Efficient Image-Based Localization Using Context

by

Charbel Azzi

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2015

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

This thesis consists of material all of which I co-authored: The work documented in Chapter 5 has been submitted to the Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference. Co-authors include: Charbel Azzi, John Zelek, Daniel Asmar and Adel Fakih. I hereby verify that I am the principal author. Also, the work documented in Chapters 2 and 4 will be submitted to Robotics and Automation Magazine, 2016 IEEE. Co-authors include: Charbel Azzi, John Zelek, Daniel Asmar and Adel Fakih. I hereby verify that I will be the principal author.

Abstract

Image-Based Localization (IBL) is the problem of computing the position and orientation of a camera with respect to a geometric representation of the scene. A fundamental building block of IBL is searching the space of a saved 3D representation of the scene for correspondences to a query image. The robustness and accuracy of the IBL approaches in the literature are not objective and quantifiable.

First, this thesis presents a detailed description and study of three different 3D modeling packages based on SFM to reconstruct a 3D map of an environment. The packages tested are VSFM, Bundler and PTAM. The objective is to assess the mapping ability of each of the techniques and choose the best one to use for reconstructing the IBL 3D map. The study results show that image matching which is the bottleneck of SFM, SLAM and IBL plays the major role in favour of VSFM. This will result in using wrong matches in building the 3D map. It is crucial for IBL to choose the software that provides the best quality of points, *i.e.* the largest number of correct 3D points. For this reason, VSFM will be chosen to reconstruct the 3D maps for IBL.

Second, this work presents a comparative study of the main approaches, namely Brute Force Matching, Tree-Based Approach, Embedded Ferns Classification, ACG Localizer, Keyframe Approach, Decision Forest, Worldwide Pose Estimation and MPEG Search Space Reduction. The objective of the comparative analysis was to first uncover the specifics of each of these techniques and thereby understand the advantages and disadvantages of each of them. The testing was performed on Dubrovnik Dataset where the localization is determined with respect to a 3D cloud map which was computed using a Structure-from-Motion approach. The study results show that the current state of the art IBL solutions still face challenges in search space reduction, feature matching, clustering, and the quality of the solution is not consistent across all query images.

Third, this work addresses the search space problem in order to solve the IBL problem. The Gist-based Search Space Reduction (GSSR), an efficient alternative to the available search space solutions, is proposed. It relies on GIST descriptors to considerably reduce search space and computational time, while at the same exceeding the state of the art in localization accuracy. Experiments on the 7 scenes datasets of Microsoft Research reveal considerable speedups for GSSR versus tree-based approaches, reaching a 4 times faster speed for the Heads dataset, and reducing the search space by an average of 92% while maintaining a better accuracy.

Acknowledgements

I would like to thank my supervisor Dr John Zelek. His insight and guidance have made my study exciting, and led this thesis to intriguing places.

I would also like to thank Dr Daniel Asmar. His continuous advice guidance and feedback lead to me to accomplish this work.

I extend many thanks to my reader Dr Steve Waslander for his valued feedback.

Special thanks to Dr Adel Fakih for his advice, support and hands-on help that he provided throughout this thesis.

To Bank Audi who funded my expenses throughout this degree I would like to thank you a lot for believing in me. I would not have accomplished this work without your full support. I will always owe you for the opportunities you have given me.

I would like to thank Ontario Centres of Excellence(OCE) and Natural Sciences and Engineering Research Council(NSERC) for funding this project and making it possible.

Finally my ultimate thanks goes to my family for the spiritual support they gave me by always motivating me to overcome the hardships I faced. Without them I would have never achieved this work.

This work is dedicated to my parents for their sacrifices, support and belief in me.

Table of Contents

List of Tables

List of Figures

Nomenclature

Symbol	Description
IBL	Image-Based Localization
GSSR	GIST-based Search Space Reduction
2D-2D	Two dimensional Space to two dimensional space matching
2D-3D	Two dimensional Space to three dimensional space matching
BF	Brute Force

Acronyms

SLAM	Simultaneous Localization And Mapping
KF	Keyframe
BA	Bundle Adjustment
RT	Re-Triangulation
AR	Augmented Reality
FLANN	Fast Library of Approximate Nearest Neighbor
ANN	Approximate Nearest Neighbor
NN	Nearest Neighbor
SFM	Structure From Motion
BF	Brute Force
PTAM	Parallel Tracking And Mapping

Chapter 1

Introduction

Image-Based localization (IBL) addresses the problem of estimating the 6 DoF camera pose in an environment, given a query image and a representation of the scene (*i.e.*, map). This is different from SLAM (Simultaneous Localization And Mapping) in that in SLAM, the camera pose is tracked while moving through the scene and is prone to drift errors, which are usually reduced by looking for loop closures to remove the errors resulting from accumulated drift. In addition, in IBL, typically the map or scene representation is not being modified while in SLAM it is. IBL does not have an initial seed location to initiate the search for the pose and shares a kinship with the kidnapped robot problem in that the pose of the camera is wide open to all possibilities. The techniques used in IBL can also be used to improve SLAM processes but that will not be discussed in this work. IBL is mainly implemented in many interesting applications specially robot localization and loop closure, place recognition, SFM and augmented reality(AR).

1.1 Problem Description

The implementation of IBL consists of three main stages:

1 Keypoints matching: keypoints can also be referred to as features and have been associated with a descriptor, which is usually a measure of the texture at a particular scale around the feature point. The descriptor is matched against other keypoints. Unfortunately, the keypoints are not invariant to illumination, blur , or viewpoint beyond certain limits. When dealing with an environment where there are thousands, sometimes millions of 3D points and their associated descriptors, the matching problem is the major challenge in IBL; which essentially is the curse of dimensionality problem.

2 Image registration: It is the process of getting the largest number of correct matches of a particular viewpoint amongst all the candidate scenes. An image has to have a sufficient number of correct matches in order to be registered and thus qualify for the pose estimation step.

3 Pose estimation: after an image is registered, the query image position and orientation within the map representation is estimated and then refined. Pose estimation is highly dependent on the quality of the matches from image registration.

There are two standard ways to solve the IBL problem:

1 2D-2D: where a set of 2D features from the query image is matched against the 2D features from the database image, which essentially defines the image pose.

2 2D-3D: where a set of 2D features from the query image is matched against the 3D points representing the scene and this set is used to optimize the camera pose.

1.2 Motivation and Objective

Building a sparse map with the least amount of points which clearly describes the scene using a single cheap camera has been evolving for several years now. The most notable and successful approaches were based on Structure From Motions (SFM) and the best known approaches are currently Bundler [55, 56, 62] and VSFM [65, 64]. The evolution of Bundler in 2008 allowed IBL to start focusing on real-time applications using a 3D map in any environment. In this thesis, three different 3D modeling packages based on SFM are tested: VSFM [63], Bundler [62] and PTAM [24]. The objective is to assess the mapping ability of each of the techniques and choose the best one to use for reconstructing the IBL 3D map. These approaches have a major shortcoming of using wrong matches in building the map. It is crucial for IBL to choose the software that provides the best quality of points, *i.e.* the largest number of correct 3D points. For this reason, this thesis will show that VSFM is the best package to reconstruct the 3D maps for IBL.

This thesis presents a comparative study of the main state of the art for IBL, namely Brute Force Matching, Tree-Based [35], Embedded Ferns Classification [12], ACG Localizer [46], Keyframe Matching Approach [16], Decision Forest [49], Worldwide Pose Estimation[28] and MPEG Search Space Reduction [19]. The objective of this comparison was to first uncover the specifics of each of these techniques and thereby understand their advantages and disadvantages. These approaches have many shortcomings in terms of accuracy and computational performance

mainly in search space reduction, clustering, feature matching and the quality of the solution is not consistent across all query images.

The focus is on reducing the search space problem as mean to solve the IBL problem. Most of the previous work focuses on reducing the search space in order to to improve the Visual Words system [46] instead of presenting a new localization system. Recently, Heisterklaus et al. [19] presented a new localization system to tackle the search space problem by using MPEG descriptors to generate artificial images to cover the space.

Sattler et al. [46] provided one of the most accurate and robust search space systems for IBL called the Visual Words approach. It aimed to accelerate the Keypoint Matching step by reducing the search space via clustering features into visual words. However the comparison study presented in this work shows that the Visual Words approach loses information due to the quantization effect. This work proposes a new IBL system. It is named Gist-based Search Space Reduction (GSSR). GSSR uses global descriptors to find candidate keyframes in the database, then matches against the 3D points that are only seen from these candidates using local descriptors stored in a 3D cloud map.

1.3 Statement of Contributions

The major contributions which this thesis presents are:

1 The main contribution is to solve the search space problem in IBL by using context and combining global and local descriptors using an SfM map. Even though GIST descriptors [58] have been used for many purposes, especially for topological localization[36, 52],

this work appears to be the first work to present a keyframe approach with global descriptors in IBL. This work is documented is Chapter 5. The material of this work has been submitted to the Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference. Co-authors include: Charbel Azzi, John Zelek, Daniel Asmar and Adel Fakih. I hereby verify that I am the principal author.

2 This work aims to provide researchers with a knowledge about this rapidly growing problem: main steps, main approaches and drawbacks. To our knowledge this work appears to be the first to present a comprehensive study on IBL. This work is documented is Chapters 2 and 4. The material of this work will be submitted to Robotics and Automation Magazine, 2016 IEEE. Co-authors include: Charbel Azzi, John Zelek, Daniel Asmar and Adel Fakih. I hereby verify that I will be the principal author.

1.4 Thesis Outline

The remainder of this report is structured as follows:

- Chapter 2 will give a brief history about IBL. It also introduces the IBL problem and inputs. Then, the IBL main stages are introduced and described in details; keypoint matching, image registration and pose estimation.
- Chapter 3 presents an introduction of the scene representation. It also describes in detail the three chosen SFM reconstruction packages for testing. The chapter also presents the results and shortcomings of the testing to choose the best package for constructing the IBL map.

- Chapter 4 is a detailed description of the start-of-the-art methods. It also presents the results from the comparison study and states and analyses the results for each of the main approaches along with the standard datasets used.

- Chapter 5 starts by describing the search space problem. Then it presents the details of the proposed system along with a brief description on GIST. It also presents the experiments, the results and discussion on the proposed technique.

- Finally, Chapter 6 summarizes and concludes this thesis.

Chapter 2

Literature Review[1]

2.1 History of IBL

In the earliest stages, IBL was cast as an image retrieval problem, which consists of estimating the location by matching a query image to a database of images.

The earliest working IBL system was developed in 2004 by Robertson et al. [41] when they represented the scene by a 2D map of manually rectified images. They rectified a query image and matched it to all the database images, then estimated the image pose. Zhang et al. [67] improved the previous system by using SIFT [30] features for matching and referring to the GPS tags. Schindler et al. [48] and Knopp et al. [26] solved the problem by efficiently matching the query image to a certain number of database images instead of matching against all of them on a

[1]The content of this chapter will be submitted to the Robotics and Automation Magazine, 2016 IEEE. Co-authors include: Charbel Azzi, John Zelek, Daniel Asmar and Adel Fakih. I hereby verify that I will be the principal author. The material will be paraphrased.

city-scale scene. These were the first large-scale IBL approaches. These approaches worked on a database consisting of tens of thousands of images. Hays et al [18], Avrithis [5] and Chen et al [9] improved IBL to deal with a database that consists of more than a million images. Hays et al. [18] incorporated a probabilistic model to estimate the query image position from a database of millions of images. Avrithis [5] projected all the image features into a global coordinate building a 2D scene map. Chen et al. [9] achieved localization by dividing the database images into sets and individually matching to each set. Zamir et al. [66] used a GPS tags to find the nearest neighbours based on SIFT descriptor matching and then estimated the camera pose.

The representation of the unknown environment can be done in two ways: either a 3D point cloud map consisting of 3D points along with their feature descriptors and their visible keyframes obtained from SfM, or a dense, featureless 3D point cloud obtained from a RGB-D technique [49, 16, 37].

Recently, some powerful SfM techniques, mainly Bundler [62, 55, 56], allowed the representation of an environment by a 3D point cloud map. This improvement allowed IBL to solve the localization by matching to a 3D map instead of matching to images. IBL relied on Bundler to represent city-scale scenes accurately and robustly with rich and dense information. The first approaches [21, 3, 61] focus on trying to get as many matches as possible between the query image and the 3D map. Irschara et al. [21] presented the first successful IBL system to match a query image to a 3D point cloud map constructed from a database of retrieved photos.

Li et al. [28, 29] and Sattler et al. [45, 46] tackled the problem of dimensionality in IBL when they tried to solve the IBL problem using city-scale datasets. Their aim was to perform the correspondence matching using only a small subset of possible matches. Their objective was

also to get the camera pose based on this subset. Recently, Shotton et al. [49] tried to solve the 2D-3D correspondence problem in small indoor scenes using a featureless 3D map. Donoser et al. [12] introduced the Embedded Random Ferns approach, where they used classifiers instead of descriptors to solve the matching problems in city-scale scenes. More recently, Heisterklaus et al. [19] tried to solve the IBL problem by reducing the search space for small-scale indoor environments. They classified the database images into multiple views and tried to find the view from where the query image was taken. Lately, Shotton et al. [16, 17] adopted the 2D-2D approach to solve the IBL problem in small-scale indoor scenes. They introduced a keyframe approach based on random ferns to try to find the closest keyframes to the query image and then performed a simple 2D-2D match without the use of any 3D map.

All of the techniques show shortcomings in their result, particularly in search space reduction, feature matching, clustering and sensitivity to where the query image is taken. At this time, the work of Sattler et al. has produced some of the bast results in IBL. The latest approaches focused on improving their work by trying to mainly reduce the search space or achieve better matching accuracy. This thesis will present a review of Sattler's et al. and the other main approaches. Then, a comparative study on each approach will be presented to benchmark the different IBL systems. The study will also reveal the shortcomings of each approach.

2.2 IBL Problem

Solving the IBL problem consists of estimating the 6 DoF camera pose of a query image in an unknown scene, given a representation of that scene.

IBL can be solved using one of two approaches, namely 2D-3D matching or 2D-2D matching. 2D-3D matching consists of matching 2D features from a query image to a 3D point cloud map of the environment. The 3D map can be extracted either using Structure from Motion (SfM) techniques [62, 55, 56, 65, 59], or via RGB-D techniques [16, 49, 37]. In the former case, the map consists of a set of 3D points, along with their corresponding feature descriptors. In the latter case, the map consists of a featureless 3D point cloud. Alternatively, 2D-2D matching consists of matching a query image to a group of keyframe images that represent the scene. Figure 2.1 illustrates the main components in IBL.

IBL solutions face several challenges that can affect their robustness, accuracy and speed. Firstly, feature matching is dependent on feature viewpoint invariance; otherwise it is susceptible to false matches due to scale, blur, and illumination changes. Secondly, IBL is also prone to the curse of dimensionality when dealing with scenes where there are thousands, sometimes millions of 3D points and their associated descriptors. In such situations, finding the exact correspondences becomes challenging. In addition, repeated patterns and structures as well as reflected surfaces can also compound the problem.

IBL relies on three main building blocks for its solution as shown in Figure 2.1: (1) Keypoint matching which involves extracting the features and descriptors, then matching them to get the correspondences (2) Image registration to remove the wrong matches returned from the previous step and to send the images with enough inliers to the next step and (3) Pose estimation where the query pose is estimated and refined.

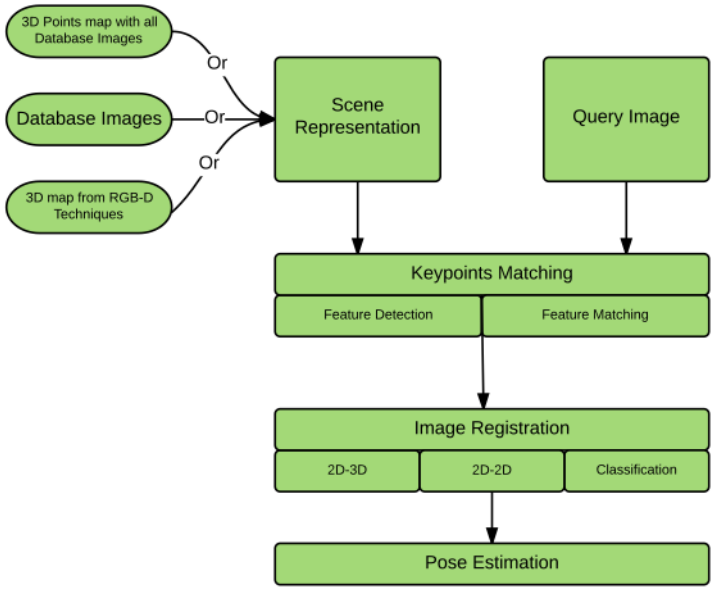


Figure 2.1: Image-Based Localization Main Components

2.3 KeyPoint Matching

The first step of IBL consists of establishing correspondences between keypoints extracted from query images with features extracted from a database keyframe. First, features are detected inside an image and then they are matched.

2.3.1 Feature Extraction

A feature is a salient point inside an image. Its robustness depends on its invariance to changes in scale and orientation. Many different types of feature detectors are available in the literature, with each one exhibiting different advantages and disadvantages. SIFT [30] has traditionally become the most popular feature detector; while it is somewhat robust to changes in viewpoint,

the computational overhead is very high. SURF [6] is another type of feature. SURF has a poor performance with changes in rotation when it is compared to SIFT which significantly influences the accuracy of the extracted feature points.The PCA-SIFT [23] proposes to reduce the computational overhead of SIFT by reducing its dimensionality; however, this comes at the cost of reduced accuracy and robustness. Affine SIFT [32] incorporated greater invariance to affine transformations, which showed more robust and accurate performances than the vanilla SIFT, but the feature extraction process takes a longer time. BRIEF [7] uses binary tests to classify trees. Their descriptor is a binary descriptor of 256 entries. It performs very poorly with changes in rotation, which restrict its use in IBL since it will affect the localization accuracy due to incorrect matches. ORB [43] is a fast binary descriptor based on BRIEF [7] and was introduced to solve the rotation problems with BRIEF. It performs better then regular BRIEF and faster then SIFT but still less accurately and less robustly than SIFT. FAST [42] uses a Harris corner filter and gives fast performance but suffers from sensitivity to orientation. This makes its use in IBL very specific to some scenes, mainly indoor scenes where the rotation effect is present. GPU-SIFT [54] is a parallel processing implementation of SIFT on GPU in real time to speed up the SIFT performance.

SIFT is most commonly used feature detector in IBL. It is mainly the most popular feature detector. Nevertheless, SIFT might not be the best accurate and robust feature detector method but this is not discussed in this thesis and is left for future work.

Once features are detected, one needs to recognize them from different viewpoints, and accordingly, each image feature is associated with a description of its local neighbourhood, which is known as an image descriptor.

2.3.2 Feature Matching

Features are matched across frames by relying on their descriptors, which encode the image appearance in the local neighbourhood of each feature. Finding the matches between descriptors is usually solved using linear search(or brute force) in IBL. The most obvious method for matching relies on brute force, where each image feature is compared to each feature descriptor in the database. Unfortunately, although effective, the process is very slow especially in large-scale scenes, where thousands or millions of 3D points have to be tested for a match.

Nearest Neighbour Search (NN) [14, 60] is the most common method used in IBL. Here, the search space is divided into subspaces of lower dimensions, thereby resulting in searching for matches in lower image scales. The disadvantage of such techniques is that the computational time grows exponentially when the size of the scene increases. To address its shortcoming known as Approximate Nearest Neighbour Search (ANN) [2, 4] attempts to find the neighbour that is most similar to the matched feature. Although it does not necessarily find the exact neighbour, it finds the most similar neighbour which in some cases might be the exact one.

To this date there are not any exact search methods that are faster then BF. All other methods like NN and ANN are optimization methods and are not considered exact. The main challenge of Image-based localization'is to find the correct matches. Thus the optimized methods used have to minimize the number of false matches while achieving fast computational times.

Fast Library of ANN (FLANN) [35] is a library of fast ANN methods to speed up the search in high dimensional spaces. These methods use either multiple randomized kd-trees or hierarchical k-means trees:

1 Kd-trees [35] iteratively split the k-dimensional set of descriptors space. The data is split at each tree level at the value where the data scores the largest variance. This results in a hierarchy of splits called search trees. It tries to find the nearest neighbours by traversing all the leaves in the tree. Increasing the search space dimension requires a large amount of time to search all the leaves. An alternative more efficient approach proposed by [53] consists of visiting a reduced number of leaf nodes to find the approximate nearest neighbour. In the method known as the Forest of Randomized kd-trees [34], the split is chosen randomly among the ones featuring the largest variance. It increases the efficiency and gives similar accuracy to the kd-trees in approximating the nearest neighbour but at the cost of high memory requirements. The kd-trees method is the most common approach used in IBL [45] for matching. It presents the best compromise between accuracy and computational time. Nevertheless, in larger environments where there are thousands or millions of descriptors, the matching speed of kd-trees decreases.

2 Hierarchical k-means trees [34] also known as a vocabulary trees [38] use k-means clustering to iteratively split the descriptors group into k clusters. The clustering stops when each cluster contains less then k descriptors. The approach traverses the tree to find the approximate nearest neighbour corresponding to the closest cluster centre in each node. Due to its efficiency, hierarchical k-means is commonly used in IBL for matching descriptors. Nevertheless, it is subjected to miss correct correspondences due to quantization effects where a feature can get assigned to a cluster that does not contain many of its matches.

As an alternative to matching features, another approach is based on learning classes of features. Here, descriptors corresponding to the same feature are used to train a machine-learning

algorithm; then, any new feature is classified based on the attributed class of its descriptor.

Donoser et al. [12] presented an alternative to the Approximate Nearest neighbour (ANN) by introducing a discriminative classification step called embedded random ferns. Their goal was to improve the feature matching by considering previous sightings of a specific feature as a class. This system scored a higher number of matches than tree-based but the quality of these matches remained questionable. The major shortcoming is the weakness of the classifier in global matching and its reliance on GPS tags to partition the search space into smaller regions. This approach will be further explained and discussed in Section 4.

2.4 Image Registration

The second building block of IBL is image registration, which is performed via 2D-3D correspondence, 2D-2D image matching, or classification.

2.4.1 2D-3D

The image registration step aims to find the correct matches (inliers) between a query image and a database of 3D points. . The standard approach to find the set of 2D-3D correspondences is by utilizing the tree-based approach(based on FLANN). To clarify this point, a number of features that are matched using the keypoint matching technique are not correct and using them would result in inaccurate localization. Therefore, it is necessary to follow the keypoint-matching step by image registration to help reduce the number of erroneous matches. To help remove false matches, one must perform what is known as the ratio test [30], where a match is only accepted

if the similarity between the distances to the first and second nearest neighbour is less then a certain threshold. Then RANSAC [40] is applied to remove the remaining outliers. RANSAC iteratively selects a random subset of all the matches, then uses this subset to estimate the camera transformation, and verifies the estimated transformation against all other matches. If the number of inliers after RANSAC applied is higher then 12 then the image can be considered registered and subsequently qualifies for pose estimation. Otherwise, it is discarded. The threshold of 12 inliers proposed by [29] is chosen to be high enough to make it unlikely for a false candidate to have this many inliers and low enough for true candidates with a low number of features not to be rejected.

The Tree-based approach (FLANN) is so slow when dealing with medium to large scale environments because it performs the matching against the whole search space. During the past years, several improvements to image registration were proposed. Li et al. [28] presented a visibility graph(P2F), which sorts the 3D points in a map in terms of their visibility from the camera viewpoint corresponding to the different photos that were used to construct the map. Then, they used 3D-2D to guarantee that a sufficient number of inliers is found. Their algorithm stops after 100 correspondences are found. Then RANSAC is used to remove the outliers. Their localization results outperformed the tree-based approach in terms of speed but were less accurate. Li et al. [29] further improved the image registration by introducing co-occurrence RANSAC. This consists of a probabilistic model that uses a visibility model [29] to choose the highest set of 2D-3D matches that tends to co-occur using RANSAC. Then, they used 3D-2D checking to guarantee that a sufficient number of inliers were found. This approach slowed their previous approach but resulted in higher accuracy.

Sattler et al. [45] introduced a Vocabulary-Based Prioritized System (VPS) to improve im-

age registration. They clustered 3D points into bag-of-words and then sorted them based on a priority cost before matching them through a tree-based approach, stopping after one hundred matches were found. Their system is considerably faster then tree-based, P2F, and Co-occurrence RANSAC, but not as accurate as tree-based. Sattler et al. [46] improved their VPS system for registration by introducing active search, where the surroundings of a 2D-3D match are searched to find its nearest neighbours., This is followed by a 3D-2D matching to recover the matches from their VPS. Their system was faster then all the other approaches but lost some accuracy to VPS. All these registration approaches tackled city-scale scenes.

Shotton et al. [49] presented a different registration approach. They used a regression random forest method to train a featureless 3D map, reconstructed from RGB-D techniques. They matched the 2D points of a query image to their trained map to get the 2D-3D correspondences. Their method tackled small-scale environments and was faster, although less accurate than tree-based. Their work was not tested on city-scale sets and therefore cannot be easily compared to the previously mentioned registration approaches.

2.4.2 2D-2D

Although 2D-3D is more accurate and faster than 2D-2D in re-localization applications, image registration is performed using 2D-2D techniques to save computational time needed for fast localization. The scene is represented by a database of keyframes that cover that environment. 2D-2D image registration consists of assigning to each query image the corresponding keyframe that is most similar. Then, the set of 2D-2D correspondences between the matched images is computed. Similar to the 2D-3D matching, these matches are subjected to the ratio test [30] and

RANSAC [40] in order to remove the outliers. Again, if a sufficient number of inliers are found, the image is considered registered and the image qualifies for pose estimation.

In the work of Shotton et al. [16], an efficient encoding of each keyframe is performed by training their internal 2D pixels using random ferns and then matching each keyframe to the query image. The keyframes with the smallest distance to the query are then used to perform a 2D-2D match and thereby guarantee robust image registration. Their system performed better (in terms of accuracy and speed) at re-localization when a query image was close to a keyframe.

2.4.3 Classification

In addition to the more common 2D-3D, and 2D-2D image registration techniques, classification techniques can also be used to improve image registration. It is notable that although classification can be used as an alternative to the above techniques, it can also be used with 2D-3D or 2D-2D to further improve the registration.

In classification techniques, Heisterklaus et al. [19] images are binned into multiple views using global descriptors. Then synthetic camera poses are created to cover all the remaining spaces in the environment; this is needed to ensure more robust correspondences in less time. Their system showed promising registration improvements.

2.5 Pose Estimation

Any image in which a sufficient number of features are matched and subsequently image registration is successful can be used for estimating the camera pose (translation and rotation). Cal-

culating the camera pose of a camera depends on the underlying matching that was performed (i.e., 2D-3D or 2D-2D).

In the case of 2D-3D matching, the n-point perspective (pnp) method is used. It starts by computing the location of the 3D map points in the local frame of the camera. Then, the camera pose is estimated by calculating a rigid geometric transformation between the position of the points in the local frame and their position in the global frame. The computation of the transformation between local and global frames requires information about the intrinsic camera parameters. This information can be either known or unknown. In the case of known intrinsic parameters (mainly focal length and distortions), the rigid transformation is estimated from three 2D-3D matches and is referred to as three-point perspective pose problem (p3p). P3P defines the pose by aligning local and global point positions and yields up to 4 solutions. Kneip et al. [25] presents a very efficient solution to IBL using p3p. The six-point perspective pose problem (p6p) is widely used to compute the transformation in the case of unknown intrinsic parameters. It computes a full projection camera matrix including the focal length estimation from six 2D-3D matches and yields a single solution. Recently, Sattler et al. [47] presented a new method to estimate the transformation by using p3p while sampling the focal length. They achieved same accuracy as standard PnP with much faster speed. It is used to avoid evaluating all the solutions returned from PnP. Iterative pnp can optionally be used to refine the estimated camera pose.

In the case of the 2D-2D correspondence case, the camera pose is estimated using the fundamental matrix of the camera. The method consists of estimating the camera fundamental matrix from a set of points using the epipolar geometry constraint between two cameras in the case of unknown intrinsic parameters. In the case of a calibrated camera (known intrinsic parameters), the 5-point algorithm estimates the projection matrix, here called essential matrix, from at least

5 2D-2D matches. The pose is usually then refined by using non linear error minimization techniques. In IBL, Levenberg-Marquardt (LM) [15], Gauss-Newton [44] and M-estimators [20] are the refinement approaches commonly used.

2.6 Summary

This chapter presented a literature review about IBL; The history of IBL was presented. Then the IBL problem was described and its three main stages (Keypoints matching, image registration and pose estimation) were presented. Each stage was fully described and the main works done in each stage were presented. These main approaches appear to have many shortcomings in terms of accuracy and computational performance mainly in search space reduction, clustering, feature matching and the quality of the solution is not consistent across all query images. The main approaches will be studied and the main shortcomings of each will be revealed in Chapter 4 to prove that IBL problem is not yet solved. In the next chapter, the focus will be on choosing the best software to reconstruct the 3D map of the environment.

Chapter 3

Creating The Scene Representation

3.1 Introduction

In this modern era, cameras are found everywhere. They are relatively cheap, light, and produce high-resolution images. These factors, along with the advances in Computer Vision, make a camera the sensor of choice for producing 3D models of any environment. Applications range from aerial mapping to mapping of indoor and outdoor land scenes, to mapping of underwater environments. Both filter-based techniques (Visual Simultaneous Localization and Mapping or Visual SLAM) [57, 11] and non-filtering methods (e.g., Structure From Motion or SFM) [31, 33, 24, 65] produce maps by concurrently localizing the position of the camera in the map. While all types of localization and mapping produce maps during localization, the quality of the maps differs based on the specifics of each implementation. The majority of real time localization applications tend to build the maps on the fly. This poses limitations that are still unsolved using

a monocular camera. There are many notable limitations. First, obtaining an accurate real world scale that is crucial for many applications like Augmented Reality(AR) is very hard with the use of a single monocular camera. Second, the computational time and memory requirements related to the platform that these applications are run on limit the algorithm use and can pose some constraints on the map and the environment size. For instance, PTAM [24] was mainly built for indoor AR applications and PTAMM [8] was built for mobile platforms with limitations on the size of the maps. The memory of the phone can deal with sparse maps with a limited number of features. Third, the built map quality is affected by the time and memory complexity which affects the robustness of the tracking. There are lots of applications that do not require the maps to be built on the fly. Having a good sparse 3D map which is used only for tracking can solve the above limitations concerning the IBL problem.

Building a sparse map with the least amount of points which clearly describes the scene using a single cheap camera has been evolving for several years now. The most notable, successful approaches were based on SFM and the best known approaches are currently Bundler [55, 56, 62] and VSFM [65, 64]. The evolution of Bundler in 2008 allowed IBL to start focusing on real-time applications using a 3D map in any environment. In this chapter, three different 3D modeling packages based on SFM are tested: VSFM [63], Bundler [62] and PTAM [24]. The objective is to assess the mapping ability of each of the techniques and choose the best one to use for reconstructing the IBL 3D map.

3.2 Packages Description

In this section we describe the basic functionality steps behind each of the packages. We will present the main methods used to do the 3D reconstruction so that we can differentiate between each package and understand the reasons behind the different results that will be shown later.

3.2.1 Bundler

The latest version of Bundler was released in 2010. This software aims to demonstrate the success of SFM techniques on unorganized images sets that may be found on the Web. The package uses a robust modified SFM approach to reconstruct 3D scenes out of these unordered images. The main methods upon which Bundler works are described below [55]:

A) Feature matching: Bundler uses the SIFT feature detector. Then, each pair of images is matched to get the Fundamental matrix. Here Bundler runs its own optimization to get a robust Fundamental matrix(F-matrix) using RANSAC as follows:

   1) Compute a candidate F-Matrix for each RANSAC iteration using the eight point algorithm.

   2) Run non-linear refinement of it.

   3) Remove the outlier matches and get the recovered F-matrix.

   4) Check if the number of remaining matches is less then 20, then remove all the matches.

B) Modified SFM: Bundler organizes the matches into a connected set of matched keypoints across multiple images called a 'track'. The modified SFM approach is summarized below:

1) Bundler initializes cameras using pose estimation to avoid getting stuck in bad local minima. This is done by adding multiple cameras at a time.

2) Bundler uses different approaches to choose the initial 2 images. It chooses the pair of images that have the largest number of matched features and then estimates the camera parameters of this pair.

3) Bundler starts to add multiple cameras to the optimization. It begins by adding the camera with the greatest number of matches (whose 3D position has been already estimated) then follows that by adding any camera that has 0.75 of the total number of matches.

4) For each added camera, Bundler initialize the extrinsic and intrinsic parameters using Direct Linear Transformation (DLT). It also reads EXIF tags of the image where they take the focal length and compare it to the estimated one from DLT to initialize it.

5) For each added camera, Bundler adds tracks observed by that camera. Each track is added if it was observed by at least one recovered camera.

6) Bundler uses sparse BA to minimize the reprojection error at each iteration. After every run of the optimization, Bundler detects 3D outlier points that have high reprojection error in a track and then removes that track. Then optimization is rerun again until no outliers remain.

3.2.2 VisualSFM

The latest version of VSFM was released in 2013. The main target of this package was to reach a linear time incremental SFM. Wu in his paper [64] explains the major improvements that his

software presented to build a better 3D reconstruction. Basically, VSFM improved the SFM algorithm done in Bundler. The main methods upon which this software work are described below:

1) VSFM uses a new feature matching approach called "Preemptive Feature Matching". This method consists of: a) Sorting the SIFT features of each image in a decreasing scale order, b) Generating the frame pairs to be either fully matched or by taking a subset of images to be matched, c) Looping over each image pair by first choosing a number h corresponding to the first h features to be matched. Then, it is essential to check if the number of matches is less than a certain threshold, then repeat the matching, and if it is not then do regular matching and geometry estimations.

2) VSFM uses multicore bundle adjustment [65] where the aim of BA is to refine the 3D position of features and camera parameters by minimizing the non-linear reprojection error function. This method uses implicit multiplication of the known Hessian matrices and Schur complements by the use of the Jacobian matrix. In this way, the function is linearized at each iteration.

3) VSFM uses a modified version of incremental SFM from Bundler. First, the software starts to do full BA only when the size of a model is increased by a certain threshold ratio due to the large amounts of cameras being added. But to reduce the error accumulation, VSFM always runs local partial/local BA on a certain number of recent frames. Second, point filtering is done on the 3D points that had large reprojection errors.

4) The last part of incremental SFM is Retriangulation (RT) where VSFM retriangulate the

failed feature matches. This is done to recover these points and to have more features in the scene. Here the reprojection error threshold is increased to reach that goal. Then a full BA and point filtering is run again to improve the reconstruction and reduce the errors.

Thus, VSFM is characterized by the preemptive feature matching, multicore BA and incremental SFM that uses a mix of BAs and RT to maintain the accuracy of the 3D reconstruction.

3.2.3 PTAM

PTAM was released in 2007. Many updates were made on it and the latest was PTAMM (parallel tracking and multiple mapping) in 2011. The main approach is described in [24] and [8]. It splits the tracking and mapping into 2 parallel threads. The main approach is described below:

A) Mapping is the main part that we are interested in when dealing with 3D scene reconstruction. The main difference from other SFM softwares is that they only update the map based on a keyframe. In other words, they have their own way to add frames to the mapping thread. The main algorithm works as follows:

1) PTAM uses stereo initialization: They take the first 2 keyframes, run FAST-10 features extraction, match the 2 images to get the F-matrix, use RANSAC to remove outliers, recalculate F-matrix using inliers, optimize for getting the correct Essential matrix and finally triangulate the 3D points.

2) Adding Keyframes: PTAM only adds a frame, which becomes a keyframe, to continue constructing the map based on satisfying the following 3 conditions: a) if tracking quality

is good, b) if a minimum of 20 frames has passed since the last keyframe was added and c) if the camera has moved a minimum distance from the last keyframe pose.

3) 3D points are added to the map by feature matching along epipolar lines between the latest keyframe added to the map and its closest keyframe in terms of camera position using triangulation.

4) BA: LM algorithm is used to refine the camera positions and 3D triangulated points. When a new keyframe is added, BA is interrupted.

B) Tracking is run parallel to the mapping thread. The main steps are summarized below:

1) At each frame grabbed by the camera, the image is converted to grayscale. Then a 4 level pyramid is created for each frame. Fast-10 feature detector is run at each level.

2) A predicted camera pose is estimated using a decaying velocity model.

3) 3D map points are projected into the image according to the frame predicted pose in point 2 using a calibrated pin hole camera model.

4) 50 3D points are projected at coarse level into the image plane and searched for. Given successful patch matches between the new image and its closest keyframe, the camera pose is updated by minimizing an objective function that accounts for the reprojection error. Then 1000 points are projected at a fine level and the same procedure is repeated to refine the updated pose.

3.3 Results

3.3.1 Datasets and Testing Methodology

Three datasets composed of hundreds of images taken by our camera (point grey as raw data, logitech as png). Each dataset is described below:

1. Jbeil Roman Theatre:

   Images of a very old Roman Theatre located in Byblos-Jbeil were taken by a non-calibrated Logitech camera. The images were taken as Keyframes inside PTAM where 31 keyframes were stored for testing for this scene as input images to other packages to ensure having a fair comparison.

2. University of Waterloo Robotics Group Lab:

   2000 Images for a well dense lab located at the University of Waterloo were taken as raw data from an uncalibrated Point Grey Camera. The best 400 frames out of the 2000 were chosen to reconstruct this scene. The aim was to show the camera effect of the reconstruction by taking the images as raw uncompressed data.

3. University of Waterloo Engineering 5 Building:

   400 Images for a symmetrical shaped building were taken as raw data from our Point Grey uncalibrated camera to be the input images to the packages. This building was chosen due to its symmetrical shape effect on the reconstruction along with its reflection effects caused by the glass

The three packages were used as blackboxes where some parameters were tuned to have a fair comparison. In VSFM, we changed the maximum feature matching number, and it was tuned so that more features can be matched. The minimum and maximum reprojection error thresholds were tuned by trial and error and we set the best ones equally for the three packages. In Bundler, the camera initialization thresholds were tuned. In PTAM, the number of minimum keyframes was lowered along with the minimum distance threshold from the camera to get more keyframes, since the goal is to tackle the mapping thread of PTAM.

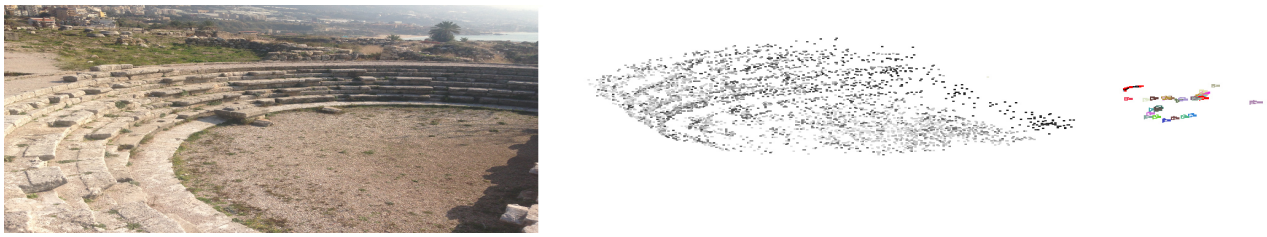
3.3.2 Results

This section will present the reconstructed 3D maps scenes for each dataset used along with their corresponding reprojection errors.

Roman Theater Dataset The reconstructed 3D sparse maps for Jbeil Roman theater are shown in Fig. 3.1. Fig. 3.1a and 3.1b show a side and a front view of the resultant sparse map of the theater from VSFM respectively. Fig 3.1c and 3.1d also show a side and a front view for the map reconstructed from PTAM respectively. Fig. 3.1e show the Bundler reconstructed map. It is clear that the reconstruction from VSFM and PTAM was good and it clearly shows the layout of the theater, with a fair advantage to VSFM where it shows a more robust structure of the monument. In contrast, the Bundler reconstruction was poor and did not even reflect the theater aspect.

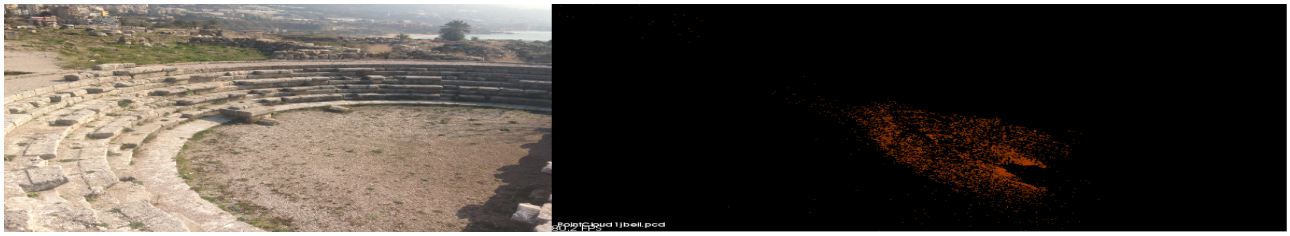
The reprojection error in pixels associated with the reconstructed maps from Bundler and VSFM are shown in Fig. 3.2. Fig. 3.2 shows that the error coming from VSFM is a bit higher than the one coming from Bundler. The total average reprojection error for VSFM was 3.09

whereas it scored 2.47 for Bundler. The main reason why Bundler had a lower error than VSFM is simply because of the number of 3D features. The sparse map from VSFM had 8661 features whereas PTAM's map had 7100 and Bundler's map had only 3246. So Bundler's reprojection error was lower than the VSFM reprojection error because it was computed on 2.8 times less number of features. The number of features also reflects the fact that VSFM and PTAM returned good maps with a fair advantage to VSFM's map.



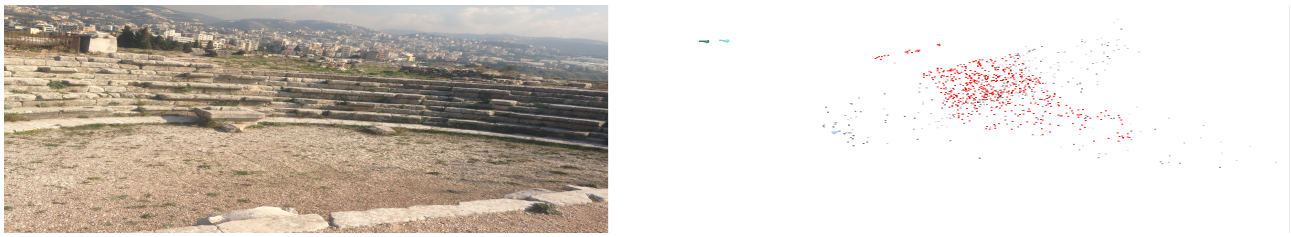
(a) Jbeil map from VSFM side view

(b) Jbeil map from VSFM front view



(c) Jbeil map from PTAM side view

(d) Jbeil map from PTAM front view



(e) Jbeil map from Bundler

Figure 3.1: Reconstructed 3D sparse maps for Jbeil Roman Theatre

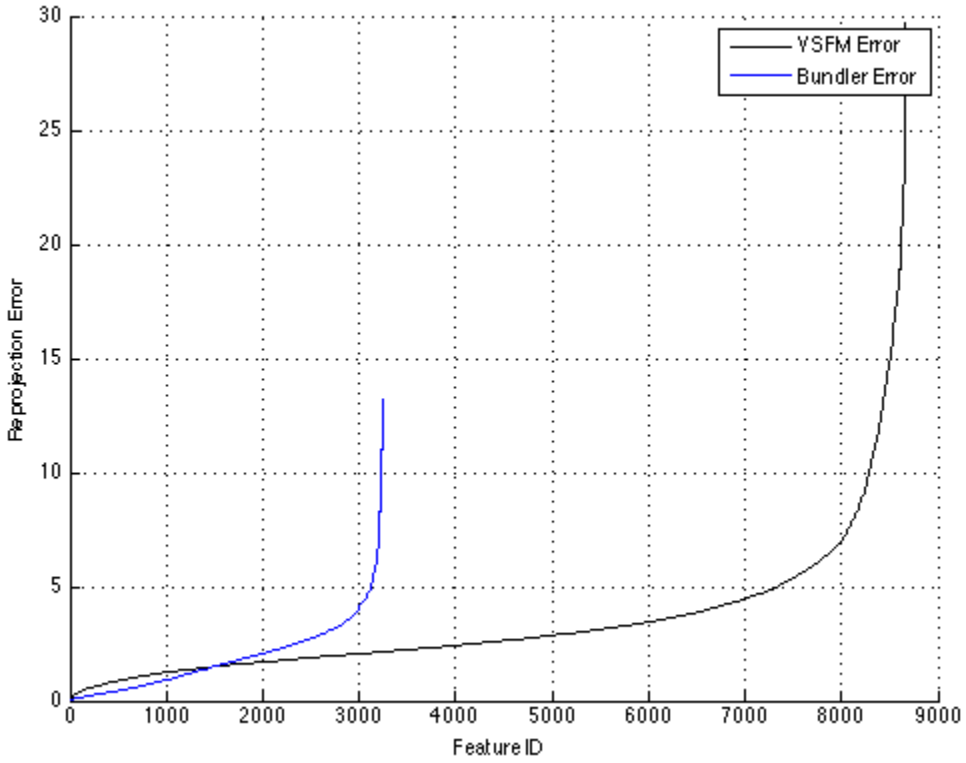
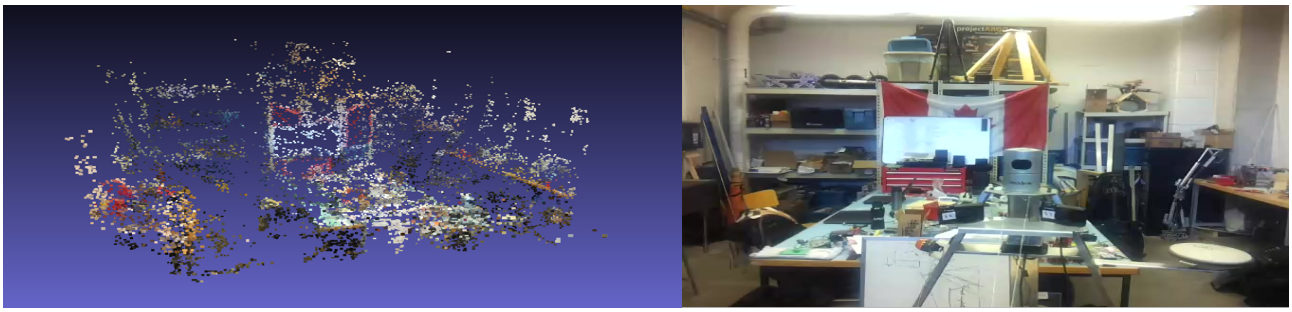


Figure 3.2: Reprojection error Vs feature ID associated with the reconstructed maps for Jbeil Roman Theatre

University of Waterloo Robotics Group Lab Dataset The reconstructed 3D sparse maps for the University of Waterloo Robotics Lab are shown in Fig. 3.3. Fig. 3.3a shows the resultant sparse map of the lab from VSFM. Fig. 3.3b shows the Bundler reconstructed map. This time, Bundler showed an acceptable reconstruction where the lab can be identified. But compared to VSFM the Bundler results become marginally bad since VSFM showed again a really good 3D sparse map. In Fig. 3.3a we can clearly see that VSFM maintained the lab structure and main components and a person can clearly identify some of the lab components such as the Canadian



(a) UW Robotics Lab map from VSFM

(b) UW Robotics Lab map from Bundler

Figure 3.3: Reconstructed 3D sparse maps for UW Robotics Lab

flag.

The reprojection error in pixels associated with the reconstructed maps from Bundler and VSFM are shown in Fig. 3.4. The figure shows that the error coming from VSFM is lower than the one coming from Bundler. The total average reprojection error for VSFM was 1.35 whereas it scored 3.01 for Bundler. The sparse map from VSFM had 12881 features whereas Bundler's map had 8962. Those numbers reflect the sparsity of the map and the fact that VSFM again returned the best map. Alternatively, Bundler this time was able to get an acceptable number of features to build its map.

University of Waterloo Engineering 5 Building Dataset The reconstructed 3D sparse maps for the University of Waterloo's Engineering 5 Building are shown in Fig. 3.5. Fig. 3.5a shows the resultant sparse map of the lab from VSFM. Fig. 3.5b shows the Bundler reconstructed map. Again VSFM showed a good reconstruction where the E5 building structure was clearly visible. Although the map was visually good, we must note that it was returned in 2D plane and not in 3D plane. This is due to the symmetrical shape and the lighting reflections coming from the glass of that building. In contrast, the Bundler reconstruction was pretty poor and did not even reflect

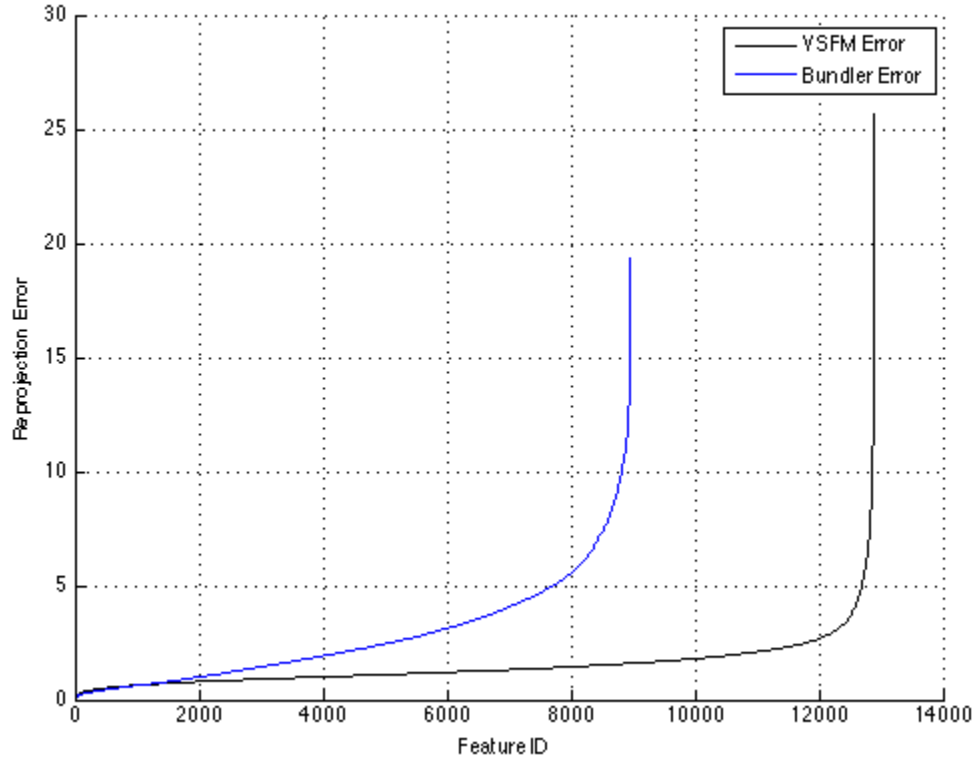
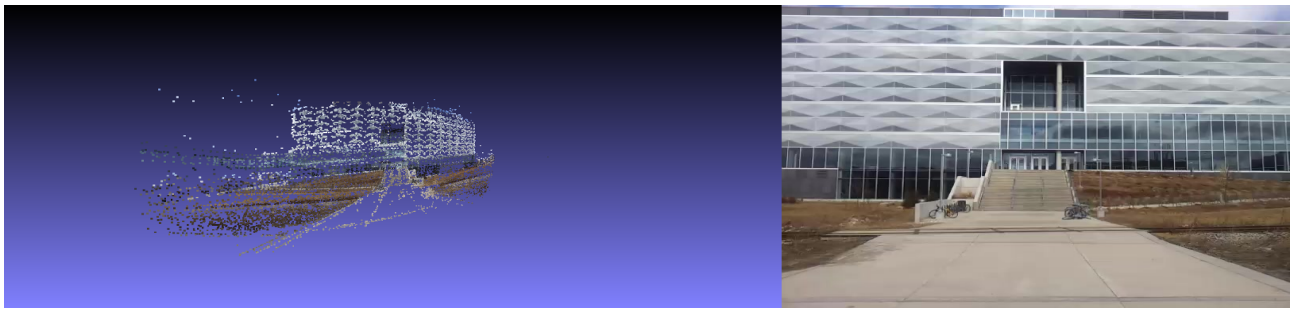


Figure 3.4: Reprojection error Vs feature ID associated with the reconstructed maps for UW Robotics Lab

the building aspect.

The reprojection error in pixels associated with the reconstructed maps from Bundler and VSFM are shown in Fig. 3.6. The figure shows that the error coming from VSFM is more then 2 times higher than the one coming from Bundler. The total average reprojection error for VSFM was 13.24 whereas it scored 5.44 for Bundler. Those numbers show that the VSFM reconstructed map is indeed a decent one, but there is something wrong in the 2D reprojections. The sparse map from VSFM has 16478 features, while Bundler's map had only 2994. So Bundler's reprojection error was lower than the VSFM reprojection error because it was computed on 5.5 times less number of features. The number of features also demonstrates the fact that VSFM is again the best package.



(a) UW E5 Building map from VSFM

(b) UW E5 Building map from Bundler

Figure 3.5: Reconstructed 3D sparse maps for UW Engineering 5 (E5) Building

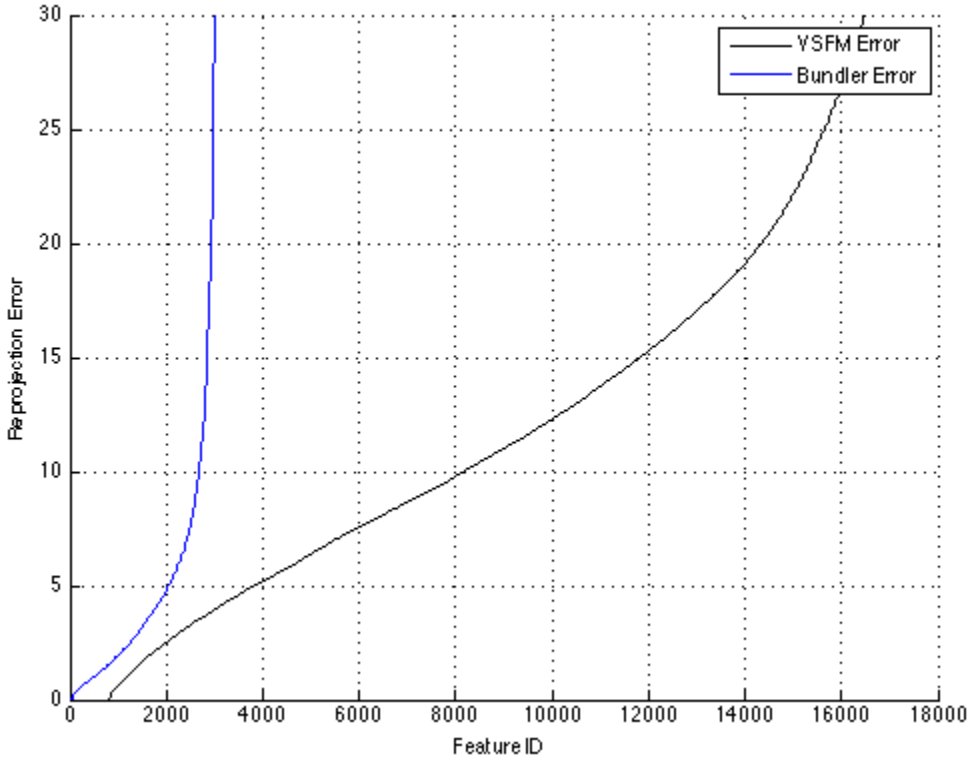


Figure 3.6: Reprojection error Vs feature ID associated with the reconstructed maps for UW E5 building

3.4 Analysis and Discussion

In this section, the analysis of the results is discussed and the shortcomings of each tested package are revealed, along with their effect on IBL.

First, VSFM''s robust and accurate mixing of Re-triangulation (RT) and BA is a major reason it produced the best sparse maps. Reconstruction shows that the RT step is handling the drifting errors where some features marked as outliers were re-optimized using RT. a Many of them were found to be inliers and were refined using BA. This gave VSFM a main advantage over PTAM and Bundler since t PTAM and Bundler only run optimization and do not re-triangulate the outliers.

Second, it was noticed that VSFM's multicore BA gave their optimization of the camera parameters and features positions more accuracy over Bundler and PTAM. Bundler used only regular global BA when they had to add multiple cameras to the optimization to save time. - Thus, their BA estimations returned lots of outliers. PTAM, on the other hand, used a mixed global and local BA which gave their maps a large number of inliers and helped to reduce the accumulation error.

Third, image matching which is the bottleneck of SFM, SLAM and IBL plays the major role in favour of VSFM. VSFM's preemptive feature matching approach gave VSFM a high number of robust matches relative to PTAM and Bundler. This method offers a very low computational cost and allows one to focus efforts on the features that are most likely to be matched. This, combined with the mixing of RT and BA, allowed a large number of inliers, and so a large number of features in the reconstructed map. This is a major necessity for IBL since good quality matches in the map lead to more inliers and better localization accuracy. Bundler''s major

problem causing poor results was its own image matching. It is summarized by the numbers of cameras registered for the reconstruction. A very large number of cameras failed to initialize, leaving the reconstruction with a small amount of registered cameras (for example only 328 images out of the 400 were registered for the E5 Building dataset). This affected the number of features, thus the complete reconstruction. It is mainly caused by two major things. First, the images could not be matched because they belong to a part of the scene which might be disconnected from each other. In addition, there was excessive blur and noise, and little overlap with other images. This results in a very few number of matches, where Bundler set a threshold of 20 remaining matches in order to register the camera. Second, Bundler uses a prior pose estimation to initialize the camera to avoid getting stuck in local minima. W This failed in l many cases and marked the images as bad ones, and therefore did not initialize them. The poor quality of matches that Bumdler returns will give less localization accuracy.

Fourth, Bundler used a camera model that does not handle the lens distortion, which causes large reconstruction errors. This is why in Bundler's maps there exist a lot of reconstructed features that should be marked as outliers.

Fifth, it was noticed that PTAM sometimes inserted wrong features into the map with high errors. This happened when tracking quality was poor.

Finally, some chosen environments might be challenging to reconstruct, especially those who have symmetrical or repeating structures or do not have strong features like the E5 Building dataset. Even VSFM had problems reconstructing such scenes.

To summarize, all of these main algorithmic differences explained above are tabulated in Table 3.1. It is crucial for IBL to choose the software that provides the best quality of points, i.e.

		VSFM	PTAM	Bundler
Image Matching	SIFT	✓	×	✓
	FAST	×	✓	×
	Preemptive Matching	✓	×	×
	Prior Pose Estimation for Initialization	×	×	✓
SFM	RT and BA mix	✓	×	×
	Global BA	✓	✓	✓
	Local BA	✓	✓	×
	Multicore BA	✓	×	×
	Motion Model Handle Lens Distortion	✓	✓	×
	Affected by Tracking	×	✓	×

Table 3.1: Main algorithmic differences for each package.

the largest number of correct 3D points. For this reason, VSFM was chosen to reconstruct the 3D maps for IBL.

Chapter 4

IBL State of the Art Evaluation[1]

In this chapter the state of the art in IBL techniques will be described. Also, the different datasets used in IBL will be discussed. The methodology that was followed to ensure a fair comparison is also presented. Then, the results of the study on each of the available approaches will be presented. This will be followed with a comparison utilizing the results taken from the papers describing closed-source systems. Finally, the results will be analyzed and discussed by presenting the shortcomings and strengths of each approach.

4.1 Main Approaches Description

In this section the state of the art in IBL techniques are presented. Table 4.1 lists six different systems taken from the literature, along with the advantages and disadvantages of each system.

[1]The contents of this chapter will be submitted to the Robotics and Automation Magazine, 2016 IEEE. Co-authors include: Charbel Azzi, John Zelek, Daniel Asmar and Adel Fakih. I hereby verify that I will be the principal author. The material will be paraphrased.

Table 4.1: This table shows the area of contributions for each major approach in the main steps of IBL and its scalability application

		Main Steps in Image-Based Localization			Scale Applicability	
		Keypoint Matching	Registration	Pose Estimation (Localization)	City-Scale	Small-Scale
Decision Forest - Shotton	Shotton's Learning Approach	✓	×	×	×	✓
	Modified RANSAC	×	×	✓	×	✓
Keyframe Approach - Shotton	Shotton's Relocalization Approach	×	×	✓	×	✓
Visual Words - Sattler	Visual Words Vocabulary Tree	✓	×	×	✓	✓
	Active Search	×	✓	×	✓	✓
Worlwide Pose Estimation - Snavely	Co-occurence RANSAC	×	✓	×	✓	×
	Bi-directional Search	×	✓	×	✓	×
Embedded Ferns - Donoser	Classification Approach Using Ferns	✓	×	✓	×	✓
MPEG Search Space Reduction - Heisterklaus	Synthetic Camera Generation	✓	×	×	✓	×

IBL techniques are typically designed for two different types of scale; namely, (1) small indoor scales, which consist of hundreds of images and result in tens of thousands of 3D points and (2) a large city-scale, in which the representative database contains thousands of images and millions of 3D points. Some of these selected techniques are open-source in nature and they are the most commonly referred to. The list includes the decision forest [49], the Keyframe approach [16], Visual Words [46], Worldwide Pose Estimation [28], Embedded Ferns [12] and MPEG search space reduction [19].

4.1.1 Decision Forest

Shotton et al. [50] presented an IBL system that focused on improving the Keypoints matching and pose estimation steps. Their main contributions were: (1) their regression forest to represent the scene and (2) their modified RANSAC for pose estimation.

Their main technique is illustrated in Figure 4.1 They used an RBD-D sensor to get both RGB images and their corresponding depth. The images have known poses computed from the RGB-D technique [49, 16, 37]. They determined the pixel location of each image and used this position to train a regression forest [1, 27, 51]. The depth and camera poses are used to compute the 3D scene coordinates by training the forest at every image pixel. The forest is mainly used to generate a mathematical representation of the scene from the input database images from which the output will be the 3D position of each pixel point. This will result in a featureless 3D map. The pixels of a query image are matched to the 3D map points through the regression forest. Then a modified version of Preemptive RANSAC based on energy minimization is used to remove the outliers. If more then 12 inliers are found, the image is registered and its pose is estimated and optimized.

4.1.2 Keyframe Approach

Shotton et al. [16] presented a keyframe approach for re-localization application. Their main idea focuses on finding the closest keyframe to the query image to correct the current pose estimation using a new simplified random ferns [39] approach.

Their database consists of RGB-D keyframes of indoor scenes. They start by dividing each

keyframe in the database into *m* equal pixel locations. Thus each image is divided into m ferns and each fern is divided into 4 nodes to represent the intensity of each pixel (RGB and Depth). They encode each fern in the image by doing a simple binary test. A query image is taken and divided to m equal locations and compared respectively to all the keyframes in the database using the trained ferns. Then, the block hamming distance (BlockHD) between the query image and each keyframe is computed from the resulting binary test. The hamming distance is a number that denotes the difference between two binary blocks and the blockHD counts the number of differing blocks. The closest keyframes with the smallest distance to the image are chosen and matched via standard FLANN in order to obtain the 2D-2D correspondences. If more then 12 inliers are found after RANSAC, than the image is registered and its pose is estimated.

4.1.3 ACG Localizer

Sattler et al. [46, 45] presented a complete IBL system that aimed to reduce the search space problem when dealing with large city-scale environments. They improved the keypoint matching step by their Vocabulary Prioritized Search (VPS) [45] algorithm, described in the previous section, and their active search method improved the image registration step. They used a 3D point cloud map where they presented their VPS, illustrated in Figure 4.2, to cluster the 3D points into bag-of-words and form a vocabulary tree. Each point is stored by the mean of all its SIFT descriptors [30]. The tree is sorted based on a priorities strategy that takes into account the co-visibility of each 3D point in the database image. Then, they start their active search algorithm, illustrated in Figure 4.3 by performing a standard FLANN 2D-3D matching between the query image descriptors and the 3D point in the vocabulary tree until one hundred matches are found.

There is a high probability that the nearest neighbours of a matched 3D point will have matches in the query image. Thus, the neighbours of each matched 3D point undergo a 3D-2D matching with 2D features in the query image. The outliers are then removed via RANSAC and if more then 12 inliers are found, the image is registered.

4.1.4 Worldwide Pose Estimation

Snavely et al. [28] address the image registration step in IBL and propose an improvement based on their co-occurrence RANSAC and bi-directional contributions. They worked on a city-scale mapping of the environments. Their work consists of performing a standard FLANN 2D-3D between the query features and all the 3D maps until one hundred matches are found. Co-occurrence RANSAC is then applied on those matched. It consists of dividing the resultant matches into subsets. Then they use a probabilistic model to return the subsets with the highest probability. These subsets are the starting set that RANSAC will begin with to remove the outliers. If more then 12 inliers are found, then the query is registered and qualifies for pose estimation. Otherwise, the matches undergo a bi-directional search, which consists of performing a 3D-2D matching to guarantee that a sufficient number of inliers is found.

4.1.5 Embedded Ferns

Donoser et al. [12] presented a new keypoints matching technique that can be used for IBL. Their main contribution was presenting a new classification technique called embedded ferns.

They followed the principle of 2D-3D approach by [46] but they replaced the standard

FLANN to get the 2D-3D by a discriminative classification step called embedded random ferns. They basically used all the descriptors representing each 3D point to train a classifier, then used random ferns based on the projections. They only stored the classifier and removed the images and descriptors to save memory. They required a GPS prior on camera position to restrict the classification into certain areas of the scene from which the query image is taken.

4.1.6 MPEG Search Space Reduction

Heisterklaus et al. [19] improved the keypoints matching and image registration steps. Their main idea is to reduce the search space in large city-scale environments by presenting synthetic views to cover the space for faster 2D-3D matching.

For each keyframe in the database, they extract the MPEG Compact Descriptors for Visual Search descriptor (CDVS) [13]. Each one consists of a global descriptor and compressed local descriptors. A CDVS test model is used to generate a compact model of the real and the synthetic camera views within an image based on frustum culling. Thus, for each keyframe, its corresponding three hundred most relevant features and its global descriptor are stored to form the compact 3D model. The three hundred most relevant SIFT descriptors are extracted from a query image along with its CDVS descriptor. These 300 descriptors are matched via 2D-2D with the 3D model to get a score for match. Then the 2D-3D correspondences are computed from the highest scored matched 2D-2D. If enough inliers are found after RANSAC, then the image is registered and its pose is estimated.

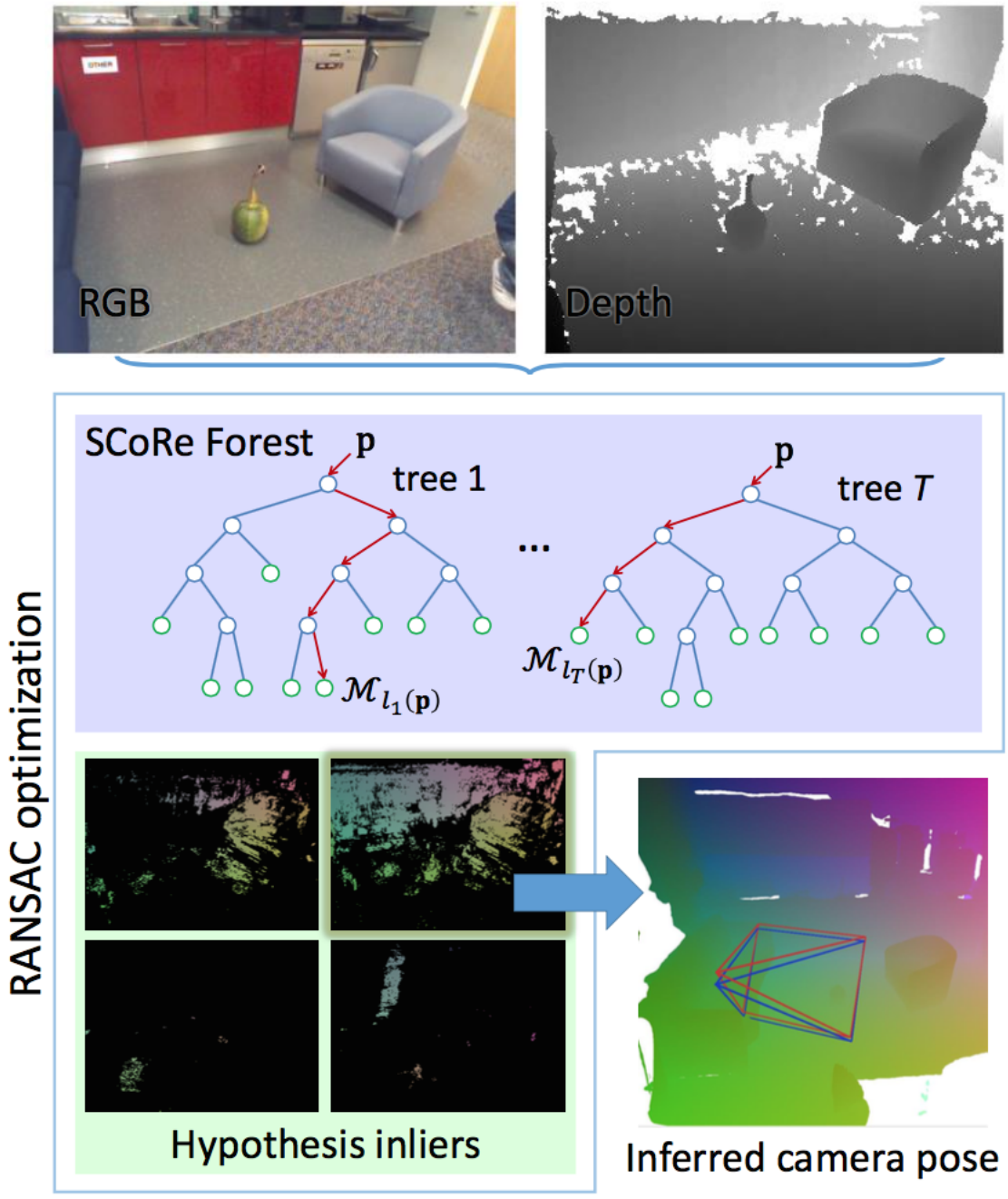


Figure 4.1: Decision Forest Pose IBL System. (Shotton et al., 2013)

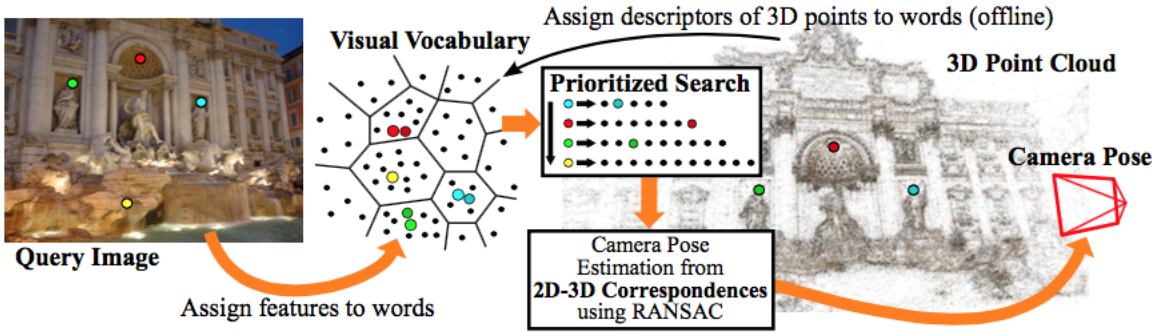


Figure 4.2: Vocabulary-based Prioritized Search(VPS). (Sattler et al., 2011)

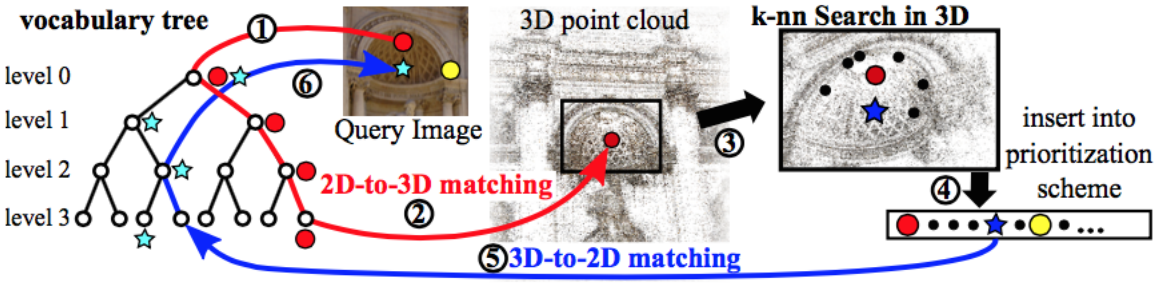


Figure 4.3: Active Search System. (Sattler et al., 2012)

4.2 Datasets and Methodology

Table 4.2 presents the eight datasets used for the assessment of IBL systems. The first six datasets are large and represent some major cities in the world. The sets were reconstructed using Bundler [62] from images that are available on the web.

The query images are the test images against which the approaches are validated. The table shows the number of 3D points that each map consists of. Dubrovnik [29], Rome [29], Quad and Vienna [22] datasets consist of a few million 3D points reconstructed from images taken by Flickr users. The Aachen [46] dataset also has a few millions 3D points and the query images were taken by Flickr users via telephone over a two year period. This makes them more difficult to process than the previous three datasets due to geometrical defection and changes that could have happened during two years. Landmark 1k [28] is the most popular 1k landmark on Flickr and together with San Francisco [28], they are considered the largest datasets, featuring tens of million of 3D points. Microsoft researcher introduced the 7 scene Dataset [50] which consists of images for seven different indoor scenes taken using an RGB-D Kinect.

Table 4.2: The major datasets used in IBL along with the total number of database images, query images and 3D points.

Dataset	Total # of Images	# of Query Images	# of 3D points [Million]
Dubrovnik [29]	6044	800	1.9
Rome [29]	15179	1000	4.1
Vienna [22]	1324	266	1.1
Quad [10]	6514	348	1
Aachen [46]	3047	369	1.5
Landmarks 1k [28]	204000	1000	38
San Francisco [28]	610000	803	30
7 Scene [50]	26000	17000	-

To ensure a fair comparison, the evaluation methodology that was followed is presented. All testing was done on an Intel Core i7-4930K CPU with 3.40GHz x12 with 32 GB Memory. Only three out of the six main approaches provided an open source for testing. Their source code is in the evaluation pipeline. Since Dubrovnik dataset provides metric ground truth and is mostly used for testing localization accuracy in IBL, it was chosen to run the testing. First, Brute Force and tree-based are used to match all the query images to the 3D map in the dataset. Then the resultant correspondences undergo first a ratio test to remove the ambiguous matches, then a cross match to remove the repeated matches. Following this, RANSAC removes the remaining outliers from those matches. If more then 12 inliers are found, then the image is registered and its pose is estimated using P6P. The estimated pose is compared to the metric provided with the dataset. For ACG localizer, a clustering file is provided where all 3D points are clustered into 100k clusters. The provided file is used to run their pipeline. Then, another clustering file is created using the same library they used to test the quantization effect. The same thresholds values used in tree-based and BF are set for the ratio test, RANSAC re-projection error and number of iterations. The resulting correspondences from their software are subjected to the cross-matching step. For Embedded Ferns, only a general classifier code was provided. It was needed to parse the Dubrovnik dataset in the way described in their paper to be able to test it on the provided classifier. Their system was automated to be run on each query image through the classifier. The resultant matches were subjected to the ratio test, the cross matching step, and then RANSAC using the same threshold values. As for the Keyframe Approach, their keyframe approach was implemented for each image in the dataset. Then, for each query image its corresponding fern was created and matched to the dataset. The closest 100 keyframes in the database were chosen and matched against their corresponding 3D map points. The resultant matches also undergo the

ratio test, the cross matching step, and then RANSAC using the same threshold values.

4.3 Results

Table 4.3 shows the results of our study on the Dubrovnik dataset conducted on the 3 provided main approaches (ACG localizer, Embedded Ferns and Keyframe Approach) along with the results from Brute Force and FLANN.

As expected, Brute Force and tree-based registered almost all the images with the best localization accuracy by the best mean and median errors with a slight advantage to Brute Force. Nevertheless, these approaches were very slow where brute force needs an average of 28.9s to register one query image and tree-based 3.6s to do the same job. Thus, these two approaches cannot be used in real time. The main goal of all of the other approaches was to make IBL feasible for real time application by speeding the registration while trying to maintain the same level of accuracy as tree-based . The results from ACG Localizer using the provided clustering approximately matched the results reported in their paper. ACG localizer gave a good mean error compared to tree-based with almost the same median error while being more then 10 times faster. Since this approach relies on clustering, another clustering file was created to test the sensitivity to the well known problems of clustering on the same dataset. The approach lost significant accuracy with the new clustering in the mean and median error but maintained the same speed. Although the Keyframe Approach scored the fastest registration times, their localization results on this dataset were poor since the average mean and median were very high compared to the other approaches. Embedded Ferns presented the worst results. Their offline training took a few days. Concerning their results, they returned the worst mean and median average errors with

Table 4.3: Results on Dubrovnik dataset for the provided main approaches using our methodology. The results include the number of registered images out of the total images, the average mean and the average median in meters, and the average registration time in seconds.

Approach	Total # Images	# Registered Images	Mean [m]	Median [m]	Average Time [s]
Brute Force	800	798	8.7	0.8	28.9
FLANN	800	794	10.4	1.1	3.6
ACG Localizer with provided clustering	800	797	31.4	1.3	0.28
ACG Localizer with our clustering	800	754	52.3	7.8	0.29
Keyframe Approach	800	697	182.1	64.3	0.21
Embedded Ferns	800	769	252.4	92.9	5.2

slow speed.

Table 4.4 reports the results of the other 3 main approaches in IBL that did not provide open source codes for testing. Their experiments were conducted on other major dataset stated in Table 4.2. The Decision Forest approach tackled small indoor scenes that were tested on 7 scene dataset. The average results on all the 7 datasets were taken. 68.3% of their images had average mean translational error less then 5cm and rotational error less then 5 degrees. The median on these registered images was less then 1% and the average time for registration was very fast (0.1s). On the other hand, Worldwide Pose Estimation registered 68.4% of their images, tested on Quad dataset, with average mean and median error very close to tree-based but with slower speed. The MPEG Search Space Reduction approach was tested on Aachen, and they only reported an average median error of 5m.

Table 4.4: Results taken from the corresponding papers of the non-available approaches. They include the dataset each approach was tested on, the % of registered images, the average time in second, and the average mean and median.

Approach	Dataset	% Registered Images	Mean	Median [m]	Average Time [s]
Decision Forest [49]	7 Scene Dataset	68.3	T < 5cm R < 5deg	<1%	0.1
Worldwide Pose Estimation [28]	Quad	68.4	5.5m	1.6	'few seconds'
MPEG Search Space Reduction [19]	Aachen	-	-	5	-

4.4 Analysis and Discussion

The results in the previous section show that IBL still lacks accuracy and robustness particularly in the main approaches. In this section the shortcomings found for each approach will be presented.

All the approaches share some common problems. These problems are the main ones that IBL still faces, though it was improved to some extent. RANSAC performance in the Image Registration phase is not robust in many situations. This problem is important and has not been addressed for a while. SIFT extraction and the nature of the descriptors in the matching phase are also still challenging problems.

ACG localizer suffered from the quantization effect in three ways: (1) the problem resides in missing the best correspondences because the matched descriptor might not be assigned to the correct visual word, (2) the clustering will result in an uneven distribution between visual words where some of them will contain a large number of descriptors, thus technically a non-efficient search space reduction and (3) the k-means clustering itself which in large-scale scenes faces challenges in scaling to this kind of size due to its inherent sequential nature. The Keyframe Approach was sensitive to the distance between the keyframes and the query needs to be taken as close as possible to the path followed while taking the keyframes. For this reason, this approach will perform much better for small or building scale scenes, not on a city-scale like Dubrovnik. As for Embedded Ferns, the paper addresses the matching as a classification step and they reported results better than tree-based in image registration. Nevertheless, they only provided a general classification code. Their main problem was that their classifier is not good enough for global matching. This is due to the need to use GPS prior on camera positions to restrict classifi-

cation by dividing city-scale scenes into small groups where each query image will be placed in one specific group based on the GPS tag.

Decision Forest reports problems of ambiguity in the environment. This approach relies on the depth provided, thus it will require a significant amount of work be done on the RGB images. Also their modified RANSAC only works for the decision forest modeled in their paper. As for Worldwide Pose Estimation, their co-occurrence RANSAC will fail when all features are from the same environment. Also, similar or identical features will probably cause false registrations. MPEG Search Space Reduction reports that their approach is not robust to illumination changes. Also, bad estimation of the focal length and skew coefficient were the cause of incorrect localization.

Solving IBL consists of solving its three main steps: keypoints matching, image registration and pose estimation. This comparison revealed major deficiencies that still face IBL: Matching is one of the major problems that affects the robustness and accuracy of IBL due to the nature of the descriptors which causes lots of false matches (scale, blur, illumination and camera perspective). Another major problem is the dimensionality of the environment consisting of millions of 3D points where the need of efficient and robust matching arises along with managing memory consumption.

Chapter 5

GIST-based Search Space Reduction (GSSR) System[1]

The comparative study proved that IBL still faces many problems. The shortcomings of the main approaches were discussed in the previous section. In the following section it was chosen to focus on reducing the search space problem as the mean to solve the image-based localization problem. A new image-based localization approach based on reducing the search space is proposed. It consists of using global descriptors to find candidate keyframes in the database and then search against the 3D points that are only seen from these candidates using local descriptors stored in a 3D cloud map as shown in Figure 5.1. The proposed novel solution has the desirable properties of speed and accuracy built in.

[1]The contents of this chapter have been submitted to the Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference. Co-authors include: Charbel Azzi, John Zelek, Daniel Asmar and Adel Fakih. I hereby verify that I am the principal author. The material used was paraphrased.

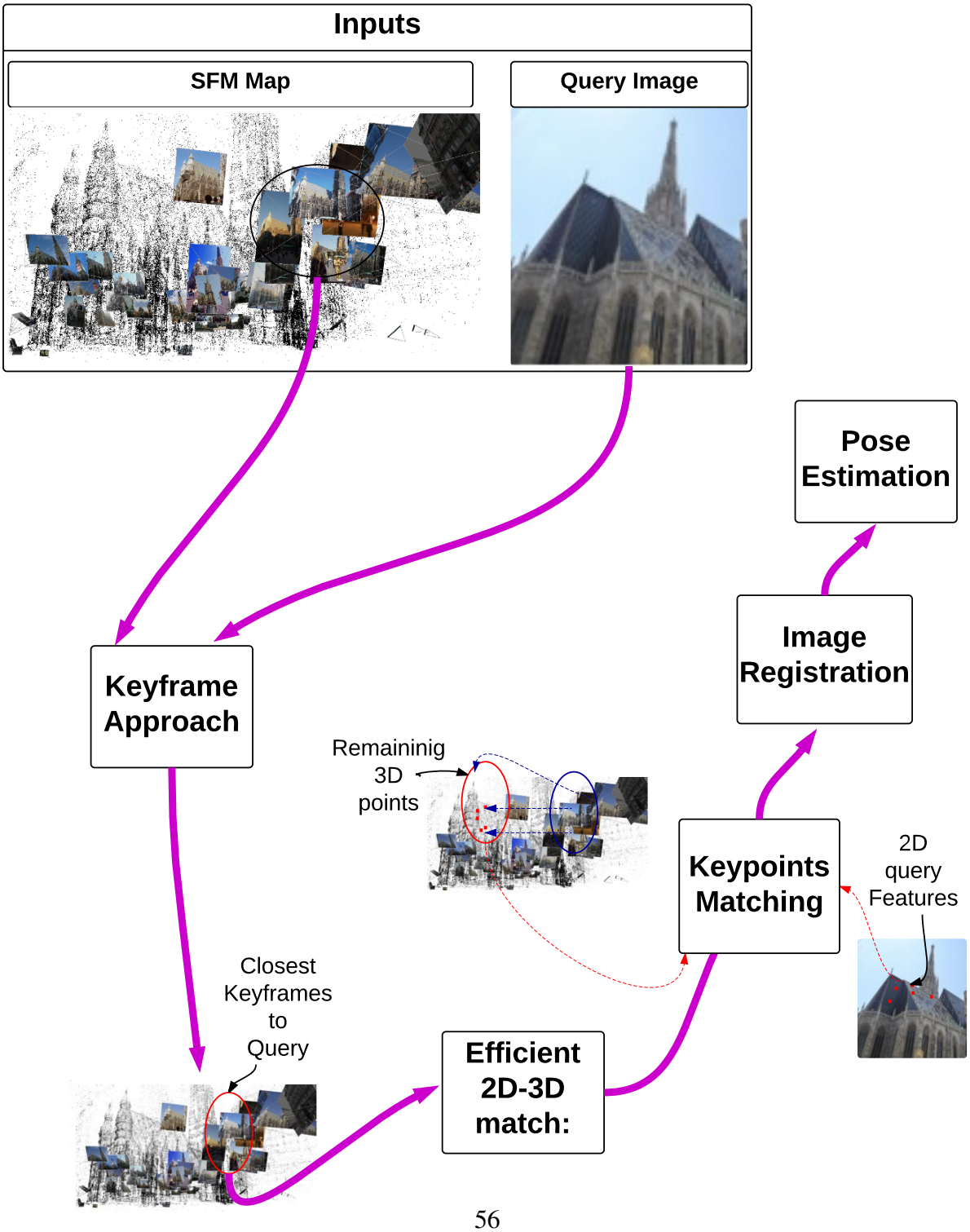


Figure 5.1: GSSR System

5.1 Search Space Problem

In the literature, there are two main approaches for the search space problem; the first one proposed by Sattler et al. [46], also known as the ACG localizer, and reduces the search space by clustering features into visual words. This method has three notable disadvantages. First, it risks missing the best correspondences because the matched descriptor might not be matched to the correct visual word. Second, the clustering sometimes results in an uneven distribution between visual words, resulting in inefficient search space reduction. Third, it relies on K-means clustering, which in large-scale scenes faces challenges in scaling to this kind of size due to its inherent sequential nature. Heisterklaus et al. [19] tackles the search space problem by using an MPEG descriptor in order to generate artificial images to cover the space. The technique is not invariant to lighting changes and is extremely sensitive to inaccuracies in the intrinsic camera parameters.

5.2 GIST-based Search Space Reduction (GSSR)

Gist-Based Search Space Reduction (GSSR), illustrated in Figure 5.1, is proposed to overcome scaling issues as well as other traits such as illumination variance. GSSR relies on the GIST global scene descriptor [58], which has shown to have relation to how humans perceive the GIST of a scene. GIST scene measures are not dependent on illumination changes. While most search space reduction methods rely on bag of words to do so, GSSR relies on GIST descriptors [58] to establish context for both the saved images in the database as well as for any query image.

The GIST descriptor proposed in [58] aims to develop a low-dimensional representation for each image. Figure 5.2 illustrates the GIST pipeline. The proposed descriptor represents the

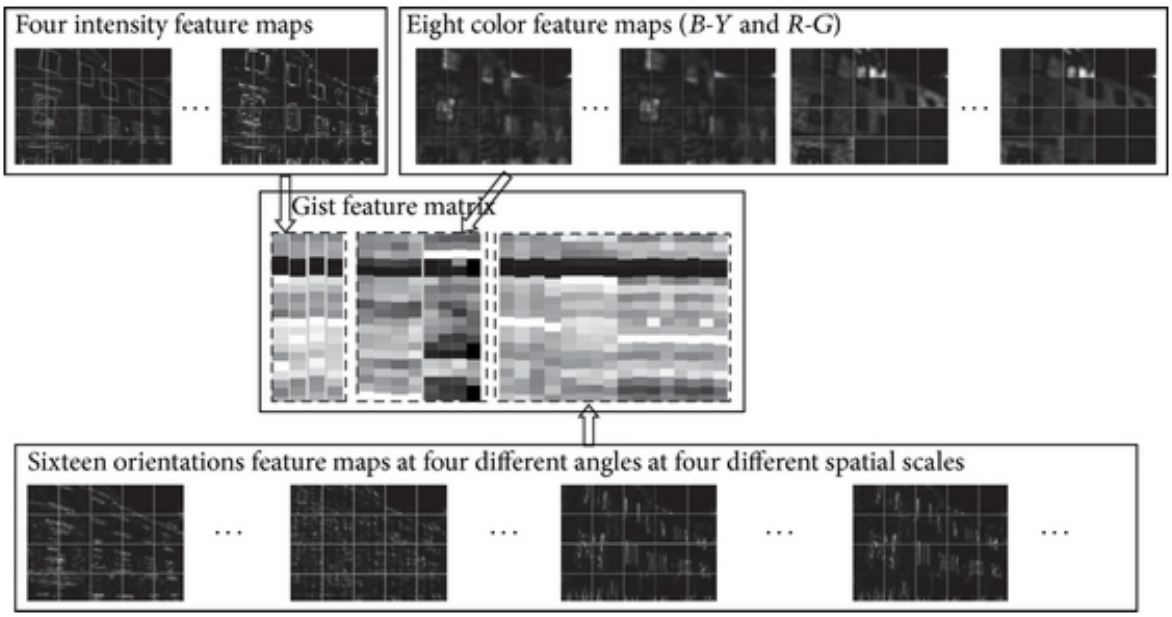


Figure 5.2: GIST descriptor computation.(Torralba et al.,2006)

dominant spatial structure of a scene. Thus, GIST summarizes the gradient information (scale and orientation) for different parts of an image. It starts by convolving the image with 32 Gabor filters at 4 intensity scales, 8 color feature orientations, producing 32 feature maps of the same size of the input image. Then it divides each feature map into 16 regions of 4x4 grids. Finally, a 512 GIST descriptor is computed by concatenating the 16 averaged values of all 32 feature maps (16x32 = 512).

GSSR is shown in Algorithm 1. In a pre-processing offline stage, a 3D map of the target environment is built using SfM. The resulting map is parsed and contains the following (see Figure 5.3):

- 3D points + their SIFT [30] descriptors + keyframes in which each 3D point was observed.

- Keyframes + a single GIST for each keyframe

Once GSSR is initiated, each query is processed to produce its own GIST descriptor as well as its SIFT descriptors. The GIST distance between the query and all the keyframes is computed (L2 norm of the GIST feature). If the distance is below a threshold then the keyframe is considered a candidate match, otherwise it is discarded since it does not belong to the same view of the query image. The threshold chosen here is determined empirically and can lead to unsuitable keyframes that do not share a large enough number of 3D points with the query image. In order to remove these outlier keyframes, a simple random consensus test is done as a refinement step. Each candidate keyframe is checked with all the other candidates according to:

$$F_k = \frac{\sum_{i=1}^{N} P_i(KF_i, KF_k)}{N}, \tag{5.1}$$

where N is the total number of candidate keyframes and P_i is the number of 3D points in common between the tested candidate KF and the keyframe at i, KF_i. P_i is computed in an offline stage after the map is reconstructed and parsed. If the ratio F_k is high enough then the candidate keyframe qualifies for localization, otherwise it is discarded. The net result at this point is a constellation of keyframes that qualify for localization. This technique is similar in spirit to the work of Shotton [16, 17].

In the next stage, the search space is reduced by matching only to 3D points seen in the qualified keyframes. This consists of only considering 3D points that are viewed by the constellation of keyframes. Then 2D-3D matching between SIFT descriptors and retained 3D points (i.e., only seen by the qualified keyframes) is done using ANN [35]. If more than 12 inliers are found, the

image is registered, if not it is discarded. The empirical value of 12 is taken from the relevant body of literature [29]. Finally, the pose is refined using the qualified registered image.

Figure 5.3: Inputs into the GSSR system

Algorithm 1: GSSR Algorithm

Get the GIST for each KF(Keyframe) + the 3D pts and all the Kf's each pt is visible in from VSFM map + the camera transformation estimates from VSFM
Take a query image Q and extract its GIST
for *all database KFs* **do**
    Compute the cost $C(Q, KF_i)$ = GIST distance between Q and KF_i
    if $C(Q, KF_i) < N_{(min)threshold}$ **then**
        Qualify KF_i for next step
    else if $C(Q, KF_i) > N_{(max)threshold}$ **then**
        Discard KF_i
for *All qualified KFs* **do**
    if KF_i *have enough number of visible 3D pts between the other KFs* **then**
        Qualify KF_i to localization step
    else
        Discard KF_i
▷ **Match the query to the 3D pts coming from the final qualified KFs:**
Take the 3D pts viewed only in the qualified KFs
Perform a 2D-3D match between the query and those 3D pts
Image Registration: Reject outliers via RANSAC and ratio test. If enough Inliers are found then Image qualifies to the Pose Estimation otherwise discard the image
Pose Estimation

5.3 Experiments

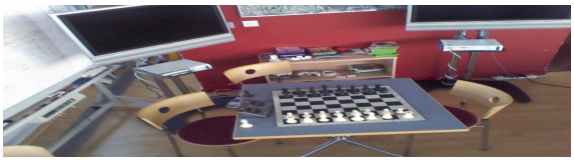
This section describes the experiments that were conducted to validate GSSR. First, the datasets are described (including ground truth) and then the evaluation methodology is presented for the benchmarking of GSSR against the tree-based technique.

5.3.1 Datasets

For the evaluation of GSSR, the 7 scenes datasets provided by Microsoft Research [16, 49] is used. These datasets are presented in Table 5.1 and consist of seven different indoor locations; each mapped using an RGB-D Kinect camera, resulting in a 3D metric ground truth map for each scene. The choice of each scene represents different photometric challenges, namely (1) motion blur and illumination changes, (2) flat surface and repetitive structures (especially in the Stairs dataset), and (3) reflectivity (especially in the RedKitchen dataset). Each dataset offers training data, which can be used for the 3D reconstruction of the scene, as well as a test dataset, which can be used as query images. These datasets are used to build the ground truth model and the 3D map in order to evaluate GSSR and compare it against other IBL techniques.

5.3.2 Evaluation Methodology

This section describes how the 3D map is built as well as how the ground truth with metric scale is obtained.



(a) Chess scene

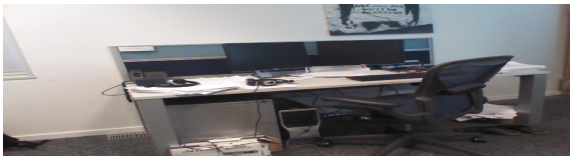
(b) Chess 3D Map

(c) Fire scene

(d) Fire 3D Map

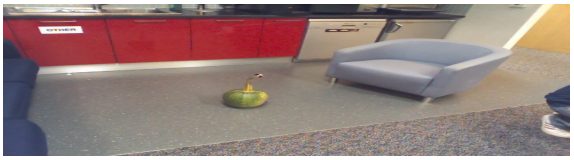
(e) Heads scene

(f) Heads 3D Map

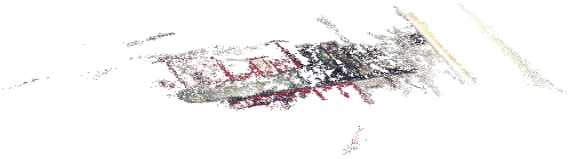


(g) Office scene

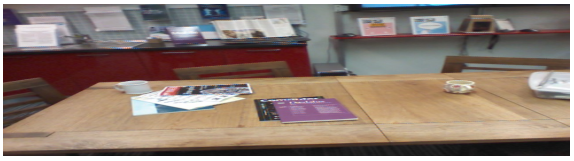
(h) Office 3D Map



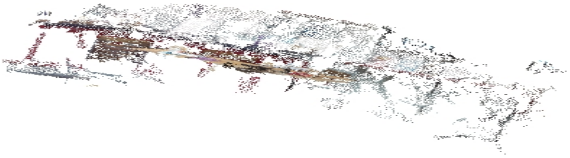
(i) Pumpkin scene



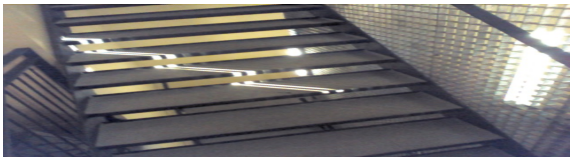
(j) Pumpkin Map



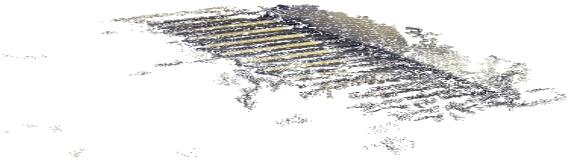
(k) RedKitchen scene



(l) RedKitchen 3D Map



(m) Stairs scene



(n) Stairs 3D Map

Figure 5.4: 3D point cloud map for each scene reconstructed from VSFM [65]

Table 5.1: The 7 scenes dataset by Microsoft research

Dataset	# Images		#3D Points
	Keyframes	Query	
Chess	4000	2000	69557
Fire	2000	2000	146973
Heads	1000	1000	87332
Office	6000	4000	77926
Pumpkin	4000	2000	55536
RedKitchen	7000	5000	80172
Stairs	2000	1000	42463

First, the training and test images in each dataset are processed by VSFM [65] to yield a full 3D reconstruction for each scene. Next, the test images are considered as ground truth by removing their corresponding 3D points from the reconstructed model, along with their corresponding SIFT descriptors and poses. Each 3D point is then represented by the mean value of all of its SIFT desrcriptors. Finally, ground truth provided by the dataset [16, 49] is used to obtain a metric scale for the ground truth. This is done by aligning the ground truth map from [16, 49] to the ground truth and then extracting scale between the two.

In the evaluation of GSSR against the tree-based approarch the following should be noted:

- All tests were performed on an Intel Core 7-4930K CPU with 3.4 GHzx12 with 32GB memory.
- In the keyframe approach of GSSR, keyframes that have a normalized distance to a query image less then 0.2 are accepted for the refinement step.

- Only 3D map points that are seen from the qualified keyframes are matched to the 2D features from the query. This is in contrast to the tree-based method, where all 2D features are matched to all 3D map points.

- To ensure a fair comparison, tree-based and GSSR are first subjected to Lowe's ratio test [30] with a value of 0.8 to remove the ambiguous matches, then a cross-match is done to remove the repeated matches. RANSAC and PnP [25] removes the remaining outliers from those matches.

- As proposed in all the IBL approaches, a query image is registered if more then 12 inliers are found and its pose is estimated using PnP. The threshold of 12 inliers is proposed by [29], it is chosen to be high enough to make it unlikely for a false candidate to have this many inliers and low enough for true candidates with low number of features not to be rejected.

- The estimated pose is then compared to the metric ground truth.

- The main criterion for comparison is to evaluate the pose (translational and rotational) errors for each scene with respect to the ground truth. An accurately localized image is considered if it has a translational error less then 2cm and a rotational error less then 2 degrees.

5.4 Results

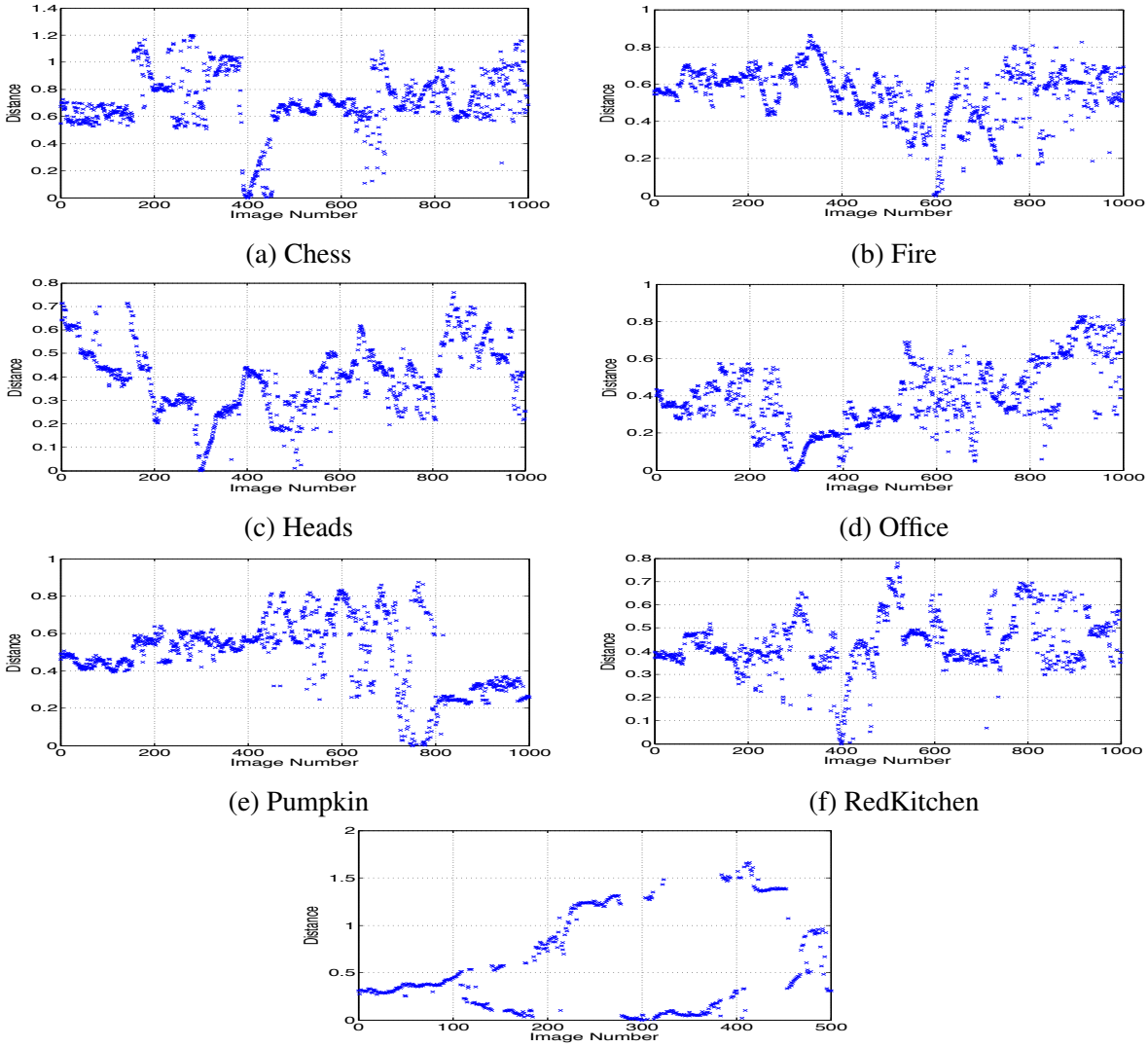
This section presents the results of the experiments conducted in Section 5.3. Results include performance of the GIST matching, results of the GSSR, as well as benchmarking against other IBL systems.

5.4.1 GIST Matching

Samples of query images from each scene to show the GIST clustering capability to the closest keyframes to the query image. Figure 5.5 shows the distance between the GIST descriptor of a query image taken at random and 1000 keyframes from the Chess dataset. For example in 5.5a the query image is number 400 on the x axis and the clusters of points(KFs) around it are the ones who have the smallest distance. Note the close matching of the GIST descriptors of the keyframes located in the vicinity of Frame 400. Another cluster is noted around images number 600-700 where the user revisits part of scene corresponding to Image 400. Results on the other six datasets are similar.

5.4.2 Performance of GSSR

Table 5.2 presents the results of the GSSR approach benchmarked against the tree-based approach. On all seven databases GSSR produces superior localization accuracy in terms of translational and rotational errors and standard deviations. The main advantage of the GSSR appears in the computational time, featuring significant speed-ups in performance, where for the Heads dataset note a speed-up of roughly 4 times faster. Finally, using GSSR, there is a notable increase



(a) Chess

(b) Fire

(c) Heads

(d) Office

(e) Pumpkin

(f) RedKitchen

(g) Stairs

Figure 5.5: This graph shows the GIST distance between a sample query image randomly taken from each dataset to a 1000 Keyframes in the corresponding dataset. The query is taken among from the 1000 Keyframes for better visualization where for example in 5.5a the query image is number 400 on the x axis and the clusters of points around it are the ones who have the smallest distance to it

in the percentage of inliers, where for the stairs dataset, more than twice the number of inliers was detected. These results are better appreciated in graphical form in Figure 5.6a.

In terms of the 3D points used for each scene, note in Table 5.4 the efficiency of GSSR at reducing the search space, where for the Heads database it reaches a reduction nearing 96%, with an average of 92% of the original data points while maintaining a better accuracy then tree-based. This reduction has a direct implication on the reduced computational costs incurred in GSSR versus the tree-based approach. Figure 5.6b shows these results in graphical form.

In addition to this direct comparison, GSSR is further compared to the published results of Shotton et al. [16, 49] on the same datasets using the decision forest [49] and the the Keyframe approach [16]. The results are shown in Table 5.3. The percentage accuracy reported in the table is calculated as the number of query images featuring a translational error less than 2cm and at the same time a rotational error less than 2 degrees. The exception is for the decision forest technique where the published accuracy is based on translational errors smaller than 5cm and rotational errors less than 5 degrees. Note the considerable improvement in GSSR over the other techniques. Even for difficult scenes like Pumpkin, RedKitchen and stairs, featuring ambiguous scenes GSSR performance is superior to the prior art.

Table 5.2: GSSR performance benchmarked against the tree-based approach

Dataset	R Error (Deg) Mean/SD		T Error (cm) Mean/SD		Average Time(ms)		% Inliers	
	GSSR	Tree-Based	GSSR	Tree-Based	GSSR	Tree-Based	GSSR	Tree-Based
Chess	0.2914/0.1	0.3139/0.25	0.315/0.17	0.3235/0.19	34	107	58.7	52.5
Fire	0.148/0.02	0.149/0.02	0.5412/0.23	0.5557/0.26	56	201	65.1	50.0
Heads	0.226/0.05	0.232/0.05	0.7275/0.62	0.7379/0.61	18	93	50.8	42.5
Office	0.234/0.05	0.238/0.05	0.2814/0.05	0.2959/0.06	28	98	51.5	45.2
Pumpkin	0.291/0.08	0.311/0.15	0.2839/0.07	0.2902/0.11	18	54	60.0	58.2
RedKitchen	0.151/0.01	0.156/0.03	0.4842/0.36	0.4922/0.38	28	108	55.4	49.9
Stairs	1.647/1.32	1.652/1.58	2.751/2.05	2.8517/2.09	22	31	28.1	13.1
Average	***0.426/0.24***	***0.436/0.30***	***0.784/0.50***	***0.793/0.53***	***29***	***99***	***52.8***	***44.5***

Table 5.3: GSSR performance in terms of accuracy benchmarked against the tree-based, Decision Forest, and Keyframe approach. Note the improvement over the tree-based approach for all datasets and the considerable improvements over the decision forest and the Keyframe approach

Dataset	#Query Images	% Accurate Images			
		GIST Approach	Tree-Based	Decision Forest	Keyframe Approach
Chess	2000	97.4	97	92.6	85.3
Fire	2000	98	96.5	82.9	72
Heads	1000	93.2	93	49.4	79.8
Office	4000	99.2	98.8	79.1	74.7
Pumpkin	2000	99.3	98.1	73.7	62.8
RedKitchen	5000	97.5	96	72.9	54.1
Stairs	1000	40.3	36	27.8	34.1
Average		***89.3***	***87.9***	***68.3***	***66.1***

5.5 Discussion

The results presented in the previous section demonstrate the accuracy and robustness of GSSR where it exhibits powerful capabilities of querying the search space. Indeed, the fact that GSSR presents higher inlier ratios than tree-based justifies the results of higher accuracies.

Since the 2D features of a query image is matched to 3D points belonging exclusively to the constellation of keyframes, a large number of correct matches are more likely to be found.

Figure 5.7 shows the camera trajectories on the Heads, RedKitchen and Stairs scenes. Note the accuracy and consistency of GSSR in following the ground truth path of the camera. The graph also shows a smoother track than the tree-based approach, especially in regions where tree-based looses track but GSSR does not.

As far as search space reduction, the experimental results show a marked decrease in search space by an average of approximately 92% and yet the system maintains superior accuracy than the tree-based method.

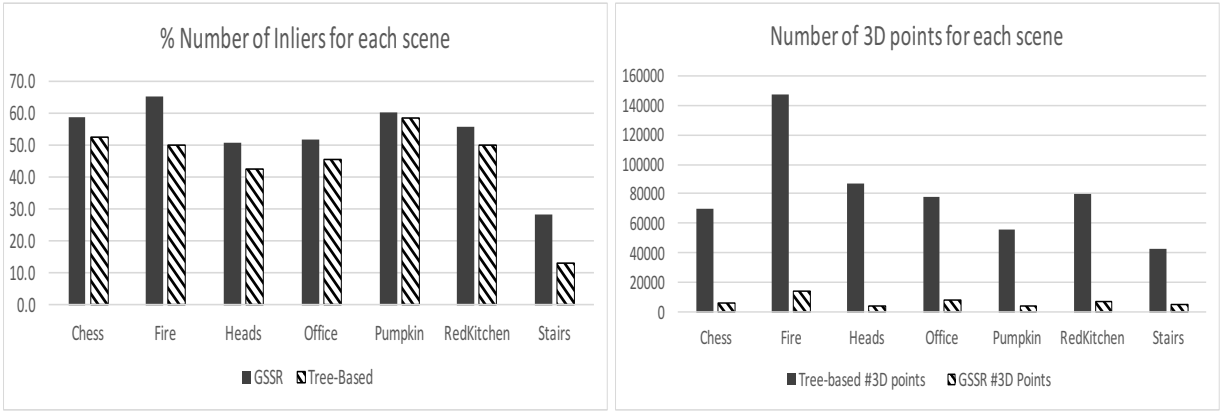
It is notable that the translation and rotational errors differed from one scene to another. The most significant drop in accuracy is for both GSSR and tree-based is for the Stairs dataset. This is most probably due to the repetitive structure of the stairs in that scene.

GSSR showed notable improvement over the Decision Forest and Keyframe systems, which performed decently on the Pumpkin and RedKitchen datasets. This is probably due to the reflectivity nature of those scenes, where the ground, the cupboards and the kitchen structure cause lots of reflections in the images, which may result in false positive matches.

Like all IBL system, GSSR performs generally decently with structures such as the Stairs dataset and relatively better than state-of-the art approaches, which register very few images. Note that most of the state-of-the-art approaches other then Descion Forest and Keyframe Approach did not even register any image. This is due to the repetitive nature of the structure of the scene, which causes false positive matches.

Table 5.4: Search space reduction efficiency of GSSR

Dataset	Intial #3D Points	#3D Points after GSSR	%Search Space Reduction
Chess	69557	6248	91.0
Fire	146973	14470	90.2
Heads	87332	3727	95.7
Office	77926	7603	90.2
Pumpkin	55536	4444	92.0
RedKitchen	80172	7428	90.7
Stairs	42462.5	5027	88.2
Average	***79994***	***6992***	***91.1***



(a) (b)

Figure 5.6: 5.6a shows the average numbers of inliers for each scene. 5.6b shows the number of 3D points for each scene initially in the map and after the GSSR

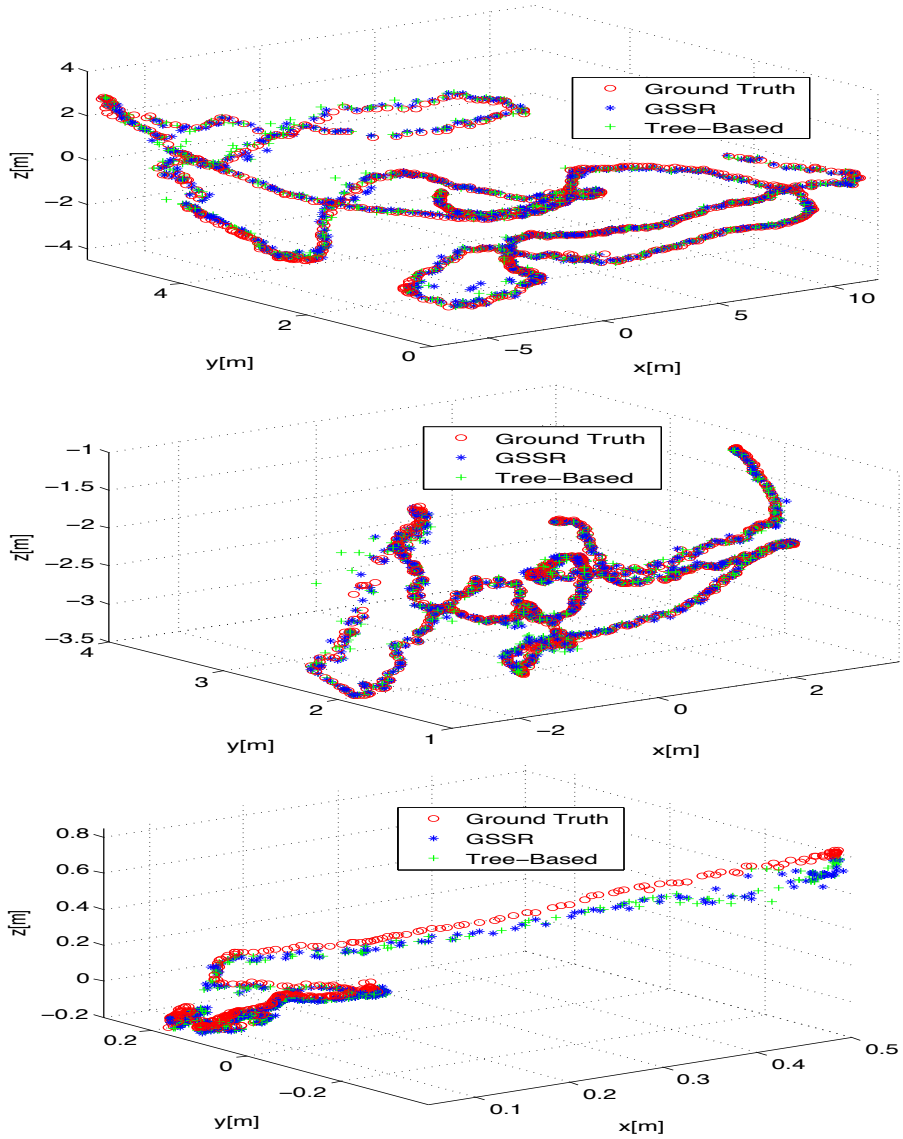


Figure 5.7: Tracking of the camera motion using GSSR (*) and tree-based IBL (+) for the Heads, RedKitchen and Stairs scenes. Ground truth is shown as (o).

Chapter 6

Conclusion

Solving IBL consists of solving its three main steps: keypoints matching, image registration and pose estimation. Matching is one of the major problems that affects the robustness and accuracy of IBL due to the nature of the descriptors which causes lots of false matches (scale, blur, illumination and mainly camera perspective). Another major problem is the dimensionality of the environment consisting of millions of 3D points where the need of an efficient and robust matching procedure arises along with managing memory consumption.

This thesis presents a detailed description and study of three different 3D modeling packages based on SFM to reconstruct a 3D map of an environment. The packages tested are VSFM, Bundler and PTAM. Image matching which is the bottleneck of SFM, SLAM and IBL plays the major role in favour of VSFM. VSFM's preemptive feature matching approach combined with the mixing of RT and BA gave VSFM a high number of robust matches relative to PTAM and Bundler. These allowed a large number of inliers, and so a large number of features in the

reconstructed map. This is a major necessity for IBL since good quality matches in the map lead to more inliers and better localization accuracy. For this reason, VSFM was chosen to reconstruct the 3D maps for IBL.

This work presents a detailed description of the pipeline of IBL and the main approaches to solve this problem. Also, a comparison study presented benchmark results for the available main approaches on the Dubrovnik dataset used in this field. Brute force and tree-based showed the best localization accuracy but are very slow. The results showed that Visual Words (ACG Localizer) presented the best results through its sensitivity to clustering. The Keyframe Approach should be used with small-scale applications. Embedded ferns gave good registration performance but very bad localization quality and poor time efficiency due to its restriction to divide the large-scenes into small subsets. To sum up, Visual Words has the best IBL system that tackles the search space problem. Nevertheless, this system and the other main approaches still lack accuracy and robustness and the major problems of IBL are still problematic.

This work also presents a novel approach for the search space reduction problem in IBL. GSSR was bench-marked on the 7 scenes dataset of Microsoft. Results show better localization accuracy then tree-based, decision forest, and Keyframe approaches. More importantly, GSSR showed considerable speed-ups in computational times, sometimes 4 times a faster then the tree-based approach. The speed-up is primarily due to the ability of GSSR to considerably reduce the search space and yet produce superior accuracy compated to other state-of-the-art techniques.

References

[1] Amit, Y., Geman, D., 1997. Shape quantization and recognition with randomized trees. Neural computation 9 (7), pp. 1545–1588.

[2] Andoni, A., Indyk, P., 2006. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In: Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on. IEEE, pp. 459–468.

[3] Arth, C., Wagner, D., Klopschitz, M., Irschara, A., Schmalstieg, D., 2009. Wide area localization on mobile phones. In: Mixed and Augmented Reality, 2009. ISMAR 2009. 8th IEEE International Symposium on. IEEE, pp. 73–82.

[4] Arya, S., Mount, D. M., Netanyahu, N., Silverman, R., Wu, A. Y., 1994. An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. In: Proc. 5th ACM-SIAM Sympos. Discrete Algorithms. pp. 573–582.

[5] Avrithis, Y., Kalantidis, Y., Tolias, G., Spyrou, E., 2010. Retrieving landmark and non-landmark images from community photo collections. In: Proceedings of the international conference on Multimedia. ACM, pp. 153–162.

[6] Bay, H., Tuytelaars, T., Van Gool, L., 2006. Surf: Speeded up robust features. In: Computer vision–ECCV 2006. Springer, pp. 404–417.

[7] Calonder, M., Lepetit, V., Strecha, C., Fua, P., 2010. Brief: Binary robust independent elementary features. Computer Vision–ECCV 2010, pp. 778–792.

[8] Castle, R., Klein, G., Murray, D. W., 2008. Video-rate localization in multiple maps for wearable augmented reality. In: Wearable Computers, 2008. ISWC 2008. 12th IEEE International Symposium on. IEEE, pp. 15–22.

[9] Chen, D. M., Baatz, G., Köser, K., Tsai, S. S., Vedantham, R., Pylvä, T., Roimela, K., Chen, X., Bach, J., Pollefeys, M., et al., 2011. City-scale landmark identification on mobile devices. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE, pp. 737–744.

[10] Crandall, D., Owens, A., Snavely, N., Huttenlocher, D., 2011. Discrete-continuous optimization for large-scale structure from motion. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE, pp. 3001–3008.

[11] Davison, A. J., 2003. Real-time simultaneous localisation and mapping with a single camera. In: Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on. IEEE, pp. 1403–1410.

[12] Donoser, M., Schmalstieg, D., 2014. Discriminative feature-to-point matching in image-based localization. In: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on. IEEE, pp. 516–523.

[13] Duan, L.-Y., Gao, F., Chen, J., Lin, J., Huang, T., 2013. Compact descriptors for mobile visual search and mpeg cdvs standardization. In: Circuits and Systems (ISCAS), 2013 IEEE International Symposium on. IEEE, pp. 885–888.

[14] Fukunaga, K., Narendra, P. M., 1975. A branch and bound algorithm for computing k-nearest neighbors. Computers, IEEE Transactions on 100 (7), pp. 750–753.

[15] Gavin, H., 2011. The levenberg-marquardt method for nonlinear least squares curve-fitting problems. Department of Civil and Environmental Engineering, Duke University, pp. 1–15.

[16] Glocker, B., Izadi, S., Shotton, J., Criminisi, A., 2013. Real-time rgb-d camera relocalization. In: Mixed and Augmented Reality (ISMAR), 2013 IEEE International Symposium on. IEEE, pp. 173–179.

[17] Glocker, B., Shotton, J., Criminisi, A., Izadi, S., 2015. Real-time rgb-d camera relocalization via randomized ferns for keyframe encoding. Visualization and Computer Graphics, IEEE Transactions on 21 (5), pp. 571–583.

[18] Hays, J., Efros, A., et al., 2008. Im2gps: estimating geographic information from a single image. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. IEEE, pp. 1–8.

[19] Heisterklaus, I., Qian, N., Miller, A., 2014. Image-based pose estimation using a compact 3d model. In: Consumer Electronics Berlin (ICCE-Berlin), 2014 IEEE Fourth International Conference on. IEEE, pp. 327–330.

[20] Hoseinnezhad, R., Bab-Hadiashar, A., 2011. An m-estimator for high breakdown robust

estimation in computer vision. Computer Vision and Image Understanding 115 (8), pp. 1145–1156.

[21] Irschara, A., Zach, C., Frahm, J.-M., Bischof, H., 2009. From structure-from-motion point clouds to fast location recognition. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, pp. 2599–2606.

[22] Irschara, A., Zach, C., Frahm, J.-M., Bischof, H., 2009. From structure-from-motion point clouds to fast location recognition. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, pp. 2599–2606.

[23] Ke, Y., Sukthankar, R., 2004. Pca-sift: A more distinctive representation for local image descriptors. In: Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on. Vol. 2. IEEE, pp. II–506.

[24] Klein, G., Murray, D., 2007. Parallel tracking and mapping for small ar workspaces. In: Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on. IEEE, pp. 225–234.

[25] Kneip, L., Scaramuzza, D., Siegwart, R., 2011. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE, pp. 2969–2976.

[26] Knopp, J., Sivic, J., Pajdla, T., 2010. Avoiding confusing features in place recognition. In: Computer Vision–ECCV 2010. Springer, pp. 748–761.

[27] Lepetit, V., Fua, P., 2006. Keypoint recognition using randomized trees. Pattern Analysis and Machine Intelligence, IEEE Transactions on 28 (9), pp. 1465–1479.

[28] Li, Y., Snavely, N., Huttenlocher, D., Fua, P., 2012. Worldwide pose estimation using 3d point clouds. In: Computer Vision–ECCV 2012. Springer, pp. 15–29.

[29] Li, Y., Snavely, N., Huttenlocher, D. P., 2010. Location recognition using prioritized feature matching. In: Computer Vision–ECCV 2010. Springer, pp. 791–804.

[30] Lowe, D. G., 2004. Distinctive image features from scale-invariant keypoints. International journal of computer vision 60 (2), pp. 91–110.

[31] Molton, N., Davison, A. J., Reid, I., 2004. Locally planar patch features for real-time structure from motion. In: BMVC. pp. 1–10.

[32] Morel, J.-M., Yu, G., 2009. Asift: A new framework for fully affine invariant image comparison. SIAM Journal on Imaging Sciences 2 (2), pp. 438–469.

[33] Mouragnon, E., Lhuillier, M., Dhome, M., Dekeyser, F., Sayd, P., 2009. Generic and real-time structure from motion using local bundle adjustment. Image and Vision Computing 27 (8), pp. 1178–1193.

[34] Muja, M., Lowe, D. G., 2009. Fast approximate nearest neighbors with automatic algorithm configuration. VISAPP (1) , 2.

[35] Muja, M., Lowe, D. G., 2014. Scalable nearest neighbor algorithms for high dimensional data. Pattern Analysis and Machine Intelligence, IEEE Transactions on 36 (11), pp. 2227–2240.

[36] Murillo, A. C., Singh, G., Kosecka, J., Guerrero, J. J., 2013. Localization in urban environments using a panoramic gist descriptor. Robotics, IEEE Transactions on 29 (1), 146–160.

[37] Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., Kohi, P., Shotton, J., Hodges, S., Fitzgibbon, A., 2011. Kinectfusion: Real-time dense surface mapping and tracking. In: Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on. IEEE, pp. 127–136.

[38] Nister, D., Stewenius, H., 2006. Scalable recognition with a vocabulary tree. In: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. Vol. 2. IEEE, pp. 2161–2168.

[39] Özuysal, M., Calonder, M., Lepetit, V., Fua, P., 2010. Fast keypoint recognition using random ferns. Pattern Analysis and Machine Intelligence, IEEE Transactions on 32 (3), 448–461.

[40] Raguram, R., Chum, O., Pollefeys, M., Matas, J., Frahm, J., 2013. Usac: a universal framework for random sample consensus. Pattern Analysis and Machine Intelligence, IEEE Transactions on 35 (8), 2022–2038.

[41] Robertson, D. P., Cipolla, R., 2004. An image-based system for urban navigation. In: BMVC. Citeseer, pp. 1–10.

[42] Rosten, E., Drummond, T., 2006. Machine learning for high-speed corner detection. In: Computer Vision–ECCV 2006. Springer, pp. 430–443.

[43] Rublee, E., Rabaud, V., Konolige, K., Bradski, G., 2011. Orb: an efficient alternative to sift

or surf. In: Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, pp. 2564–2571.

[44] Sarkis, M., Diepold, K., 2012. Camera-pose estimation via projective newton optimization on the manifold. Image Processing, IEEE Transactions on 21 (4), 1729–1741.

[45] Sattler, T., Leibe, B., Kobbelt, L., 2011. Fast image-based localization using direct 2d-to-3d matching. In: Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, pp. 667–674.

[46] Sattler, T., Leibe, B., Kobbelt, L., 2012. Improving image-based localization by active correspondence search. In: Computer Vision–ECCV 2012. Springer, pp. 752–765.

[47] Sattler, T., Sweeney, C., Pollefeys, M., 2014. On sampling focal length values to solve the absolute pose problem. In: Computer Vision–ECCV 2014. Springer, pp. 828–843.

[48] Schindler, G., Brown, M., Szeliski, R., 2007. City-scale location recognition. In: Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on. IEEE, pp. 1–7.

[49] Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., Fitzgibbon, A., 2013. Scene coordinate regression forests for camera relocalization in rgb-d images. In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on. IEEE, pp. 2930–2937.

[50] Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., Fitzgibbon, A., 2013. Scene coordinate regression forests for camera relocalization in rgb-d images. In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on. IEEE, pp. 2930–2937.

[51] Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., Moore, R., 2013. Real-time human pose recognition in parts from single depth images. Communications of the ACM 56 (1), 116–124.

[52] Siagian, C., Itti, L., 2009. Biologically inspired mobile robot vision localization. Robotics, IEEE Transactions on 25 (4), pp. 861–873.

[53] Silpa-Anan, C., Hartley, R., 2008. Optimised kd-trees for fast image descriptor matching. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. IEEE, pp. pp. 1–8.

[54] Sinha, S. N., Frahm, J.-M., Pollefeys, M., Genc, Y., 2006. Gpu-based video feature tracking and matching. In: EDGE, Workshop on Edge Computing Using New Commodity Architectures. Vol. 278. pp. pp. 4321–4330.

[55] Snavely, N., Seitz, S. M., Szeliski, R., 2006. Photo tourism: exploring photo collections in 3d. ACM transactions on graphics (TOG) 25 (3), 835–846.

[56] Snavely, N., Seitz, S. M., Szeliski, R., 2008. Modeling the world from internet photo collections. International Journal of Computer Vision 80 (2), 189–210.

[57] Strasdat, H., Montiel, J. M., Davison, A. J., 2012. Visual slam: Why filter? Image and Vision Computing 30 (2), 65–77.

[58] Torralba, A., Oliva, A., Castelhano, M. S., Henderson, J. M., 2006. Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search. Psychological review 113 (4), 766.

[59] von Hiestand, R., 2015. Regard3d. http://www.regard3d.org, accessed: 2015-11-23.

[60] Weber, R., Schek, H.-J., Blott, S., 1998. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In: VLDB. Vol. 98. pp. 194–205.

[61] Wendel, A., Irschara, A., Bischof, H., 2011. Natural landmark-based monocular localization for mavs. In: Robotics and Automation (ICRA), 2011 IEEE International Conference on. IEEE, pp. 5792–5799.

[62] Wu, C., 2011. Bundler: Structure from motion (sfm) for unordered image collections. http://www.cs.cornell.edu/~snavely/bundler/, accessed: 2015-11-25.

[63] Wu, C., 2011. Visualsfm: A visual structure from motion system. http://ccwu.me/vsfm/, accessed: 2015-11-25.

[64] Wu, C., 2013. Towards linear-time incremental structure from motion. In: 3DTV-Conference, 2013 International Conference on. IEEE, pp. 127–134.

[65] Wu, C., Agarwal, S., Curless, B., Seitz, S. M., 2011. Multicore bundle adjustment. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE, pp. 3057–3064.

[66] Zamir, A. R., Shah, M., 2010. Accurate image localization based on google maps street view. In: Computer Vision–ECCV 2010. Springer, pp. 255–268.

[67] Zhang, W., Kosecka, J., 2006. Image based localization in urban environments. In: 3D Data Processing, Visualization, and Transmission, Third International Symposium on. IEEE, pp. pp. 33–40.