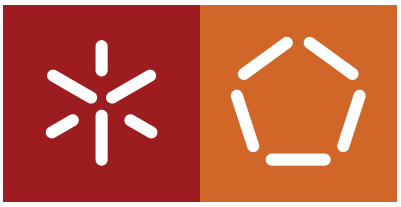

Universidade do Minho
Escola de Engenharia
Departamento de Informática

Master Course in Computing Engineering

Nuno Mourão de Amorim

Image Geocoding

Master dissertation

Supervised by: Jorge Gustavo Rocha

Braga, October 31, 2014

ACKNOWLEDGEMENTS

My acknowledgements will be brief as the people which I would like to thank already known that I am eternally indebted to them. First of all I would like to thank my supervisor Professor Jorge Gustavo Rocha for offering me the opportunity of researching this interesting area of Computer Vision and providing the necessary tools to perform this research. I also would like to give a special thanks to PhD student Paulo Almeida for his patience and availability on providing me guidance during this year.

And finally, I would like thank my girlfriend, family and friends for supporting me and providing a pleasant atmosphere to inspire me on my research.

ABSTRACT

The Global Positioning System is well known for not reaching indoor environments. Several Indoor Positioning System's have been proposed, but most of these solutions either deliver high accuracy errors or require expensive material to attenuate positioning errors. In this thesis we propose the use of two image based location recognition routines to compute the position and orientation on indoor environments. These routines are based on Structure from Motion, an incremental algorithm which recovers the 3D structure from related photographs. 3D structures generated are geocoded, compressed and stored in a database. New photographs taken from cheap cameras can be geocoded at any time. By combining Structure from Motion with the already existing Synthetic Views and Prioritized Features algorithms for fast location recognition, we are able to compute indoor GPS coordinates and orientation of new photographs within few seconds.

RESUMO

O Sistema de Posicionamento Global é bem conhecido por não alcançar espaços interiores. Diversos Sistemas de Posicionamento Interior foram propostos, mas a maioria destas soluções calculam grandes erros de precisão ou necessitam de material dispendioso para atenuar estes erros. Nesta tese, nós propomos o uso de duas rotinas de reconhecimento de locais baseadas em imagem para calcular a posição e orientação em espaços interiores. Estas rotinas são baseadas em *Structure from Motion*, um algoritmo incremental que recupera a estrutura 3D através de fotografias associadas. As estruturas 3D geradas são georreferenciadas, comprimidas e armazenadas numa base de dados. Ao combinar *Structure from Motion* com os algoritmos existentes *Synthetic Views* e *Prioritized Features* para um reconhecimento de localização rápida, nós conseguimos calcular as coordenadas GPS e orientação de novas fotografias dentro de poucos segundos.

CONTENTS

Contents . . . iii

i INTRODUCTORY MATERIAL . . . 3
1 INTRODUCTION . . . 4
1.1 State of the Art . . . 5
1.2 Research Hypotheses . . . 7
1.3 Methodology . . . 7
1.4 Document structure . . . 8
1.5 Summary . . . 8
2 STRUCTURE FROM MOTION . . . 10
2.1 Concept . . . 10
2.2 Algorithm . . . 10
2.3 Feature Extraction . . . 11
2.4 Coarse Matching . . . 12
2.4.1 Vocabulary Trees . . . 12
2.4.2 Preemptive Feature Matching . . . 15
2.5 Feature Matching . . . 16
2.6 Geometric Verification and Projection . . . 17
2.6.1 Projection Matrix . . . 18
2.7 Bundle Adjustment . . . 19
2.8 3D Scene Representation . . . 21
2.9 Structure from Motion Software's . . . 21
2.10 Summary . . . 22

ii CORE OF THE DISSERTATION . . . 23
3 FROM OUTDOOR TO INDOOR REPRESENTATIONS . . . 24
3.1 Challenges on Building Indoor SFM Models . . . 24
3.2 Summary . . . 25
4 IMAGE BASED LOCALIZATION . . . 26
4.1 GPS Transformation Matrix (Position and Orientation) . . . 26
4.2 Proposed Methods for Indoor Geocoding . . . 27
4.2.1 Synthetic Views . . . 27
4.2.2 Prioritized Features . . . 28
4.3 Implementation Details . . . 29

4.3.1 **Implementation Details: Synthetic Views** . . . 29
4.3.2 **Implementation Details: Prioritized Features** . . . 33
4.4 **Summary** . . . 35
5 EXPERIMENTAL RESULTS AND DISCUSSION . . . 36
5.1 **Datasets** . . . 36
5.1.1 **Indoor GPS Approximation** . . . 37
5.2 **Experimental Results** . . . 38
5.2.1 **Experimental Results: Synthetic Views** . . . 39
5.2.2 **Experimental Results: Prioritized Features** . . . 40
5.3 **Discussion** . . . 42
5.3.1 **Discussion: Synthetic Views** . . . 43
5.3.2 **Discussion: Prioritized Features** . . . 46
5.3.3 **Synthetic Views vs Prioritized Features** . . . 48
5.4 **Summary** . . . 48
6 CONCLUSION AND FUTURE WORK . . . 50
6.1 **Future Work** . . . 50
6.1.1 **SIFT descriptor accuracy** . . . 51
6.1.2 **Refine 3D models** . . . 51
6.1.3 **Photographs Focal Length and Distortion** . . . 52
6.1.4 **Vocabulary Tree with GPU scoring functions** . . . 52
6.1.5 **Co-occurence Sampling** . . . 52
6.1.6 **Larger Database** . . . 53
6.1.7 **Updating SFM Models** . . . 53
6.1.8 **Image Geocoding as Service** . . . 53

LIST OF FIGURES

Figure 1 Example of a 3D structure reconstructed with a sequence of photographs. Image adapted from Enqvist et al. (2011). 11
Figure 2 Sequential diagram of the main tasks performed by the Structure from Motion algorithm. 11
Figure 3 Example of keypoints extracted on Santa Luzia Church in Viana do Castelo (at the right). 12
Figure 4 Example of a vocabulary structure. 13
Figure 5 Example of a vocabulary tree filled with documents. 14
Figure 6 Matching of keypoints between two different images of Santa Luzia Church. 16
Figure 7 Example of an uncalibrated (left) and calibrated (right) image of a chessboard pattern. As we can see, the curves on the left were adjusted to look like lines on the right. 20
Figure 8 Example of two reconstructions where all the photograph were taken at the same altitude. The coloured polygons represent the camera positions. The reconstruction on the left is made without bundle adjustment, and the one on the right is optimized with bundle adjustment. 20
Figure 9 Relation between the information stored for a structure from motion model. 21
Figure 10 Amount of features retrieved from an outdoor photograph (about 10000+ on the left) and an indoor photograph (1621 on the right). 24
Figure 11 Structure of the generated 3D documents using Synthetic Views. 32
Figure 12 Structure of the generated clouds using Prioritized Features. 33
Figure 13 A sample of indoor (top) and outdoor (bottom) photographs from our data set. 37
Figure 14 Sparse reconstructions from our data set visualized with VisualSFM. The center and left reconstruction belong to indoor environments and the right one belongs to outdoor. 37
Figure 15 An image from our data set (left) and a query photograph (right). Although the environment is clearly distinct, similar objects such as the extinguishing hose box may be found. These similarities may confuse the vocabulary top matches retrieved. 44

LIST OF TABLES

Table 1 Compression rate performed by Synthetic Views when using the mean of the descriptors for indoor models. 39
Table 2 Compression rate performed by Synthetic Views when using the mean of the descriptors for outdoor models. 39
Table 3 Overall statistics from pose estimating 25 indoor photographs of resolution 3000 and 1000 with Synthetic Views. 39
Table 4 Overall statistics from pose estimating 25 outdoor photographs of resolution 3000 and 1000 with Synthetic Views. 40
Table 5 Mean time spent by each operation made by Synthetic Views when geocoding indoor photographs. 40
Table 6 Mean time spent by each operation made by Synthetic Views when geocoding outdoor photographs. 40
Table 7 Compression rate performed by Prioritized Features when using the mean of the descriptors for indoor models. 41
Table 8 Compression rate performed by Prioritized Features when using the mean of the descriptors for outdoor models. 41
Table 9 Overall statistics from pose estimating 25 indoor photographs of resolution 3000 and 1000 with Prioritized Features. 41
Table 10 Overall statistics from pose estimating 25 outdoor photographs of resolution 3000 and 1000 with Prioritized Features. 42
Table 11 Mean time spent by each operation made by Prioritized Features when geocoding indoor photographs. 42
Table 12 Mean time spent by each operation made by Prioritized Features when geocoding outdoor photographs. 42

Part I

INTRODUCTORY MATERIAL

1

INTRODUCTION

Used for civil, commercial and military purposes, the Global Positioning System (GPS) has proven to be a resourceful and useful service. By using 4 or more satellites, this system uses a trilateration process which allows world wide location recognition. But since this process requires the communication between the subject to be located and the satellites, occluded zones (by bad weather or buildings) often hinder this communication and render the GPS ineffective.

To complement the GPS in these zones, several Indoor Positioning System (IPS) solutions were researched. These solutions were evaluated in several performance metrics were the most relevant are: accuracy, precision, complexity, robustness, scalability and equipment cost. Presently there is not an official IPS because the existing solutions do not balance these metrics, where the cost/accuracy relation is often inversely proportional.

Further tackling the indoor localization problem, the Computer Vision community has been supporting the use of images as a mean to recognize locations. This system uses a database of pre-registered photographs, and based on the visual similarity of new photographs to the database, a location is returned. The clear advantage of image geocoding is not requiring additional infrastructure to deploy the system, which greatly reduces the cost of solutions. Simple photographs taken from cheap cellphone cameras can be used. The second clear advantage of this method is the ability to compute the orientation along with the position. While other methods may achieve this by performing time lapse measurement, image geocoding can compute both on a single query.

But most of the image location recognition systems developed were only tested on outdoor environments, where the GPS signal is strong the majority of time. Besides the implementation of their complete pipeline is not publicly available.

So, to offer continuity to this research, we were motivated to develop a prototype which uses existing Computer Vision image localization methods to perform image geocoding on occluded zones, without any prior information of where photographs were taken.

In this thesis we want to prove that already existing methods can be applied into indoor with just few modifications. By developing a prototype, we will offer the necessary tools for the Computer Vision community to experiment and improve this research.

1.1 STATE OF THE ART

In this section we present the current state of art on proposed localization systems. First we address researched methods which are not related to Computer Vision to justify the shared disadvantage of these systems. And then, image base localization methods are presented, briefly describing each work to re-enforce the utility of image based localization.

UltraWide Band Waves

By using Ultrasonic waves, Minami et al. (2004); Priyantha (2005); Hazas and Hopper (2006) proposed solutions which use these high accuracy waves for indoor localization. Although the precision is often high, an expensive infrastructure is needed, which limits this system to smalls areas where it can reach its peak of accuracy to compensate the high cost.

Inertial Measurement Units

Taking advantage of Inertial Measurement Unit's (IMU) device, Woodman and Harle (2008); Patrick et al. (2009) proposed an indoor location system using this measurement device. The IMU are known for high drifting errors which are accumulated over time. Although the accuracy of these systems is limited, the infrastructure cost is of little expenses. To compensate errors, other research's propose to use a combined IMU system with RFID's Ruiz et al. (2012) or Ultra-Wideband measurements Hol et al. (2009), but using adicional hardware increases the expenses in deploying and maintaining the system.

Radio Frequency

Due to high range coverage and low cost hardware usage, Hightower et al. (2000); NI et al. (2003); Zhang et al. (2010); Das and Agrawal (2014) proposed IPS solutions which use radio frequencies for location recognition. Subjects to be located are required to carry a small RFID device which acts as a receiver of tracking information. Although RFID solutions are able to compute positions on indoor environments, radio frequencies are affected by signal interference caused by infrastructures.

Infrared

In Mao et al. (2013); Jung et al. (2014), solutions for an indoor location system using a infrared system were proposed. Subjects to be located are required to use an infrared device which periodically sends information to infrared sensors positioned along the building structure. The nearby sensors which are able to capture the messages, consequently compute the current position of users.

Image Based Localization

In Computer Vision, the problem of location recognition as been addressed by a variety of approaches. The basic idea behind the researched methods is to compute the position of a query photograph with respect to a database of registered reference images or 3D models.

Early work started with Ravi et al. (2006), where it was developed an indoor location recognition system using mobile phones. Photographs taken are uploaded into a server, and using off the shelf algorithms such as Color Histograms Niblack and Barber (1993); Vellaikal and Kuo (1995), Wavelet Decomposition Jacobs et al. (1995) and Shape Matching Kato et al. (1992), photographs are compared to a geocoded image database to return a pose. Also using a database of geocoded photographs, Kawaji et al. used a combination of PCA-SIFT Ke and Sukthankar (2004) features with a Locality Sensitive Hashing Datar et al. (2004) searching algorithm to query the position of new photographs on large databases.

Based on a combination of an image database and Structure from motion point clouds, Huitl et al. (2012) developed an application which uses the algorithm described in Schroth et al. (2011) to compute the location of new indoor photographs. New photographs are sent to a server and queried to several vocabulary trees (where each represents an area) which efficiently stores database images. From these vocabulary trees, the top most similar documents are retrieved and matched to the query photographs. The location is computed on a successful match.

Regarding research's on outdoor image location recognition, other relevant researches such as Schindler et al. (2007) developed a city scale location recognition which also uses a trained vocabulary tree along with the Scale Invariant Feature Feature Transform (SIFT) Lowe (2004) descriptors to store a database of registered images. Being aware of the heavy weight of the SIFT descriptors, later Fraundorfer et al. (2008) also proposed a vocabulary tree based location recognition but with the compressed PCA-SIFT Ke and Sukthankar (2004) descriptors to refine the matching new images.

Irschara et al. (2009) devised Synthetic Views, an algorithm based on structure from motion models which allows fast location recognition at large scale databases. Environments to be stored in their database are first reconstructed into 3D models. Models are then compressed in order to remove unnecessary information, and their related information stored within a vocabulary tree. New photographs are queried to the vocabulary tree which in sequence, delivers enough information to compute the pose of new photographs (if there is related visual information). With this algorithm, they achieve real time location recognition in a database of 1054 photographs.

Also based on structure from motion models, Li et al. (2010, 2012) devised a location recognition algorithm using Prioritized Features. Their research is divided into two papers. In their first paper Li et al. (2010), they have proposed the prioritization of relevant features on 3D models to remove irrelevant visual information, and allow faster queries of new photographs on large databases. Here, they studied and compared the standard matching of query photographs to point clouds and the inverse operation. Both advantages and disadvantages of these matching methods were carefully analysed and later in Li et al. (2012) they propose a conjugation of both matching methods, devising this way, the

novel Bidirectional Matching scheme. On a world scale outdoor database of 3D models, they achieve successful position and orientation estimation in few seconds per query.

1.2 RESEARCH HYPOTHESES

By analysing the state of art on Indoor Positioning Systems, it is noticeable that these systems either compute the indoor position with high accuracy errors or require additional hardware to attenuate these errors. In contrast to these research's, image based location recognition methods only require photographs taken from a cheap camera to compute the associated location. Besides, orientation can also be retrieved on a single query without performing time lapse of the computed locations.

Although present proposed algorithms allow image geocoding within indoor environments, these algorithms rely on heavy databases of geocoded photographs and models to retrieve the location of new photographs. Thus, in this thesis we question the possibility of using Synthetic Views and Prioritized Features algorithms to address the scalability of the geocoded database required by indoor image based localization. Assuming that the compression of a geocoded database will remove information to pose estimate new photographs we then question how much precision and speed can we get from the location estimation, and if the balance between these metrics is achievable.

1.3 METHODOLOGY

We believe that the development of a prototype will help us answer our research questions. This prototype will be based on the image location recognition methods, Synthetic Views and Prioritized Features, two Structure from Motion based methods which were successful on geocoding outdoor photographs.

We will start by exploring the Structure from Motion algorithm, understanding each step performed from receiving photographs to build 3D models and position photographs relatively to each other. From here, we will develop a process to relate the model coordinate system with the respective GPS coordinate. This will allow automatic geocoding when adding new photographs to a model.

Because 3D Structure from Motion models will be needed to support our geocoding process, and their construction is time expensive, we will search for available indoor and outdoor models. If models are not found, then we will gather a data set of geocoded photographs to build our own models using existing Structure from Motion reconstruction software.

Then we will proceed to the full implementation of Synthetic Views and Prioritized Features, since there is not available code on their complete pipeline. Both methods will be divided in two stages:

- an offline stage where models reconstructed are geocoded by the GPS coordinates associated to the photographs, compressed into a smaller yet representative set of information and added to a database of 3D models.

- an online stage where new photographs are compared with the database models to retrieve their relative position and further compute their GPS coordinates and direction.

Here, we will learn how to use vocabulary trees Nist and Stew (2006) and Approximate Nearest Neighbors search algorithms Indyk and Motwani (1998), which allow scalable object recognition and use efficient software Wu (a) which benefit the GPU processor to speed heavy matrix operations.

After developing the Synthetic Views and Prioritized Features prototype, we will evaluate their view registration performance. New indoor and outdoor photographs will be geocoded with both methods and the geocoding rate, speed and precision will be registered. A careful analysis on experimental results from each method individually will validate their use on performing indoor image gecoding. Then a final comparison between both methods will define which method is advised for a quick and accurate location recognition on indoor environments.

1.4 DOCUMENT STRUCTURE

This thesis is divided into 5 chapters:

- **Chapter 2** describes the Structure from Motion process, unveiling the mechanism behind the 3D model acquisition using photographs. Then we present available software which perform Structure from Motion.
- **Chapter 3** will address the indoor model reconstruction and its difficulties. Here we distinguish and debate differences between 3D structure from motion outdoor models with indoor.
- **Chapter 4** offer an overview of both Synthetic Views and Prioritized Features algorithms for fast location recognition using structure from motion models. Problems related to the scalability and accuracy are exposed and solved using these methods. For each method, we explain their implementation and adaptation to indoor models;
- **Chapter 5** identifies the data set gathered as well the 3D models reconstructed to test the veracity of the proposed methods to solve the indoor positioning system problem. Experimental results are presented and debated individually for each method. Then a comparison between both methods is provided.
- **Chapter 6** concludes this thesis with a summary of the outcome of our research. Ideas for future work are then presented.

1.5 SUMMARY

In this chapter it was introduced the main theme of this thesis. A contextualization was given to expose the Indoor Positioning System problem and relate it with Computer Vision research. Then the

current state of art was presented, where we offered a brief overview of researched methods which use infrared rays, inertial measurement units, ultrasonic waves, radio frequencies and images to perform location recognition.

Based on state of art research, we formulated our hypotheses where we questioned the possibility of efficiently performing indoor image geocoding using the algorithms Synthetic Views and Prioritized Features, while addressing the compression of the required image/model geocoded database. We also questioned if a balance between speed and accuracy is achievable with this solution due to the compression process done by both methods.

The methodology which will allow us answer our hypotheses was then presented. Here we anticipate the implementation of Synthetic Views and Prioritized Features algorithms.

And before closing this chapter we presented the structure of this thesis.

2

STRUCTURE FROM MOTION

As we are required to explore the Structure from Motion algorithm, in this chapter we will present the explanation of each step performed by this algorithm. We will start by introducing the concept of Structure from Motion followed by the composition of its algorithm. Then a detailed explanation of the core steps will be offered. Here we will approach the feature extraction and matching process, geometric projections and the bundle adjustment optimization. Before closing the chapter, we will present the typical output of 3D Structure from Motion models and afterwards we will indicate available software which may be used to construct these models.

2.1 CONCEPT

Structure from Motion consists in a method for estimation the three-dimensional structure from two dimensional image sequences. As humans precept their surroundings over-time, this algorithm does the same by computing the "motion" between related images and use it to position them relatively to each other. Then it projects common visual information found on each image into a single coordinate system in order to build a 3D model of the scene.

2.2 ALGORITHM

Generally, the Structure from Motion is composed by the following steps:

- **Feature Extraction** - Extraction of distinctive points which best define an image;
- **Coarse Matching** - Infer an initial subset of image pairs to match;
- **Detailed Matching** - Establish correspondences between the chosen pairs;
- **Geometric Verification** - Establish the geometry correlation between stable pairs;
- **Projection** - Project 2D information into the 3D coordinate system;
- **Bundle Adjustment** - Optimize projections to reduce positioning errors;

Figure 1: Example of a 3D structure reconstructed with a sequence of photographs. Image adapted from Enqvist et al. (2011).

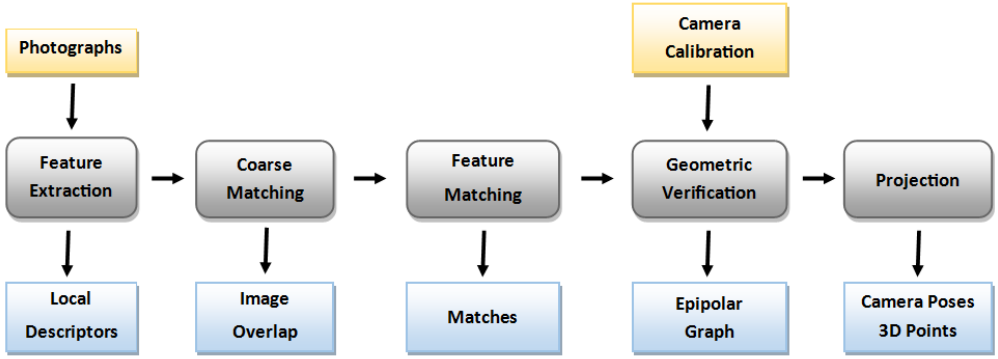

Figure 2: Sequential diagram of the main tasks performed by the Structure from Motion algorithm.

2.3 FEATURE EXTRACTION

To build a 3D model, the Structure from Motion algorithm needs to track similarities between photographs in order to relate them. The tracking process is based on the concept of feature. In Computer Vision, a feature is a piece of visual information which best defines an image region. This piece of information is detected by searching the image for edges, corners, blobs or ridges. At each feature extracted, a keypoint (location of the feature) and a descriptor (characterization of the appearance of the feature) are defined. Over the past decades a variety of feature detectors and descriptors were proposed and extensive evaluation Mikolajczyk and Schmid (2005) on points of interest and descriptors shows that the Scale Invariant Feature Transform (SIFT) Lowe (2004) and Speeded Up Robust Features (SURF) Bay et al. (2006) are among the top performing in terms of accuracy. These features

are scale and rotation invariant and robust against illumination changes which makes them suitable for image matching.

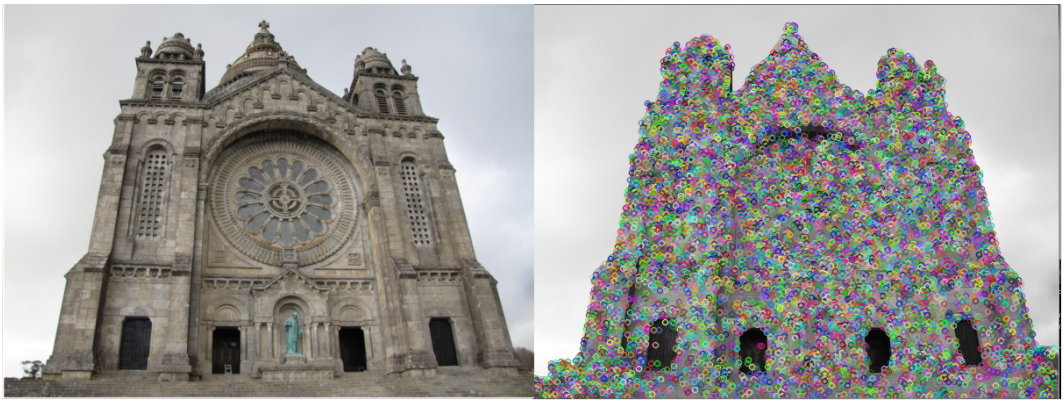

Figure 3: Example of keypoints extracted on Santa Luzia Church in Viana do Castelo (at the right).

Usually an image feature which is scale and rotation invariant is composed by:

- a 2D position, which locates the center of the feature patch within the photograph;
- a descriptor, which characterizes the feature patch;
- a scale, which defines the level of visibility of a feature;
- and an orientation, which defines the orientation of the feature;

2.4 COARSE MATCHING

After extracting features for the entire set of photographs, the idea now is to define correspondences between them. But as this process is often heavy since it needs to compare every pair of photographs, it is wise to define a set of potential images to match instead of blindly matching them all. This selection process is called Coarse Matching. On the following subsections we address the very efficient vocabulary trees and preemptive feature matching to reducing the amount of matches to perform.

2.4.1 *Vocabulary Trees*

In Computer Vision, a vocabulary tree is a structure which allows the fast and accurate object recognition. This structure is based on the concept of bag-of-words Sivic and Zisserman (2003), which describes local descriptors as a set of words. Images are directly described based on a histogram of quantized feature occurrences with respect to a code book of pre-generated visual words. When quering new photograph, this histogram is then used to define a similarity between query and database photographs. The advantages of using this structure are:

- High dimension descriptors are quantized into integers (compactness);
- Fast queries (efficiency);
- Adding and removing new photographs on the fly (flexibility);
- Benefits from large image database, due to descriptor variety (scalability).

The explanation of the next subsections are heavily influenced by the explanation of the vocabulary tree structure provided in (Irschara, 2012, p. 24-33)

Building a Vocabulary

The first step to build a vocabulary tree is to generate a code book. This code book is built based on a large training set of rich descriptors which allow the generation of distinctive words. The idea is to clusterize the set of descriptors and each cluster center will represent a tree node. Initially, the set of descriptors are partitioned into K clusters where K is the branching factor of the tree. The descriptor closest to each cluster center will represent a tree node. This process is then repeated $L - 1$ times for each cluster previously created, where L is the number of levels of the tree.

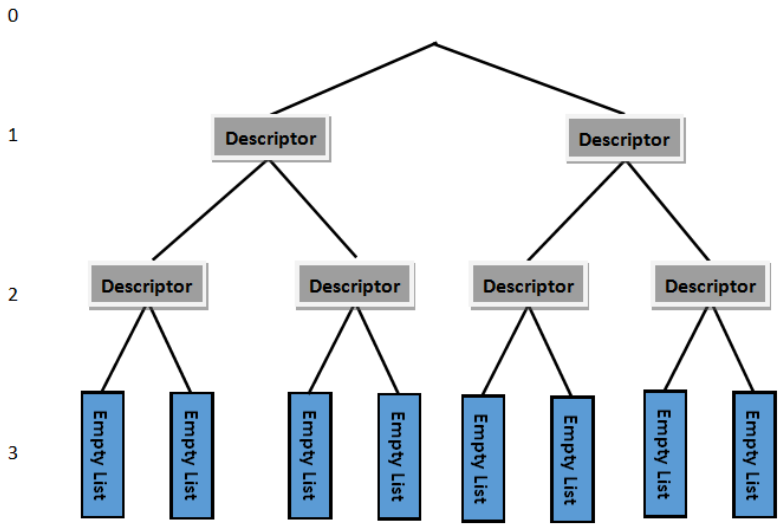

Figure 4: Example of a vocabulary structure.

Based on this representation, the memory usage of this structure is linear to the number of leaf nodes K^L.

Inverted Files

With the code-book generated, photographs may be described as a set of visual words. The descriptors of these photographs are propagated through the tree by comparing the query descriptors with the

K candidate cluster centers, and choosing the closest one. Each time a descriptors reaches a bottom node of the tree, the identifier of the photograph which the descriptors belongs and the frequency of the visual word throughout the tree are stored to compare this photograph with further query photographs.

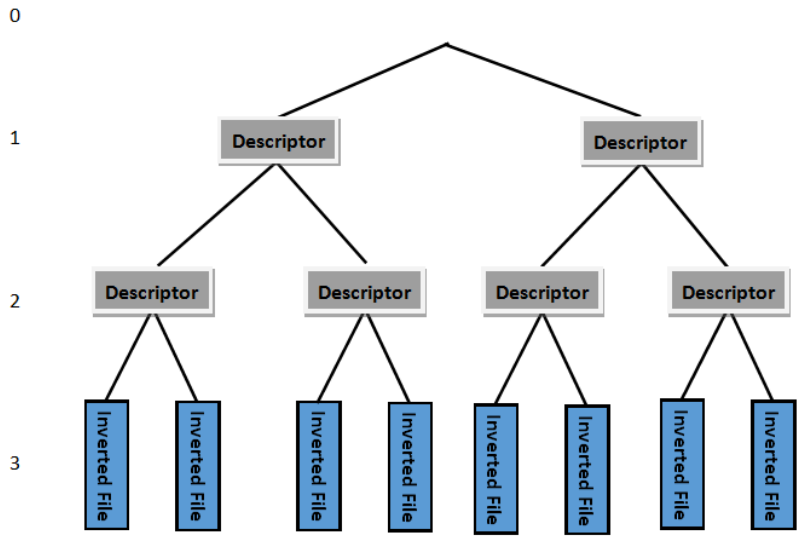

Figure 5: Example of a vocabulary tree filled with documents.

With this scheme, descriptors of new photographs are not required to be stored, and the K^L descriptors is rather appealing compared to saving every single descriptor.

Scoring Function

After building the code-book and insert a set of photograph within the structure, the idea now is to determine the relevance of new photographs to the database. The relevance is a factor defined by how similar the paths of the descriptors of query photographs are to the images stored within the structure. The similarity factor is returned by a scoring function and we show the most used.

Scoring Function: Jaccard Scoring

The Jaccard scoring or Jaccard similarity coefficient is a statistic metric used for comparing the similarity and diversity of sample sets. The metric itself is defined by the division between the intersection with the union of two sample sets,

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

where A and B are binary occurrences of visual words in image i and j, respectively.

Scoring Function: TF-IDF Scoring

Term Frequency-Inverse document frequency is a common scoring function which weights the relevance of a visual word relatively to a collection of words. As the name suggests, this metric is computed by the product of two statistics: term frequency and the inverse document frequency. Let V be a vocabulary of visual words. Then each document is represented by a vector,

$$v_d = (t_1, ..., t_i, t_|V|) \tag{2}$$

of tf-idf weighted word frequencies with components,

$$t_i = \frac{n_i d}{n_d} log \frac{N}{n_i} \tag{3}$$

where $n_i d$ is the number of occurrences of word i in document d, n_d is the total number of the document d, n_i is the number of documents containing the term i and N the number of documents in the whole database. Given two tf-idf vectors v_1 and v_2, the similarity is returned by,

$$cos(o) = \frac{v_1 v_2}{||v_1||||v_2||} \tag{4}$$

which is then ranked into three levels: 1 meaning that the documents are exactly the same, 0 independent and -1 exactly opposite.

Scoring Function: Probabilistic scoring

Probabilistic scoring is a scoring function used on Irschara et al. (2009) work. Here, they define two metrics, R and $\bar{R}$ where the first corresponds to the set of documents which overlap the query image and the second the set of documents which do not share any visual information. The similarity score of a query image is computed based on the probability $P(v_i|R)$ of visual words v_i being present on images randomly selected from a set R of relevant images to the query, and the probability $P(\bar{v}_i|R)$ of visual words not being present on images with potential overlap to the query.

$$sim(D_j, Q) \frac{(\prod_{g_i(D_j)=1} P(v_i|R)) \times (\prod_{g_i(D_j)=0} P(\bar{v}_i|R))}{(\prod_{g_i(D_j)=1} P(v_i|\bar{R})) \times (\prod_{g_i(D_j)=0} P(\bar{v}_i|\bar{R}))} \tag{5}$$

2.4.2 *Preemptive Feature Matching*

This coarse matching method was introduced by (Wu, 2013, p. 2). Its basis lies on the fact that feature scales represent the visibility of keypoints, where features with high scales are defined as popular points (points which have an high probability of appearing in other photographs). This method first sorts keypoints of photographs to match by decreasing order, based on their scale. It then sets a threshold of h keypoints. If from h most visible keypoints, at least h_t successfully match, then the full set of keypoints should be matched. Else, the pair of photographs is defined as a false match and

feature matching is not applied. If a low enough h value is applied, such as 100 keypoints, the coarse matching process is applied in an instant. Without requiring additional structures, this method was able to eliminate up to 75-95% of false matches.

2.5 FEATURE MATCHING

Feature matching is responsible for establishing correspondences of photograph pairs. Correspondences are then used to track the motion between the entire set of photographs.

Figure 6: Matching of keypoints between two different images of Santa Luzia Church.

Feature matching comes from the Nearest Neighbors optimization problem where the goal is to find a point of d-dimensionality closest to a query point. As this search is slow in high dimensional spaces such as 128-SIFT descriptor space, several approaches such as kd-randomized tree Silpa-Anan and Hartley (2008) with priority search and the hierarchical k-mean tree Fukunaga and Narendra (1975) were proposed, where they provide speedups of about one or two orders of magnitude over the linear search on a single CPU. But this speed up comes with a cost of accuracy.

More recent approaches use graphic processing unit to perform feature matching faster than approximate nearest search and with the same accuracy. The core of these approaches comes from the Euclidean distance, which defines the similarity between two descriptors vectors x and $\hat{y}$ on K-dimensional space, being computed as,

$$||f - \hat{f}||_2^2 = 2 - 2\sum_{i=1}^{K} f_i.\hat{f}_i \tag{6}$$

And so, the matching between several descriptor vectors can be represented as a dense matrix multiplication,

$$\begin{matrix} f_1 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ f_N \end{matrix} \begin{pmatrix} d_{1,1} & \cdots & d_{1,K} \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ d_{N,1} & \cdots & d_{N,K} \end{pmatrix} \begin{pmatrix} d_{1,1} & \cdots & \cdots & d_{1,M} \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ d_{K,1} & \cdots & \cdots & d_{K,M} \end{pmatrix} = \begin{pmatrix} c_{1,1} & \cdots & \cdots & \cdots & c_{1,K} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ c_{N,1} & \cdots & \cdots & \cdots & c_{N,M} \end{pmatrix} \tag{7}$$

Here, the library CUBLAS NVidia, which specializes in algebra operations on GPU can be used to efficiently solve this multiplication. The table provided by Irschara PhD thesis (Irschara, 2012, p. 34) presents a comparison between the computational speed by the CPU and GPU implementation of feature matching on a Pentium 3.2 GHz CPU with a nVidia GTX 280, where it is proven that GPU matching is faster than CPU implementations.

For the dense matrix multiplication, the following measure considering the first and second closest neighbors can be used to determine the acceptance of matches,

$$\frac{d(f, f_{1st})}{d(f, f_{2nd})} < t_r \tag{8}$$

where t_r is a fixed threshold usually set to 0.8 as suggested by Lowe (2004). The explanation of this section was also influenced by the detailed overview of feature matching found at (Irschara, 2012, p. 33-35)

2.6 GEOMETRIC VERIFICATION AND PROJECTION

After establishing the motion between images, the Structure from Motion algorithm is ready to start the reconstruction.

For each pair of photograph successfully matched, common keypoints are projected into an individual coordinate system where projected points are triangulated to the respective camera positions. The output of all individual projections is denoted as an epipolar graph, a structure composed of vertexes $V = V_1 ... V_N$ corresponding to the photographs and a set of edges $E = e_{ij} | i, j \in V$ which are pairwise reconstructions. From here, a baseline pair is chosen (usually the pair with most features projected) to start an incremental reconstruction. The remaining pairs are iteratively added to the baseline reconstruction by using the epipolar graph correspondences.

The next subsection will give a brief explanation on the feature projection and triangulation between images.

2.6.1 *Projection Matrix*

So, to reconstruct a 3D model out of 2D images, 2D points must be projected into a 3D referential. A 2D point x from an image will be mapped to the 3D position X by a projection matrix. This matrix is called Camera Matrix and we will name it the P matrix.

$$x = P.X \tag{9}$$

Theoretically what is expected is that this projection matrix locates the position of the camera which shot the image, from that position compute the pixel positions and then, from the center of the camera, project pixels into a 3D referential. So, this matrix may be defined by two separate matrices: the intrinsic and extrinsic camera matrices. The intrinsic camera matrix (K) represents the pixels position relatively to the camera position while the extrinsic camera matrix (RT) relates the camera position and orientation in the real world.

The intrinsic matrix contains the internal parameters of the camera and is represented as the following.

$$K = \begin{bmatrix} fx & s & cx \\ 0 & fy & cy \\ 0 & 0 & 1 \end{bmatrix}$$

The fx and fy are the respective focal length for the camera in x and y axis. The two axes are distinguished since the presumption of non square pixels is assumed. The parameter s is the skew factor and cx and cy the center of the image (the center of the image needs to follow the distortion of the image). The skew factor is the angle in which the pixels slope. Several methods are proposed to compute the parameters of this matrix Zhang and Kang (2004).

As for the extrinsic matrix, a bit more work is needed. Since all the reconstruction is made in a fictional coordinate system, it doesn't matter if the first camera reconstructed is placed in a random position with a random orientation, as long as those parameters are correctly mapped to the real world position. The problem comes with the following cameras, as the idea is to preserve the distance and orientation of the several cameras processed. So, in order to compute the extrinsic matrix, we need to find the epipolar geometry between images.

The epipolar geometry is the intrinsic projection between two views and may be represented by the Fundamental or Essential matrix. Starting by the Fundamental matrix (F) (Hartley and Zisserman, 2004, p. 239-256), this matrix relates the intrinsic projection of common points between two uncal-

ibrated images. If a 3D point X is represented as x in the first image and x' in the second, then the points from both views need to satisfy the relation $x'^T Fx = 0$.

The essential matrix (E) (Hartley and Zisserman, 2004, p. 257-260) does the same, but for calibrated images.

Since it is preferable to 3D reconstruct with calibrated cameras, the essential matrix is computed by the following equation, where K and K are the intrinsic matrices from the pair of cameras belonging to both images to triangulate.

$$E = K'^T.F.K$$

The essential matrix is then used to compute the rotation and translation matrices, through the method of Singular Value Decomposition (Hartley and Zisserman, 2004, p. 257-260).

$$R = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}$$

$$T = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix}$$

These two matrices represent the orientation and location on the extrinsic matrix.

$$R.T = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_1 \\ R_{21} & R_{22} & R_{23} & T_2 \\ R_{31} & R_{32} & R_{33} & T_3 \end{bmatrix}$$

But up until now it was assumed that there are no distortions. Typically, when photographs are taken with cheap cameras, from certain angles one may notice that lines from the real world are represented as curves on the image. This effect is caused by non-planar lens surfaces.

The distortion is represented as an array containing coefficients (radial and tangential distortion) and can be obtained through Zhang and Kang (2004). Basically what these distortion coefficients do is help correcting the position of 2D points on images to optimize the projection of those points into a 3D coordinate system. A distorted 2D point x is now represented as the undistorted $x_{undistort} = x + \delta$ where δ is the distortion correction of x.

2.7 BUNDLE ADJUSTMENT

Although the camera calibration is useful to adjust the cameras to the real world, the visual projection of information is merely an approximation to the reality, which means that the reconstruction may

Figure 7: Example of an uncalibrated (left) and calibrated (right) image of a chessboard pattern. As we can see, the curves on the left were adjusted to look like lines on the right.

accumulate errors. The process of computing all the cameras locations, orientations and the point projections with distorted lens to the same coordinate system, contains a lot of variables and can easily create distorted reconstructions. Bundle Adjustment consists in an optimization problem which aims to minimize the projection error of between the 2D feature locations with the predicted image measurements of 3D points. This error is computed by analysing each 3D point projected by 2D features shot from cameras with a given pose and internal calibration parameters (Triggs et al., 2000, p. 302).

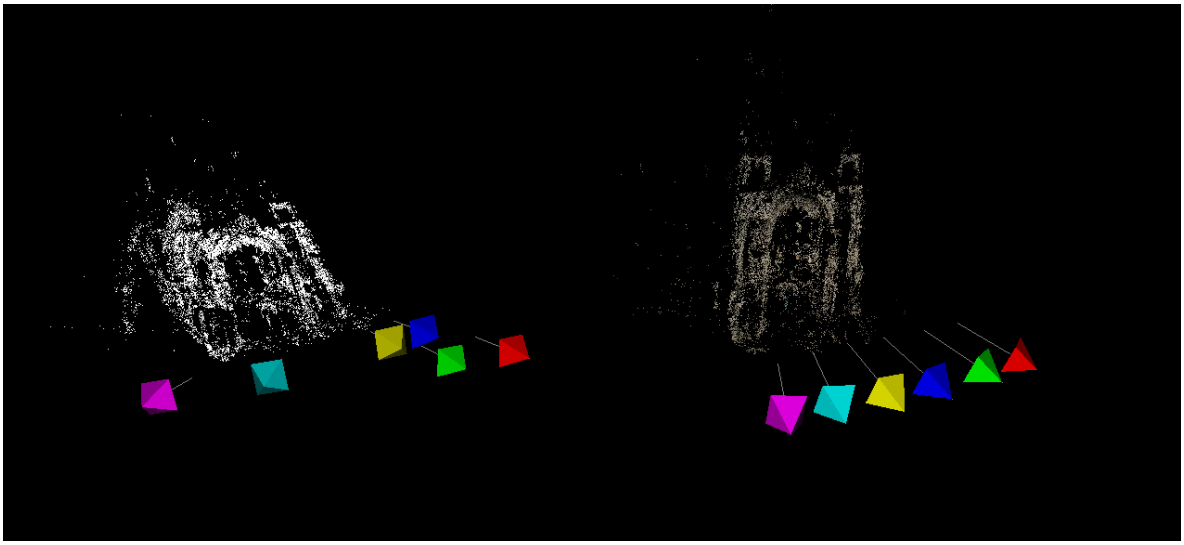

Figure 8: Example of two reconstructions where all the photograph were taken at the same altitude. The coloured polygons represent the camera positions. The reconstruction on the left is made without bundle adjustment, and the one on the right is optimized with bundle adjustment.

Although this optimization improves the accuracy of the reconstruction, it poses for computational speed and memory consumption as the optimization problem grows proportionally to the number of projections to optimize. Being aware of this trade off, Agarwal et al. (2010) described a new Inexact Newton type of bundle adjustment which uses substantially less time and memory than the standard bundle adjustment algorithm without compromising the quality of the solution. Also, by taking advantage of the GPU/CPU parallelism of recent computer machines, Wu et al. (2011) proposed the use of a bundle adjustment algorithm which exploits this parallelism to efficiently optimize large scale 3D scene reconstructions. Their research shows that this algorithm executes 10 times faster on CPU and 30 times faster on GPU than the current state of the art methods. Recent work made by (Wu, 2013, p. 4) states that running bundle adjustment sporadically (each time the reconstruction grows a certain ratio) is enough to speed the optimization while maintaining consistency in incremental reconstruction.

2.8 3D SCENE REPRESENTATION

Typically, a Structure from Motion model follows the structure presented on Figure 9, where each 3D model is composed by a point cloud of 3D points. The 3D points have one or more image features which projected them. As explained in section 2.3, image features are composed by a 2D position, descriptor, scale and orientation.

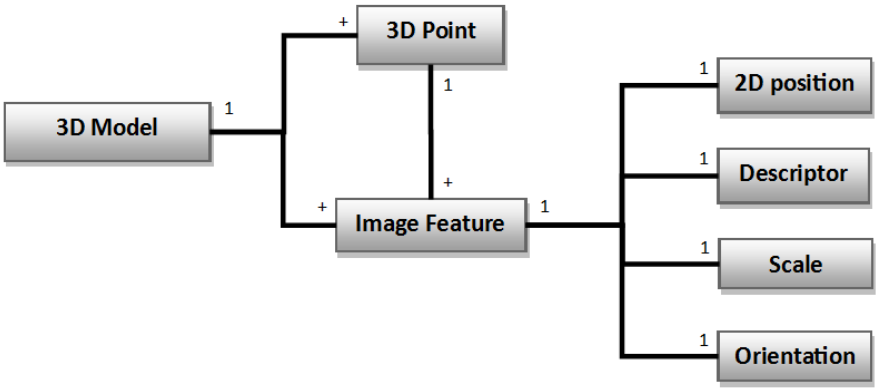

Figure 9: Relation between the information stored for a structure from motion model.

2.9 STRUCTURE FROM MOTION SOFTWARE'S

Presently, there are several free and non free available software's which perform structure from motion to reconstruct 3D model from simple photographs. From those current available, the most relevant are: AgiSoft PhotoScan (non-free) AgiSoft (2006), Smart3DCapture (non-free) Acute3D (2011), Bundler

(free) Snavely (2006), VisualSFM (free) Wu (b), 3DF Samantha (free) 3DFLOW, SFMToolkit (free) Henri. From the available free to use, we highlight VisualSFM for the speed on processing incremental structure from motions. It uses a combination of SiftGPU Wu (a) and PBA Wu et al. (2011) which both exploit the GPU to fasten heavy matrix operations.

2.10 SUMMARY

In this chapter the main components of the structure from motion were described and the current state of art on solving each step was provided. We started by introducing the concept of feature keypoints and descriptors and their uses on computer vision. Here we highlighted SIFT for being one of the most consistent feature processors available. Since feature matching is an heavy matrix operation, we pointed the efficient vocabulary trees and preemptive feature matching to reduce the number of image matches to perform while avoiding removing positive matches. Then we described feature matching as a dense matrix operation which can be computed on GPU to deliver correspondences between images. Afterwards, the projection and triangulation of these correspondences were explained. Since the projection stage poses for geometric errors, bundle adjustment was then addressed where we indicated a GPU optimization which performs 30 times faster than standard bundle adjustments. At the end, we presented the typical output of structure from motion and then we mention available Structure from Motion software's.

Part II

CORE OF THE DISSERTATION

3

FROM OUTDOOR TO INDOOR REPRESENTATIONS

As explained in Chapter 2, the structure from motion pipeline allows the pose estimation of new photographs as long as these photographs are related to the reconstructed models. Provided that this reconstruction algorithm is also applicable to indoor environments, then we may assume that indoor image location recognition is also possible. In this chapter we address the obstacles on generating indoor 3D models with structure from motion.

3.1 CHALLENGES ON BUILDING INDOOR SFM MODELS

The structure from motion reconstruction depends highly on an input of distinctive photographs, with overlap to each other in order to be related and compared. The higher the number of features shared among photographs, the more consistent and accurate the 3D models generated will be. Outdoor environments often contain an high amount of edges and ridges where features accumulate, and thus photographs on this environments are descriptive. Opposed to outdoor, indoor photographs are less descriptive due to blank walls and areas with few decoration, as shown in Figure 10. The relation of features extracted on outdoor to indoor reaches the five times features decrease on our set of photographs.

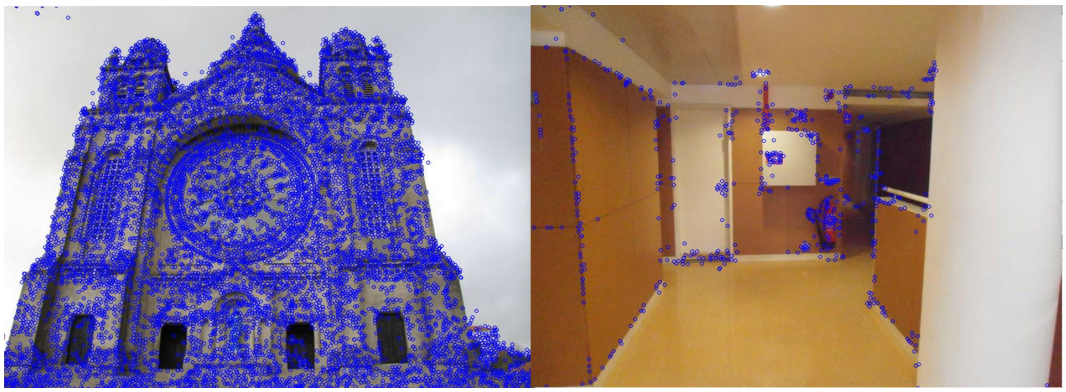

Figure 10: Amount of features retrieved from an outdoor photograph (about 10000+ on the left) and an indoor photograph (1621 on the right).

As the number of features per indoor photograph is lower, a strategy to increase the number per photograph is to take photographs on points of view which allows the capture of a wide environmental area. But rather than outdoor which contains wide streets and plazas, indoor environments are narrow and contain a complex structure which prevents good wide area photographs.

Also, the illumination exposure within indoor is weaker than outdoor. Although the SIFT features contain a certain level of illumination robustness, this robustness is limited which compromises the effectiveness of feature extraction and matching. Areas with poor illumination may create artifacts which generate erroneous features.

Due to these facts, after we attempt reconstructing indoor environments, we noticed that:

- either the indoor models can be built with few photographs, where few popular features were shared by photographs. The overall number of projected points was low;
- popular features were not shared throughout the photographs and a single model is partitioned into three or more sub-models.

Both cases are still usable to compute the position and orientation of new photographs. But practically, with the latest case, each sub-model may contain different levels of drifting (error acumulated from structure from motion estimations). Computing the pose on these sub-models may deliver different levels of precision even when belonging to the same environment.

Regardless, indoor model reconstruction is possible and new photographs can be pose estimated with structure from motion with respect to a single model.

3.2 SUMMARY

In this chapter we presented the challenges on indoor model reconstruction. Here it was identified that indoor photographs are less descriptive than outdoor, narrow areas hinder good base line photographs to begin an incremental reconstruction, and illumination changes may create artefact's which reduce the effectiveness of feature extraction and matching. Two reconstruction cases were defined where either models were reconstructed with a low number of point clouds, or indoor models were partitioned into several sub-models due to the lack of correspondences. Despite these problems, structure from motion is applicable to indoor and new photographs can be pose estimated to a single model.

4

IMAGE BASED LOCALIZATION

In the previous chapter it was proved that structure from motion can be used to build 3D indoor models. New photographs can be pose estimated, but only individually to each model built. Besides, the output data of structure from motion do not contain any geographic information which allows the calculus of the geographic coordinates of new photographs. Thus, in this chapter we present a simple solution to geocode structure from motion models based on their associated photographs. Then, we address two image location recognition methods which allows the pose estimation of new photographs with respect to several structure from motion models. For each method a brief explanation of their functionality is offered and then we proceed to the implementation details. Since these methods were only used on outdoor geocoding, we identify some changes we had to perform to adapt them to the indoor variants detected on Chapter 3.

4.1 GPS TRANSFORMATION MATRIX (POSITION AND ORIENTATION)

To relate the 3D referential (where models are reconstructed) to the GPS coordinate system (where models are located in the world), it is required that photographs used to build models contain geographic coordinates. With both coordinate systems (3D and GPS) available, it is possible to approximate a 4x4 transformation matrix G which maps 3D positions X into GPS coordinates.

$$gps_{coords} = X.G \tag{10}$$

The G matrix can be obtained by computing an affine transformation using a set of coordinates from both 3D and GPS systems. Although to obtain this matrix it is not required the GPS coordinates of all the photographs, the sample of coordinates must be sparsely distributed on the geometric space to allow the affine transformation to infer the metric transformation for each axis.

Also, the utility of the matrix G is not limited to the calculus of positions, since it can also be used to transform the camera direction vector computed by structure from motion. This is because a vector can be represented as two points. In this case, the GPS direction may be represented in cardinal and ordinal directions (i.e. North, West, North West, etc.).

4.2 PROPOSED METHODS FOR INDOOR GEOCODING

As described in Chapter 2, the Structure from Motion algorithm allows new photographs to be added to models. This ability combined with coarse matching methods (section 2.4), and the GPS matrix allows a geocoding process for a single model. But to extend this ability to larger databases, photographs need to be efficiently geocoded to several models. As expected, there are some issues to address when processing a large scale database of structure from motion models being them:

- The weight of the database - Since the SIFT descriptors are composed by 128 unsigned char (128 bytes), small models with millions of projected features will weight considerably when considering a database with several models;
- Redundant information - Due to image overlap required by the structure from motion reconstruction, models often contain redundant descriptors. Since it is required a process which efficiently pose estimate photographs, matching these photographs to each redundant descriptors is inefficient since they contribute equal value to perform a pose;
- Coarse Matching on Multiple Models - Since location recognition requires pose estimation of photographs to a database of models, a coarse matching scheme is needed to find the correct model to geocode new photographs.

The following two sub-sections address the two chosen methods, Synthetic Views and Prioritized Features, to efficiently solve the image geocoding process on large scale 3D structure from motion databases.

4.2.1 *Synthetic Views*

Knowing the inherent problems of scalability and efficiency from location recognition based on structure from motion, Irschara et al. (2009) presented a location recognition routine which focus the real time view registration. The 3D models used by this method are assumed to be geocoded by their associated photographs.

On an offline stage, 3D structure from motion models are carefully compressed. A global threshold known as Mean Shift Clustering Comaniciu et al. (2002) is applied to group existing descriptors by their similarity, allowing the reduction of their raw amount into more than half without compromising the pose estimation of new photographs. Then the 3D scene is partitioned into a representative set of views. Several artificial cameras are uniformly placed around models and visible 3D information is reprojected into each camera. The best point of view cameras are then selected, and their 2D/3D information stored into 3D documents. Descriptors contained within 3D documents are propagated into a vocabulary tree, for a global indexation.

On an online stage, the conjunction of a vocabulary tree with stored 3D documents allows a fast and direct 2D to 3D matching of query photographs. Based on the similarity of query descriptors, the top documents are retrieved and matched iteratively until a pose estimation is found. Since a 3D document has an associated model, the pose estimation location of new photographs can be easily known. Combining the scalability of vocabulary trees and the utilization of algorithms which benefit the graphic processing unit, (Irschara et al., 2009, p. 6) was able to perform location recognition of unknown photographs with timings ranging the 72 to 297 milliseconds on a database with 1054 outdoor images.

Since we already proved that indoor structure from motion reconstruction is possible, this method will allow the removal of redundant information contained within reconstructed indoor models, and allow efficient pose estimation by reducing the amount of photographs used per model into a representative set of 3D documents.

4.2.2 *Prioritized Features*

Opposed to Synthetic Views which defines 3D models as a representative set of views, Prioritized Features selects a relevant set of 3D points to represent models. Here, views are not required to perform pose estimation and all the 2D information is removed except the descriptors. This method also assumes that 3D models are geocoded by their associated photographs, to allow the geocoding of new photographs. It is also assumed that each model contains a bipartite graph B which relates 3D points to views in which they are seen.

On an offline stage, models are compressed into a representative set of 3D points. Here it is either proposed the use of the total number of descriptors or the mean of associated descriptors to each 3D point. Then, 3D points are ranked by their visibility (the amount of views in which they are seen). Two compressed point clouds P_s and P_c are generated with the prioritized points (points with the highest rank of visibility), where the first acts as a seed cloud used for fast direct localization, and the second as a compact yet more exhaustive and accurate representation of the model itself used to complement missing information on the seed cloud. Each cloud generated is added into a global seed and compressed cloud to allow scalability.

On an online stage, new photographs are queried to the global clouds using Approximate Nearest Neighbors search. Here, it was introduced a novel bidirectional matching scheme (Li et al., 2012, p. 7):

- Feature to Point (F2P) which matches points from the query photographs to P_s, to find a set of primary matches with a low distance ratio (and thus of higher confidence). On each successful match, associated 3D points are prioritized using the bipartite graph B;
- Point to Feature (P2F) which inversely matches prioritized points from P_c to the query photograph, augmenting the set of primary matches;

These two steps are applied in cascade, so if enough matches are found after F2P, a pose estimation is computed and if successful, the P2F is skipped. As stated in (Li et al., 2010, p. 6), the advantage of this scheme is that, although we do not know the relevance of query descriptors to the point clouds, we know the importance of cloud points to the query photograph by the prioritization process.

To retrieve a pose using this method, (Li et al., 2012, p. 5) states that too much outliers are grouped together by using matches from cloud to the query photograph. Here, they propose the use of co-occurrence sampling (Li et al., 2012, p. 5). When computing a pose, rather than searching random samples on each round to find the correct inliers to compute a pose as Random Sample Consensus (RANSAC) Fischler and Bolles (1981) does, co-occurrence sampling gathers points which are correlated by the bipartite graph B on each round, and then perform perspective-3-point to compute a pose. By using this scheme, they were able to perform pose estimations with an inlier ratio of 1%.

Although this scheme for location recognition was proposed in a worldwide outdoor scale, we believe that it is possible to benefit from the prioritization given to features on indoor locations. In a large enough indoor environment, there will be places where features have an higher popularity (visibility) index which may be used to fasten the location process if photographs are taken in those places.

4.3 IMPLEMENTATION DETAILS

In this section, implementation details of our prototype are provided. Following the Synthetic Views Irschara et al. (2009); Irschara (2012) and Prioritized Features Li et al. (2010, 2012) detailed explanation, we describe the entire pipeline developed for the proposed methods, and adaptations to the indoor variants. Both methods, were developed using available open source software such as OpenCV Ope (2000) which provides a variety of image processing methods and SiftGPU Wu (a) which allows fast feature detection and extraction. Also, some pose estimation methods were adapted from the structure from motion library at Royshil (2013). Other software used will be presented at the appropriate time along with the implementation details. The code developed for our prototype was written in C++, is open source and can be found at Amorim (2014b).

4.3.1 *Implementation Details: Synthetic Views*

Model Compression

When compressing the visual information of 3D models, it is important to avoid a degenerative compression which removes relevant information to pose estimate new photographs. But a naive compression will not remove the redundancy of descriptors within models. To avoid both cases, Synthetic Views use Mean Shift Clustering to apply a global threshold to the stored descriptors. Here we opted to use the mean of the descriptors associated to each 3D point. Although this leads to a more ag-

gressive compression, it still does not degenerates models and delivers higher compression rates than Mean Shift Clustering.

Synthetic Views Generation

After compressing the amount of descriptors, we place several artificial views around the 3D model. The idea is that each artificial view takes a snapshot of the sparse model to capture several angles of the 3D structure. Each view is evaluated in terms of visibility and the best views are stored in the database.

We start by computing a ground plane of the 3D model, and therefore we use a polynomial approximation to approximate a plane to every camera position. To simplify the ground plane generation, it is assumed that every camera is located at the same height above the ground.

Afterwards, the computed plane is divided into a grid and twelve 1024x1024 artificial views are placed into each grid position, each with 1000 focal length. Since at every grid position we want to cover the entire scene, each of the twelve views have a 30^{o} difference between each other to focus the entire panorama of the 3D environment.

Since outdoor models often contain wide streets, plazas, etc, each view direction is lifted by 10^{o} (Irschara et al., 2009, p. 6) to avoid focusing 3D points on the ground (which are not useful to location recognition). As for indoor models, we do not lift views since most of the models are narrow. The same assumption that features on the ground are irrelevant is applied to features on the ceiling.

For each artificial view placed, visible 3D points are re-projected into a 2D planes and artificial photographs are generated.

As in (Irschara et al., 2009, p. 3) a 3D point is only visible to a given view if:

- Its position lies within the artificial camera view frustum culling;
- The difference of its 3D orientation with the viewing direction of the current synthetic view is lower than 30^{o} (face culling);
- The scale of 3D point is higher than 1 in terms of *Difference of Gaussian (DoG)*;

Using the pre-set width, height and focal length we define a frustum culling for each view. The orientation and scale of the 3D point is given by their associated 2D feature orientation and scale. Both these values are extrapolated into 3D to represent the direction and visibility of the 3D points.

To extrapolate the orientation of a 3D point, we start by projecting the 2D feature orientation of each associated feature to the 3D referential by using the camera and projection matrix. As we are compressing the descriptors of 3D points to the mean of all associated descriptors, here we perform the mean of all the projected orientations vectors into a single orientation. As the 2D scale is a scalar

value we ca not extrapolate it by using the projection matrices. Here the resulting value is extrapolated with the following Equation 11 as described in (Irschara, 2012, p. 127).

$$scale_{3D} = \frac{scale * distance_{viewTo3DPoint}}{focallength} \tag{11}$$

The mean of all computed 3D scales will represent the visibility of a 3D point.

The three conditions are tested by the following order: for each placed view we verify if a given point is within the frustum culling; if it is, the difference between the 3D point orientation with the view direction is computed; if the difference is lower than 30^{o}, then the point is directed into the view; and as a final test, we re-extrapolate the 3D scale into the 2D plane of the view and check if the resulting value is higher than 1 in terms of Difference of Gaussian. This threshold allows the removal of points directed to the camera but invisible to the view for being too far.

With the synthetic photographs generated, we evaluate the spatial distribution of the points throughout the photograph. This allows to remove photographs which are able to see the 3D model, but from acute, undesirable angles. Here, we simply compute the spatial histogram for both x and y axis and verify if the point distribution is close to uniform.

After generating all the synthetic photographs, we have the entire 3D model represented with views from different angles. The next step is to select the top views which cover the entire set of synthetic views. In order for a view A to be defined as covering B, we require that A sees at least 150 points of B for outdoor models and 30 for indoor models. We decrease by 5 times for indoor models based on the relation of features extracted on outdoor (about 10000+) to indoor models (around 1500 to 2000) (section 3.1). Based on the visibility coverage of all original views and synthetic views, a binary matrix is built, where 1 means "A covers B", and 0 "A does not cover B". It is relevant to consider the original views on the coverage selection as they may already be the best views to select. The greedy algorithm described in (Irschara et al., 2009, p. 4) is then applied to this matrix to retrieve the minimum number of good views to represent the 3D model.

Scalable Location Recognition

After the best views generation, their associated information is stored into a 3D document which contains the 2D and 3D points, descriptors and the related GPS transformation matrix (computed with the model geocoded photographs).

All the descriptors are then propagated into a vocabulary tree to speed up the retrieval of documents when querying new photographs. Here we use the implementation Snavely (2013) which corresponds to the vocabulary tree described in Nist and Stew (2006). This vocabulary tree uses inverse files for document storing and a TF-IDF scoring function to compute the relevance of documents to query photographs.

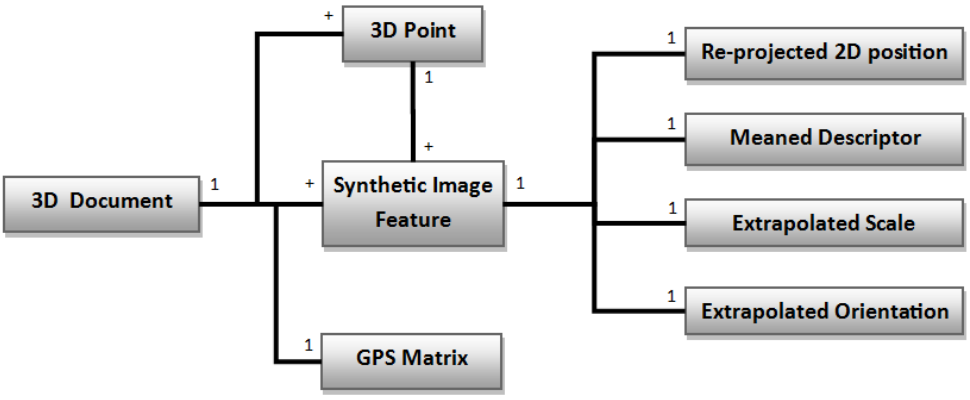

Figure 11: Structure of the generated 3D documents using Synthetic Views.

Pose Estimation Pipeline

Having all the information related to each 3D model stored in a vocabulary tree, we use the following pipeline to pose estimate new photographs.

- Feature extraction on the query photograph;
- Query the vocabulary tree for the top matches;
- Feature match between the query image and the synthetic views retrieved;
- Geometry verification to validate pose estimations;
- Compute *GPS* position and orientation.

Feature extraction is essential to perform a pose estimate as it extracts the required descriptors for the remaining pose estimation steps. We rely on the SiftGPU Wu (a) which allows fast keypoint and descriptors extraction by benefiting the graphic processing unit to fasten matrix operations. The descriptors extracted are queried to the vocabulary tree, which consequently returns the top 10 most similar 3D documents. Then we iteratively perform feature match (with SiftGPU) between the query descriptors and each retrieved document. For each match, we compute the fundamental matrix followed by the homography between the photograph and the 3D document (inspired by the implementation on (Baggio et al., 2012, p. 139)). Both these matrices are used to remove outliers and validate inliers for a successful pose estimation. Although we are not sure if Synthetic Views performs this step to refine matches, we still applied it since it is not time consuming and improves the outlier removal to perform the pose estimation. Afterwards we perform OpenCV Efficient Perspective-n-Point (EPnP) Moreno-Noguer et al. (2007) to check the coherency between the photograph keypoints and the 3D document points. Since the set of inliers is already refined before computing EPnP, the processing

time of this non-linear pose estimator will be greatly decreased. If from EPnP, at least 10 inliers are coherent (Irschara, 2012, p. 136), then a projection matrix containing the position and orientation of the new photo is computed. The GPS coordinates are then computed by using the GPS matrix related to the 3D document which allowed the pose estimation.

4.3.2 *Implementation Details: Prioritized Features*

Model Compression

Given a 3D point cloud, initially we rank each 3D point by their visibility and compress the amount of descriptors before the two compressed clouds generation. As in section 4.2.2 the visibility is given by the amount of related views given by the associated bipartite graph, which relates points to views. For the descriptor compression, we use the mean of the descriptors to represent each 3D point as we find this measure to be enough to allow pose estimation of new photographs.

Then, the given model is compressed into the seed (P_s) and compressed (P_c) clouds where the most visible points are filtered to each cloud. For the seed cloud, a maximum of 2000 points until each view is covered by 5 points for both indoor and outdoor variants. As for the P_c cloud, points are selected until each view is covered by 100 points for outdoor and 20 for indoor. As in synthetic views, here we also relieve the coverage threshold for indoor based on the amount of features expected on indoor photographs (five times less compared to outdoor) (section 3.1).

Finalizing the compression, the 3D points, visibility ranks, descriptors, the associated bipartite graph and GPS transformation matrix are added to a global seed and compressed model.

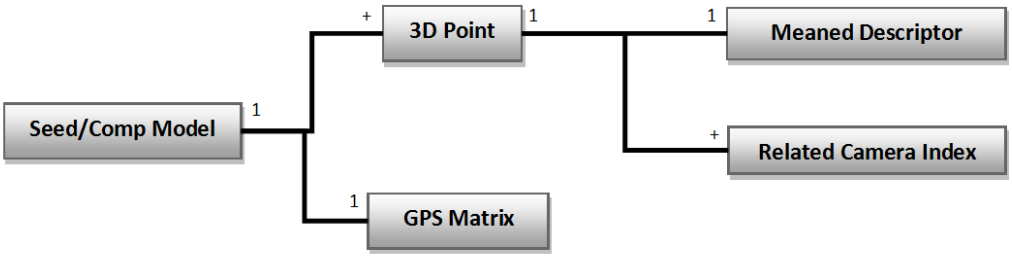

Figure 12: Structure of the generated clouds using Prioritized Features.

Pose Estimation Pipeline

The pose estimation process for Prioritized Features follows the next steps:

- Feature Extraction on the query photograph;
- Query the new photograph descriptors to the P_s cloud (Feature-to-Point);

- Query the P_c descriptors to the new photograph (Point-to-Feature);
- Perform co-occurrence sampling;
- Perform pose verification;
- Use Bundle Adjustment to optimize the final position;
- Compute GPS coordinates.

So, to find the best correspondences of the query image, we start by performing Feature-to-Point (F2P) matching, where query descriptors are matched to the P_s using the Approximate Nearest Neighbors search (ANN) Mount and Arya (2010). Here, we guarantee that successful points matched are of high confidence by applying a 0.5 ANN distance threshold. Each time a descriptor is successfully matched to a 3D point, associated points on the bipartite graph B are prioritized by a factor w (10 in our case). If 12 or more correspondences are found, we sort matches using co-occurrence sampling, attempt a pose estimation and return the location of the new image. Unless a successful pose is found, we continue the search until all query points are matched.

If matches found with F2P are insufficient to compute a pose, we advance to Point-to-Feature (P2F) matching. As we were prioritizing points while performing F2P, we start matching P_c points with the highest prioritization. By doing this we are ensuring that points matched from P_c are related by their views to matched seed points. Again we use Approximate Nearest Neighbor search but with a 0.7 threshold since P_c contains both high and low level confidence points. If more than 12 points are positively matched we sort matches using co-occurrence sampling and attempt to pose estimate the query photograph. The algorithm stops if $500N$ features are searched, where N equals to the number of points expected to match (Li et al., 2010, p. 8).

To perform co-occurrence sampling, each time a match is found, we store the set of views in which the matched 3D point is seen. The generation of inlier sets is initialized with matches in which 3D points came from the seed cloud (and thus of higher confidence). For each unrelated 3D seed point, a separate set of inliers is created. Associated points to each group are added by choosing matches which are related by the bipartite visibility graph B and has the highest amount of views in common. Each group with more than 12 points is pose estimated using OpenCV Perspective-3-Point Ameller et al. (2000), where we accept a pose if 12 or more inliers are validated (Li et al., 2012, p. 9). The final position and orientation are then enhanced using bundle adjustment. Here we use Multicore Bundle Adjustment Wu (2011), a free bundle adjustment implementation which uses CPU parallelism and GPU processor to speed the heavy optimization problem.

Both the final position and orientation are extrapolated into GPS coordinates using the GPS matrix related to the model in which the matches belong.

4.4 SUMMARY

In this chapter we explained how image based localization can be achieved. We started by proposing a simple affine transformation to geocode 3D Structure from Motion models, although we are aware that a more consistent method should be applied. With the 3D models geocoded, the GPS position and orientation of new photographs can be automatically retrieved. Afterwards we presented the inherent problems of geocoding with large scale databases, where we explain the reason for 3D models being heavy weighted and the slow geocoding speed when geocoding with several models.

Facing these problems, we proposed the use of Synthetic Views and Prioritized Features. We offered a brief overview of their concept and functionality and how they solve both scalability and performance problems on large scale databases. The implementation details and adaptions to indoor environments were then provided.

5

EXPERIMENTAL RESULTS AND DISCUSSION

In this chapter, we will describe the process used to experiment the indoor geocoding ability of our prototype using both Synthetic Views and Prioritized Features. We will begin by explaining how we gathered our data set of structure from motion models and photographs to be geocoded. Then, the experimental results of compressing models and geocoding photographs with both methods will be provided. These results will be divided by each image geocoding method and then by indoor and outdoor photographs of two different resolutions. The evaluation will be classified by compression rate, geocoding rate, speed and accuracy.

Afterward, a detailed discussion where we justify the experimental results independently for each method will be offered. And then, we will provide a comparison between both methods distinguishing significant differences.

5.1 DATASETS

The data set used to experiment our image geocoding prototype consists in a set of photographs taken on indoor sections on Minho University and Matosinhos in Portugal. Other indoor environments such as malls would be interesting to approach, but due to privacy concerns, we were not allowed to take photographs within these buildings. From our indoor data set we retrieved 443 geocoded photographs. As we want to distinguish the indoor from outdoor geocoding, we also gathered 802 geocoded photographs from buildings on Minho University Braga, Viana do Castelo and roads at Montalegre in Portugal.

From this data set, we have built six indoor and six outdoor models with VisualSFM Wu (b). Here the default options of VisualSFM were used to extract features. To match photographs, we provided the list of specified matches to perform on models in which photographs had an high level of overlap and symmetric environments. This prevented the acceptance of matches between similar but uncorrelated photographs.

Although we had entire floors represented with indoor photographs, due to areas with a low number of features, these models were partitioned into sub-models. In an attempt to reconnect these partitions, we increased the number of photographs and the overlap between photographs taken on these areas, which slightly increased the number of projected 3D points. Still, we were not able to connect the

Figure 13: A sample of indoor (top) and outdoor (bottom) photographs from our data set.

several partitions returned by structure from motion, and thus we used the six partitions with the higher number of 3D points.

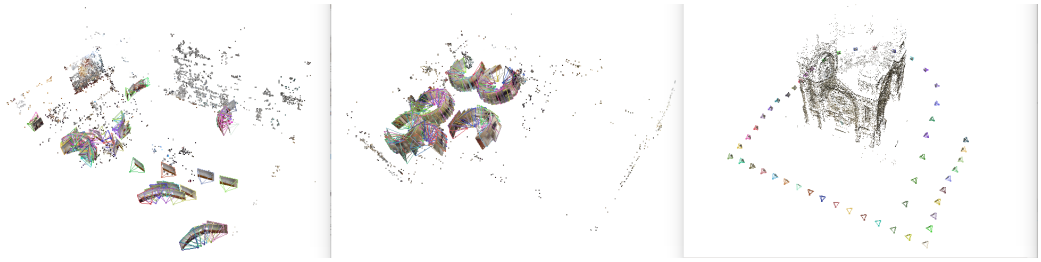

Figure 14: Sparse reconstructions from our data set visualized with VisualSFM. The center and left reconstruction belong to indoor environments and the right one belongs to outdoor.

With the GPS coordinates associated to photographs, we used the affine transformation described in section 4.1 to geocode the reconstructed models.

5.1.1 *Indoor GPS Approximation*

As we could not compute accurate GPS coordinates on indoor environments, we relied on a GPS approximation. This approximation was made by carefully registering each photograph position in meters. From Google Maps Google (2005) we retrieved one approximated GPS coordinate to each respective indoor model and assigned that coordinate to a randomly chosen photograph. Then, based on the assigned coordinate, we measured the distance between all the photographs and converted the distance from meters to GPS. Although in practice, we will not be computing real coordinates even

when correctly pose estimated, we are ensuring that any error computed between the GPS estimated by our prototype and the ground truth GPS of new photographs is real.

5.2 EXPERIMENTAL RESULTS

In this section we provide the results of pose estimating new photographs with both methods. First we will approach Synthetic Views, describing the compression rate and the geocoding rate, speed and accuracy. And then we move to Prioritized Features.

Our set of photographs to geocode consists of 25 indoor and 25 outdoor photographs from the 12 models reconstructed. These photographs were not used on the reconstruction stage. For this evaluation, we only used the latitude and longitude coordinates to measure positioning errors.

The image geocoding prototype for each method was run twice: first with the 3000x2250 photographs, as this is the maximum size that our GPU supports without SiftGPU Wu (a) performing down-sampling to allocate a texture to process photograph pixels. And then with the 1000x750 version of the original size. To maintain the aspect ratio of some outdoor photographs, we re-sized them to 3000x2510 and 1000x837 respectively. When presenting the experimental results, both resolutions will be refered as photographs of resolution 3000 and 1000 respectively.

Since the specific camera parameters for focal length were not available for the cameras used, we relied on the approximation described in VisualSFM documentation which corresponds to the a medium viewing angle.

$$focallength = max(width, height) * 1.2 \tag{12}$$

As SiftGPU was used to perform feature extraction, we indicate some of the relevant options used on both Synthetic Views and Prioritized Features prototypes.

- -cuda - Activate the CUDA version, for a faster extraction;
- -fo -1 - Allows up-sampling to extract very small features, refining the pool of feature extracted;
- -tc1 - Soft limiter to control the number of features extracted. Skips the lowest levels features to reduce the output number.
- -mo 1 - Sets the number of computed orientations per features to 1, since we do not process the descriptor orientation for both image geocoding methods.

Experimental results obtained on the following sections were achieved in a CPU Intel Core i5-4200U 1.6GHz with a nVidia GeForce 820m *GPU*. Also the data set of query photographs used can be found at Amorim (2014a) to be experimented with the example provided at Amorim (2014b) repository.

5.2.1 *Experimental Results: Synthetic Views*

First we applied the Synthetic Views compression to the data set retrieved. We used synthetic views with 1024 width and height and 1000 focal length. For outdoor models, views were tilted by 10^{o} upward and the coverage threshold was set to 150. For indoor, views were not tilted and the coverage threshold was reduced to 30 to match the decrease of feature correspondences between photographs. The mean of the descriptors was applied to remove redundant information and the best views were selected to represent the data set models. By applying this method, we were able to reduce the amount of descriptors by 79.62% for indoor and 69.73% for outdoor as Table 1 and 2 shows.

Indoor	Original	Compressed	Compression Percentage
N^{o} Descriptors	128543	26192	79.62%

Table 1: Compression rate performed by Synthetic Views when using the mean of the descriptors for indoor models.

Outdoor	Original	Compressed	Compression Percentage
N^{o} Descriptors	679750	205750	69.73%

Table 2: Compression rate performed by Synthetic Views when using the mean of the descriptors for outdoor models.

The documents generated from the compression stage were stored in a vocabulary tree of 5 levels and 10 branches per level. The code-book of words used for the vocabulary of this tree can be found at Snavely (2013).

With all the information set, we iteratively geocoded photographs to the generated models. Here, to extract features of query photographs, we used the SiftGPU option "-tc1 1600" to apply a soft limit to the number of features extracted to 1600. This option was activated to allow a closer comparison to timings presented in (Irschara et al., 2009, p. 6). The vocabulary tree is queried for the top 10 similar documents. When matching each document with the query photograph, a pose estimation is accepted if efficient perspective-n-point validates at least 10 inliers. Tables 3 and 4 show the experimental results of geocoding the 25 indoor and outdoor photographs.

Indoor	3000	1000
Posed Photographs	15 (60%)	16 (64%)
Mean Extracted Features	1773	1420
Mean Position Error	2.470 m	1.490 m
Error Range	[0.411, 11.400] m	[0.400, 5.160] m
Mean Time	838 ms	484 ms

Table 3: Overall statistics from pose estimating 25 indoor photographs of resolution 3000 and 1000 with Synthetic Views.

Outdoor	3000	1000
Posed Photographs	21 (84%)	20 (80%)
Mean Extracted Features	2102	1948
Mean Position Error	2.930 m	2.750 m
Error Range	[0.526, 9.270] m	[0.252, 8.290] m
Mean Time	884 ms	600 ms

Table 4: Overall statistics from pose estimating 25 outdoor photographs of resolution 3000 and 1000 with Synthetic Views.

With this method, 15 to 16 indoor and 21 to 20 outdoor photographs were geocoded for each resolution respectively. Photographs were never pose estimated on wrong models. High resolution photographs were processed from 838 to 884 milliseconds while the low resolution speed varies from 484 to 600 milliseconds. As for accuracy, the mean geocoding error for high resolution photographs hits the 2.470 meters for indoor and 2.930 meters for outdoor. Although the mean error is lower for indoor, the range of errors obtained is wider than outdoor. For low resolution photographs, indoor geocoding shows a lower mean error and a smaller range errors than outdoor.

For a more consistent discussion, we have also registered timings for each operation in Tables 5 and 6.

Indoor	3000	1000
Keypoint Extraction	516 ms	218 ms
Vocabulary Query (Top N)	203 ms	178 ms
Detailed Match	20 ms * N	17 ms * N
PnP Ransac	$<$ 1 ms * N	$<$ 1 ms * N
GPS retrieval	$<$ 1 ms	$<$ 1 ms

Table 5: Mean time spent by each operation made by Synthetic Views when geocoding indoor photographs.

Outdoor	3000	1000
Keypoint Extraction	534 ms	243 ms
Vocabulary Query (Top N)	255 ms	247 ms
Detailed Match	39 ms * N	36 ms * N
PnP Ransac	$<$ 1 ms * N	$<$ 1 ms * N
GPS retrieval	$<$ 1 ms	$<$ 1 ms

Table 6: Mean time spent by each operation made by Synthetic Views when geocoding outdoor photographs.

5.2.2 *Experimental Results: Prioritized Features*

Like in Synthetic Views, using Prioritized Features we compressed the descriptors by computing their mean. For each model, the seed cloud was generated by choosing a maximum of 2000 points until each view is covered by 5 points, and the compressed cloud by choosing points until each view is

covered by 100 points for outdoor and 20 points for indoor. Both clouds were then added to a global seed and compressed cloud. By applying this method compression, we were able to compress the number of descriptors to 99.18% for indoor and 97.51% for outdoor models and shown in Table 7 and 8.

Indoor	Original	Compressed	Compression Percentage
Nº Descriptors	128543	1055	99.18%

Table 7: Compression rate performed by Prioritized Features when using the mean of the descriptors for indoor models.

Outdoor	Original	Compressed	Compression Percentage
Nº Descriptors	679750	16912	97,51%

Table 8: Compression rate performed by Prioritized Features when using the mean of the descriptors for outdoor models.

At this point, we started pose estimating new photographs. For the feature extraction we used the SiftGPU option "-tc1 3200" to limit the number of features extracted. Here, we increase the number of features extracted to the double compared to synthetic views since prioritized features is more sensible to the pool of features extracted. On Feature-to-Point matching we accept matches with a distance threshold lower than 0.5 for high level confidence selection. As for Point-to-Feature we relax the distance threshold to 0.7. We accept a pose estimation after the co-occurrence sampling along with perspective-3-point validates at least 12 inliers. The stopping criteria of bidirectional matching is set to 6000 points ($500 * 12$ inliers). We do not adopted the threshold proposed in (Li et al., 2010, p. 8), as this threshold was high enough to allow searches on our entire database.

The following Tables 9 and 10 show the experimental results of geocoding the 25 indoor and outdoor photographs.

Indoor	3000	1000
Posed Photographs	15 (60%)	14 (56%)
Mean Extracted Features	2659	1420
Mean Position Error	3.430 m	2.140 m
Error Range	[0.116, 18.000] m	[0.107, 4.770] m
Mean Time	1.831 secs	1.011 secs

Table 9: Overall statistics from pose estimating 25 indoor photographs of resolution 3000 and 1000 with Prioritized Features.

With this method, 15 to 14 indoor and 20 to 16 outdoor photographs were successfully geocoded for each resolution respectively. Photographs were never geocoded on wrong models. High resolution photographs were processed with a processing time ranging the 1.831 to 2.100 seconds with a mean error 3.430 meters for indoor and 5.460 meters for outdoor. Although the mean error is lower for indoor, the range of errors obtained was wider than outdoor. As for the low resolution, photographs

Outdoor	3000	1000
Posed Photographs	20 (80%)	16 (64%)
Mean Extracted Features	4011	3813
Mean Position Error	5.460 m	5.980 m
Error Range	[2.390, 9.720] m	[2.040, 10.600] m
Mean Time	2.100 secs	2.289 secs

Table 10: Overall statistics from pose estimating 25 outdoor photographs of resolution 3000 and 1000 with Prioritized Features.

were processed on 1.011 to 2.289 seconds with a mean precision error of 2.140 for indoor and 5.980 for outdoor. Indoor photographs geocoding shows lower mean errors and a smaller range of errors compared to outdoor.

To further refine our discussion, we have also registered the time performance of each operation spent by both methods on Tables 11 and 12.

Indoor	3000	1000
Keypoint Extraction	539 ms	226 ms
ANN	1.281 ms	775 ms
Co-occurrence Sampling	6 ms	5 ms
P3P Ransac	$<$ 1 ms	$<$ 1 ms
Parallel Bundle Adjustment	1 ms	$<$ 1 ms
Compute GPS	$<$ 1 ms	$<$ 1 ms

Table 11: Mean time spent by each operation made by Prioritized Features when geocoding indoor photographs.

Outdoor	3000	1000
Keypoint Extraction	566 ms	304 ms
ANN	1.524 ms	1.975 ms
Co-occurrence Sampling	3 ms	6 ms
P3P Ransac	$<$ 1 ms	$<$ 1 ms
Parallel Bundle Adjustment	3 ms	1 ms
Compute GPS	$<$ 1 ms	$<$ 1 ms

Table 12: Mean time spent by each operation made by Prioritized Features when geocoding outdoor photographs.

5.3 DISCUSSION

In this section we provide a detailed discussion of the experimental results obtained with both methods. Each method will be evaluated in terms of compression rate, geocoding rate, speed and accuracy, followed by an individual overall discussion. Afterwards, we compare both methods by the same evaluation metrics.

5.3.1 *Discussion: Synthetic Views*

Compression Rate

Due to the overlap of photographs required to build indoor models as explained in section 5.1, it was expected that indoor models had an higher level of redundant descriptors compared to outdoor. As shown in Tables 1 and 2, we achieved a compression rate of 79.62% for indoor compared to the 69.73% for outdoor, which proves the presence of redundancy on indoor models.

Geocoding Rate

Although the compression rate is high, visual information kept within our database still allowed the geocoding of new photographs. While for outdoor photographs the geocoding rate hits the 84% of the queried photographs, for indoor only a maximum 64% was successful, which proves that the associated difficulty of geocoding indoor photographs is higher.

Investigating the core of this problem, we noticed that when indoor models were compressed, an higher number of synthetic views were needed to entirely cover models from different points of view. This indicates that indoor models are harder to represent with synthetic views than outdoor. While the threshold of 30 points coverage offered a good view coverage for some indoor models, for others it did not work as it was supposed to, and thus, artificial views placed on similar points of view were also stored. Storing these views with similar visual information will add entropy to the vocabulary tree which in consequence difficult the retrieval of the top similar and ideal documents to pose estimate new photographs. We believe that the source of this problem comes from the lack of feature points shared between photographs used to build models, which in consequence, prevents synthetic views placed from covering large structural areas. Lowering this threshold even more to adapt to the lack of points shared only degenerated indoor models as it removed too much information to pose estimate new photographs.

Despite not being a significant change, the geocoding of one more indoor photograph when lowering the resolution is an interesting result to be analysed. While at first glance, lowering photographs resolution implies the removal of relevant visual information by the pixel compression, it also removes misleading features which may confuse the vocabulary tree queries. The extra photograph that was geocoded is presented on Figure 15. On our indoor data set, we have two models which contain a similar extinguish hose box, but on different locations. While on an higher resolution, this box will have an heavier weight on computing the similarity of documents (since more features are extracted on the box), on a lower resolution the similarity is distributed by features around the extinguish hose box.

As for outdoor geocoding rate, decreasing the photographs resolution also does not show significant changes. But here, the geocoding of one less photograph on resolution 1000 shows the standard reactions of removing relevant features by the pixel compression.

Figure 15: An image from our data set (left) and a query photograph (right). Although the environment is clearly distinct, similar objects such as the extinguishing hose box may be found. These similarities may confuse the vocabulary top matches retrieved.

Accuracy

Comparing the accuracy between indoor and outdoor photographs, indoor photographs have a lower mean positioning error than outdoor. Several facts may justify this. First, the geocoding rate on outdoor is higher than indoor which affects the magnitude of the mean error. Second, on outdoor we compute a GPS matrix which correlates two coordinate systems of a large space unit. This means that any positioning error returned by the pose estimation process will be mirrored by the same magnitude into the computed GPS coordinates. Thus, it is natural that outdoor photographs are computed with an higher GPS deviation to their ground truth positions. And third, we are estimating the real focal length and are not applying the distortion of photographs, so positioning errors may be inherited from these estimations.

The indoor positioning errors have a considerable decrease when lowering images resolution compared to outdoor. We believe that this difference is caused by the consistency of synthetic views generated referred on the previous topic. Lowering the resolution, implies extracting different features. In consequence, the vocabulary tree returns different documents based on the similarity of the query features.

As for outdoor, the mean positioning error decreases along with the geocoding rate. The extra photograph that was geocoded on resolution 3000 had an associated positioning error of 6.5 meters. Removing this photograph from the calculus of the mean error will decrease the mean error computed, which is what was observed.

Speed

The mean speed of geocoding the 25 photographs with our prototype is 838 milliseconds for indoor and 884 milliseconds for outdoor. Lowering the resolution, increased the speed considerably by 354 milliseconds for indoor and 284 milliseconds for outdoor. Analysing Tables 5 and 6, this is justified by the decrease on time spent extracting and matching features, since we are lowering the number of

pixels to process. The number of features extracted is consequently lower when reducing image sizes as proved on Tables 3 and 4. All the remaining operations have small changes on both sizes.

It is also important to note that our vocabulary tree is run entirely on CPU. Computing the score for a different amount of descriptors deliver different processing speeds. This justifies indoor photographs query being 52 and 69 milliseconds faster than outdoor for each resolution respectively, due to difference on the amount of features extracted.

Overall Discussion

As an overall evaluation, our synthetic views prototype suffered from a low indoor geocoding rate compared to outdoor. We believe that the core of this problem comes from the consistency indoor models, as points shared between views were not enough for a consistent coverage.

Lowering the resolution presented small changes in which we believe that this method is resilient to process small amounts of visual features. As for accuracy, high positioning errors were obtained. While for outdoor, 10 meters may not imply an incorrect localization, for indoor, even 1 meter may geocode our photographs behind walls, bellow floors or above ceilings. Since we used Efficient Perspective-N-Point for the pose estimation, we also tried Perspective-3-Point as proposed in (Irschara et al., 2009, p. 6). But using this weaker pose estimation algorithm, only increased our geocoding positioning errors and did not improved our geocoding rate.

Provided that we did not reach the real time obtained by Irschara (Irschara et al., 2009, p. 6), we can assume that our pipeline is not refined to performance fast image geocoding. Although we are not aware of which parameters Irschara used to extract features with SiftGPU, we believe that our parameters are necessary to avoid hindering the indoor image geocoding. As for the vocabulary tree, improvements can be made. To speed the query of the top documents, in (Irschara et al., 2009, p. 5-6) it was implemented a vocabulary tree which resources to the GPU and CPU parallelism to fasten matrix operations when processing descriptors and computing scores. Here they achieved an impressive query time of 19 ms to retrieve top 10 documents on a 3 levels, 50 branches vocabulary tree on their Intel Pentium D 3.2Ghz and a GeForce GTX 280 system . Furthermore, it uses a probabilistic function to refine the false positive documents. With this improvement, we are assuming that our implementation would achieve a geocoding speed ranging the 254 to 407 milliseconds (28.09 to 38.50 % faster) for indoor and 298 to 622 milliseconds (26.82 to 43.35 % faster) for outdoor with 1000x750 photographs.

5.3.2 *Discussion: Prioritized Features*

Compression Rate

The compression rate of prioritized features show an higher redundancy removal for indoor model compared to outdoor. As shown in Tables 7 and 8, we achieved a compression rate of 99.18% for indoor and a lower compression rate of 97.51% for outdoor models, which again proves the removal of redundancy present on indoor models.

Geocoding Rate

Provided that we removed more than 97% visual information from our database, we were expecting a low geocoding rate for both outdoor and indoor. This was not the case, as we were able to successfully geocode a maximum of 15 indoor and 20 outdoor photographs for the resolution 3000. This method shows less resilience to decreasing resolution as both indoor and outdoor geocoding were able to geocode less photographs. Since this method relies on re-directing query features to relevant data base features through bidirectional matching, having a larger pool of relevant features will re-enforce this redirection to the correct 3D points to allow a pose estimation. Misleading features within this large pool are simply removed by the co-occurrence sampling outlier removal.

Comparing the indoor to outdoor geocoding rate, indoor photographs proved to be harder to pose estimate than outdoor. Since this method compresses relevant information by selecting points which are seen by several views, processing models with weak photograph correlations will cause this algorithm to discard some relevant features which may help pose estimating new photographs.

Accuracy

Proportional to lowering the resolution of query photographs, the mean positioning errors was increased for outdoor models, but not for indoor. The explanation to this is simple. On indoor, the photograph that was not geocoded on resolution 1000, was geocoded with a positioning error of 8.3 meters for resolution 3000. On outdoor, photographs which were not geocoded on resolution 1000, were geocoded on resolution 3000 with a positioning error bellow the 5.460 meters mean error. While on indoor, avoiding the pose estimation of the missing photograph will decrease the mean error computed, for outdoor removing photographs which were geocoded with errors below the mean will increase the mean error computed.

As a comparison of indoor to outdoor errors, indoor shows lower errors than outdoor. Again this may be caused by outdoor models having an higher space unit, and any positioning error computed by the pose estimation is mirrored by the GPS matrix.

Although a local bundle adjustment is applied to each computed pose, the most problematic errors obtained were higher than 4.770 meters which is unacceptable for a precise localization. Since we are estimating the focal length and ignoring the distortion of photographs, we believe that a portion of

these errors are being caused by these estimations.

Speed

Lowering the size of query photographs has a positive effect on the processing speed per photograph. Analysing Tables 11 and 12, it is noticeable that timings on feature extraction are lower for resolution 1000 than 3000, due to the reduction of the number of pixels to process. As the number of features extracted are reduced due to the compression of pixels as shown on Table 9 and 10, the Approximate Nearest Neighbors search consequently should be faster for the photographs resolution 1000.

Although this is true for indoor photographs, outdoor photographs spent more time on the ANN search. This result can be justified with the geocoding rate of outdoor photographs. While on indoor, only 1 more photograph required a deeper ANN search until our algorithm hits the searching threshold of 6000 points, for outdoor ANN search hits this threshold four more times for the lower resolution, and thus, more processing time spent.

Comparing indoor to outdoor speed, we noticed significant decrease on processing the ANN search from outdoor to indoor. Two hypotheses may be formulated from this. Either indoor photographs found the relevant 3D points faster than outdoor, and a successful pose estimation was achieved at an early stage. Or the number of features extracted from indoor photographs is lower than outdoor, which then relieves the ANN search. We believe that the second conclusion is the right one, as the co-occurrence sampling processing time was higher on indoor environments than outdoor. This means that matches found were rejected more often than outdoor, and thus ANN search needed to search more points to complement the pool of matches, which does not fit on the first hypothesis.

Overall Discussion

As an overall evaluation, our prioritized features prototype also suffered from a low indoor geocoding rate compared to outdoor. We believe that the lack of correspondences shared between 3D model photographs is the source of this problem as the prioritization process will be affected by the number of views associated to each 3D points.

Here we also tried the EPnP algorithm for the pose estimation, but doing so decreased the pose estimation rate on both indoor and outdoor. Co-occurrence sampling was specially developed to work under the assumption of a weak pose, since it has an high accuracy on gathering inliers than the Random Sample Consensus selection.

Lowering the resolution of photographs impacts negatively the geocoding rate and precision. This is caused by the lack of relevant features extracted used to direct the query descriptors to the correct 3D points. Inversely, the processing speed is increased when lowering the resolution since we are extracting less features from a smaller amount of pixels.

Since (Li et al., 2012, p. 12) only exposed a partial performance speed of few seconds per photographs, we are assuming that our timings are within their boundaries.

5.3.3 *Synthetic Views vs Prioritized Features*

To complete our discussion, we now compare the performance of both prototypes developed. Through our experimental results, we proved that Synthetic Views and Prioritized Features are able to geocode outdoor and indoor photographs. Although their geocoding rate have shown a considerable discrepancy from outdoor to indoor, we believe that indoor models should be improved with wide angle photographs before inferring if further adaptations of this algorithms to indoor environments should be done.

In terms of compression rate, we observed a much higher compression from our prioritized Features Prototype while maintaining a similar geocoding rate for indoor and outdoor when compared to Synthetic Views. From this, the prioritization of points proves to be ideal to 3D model compressed representation, if we are focusing the weight of models.

Synthetic Views have shown to be more resilient to pose estimating low resolution photographs than Prioritized Features. While a weaker pool of features will ideally uniformly affect the similarity score computed by vocabulary queries, on Prioritized Features this weaker pool misleads the search and prioritization of 3D points, which consequently allows less geocoded photographs.

Evaluating the accuracy, both methods presented high positioning errors, which do not allow precise geocoding. Here, photographs with known focal length and distortion factors should be used for a more accurate evaluation. We believe that these errors will be greatly improved provided that we have this information.

As for the geocoding speed per photograph, experimental results shown that computational timings are mostly influenced by the extraction of features and coarse matching to retrieve relevant information to perform a successful pose. Since we used SiftGPU, which benefits the GPU to fasten the extraction of features, our second idea to improve the extraction speed is by lowering the resolution of query photographs. As for coarse matching, the vocabulary tree query requires less time to find relevant information than approximate nearest neighbors search. From this two facts, we conclude that Synthetic Views is ideally adapted to a faster location recognition than Prioritized Features.

5.4 SUMMARY

In this chapter we started by presenting the data set of indoor and outdoor photographs gathered to evaluate the performance of view registration from our prototype. Then we described the experimental process of geocoding new photographs using both methods. Here we provided the experimental results sorted by indoor and outdoor geocoding, with high and low resolution photographs. For each variant, we evaluated both methods by compression rate, geocoding rate, geocoding speed and precision.

Afterwards, we provided a detailed explanation of the experimental results obtained, while focusing the comparison between indoor and outdoor geocoding. An overall discussion comparing both methods was then provided where we conclude that both methods are able to geocode new indoor

photographs, despite their compression rate. But indoor models should be refined before inferring if both methods require further adaptation to the indoor varaints. Also, we concluded that Synthetic Views performs a faster location recognition by using a vocabulary tree and is more resilient to low resolution photographs when compared to Prioritized Features.

6

CONCLUSION AND FUTURE WORK

In this thesis we addressed the problem of efficient localization on indoor environments using Structure from Motion models. Two Computer Vision algorithms, Synthetic Views and Prioritized Features, which allow efficient geocoding of outdoor photographs were presented. We studied the structural differences between outdoor and indoor SFM models, and adapted both methods into the indoor variant. Provided that GPS coordinates of associated photographs are present, a simple affine transformation allows the calculus of GPS coordinates from new pose estimated photographs. Although the complete pipeline of both methods was not implemented, we answered our main research hypothesis by being able to geocode unknown indoor photographs on a database of 1245 indoor and outdoor photographs with both methods, while addressing the compression of the database. Relatively to our secondary questions, the geocoding process can be executed in less than a second with our Synthetic Views prototype. Provided that the vocabulary tree which benefits the GPU is implemented, faster processing timings are achievable. As far as how much accurate can the indoor geocoding process be, with our prototype we obtained 1.490 and 2.140 meters of mean position error with Synthetic Views and Prioritized Features respectively. Still, the range of errors reached the 5.160 and 4.770 meters in our experimental results, which are high enough for an incorrect GPS estimation. Provided that the focal length and distortion factors are known for query photographs, we believe that both speed and accuracy can be balanced with Synthetic Views. Through our experimental results, it was also proven that image geocoding can be performed with low resolution photographs with Synthetic Views.

6.1 FUTURE WORK

The work of this thesis is complete, since we were able to proved that indoor image geocoding is possible within a second. In this section we provide some ideas which address the improvement of the prototype developed.

6.1.1 *SIFT descriptor accuracy*

Our image geocoding prototype was based on SIFT features to extract both keypoints and descriptors from photographs. The choice of using this type of features was based on the fact that both Synthetic Views and Prioritized Features used the same feature type to process photographs. Although we are satisfied with the results performed from SIFT descriptors, we do not want to discredit other feature types such as SURF which were proven to be as accurate as SIFT. Regarding the size of the SIFT features, the performance of PCA-SIFT Ke and Sukthankar (2004) should also be experimented as these features are a compressed representation of SIFT that could potentially reduce the weight of our database.

6.1.2 *Refine 3D models*

There are two improvements that should be done to our models, where the first addresses their reconstruction and the second the GPS acquisition and respective model geocoding.

As stated in sections 5.3.1 and 5.3.2, our prototype delivered a lower geocoding rate to indoor than outdoor. Through our experimental results discussion, we believe that this performance decrease on indoor was caused by the consistency of indoor models, where indoor photographs contained a low number of features and correspondences. To increase the number of features per photograph we advise the use of wide angle cameras to represent indoor environments. Since an wider area will be covered by each of these photographs, more features will be detected and thus, more correspondences will be found when overlapping photographs. In consequence, an higher number of 2D points will be projected into 3D and the coverage thresholds from Synthetic Views and Prioritized Features will work as expected. Currently there is an available data set of indoor photographs which are ideal for structure from motion reconstruction at Huitl et al. (2012). This data set was not tested with our prototype since the point cloud and photographs available did not contain feature keypoints, descriptors, scale and orientations required by our prototype. Also we wanted to show potential differences between indoor and outdoor image geocoding, and this data set only contained indoor photographs taken from cameras which we did not have.

In section 5.1.1 we stated that we could not retrieve precise indoor coordinates. Instead, we reverse engineered a GPS approximation from the metrical distances between photograph positions. Also, computing the GPS transformation matrix as we proposed will absorb the positioning error from all associated photographs. Geocoding new photographs will mirror this error. As a future task we propose to directly geocode the structure from motion model rather than from its associated photographs. Applications such as Google Maps Google (2005) and OpenStreetMaps Coast (2004) allows the access to a consistent global mapping of buildings. So, if it developed an algorithm which fits the reconstructed models into their associated building boundaries (on these global maps), we can directly

retrieve the GPS coordinates of models. By doing this we will be able to automatically geocode indoor structures and focus positioning errors model wise, allowing more flexible adjustments if required.

6.1.3 *Photographs Focal Length and Distortion*

As explained in section 5.2, we did not know the real focal length for the photographs used, and thus we used a focal length approximation. In section 5.3.3, we affirmed that errors computed on the geocoding of new photographs may be due the focal length approximation.

Besides, we assumed that photograph machines lenses are completely planar. Although present professional photograph machines are quite refined in terms of image distortion, this system should be generalized to support cheap equipment such as mobile phone cameras. In which case, we should take in account the distortion of cheap lenses when querying new photographs for GPS coordinates. To do so, a more accurate information of the machine lens is needed rather than just estimating the focal length. The information about distortion can be retrieved with the method proposed in Zhang and Kang (2004) which requires several photographs on a chessboard pattern to compute the distortion factors. Alternatively, this information can also be retrieved from the lens providers.

Applying the correct size of the focal length and distortion correction of photographs used on model reconstruction will improve the consistency of models. Also, geocoding photographs with this correction will decrease positioning errors as discussed in section 5.3.3.

6.1.4 *Vocabulary Tree with GPU scoring functions*

As refereed in the overall discussion of section 5.3.1, we propose the use of the vocabulary tree described in (Irschara et al., 2009, p. 5). This implementation resorts to CPU parallelism to compute the visual words for each feature extracted from query photographs and the GPU to speed the calculus of the respective visual words. Compared to the vocabulary tree used in our prototype, the GPU implementation should deliver faster queries, which will greatly improve the geocoding speed of new photographs.

6.1.5 *Co-occurence Sampling*

As described in (Li et al., 2012, p. 5), it was implemented a variant version of Random Sample Consensus where on each round, they select the co-occurrence samples and test a pose with them. On our version of Prioritized Features, we select the samples and then perform Perspective-3-Point RANSAC to pose estimate the new photograph, which results in a much slower process if enough matches to pose are found very often. Although in theory our implementation should deliver the same results as theirs, the correct implementation should be addressed.

6.1.6 *Larger Database*

Both methods should be tested in larger indoor databases. Since location recognition through optical methods strongly relies on the similarity of features, it is expected self confusion caused when by similar environment models. This self confusion can even occur within the same building where hallways have same structure through different floors or even rooms on the same floor. To avoid returning erroneous location and orientation by similarity, a possible solution would be the integration of a time lapse positioning log which prevents the acceptance of geocoded photographs greatly distanced from the last position registered.

6.1.7 *Updating SFM Models*

The image geocoding process proposed on this thesis strongly depends on the similarity of environments. This being said, the model information stored within the database will be outdated very often if the environments represented are constantly changing. To avoid outdated information, we advise the implementation of an update process using structure from motion. Since this algorithm allows incremental reconstruction, models can be updated at any time by replacing the information of outdated views with the new visual information. Although we are aware that this updating process is time expensive, it is only required when significant changes occur and can be done in background.

6.1.8 *Image Geocoding as Service*

The main utility of our research is the applicability of the image geocoding process into a client server service as in Huitl et al. (2012), where clients send photographs from their location to the server, and the server quickly returns their location based on the similarity of photographs to a database of geocoded models. Regarding the constant evolution of mobile phone devices and the fact that almost everyone has a mobile phone with a photograph machine, it would be interesting to research how the image geocoding pipeline can be divided between clients and servers.

For instance, as discussed in sections 5.3.1 and 5.3.2, the most expensive operations performed by the image geocoding pipeline is the extraction of features. Addressing the powerful mobile phones GPU, efficient extraction software which uses this processor can be implemented to extract photograph features. This opens the possibility of processing feature extraction on the client side, which in turn sends the keypoints and descriptors to the server rather than the entire photograph, and thus, the response speed of new queries is increased.

BIBLIOGRAPHY

(2000). OpenCV (Open Source Computer Vision). http://opencv.org/. Accessed 7 May 2014.

3DFLOW. 3DF Samantha – Structure from Motion at its finest. http://www.3dflow.net/technology/samantha-structure-from-motion/. Accessed 14 February 2014.

Acute3D (2011). Smart3DCapture. http://www.acute3d.com/smart3dcapture/. Accessed 14 February 2014.

Agarwal, S., Snavely, N., M. Seitz, S., and Szeliski, R. (2010). Bundle Adjustment in the Large. *Computer Vision – ECCV*, 6312:29–42. DOI: 10.1007/978-3-642-15552-9_3.

AgiSoft (2006). AgiSoft PhotoScan. http://www.agisoft.com/. Accessed 14 February 2014.

Ameller, M.-A., Triggs, B., and Quan, L. (2000). Camera pose revisited: New linear algorithms. *ECCV'00*, page 13.

Amorim, N. M. (2014a). Image Geocoding Dataset. https://www.dropbox.com/sh/ns2h7x3iv9ne5ss/AAA1lhovCywGseIzXTxn4L5Wa?dl=0. Accessed 31 October 2014.

Amorim, N. M. (2014b). Indoor Image Geocoding Prototype. https://bitbucket.org/nam81/qt-sfm. Accessed 26 October 2014.

Baggio, D. L., Emani, S., Escriva, D. M., Levgen, K., Mahmood, N., Saragih, J., and Shilkrot, R. (2012). *Mastering OpenCV with Practical Computer Vision Projects*.

Bay, H., Tuytelaars, T., and Van Gool, L. (2006). SURF: Speeded Up Robust Features. *Computer Vision - ECCV*, 3951:404–417. DOI: 10.1007/11744023_32.

Coast, S. (2004). Open Street Maps. www.openstreetmap.org/. Accessed 12 October 2014.

Comaniciu, D., Meer, P., and Member, S. (2002). Mean Shift: A Robust Approach Toward Feature Space Analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619. DOI: 10.1109/34.1000236.

Das, P. and Agrawal, D. P. (2014). RFID for Indoor Position Determination. *Instrumentation and Measurement, IEEE Transactions*, 28:561–567. DOI: 10.1007/978-3-319-07350-7_62.

Datar, M., Immorlica, N., Indyk, P., and Mirrokni, V. S. (2004). Locality-sensitive hashing scheme based on p-stable distributions. *Proceedings of the twentieth annual symposium on Computational geometry - SCG '04*, page 253. DOI: 10.1145/997817.997857.

Enqvist, O., Olsson, C., and Kahl, F. (2011). Stable Structure from Motion using Rotational Consistency. http://www.maths.lth.se/matematiklth/personal/calle/tech_rep/tech_rep.html.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395. DOI: 10.1145/358669.358692.

Fraundorfer, F., Wu, C., Frahm, J.-m., and Pollefeys, M. (2008). Visual Word based Location Recognition in 3D models using Distance Augmented Weighting. *3DPVT08*. DOI: 10.1.1.325.8145.

Fukunaga, K. and Narendra, P. M. (1975). A Branch and Bound Algorithm for Computing k-Nearest Neighbors. *IEEE Transactions on Computers*, C-24(7):750–753. DOI: 10.1109/T-C.1975.224297.

Google (2005). Google Maps. https://www.google.pt/maps/. Accessed 12 October 2014.

Hartley, R. and Zisserman, A. (2004). Epipolar Geometry and the Fundamental Matrix. *Multiple View Geometry in Computer Vision*, pages 239–261.

Hazas, M. and Hopper, A. (2006). Broadband ultrasonic location systems for improved indoor positioning. *Mobile Computing, IEEE Transactions*, pages 536–547. DOI: 10.1109/TMC.2006.57.

Henri, A. SFMToolkit. http://www.visual-experiments.com/demos/sfmtoolkit/. Accessed 14 February 2014.

Hightower, J., Want, R., and Borriello, G. (2000). SpotON: An indoor 3D location sensing techology based on RF signal strength. *UW CSE00-02-02*.

Hol, J. D., Dijkstra, F., Luinge, H., and Schon, T. B. (2009). Tightly Coupled UWB/IMU Pose Estimation. *Proceedings of the IEEE International Conference on Ultra-Wideband (ICUWB)*.

Huitl, R., Schroth, G., Hilsenbeck, S., Schweiger, F., and Steinbach, E. (2012). TUMindoor dataset. http://www.navvis.de/dataset. Accessed 31 October 2014.

Indyk, P. and Motwani, R. (1998). Approximate nearest neighbors: towards removing the curse of dimensionality. *STOC '98 Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613. DOI: 10.1145/276698.276876.

Irschara, A. (2012). *Scalable Scene Reconstruction and Image Based Localization*. PhD thesis.

Irschara, A., Zach, C., Frahm, J.-M., and Bischof, H. (2009). From structure-from-motion point clouds to fast location recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2599–2606. DOI: 10.1109/CVPR.2009.5206587.

Jacobs, C. E., Finkelstein, A., and Salesin, D. H. (1995). Fast multiresolution image querying. *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. DOI: 10.1145/218380.218454.

Jung, H. O., Doojin, K., and Lee Beom, H. (2014). An Indoor Localization System for Mobile Robots Using an Active Infrared Positioning Sensor. *Journal of Industrial and Intelligent Information*, pages 35–38.

Kato, T., Kurita, T., Otsu, N., and Hirata, K. (1992). A sketch retrieval method for full color image database. *Proceedings of International Conference on Pattern Recognition*. DOI: 10.1109/ICPR.1992.201616.

Kawaji, H., Hatada, K., Yamasaki, T., and Aizawa, K. Image-based Indoor Positioning System: Fast Image Matching using Omnidirectional Panoramic Images. pages 1–4. DOI: 10.1145/1878039.1878041.

Ke, Y. and Sukthankar, R. (2004). PCA-SIFT: a more distinctive representation for local image descriptors. *CVPR'04 Proceedings of the IEEE computer society conference on Computer vision and pattern recognition*, pages 506–513. DOI: 10.1109/CVPR.2004.1315206.

Li, Y., Snavely, N., Huttenlocher, D., and Fua, P. (2012). Worldwide Pose Estimation using 3D Point Clouds. *ECCV'12 Proceedings of the 12th European conference on Computer Vision*, pages 15–29. DOI: 10.1007/978-3-642-33718-5_2.

Li, Y., Snavely, N., and P Huttenlocher, D. (2010). Location Recognition using Prioritized Feature Matching. *ECCV'10 Proceedings of the 11th European conference on Computer vision*, pages 791–804. DOI: 10.1007/978-3-642-15552-9_57.

Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, pages 91–110. DOI: 10.1023/B:VISI.0000029664.99615.94.

Mao, L., Chen, J., Li, Z., and Zhang, D. (2013). Relative Localization Method of Multiple Micro Robots Based on Simple Sensors. *International Journal of Advanced Robotic Systems*. DOI: 10.5772/55587.

Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions*, 27(10):1615–1630. DOI: 10.1109/TPAMI.2005.188.

Minami, M., Fukuju, Y., Hirasawa, K., Yokoyama, S., Mizumachi, M., Morikawa, H., and Aoyama, T. (2004). Dolphin: A pratical approach for implementing a fully distributed indoor ultrasonic positioning system. *Ubicomp*, pages 347–365. DOI: 10.1007/978-3-540-30119-6_21.

Moreno-Noguer, F., Lepetit, V., and Fua, P. (2007). Accurate non-iterative o(n) solution to the pnp problem. *IEEE 11th International Conference on Computer Vision*, pages 1–8. DOI: 10.1109/ICCV.2007.4409116.

Mount, D. M. and Arya, S. (2010). ANN: A Library for Approximate Nearest Neighbor Searching. http://www.cs.umd.edu/~mount/ANN/. Accessed 15 May 2014.

NI, L. M., Yunhao, L., Lau Yiu, C., and P. Patil, A. (2003). LANDMARC: Indoor Location Sensing Using Active RFID. *Pervasive Computing and Communications. Proceedings of the First IEEE International Conference*, pages 407–415. DOI: 10.1109/ PERCOM.2003.1192765.

Niblack, W. and Barber, R. (1993). The qbic project: querying images by content using color, texture and shape. *Proceedings of SPIE Storage and Retrieval for Image and Video Databases*. DOI: 10.1117/12.143648.

Nist, D. and Stew, H. (2006). Scalable Recognition with a Vocabulary Tree. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2161–2168. DOI: 10.1109/CVPR.2006.264.

NVidia. CUBLAS: CUDA Basic Linear Algebra Subprograms. http://docs.nvidia.com/cuda/cublas/#abstract. Accessed 27 May 2014.

Patrick, R., Angermann, M., and Bernhard, K. (2009). Simultaneous localization and mapping for pedestrians using only foot-mounted inertial sensors. *Proceedings of the 11th international conference on Ubiquitous computing*, pages 93–96. DOI: 10.1145/1620545.1620560.

Priyantha, N. B. (2005). The cricket indoor location system. *PhD Thesis, Massachusetts Institute of Technology*, page 199.

Ravi, N., Shankar, P., Frankel, A., Elgammal, A., and Iftode, L. (2006). Indoor Localization Using Camera Phones. *Seventh IEEE Workshop on Mobile Computing Systems & Applications (WMCSA'06)*, page 49. DOI: 10.1109/WMCSA.2006.12.

Royshil (2013). Toy Structure From Motion Library using OpenCV. https://github.com/royshil/SfM-Toy-Library. Accessed 15 May 2014.

Ruiz, A., Granja, F., Priesto, Honorato, J., and Rosas, J. (2012). Accurate Pedestrian Indoor Navigation by Tightly Coupling Foot-Mounted IMU and RFID Measurements. *IEEE Transactions on Instrumentation and Measurement*, pages 178–189. DOI: 10.1109/ICUWB.2009.5288724.

Schindler, G., Brown, M., and Szeliski, R. (2007). City-Scale Location Recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7. DOI: 10.1109/CVPR.2007.383150.

Schroth, G., Huitl, R., Chen, D., Abu-Alqumsan, M., AlNuaimi, A., and Steinbach, E. (2011). Mobile visual location recognition. *IEEE Signal Processing Magazine*, 28(4):4. DOI: 10.1109/MSP.2011.940882.

Silpa-Anan, C. and Hartley, R. (2008). Optimised KD-trees for fast image descriptor matching. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. DOI: 10.1109/CVPR.2008.4587638.

Sivic, J. and Zisserman, A. (2003). Video google: A text retrieval approach to object matching in videos. *IEEE International Conference on Computer Vision*, 2:1470–1477. DOI: 10.1109/ICCV.2003.1238663.

Snavely, N. (2006). Bundler. http://www.cs.cornell.edu/~snavely/bundler/. Accessed 7 February 2014.

Snavely, N. (2013). VocabTree2. https://github.com/snavely/VocabTree2. Accessed 7 May 2014.

Triggs, B., F Mclauchlan, P., I Hartley, R., and W Fitzgibbon, A. (2000). Bundle Adjustment — A Modern Synthesis. *Vision Algorithms: Theory and Practice*, pages 298–372. DOI: 10.1007/3-540-44480-7_21.

Vellaikal, A. and Kuo, C. (1995). Content-based retrieval using multiresolution histogram representation. *Digital Image Storage Archiving Systems*, 2602.

Woodman, O. and Harle, R. (2008). Pedestrian localisation for indoor environments. *UbiComp '08 Proceedings of the 10th international conference on Ubiquitous computing*, pages 114–123. DOI: 10.1145/1409635.1409651.

Wu, C. SiftGPU. http://cs.unc.edu/~ccwu/siftgpu/. Accessed 8 May 2014.

Wu, C. VisualSFM - A Visual Structure from Motion System. http://ccwu.me/vsfm/. Accessed 7 May 2014.

Wu, C. (2011). Multicore Bundle Adjustment. http://grail.cs.washington.edu/projects/mcba/. Accessed 7 February 2014.

Wu, C. (2013). Towards Linear-Time Incremental Structure from Motion. *International Conference on 3D Vision*, pages 127–134. DOI: 10.1109/3DV.2013.25.

Wu, C., Agarwal, S., Curless, B., and M. Seitz, S. (2011). Multicore bundle adjustment. *CVPR 2011*, (1):3057–3064. DOI: 10.1109/CVPR.2011.5995552.

Zhang, D., Yang, Y., Cheng, D., Liu, S., and NI, L. (2010). COCKTAIL: An RF-based Hybrid Approach for Indoor Localization. *IEEE International Conference on Communications (ICC)*, pages 1–5. DOI: 10.1109/ICC.2010.5502137.

Zhang, Z. and Kang, S. B. (2004). Camera Calibration. *Emergin Topics in Computer Vision*, pages 4–43.