ON THE IMPROVEMENT OF THREE-DIMENSIONAL RECONSTRUCTION FROM LARGE DATASETS

GUILHERME AUGUSTO POTJE

ON THE IMPROVEMENT OF THREE-DIMENSIONAL RECONSTRUCTION FROM LARGE DATASETS

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

Orientador: Erickson Rangel do Nascimento
Coorientador: Mario Fernando Montenegro Campos

Belo Horizonte
Março de 2016

GUILHERME AUGUSTO POTJE

ON THE IMPROVEMENT OF THREE-DIMENSIONAL RECONSTRUCTION FROM LARGE DATASETS

Dissertation presented to the Graduate Program in Ciência da Computação of the Universidade Federal de Minas Gerais in partial fulfillment of the requirements for the degree of Master in Ciência da Computação.

Advisor: Erickson Rangel do Nascimento
Co-Advisor: Mario Fernando Montenegro Campos

Belo Horizonte

March 2016

Potje, Guilherme Augusto

P863o On the improvement of three-dimensional reconstruction from large datasets / Guilherme Augusto Potje. — Belo Horizonte, 2016
xx, 67 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de Minas Gerais – Departamento de Ciência da Computação
Orientador: Erickson Rangel do Nascimento
Coorientador: Mario Fernando Montenegro Campos

1. Computação. 2. Modelo digital de elevação. 3. Visão estéreo. 4. Reconstrução 3D. 5. Veículo aéreo não tripulado. I. Orientador. II. Coorientador. III. Título.

CDU 519.6*84(043)

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

On the improvement of three-dimensional reconstruction from large datasets

GUILHERME AUGUSTO POTJE

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROF. ERICKSON RANGEL DO NASCIMENTO - Orientador
Departamento de Ciência da Computação - UFMG

PROF. MARIO FERNANDO MONTENEGRO CAMPOS - Coorientador
Departamento de Ciência da Computação - UFMG

DR. GUSTAVO MEDEIROS FREITAS
Instituto Tecnológico Vale Mineração

PROF. JEFERSSON ALEX DOS SANTOS
Departamento de Ciência da Computação - UFMG

PROF. LUCIANO REBOUÇAS DE OLIVEIRA
Departamento de Ciência da Computação - UFBA

Belo Horizonte, 30 de março de 2016.

Acknowledgments

I would like to thank to all people who somehow contributed to the development of this work, and specially to these people:

- My advisor, Erickson R. Nascimento, who guided me through this journey. He have always been very helpful and immeasurably contributed to the development of this work in every single aspect. To him I owe my deepest gratitude;
- My coadvisor Mario F. M. Campos, who undoubtedly contributed for the improvement of this work in many ways;
- My colleague Gabriel D. Resende who worked with me in the project and made this work truly better and more complete;
- To my parents Marco and Elizabeth who always have been supportive;
- To all colleagues from VeRLab, specially Hector and Igor who worked with me on the ITV project, and Balbino who always assisted me in anything I needed in the lab.

I also want to thank Vale Institute of Technology (ITV) for founding this project, and all people from Vale who participated in the UAV mapping project. They were always prompted to help in anything we needed.

Resumo

O advento das câmeras digitais permitiu se estimar a estrutura 3D a partir de imagens que são adquiridas por estes dispositivos de forma rápida e barata. Ao longo dos anos, inúmeras técnicas surgiram, e os algoritmos do estado-da-arte agora são capazes de prover resultados a partir de sensores de baixo custo com qualidade e resolução comparável aos sistemas padrão da indústria. Câmeras atuais capazes de produzir imagens de alta definição são compactas, leves, e podem ser facilmente acopladas a veículos aéreos não tripulados (VANTs), em contraste a outros meios de aquisição de dados 3D, como LiDAR, que está associados a altos custos financeiros e logísticos. No entanto, o tempo de processamento das imagens coletadas rapidamente se torna proibitivo conforme o número de imagens de entrada aumenta, exigindo hardware poderoso e dias de tempo de processamento para se gerar modelos 3D de grandes conjuntos de dados. Neste trabalho, é proposta uma abordagem eficiente baseada na técnica de estrutura a partir do movimento incremental (*Structure-from-Motion*) e técnicas de reconstrução estéreo para gerar automaticamente MDE - Modelos Digitais de Elevação - a partir de imagens aéreas e também modelos 3D em geral. A abordagem proposta usa a informação de GPS para inicializar a estrutura de grafo usada no algoritmo, uma pontuação baseada em árvore de vocabulário para reduzir o número de pares a serem considerados na etapa de correspondência, uma técnica de filtragem de pontos de interesse na imagem que mantém a alta repetibilidade de pontos e reduz o custo computacional, e múltiplas otimizações locais em vez da clássica otimização global é empregado em um novo esquema para acelerar o processo incremental de estimação. Resultados obtidos com seis grandes conjuntos de imagens aéreas obtidas por VANTs e quatro conjuntos de dados terrestres mostram que a abordagem adotada supera as estratégias atuais em tempo de processamento, e também é capaz de proporcionar resultados equivalentes ou melhores em precisão comparado com três métodos do estado-da-arte.

Palavras-chave: Modelo Digital de Elevação, Visão Estéreo, Reconstrução 3D, Veículo Aéreo Não Tripulado, Estrutura a partir do movimento.

Abstract

The advent of digital cameras heralded many possibilities of structure and shape recovery from imagery that are quickly and inexpensively acquired by such devices. Throughout the years numerous techniques have emerged, and state-of-art algorithms are now able to deliver 3D structure acquisition results from low cost sensors with quality and resolution comparable to industry standard systems such as LIDAR and expensive photogrammetric equipments. Current imaging devices capable to produce high-definition images are compact, lightweight, and can be easily attached to unmanned aerial vehicles (UAVs), in contrast to other means of 3D data acquisition such as LiDAR, which is associated to high financial and logistical costs. However, the processing time of the collected imagery to produce a 3D model quickly becomes prohibitive as the number of input images increases, demanding powerful hardware and days of processing time to generate full DEMs of large datasets containing thousands of images. In this work we propose an efficient approach based on Structure-from-Motion (SfM) and Multi-view Stereo (MvS) reconstruction techniques to automatically generate DEM – Digital Elevation Models – from aerial images and also 3D models in general. Our approach, which is image-based only, uses the increasingly meta-data information such as GPS in EXIF tags to initialize our graph structure, a keypoint filtering technique to maintain high repeatability of matches across pairs and reduce the matching effort, a vocabulary tree score to reduce the space search of matching and multiple local bundle adjustment refinement instead of the global optimization in a novel scheme to speed up the incremental SfM process. The results from six large aerial datasets obtained by UAVs with minimal cost and four terrestrial datasets show that our approach outperforms current strategies in processing time, and is also able to provide better or at least equivalent results in accuracy compared to three state-of-the-art methods.

Keywords: Digital Elevation Model, Multi-View Stereo, 3D Reconstruction, Unmanned Aerial Vehicles, Structure-from-Motion.

List of Figures

List of Tables

Contents

List of Acronyms

Acronym	*Description*
AC-RANSAC	*A contrario RAndom Sample Consensus*
BA	*Bundle Adjustment*
DEM	*Digital Elevation Model*
ICP	*Iterative Closest Point*
IMU	*Inertial Measurement Unit*
ITV	*Instituto Tecnológico Vale*
GPS	*Global Positioning System*
LBA	*Local Bundle Adjustment*
LiDAR	*Light Detection And Ranging*
MLE	*Maximum Likelihood Estimator*
MST	*Maximum Spanning Tree*
MvS	*Multi-view Stereo*
NCC	*Normalized Cross-Correlation*
ORB	*Oriented Brief (Keypoint detector & descriptor)*
RANSAC	*RANdom SAmple Consensus*
RMSE	*Root Mean Squared Error*
SfM	*Structure-from-Motion*
SIFT	*Scale Invariant Feature Transform (Keypoint detector & descriptor)*
SURF	*Speeded Up Robust Features (Keypoint detector & descriptor)*
SBA	*Sparse Bundle Adjustment*
TLS	*Terrestrial Laser Scanner*
UAV	*Unmanned Aerial Vehicle*

List of Parameters

Parameter	*Description*
$\mathbf{E_{ij}}$	*Essential matrix of image pair ij*
$\mathbf{F_{ij}}$	*Fundamental matrix of image pair ij*
$\mathbf{K_i}$	*Intrinsic calibration matrix of image i*
$\mathbf{P_i}$	*Projection matrix of image i*
$coarse\_inlier\_rf$	*Inlier ratio of the coarse fundamental matrix estimation step*
$contrast\_threshold$	*Contrast threshold used in the SIFT detector algorithm which is used to evaluate the keypoint quality.*
$d\_nearest$	*Amount of the closest images that are kept for each vertice in the initial graph using the GPS coordinates*
k_top	*Amount of the biggest scaled keypoints used in the coarse estimation for the fundamental matrix*
MAX_RE	*Threshold in pixels to consider if a camera was correctly estimated considering its threshold_resec*
$threshold\_fm$	*Inlier threshold value in pixels for the fundamental matrix estimation*
$threshold\_resec$	*Threshold value estimated for the AC-RANSAC based camera resectioning in pixels*
τ	*Minimum amount of inliers a pair must have to be considered geometrically consistent in order to be present in the final epipolar graph*
voc_tree_bf	*Branching factor of the tree used in the vocabulary tree based image recognition approach*
$window\_size$	*Number of cameras that are optimized in the local bundle adjustment*

Chapter 1

Introduction

Geometric reconstruction of the world from a sequence of images remains one of the key-challenges in Computer Vision. Three-dimensional recovery of the geometry of an object or a scene has several applications in Computer Vision and Robotics, such as scene understanding [Li et al., 2009], object recognition and classification [Belongie et al., 2002] [Gehler and Nowozin, 2009], digital elevation mapping and autonomous navigation, to name a few.

In Robotics, 3D information is crucial to mobile robots that navigates autonomously in the environment, because it gives much more information about the ambient [Wurm et al., 2010]. Semantic mapping is one of the computer vision techniques that uses 2D-3D information to extract high-level features of the environment that can improve the agent's decision [Henry et al., 2010].

Due to the recent development of low-cost RGB-D sensors, semantic mapping techniques have been gaining more attention because of the easy accessibility of these devices that can be attached in the robots and provide reliable 2D and 3D information that together can be explored efficiently to generate semantic maps [Hermans et al., 2014]. However, these devices, *e.g.* the Kinect, which provides linked radiometric and depth information, only works well in estimating surface depth that is within a certain distance range of the sensor in indoor environments (3 to 5 meters at most), and much less in outdoor scenes in daylight, because they are generally based on infrared emission, limiting the use of such sensors in a myriad of real world problems. Aligning the sensor readings in a global frame (registration) without the use of accurate Inertial Measurement Units (IMU) is another challenge [Henry et al., 2012]. In contrast, image-based 3D acquisition can be widely applied in many types of environment with consumer grade cameras, which are increasingly accessible to everybody nowadays. Image based techniques can be applied both indoor and outdoor with no restrictions

at all, when there is enough baseline, overlap and texture present in the acquired images [Westoby et al., 2012].

Light Detection and Ranging (LiDAR) systems generally require accurate IMU and GPS rigidly attached and well-calibrated to obtain a global reference frame of the sensor readings, which makes the use of such approach in a campaign, expensive [Liu, 2008]. Although, recent methods based on laser [Bosse et al., 2012] are able to provide accurate 3D reconstruction results without the the requirement of GPS, and is applicable to both indoor and outdoor environments. However, LIDaR based systems are only able to measure depth and the intensity of the returned pulse, and texture information can not be directly obtained from the data.

For applications, such as aerial mapping, for instance, image-only based pipelines that incorporate recent SfM (Structure-from-Motion) and Multi-view Stereo (MvS) techniques are strong competitors to LiDAR based surface measurements [Leberl et al., 2010]. Two of the advantages of image-based reconstruction when compared to LiDAR is that several mapping tasks may also require digital images of the scene, and radiometric information is directly registered with depth.

In particular, approaches that estimate DEMs using only images gained attention recently, specially due to the increasing availability of high quality cameras and of UAVs. Camera equipped UAVs are a low-cost and lightweight autonomous platforms that can be readily applied to acquire data processed by software packages, generating full three-dimensional models of outdoor scenes in remote areas [James and Robson, 2012]. In addition, these easy-to-use platforms can allow people with no knowledge at all in Robotics and Computer Vision to use complete image-based 3D reconstruction pipelines with minimal costs. Figure 1.1 shows a DEM estimated by our approach, and Figure 1.2 shows a sparse point cloud in VisualSFM's graphical user interface.

A large number of techniques for recovering 3D data which describes the geometry of a scene or an environment have been proposed in the literature. In the past few years, state-of-the-art techniques in 3D photography and laser-based sensors have set the bar in accuracy in the order of a few centimeters in elevation measurement [James and Robson, 2012]. Recent methods based only on sequences of images attained significant improvements, thanks to the advancement of camera sensor technology and computer vision techniques.

Recent technological advances in imaging sensors enabled the production of high resolution, low cost digital cameras which can provide high quality images and also meta-information such as noisy GPS and camera intrinsic parameters. However, the high resolution images and the rich set of features output by these cameras impose heavy time and memory constraints for the use of state-of-the-art methodologies in

Figure 1.1. DEM reconstructed by our pipeline followed by a sample of 4 images out of 220 used in the estimation from a construction site near ICEx. The model was obtained with minimal budget using a smartphone camera and a low-cost quadrotor.

large datasets obtained from their data stream.

A relatively new approach developed in Computer Vision field called Structure-from-Motion significantly advanced in accuracy and scalability in the past few years, and is able to retrieve the camera parameters and an initial sparse set of 3D points. Given sets of point correspondences between image pairs and the intrinsic parameters of each image, SfM pipelines are able to compute a global consistent pose (translation and orientation vector) for each image (where they were taken in the 3D space) up to an undetermined scale and an arbitrary coordinate system. Although, challenges and problems still remain unsolved. Some of the problems are that these approaches can provide wrong results when the optimization of the parameters get stuck at local minima, and the processing cost of the optimization step still is too costly. Limitations are also present, for example the lack of texture in the images makes image-based techniques useless in some cases.

After a structure-from-motion algorithm is used to obtain a 3D model from the images, a similarity transform can be used to geo-reference the 3D model into a known global reference system and scale, allowing one to measure distances in the scene in a metric scale. This can be performed by using ground control points and inaccurate GPS

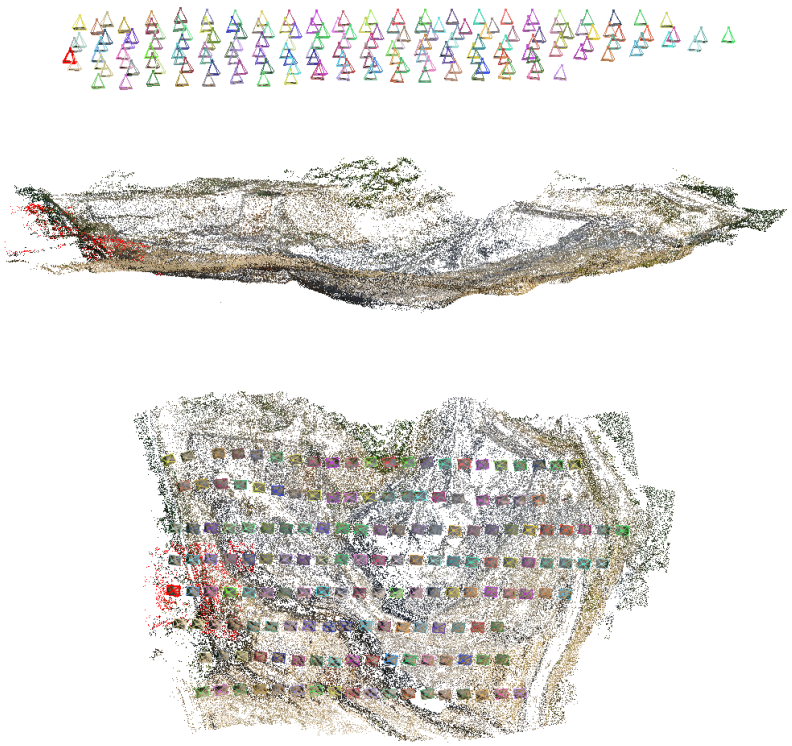


Figure 1.2. VisualSFM [Wu, 2013] graphical user interface showing two different view points of the same reconstructed model. The colored rectangles represents the poses of the cameras in 3D. The software, which is free for research purposes, provides a friendly interface covering the complicated structure-from-motion algorithm, and allows people with no expertise in computer vision to use advanced techniques to estimate 3D models from images.

data. This promoted a huge advancement in 3D photography, and now it is possible to obtain 3D models without many prerequisites previously required in photogrammetry, such as structured acquisition of images, expensive and accurate GPS/IMU devices rigidly attached to the camera sensor, resulting in a costly and hard to apply approach to small and medium size mapping projects. Semi-automatic selection of point measurements in overlapping images were also required, increasing even more the costs [Irschara et al., 2012].

Nowadays, even consumer-grade cameras have considerable amount of resolution, with sufficient radiometric and geometric stability to be used for 3D reconstruction, depending on the accuracy needed [James and Robson, 2012]. Ultimately, with the popularization of UAVs, high resolution image acquisition using these vehicles showed to be a high cost-efficient and automated method that summing with all above qualities mentioned makes 3D photography the most time-and-cost efficient method to obtain digital terrain models [Westoby et al., 2012].

However, in general the processing time of SfM methods increases non-linearly with respect to the number of pictures, which makes the processing time for most real outdoor scenes, such as open-pit mines and large areas of cities, undesired, specially on consumer-grade computers. The complexity of the image registration step, which is at the core of SfM algorithms, has a time complexity of $O(n^2)$ when using a brute force approach, where every image is tested against all the dataset to validate the geometry of a valid correspondence. Since this step is expensive in terms of processing time, naive methods already boggle down with just few hundreds of images, and become prohibitively slow when thousands of images are considered. In addition to the registration step, another barrier to be overcame is the costly non-linear optimization of the camera parameters required in SfM methods called Bundle Adjustment (BA). This step is extremely important to avoid drifting during the reconstruction and provides the optimal solution for the SfM problem, being the maximum likelihood estimator for camera pose and 3D points considering that the measured points in the images have a Gaussian noise in their position.

Thus, despite the advances of SfM methods that estimate the three-dimensional structure from a sequence of images, there are still several challenges that need to be overcame to compute high quality 3D data from a large number of high definition images.

In this thesis, we combine several ideas in a novel scheme, which some of them have already proven to be efficient separately, like making use of GPS information [Frahm et al., 2010], considering only the most discriminant features of the images to make a coarse estimation of the pair geometry [Wu, 2013], the use of nearest neighbour search of the corresponding features of the valid pairs, and leveraging the $O(\log(n))$ complexity of the vocabulary tree search to speed up the matching phase [Nister and Stewenius, 2006]. These steps combined outperform the previously proposed approaches individually, and drastically reduces the time required to perform the matching step when compared to the brute force approach. Also, our proposed pipeline detailed in the methodology section is able to meticulously select image pairs with high quality of overlapping area. These steps can decrease the possibility of using bad pairs in the incremental structure and motion estimation step, and drastically reduces the space search of the problem, speeding up the epipolar graph build time.

Another contribution of our work is the proposal of an overlapping local bundle adjustment (LBA) window approach, since for large datasets, the global optimization can hugely contribute for the slowness of the process. We locally optimize the camera poses and points, but we also consider previously done bundle adjustment steps to maintain the consistency of the model, which is the novelty in this approach.

Therefore, as shown in Chapter 5, our approach is capable of computing the DEM faster than the other methods used in the experiments in all the tested datasets while preserving the quality of the results.

It is worth mentioning that this thesis is the result of a subproject founded by Instituto Tecnológico Vale (ITV), which is contained in a bigger context. The full idea of the project is to create a complete system that is able to map a remote area using cooperative coordination among many micro aerial vehicles and build 3D maps of the environment that also have other relevant information associated to them, *e.g.* magnetic information obtained by magnetometers.

1.1 Objective and Contributions

Our goal was to develop a SfM algorithm capable of producing accurate results comparable with the state-of-the-art SfM techniques, while focusing in time performance gains.

The general contribution of our work is the development of a new SfM pipeline that is able to deliver high quality DEMs at a low processing time cost. The main contributions can be summarized as follows:

- Proposal and implementation of a new SfM pipeline which incorporates and adapts the best techniques, both focused in time performance and accuracy, into a single algorithm;
- Proposal and implementation of an adapted overlapping local bundle adjustment window approach for large-scale datasets;
- A comparison and analysis of state-of-the-art softwares used in aerial mapping with large real world datasets acquired by UAVs.

1.2 Thesis Organization

In Chapter 2 we explore and discuss the main concepts of 3D reconstruction techniques, including the projective reconstruction theorem and MvS dense reconstruction techniques.

In Chapter 3 we review and detail recent state-of-the-art SfM techniques present in the literature.

In Chapter 4 we present the proposed pipeline, divided in five main steps.

Sequentially, Chapter 5 contains the experiments and results obtained by exhaustively testing seven datasets with the proposed approach and three state-of-the-art SfM implementations.

Finally, in Chapter 6 we discuss the results achieved.

Chapter 2

Theoretical Background

In this chapter, we explore and detail important concepts and techniques used in 3D photography, which is the basis for all approaches of 3D reconstruction using collection of images.

2.1 Epipolar Geometry

The epipolar geometry in stereo vision is the intrinsic projective geometry between two pinhole cameras that view the same static 3D scene, where the geometric relations between the 3D points and their 2D perspective projections onto the images are constrained by the camera internal parameters and their relative pose [Hartley and Zisserman, 2004].

The fundamental matrix $\mathbf{F_{ij}}$ is a 3×3 matrix of rank 2 which describes the geometry between a pair of images i and j according to the corresponding points that are consistent with the epipolar geometry constraint:

$$\mathbf{x_j^T F_{ij} x_i} = 0, \tag{2.1}$$

where $\mathbf{x_i}$ and $\mathbf{x_j}$ are the projected 2D coordinates of the same 3D point in pixels in the images taken from different viewpoints.

With $\mathbf{F}$, we can map a point in the left image to a line (namely, epipolar line) in the right and vice-versa. Figure 2.1 depicts this constraint. The camera motion can be obtained from $\mathbf{F}$ up to a projective transformation. If the intrinsic parameters of the cameras are known, we can obtain a new matrix called the essential matrix $\mathbf{E}$ which is a special case of the fundamental matrix where the intrinsic calibration matrices are identity matrices (this happens if we normalize the image coordinates by multiplying

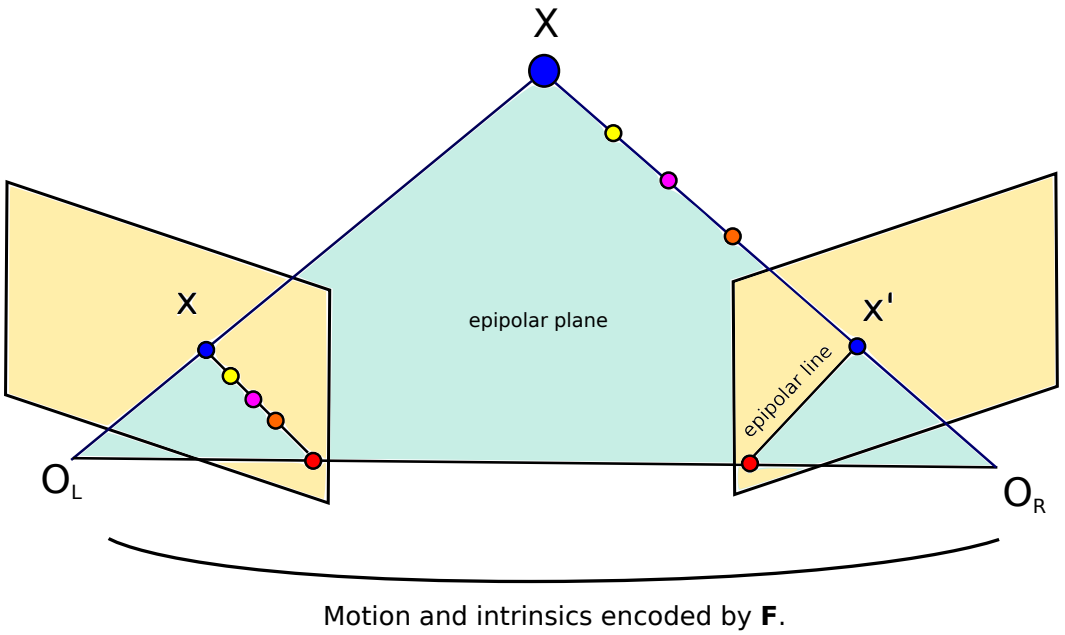


Figure 2.1. Epipolar geometry between two cameras. Given the 2D coordinate $\mathbf{x}'$ in the right image, which is the perspective projection of $\mathbf{X}$ onto it, the same projection in the left image is constrained by the epipolar line, and must project along the line segment. The epipolar line of each camera can be seen as the intersection of the epipolar plane and the respective camera plane. The epipolar plane is defined by the optical center of the cameras and the 3D point, in which their 2D projections in both camera planes also lies in the epipolar plane, so this geometry does not depend on the scene structure. Note that there is an infinite number of possible epipolar lines as we move $\mathbf{X}$ in the 3D space, and just one case is represented in the image.

the points by the calibration matrices or the fundamental matrix itself). We can update the fundamental matrix to the essential matrix by using the calibration matrices:

$$\mathbf{E_{ij}} = \mathbf{K_j}^T \mathbf{F_{ij}} \mathbf{K_i}, \tag{2.2}$$

where $\mathbf{K_j}$ and $\mathbf{K_i}$ are the respective calibration matrices of cameras i and j. The calibration matrix is a 2D transformation matrix in the form:

$$\mathbf{K} = \begin{pmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}, \tag{2.3}$$

where (f_x, f_y) are the focal length of the camera in pixel units (usually they have the same value) and (c_x, c_y) is the center of projection expressed in pixels. Finally, s is the skew factor and is usually 0. With $\mathbf{E}$, it is possible to recover the relative euclidean

motion of the two cameras in the 3D space up to an ambiguity of scale, because we can only recover the direction of the relative translation from it.

The fundamental and essential matrix can be estimated from a set of point correspondences between two images (normalized image coordinates, in the case of the essential), and robust techniques like the normalized eight-point algorithm [Hartley, 1997] and the five-point algorithm [Nistér, 2004] developed in the past years allow a robust estimation from noisy image coordinates.

2.2 Structure from Motion (SfM)

In general, SfM pipelines are based on feature matching and stereo vision techniques. Recent advancements in robust feature detection and matching across images allowed the use of stereo methods that can be used to estimate the extrinsic parameters between each valid pair of cameras (sharing a portion of view in the scene) automatically. SfM core methods take as input the relative extrinsic parameters between the pairs and put

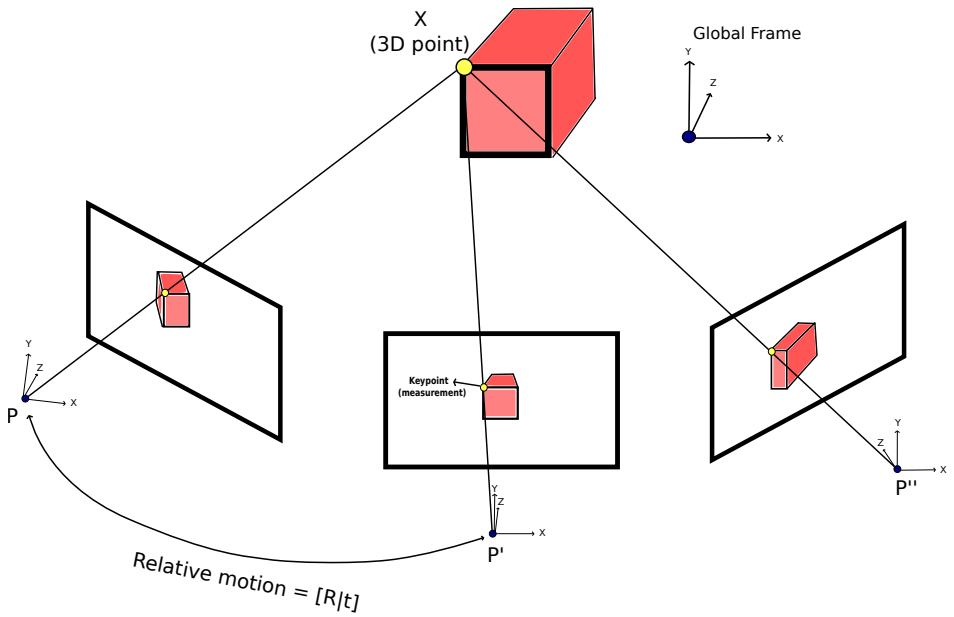


Figure 2.2. An example of the SfM problem. We want to find the projection matrices $\mathbf{P}, \mathbf{P}'$, $\mathbf{P}''$ and $\mathbf{X}$ that are in the same global frame, so that we can project $\mathbf{X}$ into the 3 images using their respective projection matrices and the error between the projected and measured 2D coordinates of the same 3D point in the scene is minimal.

them into a single reference coordinate system [Snavely et al., 2008b]. In Figure 2.2, we have a simple example of the structure-from-motion problem. Having the projection matrix for each camera relative to a global frame, it is possible to use MvS techniques to obtain a dense 3D model of the scene.

One of the most representative works that inspired numerous existing cutting-edge state-of-the-art SfM techniques until now is the well known Bundler software [Snavely et al., 2008a], which can handle a few hundred of images in a time span of days, in a consumer-grade computer. The authors used images from the Internet to create 3D models of well-known world sites, *e.g.* Notre Dame church, the Coliseum in Rome and the Trafalgar's Square, obtained from social media websites.

2.2.1 Epipolar Graph

Keypoint matching is one of the most time consuming steps in a SfM pipeline. Techniques like ORB [Rublee et al., 2011], SURF [Bay et al., 2008] or SIFT [Lowe, 2004] and many others can be used to detect and match points across images, which requires a lot of computational effort to process high resolution photographs in the detection and description phase. In addition, the matching step, that requires comparing each descriptor of the keypoint in one image to all descriptors of all keypoints in the other to find its nearest neighbor, is also another time bottleneck in this phase, requiring $O(n^2)$ comparisons between high-dimension descriptor vectors to perform the match of two images optimally, considering that n is the number of keypoints in the images.

The epipolar graph is widely used to represent the geometric relation between each pair of image in the scene and can be defined as follows: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each vertex $v \in \mathcal{V}$ represents an image and there is an edge $e \in \mathcal{E}$ between two vertices if there is a valid epipolar geometry relation between the images which is described by the fundamental or essential matrix. A simple epipolar graph is shown in Figure 2.3.

In the naïve approach, each image is matched against all other images in the dataset using a brute force nearest neighbor search for each possible pair to attribute the correspondence for each point, and then RANSAC [Fischler and Bolles, 1981] is used to robustly estimate the essential matrix between each pair, as also to remove the outliers from the correspondences.

2.2.2 Bundle Adjustment

Besides the feature matching and geometric validation step, another bottleneck of SfM approaches is the optimization of the camera parameters and 3D points, required in

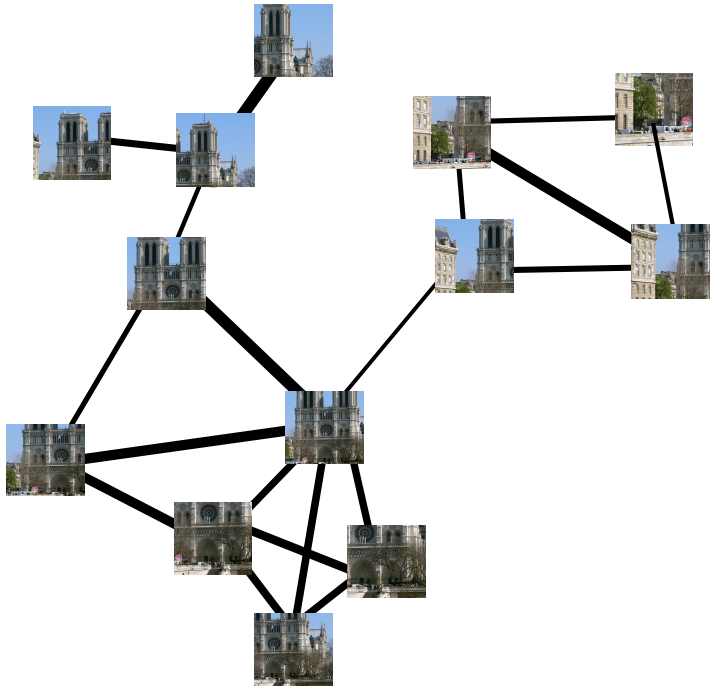


Figure 2.3. An epipolar graph with 13 images from Notre Dame. The thickness of the edges indicates a higher number of correspondences between pairs.

all approaches with no exception, to reduce drifting and improve accuracy. Once new camera poses are estimated and their 2D features are triangulated into 3D points, there is a need to optimize these estimated parameters due to estimation errors which accumulates in the model. The optimal solution for this problem considering Gaussian noise in the position of the keypoints, is the maximum likelihood estimator (MLE).

In this optimization problem, we want to estimate the camera poses and 3D points that minimizes the squared re-projection error in pixels between the measured and predicted projections. Considering that each camera pose j is parameterized by a projection matrix $\mathbf{P_j}$ and each 3D point i by a vector $\mathbf{X_i}$, we can write the optimization problem as minimizing:

$$\min_{\mathbf{P_j},\mathbf{X_i}} \sum_{i=1} \sum_{j=1} \parallel \mathbf{X_i P_j} - x_{ij} \parallel^2, \tag{2.4}$$

where $\mathbf{X_i P_j}$ is the predicted projection of point i on image j, and x_{ij} is the measured 2D coordinate of the projection. The optimization step requires an initial guess of the camera parameters and 3D points which is provided by the SfM algorithm, and since it is based on iterative methods that solves for a non-convex cost function in a high-dimensional non-linear parameter space, it can be stuck at a local minima if a bad initial solution is provided.

Considering that the parameters of each projection matrix has 6 degrees of freedom (three for the position and three for orientation) and each 3D point has 3 degrees of freedom (X,Y and Z coordinates), the total dimensions of the problem to solve can be calculated as $6j + 3i$ in this case. Other parameters as focal length, principal point and distortion coefficients can also be considered in the optimization, increasing even more the number of parameters of each camera. For large datasets, such as the ones containing thousands of cameras and millions of 3D points, the dimensionality of the problem is extremely high, and minimizing its cost function demands highly specialized algorithms that need to be extremely efficient and well-implemented.

In spite of the efforts of the community and the improvements already made, specially in exploiting the sparse block structure that arises in bundle adjustment to speed up the computation [Lourakis and Argyros, 2009], the problem is still costly to solve for large datasets. However, Eudes and Lhuillier [2009] shows that using a local bundle adjustment instead performing a global optimization in the incremental process on video-based reconstructions can achieve good quality results and provides a considerable speed up gain. Another solution for this problem is to use a divide-and-conquer approach [Ni et al., 2007], which can also accelerate the optimization while maintaining good accuracy relative to global bundle adjustment.

2.3 Multi-view Stereo Algorithms (MvS)

Multi-view stereo algorithms take as input fully calibrated intrinsic and extrinsic camera parameters and their respective images, and generate a *quasi*-dense 3D model based on correspondences between images. They can be roughly classified into four classes according to the underlying object models, being them shape-from-sillouetes, voxel-based, patch-based and graph-based. Each of them has its limitations, specially for considering some assumptions that are not general for every scenario. This limits the dataset type a technique can be applied, being them object and scene datasets.

A key process in all kinds of these algorithms is to check the consistency of the projected 3D points into the images [Furukawa and Ponce, 2010]. The limitation of these techniques is that they provide poor results when weak texture regions in the images and occlusions are present, because they use the intensity information to perform the consistency checks. All below techniques assumes that the intrinsic and extrinsic parameters of each camera are already known, which can be obtained by manual calibration or structure-from-motion techniques.

2.3.1 Photo-consistency

Photo-consistency tests are widely used in the techniques described below. Such tests are based on color or greyscale variance information that can be used as constraints in 3D reconstruction as a valid three-dimensional point in a world's surface as it's projection onto the visible cameras will theoretically have the same intensity or color, considering small variations of illumination and Lambertian reflectance. One of the most used scores to measure the similarity of patches in images is the normalized cross-correlation, which is given by the formula:

$$NCC = \frac{1}{n} \sum_{x,y} \frac{(f(x,y) - \mu_f)(t(x,y) - \mu_t)}{\sigma_f \sigma_t}, \tag{2.5}$$

where f and t are two corresponding patches in the images, n the total amount of pixels in the patches, and σ and μ the respective standard deviation and mean of the patch. A fixed threshold is usually set, and patches are declared inconsistent if they are not similar enough.

2.3.2 Voxel-based Approaches

In voxel-based approaches, a bounding box containing the volume of the scene is initialized, and every unit of this volume is formed by a *voxel*. We can do a direct analogy of voxels as being 3D pixels, like we have pixels in ordinary images. They have 3D coordinates and a color, like a pixel in an image has 2D coordinates and also a color. But this limits the technique, because there is a need to know a valid volume containing the scene, limiting the technique to object datasets only, and the quality of the model as also the computational cost is dependent of the resolution of the voxel space. Then, an iteration is made to verify the photo-consistency of the voxel, achieved by projecting the voxel onto all camera planes that can see it, and color variance is analyzed to determine if the voxel is photo-consistent or not. In case it is not, it is removed from the volume space.

The first attempts in reconstruction of 3D shapes through images used the silhouettes as source of shape characteristics. Szeliski [1993] and Niem and Buschmann [1994] proposed methods that uses calibrated cameras to produce 3D object models using the silhouettes. First, images of the object are taken in different poses around it, then, segmentation techniques are used to extract the silhouettes. Finally, the shapes of the silhouettes are used to define the volume intersection of the model generating the 3D shape of the object. But as proved in the work of Laurentini [1997], the sil-

houette information has not enough information to converge into the real shape of the object, depending on the shape of it, even when there is a possibility to obtain infinite number of images of the object in all possible poses. Another limiting factors of shape-from-silhouettes techniques are the number of images necessary to provide a good approximation of the shape, and also the pose of the cameras of the images. If there is too few images available, or a bad distributed viewpoints of the object, these kind of approaches can provide very rough results. A method presented by Shanmukh and Pujari [1991] considers some prior knowledge of the object shape and provides a solution that optimises the reconstruction specifying the viewpoints necessary.

2.3.3 Multiple Depth Maps

These techniques rely on estimated depth maps for each pair of image. Once the depth maps are obtained using stereo algorithms, they are merged onto a single model. These kind of techniques are simple and more flexible but requires many well-distributed views of the object to achieve good results. An example of the power of this technique, Irschara et al. [2012] developed a full methodology to obtain a dense model from large scale and highly overlapping aerial images. The core component of the approach is a multi-view dense matching algorithm that explores the redundancy of the data. A multi-view plane sweep technique is applied to perform the match, where the 3D space is iteratively traversed by parallel planes which is usually aligned with a particular key view. For each depth in the plane, sensor images are projected onto the plane and a similarity function is used to compute a cost. After, a depth map can be extracted using a minimum graph cut algorithm. The final result of the approach is a model with depth value estimated for every possible pixel in the images.

2.3.4 Patch-based Methods

Being one of the most flexible techniques, patch-based approaches can achieve good results in the majority of datasets (objects and scenes), except in texture-less or occluded regions. These approaches first generate a set of sparse 3D oriented points using feature matching correspondences across images, and then iteratively expand these patches to increase density.

Techniques based on patches are the most versatile, being robust to calibration errors and does not need any prerequisite like initializing a visual hull, bounding box or valid depth ranges, not being restricted only to object datasets. The approach presented in [Furukawa and Ponce, 2010] shows a hybrid approach that outputs a dense

collection of small oriented rectangular patches obtained from pixel-level correspondences considering the images and their respective calibration. The algorithm consists of a simple match, expand and filter procedure, and as first step patches are created considering sparse points obtained with feature matching techniques, and their orientation are calculated considering the cameras centers that observe the point. Then, cells are calculated by the projection of the patch in all cameras. A cell $C = (x, y)$ is defined by a window with size m with its coordinate at the central pixel. The expansion is done by creating other cells in the region of an existing cell, and photo-consistency tests are realized as well as occlusion tests to determine if a patch will be created or not, considering the new cells created (filtering step). Another similar technique is presented in [Goesele et al., 2007], which also uses patches, also known as *surfels* to reconstruct the surface of the scene. In a similar way, the initial sparse set of surfels are obtained by the result of a sparse reconstruction of a structure from motion approach as well as the cameras parameters. The algorithm iteratively grows surfaces through the initial sparse set, optimizing surface normals within a photo-consistency measure, which significantly improve the matching. The results obtained are very accurate, and direct comparisons between a model generated with laser scan and the technique shows the robustness of the technique.

Chapter 3

Related Work

In this chapter, we discuss relevant methods present in the literature that try to solve the Structure-from-Motion problem, pointing out the advantages and disadvantages of each one.

3.1 Structure-from-Motion

Incremental SfM reconstruction techniques aim at solving the problem incrementally. Our method fits in this category, being an extended pipeline, that is, in the end of the pipeline we also have the dense reconstruction and the estimated mesh with the projected textures.

Incremental SfM approaches are able to handle unordered collection of images. In other words, they do not make any assumption of temporal sequence in the frames, do not require high redundancy of images, and do not rely in any loop closing technique, since the feature tracking among cameras occurs globally considering the entire dataset. Such scheme allows SfM techniques to be robust, accurate and near-optimal in most cases after global bundle adjustment, although it might get stuck at a local minima eventually.

The incremental approach gained attention in the past ten years, because of its robustness to outliers (wrong correspondences and relative motion estimation) and missing data, such as the absence or wrong intrinsic parameters. Other methods such as factorization-based [Tomasi and Kanade, 1992] and global SfM [Crandall et al., 2011] can provide results in less time than incremental SfM because they do not need to optimize the model constantly, however such methods tend to be very sensitive to outliers and missing data, and generally cannot be applied to images in the wild, such

the ones obtained on the Internet or datasets commonly gathered by inexperienced people.

Bundler [Snavely et al., 2008a] is based on incremental SfM. First, features and meta-data are extracted for each image, and then an exhaustive brute-force matching of features is performed between each possible pair. Then, the fundamental matrix is estimated and finally the incremental reconstruction is performed by adding new cameras in a greedy manner through camera resectioning and triangulating new points with the before estimated cameras. The major drawback of the approach is the number of images that can be used for the reconstruction, which is bounded to a few hundreds, since the time required to process more than that rapidly becomes prohibitive due to both brute-force matching and multiple global BA calls that are required during the reconstruction. In the end, the pipeline provides the projection matrices for the images and a sparse point cloud.

The method of Frahm et al. [2010] is applicable to the structure and motion estimation of large-scale datasets. To deal with the high redundancy of images from image queries from the internet, they first clusterize the images and for each cluster they consider just an iconic image. Then, a result retrieved from a vocabulary tree search is used to perform the feature matching and geometry validation of the k closest images to the query image defined by a similarity score, giving a huge speed up of the epipolar graph building. However, their method tends to reconstruct unconnected clusters consisted of subsets of the original dataset.

The geo-location occasionally available can also be exploited to deal with the problem of fragmented models generated by large-scale SfM. Strecha et al. [2010] leverage the geo-location available, among other meta-data (DEMs and 2D building models), to deal with the fragmentation problem and also allow the update of the estimations when new images of the region become available without the necessity of redoing the process from scratch. But the method depends on reliable information to generate good results, such as accurate GPS.

Other approaches take advantage of the meta-data increasingly available in recent imagery to deal with the struggle in the matching step. Agarwal et al. [2009], which also focus on large-scale internet photo collections, use the noisy geo-location of the images to remove comparison between far away pairs, performing the matching only between close cameras, selected according to an arbitrary threshold, reducing efforts of the image matching step. However, their methodology requires a computer cluster with hundreds of CPUs to provide the complete solution within the time span of a day.

Since vocabulary tree approaches may return ambiguous pairs that can induce the reconstruction to fail, resulting in wrong models, Irschara et al. [2011] uses the

meta-data to compute a coarse projection matrix for each camera using the available GPS and IMU data. By means of a pre-existing 3D model of the scene or making weak assumptions on its maximum depth, the method is able to estimate a coarse overlap between the images. Thus the feature match process occurs among the ones with overlapping views, improving significantly the time performance compared to the brute-force matching and avoiding ambiguity. However, IMU data and pre-existing models are not commonly available, limiting the applicability of this technique.

In order to overcome the costly matching step, the work of Wu [2013] tries to reduce the time consumption by using approximate nearest neighbor search and carefully selecting subsets of *keypoints* to be matched. For a moderate number of images (few hundreds), this approach is efficient, but it does not avoid the quadratic complexity of matching. The authors also use preconditioned conjugated gradient which can accelerate the convergence of the optimization in the bundle adjustment step. Furthermore, they explore the pleasingly parallelizable characteristics of the problem to speed up the process by using multi-core processors and GPUs. The results remain as one of the state-of-art techniques, although for very large datasets, the method requires powerful hardware, such as multiple GPUs and many threads to provide the solution in acceptable time.

Other works focused at improving bundle adjustment, which is an essential part of the SfM pipeline, thus, receiving intense research in the past years. Jeong et al. [2012] perform experiments with several bundle adjustment methods present in the literature, and proposes two methods that work in the reduced camera system that leverages the natural block sparsity. While one is based in exact minimum degree ordering and block-based LDL (lower triangular and diagonal matrix decomposition) solving, the other uses a block-based preconditioned conjugate gradient. The reported results show that the methods are able to converge faster, in addition to handle memory efficiently. However, the strategies for the linear solvers were not fully investigated as pointed by the authors, and better results can be achieved with a proper investigation.

More recently, the work of Zhu et al. [2014] explores the idea that the way BA distributes the errors evenly may cause local areas to be sub-optimal, and re-optimizing them locally can improve the accuracy of the reconstruction. They use a divide-and-conquer approach to segment the model into well-conditioned regions (parts of the scene that are visible from many cameras) and re-optimizes them while also maintaining the global consistency. The result is that fine details that were be lost with the global optimization are now present in the re-adjusted model.

3.2 Video-based Methods: Visual-SLAM

Video-based methods, also known as Visual-SLAM, use similar stereo techniques to estimate the relative camera poses, but differently from the structure-from-motion for unordered collection of images, they use the temporal relation between the frames to skip the matching step. Using fast tracking techniques such as optical flow, the points are tracked in the most recent frames of a video stream, which requires a high frame-rate to maintain the relative change of frames very low. Generally, the images resolution are limited by these techniques if one wants to achieve real-time performance.

Because these methods aim at running in real-time, global bundle adjustment is undesirable at any point, and the uncertainty of the camera poses in long sequences is high. Consequently, they rely on loop-closure techniques to attempt drift correction on-the-fly, however, loop detection may be too expensive for large datasets and does not guarantee that all loops will be detected. Furthermore, the 3D model generated by these techniques can suffer from multiple estimations of the same 3D point, because they only keep track of the most recent features in the images, providing a sub-optimal solution which may lead to reduced global accuracy specially on large datasets.

Thus, these techniques are not usually used to estimate 3D models in general because of their limitation in accuracy, but they are commonly used in robotics to provide a coarse estimation of the robot's pose and the environment structure in 3D using low-cost cameras, being very useful in that case.

Pollefeys et al. [2008] proposed a real-time system that is able to deliver dense 3D information from a video stream of an urban area. They rely on accurate INS and GPS to provide the camera poses. Eight cameras were positioned in different points of view attached to a vehicle, and then a calibration was performed to compensate the difference of coordinate system of the cameras and the INS/GPS sensors. Using the camera poses provided by the sensors, a subset of frames with sufficient baseline are constantly selected, and a GPU implementation of the plane sweep algorithm originally proposed by Collins [1996] is used to achieve real-time 3D depth estimation.

More recently, Engel et al. [2014] proposed a video-based method that uses direct intensity comparisons of small-baseline consecutive frames to obtain semi-dense depth-maps. This step requires the frames to be extremely redundant. The algorithm constantly estimates relative camera poses based on those depth-maps, and solves for a graph-based pose optimization problem to obtain global camera poses, where each pose is parametrized by a similarity transform (pose and scale). These poses are updated if a loop is found to reduce scale drift using loop detection algorithms. The system works in real time using 640×480 resolution images, and the frame resolution is bounded if

one wants to achieve real-time results running in CPU. The technique also suffers from pose drifting for long sequence of frames, specially if no loop is found.

3.3 Structure-from-Motion versus LiDAR

The work of James and Robson [2012] applies a 3D photography methodology to geosciences. By using SfM and multi-view stereo (MVS) techniques, the authors generate models and compare them against laser scanned models. A consumer-grade camera, low cost UAVs and computer vision software were used in the experiments, and they concluded that the combination SfM+MVS is capable of producing useful 3D models with a decrease of 80% in the total time spent in a mapping campaign (considering the logistics until the final result) in comparison to LiDAR. Similarly, the work of Westoby et al. [2012] presents a survey concluding that 3D photo techniques are good options to produce topographic data in an efficient cost and low-time way, in contrast to the traditional surveying campaigns using lasers and manned aerial vehicles, which require high financial and logistical costs, and demand specialists to perform the data acquisition.

More recently, the work of Micheletti et al. [2015] demonstrates that even an ordinary smartphone camera with 5.0 megapixel resolution processed by a SfM pipeline is able to deliver satisfying results. The authors compare the models generated by SfM against those generated by a well established photogrammetric software and LiDAR. Their results reinforce that SfM approaches are a fully automated and inexpensive way to obtain reliable 3D information.

A survey performed by Teza et al. [2016] showed many advantages of SfM over terrestrial laser scanners (TLS) in morphological analysis for architectural applications. The first advantage is that when a photo-realistic representation is required, SfM is suitable while TLS requires additional camera sensors and rendering techniques. Another appealing advantage is that when UAV-mounted or lifting platform-mounted instrument is needed, TLS becomes unsuitable, while SfM is perfectly suitable for this task, since cameras can be easily attached to them. When a very fast survey is needed, SfM techniques demonstrated to be faster than campaigns using TLS, specially when a complex building has to be mapped. Besides, when there is limited available economic resources, TLS becomes unsuitable. A last advantage of SfM highlited by the authors is that the SfM pipelines are highly automated, while TLS requires some manual work to generate the 3D model. On the other hand, in the presence of trees or similar disturbances, SfM becomes unsuitable, as well as when observations of tall buildings or

night survey are required , while the TLS is partially suitable for these tasks.

Laser-based SLAM approaches are also gaining attention, since they do not require GPS to register the point cloud and robot's pose. A modern system proposed by [Bosse et al., 2012] uses a hand-held 2D range LiDAR sensor mounted on a spring, and an industrial-grade IMU rigidly attached to the LiDAR device to map the 3D environment. The device is built in such a way that when the operator moves through the environment, the sensor head is moved as the result of the motion induced by the spring, giving the 2D range sensor a 3D field of view. The SLAM software receives the range and IMU data and estimate the 6-degree-of-freedom robot's trajectory and also the 3D point cloud registered in a global frame. The point cloud based SLAM method first extract and match features based on the point normals, and then, similar to the classic Iterative Closest Point (ICP) algorithm, they minimize the distance between the matched points. The trajectory is optimized through a moving window of fixed size, which they call open-loop solution. A post-processing phase called closed-loop solution can be performed using the result of the open-loop solution, where all the trajectory is optimized in a single window. However, if the open-loop trajectory drifts too much, the closed-loop solution may get stuck at local minima, and loop-closure techniques may be required to generate a consistent model. One advantage of Laser-based SLAM is that it can handle textureless scenes, but they also suffer from the lack of features, in this case in the point cloud. Examples of scenes without features in the point cloud are long tunnels or plain regions, that lack abrupt normal surface changes.

Chapter 4

Methodology

In this section we detail the main steps of our methodology. It is a novel pipeline that provides two new features: An efficient epipolar graph building procedure and a local bundle adjustment adapted to large-scale reconstructions.

First, the GPS constraint in addition to the vocabulary tree score are used to efficiently prune non-overlapping pairs (Figure 4.1– I) followed by a coarse to fine geometry validation to save even more processing time in the feature matching phase (Figure 4.1– II). The epipolar graph's edges are then updated by the modifed maximum spanning tree algorithm (Algorithm 2) that carefully selects the best ones to be used in estimation of the camera parameters and the scene structure while enforcing the completeness of the graph (Figure 4.1– III). The camera motion and intrinsics, as well the 3D structure parameters are locally optimized by an overlapping window containing the most recent cameras (Figure 4.1– IV). As a direct consequence of using the proposed local bundle adjustment (LBA), our approach demands less global optimizations to provide an accurate solution in the end of the reconstruction. In the final step, our pipeline computes the dense model using a patch-based multi-view-stereo technique and Poisson reconstruction to obtain the final mesh (Figure 4.1– V).

4.1 Registration

In general, in the aerial image acquisition process, GPS data (even if noisy) will be available. UAVs require GPS to autonomously navigate through the environment, and the readings from the sensor can be directly registered with the images or obtained through smart-phones and cameras that already have GPS sensors built-in. Leveraging this meta-data, we can use it to reduce the space search of the matching step and avoid ambiguity in the scene to be considered. It is fair to assume that if the Euclidean

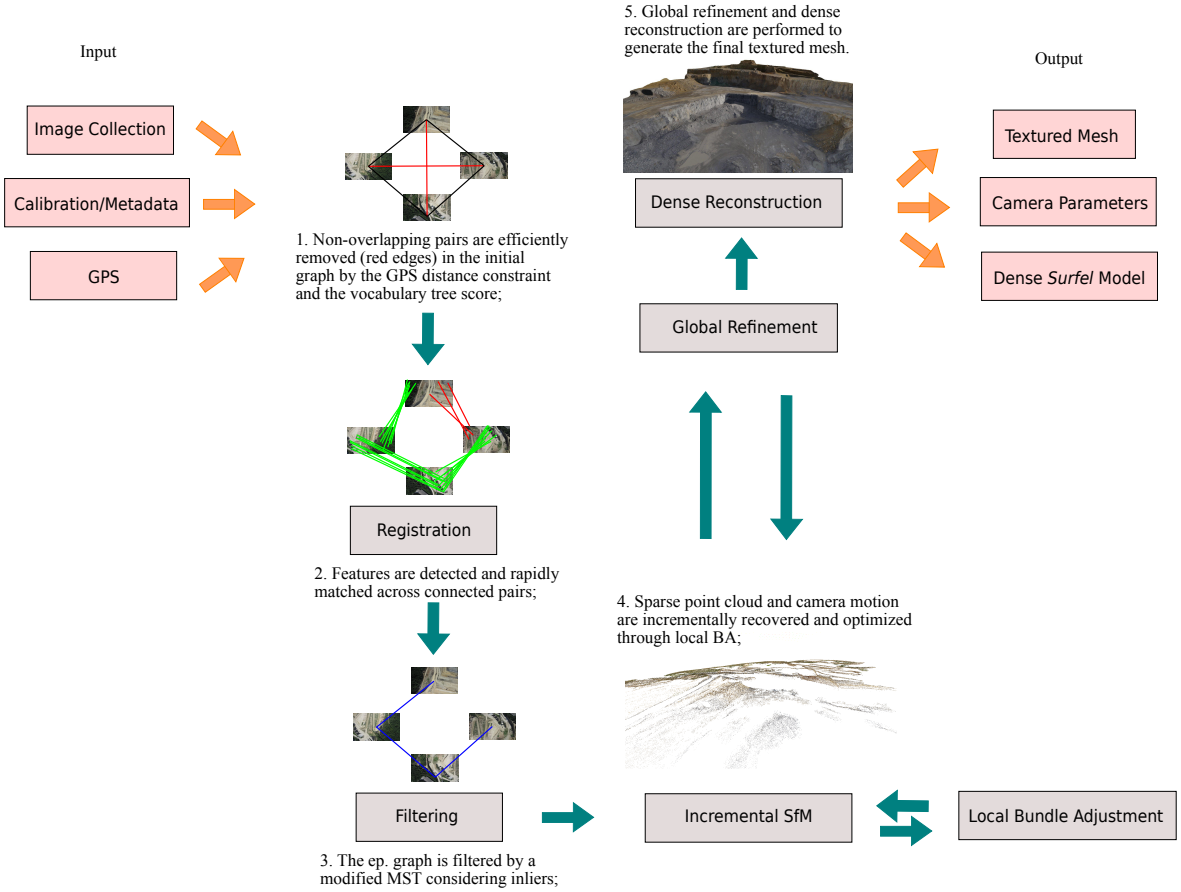


Figure 4.1. Illustration of the main steps of our methodology. We initialize the epipolar graph by connecting images with a large chance of having overlap, according to GPS data and a vocabulary tree search. In this example, the black edges are below the threshold distance, and the vocabulary tree query of at least one of the images are among the top 40 highest score matches of the other, so they are kept while the red ones are removed. After the optimized pairwise registration, we update the epipolar graph by selecting high quality matches enforcing completeness, here represented by the blue edges in step III. The camera motion is incrementally recovered for each image and a sparse point cloud generated from the matching points and optimized through robust and fast local bundle adjustment. At the end, we compute the dense model.

distance between the position of image pairs is large, they do not share any portion of view. By considering that, we generate an initial graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each vertex $v \in \mathcal{V}$ represents an image. We connect the $d\_nearest$ images according to the distance obtained by compairing each pairs' GPS coordinates. We used $d\_nearest = 40$ in our experiments, which is a sufficient value for all datasets in our experiments. Reducing this value can reduce even more the effort of matching although it can prune pairs that overlap.

The constraint increases the time performance and reduces the time complex-

ity of matching n images from $O(n^2)$ to $O(n)$ considering aerial and large datasets. Additionally, this avoids comparing ambiguous pairs, which makes the approach more robust to wrong reconstructions due to views that are actually geometrically consistent but are not viewing the same portion of the scene (*e.g.* symmetric building facades).

4.1.1 Keypoint Extraction

In general, SfM techniques look for the correspondences between images to estimate the camera extrinsic parameters and to generate the final sparse three-dimensional point cloud. We used SIFT [Lowe, 2004] to extract the keypoints and compute their descriptors due to its good invariance to scale and affine transformations that occur, as a consequence of cameras looking at the same region in many different viewpoints.

To avoid that too many keypoints are considered by our approach which is bad both due to ambiguity and unnecessary elevated processing time to match and optimize in the SfM phase, we sort the found keypoints by descending order of scale and remove the small keypoints so that we keep the features with large scale attribute up to 9.000 features per image, which is a sufficient amount of keypoints for the most scenarios, as suggested by [Wu, 2013]. The reason we select the features with large scale attribute in many steps in the approach is because they have a higher repeatability rate than small scale features and their descriptors tend to be more discriminant.

4.1.2 Vocabulary Tree Pruning

In some cases, the GPS tags are missing for some images, and it can become a problem when a dataset has most of its images without GPS information. Thus, we cannot remove the edges of the respective vertices that correspond to those images because we do not have any prior information to infer if the pairs overlap. Depending on the size of the dataset, it can cause a strong negative impact in the processing time of this phase.

To overcome this problem, we use a vocabulary tree approach similar to Nister and Stewenius [2006] to avoid the $O(n^2)$ time complexity in the matching step. Vocabulary trees are used in scalable image recognition, where similar images are returned by a recursive search in the tree given an image query (the search term). The algorithm we used to build a vocabulary tree can bee seen in Algorithm 1. Using the SIFT descriptors, we build a vocabulary tree with a branching factor (voc_tree_bf) of 9 (a reasonable value as shown in [Nister and Stewenius, 2006] experiments for large datasets) by grouping a feature set formed by 600 random keypoint descriptors obtained

in each image selected uniformly. We finally index the tree leafs using the top 3000 features (ordered from large to small scale value) of each image. Once the tree is indexed, we can query an image for images with close visual appearance to it, taking $O(\log(n))$ time complexity in a balanced tree, where n is the number of images in the entire dataset. These parameters were varied in our experiments, and we concluded that selecting 600 random keypoints produce a varied set of features and reduces the memory usage to build the tree, and querying for the 3000 largest features produces improved matching results rather than querying for the entire set, mostly because the largest features are more stable. The visual similarity score for each image is obtained by propagating again the 3000 features with largest scale attributes. The algorithm increments the bin's score of the respective indexes of the images that are present in the leaf of each descriptor propagation in the histogram of indexes for the image query. We have used the increment score as being:

$$score = \frac{1}{nl}, \tag{4.1}$$

where nl is the node level, so the most common features will contribute less to the score of the bins that are indexed in the leaf. Intuitively, the most common features will have larger clusters and the depth of the path for these less discriminant features will be larger.

In our experiments, we search for the top 60 highest score matches for each image (the most similar ones to that query according to the vocabulary tree), and we prune the edges in the epipolar graph from the query vertice to those vertices that are not among the highest scores of this query, excepting the edges that were validated by the GPS distance. The time complexity cost of the entire pruning operation is $O(n \log(n))$, being n the number of images. Here, we consider that the tree is balanced assuming that the keypoint descriptors we used to build the tree were randomly chosen, even though it may not be true if there are too many similar features in the images. This approach enforces the matching step to be linear in time since each image will have at most 60 candidates to perform the registration.

4.1.3 Geometric Validation

After the graph construction, we can efficiently match image pairs in a reduced space search, which initially had $O(n^2)$ and now has $O(n)$ pairs. Furthermore, the remaining image pairs present strong evidence that they will actually overlap. For each edge of the graph, the matching step procedure first attempts to match the descriptors of

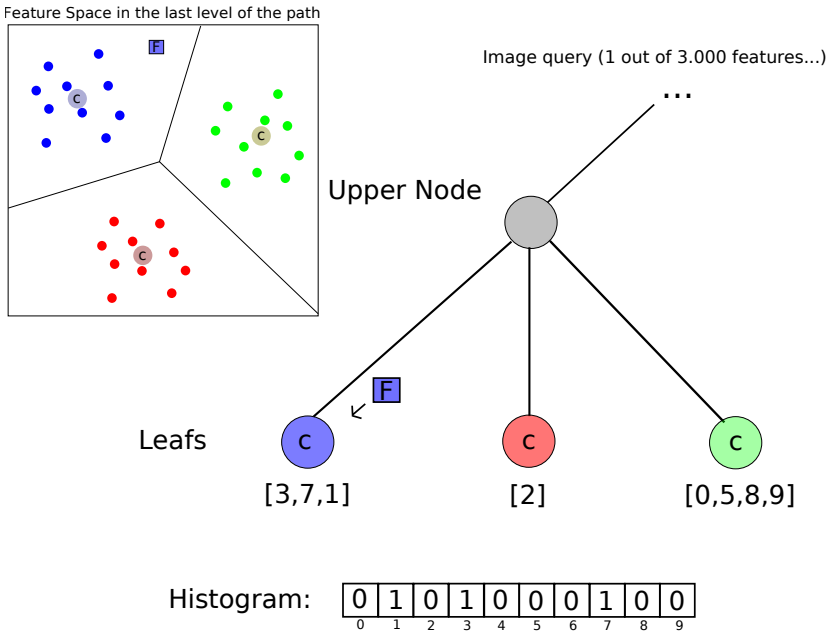


Figure 4.2. Example of a query for an image. The first feature is being propagated down. In this case, the branching factor of the tree is three, and in each level, the feature is compared to the three node centers and it is propagated to the one that is the closest to it. The process is recursively done until it reaches a leaf, where the histogram is incremented with the score (1 in this case for simplicity) for each respective index the leaf holds.

Algorithm 1 Vocabulary tree building.

procedure BUILDVOCABULARYTREE($Node$, $FeatureSet$, $BranchingFactor$)
  $\mathcal{C}_{set}$ = K-MEANS($FeatureSet$, $BranchingFactor$)
  for each cluster $\mathcal{C}$ and its respective child node **do**
    SETCHILDCENTER($ChildNode$, $Center(\mathcal{C})$)
    if $Size(\mathcal{C}) > 5 \times BranchingFactor$ **then**
      BUILDVOCABULARYTREE($ChildNode$, $\mathcal{C}$, $BranchingFactor$) ▷
Recursively divides the feature space into n Voronoi cells, where n is the branching factor.

two small sets containing the biggest (most discriminant) keypoints of their respective images, selected according to the scale attribute. If the correspondences are able to minimally satisfy the epipolar geometry constraint, a full pairwise match considering all keypoints are performed to obtain a fine pairwise registration.

Fast Geometry Verification: We perform a fast (coarse) verification of the epipolar geometry. The top k_top keypoints (in our tests, we use $k\_top = 600$) with the highest scale value are selected, which reduces the effort of matching. If the selected

pairs of keypoints do not provide a valid epipolar geometry, we remove the edge of the graph. We consider a pair as valid if the number of inlier correspondences returned by the fundamental matrix estimation 2.1 using RANSAC [Fischler and Bolles, 1981] is higher than at least 15% of the number of matches between each pair, which we call $coarse\_inlier\_rf$. The 15% value was chosen by performing tests on image pairs and we concluded when there is less than 15% of inliers in the correspondences using $k\_top = 600$, the likelihood of overlap between them is minimal. These steps are performed only between images that are connected in our graph. To avoid requiring the intrinsics for the images, we use the fundamental matrix in this step instead of the essential matrix, since for some images the intrinsics may not be available or have incorrect parameters.

Fine Pairwise Registration: To perform the fine registration, we fully match the keypoints between image pairs that have passed in the fast geometry validation. We now use all the keypoints found in both images with Fast Approximate Nearest Neighbour search (FLANN) [Muja and Lowe, 2009]. We also use the ratio test criterion, discarding similar distances of the two nearest neighbours of a query descriptor. We use a ratio of 0.8 in our experiments, which is the suggested value in the original SIFT paper [Lowe, 2004]. This step filters out ambiguous pair matches which have a higher chance to be wrong correspondences and decreases the set of points to be considered by the RANSAC. By doing that we also raise the probability of finding a valid pairwise geometric estimation (fundamental matrix), since the ratio of $inliers/total$ in the set of correspondences is increased. At last, we estimate the fundamental matrix by using the RANSAC scheme with the normalized 8-point algorithm [Hartley, 1997] to validate a pair geometry. Again, we use the fundamental matrix to avoid using wrong intrinsics.

A threshold in pixel is defined ($threshold\_fm = 0.07\%$ of the image width in our tests) to determine if the point is an inlier or not, depending on the distance that it is from the respective epipolar line in the other image, which is a similar threshold used by Bundler [Snavely et al., 2008a], and were tested many different datasets.

4.2 Filtering

We set the weights in the epipolar graph using the number of *inliers* returned by RANSAC for each estimated pair. A naïve approach would consider removing the edges with a small number of inliers using a hard threshold and perform the triangulation by using only the remaining pairs. However, this may remove edges that keep the graph connected, which results in missing parts in the final 3D model, specially because it is

difficult to define a hard threshold for this purpose, depending on many factors, *e.g.* matching quality, amount of texture in the images and overlap. Therefore, we propose applying a maximum spanning tree approach (MST) to remove only the edges with small number of inliers but enforcing the connectivity of the graph, since the MST avoids us breaking the epipolar graph into smaller connected components when we try to remove an edge with low number of inliers.

The last step of the epipolar filtering consists in extracting the sub-graph that contains the edges from the maximum spanning tree and the edges with the number of inliers larger than a defined threshold τ_i (we use a value of 60 inliers in our experiments, a standard value used by Bundler [Snavely et al., 2008a] and VisualSFM [Wu, 2013]).

This procedure is described by the Algorithm 2. The complexity of the Algorithm 2 lies in the same of Kruskal's algorithm $O(e \log(e))$ since only an additional $O(e)$ iteration is required.

4.3 Incremental SfM

Our methodology uses an incremental structure-from-motion approach. The algorithm begins the reconstruction by using a pair of images and then incrementally estimate the points and cameras parameters, adding them to the model sequentially. The camera motion estimation happens in a greedy manner with respect to the number of 2D-3D correspondences. In other words, the method estimates the camera motion through resectioning by choosing the camera that provides the largest amount of 2D-3D correspondences and then triangulates new 3D points into the model, until there is no more cameras to add.

Algorithm 2 Epipolar graph filtering.

procedure EpipolarFiltering(EG, τ_i)
  MaxSpanningTree($EG, FilteredEG$)
  for each edge e in EG **do**
    if $weight > \tau_i$ & $e \notin FilteredEG$ **then**
      Add($FilteredEG$, e)
  return $FilteredEG$
  ▷ The $FilteredEG$ contains the maximum spanning tree plus all edges higher than a threshold.

4.3.1 Robustly Choosing The Initial Pair

Choosing the initial pair is crucial to the quality of the reconstruction. If we choose a pair not having enough overlap, the reconstruction can fail immediately. But if we also choose a pair that have almost no translation motion (generally, they will overlap almost entirely), the essential matrix estimation and initial triangulated points will be ill-conditioned, because there is not enough parallax for the algorithm to infer the depth of the scene. To avoid that, we sort the edges of the graph and keep a percentile of 0.4 of the most valued edges (this value is arbitrary and is not sensitive when it is not set on the extremes like ≤ 0.10 or ≥ 0.90 according to our experiments), which contains consistent geometric pairs that undoubtedly overlap. Then, we sort this subset considering the ratio between the essential matrix inliers and the homography inliers and use a percentile of 0.25 (again, the percentile value is not sensitive and is arbitrary) of the subset containing the highest ratio between the fundamental matrix inliers and homography inliers ($F_{inliers}/H_{inliers}$), which is useful to avoid the use of small-baseline pairs in the seed reconstruction. Homographies cannot explain parallax in the scene, just the motion of planar surfaces. The number of homography inliers then will be, in general, lower than the inliers of the fundamental matrix for pairs with sufficient translation motion, except in the case when the entire scene is planar, which is fair to assume that is not in our context.

We then finally select the pair which provide the lowest mean re-projection error in this small subset of candidates. The essential matrix (2.2) is estimated using the normalized camera coordinates of the correspondences, calculated using the respective camera intrinsic parameters extracted from the EXIF meta-data or a calibration file. To perform the reconstruction of the initial pair, there must exist some source of information of the intrinsic parameters, or it will not be possible to approximately estimate the relative euclidean motion for them. An initial point cloud is created by triangulating the feature correspondences using the relative euclidean motion extracted from the essential matrix and refined using bundle adjustment.

4.3.2 Robust Incremental Estimation

From the initial point cloud, we find the image with the largest 2D correspondences with 3D points already estimated and we calculate the extrinsic parameters from the camera through camera resectioning. Camera resection techniques uses the 2D-3D correspondences to find a projection matrix $\mathbf{P_i}$ that maximizes the number of inliers of the projection of the corresponding 3D points onto the image i, generally using the direct linear transform (DLT) algorithm in a RANSAC scheme. We adopt the same

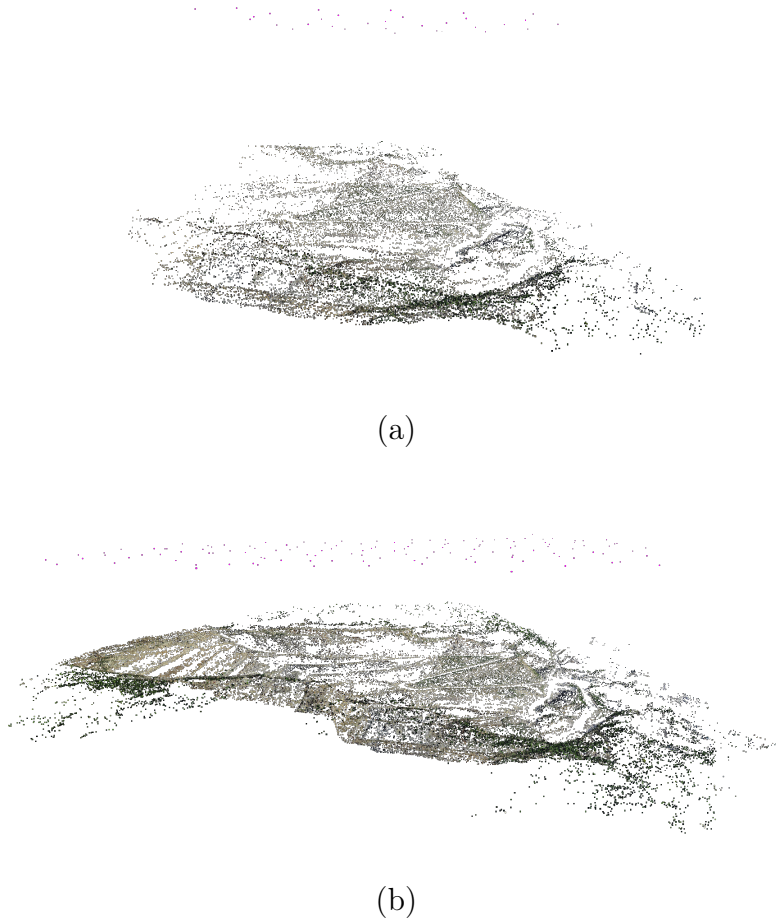
approach of Moulon et al. [2013] that uses an *a contrario* RANSAC scheme for solving for $\mathbf{P_i}$, where an inlier threshold is also estimated. The threshold choice for a pure RANSAC scheme for estimating $\mathbf{P_i}$ is usually done empirically and do not generalize well for different kind of datasets.

We first estimate a normalized $\mathbf{P_i}$ using normalized image coordinates, and then we evaluate the ratio of inliers and the threshold estimated. If the inlier ratio *inlier/total* is fewer than 0.25 or the threshold is above 24.0 pixels (MAX_RE), we assume that the intrinsic parameters of the camera are wrong. The values of these paramaters are similar to Bundler and works well for all datasets we tested. We then try to re-estimate the unnormalized $\mathbf{P_i}$ and decompose it into two matrices using the RQ decomposition. The two matrices are actually the rotation matrix (an orthogonal matrix) and the calibration matrix $\mathbf{K_i}$ of camera i. Finally, we are able to extract the new focal length from $\mathbf{K_i}$, and re-estimate the normalized $\mathbf{P_i}$. If the camera resectioning fails again, we exclude it from the model.

If a valid estimate of $\mathbf{P_i}$ is obtained, our algorithm optimizes the camera parameters including the focal length and camera distortion through a single-camera bundle

Algorithm 3 Incremental estimation.

procedure IncrementalSfM(*Structure*, *Cameras*, $\mathcal{EG}$)
  while NotAllChecked(*Cameras*) **do**
    $BC \leftarrow$ FindBestCamera(*Structure*, *Cameras*)
    ResectionNormalized(*Structure*, *BC*)
    if $BC.InlierRatio < 0.25 \vee BC.threshold\_resec > MAX\_RE$ **then**
      ResectionUnnormalized(*Structure*, *BC*)
      UpdateFocalLength(*BC*)
      ResectionNormalized(*Structure*, *BC*)
      if $BC.InlierRatio < 0.25 \vee BC.threshold\_resec > MAX\_RE$ **then**
        continue ▷ Skip this camera.
      else
        OneCameraBundleAdjustment(*BC*)
        TriangulatePoints(*Structure*, *BC*, *Cameras*, $\mathcal{EG}$)
    else
      OneCameraBundleAdjustment(*BC*)
      TriangulatePoints(*Structure*, *BC*, *Cameras*, $\mathcal{EG}$) ▷ Triangulate points with connected cameras in the graph that have already been estimated.
    if $NumberOfEstimatedCameras \mod window\_size = 0$ **then**
      LocalBundleAdjustment(*Structure*, *Cameras*)
  GlobalBundleAdjustment(*Structure*, *Cameras*) ▷ Performs a final global optimization considering all cameras and points in the end of the procedure.



(a)

(b)

Figure 4.3. Sparse reconstruction obtained from small_mine dataset (127 images) during the incremental estimation of the camera parameters and sparse 3D structure. (a) Partially reconstructed model (32 images); (b) Fully reconstructed model.

adjustment with fixed 3D points. Finally, we triangulate the points that are not in the model by visiting the connected estimated cameras using the epipolar graph. We discard the points with a triangulation angle smaller than 2.0 degrees (a standard value used in many stereo algorithms, such as [Furukawa and Ponce, 2010], [Snavely et al., 2008a], [Wu, 2013]) because the cameras do not have enough baseline to provide a good estimation of the 3D intersection.

The camera resectioning step is repeated iteratively for all cameras, and after a certain amount of camera estimations, we call a local bundle adjustment to minimize the re-projection error, consequently reducing drifting. Once there is no more cameras to be added, we run a final global optimization to obtain a set of optimized parameters for each camera, including motion, radial distortion and intrinsics, and also the refined sparse point cloud representing the *keypoints* found in the 3D space, which can be seen

in Figure 4.3. Algorithm 3 shows the incremental estimation procedure.

4.4 Local Bundle Adjustment and Global Refinement

Bundle adjustment (BA) techniques attempt to minimize the re-projection error between the observed and predicted image points in order to obtain the optimal 3D structure and camera parameters (Subsection 2.2.2).

Due to the large number of unknown parameters which contributes to the re-projection error value, a standard implementation of this optimization method would have massive computational costs when applied to the minimization problem characterized in bundle adjustment. Lourakis and Argyros [2009] proposed a method that explores the sparse block structure of the non-linear optimization problem in BA context achieving a considerable time performance gain (Equation 4.5).

However, finding the optimal solution for this problem is still time consuming when considering thousands of cameras and millions of 3D points. To tackle with this problem, we propose an overlapping local bundle adjustment window approach that optimizes the camera poses and points locally, but it overlaps with already optimized 3D points to hold the consistency and avoid fast propagation of drifting. Although this approach can be find in several video-based (*i.e.* small baseline and organized dataset) methods, in our work we apply this approach for unorganized dataset of large baselines.

Let $\mathbf{V} = (P_1, \ldots, P_m, X_1, \ldots, X_n)^T$ be a vector containing all parameters describing the m projection matrices and the n 3D points, and $\mathbf{X} = (x_{11}^T, \ldots, x_{1m}^T, \ldots, x_{n1}^T, \ldots, x_{nm}^T)^T$ the measured image point coordinates across the cameras (position of the detected keypoints). By using the parameter vector, we can create the estimated measure matrix as:

$$\hat{\mathbf{X}} = (\hat{x}_{11}^T, \ldots, \hat{x}_{1m}^T, \ldots, \hat{x}_{n1}^T, \ldots, \hat{x}_{nm}^T)^T, \tag{4.2}$$

where $\hat{x}_{ij}^T$ is the projection of the 3D point i in the camera j.

We can write the BA as the optimization problem of finding the values that minimize:

$$(\mathbf{X} - \hat{\mathbf{X}})^T \Sigma_{\mathbf{X}}^{-1} (\mathbf{X} - \hat{\mathbf{X}}) \tag{4.3}$$

over the parameter vector $\mathbf{V}$. $\Sigma_{\mathbf{X}}$ is the norm matrix. The minimization can be performed by the Levenberg-Marquardt algorithm Marquardt [1963] to solve the aug-

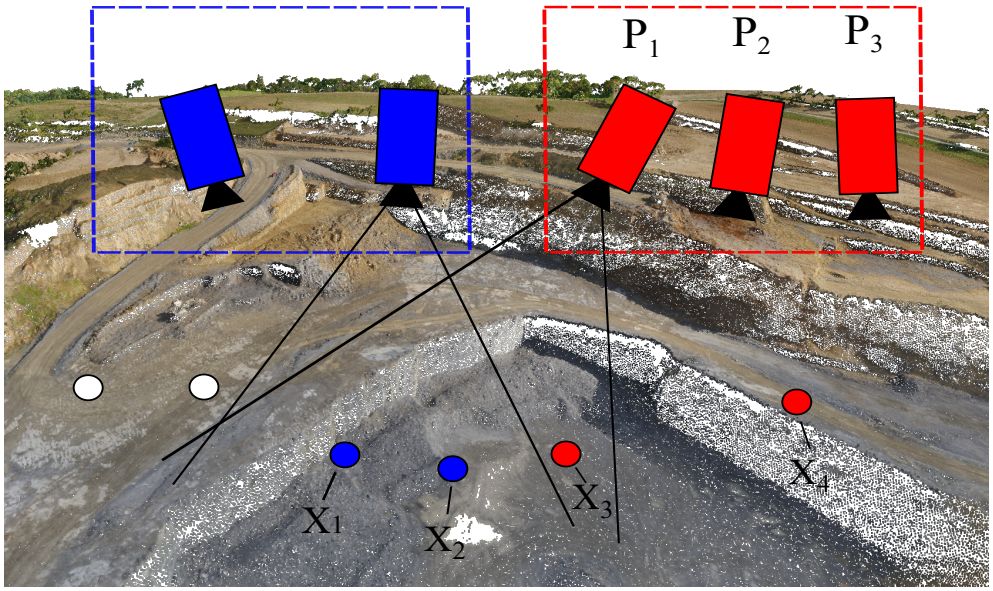


Figure 4.4. A simple case of the local window approach. The blue selection represents the points and cameras that have already been bundle adjusted, while the red selection will be optimized when the window becomes full. The green points will contribute to the minimized re-projection error of the cameras, but since they already are optimized, their parameters will remain fixed.

mented weighted normal equations:

$$(\mathbf{J}^T\Sigma_{\mathbf{X}}^{-1}\mathbf{J} + \mu\mathbf{J})\delta = \mathbf{J}^T\Sigma_{\mathbf{X}}^{-1}(\mathbf{X} - \hat{\mathbf{X}}), \tag{4.4}$$

where $\mathbf{J}$ represents the Jacobian of $\hat{\mathbf{X}}$, δ the update parameter of $\mathbf{V}$ that we are estimating and μ is the damping term which is used to change the diagonal elements of the Jacobian.

For instance, considering the camera setup and the scene structure illustrated in

Figure 4.4, the Jacobian **J** can be write as:

$$\mathbf{J} = \begin{pmatrix} \frac{\partial \hat{x}_{11}}{\partial P_1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\partial \hat{x}_{12}}{\partial P_2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\partial \hat{x}_{13}}{\partial P_3} & 0 & 0 & 0 & 0 \\ \frac{\partial \hat{x}_{21}}{\partial P_1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\partial \hat{x}_{22}}{\partial P_2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\partial \hat{x}_{23}}{\partial P_3} & 0 & 0 & 0 & 0 \\ \frac{\partial \hat{x}_{31}}{\partial P_1} & 0 & 0 & 0 & 0 & \frac{\partial \hat{x}_{31}}{\partial X_3} & 0 \\ 0 & \frac{\partial \hat{x}_{32}}{\partial P_2} & 0 & 0 & 0 & \frac{\partial \hat{x}_{32}}{\partial X_3} & 0 \\ 0 & 0 & \frac{\partial \hat{x}_{33}}{\partial P_3} & 0 & 0 & \frac{\partial \hat{x}_{33}}{\partial X_3} & 0 \\ \frac{\partial \hat{x}_{41}}{\partial P_1} & 0 & 0 & 0 & 0 & 0 & \frac{\partial \hat{x}_{41}}{\partial X_4} \\ 0 & \frac{\partial \hat{x}_{42}}{\partial P_2} & 0 & 0 & 0 & 0 & \frac{\partial \hat{x}_{42}}{\partial X_4} \\ 0 & 0 & \frac{\partial \hat{x}_{43}}{\partial P_3} & 0 & 0 & 0 & \frac{\partial \hat{x}_{43}}{\partial X_4} \end{pmatrix}. \tag{4.5}$$

In our implementation, we first used the Sparse Bundle Adjustment (SBA) library as the optimizer solver [Lourakis and Argyros, 2009], but then we verified that there is a newer, more efficient and flexible implementation of a non-linear least squares solver called Ceres [Agarwal et al., 2015] that also leverages the sparse structure of the Jacobian, which we later used to model and optimize the parameters of our SfM problem in a very practical way, and it was able to provide slightly better re-projection error results.

The incremental approach estimates camera motion and scene structure calling bundle adjustment multiple times. As the number of parameters of the model incrementally increases, the time to perform a global BA iteration rapidly grows with the number of cameras and points. Our approach proposes to fasten the parameters of the 3D points that have already been bundle adjusted and only adjusts the parameters of the newest estimated cameras and points.

The time complexity of bundle adjustment considering the sparse block structure is $O(m^3)$ [Mitra and Chellappa, 2008], where m is the number of cameras. In the the incremental approach, $O(m)$ global BA calls are required to avoid the propagation of drifting, which makes the complexity raise to $O(m^4)$. This asymptotic behavior causes

the approach to be very slow on large datasets. However, limiting the number of cameras BA will consider to a constant number, we can still obtain comparable results as shown in our experiments and able to reduce the $O(m^4)$ complexity back to $O(m^3)$. In this case, we will still require global BA calls to correct long term drifting but we can limit the number of calls to a constant value, while we reduce the fast propagation of drifting optimizing the parameters through local bundle adjustment. LBA has a time complexity of $mO(w^3)$, for a window containing w cameras. Since the number w is fixed to a constant value, we obtain an overall asymptotic behavior of $O(m) + O(m^3)$, where the $O(m)$ term is respective to LBA, and the $O(m^3)$ term the global BA part. The final time complety then is equivalent to $O(m^3)$.

The window in our case contains the most recent estimated cameras and all the 3D points that projects onto them. When the window achieves the limit of cameras, we call a BA that will optimize all cameras in the set and the points. It is important to notice that points that have been already optimized contributes to the minimized re-projection error, although their parameters remain fixed, to maintain the local consistency and prevent the fast propagation of drifting. Figure 4.4 shows two sets of cameras (blue and red). The blue set was optimized and the current iteration is trying to adjust the three new cameras (in red). The green points should not be modified in the optimization process. Thus, we set the values $\frac{\partial \hat{x}_{ij}}{\partial X_i} = 0, \forall i \leq 2$ in the Equation 4.5.

Global BA can be performed sometimes during reconstruction to obtain the optimal parameters as we do in our experiments, but much fewer global optimization calls are required (bound to a constant value), and it is optional depending on the size of the dataset and the desired accuracy.

4.5 Dense Reconstruction

Once we have the complete set of projection matrices and undistorted images estimated by our approach, we use them as input to a MvS dense reconstruction technique [Furukawa and Ponce, 2010].

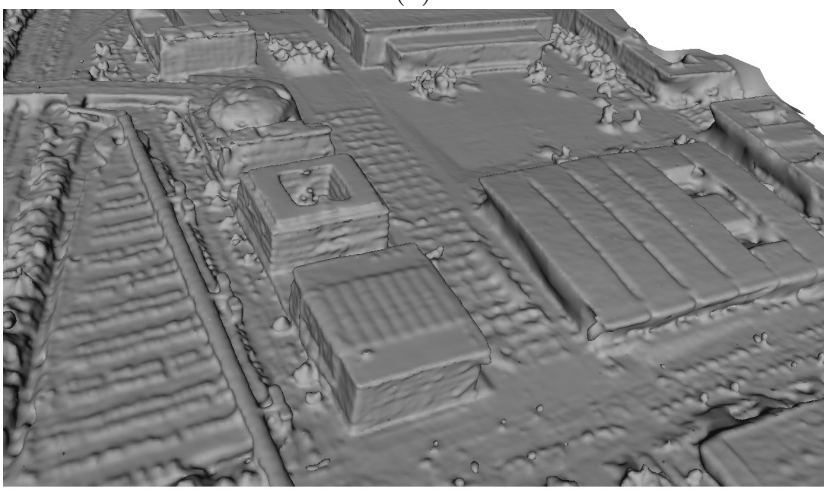
This algorithm uses a robust match, expand, and filter approach to estimate a *quasi-dense* set of small 3D rectangular patches and is composed of three main parts. First, an initial collection of small oriented rectangular patches is created by matching the features detected in input images. The algorithm initializes each patch with its center $c(p)$, normal $n(p)$, a reference image $R(p)$ and the list of visible images $V(p)$. The geometric parameters $c(p)$ and $n(p)$ are estimated by maximizing a photographic similarity cost function (Equation 2.5). Thereafter, it creates new patches according

to the neighbouring pixels of the initial matches. This performs the expansion around the vicinity of the patch. After the matching and expansion, a filtering process is performed in order to eliminate erroneous patches. The algorithm uses three filters – two of them based on visibility constraints and a third one that enforces a weak form of regularization.

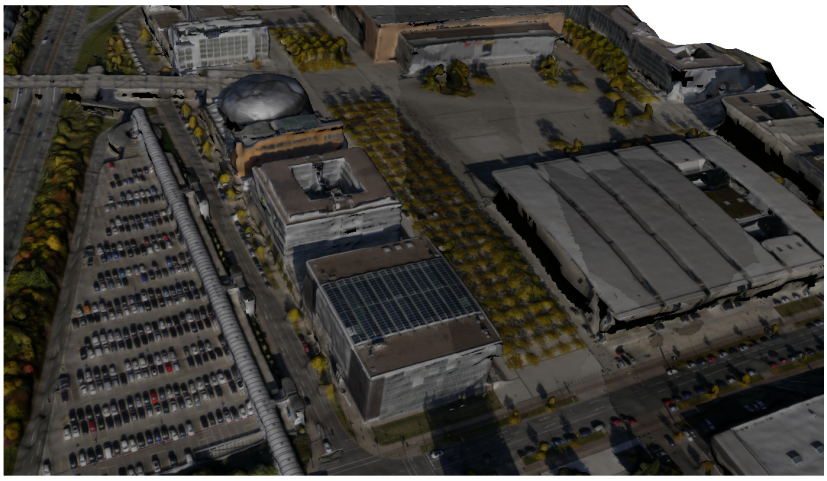
It is important to mention that the quality of the camera parameters provided by the SfM algorithm as well as the quality of the images (*e.g.* resolution, texture and image sharpness) strongly influence on the density and quality of the estimated *quasi-dense surfel* model.

By using the Poisson Surface Reconstruction method [Kazhdan et al., 2006], we convert the set of oriented points into a mesh model. The parameters used in Poisson reconstruction were 12 for the reconstruction depth and 10 for the subdivision depth. We chose those parameters empirically by varying them between 6 and 12 and analyzing the visual quality and computational resources needed. Higher values produce more detailed models but require a lot more memory and processing time. At last, we use the images and their respective projection matrices to obtain a parametrized and textured model by projecting the image rasters' into the model. Figure 4.5 shows the three models generated by the dense reconstruction phase for the expopark dataset.

(a)



(b)



(c)

Figure 4.5. The dense reconstruction obtained from exporpark dataset (1,231 images) after the estimation of the camera parameters and sparse 3D structure. (a) *Quasi-dense surfel* model estimated by the patch-based multi-view stereo algorithm [Furukawa and Ponce, 2010]; (b) Poisson surface reconstruction of a detailed region; (c) The projected textures into the mesh of the same region.

Chapter 5

Experimental Evaluation

In this chapter, we show the obtained results of our structure-from-motion approach in ten different datasets and compare them against three state-of-the-art implementations for solving moderate and large scale SfM problems, namely, Bundler [Snavely et al., 2008a], VisualSFM [Wu, 2013] and OpenMVG [Moulon et al., 2013].

5.1 Experimental Setup

5.1.1 Datasets

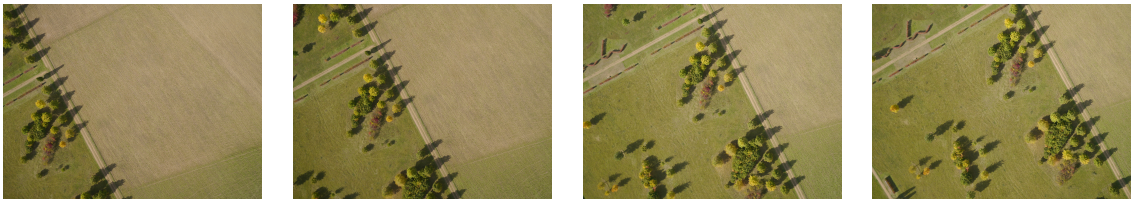
Both aerial and terrestrial datasets were used to evaluate our method, each one from a different scene. Challenging aspects are present in many of these datasets: Low-textured regions, reflective surfaces such as lakes, occlusions caused by moving objects and strong illumination and perspective changes.

Aerial Datasets: We used six large scale aerial datasets composed of high resolution overlapping images acquired by unmanned aerial vehicles with large baseline, obtained from publicly available drone websites.

- *small_mine* contains 127 images acquired from the main pit of a stone mine;
- *small_city* has 297 images from a village next to a lake in Switzerland;
- The *intergeo* presents plan terrain and low textured regions, which allow us to visually check consistencies of the generated model (479 images);
- *colombia_club* was gathered in a large region of a complex (795 images), containing a small river and a lake, which are poor in texture;

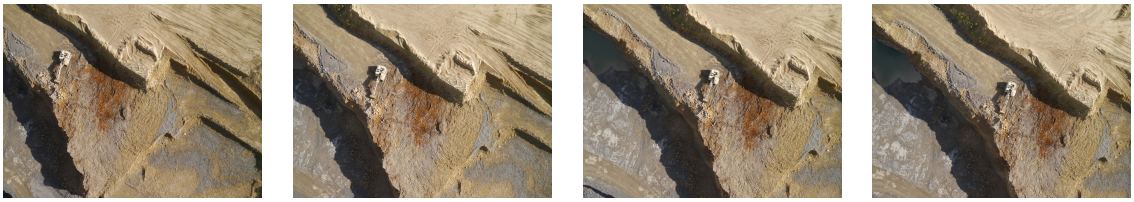
small_mine – 127 images – 1600 × 1200 pixels resolution.

small_city – 297 images – 1600 × 1200 pixels resolution.

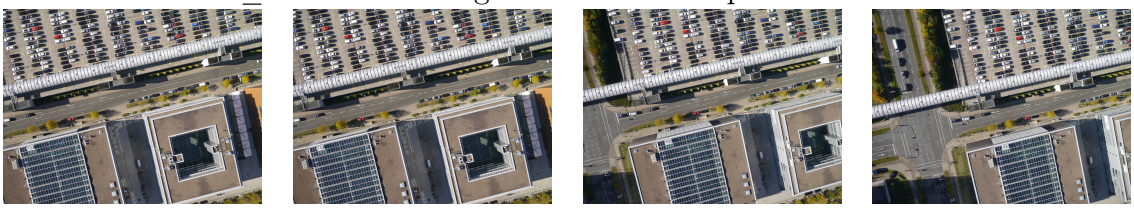


intergeo – 479 images – 1837 × 1380 pixels resolution.

colombia_club – 795 images – 1837 × 1380 pixels resolution.



sand_mine – 978 images – 1837 × 1380 pixels resolution.



expopark – 1,231 images – 1837 × 1380 pixels resolution.

Figure 5.1. Four image samples for each aerial dataset, followed by the amount of images and resolution.

- The *sand_mine* dataset was obtained in a stone and sand mining region, incorporating multiple open pits, mountains and plain regions (978 images);

- The largest dataset, named *expopark* , exhibits composite details of hangars and tall buildings from an exposition park and its surroundings (1, 231 images). It also contains small objects as cars and small trees.

Terrestrial Datasets: We also used four terrestrial datasets. NotreDame was obtained from the internet (Flickr), while the other three were made using a *Samsung S4* smartphone from VeRLab.

- *Notredame* is an unordered collection set of 715 images taken from the flickr website. The challenge of this dataset is to tackle with unknown focal lengths and distortion coefficients from the heterogeneous camera models, in addition to the GPS missing from most of images. Furthermore, strong illumination changes, occlusions and extreme viewpoints are present in this dataset.
- The *UFMG_statue* dataset was obtained by taking 23 pictures around a replica of Venus de Milo statue in front of the "Belas Artes" building. The statue has smooth texture, and fine prominences that can be used to evaluate the depth quality of the mesh.
- *UFMG_Rectory* dataset was also obtained by taking 104 pictures of the rectory building in different viewpoints. The building has reflective glasses in the window that can deteriorate the results.
- *ICEx_square* dataset contains a set of 125 images taken inside ICEx near the main entrance. The low textured and ambiguous walls of the building is challenging to all photo-based reconstruction techniques.

5.1.2 Hardware

We used a virtual machine hosted by a computer equipped with two Intel(R) Xeon(R) CPU E5-2620 @ 2.00GHz processors and 132 GB of RAM , but for a fair comparison we have set each method used in our experiments to work in a single thread and computed the time each approach used in CPU. Our focus was to test the scalability of the SfM approaches *per se* rather than testing how well they are implemented for the use in parallel systems, such as VisualSFM that has multiple GPU and multi-thread support.

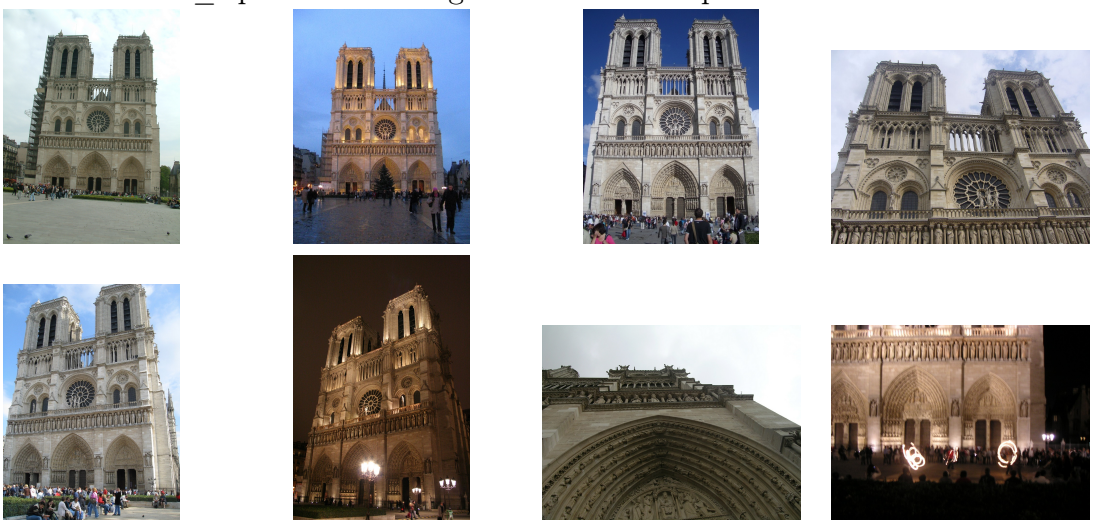
5.1.3 Evaluation Methodology

To quantitatively evaluate the output of all the SfM techniques in our experiments, we use the residual re-projection error values in pixels. For a given reconstruction, we

statue – 23 images – 2048 × 1152 pixels resolution.

rectory – 104 images – 2048 × 1152 pixels resolution.

ICEx_square – 125 images – 2048 × 1552 pixels resolution.



NotreDame – 715 images – various pixels resolution.

Figure 5.2. Four image samples for each of the terrestrial datasets, followed by the number of images and resolution.

calculate the mean and median of the vector of residuals. We used the median when comparing different pipelines because the threshold value of the outlier elimination for the four approaches vary, and the median gives a more fair value for comparison. Each element of the vector has the residual error obtained by projecting the estimated 3D point onto the image using the cameras' estimated poses and calculating its distance from the measurement. The residual vector contains all the residuals of the projections

of all the estimated 3D points onto the cameras they are visible in.

To calculate the residual value of the projection of one 3D point onto a given camera it is visible in, we first need to convert the 3D point into the camera's coordinate system:

$$\mathbf{C} = \mathbf{R_i X} + \mathbf{t_i}, \tag{5.1}$$

where $\mathbf{X}$ is the 3D point, $\mathbf{R_i}$ is the camera's rotation matrix, $\mathbf{t_i}$ is the camera translation vector, and $\mathbf{C}$ is the 3D point in the camera's coordinate system, considering the camera i. Then, we divide the point by the z coordinate (perspective division) to project the 3D point onto the camera's plane represented by the 2D vector $\mathbf{p} = \dfrac{\mathbf{C}}{\mathbf{C_z}}$. Finally, to be able to compute the residual error of the estimated projection of the point, we calculate the position of the projection in pixels $\mathbf{p_{pixel}}$, which is given by the formula:

$$\mathbf{p_{pixel}} = r(\mathbf{p}, \mathcal{D}_i)\mathbf{K_i}, \tag{5.2}$$

where $r(\mathbf{p}, \mathcal{D}_i)$ is the radial distortion function (Brown-Conrady distortion model) that distorts the point $\mathbf{p}$ according to the distortion coefficients vector $\mathcal{D}_i$, and $\mathbf{K_i}$ is the calibration matrix respective to the camera i.

Now, we can obtain the residual error value by finding the Euclidean distance between the predicted position of the projection and the measured position of the projection:

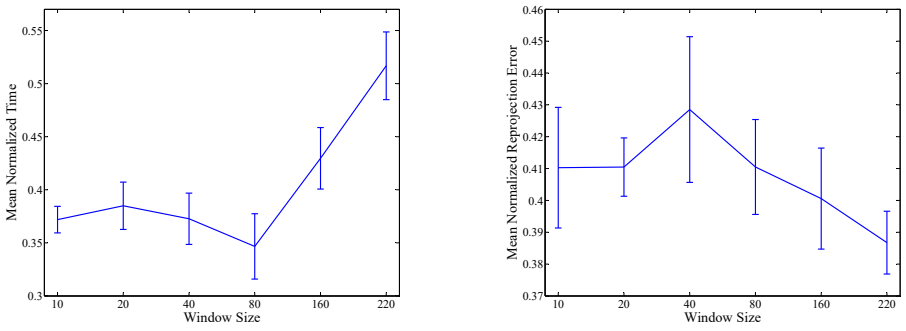
$$residual = \| \mathbf{p_{pixel}} - \mathcal{M} \|^2, \tag{5.3}$$

where $\mathcal{M}$ is the actual measured 2D position (found keypoint) of the projected 3D point on image i. Due to the lack of ground-truth data, which is very common considering large scale datasets and images in the wild, we were only able to evaluate the estimation quality achieved by measuring the re-projection error values.

5.2 Parameter Tuning

To choose the window size in the local bundle adjustment step, we ran several experiments with multiple window size values on 3 moderate-sized datasets. We chose the size equal to 80, since it provided the best time performance gains with small fluctuations in the re-projection error (Figure 5.3).

We tested several combinations of detectors and descriptors (*e.g.* ORB [Rublee et al., 2011], SURF [Bay et al., 2008] and SIFT [Lowe, 2004]) and we found that SIFT holds the best results, providing robust and accurate correspondences. However, in textureless regions of the images, depending on the thresholds used, the detector may



(a) Processing time (b) Re-projection error

Figure 5.3. Mean normalized performance achieved by varying the window size (a) and the mean normalized re-projection error of each window size (b) for the large-scale aerial datasets, with their respective standard deviation interval. We have empirically chosen the best parameters according to the graph results.

not return any keypoints in a large area of an image or no features at all, which may lead to a bad pose estimation and loss of overlap that may be crucial to the reconstruction.

We found that lowering the default contrast threshold value of the OpenCV's [Bradski, 2000] SIFT implementation from 0.04 to 0.02 still yields good keypoints with sufficient discriminant descriptors and provide a higher amount of keypoints in low textured regions of the image.

5.3 Results and Discussion

The Figure 5.5 shows the median re-projection error for each dataset and each method. Figure 5.4 shows the time performance. We set a time-out of 120 hours for the single core experiment. Bundler and VisualSFM were unable to generate the results for some datasets in the established time-out. The VisualSFM re-projection error was computed by using the the VisualSFM method with all parallel optimization options enabled including GPU. The darker green curve of the processing time in Figure 5.4 shows the values for a execution in a machine with a Xeon E3-1200 v2/3rd Gen 8-core processor and a GeForce GTX 560 Ti GPU.

In Figure 5.4, we can clearly see an expressive increase in processing time by all implementations but ours, as the number of images increases. Our method shows a smoothed growth and it leads the performance, reflecting the optimization steps adopted in our method. Also in Figure 5.5, we can see that our approach provides the second best values of achieved re-projection error, slightly smaller (less than 0.09 pixels on average) than OpenMVG. This is the result of a careful selection of pairs

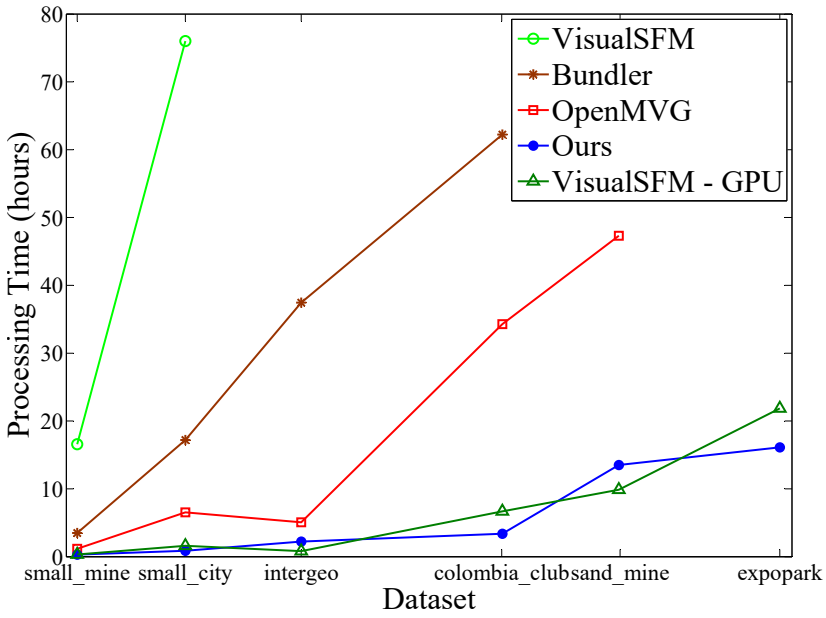


Figure 5.4. Time performance considering the entire pipeline. Our approach was the only able to provide the results for the expopark dataset within the time-out value of 120 hours.

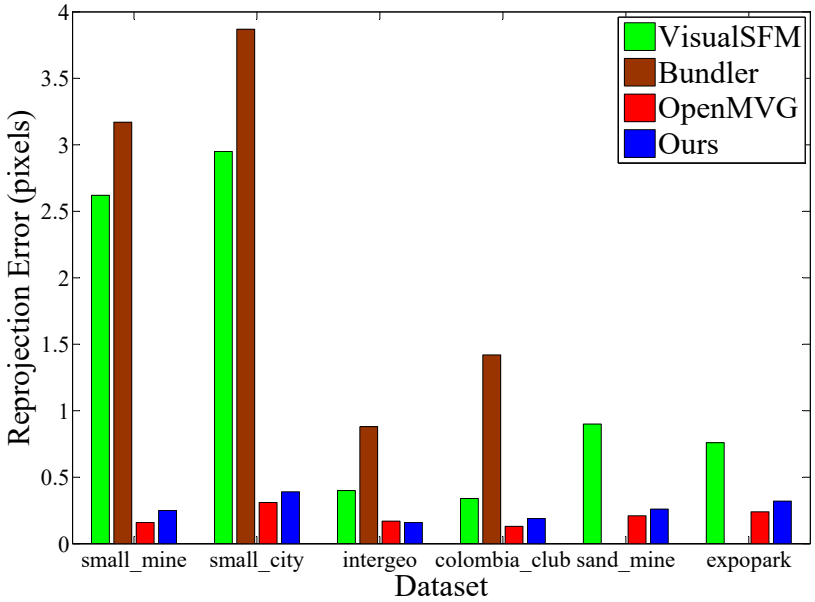


Figure 5.5. Median re-projection error results of each approach for the large aerial datasets. VisualSFM-GPU re-projection error is equivalent to VisualSFM without GPU.

to be matched through the filtering performed by Algorithm 2, which avoids false positive matches that can lead to an increase of the re-projection error of the cloud

compromising the model accuracy, besides assuring the completeness in the model estimation. As can be seen in Figures 5.6, 5.10, 4.5 and 5.11, the final model for all sets of aerial images do not present any discontinuity on the mesh, except in reflective surfaces as lakes and water.

We used a collection of 715 unorganized images from the Notre Dame dataset [Snavely et al., 2008a] to show the capability of our approach to deal with unordered collection of images in the wild. The dense 3D model generated by our method is shown in Figure 5.9. For this experiment, our method estimated the model with a re-projection error of 0.43 pixels in 27.4 hours. Bundler method, for its turn, spent 86.7 hours and got a larger error (0.47 pixels). VisualSFM was not able to provide the results within the established time-out value of 96 hours, and OpenMVG could not handle with the missing focal lengths of a good portion of the images, not returning any results.

For the rest of the terrestrial datasets we compared them with the most accurate approach, OpenMVG, and we were able beat its results in some cases. Here, we use the root mean squared error value (RMSE) since OpenMVG also uses an *a contrario* estimation for outlier removal, and the objective function for the bundle adjustment is to minimize the RMSE. For the UFMG_statue dataset, our method estimated the parameters with a RMSE re-projection error of 0.24 pixels, while OpenMVG's RMSE was 0.25 pixels. In the UFMG_Rectory dataset our method's residuals were 0.36 pixels, and OpenMVG's 0.31 pixels, however, OpenMVG skips some images, and our method estimates the poses of all cameras. Finally, in the ICEx_square dataset, one of the most challenging ones due to the ambiguous and textureless walls, OpenMVG was able to estimate the pose of only 53 cameras, with a RMSE of 0.43 pixels, while our method was able to estimate all the 125 poses with a RMSE of 0.64 pixels, however the RMSE for this last case cannot be compared because there were much more parameters to be considered in estimating all the cameras. Figure 5.7 shows that the dense model generated using our estimated poses for ICEx_square dataset is visually correct from the top view, since it is known that the building is symmetric and regular.

The speedup provided by using the local bundle adjustment method proposed can be verified in Table 5.1. We compare the total time used to generate the DEM with the local bundle adjustment against the classic approach of globally optimizing the model multiple times. After the reconstruction using local BA finishes, we run a final global BA to obtain the optimal solution. We can see that even running a global optimization in the end, the speedup gain is considerable, and it is able to achieve global minima. It means that the multiple local BAs are able to maintain the necessary consistency and avoid the final minimization to fail.

	images	speedup	local error	global error
small_mine	127	2.33	0.60	0.45
small_city	297	2.51	0.56	0.53
intergeo	479	2.45	0.38	0.41
colombia_club	795	1.83	0.39	0.39
sand_mine	978	1.28	0.37	0.37
expopark	1,231	2.04	0.38	0.23

Table 5.1. Speedup gain and mean re-projection error in pixels of the local approach compared to global BA. A small oscillation can be noticed which is caused by RANSAC model estimations, but the overall accuracy remains the same, while there is a significant speedup advantage.

	colombia_club	sand_mine	expopark
Using Alg. 1	0.39	0.37	0.38
Without Alg. 1	1.01	0.73	5.37

Table 5.2. Mean re-projection error in pixels when using the filtering based on the maximum spanning tree.

We also evaluate the benefits of using the Algorithm 2 and the GPS information to filter the epipolar graph. We performed three experiments with the largest datasets. Table 5.2 shows there is a considerable gain in accuracy using Algorithm 2 in our implementation. For the GPS data, we disabled the filtering step, and our method was able to perform the reconstruction faster than all other ones (considering the final global refinement), thanks to the efficient strategies adopted in incremental reconstruction.

All the mentioned characteristics of the aerial datasets described in Subsection 5.1.1 provide us rich information about how good are the estimated results both quantitatively and qualitatively. A qualitative result from all these datasets is shown in Figure 4.5, Figure 5.6, Figure 5.7, Figure 5.8 and Figure 5.12.

It can be verified that all of the models estimated by our approach are consistent and dense, except in extremely low textured regions in the images and strongly perspective distorted regions such as the ground. However, even in low textured regions as shown in the *surfel* model of intergeo dataset, there is no apparent holes in the model. Our SfM algorithm is also able to handle the reflective surfaces present in the colombia_club and small_city datasets, which may cause the dense reconstruction to fail due to bad camera pose estimation. It is even possible to visually perceive the heigh of the cars in the largest dataset, which were consistently estimated in the mesh depicted in Figure 4.5–(b).

5.3.1 Limitations

Our approach is inspired by the classical SfM pipeline proposed by [Snavely et al., 2008a], and besides the GPS pruning, we do not treat geometric ambiguity in the scene, thus, in some scenarios the reconstruction can fail for this reason. Some works in SfM aim at solving this specific problem [Wilson and Snavely, 2013]. In aerial images, such ambiguities are rare, but in images from cities taken from the ground, they are much more common due to the symmetric nature of men made buildings.

Another issue is that if a significant drift occurs before the global optimization, which can eventually happen for some datasets, mainly because the lack of tracks on images, the reconstruction can also fail.

In our experiments, we learned that images from the ground are more challenging due to strong perspective changes, lack of features and ambiguous keypoints present in the images, specially from textureless regions and walls.

SfM pipelines are a very promising tool that offers 3D information estimation at a low financial cost and automatic way. Nonetheless, in some situations laser-based techniques are still superior in accuracy, density and completeness of the reconstructed model, specially on weak textured environments.

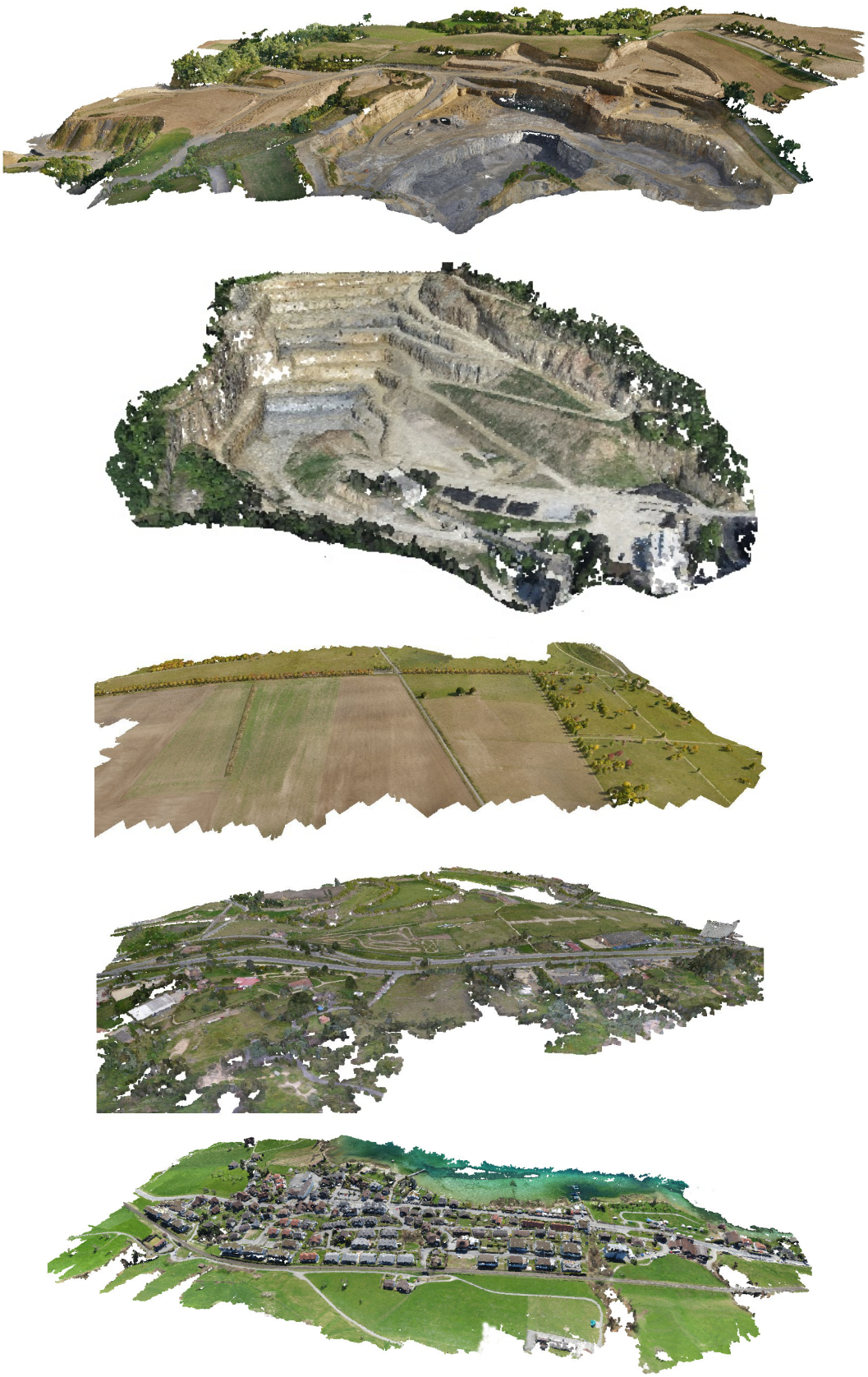
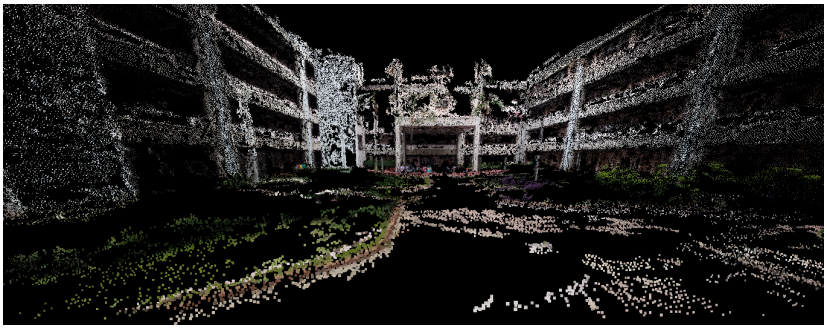


Figure 5.6. Dense *surfel* models estimated for the datasets. From up to down: sand_mine, small_mine, intergeo, colombia_club and small_city.



(a)

(b)

Figure 5.7. The *surfel* model for the ICEx_square dataset. (a) Inside view of the model; (b) Top view of the model.

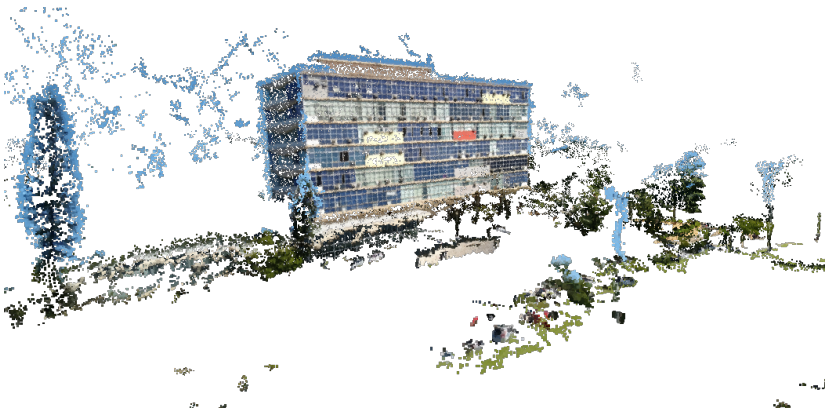


Figure 5.8. Quasi-dense *surfel* model obtained from the UFMG_Rectory dataset.

Figure 5.9. Result of the “Notre Dame" dataset experiment. Our methodology estimate this model with a re-projection error of 0.43 spending 27.4 hours, while Bundler returned an error equal to 0.47 and 86.7 hours of processing.

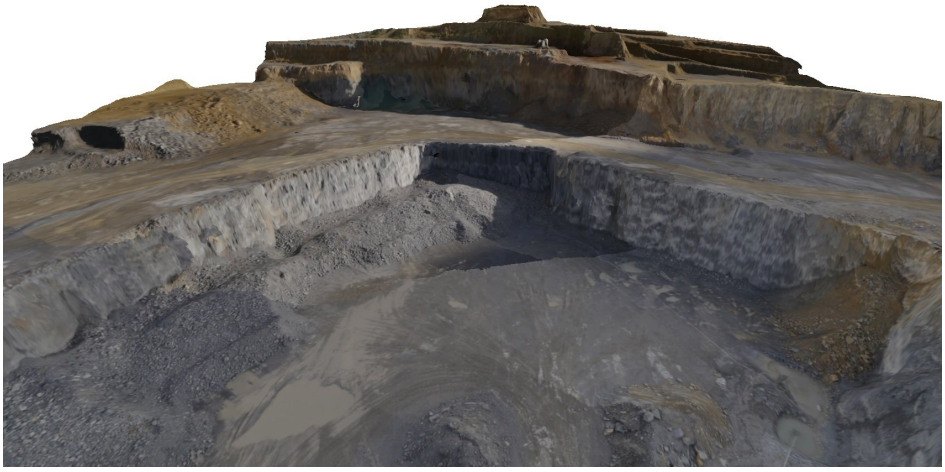


Figure 5.10. A detailed region of the final textured mesh from the second largest dataset small_mine.

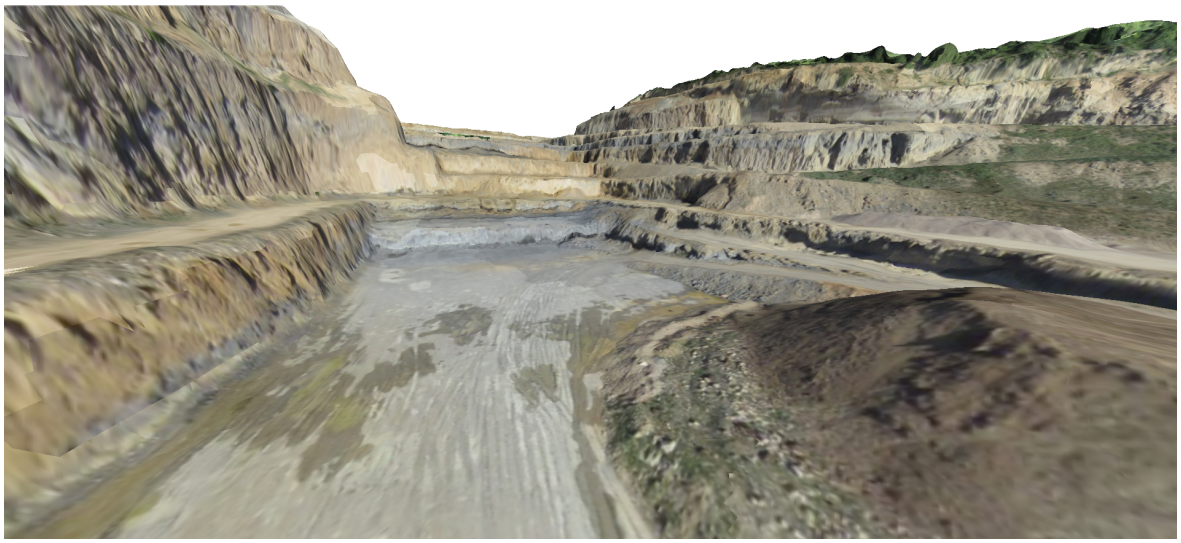


Figure 5.11. Ground-level view of the final mesh obtained from the small_mine dataset.

Figure 5.12. Four views of the final mesh for the UFMG_statue dataset. Fine geometry details can be seen.

Chapter 6

Conclusion

In this work, we proposed a novel methodology that incorporates efficient strategies in the incremental pipeline, which contributed to the time performance and scalability of the incremental structure from motion applied to DEM estimation.

Our contribution is the proposal and implementation of a new SfM pipeline adapted to high resolution aerial image (but not only limited to this kind) datasets which incorporates many previously used methods in the literature aiming at time efficiency. It is important to mention that most of these methods were used separately in previous works and we explore and adapt them into a single approach, in addition to the maximum spanning tree that ensures the graph's completeness and also contributes to a lower re-projection error of the estimation. The speed-up achieved as well as the low re-projection error can be seen in the experiments performed to evaluate the time efficiency and point cloud quality.

We performed experiments on the task of creating DEMs and Sparse 3D Model from sets of images. The proposed approach outperforms state-of-the-art SfM methodologies in terms of processing time, including VisualSFM algorithm with all GPU and multi-core optimization enabled for the largest dataset in a machine with a reasonable hardware configuration. As shown in “Notre Dame" experiment, our method can also handle unorganized collection of images in a reduced time and smaller re-projection error, even though our approach is specialized for aerial datasets.

It is worth noting that our approach is easily parallelizable in many steps and can be merged with other approaches, contributing to the development of 3D reconstruction techniques based on SfM.

6.1 Future Works

One of the possible improvements for future work is to enhance our implementation to leverage the parallelizable parts of the incremental SfM in a parallell architecture, such as multi-core processors and GPUs.

Aerial images are a good option to reconstruct large areas, specially natural environments, but in case of city areas, the facades of the buildings are usually not reconstructed due to the Z axis of the camera that faces orthogonally the ground. In the other hand, ground images provide fine details of facades and objects' sides, but the ground and top of other objects do not appear in such photos. An interesting work is to to use both aerial and ground images to obtain a more complete and detailed reconstruction of the scene. Merging these strongly different views of the same scene can also improve accuracy due to more constraints that will be imposed to the geometry of the reconstruction. However, keypoint matching techniques are usually not able to provide correspondences from such different points of view. They are robust to affine transformations up to a limit, and other approaches may be required to register aerial and ground images together.

Other interesting improvements can be done practically in all modules of SfM, and easily substituted, not only in our approach, but with respect to the state-of-the-art in Structure-from-Motion. The own nature of the SfM pipelines permits one to modify and improve isolated parts of the algorithm and replace them without any further modification in the other modules. Some of the things that can be improved in the incremental pipeline include:

1. Feature matching module, which will improve the overall accuracy of the algorithm, due to more accurate and correct matches across image pairs. Improvements in time performance and memory consumption in this phase are also important.

2. Bundle adjustment, one of the critical parts of the algorithm that is still a bottleneck and consumes a great portion of the total time to process a dataset.

There is also an important limitation of the SfM approaches that we must mention: They do not handle well textureless regions and ambiguity in the scene. Improvements can be done to treat or solve this kind of problem, and are being approached throughout the literature recently, but it is out of the scope in this work.

Bibliography

Agarwal, S., Mierle, K., and Others (2015). Ceres solver. http://ceres-solver.org.

Agarwal, S., Snavely, N., Simon, I., Seitz, S. M., and Szeliski, R. (2009). Building rome in a day. In *IEEE Int. Conf. on Comp. Vision*, pages 72--79.

Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 110(3):346--359. ISSN 1077-3142.

Belongie, S., Malik, J., and Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Analysis Machine Intelligence*, 24(4):509--522.

Bosse, M., Zlot, R., and Flick, P. (2012). Zebedee: Design of a spring-mounted 3-d range sensor with application to mobile mapping. *Robotics, IEEE Transactions on*, 28(5):1104--1119.

Bradski, G. (2000). *Dr. Dobb's Journal of Software Tools*.

Collins, R. T. (1996). A space-sweep approach to true multi-image matching. In *IEEE Conf. on Comp. Vision and Pattern Recog.*, pages 358--363.

Crandall, D., Owens, A., Snavely, N., and Huttenlocher, D. (2011). Discrete-continuous optimization for large-scale structure from motion. In *IEEE Conf. on Comp. Vision and Pattern Recog.*, pages 3001–3008. ISSN 1063-6919.

Engel, J., Schöps, T., and Cremers, D. (2014). LSD-SLAM: Large-scale direct monocular SLAM. In *Proc. of the Europ. Conf. on Comp. Vision*.

Eudes, A. and Lhuillier, M. (2009). Error propagations for local bundle adjustment. In *IEEE Conf. on Comp. Vision and Pattern Recog.*, pages 2411--2418. IEEE.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381--395. ISSN 0001-0782.

Frahm, J.-M., Fite-Georgel, P., Gallup, D., Johnson, T., Raguram, R., Wu, C., Jen, Y.-H., Dunn, E., Clipp, B., Lazebnik, S., and Pollefeys, M. (2010). Building rome on a cloudless day. In *Proc. of the Europ. Conf. on Comp. Vision*, ECCV'10, pages 368--381, Berlin, Heidelberg. Springer-Verlag.

Furukawa, Y. and Ponce, J. (2010). Accurate, Dense, and Robust Multi-View Stereopsis. *IEEE Trans. Pattern Analysis Machine Intelligence*, 32(8):1362--1376.

Gehler, P. and Nowozin, S. (2009). On feature combination for multiclass object classification. In *IEEE Int. Conf. on Comp. Vision*, pages 221--228. IEEE.

Goesele, M., Snavely, N., Curless, B., Hoppe, H., and Seitz, S. M. (2007). Multi-view stereo for community photo collections. In *IEEE Int. Conf. on Comp. Vision*, pages 1--8. IEEE.

Hartley, R. I. (1997). In defense of the eight-point algorithm. *IEEE Trans. Pattern Analysis Machine Intelligence*, 19(6):580--593. ISSN 0162-8828.

Hartley, R. I. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision.* Cambridge University Press, ISBN: 0521540518, second edition.

Henry, P., Krainin, M., Herbst, E., Ren, X., and Fox, D. (2010). Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments. In *In the 12th International Symposium on Experimental Robotics.*

Henry, P., Krainin, M., Herbst, E., Ren, X., and Fox, D. (2012). Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments. *The International Journal of Robotics Research*, 31(5):647--663.

Hermans, A., Floros, G., and Leibe, B. (2014). Dense 3d semantic mapping of indoor scenes from rgb-d images. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2631--2638. IEEE.

Irschara, A., Hoppe, C., Bischof, H., and Kluckner, S. (2011). Efficient structure from motion with weak position and orientation priors. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*, pages 21–28. ISSN 2160-7508.

Irschara, A., Rumpler, M., Meixner, P., Pock, T., and Bischof, H. (2012). Efficient and Globally Optimal Multi View Dense Matching for Aerial Images. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, pages 227–232.

James, M. R. and Robson, S. (2012). Straightforward reconstruction of 3D surfaces and topography with a camera: Accuracy and geoscience application. *Journal of Geophysical Research: Earth Surface*, 117(F3).

Jeong, Y., Nistér, D., Steedly, D., Szeliski, R., and Kweon, I. (2012). Pushing the envelope of modern methods for bundle adjustment. *IEEE Trans. Pattern Analysis Machine Intelligence*, 34(8):1605--1617.

Kazhdan, M., Bolitho, M., and Hoppe, H. (2006). Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7.

Laurentini, A. (1997). How many 2d silhouettes does it take to reconstruct a 3d object? *Computer Vision and Image Understanding*, 67(1):81–87.

Leberl, F., Irschara, A., Pock, T., Meixner, P., Gruber, M., Scholz, S., and Wiechert, A. (2010). Point clouds: Lidar versus 3d vision.

Li, L.-J., Socher, R., and Fei-Fei, L. (2009). Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. In *IEEE Conf. on Comp. Vision and Pattern Recog.*, pages 2036--2043. IEEE.

Liu, X. (2008). Airborne lidar for dem generation: some critical issues. *Progress in Physical Geography*, 32(1):31--49.

Lourakis, M. A. and Argyros, A. (2009). SBA: A Software Package for Generic Sparse Bundle Adjustment. *ACM Trans. Math. Software*, 36(1):1–30.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91--110.

Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial & Applied Mathematics*, 11(2):431--441.

Micheletti, N., Chandler, J. H., and Lane, S. N. (2015). Investigating the geomorphological potential of freely available and accessible structure-from-motion photogrammetry using a smartphone. *Earth Surface Processes and Landforms*, 40(4):473--486.

Mitra, K. and Chellappa, R. (2008). A scalable projective bundle adjustment algorithm using the l infinity norm. In *Computer Vision, Graphics & Image Processing, 2008. ICVGIP'08. Sixth Indian Conference on*, pages 79--86. IEEE.

Moulon, P., Monasse, P., and Marlet, R. (2013). Adaptive structure from motion with a contrario model estimation. In *Proceedings of the 11th Asian Conference on Computer Vision - Volume Part IV*, ACCV'12, pages 257--270, Berlin, Heidelberg. Springer-Verlag.

Muja, M. and Lowe, D. G. (2009). Fast approximate nearest neighbors with automatic algorithm configuration. In *In VISAPP International Conference on Computer Vision Theory and Applications*, pages 331--340.

Ni, K., Steedly, D., and Dellaert, F. (2007). Out-of-core bundle adjustment for large-scale 3D reconstruction. In *IEEE Int. Conf. on Comp. Vision*, Rio de Janeiro.

Niem, W. and Buschmann, R. (1994). Automatic modelling of 3d natural objects from multiple views. In *In European Workshop on Combined Real and Synthetic Image Processing for Broadcast and Video Production*. Springer.

Nistér, D. (2004). An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Analysis Machine Intelligence*, 26(6):756--770.

Nister, D. and Stewenius, H. (2006). Scalable recognition with a vocabulary tree. In *IEEE Conf. on Comp. Vision and Pattern Recog.*, volume 2, pages 2161--2168. IEEE.

Pollefeys, M., Nistér, D., Frahm, J.-M., Akbarzadeh, A., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Kim, S.-J., Merrell, P., et al. (2008). Detailed real-time urban 3d reconstruction from video. *International Journal of Computer Vision*, 78(2-3):143--167.

Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). ORB: An Efficient Alternative to SIFT or SURF. In *IEEE Int. Conf. on Comp. Vision*, Barcelona.

Shanmukh, K. and Pujari, A. K. (1991). Volume intersection with optimal set of directions. *Pattern Recognition Letters*, 12(3):165 – 170. ISSN 0167-8655.

Snavely, N., Seitz, S. M., and Szeliski, R. (2008a). Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(2):189--210. ISSN 0920-5691.

Snavely, N., Seitz, S. M., and Szeliski, R. (2008b). Skeletal graphs for efficient structure from motion. In *Proc. Computer Vision and Pattern Recognition*.

Strecha, C., Pylvänäinen, T., and Fua, P. (2010). Dynamic and scalable large scale image reconstruction. In *IEEE Conf. on Comp. Vision and Pattern Recog.*, pages 406--413.

Szeliski, R. (1993). Rapid octree construction from image sequences. *Computer Vision and Image Understanding*, 58(1):23--32. ISSN 1049-9660.

Teza, G., Pesci, A., and Ninfo, A. (2016). Morphological analysis for architectural applications: Comparison between laser scanning and structure-from-motion photogrammetry. *Journal of Surveying Engineering*, page 04016004.

Tomasi, C. and Kanade, T. (1992). Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, 9(2):137--154. ISSN 0920-5691.

Westoby, M. J., Brasington, J., Glasser, N. F., Hambrey, M. J., and Reynolds, J. M. (2012). 'Structure-from-Motion' photogrammetry: A low-cost, effective tool for geoscience applications. *Geomorphology*, 179:300--314.

Wilson, K. and Snavely, N. (2013). Network principles for sfm: Disambiguating repeated structures with local context. In *IEEE Int. Conf. on Comp. Vision*, pages 513--520.

Wu, C. (2013). Towards linear-time incremental structure from motion. In *Proceedings of the 2013 International Conference on 3D Vision*, 3DV '13.

Wurm, K. M., Hornung, A., Bennewitz, M., Stachniss, C., and Burgard, W. (2010). Octomap: A probabilistic, flexible, and compact 3d map representation for robotic systems. In *Proc. of the ICRA 2010 workshop on best practice in 3D perception and modeling for mobile manipulation*, volume 2.

Zhu, S., Fang, T., Xiao, J., and Quan, L. (2014). Local readjustment for high-resolution 3d reconstruction. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 3938--3945.