Universidad de Murcia

Facultad de Informática

Departamento de Ingeniería y Tecnología de Computadores

PhD thesis:

Algebraic epipolar constraints for efficient structureless multiview motion estimation

Antonio L. Rodríguez López

Advisors:

Pedro E. López de Teruel Alcolea

Alberto Ruiz García

June 2013

Abstract

Visual reconstruction methods such as Structure from Motion (SfM) or visual SLAM can be successfully used nowadays in tasks such as autonomous robotic navigation, augmented reality, or 3D scene reconstruction. Increasing the computational efficiency of these methods has been a persistent interest in the research community. This led to important reductions in time and energy consumption, and increased the chances of their integration in smaller or cheaper hardware, such as lightweight robotic platforms, smartphones or low-end commodity hardware. An important time-consuming operation in incremental SfM is the bundle adjustment (BA) refinement. A large number of improvements have been proposed in the literature to speed up this operation, including structureless BA, where the cost optimized is not based on the re-projection error, but on multiple view relations such as the epipolar or trifocal constraints. This way the cost does not involve the structure parameters, thus improving the computational efficiency of its optimization.

In this work we propose GEA (*Global Epipolar Adjustment*), a high-performance structureless BA correction method based on algebraic epipolar constraints. Due to the algebraic nature of the GEA cost, it can be optimized very efficiently, in most cases using a small fraction of the time required by BA to obtain the optimal configuration. Moreover, despite of this algebraic nature, under general circumstances the accuracy of the obtained camera poses is very close to that obtained with classical BA methods. We also propose a structureless incremental motion estimation procedure which uses GEA to obtain accurate initializations for the camera poses. This procedure does not require composing feature trackings or the triangulation of scene landmarks. Instead, it just requires pairwise feature correspondences detected between the input images with standard image matching methods. Both the incremental motion estimation method and GEA are designed to be robust against the unavoidable outliers found by these matching techniques. The resulting camera poses can be used afterwards to obtain highly accurate sparse or dense estimations of the scene structure.

We demonstrate the advantages, computational efficiency and practical applications of the proposed technique on a large number of real reconstruction problems, with arbitrarily large sizes and near critical configurations, and discuss possible future research lines.

Resumen

La visión por computador es la rama de la informática dedicada al estudio de algoritmos de procesamiento de imagen para su comprensión. Esta disciplina incluye, entre otros, los métodos utilizados para obtener la estructura 3D de los objetos que aparecen en una serie de imágenes de entrada determinadas.

La base matemática de estos métodos es la geometría proyectiva, que estudia aquellas propiedades geométricas que son invariantes bajo las transformaciones proyectivas. Esta teoría fue utilizada por los pintores del Renacimiento para aumentar el realismo de sus dibujos, de forma que el tamaño y la disposición de los objetos pintados respetase las reglas de la perspectiva. Esto incrementó la semejanza de sus obras con el aspecto visual de la escena real que pretendían representar en ellas. Con la aparición de las computadoras la geometría proyectiva comenzó a usarse en el desarrollo de aplicaciones de renderización 3D, capaces de recrear fielmente en las secuencias de vídeo o la pantalla del ordenador, el aspecto visual correcto de escenas 3D pregeneradas.

Más tarde, con el desarrollo de las técnicas de visión artificial se invirtió el sentido de uso tradicional que se le había dado a la geometría proyectiva. En lugar de obtener la proyección 2D en una imagen de una escena 3D conocida, los formalismos de la geometría proyectiva se combinaron con métodos de inferencia estadística y algoritmos de procesamiento de imagen para obtener la estructura y localización de los objetos físicos en una escena 3D, a partir de su apariencia 2D en múltiples imágenes. Este fue el comienzo de la geometría de múltiples vistas, basada en el estudio de las relaciones proyectivas que surgen entre múltiples imágenes, capturadas desde distintos puntos de vista [Hartley and Zisserman, 2003].

Hoy en día esta teoría matemática se utiliza en una gran variedad de métodos de reconstrucción visuales. Estos son capaces de obtener información sobre las cámaras y la estructura de la escena, a partir de las imágenes proporcionadas por una gran variedad de dispositivos de vídeo y fuentes de imagen, tales como: cámaras de vídeo embebidas en plataformas robóticas, grandes colecciones de fotos no estructuradas, imágenes satelitales, cámaras digitales manuales, o cámaras web. La mayoría de estos métodos asumen una escena física rígida, de forma que se simplifica el problema de la reconstrucción. Sin embargo, ciertas aplicaciones requieren la reconstrucción de superficies con deformaciones dinámicas, que se pueden obtener utilizando algoritmos de recuperación de formas no rígidas [Salzmann et al., 2008; Perriollat,

Hartley and Bartoli, 2011; Moreno-Noguer and Porta, 2011]. En cualquier caso, esta tesis se centra sólo en los métodos para la reconstrucción visual de escenas rígidas, dado su amplio rango de aplicaciones posibles.

Dentro de los métodos de reconstrucción visual existen dos ramas principales, o familias de métodos: *visual simultaneous location and mapping* (VSLAM), que hacen uso de técnicas de filtrado, y *Structure from Motion* (SfM), que hacen uso de técnicas estadísticas y de optimización de funciones de coste. Ambas técnicas proporcionan un balance similar entre el coste del tiempo computacional requerido para obtener la reconstrucción de la escena, y la precisión con la que se obtiene. La mayoría de sistemas de reconstrucción visual actuales utilizan técnicas de SfM incrementales. Estas técnicas suelen proporcionar información sobre la odometría y la óptica de la cámara (o cámaras) utilizada para capturar las imágenes, como un resultado secundario que en muchas ocasiones tiene una alta utilidad práctica. De esta forma los sistemas de reconstrucción SfM incrementales se pueden usar no solo en la estimación de modelos 3D, sino para obtener la localización y movimiento de la cámara.

El núcleo de la mayoría de los sistemas de reconstrucción SfM incrementales es una técnica conocida como *bundle adjustment* (BA) [Triggs et al., 2000]. Esta técnica se usa principalmente para corregir errores en la inicialización de parámetros de cámaras y de la estructura, producidos habitualmente en las iteraciones del proceso de reconstrucción incremental. Básicamente, BA consiste en la optimización del error de reproyección correspondiente a los parámetros estimados de las cámaras y la estructura, usando técnicas de alto rendimiento tales como Levenberg-Marquardt. Usando esta técnica se previene en la mayoría de los casos la divergencia del proceso incremental respecto de la solución óptima, obteniéndose reconstrucciones lo más precisas posible.

La optimización del error de reproyección suele tener un importante coste computacional, en parte debido al gran número de parámetros implicados en la estructura. Debido a esta limitación, las técnicas de SfM incrementales no son todavía adecuadas para resolver determinados problemas en la práctica, tales como reconstrucción de gran escala en hardware de bajo rendimiento, o la estimación autónoma de odometría visual en dispositivos de bajo coste. Para conseguir tales fines, es preciso reducir primero los requisitos computacionales de los algoritmos involucrados en la reconstrucción visual, tales como BA. Esto además reducirá el consumo de energía y el tiempo requerido para obtener los modelos 3D y la información de las cámaras con estos métodos.

En ciertas aplicaciones la estructura 3D no es realmente necesaria, y pasa a convertirse en un mero resultado auxiliar, usado sólo para asegurar una estimación precisa y robusta de la odometría de la cámara. Además, obtener reconstrucciones 3D de alta precisión de la escena a partir de estimaciones precisas de las poses de cámara resulta sencillo y muy eficiente, gracias a métodos de triangulación [Hartley and Sturm, 1997; Lindstrom, 2010] y de estimación de modelos 3D densos [Furukawa and Ponce, 2007]. Por estos motivos, ha surgido un reciente interés en el campo de la visión por computador hacia el desarrollo de métodos sin estructura para la corrección de movimiento, tales como *pose-graph relaxation* (PGR) [Strasdat, Montiel and Davison, 2010*a*; Strasdat et al., 2011; Lategahn et al., 2012], métodos de filtrado *delayed-state* [Lu and Milios, 1997; Eustice, Pizarro and Singh, 2004; Eustice, Singh and Ma, 2005; Ila et al., 2007; Ila, Andrade-cetto and Sanfeliu, 2007], o de promediado de desplazamientos [Govindu, 2004; Hartley, Aftab and Trumpf, 2011]. Estos métodos mejoran las estimaciones de la cámara mediante la optimización de funciones de error definidas sobre restricciones de movimiento relativo entre las cámaras. Debido a que estos métodos no involucran a la estructura, el número de grados de libertad del coste optimizado es mucho menor, respecto de otras técnicas tales como BA.

En general, los métodos PGR se utilizan en aplicaciones de estimación de odometría visual en tiempo real, como un back-end para corregir los errores de deriva en presencia de información de cierre de bucle. Estos movimientos relativos se obtienen en la mayoría de los casos a partir del movimiento de la cámara estimada, y la estructura 3D calculada por el sistema front-end de odometría visual. Otros métodos tales como los promedios de movimiento [Govindu, 2004; Hartley, Aftab and Trumpf, 2011] se utilizan en modelado 3D fuera de línea, para obtener la inicialización de la cámara en conjuntos de imágenes de tamaño grande, sin la costosa corrección de los parámetros de la estructura. En este caso, los movimientos relativos se suelen estimar a partir de la geometría epipolar definida por las correspondencias de características detectadas en las imágenes de entrada.

La precisión de estos métodos depende en gran medida de la calidad con que se estiman los movimientos relativos que se incluirán posteriormente en el coste. que debe debe ser adecuada para que la optimización del error proporcione unos resultados aceptables. La obtención de estos movimientos relativos es generalmente un problema no trivial, que en la mayoría de los casos requiere de la estimación de la estructura, o del uso de métodos sofisticados para la discriminación de errores de correspondencia de imagen. Algunos de estos movimientos relativos estimados pueden ser incorrectos, sobre todo en problemas de estimación de movimiento de gran escala, y pueden degradar significativamente la calidad de los resultados obtenidos.

En esta tesis se propone un método de corrección de movimiento sin estructura conocido como *global epipolar adjustment* (GEA) [Rodríguez, López-de-Teruel and Ruiz, 2011*b*,*a*]. Esta técnica consiste en la optimización de un coste definido sobre múltiples vistas, basado en restricciones algebraicas epipolares promediadas bajo la norma L_2. Debido al carácter algebraico de este coste, y al no incluir restricciones entre múltiples vistas, GEA sacrifica una pequeña cantidad en la precisión de la pose de cámaras estimadas, en comparación con otros métodos de corrección geométrica tales como BA. Por contra, la optimización GEA requiere un tiempo de cálculo significativamente menor que la optimización del error de reproyección, incluso la realizada por implementaciones de BA del estado del arte.

El error epipolar de múltiples vistas optimizado por GEA es más simple y más regular que el error de reproyección, dado que no involucra los parámetros de la estructura. Por este motivo, en condiciones generales el coste epipolar se puede corregir con éxito usando métodos de optimización numérica más simples que Levenberg-Marquardt, tales como Gauss-Newton. En ausencia de configuraciones de movimiento críticas, tanto GEA como BA producen estimaciones de movimiento con una precisión similar.

Las cámaras obtenidas con el método propuesto son lo suficientemente precisas para obtener estimaciones de alta calidad de la estructura de la escena. Mientras tanto, este método requiere en la mayoría de los casos una fracción del coste computacional empleado por SBA para corregir la odometría de las cámaras. A diferencia de las restricciones de movimiento relativo, utilizadas por otros métodos de corrección de odometría tales como PGR, las restricciones epipolares son más fáciles de estimar a partir de las correspondencias de imagen con una precisión suficiente para obtener resultados de alta calidad, como se describe en este trabajo. La evaluación de estas restricciones epipolares precisas es sencilla y menos propensa a errores que la estimación de los movimientos relativos. Además, GEA puede ser adaptado y utilizado con éxito tanto en aplicaciones de estimación de odometría visual en tiempo real, como en aplicaciones de reconstrucción visual fuera de línea.

En el futuro, los métodos de BA sin estructura como GEA pueden convertirse en una herramienta importante para reducir el coste computacional de las aplicaciones de reconstrucción visual y estimación de odometría. En esta tesis ofrecemos argumentos sólidos que apoyan esta afirmación, paralelamente a una revisión del estado del arte en lo que se refiere a las principales técnicas de VSLAM y SfM. Por otra parte, ofrecemos detalles computacionales cruciales para programar una implementación GEA eficiente. En particular, describimos cómo comprimir la información de correspondencias de imagen en la etapa de preprocesamiento, con el menor coste computacional posible. También proponemos una forma exacta para incrementar el número de valores nulos en el Hessiano del coste epipolar de múltiples vistas. Esto reduce significativamente el tiempo empleado en la optimización del coste, sin que suponga un sacrificio importante en la calidad de los resultados obtenidos. Estas ventajas computacionales, combinadas con las interesantes propiedades matemáticas del coste propuesto, posicionan GEA como una alternativa competitiva a BA en una amplia gama de aplicaciones SfM.

En este documento proporcionamos argumentos empíricos y teóricos que explican las propiedades de GEA descritas. Por otra parte, también describimos cómo usar el método de optimización propuesto de forma satisfactoria, tanto en aplicaciones de estimación odométrica como de reconstrucción visual, con el fin de mejorar su eficiencia computacional. Entre otros usos, proponemos el uso de la optimización para acelerar y prevenir la divergencia de los pasos intermedios de un método de estimación de odometría visual incremental. Las poses de cámara obtenidas con este método se pueden usar para estimar reconstrucciones de escena de alta calidad, tanto densas como basadas en características de imagen. El método incremental propuesto es más eficiente que los métodos clásicos de SfM incrementales, que usan BA y estiman los parámetros de estructura en los pasos intermedios de la estimación de las poses de cámara. Este método puede utilizar métodos de consenso de muestreo clásicos (por ejemplo RANSAC, o PROSAC) para obtener geometrías epipolares entre parejas de imágenes de entrada. Gracias a una técnica de robustificación de coste, GEA puede hacer frente a las posibles geometrías epipolares incorrectas, que podrían haber sobrevivido a los consensos de muestreo, debido por ejemplo a condiciones de duplicidad estructural en la escena. Esta robustificación se puede implementar con una simple modificación en la evaluación paso de Gauss-Newton ecuación. Por lo tanto, no requiere cambios en el error coste, conservando su simplicidad y ventajas computacionales.

Proporcionamos los resultados de una larga cantidad de experimentos que evalúan empíricamente la eficiencia y precisión obtenidos con GEA. Además, estos experimentos contienen configuraciones cercanas a las que podrían degradar su rendimiento. Concretamente, contienen secuencias de movimiento críticas que pueden perjudicar la precisión de los métodos de corrección basados en restricciones de pares de vistas como GEA o PGR. Nuestros experimentos demuestran que GEA es capaz de obtener configuraciones de cámaras de manera muy eficiente y precisa, tanto para problemas de reconstrucción pequeños como para otros arbitrariamente grandes, incluso en aquellos que contienen configuraciones casi críticas.

Acknowledgements

First I want to thank my PhD advisors, for providing me the opportunity to work in the computer vision research field, as well as for their fundamental guidance and support during the development of this thesis. I am glad to have had great colleagues and good friends at the UM during my PhD grant, such as Ginés, Lorenzo, Pedro Cavestany, Óscar, Félix, Juan Fernández, Ester, Santi, Raúl, Davó, Pedro Javier, Pedro Martínez and Javier Cuenca, who always kept his door open and a warm pleasant conversation ready for anyone who passed by. I am very much obliged to my great lab companions Antonio, Irene, Fran, Bernardo, Merche, Luis, and Julio for creating a great work environment. In this regard, I want to thank specially to Germán, who is an excellent person and better friend.

I would also like to thank the peer reviewers who contributed to the present work with their valuable reviews, suggestions and feedback. In special to the anonymous reviewers of the BMVC'11 and CVPR'11 conferences, as well as to Richard Newcombe, Andrew Davison and George Vogiatzis of the LDRMC'11 workshop. I also want to thank those computer vision researchers such as Chanchang Wu, Nghia Ho, Yasutaka Furukawa, Jean Ponce, Noah Snavely, and many more, who selflessly release and share state of the art computer vision code, to help anyone who needs it for research purposes.

I am grateful to my good friends outside the UM such as Chema, Sandra, Guillermo, Maria and Esparza (a.k.a. Big Kahuna) for the numerous joyful times that we spent together during these years. Finally, I am also grateful to my parents for their support during my PhD grant.

Document notation

Unless stated otherwise, in this document we assume vectors to be column vectors. We denote them using the **bold** typeface, whereas their elements are denoted by the vector name in *slanted* typeface, with the corresponding index in subscript:

$$\mathbf{u} = (u_1, u_2, u_3, u_4, ..., u_n)^T$$

We will commonly denote matrices with a capital letter in *slanted* typeface. The Moore-Penrose pseudo-inverse of a given matrix A is denoted by A^+. The matrix elements are represented with *slanted* typeface and lower case, with the corresponding matrix indexes in subscript. For example:

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \end{pmatrix}$$

The operator $[\cdot]_\times$ denotes the vector to cross-product matrix conversion. Given a vector $\mathbf{v}$ of size 3:

$$[\mathbf{v}]_\times = \begin{pmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{pmatrix}$$

This notation can be used to represent the matrix equation equivalent to the cross product of two vectors $\mathbf{v}$, $\mathbf{w}$ of size 3:

$$\mathbf{v} \times \mathbf{w} = [\mathbf{v}]_\times \mathbf{w}$$

The notation $\mathfrak{v}_A$ is used to represent a vector containing the elements of a given matrix A, in row-major order. E.g. given the following matrix:

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

The vector $\mathfrak{v}_A$ is:

$$\mathfrak{v}_A = (1, 2, 3, 4, 5, 6)^T$$

List of acronyms

AR	Augmented reality
BA	Bundle adjustment
CG	Conjugate gradient
DLT	Direct linear transform
GEA	Global epipolar adjustment
IBM	Image-based modeling
iLBA	Incremental light bundle adjustment
LBA	Local bundle adjustment
LIDAR	Light Detection And Ranging
LM	Levenberg-Marquardt
LT	Linear triangulation
MLE	Maximum likelihood estimation
PCG	Preconditioned Conjugate Gradient
RBA	Relative bundle adjustment
RMSE	Root-mean-square error
SAD	Sum of absolute differences
SfM	Structure from motion
SOCP	Second-order cone programming
sSBA	Sparse sparse bundle adjustment
SSD	Sum of squares distances
UKF	Unscented Kalman filter
VSLAM	Visual simultaneous localization and mapping

Contents

Chapter 1

Introduction

1.1 Projective geometry in computer vision

Computer vision is a branch of computer science dedicated to the study of algorithms for image processing and understanding. This discipline is related, amongst other things, with methods used to obtain the 3D structure of the objects appearing in a given set of input images.

The mathematical basis for these methods is projective geometry, which studies the geometric properties that are invariant under projective transformations. This theory was used by Renaissance painters to enhance the realism of their drawings by enforcing the rules of perspective in the shape, size and arrangement of the painted objects. This increased the resemblance of their drawings with the visual appearance of real scenes, as they would be perceived by a human observer. Later on, with the appearance of computers the projective geometry rules were used to develop 3D rendering applications which faithfully recreate in video sequences, or the computer screen, the correct visual appearance of pregenerated 3D scenes.

Computer vision researchers reversed the direction of how projective geometry was used. Instead of simply obtaining 2D projections of a known 3D scene, the formalisms of projective geometry were combined with statistical inference methods and image processing algorithms, to obtain the camera information and the 3D structure of the objects in a scene from their 2D appearance over multiple images. This was the beginning of *multiple view geometry*, which is the study of geometric projective relationships arising between multiple views [Hartley and Zisserman, 2003].

Nowadays these results are used in a large variety of visual reconstruction methods, which can estimate the structure and camera pose information from images provided by many different video devices and image sources,

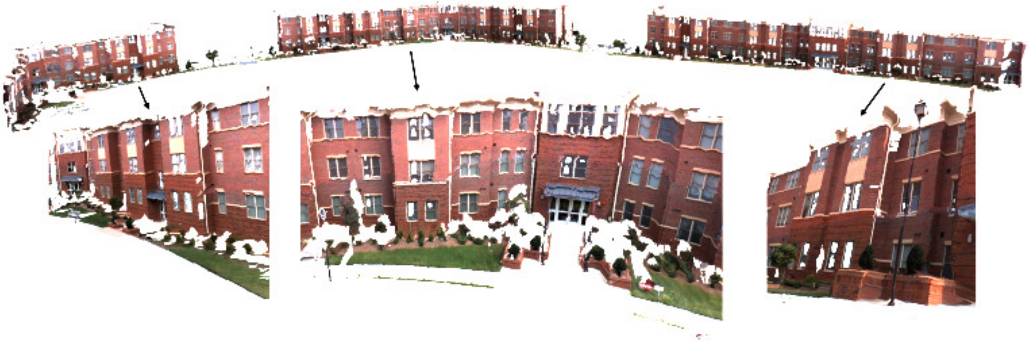


Figure 1.1: 3D model of the facades in a street side, obtained on-line from a moving vehicle with the system described at [Akbarzadeh et al., 2006]. The reconstruction application runs on an on-board computer, equipped with a powerful *Graphical Processing Units* (GPU's), and obtains a simplified 3D model of the street, consisting on planar surfaces detected at the building facades.

such as: video cameras embedded in moving robotic platforms, large unstructured photo collections, satellite images, hand-held digital cameras, webcams, surveillance *closed-circuit television* (CCTV) or stereo rigs. To succeed, most of these methods simplify the reconstruction problem by assuming a rigid physical scene. However, certain applications require the reconstruction of dynamic surfaces, which can be obtained using non-rigid shape recovery algorithms [Salzmann et al., 2008; Perriollat, Hartley and Bartoli, 2011; Moreno-Noguer and Porta, 2011]. Nevertheless, in this thesis we will focus only on methods for visual reconstruction of rigid scenes.

The practical uses of visual reconstruction methods are numerous. For example, in *Image-based modeling* (IBM) applications, we obtain textured 3D models for the objects appearing in a given set of images. Initially they were oriented mostly to automatic retrieval of 3D textured models of architectural sites. These 3D models could be obtained from sets of input images [Debevec, Taylor and Malik, 1996; Dick, Torr and Cipolla, 2004] as well as from video sequences [Pollefeys et al., 2004; Akbarzadeh et al., 2006; Cornelis et al., 2008]. On most occasions these applications take advantage of special features of urban scenes such as the planar-dominant structure of buildings and streets, to obtain visually compelling 3D scene models, as can be seen in figure 1.1.

State of the art IBM applications are able to obtain fully detailed 3D models from general large scale scenes. Using commodity hardware and low-cost cameras these applications can obtain the shape of an object on the fly [Pan, Reitmayr and Drummond, 2009; d. Hengel et al., 2009]. With

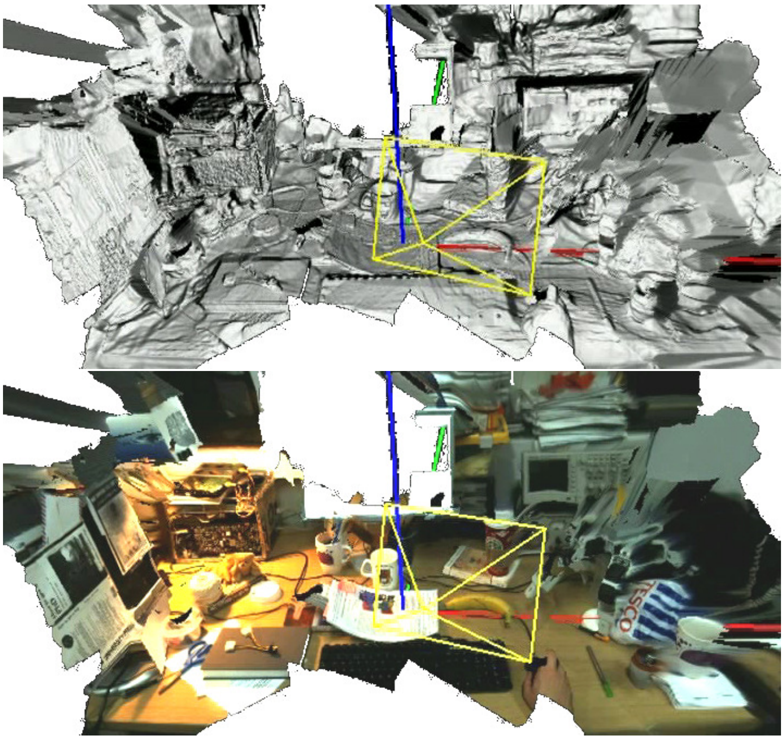


Figure 1.2: Dense 3D model reconstruction obtained on-line with the system described in [Newcombe and Davison, 2010], for a desktop scene. Upper image: dense 3D model for the desktop scene. Lower image: textured version of the model.

high performance hardware, such as cloud computing stations [Furukawa and Ponce, 2007; Furukawa et al., 2010] or desktop computers equipped with GPU [Frahm et al., 2010] these applications can obtain accurate textured 3D models under reasonable time constraints, or even in real-time for small to medium scenes [Newcombe and Davison, 2010; Newcombe, Lovegrove and Davison, 2011] as can be seen in figure 1.2.

Thanks to massive image sharing and search sites such as *Google Images* [1], *Flickr* [2] or *Picasa* [3], the Internet has become a comprehensive and increasing photographic record of the most interesting scenes and sites from around the world. These on-line services provide thousands of pictures capturing popular building facades, sculptures, streets, public indoor and city corners, under varying view points, illumination conditions, and camera models. For example, Flickr contains thousands of hits for search terms such as *"Trevi Fountain"*, *"Venus de Milo"*, *"Trafalgar Square London"*, or *"Piazza San*

[1] `http://images.google.es/`

[2] `http://www.flickr.com/`

[3] `http://picasaweb.google.com/`

Marco". Processing these image databases to obtain detailed reconstructions for those locations, such as the 3D model shown in figure 1.3, is a remarkable result of the computer vision research field [Agarwal et al., 2009].

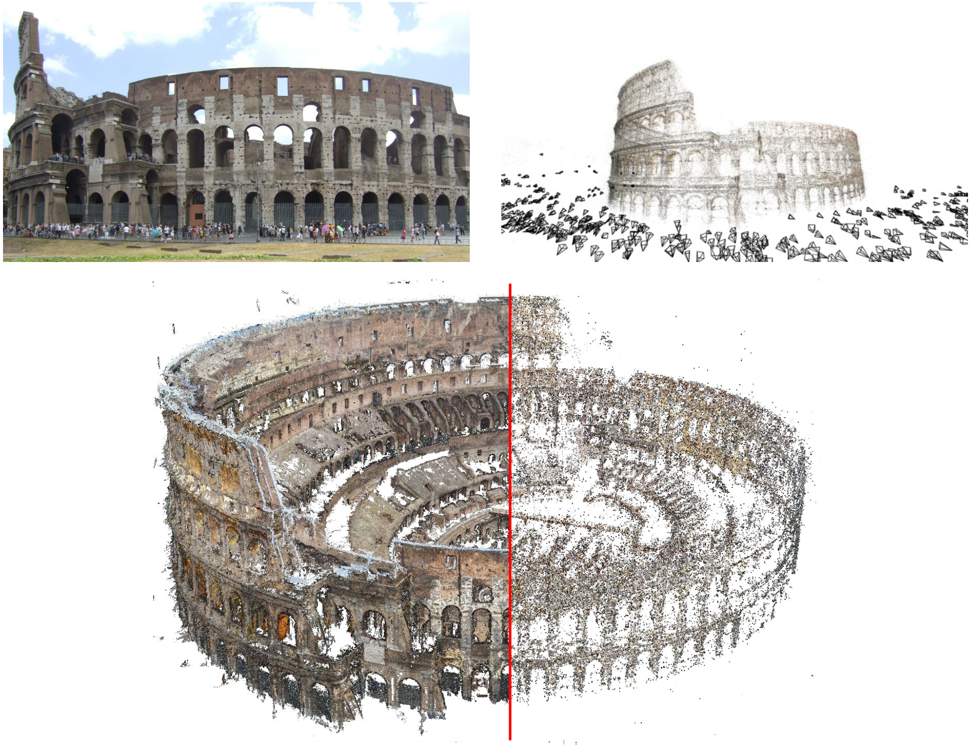


Figure 1.3: A 3D model of the Colosseum in Rome, obtained with the techniques described in [Furukawa and Ponce, 2007; Furukawa et al., 2010] from an Internet collection of pictures using open source applications for batch SfM reconstruction. **Top left:** one of the pictures used in the reconstruction process. **Top right:** sparse reconstruction obtained with the Bundler application [Snavely, 2012]. In this picture the black square pyramids represent the estimated camera poses for the input images, while the points in the cloud correspond to the image features detected on the physical surface of the Colosseum building. **Bottom:** comparison of the point-cloud sparse reconstruction (right half) with the final textured 3D model (left half) obtained using dense reconstruction applications such as PMVS2 [Furukawa and Ponce, 2012].

Structure reconstruction methods will usually provide information about the location and optics for the camera (or cameras) used to capture the images, as a highly useful side-effect result. In most applications, obtaining the camera information can be an objective by itself. For example, structure estimation methods can be used to tag images with precise geolocation

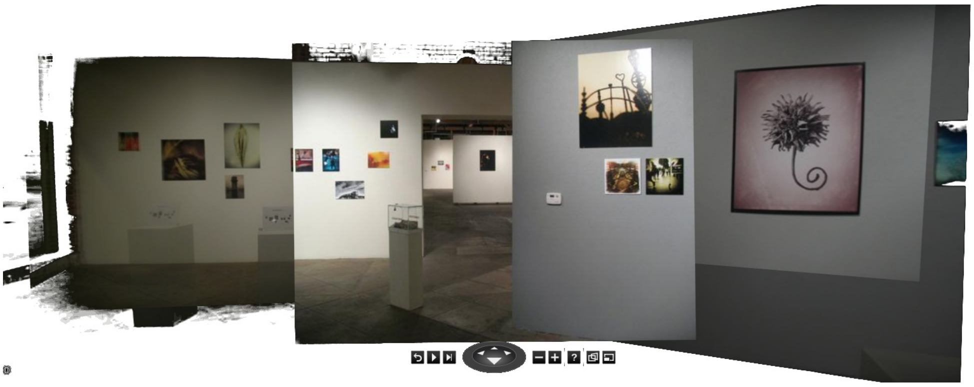


Figure 1.4: Navigable image mosaic, generated with with Photosynth [Labs, 2012] from pictures captured inside an art gallery. Photosynth is an on-line application based on the Phototourism project for SfM reconstruction [Snavely, Seitz and Szeliski, 2006]. The application renders on an image mosaic the approximate visual appearance of the scene, as seen from a given view point. The user can change this view point using the interface to navigate through the scene in a simulated virtual presence.

information [Snavely, Seitz and Szeliski, 2006; Irschara et al., 2009]. This can be useful for certain image browsing and retrieval applications such as Google Street View[4] [Anguelov et al., 2010], Photosynth[5] [Snavely, Seitz and Szeliski, 2006] (shown in figure 1.4) or the early 80' application Aspen Movie Map [Lippman, 1980]. These applications can provide to the user a kind of surrogate traveling experience, with the visual interactive navigation through the images in the database.

When the input images for the structure estimation procedure is a sequence of frames captured on-line with a video camera, these techniques can estimate the motion performed by the camera during the recording of the video sequence. This process is known as *visual odometry*, and can be used in a highly diverse range of applications. For example, in unmanned navigation applications which can guide robots, or autonomous vehicles such as cars or aerial drones through previously unknown locations. The structure reconstruction can be used to map in real-time the unknown scene, and the objects inside it, while the visual odometry is useful to detect the robot location inside this map during the process. This way the robot can transverse previously unknown scenes, avoiding any physical obstacles, as well as finding its way back through previously visited locations.

Camera motion tracking techniques can also be used in *augmented reality*

[4] http://maps.google.com/streetview/

[5] http://www.photosynth.com/

(AR) applications. The input images from the video camera can be enhanced with virtual 3D objects which appear to move solidarily with the real scene, by using the camera location information estimated with visual odometry methods [Klein and Murray, 2007; d. Hengel et al., 2009].

1.1.1 Types of scene reconstructions

The reconstruction obtained from the images can be dense or sparse, depending on its detail level. Feature-based maps (or sparse reconstructions) represent the physical scene structure with a 3D cloud of features from the scene, corresponding to certain image features detected at the input images. These features can resemble blobs, corners, lines or other geometrical shapes easily identifiable at the input images using automatic feature detection methods [Tuytelaars and Mikolajczyk, 2008]. VSLAM or SfM techniques usually match these significant features between the input images, and use regression and/or filtering methods to robustly estimate from these matchings the 3D location of the features in the scene, along with the camera poses.

Meanwhile, dense reconstructions provide a continuous representation of the 3D surface of the objects in the scene, even in areas of the scene where it is difficult to detect image features. These methods usually merge information of every pixel from the input images into data structures such as depth maps, or 3D meshes with millions of vertices. Hence they usually contain a vast amount of information from the scene, larger than the information contained in sparse reconstructions.

Nowadays, real-time applications can obtain dense mappings on the fly, using high performance hardware such as GPU [Newcombe and Davison, 2010; Newcombe, Lovegrove and Davison, 2011]. Thanks to the popularization of affordable graphical processing units, these dense systems can be used on middle to high-end commodity hardware. However, obtaining dense scene mappings still requires a high computational cost, a condition that makes it prohibitive for most portable hardware platforms such as low budget commodity laptops, smartphones, tablet PC's, or autonomous robots with small weight, size and energy consumption profiles. Working with sparse maps tends to require less computational resources than working with dense ones, so the former are commonly used in this kind of applications to perform online 3D scene reconstruction and motion estimation [Davison, 2003; Davison et al., 2007; Klein and Murray, 2007].

Another kind of scene mapping known as *topological* does not contain metric information of the scene structure, as feature or dense mappings do. Instead, topological maps can contain higher level or conceptual information, such as imprecise 3D structure measurements, overall location of the objects in the scene, connectivity between different areas of the scene, and so on.

These topological representations usually describe the scene with graph structures, derived from theories of human cognitive mapping. Nodes and links in these graphs can respectively represent scene location and known transition paths between adjacent locations.

Several probabilistic methods such as Markov localization [Simmons and Koenig, 1995] or appearance SLAM [Ho and Newman, 2007; Cummins and Newman, 2007, 2008] have been proposed to estimate these topological representations from the input image sequences.

They have the advantage of being less computationally demanding than metric mapping methods, as in most cases they only estimate visual correlation information between the different views. This kind of methods can be used to solve problems such as loop closing detection, or place recognition (also known as the *kidnapped robot* problem).

However, certain problems such as obstacle avoidance can be difficult to solve with topological maps. For this reason, applications for visual odometry and/or structure estimation usually combine topological and metric mapping to perform an efficient and robust scene reconstruction.

1.1.2 Alternatives for visual reconstruction

Some of the results described in the previous section could also be achieved with alternative non-visual techniques, based on information obtained from sensors other than cameras. For example, the device odometry and the scene map can be obtained using ultrasonic [Crowley, 1989] or sonar sensors [Leonard, Durrant-Whyte and Cox, 1990]. Laser range scanners and structured-light based depth cameras can be used to obtain clouds of points from the scene with high precision and detail [Gonzalez, Ollero and Reina, 1994; Lu and Milios, 1997]. Using high performance hardware such as GPU, these point clouds can be merged into a dense scene reconstruction while tracking the device location in real-time [Izadi et al., 2011], as can be seen in figure 1.5. The location problem can be also estimated using sensors such as GPS, accelerometers or beacon-based navigation networks [Whitcomb et al., 1999].

Nevertheless, computer vision methods for odometry and scene mapping offer competitive advantages over non visual solutions. Visual sensors provide a vast amount of raw information from the scene per time unit, larger than the information provided by other sensors. Given enough computational resources and/or efficient processing techniques, visual reconstruction methods can produce highly accurate and robust results, under a wide variety of conditions where other sensors can fail.

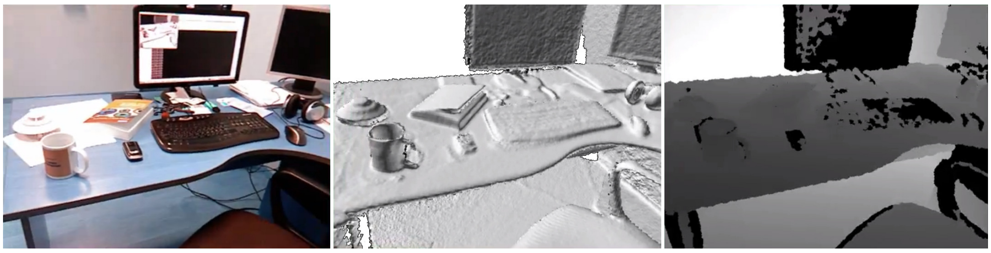


Figure 1.5: Real-time dense structure estimation with the open source Point Cloud Library (PLC) (`http://www.pointclouds.org/`) and Kinect depth camera. **Left:** input RGB. **Middle:** image of surface normals. **Right:** depth map.

Structured light depth cameras such as Kinect obtain scene measurements with too much noise when they operate in outdoor environments. Even at indoor scenes certain luminance conditions can significantly degrade their measurement accuracy in practice. Some types of surfaces reflect the structured light and produce incorrect reconstructions. Furthermore, the structured light beams from multiple devices operating simultaneously at the same area may collide, reducing the accuracy of the sensors estimations.

These depth sensors, as well as others such as *Light Detection And Ranging* (LIDAR) scanners have other important practical limitations. The effective range distance for LIDAR devices is normally less than 100 meters, while Kinect operates at an effective distance range between 0.8 and 4.0 meters. In some applications for LIDAR this range can be smaller.

Structured light devices can be combined with computer vision techniques to overcome this limitation. For example, Stanley [Thrun et al., 2006] the self-driving vehicle winner of the DARPA challenge at 2006, used the input information from an array of LIDAR sensors, along with the input of a common RGB video camera. The LIDAR system was used to obtain a reliable reconstruction of the road up to a maximum distance of 25 meters in front of the vehicle. The video camera and two RADAR sensors provided information for long-range road perception and detection of large obstacles respectively. This way the vehicle could progress at a reasonable speed without driving off-road, or colliding with other vehicles.

As long as there is enough baseline distance between camera centers for the captured images, visual techniques can obtain a precise depth estimation for both indoor and outdoor scenes, under a reasonable variety of natural and artificial luminance conditions. This is true even if the scene objects are located thousands of meters away from the cameras.

The quality of dense 3D models obtained with pure visual methods can be comparable to that obtained with these devices [Seitz et al., 2006] and does not have these practical limitations.

Devices such as accelerometers produce motion estimations which can accumulate a large drift error during long tracking sessions. These devices require the support of an alternative method for pose estimation to obtain reliable sensor motion estimations in the long run. GPS devices obtain global positioning coordinates with a measurement error larger than one meter, and do not work correctly at indoor or underwater locations.

On the other hand, a personal smartphone equipped with a video camera could perform ego-location in indoor scenarios where GPS would not be available, such as a mall, an airport, a factory or any other large building [Irschara et al., 2009; Ruiz et al., 2011]. Vehicles for biological and/or archeology study must rely on external resources to triangulate their location, such as acoustic beacon networks [Whitcomb et al., 1999], or techniques for visual odometry. In the case of underwater scenes the visual information can be mixed with the signal from the transponder net, to increase the accuracy of the measurement, or even be used solely in case the transponder net would not be available [Eustice, Pizarro and Singh, 2004; Eustice, Singh and Ma, 2005]. GPS information can be used in batch SfM systems to improve the accuracy of the obtained 3D models [Crandall et al., 2011], but undoubtedly the visual information is crucial in the structure estimation task.

1.2 Motivation of this thesis

Much research effort has been dedicated to increase the computational efficiency of techniques for visual odometry and structure estimation. As a result, these techniques can now obtain large scale 3D models under reasonable time constraints, using for example high performance hardware such as GPU or cloud computing. These techniques can also estimate the motion and scene reconstruction in mobile platforms or portable devices such as quadcopters, ground vehicles, smart phones or tablet PC's using distributed reconstruction pipelines [Wendel et al., 2012], or adapting the reconstruction techniques to perform local scene structure and motion estimation [Wagner et al., 2008; Klein and Murray, 2009].

However, due to their computational performance limitations, visual reconstruction techniques are still not adequate to solve certain practical problems, such as robust large scale and fully autonomous visual odometry and structure estimation on lightweight devices. In order to do so, we should develop techniques to accelerate the reconstruction software. This would also reduce the energy consumption, hardware cost and time required by off-line reconstruction applications for 3D object modeling.

Most visual reconstruction techniques estimate simultaneously the camera location and the scene structure. For example, *bundle adjustment* (BA) [Triggs et al., 2000] is a numerical optimization procedure which improves jointly the estimated camera pose and structure configuration by minimizing the reprojection cost error. This optimization has important computational requirements, partly due to the large number of parameters involved. In certain visual odometry applications the structure is of no interest and is a mere auxiliary measurement, used only to ensure an accurate and robust motion estimation. Furthermore, with accurate estimations of the camera poses we can obtain high quality structure reconstructions using sparse triangulation [Hartley and Sturm, 1997; Lindstrom, 2010] and dense model estimation methods [Furukawa and Ponce, 2007]. For this reason there has been a recent growing interest in the development of structureless methods for motion correction and initialization, as a mean to increase the computation efficiency of both real-time and off-line visual odometry applications. These methods do not involve the structure parameters in the optimization, hence they require much less computational cost to obtain improved motion estimations. *Motionless* methods, which correct the structure without involving the camera parameters [Li, 2010] are another promising approach to reduce the number of parameters in visual reconstruction problems. However, this research line is still in an early stage nowadays.

Some of the structureless methods, such as pose-graph relaxation [Strasdat, Montiel and Davison, 2010*a*; Strasdat et al., 2011; Lategahn et al., 2012], delayed-state filtering methods [Lu and Milios, 1997; Eustice, Pizarro and Singh, 2004; Eustice, Singh and Ma, 2005; Ila et al., 2007; Ila, Andrade-cetto and Sanfeliu, 2007], or motion averaging [Govindu, 2004; Hartley, Aftab and Trumpf, 2011] find the estimations for the camera poses by optimizing a cost defined on relative motion constraints. The accuracy of these methods depends significantly on the quality of the relative motions used, which must be adequate in order to produce accurate camera poses with the cost optimization. The estimation of these relative motions is usually a nontrivial problem, which in most cases involves the structure estimation, or highly discriminative methods for image mismatching rejection.

Pose-graph optimization methods are used in classical on-line visual odometry applications as a back-end to correct drift errors in presence of loop closing evidence. The pose-graph optimization must use accurate relative motions to provide corrected camera poses with high quality. These relative motions are obtained in most cases from the estimated camera motion and the 3D structure estimated at the front-end visual odometry system.

Other methods such as motion averaging [Govindu, 2004; Hartley, Aftab and Trumpf, 2011] are used in off-line 3D modeling to obtain the camera initialization for large sets of images, without the time consuming correction of the structure parameters. In this case the relative motions must be estimated from the epipolar geometry defined by the image feature matching information. Some of these relative motions can eventually be incorrectly estimated, and degrade the averaging results obtained. In large scale motion estimation problems it is likely that a few relative motions with too much estimation error will corrupt the results obtained, and produce incorrect camera pose estimations.

In this thesis we evaluate the advantages of a structureless motion correction known as *global epipolar adjustment* (GEA) [Rodríguez, López-de-Teruel and Ruiz, 2011*b*,*a*], which is based on algebraic epipolar constraints. This technique minimizes the least squares of the residuals for the algebraic epipolar cost defined between each view pair in the reconstruction. Compared with other geometric correction methods such as BA, GEA sacrifices a small amount of the accuracy in the estimated camera poses obtained, gaining this way a significant computational efficiency. The multiple view epipolar cost optimized by GEA is simpler and more regular than the reprojection error, provided that no structure parameters are involved. For this reason the GEA cost can be corrected with simpler numerical optimization methods, and obtain similar camera pose error correction results under general circumstances. The camera estimations obtained are sufficiently accurate to estimate a high quality scene reconstruction. Meanwhile, this optimization requires in most cases a fraction of the computational cost required by SBA to correct the camera poses for a given reconstruction configuration. Unlike other structureless motion correction methods, the epipolar constraints are easier to estimate robustly from the image matching information, as we describe in this work. Furthermore, GEA can be adapted and successfully used in either real-time or batch SfM applications without important modifications.

We describe how to integrate GEA in visual odometry or reconstruction applications to enhance their computational efficiency, and provide important details which are crucial for that purpose, such as the robustification of this motion correction technique against the appearance of image feature mismatchings.

1.3 Structure of this thesis

In chapter 2 we will review the methodology used for visual odometry and structure computation in computer vision. We will also provide and discuss the main techniques and references for structureless motion correction techniques.

In chapter 3 we describe the GEA motion correction method in detail. We provide theoretical arguments which justify the accuracy and computational efficiency of the method, as well as several details required to develop a highly efficient implementation of the optimization. We compare the practical advantages and disadvantage of the optimization with those offered by BA, and the structureless correction methods based on relative motion constraints. In this chapter we also describe the failure conditions for GEA, such as critical motion sequences, and ways to prevent them.

Chapter 4 describes how to integrate the epipolar motion correction in an efficient incremental motion initialization procedure, which can be adapted to both real-time or batch visual odometry applications. This chapter provides a description for the data processing pipelines of the most common reconstruction applications, and show how to adapt them to use GEA, and obtain the advantages of the structureless correction. The chapter also describes how to robustify GEA against matching outliers, which is an important issue for practical reconstruction applications.

Chapter 5 provides the results obtained by an extensive set of tests, performed on a large number of real reconstruction problems to evaluate the performance of the GEA correction. In these tests the accuracy and efficiency of the method is compared against a state of the art BA implementation. These tests also evaluate the robustness of the correction against critical configurations, to show that the conditions for a GEA failure are not usually met in practice. Finally, the performance of the structureless incremental motion estimation described in chapter 4 is measured on several real scene reconstruction problems.

Chapter 6 contains the conclusions of this work. It enumerates the main contributions, and also offers some proposals for future work.

Chapter 2

Methodology for high performance visual reconstruction

This chapter introduces projective and multiple view geometry, along with the two main methodologies used in computer vision to obtain the structure and camera information from a set of input images: *structure from motion* (SfM) and *visual simultaneous localization and mapping* (VSLAM). The chapter provides notions to understand these topics, as well as important references to the actual state of the art.

2.1 Projective geometry

A given point $(x_1, x_2, ..., x_n)$ in the $\mathbb{R}^n$ euclidean space is represented in homogeneous coordinates by any point in the set $\{(y_1, y_2, ..., y_{n+1})\} \in \mathbb{R}^{n+1}$ which satisfies:

$$(x_1, x_2, ..., x_n, 1) \propto (y_1, y_2, ..., y_{n+1}) \tag{2.1}$$

This notation offers several advantages over Cartesian coordinates to represent points in the projective geometry equations. For example, most of the equations become simpler, and points at the infinity can be represented with finite coordinates.

The visual appearance for the objects in the 3D space is projected into a planar region contained in the 3D euclidean space, which is known as the *image plane*. In this projection model, the lines connecting 3D points in the object with their 2D image projections must coincide in a single point, known as the *camera center*, which must be located behind the image plane.

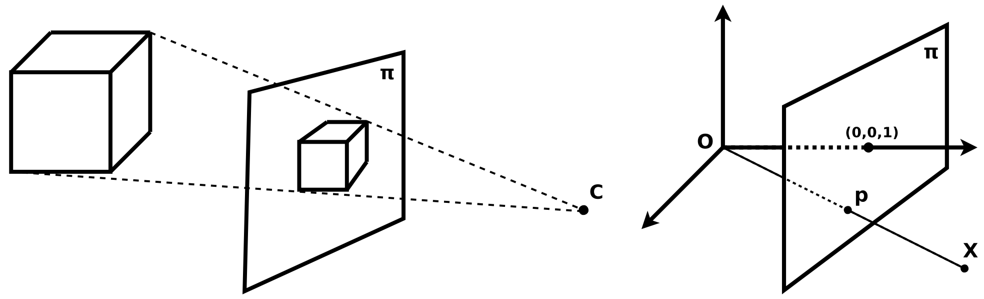


Figure 2.1: An object is projected onto the image plane π using the camera center **C** for the central projection model (**left**). A point **X** is projected to the point **p** in the image plane $\pi \equiv z = 1$, using a camera center at the origin of coordinates for the central projection model (**right**).

This image projection process preserves the true perspective of the scene, as it would be perceived by a human observer located at the camera center.

The following is a mathematical formalization of this process. Each 3D point in the euclidean space $\mathbf{X} = (x, y, z, 1)^T$, is projected with the pinhole camera model into the 2D point $\mathbf{p} = (p_x, p_y, 1)^T$ contained in the image plane $z = 1$, which satisfies the following equation:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \propto \begin{pmatrix} p_x \\ p_y \\ 1 \end{pmatrix} \tag{2.2}$$

In this equation, the camera center is assumed to be at the origin of coordinates $\mathbf{O} = (0, 0, 0)^T$. Figure 2.1 provides a graphical visualization which explains the projective equation, and illustrates how images are formed using the *p*inhole camera or central projection model.

In any real image the coordinates for the image projections will suffer linear and nonlinear distortions due to the camera optics. Furthermore, the camera can be located in a point different from the origin of coordinates, or have an arbitrary orientation.

To obtain the 2D projection of a 3D point into the projective plane under these conditions, one can simply apply the adequate transformation to the 3D points in the previous formula:

$$P\mathbf{X} \propto (p_x, p_y, 1)^T \tag{2.3}$$

In this expression P is a 3×4 matrix known as the *projective camera matrix.* This matrix encodes the camera pose, and the linear deformations of the camera optics.

The following expression is a usual way to decode these elements from the camera matrix:

$$P = KR\,[I| - \mathbf{C}] \tag{2.4}$$

In this equation K is an upper triangular matrix containing the linear distortions produced by the camera optics, R is a rotation matrix representing the camera rotation, and $\mathbf{C}$ is a 3D point representing the camera center location.

This formula does not model nonlinear distortions of the camera optics, such as the radial distortion. However it can still be used in most reconstruction problems to model the camera information accurately, without involving the nonlinear intrinsic camera parameters.

The linear distortions are modeled as follows:

$$K = \begin{pmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \tag{2.5}$$

The intrinsic parameters contained in this matrix are: horizontal and vertical focal distances (f_x, f_y), skew (s), and principal point (c_x, c_y).

The matrix R must be an element from the Special Orthogonal group $SO(3)$. This is the way to ensure that R represents a valid camera orientation. For that purpose the rotation is usually expressed with a minimal vector parametrization $\mathbf{w}$ containing 3 numerical values.

These values can be mapped to a rotation matrix $R_{\mathbf{w}} \in SO(3)$ using several methods. For example, the three values in $\mathbf{w}$ can be interpreted as Euler angles (ϕ, θ, ψ). This way, the rotation matrix can be obtained from the following expression:

$$R_{\mathbf{w}} = R_X R_Y R_Z \tag{2.6}$$

where R_X, R_Y, and R_Z are elements of SO(3) which represent rotations around the x, y and z axes respectively:

$$R_X = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{pmatrix} \tag{2.7}$$

$$R_Y = \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \tag{2.8}$$

$$R_Z = \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.9}$$

The vector $\mathbf{w}$ can also be converted to the skew symmetric matrix $[\mathbf{w}]_\times$, which is an element of the Lie algebra $\mathfrak{so}(3)$ associated to the group $SO(3)$. Each element $[\mathbf{w}]_\times \in \mathfrak{so}(3)$ can be mapped to a rotation matrix in $SO(3)$ using the following expression, which is known as the Rodrigues formula for skew-symmetric matrix exponentiation:

$$R_{\mathbf{w}} = e^{[\mathbf{w}]_\times} = I + \frac{sin\left(\|\mathbf{w}\|\right)}{\|\mathbf{w}\|}[\mathbf{w}]_\times + \frac{(1 - cos(\|\mathbf{w}\|))}{\|\mathbf{w}\|^2}\mathbf{w}\mathbf{w}^T \tag{2.10}$$

2.2 Filtering methods for Visual SLAM

Visual SLAM techniques combine the method known as Simultaneous Location And Mapping (SLAM) [Smith, Self and Cheeseman, 1988, 1986] with the projective geometry, to develop on-line camera motion and structure estimation applications.

SLAM methods were originally designed to be used in the perceptive loop of the control software of a robot, to perform a robust estimation of the sensor motion and location.These methods can be applied to a large variety of sensors, such as LIDAR, sonar sensors, depth cameras, GPS and so on.

SLAM systems use a stochastic map containing information about the expectancy and uncertainty for the location of the sensor, and in most cases the location for certain scene references, or landmarks, which are detected in the sensor signal by the SLAM system.

SLAM methods must define an observation model, which is used to evaluate the expected values for each sensor location and landmark observation from the map state. During a sensor tracking session, with each new observation the SLAM application updates the stochastic map, using filtering methods such as the *unscented Kalman filter* (UKF) [Julier, Jeffrey and Uhlmann, 2004], or particle filters [Eade and Drummond, 2006] amongst others.

In Visual SLAM applications the sensor is a video camera, which provides input frames to the SLAM system. The landmarks are image features, detected on the frames using image feature detection algorithms.

2.2.1 Feature-based mapping

In feature-based mapping applications, the map contains the sensor location and the 3D locations for the image features from the scene. The observation model in the filtering process uses projective geometry to measure the reprojection error between the expected and measured image locations for each scene feature in the map. With these errors the filter updates the map, so the discrepancy between measured and expected image feature locations is reduced, as well as the uncertainty for the estimated camera pose and the features in the map.

MonoSLAM [Davison, 2003; Davison et al., 2007] was one of the first feature-based mapping applications to successfully run a real-time camera motion estimation and feature mapping process on commodity hardware, using solely visual information from a single camera. In MonoSLAM the landmarks are detected in the input images using the Shi-Tomasi operator [Shi and Tomasi, 1994]. With each new observation an EKF filter is used in this SLAM application to correct the estimations for the camera location and the 3D scene features in the stochastic map. The information for past camera poses is marginalized in the map with each update, and the information gained over time is summarized with a probability distribution.

This SLAM system is highly efficient and robust, and obtains accurate motion and map estimations. Subsequent proposals for visual SLAM take advantage of the conditional independence inherent in the parameters for the SLAM problem to reduce the computational requirements of the EKF filter. For example, using tree structures to represent landmark estimations [Montemerlo et al., 2002] or particle filters instead of the EKF filter to merge sensor information in the map [Eade and Drummond, 2006].

Other proposals exploit the inherent sparsity in the information matrix of the EKF filter. The map update time with standard EKF grows quadratic with the number of features in the map. The conditional independence in the joint distribution of these features produces a natural quasi-sparsity in the information matrix, where many elements corresponding to weakly correlated variables will contain negligible values. Sparse Extended Information Filters (SEIF) [Thrun et al., 2004] use a sparse approximation for the information matrix, where these elements are set to zero. This way the SEIF filter can reduce the quadratic map update time to a constant time, while obtaining a similar map quality.

Yet another method to reduce the update time is to partition the map into several manageable submaps [Leonard et al., 1999; Leonard and Newman, 2003; Estrada and Tardós, 2005]. With this approach each submap contains a fixed maximum number of elements, so the update time for the EKF filter is upper-bounded. This however reduces the convergence rate to the optimal configuration of the full set of estimated features [Leonard and Newman, 2003].

2.2.2 Full and delayed-state SLAM

Eventually SLAM applications can detect image feature correspondences between previously unrelated views during the tracking process. A special kind of these correspondences are loop closing matchings, which are obtained between views sufficiently separated in the video sequence when a given area of the scene is revisited. These correspondences can be useful to correct drift errors, improving the accuracy of the estimated reconstruction parameters.

Feature-based filtering methods, which are described in the previous section, marginalize the information for past camera poses. Correcting drift errors in this kind of maps with loop closing evidence can be difficult [Estrada and Tardós, 2005; Eade and Drummond, 2008; Williams et al., 2009]. This can be a practical disadvantage for feature-based mapping filters during long camera tracking sessions, where the drift error can grow unbounded.

Feature-based filters can make incorrect updates in the map that cannot be easily undone, such as including in the map information from estimated camera poses which suffer a significant drift error. In the case of hierarchical SLAM methods the global map can be updated in presence of loop closing evidence by adjusting the transformations between submaps in a nonlinear constrained optimization [Estrada and Tardós, 2005]. Still, the problem of the reduced local convergence remains.

Full SLAM mapping is an alternative to feature-based filtering, where the stochastic map includes the trajectory performed by the sensor during the whole tracking [Thrun, Burgard and Fox, 2001; Dellaert and Kaess, 2006; Montemerlo and Thrun, 2007; Kaess, Ranganathan and Dellaert, 2008]. This way the method can include loop closing evidence and correct drift errors efficiently.

Delayed-state SLAM filtering is a different kind of mapping method which only includes the parameters of the estimated camera trajectory [Lu and Milios, 1997; Eustice, Pizarro and Singh, 2004; Eustice, Singh and Ma, 2005; Ila et al., 2007; Ila, Andrade-cetto and Sanfeliu, 2007]. In feature-based and full SLAM the map is updated to reduce the discrepancy between expected and measured image observations. In most cases this discrepancy is basically the reprojection error. In the case of delayed-state SLAM the map is updated to reduce the discrepancy between expected and measured relative camera motions. Some of these relative motions can be measured directly from the local visual odometry obtained with feature-based or full SLAM methods. Other relative motions can be estimated as well from loop closing evidence.

The main advantage of delayed-state SLAM over full SLAM is the reduced computational cost required to update the map. In full SLAM the number of feature parameters is commonly an order of magnitude or more larger than the number of camera parameters. By containing parameters only for the camera poses, maps for delayed-state filtering are much smaller, and can be updated with higher efficiency.

Furthermore, the information matrix of delayed-state maps is naturally sparse, as feature correspondences are marginalized in the relative camera motions. Null elements in the information matrix correspond to view pairs not related by point correspondences, and (vice versa) nonzero elements in the information matrix correspond to view pairs related by point correspondences. This way delayed-state maps can be updated using efficient exact sparse matrix operations [Eustice, Singh and Ma, 2005].

2.2.3 Keyframe selection and problem reduction

In real-time SLAM applications most of the scene information will usually be redundant across the different frames of the input video sequence. Views with close camera poses will contain many observations for the same map features. Hence their contribution to the SLAM problem is similar.

Using the correspondences between these views in the map correction increases the update time, as it reduces the sparseness and increases the size of the information matrix. Furthermore, due to certain linearization choices assumed in filter methods such as EKF, this also can lead to inconsistent overestimation of the reconstruction parameters in long-term experiments, for simple but realistic SLAM scenarios [Julier and Uhlmann, 2001; Bailey et al., 2006].

Several strategies alleviate the redundancy problem by reducing the number of variables in the map, and the observations involved in the map updating.

In the strategy known as keyframe based SLAM, only those frames which are key to the reconstruction problem are included in the map [Konolige and Agrawal, 2008]. This way the number of parameters can be drastically reduced (up to a tenth percent or less from the original, depending on the video sequence) while the estimated camera poses and the structure obtained will still be accurate.

The second strategy is to reduce the number of relative camera motions used to update the map in delayed-state filter methods [Eustice, Singh and Ma, 2005]. By doing so, we can control exactly the sparsity level of the information matrix, and thus the update time cost.

Both strategies can be combined in highly efficient delayed-sparse SLAM applications which use minimal stochastic maps and still obtain a highly accurate estimation of the camera tracking. The exact information gain for each element in the map can be evaluated in a closed form, so relative motions and views which are redundant in the reconstruction can be effectively identified and excluded from the map update [Ila, Porta and Andrade-Cetto, 2009].

2.2.4 Relation with SfM methods

Filtering methods such as full SLAM are closely related to BA. Assuming an isotropic Gaussian distribution with fixed known variance for the reprojection residuals, both the filtering method and the least squares optimization converge to the same statistical *maximum likelihood estimation* (MLE) for the unknown parameters.

2.3 Multiple view geometry

This section discusses multiple view geometry methods used in SfM to estimate the configuration of both the scene structure, and the set of cameras.

The optimal structure and camera configuration can be found by solving a set of equations. Given the complexity of these equations for nontrivial problems, the solutions are usually obtained with iterative optimization procedures, which reduce the least squares cost of the equations.

These cost errors however can have many local minima, hence starting the optimization from an arbitrary camera pose configuration can produce convergence to a suboptimal solution. To ensure convergence to the solution it is convenient to start the optimization from a configuration in the basin of the global optimal. Obtaining a reconstruction initialization in the basin of the optimal configuration for these cost errors is usually not trivial.

This section reviews the error costs used in the literature to obtain the solution for the equations, and also the methods used in SfM to obtain initial camera and structure configurations in the basin of the optimal configuration for these costs. This way using optimization procedures we can converge to that optimal reconstruction configuration.

Some reconstruction problems are simple enough to be correctly initialized with algebraic linear procedures. For problems with a large number of views, and a large structure, these solutions might be unreliable and bound to fail, so more sophisticated iterative initialization methods must be used to obtain a good initialization.

2.3.1 Bundle adjustment

BA is a reconstruction correction method used in SfM, which optimizes a cost error defined on the structure and camera parameters to improve their accuracy. This method was first developed in the photogrammetry research field [Brown, 1976; Granshaw, 1980; Slama, Theurer and Henriksen, 1980], and later became the core of most SfM applications.

Given the point features detected in a set of input images, the BA correction provides the least-squares solution for the following nonlinear system of equations:

$$\left\{\mathbf{p}_{ij} - \phi\left(P_i, \mathbf{X}_j\right) = \mathbf{0}\right\}_{\mathbf{p}_{ij} \in \mathcal{P}} \tag{2.11}$$

In these equations P_i is the projective matrix corresponding to the i-th camera. The vector $\mathbf{X}_j = (x, y, z, 1)^T$ contains the homogeneous representation of the coordinates for the j-th 3D feature. The set $\mathcal{P}$ contains the feature projections for the 3D points observed at the input images, and $\mathbf{p}_{ij}$ is the projection of the j-th 3D feature detected at the i-th view, represented in homogeneous coordinates. Finally ϕ is the nonlinear projection function:

$$\phi(P_i, \mathbf{X}_j) = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \propto P_i \mathbf{X}_j \tag{2.12}$$

These equations can be represented in the following general form:

$$\mathbf{f}(\mathbf{x}) = \mathbf{y} \tag{2.13}$$

In this representation $\mathbf{x}$ is known as the vector state, which contains the camera and structure configuration parameters. These are the coordinates for the 3D points, and the components of the projective matrices. The model function $\mathbf{f}$ maps the vector state to the expected coordinates for the image projections $\phi\left(P_i, \mathbf{X}_j\right)$, and the target vector $\mathbf{y}$ contains the image coordinates for the projections $\mathbf{p}_{ij} \in \mathcal{P}$.

In absence of noise, solving these equations we obtain the 3D location of the features and the projective matrices for the cameras. In practice, due to the presence of noise the equations might not be satisfied exactly, not even by the true camera and structure configurations. The least squares solution for these equations is provided by minimizing the following cost error, defined by the norm of the residuals for the equations in (2.11):

$$c(\mathbf{x}) = ||\mathbf{f}(\mathbf{x}) - \mathbf{y}||_n \tag{2.14}$$

This cost can be seen as a representation of the quality for a given estimated configuration of the cameras and the structure. The cost $c(\mathbf{x})$ can have one or several optimal solutions, depending on the choice norm n for equation (2.14), and the configuration of the reconstruction problem. Different norms will provide results with very different properties.

Assuming an euclidean norm ($n = 2$) for the equation residuals, the cost (2.14) becomes:

$$C_{RE} = \sum_{\mathbf{p}_{ij} \in \mathcal{P}} ||\mathbf{p}_{ij} - \phi(P_i, \mathbf{X}_j)||_2^2 \tag{2.15}$$

This cost is minimized in BA to correct the reconstruction accuracy, using computationally efficient numerical optimization methods such as *Levenberg-Marquardt* (LM) [Levenberg, 1944], or the trust-region based Powell's dog-leg [Lourakis and Argyros, 2005, 2009].

From a geometrical point of view, the cost C_{RE} measures the sum of squared distances between the expected and measured image locations, provided an estimated reconstruction configuration. Thus the error (2.15) represents how well the estimated structure and camera poses are related to the image feature information contained in the input images.

From a statistical point of view, BA is a regression procedure which fits the model parameters (camera poses and 3D structure configuration) into input noisy data (the image measurements). The optimal parameter configuration for the error (2.15) has a precise meaning when the measurement noise for the image features is normally distributed and isotropic. In this case the solution obtained with BA is the MLE for the structure and the cameras configuration.

Due to these properties BA is nowadays widely accepted as the gold-standard method for obtaining an optimal estimation of the camera and structure parameters from visual correspondence information.

However, the BA correction is highly sensitive to the initial parameter configuration used in the optimization. Like other costs based on the L_2 norm, the reprojection error has many local minima. To reach the optimal configuration the Levenberg-Marquardt correction of the reprojection error requires a sufficiently good starting point, hopefully inside of the basin for the optimal configuration. Otherwise the optimization can get stuck in a suboptimal solution.

2.3.2 Two view epipolar correction methods

Any given pair of image point projections $\mathbf{p} = (p_x, p_y, 1)^T$, $\mathbf{q} = (q_x, q_y, 1)^T$ detected in two different images, and corresponding to the same 3D point in the scene must satisfy the epipolar constraint:

$$\mathbf{q}^T F \mathbf{p} = 0 \tag{2.16}$$

The matrix F of size 3×3 in this expression, known as the fundamental matrix, encodes the camera information for the views i and j. Assuming that the cameras are parametrized as projective camera matrices P_1, P_2, the F matrix can be obtained with the following expression:

$$F = [P_2 \mathbf{C}_1]_\times P_2 P_1^+ \tag{2.17}$$

In this expression the vector $\mathbf{C}_1$ is the null-vector of P_1 (as well as the camera center for the first view). The operator $[\cdot]_\times$ denotes the vector to cross-product matrix conversion:

$$[\mathbf{v}]_\times = \begin{pmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{pmatrix} \tag{2.18}$$

In the calibrated case, the fundamental matrix in the expression (2.16) becomes an essential matrix E which can be evaluated using the following expression:

$$E = R_2 \left[\mathbf{C}_2 - \mathbf{C}_1\right]_\times R_1^T \tag{2.19}$$

where R_1 and R_2 are respectively the camera orientation matrices for the first and second camera poses. The vectors $\mathbf{C}_1$, $\mathbf{C}_2$ are the coordinates for the first and second camera centers.

Many cost errors based on the epipolar constraint have been proposed for optimal camera pose estimation in two-view reconstructions. These errors are usually based on the epipolar geometry. In this section we discuss several of these costs, which have been used to correct the initial pairwise camera pose estimations.

Given the set of pairwise matchings $\mathcal{M} = \{\mathbf{p}_k \leftrightarrow \mathbf{q}_k\}_{k=1..n}$ corresponding image point projections detected in two different images the optimization of the following geometric cost will provide the MLE for the camera pose parameters [Vidal et al., 2001], assuming an isotropic measurement noise in the image feature coordinates:

$$C_{GE}(\mathcal{M}, F) = \sum_{k=1}^{n} \|\mathbf{p}_k - \hat{\mathbf{p}}_k\|^2 + \|\mathbf{q}_k - \hat{\mathbf{q}}_k\|^2 \tag{2.20}$$

The auxiliary parameters $\hat{\mathbf{p}}_k$, $\hat{\mathbf{q}}_k$ are vectors of size 3 which contain the estimated image feature coordinates for the k-th matching in $\mathcal{M}$ in homogeneous coordinates. This cost must be optimized subject to the following constraints:

$$\hat{\mathbf{q}}_k^T F \hat{\mathbf{p}}_k = 0, \qquad \|F\| \neq 0, \qquad \hat{\mathbf{p}}_k^T \mathbf{e}_3 = \hat{\mathbf{q}}_k^T \mathbf{e}_3 = 1, \tag{2.21}$$

where $\mathbf{e}_3 = (0, 0, 1)^T$. The auxiliary parameters must be initialized and corrected with the camera parameters during the optimization. This increases the complexity of the optimization procedure, and the time required to reach the optimal camera pose configuration.

We can convert the constrained optimization problem to an unconstrained one. The following closed form expression known as the *normalized crossed epipolar cost* [Sastry, 1999] can be obtained from the cost error in equation (2.20) and the constraints in equation (2.21) using Lagrange multipliers [Sastry, 1999; Vidal et al., 2001]:

$$C_{NCE}(\mathcal{M}, F) = \sum_{k=1}^{n} \frac{\left(\mathbf{q}_k^T F \hat{\mathbf{p}}_k + \hat{\mathbf{q}}_k^T F \mathbf{p}_k\right)^2}{\|[\mathbf{e}_3]_\times F \hat{\mathbf{p}}_k\|^2 + \|\hat{\mathbf{q}}_k^T F [\mathbf{e}_3]_\times^T\|^2} \tag{2.22}$$

To accelerate even more the camera parameter estimation we can optimize the following cost error known as the *Sampson distance* for conic fitting [Bookstein, 1979; Sampson, 1982], instead of optimizing the normalized crossed cost:

$$C_{Sam}(\mathcal{M}, F) = \sum_{k=1}^{n} \frac{\left(\mathbf{q}_k^T F \mathbf{p}_k\right)^2}{\mathbf{p}_k^T F^T Z F \mathbf{p}_k + \mathbf{q}_k^T F Z F^T \mathbf{q}_k} \tag{2.23}$$

where:

$$Z = diag(1, 1, 0) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{2.24}$$

This cost depends only on the camera parameters, and does not involve auxiliary parameters such as Lagrange multipliers or coordinates for estimated image projections. Hence the optimization of this cost is much more efficient than the geometric costs, and obtaining the optimal camera parameters requires less computation time. The cost can be derived from the first degree Taylor approximation of the normalized crossed epipolar cost [Zhang and Shan, 2003]. Hence the solution obtained is a fair approximation, but not exactly equal to the optimal configuration for the geometric cost.

The Sampson distance can be further simplified into the following algebraic epipolar cost:

$$C_{Alg}(\mathcal{M}, F) = \sum_{k=1}^{n} \left(\mathbf{q}_k^T F \mathbf{p}_k\right)^2 \tag{2.25}$$

This is known as the linear algebraic epipolar cost, as the expression is linear on the elements of the fundamental matrix. The evaluation of this cost requires less computation time than either the Sampson error or the geometric epipolar cost. Hence obtaining the optimal configuration requires even less time than with the other costs.

2.3.3 Camera parametrization singularities

Certain singularities can arise and create problems during the optimization of the costs described so far, depending on the rotation parametrization used. The methods described in section 2.1 for minimal parametrization have discontinuities and singularities for certain critical rotation configurations. An example of these configurations is the *gimbal lock* for the Euler parametrization.

A solution is to use a non minimal parametrization such as quaternions [Hamilton, 1844; Wikipedia, 2012*b*] to represent the camera orientation. Quaternions are an extension of complex numbers. Each quaternion **q** is defined by one real and three imaginary coordinates:

$$\mathbf{q} = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k} \tag{2.26}$$

The rotation matrix for a given quaternion **q** can be evaluated with the following expression:

$$R_{\mathbf{q}} = \frac{1}{\|\mathbf{q}\|} Q_{\mathbf{q}} \tag{2.27}$$

where:

$$Q_{\mathbf{q}} = \begin{pmatrix} a^2+b^2-c^2-d^2 & 2bc-2ad & 2bd+2ac \\ 2bc+2ad & a^2-b^2+c^2-d^2 & 2cd-2ab \\ 2bd-2ac & 2cd+2ab & a^2-b^2-c^2+d^2 \end{pmatrix} \tag{2.28}$$

and

$$\|\mathbf{q}\| = \sqrt{a^2+b^2+c^2+d^2} \tag{2.29}$$

In practice, the occurrence of a camera configuration such that $\|\mathbf{q}\| = 0$ during an optimization procedure is unlikely, as this configuration does not correspond to a valid quaternion, nor a valid rotation.

Another solution to prevent the gimbal lock in the Euler angles is to parametrize the rotation as a local perturbation of an existing rotation [Triggs et al., 2000; Engels, Stewénius and Nistér, 2006]:

$$R = R_{\mathbf{w}} R_0 \tag{2.30}$$

The rotation R_0 can be the initial camera orientation for the optimization process, or any fairly good approximation for the true camera orientation. During the error optimization, the local perturbation $R_{\mathbf{w}}$ will commonly be close to the identity, and hence away from problematic configurations.

2.3.4 Algebraic camera initialization methods

Several methods have been proposed to obtain initial camera pose configurations sufficiently close to the optimum, for the optimization of two-view epipolar costs.

These methods first obtain the fundamental matrix for the camera configuration by solving an homogeneous linear system. Each term in the summatory for the algebraic epipolar cost (2.25) can be rewritten with the following equality:

$$\mathbf{q}^T F \mathbf{p} = \mathbf{u}^T \mathfrak{v}_F \tag{2.31}$$

where $\mathfrak{v}_F$ is a vector of size 9 containing the elements of the matrix F in row-major order, and $\mathbf{u}$ is the following vector obtained with the *direct linear transform* (DLT) [Hartley and Zisserman, 2003] from the image coordinates of the points $\mathbf{p}$ and $\mathbf{q}$ in the matching:

$$\mathbf{u} = (q_x p_x, q_x p_y, q_x, q_y p_x, q_y p_y, q_y, p_x, p_y, 1)^T \tag{2.32}$$

Stacking the vectors $\{\mathbf{u}_i\}_{i=1..|\mathcal{M}|}$ for the image matchings in $\mathcal{M}$ we obtain the following measurement matrix of size $|\mathcal{M}| \times 9$:

$$U = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{|\mathcal{M}|})^T \tag{2.33}$$

This way, the expression for the algebraic cost error in (2.25) can be rewritten as follows:

$$C_{Alg}(\mathcal{M}, F) = \|U\mathfrak{v}_F\|_2^2 = \mathfrak{v}_F^T U^T U \mathfrak{v}_F \tag{2.34}$$

An approximation to the fundamental matrix which minimizes this cost can be found by solving the following overdetermined homogeneous linear system:

$$U\mathfrak{v}_F = \mathbf{0} \tag{2.35}$$

In presence of noise, the matrix obtained F^* will not be a valid fundamental matrix, as the solution for the homogeneous equation does not enforce the singularity constraint ($det(F^*) = 0$). The smallest singular value for this matrix can be truncated to zero using a SVD decomposition.

This provides the singular matrix closest to F^* in the Frobenius norm. Using a simple normalization of the input image feature coordinates, the singular matrix will also be a close approximation to the real fundamental matrix which optimizes the algebraic cost [Hartley, 1997].

Once obtained, the fundamental matrix can be factorized into the projection matrices P_1 and P_2 containing the camera information for the two views.

This estimation is up to scale, and it can also suffer other ambiguities which in most cases can be solved by enforcing a correct cheirality on the 3D points in the reconstructed structure [Longuet-Higgins, 1981; Robert and Faugeras, 1995] and the position of the plane at infinity [Pollefeys et al., 1998].

This process is known as the 8-point algorithm for fundamental matrix estimation [Longuet-Higgins, 1981; Tsai and Huang, 1984; Wolfe et al., 1991; Zhang, 1998]. More sophisticated closed form methods have been proposed. For example, the method proposed in [Hartley, 1994*b*] finds the fundamental matrix which satisfies the singularity constraint. This way the procedure does not require the truncation of the smallest singular value. Furthermore, thanks to the extra constraint, this method can find the fundamental matrix using only 7 image point correspondences, instead of the 8 points required by the original procedure.

Several other techniques obtain improved algebraic estimations of the fundamental or essential matrix by adding additional constraints. For example, some procedures obtain the essential matrix in the calibrated case from a set of five image correspondences by enforcing the equality of nonzero singular values in the matrix [Nistér, 2004; Stewénius, Engels and Nistér, 2006; Li and Hartley, 2006].

For reconstructions containing more than two views, similar algebraic procedures can be used to estimate the camera information, such as the trifocal [Torr and Zisserman, 1997] and quadrifocal [Hartley, 1998*a*] tensors for three and four views respectively, or the factorization method [Tomasi and Kanade, 1992; Sturm and Triggs, 1996] for n-view initialization.

These algebraic initialization procedures show important disadvantages to be used in practice. Most of these procedures are limited to use only structure features which are visible in all views. This can be an important limitation in practical reconstruction problems with a large number of views. Due to occlusions and low feature repeatability, it may result difficult to find a sufficiently large number of 3D points appearing in every view.

Furthermore, these techniques are not able to deal with significant noise, or outliers in the input matchings. A small fraction of input mismatchings can corrupt the results obtained with them. Even if the input matchings are free from outliers, due to the algebraic linear nature of projective factorization methods, they can obtain inaccurate results for certain reconstruction problems in presence of a significant measurement noise.

Certain factorization methods were developed which can work with missing data [Hartley and Schaffalitzky, 2003]. However, robustifying these methods to measurement noise and matching failures is still an open problem.

Methods based on cost error optimization such as Bundle Adjustment are used in large scale reconstruction problems, as they can handle matching failures and initialization errors easily by using techniques such as cost robustification.

2.3.5 Structure triangulation

Once the camera information is estimated, the structure parameters can be triangulated. If the camera pose estimations are sufficiently accurate, they can be used with the correspondences between image projections to compute good estimations for the 3D location of the features in the structure.

Triangulation techniques are used to find the 3D location for a point in the scene, given its projections in two or more views with known camera poses. If the calibration for the cameras is known, the triangulated points will be a metric representation of the scene geometry, up to scale.

The most basic and oldest triangulation technique is known as the *midpoint triangulation* [Hartley and Sturm, 1997; Kanatani, Sugaya and Niitsuma, 2008]. This method estimates the 3D point location as the intersection of the two projective rays which connect the camera pose center of each view, with the projections of the 3D point on the respective image planes for the views. Due to image measurement noise, in practice the two rays will not usually intersect. For this reason the point triangulation is estimated as the midpoint of the shortest line segment connecting both rays, i.e. the common perpendicular line.

This method is restricted to the triangulation of 3D points with only two image projections. Furthermore, the reconstruction obtained is neither affine nor projective invariant, since certain properties such as perpendicularity or mid-point distances are not preserved by these transformations.

Another method for structure point estimation is known as the *linear triangulation* (LT) [Hartley, Gupta and Chang, 1992]. This method finds the 3D location for each point by solving a linear system of equations. Given the 2D image projection $\mathbf{p}$ on a view for the 3D point, and the camera projection matrix for that view:

$$P = (\mathbf{P}_1 \mathbf{P}_2 \mathbf{P}_3)^T \tag{2.36}$$

where each column vector $\mathbf{P}_i$ contains the elements for the i-th row in the matrix P, we can rewrite the projective equation in (2.3) as the following pair of homogeneous equations:

$$(p_x \mathbf{P}_3 - \mathbf{P}_1)^T \mathbf{X} = 0 \tag{2.37}$$

$$(p_y \mathbf{P}_3 - \mathbf{P}_2)^T \mathbf{X} = 0 \tag{2.38}$$

Stacking the equations corresponding to the projections from two or more views we obtain an overdetermined homogeneous linear system, which we can solve for the homogeneous coordinates $\mathbf{X} = (x, y, z, k)^T$ of the 3D point. This method is known as the Linear-Eigen triangulation. We can also fix the k coordinate of the 3D point to a constant value, and convert the homogeneous system into an inhomogeneous one. Finding the 3D point coordinates by solving this system is known as the *linear least squares* (Linear-LS) triangulation method.

Both of these linear methods are not projective invariant. The inhomogeneous method has an additional disadvantage. The solution point is assumed to be not on the plane at infinity. In certain circumstances it can be convenient to estimate points with the homogeneous coordinate set to zero. Nevertheless, this method is affine invariant, whereas the homogeneous method is not. If the linear system is solved with an iterative procedure, the linear triangulation is still not projective-invariant, though experiments show that they are quite insensitive to projective transformations [Hartley and Sturm, 1997].

Finally, nonlinear methods triangulate the point by finding the optimal solution for a nonlinear equation. The point can be triangulated by minimizing the L_2 norm of the reprojection error [Hartley and Sturm, 1997; Lindstrom, 2010], which provides the MLE solution. This method improves the results obtained by linear triangulation, and it is invariant to projective deformations. There is a closed form solution for the two-view case, which requires solving a 6-degree polynomial [Hartley and Sturm, 1997]. For the multiple view case, LM can be used to correct the reprojection error for the 3D point under the L_2, providing a highly accurate point triangulation if a sufficiently good initial configuration is available.

The *Second Order Cone Programming* (SOCP) algorithm has been adapted to triangulate points in the structure by minimizing the uniform norm L_∞ [Hartley and Schaffalitzky, 2004]. Like many other SOCP problems, the L_∞ optimization space for the triangulation problem is convex [Hartley and Schaffalitzky, 2004; Kahl, 2005]. Hence there are no local minima, with the exception of the optimal configuration, and the optimization does not depend on the starting configuration to reach the optimal estimations.

However, the solution for the L_∞ norm is highly sensitive to outliers. In a large number of SOCP problems, including the triangulation under uniform norm, at least one of the outliers is always included in the set of measurements with the largest residuals for the optimal L_∞ configuration [Sim and Hartley, 2006*b*]. The initial solution provided by the averaging can be improved by removing these terms, and performing again the correction. After a second SOCP optimization, the solution obtained will usually be more accurate. This process can be iteratively repeated to obtain an outlier free solution.

2.4 Structure from Motion

Structure from Motion (SfM) is known as the process which integrates tools and methods from multiple view geometry, image processing and statistical regression, to obtain the 3D scene reconstruction and camera information from a sequence of input images taken with a moving camera on a static scene.

The basic workflow for a SfM application is the following. First the application detects image features in the input set of images using image processing algorithms. Then these features are matched, using feature descriptors, and sample consensus matching procedures. Finally, multiple view geometry methods such as BA, or pairwise epipolar motion estimation are used to obtain the camera information and the structure from these feature matchings.

The first methods proposed in SfM used the linear algebraic techniques described in sections 2.3.4 and 2.3.5 to obtain an initial estimation for the camera configuration and the structure, which is then corrected using BA, producing the optimal reconstruction configuration.

Due to the limitations of linear methods, this approach only works correctly for small reconstructions, with a limited number of views and structure features. More sophisticated SfM iterative methods were later developed to initialize larger reconstructions, containing hundreds of views and thousands of features in the structure.

2.4.1 Iterative initialization methods

Examples of iterative reconstruction methods methods are the hierarchical initialization [Fitzgibbon and Zisserman, 1998; Nistér, 2000] and the incremental reconstruction [Hartley, 1994*a*; Pollefeys et al., 1998; Brown and Lowe, 2005; Snavely, Seitz and Szeliski, 2006; Klein and Murray, 2007; Snavely, Seitz and Szeliski, 2008*a*].

These iterative procedures use the algebraic initialization techniques described in sections 2.3.4 and 2.3.5 to obtain small partial reconstructions containing reduced groups of views, along with the 3D features which appear on these views.

These partial reconstructions are augmented in each iteration with more views, until no more views or 3D features are left to initialize. At the end of each iteration, the partial reconstructions are refined with BA to eliminate initialization errors, and ensure the convergence of the procedure to the optimal reconstruction.

Once the whole set of views and structure features is initialized, a final global BA provides the optimal solution for the reconstruction problem. The final configuration for the cameras and the structure can then be corrected to metric using either ground truth 3D points [Hartley, Gupta and Chang, 1992] or autocalibration methods [Hartley, 1992; Triggs, 1997; Bougnoux, 1998; Pollefeys, Koch and Gool, 1999].

In [Fitzgibbon and Zisserman, 1998] the authors proposed a hierarchical initialization method, which is able to perform fully automatic robust 3D scene and camera pose recovery from a sequence of images. The proposed procedure begins by estimating optimal partial reconstructions for triplets of views adjacent in the image sequence using the trifocal tensor. These reconstructions are then merged together and *bundelized* iteratively, until a single reconstruction containing all the views and 3D features is left.

Incremental methods are nowadays the most used procedure for reconstruction initialization. These methods start with a partial reconstruction obtained with algebraic initialization methods, which contains a few views, and the structure features which appear on those views. Then the incremental method adds new cameras and features in the 3D structure on each iteration.

The new cameras are initialized using the 3D points in the partial structure with a procedure known as *camera resection*. The new features are initialized in each iteration using these new camera poses. A BA optimization corrects the partial reconstruction at the end of each iteration to prevent divergence.

The first practical implementations of iterative initialization algorithms [Pollefeys et al., 1998] obtained projective reconstructions for the scene, as the cameras were parametrized with projective matrices.

The correction to metric can fail for certain motions and structure configurations. For example, in scenes with dominant planes it might not produce a valid solution [Sturm, 1997; Kahl and Triggs, 1999]. These scenes are common in artificial human-made environments with many planar surfaces such as walls, doors or facades. Some ad-hoc solutions to this problem were proposed, such as using special reconstruction procedures which are robust to planar structure configurations [Pollefeys, Verbiest and Gool, 2002].

Recent works assume either a calibrated camera scenario [Klein and Murray, 2007], or certain reasonable constraints on the camera calibration parameters [Brown and Lowe, 2005; Snavely, Seitz and Szeliski, 2006]. This way the process is more robust, as many critical configurations (such as the aforementioned scenes containing dominant planar surfaces) are no longer a problem to SfM.

Furthermore, the projective-to-metric correction step is no longer necessary, because the reconstruction obtained this way is already metric. Hence the reconstruction obtained is already a faithful representation of the scene structure, up to scale ambiguity.

2.4.2 High performance large scale reconstruction techniques

The actual maturity of SfM methods is high. In recent years many advances have been developed to increase the computational efficiency of BA, which is typically the main computational time bottleneck in SfM applications. This way SfM applications can obtain larger reconstructions while requiring less computational time.

In hierarchical BA [Shum, Zhang and Ke, 1999; Ni, Steedly and Dellaert, 2007] the reconstruction is divided into overlapping areas of fixed size that have their own local coordinate system for the camera parametrization. Depending on the structure and size of the scene, these methods can be more efficient and scalable than a global BA correction.

Several techniques can be used to speed up the Levenberg-Marquardt optimization, such as: *Preconditioned Conjugate Gradient* (PCG) [Agarwal et al., 2010; Byrod and Astrom, 2010; Wu et al., 2011], the Schur complement trick [Engels, Stewénius and Nistér, 2006; Jeong et al., 2011], exploiting the sparsity of the Hessian matrix corresponding to the reprojection error [Engels, Stewénius and Nistér, 2006; Lourakis and Argyros, 2009; Konolige, 2010], and so on.

The speed of the LM optimization can also be increased by reducing the number of views and 3D features which are updated. In [Steedly and Essa, 2001*a*] the authors propose a method to evaluate the contribution of innovative information for each element in the reconstruction problem. This way incremental SfM applications can reduce the number of free parameters in the BA procedure to those essentially affected by new correspondence evidence.

In [Snavely, Seitz and Szeliski, 2008*b*] the authors propose to reduce the reconstruction problem into a skeletal reconstruction problem. This reduces significantly the number of views and features which must be initialized.

Large reconstruction problems initially required clusters of computers to be solved, which in some occasions took several hours to obtain the optimal reconstruction. These procedures were later ported to commodity desktop computers using GPU hardware [Frahm et al., 2010; Wu et al., 2011].

Thanks to these advances, actual SfM techniques can robustly obtain precise city-scale reconstructions from unstructured image datasets containing thousands of pictures [Agarwal et al., 2009; Furukawa et al., 2010], in a completely automatic manner and requiring reasonable computational times.

2.4.3 Methods for real-time SfM reconstruction

At the time MonoSLAM was released, SfM techniques were unable to perform real-time camera pose and structure estimation due to the computational cost of BA. Meanwhile, the quality of the reconstruction obtained with the filtering method was not significantly different from that obtained using batch SfM techniques [Strasdat, Montiel and Davison, 2010*b*; Engels, Stewénius and Nistér, 2006]. This way MonoSLAM became the first system capable of performing visual odometry with commodity hardware.

At that time, BA implementations were not efficient enough to be used in real-time applications for visual odometry. Even with the optimizations described in section 2.4.2, using nowadays BA to correct small reconstructions with a few hundred cameras on average commodity CPU hardware can still take more than a hundred milliseconds, which is prohibitive for any real-time application. The development of real-time SfM applications required explicit computational efficiency enhancements for the BA correction.

These solutions were developed, and keyframe-based SfM techniques became more suitable to compete with feature-based filtering methods for visual odometry estimation. The actual state of the art makes SfM techniques more advantageous than filtering methods to perform real-time camera location and scene mapping [Strasdat, Montiel and Davison, 2010*b*, 2012]. Filtering methods seem to be more adequate for tight computational efficiency budgets, while SfM methods offer a better trade-off between computational efficiency and reconstruction accuracy.

One of the solutions to reduce the computational requirements of SfM for real-time applications is to run the BA correction in a separated thread from the rest of the tasks involved in the visual odometry [Klein and Murray, 2007]. The camera updating, the image feature detection, and the matching for each input frame is performed on the main processing thread, while in the background thread the reconstruction is being corrected with BA. This way the camera pose can be resected using the most recently corrected estimations for the structure parameters, but the tracking process does not require to wait until BA finishes the actual correction to use them.

In practice this approach is suitable to obtain reconstructions of up to a few hundred views and a few thousand structure points, which is enough for medium size indoor scenes or desktop environments. The parallelization is not adequate however for exploratory tasks where the number of keyframes can grow unbounded. In this case the BA correction will eventually fail to keep up with the map correction as it grows indefinitely. New camera poses will be resected with uncorrected 3D features, and the reconstruction process will diverge.

A more scalable solution for real-time reconstruction estimation is to use a variant of BA known as *local bundle adjustment* (LBA) [Nistér, Naroditsky and Bergen, 2004; Mouragnon et al., 2006; Engels, Stewénius and Nistér, 2006; Eudes and Lhuillier, 2009]. In LBA, the LM optimization only adjusts a small fixed amount of views and 3D points from the reconstruction, while the rest of camera poses and features are not changed. This way, a large fraction of the reconstruction parameters and terms in the BA cost error can be ignored during the LM optimization.

To maximize the ratio between error reduction and computational cost, the LM in the local optimization should adjust those parameters that might change in light of new information [Steedly and Essa, 2001*b*; Ranganathan, Kaess and Dellaert, 2007]. That is, LBA should correct the views and 3D points which are prone to contain the largest reprojection errors.

For this reason, a usual practice in LBA is to bundelize the views and 3D features most recently included in the map. This is an adequate procedure to correct build-up errors and prevent divergence, as usually the most recent views in the reconstruction will contain the larger initialization errors.

To ensure convergence during camera tracking processes, it is usually sufficient to optimize the parameters for the 10 most recently added views in the map, and the feature points detected in those views. The camera parameters for older views, as well as the coordinates for 3D points which have no observations on those views can be left fixed in the LM optimization. The local BA correction can be performed this way each time a new keyframe is added to the map, and still satisfy a real-time constant computation time using ordinary commodity hardware [Engels, Stewénius and Nistér, 2006].

Real-time SfM applications based on LBA can produce camera pose estimations with significant dead reckoning error, especially in long-term tracking sessions. These drift errors become apparent when the camera revisits a given area of the scene. Under these conditions, the new location estimations for the features in the structure may not coincide with the location previously estimated for those features.

A solution to correct these discrepancies while satisfying a constant processing time is to use the local correction combined with a relative camera parametrization, and include loop closing correspondences in the LBA correction. This approach is known as *relative bundle adjustment* (RBA) [Holmes et al., 2009; Sibley et al., 2009, 2010*a*; Strasdat et al., 2011].

The reconstruction obtained by BA with the relative parametrization is equivalent to the reconstruction obtained with the regular *absolute reference frame* parametrization [Holmes et al., 2009; Sibley et al., 2010*b*]. The evaluation of the cost error and the Jacobians using the relative parametrization can be more complex than using a classical single reference frame parametrization, but the overall computational complexity of each iteration in the RBA correction is $O(n_v^3)$ for the worst case, being n_v the number of views, which matches the complexity of classical BA [Sibley et al., 2009].

The reconstruction obtained with the classical global BA is intended to be metric, or Euclidean. In contrast, when the relative parametrization is combined with a local correction, BA is intended to obtain reconstructions with a topology corresponding to a connected Riemannian manifold, in the sense that they are locally metric and globally topological.

The local RBA correction can be seen as a continuous submapping approach, which lacks the shortcomings of classical submapping methods such as map overlapping, data duplication, and does not need to merge the submaps into a global euclidean frame.

The reconstructions obtained with the local RBA will have a high accuracy at the local scale, and with the adequate selection of cameras to update, loop closing gaps will be corrected. However the global configuration of the structure and camera poses estimated with RBA can contain arbitrarily large errors. The authors define these Riemannian reconstructions as *topometric*.

These topometric reconstructions can be useful for certain autonomous navigation and robotic applications, where a robot must reach a target location while avoiding physical obstacles. The local metric accuracy of topometric reconstructions ensures that the robot will know the precise size and volume of obstacles in the way, while the topological properties ensure that it will be able to find a valid path to any designated target location. These results are sufficient for many autonomous navigation applications, which do not require a single global Euclidean representation, or precise large-range distance estimation.

2.5 Structureless motion correction methods

Motion correction methods are used in reconstruction applications for several different purposes, such as drift error reduction in presence of loop closing evidence, or direct camera pose initialization without structure computation.

Some structureless solutions take advantage of certain scene features to obtain a direct initialization of the camera poses, without incremental procedures which must estimate the structure. For example, including camera pose information derived from vanishing points [Sinha, Steedly and Szeliski, 2010; Crandall et al., 2011] which can be estimated from straight parallel and perpendicular lines, and are quite common in certain structured artificial scenarios such as cities.

However in more general reconstruction problems it may be difficult to find and use these visual cues to obtain a direct motion initialization.

In this section we will review structureless methods for motion estimation for general scene configuration, i.e. which do not rely on special scene features.

2.5.1 Pose graph optimization

Most visual odometry applications use LBA, which produces a significant drift error in long camera tracking sessions. Pose graph relaxation (or pose graph optimization) [Olson, Leonard and Teller, 2006; Grisetti et al., 2007;

Strasdat, Montiel and Davison, 2010*a*] methods can be used in these applications to correct the drift error using loop closing information.

These methods take the estimated camera motion provided by the visual odometry system, and use the loop closing information to correct drift errors. This is done by optimizing a measurement cost defined on pairwise relative camera motion constraints, obtained from two different sources: the input camera odometry, and visual loop closing information. The following is a general expression for the cost error usually optimized by pose graph correction methods:

$$C_{PG} = \sum_{T_{ij} \in \mathcal{T}} \mathbf{r}(T_{ij}, \hat{T}_i, \hat{T}_j)^T \; \Lambda_{ij} \; \mathbf{r}(T_{ij}, \hat{T}_i, \hat{T}_j) \tag{2.39}$$

In this equation T_{ij} represents a relative motion transformation measured between the camera poses for the views i and j. The set $\mathcal{T}$ contains all the measured motion transformations available for the pose graph correction. The terms $\hat{T}_i$, $\hat{T}_j$ represent the camera poses estimated for the views i and j, w.r.t. a global fixed reference coordinate system. The function $\mathbf{r}$ obtains the residual vector for the discrepancy between the measured and estimated motions. Finally, Λ_{ij} is the inverse of the covariance matrix for each one of the residual vectors in the cost.

We can represent the estimated camera motions $\hat{T}_i$, $\hat{T}_j$ as elements from the group of rigid Euclidean transformations $SE(3)$:

$$T = \begin{pmatrix} R & \mathbf{t} \\ 0 & 1 \end{pmatrix} \tag{2.40}$$

In this case the residual vector $\mathbf{r}(T_{ij}, \hat{T}_i, \hat{T}_j)$ can be evaluated from the composition of the measured and estimated motions as follows:

$$\mathbf{r}(T_{ij}, \hat{T}_i, \hat{T}_j) = log_{SE(3)}(T_{ij} \; \hat{T}_i \; \hat{T}_j^{-1}) \tag{2.41}$$

This is the approach proposed in [Strasdat, Montiel and Davison, 2010*a*]. Another way to evaluate the residual function is to measure the distance between the camera pose parameters directly in the Lie algebra $\mathfrak{se}(3)$ associated to $SE(3)$:

$$\mathbf{r}(T_{ij}, \hat{T}_i, \hat{T}_j) = log_{SE(3)}(T_{ij} \; \hat{T}_i) - log_{SE(3)}(\hat{T}_j) \tag{2.42}$$

This approach is similar to that proposed in [Sünderhauf and Protzel, 2012].

Assuming that the estimation errors for the measured relative camera poses are Gaussian and independent, the optimization of this cost will obtain results equivalent to a delayed-state filtering map adjustment [Lu and Milios, 1997]. The covariance matrices Λ_{ij} in the cost can be coarsely approximated [Strasdat et al., 2011]. We can also assume in most cases with adequate results as well, that the individual measurement parameters are statistically independent, and their uncertainty magnitudes are equal. In this case the covariance matrix Λ_{ij} is exactly the identity [Strasdat, Montiel and Davison, 2010*a*].

The cost (2.39) can be conceived as a graph where nodes represent poses, and the links represent measured relative pairwise motion constraints. Hence the name of pose graph optimization.

Some of the relative constraints in the set $\mathcal{T}$ are obtained from the input odometry estimated by the tracking system. The camera poses provided by the tracking are used to estimate relative pairwise motions between views:

$$\hat{T}_{ij} = \hat{T}_j \, \hat{T}_i^{-1} \tag{2.43}$$

These odometry constraints will constitute the backbone structure of the pose graph. This way the input camera poses will satisfy exactly all the above equations for the initial pose graph.

To correct drift errors, the initial graph is augmented with relative motions estimated from loop closing evidence. These relative motions can be obtained from image point matchings, detected between views previously not related by the real-time SfM camera tracker.

For example, the relative motion can be estimated using the methods described in section 2.3.4 from the epipolar geometry for the matchings. However these methods can be unreliable and produce motion estimations with significant errors. Furthermore, the estimated motion will have an inherent scale ambiguity.

The structure estimated by the real-time SfM tracker can be used to estimate the loop closing motion constraints, by finding the alignment of the 3D points contained in the loop closing views [Strasdat, Montiel and Davison, 2010*a*]. The optimization of the full pose graph containing odometry and loop closing constraints will reduce the drift errors in the estimated motion, and correct loop closing gaps in the reconstruction.

During large camera trackings the drift error can produce significant scale changes in the estimated structure. The relative motion constraints used in the pose graph correction impose a fixed baseline distance between the camera centers, which due to this scale drift can be incorrect.

To deal with this problem, instead of optimizing over rigid euclidean transformations T (elements of the group $SE(3)$) the pose averaging can optimize over elements S of the group of similarity transformations $Sim(3)$

[Strasdat, Montiel and Davison, 2010*a*]:

$$S = \begin{pmatrix} sR & \mathbf{t} \\ 0 & 1 \end{pmatrix} \tag{2.44}$$

This relative parametrization introduces the extra parameter s, representing the scale for each transformation in the cost error of the pose graph correction.

2.5.2 Motion averaging methods

Motion averaging methods can be used to obtain a direct initialization of the reconstruction parameters in SfM applications, without computationally expensive iterative or incremental procedures which must estimate the scene structure.

With accurate camera pose estimations the structure can be obtained using the triangulation methods described in section 2.3.5. This initial reconstruction configuration (averaged camera poses and triangulated structure) should be a good starting point for the BA optimization.

To obtain valid camera pose estimations these averaging methods minimize the residuals for relative motion constraints estimated from pairwise image feature correspondences, in a similar way to pose graph correction. However, the relative motions must be measured from the set of pairwise correspondences detected between the input views.

For example, this can be done solving the epipolar geometry, as described in section 2.3.4. Again, these methods can eventually provide unreliable measurements.

Different proposals for the motion averaging use different norms (such as the mean, median or the uniform norm) in the minimization, thus obtaining varying results. Assuming perfect motion estimations, the results obtained by the averaging are independent of the choice norms, and any of them should provide the optimal configuration. Each norm, however, has a different tolerance against outliers and estimation noise, so in presence of inaccurate relative motions the results obtained with different norms will vary significantly.

Motion averaging under the L_2 norm

In [Govindu, 2001] the author proposes an initialization method to estimate the camera pose orientation R_i and center $\mathbf{C}_i$ for each view, using measured pairwise relative camera motions.

The proposed procedure first estimates the pairwise relative camera orientations R_{ij} and translation directions $\mathbf{t}_{ij}$ from image feature correspondences, using pairwise camera pose initialization methods such as those described in section 2.3.4.

Each relative camera orientation R_{ij} estimated between the i-th and j-th views imposes the following constraint on the relative orientation between those cameras:

$$R_{ij} = R_j R_i^T \tag{2.45}$$

Using a quaternion rotation parametrization, each one of these nonlinear equations can be converted into the following homogeneous linear equation, where $\mathbf{q}_i$ and $\mathbf{q}_j$ are the quaternions corresponding to the absolute rotations R_i and R_j [Horn, 1987]:

$$Q\mathbf{q}_i - \mathbf{q}_j = \mathbf{0} \tag{2.46}$$

The matrix Q in this expression is:

$$Q = \begin{pmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{pmatrix} \tag{2.47}$$

where $(q_0, q_1, q_2, q_3)^T$ are the quaternion coordinates for the relative rotation R_{ij}. Stacking the linear equations for each relative rotation we can propose an homogeneous linear system, which can be solved for the quaternion parameters corresponding to the absolute camera rotations $\{R_i\}_{i=1..n}$.

Once obtained the absolute camera orientations, the translations (or center locations) can be obtained as follows. The absolute and relative camera translations for a view pair should satisfy the following equation:

$$\mathbf{t}_{ij} \propto \mathbf{C}_j - R_{ij}\mathbf{C}_i \tag{2.48}$$

which can be rewritten as the following homogeneous linear equation:

$$[\mathbf{t}_{ij}]_\times (\mathbf{C}_j - R_{ij}\mathbf{C}_i) = \mathbf{0} \tag{2.49}$$

The linear equations for each relative translation can be stacked once again to obtain an homogeneous linear system, which can be solved for the camera centers [Govindu, 2001].

In [Govindu, 2004] the author proposes an iterative correction method which refines simultaneously the camera orientations and translations under a single L_2 norm cost. Each measured relative motion T_{ij} provides the following equation for the correction:

$$T_{ij} = \hat{T}_j \, \hat{T}_i^{-1} \tag{2.50}$$

These equations are mapped to the Lie algebra $\mathfrak{se}(3)$ with the logarithmic operator:

$$log_{SE(3)}(T_{ij}) - log_{SE(3)}(\hat{T}_i) + log_{SE(3)}(\hat{T}_j) = \mathbf{0} \tag{2.51}$$

The optimal solution for the averaging method proposed in this work is equivalent to the least squares solution for these equations. In each iteration the averaging method evaluates the motion residuals in $SE(3)$, and then switches back to the Lie algebra $\mathfrak{se}(3)$ to reduce them, thus correcting the corresponding camera poses.

The results obtained by this method and the pose graph optimization are similar, as both methods obtain the least squares solution for a set of equations derived from relative pose constraints. In most cases the camera parametrization adopted by pose graph optimization techniques is the same one used in this averaging procedure. The only difference is the way in which both methods evaluate the residuals.

In the averaging method the scale ambiguity is solved by fixing the baseline distance for each relative motion T_{ij} in each iteration of the averaging procedure with a good approximation estimated from the linear equations in (2.49).

Motion averaging under the L_∞ norm

The optimization of nontrivial cost errors based on the L_2 has several practical problems to obtain valid camera pose initializations in large reconstruction problems.

These costs usually have many local minima. Hence minimization methods are likely to diverge from the optimal solution if the initial configuration is not close enough to the optimal value. An initial camera pose configuration can be obtained using the algebraic methods described in the previous section, but it may not be in the basin of the optimal configuration for the BA correction.

Furthermore, relative motions estimated with common pairwise initialization methods are prone to contain large estimation errors due to a combination of several factors. The initialization of camera poses from pairwise image correspondences is itself an ill-posed problem. Many pairwise initialization techniques will eventually produce bad relative motion estimations in large reconstruction problems. There are several factors which can contribute to this fact. For example, an insufficient number of feature matchings, the image measurement noise, or an insufficient baseline distance between the cameras. Direct averaging of these relative camera translations can produce incorrect results.

Another source of error in pairwise motion estimations can be due to the visual appearance similarity of certain areas in a scene. Image matching techniques could incorrectly associate features detected at different locations featuring visual resemblance, assuming that they correspond to the same place. This problem is known as *perceptual aliasing.*

In practical reconstruction problems the occurrence of these relative motion outliers grows with the size of the reconstruction problem. For this reason methods for motion averaging under the L_2 norm can fail to provide accurate or even valid camera pose estimations, especially in large reconstructions.

To overcome these problems, in [Sim and Hartley, 2006*a*] the authors proposed a method to average the relative translations under the uniform norm, which formulates the translation averaging problem as a SOCP problem [Hartley and Schaffalitzky, 2004]. This optimization method finds the translation configuration by minimizing the maximum angle between the measured relative translations $\mathbf{t}_{ij}$, and the corresponding expected relative translation $\hat{\mathbf{t}}_{ij} = \hat{\mathbf{C}}_i - \hat{\mathbf{C}}_j$.

Like other SOCP problems, the optimization space in this case is convex. The correction converges to the optimal configuration without requiring a good initial camera translation configuration. Furthermore, like other SOCP problems, an outlier-free solution can be usually obtained by repeating iteratively the SOCP correction after dropping terms with the largest residuals.

Motion averaging under the L_1 norm

In [Hartley, Aftab and Trumpf, 2011] the authors adapted the Weiszfeld algorithm [Weiszfeld and Plastria, 2009] to perform rotation averaging under the geometric median (L_1 norm). This improves the tolerance of the method to outliers, as this algorithm can find in principle the optimal configuration for the median norm of the residuals, with up to a 50% presence of outliers.

One of the problems of this procedure is that, like costs based on the L_2 norm, the median cost has many local minima. Unlike the L_∞ norm, a good initialization is required to reach the optimal configuration in the averaging.

2.5.3 Structureless BA correction methods

Structure-less BA methods, based on epipolar or trifocal constraints, are an alternative to optimization methods based on relative motion constraints. Like relative motion constraints, epipolar and trifocal constraints can be parametrized to depend only on the camera information, without requiring structure parameter estimations (as seen in section 2.3.2). Thanks to this property, motion correction methods based on epipolar or trifocal constraints can be designed to not depend on the structure parameters [Zhang and Shan, 2003; Steffen, Frahm and Förstner, 2010; Rodríguez, López-de-Teruel and Ruiz, 2011*b*]. For this reason, the costs optimized by these methods can have a significantly reduced amount of free parameters compared with the BA cost. Therefore these costs are smoother functions with less local minima, and can be optimized with correction methods more computationally efficient than LM.

The authors in [Vidal et al., 2001] proposed a cost error based on geometric epipolar constraints. Using Lagrange multipliers they derive a closed form expression of this error which can be optimized to correct the camera poses of the views in the reconstruction. However, this optimization still requires the structure, which must be alternatively reestimated after each camera pose correction using the optimal triangulation procedure proposed in [Sastry, 1999].

In [Zhang and Shan, 2003] the authors propose a cost error which simultaneously uses both epipolar and trifocal constraints, to correct the estimated camera poses corresponding to the views in a given image sequence. To speed up the optimization, the cost error is linearized by neglecting the second order elements in the Taylor expansion [Zhang and Shan, 2001] of the geometric cost. This way each view pair in the reconstruction contributes to the cost with a term equivalent to the *Sampson distance*. This cost is parametrized only with the camera parameters, thus it is significantly faster to optimize than the original geometric cost for two-view constraints.

Most costs errors based on epipolar and trifocal constraints (such as the two aforementioned) usually include only terms corresponding to views adjacent in the image sequence. This makes the underlying optimization unsuitable to correct dead-reckoning errors with loop closing information. A solution is to include terms in the error cost for every pair and triplet of views in the reconstruction related by point correspondences. An example of this approach is [Steffen, Frahm and Förstner, 2010], where the authors describe another geometrical cost error which includes a trifocal constraint for each triplet of the views in the reconstruction.

In [Indelman et al., 2012] the authors describe the *incremental light bundle* *adjustment* (iLBA) optimization, a motion correction method which combines a cost based on constraints defined on pairs and triplets of views with the incremental smoothing technique. In this approach not all camera poses are corrected in the optimization, which includes only those constraints which are essential for the error correction. This way the iLBA can outperform the classical BA algorithm in terms of computational complexity, requiring a significantly shorter time to obtain the corrected camera poses while achieving a similar accuracy.

The GEA correction

In [Rodríguez, López-de-Teruel and Ruiz, 2011*b*] we proposed GEA, an efficient motion correction method based on algebraic epipolar constraints. The GEA cost error includes a constraint between each pair of views which are related with point correspondences. Hence, GEA can be applied in batch or real-time SfM applications to correct reconstructions obtained either from unstructured image data-bases, or long term video sequences with significant drift errors.

The algebraic constraints do not depend on the structure parameters, and do not include auxiliary variables. For these reasons, and thanks to the algebraic reduction of the matching information, the GEA cost can be optimized very efficiently. Despite the simplicity of this algebraic epipolar error, with the adequate camera parametrization the error of the camera poses estimated with GEA will be, under general circumstances very close to the optimal reprojection error obtained with BA. Meanwhile, thanks to the structure parameter marginalization and the simplicity of the algebraic constraints, we can outperform the computational efficiency of state of the art BA implementations with GEA, even when all the views and epipolar constraints are included in the optimization. The matching data is reduced in a precomputation step which saves a significant computation time in each optimization iteration. The cost error is corrected with an efficient Gauss-Newton optimization, which can provide a better error reduction speed than LM, and exploits the sparsity of the second level system in a similar way to what most state of the art BA implementations do.

Once GEA has obtained the optimal camera pose configuration, we propose to estimate the structure using the linear triangulation methods described in section 2.3.5. For general motion sequences, the difference between the optimal reprojection error obtained with BA, and the reprojection error for the GEA cameras with the triangulated structure will be very small, in most cases marginal. The GEA algorithm can thus be used in practice to obtain a reconstruction configuration which is at a small correction away from the optimal BA configuration, using only a fraction of the computation time required by BA.

2.6 Closure

In this chapter we have discussed the two main types of techniques used in reconstruction applications: VSLAM techniques, based on filtering methods, and SfM based on error optimization methods such as BA. Both filtering and error optimization methods provide a similar trade-off between the computational time cost required, and the accuracy of the reconstruction obtained. Thanks to recent advances, these methods have become practical and powerful tools to solve problems such as visual reconstruction and robot navigation in real-time and off-line applications.

We have introduced structureless motion correction techniques, which can reduce the computational cost of reconstruction applications in comparison with other correction methods such as BA. In most cases the structure can be efficiently computed from these motion estimations. This way structureless correction methods can be used to reduce the cost of reconstruction applications without sacrificing a significant accuracy on the motion estimations obtained, where using other correction methods which involve the structure parameters would be unpractical due to their computational cost.

Some of these methods require the estimation of relative motions between view pairs. Depending on the method used to estimate these motions, they can contain a significant error [Kahl, 2005] which could degrade the results obtained in the motion correction. Other structureless correction methods such as GEA improve the estimated motions by optimizing a cost defined directly on algebraic epipolar constraints. The computational efficiency and accuracy advantages of this approach are discussed in the following chapters of this document.

Chapter 3

Global epipolar adjustment for motion correction

This chapter provides an in-depth description of the GEA method, along with implementation details to perform a high performance camera correction, and a comparison with other correction techniques used in SfM and VSLAM, such as BA and pose-graph optimization.

3.1 The GEA cost error

We will assume that we have a reconstruction problem with a set of n views, where each pair of views is related by a set of pairwise point correspondences $\mathcal{M}_{i,j} = \{\mathbf{p} \leftrightarrow \mathbf{q}\}$. The GEA cost error is defined for this reconstruction problem as follows:

$$C_{GEA} = \sum_{i=1}^{n-1} \sum_{j=i}^{n} \sum_{\mathbf{p} \leftrightarrow \mathbf{q} \in \mathcal{M}_{i,j}} \left(\mathbf{q}^T E_{ij}^{\dagger} \mathbf{p} \right)^2 \tag{3.1}$$

Matrix $E_{ij}^{\dagger}$ is a normalized version of the essential parametrization for the views i, j defined in equation (2.16):

$$E_{ij}^{\dagger} = \frac{1}{\|\mathbf{t}_j - \mathbf{t}_i\|} R_j \left[\mathbf{t}_j - \mathbf{t}_i\right]_{\times} R_i^T \tag{3.2}$$

The normalization $1/\|\mathbf{t}_j - \mathbf{t}_i\|$ is introduced in the expression to make the evaluated epipolar residuals independent of the motion scale:

$$\frac{1}{\|\mathbf{t}_j - \mathbf{t}_i\|} R_j \left[\mathbf{t}_j - \mathbf{t}_i\right]_{\times} R_i^T = R_j \left[\frac{\mathbf{t}_j - \mathbf{t}_i}{\|\mathbf{t}_j - \mathbf{t}_i\|}\right]_{\times} R_i^T \tag{3.3}$$

This also prevents the convergence of the cost error optimization to incorrect configurations for the camera poses, such as $\mathbf{t}_i = \mathbf{t}_j$ for $i \neq j$ which would produce zero epipolar residuals.

Like other algebraic epipolar costs such as the *Sampson distance* the terms in the GEA cost error are not explicitly parametrized with the 3D structure. The only parameters involved are those corresponding to the camera parameters.

3.2 Efficient GEA cost error optimization

This section details how to obtain an efficient implementation for the GEA correction procedure.

3.2.1 Compact algebraic epipolar cost

The computational time required to evaluate the algebraic cost error in equation (2.34) grows linear with the number of image correspondences $|\mathcal{M}|$. Any iterative optimization method used to minimize C_{Alg} should evaluate several times this cost before reaching the optimal camera configuration. In these evaluations the elements for the $\mathfrak{v}_F$ vector may vary, but the values for the U matrix remain fixed, as the set of feature correspondences $\mathcal{M}$ does not change during the optimization.

With certain precomputations on the coefficient matrix U the evaluation time of cost C_{Alg} becomes constant with the number of matchings, and hence the iterative optimization can be significantly accelerated. One way to do so is to substitute U in expression (2.34) by an alternative reduced matrix $\tilde{U}$ of size 9×9 such that:

$$U^T U = \tilde{U}^T \tilde{U} \tag{3.4}$$

This way the new expression for the algebraic cost will be mathematically equivalent to the original expression:

$$C_{Alg}(\mathcal{M}, F) = \mathfrak{v}_F^T U^T U \mathfrak{v}_F = \mathfrak{v}_F^T \tilde{U}^T \tilde{U} \mathfrak{v}_F \tag{3.5}$$

However, the optimization of the new cost will be significantly faster, as the evaluation time becomes constant.

The reduced matrix $\tilde{U}$ can be precomputed from U using several matrix decompositions before the minimization of the algebraic cost takes place, and be reused in each iteration of the optimization:

- Using the SVD decomposition $U = XDV^T$, the reduced matrix can be computed as $\tilde{U} = DV^T$:

$$U^T U = \left(VDX^T\right)\left(XDV^T\right) = (VD)\left(DV^T\right) = \tilde{U}^T\tilde{U} \tag{3.6}$$

- The reduced matrix can also be computed from the Cholesky decomposition of the original matrix $U^TU = LL^T$ as $\tilde{U} = L^T$:

$$U^T U = LL^T = \tilde{U}^T\tilde{U} \tag{3.7}$$

- Given the eigendecomposition $U^TU = Q\Lambda Q^T$, the reduced matrix can also be obtained from the expression $\tilde{U} = \sqrt{\Lambda}Q^T$:

$$U^T U = Q\Lambda Q^T = \tilde{U}^T\tilde{U} \tag{3.8}$$

- Finally, the QR decomposition $U = QR$ also provides the desired reduced matrix as $\tilde{U} = R$:

$$U^T U = \left(R^T Q^T\right)(QR) = R^T R = \tilde{U}^T\tilde{U} \tag{3.9}$$

Any of these approaches used to speed up the evaluation of the algebraic error by precomputing $\tilde{U}$ would certainly accelerate each step of the cost optimization, but at the expense of evaluating a matrix factorization, which can be computationally expensive. Moreover, some of these factorization methods can introduce numerical errors which may become significant in certain occasions. For example, when U has singular values near or equal to zero, the reduced matrix estimated with the Cholesky decomposition can contain large numerical errors.

An alternative method to speed up the algebraic cost optimization is to use the following substitution in the expression (2.34):

$$U^T U = \Omega \tag{3.10}$$

This way the algebraic epipolar cost C_{Alg} becomes:

$$C_{Alg}(\mathcal{M}, F) = \mathfrak{v}_F^T\,\Omega\,\mathfrak{v}_F \tag{3.11}$$

The matrix Ω is a symmetric matrix of size 9×9 which does not change during the optimization process. For this reason the evaluation time for this version of the algebraic cost requires a constant time, just like the algebraic cost in (3.5).

Obtaining Ω from matrix U does not require a factorization. Still, it can require arbitrarily large memory storage sizes depending on the way the product $U^T U$ is evaluated, as the storage size of matrix U depends on the number of feature correspondences.

An efficient way to estimate Ω from the input matchings $\mathcal{M}$ is to accumulate the element values with the following summatory:

$$\Omega = \sum_{k=1}^{|\mathcal{M}|} \mathbf{u}_k \mathbf{u}_k^T \tag{3.12}$$

In this expression, each $\mathbf{u}_k$ is the column vector defined in equation (2.32) obtained from the coordinates of the k-th feature matching. Many of the values in each term $\mathbf{u}_k \mathbf{u}_k^T$ of this summatory are repeated, due to the fact that the product of several pairwise combinations of different elements from each vector $\mathbf{u}$ will produce the same numerical value. For example, the elements $(1,9)$ and $(3,7)$ in both the matrix $\mathbf{u}_k \mathbf{u}_k^T$ and the accumulated matrix Ω will be equal, given that:

$$u_1 u_9 = u_3 u_7 = q_x p_x \tag{3.13}$$

This is also the case for other pairs of values in these matrices[1]. This way each term in the summatory (3.12) will only have 36 different values. Furthermore, most of the computations required to evaluate terms which are different in these matrices can be factorized. For example, the element $(5,8)$ can be computed by multiplying the element $(8,8)$ by q_y, and the element $(2,5)$ can be computed by multiplying the element $(5,8)$ by q_x:

$$u_5 u_8 = u_8 u_8 q_y = p_y^2 q_y \tag{3.14}$$

$$u_2 u_5 = u_5 u_8 q_x = q_x p_y^2 q_y \tag{3.15}$$

[1] As can be seen in the following expression:

$$\mathbf{u}_k \mathbf{u}_k^T = \begin{pmatrix} {q_x}^2{p_x}^2 & {q_x}^2 p_x p_y & {q_x}^2 p_x & q_x {p_x}^2 q_y & q_x p_x q_y p_y & q_x p_x q_y & q_x {p_x}^2 & q_x p_x p_y & q_x p_x \\ {q_x}^2 p_x p_y & {q_x}^2 {p_y}^2 & {q_x}^2 p_y & q_x p_x q_y p_y & q_x {p_y}^2 q_y & q_x p_y q_y & q_x p_x p_y & q_x {p_y}^2 & q_x p_y \\ {q_x}^2 p_x & {q_x}^2 p_y & {q_x}^2 & q_x p_x q_y & q_x p_y q_y & q_x q_y & q_x p_x & q_x p_y & q_x \\ q_x {p_x}^2 q_y & q_x p_x q_y p_y & q_x p_x q_y & {q_y}^2 {p_x}^2 & {q_y}^2 p_x p_y & {q_y}^2 p_x & q_y {p_x}^2 & q_y p_x p_y & q_y p_x \\ q_x p_x q_y p_y & q_x {p_y}^2 q_y & q_x p_y q_y & {q_y}^2 p_x p_y & {q_y}^2 {p_y}^2 & {q_y}^2 p_y & q_y p_x p_y & q_y {p_y}^2 & q_y p_y \\ q_x p_x q_y & q_x p_y q_y & q_x q_y & {q_y}^2 p_x & {q_y}^2 p_y & {q_y}^2 & q_y p_x & q_y p_y & q_y \\ q_x {p_x}^2 & q_x p_x p_y & q_x p_x & q_y {p_x}^2 & q_y p_x p_y & q_y p_x & {p_x}^2 & p_x p_y & p_x \\ q_x p_x p_y & q_x {p_y}^2 & q_x p_y & q_y p_x p_y & q_y {p_y}^2 & q_y p_y & p_x p_y & {p_y}^2 & p_y \\ q_x p_x & q_x p_y & q_x & q_y p_x & q_y p_y & q_y & p_x & p_y & 1 \end{pmatrix}$$

To speed up the data reduction, we can exploit these redundancies and accumulate the 36 different values for each term $\mathbf{u}_k\mathbf{u}_k^T$ during the summatory evaluation, instead of computing the 81 elements for the full $\mathbf{u}_k\mathbf{u}_k^T$ matrix. Once accumulated, the 36 unique elements can be arranged into a full Ω matrix.

The trick of the reduced matrix was originally suggested in [Hartley, 1998*b*] to speed up not only the pairwise camera pose estimation from the epipolar geometry constraint, but also several other tasks involving the optimization of an algebraic cost, such as trifocal tensor estimation and camera resection. It has been applied to obtain a significant speed up in other problems as well, such as relative motion estimation from point-cloud alignment [Ros et al., 2013]. The method described in this section, which does not require the factorization of the U^TU matrix, could also be applied to the resolution of these problems.

The evaluation time for the algebraic costs in equations (3.5) and (3.11) is similar. However the estimation of the Ω matrix requires a significantly smaller computation time than the evaluation of the reduced matrix $\tilde{U}$, and is also less prone to numerical errors which could degrade the results obtained with the cost optimization.

The results described in this section can be used to rewrite the expression (3.1) for the GEA cost error as the more compact expression:

$$C_{GEA} = \sum_{i=1}^{n-1} \sum_{j=i}^{n} \mathfrak{v}_{E_{ij}^{\dagger}}^{T} \Omega_{ij} \mathfrak{v}_{E_{ij}^{\dagger}} \tag{3.16}$$

In this new cost the term $\mathfrak{v}_{E^{\dagger}}$ is a vector of size 9 containing the elements of the matrix $E^{\dagger}$.

We can drop from this cost those terms corresponding to view pairs which are not related by feature correspondences. These terms should provide a zero contribution to the cost error, as the Ω_{ij} matrix must be zero. With this idea in mind the cost (3.16) can be simplified to our final compact expression for the GEA cost error:

$$C_{GEA} = \sum_{\Omega_{ij} \in \mathcal{O}} \mathfrak{v}_{E_{ij}^{\dagger}}^{T} \Omega_{ij} \mathfrak{v}_{E_{ij}^{\dagger}} \tag{3.17}$$

where $\mathcal{O}$ is the set of Ω_{ij} matrices which were obtained from one or more image correspondences ($|\mathcal{M}_{i,j}| = 0$):

$$\mathcal{O} = \{\Omega_{ij}\}_{|\mathcal{M}_{ij}| \neq 0} \tag{3.18}$$

3.2.2 First and second order optimization methods

To correct the GEA cost error and improve the quality of the estimated camera parameters, we must use a numerical optimization technique. Formally, these techniques find the vector $\mathbf{x}$ that minimizes a given cost error $c(\mathbf{x})$. Most of these techniques are iterative methods which start from a first initial estimation for the vector $\mathbf{x}$, which is updated in each iteration with an increment vector δ, so the cost $c(\mathbf{x}+\delta)$ is smaller than the original value $c(\mathbf{x})$. This process finishes when either the norm of δ, or the cost for the actual solution vector are sufficiently small. The algorithm 1 contains a basic scheme for this kind of numerical methods.

Algorithm 1 General scheme for numerical optimization methods

```
Input:
  $\mathbf{x} \leftarrow$ initial state vector $\mathbf{x}_0$.
  $\mathbf{c} \leftarrow$ cost function.
Method:
  repeat
    $\delta \leftarrow$ increment for the state vector towards the minimum of $c(\mathbf{x})$
    $\mathbf{x} \leftarrow \mathbf{x} + \delta$
  until optimal not reached
```

Numerical optimization methods can be used to solve model data fitting problems such as BA, by minimizing the following cost error derived from the expression in equation (2.14) under the L_2 norm:

$$c(\mathbf{x}) = \frac{1}{2} ||\mathbf{f}(\mathbf{x}) - \mathbf{y}||_2^2 \tag{3.19}$$

In the case of BA, the vector $\mathbf{x}$ is known as the state vector, and will contain the values for the camera and structure parameters. The function $\mathbf{f}$ maps the state vector to the expected image coordinates for the 3D features, provided the camera and structure configuration. The vector $\mathbf{y}$ contains the measured image coordinates for the 3D features. This way the cost evaluated corresponds to the reprojection error of the points in the reconstruction, and its optimization reduces the distance between measured and estimated image projections for the 3D features.

There is a large variety of numerical optimization methods which can be used to correct this cost error. Some optimization methods such as Powell's dog-leg can provide good correction results. However, first and second order methods are the most widely used ones for residual correction under the L_2 norm. Due to its robustness and efficiency, most BA implementations use

LM to correct the reprojection error, which combines the advantages of both first and second order optimization methods.

First-order optimization methods such as gradient descent offer a robust convergence to the optimal configuration, but at the cost of a slow error reduction speed. These methods are based on the fact that any given function decreases in the direction of the negative gradient. The gradient of c at the state vector $\mathbf{x}$ can be evaluated with the expression:

$$\nabla_c = J^T \left(\mathbf{f}(\mathbf{x}) - \mathbf{y}\right) \tag{3.20}$$

where J is the Jacobian matrix of the function $\mathbf{f}$ evaluated at the point $\mathbf{x}$. Hence, the increment vector δ in each iteration of a gradient descent optimization is obtained as follows:

$$\delta = -\alpha J^T \left(\mathbf{f}(\mathbf{x}) - \mathbf{y}\right) \tag{3.21}$$

The value α controls the size of the increment in each step. If the initial state vector $\mathbf{x}$ is inside the basin of the optimal configuration of c, and the α value is sufficiently small, the gradient descent is ensured to reduce the value $c(\mathbf{x})$ in each optimization step, and eventually reach the optimal configuration. However, the convergence to the optimal configuration can be slow, specially when the α value is too small. On the other hand, a value too large will produce large update steps, which can make the gradient descent diverge from the optimal configuration. For these reasons the gradient descent optimization method is rarely used to optimize cost errors in practice.

Second order techniques such as the *Newton* or *Gauss-Newton* methods offer a faster convergence speed. Depending on the nature of the cost error, these methods can have a quadratic rate of convergence.

The cost c can be approximated in the vicinity of a given point $\mathbf{x}$ with the following second-order Taylor series expansion:

$$c(\mathbf{x} + \delta) \simeq c(\mathbf{x}) + \delta^T J^T \left(\mathbf{f}(\mathbf{x}) - \mathbf{y}\right) + \delta^T H \delta \tag{3.22}$$

In this expression H is the Hessian matrix of the cost c evaluated at the point $\mathbf{x}$. This approximation will be accurate as long as c has the shape of a quadratic function near $\mathbf{x}$. If the basin for the optimal of c contains the state vector $\mathbf{x}$ and has a convex shape, the optimal configuration for the Taylor expansion will be a good approximation for the location of the optimal configuration for c.

The Newton method uses this fact to evaluate the increment δ in each optimization step as the vector which optimizes the Taylor expansion in equation (3.22). This vector can be obtained by solving the following linear system:

$$H\delta = -J^T \left(\mathbf{f}(\mathbf{x}) - \mathbf{y}\right) \tag{3.23}$$

The Hessian matrix can be obtained with this expression:

$$H = J^T J + \sum_{i=1}^{n} (\mathbf{f}(\mathbf{x}) - \mathbf{y})_i D^i \tag{3.24}$$

Where each matrix D^i contains the second derivatives of c with respect to $\mathbf{x}$:

$$D^i_{jk} = (\mathbf{f}(\mathbf{x}) - \mathbf{y})_i \frac{\partial^2 (\mathbf{f}(\mathbf{x}) - \mathbf{y})_i}{\partial x_j \partial x_k} \tag{3.25}$$

The *Gauss-Newton* method approximates the Hessian matrix with the following expression:

$$H \simeq J^T J \tag{3.26}$$

which is accurate, as long as the expression $\mathbf{f}(\mathbf{x}) - \mathbf{y}$ for the residual vector is approximately linear, or sufficiently small near $\mathbf{x}$. This way the equation (3.23) becomes:

$$J^T J \delta = -J^T \left(\mathbf{f}(\mathbf{x}) - \mathbf{y}\right) \tag{3.27}$$

The reason why this approximation is more commonly used than the original Newton step is that the second derivatives of certain costs such as the BA or GEA cost errors can be difficult to evaluate, and the Gauss-Newton approximation usually produces an optimization sufficiently fast and accurate.

However, the correction performed by second order techniques can be too aggressive in occasions, especially when the second order Taylor polynomial approximation differs too much from the real cost error. Hence these optimization methods can diverge during the correction of high dimensional and strongly non quadratic cost functions, such as the BA cost error. For this reason, most actual state of the art BA implementations use LM, which combines the fast error reduction speed of second order methods and the smooth convergence properties of first order methods.

LM is basically a Gauss-Newton method modified so that it can behave like a gradient descent when the quadratic error optimization diverges. In LM the increment in the state vector is obtained by solving the following step equation:

$$\left(J^T J + \lambda I\right) \delta = -J^T \left(\mathbf{f}(\mathbf{x}) - \mathbf{y}\right) \tag{3.28}$$

The damping parameter λ controls the convergence speed of the algorithm. When this value is close to zero, the step equation is equivalent to the Gauss-Newton one in (3.27). When the λ value is sufficiently high, the term $J^T J$ has a lesser influence in the solution δ for the LM step equation, and it becomes similar to a gradient descent step equation with a small α value.

During the LM optimization the λ parameter must be tuned dynamically to obtain an optimal convergence speed. When the state vector is updated in each iteration, LM evaluates the cost error. If the cost for the new state vector decreases, the optimization is converging. Hence we must be closer to the optimal value, and the λ parameter can be reduced as well, as the function c will tend to behave more quadratically. If the cost error grows the state vector $\mathbf{x}$ must be restored to the previous configuration, and the λ parameter should be increased to reduce the update step size in future LM iterations.

An outline for the LM optimization method is provided in the algorithm figure 2.

Algorithm 2 Basic Levenberg-Marquardt optimization algorithm.

```
Input:
  $\mathbf{x} \leftarrow$ initial state vector $\mathbf{x}_0$.
  $\mathbf{y} \leftarrow$ target measurements vector.
  $\mathbf{f} \leftarrow$ vector function for expected measurements.
  $\lambda \leftarrow$ initial value for the damping parameter.
Method:
  repeat
    $J \leftarrow$ Jacobian matrix for $\mathbf{f}(\mathbf{x})$
    $\delta \leftarrow$ solve from equation:
            $\left(J^T J + \lambda I\right)\delta = -J^T\left(\mathbf{f}(\mathbf{x}) - \mathbf{y}\right)$
    if $(\|\mathbf{f}(\mathbf{x}) - \mathbf{y}\| > \|\mathbf{f}(\mathbf{x} + \delta) - \mathbf{y}\|)$ then
      $\mathbf{x} \leftarrow \mathbf{x} + \delta$
      decrease $\lambda$
    else
      increase $\lambda$
    end if
  until optimal not reached
```

3.2.3 Numerical optimization of the GEA cost

Like other costs such as the reprojection error optimized in BA, the GEA cost can be efficiently optimized with a second order method. However, the GEA cost has a smaller amount of local minimal configurations due to the smaller number of free parameters. Furthermore, the results described in chapter 5 suggest that the shape of the attraction basin of the optimal configuration for this cost is in general more similar to a quadratic function. For these reasons this cost can be successfully optimized using the Gauss-Newton method, instead of the more complex LM method commonly used to optimize the reprojection error in BA.

Given a reasonably good initial configuration for the camera poses, the Gauss-Newton correction of the GEA cost will converge to the optimal configuration, in most cases. Solving the step equation requires a computational time which, in the worst case, can grow cubic with the number of variables in the cost error. Therefore, it usually has a critical impact in the computational requirements of BA, provided the large number of parameters in the optimization. For this reason, most BA implementations use sparse resolution techniques to solve efficiently the step equation, especially for large reconstruction problems [Konolige, 2010]. However, some problems can appear during the optimization depending on the method used to solve the step equation. For example, in occasions the coefficient matrix for the step equation can be near singular. This could be the case when using a quaternion parametrization for the camera orientation in GEA, instead of a minimal parametrization such as Euler angles or elements from the Lie algebra $\mathfrak{so}(3)$. This is also the case when the scale of the reconstruction is not fixed. Under these circumstances the solution to the step equation is not unique, as a continuum of configurations for the reconstruction parameters will provide the same global optimal error. The following two subsections discuss how to solve this problem, and succesfully solve the Gauss-Newton step equation in GEA using both direct, and iterative solvers.

Solving the step equation with the sparse Cholesky factorization

Many numerical software libraries provide sparse Cholesky factorization methods [Intel, 2012; Chen et al., 2008; Davis, 2012], which can solve the step equation in the LM optimization. However, Cholesky solvers can fail when the step equation is singular. In LM, this is prevented when using a nonzero value in the damping parameter. In a Gauss-Newton optimization, the singularity of the coefficient matrix can also be prevented by adding a small fixed ϵ value to the diagonal elements of the $J^T J$ matrix in the step equation, which becomes:

$$\left(J^T J + \epsilon I\right) \delta = -J^T \left(\mathbf{f}(\mathbf{x}) - \mathbf{y}\right) \tag{3.29}$$

The ϵ parameter is mathematically equivalent to the λ parameter in LM, but has a different purpose. It can be set to a fixed value, and never be changed for the whole optimization. The value should be sufficiently large to prevent the singularity of the coefficient matrix. At the same time, it should be sufficiently small so that the optimization process is not damped. This way, the error reduction speed of the optimization will be optimal.

A general scheme for this modified Gauss-Newton method can be found in the algorithm 3.

Algorithm 3 Modified Gauss-Newton optimization method.

```
Input:
  $\mathbf{x} \leftarrow$ initial state vector $\mathbf{x}_0$.
  $\mathbf{y} \leftarrow$ target measurements vector.
  $\mathbf{f} \leftarrow$ vector function for expected measurements.
  $\epsilon \leftarrow$ fixed small value, zero if using PCG.
Method:
  repeat
    $J \leftarrow$ Jacobian matrix for $\mathbf{f}(\mathbf{x})$
    $\delta \leftarrow$ solve from equation:
          $\left(J^T J + \epsilon I\right) \delta = -J^T \left(\mathbf{f}(\mathbf{x}) - \mathbf{y}\right)$
    $\mathbf{x} \leftarrow \mathbf{x} + \delta$
  until optimal not reached
```

Solving the step equation using iterative sparse solvers

An efficient solution to prevent the problems arising from factorizing singular coefficient matrices when solving the vector δ for the step equation is to use iterative sparse solvers[Agarwal et al., 2010; Jeong et al., 2011], instead of a sparse Cholesky factorization method. Iterative solvers usually obtain the increment step δ by minimizing the following quadratic cost:

$$C_{CG}(\mathbf{x}) = \|A\mathbf{x} - \mathbf{b}\|_2 \tag{3.30}$$

Where A and $\mathbf{b}$ are respectively the coefficient matrix and the right-hand side vector of the step equation. For example, in the case of a Gauss-Newton optimization method, the coefficient matrix A and the objective vector $\mathbf{b}$ for the iterative solver are respectively:

$$A = J^T J \tag{3.31}$$

$$\mathbf{b} = -J^T(\mathbf{f}(\mathbf{x}) - \mathbf{y}) \tag{3.32}$$

In the case of LM, the coefficient matrix should be:

$$A = J^T J + \lambda I \tag{3.33}$$

In most occasions these iterative solvers are more efficient than direct solvers. The *conjugate gradient* (CG)[Shewchuk, 1994] is a well known and highly efficient modification of the gradient descent algorithm. When used to solve linear equations, this method is ensured to obtain the optimal configuration in a finite number of iterations which is bounded by the number of free parameters in the cost error. In fact, the method can converge in a smaller number of iterations, depending on the numeric condition of the coefficient matrix A. This value is given by the ratio between the highest and the smallest eigenvalues in A. The largest this value, the more iterations will CG need to achieve convergence.

The numeric condition for the step equation in optimization problems such as BA will usually be high, and the standard CG will converge slowly. A way to speed up CG is to use a preconditioner for the step equation, which equalizes the eigenvalues of the coefficient matrix, thus reducing the number of iterations required by the optimization. This version of CG is known as the *Preconditioned Conjugate Gradient* (PCG), and is the preferred method for the step equation resolution in most state of the art BA implementations [Agarwal et al., 2010; Byrod and Astrom, 2010; Wu et al., 2011].

Instead of optimizing the cost in (3.30) like CG does, the PCG method optimizes the following preconditioned cost with equivalent results:

$$C_{PCG}(\mathbf{x}) = \|M^{-1}A\mathbf{x} - M^{-1}\mathbf{b}\|_2 \tag{3.34}$$

The preconditioner M is a matrix such that the condition number of $M^{-1}A$ is significantly smaller than the condition number of A alone. Different approximations can be used to obtain a good preconditioner, with varying equalization results for the eigenvalues of A. The common factor of these methods is that M is obtained as an approximation of A such that it is computationally inexpensive to invert.

The simplest method is known as the Jacobi preconditioner. In this method, the preconditioner contains only the diagonal elements of A:

$$M = diag(A) \tag{3.35}$$

A better conditioner for block-sparse matrices is the so-called block Jacobi preconditioner, where M contains the block diagonal elements of matrix of A. Depending on the complexity of the cost error, more sophisticated preconditioners can be used, such as the Generalized Subgraph Preconditioner [Jian, Balcan and Dellaert, 2011].

If an iterative method such as PCG is used to solve the step equation in the GEA optimization, the ϵ value in the modified Gauss-Newton procedure proposed in section 3.2.3 can be set to zero, as the singular condition problem of the Cholesky decomposition does not apply with PCG. In this case the modified Gauss-Newton optimization method will be equivalent to the original Gauss-Newton method, as the step equation (3.29) becomes numerically equivalent to the expression in equation (3.27).

3.2.4 Efficient step equation evaluation

Obtaining and solving the step equation are usually the two main computation time bottlenecks in optimization methods such as Gauss-Newton or LM. Performing these operations efficiently, especially the equation resolution, becomes critical to develop efficient BA or GEA implementations.

Most state of the art BA implementations exploit the sparsity of the cost error to speed up the evaluation of the coefficient matrix, and the right-hand side vector in the step equation [Engels, Stewénius and Nistér, 2006]. In this section we describe an efficient sparse block-oriented procedure to evaluate the matrix $A = J^T J + \epsilon I$ and the vector $\mathbf{b} = -J^T (\mathbf{f}(\mathbf{x}) - \mathbf{y})$ in the Gauss-Newton step equation provided in the formula (3.29).

To describe this procedure, we will assume a camera parametrization of d degrees of freedom, and a reconstruction with n views. The matrix A is configured as a square block matrix containing n^2 blocks of size d^2. The vector $\mathbf{b}$ is also configured as a block vector containing n subvectors of size d. We use the notation $A[i, j]$ for the block starting at the (id)-th row and (jd)-th column in matrix A. In a similar way, the expression $\mathbf{b}[k]$ denotes the subvector of size d which starts at the (kd)-th element of $\mathbf{b}$.

The step equation evaluation starts by initializing the matrix A to ϵI, and the vector $\mathbf{b}$ to zero. Then, each term in the GEA cost error from equation (3.17) is used to update a few specific blocks in A and $\mathbf{b}$, as follows. The procedure evaluates the vector $\mathfrak{v}_{E^\dagger_{ij}}$, and its Jacobian matrices J_i and J_j with respect to the current camera parameter estimations of the i-th and j-th views respectively. The off-diagonal block element $A[i,j]$ is set to $J_i^T \Omega_{ij} J_j$, and the block element $A[j,i]$ is set to the transpose of $A[i,j]$. The diagonal block elements $A[i,i]$ and $A[j,j]$ are respectively incremented with the expressions $J_i^T \Omega_{ij} J_i$ and $J_j^T \Omega_{ij} J_j$. Finally, the block elements $\mathbf{b}[i]$ and $\mathbf{b}[j]$ are respectively with the expressions $J_i^T \Omega_{ij} \mathfrak{v}_{E_{ij}^\dagger}$ and $J_j^T \Omega_{ij} \mathfrak{v}_{E_{ij}^\dagger}$. The outline of this method can be seen in the algorithm 4.

Notice that the sparsity of A decreases with the number of reduced matrices in $\mathcal{O}$. The number of nonzero blocks in that matrix will be at most $n_v + 2|\mathcal{O}|$, where n_v is the number of views in the reconstruction.

Algorithm 4 Efficient step equation evaluation for the GEA cost.

```
Input:
  ${\mathbf{c}_k}_{k=1..n_v} \leftarrow$ parameter configurations for the $n_v$ camera poses.
  $\mathcal{O} \leftarrow$ set of reduced matrices from equation (3.18)
Output:
  $A \leftarrow$ sparse coefficient matrix, initially set to $\epsilon I$.
  $\mathbf{b} \leftarrow$ objective vector, initially set to zero.
Method:
  for all $\Omega_{ij} \in \mathcal{O}$ do
      $J_i \leftarrow$ Derivatives for $\mathfrak{v}_{E_{ij}^\dagger}$ w.r.t. $\mathbf{c}_i$
      $J_j \leftarrow$ Derivatives for $\mathfrak{v}_{E_{ij}^\dagger}$ w.r.t. $\mathbf{c}_j$
      $\mathbf{r} \leftarrow \Omega_{ij} \mathfrak{v}_{E_{ij}^\dagger}$
      $A[i,j] \leftarrow J_i^T \Omega_{ij} J_j$
      $A[i,i] \leftarrow A[i,i] + J_i^T \Omega_{ij} J_i$
      $A[j,j] \leftarrow A[j,j] + J_j^T \Omega_{ij} J_j$
      $A[j,i] \leftarrow A[i,j]^T$
      $\mathbf{b}[i] \leftarrow \mathbf{b}[i] + J_i^T \mathbf{r}$
      $\mathbf{b}[j] \leftarrow \mathbf{b}[j] + J_j^T \mathbf{r}$
  end for
```

3.2.5 Exact cost error sparsification

Ignoring terms in the GEA cost error reduces the computation time required to obtain the Gauss-Newton step equation and increases its sparsity. Given that some of the terms in the cost error can be redundant, we can ignore them during the optimization and still obtain highly accurate camera poses, while reducing the optimization time significantly.

This approach is used in the structureless *iLBA* optimization [Indelman et al., 2012; Indelman, Roberts and Dellaert, 2013], where not all the constraints are included in the optimization to perform an efficient camera pose correction. In [Rodríguez, López-de-Teruel and Ruiz, 2011*a*] we also proposed a simple method to speed up the GEA motion correction during camera motion tracking, by ignoring those terms in the GEA cost which involve cameras outside a sliding window centered on the most recently added keyframe, or do not contribute significantly to loop-closing error correction.

In the present work we propose a simple and efficient method for term reduction based on feature visibility, which can be used to speed up the GEA correction in general unstructured reconstruction problems. This method can be seen as an iterative graph simplification procedure, similar to those proposed in [Jian, Balcan and Dellaert, 2011; Kushal and Agarwal, 2012] for PCG preconditioner estimation. We will use this method in chapter 5 to demonstrate that we can ignore a large number of terms in the GEA correction, without sacrificing error reduction efficiency.

The input of this method is the GEA cost error, presented as a graph where nodes represent views, and links connect view pairs which are constrained by an epipolar constraint in the cost error. After the simplification, the graph will contain the same amount of views, but a significantly smaller number of links. The terms in the GEA cost corresponding to the eliminated links can be considered not essential to the motion correction problem, and be ignored during the GEA optimization without expecting a significant quality degradation of the camera poses obtained.

In a similar way to what the authors propose in [Kushal and Agarwal, 2012] this graph simplification procedure uses the number of image matchings detected between each view pair as an heuristic value for the contribution of each term in the GEA cost error. Hence the graph simplification can be performed before the estimation of the reduced matrices Ω, so not only the GEA error optimization will be faster due to the sparsity increment in the step equation, but also the data reduction.

Each iteration in the simplification procedure deletes the link in the graph connecting the view pair with the smallest number of feature matchings. To preserve the connectivity in the graph, the process will not consider for elimination those links connecting nodes with a valency of s_c edges or less. Doing so might split the graph, or produce a simplified graph with a low connectivity. This link deletion process is repeated while there are links in the graph suitable for deletion, or until a certain fraction of the links s_f from the original graph have already been removed.

Figure 3.1 shows the result obtained with the graph simplification procedure for the pairwise matchings detected between the first 8 keyframes in

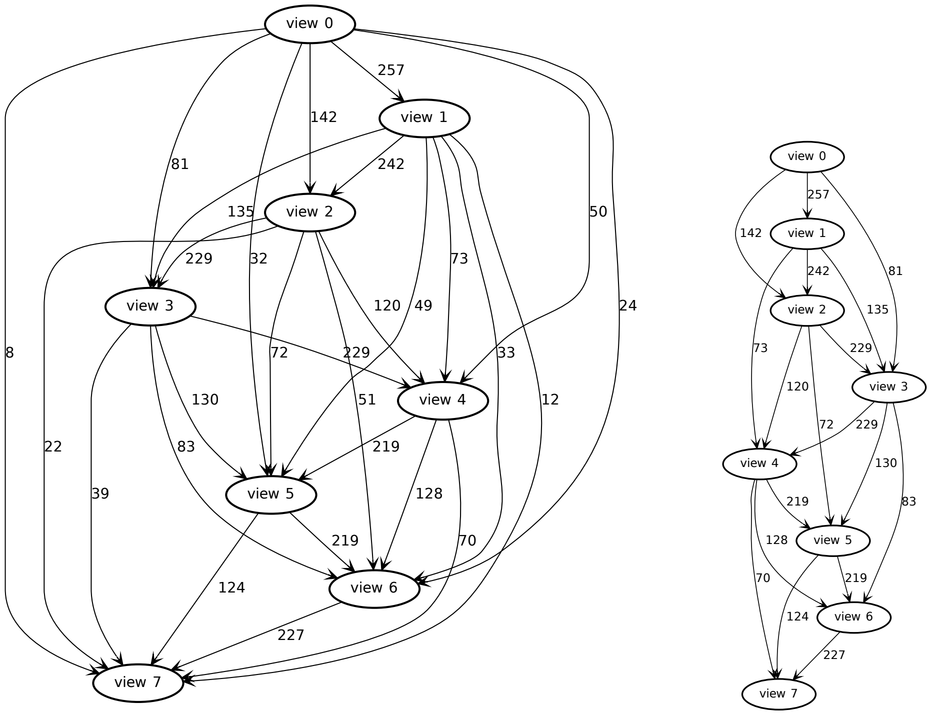


Figure 3.1: Original graph (left) and simplified graph (right) obtained with the described procedure for the correspondences detected between the first frames in the *dinosaur* dataset. Links are tagged with the number of matchings detected between each view pair. The simplification reduces the number of links (or terms in the GEA cost error) from 28 to 18, while keeping at least 3 epipolar constraints involving each one of the views.

the well known *dinosaur* sequence[2]. In this test the s_c and s_f parameters were configured with the values 3 and 0.33 respectively. The quality for the camera poses obtained by optimizing GEA cost errors simplified with this technique is evaluated in detail in section 5.3.2.

The graph simplification technique described is similar in essence to other reconstruction simplification methods, such as skeletal graphs [Snavely, Seitz and Szeliski, 2008*b*], where the overall size of the problem is reduced to a fraction of the original one. The technique proposed in this section is much simpler, but still manages to obtain accurate results, even when the number of links in the view graph is reduced significantly. In the future more sophisticated methods for link selection could be used in the graph simpli-

[2] `http://www.robots.ox.ac.uk/~vgg/data1.html`

fication procedure, such as the accuracy of the pairwise reconstruction for each view pair. This way a larger number of links not essential to the motion estimation problem could be dropped, while obtaining a similar accuracy in the corrected camera poses. The step equation simplification discussed here is also related to the exact sparsification method for delayed-state mapping discussed in section 2.2.3, in the sense that both methods control the exact sparsity level of the step equation (or the information matrix in the delayed-state procedure), and hence the time required to solve it (or the time required for the stochastic map update).

The solutions for the step equation with the simplified cost, and the original GEA cost should be quite similar. For this reason the Hessian matrix of the simplified GEA cost could be used as a preconditioner in the PCG optimization of the full cost, as it is sparser than the Hessian for the original cost, and therefore requires less computational time to be inverted.

3.3 Comparison with bundle adjustment

Under general circumstances, the quality of the optimal camera poses for the GEA cost error will be very similar to the quality of the camera poses obtained with the BA optimization, which requires the structure. Meanwhile, given the same problem and initial starting camera pose and structure configuration, GEA will converge to the optimal camera poses in a fraction of the time required by BA to reach the optimal reconstruction configuration.

In this section we provide theoretical justifications of these facts, and study the conditions where GEA can fail either to compete in computation time with SBA, or to provide accurate results. Later in sections 5.2.1, 5.2.2, and 5.2.3 we will review the results obtained in several evaluation tests which compare the performance, convergence speed and computation time for GEA and BA respectively, to support these facts with practical evidence.

3.3.1 Multiple view *vs* pairwise constraints

BA is basically a statistical method which estimates the solution for the equations (2.11) in presence of noise. Each 3D point $\mathbf{X}$ contributes with the least squares sum of residuals of the following constraints into the BA cost:

$$\left\{\mathbf{p}_i - \phi\left(P_i, \mathbf{X}\right) = \mathbf{0}\right\}_{\mathbf{p}_i \in \mathcal{P}_{\mathbf{X}}} \tag{3.36}$$

where $\mathcal{P}_{\mathbf{X}}$ is the set of image projections detected for the given 3D point. This set of equations is considered a constraint on multiple views, as the joint optimization of the least squares residuals for these equations will enforce simultaneously the camera poses P_i, as well as the coordinates for the 3D point itself.

In a similar way, we could define a multiple view set of constraints based on pairwise epipolar constraints, by combining the geometric epipolar formulations in equations (2.20), (2.21) for different view pairs in a reconstruction. The optimization of these pairwise constraints would be the geometric equivalent for the algebraic GEA cost optimization. In this procedure, each multiple view restriction from equation (3.36) in the BA cost could be converted into the following set of pairwise camera restrictions:

$$\left\{\|\mathbf{p}_i - \hat{\mathbf{p}}_i\|^2 + \|\mathbf{p}_j - \hat{\mathbf{p}}_j\|^2 = 0\right\}_{\mathbf{p}_i, \mathbf{p}_j \in \mathcal{P}_{\mathbf{X}}} \tag{3.37}$$

with the corresponding auxiliary constraints:

$$\hat{\mathbf{p}}_j^T F_{ij} \hat{\mathbf{p}}_i = 0, \qquad \|F_{ij}\| \neq 0, \qquad \hat{\mathbf{p}}_i^T \mathbf{e}_3 = \hat{\mathbf{p}}_j^T \mathbf{e}_3 = 1 \tag{3.38}$$

Where F_{ij} is the fundamental matrix parametrization of the camera poses P_i and P_j, as described in equation (2.17). If we assume a known calibration for this correction procedure, this fundamental matrix can be substituted by the essential matrix parametrization from equation (2.19).

Unless certain critical camera motion and structure configurations appear in the reconstruction problem, the least squares solution for both the multiple view constraints, and the corresponding pairwise epipolar constraints is equivalent [Heyden and Åström, 1997; Ma et al., 2000].

These critical configurations appear for example, when the centers for the camera poses are collinear, or coincident. Another critical setup for two-view constraints which does not affect multiple view constraints is produced when the 3D points in the structure have a planar configuration. In these circumstances the solution obtained from two-view constraints can be incorrect, and multiple view constraints such as trilinear ones should be used to estimate the reconstruction parameters.

Critical configurations affect the optimization of geometric and algebraic costs in a similar way. The critical configurations for the pairwise geometric constraints in equation (3.37) are the same than for the algebraic restrictions optimized in the GEA cost error.

However, the critical configuration corresponding to the planar structure does not affect correction methods which assume a calibrated camera parametrization, which is the case for the GEA method studied in this work.

3.3.2 Accurate solutions for the algebraic cost

As discussed in the previous section, in absence of critical configurations the correction of multiple views using geometric epipolar constraints will provide camera poses with an accuracy equivalent to those obtained with the BA correction. In this section we will justify that the correction of the GEA cost, which is algebraic and not geometric, will provide camera poses with a similar accuracy.

In a broad sense, the solutions obtained for algebraic costs can differ significantly from those obtained for geometric costs. Nevertheless, in [Hartley, 1998*b*] the author demonstrates the remarkable similarity between the solutions for the algebraic and geometric epipolar costs when the camera parametrization meets certain conditions. The first condition is that the fundamental matrix must be singular ($det(F) = 0$). The other is that the skew must be zero. The fundamental and essential matrix parametrization from equations (2.17), (2.19) and (3.2) used in the epipolar costs described so far in this document enforce the first condition, as the matrices obtained are bound to have rank 2. A skew value significantly different from zero indicates certain projective deformations in the image which are rarely found in practice. Furthermore, it is not practical to assume a nonzero skew in reconstruction problems [Pollefeys, Koch and Gool, 1999]. For these reasons assuming a zero skew ($s = 0$) is usually an adequate decision.

To simplify the reconstruction problem and improve the results obtained many authors assume additional constraints on the camera parametrization. Some use a fixed location of the principal point (c_x, c_y) at the center of the image. Though the estimation of the focal length can be influenced significantly by the choice of the principal point [Hartley and Kaucic, 2002] its exact location can be hard to obtain, and has a smaller influence in the reconstruction error than other calibration parameters such as the focal distances, or the skew [Ruiz, López-de-Teruel and García, 2002].

For true pinhole cameras (which is usually the case) it can be assumed that both horizontal and vertical focal distances (f_x, f_y) are equal. Thus the calibration matrix can be reduced to the following expression, which depends only on the focal distance:

$$K_f = \begin{pmatrix} f & 0 & 0.5w \\ 0 & f & 0.5h \\ 0 & 0 & 1 \end{pmatrix} \tag{3.39}$$

where w and h are respectively the width and height of the input image. Nonlinear intrinsic camera parameters such as the radial distortion cannot be modeled in linear equations based on projective matrices. Most modern cameras have a small or negligible radial distortion. In these circumstances forcing the radial distortion to be zero has a small impact on the quality of the reconstruction results obtained. In other problems the camera calibration can be known before the reconstruction estimation.

This proposed camera parametrization has 7 degrees of freedom, (one for the focal distance, three for the orientation and three for the camera center. It is used in several SfM works such as [Agarwal et al., 2009], as it simplifies the computations, prevents critical configurations and provides a reconstruction up to scale.

The camera parametrization used by the GEA correction (from equation (2.19)) assumes that the cameras are fully calibrated, thus the camera parametrization is reduced to a rotation matrix and a translation vector. This way the GEA algebraic cost meets the two conditions required to obtain accurate motion estimations, which are the singularity of the essential matrices, and a zero skew. This way, in absence of collinear camera centers the GEA correction will provide camera poses with an accuracy similar to those obtained by BA. Later in section 5.2.1 we evaluate the performance of both BA and GEA in practice, and provide practical evidence to support these considerations.

3.3.3 Structure of the step equation

The size of the Jacobian matrix in BA is $2n_r m$, where n_r is the number of image projections in the reconstruction, and m is the number of camera and structure parameters in the cost error. In practical reconstruction problems the size of the Jacobian matrix is dominated by the number of terms (or image projections) in the BA cost error (residuals). The number of structure parameters also dominates the size of the coefficient matrix in the step equation, as structure parameters usually outnumber the camera parameters in one order of magnitude, or more.

Figure 3.2 shows the sparsity structure for the Jacobian and Hessian matrices in an example reconstruction problem, which contains only three views and four points in the structure. In this case the BA error has eight terms, one for each image projection detected for the 3D points on the views. All 3D points are visible in every view, except for the third and fourth 3D points which are not visible in the second view, and the two first points, which are not visible in the third view.

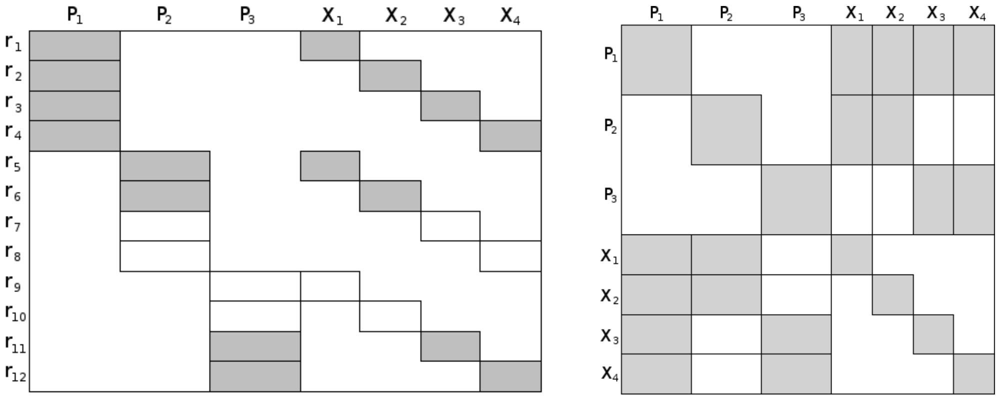


Figure 3.2: Jacobian (left) and Hessian (right) matrix sparsity structure for the reprojection error in an example reconstruction problem. The reprojection error is parametrized with the camera projection matrices $\{\mathbf{P}_i\}_{i=1..3}$ and the coordinates $\{\mathbf{X}_j\}_{j=1..3}$ for the views and 3D points respectively. Grey elements in the figures correspond to nonzero elements in the matrices. Each row corresponds to the partial derivatives of one of the terms in the summatory of the reprojection error, with respect to the camera and structure parameters.

The reduced camera system and the GEA cost Hessian

A common trick to speed up the BA correction is to reduce the LM step equation into an equivalent linear equation defined only on the increments for the camera parameters. This alternative equation system, known as the *camera system*, is smaller and can hence be solved with a significantly reduced computational cost.

To obtain it we can organize the terms in the original equation (3.28) into the following block-structure expression:

$$\begin{pmatrix} H_{CC} & H_{CS} \\ H_{CS}^T & H_{SS} \end{pmatrix} \begin{pmatrix} \delta_C \\ \delta_S \end{pmatrix} = \begin{pmatrix} \mathbf{b}_C \\ \mathbf{b}_S \end{pmatrix} \tag{3.40}$$

where δ_S and δ_C represent respectively the increments for the structure and camera parameters from the state vector, separated in two different vectors.

As can be seen in figure 3.2 the terms H_{CC} and H_{SS} in the Hessian matrix are block diagonal sparse matrices which contain the second grade derivatives of the cost, with respect to the camera and the structure parameters respectively. The term H_{CS} is also a block-sparse matrix which contains the crossed derivatives for the structure and camera parameters.

Using the Schur complement we can clear out the structure increments from equation (3.40), obtaining a linear system which depends only on the increments for the camera parameters [Engels, Stewénius and Nistér, 2006; Jeong et al., 2011]:

$$\left(H_{CC} - H_{SC}^{T}\, H_{SS}^{-1}\, H_{SC}\right) \delta_C = \mathbf{b}_C - H_{SC}^{T}\, H_{SS}^{-1}\, \mathbf{b}_S \tag{3.41}$$

We can solve this equation, known as the reduced camera system [Triggs et al., 2000], instead of the original one in equation (3.40) to find the camera increments δ_C. Then, the structure increments δ_S can be found with the following expression:

$$\delta_S = H_{SS}^{-1}\, \mathbf{b}_S - H_{SS}^{-1}\, H_{SC}\, \delta_C \tag{3.42}$$

The Schur complement trick offers important computational efficiency advantages. The computation of the reduced camera system only requires a few matrix operations, and the inversion of matrix H_{SS}. This matrix is block diagonal and thus easy to invert with a computational cost of $O(n_t)$, being n_t the number of 3D points in the structure. The reduced camera system is significantly smaller than the original step equation, provided that the number of camera parameters is usually one or several orders of magnitude smaller than the number of structure parameters in most reconstruction problems. Hence the resolution of (3.41) requires much less computation time than the resolution of the original step equation in (3.23). For these reasons the Schur complement trick is used in most BA implementations, to obtain the best computation time in the optimization [Triggs et al., 2000; Engels, Stewénius and Nistér, 2006; Konolige, 2010; Agarwal et al., 2010; Jeong et al., 2011].

Figure 3.3 shows the sparsity structure of the reduced camera system obtained with the Schur complement for the example reconstruction problem proposed at the beginning of this section. The blocks in the reduced camera matrix corresponding to the partial derivatives for the views 2 and 3 contain zero values, because no 3D point in the structure of the reconstruction problem has measurements simultaneously for the two views. Hence, these views are not related by pairwise matching information.

Given a reconstruction problem, the reduced camera system in BA will have the same size and sparsity structure than the Hessian matrix of the GEA cost error, as view pairs unrelated by point correspondences will produce null blocks in the Hessian matrix for the GEA correction, or the reduced camera matrix in BA.

3.3.4 Convergence speed comparison

As discussed in section 3.2.3 GEA can perform the error optimization using Gauss-Newton, instead of LM which is used in most cases to correct the

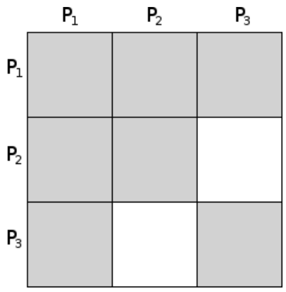


Figure 3.3: Sparsity structure of the reduced camera matrix for the sample reconstruction problem. This matrix is significantly smaller and has a more compact structure than the Hessian matrix in figure 3.2.

reprojection error in BA. The Gauss-Newton optimization of the GEA cost will commonly require less iterations to reach the optimal configuration than the LM correction in BA, as the former is not damped and will not diverge in usual conditions.

The initial value for the λ parameter in LM has a significant influence on the computation time required by the optimization to obtain the optimal configuration. When the λ parameter is too small the LM correction can diverge. In this case BA must restore the changes performed during the last LM iteration, and increase the damping parameter. This way, the computation time invested in this iteration is lost, as the optimization will not progress towards the optimal configuration. When the damping parameter is large, the error reduction produced in each iteration will be smaller. Hence LM will take a larger number of iterations to obtain the optimal configuration.

In chapter 5 we provide experimental results obtained on a large number of reconstruction problems which demonstrate these facts.

3.3.5 Step time cost comparison

Depending on the reconstruction problem configuration, and the method used to solve the step equation, the time required by the GEA correction can be a fraction of the time required by BA. In this section we will evaluate the time cost of each iteration of Gauss-Newton in the GEA correction when used in a reconstruction application, and compare it with the time cost for each LM iteration in BA. For this purpose we can identify the following steps in the optimization algorithm:

1. **Data reduction.** This includes the computation of the reduced matrices Ω from each set of correspondences detected between a view pair.

2. **Gauss-Newton iteration.** The optimization will usually require two or more Gauss-Newton iterations, to ensure convergence to the optimal GEA cost error configuration. Each iteration must evaluate the corresponding step equation, and solve it.

3. **Structure reevaluation.** To obtain a corrected full reconstruction, the points in the structure must be reevaluated after the GEA optimization. This step uses a triangulation method to estimate the 3D coordinates for these points with the corrected cameras.

We will assume a general reconstruction problem containing n_v views, n_t points and n_p projections. The data reduction step will require $O(n_p)$ to evaluate the Ω matrices with the method proposed in section 3.2.1. As the input feature matchings will not change during the camera pose correction, the data reduction can be precomputed before the Gauss-Newton optimization and be reused in each iteration, saving a significant amount of computation time.

In BA, the computational cost for the evaluation of the reduced camera system grows linear $O(n_p)$ with the number of projections, which can take any value between zero and $n_t \times n_v$. Hence the time complexity of the step equation evaluation in BA is $O(n_t n_v)$ for the worst case. In the GEA correction, this time will grow linear with the number of view pairs related to point correspondences in the reconstruction $|\mathcal{O}|$, which is bounded by $O(n_v^2)$. The method for step equation evaluation in the GEA correction described in section 3.2.4 is very efficient, as it directly obtains each block in the Hessian matrix by multiplying the reduced matrices Ω by the derivatives for the camera poses. This way, GEA does not need to reduce the step equation with the Schur complement, as the step equation already depends on the camera parameters only.

As discussed in section 3.3.3, the size and sparsity pattern of the reduced camera system in BA, and the step equation in the GEA correction are exactly the same. Hence, if the SBA implementation uses the Schur complement to speed up the optimization, the time required to solve the step equation in BA and GEA will be similar. The solving time for either GEA, or BA using the Schur trick, grows cubic with the number of views $O(n_v^3)$ in the worst case. This time can be reduced dramatically by exploiting the sparsity on the coefficient matrix [Konolige, 2010], and using iterative solvers such as PCG [Agarwal et al., 2010].

The time for the structure reevaluation in GEA will vary, depending on the trackings length, and the method used to triangulate the 3D points. For certain reconstruction problems, we could assume that the average tracking length n_p/n_t will be constant during the reconstruction process. This way the time for each triangulation becomes also constant, if we also assume a fixed choice of the triangulation method. This approximation is valid for a large range of reconstruction problems, such as real-time motion estimation in exploratory tasks. Under these circumstances, the time cost for the structure reevaluation step will grow linear $O(n_t)$ with the number of points in the structure.

When an iterative solver such as PCG is used to solve the step equation in BA, its evaluation and the structure parameters updating can become the main bottlenecks in the LM step. BA must evaluate the Schur complement of the step equation to obtain the reduced camera system, and also update the structure parameters once the camera parameter increments are found. As can be seen in equation (3.42), the time for the structure updating in the BA correction when it uses the Schur complement has a growth of $O(n_t^2 n_v + n_t n_v^2)$ w.r.t. the number of structure parameters n_t and views n_v, in the worst case.

Respectively, in large reconstructions the structure reevaluation performed by GEA can become the main bottleneck. Nevertheless, most reconstruction applications will not require to re-estimate the whole set of structure points after the camera pose correction. For example, iterative SfM applications will only require the reevaluation of certain 3D points which are required to resect new camera poses. These 3D points can be a fraction of the whole set of points in large reconstruction problems. Moreover, the reconstruction application can be designed to estimate the camera poses without requiring the reevaluation of 3D points in the structure, as discussed in the next chapter.

3.4 Epipolar constraints vs relative motion constraints

GEA is similar to motion correction procedures based on relative motion constraints, such as pose-graph optimization or motion averaging procedures, in the sense that the camera poses are corrected by enforcing pairwise camera constraints. In the case of GEA, each one of these constraints represents the epipolar geometry defined between a pair of views.

This section compares the corrections based on epipolar and motion constraints under different theoretical and practical perspectives.

3.4.1 Scale uncertainty and critical motions

Motion constraints impose a privileged scale in the baseline between the view pair. Hence, relative motion correction methods require special techniques to deal with the scale uncertainty, such as including additional parameters in the cost error which represent the estimated scale [Strasdat, Montiel and Davison, 2010*a*], or using special correction procedures which adjust it dynamically [Govindu, 2004].

This is not a problem in GEA, as epipolar constraints do not impose a privileged baseline distance between the view pair. Hence GEA does not need to introduce extra variables or adapt the cost correction method to allow the scale freedom for each view pair. The basic formulation for the multiple view epipolar error described in section 3.1 is sufficient to obtain an accurate optimization, even in presence of significant scale drift or uncertainty.

It is this scale freedom required by relative motion correction, and imposed by the epipolar constraints geometry which can raise problems with critical motion sequences during the cost error optimization. The previous section discussed how certain linear camera motion configurations can degrade the accuracy of the GEA optimization. Motion averaging and pose graph correction methods also suffer from this problem, when the true camera centers are aligned and the scale for the relative motions is not fixed during the correction. In this case, all the relative camera translations will point towards the same direction, and any possible camera configuration containing the correct camera orientations, and an arbitrary aligned configuration of the camera centers will satisfy the pairwise relative motion constraints [Kaucic, Dano and Hartley, 2001]. Like epipolar constraints, pairwise motion constraints do not encode enough information to solve the relative scale ambiguity in this case. A graphical representation of this problem can be seen in figure 3.4.

To prevent this problem several authors use multiple view constraints, such as the trifocal tensor [Sim and Hartley, 2006*a*; Indelman et al., 2012]. This can complicate the optimization procedure and increase the computational cost. As we will discuss in section 5.3.3, the GEA correction is highly robust against near critical motions anyway. Its accuracy will not degrade significantly as long as the true camera poses are not strictly aligned. A small deviation of the camera motion from the linear trajectory will produce epipolar constraints with enough information to reduce the problems due to the critical configuration, so the pairwise constraint correction can obtain accurate motion estimations.

The problems of critical motion sequences can also be prevented by preceding the reconstruction process by a careful planning of the image acquisition procedure. For example, if the camera is expected to move eventually in a straight line, the reconstruction system could use a stereo camera pair to obtain images with non co-aligned camera pose centers. This can also simplify the feature matching, and increase the robustness of the reconstruction results obtained.

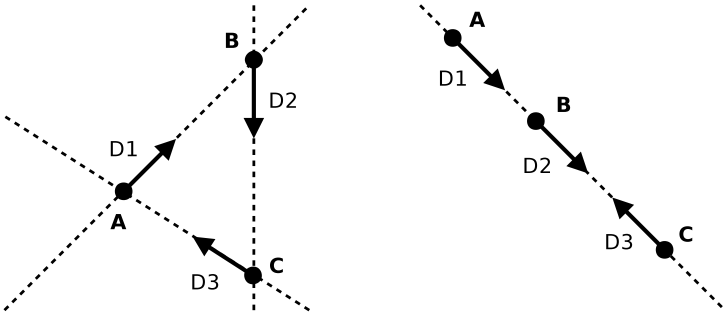


Figure 3.4: Graphical examples of how three relative motion directions D1, D2 and D3 can constrain the camera centers A, B and C. In the left configuration the motions are not parallel. There is only one possible solution for the camera centers configuration, up to similarity transformation. In the right image the directions are equal. Any arrangement of the centers on a line parallel to the motion satisfies the constraints.

3.4.2 Usage in practical reconstruction applications

Classical SfM applications use sample consensus search procedures such as RANSAC, PROSAC, or similar, to estimate valid epipolar models between the view pairs in the reconstruction problem. These procedures provide a set of feature matchings supporting a common epipolar geometry between each view pair. While the GEA correction can work directly on these sets of feature correspondences, motion averaging methods work on relative motions which must be estimated from these matchings. These relative motions are hence an indirect measurement for the true input data for the reconstruction problem, which is the epipolar geometry of the camera poses defined by the matchings.

The accuracy of the results obtained with motion averaging methods is highly sensitive to the precision of the estimated pairwise relative motions. In most cases these relative motions are obtained using classical lineal or geometric methods as those described in sections 2.3.2 and 2.3.4 [Govindu, 2004]. However, these methods introduce to some extent a certain amount of error in the estimated motions, which can be so large under certain circumstances that the averaging will produce incorrect results. More sophisticated methods for motion estimation can be used to estimate better relative motions, and prevent the failure in most occasions [Martinec and Pajdla, 2007]. Certain averaging methods use norms such as L_1 or L_∞ as well to provide an increased level of robustness against these incorrectly estimated relative motions.

Pose-graph optimization methods are usually integrated in a visual odometry application to reduce the drift error in presence of loop closing information. This application can use SfM or VSLAM methods to perform visual odometry, and initialize the backbone of the pose graph with the estimated camera motion. The relative motions obtained this way will contain an arbitrary drift error. Later, the graph can be augmented with relative loop closing motions estimated from the structure, so the pose-graph relaxation will reduce this drift error. Unlike the relative motions estimated from the visual odometry, the epipolar constraints used in GEA are free from loop closing errors.

3.5 Closure

GEA is a motion correction method which optimizes the algebraic residuals for the epipolar constraints to obtain accurate camera pose estimations. This method can be used to perform an efficient incremental motion estimation and drift error reduction without having to compute explicitly the 3D structure, or accurate relative motion between pairs of views.

When compared to other correction techniques such as BA, the cost optimized is simpler and has a smaller amount of locally optimal configurations. This way the optimization can be performed with undamped Newton methods which are more efficient than the classical LM. In the absence of critical configurations, such as rectilinear camera motions, GEA is ensured to obtain camera poses with an accuracy very close to that obtained with BA.

The GEA optimization has the advantage to work directly on the image matching information, unlike other structureless correction methods like pose graph optimization or motion averaging methods, which work on indirect measurements obtained with several different methods from the pairwise matching information. The evaluation of accurate epipolar constraints is simpler and less prone to errors than the estimation of relative motions.

Chapter 4

Using GEA in reconstruction applications

This chapter describes how to exploit the advantages of GEA when used in SfM and VSLAM applications. The first section of this chapter reviews the most common data processing pipelines implemented by reconstruction applications. The second section describes how to use the GEA correction in these pipelines, to reduce the computation requirements of reconstruction applications. Finally, we introduce in the third section a novel structureless incremental motion estimation procedure based on the GEA correction, which requires less computation time than classical incremental SfM applications to obtain accurate reconstructions.

Visual reconstruction methods must prevent the influence of feature mismatchings, which are usually produced by most image feature matching methods. Otherwise the obtained structure and camera poses could contain significant estimation errors. Several methods have been developed for that purpose, such as loss function robustification and sample consensus search. In this last section we introduce a technique based on a loss function to robustify the GEA correction against mismatchings, along with a robust motion averaging under the L_1 norm to initialize the camera pose for a given view. These two methods are used in the incremental motion estimation procedure to obtain robust estimations for the camera poses.

4.1 Structure of SfM reconstruction applications

The first step in most reconstruction applications is to detect features, such as edges [Canny, 1986], corners [Shi and Tomasi, 1994; Rosten and Drummond, 2006], blobs [Lindeberg, 1993; Lowe, 2004], or contours [Matas et al., 2002; Nistér and Stewénius, 2008] on the input images. These features are then matched in an attempt to find pairs of image features detected at different images which correspond to the same 3D feature in the physical scene. For this purpose, the image features are associated with numeric descriptors such as SIFT [Lowe, 2004], SURF [Bay, Tuytelaars and Gool, 2006] or binary descriptors [Calonder et al., 2010], which summarize the visual appearance information at the image feature location. By comparing the descriptors of two features we can establish whether they are likely to correspond to the same 3D physical feature or not. Once the feature matchings are obtained, we can use multiple view geometry techniques to estimate the camera poses and the structure for the scene.

Some of the matched features will not correspond to the same 3D feature from the scene. These mismatchings can degrade the quality of the camera poses and the structure estimated by the reconstruction process, and must be detected and eliminated to obtain valid results. Some reconstruction applications filter out most of the mismatchings from the correspondences detected at the input images using sample consensus search methods such as RANSAC [Fischler and Bolles, 1981] or PROSAC [Chum and Matas, 2005]. This way the quality and robustness of the reconstruction will be improved.

The next subsection will provide a brief overview on how do sample consensus search methods work, and the way they are used in reconstruction applications to eliminate outlier feature matchings. The following two subsections describe procedures used to obtain the camera poses and the structure from the pairwise feature matchings. The last subsection describes how these procedures are adapted to perform motion and structure computation in real-time applications.

4.1.1 Epipolar consistency for pairwise mismatching filtering

Sample consensus search methods can be used to estimate a fundamental matrix F which encodes the true camera poses for a given view pair, given the set of correspondences $\mathcal{M} = \{\mathbf{p} \leftrightarrow \mathbf{q}\}$ detected between these views.

These methods use the procedure described in section 2.3.4 to estimate fundamental matrices from sets of 7 or 8 matchings randomly selected from $\mathcal{M}$, until one of these estimated fundamental matrices is supported by a sufficiently large number of the matchings in $\mathcal{M}$, or a large number of fundamental matrices has been estimated. A feature matching $\mathbf{p} \leftrightarrow \mathbf{q}$ is said to support a given fundamental matrix F when the epipolar error for the matching and the matrix F is smaller than a given threshold value α_R. This epipolar error could be evaluated with any of the epipolar cost errors defined in equations (2.22), (2.23) or (2.25). For example:

$$\mathbf{p}F\mathbf{q} \leq \alpha_R \tag{4.1}$$

The outline for this procedure can be seen in the algorithm 5. In this case, this fundamental matrix is returned by the sample consensus procedure as the estimated epipolar geometry for the camera poses of the given view pair.

Algorithm 5 General RANSAC method for fundamental matrix estimation

```
Input:
  $\mathcal{M} = \{\mathbf{p} \leftrightarrow \mathbf{q}\}$ ← set of feature correspondences.
  $\alpha_R$ ← epipolar error threshold.
  $n_{min}$ ← minimum number of inliers for a valid model.
  $n_{mod}$ ← maximum number of estimated models.
  $C$ ← an epipolar cost error.
Method:
  $i \leftarrow 0$.
  repeat
    $\mathcal{M}'$ ← set of 7/8 matchings randomly selected from $\mathcal{M}$.
    $F$ ← fundamental matrix obtained from $\mathcal{M}'$.
    $n_{inl}$ ← number of matchings in $\mathcal{M}$ which satisfy $(C(\mathbf{p} \leftrightarrow \mathbf{q}, F) \leq \alpha_R)$.
    if $(n_{inl} \geq n_{min})$ then
      return F
    end if
    $i \leftarrow i + 1$
  until $(i > n_{mod})$
  return failure
```

The set of matchings (or inlier matchings) which support the fundamental matrix estimated with this method will contain a larger fraction of valid correspondences than the initial set of feature correspondences detected between the image pair. For this reason, the feature matchings not contained in this set can be discarded as mismatchings. However, the set of inlier matchings can still contain a few mismatchings, which support the fundamental matrix with a small epipolar residual, but do not correspond image projections for the same scene point.

4.1.2 Incremental reconstruction estimation

In an incremental reconstruction procedure the pairwise feature matchings detected at the input views are used to estimate the multiple view geometry of the reconstruction. The process starts with the bootstrapping, which estimates a small partial reconstruction containing a few camera poses, along with the 3D points which are visible in those views. These initial camera poses can be obtained from the input pairwise matchings using the methods described in section 2.3.4. Once the camera poses are known, the location for the points which appear on the initialized views can be estimated using triangulation methods, as described in section 2.3.5.

This partial reconstruction is iteratively augmented with new camera poses and structure points. Each iteration estimates the motion for views not yet included in the partial reconstruction, using correspondences between 3D points in the structure and image features detected at the views, which can be derived from the pairwise matchings.

This is usually known as camera resection. There are many methods which can be used to resect new camera poses during the incremental initialization. The simplest one obtains the camera poses from the projective matrix, which is solved from a homogeneous linear system under the L_2 norm [Hartley and Zisserman, 2003]. As any other linear method, this resection procedure provides acceptable results only when the input set of correspondences is free from outliers, hence it should be used in combination with sample consensus search. More recent proposals use SOCP to estimate the camera pose under the L_∞ norm, with the advantages of cost error convexity and iterative outlier rejection based on the largest residual values [Li, 2007]. When the input views are captured with the same physical camera and fixed calibration, we can as well estimate efficiently the camera orientation and location from the $3D \mapsto 2D$ correspondences with *n-point camera pose determination methods* (PnP) methods [Quan and Lan, 1999; Lepetit and Fua, 2005; Lepetit, Moreno-Noguer and Fua, 2009].

Once new camera poses are estimated, the incremental procedure adds new 3D points to the structure, composing the pairwise matchings into trackings, and using them with the camera poses resected so far to triangulate their location. The correct initialization of new camera poses and points during the incremental procedure depends on the accuracy for the partial reconstruction estimated so far. For this purpose the next step in the incremental process is to correct the initialization errors produced during the estimation of new camera poses and 3D points. This is usually done in classical SfM applications using BA. For this purpose, the partial reconstruction estimated so far is corrected using BA. This way the camera poses and points in the

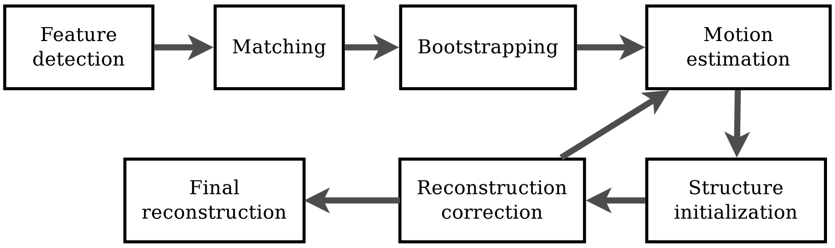


Figure 4.1: Basic scheme for the data-flow path in the pipeline of batch reconstruction applications.

structure can be used in future iterations to successfully resect new camera poses and triangulate new points. This process is repeated until no more views or 3D points can be initialized.

Figure 4.1 shows the scheme for a reconstruction pipeline which uses the incremental motion initialization procedure.

Robustification to outliers

Due to the presence of mismatchings in the tracking composition, the BA optimization can diverge from the valid reconstruction configuration as the L_2 norm is highly sensible to outliers.

A possible solution is to assume a non-Gaussian distribution for the reprojection residuals, such as the Cauchy distribution [Triggs et al., 2000; Engels, Stewénius and Nistér, 2006]. In this case, the BA cost from equation (2.15) becomes:

$$C^*_{RE} = \sum_{\mathbf{p}_{ij} \in \mathcal{P}} log \left(1 + \frac{\|\mathbf{p}_{ij} - \phi(P_i, \mathbf{X}_j)\|^2}{\sigma^2} \right) \tag{4.2}$$

This way the influence of outliers in the BA correction is reduced, and the optimization can provide accurate estimations for the reconstruction parameters, even in presence of a reasonable fraction of mismatchings.

The exact distribution assumed in practice for the reprojection residuals is less important for the robustification than the general way in which the outliers are penalized in the cost error. Some BA implementations use loss functions, such as the Tukey's biweight function [Klein and Murray, 2007] or the Huber loss function [Huber, 1964; Sibley et al., 2009] which reduce the influence of outliers with large reprojection errors in the BA correction.

Once the initialization errors in the estimated camera poses and the structure are eliminated by a robustified cost error optimization, the outliers become evident given their large reprojection residuals. This way, they can be detected and discarded during the iterative reconstruction process.

4.1.3 Direct camera pose estimation with motion averaging

In [Martinec and Pajdla, 2007] the authors describe a structureless camera pose initialization procedure. The matchings are used to estimate pairwise relative motions which are used in both a linear algebraic averaging method for rotation estimation, and the SOCP method for L_∞ translation estimation. To estimate these relative motions accurately, MSER regions, SIFT features and affine interest points [Mikolajczyk et al., 2005] are detected on the input images to obtain the largest number of pairwise feature correspondences. The procedure also uses for this purpose a robust mismatching detection method which detects between each pair of images the four correspondences most likely to be valid, from the initial set of matchings.

For certain datasets, this method provides camera poses with a high quality, which can be used to estimate a sparse structure with a small reprojection error. In most occasions this reconstruction can be corrected with BA to obtain the optimal reconstruction configuration. However, the relative motions can occasionally be incorrect due for example to perceptual aliasing, and hence the camera poses obtained and the reconstruction estimated. In these occasions BA might not be able to recover the optimal reconstruction from the motion initialization errors.

Like other results obtained with SOCP, the camera poses can be iteratively refined by discarding the measurements with the largest residuals and repeating the averaging, until accurate estimations for the camera poses are obtained. The convergence of this process to an accurate solution is not ensured, however. Not all the outliers can be discarded by the iterative rejection in some reconstruction problems, and the final reconstruction obtained can have a significant error. The reconstruction pipeline for this motion estimation procedure can be seen in figure 4.2.

4.2 GEA in classical incremental SfM applications

In this section we suggest several ways to use the GEA optimization in the SfM applications described in the previous section, as well as in on-line reconstruction pipelines.

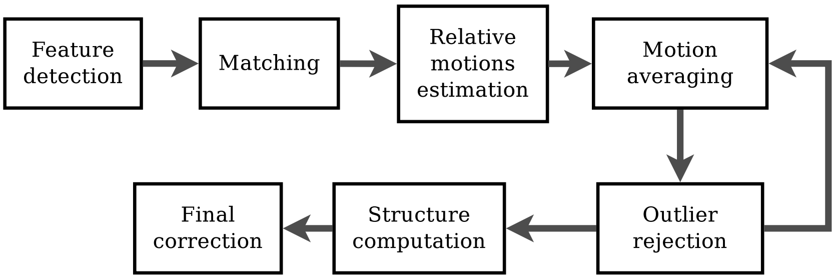


Figure 4.2: Reconstruction pipeline as suggested in [Martinec and Pajdla, 2007]. The motion is estimated using an averaging method, and the solution obtained is refined iteratively by rejecting relative motion outliers given their large residuals. Once the views are initialized, the structure is computed and the whole reconstruction is refined using BA.

4.2.1 In combination with SBA

The number of iterations required by SBA to correct the initialization error in a partial reconstruction estimated during an incremental SfM process will be larger when the error of the partial reconstruction is high. A way to reduce the computation time of SBA is to reduce the number of these iterations, starting the optimization from a configuration with less reprojection error [Rodríguez, López-de-Teruel and Ruiz, 2011*b*].

GEA can be used to correct an important amount of this initialization error. The structure points can then be re-estimated efficiently using the linear triangulation procedure with the camera poses corrected by GEA. SBA will usually require less LM iterations to polish the resulting reconstruction.

This approach will be adequate when the trade-off between the computational cost of the SBA iterations saved, and the estimation of the GEA camera poses with the linear structure computation pays. This will usually be the case when the camera poses have significant errors, due for example to incorrect initializations or loop closure gaps.

As the experiments described in chapter 5 show, GEA will usually reach the optimal camera pose configuration in a few iterations. Each one of these iterations uses a fraction of the time required by each iteration of BA. Obtaining the linear triangulation of the whole structure will usually require less computation time than performing a single BA iteration, in most reconstruction problems.

4.2.2 Substituting SBA in the correction stage

GEA could also be used to correct the initialization errors in the partial reconstructions without the final BA polishing. The main purpose of the correction step is not to obtain the optimal partial reconstruction, but to reduce the initialization errors of the partial reconstruction. This is required in order to use the camera poses and the structure in future iterations of the incremental SfM to successfully estimate new camera poses and points. This can be achieved by using GEA in combination with the linear triangulation.

The reprojection error of the corrected reconstruction will be suboptimal, but in practice it will be sufficiently close to the optimal SBA reconstruction configuration to be used in the initialization of new views and 3D points.

The time required for the SBA correction will usually be much larger than the combined time for the GEA correction and the linear 3D point triangulation. For this reason the computational efficiency improvement obtained by substituting the BA optimization by the GEA correction combined with the linear triangulation can be significant.

4.2.3 Correcting drift error in visual odometry applications

GEA can be used to correct the camera poses estimated by real-time SfM applications without requiring the correction of the structure parameters, in a similar way to pose graph optimization [Rodríguez, López-de-Teruel and Ruiz, 2011*a*].

After each GEA correction we can look for tentative loop closing keyframe pairs, which are likely to contain image projections for the same scene features. This can be done efficiently using full image descriptors. The *sum of absolute differences* (SAD) [Wong, Vassiliadis and Cotofana, 2002; Watman et al., 2004] and the *sum of squares distances* (SSD) [Klein and Murray, 2007] are two correlation measurements that have been used in many works to evaluate the visual similarity between two image blocks. We can efficiently perform full image matching between a given input frame, and a large number of the frames previously processed by subsampling them to a fraction of the original size, and using the SAD or SSD correlations between these subsampled frames as an accurate measurement for their visual similarity [Castle, Klein and Murray, 2008].

When a small SSD or SAD correlation value is detected between a given pair of frames which still do not have image feature matchings in common, we can proceed to detect feature matchings between them. Once we have a set of loop closing correspondences we can evaluate the reduced Ω matrix for the view pair, and include it in following GEA corrections to reduce the drift error between the matched frames.

The GEA correction can be performed in a parallel thread separated from the camera tracking process, much in the style of the reconstruction system proposed in [Klein and Murray, 2007]. GEA can also be applied locally on a fixed number of the most recent keyframes, to limit the maximum computation time required by each step of the Gauss-Newton correction. In this case the correction can be performed each time a new keyframe is added to the reconstruction, in the style of a local BA optimization.

In both approaches, compared with SBA, we can include in the GEA correction a larger number of free views while requiring the same computation time in the Newton step, as the experiments discussed in chapter 5 demonstrate. Hence GEA can be used to increase the scalability and robustness of on-line reconstruction applications, and increase the accuracy of the estimated motion without sacrificing real-time constraints.

4.3 Structureless incremental motion estimation with GEA

In this section we propose a novel structureless incremental motion estimation procedure which can be used to efficiently initialize the camera poses in SfM applications. It can be integrated in the classical SfM reconstruction pipeline, to obtain an accurate camera pose initialization before computing the structure. By doing so, this procedure can improve significantly the computational efficiency of classical incremental SfM methods, by providing an accurate initialization for the camera poses and the 3D structure without performing the BA correction in the intermediate steps of the process.

The procedure uses the correspondences detected by classical feature matching algorithms such as SIFT to estimate relative motions and epipolar constraints between the camera poses for the input views. We assume that these matchings have been filtered with the sample consensus search procedure described in section 4.1.1.

In a similar way to incremental SfM, the motion estimation starts with a small set of initialized cameras, which is augmented with new camera poses iteratively. The camera pose for each new view is averaged using the relative motions and the epipolar constraints. When no more camera poses can be estimated, they are added to the set of initialized cameras, which is then corrected with the GEA optimization.

With this procedure there is no need to compose the image feature trackings, or estimate the 3D locations for the corresponding points. The BA correction is neither required to prevent divergence, as GEA ensures that the initialization errors of the new camera poses are corrected in each iteration. This process is repeated until no more views can be added to the set of initialized cameras.

The set of feature matchings detected between the input views will contain a small fraction of outliers, due to matching errors. To perform an accurate motion initialization, the procedure proposed is robust against these outliers. The averaging method estimates new camera poses under the L_1 norm. This way the camera poses are accurately obtained, even when a half of the relative orientations and epipolar geometries were incorrectly estimated. Once the camera pose for an uninitialized view is obtained, we can evaluate the epipolar residuals for the matchings obtained with other views already initialized. If a sufficiently high number of correspondences is found to have a small residual for the new camera pose, the view is initialized with the estimated camera pose. Otherwise the camera pose is rejected. This way we prevent camera poses with large initialization errors to be included in the final solution.

The motion initialization also uses a robustified GEA correction which can successfully reduce initialization errors of the camera poses in presence of feature mismatchings.

Figure 4.3 shows the pipeline for a SfM application which uses the incremental motion estimation procedure proposed, to obtain a full scene reconstruction. The **feature detection** and **matching** stages are similar to the ones with the same names in incremental reconstruction pipelines. The **motion bootstrapping** stage is also similar to the bootstrapping stage in the incremental pipeline. In this case though, the structure is not computed, and the initial camera poses configuration are corrected using GEA. The **motion estimation** stage adds new camera poses to the set of initialized views using averaging. The **motion correction** stage corrects initialization errors of the camera poses with a robustified GEA correction. This way the corrected camera poses can be used in the following iterations of the motion estimation to initialize new camera poses. These two last stages are iteratively repeated until no more camera poses can be initialized. When this is not possible, the **structure computation** stage composes the trackings and obtains the linear structure using the camera poses. BA could be used optionally in the **final correction** to obtain the optimal reconstruction.

In the next subsection we characterize the types of mismatchings which can be produced by the sample consensus matching method described in section 4.1.1. This will be useful to describe the averaging method used to estimate new camera pose, and the robustified version of the GEA cost, in

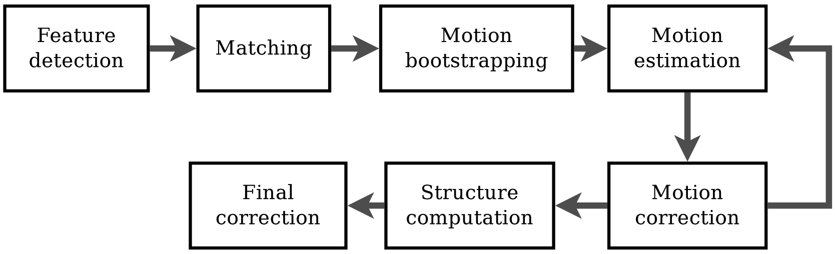


Figure 4.3: Reconstruction pipeline for structureless motion estimation. Each iteration initializes camera poses for new views with a robust averaging method, which uses relative constraints between views already initialized, and uninitialized ones. Outliers can be rejected at the motion correction step.

the second and third subsections respectively. In section 5.4 of the following chapter we evaluate the performance of this pipeline on several scenes and image datasets.

4.3.1 Sample consensus mismatching characterization

The feature correspondences provided by the sample consensus search can contain two kinds of outliers, which must be dealt with in different ways. In the first kind we have matchings satisfying the valid epipolar geometry defined for the true camera poses. In this case the sample consensus search has estimated the correct epipolar geometry and the fundamental (or essential) matrix encodes the true camera poses for the view pair. Most of the matchings supporting this epipolar geometry will be valid correspondences. Some of them will be invalid matchings, which happen to provide a small epipolar residual for the fundamental matrix.

On the other hand, the sample consensus search can fail to estimate a valid epipolar model. In this case, the matchings obtained can still have a small epipolar residual for a given epipolar geometry which does not correspond to the true camera pose configuration. That is, this matrix does not encode correctly the relative motion between these camera poses. The set of matchings hence obtained will contain mostly outliers, which will have a large epipolar residual error for the real fundamental matrix encoding the true camera poses for the view pair.

Figure 4.4 shows examples of a correct, and incorrect sample consensus search matching results. Figure 4.5 shows the epipolar lines for the incorrect matchings, obtained in presence of perceptual aliasing.

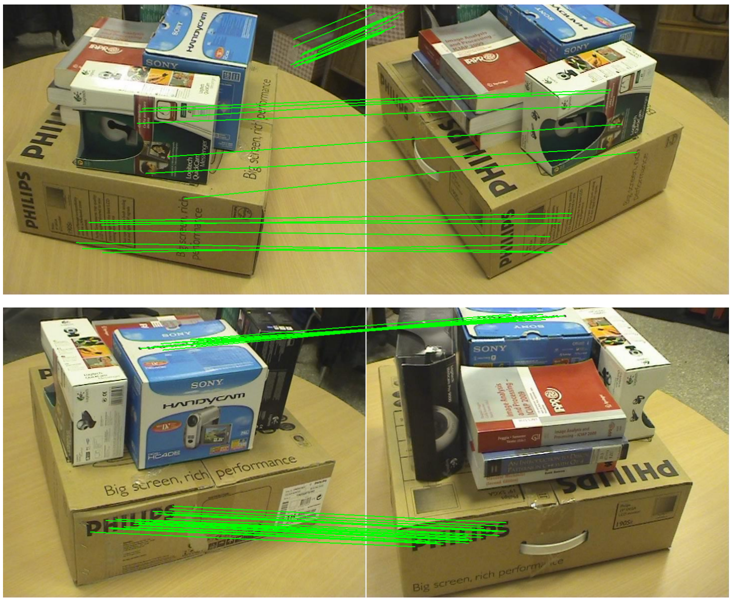


Figure 4.4: Matchings detected between two image pairs using a sample consensus search. The epipolar geometries were correctly estimated (upper row) and incorrectly estimated due to perceptual aliasing (lower row).

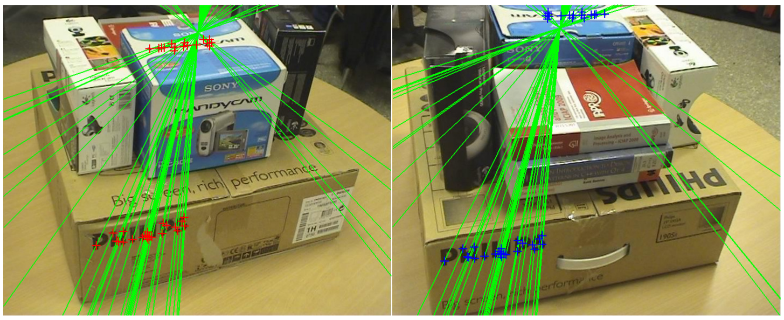


Figure 4.5: Epipolar lines for the mismatchings shown in figure 4.4 which were obtained with an incorrect epipolar geometry. In this case the image features (red and blue crosses) are sufficiently close to the epipolar lines (green lines) obtained with an incorrect epipolar geometry. This misleads the sample consensus search into considering the mismatchings and the epipolar geometry as valid.

4.3.2 Motion averaging for camera pose initialization

This section describes a structureless robust camera initialization procedure, which can be used to initialize the camera pose for an input view I_k, given a set of other views $\{I_i\}_{i=1..n}$ with known camera orientations $\{R_i\}_{i=1..n}$ and translations $\{\mathbf{C}_i\}_{i=1..n}$. The method requires as well the image feature correspondences $\mathcal{M}_{ik}$ detected between the view I_k and the views in $\{I_i\}$. Unlike the camera resection procedure described in section 4.1.2, this initialization procedure does not require 3D points or tracking information to estimate the camera pose for I_k.

The first step of the structureless camera initialization method is to estimate the orientation R_k for the k-th camera pose. To do so, we estimate relative rotations R_{ik} between the view I_k and each view I_i from the image correspondences $\mathcal{M}_{ij}$ using algebraic epipolar initialization methods as discussed in section 2.3.4. Assuming an error free estimation for both the relative rotation R_{ik}, and the absolute rotation R_i, the unknown absolute rotation R_k must satisfy the following equation:

$$R_k = R_{ik} R_i \tag{4.3}$$

Using this expression, we can obtain an estimation R_k^i for the absolute camera orientation R_k from each one of the relative rotations R_{ik} and the corresponding absolute camera orientation R_i. Averaging these rotation estimations we can obtain a single more accurate estimation for R_k^*.

Once obtained the orientation R_k^*, the problem of camera translation and structure computation can be reduced to solving a linear system of equations [Rother and Carlsson, 2001]. In this case we will only compute the camera translation $\mathbf{C}_k$ for the view I_k, which can also be done by solving a linear problem. As described in section 2.3.4, the algebraic epipolar cost defined between the views i and k can be expressed with the following compact equation derived from (2.35):

$$U_{ik} \mathfrak{v}_{E_{ik}} = \mathbf{0} \tag{4.4}$$

In this equation the coefficient matrix U_{ik} depends only on the coordinates for the point correspondences. The vector $\mathfrak{v}_{E_{ik}}$ contains the elements for the essential matrix, and depends on the camera orientations R_i, R_k^* and locations $\mathbf{C_i}$, $\mathbf{C_k}$ for the view pair. This equation can be rewritten as:

$$U_{ik} \rho_{ik} \left(\mathbf{C_i} - \mathbf{C_k} \right) = \mathbf{0} \tag{4.5}$$

Where ρ_{ik} is a 9×3 matrix, whose coefficients depend only on the components of the relative orientation between the views I_k and I_i, which is defined as follows:

$$\rho_{ik} = \begin{pmatrix} [\mathbf{r}_1]_\times \\ [\mathbf{r}_2]_\times \\ [\mathbf{r}_3]_\times \end{pmatrix} \tag{4.6}$$

where $\mathbf{r}_j$ is the j-th row vector of matrix $R_k^* R_i^T$. Given the estimations for the orientations R_k and R_i, and the translation $\mathbf{C}_i$, the expression (4.5) becomes the following inhomogeneous equation:

$$U_{ik}\rho_{ik}\mathbf{C}_k = U_{ik}\rho_{ik}\mathbf{C}_i \tag{4.7}$$

In this equation the matrix $U_{ik}\rho_{ik}$ on the right hand side and the vector $U_{ik}\rho_{ik}\mathbf{C}_i$ on the left side are known, while the translation $\mathbf{C}_k$ remains unknown. Stacking this equation for two or more relative orientations we obtain a linear system which we can solve for the camera pose location $\mathbf{C}_k$ of the view I_k.

This motion estimation procedure is robust against those feature mismatchings which support the epipolar geometry for the true camera pose configuration. These mismatchings will not degrade the quality of either the relative rotations estimated with algebraic epipolar initialization methods, or the camera centers estimated from the linear equations, as both are based on pairwise epipolar constraints. However, if the sample consensus search obtains mismatchings which satisfy an incorrect fundamental matrix between the view I_k and one or several views in $\{I_i\}$, the orientation and camera center estimated from them can be incorrect.

We can prevent this by estimating the rotation and translation for the view I_k under the L_1 norm. The method proposed in [Hartley, Aftab and Trumpf, 2011] based in the Weiszfeld algorithm can be used to obtain the geometric median of the rotations R_k^i in the Lie group $SO(3)$. This way we obtain an accurate estimation for the camera orientation, even if half of the relative rotations were incorrectly estimated.

We can also adapt the Weiszfeld algorithm to solve the algebraic equations for the translation under the L_1 norm as follows. The procedure starts with an initial solution $\mathbf{x}_0$ obtained by solving the set of equations as usual, under the L_2 norm:

$$\begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{pmatrix} \mathbf{x}_0 = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_n \end{pmatrix} \tag{4.8}$$

In this equation each matrix A_l and the vector $\mathbf{b}_l$ denote respectively the coefficient matrix and the objective vector for the linear system in 4.7 corresponding to a pair wise camera pose restriction. In subsequent iterations of the procedure, each equation is averaged with the inverse for its residual, to penalize terms with large residuals. The solution vector $\mathbf{x}_{t+1}$ is updated in each iteration by solving the following linear system:

$$\begin{pmatrix} \lambda_1 A_1 \\ \lambda_2 A_2 \\ \vdots \\ \lambda_n A_n \end{pmatrix} \mathbf{x}_{t+1} = \begin{pmatrix} \lambda_1 \mathbf{b}_1 \\ \lambda_2 \mathbf{b}_2 \\ \vdots \\ \lambda_n \mathbf{b}_n \end{pmatrix} \tag{4.9}$$

where each factor λ_i is obtained with the expression:

$$\lambda_i = \left[\| A_i \mathbf{x}_t - \mathbf{b}_i \| \sum_{l=1}^{n} \frac{1}{\| A_l \mathbf{x}_t - \mathbf{b}_l \|} \right]^{-1} \tag{4.10}$$

This version of the Weiszfeld averaging method will usually provide a robust estimation for the camera pose translation in a few iterations, as long as 50% or more of the linear equations in (4.7) were obtained from matchings satisfying the correct epipolar geometry.

4.3.3 Robustified GEA correction

The GEA correction described in chapter 3 is robust against mismatchings which satisfy the epipolar geometry defined for the true camera pose configuration. These mismatchings, which can be harmful for the accuracy of other correction methods such as BA, do not degrade the quality of the results obtained by GEA. Each one of these mismatchings $\mathbf{p} \leftrightarrow \mathbf{q}$ will have a small epipolar residual with the essential matrix parametrization for the true camera pose configuration:

$$\mathbf{q} E_{ij}^{\dagger} \mathbf{p} \simeq 0 \tag{4.11}$$

Nevertheless, the influence of mismatchings obtained with incorrect epipolar geometries must be prevented from the GEA correction, as these mismatchings will not have small epipolar residuals for the true camera poses. We can detect the terms in the GEA cost which contain mismatchings that do not support the valid epipolar geometry, and reduce their influence on the results obtained by the correction. The expression for the GEA cost error in (3.17) can be rewritten for this purpose to:

$$C_{GEA} = \sum_{\Omega_{ij} \in \mathcal{O}} \Phi_{|\mathcal{M}_{i,j}|} \left(\mathfrak{v}^T_{E^\dagger_{ij}} \Omega_{ij} \mathfrak{v}_{E^\dagger_{ij}} \right) \tag{4.12}$$

where $\Phi_n(r)$ is a loss function which should reduce the contribution of terms which are likely to contain mismatchings obtained from incorrectly estimated epipolar geometries.

To design a loss function suitable for the GEA correction we can use the average residual contribution for a term in the GEA cost, which is evaluated with the expression:

$$|\mathcal{M}_{i,j}|^{-1} \mathfrak{v}^T_{E^\dagger_{ij}} \Omega_{ij} \mathfrak{v}_{E^\dagger_{ij}} \tag{4.13}$$

Where $|\mathcal{M}_{i,j}|$ is the number of matchings used to obtain the reduced matrix Ω_{ij} for the GEA cost term. The average residual contribution in the GEA cost error will usually be larger for terms containing mismatchings which support invalid epipolar geometries, in comparison with the rest of terms estimated from matchings which support valid epipolar geometries. With this idea in mind, a possible choice for this function could be the following:

$$\Phi_n(r) = \begin{cases} |r| & |\frac{r}{n}| < \mu \\ 0 & |\frac{r}{n}| \geq \mu \end{cases} \tag{4.14}$$

In this function r is the epipolar residual for the GEA term, and n is the number of correspondences used to estimate the Ω matrix. Figure 4.6 shows the plot for this function. By using this function in the cost (4.12), those terms with an average residual (as defined by expression (4.13)) larger than the threshold value μ will have a zero contribution to the GEA cost error. Meanwhile, the contribution of terms with an average absolute residual smaller than μ to the GEA cost will be the same than their contribution for the original GEA cost defined in equation (3.17).

Despite being discontinuous, the proposed loss function can be easily incorporated in the GEA correction by modifying the Gauss-Newton optimization procedure, without changing the derivatives of the original GEA cost error.

To implement this idea in the GEA correction method described in chapter 3, we only have to modify the step equation evaluation procedure described in section 3.2.4. Before updating the Hessian matrix and the objective vector with the derivatives estimated for a given term in the GEA cost error, we can evaluate the average residual with the expression (4.13).

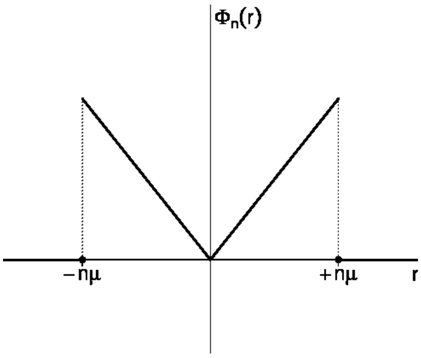


Figure 4.6: Plot for the loss function proposed.

If this value is larger than μ, we can ignore the term and proceed to update the estimated step equation with the following term in the GEA cost. This way we drop those GEA terms from the step equation evaluation which are likely to correspond to invalid epipolar geometries. The schema for this robustified algorithm for step equation evaluation can be seen in the algorithm 6.

Due to discontinuities in the proposed loss function, this method can fail to obtain the exact step equation for the robustified GEA cost. When $r = \pm n\mu$ the derivative for the loss function is undefined. However, in this case the modified Gauss-Newton procedure will evaluate the derivative for the loss function as zero. Despite this difference, the Gauss-Newton optimization will still work correctly in practice, and it will provide an accurate approximation to the robustified GEA cost error optimum, as the experiments described in the following chapter show.

4.4 Closure

In this chapter we have shown how to use GEA in classical SfM applications.

Techniques such as sample consensus search or cost error robustification play a capital role in reconstruction applications to reduce the influence of outliers, as classical image matching procedures eventually obtain incorrect feature correspondences due to noise and perceptual aliasing. In this chapter we reviewed how the GEA correction can be robustified, so it can be used in practical reconstruction applications where these outliers can appear. The inlier matchings obtained in SfM applications with sample consensus methods will usually have a small residual for the epipolar geometry corresponding to the valid camera pose configuration. Hence, these inliers can be used safely in the GEA correction, even if they contain mismatchings. When the

Algorithm 6 Robustified step equation evaluation for the GEA cost.

```
Input:
  $\{\mathbf{c}_k | k = 1..n_v\} \leftarrow$ parameter configuration for the $n_v$ camera poses.
  $\mathcal{O} \leftarrow$ set of reduced matrices from equation (3.18)
Output:
  $A \leftarrow$ sparse coefficient matrix, initially set to zero.
  $\mathbf{b} \leftarrow$ objective vector, initially set to zero.
Method:
  for all $\Omega_{ij} \in \mathcal{O}$ do
    $J_i \leftarrow$ Derivatives for $\mathfrak{v}_{E^\dagger_{ij}}$ w.r.t. $\mathbf{c}_i$
    $J_j \leftarrow$ Derivatives for $\mathfrak{v}_{E^\dagger_{ij}}$ w.r.t. $\mathbf{c}_j$
    $\mathbf{r} \leftarrow \Omega_{ij}\mathfrak{v}_{E^\dagger_{ij}}$
    if $(\mathfrak{v}^T_{E^\dagger_{ij}}\mathbf{r} < |\mathcal{M}_{i,j}|\mu)$ then
      $A[i,j] \leftarrow J_i^T \Omega_{ij} J_j$
      $A[i,i] \leftarrow A[i,i] + J_i^T \Omega_{ij} J_i$
      $A[j,j] \leftarrow A[j,j] + J_j^T \Omega_{ij} J_j$
      $A[j,i] \leftarrow A[i,j]^T$
      $\mathbf{b}[i] \leftarrow \mathbf{b}[i] + J_i^T \mathbf{r}$
      $\mathbf{b}[j] \leftarrow \mathbf{b}[j] + J_j^T \mathbf{r}$
    end if
  end for
```

sample consensus fails, it obtains an invalid epipolar geometry and a large fraction of the inlier matchings provided will be incorrect. In this case, the corresponding term in the GEA correction can be detected by measuring the average residual with the initial camera poses. This fact can be used to robustify the GEA optimization against the influence of epipolar constraint outliers which could degrade the accuracy of the camera poses obtained.

In this chapter we have also shown how to obtain a robust motion estimation free from the influence of image feature mismatchings with a completely structureless incremental camera pose initialization procedure, which uses the robustified GEA to correct the initialization errors in each iteration and prevents divergence from the optimal camera configuration.

Chapter 5

Performance evaluation for GEA

This chapter describes the results obtained in several performance tests which evidence the advantages of using GEA in SfM and SLAM applications. It is divided in four sections. In the first section we define the methodology used to evaluate the accuracy and the computational cost of GEA. In the second section, we use this methodology to compare the speed and accuracy of both the GEA correction, and a state of the art SBA implementation. In the third section we test the failure conditions of the GEA algorithm by evaluating the robustness of the correction against outliers, critical motion sequences and cost sparsification. Finally, the last section shows several results of the structureless incremental motion estimation procedure described in section 4.3. We demonstrate the accuracy of the camera poses obtained by using them to estimate both sparse and dense reconstructions.

5.1 Methodology for the GEA performance evaluation

In this chapter we will use several datasets to evaluate the GEA performance. These datasets can contain a set of images captured with one or several cameras, feature trackings detected in them, the calibration of the cameras, as well as an initial configuration for the camera poses and the structure obtained using SfM or SLAM techniques. This initial reconstruction configuration has in occasions a significant reprojection error, which we can reduce using different correction methods to compare their performance, and determine which one is best for the given reconstruction problem.

5.1.1 Measuring the GEA correction accuracy

The accuracy of structureless motion correction methods is often compared with a ground truth configuration for the camera poses. For example, in [Strasdat, Montiel and Davison, 2010*a*; Sünderhauf et al., 2012] the accuracy of pose-graph optimization methods is measured with the *root-mean-square error* (RMSE) discrepancy between the camera poses corrected, against the ground truth camera pose configuration.

To measure the performance of GEA and compare it with the performance of SBA, it could be desirable to evaluate not only the quality for the camera poses obtained, but also the accuracy of the scene structure which can be estimated from these cameras, i.e. using a given triangulation method. In this work we will use the reprojection error measured for both, the camera poses corrected with GEA, and the structure triangulated using them, as a measure of the quality for the camera poses. If the triangulation method used to estimate the structure is sufficiently accurate, this reprojection error can be considered a good measurement for the GEA accuracy.

We evaluate the reprojection error using the following equalization for the expression in equation (2.15):

$$C^*_{RE} = 1000\sqrt{\frac{1}{2|\mathcal{P}|}C_{RE}} \tag{5.1}$$

Provided that we work with calibrated cameras, this reprojection error is measured directly on image plane coordinates. In this expression the error is divided by the size of the residuals vector, which is two times the number of image projections $|\mathcal{P}|$ in the reconstruction. This way the expression evaluates the standard deviation of the measurement error in the image features. We can use this value to compare the error obtained in different reconstruction problems with a varying number of image feature projections. The factor 1000 scales the value obtained for readability purposes.

In principle we can use for the structure triangulation any of the methods described in section 2.3.5. Unless stated otherwise, in this work we will use for this purpose the *Linear-LS* triangulation method, as it offers a good balance between accuracy and computational cost, provided that it is fast and also highly reliable for most reconstruction problems.

Due to the appearance of outliers and certain ill-posed problem configurations, this triangulation method can provide incorrect estimations for certain points which will degrade significantly the average reprojection error, even if the estimations for the cameras and a large fraction of the points in the structure are accurate. Furthermore, most SBA implementations are robustified against these outliers and will have little influence during the correction. For these reasons, it is likely that these points will have large residuals for the reconstructions obtained by either SBA, or the GEA with the Linear-LT structure.

In practice, these outliers would be discarded during the reconstruction process. As long as the remaining structure obtained contains a sufficiently large number of 3D points, the reconstruction can be considered successful. Hence the residual contribution of these outlier points to the reprojection error will not represent the quality of the estimated camera poses and the remaining correct points in the structure.

In this work we ignore a small fraction of the points with the largest residual contribution during the evaluation of the average reprojection error. This way the reprojection error is a faithful and valid measurement of the camera poses quality. Unless stated otherwise, the evaluation of the reprojection error in the tests described in this chapter does not include the 1% of the points with the largest average residual contribution.

5.1.2 SBA and GEA implementations

To compare the performance of the GEA and BA algorithms we evaluate the speed and accuracy of one implementation for each algorithm on different reconstruction problems. The GEA implementation is based on the code provided by the open source QVision library[1] [Rodríguez et al., 2008]. This implementation uses the $\mathfrak{so}(3)$ parametrization to represent the orientation of each camera pose (as described in section 2.3.3) and a 3D point for the camera center. The cameras are assumed to be calibrated, so this parametrization does not include intrinsic parameters such as the focal distance, principal point coordinates, or radial distortion coefficients.

During the step equation evaluation, the derivatives J_i, J_j for each term in the GEA cost error are efficiently evaluated from the camera poses and the reduced matrix Ω using code automatically generated with a symbolic algebra package, such as Maple [Monagan et al., 2005] or Mathematica [Wikipedia, 2012*a*]. To obtain the solution for this equation, this GEA implementation provides different dense and sparse solvers, such as the sparse Cholesky factorization routines in the MKL [Intel, 2012] or CHOLMOD [Chen et al., 2008] libraries.

The performance of this GEA implementation is tested against *sparse*

[1] `http://qvision.sourceforge.net/`

sparse bundle adjustment (sSBA)[2] [Konolige, 2010], which is a state of the art implementation of BA included in the ROS library [Quigley et al., 2009]. There are several other open source SBA implementations which could be used to compare the performance of GEA with BA [Lourakis and Argyros, 2009; Kummerle et al., 2011; Wu et al., 2011]. However sSBA combines several advantages which make it adequate for the performance comparison with GEA. Like the GEA implementation sSBA assumes a calibrated scenario, hence the parametrization for the cameras only includes the orientation and the camera center. Both implementations were coded using design choices with comparable computational efficiency. They use object oriented data containers which can increase significantly the computational time cost. Neither of them uses high performance parallel computing techniques, GPU hardware or multimedia extensions to accelerate the optimization, with the exception of the step equation resolution which is done using third-party libraries. Nevertheless, if the same sparse solver implementation were used for both correction methods, and the Schur complement is used in SBA, the time required for this operation should be the same. Hence the times evaluated for this operation in our tests should not be considered in the computational efficiency comparison of both GEA and SBA. The implementation is also highly accurate, and like GEA, it is freely available and open source.

In our tests, both implementations were configured to solve the step equation using a block Jacobi preconditioned CG. This method requires a maximum of $n_v \times n_p$ iterations, where n_v is the number of views in the reconstruction and n_p is the number of parameters for each camera. In practice, PCG usually requires a fraction of those iterations to obtain an accurate approximation for the solution. We configured sSBA and GEA to perform a minimum of 10 iterations, and a maximum of $n_v/4$, obtaining this way solutions for the step equations which were quite similar to those obtained by direct solvers.

Unless stated otherwise, both GEA and SBA are configured to perform 10 iterations in the optimization of the cost error with the modified Gauss-Newton and Levenberg-Marquardt respectively. The initial damping parameter for LM is set to $\lambda = 10^{-3}$, while this parameter is fixed in the Gauss-Newton optimization used in GEA to $\lambda = 10^{-3}$. This parameter configuration was adequate to reach the optimal configuration for most datasets using both correction methods.

[2] The adjective *sparse* is repeated in the denomination of this implementation to indicate that it exploits the sparsity on both the evaluation of the reprojection error derivatives, and the resolution of the reduced camera system obtained with the Schur complement.

5.1.3 Datasets used in this work

In this work we have measured the performance of the GEA correction on a large number of reconstruction problems. The datasets *dinosaur*[3], *corridor*[4], *model house*[4], and *maquette*[5] contain feature point trackings detected in short image sequences, along with the camera calibration and a suboptimal initial configuration for the camera poses and the scene structure.

Dataset	**Views**	**Points**	**Projections per view**		
			min	**mean**	**max**
dinosaur	36	4983	257	456	602
wardham	5	1331	347	603	887
modelhouse	10	672	102	284	460
corridor	11	737	260	366	490
boxes2	34	374	72	116	160
synthetic	20	256	256	256	256

Table 5.1: Size of several datasets used in this work, in number of views, 3D points, and projections per view.

The datasets *trafalgar*, *venice*, *dubrovnik* , *ladybug*, and *final*, were used by Agarwal et alt. to evaluate the performance of their BA implementation in large scale reconstructions[6] [Agarwal et al., 2010].

The datasets *trafalgar*, *venice*, and *dubrovnik* were generated from large collections of images downloaded from the Internet, which capture certain popular sites across the world. The dataset *ladybug* was created from images captured with a video camera embedded in a robot, which performed precise straight line translations across several corridors inside a building during the dataset generation. Each one of these datasets contains several suboptimal partial initializations for the reconstruction problem, which were obtained using an incremental SfM reconstruction procedure. These datasets contain the partial reconstructions obtained in each iteration of the reconstruction process, just after the initialization of new camera poses and 3D points, and before the BA correction. Hence these datasets are very useful for the performance evaluation of a correction method, as they contain initialization errors which can be found in practice during the reconstruction process.

[3] Thanks to Wolfgang Niem, University of Hannover.

[4] Oxford's VGG group:
`http://www.robots.ox.ac.uk/~vgg/data/data-mview.html`.

[5] Provided with the source code of laSBA [Lourakis and Argyros, 2009]:
`http://www.ics.forth.gr/~lourakis/sba`.

[6] BA in the large: `http://grail.cs.washington.edu/projects/bal/`

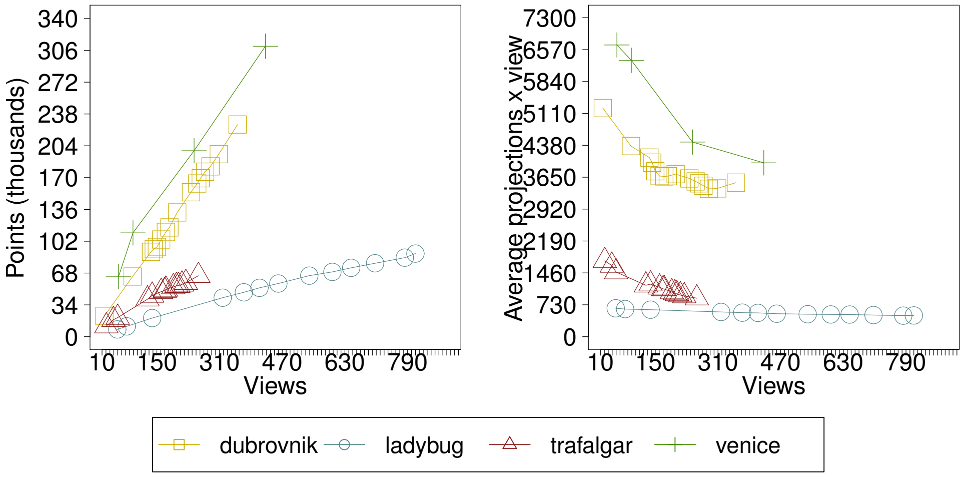


Figure 5.1: Number of points, and image projections in the datasets *trafalgar*, *dubrovnik*, *venice*, and *ladybug*.

The table 5.1 and the figure 5.1 show statistical information about the datasets referenced so far, such as number of feature projections, views and 3D points. To evaluate the GEA performance on these datasets, we must obtain pairwise image feature correspondences from the trackings contained in their reconstructions. Each pair of feature projections in a given tracking produces one pairwise matching between these projections which will be used in the GEA optimization.

These pairwise matchings will contain a minimal or zero fraction of the mismatchings produced by typical image feature correspondence detection methods, as most of them will have been discarded to compose the trackings. To evaluate the robustness of the incremental motion estimation procedure described in section 4.3 in practical working conditions we must use all the feature correspondences obtained by the image feature matching, including the mismatchings. For this purpose we have organized a different group of datasets, each one of them containing a set of images capturing a given scene. These datasets contain the pairwise feature correspondences detected between these images using sample consensus search methods and SIFT feature descriptors, as well as the intrinsic camera parameters and a ground truth reconstruction obtained with a classical incremental motion estimation procedure.

The data sets *leuvencastle* and *medusa*[7] contain image sequences previously used in [Pollefeys et al., 2004] to evaluate the performance of an incre-

[7] `http://www.cs.unc.edu/~marc/`

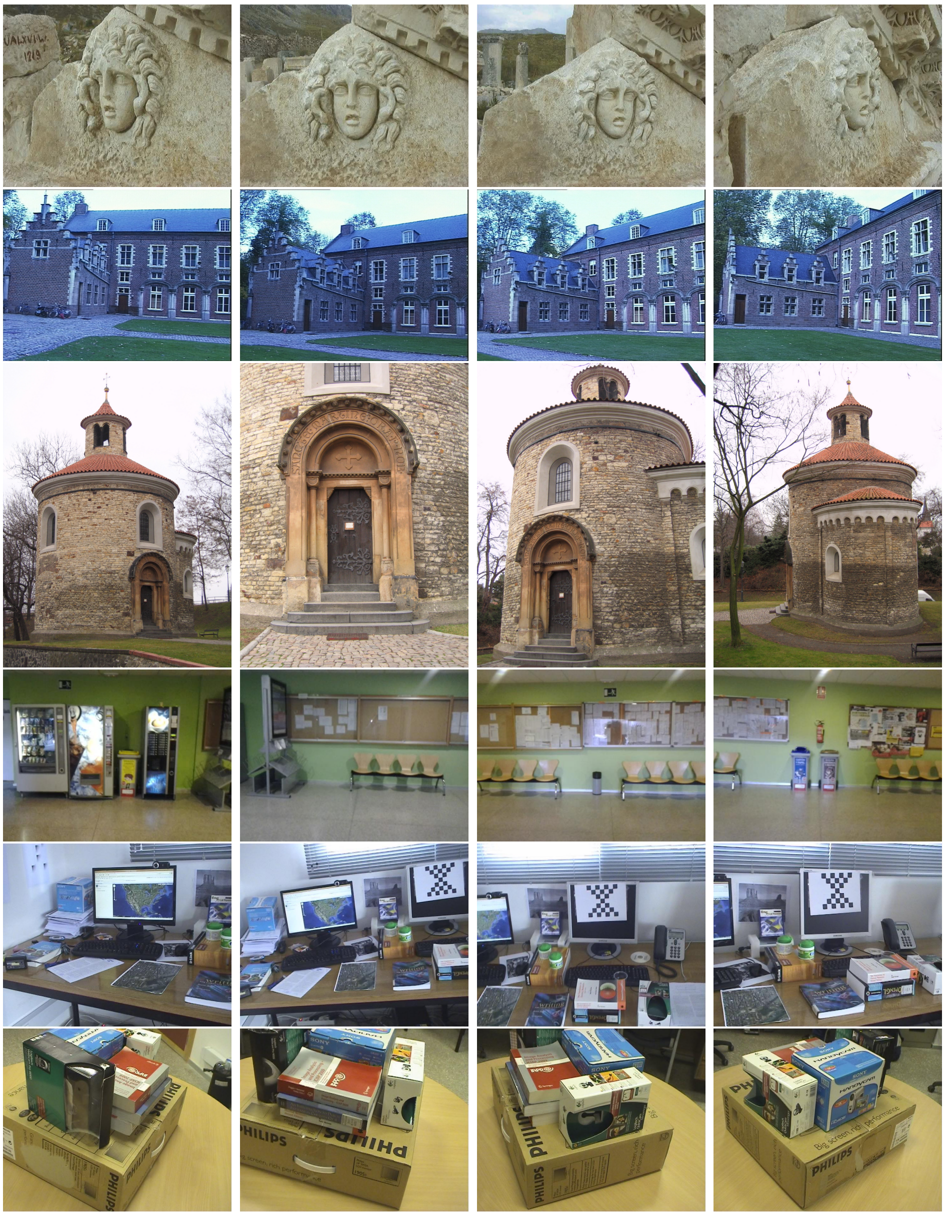


Figure 5.2: Selected frames from the datasets *medusa*, *leuvencastle*, *stmartin*, *hallwall*, *desktoplong* and *boxes8*.

mental SfM procedure. The *desktoplong* and *boxes8* datasets contain frames obtained from the video sequences used in [Rodríguez, López-de-Teruel and Ruiz, 2011*a*] to evaluate the performance of the GEA correction on real-time SfM applications[8]. The dataset *hallwall* contains frames from the video se-

[8] `http://perception.inf.um.es/gea/`

quence used in [Ruiz et al., 2011] to perform visual egolocation on mobile devices by matching image features with points in sparse reconstructions. The *stmartin* dataset contains the images[9] used to test the structureless motion initialization procedure described in 4.1.3.

The figure 5.2 shows some of the images contained in each one of these datasets. The pairwise matchings, the camera calibrations and the ground truth reconstruction contained in these datasets were obtained from the images using the *VisualSfM* framework[10].

Al the datasets referenced contain estimations for the linear intrinsic calibration parameters, which we use to correct the image projection coordinates before the GEA or sSBA optimization. Nonlinear camera parameters such as the radial distortion are ignored. In most cases they contain negligible values, and using them to correct the image coordinates would have a minor contribution in the accuracy of the results obtained.

5.2 GEA and sSBA performance comparison

In this section we compare the performance of the GEA and sSBA optimizations on several datasets. We demonstrate that the GEA correction obtains accurate camera pose estimations, which can be used to obtain the scene structure with a quality very similar to that obtained by BA. We also demonstrate that GEA can reduce the error of the estimated camera poses in a fraction of the time required to correct the full reconstruction (camera poses + structure) using sSBA.

5.2.1 Optimal error configuration

The table 5.2 and the figure 5.3 show the following reprojection errors obtained with sSBA and GEA on several datasets:

Initial: reprojection error for the initial reconstruction stored in the dataset.

SBA: reprojection error for the initial reconstruction refined with SBA.

GEA: reprojection error for the reconstruction after a GEA correction and a linear re-estimation of the points in the structure.

GEA+SBA: reprojection error for the previous reconstruction configuration after a SBA refinement.

[9] http://cmp.felk.cvut.cz/~martid1/demoCVPR07/

[10] http://homes.cs.washington.edu/~ccwu/vsfm/

Dataset	views	Reprojection error			
		Initial	SBA	GEA	GEA+SBA
dinosaur	36	22.25	0.81	2.04	0.17
wardham	5	4.42	0.15	0.36	0.15
modelhouse	10	16.26	0.71	1.92	0.71
corridor	11	18.60	0.82	1.00	0.82
boxes2	34	2.91	2.15	2.18	2.15
synthetic	20	2.18	0.96	0.98	0.96

Table 5.2: Reprojection errors obtained for several datasets with sSBA and GEA.

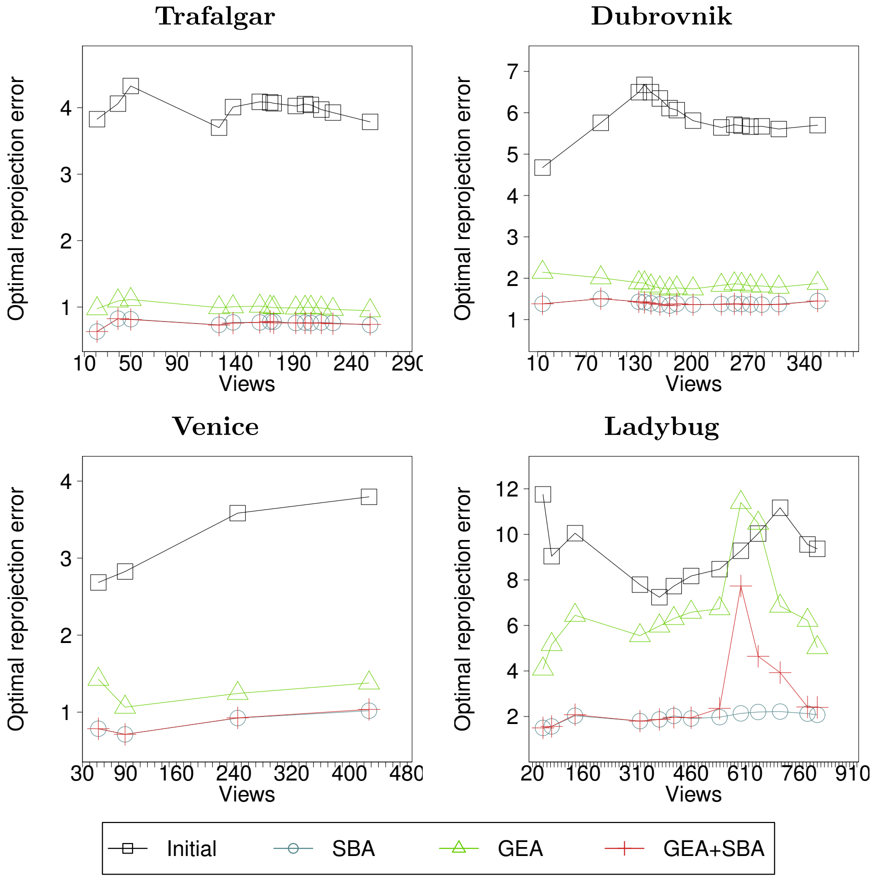
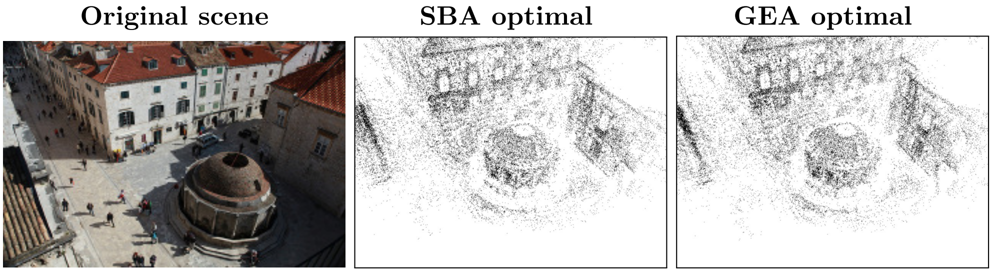


Figure 5.3: Reprojection errors obtained with sSBA and GEA for the reconstruction problems *trafalgar*, *dubrovnik*, *venice*, and *ladybug*.

The reprojection error of the GEA reconstruction is in most cases very similar to the optimal reprojection error obtained with SBA. The similarity between the reprojection errors **GEA+SBA** and **SBA** suggest that the **GEA** reconstruction configuration is inside the basin of the optimal reprojection error configuration.


Original scene

SBA optimal

GEA optimal

Figure 5.4: Optimal reconstructions obtained with GEA and BA for the dataset *dubrovnik* (88 views). The 3D points in the GEA reconstruction were estimated from the optimal configuration for the camera poses provided by GEA, and the linear triangulation method.

The similarity between the structures obtained from the GEA camera poses, and SBA can be compared with visual inspection in figure 5.4, which shows a detail of the optimal reconstruction obtained with both methods GEA and SBA, for the dataset *dubrovnik*.

Dataset	**views**	**Num. points**		**Min. proj. per view**	
		original	**evaluated**	**original**	**evaluated**
dinosaur	36	4983	4833	257	245
wardham	5	1331	1292	347	328
modelhouse	10	672	656	102	94
corridor	11	737	721	260	250
boxes2	34	374	368	72	69
synthetic	20	256	251	256	251

Table 5.3: **Num. points:** number of total points in the structure , and inlier points used in the evaluation of the reprojection error. **Min. proj. per view:** minimum number of point projections from the structure, and minimum number of inlier points used to evaluate the reprojection error of a view in the datasets.

As we described in section 5.1.1 in the reprojection error evaluation we have ignored some of the points with the largest residuals. This prevents the influence of outliers which can appear during the Linear-LT structure estimation, or the BA correction due to incorrect dataset initialization, or structure triangulation.

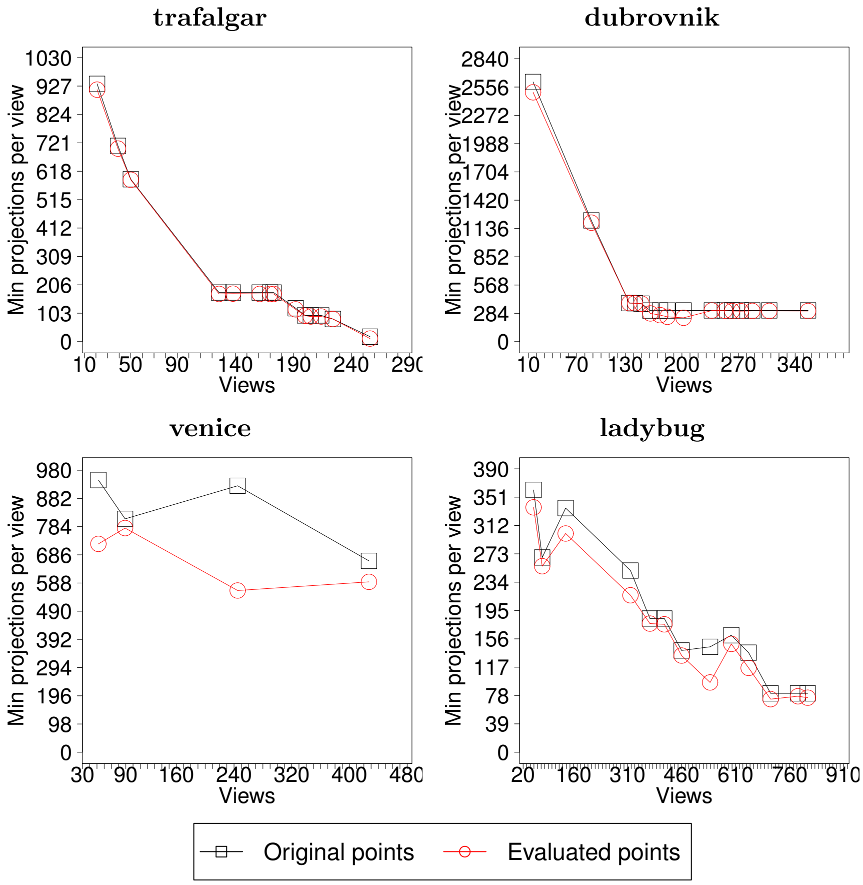


Figure 5.5: Minimum number of projections per view, for the points in the reconstruction problem, and the inlier points used in the reprojection error evaluation for the datasets *trafalgar*, *dubrovnik*, *venice*, and *ladybug*.

It could be argued that by ignoring the points with the largest residuals we could obtain a small average reprojection error, even if one of the views has an incorrect camera pose. For example, if that incorrect view contains projections only for the 3D points with the largest residuals ignored in the reprojection error evaluation.

This was not the case for the tests described in this chapter. The value *minimum projections per view* shown in table 5.3 and in the figure 5.5 is the number of 3D point projections used in the evaluation of the average reprojection error for the views obtained with GEA. These figures demonstrate that the quality for the camera pose in every view contributes to the average reprojection error measured in our tests.

5.2.2 Convergence speed

Figures 5.6 and 5.7 show the convergence speed to the optimal configuration for the GEA and SBA correction methods in several datasets. We have evaluated the convergence with different damping values in the Levenberg-Marquardt and Gauss-Newton algorithms, to evidence that GEA requires a few iterations to obtain camera poses very close to the optimal configuration when using either large or small damping parameters, even when this value is zero. Hence to provide the optimal configuration GEA does not require a fine tuning for this value.

Setting the LM damping parameter with large initial values will provide a slow error reduction speed, while in most cases these values will still provide a fast convergence with GEA. Meanwhile, BA can diverge from the optimal configuration when the damping parameter is initialized with a value too small. In this case the reprojection error does not have a convex shape at the initial reconstruction configuration, and the solution for the second order approximation to the cost error is far from the real optimum.

Most BA implementations increase the damping parameter in this case to prevent divergence. This way the optimization can reach the optimal configuration in a reasonable number of iterations. Nevertheless this increases the optimization time, as LM must waste several iterations tuning the damping parameter.

In the case of GEA, a sufficiently small or zero value ensures a fast convergence to the optimal solution for all the datasets used in the performance tests. An exception is the dataset *ladybug*, which features critical configurations that we will discuss latter in section 5.3.3.

5.2.3 Optimization step time

In this section we show the time required by each stage of the GEA and sSBA optimizations to perform a single optimization iteration for the reconstructions in several datasets[11]. In the figures 5.8 and 5.9 the stage *reduce* estimates the reduced coefficient matrices Ω from the pairwise image matchings. For the SBA correction, the stage *setup* obtains the reduced camera system, from the image projections and the actual reconstruction configuration. In this stage, each projection in the reconstruction produces an increment in two block-elements of the camera matrix. Hence the computation time of this step grows significantly with the number of projections. Meanwhile for the GEA correction this stage obtains the step equation for

[11] These tests were executed on an Intel Core i3 CPU, with 3.20GHz and 4Gb of RAM memory.

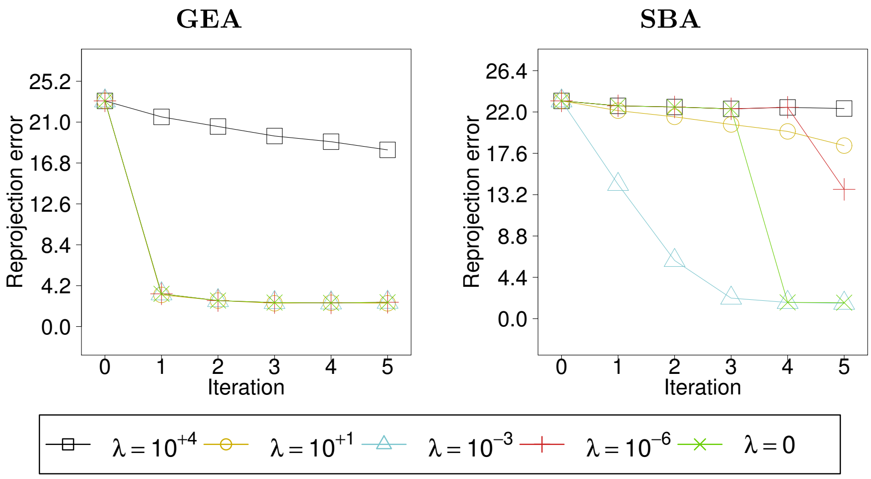


Figure 5.6: Convergence for GEA and SBA on the dataset *dinosaur*, using different values for the damping parameter λ in the optimization. The initial error (iteration 0) is different in both graphs because in the case of GEA, the reprojection error is evaluated on the linear triangulation of the 3D points. Meanwhile in the SBA plot the initial error corresponds to the structure configuration originally contained in the dataset.

the Gauss-Newton optimization. The GEA step equation is evaluated with less computational cost for GEA than for SBA.

In both methods sSBA and GEA, the *solve* stage obtains the solution for the step equation. The times for this stage are different because both methods use a different implementation for the iterative solver. If both sSBA and GEA implementations were using the exact same code for solving the step equation, the times for this stage would be the same.

The *triangulate* step in the GEA correction estimates the 3D location for every point in the structure, using the GEA camera poses.

The computational cost of GEA in practice will vary depending on the design of the final reconstruction application, as the main computation time bottleneck in the GEA correction is the linear triangulation of the structure. Compared with sSBA, the computational efficiency of GEA will grow with the number of 3D points and image projections in the reconstruction.

A large fraction of the triangulation time with GEA can be saved in practice, given that most or all the 3D points will not need to be updated. Incremental SfM pipelines only require to re-estimate the 3D points which are needed to resect new camera poses in each iteration. The incremental motion estimation procedure described in section 4.3 does not require to evaluate the structure at all during the motion estimation.

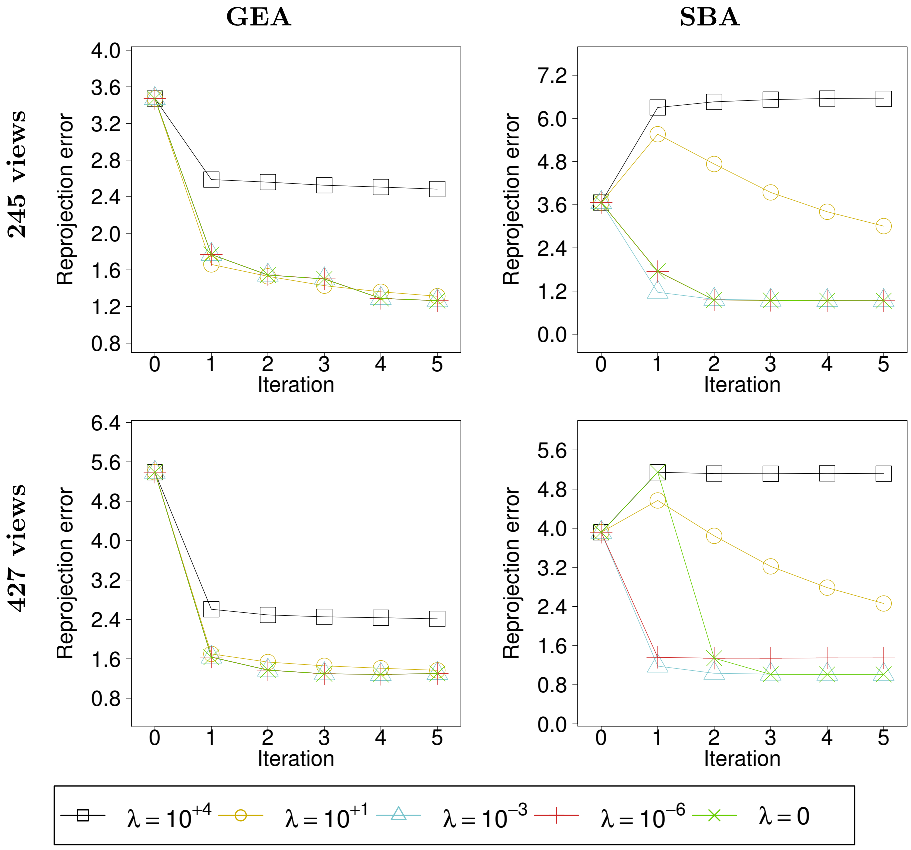


Figure 5.7: Convergence for GEA and SBA on the dataset *venice* using different values for the damping parameter λ in the optimization. Again, the initial error (iteration 0) is different in both graphs, because the reprojection error obtained with GEA is evaluated with the linear triangulation of the points, whereas the initial SBA error corresponds to the dataset configuration.

Figure 5.10 shows a comparison of the time required by the data reduction step in the GEA optimization for the different methods described in section 3.2.1. The direct method which uses the Ω matrix takes a fraction of the computation time required by the decomposition methods which were initially proposed in [Rodríguez, López-de-Teruel and Ruiz, 2011*b*] to reduce the matching data.

5.3 Robustness of the GEA optimization

In this section we evaluate the performance of the GEA correction under possible failure conditions such as critical motion sequences or feature mis-

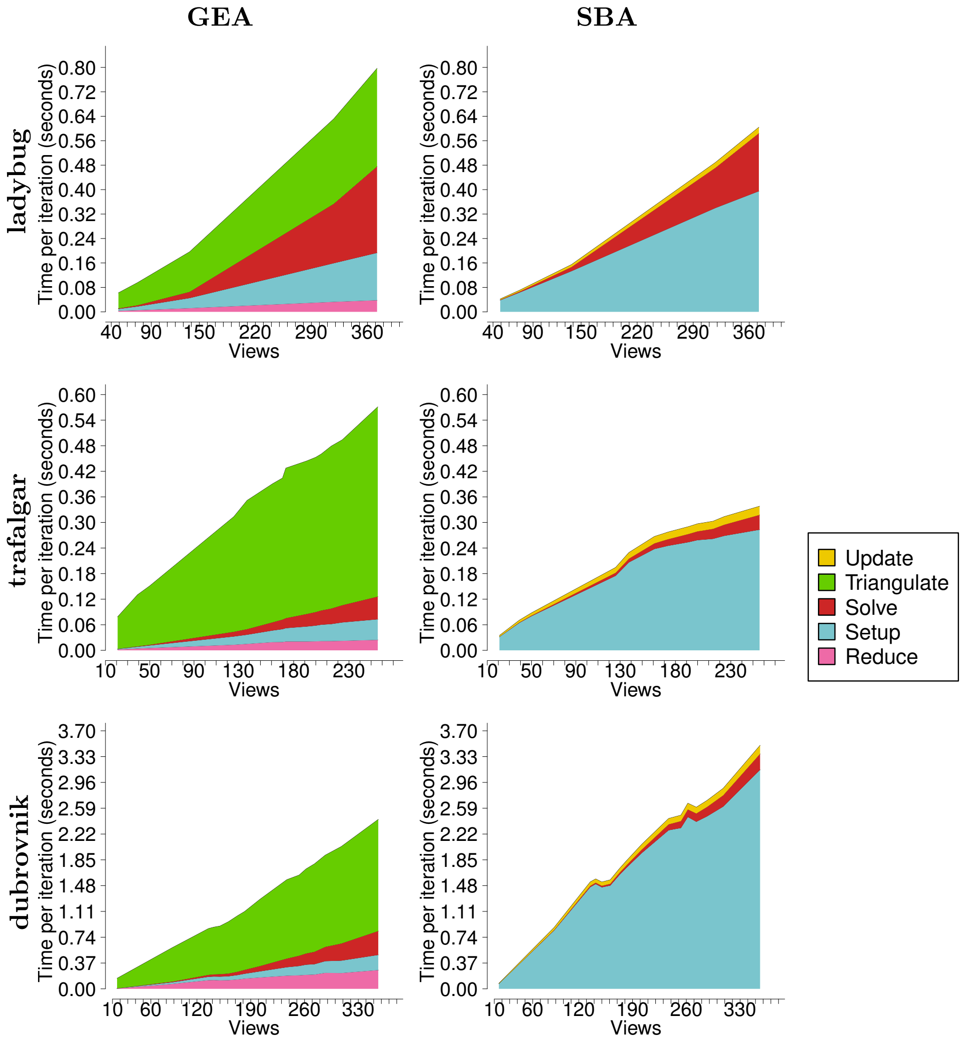


Figure 5.8: Time per iteration required by GEA and SBA for several datasets. The *reduce* and *triangulate* steps are performed once only in each GEA correction, even if the Gauss-Newton optimization requires several iterations. Furthermore, the *triangulate* step is not required to perform structureless motion estimation.

matching appearance. We demonstrate with empirical results that, under general circumstances, the GEA optimization described during the previous chapters is sufficiently robust against these conditions.

5.3.1 Robustness against feature mismatchings

To demonstrate the robustness of GEA against the appearance of pairwise feature mismatchings we performed several optimization tests. In the first

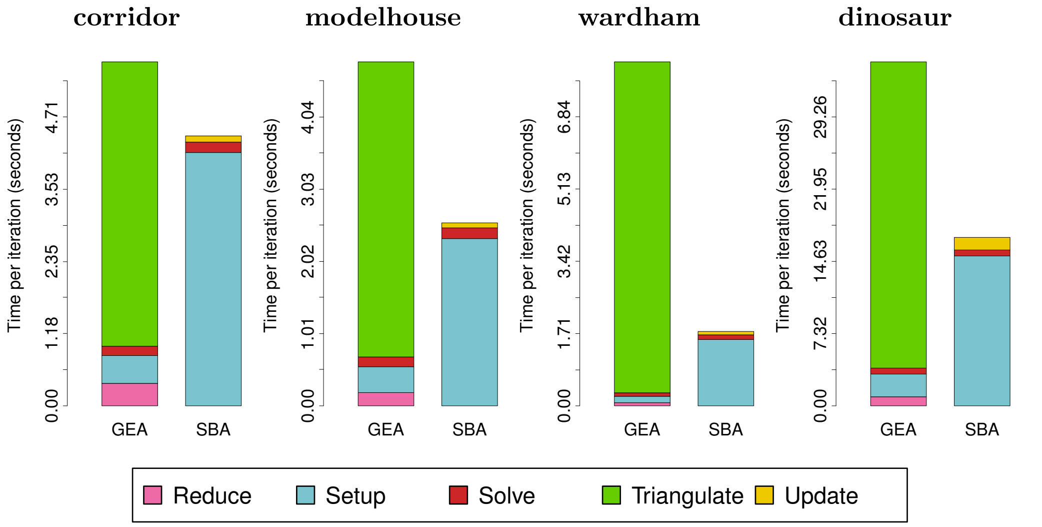


Figure 5.9: Time (in miliseconds) required by each stage of the GEA and SBA corrections for the datasets *corridor*, *modelhouse*, *wardham* and *dinosaur*. The *triangulate* step obtains the 3D sparse structure, and it is not required by GEA to estimate the camera poses.

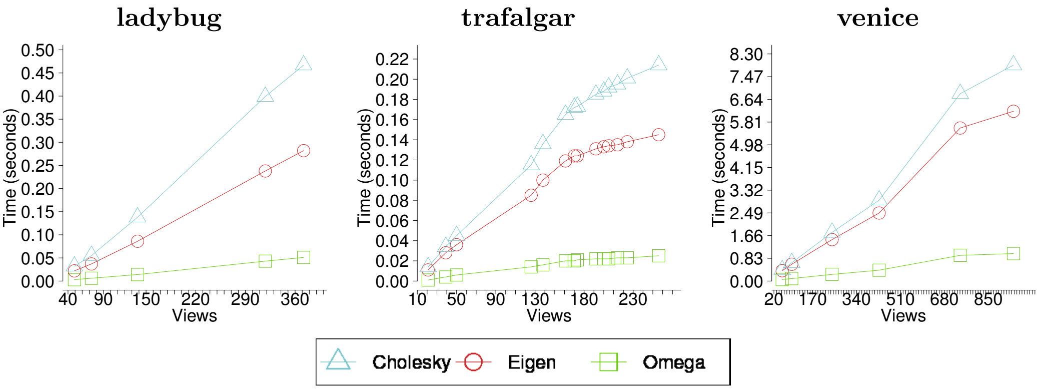


Figure 5.10: Computation time required to evaluate the reduced matrices from the matching data for different datasets, using two methods: the one proposed in [Hartley, 1998*b*; Rodríguez, López-de-Teruel and Ruiz, 2011*b*] with both the Cholesky and eigen decompositions for the factorization of the reduced data matrix, and the Ω method proposed in section 3.2.1 which does not require a matrix factorization.

set of these tests we introduced a significant number of artificially generated mismatchings in the GEA correction of the datasets *corridor*, *boxes2*, *trafalgar*, *venice* and *dubrovnik*. These mismatchings were generated to have a small epipolar residual with the ground truth configuration for the camera poses. This ground truth configuration is assumed to be the optimal GEA configuration obtained from the camera poses, and the original feature trackings in the dataset.

Hence, each one of these mismatchings $\mathbf{p} \leftrightarrow \mathbf{q}$ is generated as follows. The coordinates for the first feature location $\mathbf{p} = (x_p, y_p, 1)$, and the horizontal coordinate x_q for the second feature location $\mathbf{q} = (x_q, y_q, 1)$ are randomly generated from the set $[-1, 1] \in \mathbb{R}$. This locates the point $\mathbf{p}$, and the coordinate x_q inside a square box of size 2, centered at the origin of coordinates in the image plane. The remaining image coordinate y_q is solved from this equation:

$$\mathbf{q}^T E_{ij} \mathbf{p} = r \tag{5.2}$$

where r is the epipolar residual that we want for the generated mismatching, and E_{ij} is the essential matrix parametrization of the ground truth camera poses for the views i and j.

Dataset	**Fraction of synthetic mismatchings**				
	0%	10%	25%	50%	75%
dinosaur	1.9367	2.3632	2.2872	2.8388	4.4753
wardham	0.3579	0.3328	0.4439	0.3917	0.3511
modelhouse	1.9446	1.9016	1.8101	1.8893	1.8905
trafalgar-50	1.1087	1.1159	1.1900	1.2886	1.3578
dubrovnik-88	2.0219	3.1046	2.5740	3.0385	2.3530
venice-52	1.4357	1.3812	1.3895	1.4347	1.4876

Table 5.4: Reprojection errors obtained using GEA on several datasets which include a fraction of randomly generated mismatchings with a small epipolar residual for the ground truth camera poses.

The table 5.4 shows the results obtained for this first set of experiments, where a fraction of mismatchings is added to the list of feature correspondences used in the GEA correction. The residual r for each artificial mismatching is randomly selected as the residual for one of the original matchings obtained from the trackings in the dataset. This way the residual distribution of both the mismatchings and the original matchings is similar. The tests show that GEA provides accurate camera poses in most cases, even when 75% of the image correspondences are synthetic mismatchings. As suggested in section 4.3.3, feature mismatchings satisfying the correct epipolar geometry for the camera poses do not degrade the performance of the GEA correction.

The table 5.5 shows the results obtained in a second set of experiments, where the synthetic mismatchings introduced in the GEA correction have a

Dataset	**Fraction of synthetic matchings**					**Random mismatchings**
	0%	10%	25%	50%	75%	
corridor	0.9708	1.3173	1.7649	1.8767	2.0825	20.9127
boxes2	2.1873	4.5058	5.3289	9.3690	14.1217	18.5555
trafalgar-21	0.9517	1.1744	2.0826	4.5713	6.4662	72.4834
trafalgar-50	1.0993	2.0147	3.3696	5.6733	7.5078	77.0119
trafalgar-126	0.9897	2.2951	2.3661	3.6071	5.2475	110.9736
trafalgar-206	0.9924	1.8839	2.7068	3.6410	5.1623	81.8933
trafalgar-257	0.9328	1.7529	2.2349	3.3266	5.0442	127.9680
dubrovnik-88	2.0177	3.0652	4.3996	7.8411	12.3717	273.8359
dubrovnik-150	1.8218	3.2866	5.1214	8.9883	15.2893	170.2761
dubrovnik-308	1.7959	3.0369	4.3747	7.6861	15.8416	411.2761
venice-52	1.4199	2.0235	2.5989	3.5267	4.8030	40.1494
venice-89	1.0295	1.3762	1.7545	2.4587	3.7931	52.9438
venice-245	1.2449	18.1216	18.3851	18.6509	22.6964	79.3621
venice-427	1.3904	8.1374	7.2535	9.7570	10.4587	226.3414

Table 5.5: Reprojection errors obtained with the **GEA correction** on several datasets. A fraction of synthetic matchings containing a significant epipolar residual with the ground truth camera pose configuration was added to a 10% of the view pairs in the optimization. The column **random mismatchings** shows the reprojection error obtained when this 10% of view pairs contains a 75% of synthetic matchings with a very large epipolar residual.

large epipolar residual error, up to 50 times larger than the residuals for the original matchings in the dataset. In this case, the mismatchings were introduced only in a 10% of the view pairs, to simulate that the sample consensus matching provided for these view pairs a given fraction of correspondences with large epipolar residuals. We can see that the error of the optimal GEA camera pose configuration increases with the percentage of these synthetic mismatchings, as expected. The last column in this table shows the reprojection error obtained when a 75% of the correspondences in these view pairs are artificial mismatchings, where the coordinate y_q was randomly generated without satisfying the epipolar geometry of equation (5.2) at all. The mismatchings in these tests represent matching failures where the sample consensus search obtains incorrect epipolar models which are quite different from the valid ones. In this case the reprojection error becomes quite large.

The table 5.6 shows the results obtained in the same experiments when the GEA correction is robustified with the procedure described in section 4.3.3, using a robustification threshold parameter μ set to 10^{-4}. The error obtained with the robustified GEA correction decreases, and in most cases

Dataset	**Fraction of synthetic matchings**					**Random mismatchings**
	0%	10%	25%	50%	75%	
corridor	0.9849	1.2857	1.0101	0.9744	0.9744	0.9744
boxes2	2.1873	2.1885	2.1885	2.1885	2.1885	2.1885
trafalgar-21	0.9517	1.1146	0.9510	1.0076	1.0205	0.9188
trafalgar-50	1.0939	1.3424	1.3102	1.1702	1.1798	1.1590
trafalgar-126	0.9857	1.2451	1.1678	1.1206	1.1263	1.0099
trafalgar-206	0.9929	1.1371	1.1545	1.1429	1.1260	1.0069
trafalgar-257	0.9340	1.0860	1.0866	1.0330	1.0296	0.9455
dubrovnik-88	2.0216	2.0291	2.0240	2.0231	2.0253	2.0234
dubrovnik-150	1.7916	2.1283	1.9164	2.1533	2.3135	1.7924
dubrovnik-308	1.8080	1.8312	1.8312	1.8406	1.8348	1.8100
venice-52	1.4568	1.4103	1.4381	1.4319	1.4592	1.4626
venice-89	1.0338	1.0957	1.0748	1.0381	1.3269	1.0591
venice-245	1.2228	18.1930	17.0374	17.4025	18.3389	1.2190
venice-427	1.3521	7.1942	6.4374	6.8524	7.1689	1.3580

Table 5.6: Reprojection errors obtained with the *robustified* **GEA correction** on several datasets. A fraction of synthetic matchings containing a significant epipolar residual with the ground truth camera pose configuration was added to a 10% of the view pairs in the optimization. The column **random mismatchings** shows the reprojection error obtained when this 10% of view pairs contains a 75% of synthetic matchings with a very large epipolar residual.

it is very close to the error obtained with the mismatching-free GEA correction. When the mismatchings are totally randomly generated, without satisfying the epipolar geometry for the camera poses at all, the robustification method successfully discards all the terms in the GEA cost error containing mismatchings. In this case, the robustified GEA correction obtains camera pose estimations with an accuracy similar to those obtained by the classical GEA correction with the original set of feature correspondences, which is free from the artificial mismatchings.

5.3.2 Evaluation of cost error sparsification

In this section we evaluate the results obtained with the graph reduction procedure described in section 3.2.5, which sparsifies the step equation in the GEA optimization.

Table 5.7 shows the optimal GEA reprojection error obtained for different degrees of sparsification. In these tests the parameter s_c is set to 10, so the graph reduction procedure will not delete links connecting nodes which are themselves linked to 10 or less other nodes in the graph.

Dataset	**Cost terms used**			**Iteration time**			**Rep. error**		
trafalgar-50	921	608	304	6.92	4.73	2.43	1.11	1.15	1.36
trafalgar-126	2969	1960	980	29.75	19.75	10.19	0.99	1.01	1.19
trafalgar-170	4725	3119	1559	50.25	34.35	17.42	1.00	1.03	1.14
trafalgar-206	5961	3934	1967	69.74	47.39	24.35	1.00	1.01	1.16
trafalgar-257	7723	5097	2548	99.75	67.69	35.08	0.96	0.97	1.13
dubrovnik-88	3334	2201	1100	30.29	20.07	9.89	2.01	2.05	2.38
dubrovnik-135	7168	4731	2365	74.16	49.66	24.95	1.85	1.88	2.16
dubrovnik-150	7916	5225	2612	84.86	56.73	28.55	1.77	1.84	2.20
dubrovnik-202	12914	8523	4261	164.50	106.93	53.67	1.71	1.74	1.95
venice-52	1295	855	427	10.11	6.65	3.45	1.43	1.50	1.82
venice-245	19439	12830	6415	269.49	180.75	87.63	1.27	1.29	1.35
Parameter s_f	100%	66%	33%	100%	66%	33%	100%	66%	33%

Table 5.7: Reprojection errors obtained after 10 GEA iterations, with the graph reduction elimination of terms proposed in section 3.2.5. The iteration time (in milliseconds) includes the evaluation and solving of the Gauss-Newton step equation.

The number of terms in the GEA cost error can be significantly decreased with the graph reduction technique, increasing the sparsity of the coefficient matrix in the step equation, and reducing significantly the computation time required in the Gauss-Newton optimization as can be seen in the table 5.7. Meanwhile the GEA correction still obtains cameras with a high accuracy, almost equivalent to the camera pose accuracy obtained with the correction of the original cost.

5.3.3 Influence of critical sequences

The accurate results obtained by GEA on the datasets *dubrovnik*, *trafalgar*, and *venice* demonstrate that the correction method can be used without problems in most reconstruction problems where the camera motion is not a linear translation. However, as discussed in sections 3.3.1 and 3.4.1 motion correction methods based on pairwise constraints which do not enforce the scale, such as GEA, pose-graph relaxation or motion averaging, will not be able to estimate accurately the camera poses on reconstruction problems which contain linear camera motion configurations.

As can be seen in figure 5.11, when one of these critical motion sequences is present in the reconstruction the optimization diverges from the optimal solution, and the error in the estimated camera poses grows with each iteration.

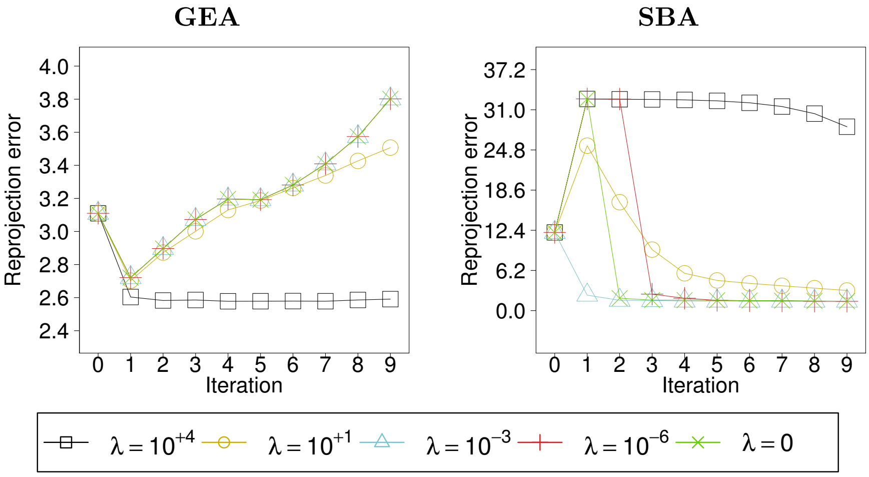


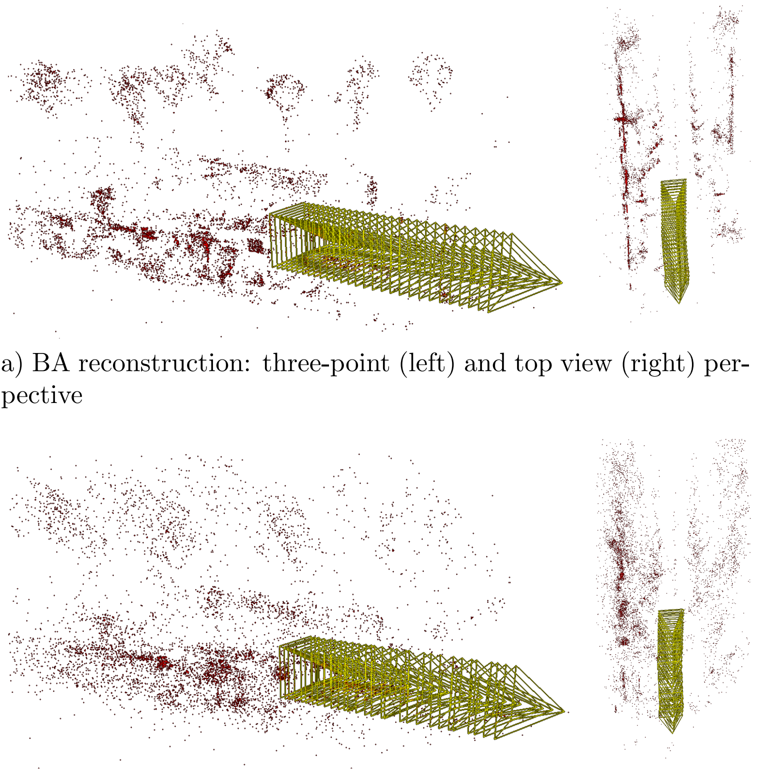
Figure 5.11: Convergence for GEA and SBA on the dataset *ladybug* (with 49 views) using different values for the damping parameter λ in the optimization. The GEA optimization diverges from the optimal configuration, as the reconstruction in this dataset contains a critical motion.

Figure 5.12 shows a visual comparison between the optimal configuration obtained with BA on the dataset *ladybug*, and the configuration obtained with GEA after a given number of iterations. The camera poses obtained by GEA are correctly arranged on the line of the robot translation, but the estimated location of these camera centers on this line is incorrect. For this reason, the structure triangulated with the camera poses corrected with GEA is noticeable different from the optimal BA reconstruction, as it contains a significant error.

In practice, GEA can be applied successfully to a large number of SfM problems which contain nearly critical camera pose configurations. Any deviation of the camera motion from the perfect linear translation can prevent the problems of critical motions with GEA. To demonstrate this fact, we have evaluated the GEA accuracy and robustness on other datasets which contain near linear translation camera motions. In these datasets, even after a large number of iterations the GEA correction converges to a camera pose configuration close to the BA solution.

For example, in the dataset *corridor* the camera poses are arranged in a quasi linear configuration with a slight curve deviation. This is also the case for the *modelhouse* and *wardham* datasets.

The figures 5.14 and 5.15 show the reconstruction obtained with BA and GEA for the datasets *corridor* and *wardham* respectively. As can be seen in the figure 5.13, both GEA and SBA converge in these datasets to camera pose



(a) BA reconstruction: three-point (left) and top view (right) perspective

(b) GEA reconstruction: three-point (left) and top view (right) perspective

Figure 5.12: *Ladybug* dataset (49 views). The GEA reconstruction diverges from the optimal reprojection error configuration when a critical motion configuration is found. The figures show the reconstructions obtained with GEA and SBA after 60 iterations using a conservative value of 1.0 for the damping parameter. The dataset *ladybug-49* contains views for two different camera trajectories: one with camera facing towards, and other with the camera facing sideways. Only one of them is shown in the images for the sake of clarity.

configurations with a similar small reprojection error. The exact location of the camera centers for the *corridor* dataset is accurately estimated using the GEA correction, thanks to the slight curve in the trajectory which prevents the GEA failure due to the critical linear translation.

Robustness evaluation on synthetic datasets

To evaluate with precision the influence of near critical camera configurations in the GEA correction, we have performed another set of experiments with synthetic reconstruction configurations which contain nearly aligned camera

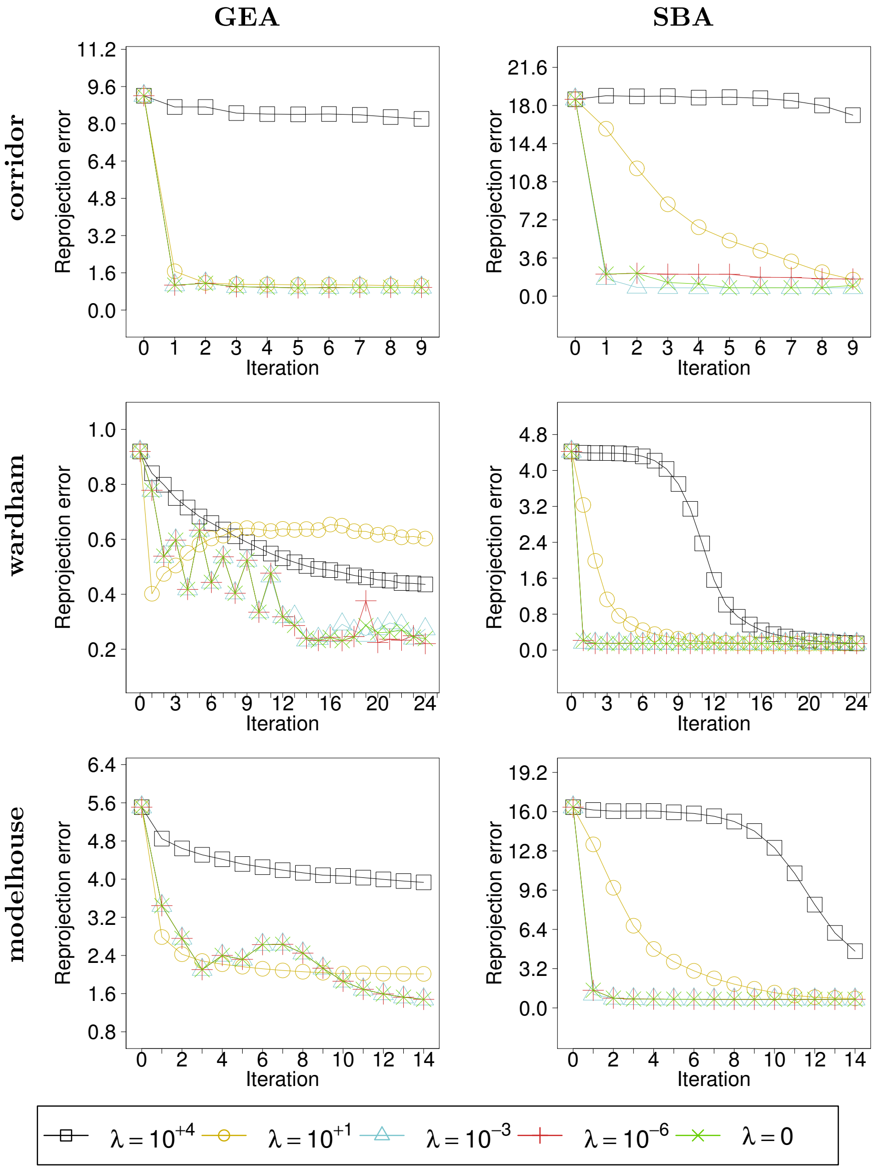
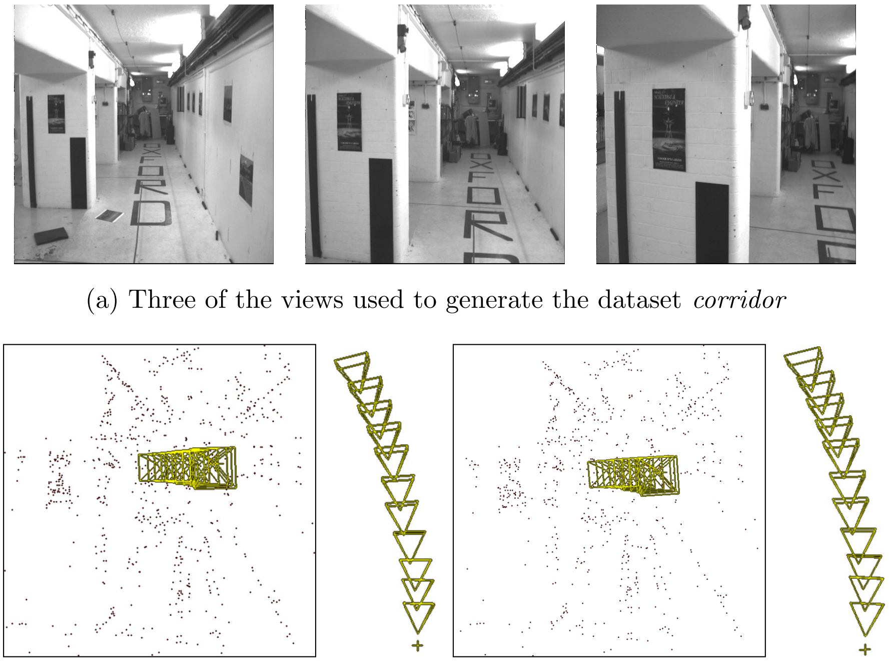


Figure 5.13: Convergence for GEA and SBA on the datasets *corridor*, *wardham* and *modelhouse*, where the camera translates in a near-linear motion.

poses.

Figure 5.16 shows the basic configuration of these synthetic reconstructions. Each one of these configurations contains five views and 100 feature points. The points are randomly located in front of the cameras, which are arranged along a straight line with a baseline distance between the first and the last camera center of 1. Uniform noise is added to the image coordi-



(a) Three of the views used to generate the dataset *corridor*

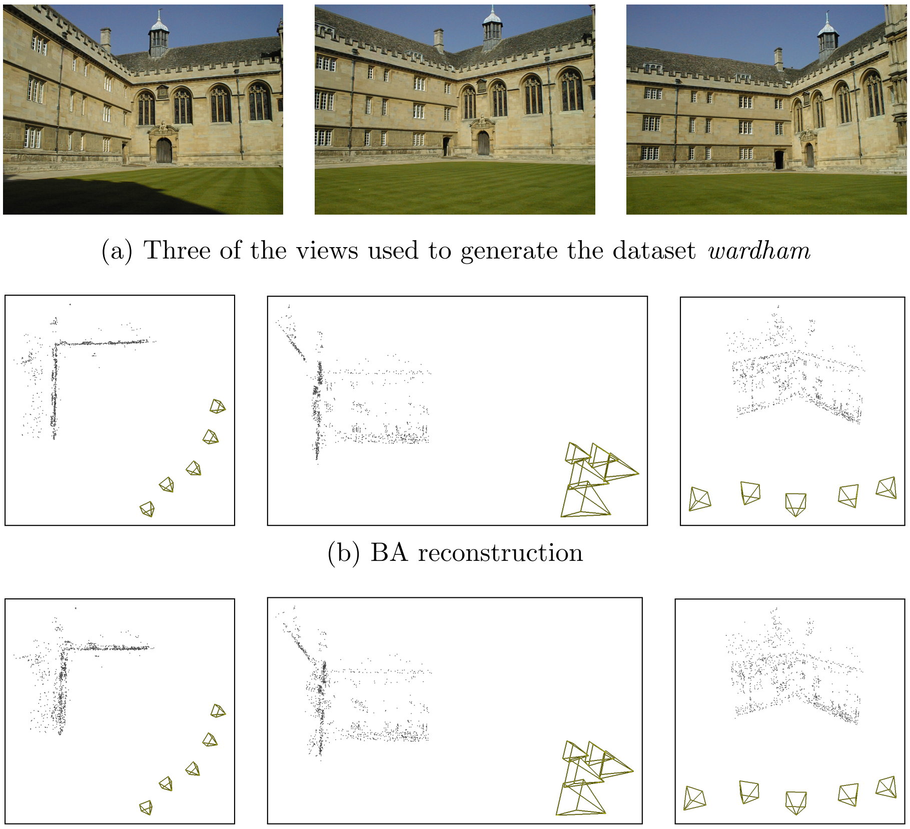
(b) BA reconstruction

(c) GEA reconstruction

Figure 5.14: Reconstructions obtained from the dataset *corridor* using the GEA and SBA corrections. **Top row:** some of the images used to obtain the dataset. **Bottom row:** reconstructions and camera motion obtained with the GEA and BA corrections.

nates, to make the experiment more similar to real image data conditions. These reconstructions have a free parameter which indicates the distance of the third camera pose to the motion line for the rest of the cameras. This distance breaks the linearity of the camera motion in the reconstruction.

In the same figure we can see the accuracy obtained with GEA for a large number of these synthetic reconstructions, compared with the accuracy obtained using BA, with a varying offset distance for the third view. This figure demonstrates that for certain reconstruction problem configurations, having one of the cameras located a small offset distance away from the aligned configuration is a sufficient condition to prevent the performance degradation of GEA due to critical configurations.



(a) Three of the views used to generate the dataset *wardham*

(b) BA reconstruction

(c) GEA reconstruction

Figure 5.15: Reconstructions obtained from the dataset *wardham* using the GEA and SBA corrections. **Top row:** some of the images used to obtain the dataset. **Middle row:** top, side and front view of the reconstruction obtained with the SBA correction. **Bottom row:** top, side and front view of the reconstruction obtained with the GEA correction.

5.4 Incremental motion estimation evaluation

In this section we evaluate the performance of the structureless incremental motion estimation procedure described in section 4.3 in several reconstruction problems. With these experiments, we demonstrate that the structureless motion estimation method proposed can accurately initialize a large number of views for most of these reconstruction problems, as well as be used to obtain an accurate structure estimation from the input image matchings.

In a first set of experiments, we evaluate the motion initialization method on the datasets used in the previous section to evaluate the GEA performance. We use the matchings obtained from the trackings in these datasets

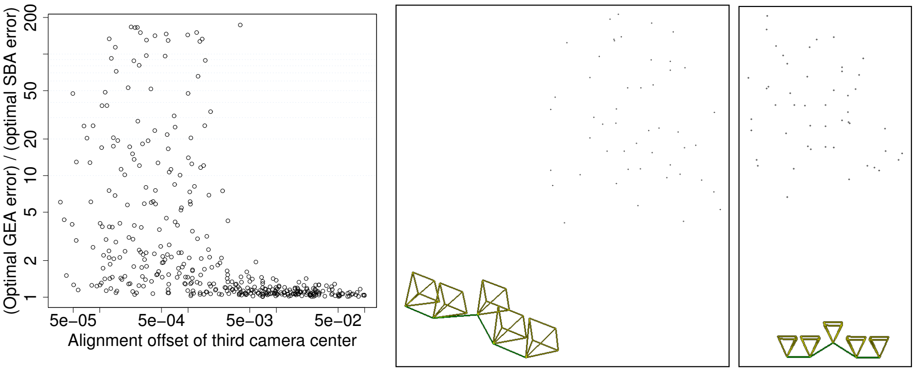


Figure 5.16: **Left:** influence of near critical motion configurations in the GEA correction of synthetic reconstruction configurations. The horizontal axis represents the distance between the camera center of the third view with the translation line. The vertical axis represents the ratio between the reprojection errors obtained with GEA and SBA. **Middle and right:** top and side view of an example synthetic reconstruction configuration used to evaluate the GEA tolerance to near linear motions.

as input feature correspondences for the incremental motion estimation procedure. These pairwise matchings will be used with the incremental motion estimation to obtain the camera poses without computing the structure. In a second set of experiments we evaluate the motion initialization method on image matchings detected on image sequences using SIFT. Most of the mismatchings were filtered using pairwise sample consensus search, as described in section 4.1.1.

5.4.1 Using multiple view feature trackings

In this subsection we evaluate the performance of the structureless incremental motion estimation procedure on the datasets *trafalgar*, *dubrovnik* and *venice*. In these experiments the input pairwise feature matchings for this algorithm are obtained from the trackings contained in the datasets.

The camera poses estimated with this procedure are compared with a ground truth camera pose configuration, which is obtained by correcting the initial camera pose configuration in the dataset with GEA. We re-estimate the structure with the linear method using these camera pose configurations, and the trackings contained in the datasets.

The number of views and the reprojection error of both reconstruction configurations can be seen in table 5.8. Notice that our method does not initialize certain views which can increase significantly the reprojection error. Hence the error obtained with the method proposed can be smaller than the error obtained with other SfM methods, such as the one used to obtain the ground truth. For most datasets, the initialization procedure obtains accurate camera poses for a large fraction of the views, independently of the reconstruction problem size. Figure 5.17 shows a detail of the reconstructions obtained for the dataset *trafalgar-161*.

Dataset name	**Ground truth**		**Initialization**	
	Views	**Error**	**Views**	**Error**
dinosaur	36	2.493	18	1.375
modelhouse	10	1.748	9	1.452
corridor	11	0.716	6	0.667
trafalgar	21	0.973	21	0.792
	50	1.122	50	1.020
	161	1.034	156	1.030
	200	1.011	192	0.998
	256	0.958	219	0.940
dubrovnik	16	1.300	6	0.666
	88	1.998	85	1.781
	142	1.806	138	1.628
	182	1.728	155	4.760
	202	1.692	135	24.846
venice	52	1.470	52	0.970
	89	1.081	89	0.899
	245	1.208	240	1.087
	427	1.307	415	1.174

Table 5.8: Comparison between reprojection error and number of views estimated with the proposed incremental camera pose initialization method (**initialization**), and the reconstruction obtained with the optimal GEA camera poses (**ground truth**).

5.4.2 Using pairwise feature matchings

The table 5.9 shows the results for a set of experiments where we use the pairwise feature matchings detected using SIFT and sample consensus search in the image sequences *medusa*, *leuvencastle*, *stmartin*, *hallwall*, *desktoplong* and *boxes8* to estimate the camera poses with the incremental motion estimation procedure. We obtain the sparse structure from the camera poses by composing the pairwise correspondences into trackings, and using the linear triangulation to obtain the 3D point locations. The figures 5.18 and 5.19 show respectively the sparse and dense reconstructions obtained from

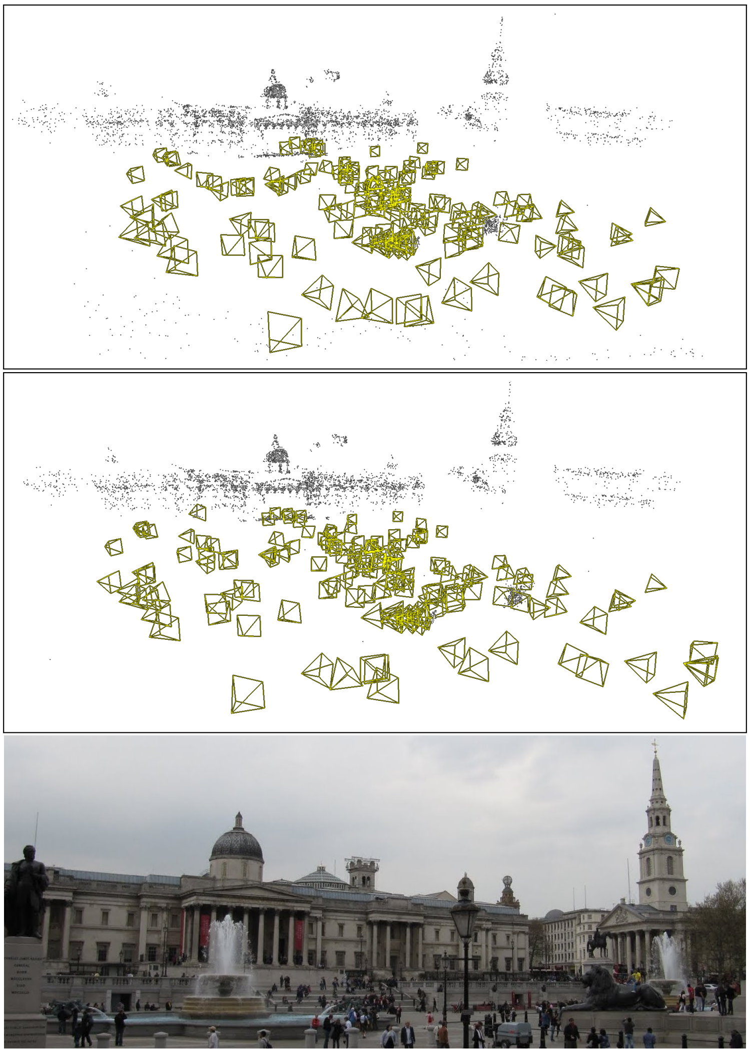


Figure 5.17: Camera poses and detail of the sparse scene structure obtained from the dataset *trafalgar-161*, using the following methods: **Top:** correction of the camera poses contained in the dataset with GEA, and re-estimation of the structure with the linear triangulation. **Middle:** incremental motion estimation and the linear structure triangulation from the pairwise matching information contained in the dataset. **Bottom:** picture from the original scene (Author: `https://ssl.panoramio.com/user/3029536`).

the initialized camera poses. As can be seen in these figures, the structure estimated with these camera poses is highly accurate.

The reduced number of initialized views for certain datasets could be explained by the poor pairwise matching connectivity defined between their views. As can be seen in figure 5.20 some of the datasets used in these tests

Dataset name	**Total views**	**Ground truth**				**Initialization**			
		error	PTS	views	PPV	error	PTS	views	PPV
medusa	18	0.60	2186	19	679.05	0.99	1715	18	632.44
leuvencastle	28	0.47	5256	28	1591.00	0.83	7168	28	2498.75
stmartin	124	2.03	41058	116	2191.03	0.53	33537	85	2628.94
hallwall	230	1.07	11612	230	685.77	1.13	7726	190	149.62
desktoplong	100	0.97	10586	100	1306.24	0.76	22768	100	1142.65
boxes8	91	0.79	9161	91	730.33	0.83	2236	25	702.84

Table 5.9: Comparison between the reconstructions obtained with the incremental motion estimation (initialization), and a classical incremental SfM method (ground truth). Trackings with 3 or less image projections and their corresponding points in the structure were ignored in the estimation of these values. **Error**: reprojection error for the reconstruction. **PTS**: 3D points initialized. **Views**: views initialized. **PPV**: image points per view which were included in one or more trackings.

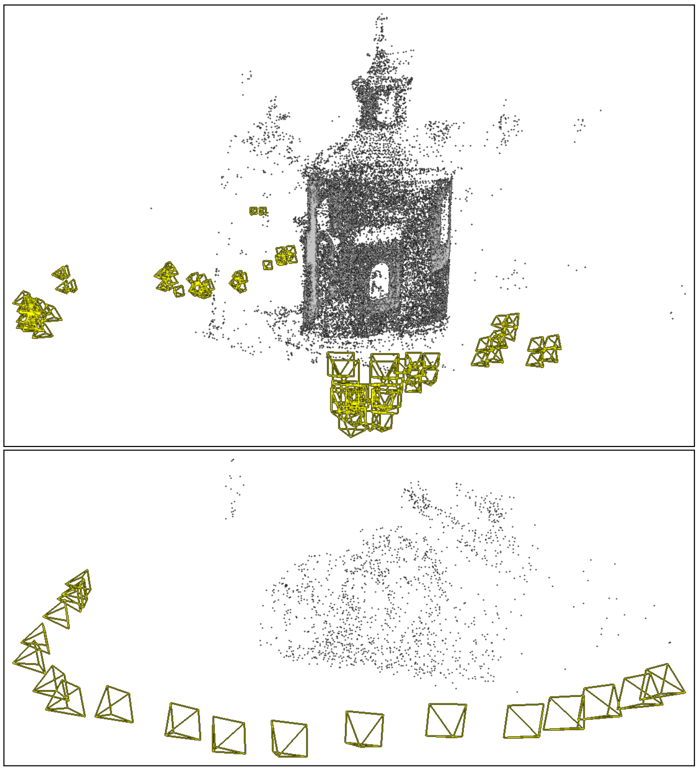


Figure 5.18: Sparse reconstructions obtained for the video sequences *stmartin* (**top**) and *medusa* (**bottom**) using the structureless incremental motion estimation procedure, and the linear structure triangulation.

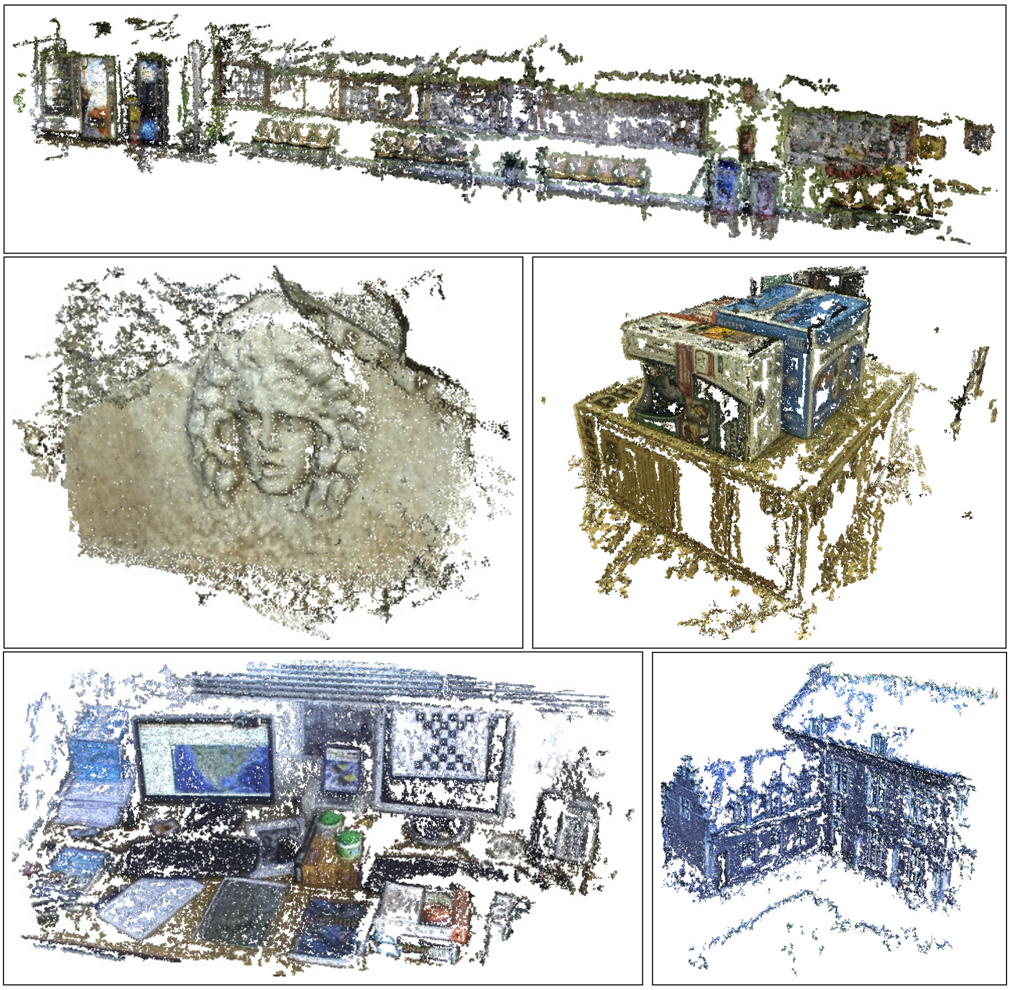


Figure 5.19: Dense scene structure obtained by PMVS2 using the camera poses estimated with the structureless incremental motion estimation procedure, for the datasets *hallwall* (**top row**), *medusa* (**middle row left**), *boxes8* (**middle row right**), *desktoplong* (**bottom row left**), and *leuvencastle* (**bottom row right**).

have a small number of view pairs corresponded with feature matchings, and some of these pairs have a small number of feature matchings. For this reason, the motion estimation procedure is not able to initialize a large fraction of the views in datasets such as *boxes8* or *stmartin.* In other datasets, the procedure provides a number of 3D points significantly smaller than those obtained with the classical incremental SfM method. However, the results obtained with the structureless incremental procedure could still be used as a starting point in a classical incremental SfM procedure, hence reducing the computation time required to obtain the full reconstruction.

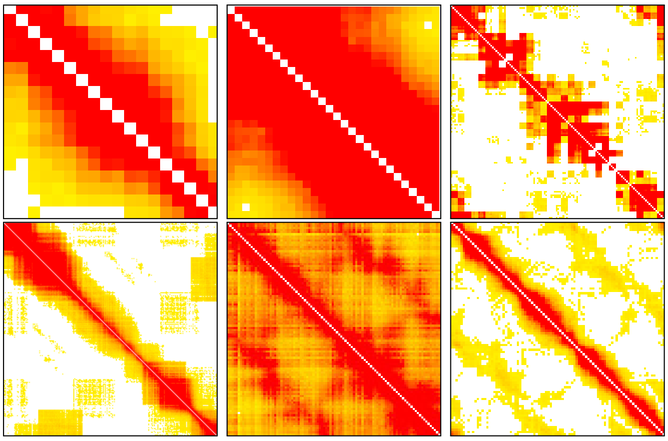


Figure 5.20: Feature matching correlation matrices for the views in the datasets *medusa*, *leuvencastle*, *stmartin* (**top row**), and *hallwall*, *desktoplong*, *boxes8* (**bottom row**). Each element in these matrices contains the number of feature correspondences detected by the sample consensus search method for a given view pair in the dataset. The red color in one of these elements indicates that the view pair is corresponded with more than 256 feature matchings. A color varying from yellow to red indicates that the view pair is corresponded respectively with 1 to 255 feature matchings. The white color indicates that the view pair is not correlated with feature matchings.

The figure shows the correlation between the amount of views and points correctly estimated with the reconstruction method, and the sparsity of the view correspondences graph. For those datasets containing a high pairwise view matching correlation, the averaging works better and the incremental motion estimation initializes a larger number of views and 3D points.

5.5 Closure

Despite optimizing an algebraic pairwise error, the experiments demonstrate that in general conditions GEA obtains highly accurate camera pose estimations, very close to the optimal configuration obtained with a geometric cost such as the reprojection error optimized by BA.

The GEA convergence speed suggest that the GEA cost error is more convex in the vicinity of the optimal configuration than the reprojection error. This fact is also supported by the significantly smaller number of free parameters and terms of the GEA cost error. Hence, the GEA cost is *easier* to optimize, in the sense that the basin for the optimal configuration is probably larger, and the optimization is smoother and less prone to divergence from the optimal configuration.

In practice, GEA can be used safely in most video sequences captured with hand-held or wearable cameras, video devices embedded in vehicles, or similar, as long as the camera does not move in straight line translations.

Critical motions will rarely affect the GEA optimization in practical reconstruction problems. As long as each view in the reconstruction has correspondences with other two non collinear views, even if they are only slightly non collinear, the problem of critical motion does not affect the GEA optimization. In monocular reconstruction applications, the camera motion will usually have slight or large deviations from the pure rectilinear translation.

The results discussed in this chapter also demonstrate the effectiveness of the robustification technique proposed for GEA in the previous chapter.

Chapter 6

Conclusions

6.1 Contributions

Structureless optimization methods, such as GEA, could become an important tool to reduce the computational cost in future reconstruction and visual odometry applications. We believe that this thesis provides solid arguments in this sense. Our main contributions are the following:

A bibliographic review of structureless BA methods.

We have discussed the state of the art in structureless BA methods for visual reconstruction. We have also reviewed alternative techniques for camera pose correction and initialization, such as classical BA, pose-graph optimization, or motion averaging.

A high-performance multiview structureless BA method based on algebraic epipolar constraints.

We have proposed GEA, a structureless BA approach which optimizes a multiple-view algebraic cost based on pairwise camera constraints using standard second-order optimization methods. We provided important computational details to obtain an efficient implementation of this optimization. In particular, we propose an efficient way to compress the feature correspondence information which dramatically reduces the computational cost of GEA. These computational advantages, combined with the nice mathematical properties of the proposed cost, give rise to a competitive alternative in a wide range of SfM applications.

Methods for using GEA in practical reconstruction applications.

We have reviewed several ways to use GEA in classical incremental SfM applications. Specifically, we proposed using it to speed up and prevent divergence in the intermediate steps of an incremental motion estimation procedure. The camera poses obtained this way can be used to estimate accurate dense or sparse reconstructions. The proposed incremental motion estimation is more efficient than classical incremental SfM methods which use BA and estimate the structure parameters.

Robustness to outliers is also an important issue in motion estimation algorithms. Our incremental method uses classical pairwise sample consensus to obtain epipolar geometries from the input matchings. Thanks to an additional cost robustification technique, GEA can deal with possible incorrect epipolar geometries, which could have survived due for example to perceptual aliasing. This robustification can be implemented with a simple modification in the Gauss-Newton step equation evaluation. Hence, it does not require changes in the GEA cost error, preserving its simplicity and computational advantages.

A thorough empirical evaluation of the GEA performance.

We have experimentally demonstrated that, under general circumstances, both the GEA and BA corrections obtain camera poses with a similar accuracy, with GEA requiring a significantly shorter computation time. We have also provided theoretical arguments which explain these two facts. Furthermore, we have evaluated the conditions which can degrade the GEA performance, specially the critical motion sequences which can reduce the accuracy of correction methods based on pairwise constraints. Our experiments showed that GEA can efficiently obtain accurate camera poses in arbitrarily large datasets, even in near-critical configurations.

6.2 Future work

In this section we describe what we believe are the most promising ideas to improve the results described in this thesis:

Uncalibrated motion estimation

An interesting future application of algebraic epipolar constraints might be uncalibrated motion estimation for 3D reconstruction from images with unknown intrinsic parameters. This would require using GEA with a camera parametrization where the focal distance, or other linear camera parameters are not fixed. In theory, the correction of the algebraic epipolar cost should still obtain accurate results under these circumstances.

Relative camera parametrization

In a similar way to [Sibley et al., 2009], we could use GEA with a relative camera pose parametrization to reduce drift error in real-time camera tracking applications. This way we could improve the quality of the *topometric* reconstructions obtained with respect to relative BA, as the number of views in the correction window could be increased without requiring a larger computation time in the cost optimization.

Improving incremental motion estimation

The results obtained with the structureless incremental motion estimation procedure could be improved by detecting more pairwise matchings between the input views, as well as using a better averaging method to initialize the camera poses. This way we could obtain camera pose configurations with a larger number of initialized views. In [Martinec and Pajdla, 2007] the authors use several kinds of features to obtain the largest possible number of matchings between the input images. This improves the averaging results obtained, by providing more relative motion constraints to estimate the camera pose. Furthermore, it is likely that using a geometric averaging method to estimate the camera location, instead of an algebraic one as our camera pose estimation does, would improve the view initialization results.

Structureless real-time motion estimation

The performance advantages of structureless motion estimation could be exploited in real-time visual odometry applications. For this purpose we could use the proposed motion estimation procedure, in combination with an efficient structureless method to detect image correspondences between the input keyframes. Image correspondences between keyframes close in the video sequence could be obtained either with a point tracking method such as KLT [Shi and Tomasi, 1994], using reduced or lightweight descriptors [Wagner et al., 2008; Calonder et al., 2010], or using a descriptor-less tracker which assumes a planar or euclidean image transference model [Tordoff and Murray, 2005; Rodríguez, López-de-Teruel and Ruiz, 2009]. Loop closing correspondences could be efficiently detected between keyframes which are distant in the video sequence, and have a small SAD or SSD correlation value.

Bibliography

Agarwal, S., N. Snavely, I. Simon, S. M. Seitz and R. Szeliski. 2009. Building Rome in a day. In *Proceedings of the 12st International Conference on Computer Vision.* IEEE.

Agarwal, S., N. Snavely, S. M. Seitz and R. Szeliski. 2010. Bundle adjustment in the large. In *Proceedings of the 11th European Conference on Computer Vision.* Springer-Verlag.

Akbarzadeh, A., J. Frahm, P. Mordohai, B. Clipp, C. Engels, D. Gallup, P. Merrell, M. Phelps, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewénius, R. Yang, G. Welch, H. Towles, D. Nistér and M. Pollefeys. 2006. Towards Urban 3D Reconstruction from Video. In *Proceedings of the 3rd International Symposium on 3D Data Processing, Visualization, and Transmission.* IEEE Computer Society.

Anguelov, D., C. Dulong, D. Filip, C. Frueh, S. Lafon, R. Lyon, A. S. Ogale, L. Vincent and J. Weaver. 2010. "Google Street View: Capturing the World at Street Level." *Computer* (6).

Bailey, T., J. I. Nieto, J. E. Guivant, M. Stevens and E. M. Nebot. 2006. Consistency of the EKF-SLAM Algorithm. In *Proceedings of the International conference on Intelligent Robots and Systems.* IEEE.

Bay, H., T. Tuytelaars and L. V. Gool. 2006. Surf: Speeded up robust features. In *Proceedings of the 9th European Conference on Computer Vision.*

Bookstein, F. L. 1979. "Fitting conic sections to scattered data." *International Journal of Computer Graphics and Image Processing* (1).

Bougnoux, S. 1998. From Projective to Euclidean Space Under any Practical Situation, a Criticism of Self-Calibration. In *Proceedings of the 6th International Conference on Computer Vision.*

Brown, D. C. 1976. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences.* Number 3.

Brown, M. and D. G. Lowe. 2005. “Unsupervised 3D Object Recognition and Reconstruction in Unordered Datasets.” *Proceedings of the 2005 International Conference on 3D Digital Imaging and Modeling* .

Byrod, M. and K. Astrom. 2010. Conjugate gradient bundle adjustment. In *Proceedings of the 11th European Conference on Computer Vision.* Springer-Verlag.

Calonder, M., V. Lepetit, C. Strecha and P. Fua. 2010. BRIEF: binary robust independent elementary features. In *Proceedings of the 11th European Conference on Computer Vision.*

Canny, J. 1986. “A Computational Approach to Edge Detection.” *International Journal of ACM Transactions on Pattern Analysis and Machine Intelligence* (6).

Castle, R. O., G. Klein and D. W. Murray. 2008. Video-rate Localization in Multiple Maps for Wearable Augmented Reality. In *Proceedings of the 12th International Symposium on Wearable Computers.*

Chen, Y., T. A. Davis, W. W. Hager and S. Rajamanickam. 2008. “Algorithm 887: CHOLMOD, Supernodal Sparse Cholesky Factorization and Update/Downdate.” *International Journal of ACM Transactions on Mathematical Software* .

Chum, O. and J. Matas. 2005. Matching with PROSAC - Progressive Sample Consensus. In *Proceedings of the 18th Computer Vision and Pattern Recognition.*

Cornelis, N., B. Leibe, K. Cornelis and L. Gool. 2008. “3D Urban Scene Modeling Integrating Recognition and Reconstruction.” *International Journal of Computer Vision* (2-3).

Crandall, D., A. Owens, N. Snavely and D. P. Huttenlocher. 2011. Discrete-Continuous Optimization for Large-Scale Structure from Motion. In *Proceedings of the 24th Computer Vision and Pattern Recognition.*

Crowley, J. L. 1989. World modeling and position estimation for a mobile robot using ultrasonic ranging. In *Proceedings of the International Conference on Robotics and Automation.* IEEE.

Cummins, M. and P. Newman. 2007. Probabilistic Appearance Based Navigation and Loop Closing. In *Proceedings of the International Conference on Robotics and Automation.*

Cummins, M. and P. Newman. 2008. “FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance.” *International Journal of Robotics Research* .

d. Hengel, A. V., R. Hill, B. Ward and A. Dick. 2009. In situ image-based modeling. In *Proceedings of the 8th International Symposium on Mixed and Augmented Reality.* IEEE.

Davis, T. 2012. “CSparse: a small yet feature-rich sparse matrix package.” `http://www.cise.ufl.edu/research/sparse/CSparse/`.

Davison, A. J. 2003. Real-Time Simultaneous Localisation and Mapping with a Single Camera. In *Proceedings of the 9th International Conference on Computer Vision.* IEEE Computer Society.

Davison, A. J., I. D. Reid, N. D. Molton and O. Stasse. 2007. “MonoSLAM: Real-Time Single Camera SLAM.” *International Journal of ACM Transactions on Pattern Analysis and Machine Intelligence* (6).

Debevec, P. E., C. J. Taylor and J. Malik. 1996. Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. In *Proceedings of the 23rd annual conference on Computer Graphics and Interactive Techniques.* ACM.

Dellaert, F. and M. Kaess. 2006. “Square Root SAM: Simultaneous localization and mapping via square root information smoothing.” *International Journal of Robotics Research* .

Dick, A. R., P. H. S. Torr and R. Cipolla. 2004. “Modelling and Interpretation of Architecture from Several Images.” *International Journal of Computer Vision* (2).

Eade, E. and T. Drummond. 2006. Scalable Monocular SLAM. In *Proceedings of the 19th Computer Vision and Pattern Recognition.* IEEE Computer Society.

Eade, E. and T. Drummond. 2008. Unified loop closing and recovery for real-time monocular SLAM. In *Proceedings of the 19th British Matchine Vision Conference.*

Engels, C., H. Stewénius and D. Nistér. 2006. Bundle adjustment rules. In *Photogrammetric Computer Vision.*

Estrada, J. N. C. and J. D. Tardós. 2005. “Hierarchical SLAM: real-time accurate mapping of large environments.” *IEEE Transactions on Robotics* (4).

Eudes, R. and M. Lhuillier. 2009. Error propagations for local bundle adjustment. In *Proceedings of the 22th Computer Vision and Pattern Recognition*.

Eustice, R. M., H. Singh and W. H. Ma. 2005. Exactly sparse delayed-state filters. In *Proceedings of the International Conference on Robotics and Automation*.

Eustice, R., O. Pizarro and H. Singh. 2004. Visually Augmented Navigation in an Unstructured Environment Using a Delayed State History. In *Proceedings of the International Conference on Robotics and Automation*.

Fischler, M. A. and R. C. Bolles. 1981. “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography.” *Communications of the ACM* (6).

Fitzgibbon, A. W. and A. Zisserman. 1998. Automatic Camera Recovery for Closed or Open Image Sequences. In *Proceedings of the 5th European Conference on Computer Vision*.

Frahm, J., P. Fite-Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y. Jen, E. Dunn, B. Clipp, S. Lazebnik and M. Pollefeys. 2010. Building Rome on a cloudless day. In *Proceedings of the 11th European Conference on Computer Vision*. Springer-Verlag.

Furukawa, Y., B. Curless, S. M. Seitz and R. Szeliski. 2010. Towards Internet-scale multi-view stereo. In *Proceedings of the 23rd Computer Vision and Pattern Recognition*.

Furukawa, Y. and J. Ponce. 2007. Accurate, Dense, and Robust Multi-View Stereopsis. In *Proceedings of the 20th Computer Vision and Pattern Recognition*.

Furukawa, Y. and J. Ponce. 2012. “Patch-based Multi-view Stereo Software (PMVS - Version 2).” `http://http://grail.cs.washington.edu/software/pmvs/`. [Online; accessed April-2012].

Gonzalez, J., A. Ollero and A. Reina. 1994. Map building for a mobile robot equipped with a 2D laser rangefinder. In *Proceedings of the International Conference on Robotics and Automation*.

Govindu, V. M. 2001. Combining Two-view Constraints For Motion Estimation. In *Proceedings of the Computer Vision and Pattern Recognition.*

Govindu, V. M. 2004. Lie-Algebraic Averaging for Globally Consistent Motion Estimation. In *Proceedings of the 17th Computer Vision and Pattern Recognition.*

Granshaw, S. I. 1980. "Bundle adjustment methods in engineering photogrammetry." *The Photogrammetric Record* (56).

Grisetti, G., C. Stachniss, S. Grzonka and W. Burgard. 2007. A tree parameterization for efficiently computing maximum likelihood maps using gradient descent. In *Proceedings of the 2007 Robotics Science and Systems.*

Hamilton, W. R. 1844. "On Quaternions; or on a new System of Imaginaries in Algebra." *The Philosophical Magazine* .

Hartley, R. and A. Zisserman. 2003. *Multiple View Geometry in Computer Vision.* 2 ed. Cambridge University Press.

Hartley, R. and F. Schaffalitzky. 2003. PowerFactorization: 3D reconstruction with missing or uncertain data. In *Proceedings of Australia-Japan Advanced Workshop on Computer Vision.*

Hartley, R. and F. Schaffalitzky. 2004. L_∞ Minimization in Geometric Reconstruction Problems. In *Proceedings of the 17th Computer Vision and Pattern Recognition.*

Hartley, R. I. 1992. Estimation of Relative Camera Positions for Uncalibrated Cameras. In *Proceedings of the 2nd European Conference on Computer Vision.*

Hartley, R. I. 1994*a*. Euclidean reconstruction from uncalibrated views. In *Proceedings of the Second Joint European - US Workshop on Applications of Invariance in Computer Vision.* Springer-Verlag.

Hartley, R. I. 1994*b*. "Projective reconstruction and invariants from multiple images." *International Journal of ACM Transactions on Pattern Analysis and Machine Intelligence* (10).

Hartley, R. I. 1997. "In Defense of the Eight-Point Algorithm." *International Journal of ACM Transactions on Pattern Analysis and Machine Intelligence* .

Hartley, R. I. 1998*a*. Computation of the Quadrifocal Tensor. In *Proceedings of the 5th European Conference on Computer Vision.*

Hartley, R. I. 1998*b*. Minimizing Algebraic Error in Geometric Estimation Problems. In *Proceedings of the 6th International Conference on Computer Vision.*

Hartley, R. I. and P. F. Sturm. 1997. "Triangulation." *Journal of Computer Vision and Image Understanding* (2).

Hartley, R. I., R. Gupta and T. Chang. 1992. Stereo from Uncalibrated Cameras. In *Proceedings of the Computer Vision and Pattern Recognition.*

Hartley, R. I. and R. Kaucic. 2002. Sensitivity of Calibration to Principal Point Position. In *Proceedings of the 7th European Conference on Computer Vision.*

Hartley, R., K. Aftab and J. Trumpf. 2011. L_1 rotation averaging using the Weiszfeld algorithm. In *Proceedings of the 24th Computer Vision and Pattern Recognition.*

Heyden, A. and K. Åström. 1997. "Algebraic Properties of Multilinear Constraints." *International Journal of Mathematical Methods in the Applied Sciences* (13).

Ho, K. L. and P. Newman. 2007. "Detecting Loop Closure with Scene Sequences." *International Journal of Computer Vision* (3).

Holmes, S., G. Sibley, G. Klein and D. W. Murray. 2009. A relative frame representation for fixed-time bundle adjustment in SfM. In *Proceedings of the International Conference on Robotics and Automation.* IEEE Press.

Horn, B. K. P. 1987. "Closed-form solution of absolute orientation using quaternions." *Journal of the Optical Society of America* .

Huber, P. J. 1964. "Robust Estimation of a Location Parameter." *The Annals of Mathematical Statistics* (1).

Ila, V., J. Andrade-cetto and A. Sanfeliu. 2007. Outdoor Delayed-state Visually Augmented Odometry. In *Proceedings of the 6th IFAC Symposium on Intelligent Autonomous Vehicles.*

Ila, V., J. Andrade-Cetto, R. Valencia and A. Sanfeliu. 2007. Vision-based loop closing for delayed state robot mapping. In *Proceedings of the International Conference on Intelligent Robots and Systems.* IEEE.

Ila, V., J. M. Porta and J. Andrade-Cetto. 2009. Reduced state representation in delayed-state SLAM. In *Proceedings of the International conference on Intelligent Robots and Systems.* IEEE Press.

Indelman, V., R. Roberts, C. Beall and F. Dellaert. 2012. Incremental Light Bundle Adjustment. In *Proceedings of the 23rd British Matchine Vision Conference.*

Indelman, V., R. Roberts and F. Dellaert. 2013. Probabilistic Analysis of Incremental Light Bundle Adjustment. In *IEEE Workshop on Robot Vision (WoRV).*

Intel. 2012. "Math Kernel Library." `http://developer.intel.com/software/products/mkl/`.

Irschara, A., C. Zach, J. Frahm and H. Bischof. 2009. From structure-from-motion point clouds to fast location recognition. In *Proceedings of the 22th Computer Vision and Pattern Recognition.* IEEE.

Izadi, S., R. A. Newcombe, D. Kim, O. Hilliges, D. Molyneaux, S. Hodges, P. Kohli, J. Shotton, A. J. Davison and A. Fitzgibbon. 2011. KinectFusion: real-time dynamic 3D surface reconstruction and interaction. In *Proceedings of the ACM SIGGRAPH 2011 Talks.* ACM.

Jeong, Y., D. Nistér, D. Steedly, R. Szeliski and I. Kweon. 2011. "Pushing the Envelope of Modern Methods for Bundle Adjustment." *International Journal of ACM Transactions on Pattern Analysis and Machine Intelligence* (PrePrints).

Jian, Y., D. C. Balcan and F. Dellaert. 2011. Generalized subgraph preconditioners for large-scale bundle adjustment. In *Proceedings of the 13st International Conference on Computer Vision.*

Julier, S. J. and J. K. Uhlmann. 2001. "A counter example to the theory of simultaneous localization and map building." *Proceedings of the International Conference on Robotics and Automation* (1).

Julier, S. J., Jeffrey and K. Uhlmann. 2004. Unscented Filtering and Nonlinear Estimation. In *Proceedings of the IEEE.*

Kaess, M., A. Ranganathan and F. Dellaert. 2008. "iSAM: Incremental Smoothing and Mapping." *IEEE Transactions on Robotics* (6).

Kahl, F. 2005. Multiple View Geometry and the L_∞ Norm. In *Proceedings of the 10th International Conference on Computer Vision.*

Kahl, F. and B. Triggs. 1999. Critical Motions in Euclidean Structure from Motion. In *Proceedings of the Computer Vision and Pattern Recognition*. IEEE Computer Society.

Kanatani, K., Y. Sugaya and H. Niitsuma. 2008. Triangulation from Two Views Revisited: Hartley-Sturm vs. Optimal Correction. In *Proceedings of the 19th British Matchine Vision Conference*.

Kaucic, R., N. Dano and R. Hartley. 2001. Plane-based Projective Reconstruction. In *Proceedings of the 8th International Conference on Computer Vision*.

Klein, G. and D. Murray. 2007. Parallel Tracking and Mapping for Small AR Workspaces. In *Proceedings of the 6th International Symposium on Mixed and Augmented Reality*.

Klein, G. and D. Murray. 2009. Parallel Tracking and Mapping on a Camera Phone. In *Proceedings of the 8th International Symposium on Mixed and Augmented Reality*.

Konolige, K. 2010. Sparse Sparse Bundle Adjustment. In *Proceedings of the 21st British Matchine Vision Conference*.

Konolige, K. and M. Agrawal. 2008. "FrameSLAM: From Bundle Adjustment to Real-Time Visual Mapping." *IEEE Transactions on Robotics* (5).

Kummerle, R., G. Grisetti, H. Strasdat, K. Konolige and W. Burgard. 2011. g2o: A General Framework for Graph Optimization. In *Proceedings of the International Conference on Robotics and Automation*.

Kushal, A. and S. Agarwal. 2012. Visibility Based Preconditioning for Bundle Adjustment. In *Proceedings of the 25th Computer Vision and Pattern Recognition*.

Labs, M. L. 2012. "PhotoSynth: Capture your world in 3D.". [Online; accessed April-2012].
URL: *http://www.photosynth.com*

Lategahn, H., A. Geiger, B. Kitt and C. Stiller. 2012. Motion-without-Structure: Real-time Multipose Optimization for Accurate Visual Odometry. In *Proceedings of IEEE Intelligent Vehicles Symposium*.

Leonard, J., H. Durrant-Whyte and I. J. Cox. 1990. Dynamic map building for autonomous mobile robot. In *Proceedings of the IEEE International Workshop on Intelligent Robots and Systems*. IEEE.

Leonard, J. J., H. J. S. Feder, H. Jacob and S. Feder. 1999. “Decoupled Stochastic Mapping.” *IEEE Journal of Oceanic Engineering* .

Leonard, J. and P. Newman. 2003. Consistent, convergent, and constant-time SLAM. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence.* Morgan Kaufmann Publishers Inc.

Lepetit, V., F. Moreno-Noguer and P. Fua. 2009. “EPnP: An Accurate O(n) Solution to the PnP Problem.” *International Journal of Computer Vision* (2).

Lepetit, V. and P. Fua. 2005. “Monocular model-based 3D tracking of rigid objects.” *Foundations and Trends in Computer Graphics and Vision* (1).

Levenberg, K. 1944. “A method for the solution of certain problems in least squares.” *Quarterly of Applied Mathematics* .

Li, H. 2007. A practical algorithm for L_∞ triangulation with outliers. In *Proceedings of the 20th Computer Vision and Pattern Recognition.*

Li, H. 2010. Multi-view structure computation without explicitly estimating motion. In *Proceedings of the 23rd Computer Vision and Pattern Recognition.* IEEE.

Li, H. and R. Hartley. 2006. Five-Point Motion Estimation Made Easy. In *Proceedings of the 18th International Conference on Pattern Recognition.* Proceedings of the 18th International Conference on Pattern Recognition IEEE Computer Society.

Lindeberg, T. 1993. “Detecting Salient Blob-Like Image Structures and Their Scales with a Scale-Space Primal Sketch: A Method for Focus-of-Attention.” *International Journal of Computer Vision* .

Lindstrom, P. 2010. Triangulation made easy. In *Proceedings of the 23rd Computer Vision and Pattern Recognition.* IEEE.

Lippman, A. 1980. Movie-maps: An application of the optical videodisc to computer graphics. In *Proceedings of the 7th annual conference on Computer Graphics and Interactive Techniques.* ACM.

Longuet-Higgins, H. C. 1981. “A computer algorithm for reconstructing a scene from two projections.” *Nature* .

Lourakis, M. and A. Argyros. 2005. Is Levenberg-Marquardt the Most Efficient Optimization Algorithm for Implementing Bundle Adjustment? In *Proceedings of the 10th International Conference on Computer Vision.*

Lourakis, M. and A. Argyros. 2009. "SBA: A software package for generic sparse bundle adjustment." *International Journal of ACM Transactions on Mathematical Software* (1).

Lowe, D. G. 2004. "Distinctive Image Features from Scale-Invariant Keypoints." *International Journal of ACM Transactions on Pattern Analysis and Machine Intelligence* (2).

Lu, F. and E. Milios. 1997. "Globally Consistent Range Scan Alignment for Environment Mapping." *Journal of Autonomous Robots* (4).

Ma, Y., S. Soatto, J. Košecká and S. Sastry. 2000. "Euclidean Reconstruction and Reprojection Up to Subgroups." *International Journal of Computer Vision* .

Martinec, D. and T. Pajdla. 2007. Robust Rotation and Translation Estimation in Multiview Reconstruction. In *Proceedings of the 20th Computer Vision and Pattern Recognition.* IEEE Computer Society.

Matas, J., O. Chum, M. Urban and T. Pajdla. 2002. Robust Wide Baseline Stereo from Maximally Stable Extremal Regions. In *Proceedings of the 13rd British Matchine Vision Conference.* British Machine Vision Association.

Mikolajczyk, K., T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir and L. V. Gool. 2005. "A Comparison of Affine Region Detectors." *International Journal of Computer Vision* (1-2).

Monagan, M. B., K. O. Geddes, K. Heal, G. Labahn, S. M. Vorkoetter, J. McCarron and P. DeMarco. 2005. *Maple 10 Programming Guide.* Maplesoft.

Montemerlo, M. and S. Thrun. 2007. *FastSLAM: A Scalable Method for the Simultaneous Localization and Mapping Problem in Robotics.* Springer-Verlag New York, Inc.

Montemerlo, M., S. Thrun, D. Koller and B. Wegbreit. 2002. FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence.*

Moreno-Noguer, F. and J. M. Porta. 2011. Probabilistic simultaneous pose and non-rigid shape recovery. In *Proceedings of the 24th Computer Vision and Pattern Recognition.*

Mouragnon, E., M. Lhuillier, M. Dhome, F. Dekeyser and P. Sayd. 2006. Real Time Localization and 3D Reconstruction. In *Proceedings of the 19th Computer Vision and Pattern Recognition.* IEEE Computer Society.

Newcombe, R. A. and A. J. Davison. 2010. Live Dense Reconstruction with a Single Moving Camera. In *Proceedings of the 23rd Computer Vision and Pattern Recognition.* Number 13-18 June 2010.

Newcombe, R. A., S. J. Lovegrove and A. J. Davison. 2011. "DTAM: Dense tracking and mapping in real-time." *Proceedings of the 13st International Conference on Computer Vision* .

Ni, K., D. Steedly and F. Dellaert. 2007. Out-of-Core Bundle Adjustment for Large-Scale 3D Reconstruction. In *Proceedings of the 11st International Conference on Computer Vision.*

Nistér, D. 2000. Reconstruction from Uncalibrated Sequences with a Hierarchy of Trifocal Tensors. In *Proceedings of the 6th European Conference on Computer Vision.* Springer-Verlag.

Nistér, D. 2004. "An Efficient Solution to the Five-Point Relative Pose Problem." *International Journal of ACM Transactions on Pattern Analysis and Machine Intelligence* .

Nistér, D. and H. Stewénius. 2008. Linear Time Maximally Stable Extremal Regions. In *Proceedings of the 10th European Conference on Computer Vision.* Springer-Verlag.

Nistér, D., O. Naroditsky and J. R. Bergen. 2004. Visual Odometry. In *Proceedings of the 17th Computer Vision and Pattern Recognition.*

Olson, E., J. Leonard and S. Teller. 2006. Fast iterative alignment of pose graphs with poor initial estimates. In *Proceedings of the International Conference on Robotics and Automation.*

Pan, Q., G. Reitmayr and T. Drummond. 2009. ProFORMA: Probabilistic Feature-based On-line Rapid Model Acquisition. In *Proceedings of the 20th British Matchine Vision Conference.*

Perriollat, M., R. I. Hartley and A. Bartoli. 2011. "Monocular Template-based Reconstruction of Inextensible Surfaces." *International Journal of Computer Vision* (2).

Pollefeys, M., F. Verbiest and L. J. V. Gool. 2002. Surviving Dominant Planes in Uncalibrated Structure and Motion Recovery. In *Proceedings of the 7th European Conference on Computer Vision.*

Pollefeys, M., L. V. Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops and R. Koch. 2004. "Visual Modeling with a Hand-Held Camera." *International Journal of Computer Vision* .

Pollefeys, M., R. Koch and L. V. Gool. 1999. "Self-Calibration and Metric Reconstruction Inspite of Varying and Unknown Intrinsic Camera Parameters." *International Journal of Computer Vision* .

Pollefeys, M., R. Koch, M. Vergauwen and L. V. Gool. 1998. Metric 3D surface reconstruction from uncalibrated image sequences. In *3D Structure from Multiple Images of Large Scale Environments. LNCS series.* Springer-Verlag.

Quan, L. and Z. Lan. 1999. "Linear N-Point Camera Pose Determination." *International Journal of ACM Transactions on Pattern Analysis and Machine Intelligence* (8).

Quigley, M., K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler and A. Y. Ng. 2009. ROS: an open-source Robot Operating System. In *ICRA Workshop on Open Source Software.*

Ranganathan, A., M. Kaess and F. Dellaert. 2007. Loopy SAM. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence.*

Robert, L. and O. D. Faugeras. 1995. "Relative 3D positioning and 3D convex hull computation from a weakly calibrated stereo pair." *International Journal of Image and Vision Computing* (3).

Rodríguez, A., P. López-de-Teruel and A. Ruiz. 2009. Real-Time Descriptorless Feature Tracking. In *Proceedings of the 15th International Conference on Image Analysis and Processing.* Springer-Verlag.

Rodríguez, A., P. López-de-Teruel and A. Ruiz. 2011*a*. GEA optimization for live structureless motion estimation. In *Proceedings of ICCV'11 workshop on the 1st Live Dense Reconstruction from Moving Cameras.*

Rodríguez, A., P. López-de-Teruel and A. Ruiz. 2011*b*. Reduced epipolar cost for accelerated incremental SfM. In *Proceedings of the 24th Computer Vision and Pattern Recognition.*

Rodríguez, A., P. López-de-Teruel, A. Ruiz, G. García and L. Fernández. 2008. QVision, a Development Framework for Real-time Computer Vision and Image Processing Research. In *Proceedings of the 2008 International Conference on Image Processing, Computer Vision, and Pattern Recognition.* CSREA Press.

Ros, G., J. Guerrero, A. D. Sappa, D. Ponsa and A. M. López. 2013. Pose initialization via Lie groups and Lie algebras optimization. In *Proceedings of the International Conference on Robotics and Automation.*

Rosten, E. and T. Drummond. 2006. Machine learning for high-speed corner detection. In *Proceedings of the 9th European Conference on Computer Vision.*

Rother, C. and S. Carlsson. 2001. Linear Multi View Reconstruction and Camera Recovery. In *Proceedings of the 8th International Conference on Computer Vision.*

Ruiz, A., O. Cánovas, R. Rubio and P. López-de-Teruel. 2011. Using SIFT and WiFi signals to provide location-based services for smartphones. In *Proceedings of the 8th International Conference on Mobile and Ubiquitous Systems:Computing, Networking and Services.*

Ruiz, A., P. López-de-Teruel and G. García. 2002. A Note on Principal Point Estimability. In *Proceedings of the 16th International Conference on Pattern Recognition.*

Salzmann, M., F. Moreno-Noguer, V. Lepetit and P. Fua. 2008. Closed-Form Solution to Non-rigid 3D Surface Registration. In *Proceedings of the 10th European Conference on Computer Vision.*

Sampson, P. D. 1982. "Fitting Conic Sections to 'Very Scattered' Data: An Iterarive Refinement of the Bookstein Algorithm." *International Journal of Computer Graphics and Image Processing* (1).

Sastry, S. 1999. Optimization Criteria, Sensitivity and Robustness of Motion and Structure Estimation. In *Proceedings of ICCV'99 workshop on Vision Algorithms.*

Seitz, S. M., B. Curless, J. Diebel, D. Scharstein and R. Szeliski. 2006. A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms. In *Proceedings of the 19th Computer Vision and Pattern Recognition.* IEEE Computer Society.

Shewchuk, J. R. 1994. An Introduction to the Conjugate Gradient Method Without the Agonizing Pain. Technical report.

Shi, J. and C. Tomasi. 1994. Good features to track. In *Proceedings of the Computer Vision and Pattern Recognition.* IEEE.

Shum, H., Z. Zhang and Q. Ke. 1999. “Efficient Bundle Adjustment with Virtual Key Frames: A Hierarchical Approach to Multi-Frame Structure from Motion.” *Proceedings of the Computer Vision and Pattern Recognition* .

Sibley, G., C. Mei, I. D. Reid and P. M. Newman. 2010*a*. “Vast-scale Outdoor Navigation Using Adaptive Relative Bundle Adjustment.” *International Journal of Robotics Research* (8).

Sibley, G., C. Mei, I. Reid and P. Newman. 2009. Adaptive Relative Bundle Adjustment. In *Proceedings of the 2009 Robotics Science and Systems.*

Sibley, G., C. Mei, I. Reid and P. Newman. 2010*b*. “Vast-scale Outdoor Navigation Using Adaptive Relative Bundle Adjustment.” *International Journal of Robotics Research* (8).

Sim, K. and R. Hartley. 2006*a*. Recovering Camera Motion Using L_∞ Minimization. In *Proceedings of the 19th Computer Vision and Pattern Recognition.*

Sim, K. and R. Hartley. 2006*b*. Removing Outliers Using The L_∞ Norm. In *Proceedings of the 19th Computer Vision and Pattern Recognition.*

Simmons, R. and S. Koenig. 1995. Probabilistic Robot Navigation in Partially Observable Environments. In *Proceedings of the International Joint Conference on Artificial Intelligence.*

Sinha, S. N., D. Steedly and R. Szeliski. 2010. A multi-stage linear approach to structure from motion. In *Proceedings of the 11th European Conference on Computer Vision.*

Slama, C. C., C. Theurer and S. W. Henriksen. 1980. *Manual of photogrammetry.* American Society of Photogrammetry.

Smith, R., M. Self and P. Cheeseman. 1986. Estimating Uncertain Spatial Relationships in Robotics. In *Proceedings of the 2nd Annual Conference on Uncertainty in Artificial Intelligence.* AUAI Press.

Smith, R., M. Self and P. Cheeseman. 1988. A stochastic map for uncertain spatial relationships. In *Proceedings of the 4th International Symposium on Robotics Research*, ed. O Faugeras and GEditors Giralt. MIT Press.

Snavely, N. 2012. “Bundler: Structure from Motion (SfM) for Unordered Image Collections.” `http://http://phototour.cs.washington.edu/bundler/`. [Online; accessed September-2012].

Snavely, N., S. M. Seitz and R. Szeliski. 2006. Photo Tourism: exploring photo collections in 3D. In *Proceedings of the 33rd annual conference on Computer Graphics and Interactive Techniques.* ACM.

Snavely, N., S. M. Seitz and R. Szeliski. 2008*a*. “Modeling the World from Internet Photo Collections.” *International Journal of Computer Vision* .

Snavely, N., S. M. Seitz and R. Szeliski. 2008*b*. “Skeletal graphs for efficient structure from motion.” *Proceedings of the 21th Computer Vision and Pattern Recognition* .

Steedly, D. and I. Essa. 2001*a*. Propagation of Innovative Information in Non-Linear Least-Squares Structure from Motion. In *Proceedings of the 8th International Conference on Computer Vision.*

Steedly, D. and I. Essa. 2001*b*. Propagation of Innovative Information in Non-Linear Least-Squares Structure from Motion. In *Proceedings of the 8th International Conference on Computer Vision.*

Steffen, R., J. Frahm and W. Förstner. 2010. Relative Bundle Adjustment based on Trifocal Constraints. In *Proceedings of the 11th European Conference on Computer Vision.*

Stewénius, H., C. Engels and D. Nistér. 2006. “Recent Developments on Direct Relative Orientation.” *Journal of Photogrammetry and Remote Sensing* .

Strasdat, H., A. J. Davison, J. M. M. Montiel and K. Konolige. 2011. Double window optimisation for constant time visual SLAM. In *Proceedings of the 13st International Conference on Computer Vision.* IEEE Computer Society.

Strasdat, H., J. M. M. Montiel and A. Davison. 2010*a*. Scale Drift-Aware Large Scale Monocular SLAM. In *Proceedings of the 2009 Robotics Science and Systems*.

Strasdat, H., J. M. M. Montiel and A. J. Davison. 2010*b*. Real-time monocular SLAM: Why filter? In *Proceedings of the International Conference on Robotics and Automation*. IEEE.

Strasdat, H., J. M. M. Montiel and A. J. Davison. 2012. "Editors Choice Article: Visual SLAM: Why filter?" *International Journal of Image and Vision Computing* (2).

Sturm, P. 1997. Critical Motion Sequences for Monocular Self-Calibration and Uncalibrated Euclidean Reconstruction. In *Proceedings of the Computer Vision and Pattern Recognition*. IEEE Computer Society.

Sturm, P. and B. Triggs. 1996. "A factorization based algorithm for multi-image projective structure and motion." *Proceedings of the 4th European Conference on Computer Vision* .

Sünderhauf, N., M. Obst, G. Wanielik and P. Protzel. 2012. Multipath Mitigation in GNSS-Based Localization using Robust Optimization. In *Proceedings of Intelligent Vehicles Symposium; Workshop of Navigation, Perception, Accurate Positioning and Mapping for Intelligent Vehicles*.

Sünderhauf, N. and P. Protzel. 2012. Towards a Robust Back-End for Pose Graph SLAM. In *Proceedings of the International Conference on Robotics and Automation*.

Thrun, S., M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. v. Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian and P. Mahoney. 2006. "Stanley: The robot that won the DARPA Grand Challenge: Research Articles." *Journal of Robotic Systems* .

Thrun, S., W. Burgard and D. Fox. 2001. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*.

Thrun, S., Y. Liu, D. Koller, A. Y. Ng, Z. Ghahramani and H. Durrant-Whyte. 2004. "Simultaneous localization and mapping with sparse extended information filters." *International Journal of Robotics Research* (7/8).

Tomasi, C. and T. Kanade. 1992. “Shape and motion from image streams under orthography: a factorization method.” *International Journal of Computer Vision* (2).

Tordoff, B. J. and D. W. Murray. 2005. “Guided-MLESAC: Faster image transform estimation by using mathcing priors.” *International Journal of ACM Transactions on Pattern Analysis and Machine Intelligence* .

Torr, P. H. S. and A. Zisserman. 1997. “Robust Parametrization and Computation of the Trifocal Tensor.” *International Journal of Image and Vision Computing* (8).

Triggs, B. 1997. Autocalibration and the absolute quadric. In *Proceedings of the Computer Vision and Pattern Recognition.*

Triggs, B., P. F. McLauchlan, R. I. Hartley and A. W. Fitzgibbon. 2000. Bundle Adjustment - A Modern Synthesis. In *Proceedings of ICCV'99 workshop on Vision Algorithms.* Springer-Verlag.

Tsai, R. Y. and T. S. Huang. 1984. Uniqueness and Estimation of Three Dimensional Motion Parameters of Rigid Objects With Curved Surfaces. In *International Journal of ACM Transactions on Pattern Analysis and Machine Intelligence.*

Tuytelaars, T. and K. Mikolajczyk. 2008. *Local Invariant Feature Detectors: A Survey.*

Vidal, R., Y. Ma, S. Hsu and S. Sastry. 2001. Optimal Motion Estimation from Multiview Normalized Epipolar Constraint. In *Proceedings of the 8th International Conference on Computer Vision.*

Wagner, D., G. Reitmayr, A. Mulloni, T. Drummond and D. Schmalstieg. 2008. Pose tracking from natural features on mobile phones. In *Proceedings of the 7th International Symposium on Mixed and Augmented Reality.*

Watman, C., D. Austin, N. Barnes, G. Overett and S. Thompson. 2004. Fast sum of absolute differences visual landmark detector. In *Proceedings of the International Conference on Robotics and Automation.*

Weiszfeld, E. and F. Plastria. 2009. “On the point for which the sum of the distances to n given points is minimum.” *The Annals of Operations Research* (1).

Wendel, A., M. Maurer, G. Graber, T. Pock and H. Bischof. 2012. Dense Reconstruction On-the-Fly. In *Proceedings of the 25th Computer Vision and Pattern Recognition.*

Whitcomb, L., D. Yoerger, H. Singh and J. Howland. 1999. Advances in Underwater Robot Vehicles for Deep Ocean Exploration: Navigation, Control, and Survey Operations. In *Navigation, Control and Survery Operations in The 9th International Symposium on Robotics Research.* SpringerVerlag.

Wikipedia. 2012*a*. "Mathematica — Wikipedia, The Free Encyclopedia.". [Online; accessed Oct-2012].
URL: *https://en.wikipedia.org/wiki/Mathematica*

Wikipedia. 2012*b*. "Quaternion — Wikipedia, The Free Encyclopedia.". [Online; accessed July-2012].
URL: *http://en.wikipedia.org/wiki/Quaternion*

Williams, B., M. Cummins, J. Neira, P. Newman, I. Reid and J. Tardós. 2009. "A comparison of loop closing techniques in monocular SLAM." *Journal of Robotics and Autonomous Systems* (12).

Wolfe, W. J., D. Mathis, C. W. Sklair and M. Magee. 1991. "The Perspective View of Three Points." *International Journal of ACM Transactions on Pattern Analysis and Machine Intelligence* .

Wong, S., S. Vassiliadis and S. Cotofana. 2002. A Sum of Absolute Differences Implementation in FPGA Hardware. In *Proceedings of the 28th EUROMICRO Conference.*

Wu, C., S. Agarwal, B. Curless and S. M. Seitz. 2011. "Multicore Bundle Adjustment." *Proceedings of the 24th Computer Vision and Pattern Recognition* .

Zhang, Z. 1998. "Determining the Epipolar Geometry and its Uncertainty: A Review." *International Journal of Computer Vision* .

Zhang, Z. and Y. Shan. 2001. Incremental motion estimation through local Bundle Adjustment. Technical Report MSR-TR-01-54 Microsoft Research.

Zhang, Z. and Y. Shan. 2003. Incremental motion estimation through modified bundle adjustment. In *Proceedings of the 10th International Conference on Image Processing.*