Australian Government
Department of Defence
Science and Technology

Sparse Reconstruction of a Scene and Camera Poses from the Scene Images with MATLAB®

Leonid K Antanovskii

Weapons and Combat Systems Division
Defence Science and Technology Group

DST-Group–TR–3346

ABSTRACT

This report addresses the description and implementation of numerical algorithms for the reconstruction of world points representing a scene, and pinhole camera poses from the scene images. The *Image Processing* and *Computer Vision System* toolboxes of MATLAB are used for detecting, extracting and matching features in images. A camera graph is introduced to indicate which image pairs to process, and a homography graph is derived as a sub-graph of the line graph of the camera graph to parametrize three-dimensional transition homographies. The estimated transition homographies are applied to world points and cameras, locally reconstructed from image pairs, to bring them to a global frame of homogeneous coordinates. Then potentially duplicate points are eliminated using an introduced metric between two projective points with respect to cameras, and a visibility relation for *Bundle Adjustment* is computed. This approach properly addresses the common situation when a point disappears from a camera view and reappears later. Compatibility cocycle conditions for keypoint matching relations over the camera graph cycles and for transition homographies over the homography graph cycles are discussed.

RELEASE LIMITATION

Approved for public release

Published by

Weapons and Combat Systems Division
Defence Science and Technology Group
PO Box 1500
Edinburgh, South Australia 5111, Australia

Telephone: *1300 333 362*
Facsimile: *(08) 7389 6567*

February, 2017
AR-016–810

APPROVED FOR PUBLIC RELEASE

Sparse Reconstruction of a Scene and Camera Poses from the Scene Images with MATLAB®

Executive Summary

This report addresses the description and implementation of numerical algorithms for the reconstruction of world points representing a scene, and pinhole camera poses from the scene images. The *Image Processing* and *Computer Vision System* toolboxes of MATLAB are used for detecting, extracting and matching features in images. A camera graph is introduced to indicate which image pairs to process, and a homography graph is derived as a sub-graph of the line graph of the camera graph to parametrize three-dimensional transition homographies. The estimated transition homographies are applied to world points and cameras, locally reconstructed from image pairs, to bring them to a global frame of homogeneous coordinates. Then potentially duplicate points are eliminated using an introduced metric between two projective points with respect to cameras, and a visibility relation for *Bundle Adjustment* is computed. This approach properly addresses the common situation when a point disappears from a camera view and reappears later. Compatibility cocycle conditions for keypoint matching relations over the camera graph cycles and for transition homographies over the homography graph cycles are discussed.

THIS PAGE IS INTENTIONALLY BLANK

Author

Leonid K Antanovskii
Weapons and Combat Systems Division

Born in Siberia, Leonid Antanovskii holds a Master of Science (with distinction) in Mechanics and Applied Mathematics from the Novosibirsk State University and a PhD in Mechanics of Fluid, Gas and Plasma from the Lavrentyev Institute of Hydrodynamics of the Russian Academy of Science. Since graduation he worked for the Lavrentyev Institute of Hydrodynamics (Russia), Microgravity Advanced Research & Support Center (Italy), the University of the West Indies (Trinidad & Tobago), and in private industry in the USA and Australia.

Leonid Antanovskii joined the Defence Science and Technology Group in February 2007 working in the area of weapon–target interaction.

THIS PAGE IS INTENTIONALLY BLANK

Contents

List of Figures

THIS PAGE IS INTENTIONALLY BLANK

1 Introduction

The reconstruction of a real-world scene from multiple images of it, taken from the various positions of cameras, is a central problem in *Computer Vision*. In the general case, the calibration and poses of cameras are unknown, and hence are part of solution. Sparse reconstruction deals with the estimation of a point cloud, representing the scene, and is the first step in the reconstruction procedure. Based on the estimated point cloud, the objective of a dense reconstruction is to build a mesh of the scene by estimating line segments and facets passing through the world points.

Taking an image of a scene is a simple task, which is generally achieved by projecting objects in the three-dimensional space to the two-dimensional image plane of a camera. By adopting the pinhole camera model [Forsyth & Ponce 2003, Hartley & Zisserman 2003] the projection is represented by a linear map in the homogeneous coordinates of the space and the image plane. The inverse problem of reconstructing a scene from images is more complex in nature as compared with the forward problem of taking images.

In general, inverse problems are quite difficult to solve numerically [Tarantola 1987], mainly because of inherent poor conditioning of involved matrices that requires the application of suitable regularization techniques, called preconditioning. The accuracy of a reconstruction procedure depends on how many features in images are detected and how well do they match. In most circumstances these features are isolated points of interest, called keypoints, but sometimes line segments or curves or even regions can be also captured. The matched keypoints obtained by an automated procedure are in putative correspondence as they may contain outliers which should be reliably eliminated; otherwise the accuracy of reconstruction will be significantly deteriorated. The elimination procedure is based on estimating geometric constraints, given in terms of homogeneous multi-focal tensors, using the Random Sample Consensus (RANSAC) algorithm [Fischler & Bolles 1981]. Then, working with inliers only, three-dimensional world points and camera poses are estimated up to a non-degenerate projective transformation (three-dimensional homography). The world points and camera maps are computed in their own homogeneous coordinate charts determined by a pair of images if a bifocal constraint is used. The next step is to bring all these objects, the world points without duplication and the camera maps, into a global coordinate frame. Each of these steps introduces an unavoidable error which can become very significant. Therefore, the final important step is to apply *Bundle Adjustment*, which minimizes an objective function given in terms of the squares of Euclidean distances between image keypoints and reprojected world points. The Levenberg–Marquardt solver [Levenberg 1944, Marquardt 1963] is usually employed for the minimization of the objective function. In the end of this procedure a cloud of world points is obtained along with camera maps, which fit best to the imagery.

Comprehensive background material and basic numerical methods for structure reconstruction are provided in [Hartley & Zisserman 2003]. It is worthwhile emphasizing that this monograph does not address the feature detection technology, which constitutes a broad subject of research in its own, and in most situations assumes that all world points of interest are visible in all views. This is obviously not the case in real-world scenarios due to occlusion.

Feature detection is a low-level image processing operation based on examining every pixel of an image and its immediate neighbourhood to associate feature descriptors to the detected keypoints; the extracted descriptors are then used to match the keypoints in two images.

Several feature detection algorithms are publicly available. Some of them are listed below in the chronological order of development:

- *Combined Corner and Edge Detector* [Harris & Stephens 1988];
- *Minimum Eigenvalue* [Shi & Tomasi 1994];
- *Scale-Invariant Feature Transform* [Lowe 1999, Lowe 2004];
- *Maximally Stable Extremal Regions* [Matas et al. 2002, Mikolajczyk et al. 2005, Nister & Stewenius 2008, Obdrzalek et al. 2009];
- *Features from Accelerated Segment Test* [Rosten & Drummond 2005];
- *Speeded-Up Robust Features* [Bay, Tuytelaars & Van Gool 2006, Bay et al. 2008, Bradski & Kaehler 2008]; and
- *Binary Robust Invariant Scalable Keypoints* [Leutenegger, Chli & Siegwart 2011].

The above feature detection algorithms, except for the patented *Scale-Invariant Feature Transform* algorithm, are implemented in the *Image Processing* and *Computer Vision System* toolboxes of the latest release of MATLAB®.

The author's previous publication [Antanovskii 2014] is related to some theoretical aspects and application of *Geometric Algebra* to 3D reconstruction. The paper [Antanovskii 2016*b*] describes the sparse reconstruction of a scene and camera poses from images processed sequentially, akin to a video stream. The paper [Antanovskii 2016*a*] provides a theoretical background to the more general problem of an arbitrary set of images matched pairwise according to a given camera graph. This report extends [Antanovskii 2016*a*] with the description and implementation of numerical algorithms for the reconstruction of world points and camera poses from a scene imagery. The *Image Processing* and *Computer Vision System* toolboxes of MATLAB are used for detecting, extracting and matching features in images. Following [Antanovskii 2016*a*] we employ a camera graph indicating which image pairs to process, and a homography graph parametrizing three-dimensional transition homographies between two coordinate systems defined by the adjacent edges of the camera graph. The homography graph is thus a sub-graph of the line graph of the camera graph [Harary 1972]. The estimated transition homographies are applied to world points and cameras, locally reconstructed from image pairs, to bring them to a global frame of homogeneous coordinates. Then potentially duplicate points are eliminated using an introduced metric between two projective points with respect to cameras, and a visibility relation for *Bundle Adjustment* is computed. Compatibility cocycle conditions for keypoint matching relations over the camera graph cycles and for transition homographies over the homography graph cycles are discussed. This rather academic approach properly addresses the common situation when a point disappears from a camera view and reappears later. However, it may not be suitable to large-scale simulations because of extra processing which inevitably deteriorates performance.

Using a randomly-generated benchmark model, numerical experiments are conducted to verify the developed MATLAB code and establish correlation between artificially introduced Gaussian noise and reconstruction errors. The code has been also applied to the airborne mid wave infra-red (MWIR) imagery of a scene containing a building in the Ottawa (Canada) area, and the airborne visible-spectrum imagery of a scene containing a church. The images and metadata files with camera poses have been provided to Defence Science and Technology

Group by Defence Research and Development Canada.

2 Problem formulation

Employing the pinhole camera model [Forsyth & Ponce 2003, Hartley & Zisserman 2003], the projection of a three-dimensional world point to the image plane of a camera reduces to the matrix multiplication $x = X\,P$ where x is the row-vector of three homogeneous coordinates of the image point, $x = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}$, X the row-vector of four homogeneous coordinates of the world point, $X = \begin{bmatrix} X_1 & X_2 & X_3 & X_4 \end{bmatrix}$, and P the 4-by-3 matrix of the homogeneous coordinates of the camera map,

$$P = \begin{bmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ P_{31} & P_{32} & P_{33} \\ P_{41} & P_{42} & P_{43} \end{bmatrix} . \tag{1}$$

The right action of camera maps is adopted throughout the paper, so the conventional camera matrix is transposed to the one used here. Recall that the homogeneous coordinates are defined up to a nonzero scale. The camera matrix P is assumed non-degenerate (of rank 3). In this case the left kernel of the matrix P is one-dimensional, and thus a nonzero element of the kernel defines the camera centre C. Since $C\,P = 0$, the camera centre is not projected to a valid image point. All the other three-dimensional points are mapped to the projective image plane of the camera; however, only the points with positive depth (in front of the camera) and within a camera-specific cone of vision will be seen in the image plane [Forsyth & Ponce 2003]. The view frustum obtained by truncating the cone of vision with parallel planes is typically a pyramid with a rectangular base in the virtual image plane located in front of the camera for convenience as opposed to the real image plane behind the camera centre.

The inhomogeneous Euclidean coordinates of the image point, $x = \begin{bmatrix} x_1 & x_2 \end{bmatrix}$, are given by the following formula

$$x = \varphi\,(X, P) \tag{2}$$

where

$$\varphi\,(X, P) = \begin{bmatrix} \varphi_1\,(X, P) & \varphi_2\,(X, P) \end{bmatrix} , \tag{3}$$

$$\varphi_1\,(X, P) = \frac{X_1\,P_{11} + X_2\,P_{21} + X_3\,P_{31} + X_4\,P_{41}}{D\,(X, P)} , \tag{4}$$

$$\varphi_2\,(X, P) = \frac{X_1\,P_{12} + X_2\,P_{22} + X_3\,P_{32} + X_4\,P_{42}}{D\,(X, P)} , \tag{5}$$

$$D\,(X, P) = X_1\,P_{13} + X_2\,P_{23} + X_3\,P_{33} + X_4\,P_{43} . \tag{6}$$

The projection map $\varphi(X, P)$ depends on the 4 homogeneous coordinates of X and 12 entries of P, which, after ignoring arbitrary scales, form 14 independent parameters mapped to the 2 inhomogeneous coordinates of x. It is straightforward to check that $\varphi(X, P)$ is invariant with respect to any three-dimensional homography H given by a non-singular 4-by-4 matrix

$$H = \begin{bmatrix} H_{11} & H_{12} & H_{13} & H_{14} \\ H_{21} & H_{22} & H_{23} & H_{24} \\ H_{31} & H_{32} & H_{33} & H_{34} \\ H_{41} & H_{42} & H_{43} & H_{44} \end{bmatrix} \tag{7}$$

which acts on X and P according to the rules

$$X \mapsto X\,H\,, \quad P \mapsto H^{-1}\,P\,. \tag{8}$$

In other words,

$$\varphi(X, P) \equiv \varphi\left(X\,H, H^{-1}\,P\right)\,. \tag{9}$$

Note that the projection map φ is undefined at the camera centre $X = C$, more generally, at the principal plane $D(X, P) = 0$. Since we are interested in the inverse problem of data fitting, the given image points are always finite.

In the most general formulation, the sparse reconstruction of a scene takes a collection of images of the scene as an input along with control parameters, including an instruction for which image pairs to match, and generates world points and camera maps as an output. We assume that the images are first preprocessed to detect interest points, which will form a collection of arrays $x^{(k)}$ of two-dimensional Euclidean coordinates. Here the index k runs from 1 to the total number of images or views, say N_V. In other words, the array $x^{(k)}$ is a two-column matrix whose number of rows may vary with k. The instruction for which image pairs to match will be specified by a two-column image-pair connectivity matrix E with a typical row of the form $\begin{bmatrix} k_1 & k_2 \end{bmatrix}$. For each row $\begin{bmatrix} k_1 & k_2 \end{bmatrix}$ of E enumerated by the row index e running from 1 to N_E, the interest points $x^{(k_1)}$ and $x^{(k_2)}$ are matched, resulting in a two-column matrix $M^{(k_1,k_2)}$ whose typical row $\begin{bmatrix} m_1 & m_2 \end{bmatrix}$ gives the row index m_1 to $x^{(k_1)}$ and m_2 to $x^{(k_2)}$ of keypoints in putative correspondence. We are not making any assumption on the quality of the matching procedure, so outliers may be present.

In an ideal case, the correspondences $M^{(k_1,k_2)}$ should induce a bijection between putatively matched keypoints. However, this is not guaranteed by a feature matching algorithm, so we will assume that $M^{(k_1,k_2)}$ is a binary relation, which we will call the *keypoint matching relation*.

Recall that a binary relation R is by definition a set of ordered pairs of elements $a \in A$ and $b \in B$ [Schmidt 2011]. In other words, R is a subset of the direct product $A \times B$. The sets A and B are called the source and target of R, respectively. By definition, two elements a and b are R-related, written $a\,R\,b$, if $(a, b) \in R$. Binary relations form a category whose objects are sets, and morphisms are the relations themselves [Mac Lane 1998]. Binary relations can be composed (the composition is associative) and reverted, and the identity relation on a set A plays the role of the unit 1_A which is the neutral element in the category of relations. Indeed, the identity relation is defined as $1_A = \{(a, a) : a \in A\}$, and the composition of two relations R and S can be defined if the target set of R is the source set of S. In this case, if

$R \subset A \times B$ and $S \subset B \times C$, then $R\,S \subset A \times C$ is defined as the set of pairs $(a, c) \in A \times C$ such that $(a, b) \in R$ and $(b, c) \in S$ for some $b \in B$. Note that the relation $R\,S$ may be empty even when both R and S are non-empty. It is straightforward to check that $1_A\,R = R\,1_B = R$. The reverted relation R^* is defined as the set of $(b, a) \in B \times A$ such that $(a, b) \in R$. The reverse operation is obviously an involution, because $(R^*)^* = R$. It is also called the inversion, or conversion, or transposition. Note that $R\,R^* \neq 1_A$ in general even when the domain of R coincides with A.

The image-pair connectivity matrix E defines the edges of a simple graph [Harary 1972] whose vertices are the integers $k = 1, \ldots, N_V$. We call this graph the *camera graph* [Antanovskii 2016*a*]. It is natural to extend this simple graph to a directed graph by assigning an orientation to each edge (k_1, k_2). Let us attach $x^{(k)}$ to each vertex k and $M^{(k_1,k_2)}$ to each edge $e = (k_1, k_2)$ of the camera graph as values (weights), and denote the *valued* camera graph by $\mathcal{G}$. Thus, the graph $\mathcal{G}$ has size N_E and order N_V. It is natural to symmetrize this graph by adding reversed arrows and setting

$$M^{(k_2,k_1)} = \left[M^{(k_1,k_2)}\right]^* . \tag{10}$$

This convention will be useful when we define composition of relations along a cycle of $\mathcal{G}$. Note that the order of the symmetrized camera graph is doubled, but we will always refer to the underlying simple graph.

The sparse reconstruction problem, modulo interest point detection and matching, takes the following compact form. Given camera graph $\mathcal{G}$ with vertex values $x^{(k)}$ and edge values $M^{(k_1,k_2)}$, reconstruct world points $X_{[n]}$, $n = 1, \ldots, N_X$, camera maps $P_{[k]}$ and a *visibility relation* $\mathcal{V} \subset \{1, \ldots, N_X\} \times \{1, \ldots, N_V\}$ such that

$$x^{(k)}_{i_{n\,k}} = \varphi\left(X_{[n]}, P_{[k]}\right) \tag{11}$$

for each $(n, k) \in \mathcal{V}$ and some row index $i_{n\,k}$ into the matrix $x^{(k)}$.

Note that the collection of the homogeneous coordinates of the world points $X_{[n]}$ is given by an N_X-by-4 matrix where N_X is unknown. The visibility relation $\mathcal{V}$ tells us whether a point $X_{[n]}$ is visible in the view k or not. If it is, the index $i_{n\,k}$ will indicate the image keypoint in the array $x^{(k)}$ corresponding to $X_{[n]}$.

Ignoring arbitrary scales in $X_{[n]}$ and $P_{[k]}$, the total number of scalar unknowns (degrees of freedom) in the reconstruction problem (11) is equal to $N_1 = 3N_X + 11N_V$. If all the world points are visible in all the image planes, the number of scalar equations is equal to $N_2 = 2N_X\,N_V$. Since N_1 grows linearly with respect to N_X and N_V, but N_2 grows quadratically, the system of defining equations (11) becomes overdetermined, $N_1 < N_2$, even for moderate N_X and $N_V \geq 2$. The visibility relation $\mathcal{V}$ may change this proportion but not dramatically provided that N_X and N_V are sufficiently large. Therefore, the number of scalar equations is expected to exceed the number of model parameters.

Since noise is always present in images, the system of equations (11) is never satisfied exactly. Therefore, an optimal solution should be obtained by minimizing the total reprojection error. Assuming that this error is the sum of the Euclidean distances squared between the measured and reprojected image points, the least-squares objective function assumes the form

$$f\,(w) = \sum_{(n,k)\in\mathcal{V}} \left\| \varphi\left(X_{[n]}, P_{[k]}\right) - x^{(k)}_{i_{n\,k}} \right\|^2 \tag{12}$$

which has to be minimized over the model parameters $w = \{X_{[n]}, P_{[k]}\}$.

A solution w to the above minimization problem is by no means unique, because any 3D homography H applied simultaneously to all the points and camera maps by the rules (8) does not change the objective function $f(w)$. In particular, the reconstructed scene is determined up to an arbitrary three-dimensional homography. Reconstruction up to a 3D homography is called projective.

The initially specified camera graph may change in the process of reconstruction if the number of matched keypoints appears to be insufficient; for example, in the presence of a large proportion of outliers. In this case the corresponding edge $e = (k_1, k_2)$ of the camera graph has to be removed along with its value $M^{(k_1,k_2)}$. In principle, this edge-elimination procedure can produce a disconnected camera graph even when the initial camera graph is complete, that is when all the $\frac{1}{2} N_V (N_V - 1)$ image pairs are instructed to be matched.

3 Solution outline

The problem of sparse reconstruction belongs to the class of inverse problems, such as data fitting or model parameter estimation. In this context the given image keypoints represent measured data, and the coordinates of the world points and camera maps constitute the model parameters to be estimated. The most reliable way to solve the reconstruction problem is to minimize the objective function (12) as it contains observable values only. Finding the global minimum of a function of many variables is a formidable task in most circumstances, therefore only an iterative procedure, such as the Levenberg–Marquardt algorithm, widely used in nonlinear regression analysis, can be practically afforded. However, an iterative solution may find a local minimum of the objective function if the initial guess is not close enough to the exact solution. Therefore, it is of a paramount importance to estimate an approximate solution as accurately as possible, and then use it as an initial guess for the Levenberg–Marquardt solver.

Recall that the sparse reconstruction problem is projective in nature, so the solution, if exists, will be obtained up to a 3D homography H with the action (8). A projective space does not have a canonical metric to measure distances between 3D points which is needed for building the visibility relation $\mathcal{V}$ by eliminating potentially duplicate points. However, when a collection of camera maps $P = \{P_k\}$ $(k = 1, \ldots, K)$ is given, the associated *metric* δ_P between two projective 3D points X and Y with respect to the camera maps can be defined by the expression

$$\delta_P(X, Y) = \frac{1}{K} \sum_{k=1}^{K} \|\varphi(X, P_k) - \varphi(Y, P_k)\|^2 \tag{13}$$

where $\|x\|$ denotes the Euclidean norm in the image plane.

Strictly speaking, δ_P is not a distance squared as the axiom $\delta_P(X, Y) = 0$ if and only if $X = Y$ may be violated for a degenerate configuration of camera maps P. This situation always occurs for one camera as the whole line through the camera centre is projected to a single image point. The rest axioms of a distance, the symmetry $\delta_P(X, Y) = \delta_P(Y, X)$ and the triangle inequality $\sqrt{\delta_P(X, Z)} \le \sqrt{\delta_P(X, Y)} + \sqrt{\delta_P(Y, Z)}$, are always satisfied. The

metric δ_P is independent of a 3D homography H due to (9), and hence can be safely used for detecting duplicate points, as well as for code validation against 'ground truth' data.

3.1 Local reconstruction

Local reconstruction involves finding a solution for each image pair as specified by the edges of the camera graph. The solution, consisting of an array of world points and two camera maps, is obtained in a local coordinate frame, and hence the name of local reconstruction.

For a given pair of camera maps, say P and Q, the homogeneous coordinates of the *bifocal tensor* are formed by the 3-by-3 matrix

$$T = \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{bmatrix} \tag{14}$$

where

$$T_{ij} = \det \begin{bmatrix} P_{1\,\sigma(i,1)} & P_{1\,\sigma(i,2)} & Q_{1\,\sigma(j,1)} & Q_{1\,\sigma(j,2)} \\ P_{2\,\sigma(i,1)} & P_{2\,\sigma(i,2)} & Q_{2\,\sigma(j,1)} & Q_{2\,\sigma(j,2)} \\ P_{3\,\sigma(i,1)} & P_{3\,\sigma(i,2)} & Q_{3\,\sigma(j,1)} & Q_{3\,\sigma(j,2)} \\ P_{4\,\sigma(i,1)} & P_{4\,\sigma(i,2)} & Q_{4\,\sigma(j,1)} & Q_{4\,\sigma(j,2)} \end{bmatrix} \tag{15}$$

and σ is the circular shift function defined by $\sigma(p, q) = 1 + (p + q - 1) \bmod 3$. The expression (15) has a functorial nature [Antanovskii 2016*a*]. The important property of the homogeneous matrix T is that its rank is equal to 2, in particular $\det T = 0$. Note that the conventional fundamental matrix is transposed to T.

Given a bifocal tensor T with the constraint $\det T = 0$, two camera maps P and Q can be chosen in many ways such that the conditions (15) are satisfied [Hartley & Zisserman 2003]. One camera map can be arbitrarily selected, and the other will be obtained, still in many ways.

The bifocal tensor T imposes a geometric constraint on the corresponding image points, namely [Hartley & Zisserman 2003]

$$\begin{bmatrix} x_1 & x_2 & 1 \end{bmatrix} \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ 1 \end{bmatrix} = 0 \tag{16}$$

where $x = \varphi(X, P)$ and $y = \varphi(X, Q)$ with X being the homogeneous coordinates of some world point. Knowing keypoint correspondences given by the matrix $M^{(e)}$, the equation (16) becomes a linear homogeneous equation in the entries of T, and can be solved by the *Singular Value Decomposition* (SVD) algorithm [Golub & Van Loan 1996], which finds the best fit to the kernel of a usually over-determined system of equations. The broad class of methods for

solving homogeneous equations, almost invariably based on the SVD algorithm, are called the *Direct Linear Transformation* (DLT) algorithms [Hartley & Zisserman 2003].

It is worthwhile emphasizing that, since a solution to a homogeneous equation provides the homogeneous coordinates of a projective space object of some type (a projective point, tensor, or map), the application of the SVD algorithm is ambiguous as it depends on the frame of homogeneous coordinates. In other words, the SVD minimization procedure is not invariant with respect to homogeneous coordinate transformation. To partially overcome this problem, suitable normalization techniques are developed, which are closely related to the preconditioning of a linear system of equations. In general, a normalization procedure produces a more accurate solution, which is still not invariant with respect to projective transformations. To make the solution really invariant, one needs to minimize a reprojection error using a suitable non-linear iterative algorithm, starting with the DLT solution as an initial guess.

Since a homogeneous 3-by-3 matrix has 8 degrees of freedom, it suffices to provide 8 point correspondences when assembling equations (16). The right kernel of the 8-by-9 matrix defining T will be one-dimensional in general configuration, and hence will produce a unique tensor T up to a scale. The normalized 8-point algorithm, with an appropriate image point preconditioning and coercing the obtained matrix to ensure $\det T = 0$, is given in [Hartley & Zisserman 2003, Page 282]. The coercion of the bifocal tensor is an important step because of unavoidable noise in images, though may introduce errors. This method can be applied without alteration to estimate the bifocal tensor from any number of point correspondences, greater than or equal to 8.

Actually, the bifocal tensor T has 7 degrees of freedom, because $\det T = 0$, and therefore it suffices to use 7 point correspondences to determine T [Hartley & Zisserman 2003]. However, the DLT algorithm is no longer applicable since the constraint $\det T = 0$ is nonlinear in entries of T. A nonlinear method for the estimation of T from 7 point correspondences is described in [Hartley & Zisserman 2003, Page 281]. Briefly, for 7 point correspondences, the 7-by-9 matrix defining T has a two-dimensional right kernel, and therefore two basis elements of the kernel, say $T_{(1)}$ and $T_{(2)}$, are available. Substituting the general solution $T = \alpha\, T_{(1)} + (1-\alpha)\, T_{(2)}$ into the equation $\det T = 0$ results in a cubic polynomial equation for α, which provides at least one real root. Note that the polynomial $p(\alpha) = \det(\alpha\, A + B)$ where A and B are matrices, is called the characteristic polynomial of the matrix pencil [Gantmacher 1959].

The RANSAC algorithm [Fischler & Bolles 1981] is used to eliminate possible outliers which do not satisfy the geometric constraint (16) given some tolerance. The matrix T is explicitly computed from 7 randomly sampled keypoint correspondences in a general configuration. The struggle for the minimum set of keypoint correspondences pays off, making the RANSAC solver more reliable as its success depends on the probability that all 7 randomly sampled correspondences represent inliers. The RANSAC solver returns inlier indices from which the bifocal tensor is re-estimated by the normalized DLT algorithm using the whole set of inliers.

In the end of this procedure, the keypoint matching relations $M^{(k_1,k_2)}$ are modified by keeping inlier correspondences only. Actually, the relations $M^{((k_1,k_2))}$ should be bijective. The camera graph can be also modified if the number of bijective correspondences is insufficient (less than 7) to recover the bifocal tensor T. Moreover, the following *camera cocycle condition*

must be satisfied

$$M^{(k_1,k_2)}\, M^{(k_2,k_3)}\, \cdots\, M^{(k_p,k_1)} \subset 1_{k_1}\,. \tag{17}$$

Here $\langle k_1, k_2, \ldots, k_p\rangle$ is any cycle of the camera graph, and the symbol 1_{k_1} in the right-hand side denotes the identity relation of the image plane k_1. It suffices to test the camera cocycle condition (17) on the *fundamental cycles* of the camera graph. Recall that a graph cycle must contain at least three vertices, so $p \geq 3$. If the superposition of relations in (17) is empty, the cocycle condition will be automatically satisfied.

As soon as the bifocal tensor T is obtained, two canonical camera matrices, P and Q, compatible with T are chosen and world points reconstructed by the triangulation algorithm [Hartley & Zisserman 2003]. This is accomplished by the SVD subroutine. Note that the triangulation procedure leads to the following linear equations

$$\begin{bmatrix} X_1 & X_2 & X_3 & X_4 \end{bmatrix} A = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix} \tag{18}$$

for the homogeneous coordinates of the world point X, where the 4-by-4 matrix A given by the expression

$$A = \begin{bmatrix} x_1\, P_{13} - P_{11} & x_2\, P_{13} - P_{12} & y_1\, Q_{13} - Q_{11} & y_2\, Q_{13} - Q_{12} \\ x_1\, P_{23} - P_{21} & x_2\, P_{23} - P_{22} & y_1\, Q_{23} - Q_{21} & y_2\, Q_{23} - Q_{22} \\ x_1\, P_{33} - P_{31} & x_2\, P_{33} - P_{32} & y_1\, Q_{33} - Q_{31} & y_2\, Q_{33} - Q_{32} \\ x_1\, P_{43} - P_{41} & x_2\, P_{43} - P_{42} & y_1\, Q_{43} - Q_{41} & y_2\, Q_{43} - Q_{42} \end{bmatrix} \tag{19}$$

has rank 3 in the general configuration. The triangulation algorithm has to be repeated for all point correspondences (x, y) thus producing the point cloud X.

The final step is to apply the *Gold Standard* method [Hartley & Zisserman 2003, Page 285] which adjusts the obtained camera maps P and Q, and world points X, by minimizing the reprojection error in both images. The reduced objective function (12), which now takes the form

$$f\,(w) = \sum \Big(\|\varphi\,(X, P) - x\|^2 + \|\varphi\,(X, Q) - y\|^2 \Big) \tag{20}$$

where the summation is taken over all the reconstructed points, is minimized using the Levenberg–Marquardt solver. The least-squares Jacobian matrix is analytically computed in a straightforward manner; for example

$$\frac{\partial \varphi_1\,(X, P)}{\partial X_1} = \frac{P_{11} - \varphi_1\,(X, P)\; P_{13}}{D\,(X, P)}\,, \tag{21}$$

$$\frac{\partial \varphi_1\,(X, P)}{\partial P_{11}} = \frac{X_1}{D\,(X, P)}\,, \tag{22}$$

$$\frac{\partial \varphi_1\,(X, P)}{\partial P_{13}} = -\frac{\varphi_1\,(X, P)\; X_1}{D\,(X, P)}\,. \tag{23}$$

In the end of this procedure, we obtain two camera maps and a cloud of world points corresponding to the detected inliers for each image pair specified by the rows of E. As mentioned before, the camera graph $\mathcal{G}$ may be altered by deleting those edges which have an insufficient number of inliers.

3.2 Global reconstruction

After the local reconstruction is completed, camera maps and world points are computed in their own homogeneous coordinate charts, and hence the next step is to bring them together to a global reference frame. This is accomplished by estimating and applying three-dimensional transition homographies, between two charts of homogeneous coordinates, parametrized by the edges of the *homography graph* [Antanovskii 2016*a*]. The homography graph, denoted here by $\mathcal{H}$, is a symmetrized graph built from the line graph of the camera graph $\mathcal{G}$. The vertices of the homography graph are the edges of the camera graph (the matrix E), and the edges of the homography graph are those pairs (e_1, e_2) of the adjacent edges of the camera graph which have a sufficient number of common keypoints in the shared image to determine a transition homography from local coordinates associated to e_1 to those of e_2. Let $H^{(e_1,e_2)}$ denotes the corresponding transition homography.

Similarly to the symmetrization procedure of the camera graph, we assign the inverse transition homography to an edge with reversed orientation

$$H^{(e_2,e_1)} = \left[H^{(e_1,e_2)}\right]^{-1} . \tag{24}$$

It is natural to apply the transition homographies to the world points and camera maps in order to bring all of them to a global reference frame. However, this procedure may break unless the following *homography cocycle condition* is met

$$H^{(e_1,e_2)}\, H^{(e_2,e_3)} \cdots H^{(e_p,e_1)} = I\,. \tag{25}$$

Here $\langle e_1, e_2, \ldots, e_p\rangle$ is any cycle of the homography graph, and I is the unit homography. As with the camera graph, it suffices to test the homography cocycle condition (25) on the fundamental cycles of the homography graph.

The transition homography $H^{(a,b)}$ maps local coordinates of the world points $X^{(a)}$ to $X^{(b)}$. The defining equations for $H^{(a,b)}$ take the form

$$\left[X^{(a)}\, H^{(a,b)}\right] \wedge X^{(b)} = 0 \tag{26}$$

where $\wedge$ is the exterior product of the Grassmann algebra $\Lambda\left(\mathbb{R}^4\right)$ over the four-dimensional vector space of homogeneous coordinates. The defining equations are linear in the entries of $H^{(a,b)}$, and hence the SVD algorithm can be employed again to find the solution. However, this approach depends on homogeneous coordinates, and therefore some preconditioning must be applied. We used the scaling of the homogeneous coordinates of $X^{(a)}$ and $X^{(b)}$, projecting them to the unit sphere. Though this procedure is not invariant, the matrix entries of the defining equation (26) for $H^{(a,b)}$ are made of order 1. The RANSAC solver is applied to eliminate possible outliers, and then the transition homography $H^{(a,b)}$ is re-estimated from inliers only. If an insufficient number of matches is found, the corresponding arrow (a, b) of the homography graph $\mathcal{H}$ is deleted along with (b, a). In principle, this may result in a discontinuous homography graph. Then, using the initial guess $H^{(a,b)}$, the Levenberg–Marquardt solver is applied to minimize the following objective function

$$\delta_{P^{(a)}}\left(X^{(a)}, X^{(b)}\, H^{(b,a)}\right) + \delta_{P^{(b)}}\left(X^{(a)}\, H^{(a,b)}, X^{(b)}\right) \tag{27}$$

where $P^{(a)}$ and $P^{(b)}$ are the corresponding two-element families of camera maps in local coordinates. Note that the symmetric objective function depends on 16 parameters, the entries of $H^{(a,b)}$, because $H^{(a,b)}\,H^{(b,a)} = I$ and $X^{(a)}$, $P^{(a)}$, $X^{(b)}$, $P^{(b)}$ are given.

The estimated transition homographies $H^{(a,b)}$ are used to construct the net homography along an oriented path of the homography graph. The homography cocycle conditions guarantee that the net homography is path independent. Starting from a source vertex of the homography graph, which is an edge of the camera graph, we can choose the shortest path leading to a given edge of the camera graph, compute the net homography along the path, and apply it to the world points and camera maps. We select the camera graph edge with the maximum number of inliers as the source vertex of the homography graph. To reduce the number of multiplications of the transition homographies to a minimum, we identify the path distance with the number of the path edges when computing the shortest path. Dijkstra's algorithm [Diestel 2005] for finding the shortest paths has been implemented.

3.3 Bundle adjustment

Next step is to eliminate duplicate world points and to recalculate all the involved indices mapping the keypoints to the world points, thus computing the visibility relation $\mathcal{V}$ needed for the definition (12) of the objective function. Given some tolerance the duplicate world points are eliminated using the introduced metric (13). Proper elimination of duplicates reduces the number of model parameters and hence improves the performance of the bundle adjustment.

As soon as the visibility relation $\mathcal{V}$ is computed, along with the row indices $i_{n\,k}$ into the matrix $x^{(k)}$, defined on $\mathcal{V}$, the Levenberg–Marquardt solver is applied to minimize the objective function (12). A potentially large system of linear equations involving the least-squares Jacobian matrix has to be solved at each iteration of the minimization algorithm. It is important to emphasize that the Jacobian matrix is sparse. The Gaussian elimination algorithm becomes more efficient when a sparse matrix defining a system of linear equations is stored in the sparse rather than dense format. In this case the number of computer operations is considerably reduced by avoiding unnecessary multiplication by zeros. Therefore, with the help of analytic expression of the Jacobian matrix stored in a sparse format, the final optimization step performs efficiently. The Levenberg–Marquardt solver is terminated either when the relative variation of the objective function becomes smaller than a given tolerance ε, namely

$$\frac{f\,(w_{i-1}) - f\,(w_i)}{f\,(w_{i-1}) + f\,(w_i)} < \varepsilon\,, \tag{28}$$

or when the iteration count i reaches a given iteration threshold.

4 Numerical experiments

A benchmark test fixture for the verification of the developed MATLAB code has been designed. Almost all configuration parameters of the benchmark model described below are

randomly sampled using Gaussian distribution characterized by expectation (mean value) and standard deviation.

The Euclidean coordinates of the three-dimensional points are randomly generated with zero expectation and the standard deviation of 1 unit length, producing a sphere-like point cloud around the origin. The cameras have fixed intrinsic parameters, namely, the view frustum rectangle is a square of 1000-pixel size, the principal point is at the centre of the view frustum rectangle, the pixel aspect ratio is 1 and the pixel skew parameter is zero. The focal lengths of the cameras have the expectation of 1000 and the standard deviation of 100 pixels, and the expectation and standard deviation of the distance of the cameras from the origin (the camera range) are equal to 10 and 1, respectively. The principal axes of the cameras point at the origin, and the angle of camera rotation about the principal axis is uniformly sampled from the interval $[0, 2\pi]$.

The randomly generated benchmark model having 120 points and 10 cameras is shown in Figure A1. The principal axes of the cameras are displayed in red.

The virtual images of the point cloud are also randomly generated by setting the range of keypoint detection probability to $[0.8, 1.0]$, the range of matching probability to $[0.6, 1.0]$, and the range of outlier probability to $[0.0, 0.1]$. It is ensured that only the points projected inside the view frustum rectangle of the camera are selected. Then Gaussian noise is added to the dimensional Cartesian coordinates of the image points. We refer to its standard deviation as the noise level. The dimension of the Gaussian noise level is in pixels.

Table 1: Solver control parameters

Bifocal tensor	
RANSAC trial threshold	1000
RANSAC inlier tolerance	10^{-2}
RANSAC inlier ratio	0.2
RANSAC confidence probability	0.99
Iteration threshold	4000
Convergence tolerance	10^{-6}
Transition homography	
RANSAC trial threshold	1000
RANSAC inlier tolerance	10^{-4}
RANSAC inlier ratio	0.2
RANSAC confidence probability	0.99
Iteration threshold	4000
Convergence tolerance	10^{-6}
Visibility relation	
World point uniqueness tolerance	10^{-4}
Bundle adjustment	
Iteration threshold	4000
Convergence tolerance	10^{-6}

Numerical experiments for the benchmark model are conducted for a gradually increasing Gaussian noise level, averaged over 20 random samples. The solver control parameters are given in Table 1. The complete initial camera graph is specified.

The mean values of the dimensionless point reprojection error, camera reconstruction error and homography cocycle error as functions of the image noise level are shown in Figures A2, A3 and A4, respectively. The camera cocycle condition (17) was always fulfilled. The dimension of reprojection error is in pixels, whereas camera reconstruction and homography cocycle errors are dimensionless. They are defined in terms of the following metric

$$\min\left(\left\|\frac{P}{\|P\|_F}-\frac{Q}{\|Q\|_F}\right\|_F,\left\|\frac{P}{\|P\|_F}+\frac{Q}{\|Q\|_F}\right\|_F\right) \tag{29}$$

between two projective objects P and Q, say camera maps or homographies, where $\|\cdot\|_F$ denotes the Frobenius norm [Golub & Van Loan 1996, Page 55] of the corresponding matrix of homogeneous coordinates of the projective object. This metric is not invariant with respect to projective transformations, but makes sense for two given camera matrices one of which is a reference 'ground truth' camera, or for two homographies one of each is the unit homography I arising in (25).

As expected, on average, these errors tend to increase with the noise level, but vanish at zero level. It is seen from Figure A2 that the reprojection error is roughly proportional to the noise level with the coefficient of proportionality between 2 and 3. This interesting result allows one to estimate the image noise in terms of the reprojection error provided that its distribution is Gaussian.

5 Simulation results

The developed MATLAB code has been partially validated against two sets of imagery provided to Defence Science and Technology Group by Defence Research and Development Canada.

The first set is the airborne mid wave infra-red (MWIR) imagery of a scene containing a building in the Ottawa (Canada) area. Figure A5 shows one of the MWIR images.

The second set is the airborne imagery of a scene containing a church. The texture of the buildings is altered, but the rest scene is intact. Figure A9 shows one of the images.

A metadata file with camera poses are supplied with each imagery dataset. It is acknowledged that the GPS camera positions are not accurate enough, so we used the camera centres only for visualization purposes by bringing all the projective objects to the reference frame defined by the camera centres using *Least Squares* estimation. Geographic coordinates are preliminary converted to the UTM coordinates.

Figure A6 shows the reconstructed point cloud and camera trajectory from 60 images of the building in the Ottawa area. The first camera pose is shown by the red asterisk. There are 6,288 world points reconstructed. The reprojection error is shown in Figure A7. The

maximum error is about 7 pixels, which is reasonable enough given that the images have the size of 480-by-640 pixels. As is expected, the reconstruction error of camera poses is large as shown in Figure A8.

Figure A10 shows the reconstructed point cloud and camera trajectory from 71 images of the church building. The first camera pose is shown by the red asterisk. There are 4,297 world points reconstructed. The reprojection error is shown in Figure A11. The maximum error is about 10 pixels, which is reasonable enough given that the images have the size of 720-by-1280 pixels. As is expected, the reconstruction error of camera poses is large as shown in Figure A12.

6 Discussion

A theoretical background for the sparse reconstruction of a point cloud and camera maps from image points of interest in putative correspondence, is presented. A prototype code has been developed in MATLAB, thoroughly verified by designed unit tests and a benchmark test fixture, and partially validated against real-world imagery provided by Defence Research and Development Canada. The developed code is currently based on the *Image Processing* and *Computer Vision System* toolboxes of MATLAB.

Acknowledgements

The author is grateful to Jonathan Fournier from Defence Research and Development Canada for providing the airborne imagery for code validation and benchmarking. Valuable discussion with Dr Leszek Swierkowski from the Defence Science and Technology Group is very much appreciated.

7 References

Antanovskii, L. K. (2014) *Implementation of geometric algebra in MATLAB® with applications*, Technical Report DSTO–TR–3021, DSTO, Edinburgh, Australia.

Antanovskii, L. K. (2016*a*) *Mathematical aspects of computer vision*, Technical Report DST-Group-TR-3214, DST Group, Edinburgh, Australia.

Antanovskii, L. K. (2016*b*) *Projective reconstruction of world points and camera matrices from a sequence of images with MATLAB®*, Technical Report DST-Group-TR-3213, DST Group, Edinburgh, Australia.

Bay, H., Ess, A., Tuytelaars, T. & Van Gool, L. (2008) Speeded-up robust features (SURF), *Computer Vision and Image Understanding* **110**(3), 346–359.

Bay, H., Tuytelaars, T. & Van Gool, L. (2006) SURF: Speeded up robust features, *in Proc. 9th European Conf. Computer Vision*.

Bradski, G. & Kaehler, A. (2008) *Learning OpenCV: Computer Vision with the OpenCV Library*, O'Reilly, Sebastopol, CA.

Diestel, R. (2005) *Graph Theory*, 3rd edn, Springer, Berlin.

Fischler, M. A. & Bolles, R. C. (1981) Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *Comm. Assoc. Comp. Mach.* **24**(6), 381–395.

Forsyth, D. & Ponce, J. (2003) *Computer Vision: A Modern Approach*, Prentice-Hall, Upper Saddle River, NJ.

Gantmacher, F. R. (1959) *The Theory of Matrices*, Chelsea, New York.

Golub, G. H. & Van Loan, C. F. (1996) *Matrix Computations*, 3rd edn, The John Hopkins University Press, Baltimore.

Harary, F. (1972) *Graph Theory*, Addison-Wesley, Reading, MA.

Harris, C. & Stephens, M. (1988) A combined corner and edge detector, *in Proc. 4th Alvey Vision Conf.*, pp. 147–151.

Hartley, R. & Zisserman, A. (2003) *Multiple View Geometry in Computer Vision*, 2nd edn, Cambridge University Press, Cambridge.

Leutenegger, S., Chli, M. & Siegwart, R. (2011) BRISK: Binary robust invariant scalable keypoints, *in Proc. IEEE Int. Conf. Computer Vision*.

Levenberg, K. (1944) A method for the solution of certain non-linear problems in least squares, *Quart. Appl. Math.* **2**, 164–168.

Lowe, D. G. (1999) Object recognition from local scale-invariant features, *in Proc. Int. Conf. Computer Vision*, Vol. 2, pp. 1150–1157.

Lowe, D. G. (2004) Distinctive image features from scale-invariant keypoints, *in Int. J. Computer Vision*, Vol. 60, pp. 91–110.

Mac Lane, S. (1998) *Categories for the Working Mathematician*, 2nd edn, Springer, New York.

Marquardt, D. (1963) An algorithm for least-squares estimation of nonlinear parameters, *SIAM J. Appl. Math.* **11**(2), 431–441.

Matas, J., Chum, O., Urba, M. & Pajdla, T. (2002) Robust wide baseline stereo from maximally stable extremal regions, *in Proc. British Machine Vision Conf.*, pp. 384–396.

Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Kadir, T. & Van Gool, L. (2005) A comparison of affine region detectors, *Int. J. Computer Vision* **65**(1–2), 43–72.

Nister, D. & Stewenius, H. (2008) Linear time maximally stable extremal regions, *in Proc. 10th European Conf. Computer Vision*, Vol. 5303 of *Lecture Notes in Computer Science*, Marseille, France, pp. 183–196.

Obdrzalek, D., Basovnik, S., Mach, L. & Mikulik, A. (2009) Detecting scene elements using maximally stable colour regions, *in Communications in Computer and Information Science*, Vol. 82, La Ferte-Bernard, France, pp. 107–115.

Rosten, E. & Drummond, T. (2005) Fusing points and lines for high performance tracking, *in Proc. IEEE Int. Conf. Computer Vision*, Vol. 2, pp. 1508–1511.

Schmidt, G. (2011) *Relational Mathematics*, Vol. 132 of *Encyclopedia of Mathematics and its Applications*, Cambridge University Press, Cambridge.

Shi, J. & Tomasi, C. (1994) Good features to track, *in Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 593–600.

Tarantola, A. (1987) *Inverse Problem Theory: Methods for Data Fitting and Model Parameter Estimation*, Elsevier Science Publishers, Amsterdam.

Appendix A: Figures

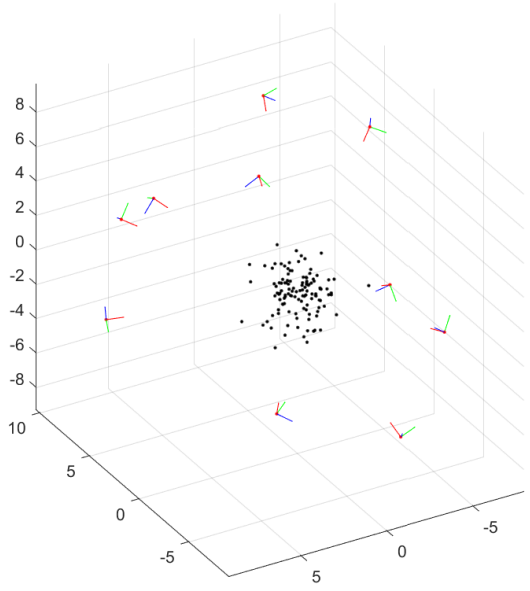

Figure A1: Benchmark model

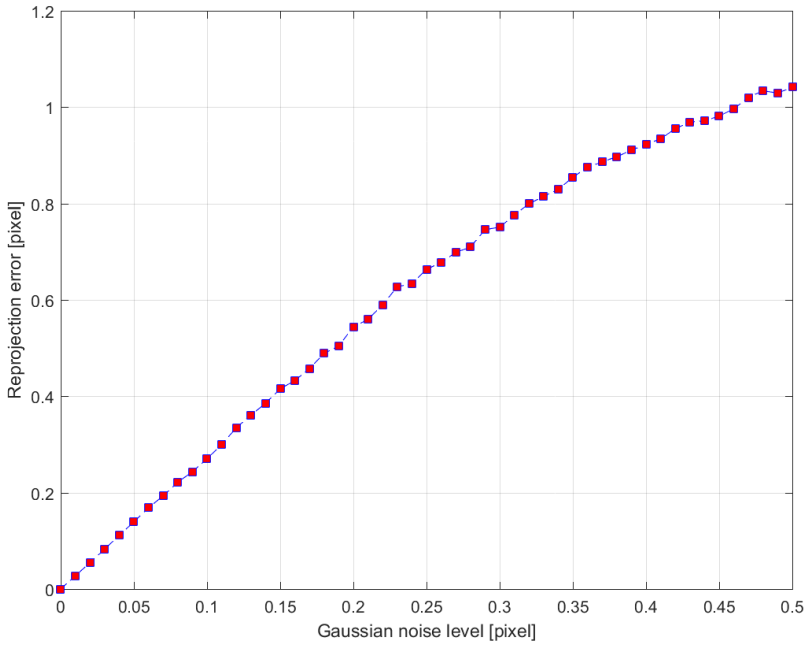

Figure A2: Point reprojection error versus Gaussian noise level

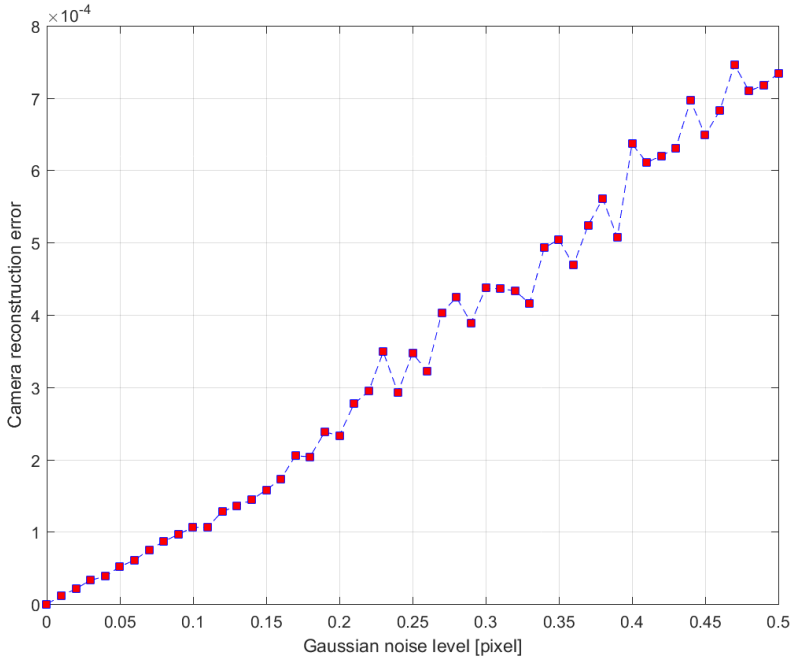

Figure A3: Camera reconstruction error versus Gaussian noise level

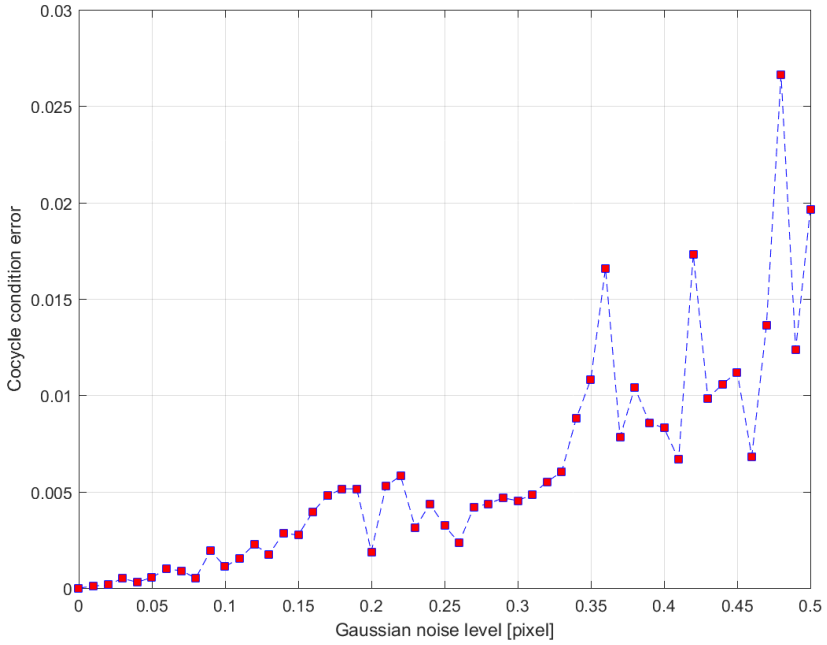

Figure A4: Homography cocycle condition error versus Gaussian noise level

Figure A5: MWIR image of a scene with the building in the Ottawa area

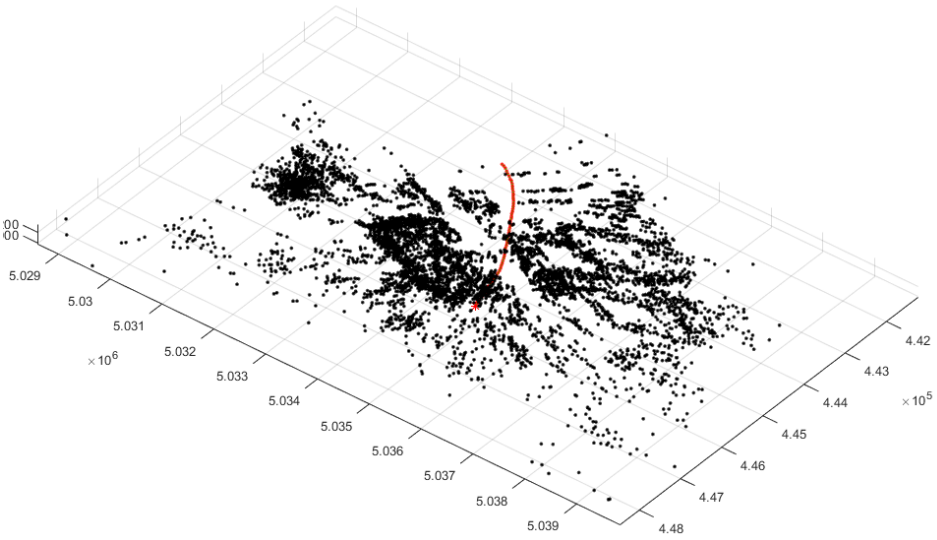

Figure A6: Reconstructed point cloud and camera trajectory (MWIR imagery)

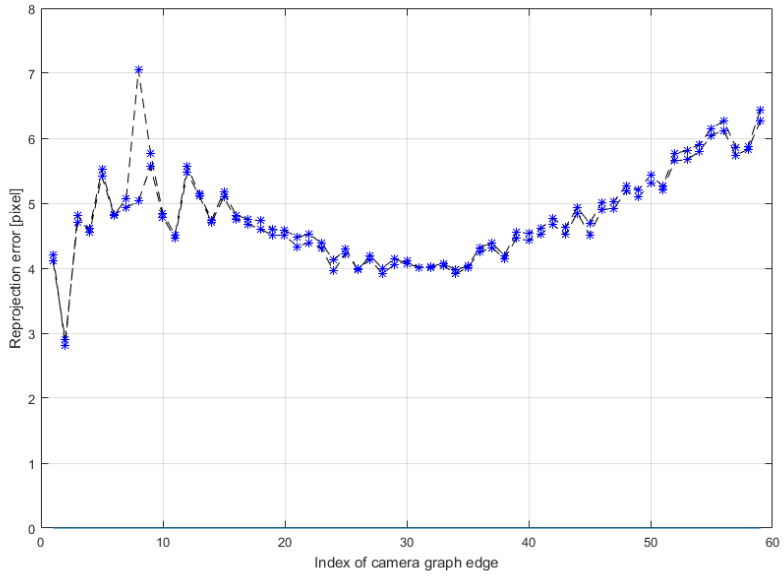

Figure A7: Reprojection error (MWIR imagery)

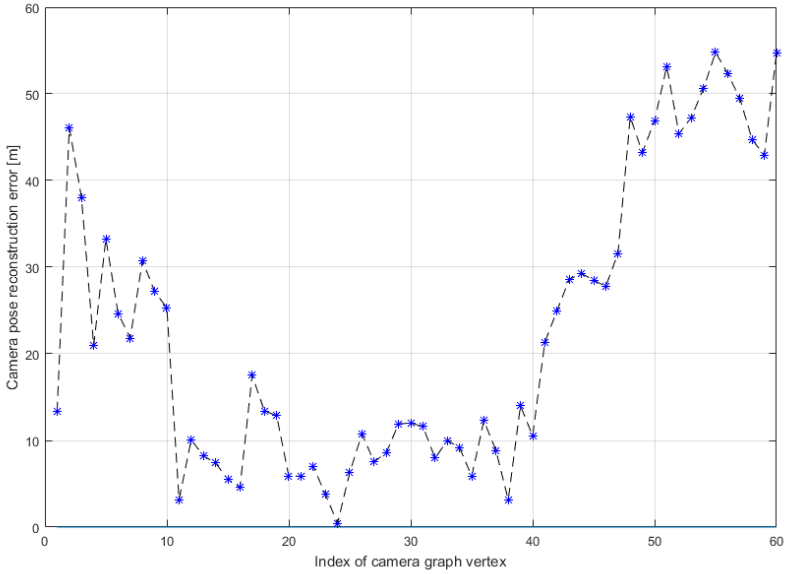

Figure A8: Camera pose reconstruction error (MWIR imagery)

Figure A9: Image of a scene with the building of a church

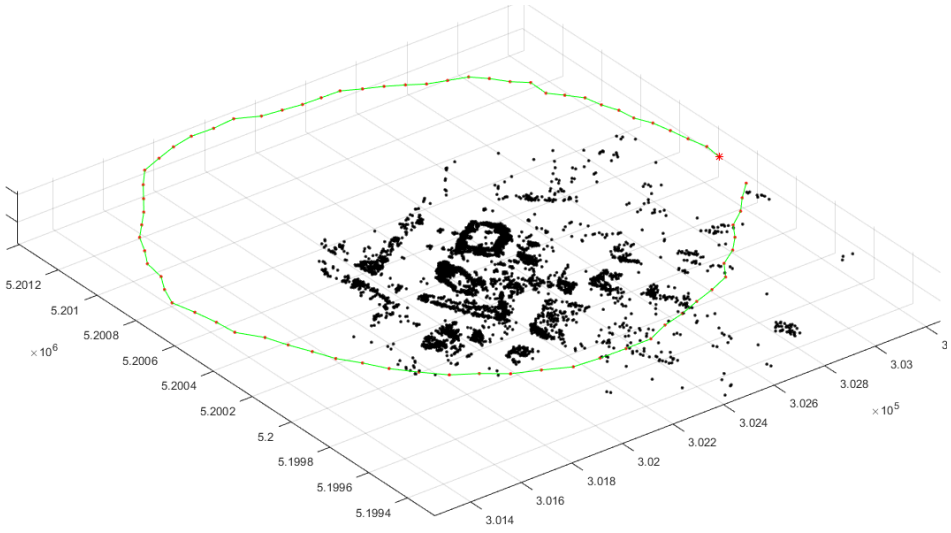

Figure A10: Reconstructed point cloud and camera trajectory (visible-spectrum imagery)

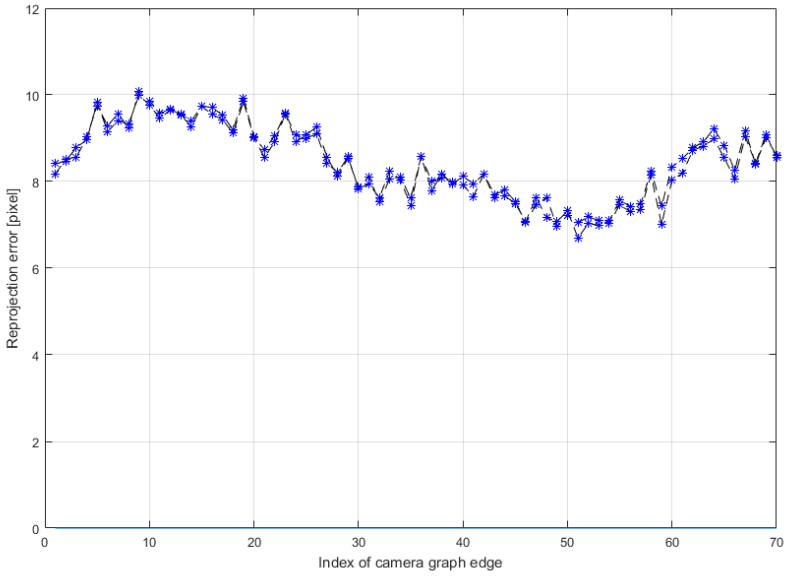

Figure A11: Reprojection error (visible-spectrum imagery)

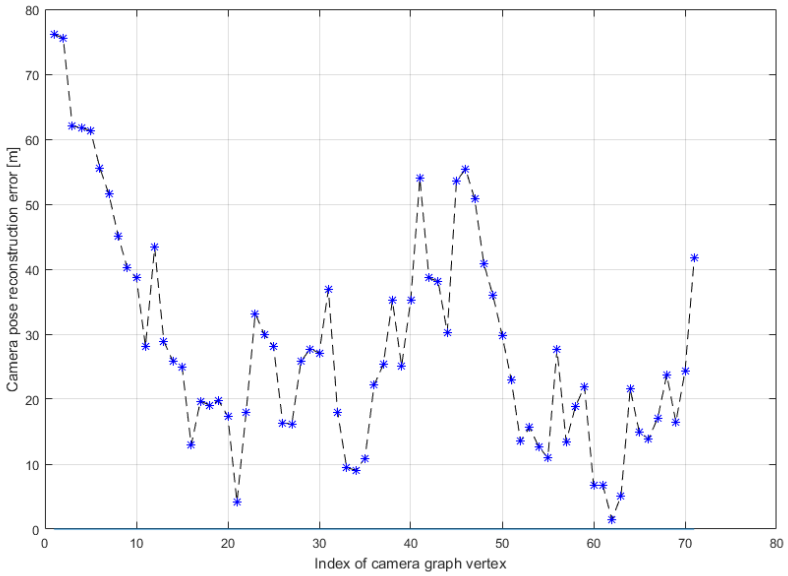

Figure A12: Camera pose reconstruction error (visible-spectrum imagery)

UNCLASSIFIED

DEFENCE SCIENCE AND TECHNOLOGY GROUP DOCUMENT CONTROL DATA	1. DLM/CAVEAT (OF DOCUMENT)

2. TITLE	3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION)
Sparse Reconstruction of a Scene and Camera Poses from the Scene Images with MATLAB®	Document (U) Title (U) Abstract (U)
4. AUTHOR	5. CORPORATE AUTHOR
Leonid K Antanovskii	Defence Science and Technology Group PO Box 1500 Edinburgh, South Australia 5111, Australia

6a. DST Group NUMBER	6b. AR NUMBER	6c. TYPE OF REPORT	7. DOCUMENT DATE
DST-Group–TR–3346	016–810	Technical Report	February, 2017

8. Objective ID	9. TASK NUMBER	10. TASK SPONSOR
AV12713971	AIR07/213	RAAF Air Combat Group

13. DST Group Publications Repository	14. RELEASE AUTHORITY
`http://dspace.dsto.defence.gov.au/dspace/`	Chief, Weapons and Combat Systems Division

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

Approved for public release

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, EDINBURGH, SOUTH AUSTRALIA 5111

16. DELIBERATE ANNOUNCEMENT

17. CITATION IN OTHER DOCUMENTS

No Limitations

18. RESEARCH LIBRARY THESAURUS

Science, Mathematics, Algorithms, Computer Vision, Structure Reconstruction, Camera Calibration

19. ABSTRACT

This report addresses the description and implementation of numerical algorithms for the reconstruction of world points representing a scene, and pinhole camera poses from the scene images. The *Image Processing* and *Computer Vision System* toolboxes of MATLAB are used for detecting, extracting and matching features in images. A camera graph is introduced to indicate which image pairs to process, and a homography graph is derived as a sub-graph of the line graph of the camera graph to parametrize three-dimensional transition homographies. The estimated transition homographies are applied to world points and cameras, locally reconstructed from image pairs, to bring them to a global frame of homogeneous coordinates. Then potentially duplicate points are eliminated using an introduced metric between two projective points with respect to cameras, and a visibility relation for *Bundle Adjustment* is computed. This approach properly addresses the common situation when a point disappears from a camera view and reappears later. Compatibility cocycle conditions for keypoint matching relations over the camera graph cycles and for transition homographies over the homography graph cycles are discussed.

UNCLASSIFIED