Camera Registration from Image Correspondences

Ánoq of the Sun, Hardcore Processing *

December 24, 2009

Quoted as e.g.: Ánoq of the Sun, Ánoq, o., Ánoq, o. t. S. or Ánoq, of the Sun. Not: Sun, Á.
Ánoq is considered the "family name", always written and pronounced first.

Contents

1 Introduction

This report continues the report [Anoq09] but is written so as to be as independent from that report as possible. In the few places where familiarity with the previous report is needed, relevant references will be given. This report is condensed and contains much information. *Section 5 is useful for reference*.

The focus is on *establishing a camera view relationship* between *two images* from *detected* and *matched point correspondences* from input images, *without user input or calibration*. The matched point correspondences are obtained by methods from the previous report. The view relationship will be given in the form of a $3x3$ *fundamental matrix*. It can also be given by a $3x3$ perspective transformation matrix, i.e. a *homography*, or by a rotational model, but these models will not be used, only suggested. The view relationship model can be turned into individual $3x4$ homogenious *camera projection matrices* for each view of the same scene. The previous report contains further introduction to the overall setting.

2 Previous Work

[Hart03] gives a very thorough overview of most methods needed for this project, so it will serve as the primary reference. Particularly chapters 2, 3, 4, 5, 9, 10, 11, 12, 18 and appendices 4, 5 and 6 are important. It presents some commonly used methods for estimating a fundamental matrix, particularly the 8-point algorithm, its 7-point variant, The Gold Standard algorithm and an extension of it, which uses RANdom SAmpling Consensus (RANSAC). These methods rely on numerical methods, such as Singular Value Decomposition (SVD), linear least-squares optimization and non-linear optimization by Levenberg-Marquardt. References such as [Trig99], [Poll00] and [Cyga09] also give relevant overviews and present some of the same methods, although in less complete ways. The article [Chum05] describes PROgressive SAmpling Consensus (PROSAC), an alternative to RANSAC, often with much faster convergence.

[Hart03] also describes multi-view methods. Some are based on extensions on the above, while others consider three or four views simultaneously, e.g. by estimating a trifocal tensor, instead of the fundamental matrix. It also describes specialized methods, such as affine methods or calibrated methods, where a priori assumptions are being made on either the input images, expected scenes or camera calibration. Importantly, it also describes camera auto-calibration methods. [Kana98] and [Trig01] describe model selection methods, e.g. for choosing between estimating a fundamental matrix or a homography, which is relevant, since a homography is only valid for planar scenes and a fundamental matrix becomes degenerate for planar scenes.

References like [Oshe06] and [Mona99] describe alternative algorithms, such as a voting scheme and incremental geometric guidance. The general joint feature correspondences method from [Trig01] is also interesting. These and other alternative methods are not mutually exclusive to the above algorithms and can be used as additional enhancements.

3 Conclusions of Previous Work

A few important conclusions drawn in previous work are:

- Correspondence points should generally be normalized before reconstruction (11.2 and 4.4 [Hart03])
- A quote, which sums up a lot of analysis: "To be certain of getting the best results, if Gaussian noise is a viable assumption, implement the Gold Standard algorithm" (11.5.1 [Hart03])
- Another quote, which sums up a lot of analysis: "Bundle adjustment should generally be used as a final step of any reconstruction algorithm" (18.1 [Hart03])

4 Design Analysis

This section gives the reasons for choosing the selected methods.

The previous report [Anoq09] considered point correspondences between images, so it is natural to use points for the camera view estimation. However, it is also possible to use other primitives, such as lines or conics. It is argued in [Cyga09] (sec. 6.8.3 p. 295) that, more often than not, lines arise out of occlusions between objects. Also, along the direction of a line, there is no fixed location to lock on to, so using lines seems unattractive. A method like the Fast Level Set Transform (FLST) (sec. 7.3.1, [Oshe06]), which was mentioned in the previous report, extracts regions for the entire image, whose edges are not necessarily lines between objects, since they are extracted as changes between increasing versus decreasing intensity gradients. Such edges might be interesting to consider. Similarly, the edges of detected Maximally Stable Extremal Regions (MSERs) [Mata02], as implemented in the previous report, could potentially be more reliable than individual lines, since entire regions are matched between the images. The ellipses, which are commonly used as the measurement regions for affine invariant feature matching, are conics and could be used for camera view estimation as well, which might be interesting. To limit the scope of this report, we will not consider using other primitives than points.

It could be considered to use methods taking advantage of more than two views, such as using the trifocal tensor, but that requires at least three images as input. However, the developed method should also work for two views, so this can at most be relevant as an improvement. To limit the scope of this report, only methods based on two views will therefore be considered.

We will not consider specialized methods, which make assumptions on either input images, scenes or camera calibration, since the goal is to avoid making such assumptions.

According to the conclusions hinted at in section 3, it does not seem appropriate to aim any lower than for the Gold Standard algorithm. For automatic estimation of the fundamental matrix, the extended version of the Gold Standard algorithm using RANSAC, shown as algorithm 11.4 on page 291 in [Hart03], seems like the best starting point.

From the above starting point, we can consider various extensions and improvements, such as geometric guidance or varying the feature detection methods in the individual phases of the pipeline.

5 Definitions and Theory

This section lists some general needed definitions and theory. *Refer to this section when reading formulas.*

5.1 Points, Lines and Transformations

- We often consider two-dimensional **homogenious points** $\mathbf{x} = (x, y, w)^T$, which are equivalent to two-dimensional points by $(x', y')^T = (x/w, y/w)^T$, but where homogenious points with $w = 0$ are also valid, being points at infinity. 3D homogenious points $(x, y, z, w)^T$ are similar

- Two-dimensional homogenious points, e.g. $\mathbf{p}$, can be *transformed* by $3x3$ matrices, e.g. H, by usual matrix multiplication: $\mathbf{p}' = H\mathbf{p}$ (section 2.3 pages 32-33 in [Hart03])

- A $3x3$ matrix is called a **homography** or a **projective transformation**, when it is invertible and preserves lines. Formally, preserving lines means that, any set of three points form a line if and only if their transformed points do (definition 2.9 and pages 32-33 in [Hart03])

- Two-dimensional **homogenious lines**, e.g. $\mathbf{l}$, can be represented by $\mathbf{l} = (a, b, c)^T$, which corresponds to the line satisfying the implicit equation: $ax + by + c = 0$ (sec. 2.2.1 p. 26, [Hart03])

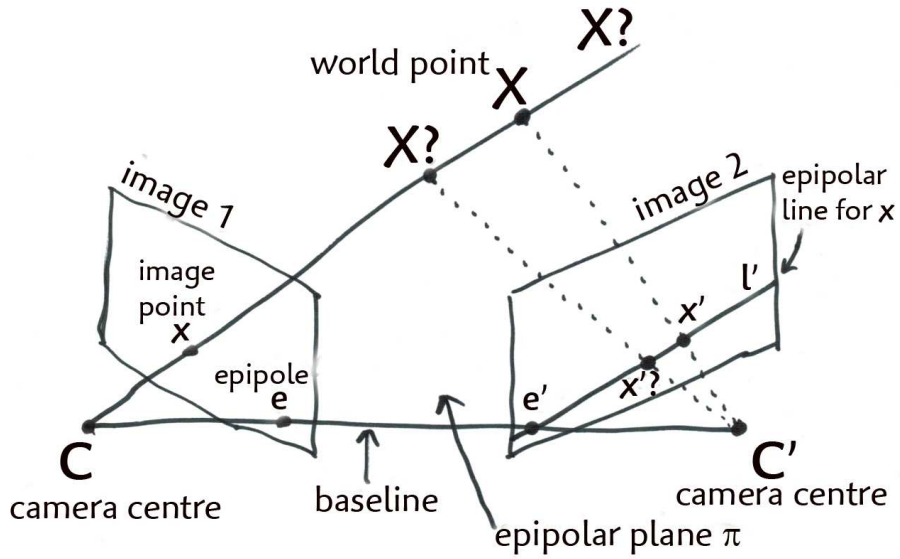

Figure 1: Two images taken of the same 3D scene with two cameras having C and C' as their centres of projection. The baseline joins the two camera centres and intersects the image planes in the epipoles $\mathbf{e}$ and $\mathbf{e}'$. For the point $\mathbf{x}$ in the first image, $\mathbf{l}'$ is its corresponding epipolar line. The point $\mathbf{x}'$ in the second image corresponding to $\mathbf{x}$ in the first image is somewhere on $\mathbf{l}'$, depending on the depth in the first image of the world-point $\mathbf{X}$ corresponding to $\mathbf{x}$. An epipolar plane π is spanned by the baseline and the image point $\mathbf{x}$

- Two-dimensional homogenious *lines*, e.g. $\mathbf{l}$, can be *transformed* by the homography H by: $\mathbf{l}' = H^{-T}\mathbf{l}$. Notice here that, the superscript T is for matrix transposition and the minus in front is for matrix inversion, so the formula uses the transposed of the inverse of H (sec. 2.3.1, [Hart03])
- A homography H between two images of a 3D scene describes a mapping from points *on a plane* in the 3D scene in one image into points on the same plane in the same scene in the other image. Thus, it is a relationship between both the camera views and a plane in the scene
- We can *transform* between two-dimensional homogenious *points*, e.g. $\mathbf{p}$, and *lines*, e.g. $\mathbf{l}$, by $\mathbf{l} = A\mathbf{x}$. The mapping A is a **correlation**, a 3x3 matrix. A *proper correlation* is invertible, i.e. A would be a non-singular matrix (definition 2.29 p. 59 and section 9.2.4 p. 246 in [Hart03])
- If $\mathbf{a} = (a_1, a_2, a_3)^T$ is a 3-vector, then a corresponding **skew-symmetric matrix** $[\mathbf{a}]_\times$ is defined by:

$$[\mathbf{a}]_\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \quad (1)$$

  Matrix $[\mathbf{a}]_\times$ is singular and $\mathbf{a}$ is its null-vector (right or left). The *cross product* is related to skew symmetric matrices by: $\mathbf{a} \times \mathbf{b} = [\mathbf{a}]_\times \mathbf{b} = (\mathbf{a}^T[\mathbf{b}]_\times)^T$ (A4.5 p. 581 in [Hart03])

5.2 Epipolar Geometry and the Fundamental Matrix

- The **baseline** between two images of the same scene is the line between the centres of projection $\mathbf{C}$ and $\mathbf{C}'$ of the two cameras, which took the images (see figure 1) (figure 9.2 p. 240 in [Hart03])
- The **epipole** $\mathbf{e}$ of an image is the point, at which the camera ray, starting from the camera's centre of projection $\mathbf{C}$, hits the centre of projection $\mathbf{C}'$ of the camera of a second image of the same scene. The ray is the baseline between the pair of images (see figure 1)
- An **epipolar plane** π for an image pair is a plane containing the baseline. There exists a one-dimensional family, called a *pencil*, of epipolar planes, all containing the baseline (see figure 1)
- An **epipolar line** is a line $\mathbf{l}'$ in the image, where the image plane intersects an epipolar plane π. Epipolar lines thus always passes through the epipole $\mathbf{e}'$ (see figure 1) (figure 9.2 in [Hart03])
- For two images acquired by cameras with non-coincident centres, the **fundamental matrix** F is defined as the unique 3x3 rank 2 homogenious matrix which satisfies $\mathbf{x}'^T F\mathbf{x} = 0$ for all corresponding image points $\mathbf{x} \leftrightarrow \mathbf{x}'$ (definition 9.4 p. 245 in [Hart03])
- Intuition: A *fundamental matrix* F is a correlation, a 3x3 matrix, which maps a *point* $\mathbf{x}$ in one image of a 3D scene into the corresponding epipolar *line* $\mathbf{l}'$ in a second image of the same scene, expressed by $\mathbf{l}' = F\mathbf{x}$. Points $\mathbf{x}_k$ on the same epipolar line $\mathbf{l}$ in the first image maps to the same line $\mathbf{l}'$, so F is *singular* and of rank 2. F has 7 degrees of freedom: 9 entries with 2 degrees of freedom lost because $\det F = 0$ and the common scaling of entries is insignificant. The true corresponding point $\mathbf{x}'$ in the other image is somewhere on the line $\mathbf{l}'$ (consider $\mathbf{x}'^T\mathbf{l}' = \mathbf{x}'^T F\mathbf{x} = 0$). In this way, the fundamental matrix describes *the relationship between the camera views* in *two images*, *independently of the scene* (see figures 1 and 2) (sections 9.2.3 and 9.2.4 in [Hart03])
- If F is a fundamental matrix between two images of the same 3D scene, where image point correspondences are given by $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ and the epipoles are $\mathbf{e}$ and $\mathbf{e}'$ (see figure 1), then:
  - F is a singular homogenious 3x3 matrix of rank 2 with 7 degrees of freedom, so $\det F = 0$
  - If a point $\mathbf{x}$ in one image corresponds to a point $\mathbf{x}'$ in the other image, then $\mathbf{x}'^T F\mathbf{x} = 0$
  - $\mathbf{x}'^T F\mathbf{x} = 0$ only implies that $\mathbf{x}'$ is somewhere on the epipolar line corresponding to $\mathbf{x}$
  - F^T corresponds to F with the roles of the two images switched
  - $\mathbf{l}' = F\mathbf{x}$ is the epipolar line corresponding to $\mathbf{x}$
  - $\mathbf{l} = F^T\mathbf{x}'$ is the epipolar line corresponding to $\mathbf{x}'$
  - $F\mathbf{e} = 0$ and $F^T\mathbf{e}' = 0$. Thus, the left and right null-spaces of F are generated by the vectors representing (in homogenious coordinates) the two epipoles $\mathbf{e}$ and $\mathbf{e}'$ in the two images
  - Two *epipolar lines* $\mathbf{l}$ *and* $\mathbf{l}'$ *correspond* $\mathbf{l} \leftrightarrow \mathbf{l}'$ if they are formed by intersection of the same epipolar plane π with the two image planes
  - If $\mathbf{l}$ and $\mathbf{l}'$ are corresponding epipolar lines and $\mathbf{k}$ is any line not passing through the epipole $\mathbf{e}$, then $\mathbf{l}$ and $\mathbf{l}'$ are related by: $\mathbf{l}' = F[\mathbf{k}]_\times\mathbf{l}$. Symmetrically: $\mathbf{l} = F^T[\mathbf{k}']_\times\mathbf{l}'$.
    A convenient choice for $\mathbf{k}$ is $\mathbf{e}$ (and similarly $\mathbf{e}'$ for $\mathbf{k}'$), so: $\mathbf{l}' = F[\mathbf{e}]_\times\mathbf{l}$ and $\mathbf{l} = F^T[\mathbf{e}']_\times\mathbf{l}'$
  - F may be written as: $F = [\mathbf{e}']_\times H_\pi$, where H_π is the transfer mapping (such as an image homography) from the first image to the second via any plane π. Since $[\mathbf{e}']_\times$ has rank 2 and H_π has rank 3, F has rank 2 (result 9.1 p. 243 in [Hart03])

  (most of the above can be found in sections 9.2.4 and 9.2.5 p. 245-247 in [Hart03])

5.3 The Fundamental Matrix and Camera Matrices

- If P and P' are the two 3x4 **camera projections** for two images of the same 3D scene, then:
  - The **camera centres** $\mathbf{C}$ and $\mathbf{C}'$ are defined by: $P\mathbf{C} = \mathbf{0}$ and $P'\mathbf{C}' = \mathbf{0}$ ($\mathbf{0}$ are zero vectors)
  - The **pseudo inverse** P^+ of P is defined by: $PP^+ = I$ (where I is the identity matrix)
  - The *epipole* $\mathbf{e}'$ in the *second image* is given by: $\mathbf{e}' = P'\mathbf{C}$
  - Fundamental matrix F from **general cameras** (with differing camera centres): $F = [\mathbf{e}']_\times P'P^+$
  - Fundamental matrix F from **canonical cameras** $P = [I|\mathbf{0}]$ [1] and $P' = [M|\mathbf{m}]$, which have *projection centres at infinity*: $F = [\mathbf{m}]_\times M = [\mathbf{e}']_\times M = M^{-T}[\mathbf{e}]_\times$ (notice that M^{-T} is the transpose of the inverse of M), where $\mathbf{e}' = \mathbf{m}$ and $\mathbf{e} = M^{-1}\mathbf{m}$
  - Fundamental matrix F from **cameras not at infinity** $P = K[I|\mathbf{0}]$ and $P' = K'[R|\mathbf{t}]$: $F = K'^{-T}[\mathbf{t}]_\times RK^{-1} = [K'\mathbf{t}]_\times K'RK^{-1} = K'^{-T}RK^T[KR^T\mathbf{t}]_\times$

  (most of the above can be found in table 9.1 p. 246 in [Hart03])
- The 3x4 *camera matrices* P and P' corresponding to a fundamenal matrix F may be chosen as: $P = [I|\mathbf{0}]$ and $P' = [[\mathbf{e}']_\times F|\mathbf{e}']$, where $\mathbf{e}'$ is the epipole in the second image (result 9.14 p. 256 in [Hart03])
- The general formula for a pair of canonical 3x4 camera matrices P and P' corresponding to a fundamental matrix F is given by: $P = [I|\mathbf{0}]$ and $P' = [[\mathbf{e}']_\times F + \mathbf{e}'\mathbf{v}^T|\lambda\mathbf{e}']$, where $\mathbf{v}$ is any 3-vector and λ a non-zero scalar (result 9.15 p. 256 in [Hart03])
- The **projective camera ambiguity**, when given a fundamental matrix F, can be fully described as: Let F be a fundamental matrix and let (P, P') and $(\tilde{P}, \tilde{P}')$ be two pairs of camera matrices such that F is the fundamental matrix corresponding to each of these pairs. Then there exists a non-singular 4x4 matrix H such that $\tilde{P} = PH$ and $\tilde{P}' = P'H$ (theorem 9.10 p. 254 in [Hart03])
- If the camera internal calibration is known, the Euclidean camera motion between views may be computed from the fundamental matrix up to a finite number of ambiguities (chapter 9, [Hart03])

6 The Implemented Methods

This section is quite long and describes the implemented methods and suggested improvements.

6.1 Coordinate Normalization

For each input image, we shall normalize the coordinates of the detected feature points according to section 4.4 in [Hart03]. This in some sense gives a canonical coordinate system for the points, which helps minimizing the divergence caused by the inaccuracies of the feature points, when computing the fundamental matrix. The normalization consists of:

- Translate all feature points, such that their centroid $\mathbf{c} = (x_c, y_c)$ (i.e. average position) becomes the new coordinate origin $(0, 0)$
- Then scale all feature points, such that their average distance to the (new) origin becomes $\sqrt{2}$

[1] I is the 3x3 identity matrix and $\mathbf{0}$ the null 3-vector

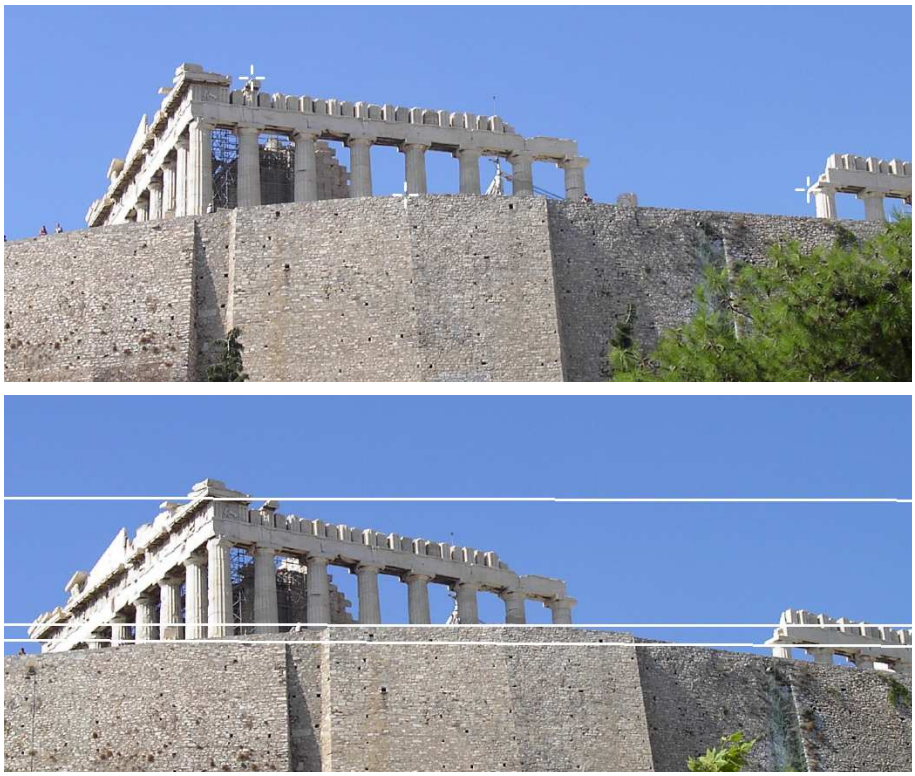

Figure 2: A zoomed-in view of images 6 and 7 of the Athens *Ακρόπολη* image sequence photographed by the author in August 2002. This image pair is further presented in figure 5. A fundamental matrix was estimated between these images by manual selection of 8 point correspondences, followed by execution of the normalized 8-point algorithm from section 6.3, but without the singularity constraint enforcement from section 6.2.1. The three points illustrated by cross-hairs in the upper image were additionally selected manually and the three lines in the lower image are their corresponding epipolar lines, computed with the estimated fundamental matrix. Notice in particular the point at the base of the 5th column of the side of the temple. This point could be both on that column in the scene, as well as on the tip of the large wall. In the second image, this can be seen by the epipolar line passing through both of these potentially matching points (count the number of columns and look at where the wall bends to realize this). When estimating the fundamental matrix with the normalized 8-point algorithm, the singularity constraint enforcement is so crude that this property does not hold any more for that line; in fact, all the epipolar lines become quite inaccurate (no pictures of this will be shown). The price paid in the images shown here is that the epipolar lines do not cross at a single point, as they should (if the fundamental matrix had had rank 2). It should be noted that the epipolar lines estimated in figure 5 for the same image pair are also not entirely accurate

This normalization is done separately for each input image. In this report, the second step, scaling, will be done isotropically by a factor $\frac{1}{s}$. Non-isotropic scaling using moments should not give any significant improvements, but a method known as *Total Least Squares - Fixed Columns*, should give a slight improvement of the fundamental matrix estimation (section 4.4.4 in [Hart03]), so this could be considered for future work.

The normalizing transformation will be used for denormalization of the estimated fundamental matrix, where the normalization and denormalization can be expressed in formulas by:

- Let the image correspondences be given by $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$
- Normalization: The image coordinates are transformed independently for the two input images by: $\hat{\mathbf{x}}_i = T\mathbf{x}_i$ and $\hat{\mathbf{x}}'_i = T'\mathbf{x}'_i$, where $T = \begin{bmatrix} \frac{1}{s} & 0 & 0 \\ 0 & \frac{1}{s} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -x_c & 0 & 0 \\ 0 & -y_c & 0 \\ 0 & 0 & 1 \end{bmatrix}$ and $T' = \begin{bmatrix} \frac{1}{s'} & 0 & 0 \\ 0 & \frac{1}{s'} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -x'_c & 0 & 0 \\ 0 & -y'_c & 0 \\ 0 & 0 & 1 \end{bmatrix}$ are the relevant normalizing transformations
- Denormalization: If $\hat{F}'$ is the fundamental matrix estimated on the normalized points, then $F = T'^T \hat{F}' T$ is the fundamental matrix corresponding to the original data $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$

6.2 Singularity Constraint

As specified in section 5.2, the fundamental matrix should be singular and of rank 2. When estimating a fundamental matrix, this property is normally used as a constraint. This section presents a crude way of enforcing this constraint.

6.2.1 Singularity Constraint by Fundamental Matrix Correction

Assume that a fundamental matrix F has been computed, e.g. by the normalized 8-point algorithm, as described below in section 6.3. F can be replaced by F', where F' minimizes the Frobenious norm $\|F - F'\|$ subject to the condition $\det F' = 0$. This can be done by Singular Value Decomposition (SVD) (see [Pres92]) as follows (from section 11.1.1 in [Hart03]):

- Compute SVD $F = UDV^T$, where D is a diagonal matrix $D = diag(r, s, t)$
- Assume that $r \geq s \geq t$. The rows and columns of U, D and V can be perturbed to achieve this
- Then $F' = Udiag(r, s, 0)V^T$ minimizes the Frobenious norm $\|F - F'\|$

As an alternative to replacing t by 0 and perturbing matrix rows and columns, in case $r \geq s \geq t$ does not hold, we can also find the smallest of r, s and t and replace that by 0. This is the way that it has been implemented for this report.

6.3 The Normalized 8-Point Algorithm

One of the simplest and most direct ways of estimating the fundamental matrix F from a set of point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ is to use the normalized 8-point algorithm, which uses linear least squares optimization. The algorithm works as follows (from section 11.2 and algorithm 11.1 in [Hart03]):

- Normalize the coordinates by $\hat{\mathbf{x}}_i = T\mathbf{x}_i$ and $\hat{\mathbf{x}}'_i = T'\mathbf{x}'_i$, as described in section 6.1
- Find the matrix $\hat{F}'$ from the correspondences $\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i$ by:
  - Linear least squares optimization: Find $\hat{F}$ from $\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i$, as described below in section 6.3.1
  - Singularity constraint enforcement: Compute $\hat{F}'$ from $\hat{F}$, as described in section 6.2.1
- Denormalize the estimated matrix by $F = T'^T \hat{F}' T$, as described in section 6.1

At least 8 points are needed for this algorithm, but more points contribute to a better solution.

6.3.1 Fundamental Matrix by Linear Least Squares Optimization

The equations $\mathbf{x}_i'^T F \mathbf{x}_i = 0$, which must hold for the point correspondences for a fundamental matrix (as defined in section 5.2), give rise to a set of equations, which can be represented in matrix form as $A\mathbf{f} = \mathbf{0}$ (section 11.1 in [Hart03]):

$$\begin{bmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_nx_n & x'_ny_n & x'_n & y'_nx_n & y'_ny_n & y'_n & x_n & y_n & 1 \end{bmatrix} \mathbf{f} = \mathbf{0} \tag{2}$$

Here $\mathbf{f}$ is the 9-vector made up of the entries of the fundamental matrix F in row-major order, i.e. the entries are ordered such that when $F = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$ then $\mathbf{f} = (a, b, c, d, e, f, g, h, i)$. The least squares solution for $\mathbf{f}$ is the singular vector corresponding to the smallest singular value of A. $\mathbf{f}$ can be obtained from A by Singular Value Decomposition (SVD) (see [Pres92]) as follows (sec. 11.1, [Hart03]):

- Compute SVD $A = UDV^T$
- Then $\mathbf{f}$ is the column of V, which corresponds to the smallest singular value of A

6.4 A 7-Point Algorithm and Degeneracies

The way of enforcing the singularity constraint described in section 6.2.1 is quite crude. The following section describes one way of improving this, resulting in an algorithm for estimating the fundamental matrix F using only 7 point correspondences. Various degenerate cases will be considered afterwards.

6.4.1 The 7-Point Algorithm

The 8-point algorithm assumes that the matrix A (from section 6.3.1) has rank 8, which is true if the points are in general position or are noisy, but it may even have rank 9 (section 11.1 in [Hart03]). It finds the fundamental matrix up to scale. By using only 7 point correspondences, the matrix A becomes a 7x9 matrix, generally of rank 7. The solution $\mathbf{f}$ for $A\mathbf{f} = \mathbf{0}$ becomes a two-dimensional space, where the singularity constraint can be used to both find the final solution, as well as for detecting some degeneracies. The solution to $A\mathbf{f} = \mathbf{0}$ is found as follows (from section 11.1.2 in [Hart03]):

- Find the two generators $\mathbf{f}_1$ and $\mathbf{f}_2$ for the right null-space of A. This is done by Singular Value Decomposition (SVD), as in section 6.3.1, where $\mathbf{f}_1$ and $\mathbf{f}_2$ are the last two columns of V, which correspond to the two smallest singular values. This gives two matrices F_1 and F_2 from the coefficients of $\mathbf{f}_1$ and $\mathbf{f}_2$ in row-major order

- The two-dimensional solution space is $\alpha F_1 + (1-\alpha)F_2$, where α is a scalar value
- The singularity constraint $\det A = 0$ is written as $\det(\alpha F_1 + (1-\alpha)F_2) = 0$. This gives a cubic polynomial in α. It is somewhat involved to write down the coefficients, so this will not be shown
- Find the one or three real solutions of the polynomial; discard any complex solutions. The solutions can be computed analytically by closed-form formulas, see e.g. [CubicP]
- Substitute back the polynomial solution values into $F = \alpha F_1 + (1-\alpha)F_2$, which gives one or three possible fundamental matrices. We will consider these solutions in the next section

This algorithm should also use normalized coordinates, as the normalized 8-point algorithm. Unlike the 8-point algorithm, which allows using more than eight point correspondences, the 7-point algorithm should only be used in the case where exactly seven points are known.

6.4.2 Degeneracies

Various configurations of point correspondences can lead to degenerate cases for the fundamental matrix estimation. Several of the cases can be classified by considering the dimension of the null-space of the matrix A from section 6.3.1, even though this is not enough to *robustly* detect degeneracies. Let the dimension of this null-space be denoted by $\dim(N)$, then a list summarizing the degenerate cases is given here (mostly from section 11.9 in [Hart03]):

- $\dim(N) = 0$: The matrix is *not singular* and is therefore not a proper fundamental matrix. This situation can be avoided by e.g. the singularity constraint (section 6.2)
- $\dim(N) = 1$: In this case there is *no degeneracy* and the fundamental matrix is uniquely determined. This happens when $n \geq 8$ point correspondences in general position are given. If $n > 8$ then the point correspondences must be perfect, i.e. noise-free (otherwise $\dim(N)$ becomes zero)
- $\dim(N) = 2$: There are *either 1 or 3 solutions* for the fundamental matrix. This happens when seven point correspondences are given, as in the 7-point algorithm. Seven points and the two camera centres always define a quadric surface, since such a surface has nine degrees of freedom. If this quadric surface is a ruled surface (which will not be defined here), then there are three possible solutions for the fundamental matrix, otherwise the solution is unqiue. More than seven perfect noise-free points may also lie on a ruled quadric and give three solutions. The case where there is only one solution is not a problem. If there are three solutions, all three solutions should be examined and the best one chosen, e.g. as done in the RANSAC algorithm (section 6.7)
- $\dim(N) = 3$: There is a *two-parameter family of solutions* for the fundamental matrix. This can happen for six or more perfect point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$, which are related by a homography, i.e. $\mathbf{x}'_i = H\mathbf{x}_i$. This usually happens in one of the following two cases:
  - There is no translation between the images, i.e. there is at most a rotation about the camera centre between the images. This is an important *degenerate camera motion*
  - All point correspondences lie on a plane in the scene, an important *degenerate scene structure*

As seen from this list, only the case $\dim(N) = 3$ needs special consideration in relation to the methods described in this report. This case is not handled in the implementation, but can be handled by estimating a homography (for planar or near-planar scenes), or even a rotational model (for coincident

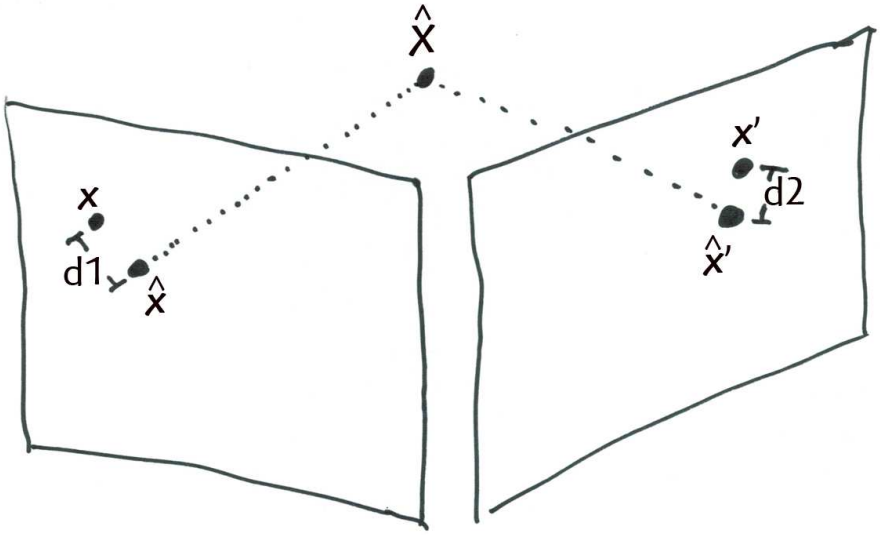

Figure 3: The geometric reprojection distance of a point correspondence $\mathbf{x} \leftrightarrow \mathbf{x}'$ with respect to a fundamental matrix F. We will use this in the fundamental matrix estimation algorithms. The fundamental matrix F is used to reconstruct an estimated 3D world point $\hat{\mathbf{X}}$ from the corresponding points $\mathbf{x}$ and $\mathbf{x}'$, using the triangulation methods from section 6.6. The estimated point $\hat{\mathbf{X}}$ is projected back into the image points $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$, as also described in section 6.6. The geometric reprojection distance is the sum of the two squared distances $d1^2$ ($d1 = d(\mathbf{x}, \hat{\mathbf{x}})$ is the distance between the points $\mathbf{x}$ and $\hat{\mathbf{x}}$ in the first image) and $d2^2$ ($d2 = d(\mathbf{x}', \hat{\mathbf{x}}')$ is the distance between the points $\mathbf{x}'$ and $\hat{\mathbf{x}}'$ in the second image)

or near-coincident camera centres), instead of a fundamental matrix. The references [Kana98], [Trig01] and [Torr95] are relevant here, as well as chapter 22 from [Hart03]. In particular, [Kana98] suggests to first test if a rotational model should be used, and if not, test if a homography model should be used, and if not, use the fundamental matrix model. Such model selection is suggested for future work, where slightly more detailed suggestions will be given in 6.13. However, the methods described in this report should actually still result in some fundamental matrix being estimated, so this limitation should not give any serious problems.

6.5 Error Measures

This section describes some error measures, two of which we will need.

6.5.1 Geometric Reprojection Error Measure for Minimization

When striving for better optimization algorithms than the normalized 8-point algorithm, we need an error measure. The error measure will be used for minimization and should be defined such that it minimizes the error of the estimated fundamental matrix. The error measure, which is used for the Gold Standard algorithm and its automated version, is called geometric reprojection error and is defined as:

$$\sum_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2 \tag{3}$$

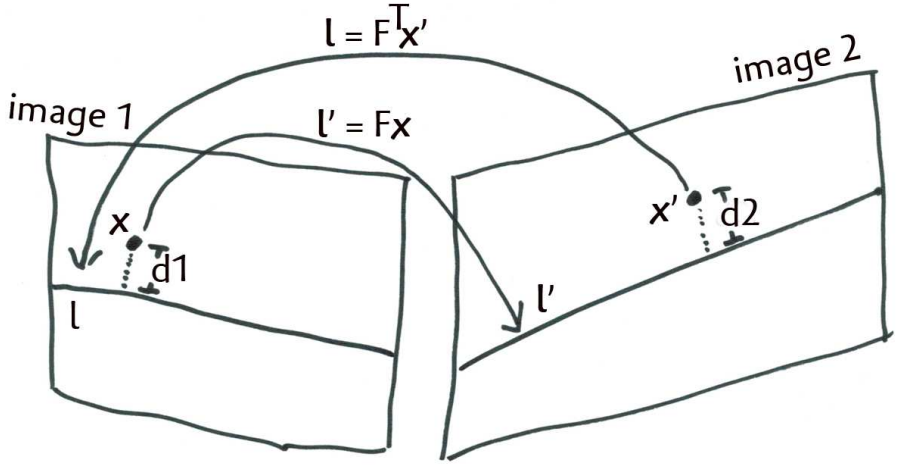

Figure 4: The symmetric epipolar distance of a point correspondence $\mathbf{x} \leftrightarrow \mathbf{x}'$ with respect to a fundamental matrix F. We will use this as a quality measure of a fundamental matrix estimation algorithms. The epipolar lines $\mathbf{l}'$ and $\mathbf{l}$ corresponding to the points $\mathbf{x}$ and $\mathbf{x}'$, respectively, are found by $\mathbf{l} = F^T\mathbf{x}'$ and $\mathbf{l}' = F\mathbf{x}$. The symmetric epipolar distance is the sum of the two squared distances $d1^2$ ($d1 = d(\mathbf{x}, \mathbf{l})$ is the distance between the point $\mathbf{x}$ and the line $\mathbf{l}$ in image 1) and $d2^2$ ($d2 = d(\mathbf{x}', \mathbf{l}')$ is the distance between the point $\mathbf{x}'$ and the line $\mathbf{l}'$ in image 2). Later on, we will also be using the average of these two squared distances and then take the square root (over multiple correspondences), to get the root mean squared (RMS) distance in pixels. This will be explained in section 7.3

The two distance terms in this formula are illustrated in figure 3. The point correspondences are given as $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$. The correspondences $\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i$ are computed by reprojection, such that they satisfy $\hat{\mathbf{x}}_i'^T F \hat{\mathbf{x}}_i = 0$ exactly for some rank-2 matrix F. Thus, $\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i$ in some sense estimate the true correspondences for a given F. The reprojection involves estimating a projective 3D reconstruction, which will be described in section 6.6.

6.5.2 Alternative Error Measures for Minimization

An alternative error measure to one the above could be to use the symmetric epipolar distance, i.e.:

$$\sum_i d(\mathbf{x}_i, F^T\mathbf{x}'_i)^2 + d(\mathbf{x}'_i, F\mathbf{x}_i)^2 \tag{4}$$

where $d(\mathbf{x}, \mathbf{l})$ is the distance between a point $\mathbf{x}$ and a line $\mathbf{l}$. The two distance terms in this formula are illustrated in figure 4. However, using this for minimization gives slightly inferior results, according to section 11.4.3 in [Hart03], so even if this is tempting, it should not be used.

Another first order approximation, called the Sampson distance, could be used though, as also described in section 11.4.3 in [Hart03]. An advantage of this error measure is that, it can be used for doing a simpler optimization than the Gold Standard method, but that will not be considered here.

6.5.3 Residual Error Measure for Determining the Accuracy of Results

The symmetric epipolar distance error measure mentioned above was the following:

$$\sum_i d(\mathbf{x}_i, F^T\mathbf{x}'_i)^2 + d(\mathbf{x}'_i, F\mathbf{x}_i)^2 \tag{5}$$

Again, $d(\mathbf{x}, \mathbf{l})$ is the distance between a point $\mathbf{x}$ and a line $\mathbf{l}$ and the two distance terms in this formula are illustrated in figure 4. Although (as explained in the previous section) this should never be used for minimization, it is useful as a measure of the accuracy of the fundamental matrix estimation algorithms. This accuracy is measured in terms of the point correspondences, which were used for estimating the fundamental matrix. Since the input point correspondences are normally not precise, this error measure cannot be expected to be zero in practice.

When using this for measuring the accuracy of results, it should only be used on points which are classified as inliers for the fundamental matrix estimation, which is relevant when considering automated methods, such as the RANSAC method described in section 6.7.

6.6 Projective Reconstruction by Triangulation

This section describes how to do triangulation, to get a projective 3D reconstruction $\hat{\mathbf{X}}_i$ of image correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$, given camera matrices P and P' for the two images. The challenge is that image correspondence points are usually skew, with respect to the camera matrices, so they don't meet at a point in 3-space. Also, the methods must be projectively invariant and concepts like distance and angles are not well-defined in the projective space, so we cannot optimize using only 3-space.

6.6.1 Linear Triangulation

Given a fundamental matrix, the 3x4 homogenious camera projection matrices P and P' can be computed by the formulas found in sections 5.2 and 5.3. The canonical camera matrices with projection centres at infinity are used in the implementation.

The method described next is from section 12.2 page 312 in [Hart03]. A given point correspondence $\mathbf{x}_i = (x_i, y_i) \leftrightarrow \mathbf{x}'_i = (x'_i, y'_i)$ ideally corresponds to a measurement, which can be be represented by the two image projections $\mathbf{x}_i = P\mathbf{X}$ and $\mathbf{x}'_i = P'\mathbf{X}$, for some 3D homogenious point $\mathbf{X}$. Going the opposite way, we estimate the 3D homogenious point $\hat{\mathbf{X}}_i$ from the correspondences by linear approximation by solving an equation of the form $A_i\hat{\mathbf{X}}_i = \mathbf{0}$, where A_i is this 4x4 matrix:

$$A_i = \begin{bmatrix} x_i\mathbf{p}^{3T} - \mathbf{p}^{1T} \\ y_i\mathbf{p}^{3T} - \mathbf{p}^{2T} \\ x'_i\mathbf{p}'^{3T} - \mathbf{p}'^{1T} \\ y'_i\mathbf{p}'^{3T} - \mathbf{p}'^{2T} \end{bmatrix} \tag{6}$$

The vector $\mathbf{p}^{jT}$ is the jth row of P and $\mathbf{p}'^{jT}$ the jth row of P', so their individual four coordinates give rise to the four columns of A_i. We find $\hat{\mathbf{X}}_i$ by minimizing $\|A_i\hat{\mathbf{X}}_i\|$ subject to $\|\hat{\mathbf{X}}_i\| = 1$. The least squares solution for $\hat{\mathbf{X}}_i$ is the singular vector corresponding to the smallest singular value of A_i. As is now becoming a habit, this is found by Singular Value Decomposition (SVD) by computing SVD $A_i = U_i D_i V_i^T$ and taking $\hat{\mathbf{X}}_i$ to the be column of V_i corresponding to the smallest singular value of A_i. Each $\hat{\mathbf{X}}_i$ is individually reconstructed in this way from a point correspondence $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$.

As usual, this reconstruction should preferably be done on point correspondences having normalized image coordinates, as described in section 6.1.

As an alternative to the above method, an inhomogenious linear method exists, described in section 12.2 [Hart03]. None of these two methods are projectively invariant though, so none of them are ideal. The inhomogenious method is affine invariant though, which would be useful for affine reconstruction, but still not any better for our purposes of projective reconstruction. The inhomogenious method however, does not allow the reconstructed 3D point to lie at infinity, which makes it slightly less desirable here, since projectively reconstructed points may lie on the plane at infinity, a concept which

will not be described in this report. Both of these methods generalize easily to multiple, i.e. more than two, views.

The optimal triangulation method mentioned below requires one of these two triangulation methods and as just argued, the homogenious one described here is the most appropriate.

6.6.2 Maximum Likelihood and Optimal Triangulation

There exists a provably optimial triangulation method, under the assumption that the noise has a Gaussian distribution. This method is a non-iterative algorithm, which requires finding the minimum of a 6th degree polynomial, as described in section 12.5 and algorithm 12.1 in [Hart03]. Using this method would be the best way of doing triangulation, but for the reasons described next and in order to limit the extent of this report, it will only be recommended for future work.

It is possible to find the Maximum Likelihood Estimate for the 3D points of the triangulation by a non-linear optimization method like Levenberg-Marquardt (section 12.3 in [Hart03]). As it turns out, this will actually be part of the optimization done by the Gold Standard optimization, which will be described later in 6.9. Hence, by the use of the Gold Standard method, we will actually be doing this optimization. Therefore, only the homogenious linear triangulation method from the previous section has been implemented, for the initialization of the Gold Standard optimization.

6.6.3 Reprojected Points at Infinity

In the above methods, the reconstructed homogenious 3D points could end up being at infinity, which will be fine here. However, if the resulting projected homogenious 2D point in any of the views somehow becomes infinite, we need to be careful when using it for computing the geometric reprojection error from section 6.5.1. The reason is that we need to divide by the last coordinate of the reprojected 2D homogenious point, in order to compute the squared distance to its original measured point. One way to handle this is to choose the distance to a reprojected point at infinity as being the distance equivalent to the maximum inlier search distance, which will be 3 pixels in each image, as argued later in section 6.7.3. This penalises the point, but no more than the maximum penalty that any other point could have, which seems like a sensible strategy here.

6.7 Robust Automated Fundamental Matrix Estimation by Using RANSAC

When estimating a fundamental matrix, we start from an automatically detected set of point correspondences. The challenge arising from this is that the point correspondences are not necessarily all correct. In fact, many of them may be wrong. The statistical sampling method that we use for finding the correct correspondences is RANdom SAmpling Consensus (RANSAC) [Fisc81].

6.7.1 The RANSAC Algorithm

RANSAC works by randomly taking out a sample set S_i of point correspondences, estimating a fundamental matrix F_i from those and checking how many of all the point correspondences are consistent with the estimated matrix F_i. This process is iterated N times, in search of the estimated fundamental matrix F_*, which were consistent with the largest number of point correspondences. When using the 7-point algorithm described in section 6.4.1, the size s of the sample set needs to be only seven point correspondences. The number N of iterations may be determined adaptively. The criterion for choosing N should ensure a very high chance of having found the fundamental matrix with the largest support,

i.e. being consistent with the largest number of point correspondences. The algorithm is as follows (sections 4.7.1 p. 117-121 and 11.6 p. 290-291 in [Hart03]):

- Initially set $N = \infty$ (total number of iterations) and $i = 0$ (completed iterations)
- Iterate the following, while $N > i$:
  - Randomly select a sample set S_i of seven point correspondences. See the description below
  - Estimate the fundamental matrix F_i from S_i. When using the 7-point algorithm, up to three fundamental matrices F_{i_j} may be estimated. For each estimated F_{i_j} do the following:
    - Compute the distance pairs $(d1_{i_{j_k}}, d2_{i_{j_k}})$ of the putative correspondences $\mathbf{x}_k \leftrightarrow \mathbf{x}'_k$ from F_{i_j} and the paired reprojection error, explained in equation 8 in section 6.7.3
    - Determine the inliers, i.e. the point correspondences consistent with F_{i_j}. These are the correspondences where $d1_{i_{j_k}} < t_1$ and $d2_{i_{j_k}} < t_2$. Section 6.7.3 describes the distance thresholds t_1 and t_2. Using two thresholds here differs from algorithm 11.4 in [Hart03]
    - If the number of inliers is higher than any previous number of inliers, these inliers and the matrix F_{i_j} are now remembered as the best set of inliers and the best fundamental matrix $F_* = F_{i_j}$. In case of an equal number of inliers, the set with the lowest standard deviation of inlier distances is chosen. The standard deviation calculation is computed by equation 10 in section 6.7.3, where the two distances $d1_{i_{j_k}}$ and $d2_{i_{j_k}}$ from the distance pairs are added as $d_{i_{j_k}} = d1_{i_{j_k}} + d2_{i_{j_k}}$
  - Set $\epsilon = 1-$(best number of inliers)/(total number of point correspondences)
  - Adaptively determine N from ϵ by the formula below in equation 7 with $p = 0.99$
  - Increment i by 1 for the next iteration

When selecting a sample of seven point correspondences, we should try to avoid selecting point correspondences, which are very close in both images, since this may harm the estimate of the fundamental matrix, when only seven points are used. One way to avoid this is by selecting the point correspondences one by one and for each new correspondence $\mathbf{x}_l \leftrightarrow \mathbf{x}'_l$ considered, check that for all other correspondences $\mathbf{x}_{l_i} \leftrightarrow \mathbf{x}'_{l_i}$ chosen so far, the image points are at least 3 pixels apart in at least one of the images. I.e. $d(\mathbf{x}_l, \mathbf{x}_{l_i}) > 3$ pixels or $d(\mathbf{x}'_l, \mathbf{x}'_{l_i}) > 3$ pixels. If only the point in one image is below this threshold, one of the correspondences could be wrong, but we have no idea which one, so we have to allow this and keep both in that case.

Remember to compensate for normalized coordinates, but the considerations of scale changes between images, which will be discussed below in section 6.7.3, may not apply here, since distances are compared in the same image. The threshold 3 is arbitrary but chosen as the inlier distance in section 6.7.3.

Having this proximity check also enables using multiple feature detectors, as done in section 6.10, without worrying about duplicate feature points being detected.

6.7.2 Determining the Number N of Iterations Adaptively

The formula used for adaptively determining N is the following:

$$N = \log(1-p)/\log(1-(1-\epsilon)^s) \quad (7)$$

The variable p models the probability of having found at least one sample set, which is free from outliers. p is typically set to 99 percent, i.e. 0.99. The variable ϵ models the probability that a point

correspondence is an outlier, so $1-\epsilon$ models the probability that it is an inlier. The size s of the sample set is 7 for the 7-point algorithm. More details on the statistical derivation can be found in section 4.7.1 in [Hart03].

As an example, for $s = 7$, $p = 0.99$ and $\epsilon = 0.5$, i.e. 50 percent of the correspondences being outliers, N becomes 588 iterations. If we had been using the 8-point algorithm, s would have been 8 and N would be 1177. Such dramatic increases of N is a good reason why the 8-point algorithm should not be used here, in addition to the fact that the 8-point algorithm gives a worse estimation of the fundamental matrix than the 7-point algorithm.

In order to avoid integer overflows on N, its initial value is not actually set to infinity, but set to 10^6, which is large enough. Also, the adaptive value of N is only updated in the algorithm, if the best number of inliers is at least 7. The reasons are that, fewer than seven inliers cannot be a correct estimate and low numbers of inliers may cause N to overflow, with the formula in equation 7.

6.7.3 Determining Inliers

In the algorithm in section 6.7.1, the geometric reprojection error measure from equation 3 in section 6.5.1 is used for determining when point correspondences are classified as inliers, except that the distance is not summed over all correspondences, but only computed for a single pair of correspondence points and computed as two individual distances:

$$d1_k^2 = d(\mathbf{x}_k, \hat{\mathbf{x}}_k)^2 \qquad d2_k^2 = d(\mathbf{x}'_k, \hat{\mathbf{x}}'_k)^2 \tag{8}$$

The terms in these formulas are illustrated and explained in figure 3. We need to specify two distance thresholds t_1 and t_2 for this. A single threshold t of 1.25 pixels is suggested in section 11.6 in [Hart03]. The reason for having two thresholds instead of one is to be able to check distances in each image individually; an example of distance discrepancies is given further below. Regarding the value 1.25, if large scale changes exist between two images, e.g. a scale factor of six, which is the best realistically achieved in the article [Dufo02], then lower accuracy might be obtained for the correspondence points. A threshold of 3 pixels will be used, as a guess of supporting six times half a pixel of inaccuracy, but this value has not been experimentally justified. However, larger inaccuracies than 1.5 pixels were indeed detected in the previous report [Anoq09], for the test images that emphasized on scale changes. Notice though that in the formula above, the distance is squared, so we would need to use the square root of this, when measuring in pixels. Also, since we are using normalized coordinates, we have to adjust the distance according to the scale change induced by the normalization from section 6.1. In particular, the distance thresholds t_1 and t_2 have to correspond to 3 pixels in each image, where the normalizing factors $\frac{1}{s}$ and $\frac{1}{s'}$ are different in each image. This means that $t_1 = 3\frac{1}{s}$ and $t_2 = 3\frac{1}{s'}$, for the threshold test $d1_k < t_1$ and $d2_k < t_2$ (or equivalently, but faster to test: $d1_k^2 < t_1^2$ and $d2_k^2 < t_2^2$).

If we had only used one distance threshold for the sum of the two image distance terms, this would allow e.g. more than 3 pixels distance in one image and less in the other image. In case of scale changes, discrepancies may then also arise in the distances, e.g. $3 \cdot 6 = 18$ pixels, in case of six times scale change, which is probably the most convincing reason to test distances in each image individually.

When a new set of inliers has been found in the algorithm in section 6.7.1, where there are as many inliers as in the previously best set of inliers, the best set of inliers is chosen as the one with the lowest standard deviation of inliers. The statistics is computed on the distance error measure. This requires the formula for computing the mean value $\bar{d}$ of K distances d_k (where each distance d_k is the sum of the pair of distances $d_k = d1_k + d2_k$):

$$\bar{d} \;=\; \frac{\Sigma_k^K d_k}{K} \tag{9}$$

The formula for the statistical variance σ^2 of K distances d_k then becomes:

$$\sigma^2 = \frac{\Sigma_k^K (d_k - \bar{d})^2}{K-1} \tag{10}$$

The denominator $K-1$ is correct here, since statistical variance computation is done by dividing by the number of degrees of freedom, not the number K of samples (see e.g. pages 12-18 in [Emde08] to understand this). It is thus not possible to compute the variance of one sample. The standard deviation σ is found as the square root of the variance σ^2.

6.8 Fundamental Matrix Parameterization

When optimizing the estimate of a fundamental matrix for an image pair, we need some way of parameterizing that matrix, in order to perform the optimization. One way is to describe it by the projection matrix $P' = [M|\mathbf{t}]$ of the second image, where the canonical projection matrices with the projection centres at infinity are used, as in section 6.6.1. As seen in section 5.3, the fundamental matrix can be computed from this as $F = [\mathbf{t}]_\times M$. This ensures that F is singular, since $[\mathbf{t}]_\times$ is.

The parameterization described here is an over-parameterization, since it has 12 degrees of freedom, whereas the fundamental matrix has only 7. Normally this should not give any problems, so this parameterization will be used in the implementation. However, better parameterizations are possible, as described in section 11.4.2 in [Hart03], but they require some bookkeeping by the implementation, in order to avoid degenerate cases of the parameterizations. Such parameterizations are suggested for future work.

6.9 The Gold Standard Optimization

The Gold Standard method for optimization is used for improving the estimate of a fundamental matrix. The optimization finds the Maximum Likelihood estimate of the fundamental matrix by non-linear optimization by Levenberg-Marquardt iteration. The method starts from a set of $n \geq 8$ point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$, which are assumed to be (mostly) inliers, and an initial estimate F of the fundamental matrix. The optimization is set up as follows (from section 11.4.1 in [Hart03]):

- Let the initial estimate be $\hat{F} = F$, e.g. estimated by RANSAC (section 6.7)
- From $\hat{F}$ compute an initial estimate of the camera matrices $\hat{P}$ and $\hat{P}'$ (section 6.6)
- From $\hat{P}$ and $\hat{P}'$ compute an initial estimate of projectively reconstructed 3D points $\hat{\mathbf{X}}_i$ from the correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ by triangulation (section 6.6)
- Project the 3D points $\hat{\mathbf{X}}_i$ by the matrices $\hat{P}$ and $\hat{P}'$ into an initial set of subsidiary point correspondences $\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i$ by $\hat{\mathbf{x}}_i = \hat{P}\hat{\mathbf{X}}_i$ and $\hat{\mathbf{x}}'_i = \hat{P}'\hat{\mathbf{X}}'_i$
- Minimize the cost $\sum_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2$ from section 6.5.1 (and figure 3) over $\hat{F}$ and all $\hat{\mathbf{X}}_i$. This minimization is done by Levenberg-Marquardt iteration and will be more precisely specified below. The projected points $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$ move in both images when the reconstructed points $\hat{\mathbf{X}}_i$ move during the optimization, even though $\hat{P}$ for the first image keeps being the identity matrix
- This minimization results in a final estimated projection matrix $\hat{P}' = [M|\mathbf{t}]$ for the second image and a final set of estimated homogenious 3D points $\hat{\mathbf{X}}_i$
- The estimated fundamental matrix is now $\hat{F} = [\mathbf{t}]_\times M$ as in section 6.8

6.9.1 Levenberg-Marquardt Iteration

The numerical Levenberg-Marquardt method was implemented from appendix 6 (particularly sections 6.1 and 6.2) in [Hart03]. The implementation will not be described, but a few remarks about it are appropriate and not necessarily obvious from the descriptions in [Hart03] (but the corresponding descriptions for homography estimation in section 4.5 in [Hart03] are somewhat more detailed):

- The method requires as input an M-dimensional goal vector $\mathbf{G}$ to optimize towards and an N-dimensional parameter vector $\mathbf{P}_0$ with an initial estimate. It also takes a function $f : \mathbb{R}^N -> \mathbb{R}^M$, which when given a parameter vector $\mathbf{P}_i$ returns a result vector $\mathbf{R}_i$. The purpose is to approximate the goal vector $\mathbf{G}$ by iteratively minimizing $\|\mathbf{G} - f(\mathbf{P}_i)\|^2$ (i.e. $\|\mathbf{G} - \mathbf{R}_i\|^2$) over $\mathbf{P}_i$. In these expressions, $\|\cdot\|^2$ is the norm of squared differences

- The initial vector $\mathbf{P}_0$ is defined by having as the first 12 elements the entries of the fundamental matrix parameterization from section 6.8 (i.e. the entries of the 3x4 projection matrix P' from the previous section) and as the last $4n$ entires, the n projectively reconstructed homogenious 3D points $\hat{\mathbf{X}}_i$ from the previous section, each one having four entires

- The goal vector $\mathbf{G}$ is defined by $4n$ entries consisting of the x and y coordinates in each image of the correspondence points $\mathbf{x}_i$ and $\mathbf{x}'_i$

- The function f is defined on its input $\mathbf{P}_i$ to extract the projection matrix $\hat{P}'$ for the second image and the estimated 3D homogenious points $\hat{\mathbf{X}}_i$. From this, it computes the projected 2D points $\hat{\mathbf{x}}_i = \hat{P}\hat{\mathbf{X}}_i$ and $\hat{\mathbf{x}}'_i = \hat{P}'\hat{\mathbf{X}}_i$ and returns them as the $4n = M$ dimensional vector $\mathbf{R}_i$

- It can now be noticed that $\|\mathbf{G} - f(\mathbf{P}_i)\|^2$ is the same as $\sum_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2$ from section 6.5.1. The minimized term is actually computed for the optimization as the four individual squared distance terms of the coordinates, two coordinate distance terms in each image for each correspondence, but the computed value is equivalent to the formula as specified here

- The algorithm thus intrinsically minimizes the geometric reprojection error, when set up as described here, so no squared distances should be explicitly computed, neither by the Levenberg-Marquardt iteration, nor by the function f. However, the exception to this is that the Levenberg-Marquardt iteration internally uses the norm $\|\mathbf{G} - f(\mathbf{P}_i)\|^2$ as a termination criterion, so for testing that criterion, the iterative algorithm will explicitly (internally to the optimization algorithm though) compute the sum of squared differences

- The termination criterion is that when improvements in the above norm is less than a small number, then the minimization stops and the final $\mathbf{P}_i$ is chosen as the estimated vector $\hat{\mathbf{P}}$, from which the estimated projection matrix $\hat{P}'$ and 3D points $\hat{\mathbf{X}}_i$ can be extracted

- The termination criterion is not given in section A6.2 on page 600 in [Hart03], but the one which is used is to compute $\epsilon_{i-1}^2 = \frac{\|\mathbf{G} - f(\mathbf{P}_{i-1})\|^2}{M}$ before minimization iteration i and $\epsilon_i^2 = \frac{\|\mathbf{G} - f(\mathbf{P}_i)\|^2}{M}$ after the iteration. The optimization is terminated when there is an improvement, i.e. $\epsilon_i < \epsilon_{i-1}$ (this is part of the way a Levenberg-Marquardt iteration is defined), and when that achived improvement is less than 10^{-2}, i.e. when $0 < \epsilon_{i-1} - \epsilon_i < 10^{-2}$. Since the ϵ_i terms are squared, computing $0 < \epsilon_{i-1}^2 - \epsilon_i^2 < (10^{-2})^2$ is slightly faster and used instead. The reason for dividing by M in the ϵ_i terms is to make the comparison threshold 10^{-2} independent of the dimension M, i.e. the number of correspondences used in the optimization. The value 10^{-2} is arbitrary, but 10^{-3} does not seem to result in any better results. 10^{-1} seems to work well too, but is possibly less stable

The current implementation does not use the sparse Levenberg-Marquardt methods, which means that it currently becomes slow already for around 50 or 100 correspondences. Implementing sparse Levenberg-Marquardt is therefore recommended for future work.

6.10 Feature Detector and Matching

This section is mostly relevant for readers who are somewhat familiar with the previous report [Anoq09], so it may be skipped; it only concerns how the matched point correspondences are found, including a few improvements since the previous report.

The previous report describes feature[2] detection, feature descriptors and matching strategies and has some suggestions for the design of the overall pipeline. The feature detector was the Maximally Stable Extremal Region (MSER) detector, as described in [Mata02], but affine invariance was not implemented. The feature descriptor was Speeded-Up Robust Features (SURF), as described in [BayH06]. The matching strategy was nearest neighbour search, where matches are only kept if the nearest-to-second-nearest neighbour ratio is 0.7 or less.

Since the completion of the previous report, the author discovered that using the matching strategy, which was referred to as brute-force *cross-correlated* two-way matching, is far superior to both the one-way matching and the conservative two-way matching, which were both used and also described in that report. Unlike conservative two-way matching, cross-correlated two-way matching is even compatible with the experimenal measurement formulas in that report, so it was easy to re-run the performance measuring program with the improved matching strategy. In summary, using cross-correlated two-way matching results in much fewer correspondences being found, but the false positive rate, measured by 1-precision, is quite low, often less than 20 percent, several times being 0 percent. Hence, many invalid matches are filtered out. The fact that affine invariance was not implemented is still very evident, since the number of matches drops rapidly for larger camera view changes.

As a note on the low number of matches, the article [Chum05] (in section 3) contains results on using the Maximally Stable Extremal Region (MSER) detector from [Mata02] and the Scale-Invariant Feature Transform (SIFT) descriptor from [Lowe04] with the nearest-to-second-nearest neighbour matching criterion, which is comparable to the methods from the previous report. They report numbers of detected inliers as low as 12 for their PLANT scene, which is considered a difficult scene. They state that the nearest-to-second-nearest neighbour matching criterion is the main reason for this, since it filters out many correspondences, retaining mostly just inliers. This concurs with the experiences of the author during the development of this report.

The $A\kappa\rho\acute{o}\pi o\lambda\eta$ image pair seen in this report (figure 5) and the previous report should also be considered a difficult scene, due to the amounts of trees, many small bricks on the large wall and repeated content on the white temple. On this image pair, only seven correspondences were found, but they were all correct or close to being correct (e.g. "off by a few stones" on the white temple, as we shall also see in this report in figure 6), according to visual human inspection by the author. This was with all parameters from the previous report unchanged, except for using two-way cross correlated matching. Despite this seeming like a miracle for that particular image pair, since seven is the minimum required number of correct point correspondences, we might not be that lucky for other image pairs, so finding more correspondences is clearly desirable. It should also be remarked that when changing the floating point representation from 32-bit, as was used in the previous report, into 64-bit, the results

[2]It has come to the attention of the author that terms like 'interest point' or 'region of interest' may be a better term here, since in some research fields, e.g. medical image analysis, the term feature is used to denote the number of relevant degrees of freedom when considering correlations between variables when analysing an image. For consistency with the previous report, we shall continue using the term feature; or better, in the case of points: feature point

degraded for this image pair, but in some other cases, it improved; more on this later in section 6.12.

Combining several feature detectors was suggested in the previous report, but even just varying the parameter Δ, as used for the Maximally Stable Extremal Region (MSER) detector, can actually also give different sets of correponding points. Running three independent rounds of feature detection and matching, with Δ values of repectively 20, 10 and 5, seems to result in three somewhat different sets of correspondences. This will therefore be done and the resulting three lists of correspondences will be concatenated, yielding a larger set of correspondences. Duplicate point correspondences will *not* be removed from the list, since if there are similar correspondences in the three lists, they may likely be the correct ones. In the RANSAC estimation of the fundamental matrix, described in section 6.7.1, the seven point correspondences will be picked such that they are not too close to each other in both images, so having duplicate correspondences is not harmful. The large values of Δ, such as 20, should be efficient for coping with image blur, while smaller values normally find many more potential regions, so it may indeed be a good idea to run the detector in this way with different settings.

Performance-wise, only the last phase of the feature detector has to be re-run three times, but this optimization has not been made, so the entire feature detection is executed three times. The descriptor calculation and matching also has to be done three times and this is currently the most expensive part of finding correspondences.

For a real application, the implementation of the feature detector should be improved, particularly by implementing affine invariance. Also, adding another feature detector is relevant, as we shall see in the next section. However, the settings described above will be used, in order to change as few things as possible from what was described in the previous report.

6.11 The Full Pipeline

The full pipeline, which is implemented for this report, consists of the phases:

- Point correspondences are computed as outlined in section 6.10
- Robust estimation of the fundamental matrix is done by RANSAC, as described in section 6.7
- The following two steps can be iterated, but the iteration and the guided matching was disabled, due to actually degrading the results:
  - Non-linear estimation of a new fundamental matrix from the correspondences currently classified as inliers. This is described in section 6.9
  - Guided matching using the estimated fundamental matrix, in order to try to find more inlier correspondences. A few details on the guided matching follow in section 6.11.1 below

It is important that the same error measure is used for all minimizations done in the algorithm. In particular, the error measure used for RANSAC (section 6.7) and the error measure used for the non-linear optimization (section 6.9) should be the same, where we use the geometric reprojection error from section 6.5.1. If these error measures differ, the algorithms will not be consistent with each other. Also, as mentioned earlier, remember to keep track of when normalized, unnormalized or squared coordinates are used, when measuring the error distance.

We shall not use or go through more refined pipelines in this report, but a few other suggestions were outlined in the previous report and in general, many combinations are possible.

6.11.1 Guided Matching

For the guided matching, the detection and matching is done as outlined in section 6.10. However, matching of each feature point $\mathbf{x}_i$ in one image is only done with feature points within a search strip around the epipolar line $\mathbf{l}'_i$ in the other image, where $\mathbf{l}'_i$ corresponds to $\mathbf{x}_i$ in the first image. This point to epipolar line correspondence is defined by the formula $\mathbf{l}'_i = \hat{F}\mathbf{x}_i$ from section 5.2, where $\hat{F}$ is the estimated guiding fundamental matrix. For the two-way matching, when the roles of the two images are switched, the transposed fundamental matrix is used in the above formula (see section 5.2).

The search strip which was used around the epipolar line was simply to include only all points within a distance of 50 pixels (with the denormalized fundamental matrix, such that the distances are really in pixels) of the epipolar line. This is quite crude. [Hart03] section 11.11 suggests searching the covariance envelope around the epipolar line, obtained from the covariance matrix of the fundamental matrix. The article [Trig01] uses Joint Feature Distributions for obtaining an even closer correspondence search range in the second image, from a feature point in the first image, where the search ranges become elliptical Guassian distribution areas. None of these two improved search methods have been used, since we will consider worse issues to be dealt with first, but they are suggested for future work.

The nearest-to-second-nearest neighbour ratio, which is used as the matching criterion, as mentioned in section 6.10 and described in the previous report [Anoq09], is effective for filtering out potential matching features, which are very alike. When features from only a smaller area of the other image are used for matching each feature point in an image, more feature points should hopefully be allowed to pass the matching criterion, resulting in a larger amount of point correspondences. The amount of correspondences did increase by this guided search, but not very much and the amount of inliers increased only slightly. Worse, the few extra inliers did not seem to particularly improve the estimate and sometimes to even make it worse. These are the main reasons why the guided search was disabled, despite it having been implemented and otherwise seeming to work.

The best suggestion to improve on this situation is to use different feature detectors and maybe a different matching criterion in this phase, compared to what is used in the initial phase. The current detector and matching criterion seem good and robust for the initial estimation, but less suited for the guided phase. The idea of using different detectors was also suggested in the previous report, but it will not be done here.

6.12 Numerical Precision and Stability

In the previous report and most of the initial development phase for this report, 32-bit floating point precision was used. Towards the end of the development phase during this report, this was changed to 64-bit floating point precision, which gave some surprises. It is easy to change the default floating point type (the type `real` in Standard ML) by a compiler option for the used compiler, MLton [MLto07].

Changing between 32-bit and 64-bit floating point precision changes the results quite a lot, even which matched point correspondences are found by the methods from the previous report. Although surprising, it may actually be reasonable. E.g. the Speeded-Up Robust Features (SURF) descriptor from [BayH06] uses and sums many pixel samples, which are weighted by a Gaussian distribution, computed by floating point numbers in the implementation used. Even a slight change in this computed orientation, either by the pixel sums mentioned above or by the numerical precision of the mathematical $\sin$ and $\cos$ functions, may affect which pixels are sampled in the oriented descriptor phase, since that phase uses floating point coordinates for the rotated samples (source code was supplied for this orientation in appendix C in the previous report [Anoq09]). Additionally, the centre point of each detected Maximally Stable Extended Region (MSER) ([Mata02]) is computed as floating point numbers

and the oriented descriptor calculation also uses this centre. Slight changes in the descriptors may affect which correspondences are matched, since there are several thousand detected feature points in each image, from which only a few final correspondences are used. It seems particularly reasonable that the output is affected in the presence of repeated image content, since this would make relevant descriptors quite similar.

The RANSAC phase in this report (section 6.7.1), although always using the same initial random seed, will be very affected if even a single correspondence is added or removed from the list of point correspondences, since the samples are selected by indices into the correspondence list. Additionally, the random number generator itself uses floating point numbers, although this has been explicitly using 64-bit floating point precision from the beginning (the Standard ML module `LargeReal`), but there are still conversions to 32-bit precision, when using 32-bit floating point numbers.

In addition to the results changing when switching between 32-bit and 64-bit floating point precision, the results sometimes also varied by even slight program modifications, which is likely due to the fact that the compiler MLton [MLto07] is a whole-program optimizing compiler, where even small program changes can make the generated code for the *entire* program change dramatically. This is perhaps an even more disturbing observation than the switch between 32-bit and 64-bit precision, but considering e.g. the x86 processor's 80-bit intermediate results on floating point computations and the whole-program compilation changes, it is perhaps still to be expected. It should be mentioned that a mistake was fixed at the end of the project, which was that the non-linear optimization (section 6.9) only optimized on the reprojection distances in the second image. Fixing this problem not only gave noticable improvements, but also seems to have made the program more numerically stable towards program changes, even though this has not been properly verified.

With the above considerations, it should make sense why the RANSAC estimated fundamental matrix may be severely affected by the floating point precision. The RANSAC estimate is the initial guess for the non-linear optimization (section 6.9), so this may have an effect on the result. This is the effect which might have diminished by correcting the problem (mentioned above) in the non-linear optimization. Also, the matched inlier correspondences most certainly has an effect on all parts of all algorithms presented in this report. Since virtually all methods in this report use floating point precision, it should be no surprise that there would at least be some differences in the precision of the results due to this.

The implemented methods generally seem to be somewhat sensitive numerically, but hopefully the reader is now convinced that this is not unreasonable for the implementation presented. The general results however, as presented in section 7, do at least appear to quite robustly give sensible fundamental matrix estimates, which is a very important achievement.

6.13 Suggested Improvements to the Implemented Methods

Implementing the optimial triangulation method, as mentioned in section 6.6.2, seems like an obvious place for making an improvement. This method might improve the quality and the convergence rate of the fundamental matrix estimate, since it should improve the positions of the initially reconstructed 3D points used in the Gold Standard optimization, described in 6.9.

Using a better parameterization of the fundamental matrix with only 7 degrees of freedom may also improve the non-linear optimization, as suggested in section 6.8.

It is relevant to consider handling the degenerate cases of planar scene geometry and coincident camera centres, in which case a homography, or even a rotational model, rather than a fundamental matrix, could be estimated, as mentioned in section 6.4.2. The article [Trig01] uses Joint Feature Distributions, which is a statistical framework. It not only allows tight search regions for correspondences between images, as was mentioned in section 6.11.1, but it also works well in the cases of planar or

near-planar scenes. Another approach to handling planar scenes, as well as coincident camera centres, is found in [Kana98], where a geometric information criterion is derived. When one model is compatible but stronger (i.e. more restrictive) than another model, such as homography estimation in comparison to fundamental matrix estimation, the criterion can give a definitive assertion of whether the stronger model is preferable. The article proposes to first test whether a rotational model should be chosen (for coincident or near-coincident camera centres), and if not, test if the homography model should be chosen, and if not, use the fundamental matrix model.

Implementing guided matching which works well is important. This would no doubt involve using different feature detectors and possibly another matching criterion for the guided matching phase, as suggested in section 6.11.1. The search method of using the epipolar covariance envelope from section 11.11 in [Hart03] or the Joint Feature Distributions from [Trig01] was also suggested in section 6.11.1.

The current implementation only supports two views. Supporting more than two views is desirable. Using pair-wise combinations of images and estimating the fundamental matrix is actually not enough to get stable results, according to chapter 18 in [Hart03]. Therefore, estimating the trifocal tensor is relevant to consider, as are other N-view methods. As mentioned in section 3, bundle adjustment for multiple views should generally be the final stage, where useful information may be found in [Trig99].

A computationally more efficient alternative to using RANSAC (section 6.7) is to use PROgressive SAmpling Consensus (PROSAC), as described in [Chum05]. This could speed up the convergence of the robust fundamental matrix estimation considerably; they report typically more than 100 times speed-up. In the worst case, it should not be worse than RANSAC. Additionally, PROSAC's use of a quality function for ordering the preference for selection of matched point correspondences can be used to remove one of the matching parameters, such as the the nearest-to-second-nearest neighbour ratio mentioned in section 6.10. This potentially allows more correspondences to be found, without decreasing the search efficiency. It can be noted that a simple experiment was made for the $A\kappa\rho\acute{o}\pi o\lambda\eta$ image pair (presented in figure 5): the nearest-to-second-nearest ratio criterion was removed, such that all nearest neigbour matches were used. This resulted in 1262 tentative correspondences being found, rather than just 24 (at that time, using 32-bit reals, out of which 18 were considered by RANSAC to be inliers). However, when this was done, RANSAC did not terminate after running for about 24 hours, which suggests that there were probably not significantly more than the 18 inliers among the 1262 correspondences. PROSAC would start its search preferring the 24 correspondencs, but still have the possibility of finding any remaining inliers amoung the 1262 correspondences. This property could also be easily achived with RANSAC though, by partitioning the tentative correspondences into two partitions, where only the 24 high-confidence correspondences are being sampled from, while all 1262 correspondences are used for inlier verification. However, this does not remove the fixed nearest-to-second-nearest ratio threshold (currently 0.7), as the PROSAC method can do, which is another reason for recommending PROSAC.

Most of the Singular Value Decomposition (SVD) computions do not use the computed matrix U in the decomposition SVD $A = UDV^T$ and according to appendix 4 in [Hart03], much improvement in speed can be gained by not computing this.

Implementing a variant of the Levenberg-Marquardt optimization which exploits the sparsity of the optimization problem, as described in appendix 6 in [Hart03], should give a faster computation of the non-linear optimization, which is currently very slow already at 100 point correspondences or more. This optimization will particularly be needed if bundle adjustment is to be done for multiple views, since the lack of speed would otherwise make the implementation useless. Also, it was attempted to match the $A\kappa\rho\acute{o}\pi o\lambda\eta$ image 6 (figure 5 top) with itself, which resulted in 4,377 correspondences, which made the program run out of memory (on a computer having 3GB available), probably since the Levenberg-Marquardt optimization involved working on (non-sparse 64-bit floating point) matrices

of size 17508x17520. For these reasons, the Levenberg-Marquardt optimization clearly needs to be improved.

7 Evaluation of the Implemented Methods

The implemented algorithms need to be evaluated. This section starts by describing some strategies and methods for doing that and then performs some of the described evaluations.

7.1 Experimenal Strategies

In order to test the correctness of the computed fundamental matrix, experiments like the following could be made:

- Use an existing data set, where camera matrices have already been estimated by other people on a set of images. The fundamental matrix can computed from these estimated camera matrices and can be used for comparison, but it may not be the accurate ground truth
- Estimate a fundamental matrix from some manually selected matching feature points in the input images, possibly using a simple estimation algorithm, such as the normalized 7-point algorithm. This estimated fundamental matrix will be subject to inaccuracies of the manually selected feature points and to the limitations of the algorithm used for estimating it
- Visual inspection of epipolar line correspondences and reconstructed 3D points. This does not give any measure for the correctness, but it should tell whether the results look sensible or wrong
- Use some kind of calibrated setup, where the fundamental matrix has been computed from real-life measurements of the camera setup. This is subject to the inaccuracies and possible mistakes of the setup
- Use synthetic rendered 3D images, where the camera settings are known. This can give an analytically accurate fundamental matrix, but the experiment would still be subject to possible discrepancies between the rendering system and the tested system. Also, synthesized images may not necessarily give a faithful impression of how the algorithms would work on real images
- Use synthetically constructed 3D points and camera matrices, which are used to project into two sets of 2D points, mimicing image points. This gives the ground truth (up to machine floating point precision) 3D points and the ground truth camera matrices, from which the ground truth fundamental matrix can be computed. This does however not test the feature detection and matching algorithms, but it tests the primary methods presented in this report

It can be noted that, none of these approaches, except for the last one, are the ground truth for comparison. Of the above possiblities, the fourth option of calibrated cameras does not seem attractive, due to the work involved in the making the setup, so it will not be used. The option of rendering images is interesting, but it requires some work to set up and does not represent real images, so it will also not be used. The first three options are possible to do without too much effort. The second option of using manually selected points was used in the early stages of the project and for generating some of the data used for unit-testing the program code, but due to its estimation inaccuracies, it will not be used for any of the presented evaluations. Figure 2 was created in this way though, by using the the normalized 8-point algorithm without enforced singularity constraint, meaning that the epipolar lines do not meet

in the same point. For the third option, inspecting reconstructed 3D points was attempted, but it did not give a very useful visualization, due to the relatively few points detected and due to the difficulties in setting up an appropriate camera projection matrix.

Making different kinds of experiments seems better than just focusing on one, so results of the following three kinds of experiments will be presented:

- Using existing data sets with estimated camera matrices (the first option above)
- Visually inspecting epipolar lines (part of the third option above)
- Using synthetically generated ground truth data (the last option above)

The last option does not test the entire pipeline, but on the other hand, it is the only experiment where the ground truth is available for comparison.

7.2 Degenerate Cases and Algorithm Robustness

When evaluating the implemented algorithms, their robustness and their handling of degenerate cases is relevant. Some ways of evaluating this could be:

- Using test images with degenerate camera motion configurations, e.g. parallel camera motion without rotation or the camera remaining at the same point
- Using test images with degenerate scene structure configurations, e.g. all points lying on a plane or nearly on a plane in the scene
- Adding random noise to the input point correspondences, before running the algorithm
- Adding some amount of random outlier correspondences, before running the algorithm

All of these cases will be handled merely by the properties of the selected image test sets and the fact that the implementation uses a real feature detector. The image test sets contain both degenerate camera motion and scene structure. The degenerate camera motion of the camera staying at the same point is tested by matching one of images with itself. Random noise and outliers are present due to the feature detector. However, explicitly perturbing the detected image points by noise may be a useful experiment, in order to quantify the effect of noise on the accuracy. Therefore, an experiment will be done where noise is added to the synthetically generated point correspondences, where the ground truth is available for comparison.

7.3 The Accuracy of a Fundamental Matrix Estimate

It is relevant to test the achieved accuracy of the fundamental matrix estimates, with respect to the detected point correspondences. This is done by using the residual error measure already described in section 6.5.3. However, some variations of the residual error is possible. In section 11.5, page 288, and in figure 11.3 on page 290 in [Hart03], they take the mean of this error over the number of correspondences. Their mean is over the number of correspondences, not the number of distance terms, even though there are two distance terms per correspondence. In chapter 5 section 5.1.3 (result 5.2 (i)) on page 136 of [Hart03], they use the Root Mean Squared (RMS) residual error.

The mean of the residual error from section 6.5.3 will be computed by dividing it by the number of squared distances terms, which is two times the number of correspondences, since there are two distance

terms for each correspondence, one distance term for each image. The square root is used because the distance terms are squared. The complete formula for the RMS residual error $\epsilon_{res_{RMS}}$ becomes:

$$\epsilon_{res_{RMS}} = \sqrt{\frac{\sum_i^N d(\mathbf{x}'_i, \hat{F}\mathbf{x}_i)^2 + d(\mathbf{x}_i, \hat{F}^T\mathbf{x}'_i)^2}{2N}} \tag{11}$$

where the estimated fundamental matrix is $\hat{F}$ and the N point correspondences are $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$. $d(\mathbf{x}, \mathbf{l})$ is the distance between the point $\mathbf{x}$ and the line $\mathbf{l}$, where the two lines in the above formula are the epipolar lines obtained by the multiplications $\hat{F}\mathbf{x}_i$ and $\hat{F}^T\mathbf{x}'_i$ (the formulas in section 5.2 explain these mappings), respectively. The distance terms were illustrated back in section 6.5.3 in figure 4. Only the correspondences considered to be inliers by RANSAC (section 6.7) will be used for this formula. This formula gives the average distance in pixels between a point and the epipolar line of its matched point, when the distances are computed on the unnormalized coordinates, as will be the case.

For the image sets where pre-estimated fundamental matrices are available, this formula will also be used for computing the RMS residual error of the pre-estimated fundamental matrix with respect to the detected correspondences.

Chapter 5 of [Hart03] suggests computing both the residual error and the estimation error, with the formulas given on page 136 in section 5.1.3 in [Hart03]. The reason why both are relevant is that the *residual* error gives the accuracy of the assumed model (i.e. in this case the fundamental matrix relationship $\mathbf{x}_i\hat{F}\mathbf{x}'_i = 0$ from section 5.2) with repect to the detected correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$. On the other hand, the *estimation* error measures how accurate the estimated correspondences are (no formulas given), provided that the model is correct. Thus, the residual error can be made arbitrarily small, by choosing a sufficiently loose model, and the estimation error can be made arbitrarily small, by choosing a sufficiently restrictive model. Good explanations of this can be found in section 2.2 of [Kana98]. To limit the extent of the evaluation in this report however, only the RMS residual error will be computed.

7.4 Comparing Fundamental Matrices

Fundamental matrices are $3x3$ matrices, but they cannot simply be compared by comparing their entries, since very different matrices can represent the same transformation. One way to compare two fundamental matrices F_1 and F_2 would be to take a set of points (e.g. placed in a regular grid or manually placed) in the first input image and map them through the two fundamental matrices being compared. The points map to epipolar lines in the other image (i.e. $\mathbf{l}'_{i_j} = F_j\mathbf{x}_i$, see section 5.2), which can then be compared, e.g. by comparing the distance between the end-points, where the lines $\mathbf{l}'_{i_1}$ and $\mathbf{l}'_{i_2}$ leave the image. The two epipolar lines could also be compared by comparing the angle between them.

All epipolar lines in the above comparison method pass through the epipole in the second image. Hence, by comparing the epipoles in the second image of the two fundamental matrices, a significant aspect of the matrices is compared. If the epipoles in both images (formulas in section 5.2) are compared, then the comparison gets even better. Recall from section 5.2 that the epipoles are the projection of the camera centre of one camera into the image taken by the other camera. Hopefully this consideration can convince the reader that comparing the two epipoles of the fundamental matrices can be a relevant way of testing the estimated relative placement between the two cameras. This does however only test four out of the seven degrees of freedom that the fundamental matrix has: the four degrees of freedom which are fixed by the placements of the x and y coordinates of the epipoles. See section 9.2.5 on page 246 in [Hart03] for more details on the remaining degrees of freedom.

Notice that if the camera placements are related by mostly a fronto-parallel translation, then the epipoles may be placed at or near infinity outside the image. This would likely result in large inaccuracies between the placements of estimated epipoles, when comparing two different fundamental matrix estimates. Therefore, in these kinds of situations, the epipole positions should probably be compared relatively to their distance from the image area, which complicates things a bit.

The first of these methods of comparing end-points of epipolar lines is used and is very useful for the automated unit-tests developed during the project. However, as argued above, there may be many potential problems with these comparisons and the meaning of the comparisons is not entirely clear, so these methods do not necessarily give a very robust comparison. Therefore, and since other research mostly use the RMS residual and estimation errors, only the RMS residual error described in the previous section will be used when comparing with pre-estimated fundamental matrices.

It should however be noted that a lower RMS residual error does not necessarily imply a better estimate, since the detected point correspondences are part of the formula and may be partly wrong, so the numbers should not be over interpreted.

7.5 Epipolar Lines for Visual Inspection of Estimated Fundamental Matrices

For visual inspection, a few epipolar correspondences of the estimated fundamental matrices are displayed, as seen in e.g. figure 5. The epipolar line homography is used for mapping an epipolar line $\mathbf{l}'_i$ in the second image back into the epipolar line $\mathbf{l}_i$ in the first image: $\mathbf{l}_i = \hat{F}^T[\mathbf{e}']_\times \mathbf{l}'_i$, where $\mathbf{e}'$ is the epipole in the second image and $\hat{F}$ is the estimated fundamental matrix. In order to get an epipolar line $\mathbf{l}'_i$ in the second image, a set of points $\mathbf{x}_i$ is manually chosen in the first image, such that their locations in the photographed 3D scene are easily recognizable in both images. The points are mapped to epipolar lines in the other image by $\mathbf{l}'_i = \hat{F}\mathbf{x}_i$ and these are in turn mapped back to the first image by the epipolar line homography, as just described. These formulas are also in section 5.2. The hope is that it will be easy to visually determine whether or not the epipolar lines correspond in the two images.

The manually chosen points will also be visualized as cross-hairs in the first image. They may also be useful for the visual inspection, for finding a relevant point on the epipolar lines for comparison. Figure 5 is an example.

This visual inspection only serves as a qualitative comparison, but it gives a good impression of whether the results are sensible or not. Also, visual comparison with the epipolar lines of *pre-estimated* fundamental matrices will be done, which may be a good alternative to the analytical fundamental matrix comparions considered in the previous section.

7.6 Test Image Sets

A set of test images has been chosen for this report. The following test image sets were photographed by the author:

- Figure 5: Images 6 and 7 of the Athens $A\kappa\rho\acute{o}\pi o\lambda\eta$ image sequence (August 2002)
- Figure 7: Images 1 and 3 of the Copenhagen Nordic Mythology Graffiti Wall image sequence (November 2009)
- Figure 8: Images 46 and 47 of the Copenhagen St. Alban Anglican Church image sequence (June 2009)
- Figure 9: Images 5 and 6 of the Paris Tour St. Jacques image sequence (January 2009)

The three image pairs, presented in figures 10 (the Model House images 0 and 1), 14 (the Oxford Corridor images 0 and 1) and 16 (the Dinosaur images 1 and 2), are from the website:

- `http://www.robots.ox.ac.uk/~vgg/data/data-mview.html`

A few other image sets were tried, where the feature matching failed to find enough correspondences, but these were on more extreme image pairs which would no doubt be considered difficult by any method. Image sets also generally fail if they have too large differences in camera viewing angles, most likely due to affine invarance still not having been implemented for the feature point descriptor. The failure was thus not due to the methods presented in this report.

The Model House and the Dinosaur image pairs have pre-estimated camera projection matrices:

$$P_{House_0} = \begin{bmatrix} -667.1324398703851557 & 15.186601706999681483 & -399.12216996267011382 & -64.171047371437467177 \\ 0.26127780106302650465 & -664.13069781367391897 & -289.01467806003762462 & -0.76296166656404640349 \\ -0.00013667887261007119113 & 0.034010281383445604975 & -1.0006416157197026706 & 0.016977709775627819466 \end{bmatrix} \quad (12)$$

$$P_{House_1} = \begin{bmatrix} -575.7095077136132204 & 53.647203026738807807 & -497.05861320342859244 & -696.35839502775650089 \\ 3.4724787872714770742 & -647.35477633899131433 & -286.99029746051945722 & -33.741684879380485995 \\ 0.17254517683786235738 & 0.012316353011474379109 & -0.97391959245717629745 & -0.0026678054518149288757 \end{bmatrix} \quad (13)$$

$$P_{Dino_1} = \begin{bmatrix} 3.99235687564161 & 39.4176809830138 & -0.763289879714919 & 3.95917550891323 \\ -14.4302310113271 & -0.941441580237717 & -27.4509701085667 & -14.4294334377681 \\ 0.0122492403549385 & -0.00014574603756147 6 & -0.000569307087309742 & 0.0122493586975179 \end{bmatrix} \quad (14)$$

$$P_{Dino_2} = \begin{bmatrix} 10.7732491138691 & 38.1264946071842 & -0.763289879714919 & 3.95917550891323 \\ -14.3746180623658 & 1.57741397558573 & -27.4509701085667 & -14.4294334377681 \\ 0.0120380326410739 & -0.00226955971786568 & -0.000569307087309742 & 0.0122493586975179 \end{bmatrix} \quad (15)$$

The fundamental matrices between the image pairs can be found by the formula $F = [\mathbf{e}']_\times P' P^+$ from section 5.3. The pseudo inverse P^+ is found by first computing the Singular Value Decomposition (SVD) of P: SVD $P = UDV^T$. Then $P^+ = VD^+U^T$, where D^+ is the diagonal matrix with entries

$$d_{ii}^+ = \begin{cases} 0 & if\ d_{ii} = 0 \\ \frac{1}{d_{ii}} & if\ d_{ii} \neq 0 \end{cases} \quad (16)$$

where d_{ii} are the entries of D (appendix A5.2 p. 590 in [Hart03]). Since P has fewer rows than columns, P must first be extended with a row of zeros. Although it does not seem to be mentioned in [Hart03], the resulting matrix P^+ from the above computation will thus have an extra column of zeros, which should be removed. The rest of the formula $F = [\mathbf{e}']_\times P' P^+$ involves computing the camera centre from P (by using SVD), in order to find $\mathbf{e}'$. The formulas are in section 5.3.

The resulting fundamental matrices become:

$$F_{House_{0\to1}} = \begin{bmatrix} 0.00842337618233 & 0.0352341565115 & -35.2240134541 \\ -0.197771227607 & 0.0212943017827 & 677.325329282 \\ 31.7672833959 & -625.885556948 & 1013.55795724 \end{bmatrix} \quad (17)$$

$$F_{Dino_{1\to2}} = \begin{bmatrix} -7.41153115742 \cdot 10^{-6} & -1.47929568885 \cdot 10^{-4} & -0.0352496728166 \\ -1.14686144642 \cdot 10^{-4} & 5.41280808427 \cdot 10^{-6} & 4.88688852488 \\ -0.269313932911 & -4.78925490677 & 106.72471217 \end{bmatrix} \quad (18)$$

7.7 Synthetic 3D Data Set Projected as Point Correspondences

Figure 18 shows two images of a synthetically generated 3D data set. The images have been formed by projecting the 3D points with two projection matrices P and P' of the same form as those used for the 3D projective reconstruction (see sections 6.6 and 6.8), i.e.:

$$P = [I|\mathbf{0}] \quad (19)$$

$$P' = [M|\mathbf{t}] \quad (20)$$

The data set resembles a collection of the Sidewinder space ships from the classical computer game Elite [ElitWi], from the days of the Sinclair ZX Spectrum. There are 100 projected corresponding points, shown as large dots connected with lines, and 20 uncorrelated outlier points, shown as small unconnected dots. Figure 19 is another version of the data set, with only the first 20 projected points and the 20 outlier points.

For the construction of this data set, the 3D points had their z-coordinate divided by 1000 in world-space, since the homogenious identity camera projection matrix P, used for the first view, is equivalent to a very wide field-of-view of the camera, which would make the geometry look very unnatural. Computing the second projection matrix was done by first scaling the z-coordinate back up with a factor 1000, then rotating the world-space and finally dividing the z-coordinate by 1000 again, thus simulating that the z-coordinate division by 1000 happens in the camera coordinate system.

The well-known $3x3$ matrices (they are not even introduced in [Hart03], but implicitly used in e.g. section 2.4 in [Hart03]) for scaling, $S(x, y, z)$, and rotation around the y and z-axes, respectively $R_y(\theta_y)$ and $R_z(\theta_z)$, are:

$$S(x,y,z) = \begin{bmatrix} \mathsf{x} & 0 & 0 \\ 0 & \mathsf{y} & 0 \\ 0 & 0 & \mathsf{z} \end{bmatrix} \tag{21}$$

$$R_y(\theta_y) = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix} \tag{22}$$

$$R_z(\theta_z) = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{23}$$

Using these matrices, the second camera matrix $P' = [M|\mathbf{t}]$ is precisely given by $t = (-20, 0, 0)^T$ and

$$M = S(1,1,1/1000)R_z(-0.1)R_y(0.2)S(1,1,1000) \tag{24}$$

where it should be noted that the rotation parameters -0.1 and 0.2 are in radians, not degrees.

7.8 Visual Inspection

7.8.1 Presentation of the Inspection Images

The images for visual inspection by epipolar line correspondences and four manually selected points, as described in section 7.5, are shown in figures 5, 7, 8, 9, 10, 14 and 16.

Figure 12 shows the epipolar lines of matching the Model House image 0 with itself, which is an important case of degenerate camera motion: when the camera does not move. This degeneracy can make some methods break down, but as the figure shows, the achieved results look sensible.

The figures 11 and 17 similarly show epipolar line correspondences, only this time for the pre-estimated fundamental matrices, as derived in section 7.6, for the Model House and the Dinosaur image pairs.

Some of the image pairs contain point correspondences, which are wrongly taken to be inliers. This can be seen in figures 6, 13 and 15. These few wrong inliers seem to affect the results negatively.

The synthetic Elite Sidewinders scene with 100 correspondences and 20 outliers is shown without any injected noise into the correspondences in figure 20. The four manually selected points are true correspondence points and their computed epipolar lines are shown for inspection. In the version of the synthetic Elite Sidewinders scene shown in figure 21, one pixel of noise has been injected into all

correspondence points for the fundamental matrix estimation, but the four points used for visualizing the epipolar lines are still the true correspondence points. Injected noise is done by randomly adding an offset between -1 and 1 pixel (in figure 21) to each of the x and y-coordinates, so the noise distribution is not Gaussian, but a "flat" or "box" distribution, i.e. white noise within a bounded square area. No images are shown for the eight other versions of the synthetic scenes.

7.8.2 Inspection Conclusions

A few inspection comments were already given above, but more detailed comments for each image is to be found on the individual image figure descriptions.

First off, it should be noted that the epipolar lines have actually varied significantly by particularly switching between 32-bit and 64-bit floating point precision but also by slight program modifications (although this was mostly before correcting the mistake in the non-linear optimization), as argued in section 6.12. Such observed variances point towards the results not being reliable, when only few point correspondences are used.

Despite the above variances and numerical sensitivity, the methods do seem to quite robustly give sensible fundamental matrix estimates, which is a very important achievement. This can be confirmed for the images where either pre-estimated camera matrices or the ground truth is available for comparison, where there are differences in the estimated epipolar lines, but not significant differences. This can be seen by comparing the figures 10 with 11, 16 with 17 and 21 with 20.

In several of the images, the epipolar lines seem to follow parts of the structure in the scene, which is somewhat suspicious. For the Model House image pair, this also disagrees with the pre-estimated fundamental matrices, as seen by comparing the orientation of the epipolar lines in figures 10 and 11. This may be the explained by the correspondences wrongly taken to be inliers, typically due to repeated image content, as seen in several of the presented images, e.g. figures 6, 13 and 15. A conclusion drawn from this is that, the current methods are not yet good enough to handle repeated image content without mistakes, but this is also a hard problem in general.

The planar scene in figure 7 gives a degenerate configuration of the epipolar lines, since the epipoles should not be located in the images. This is to be expected, when the implemented methods do not handle the planar scene structure degeneracy. Still, the methods do not break down completely. Similarly, the degenerate case of no camera motion in figure 12 does not make the methods break down. However, the author believes that the fact that figure 7 has degenerate epipolar lines and figure 12 not, could be somewhat a coincidence; it could possibly have been vice versa, or both cases having degenerate lines. At least there is supposedly a well-known kind of ambiguity, which is mentioned in [Torr96] (in section 11.4 and figure 18) and explained in [Torr95]. Therefore, looking into these degeneracies will be suggested for future work. The degenerate forward motion in figure 14 seems to be handled fairly well. The conclusion is that, while improvements should be made for handling some of these degeneracies, no disasters or unreasonable results come from the methods used.

Among the phenomenas that the methods seem to, and intuitively should, handle well are such things as occlusion and parallax effects, as seen in figure 8, but this has not been investigated very thoroughly. Also, correspondences wrongly taken to be inliers are at least being somewhat well coped with. This goes for both randomly scattered correspondences, due to differing and truly unmatchable image contents, such as the differing trees in the foreground in the figure 6, as well as for more systematically placed wrong correspondences, such as the repeated image contents in figures 6, 13 and 15.

All in all, for all image sets, the matching and estimation seems to be quite robust for getting some kind of sensible results, but are not necessarily very accurate. Some of the image sets, particularly the $Ακρόπολη$ image set, are also fairly challenging, yet the algorithms succeed.

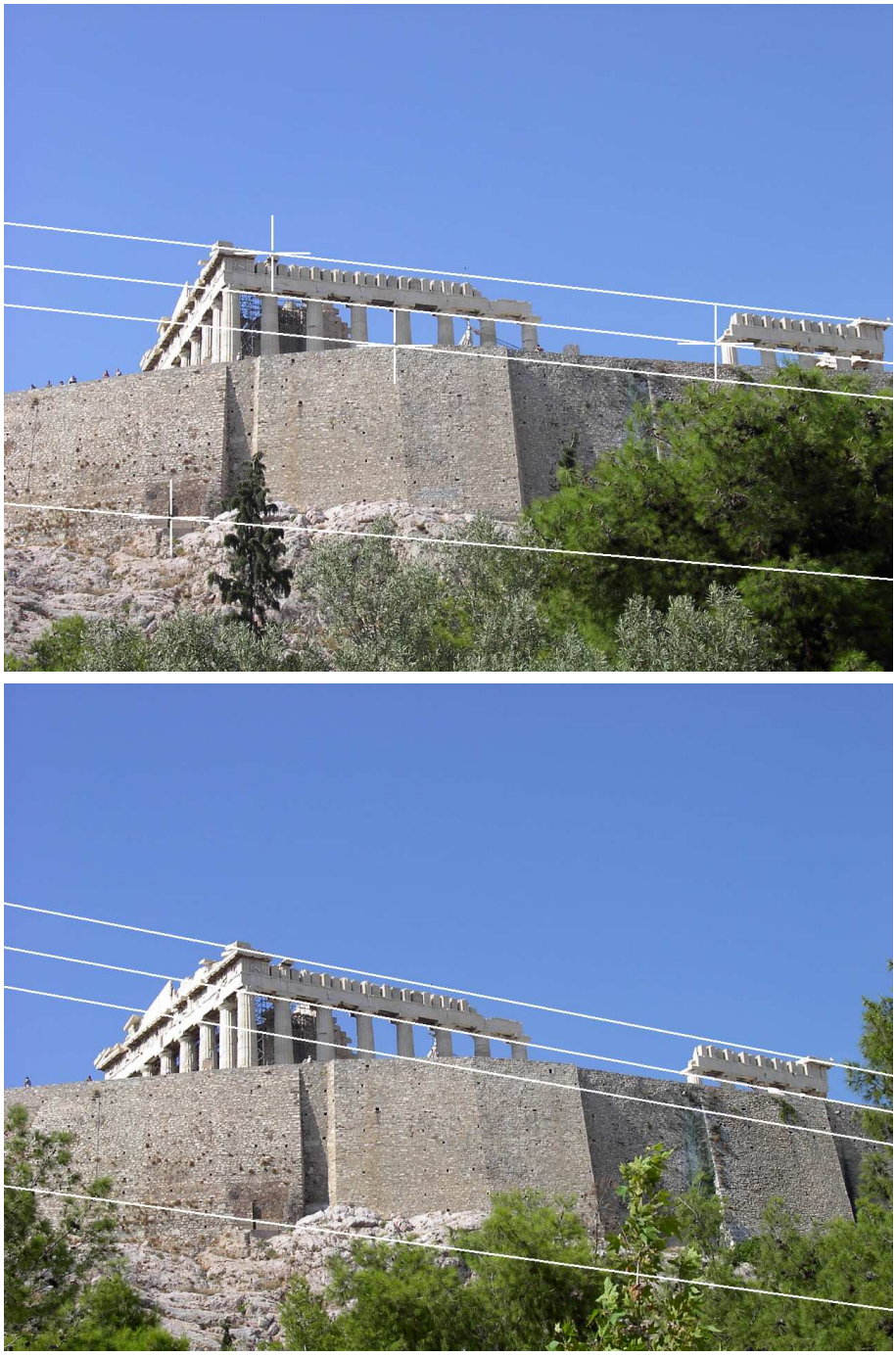

Figure 5: Images 6 (top) and 7 (bottom) of the Athens *Ακρόπολη* image sequence, photographed by the author in August 2002. The images were photographed at a resolution of 2048x1536 pixels and then scaled down to 1280x960 pixels. Four manually selected points, shown as cross-hairs, and their computed epipolar lines are shown for visual inspection. The epipolar lines here are not entirely accurate. E.g. the point at the tip of the large wall in the first image is not matched with the epipolar line in the second image. This is likely due to correspondences wrongly taken to be inliers, as seen in figure 6

Figure 6: This is the image pair from figure 5, where white lines connect the detected inlier points with their corresponding point positions in the other image. These lines have a white dot at the point in the image and a black dot at the corresponding point from the other image. Notice the two correspondences to the right on the white temple, which are "off by a few stones", meaning that these points are wrongly taken to be inliers. There are also a few other correspondences, which seem to defy the overall camera motion. In particular, the trees and bushes in the foreground are different in the two images, so it is not even possible to match these correctly

Figure 7: Images 1 (top) and 3 (bottom) of the Copenhagen Nordic Mythology Graffiti Wall image sequence, photographed by the author in November 2009. The images were photographed at a resolution of 2048x1536 pixels and then had only the wall cropped out, which makes them around 1800x425 pixels. This image pair is an example of degenerate planar scene structure. Four manually selected points, shown as cross-hairs, and their computed epipolar lines are shown for visual inspection. Notice that the epipolar lines intersect in the middle of the image at the same point, which suggests that the fundamental matrix has been estimated as if there is a forward camera motion between the images, which is actually not the case; the camera positions are side-by-side. A plausible reason for this estimation degeneracy is the planar scene structure

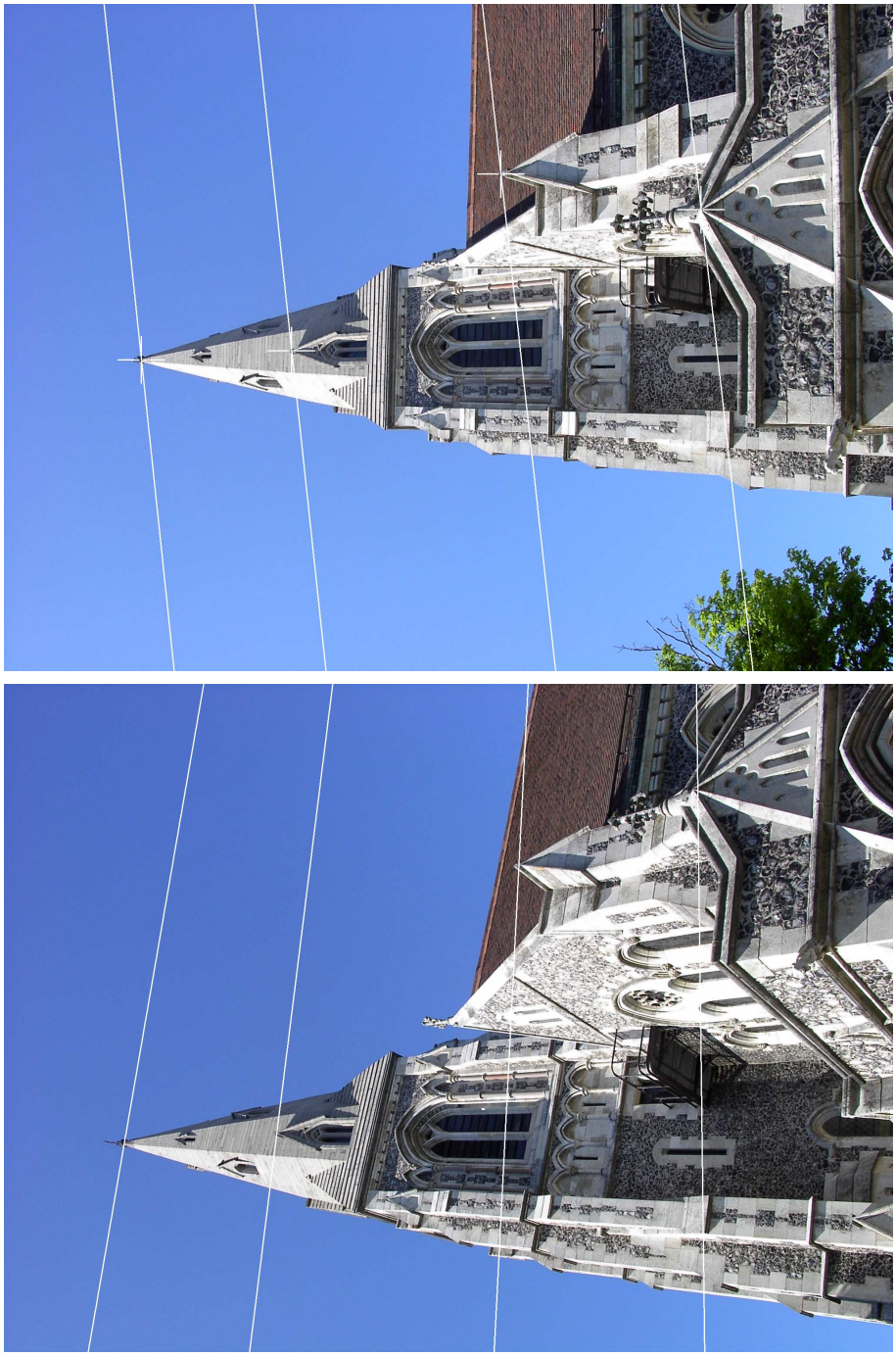

Figure 8: Images 46 (top) and 47 (bottom) of the Copenhagen St. Alban Anglican Church image sequence, photographed by the author in June 2009. Both pictures were taken with the orientation seen here, and have not been rotated 90 degrees, for demonstrating that the algorithms don't care about orientation. Notice the substantial occlusion by the foreground entrance towards the top right of the images (with the orientation shown here). Four manually selected points, shown as cross-hairs, and their computed epipolar lines are shown for visual inspection. Notice how the right-most epipolar line passes through the tip of the entrance in both images, even though there is a significant parallax motion on that entrance, in comparison to the rest of the scene

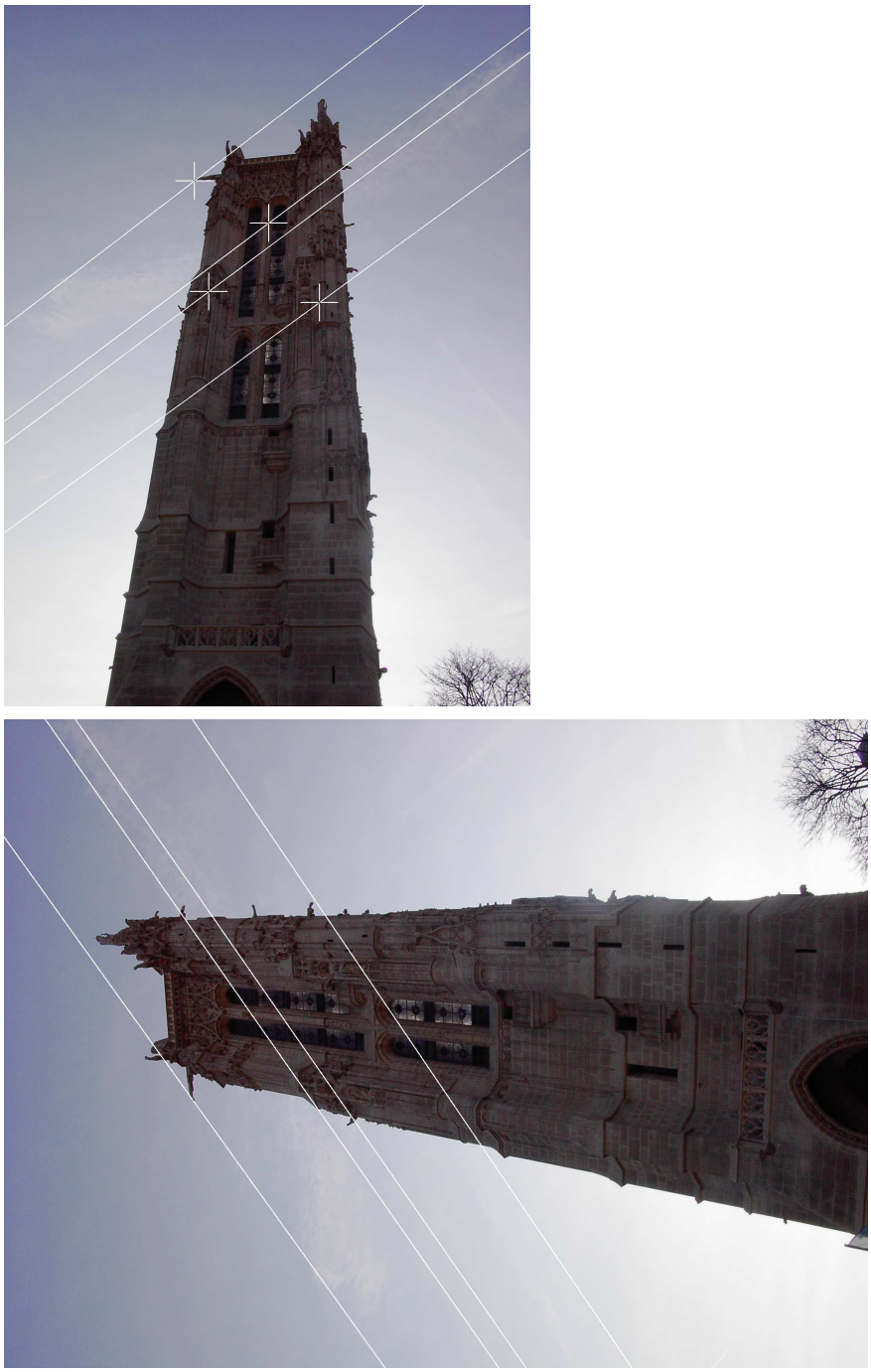

Figure 9: Images 5 (top) and 6 (bottom) of the Paris Tour St. Jacques image sequence, photographed by the author in January 2009. Both images were taken with the orientation seen on image 6 (bottom), but image 5 (top) was rotated 90 degrees, to make the tower upright, for demonstrating the support for rotation invariance. Four manually selected points, shown as cross-hairs, and their computed epipolar lines are shown for visual inspection. For this image set, when certain parameters were changed during development (e.g. 32-bit vs. 64-bit reals or normalized vs. unnormalized coordinates), the epipolar lines changed quite a bit, even going from vertical to horizontal. Possible explanations are that there is a lot of repeated scene content, giving false inliers, and that the facade of the tower is nearly planar, which is a degenerate scene structure. Both may affect

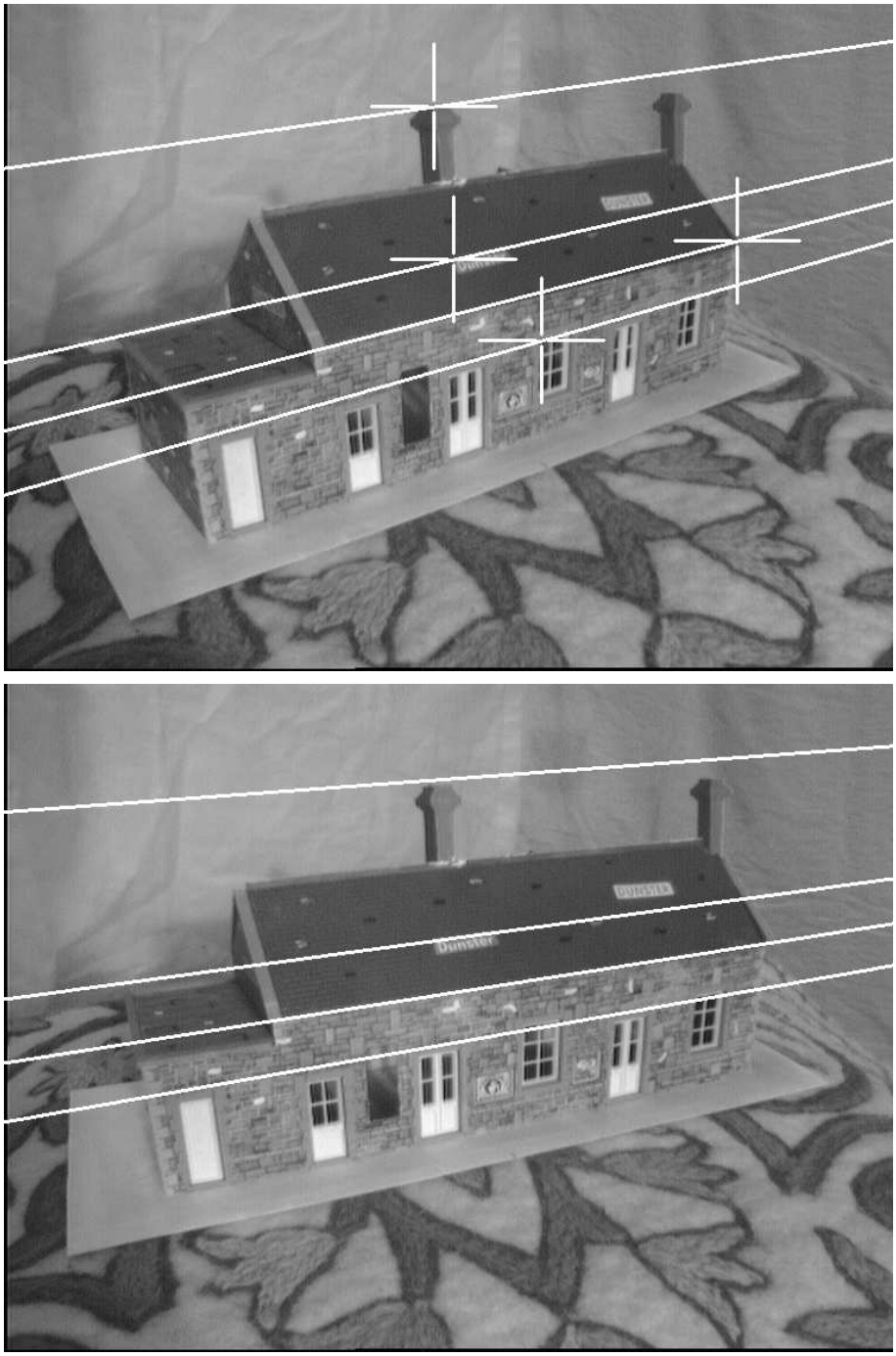

Figure 10: Images 0 (top) and 1 (bottom) of the Model House image sequence from http://www.robots.ox.ac.uk/ vgg/data/data-mview.html. Four manually selected points, shown as cross-hairs, and their computed epipolar lines are shown for visual inspection. The epipolar lines follow the structure of the house, which is suspicious and likely due to correspondences wrongly taken to be inliers, due to repeated image contents, as seen in figure 13

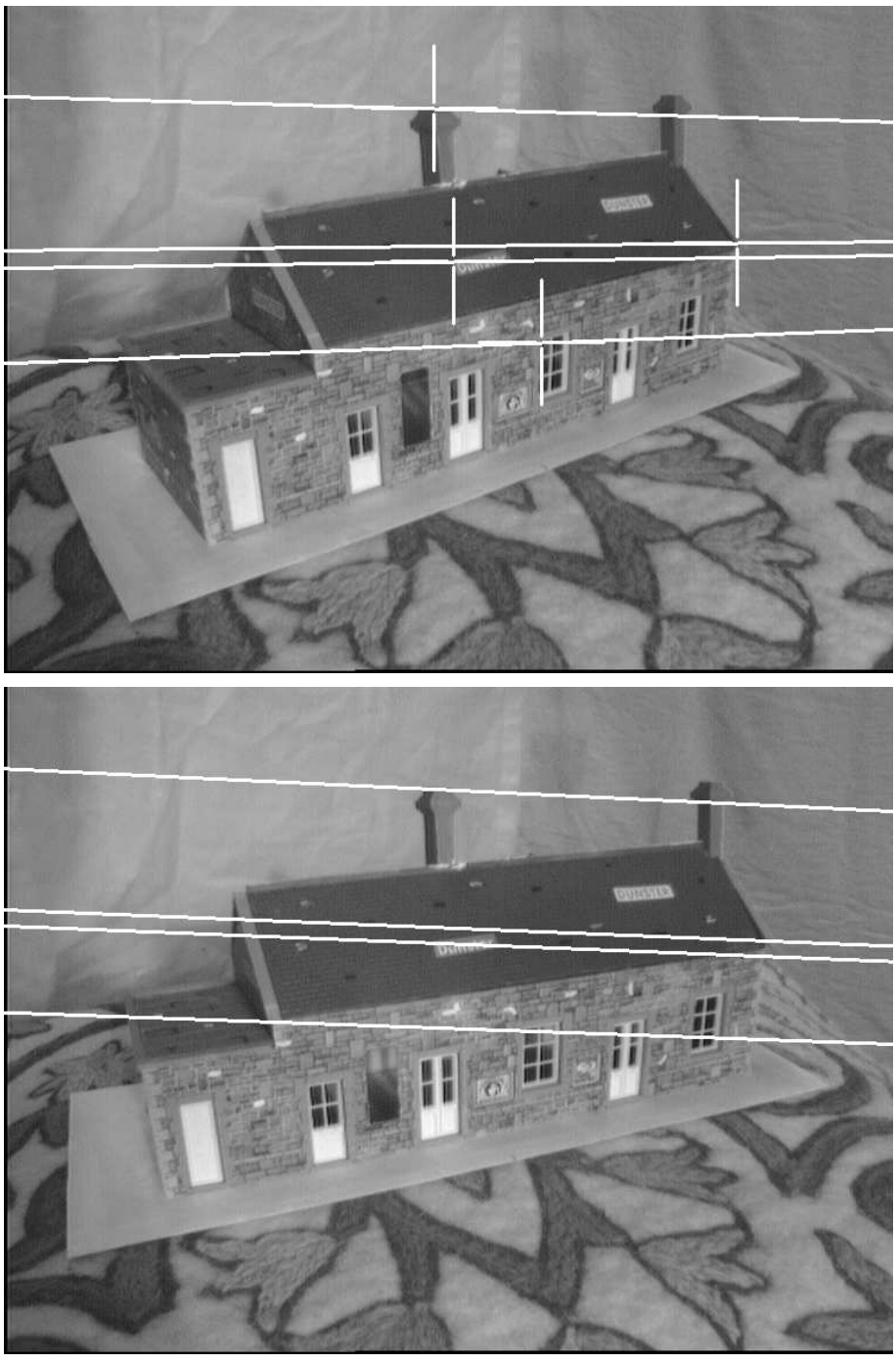

Figure 11: This figure is similar to figure 10, except that it shows the epipolar lines of the pre-estimated fundamental matrix from http://www.robots.ox.ac.uk/ vgg/data/data-mview.html

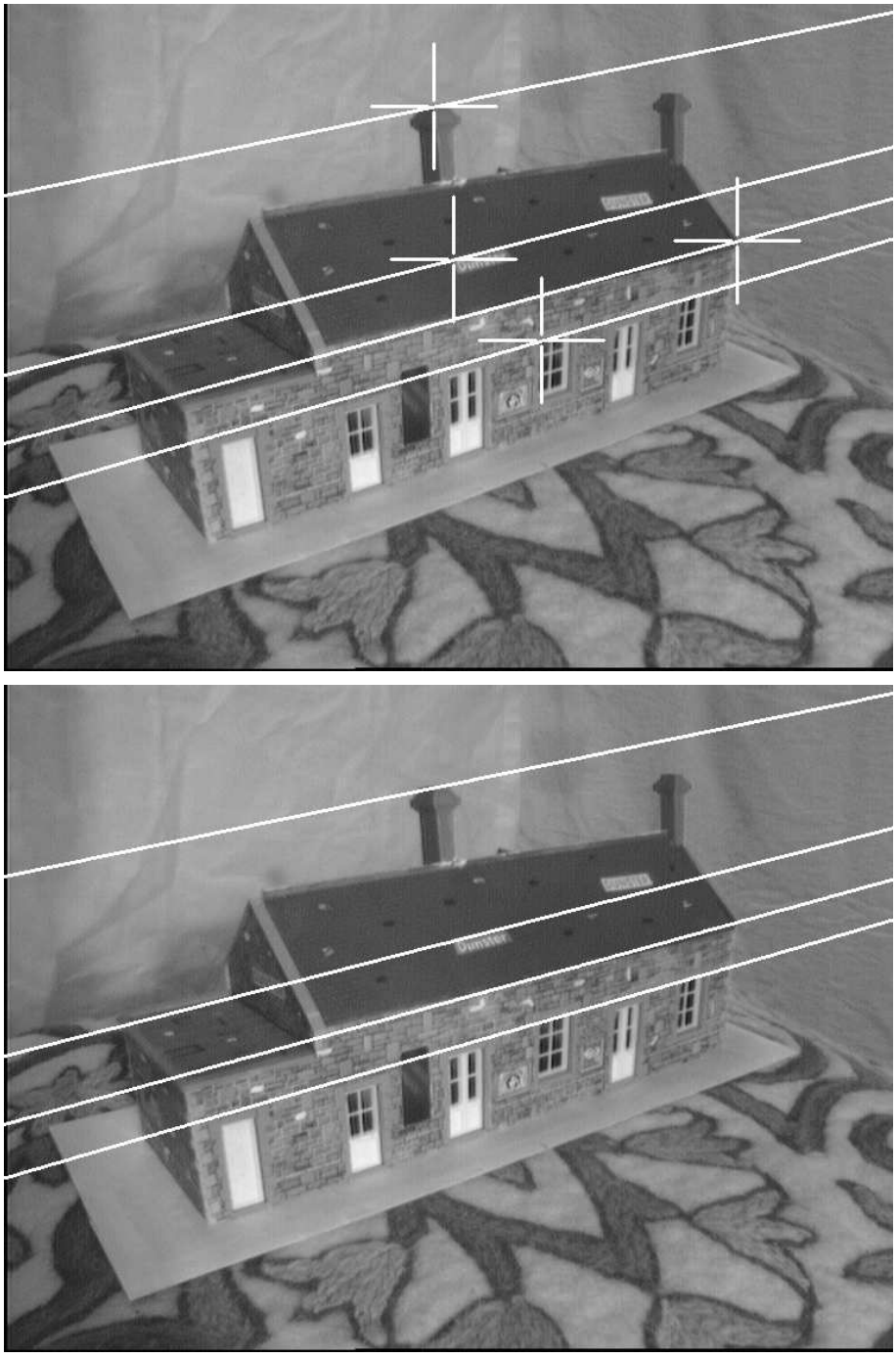

Figure 12: This figure is similar to figure 10, except that both images are image 0 of the Model House. This gives an important degenerate camera motion: when the camera does not move. This case makes some methods break down, but the results here look sensible. The author checked that all detected points were matched to the points themselves, by inspecting the image of matched correspondence points, but that image will not be shown, since it just contains black dots at the detected points. This also concurs with the obtained RMS residual error from section 7.9, which is on the order of 10^{-14} pixels. So, the fact that the lines follow the structure of the house, although it looks suspicious, cannot be attributed to correspondences wrongly taken to be inliers

Figure 13: This is the image pair from figure 10, where white lines connect the detected inlier points with their corresponding point positions in the other image. These lines have a white dot at the point in the image and a black dot at the corresponding point from the other image. Notice that there are a few correspondences on the side of the house which are quite long and seem to match "the wrong windows or doors" on the house, so these points are wrongly taken to be inliers

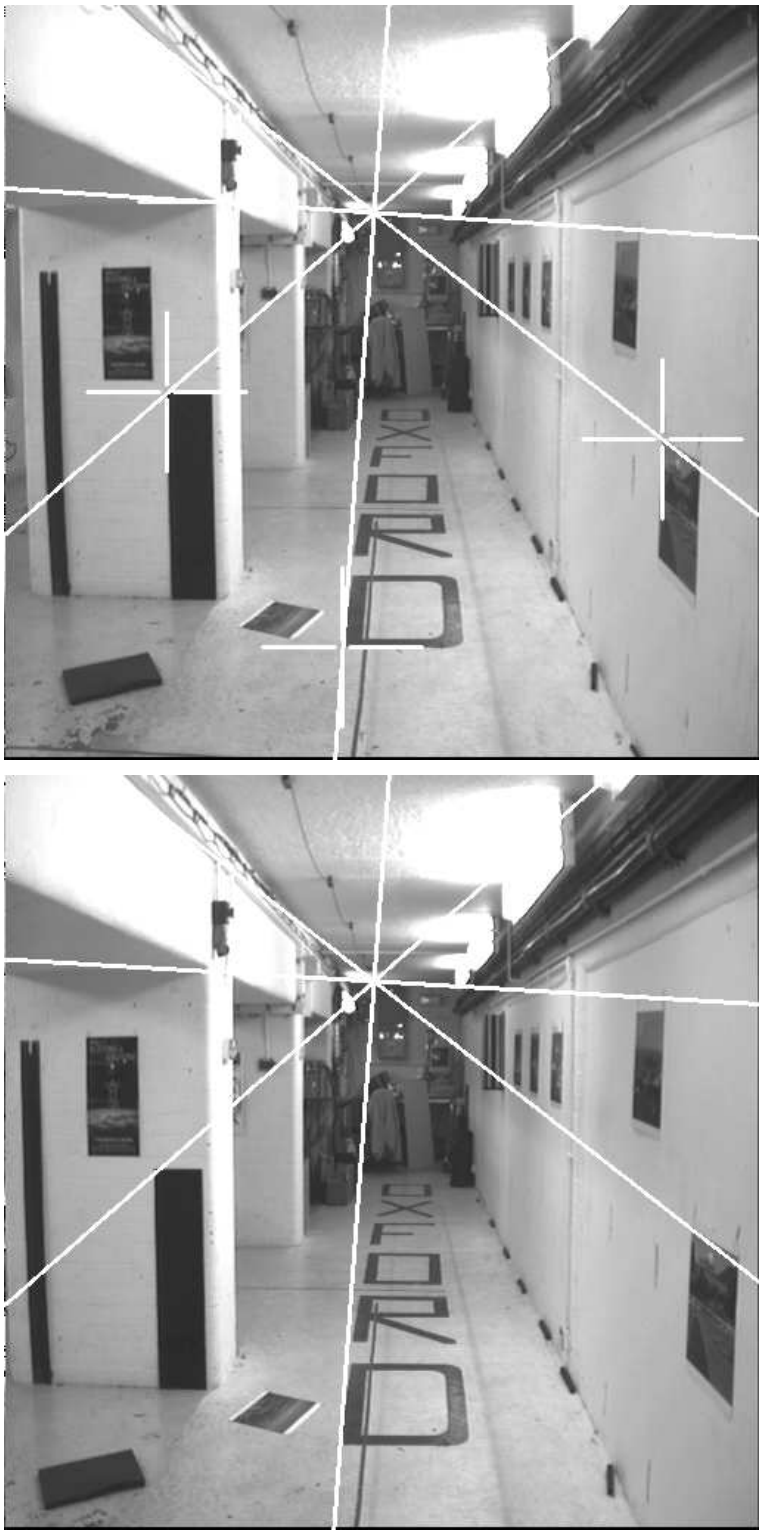

Figure 14: Images 0 (top) and 1 (bottom) of the Oxford Corridor image sequence from http://www.robots.ox.ac.uk/ vgg/data/data-mview.html. This image pair is an example of degenerate camera motion: straight forward motion without rotation. Four manually selected points, shown as cross-hairs, and their computed epipolar lines are shown for visual inspection

Figure 15: This is the image pair from figure 14, where white lines connect the detected inlier points with their corresponding point positions in the other image. These lines have a white dot at the point in the image and a black dot at the corresponding point from the other image. Notice that there are a few correspondences at the right edge of the floor, which seem to be longer than they should, e.g. longer than other lines on the floor which are even closer to the camera. These correspondences are therefore wrongly taken to be inliers

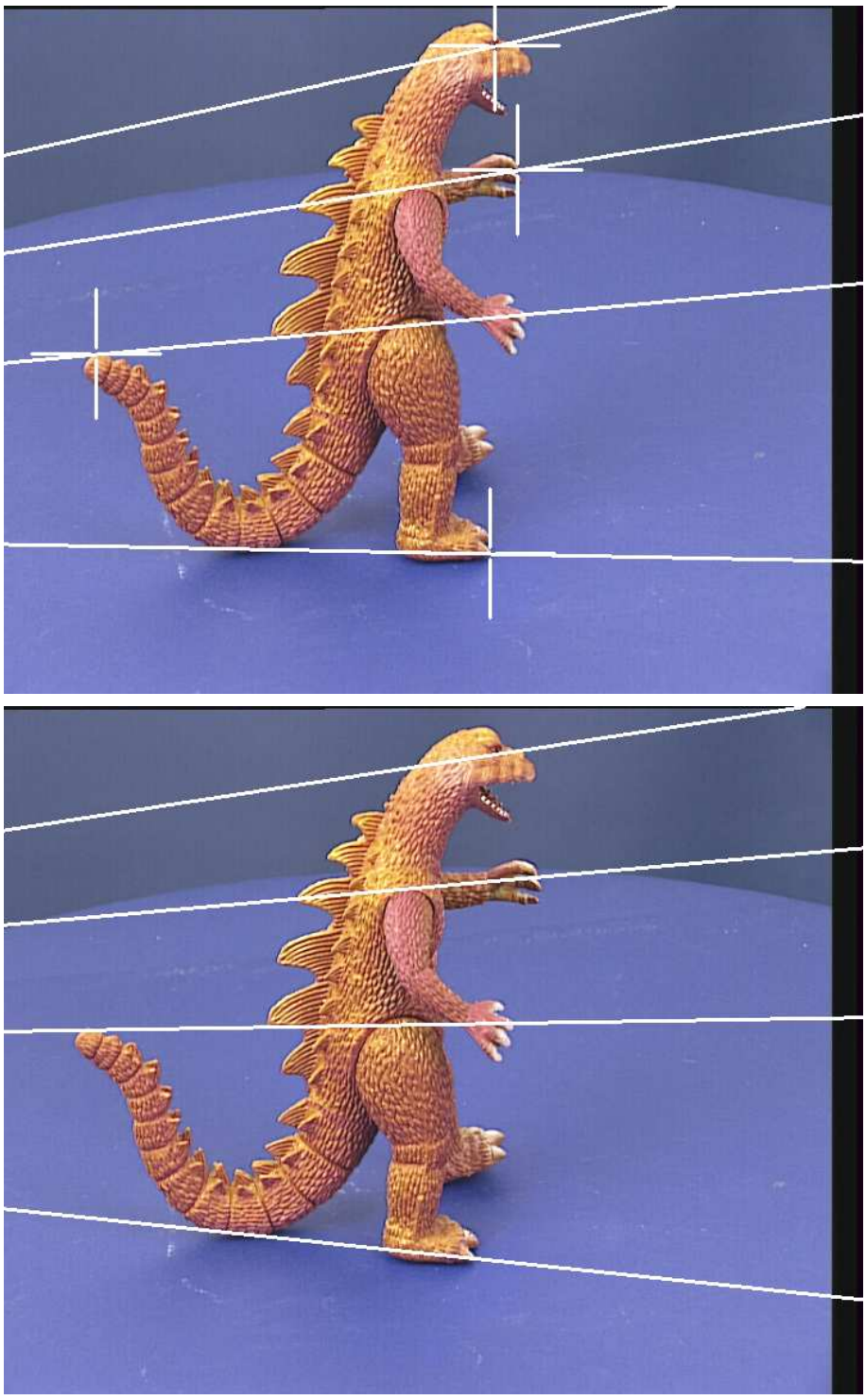

Figure 16: Images 1 (top) and 2 (bottom) of the Dinosaur image sequence from http://www.robots.ox.ac.uk/ vgg/data/data-mview.html. Four manually selected points, shown as cross-hairs, and their computed epipolar lines are shown for visual inspection

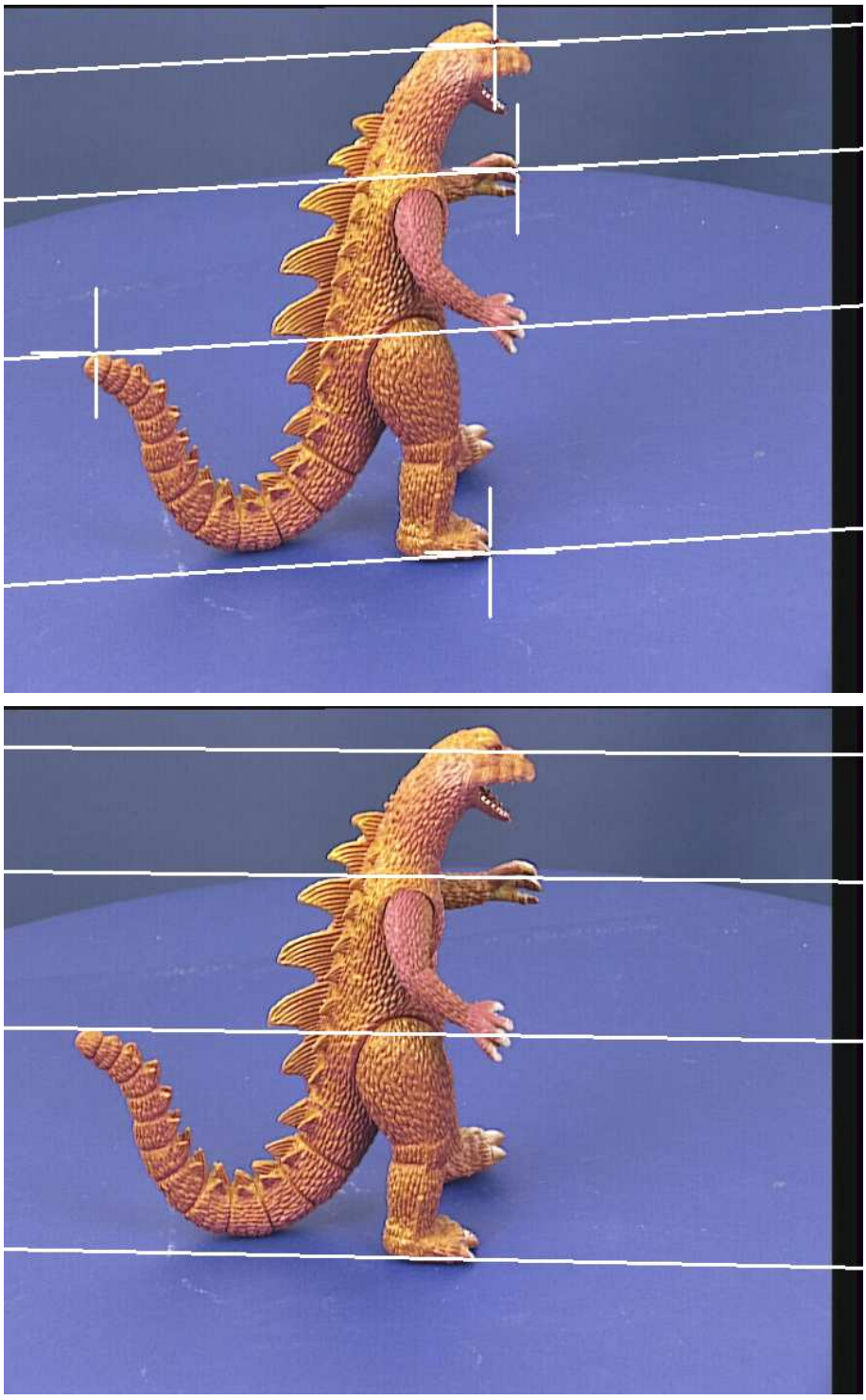

Figure 17: This figure is similar to figure 16, except that it shows the epipolar lines of the pre-estimated fundamental matrix from http://www.robots.ox.ac.uk/ vgg/data/data-mview.html

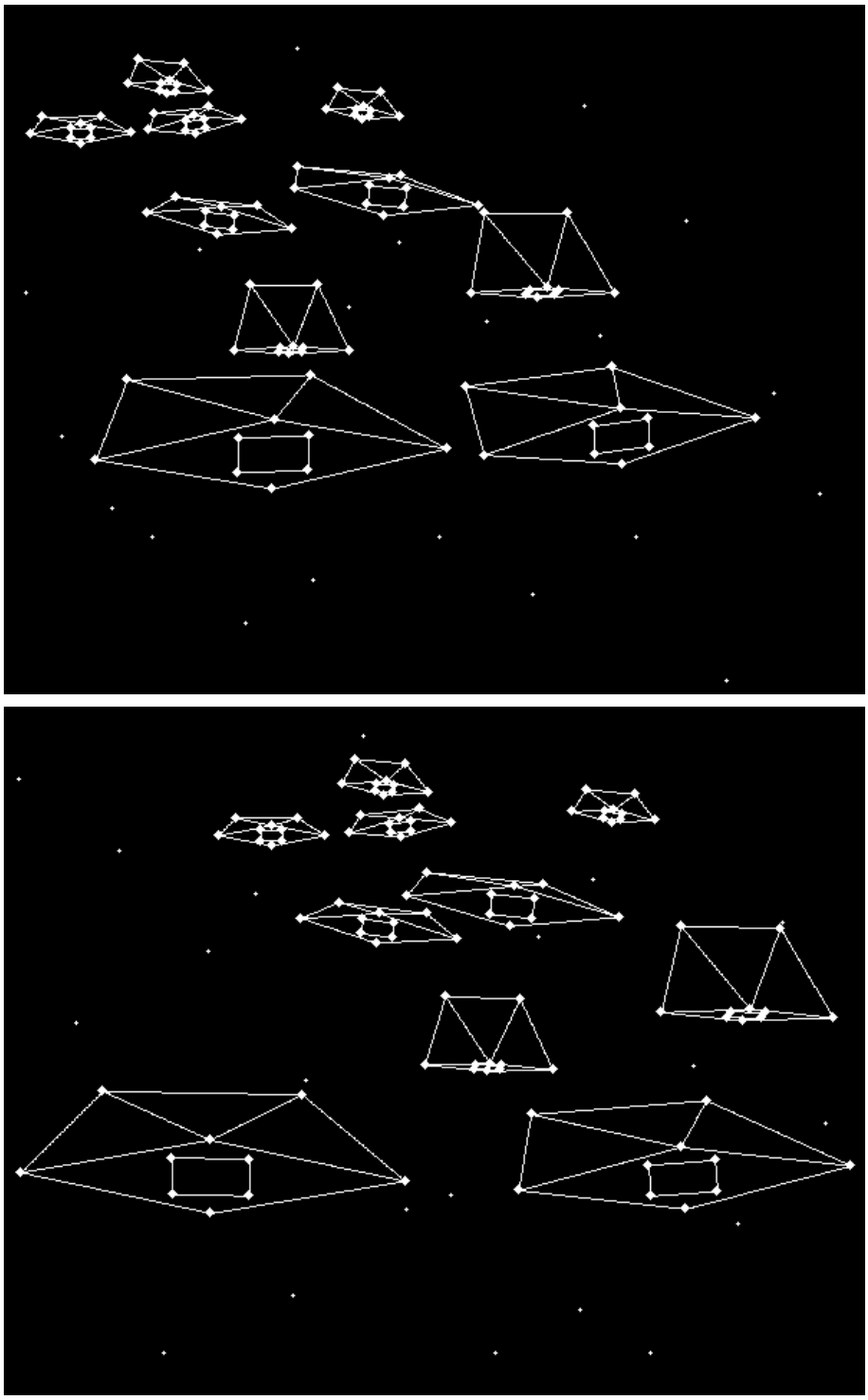

Figure 18: Images 1 (top) and 2 (bottom) of the synthetically generated Elite Sidewinders 3D scene with 100 true point correspondences and 20 outlier correspondences. The true point correspondences are shown as large dots connected with lines, for clearer visualization. The outlier points are the smaller unconnected dots. This data set is the only set of correspondence points where the ground truth point correspondences are available for comparison

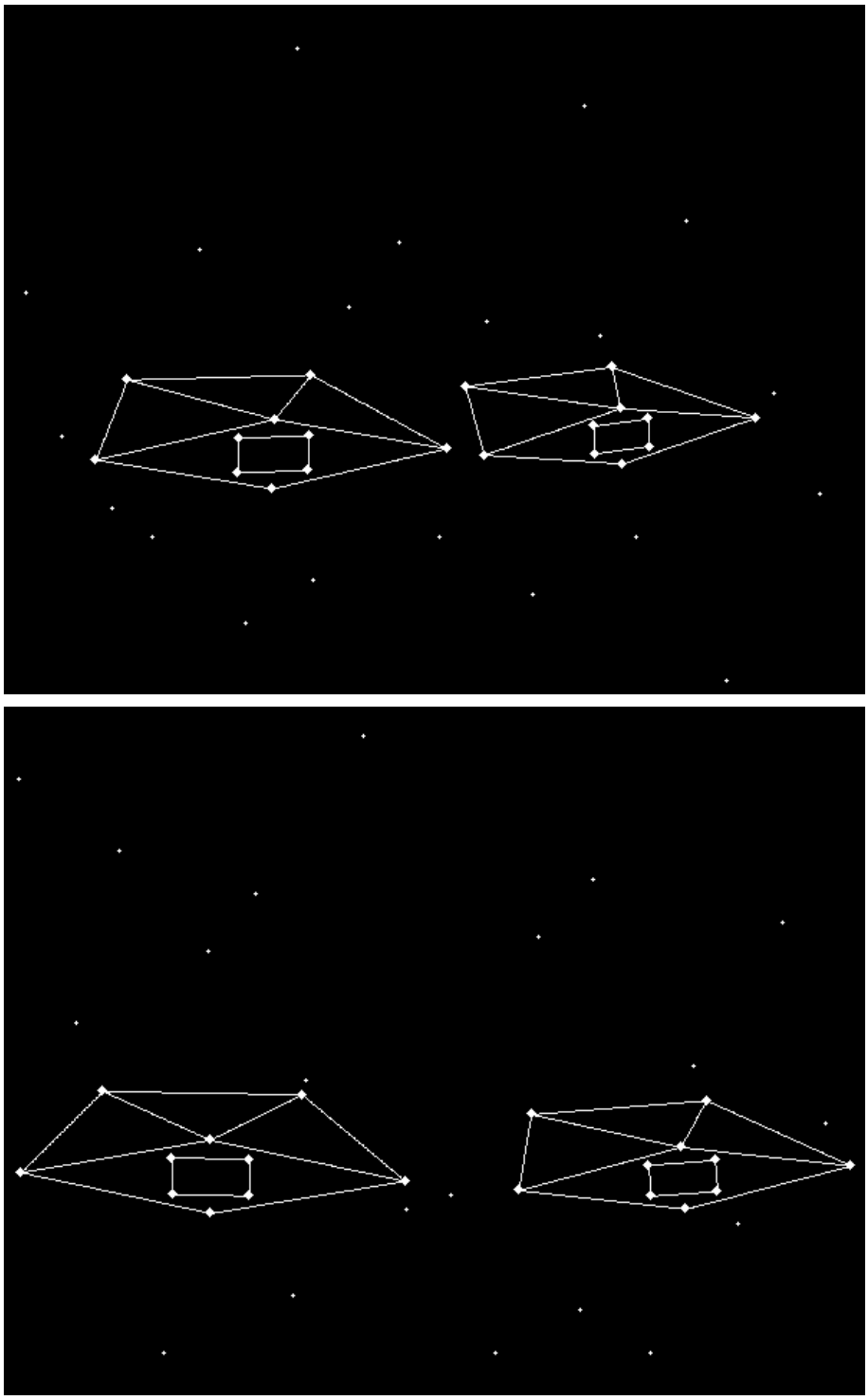

Figure 19: Images 1 (top) and 2 (bottom) of the synthetically generated Elite Sidewinders 3D scene with 20 true point correspondences and 20 outlier correspondences. The true point correspondences are shown as large dots connected with lines, for clearer visualization. The outlier points are the smaller unconnected dots. This data set is a reduced version of the data set with 100 true point correspondences, where the ground truth point correspondences are available for comparison

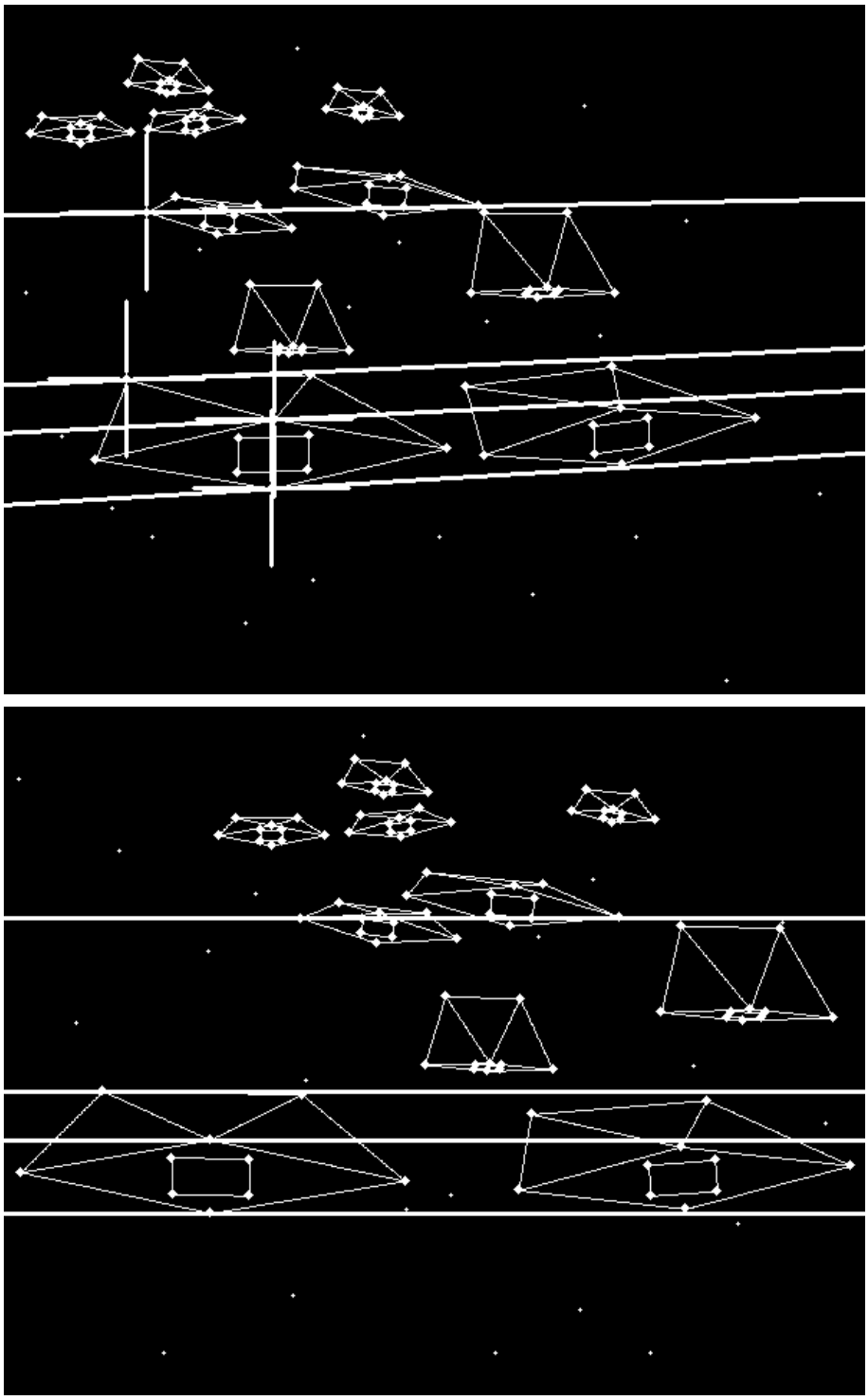

Figure 20: Images 1 (top) and 2 (bottom) of the synthetic Elite Sidewinders scene with epipolar lines for visual inspection. No noise has been injected into any coordinates in the version of the images shown here. Combining this fact with the obtained RMS residual error from section 7.9, which is on the order of 10^{-14} pixels from the ground truth correspondences, gives evidence that these epipolar lines can be taken to be the ground truth

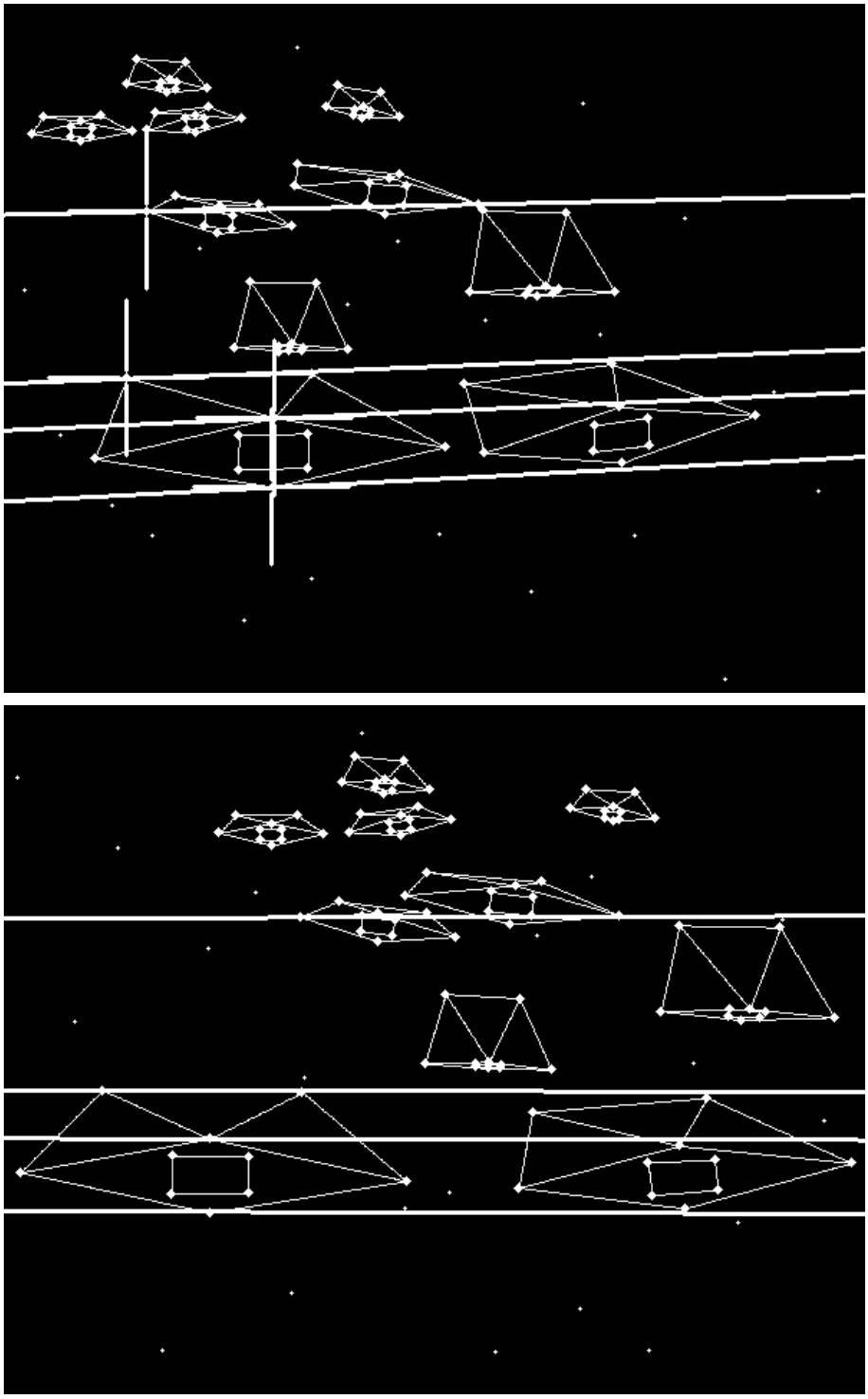

Figure 21: Images 1 (top) and 2 (bottom) of the synthetic Elite Sidewinders scene with epipolar lines for visual inspection. 1 pixel of noise has been injected into all coordinates in the version of the images shown here. Notice that the epipolar lines do not deviate significantly from the ground truth lines in figure 20

7.9 Measurements of Fundamental Matrix Estimation Accuracy

Below is a table presenting the residual RMS error for the various data sets, computed as described in section 7.3 from the detected inlier correspondences. The synthetic Elite scenes have various amounts of noise injected into the point correspondences before the fundamental matrix estimation is done. If the specified amount of maximum noise is p pixels, then the new points are computed by adding a random offset between $-p$ and p pixels in each of the x and y-coordinates of the points, meaning that the noise is not Gaussian, but a "flat" or "box" distribution, i.e. white noise inside a bounded square area.

Test data set	Number of inliers	Residual RMS error of **estimated** matrix versus **detected** (noisy) inliers
$Ακρόπολη$	19	1.088 pixels
Noric Mythology Graffiti Wall	213	0.389 pixels
Tour St. Jacques	69	0.545 pixels
St. Alban Anglican Church	175	0.897 pixels
Model House	36	1.069 pixels
Model House image 0 with itself	761	$8 \cdot 10^{-14}$ pixels
Oxford Corridor	28	0.506 pixels
Dinosaur	20	0.876 pixels
Synthetic Elite (100), max 0.00 pixels noise	100	$2 \cdot 10^{-14}$ pixels
Synthetic Elite (100), max 0.25 pixels noise	100	0.195 pixels
Synthetic Elite (100), max 0.50 pixels noise	98	0.403 pixels
Synthetic Elite (100), max 0.75 pixels noise	97	1.208 pixels
Synthetic Elite (100), max 1.00 pixels noise	91	0.838 pixels
Synthetic Elite (20), max 0.00 pixels noise	20	$4 \cdot 10^{-14}$ pixels
Synthetic Elite (20), max 0.25 pixels noise	20	0.187 pixels
Synthetic Elite (20), max 0.50 pixels noise	20	0.447 pixels
Synthetic Elite (20), max 0.75 pixels noise	21	0.910 pixels
Synthetic Elite (20), max 1.00 pixels noise	20	1.254 pixels

The following table shows the RMS residual error of the pre-estimated fundamental matrices versus the detected inlier correspondences for the Model House and the Dinosaur image sets.

Test data set	Number of inliers	Residual RMS error of **pre-estimated** matrix versus **detected** (noisy) inliers
Model House	36	14.368 pixels
Dinosaur	20	0.880 pixels

For the synthetic Elite scene, the ground truth is available for comparison. Below is a table which presents the residual RMS error of the estimated fundamental matrix with respect to the true inlier correspondences, i.e. before noise perturbation. Except for the true correspondence points being used for computation, the formula is still the one from section 7.3.

Test data set	Number of inliers	Residual RMS error of **estimated** matrix versus **ground truth inliers**
Synthetic Elite (100), max 0.00 pixels noise	100	$2 \cdot 10^{-14}$ pixels
Synthetic Elite (100), max 0.25 pixels noise	100	0.066 pixels
Synthetic Elite (100), max 0.50 pixels noise	98	0.159 pixels
Synthetic Elite (100), max 0.75 pixels noise	97	1.070 pixels
Synthetic Elite (100), max 1.00 pixels noise	91	0.427 pixels
Synthetic Elite (20), max 0.00 pixels noise	20	$4 \cdot 10^{-14}$ pixels
Synthetic Elite (20), max 0.25 pixels noise	20	0.189 pixels
Synthetic Elite (20), max 0.50 pixels noise	20	0.266 pixels
Synthetic Elite (20), max 0.75 pixels noise	21	1.363 pixels
Synthetic Elite (20), max 1.00 pixels noise	20	1.438 pixels

A few of the RMS residual error values in the above tables are not impressive, but most others are acceptable. Some of the worst values have explanations:

- Several correspondences are wrongly taken to be inliers in the $A\kappa\rho\acute{o}\pi o\lambda\eta$ image set, as seen in figure 6. This can explain the relatively high error
- Some correspondences are significantly wrong in the Model House image set, due to repeated image contents, as seen in figure 13. These large inlier offsets, and the different epipolar line orientations for the pre-estimated fundamental matrix, can explain the extremely big difference between the pre-estimated fundamental matrix and the detected inlier correspondences
- In some of the synthetic Elite evaluations, a different number than the 100 or 20 expected inlier correspondences are sometimes found. It is particularly in these cases that the error rates are high, as might be expected

The reader should be warned about making too direct comparisons based on the presented numbers. As an example, comparing the RMS residual error of the pre-estimated fundamental matrices (call them ϵ_{PRE}) with those estimated by the implemented methods (ϵ_{IMPL}) does not in itself say anything about which estimate is best. E.g. if $\epsilon_{PRE} > \epsilon_{IMPL}$, this may either be due to a worse fundamental matrix pre-estimate (counting in favour of the implementation) or due to worse detected point correspondences (counting in favour of the pre-estimated matrices). However, we have clear evidence of having both a low number of correspondences and wrongly classified inliers, so from this we assume the presented implemetation's estimates to be the worst; this fact is just not derivable from the presented numbers.

As a warning related to the interpretation of the magnitude of the numbers, the magnitude may well be related to how large the tolerated inlier distance is in the estimation algorithms. We allow a distance of 3 pixels in *each* image (see section 6.7.3), whereas e.g. section 11.6 in [Hart03] allows only 1.25 pixels, which even seems to be in the sum of the two image distances. A larger allowed distance (as used here) may handle large scale changes better, but may give less accuracy, so there is a trade-off to be made, which is not necessarily related to the quality of the implemented methods.

As already mentioned, these results have not remained completely stable when small changes were made to the program (see section 6.12), but this stability actually seems (without having been properly verified) to have improved, after correcting a mistake of the non-linear optimizating only optimizing on distances in the second image. It would be appropriate to run the evaluations multiple times (with differing initial random seeds for the RANSAC algorithm) and average the results, particularly for the synthetic Elite scenes. This has not been done, but is recommended for future work.

It can be concluded that most of the inaccuracies seem to come from correspondences wrongly classified as inliers, often due to repeated image content. The most feasible way to solve that is probably to have more correspondences. Another source of inaccuracies may be the relatively low number of correspondences, which suggests adding an additional feature detector in the guided matching phase - and enabling and making the guided matching phase more robust in the first place.

The general conclusions still concur with those of the visual inspection, from section 7.8.2.

7.10 Various Observations During the Development Phase

The development of the evaluation software has been made incrementally in quite well-defined steps. During many of these steps, specific expected improvements were observed, even though in many cases the observations were made only for a single set of test data. Some of the observed improvements were between the following phases:

- The 8-point algorithm (section 6.3) without coordinate normalization (section 6.1) and without enforcement of the singularity contraint (section 6.2.1) was implemented and tested initially
- A small program for manually selecting 8 point correspondences and estimating a fundamental matrix by using the 8-point algorithm was made. This program was made for visual testing of the 8-point algorithm and for acquiring coordinates for automated unit-testing of the various implemented fundamental matrix estimation algorithms, i.e. unit-test data for the 8-point algorithm (section 6.3), the 7-point algorithm (section 6.4.1), RANSAC (section 6.7) and the non-linear optimization (section 6.9)
- The singularity constraint (section 6.2.1) was implemented for the 8-point algorithm. It was noticed that epipolar lines did not meet in a single point until after this was implemented, as expected. However, this enforcement was seen to be so crude that in the case of only 8 point correspondences, the quality of some of the individual epipolar lines was severely degraded, in comparison to performing no singularity constraint enforcement
- Coordinate normalization (section 6.1) was implemented for the 8-point algorithm, which in the unit-test gave rise to a smaller difference between the estimated and the expected epipolar lines. This difference was measured on the positions of the end-points of the epipolar lines, where they leave the image, as explained in section 7.4
- The RANSAC algorithm (section 6.7) was implemented and unit-tested, initially using the normalized 8-point algorithm and the (sub-optimal) residual error measure (section 6.5.3)
- The 7-point algorithm (section 6.4.1) was implemented. The RANSAC algorithm was updated to use this, which seemed to significantly improve the quality of the RANSAC fundamental matrix estimate. Also the computation time decreased
- The linear triangulation algorithm (section 6.6.1) was implemented, along with the geometric reprojection error measure (section 6.5.1). Replacing the residual error measure by the geometric reprojection error measure in the RANSAC algorithm improved the quality of the RANSAC fundamental matrix estimate, as one might hope
- For the RANSAC unit-test it was observed that when having 50 percent outliers and 50 percent inliers, at least 12 liniers (i.e. 24 correspondences in total) were necessary (in the single point correspondence set considered) to make a correct fundamental matrix estimate. This seems

natural, since there would at least have to be a small number of point correspondences, in addition to the 7 estimation correspondences, to verify that the estimated fundamental matrix is correct for more than the 7 estimation correspondences

- The non-linear optimization (section 6.9) was implemented and an estimation pipeline only lacking guided matching was formed (section 6.11). The non-linear optimization seemed to improve the fundamental matrix estimate slightly, but this has only been qualitatively verified by visual inspection by the author, not by unit-test measurements

- One round of guided correspondence matching (section 6.11) and one extra round of non-linear optimization (section 6.9) was added to the pipeline. This gave only a few extra inliers (after re-running RANSAC) in the test-image pairs evaluated. However, it actually did not seem to improve the RMS residual errors, probably due to fewer wrong inliers being filtered out by the nearest-to-second-nearest neighbour ratio matching criterion. A conclusion to be drawn from this may be that the Maximally Stable Extremal Region (MSER) detector with Speeded-Up Robust Features (SURF) descriptors and the nearest-to-second-nearest neighbour ratio are already quite robust, even without geometric guidance, and that an additional feature detector should be used in this phase of the pipeline, in order to get any improvement. The guided geometric matching was removed again and is not used in the evaluation

- Up to this point, most of the previously described improvements were observed when using RANSAC and non-linear optimization with unnormalized coordinates and 32-bit floating point numbers. When enabling normalized coordinates (section 6.1), the estimates were somewhat degraded, probably due to the limitations of the 32-bit precision, since the normalized coordinate values are smaller. Changing the 32-bit floating point precision to 64-bit precision significantly improved the results and unfortunately this was done only towards the end of the project

- At the end of the project, it was discovered that the non-linear optimization (section 6.9) only optimized the distances in the second of the two images. Correcting this mistake made the results improve noticably, in terms of consistently lower residual RMS error values. The numerical stability of the residual RMS errors even seems to have improved by this, although this has not been verified

- The RANSAC algorithm was changed at the end of the project to select inliers by testing distances in each image individually (section 6.7.3), rather than in the sum of the two image distances. This seemed to improve things slightly, in terms of a few inlier counts changing, mostly giving improved residual RMS errors. The changes happened consistently when enabling and disabling this before, respectively after, correcting the mistake mentioned above in the non-linear optimization

The above observations may not be reliable, but they seem to indicate that all the implemented methods actually contribute to a better solution, which gives some measure of justification for the implementation effort.

8 Future Work

- Implement the optimal triangulation method, which might improve the quality and increase the convergence rate of the fundamental matrix estimate (sections 6.6.2 and 6.13)

- Consider using a better parameterization of the fundamental matrix for the non-linear optimization, which also might improve the fundamental matrix estimation (sections 6.8 and 6.13)

- Consider handling the degenerate cases of coincident camera centres and planar scene geometry by estimating a homography or the camera rotation (sections 6.4.2, 6.13 and 7.8.2)
- Supporting more than two views (section 6.13)
- Consider using the method *Total Least Squares - Fixed Columns* for point coordinate normalization, which might improve the fundamental matrix estimates slightly (section 6.1)
- Implementing a sparse version of the Levenberg-Marquardt optimization should allow support for more correspondences and give faster computation of the non-linear optimization, which is currently very slow (section 6.13)
- Replace the RANdom SAmpling Consensus (RANSAC) method with PROgressive SAmpling Consensus (PROSAC) from [Chum05], which should give a significant speed-up and may even give other benefits and improvements (section 6.13)
- Avoiding needless computations in the Singular Value Decomposition (SVD) should give some speed-up (section 6.13)
- Improve on the methods from the previous report [Anoq09] from its suggested future work
- Perform some of the evaluations with larger sets of point correspondences and more test images, as well as performing them as averages over multiple runs with differing RANSAC random seeds
- Make the guided matching phase work properly and robustly (section 6.11.1)
- Extend the pipeline to include dense metric 3D model reconstruction, where interesting and relevant road maps can be found in e.g. [Poll00] and [Poll04]

9 Conclusions

The contributions of this report are the following:

- A check for the sample point selection in the RANSAC algorithm, which prevents duplicate or close correspondences from being selected. This improves the possibility of using multiple feature detectors and avoids certain poor fundamental matrix estimates (section 6.7.1)
- Use of the Maximally Stable Extremal Region (MSER) detector with multiple different settings at the same time, which was not described in the article [Mata02] (section 6.10)
- A complete fundamental matrix estimation pipeline has been presented with most algorithms described in enough detail that they can be implemented from this report
- An quite thorough evaluation framework has been presented, which will be useful for future work (section 7)
- A thorough road-map for future work has been presented (section 6.13)
- The collection of formulas in section 5 is well worth mentioning; it is very useful

Most conclusions regarding the implementation were made in sections 7.8.2 and 7.9. The main conclusion to be drawn is that the implemented methods seem to work quite well and robustly, even for difficult image pairs. However, the implementation still only gives a rough fundamental matrix estimate and suffers from some problems and limitations, which have to be addressed. Many of the problems have been identified quite concretely and solutions have been suggested and recommended for future work.

The methods presented in this and the previous report [Anoq09] should provide a good basis for parts of the technology of Hardcore Processing's upcoming commercial application, *CeX3D Inverse* [C3DI10], which was the motivation for writing these reports.

10 Acknowledgements

Thanks to Søren Ingvor Olsen at the Computer Science Department of the University of Copenhagen (DIKU) for supervising this project with constructive feedback and good litterature references.

Appendices

A: Execution Log of the Evaluation Program

This section contains the output of running the implemented evaluation program.

```
--- Akropolh August 2002 image pair 6 and 7 (1280x960) ---:
Matched: 6(Delta20) + 13(Delta10) + 11(Delta5) = 30 correspondences [38.48secs]
RANSAC inliers: 19
Total time spent on the estimation (including matching): [38.62secs]
The estimated fundamental matrix is:
((4.11581098014E~9, ~4.52110924591E~7, 8.16536518609E~4),
(2.57537072766E~7, ~1.18248965757E~7, ~0.00344591843491),
(~4.85507351487E~4, 0.00359425356105, ~0.0305161154044))
Saving image ../outputData/images/akropolh2002_6_match6_7.bmp
Saving image ../outputData/images/akropolh2002_7_match6_7.bmp
Residual RMS error of estimated matrix vs. detected inliers: 1.08837510259
Saving image ../outputData/images/akropolh2002_6_epipolar6_7.bmp
Saving image ../outputData/images/akropolh2002_7_epipolar6_7.bmp
--- Nordic Mythology Graffiti Wall image pair 1 and 3 (approx. 1800x425) ---:
Matched: 26(Delta20) + 75(Delta10) + 117(Delta5) = 218 correspondences [25secs]
RANSAC inliers: 213
Total time spent on the estimation (including matching): [82.83secs]
The estimated fundamental matrix is:
((~1.36894322818E~8, ~6.68390341833E~6, 0.0021118951926),
(6.08233326577E~6, ~6.14036823504E~8, ~0.00707017689327),
(~0.00173435360195, 0.00746957792206, ~0.320787145778))
Saving image ../outputData/images/NordicMythologyGraffitiWall_1_match1_3.bmp
Saving image ../outputData/images/NordicMythologyGraffitiWall_3_match1_3.bmp
Residual RMS error of estimated matrix vs. detected inliers: 0.389475383538
Saving image ../outputData/images/NordicMythologyGraffitiWall_1_epipolar1_3.bmp
Saving image ../outputData/images/NordicMythologyGraffitiWall_3_epipolar1_3.bmp
--- Model House image pair 0 and 1 (768x576) ---:
Matched: 7(Delta20) + 13(Delta10) + 23(Delta5) = 43 correspondences [7.93secs]
RANSAC inliers: 36
```

```
Total time spent on the estimation (including matching): [8.17secs]
The estimated fundamental matrix is:
((1.72950216589E~7, 3.77450542724E~6, 1.86945356174E~4),
(~4.75422252027E~6, 1.42849663833E~6, 0.00954913653621),
(~8.54049013241E~4, ~0.00978594669876, 0.313070851963))
Saving image ../outputData/images/ModelHouse_0_match0_1.bmp
Saving image ../outputData/images/ModelHouse_1_match0_1.bmp
Residual RMS error of estimated matrix vs. detected inliers: 1.06928185385
The following pre-estimated fundamental matrix will be used:
((0.00842337618233, 0.0352341565115, ~35.2240134541),
(~0.197771227607, 0.0212943017827, 677.325329282),
(31.7672833959, ~625.885556948, 1013.55795724))
Residual RMS error of pre-estimated matrix vs. detected inliers: 14.3677884046
Saving image ../outputData/images/ModelHouse_0_preepipolar0_1.bmp
Saving image ../outputData/images/ModelHouse_1_preepipolar0_1.bmp
Saving image ../outputData/images/ModelHouse_0_epipolar0_1.bmp
Saving image ../outputData/images/ModelHouse_1_epipolar0_1.bmp
--- Corridor image pair 0 and 1 (512x512) ---:
Matched: 7(Delta20) + 12(Delta10) + 9(Delta5) = 28 correspondences [2.81secs]
RANSAC inliers: 28
Total time spent on the estimation (including matching): [2.92secs]
The estimated fundamental matrix is:
((3.72598189554E~7, ~5.7733927216E~5, 0.00801259564956),
(5.77445097527E~5, ~1.20013974448E~7, ~0.0144488245696),
(~0.00814563440238, 0.0144822784422, 0.00724155081861))
Saving image ../outputData/images/Corridor_0_match0_1.bmp
Saving image ../outputData/images/Corridor_1_match0_1.bmp
Residual RMS error of estimated matrix vs. detected inliers: 0.506176248888
Saving image ../outputData/images/Corridor_0_epipolar0_1.bmp
Saving image ../outputData/images/Corridor_1_epipolar0_1.bmp
--- Dinosaur image pair 1 and 2 (720x576) ---:
Matched: 9(Delta20) + 10(Delta10) + 7(Delta5) = 26 correspondences [7.82secs]
RANSAC inliers: 20
Total time spent on the estimation (including matching): [7.88secs]
The estimated fundamental matrix is:
((3.95523743957E~7, 5.08496234977E~6, ~0.00162182669416),
(~4.70152493724E~6, 2.10576190285E~7, ~0.00642580003115),
(0.0018532304758, 0.00613809634782, ~0.103473186724))
Saving image ../outputData/images/Dinosaur_1_match1_2.bmp
Saving image ../outputData/images/Dinosaur_2_match1_2.bmp
Residual RMS error of estimated matrix vs. detected inliers: 0.876073736902
The following pre-estimated fundamental matrix will be used:
((~7.41153115742E~6, ~1.47929568885E~4, ~0.0352496728166),
(~1.14686144642E~4, 5.41280808427E~6, 4.88688852488),
(~0.269313932911, ~4.78925490677, 106.72471217))
Residual RMS error of pre-estimated matrix vs. detected inliers: 0.879692188109
Saving image ../outputData/images/Dinosaur_1_preepipolar1_2.bmp
Saving image ../outputData/images/Dinosaur_2_preepipolar1_2.bmp
Saving image ../outputData/images/Dinosaur_1_epipolar1_2.bmp
Saving image ../outputData/images/Dinosaur_2_epipolar1_2.bmp
--- Tour St. Jacques image pair 5 and 6 (2048x1536) ---:
Matched: 2(Delta20) + 18(Delta10) + 55(Delta5) = 75 correspondences [28.48secs]
RANSAC inliers: 69
```

```
Total time spent on the estimation (including matching): [30.20secs]
The estimated fundamental matrix is:
((~1.35050768209E~7, ~1.60536724091E~8, ~0.00205758137282),
(~9.39049221331E~9, ~1.2737480168E~7, 0.00160875527346),
(0.00168713229083, 0.00223426304343, ~2.51305339424))
Saving image ../outputData/images/TourStJacques_5_match5_6.bmp
Saving image ../outputData/images/TourStJacques_6_match5_6.bmp
Residual RMS error of estimated matrix vs. detected inliers: 0.544708971932
Saving image ../outputData/images/TourStJacques_5_epipolar5_6.bmp
Saving image ../outputData/images/TourStJacques_6_epipolar5_6.bmp
--- St. Alban Anglican Church image pair 46 and 47 (2048x1536) ---:
Matched: 39(Delta20) + 72(Delta10) + 116(Delta5) = 227 correspondences [179.13secs]
RANSAC inliers: 175
Total time spent on the estimation (including matching): [210.03secs]
The estimated fundamental matrix is:
((4.61278454492E~9, ~1.3128426811E~8, ~0.00232071386754),
(3.25717678708E~7, ~3.64817207342E~8, ~4.76998953311E~4),
(0.00198414349939, ~2.03072003868E~4, 0.609754694474))
Saving image ../outputData/images/StAlbanAnglicanChurch_46_match46_47.bmp
Saving image ../outputData/images/StAlbanAnglicanChurch_47_match46_47.bmp
Residual RMS error of estimated matrix vs. detected inliers: 0.896596246395
Saving image ../outputData/images/StAlbanAnglicanChurch_46_epipolar46_47.bmp
Saving image ../outputData/images/StAlbanAnglicanChurch_47_epipolar46_47.bmp
--- Model House image 0 with itself (768x576) ---:
Matched: 55(Delta20) + 223(Delta10) + 483(Delta5) = 761 correspondences [7.29secs]
RANSAC inliers: 761
Total time spent on the estimation (including matching): [2554.8secs]
The estimated fundamental matrix is:
((1.841199503E~22, ~2.70403374776E~6, ~0.00126933894437),
(2.70403374776E~6, 6.79643811355E~23, ~0.00832868215053),
(0.00126933894437, 0.00832868215053, 7.26198615131E~16))
Saving image ../outputData/images/ModelHouse_0_match0_0_0.bmp
Saving image ../outputData/images/ModelHouse_0_match0_0_1.bmp
Residual RMS error of estimated matrix vs. detected inliers: 8.00109609011E~14
Saving image ../outputData/images/ModelHouse_0_epipolar0_0_0.bmp
Saving image ../outputData/images/ModelHouse_0_epipolar0_0_1.bmp
--- Synthetic Elite Side Winders ---:
Saving image ../outputData/images/EliteSideWinders100_noise0.0_1.bmp
Saving image ../outputData/images/EliteSideWinders100_noise0.0_2.bmp
RANSAC inliers: 100
Residual RMS error of estimated matrix vs. noisy inliers: 1.66432414537E~14
Residual RMS error of estimated matrix vs. true inliers: 1.58661484277E~14
Saving image ../outputData/images/EliteSideWinders100_noise0.0_1_epipolar1_2.bmp
Saving image ../outputData/images/EliteSideWinders100_noise0.0_2_epipolar1_2.bmp
Saving image ../outputData/images/EliteSideWinders100_noise0.25_1.bmp
Saving image ../outputData/images/EliteSideWinders100_noise0.25_2.bmp
RANSAC inliers: 100
Residual RMS error of estimated matrix vs. noisy inliers: 0.19519467848
Residual RMS error of estimated matrix vs. true inliers: 0.0663453669902
Saving image ../outputData/images/EliteSideWinders100_noise0.25_1_epipolar1_2.bmp
Saving image ../outputData/images/EliteSideWinders100_noise0.25_2_epipolar1_2.bmp
Saving image ../outputData/images/EliteSideWinders100_noise0.5_1.bmp
Saving image ../outputData/images/EliteSideWinders100_noise0.5_2.bmp
```

```
RANSAC inliers: 98
Residual RMS error of estimated matrix vs. noisy inliers: 0.403159874321
Residual RMS error of estimated matrix vs. true inliers: 0.158658619801
Saving image ../outputData/images/EliteSideWinders100_noise0.5_1_epipolar1_2.bmp
Saving image ../outputData/images/EliteSideWinders100_noise0.5_2_epipolar1_2.bmp
Saving image ../outputData/images/EliteSideWinders100_noise0.75_1.bmp
Saving image ../outputData/images/EliteSideWinders100_noise0.75_2.bmp
RANSAC inliers: 97
Residual RMS error of estimated matrix vs. noisy inliers: 1.20771251114
Residual RMS error of estimated matrix vs. true inliers: 1.06978771966
Saving image ../outputData/images/EliteSideWinders100_noise0.75_1_epipolar1_2.bmp
Saving image ../outputData/images/EliteSideWinders100_noise0.75_2_epipolar1_2.bmp
Saving image ../outputData/images/EliteSideWinders100_noise1.0_1.bmp
Saving image ../outputData/images/EliteSideWinders100_noise1.0_2.bmp
RANSAC inliers: 91
Residual RMS error of estimated matrix vs. noisy inliers: 0.8384577748
Residual RMS error of estimated matrix vs. true inliers: 0.42724078232
Saving image ../outputData/images/EliteSideWinders100_noise1.0_1_epipolar1_2.bmp
Saving image ../outputData/images/EliteSideWinders100_noise1.0_2_epipolar1_2.bmp
Saving image ../outputData/images/EliteSideWinders20_noise0.0_1.bmp
Saving image ../outputData/images/EliteSideWinders20_noise0.0_2.bmp
RANSAC inliers: 20
Residual RMS error of estimated matrix vs. noisy inliers: 3.95265706429E~14
Residual RMS error of estimated matrix vs. true inliers: 3.95140544932E~14
Saving image ../outputData/images/EliteSideWinders20_noise0.25_1.bmp
Saving image ../outputData/images/EliteSideWinders20_noise0.25_2.bmp
RANSAC inliers: 20
Residual RMS error of estimated matrix vs. noisy inliers: 0.186843690781
Residual RMS error of estimated matrix vs. true inliers: 0.189154095855
Saving image ../outputData/images/EliteSideWinders20_noise0.5_1.bmp
Saving image ../outputData/images/EliteSideWinders20_noise0.5_2.bmp
RANSAC inliers: 20
Residual RMS error of estimated matrix vs. noisy inliers: 0.446935852978
Residual RMS error of estimated matrix vs. true inliers: 0.265604040186
Saving image ../outputData/images/EliteSideWinders20_noise0.75_1.bmp
Saving image ../outputData/images/EliteSideWinders20_noise0.75_2.bmp
RANSAC inliers: 21
Residual RMS error of estimated matrix vs. noisy inliers: 0.909863117919
Residual RMS error of estimated matrix vs. true inliers: 1.36253954615
Saving image ../outputData/images/EliteSideWinders20_noise1.0_1.bmp
Saving image ../outputData/images/EliteSideWinders20_noise1.0_2.bmp
RANSAC inliers: 20
Residual RMS error of estimated matrix vs. noisy inliers: 1.25424390634
Residual RMS error of estimated matrix vs. true inliers: 1.43847725411
```

References

[Anoq09] Ánoq of the Sun. *Image Correspondences for Camera Registration*. Hardcore Processing, 2009.

[BayH06] Herbert Bay, Tinne Tuytelaars, Luc Van Gool. "SURF: Speeded Up Robust Features". 2006.

[C3DI10] Website for *CeX3D Inverse*, Hardcore Processing's upcoming (2010) commercial product: `http://www.cex3d.net/inverse`

[Chum05] O. Chum and J. Matas. "Matching with PROSAC - PROgressive SAmple Consensus". In *IEEE Computer Society Conference of Computer Vision and Pattern Recognition (CVPR 2005), (San Diego, CA), Volume I*, p. 220-226. June 2005.

[CubicP] Derivation of cubic polynomial formula, as seen on the 17th of October 2009 at: `http://mathworld.wolfram.com/CubicFormula.html`

[Cyga09] Boguslaw Cyganek, J. Paul Siebert. *An Introduction to 3D Computer Vision Techniques and Algorithms*. John Wiley & Sons Ltd, 2009.

[Dufo02] Yves Dufournaud, Cordelia Schmid and Radu Horaud, *Image Matching with Scale Adjustment*. Scientific report, INRIA, 2002.

[ElitWi] The *Elite* video game, as seen on the 22nd of November 2009 at: `http://en.wikipedia.org/wiki/Elite_(video_game)`

[Emde08] Helmut van Emden. *Statistics for Terrified Biologists*. Wiley-Blackwell, April 2008.

[Fisc81] Martin A. Fischler and Robert C. Bolles. "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". In *Graphics and Image Processing*. 1981.

[Hart03] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*, Second Edition. Cambridge University Press, 2003.

[Kana98] K. Kanatani. "Geometric Information Criterion for Model Selection". In *International Journal of Computer Vision, 26(3)*, pages 171-189. 1998.

[Lowe04] David G. Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". In *International Journal of Computer Vision, 2004*. January 5, 2004.

[Mata02] J. Matas, O. Chum, M. Urban, T. Pajdla. "Robust Wide Baseline Stereo from Maximally Stable Extremal Regions". In *Proc. British Machine Vision Conference BMVC2002*, 2002.

[Mona99] Pascal Monasse. "Contrast Invariant Registration of Images". In *Proc. of Int. Conf. on Acoustics, Speech and Signal Processing* p. 3221-3224, Phoenix Arizona, 1999.

[MLto07] Online reference for the *MLton* compiler, by Stephen Weeks et al, version 20070826: `http://www.mlton.org`

[Oshe06] Stanley Osher, Nikos Paragios (Editors). *Geometric Level Set Methods in Imaging, Vision and Graphics*. Springer Science+Business Media LLC, 2006.

[Poll00] Marc Pollefeys. *SIGGRAPH 2000 Course Notes 12: Obtaining 3D Models With a Hand-Held Camera*. ACM SIGGRAPH, 2000.

[Poll04] Marc Pollefeys, Luc Van Gool, Maarten Vergauwen, Frank Verbiest, Kurt Cornelis, Jan Tops and Reinhard Koch. "Visual Modelling with a Hand-held Camera". In *International Journal of Computer Vision, Volume 59, Issue 3* pages 207-232. 2004.

[Pres92] William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery. *Numerical Recipes in C. The Art of Scientific Computing, Second Edition*. Cambridge University Press, 1992.

[Torr95] P.H.S. Torr, A. Zisserman and S. Maybank. "Robust Detection of Degeneracy". In *Proc. 5th Int. Conf. on Computer Vision, Boston, MA*, pp. 1037. IEEE Computer Society Press: Los Alamitos CA, 1995.

[Torr96] P.H.S. Torr and D.W. Murray. *The Development and Comparison of Robust Methods for Estimating the Fundamental Matrix*. 1995, revised 1996.

[Trig99] Bill Triggs, Philip McLauchlan, Richard Hartley and Andrew Fitzgibbon. "Bundle Adjustment - A Modern Synthesis". In *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, pages 298-372. 1999.

[Trig01] Bill Triggs. "Joint Feature Distributions for Image Correspondence". In *Proceedings of the 8th International Conference on Computer Vision*, Vancouver, Canada, pages 201-208. 2001.