UNIVERSITÄT OSNABRÜCK

*Bachelor's Thesis*

# Designing and Implementing a Rephotography Application for iOS

Rasmus Diederichsen

**First Supervisor:**    Prof. Dr. Oliver Vornberger
**Second Supervisor:**  Dr. Thomas Wiemann

Department of Computer Science
Department of Cognitive Science

# CONTENTS

## LIST OF TABLES

# LIST OF FIGURES

# 1

## INTRODUCTION

This chapter will introduce the notion of rephotography, elaborate on the process of how to make such a photograph and survey existing approaches to simplify it. These include two applications for mobile operating systems which will be briefly discussed. Furthermore, a summary of more sophisticated work by MIT researchers will be given, leading to the problem statement and the goal of this work.

### 1.1 OVERVIEW

*Rephotography* or repeat photography denotes the retrieval of the precise viewpoint used for taking a—possibly historic—photograph and capturing another image from the same spot, ideally with the same camera parameters. This allows for documentation and visualisation of changes which the scene has undergone between the two or more captures. For instance when documenting urban development, one can present progress of construction, restoration efforts or changes in the surroundings in a visually striking manner, e.g. by blending the photographs together. Figures 1.1 and 1.2 show examples.

When done manually, the photographer must attempt to find the original viewpoint usually by visual inspection of the original image and trying to match the current camera parameters—camera position, camera rotation, focal length, possibly principal point—to the original. The procedure is often carried out by placing the camera on a tripod and comparing a printout of the original image with what can be seen through the viewfinder or the camera screen. The number of parameters to match as well as the difficulty to estimate them purely from comparing two-dimensional images makes the process error-prone and tedious. Visual acuity and experience of the photographer thus place limits on the accuracy with which the camera pose of the reference image can be reconstructed. Some corrections can be done by post-processing the images and warping the rephotograph with a homography to better match the original.

At the time of writing, few computerised aids are available to the photographer (see below). The advancement of mobile phones and tablet computers with integrated cameras and larger screens presents the opportunity to develop applications which can assist in this endeavour, moving away from the traditional trial-and-error approach. On current digital cameras [1] this is impossible due to their closed infrastructure not permitting running user programs.

---

[1] At the time of writing, no commercial manufacturer produces a camera with user-modifiable firm- or software. A project at Stanford (Adams et al., 2010) was discontinued Levoy (2009)

Figure 1.1: Residenzschloss Dresden, destroyed during World War II, © Sergey Larenkov, printed with permission



Figure 1.2: Frauenkirche Dresden, destroyed during World War II, © Sergey Larenkov, printed with permission

## 1.2    PREVIOUS APPROACHES

### 1.2.1    *Mobile Applications*

Two applications have been developed to assist a photographer in taking rephotographs. For smartphone operating systems, *rePhoto*[2] and *Timera*[3] exist, both available for Android and iOS devices. These applications support the user by placing a transparent version of the original image over the current camera image, allowing for easier alignment. The captured rephotograph is then presented together with the original image in a blend (c.f. **??**) which can be customized in *Timera*.

What is characteristical about both of these applications is that the user must still determine on their own how to actually move the camera. An overlay simplifies the procedure, eliminating some of the inaccuracy introduced into the manual approach by the necessity to move the eyes from printout to camera, but it is still the user's responsibility to determine the necessary motion between the current camera position and the goal position (that of the original image).

### 1.2.2    *Computational Re-Photography*

A more sophisticated automated approach was presented in by MIT researchers in 2010. Bae et al. (2010) found in preliminary studies that neither a side-by-side view as would be used in the manual approach, nor a linear blend provided by the above applications result in accurate rephotographs.

In this setup, the relevant parameters of a historic image's camera are reconstructed, including the focal length, principal point and the six degrees of freedom in camera pose. This subsection will give a high-level overview, while a more in-depth discussion of the relevant concepts is deferred until **??**.

The software runs on a laptop connetcted to a digital camera. After reconstructing the scene in 3D by use of two images captured by the user, they are then directed by the software to the desired viewpoint, the user does not have to find it by themselves. On the screen, they are shown the current camera image alongside two arrows indicating in which direction to move—one for movement in the sensor plane and one for movement along the optical axis.

Bae et al. (2010) identify five primary obstacles in viewpoint reconstruction of a historic photograph.

1. The necessary camera motion has six degrees of freedom—three for translation and three for rotation—which are challenging for the user to adjust simultaneously, as changing one parameter will often necessitate adjustments for the others to improve the matching. Furthermore, the number of degrees of freedom

---

2 http://projectrephoto.com/
3 http://www.timera.com/Explore

makes it difficult to communicate to the user how they must move the camera.

2. Computing relative translation between two cameras from corresponding image points is possible only up to an unknown scale (see **??**), meaning it is impossible to determine e.g. if an object viewed by the camera is small and close or large and further away. This poses the problem of how to determine if the user is close to the desired viewpoint and whether or not they have come closer or moved further away over iterations.

3. Relative pose estimation from corresponding points fails when the motion between the two cameras approaches zero, which is the ultimate goal one wishes to achieve. When naïvley comparing the current camera image to the reference photograph, the estimate for relative rotation and translation would become increasingly unreliable as the camera approaches the original viewpoint.

4. Automated computation of relative camera pose will rely on feature detection to find correspondences. However, historical images will often be vastly different from the current scene. Not only may the scene itself have changed considerably, but also the historical image—having been taken by a historical camera—may differ in contrast, sharpness and colours. Feature detectors may not be able to reliably find correspondences when comparing an old with a new photograph.

5. The calibration data—most importantly, focal length and principal point—of the historical camera are often unknown. The calibration data is needed for relative pose computation.

Initially, after loading a historical image, the user is instructed to take two photographs of the scene with a reasonably wide baseline (about 20°). One of them, termed *first frame* is supposed to be taken from some distance from the original viewpoint, the *second frame* should be the user's best eyeballed approximation of it. The wide baseline allows for a more reliable 3D-reconstruction of the scene used to tackle problems 2. and 3.

SIFT features (Lowe, 2004) are computed and matched between the two images. Given these correspondences, 3D coordinates of the points can be computed. A selection of these is reprojected into the second frame after which the user identifies six or more points in the historical photograph corresponding to these points in the second frame. This allows estimating extrinsic and intrinsic camera parameters of the historical camera by running an optimisation algorithm on an initial estimate for relative rotation and translation between first frame and reference image as well as sensor skew, focal length and principal point of the historical camera (problem 5.). The principal point's initial guess is found again with help of the user who identifies three sets of parallel lines in the historical image (see Hartley and Zisserman, 2004, chapter 8.8).

The result is that the location of the reference camera relative to the first camera is known. During the homing process, the current camera frame is compared to the first frame (not the reference frame, avoiding problem 3.), which avoids degeneracy due to the wide baseline. Given the locations of the reference camera and the current frame's camera, each relative to the first frame, one can compute the location of the reference relative to the current frame and thus guide the user in the right direction. Hence, the reference photograph is not needed anymore after this initial step, circumventing problem 4.

During homing, the current camera frame is warped according to the necessary rotation before being shown to the user, allowing them to focus only on the translation (problem 1.). This is possible since for rephotography dealing with structures usually at some distance, the rotation will be small, otherwise the warped image would be unusable.

A remaining problem (2.) is that the scale of the necessary translation is unknown, so that only the direction is known. This poses the question of how to determine whether the user has come closer to the goal or not. It may be feasible to find the original viewpoint nonetheless, if it was possible to determine at least when the user reaches it, but this is impossible without further information. On top of that, it would make for a better user experience if also the distance to the goal could be communicated.

A key observation in this regard is that the actual scale of the translation is irrelevant, it is sufficient that there be a way to make the scale consistent accross iterations. That is, it is unnecessary to know whether the goal is a specific distance aways, if one can ensure that the translations computed one after the other can be somehow meaningfully compared. For this, Bae et. al observe that when triangulating 3D coordinates from corresponding points, their computed distance from the camera (the first frame) is inversely proportional to the distance between the cameras. Therefore, in each iteration, the scale of the world is computed by triangulating correspondences between the first and current frames. The scale is compared to the scale computed in the initial step for the first and second frames. Scaling the current translation vector by the ratio of the two scales makes the length consistent across iterations and decreasing with the distance to the goal.

The results of this method appear to be very successful, but two main drawbacks exist.

- The prototype is not very convenient, as it requires a (laptop) computer and a digital camera to carry out which is impractical for spontaneous rephotography.

- The application is not available to the public, neither in source nor binary form. It is therefore impossible to evaluate adapt for more mobility.

## 1.3  GOALS OF THIS THESIS

This work's objective can thus be summarised as follows.

1. Implement in a prototypal fashion the process from (Bae et al., 2010) for a mobile operating system so it can be run on a smartphone or tablet and direct the user in approximate real-time.

2. Evaluate the approach and attempt to reproduce the results.

For a proof-of-concept application, several simplifying assumptions are made. Firstly, it is assumed that the "historic" photograph is captured with the same camera as the application is run on and that the camera is calibrated. Secondly, no strong visual differences between the reference and current scenes are assumed so that the reference image is accessible to the same feature detection algorithm without the user manually labelling correspondences.

The application targets iOS 8 and current hardware, as image processing is computationally intense, and has been tested on an iPad Air 2.

# 2

## CAMERA GEOMETRY

This chapter will introduce the geometry of image projection, largely following (Hartley and Zisserman, 2004, chapters 6,7) and the geometry of two views (the epipolar geometry, Ma et al. (2003, chapter 5.1)) and how it can be used to recover relative camera position from two images of the same scene. It largely follows.

### 2.1 CAMERA MODELS

The camera model used is the ideal pinhole camera model which postulates several idealised assumptions.

1. The aperture size is infinitely small

2. There are no lens effects (*thin lens* assumption)

3. The angle of view is arbitrarily large

4. All world points projected onto the image plane are in focus, owing to the small aperture

Homogeneous vectors are the elements of projective geometry. They can be obtained from cartesian coordinates by appending a 1-element. All projective entities which differ only by a scalar factor are equivalent, one writes $\mathbf{x} \sim \mathbf{y}$ if $\mathbf{x} = \lambda \mathbf{y}, \lambda \neq 0$. This has the added effect that points at infinity can be represented by vectors whose last coordinate is zero.

Given a camera $C$ whose center of projection is the origin and a point $\mathbf{X}$ in camera-centric coordinates, the central projection of $\mathbf{X}$ onto $C$'s image plane is depicted in a side-view in Figure 2.1. The image plane is virtually moved to the front of the camera, otherwise the image would be mirrored at the principal point as in real cameras. Let $f$ be the focal length, which is the distance of the image plane to the optical centre. If $\mathbf{X} = (X, Y, Z)$, then $x = \left(f\frac{X}{Z}, f\frac{Y}{Z}, f\right)$ by use of the intercept theorem.
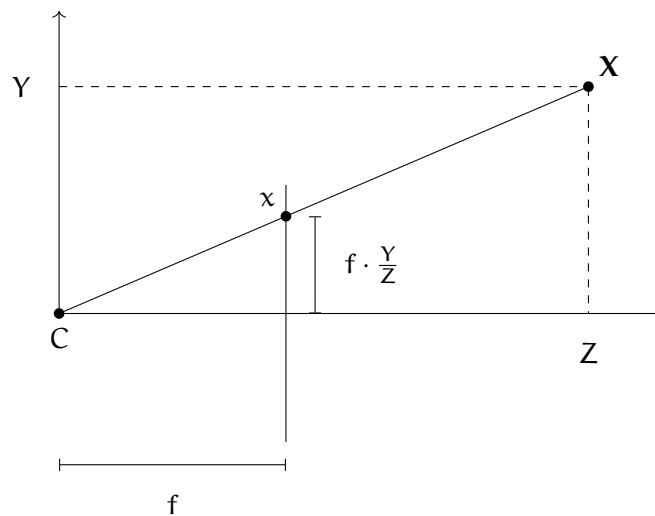


Figure 2.1: Central projection for a pinhole camera

When representing the points as homogeneous quantities, the central projection can be expressed by a matrix multiplication. This can be written with homogeneous coordinates as

$$
\begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \underbrace{\begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{\text{Projection Matrix of C}} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{2.1}
$$

or in short

$$
\mathbf{x} \sim P\mathbf{X} \tag{2.2}
$$

### 2.1.1 Camera Extrinsics

The above situation is a special case wherein the camera centre C defines the origin and the optical and image axes are the coordinate axes. Thus, the rotation and translation of the camera relative to this coordinate system is zero. More generally, there might be a world coordinate frame with different origin and different axes, so that a coordinate transform must be applied to $\mathbf{X}$ before the projection.

Let $R \in \mathbb{R}^{3 \times 3}$ be a rotation matrix giving the camera's rotation relative to the world frame and $t \in \mathbb{R}^{3 \times 1}$ its translation such that

$$
\mathbf{X}_{\text{cam}} = R\mathbf{X}_{\text{world}} + t \tag{2.3}
$$

Then the projection of a point $\mathbf{X}$ in world coordinates onto the image plane becomes

$$
\mathbf{x} = P\mathbf{X} \tag{2.4}
$$

$$
\mathbf{x} = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} [R \mid t]\, \mathbf{X} \tag{2.5}
$$

### 2.1.2 Camera Intrinsics

Above, the resulting image points $\mathbf{x}$ were in *normalised* image coordinates. In particular, the principal point—the intersection of the image plane with the optical axis—was assumed to be $(0,0)$. But generally, image coordinates are expressed in pixels relative to the upper left corner of the sensor. To convert between normalised and pixel coordinates, the camera's five intrinsic parameters and can be written in matrix form and premultiplied in equation (2.5) as

$$
\mathbf{x} = \begin{pmatrix} s_x & s & c_x \\ 0 & s_y & c_y \\ 0 & & 1 \end{pmatrix} \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} [R \mid t]\, \mathbf{X} \tag{2.6}
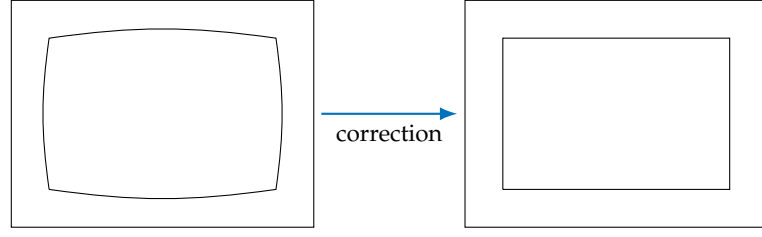$$

Figure 2.2: Radially distorted image on the left, the corrected image on the right.

where $s_x$ and $s_y$ are the focal lengths in x- and y-directions expressed in pixel units per world unit (e.g. cm; $s_x$ and $s_y$ are not necessarily identical, if the sensor has non-square pixels), s the sensor skew (the pixels may not be rectangular; their edges may not be perpendicular) which is usually zero, and the coordinates of the principal point $(c_x, c_y)$ with respect to the origin of the image plane which usually placed at the upper left corner. The intrinsic camera parameters are assembled in

$$K = \begin{pmatrix} fs_x & s & c_x \\ 0 & fs_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \qquad (2.7)$$

are therefore essential to relate world points to image points which will be important for this application. The normalised coordinates $\hat{x}$ for a pixel coordinate $\mathbf{x}$ can be computed as

$$\hat{x} = K^{-1}\mathbf{x}, \qquad (2.8)$$

which will remove the effects of the calibration parameters and thus make the image coordinates independent of the camera's internal characteristics.

In theory, these parameters could be obtained from the camera's vendor who knows the precise manufacturing specifications. In practice, only the focal lengths $f_x, f_y$ are known, in most cases only one with the assumption of square pixels. Usually, the principal point is assumed to be at the sensor centre and the pixels are assumed to be rectangular. In practice however, there are variances introduced by different causes such as imprecise manufacturing or physical impacts which may decentre the lens such that the principal point is no longer at the centre.

A further compilcation is introduced by the camera lens which will often have a non-negligible distortion, most prominently radial distorion as depicted in Figure 2.2. It can be modeled by the application of a distortion factor to the ideal undistorted image coordinates $(\tilde{x}, \tilde{y})$.

$$\begin{pmatrix} x_d \\ y_d \end{pmatrix} = L(r) \begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix} \qquad (2.9)$$

where $L$ is a nonlinear function of the distance $r$ from the distortion center—usually coincident with the principal point. The function can be approximated as an exponential with a Taylor expansion

$$L(r) = 1 + \sum_{i=1}^{k} \kappa_i r^i \tag{2.10}$$

for a given $k$ (see Hartley and Zisserman, 2004, chapter 7.4). The intrinsic camera parameters which consist in the entries of $K$ and distortion coefficients $\kappa_i$ must be determined in order to accurately relate world coordinates to image coordinates. They can be found by calibrating the camera. Different methods exist (e.g Zhang, 2000) but will not be examined here.

## 2.2 EPIPOLAR GEOMETRY

Epipolar geometry is the geometry which relates the image points in two views of the same scene. Figure 2.3 shows the basic setup.

We consider a scene viewed by two cameras with optical centres $c_1$ and $c_2$, where $c_1$ defines the origin, world points $\mathbf{X}_i \in \mathbb{R}^3$, where the subscript denotes the coordinate frame—the first camera, arbitrarily chosen to be the left one, or the second camera—and homogeneous image points $\mathbf{x}_i$ which are the projections of $\mathbf{X}_i$ onto the image planes and thus correspond to the same world point. The cameras are related by a rigid body transform $(R, T)$, where $R$ is a $3 \times 3$ rotation matrix and $T$ the translation between the camera centres. Throughout this work, the direction of coordinate frame transformation will be such that

$$\mathbf{X}_2 = R\mathbf{X}_1 + T. \tag{2.11}$$

It is obvious that the following relation holds,

$$\lambda_i \mathbf{x}_i = \mathbf{X}_i, \ \lambda_i > 0 \tag{2.12}$$

that is, the world point lies on a ray through the optical centre and the image point.

Given the corresponding points $\mathbf{x}_i$ in two images, the ultimate goal is to retrieve the euclidean transform $(R, T)$.

In case the image coordinates for both cameras are normalised (c.f. subsection 2.1.2), they have equal units, so starting from equation (2.11), one can derive

$$
\begin{aligned}
\mathbf{X}_2 &= R\mathbf{X}_1 + T \\
\lambda_2 \mathbf{x}_2 &= R\lambda_1 \mathbf{x}_1 + T && \text{(by equation (2.12))} \\
\lambda_2 \widehat{T}\mathbf{x}_2 &= \widehat{T}R\lambda_1\mathbf{x}_1 + \widehat{T}T && \widehat{T} \in \mathbb{R}^{3\times3} \text{ with } \widehat{T}\mathbf{x} = T \times \mathbf{x} \\
\lambda_2 \mathbf{x}_2^\mathsf{T}\widehat{T}\mathbf{x}_2 &= \mathbf{x}_2^\mathsf{T}\widehat{T}R\lambda_1\mathbf{x}_1 + 0 && T \times T = 0 \\
\lambda_2 \cdot 0 &= \mathbf{x}_2^\mathsf{T}\widehat{T}R\lambda_1\mathbf{x}_1 && \widehat{T}\mathbf{x}_2 \text{ is perpendicular to } \mathbf{x}_2 \\
0 &= \mathbf{x}_2^\mathsf{T}\underbrace{\widehat{T}R}_{E}\mathbf{x}_1 && \tag{2.13}
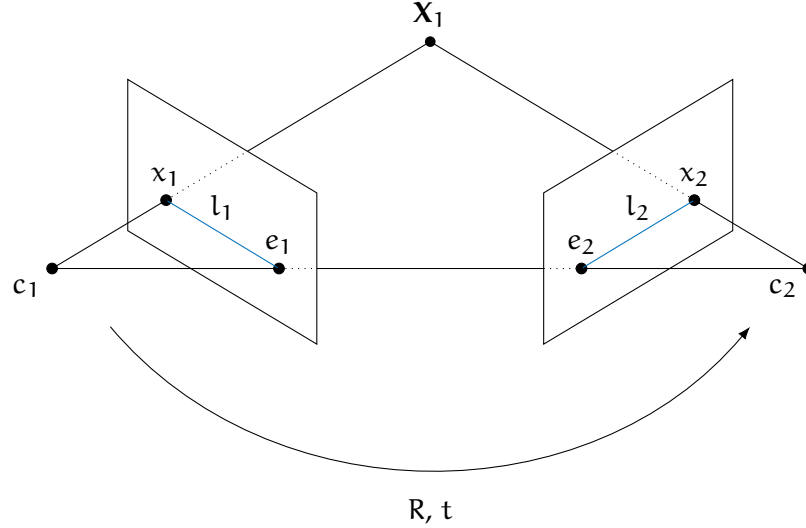\end{aligned}
$$

Figure 2.3: Basic epipolar geometry with camera centres $c_1$, $c_2$, image points $x_1$, $x_2$, a world point $X_1$, epipoles $e_1$, $e_2$ and epipolar lines $l_1$, $l_2$

The product $E = \widehat{T}R$ is the *essential matrix* and the constraint it imposes on corresponding image points the *essential constraint* (see Ma et al., 2003, chapter 5).

An intuition for the essential matrix can be obtained from Figure 2.3. Given an image point in one frame $x_1$, in attempting to find the point $x_2$ corresponding to the same world point $X_1$, the epipolar geometry restricts the search space to one dimension—the epipolar line of $x_1$ in the second camera's image plane. The camera centres and the world point define an epipolar plane. The backprojection of $x_1$ is the ray through $x_1$ and the optical centre $c_1$. All points on this ray are mapped to the same point on the image plane of $c_2$. Depending on how far away the world point $X_1$ is from $c_1$, its image on the second camera's image plane will vary—but it will be on the intersection of the image plane and the epipolar plane, the epipolar line of $x_1$. The line $l_2$ may be identified with its coimage (the orthogonal complement of its preimage) $l_2 = e_2 \times x_2 \in \mathbb{R}^3$ so that

$$\forall x : x \in l_2 \Leftrightarrow x \cdot l_2 = 0. \tag{2.14}$$

The coimage of the epipolar line is the vector perpendicular to the epipolar plane, so every vector in this plane will have an inner product of 0 with this vector. Constrained to vectors in the image planes, this means that all vectors on the epipolar line will have an inner product of 0 with this vector $l_2$. Referring back to equation (2.13), it can be seen that multiplication with $E$ will yield a term $x_2^\mathsf{T} E = l_2$ which fulfils

$$x_1 \cdot l_2 = 0, \tag{2.15}$$

which is precisely the relation stated in equation (2.14). The essential matrix thus maps an image point onto its epipolar line in the other image.

## 2.3 ESSENTIAL MATRIX ESTIMATION

Estimating the essential matrix between two cameras and decomposing it into relative rotation and translation is a necessity in the endeavour to communicate a necessary camera movement to the application's user. The most prominent algorithms in this regard are the 8-Point algorithm introduced in its original form by Longuet-Higgins (1987) and improved upon by Hartley (1997), and the 5-Point algorithm proposed by Nistér (2004). To illustrate the mathematical tractability of the problem, the former will be presented below.

### 2.3.1 *The 8-Point Algorithm*

The essential matrix has nine elements, but since all image coordinates are projective quantities, the set of all essential matrices differing by a constant factor is equivalent so that one degree of freedom is removed and at most eight remain. If one were to formulate constraints on the elements as a system of linear equations, eight of those should be sufficient to uniquely determine an essential matrix. An improvement suggested in (Hartley, 1997) is the preprocessing of the input data (the image coordinates) by translation and scaling. This improves the robustness of the algorithm, but will be omitted here.

For each point correspondence $\{\mathbf{x}_1 = (x_1^i, y_1^i, 1), \mathbf{x}_2 = (x_2^i, y_2^i, 1)\}$, one linear equation

$$\mathbf{x}_2^\mathsf{T} E \mathbf{x}_1 = 0 \tag{2.16}$$

is generated, which can be rewritten as

$$\begin{aligned}
0 = {} & x_2^i x_1^i e_{11} + x_2^i y_1^i e_{12} + x_2^i e_{13} \\
& + y_2^i x_1^i e_{21} + y_2^i y_1^i e_{22} + y_2^i e_{23} \\
& + \quad x_1^i e_{31} \quad + \quad y_1^i e_{32} \quad + \quad e_{33}
\end{aligned} \tag{2.17}$$

Let $\mathbf{e}$ denote the vector of E's entries in row-major order, then

$$0 = \left(x_2^i x_1^i, x_2^i y_1^i, x_2^i, y_2^i x_1^i, y_2^i y_1^i, y_2^i, x_1^i, y_1^i, 1\right) \cdot \mathbf{e} \tag{2.18}$$

If $n$ correspondences are given, they each contribute one row to

$$A\mathbf{e} = \begin{bmatrix} x_2^1 x_1^1 & x_2^1 y_1^1 & x_2^1 & y_2^1 x_1^1 & y_2^1 y_1^1 & y_2^1 & x_1^1 & y_1^1 & 1 \\ \vdots & & & & & & & & \vdots \\ x_2^n x_1^n & x_2^n y_1^n & x_2^n & y_2^n x_1^n & y_2^n y_1^n & y_2^n & x_1^n & y_1^n & 1 \end{bmatrix} \mathbf{e} = 0 \tag{2.19}$$

For eight noise-free point correspondences in general position, there is a unique solution (up to scale) besides the trivial zero, but in practice, one uses more correspondences and the system is overdetermined, so a least-squares-solution minimising $\|A\mathbf{e}\| = \sum_{ij} (A\mathbf{e})_{ij}^2$ is sought. The solution is unique up to scale, since all multiples of $\mathbf{e}$ will satisfy equation (2.19). One therefore introduces the constraint $\|\mathbf{e}\| = 1$ which also excludes a trivial zero solution.

This solution vector is the singular vector with the smallest singular value in the singular value decomposition of $A$ or equivalently, the smallest eigenvector of $A^\mathsf{T}A$ (see Hartley, 1997).

### 2.3.2    *Further Algorithms*

The 8-Point Algorithm is mathematically straightforward and linear, but in practice it suffers from noise (see Luong et al., 1993) and in real applications approaches the robustness of other methods only in its normalised form from (Hartley, 1997) (not related to *normalised image coordinates*).

It has been noted above that $E$ can have at most eight degrees of freedom. In reality, it has only five, as rotation and translation have three degrees of freedom each, and one is lost due to the indeterminable scale. In theory five constraints from five pairs of image points thus are sufficient for finding $E$. A solution was put forth by Nistér (2004), improved in (Stewénius et al., 2006). The algorithm is nonlinear and thus much less easily understood and implemented, involving computing the roots of a ten degree polynomial, and requires only five points, but can also be applied to more. It can be considered a state-of-the-art solution to the relative pose estimation problem; its performance in overdetermined cases in the presence of noise compares favourably to other direct methods requiring six (Pizarro et al., 2003), seven or eight points.

Many other methods exist. Some—like the algorithms described above—find a globally optimal solution in closed form—, while others employ heuristic methods to iteratively approach a local optimum. A review is given in (Zhang, 1998).

Direct methods such as the five-point or eight-point algorithms are frequently used in schemes like RANSAC, which make the estimation more robust to outliers (correspondences which are imprecise or incorrect). For a number of iterations, a hypothesis for $E$ is computed on a minimal number of correspondences and then evauluated on the whole data set. If the inliers in the data far outweigh the outliers, it is probable that a noise-free subsample is selected. The best hypothesis is kept. The simplified algorithm is shown in algorithm 1 (c.f. Hartley and Zisserman, 2004, chapter 4.8) and requires an error measure for $E$.

### 2.4    DECOMPOSING THE ESSENTIAL MATRIX

One step remains to recover the relative camera pose from corresponding points. As per the derivation in section 2.2, the rotation and translation between the two cameras is encoded in $E$. Given an essential Matrix, Hartley and Zisserman (2004, chapter 9.6) show that there are four mathematically valid decompositions of $E$ into $R$ and $T$, corresponding to four distinct geometrical scenarios (see Hartley and Zisserman, 2004, chapter 9.6). Only one of the solutions will place a point $\mathbf{X}_2 = R\mathbf{X}_1 + T$ in front of both cameras, the others cannot be

---

**Data**: $n$ point correspondences
**Result**: a best-fitting essential matrix

1 Let $c_{best} := 0$;
2 **for** $i := 0, i < maxIter$ **do**
3     Select randomly a minimal number of points to estimate $E_i$;
4     Compute error measure for $E_i$ on all $n$ points;
5     Let $C_i$ be the set of point pairs whose error does not exceed $\varepsilon$;
6     **if** $|C_i| > c_{best}$ **then**
7        $c_{best} := |C_i|$;
8        $E_{best} := E_i$;
9     **end**
10 **end**
11 return $E_{best}$

---

**Algorithm 1:** Simplified RANSAC scheme for essential matrix estimation

realised in practice. Triangulating one point from its corresponding image points in the two views will therefore reveal the one correct solution, with the translation scale unkown.

# 3

EVALUATION

The approach has been evaluated on two realistic datasets. The most important questions are whether the direction of the necessary translation is correctly identified and its scale decreasing with distance to the target. For both sets of images, the ground-truth translation between each image and the first frame has been measured with centimetre accuracy, while the ground-truth rotation has been estimated from manually labeled correspondence as it is difficult to measure without the proper instruments. Since for the case of noise-free correspondences in a non-degenerate configuration, relative pose estimation algorithms are mathematically correct, this has been deemed sufficiently accurate to evaluate the procedure. For each image pair, 19–27 correspondences have been labeled, of which the majority is used for pose recovery. For pose recovery, RANSAC is used in conjunction with the five-point solver, a point is considered an inlier for a given essential matrix if its distance to its epipolar line is no more than three pixels. These parameters lead to the majority of points being inliers of the pose recovery, the few outliers can be explained by imprecise labeling.

In both data sets, the translation was mostly in the horizontal direction and along the optical axis; the vertical translation is thus neglected. Similarly, rotation was applied mainly around the vertical axis.

In order to idealise the condition, the reference photograph has been used to fill the role of the second frame for world scale computation. In reality, since the reference location is unknown, the reference world scale is obtained from a position somewhat off.

## 3.1 TRAIN STATION DATA SET

A schematic bird's eye view of the captures is shown in Figure 3.1.
Table 1 summarises the ground truth for the five images.

### 3.1.1 *Scale Estimation*

Figure 3.2 shows how the average distance of points to the first frame's camera varies with the second image used for triangulation. The plot illustrates that—especially at full resolution—the ratio based on feature matching closely mirrors the real value. The difference increases with decreasing image scale, but the the slope of the graphs is quite similar. This shows that indeed with increasing distance to the first frame, the ratio decreases, allowing a deduction as to how close the camera is to the target, at least with respect to previous iterations.
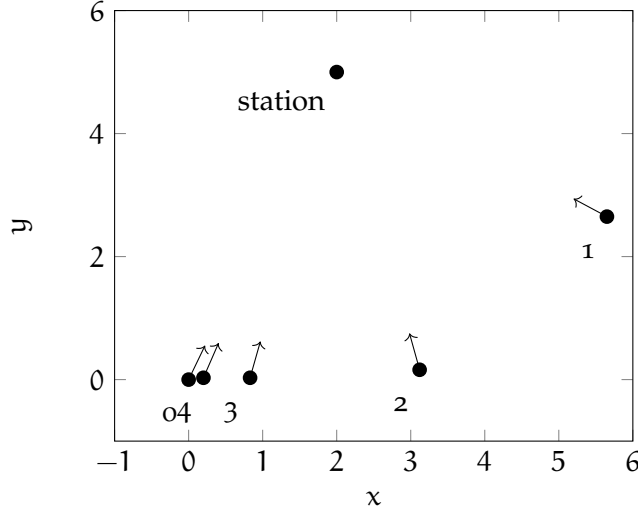
Figure 3.1: Schematic representation of the train data set. Lengths and angles are not precise.

Table 1: Ground truth for the train data. Image 0 is the reference frame, translations and rotations are given as in equation (2.11).

| Image number | Translation to reference $[x, y, z]$ | Rotation to reference $[\theta_x, \theta_y, \theta_z]$ | ratio |
|---|---|---|---|
| 0 | $[0, 0, 0]$ | $[0, 0, 0]$ | 1 |
| 1 | $[.9053, 0, .4246]$ | $[-3.3779, -9.3779, 1.05121]$ | 3.8936 |
| 2 | $[.9986, 0, .0512]$ | $[-1.3274, -5.7134, -.1884]$ | 1.6461 |
| 3 | $[.9993, 0, .0361]$ | $[-1.7156, -2.4761, .3469]$ | 1.0965 |
| 4 | $[.9950, 0, .0995]$ | $[.054606, -4.4867, .2452]$ | 1.0343 |

The situation is somewhat worse at the smallest scale with the ratio between the fifth third and fourth images barely decreasing.

## 3.2 MANOR DATA SET

The schematic positions for the seven manor captures (including reference photograph) are shown in Figure 3.3. In contrast to the train station, there is prominent movement along the optical axis.
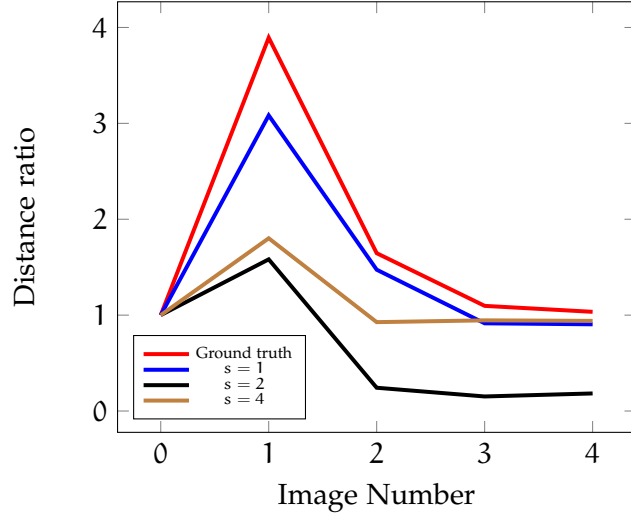
Figure 3.2: Evolution of the distance ratio between images in the Train Station data set with KAZE features
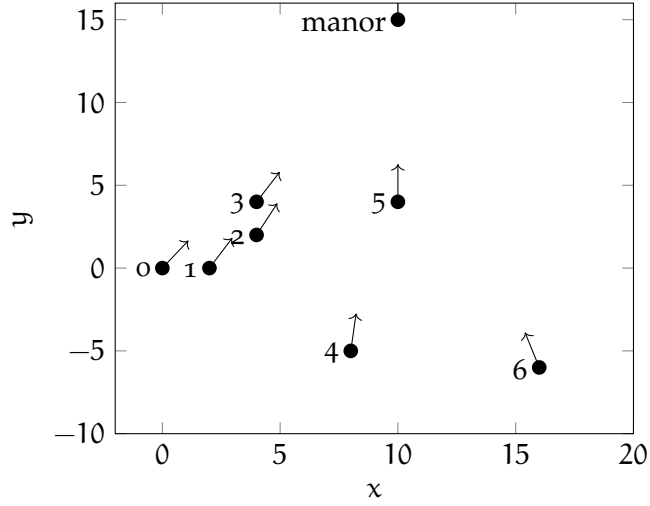


Figure 3.3: Schematic representation of the manor data set. Lengths and angles are not precise.

Table 2: Ground truth for the manor data. Image 0 is the reference frame, translations and rotations are given as in equation (2.11).

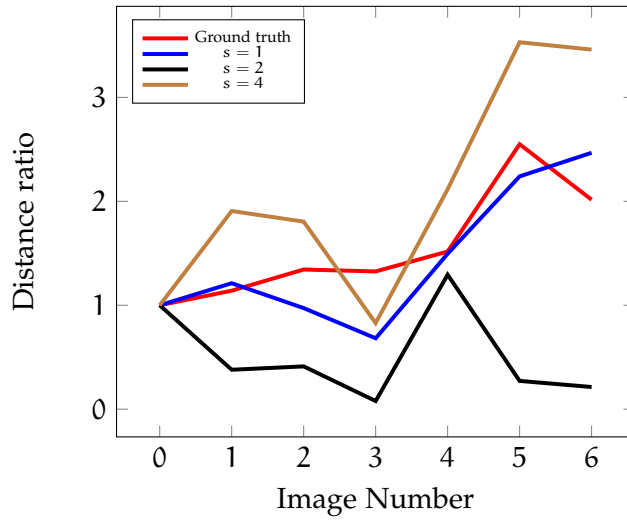| Image number | Translation to reference $[x, y, z]$ | Rotation to reference $[\theta_x, \theta_y, \theta_z]$ | ratio |
|---|---|---|---|
| 0 | $[0, 0, 0]$ | $[0, 0, 0]$ | 1 |
| 1 | $[1, 0, 0]$ | $[-1.7857, -5.4827, 2.1073]$ | 1.1401 |
| 2 | $[0.8944, 0., 0.4472]$ | $[-2.1428, -6.5773, 1.6584]$ | 1.3437 |
| 3 | $[0.7071, 0., 0.7071]$ | $[0.7263, -5.0686, 2.6176]$ | 1.3254 |
| 4 | $[0.8479, 0., -0.5299]$ | $[-1.4146, -10.7998, 2.2250]$ | 1.5168 |
| 5 | $[0.9284, 0., 0.3713]$ | $[-0.1887, -16.6670, 1.2211]$ | 2.5495 |
| 6 | $[0.9363, 0., -0.3511]$ | $[-0.8725, -18.0933, 1.5385]$ | 2.0155 |

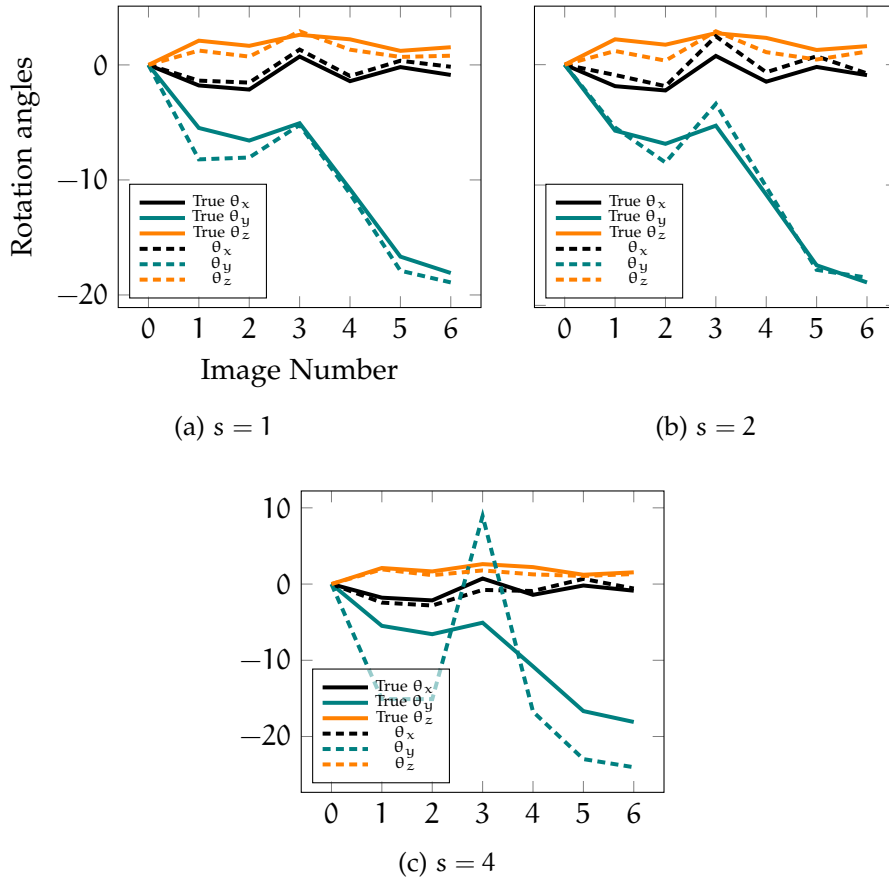Figure 3.4: Evolution of the distance ratio between images in the Manor Station data set with KAZE features



(a) s = 1

(b) s = 2

(c) s = 4

Figure 3.5: Angles of rotation relative to reference with KAZE features on full, quater and sixteenth resolution
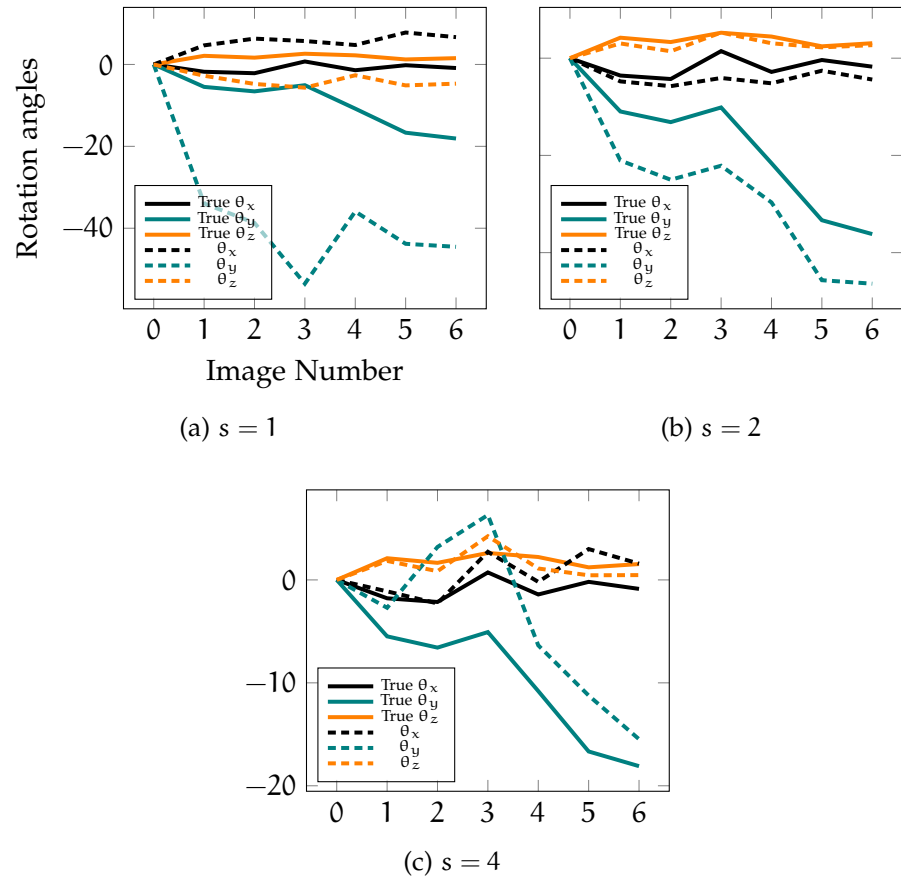
(a) s = 1

(b) s = 2

(c) s = 4

Figure 3.6: Angles of rotation relative to reference with SIFT features on full,
quater and sixteenth resolution

## BIBLIOGRAPHY

Adams, A., Talvala, E.-V., Park, S. H., Jacobs, D. E., Ajdin, B., Gelfand, N., Dolson, J., Vaquero, D., Baek, J., Tico, M., Lensch, H. P. A., Matusik, W., Pulli, K., Horowitz, M., and Levoy, M. (2010). The frankencamera: An experimental platform for computational photography. *ACM Transaction on Graphics*, 29(4):29:1–29:12.

Bae, S., Durand, F., and Agarwala, A. (2010). Computational re-photography. *ACM Transactions on Graphics*, 29(3).

Hartley, R. I. (1997). In defense of the eight-point algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(6):580–593.

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

Levoy, M. (2009). Camera 2.0: New computing platforms for computational photography. `http://graphics.stanford.edu/projects/camera-2.0/`. accessed 2015-07-28.

Longuet-Higgins, H. C. (1987). Readings in computer vision: Issues, problems, principles, and paradigms. chapter A Computer Algorithm for Reconstructing a Scene from Two Projections, pages 61–62. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

Luong, Q.-T., Deriche, R., Faugeras, O., and Papadopoulo, T. (1993). On determining the fundamental matrix : analysis of different methods and experimental results. Research Report RR-1894.

Ma, Y., Soatto, S., Kosecka, J., and Sastry, S. S. (2003). *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer Verlag.

Nistér, D. (2004). An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(6):756–777.

Pizarro, O., Eustice, R., and Singh, H. (2003). Relative pose estimation for instrumented, calibrated imaging platforms. In *Proceedings of Digital Image Computing: Techniques and Applications*, pages 601–612, Sydney, Australia.

Stewénius, H., Engels, C., and Nistér, D. (2006). Recent developments on direct relative orientation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60:284–294.

Zhang, Z. (1998). Determining the epipolar geometry and its uncertainty: A review. *Int. J. Comput. Vision*, 27(2):161–195.

Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334.