Calibration of Images with 3D range scanner data

École Polytechnique Fédérale de Lausanne

Signal Processing Laboratory (LTS4)

Víctor Javier Adalid López

Supervisors: Z. Arican, E. Vural, Prof. P. Frossard

June 2009

Abstract

Víctor Javier Adalid López

Signal Processing Laboratory (LTS4)

Supervisors: Z. Arican, E. Vural, Prof. P. Frossard

June 2009, 66 pages

This project aims to develop methods to calibrate camera images that are arbitrarily positioned in a scene with a 3D range scanner data using feature matching. The way to obtain these results is by using first a planar pattern to obtain the intrinsic parameters of the camera, hence obtaining the rotation and translation by estimating the camera projection matrix from matches of features, estimated via RANSAC to obtain robust result. The testing is done for a set of camera images in some arbitrary positions around the scanner point of view, in order to reflect how the algorithm reacts to the change in the point of view. The results show that it is possible to obtain a satisfactory projection matrix with this approach, though it requires that the images are similar to the scan, with a closer point of view, small deformation of the object shapes in the 3D scan and the scanned surface should have solid colours with similar reflectance both in infrared and visible spectra. Even with these conditions, the amount of matches fitting in the model could be too low for a good estimation.

Keywords: 3D Range Scanner Calibration.

Acknowledgements

Firstly, I would like to thank my thesis supervisors, Zafer Arican and Elif Vural, for their guidance, advice and suggestions.

Secondly, thank my parents, grandparents and the rest of my family, for their support and patience.

Finally, I would like to thank all the friends I made in Lausanne, their companion and friendship made my stay so enjoyable. It has been a pleasure to share all this moments with all of you, and I hope we all could keep in touch in the future.

Contents

Chapter 1

Introduction

3D laser range scanners are used in extraction of the 3D data in a scene. Main application areas are architecture, archeology and city planning. Thought the raw scanner data has a gray scale values, the 3D data can be merged with colour camera image values to get textured 3D model of the scene.

Also these devices are able to take a reliable copy in 3D form objects, with a high level of accuracy. Therefore, they scanned scenes can be used to validate the experimental results achieved with 3D reconstruction algorithms based on multiple view camera images or image sequences.

These applications of the 3D scanners can be achieved via the external calibration of standard cameras with respect to the 3D scanner. For this, multiple images from standard cameras are taken by placing them on the scanner by manual calibration.

This project aims to develop methods to efficiently calibrate the cameras that are arbitrary positioned in a scene with the 3D laser range scanner and register the images with the 3D data.

1.1 Outline of the thesis

In Chapter 2, some basics concepts and algorithms used in this thesis are explained. Firstly some information is given information about the pinhole camera model, as well as about the 3D range scanners, its output data and the visual aspects of this data.

The next section explains the detection of features in pixel maps, the matching of correspondences between two images and the relation between a 2D point and its 3D correspondence. In the last part of the chapter, RANSAC is explained, which is a robust algorithm for model estimation from data possibly contaminated with errors.

Chapter 3 shows the computation models to obtain the camera projection matrix, the intrinsic camera matrix from a calibration pattern and, from the previous two, the rotation and translation matrix between the camera coordinate system and the one from the scanner. Furthermore, is explained the objective method for testing the quality of the results, based in the rotation matrix.

The Chapter 4 describes the parameters used in the testing of the algorithms discussed in Chapters 2 and 3 and presents the results achieved.

Finally, Chapter 5 is dedicated to the conclusions drawn from this thesis and about future work plan on this topic.

1.2 Overview of the Thesis

A solution to the problem of calibration of camera images to 3D range scanner is proposed in this thesis. The calibration is achieved by using the reflection map obtained from the scanner as an usual image. This map is used to detect features in both the scanner map and the camera image and match its correspondences. Once the matches between the image pixels and the 3D coordinates from the scanner are detected, they are used to estimate the projection matrix. In order to prevent the contamination in the estimation originated from erroneous matches the RANSAC method is used in the projection matrix estimation. The implementation is based the Matlab RANSAC Toolbox from Marco Zuliani [9].

To obtain extrinsic parameters like the rotation matrix and the translation vector from the projection matrix, the intrinsic parameters of the camera have to be determined by a separate way. The method used in this thesis is the Zhang's [2] camera calibration with one-dimensional objects. The detection of features use the Scale-Invariant Feature Transform proposed by David Lowe [5], using a modified version of the implementation for Matlab by Andrea Veldaldi from the Department of Computer Science UCLA Vision Lab [10].

Finally, the quality of the projection matrix is tested by two proposed estimators based on the orthogonality of the rotation matrix.

Chapter 2

2D/3D Matching

The aim of this chapter is to introduce the basic concepts about the camera model, the methods for detecting features in camera images and matching them and the 3D scanner data.

The first part of the chapter presents the basic model for the camera images and the 3D scanner particularities. The second part studies the problem of feature detection and matching within the 2D and the 3D images by using existing feature detection algorithm SIFT. The last part introduces of RANSAC algorithm, used to keep only the matches fitting in the model.

2.1 Camera Model

The camera model aims to represent a transformation from 3D world coordinate system to a 2D image plane, introducing the optic effects of the camera. The transformation is usually represented by a 3 x 4 matrix called *Projection Matrix*, which maps homogeneous 3D world coordinates to homogeneous 2D image plane coordinates.

$$\boldsymbol{P} = \begin{pmatrix} P_{11} & P_{12} & \dots & P_{14} \\ P_{21} & \vdots & \ddots & \vdots \\ P_{31} & \dots & \dots & P_{34} \end{pmatrix} \tag{2.1}$$

The projection matrix (2.1) contains information about focal length and principal point, the so called intrinsic parameters, as well as the extrinsic parameters, rotation and translation.

Pinhole Camera Model

The basic pinhole model (Figure 2.1) assumes that 3D points in space are projected onto an image plane in the intersection of this plane with the line linking both the 3D point and the projection center.

The intersection of the line with the plane is the projection point, $\boldsymbol{f}$ is the focal length, $\boldsymbol{P}$ is the principal point, $\boldsymbol{X}$ is a 3D point and $\boldsymbol{x}$ is the projection of $\boldsymbol{X}$. The ray which is perpendicular to the image plane passing through the camera center is called principle axis, and the point of intersection of this ray with the image plane is known as principal point.

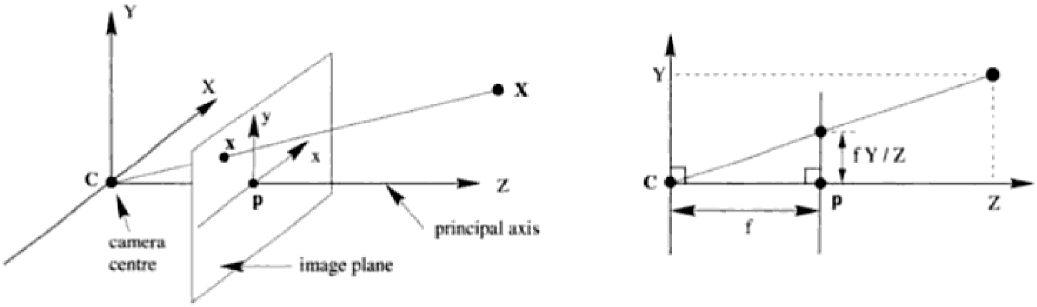


Figure 2.1: Pinhole camera geometry. C is the camera centre and p the principal point. The camera centre is here placed at the coordinate origin. Note the image plane is placed in front of the camera centre.[1]

The 3D point X is projected to the point x. If the system coordinate is $(XYZ)^T$, then the projected coordinates can be calculated as $(fX/Z, fY/Z, f)^T$ The projection matrix will then be

$$P = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.2}$$

Translations in the Image Coordinate System

The previous model (2.2) assumes the centre of the image plane as the origin, however the lower left corner is utilized as the image origin (Figure 2.2). In this case, the matrix is

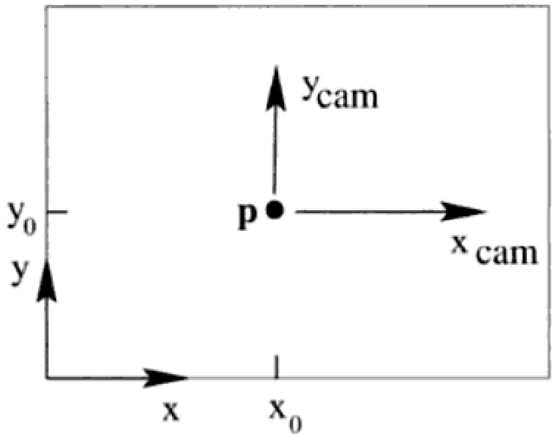


Figure 2.2: Image and camera reference systems. [1]

$$P = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} [I|0] \tag{2.3}$$

$$x = \boldsymbol{K}[\boldsymbol{I}|\mathbf{0}]X \tag{2.4}$$

where K denotes the *camera calibration matrix*.

World Coordinate System Changes

In the current projection matrix, the camera coordinate system is the same as the world coordinate system. The projection matrix should be updated for 3D coordinates measured from an arbitrary system.

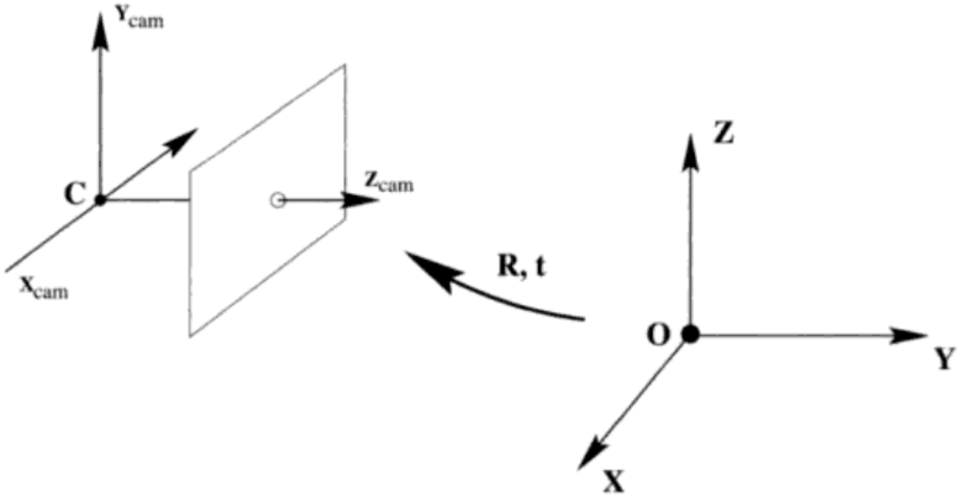


Figure 2.3: The Euclidean transformation between the world and camera coordinate frames.[1]

Figure 2.3 shows the origin of the world coordinate system in an arbitrary position with respect to the camera coordinate system. The relation between the two reference systems are defined by a 3 x 1 translation vector t and a 3 x 3 rotation matrix R in the following way

$$X_{cam} = R(X - C) \; with \; t = -RC$$

therefore (2.4) would be

$$x = K[I|0]X_{cam} \rightarrow x = K[I|0][R|t]X \rightarrow x = K[R|t]X \tag{2.5}$$

and hence

$$x = PX \ with \ P = K[R|t] \tag{2.6}$$

The K matrix contains the intrinsic parameters, the rotation and translation are the extrinsic parameters.

CCD camera distortions

While the pinhole camera model introduced in Section 2.1 is an ideal model for standard cameras, in practice, the CCD camera sensor could present some distortions, for example non-square pixels. Including this distortion the K matrix has the form

$$K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \tag{2.7}$$

There is also the possibility of skew, this is, the pixel shape not being a rectangular parallelogram, but a common parallelogram. That was common in some old cameras, The skew parameter should be 0 for most of the cameras nowadays.

So the final camera matrix would be

$$K = \begin{pmatrix} \alpha_x & s & c_x \\ 0 & \alpha_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \tag{2.8}$$

where

- α_x is the scale factor in the x-coordinate direction
- α_y is the scale factor in the y-coordinate direction
- s is the skew,
- $(x_0, y_0)^T$ are the coordinates of the principal point.

α_x and α_y represent the focal length of the camera in terms of pixel dimensions in the x and y axis respectively.

$$x = PX$$

$$P = K[R|t]$$

Final camera projection matrix has 11 degrees of freedom, 5 intrinsic and 6 extrinsic parameters.

2.2 3D Scanner data

A 3D scanner is a device that captures scenes or objects position in real 3D space, creating an accurate *point cloud* recreation of the shape of the real scene. This has many applications in architecture, industrial design, quality checking, forensics and audiovisual industry among others.

2.2.1 Scanning Hardware

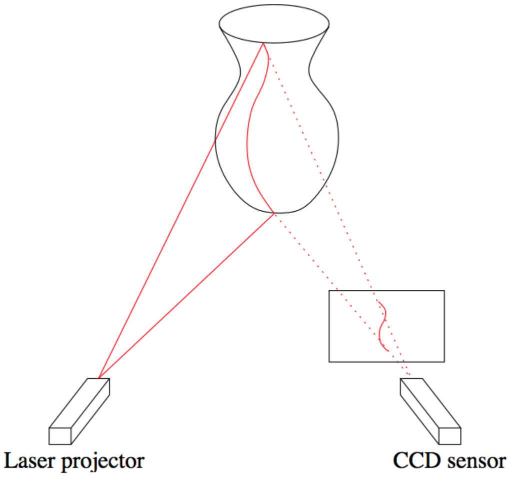


Figure 2.4: Laser scanning [4]

There are different devices commercially available to obtain 3D scans. To build a model, a 3D scanner can be treated as a *black box* that produces a 3D *point cloud*. However, it is necessary to get an idea of the basic physics underlaying in scanners.

The range scanner used in this case is a phase-shift system. This kind of scanners use light at three different phases, measuring the distance to an object by determining the phase shift detected by the sensor. A laser beam is projected onto the scanned space, and a sensor senses the reflected light from the objects (Figure 2.4)

The distance detected by the sensor can be converted to 3D points in the scanner coordinate systems, using the calibrated position and orientation of the mirror angle sensor in the scanner, which determines the θ and ϕ angles needed to determine the actual position of the point.

This kind of scanner has a small error in a range of tens of meters, and its scanning speed is in the order of hundred of thousand points per second, which allows to scan a complete scene in a few minutes.

2.2.2 3D Data

The 3D scanner data is presented in an ASCII file with the data distributed in rows and columns as:

row	column	x	y	z	reflection	reflection	reflection
0	0	-0.70410000	-0.76240000	-0.14410000	105	105	105
0	1	-0.70220000	-0.75940000	-0.14370000	112	112	112
0	2	-0.69980000	-0.75580000	-0.14310000	114	114	114
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Row and column correspond to the map coordinates of the 3D image. This is, like a Mercator projection in a map, each of them would correspond to a pixel in the projection map. Using them as pixel positions and the reflection data as intensity value, a projection of the cloud of points can be obtained. This projection will, indeed, suffer from the same projection problems as a map, changing the shape and form of the objects represented.

(X Y Z) stands for the 3D point, in Cartesian coordinates, given in meters.

The three reflections are the laser RAW reflection. The values are usually not using the complete margin of values, so they have to be normalized with

$$X_{i\,norm} = \frac{X_i - \min(\boldsymbol{X})}{\max \boldsymbol{X}} \tag{2.9}$$

and adjusted to the desired range.

2.2.3 Visual appearance

The 3D scan aspect widely differs from the usual camera image. The reflection value for some material, specially the textiles and other dyed surfaces can have a significant difference at laser wavelength than in the visible spectra.

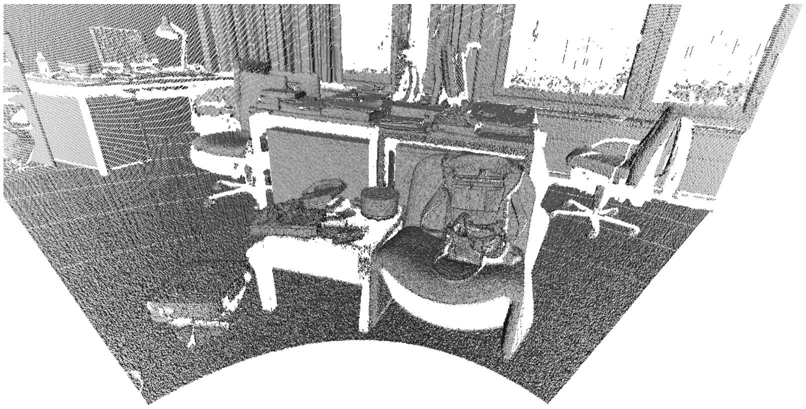


Figure 2.5: The 3D point cloud structure, without reflection data.

As shown in Figure 2.5 the points with small reflection value, too darker or transparent or so far away to get any reflection are discarded by the scanner.

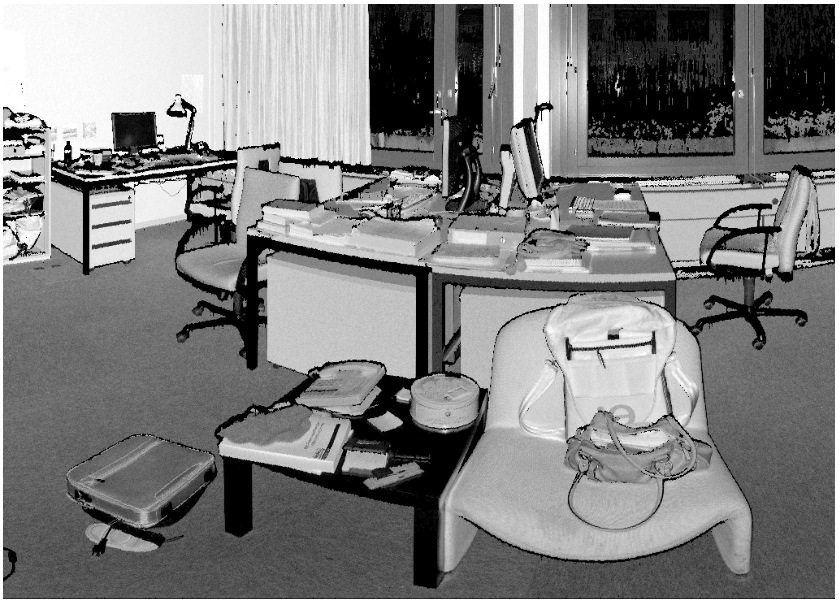


Figure 2.6: The reflection map. The points with no reflection are shown as black points.

The points behind an object cannot be shown, because the scanner requires a direct line-of-sight. Therefore, there is nothing between the scanner and the objects shown. To get an effective scan of the complete object multiple scans have to be performed from different points of view.

Furthermore, can be take for granted that in a scan from one only point of view there will not be occluded points. Hence, all the points are visible from the observation point and can be projected from this point to a plane.

The spherical 360 degrees are scanned following the meridians of the sphere, with the same number of points at the equator as at the poles, resulting a rectangular pixel map, with different reflection values, which can be represented as a picture.

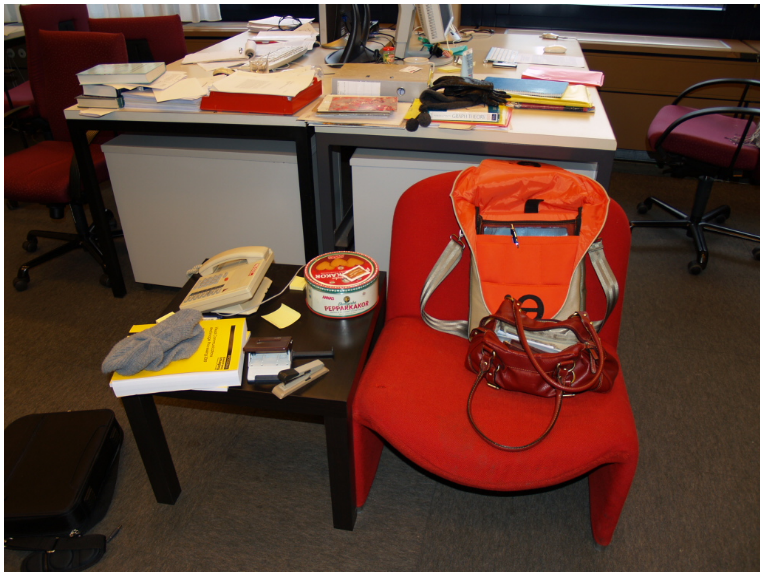


Figure 2.7: A camera picture from the same scene. The gray levels for some objects are quite different in a picture from the 3D reflectance map.

The reflection values are also different in some objects, specially in the dyed ones. Also the image is obtained by a laser illuminating the surface of the objects, being the source at the same point, the image has no shadows, by contrast the camera images can have multiple sources and the image has shadows, changing the illumination and the shapes of the objects.

2.3 Feature detection

One of the aims of the project is to test current automatic features detection algorithms and check how useful are they in detecting and matching image features between 2D pictures and 3D scans. The methods to detect features are explained in this section.

2.3.1 Scale-Invariant Feature Transform (SIFT)

The method used for detecting features is the Scale-Invariant Feature Transform proposed by David Lowe [5]. The objective is to find features invariant to changes in illumination, contrast, rotation in 3D space, distortions and additive noise, in a way that can be easily compared in a database.

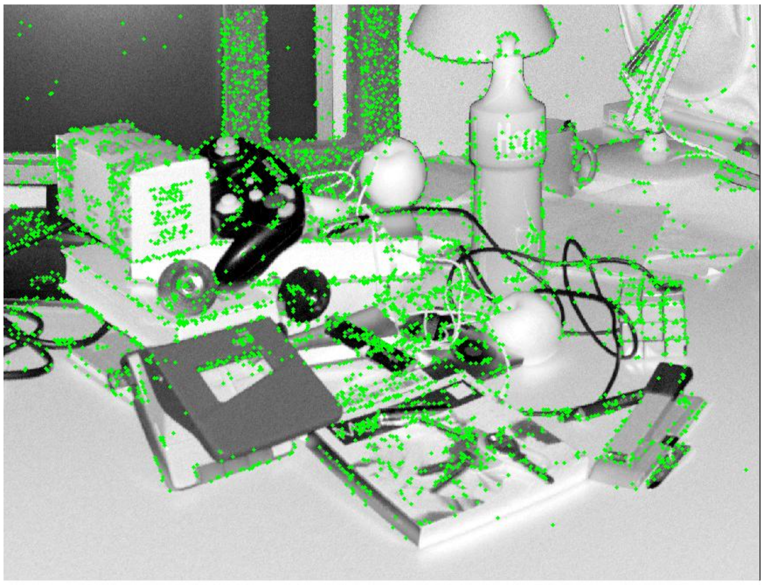


Figure 2.8: Detected features in a 3D map image

This method obtains the features by a difference between different scaled images convolved with the Gaussian 2D functions.

$$D(x, y, \sigma) = L(x, y, k_i\sigma) - L(x, y, k_j\sigma), \tag{2.10}$$

where $L(x, y, k\sigma)$ is the convolution of the image $I(x, y)$ with the 2D Gaussian

$$L(x, y, k\sigma) = G(x, y, k\sigma) * I(x, y), \tag{2.11}$$

and Gaussian blur 2D is

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \tag{2.12}$$

Denoted by k_i, is the level of blur between two images, being their difference two, which is called octave, since the σ value is doubled. Then, the DoG is the difference between two adjacent octaves. Then the image is scaled by a 1.5 factor and then the blur process is repeated, building a kind of *pyramid* (Picture 2.9) with different scales of Gaussian images.

Subsequently are found the maxima and minima in a neighborhood of the points and its condition of maximum for the neighbourhood also in the upper and the lower level of the pyramid. If the condition is maintained, then the point is added to the list of keypoint candidates.

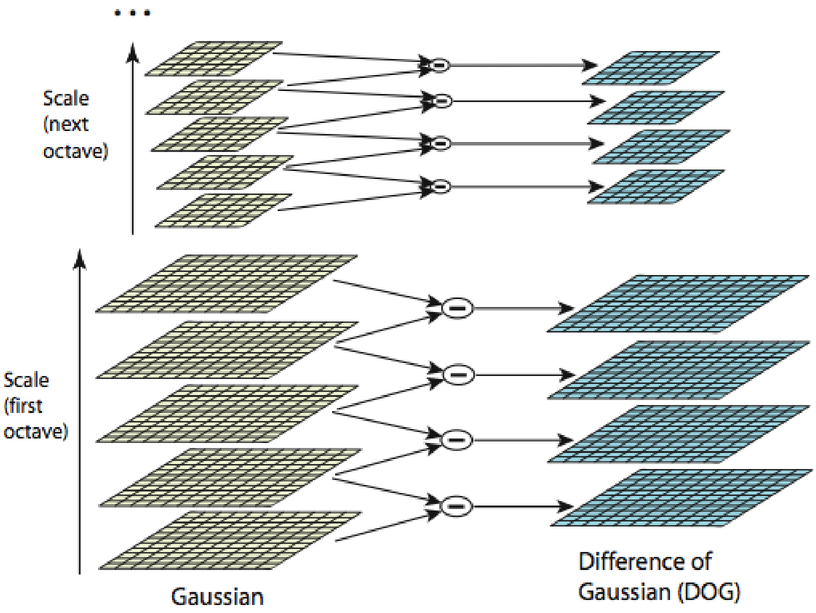


Figure 2.9: For each octave of scale space, the initial image is repeatedly convolved with Gaussians to produce the set of scale space images shown on the left. Adjacent Gaussian images are subtracted to produce the difference-of-Gaussian images on the right. After each octave, the Gaussian image is down-sampled by a factor of 2, and the process repeated. [5]

Stability of the point

To characterize each point the magnitude and orientation of the divergence from the scale space images, using the pixel difference

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \quad (2.13a)$$

$$\theta(x,y) = \tan^{-1} \frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)} \quad (2.13b)$$

Each keypoint gets assigned by a canonical orientation, in a way that the descriptors are invariant to rotation. To keep it as stable as possible to illumination and contrast, the orientation is determined by the peak of an histogram of local image gradient orientations. That histogram has 36 bins to cover the full 360 degrees of rotation and sharped previously to the peak selection. Finally, the keypoint descriptor vectors are filled by the orientation histograms around the keypoints.

2.4 Matching Features

The last stage of feature detection is the matching of the descriptor vector of both images once both images have their keypoints detected. The matching process is a key point in the process, since the quality and the quantity of matches are determinant to the quality of the final estimation. Nevertheless, the quality of the estimation lays utterly in the keypoints descriptor similarities, a big difference in the basics of the image affecting the descriptors drastically decreases the percentage of well fitted correspondences.

2.4.1 Matching

For each keypoint K_1 belonging to the image 1, its match K_2 in image 2 is determined by comparing the keypoint descriptor vector from image 2 to the K_1 descriptor vector. The descriptor in image 2 having the smallest Euclidean distance to the descriptor vector in K_1 is determined as its match K_2.

2.4.2 Filtering

As a result of the matching algorithm, a group of points from the 3D image correspond to the same point in the 2D image 2.10.

These points have to be removed before using the RANSAC method for estimation, since a large number of correspondences incorrectly matched to the same point produces an erroneous projection matrix with a large amount of inliers, and, for the reason that RANSAC chooses the model with a largest number of inliers, the

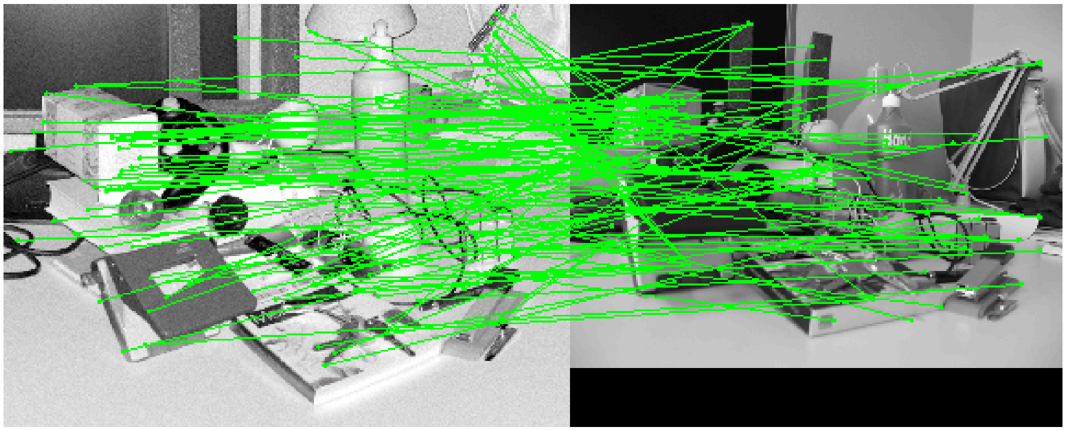


Figure 2.10: As can be seen, some points are incorrectly matched to the same point in the camera image.

algorithm is unable to find the correct model.

Therefore, all the points in the 2D image are compared between themselves and all the matches that share one point are removed, using the next algorithm in pseudo code:

Algorithm to eliminate correspondences with shared points

Input: The image coordinates of the matches Output: The matches list without shared points

```
1. Given $M =$ total number of matches
2. Initialize $i = 0$
3. Repeat while $i < M - 1$
    (a) Initialize $j$ $j = i + 1$
    (b) Repeat while $j < M$
```

```
            i. if point($i$) is equal to point($j$) delete point(j)
           ii. Increment $j$ ($j = j + 1$)
      (c) Increment $i$ ($i = i + 1$)
```

2.4.3 Mapping 3D points

Finally, to relate the obtained points to the actual 3D points from the scan, a matrix of the same size of the map is filled with the pointers to the 3D data. The pairs 2D-3D are now available for the camera projection matrix calculation.

As a resume:

1. Find, for each keypoint from the first image the keypoint in the second image with the closer descriptor (lesser Euclidean distance)

2. If the distance ratio between both keypoints is below a given threshold, both of them are added to the correspondences list.

3. Erase the correspondences sharing any common keypoint

4. Match the 2D correspondence in the 3D map to its 3D coordinates

2.4.4 Random Sample Consensus (RANSAC)

Section 3.2 explains the equations related to the calulation of the camera projection matrix. The details in this section are about the RANSAC algorithm.

RANSAC (Random Sample Consensus) is an algorithm for fitting a model to experimental data containing considerable gross errors [3]. In this method, subsets of the data are selected randomly and the model is tested only with this small subset. The goodness of the model is determined by the best subset consistent with the model, which is saved when the likelihood of finding a better model is lower than a given percentage, or a maximum number of iterations is reached.

When the subset of data is the smallest number required by the model parameters, RANSAC can achieve, if there is any, an uncontaminated solution, even if there is only one. RANSAC can handle data more than half of outliers, if the model could afford it.

For each set of points a camera matrix is estimated, therefore the matrix is tested as shown in equation 2.14. All the 3D keypoints from the 3D image are projected and its distance to the corresponding keypoint from the camera image as $dist(x, \hat{x})$, assuming that the point fits in the model if its distance is below a threshold.

$$\hat{\boldsymbol{P}}\boldsymbol{X} = \hat{\boldsymbol{x}} \tag{2.14}$$

If the number of the inliers is bigger than the previous model, the new model is assumed as the correct one.

Given the random nature of the algorithm, the termination mechanism of the iteration is associated to the number of iterations needed to assure that the probability of finding other better model is below a given value.

Being I the number of inliers and the number of total matches N, the probability of randomly choosing an inlier of all matches is $P_I = I/N$. For a samples set containing m correspondences, the probability of having all inliers is P_I^m. Hence, the probability of having a set free of outliers until the k^{th} iteration is

$$\eta = 1 - (1 - P_I^m)^k \tag{2.15}$$

Therefore, the maximum of iterations for a given probability value is

$$k_{MAX} = \frac{\log(1-p)}{\log(1-\eta^n)} \tag{2.16}$$

The threshold of maximum iterations is then updated each time the model is updated with a better one, until the number of iterations reaches k_{MAX}.

As a resume:

```
Projection Matrix Estimation with RANSAC
Input: A set of correspondences between a 2D and a 3D Image
Output: The projection camera matrix relating the two images

1. Initialize
    (a) Set the number of iterations to 0 ($k = 0$).
    (b) Set the upper bound for the number of iterations to an initial value ($k_{MAX} = M$).
    (c) Set the number of inliers to 0 ($I_{MAX} = 0.$)
2. While ($k < k_{MAX}$)
    (a) Choose a random sample set of six correspondences.
    (b) Compute the projection matrix from the sample set as shown in section 3.2.
    (c) Check the model by calculating the pixel distance for each match and if the error is below some given threshold adding it to the inlier list.
    (d) Update the number of inliers $I$ found in the previous step.
    (e) If ($I > I_{MAX}$)
        i. Update the best model projection matrix.
        ii. Set $I_{MAX} = I$.
        iii. Update $k_{MAX}$ as explained in eq. (2.16).
    (f) Increment the number of iterations ($k = k + 1$).
```

Chapter 3

2D/3D Transform

This chapter describes numerical methods for estimating the camera projection matrix from corresponding 3D-space and image entities. The computation of the camera matrix is known as resectioning.

The chapter is divided into three sections. In the first one, the calculation of the intrinsic matrix is discussed, in the second one is explained the calculation of the projection matrix. The last section is dedicated to the methods to estimate the quality of the calculated projection matrix.

3.1 Intrinsic Matrix Calculation

A flat calibration pattern can be used to obtain the intrinsic matrix. In this study a pattern shaped like a 9 x 7 checkerboard is used.

The calibration procedure follows Zhang's method [2]. The flat pattern is taken by different camera position as in Figure B.1. This method assumes the plane of the pattern as the plane $z = 0$, with the x and y axes parallel to the rows and columns of the checkerboard.

The set of images is used to obtain the parameters using the OpenCV library demonstration for stereo calibration. The intrinsic parameters obtained are in section 4.8

The relation between 3D and 2D points in homogeneous coordinates is

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} r_1 & r_2 & r_3 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}$$

$$= K \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = H \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad (3.1)$$

where $H = [h_1 h_2 h_3]$ and

$$s\hat{m} = H\hat{M},\ \hat{m} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \text{ and } \hat{M} = \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{3.2}$$

A homography can easily be estimated with an image of the model plane. To determine an estimation of H can by determined by minimizing,

$$\zeta = \sum_i ||m_j - \hat{m}_i||^2 \text{ where } \hat{m}_i = \frac{1}{\tilde{h}_3^T M_i} \begin{bmatrix} \tilde{h}_1^T & M_i \\ \tilde{h}_3^T & M_i \end{bmatrix} \tag{3.3}$$

This minimization is performed by using Levenberg-Marquardt method; this method however requires an initial guess. This initial guess can be obtained by the right singular vector of a concatenation of equations obtained by the rearrangement of (3.2)

$$\begin{bmatrix} \tilde{M}^T & 0^T & -u\tilde{M}^T \\ 0^T & \tilde{M}^T & \tilde{M}^T \end{bmatrix} x = 0 \tag{3.4}$$

After finding the homography, the matrix should be decomposed into K, R and t as follows:

In Equation 3.1, one has,

$$\begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix} = \lambda K \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \tag{3.5}$$

And given that the R matrix columns are orthonormal

$$h_1^T K^{-T} K^{-1} h_2 = 0 \tag{3.6}$$

$$h_1^T K^{-T} K^{-1} h_1 = h_2^T K^{-T} K^{-1} h_2 \tag{3.7}$$

Having $B = K^{-T} K^{-1}$ five different parameters, the same strategy used for solving H can be performed to calculate the B matrix parameters. Once the estimation of B is achieved, the parameters for K matrix can be obtained from,

$$v_0 = \frac{B_{12}B_{13} - B_{11}B_{23}}{B_{11}B_{22} - B_{12}^2} \tag{3.8}$$

$$\lambda = B_{33} - [B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23})]/B_{11} \tag{3.9}$$

$$\alpha = \sqrt{\lambda/B_{11}} \tag{3.10}$$

$$\beta = \sqrt{\lambda B_{11}/(B_{11}B_{22} - B_{12}^2)} \tag{3.11}$$

$$\gamma = -B_{12}\alpha^2\beta/\lambda \tag{3.12}$$

$$u_0 = \gamma v_0/\beta - B_{13}\alpha^2/\lambda \tag{3.13}$$

3.2 Projection Matrix Estimation

Provided with a set of matches of 3D points $\{\boldsymbol{X}\}$ and 2D points $\{\mathbf{x}\}$, to estimate the camera matrix (or projection matrix), the equation $x = PX$ will be reformulated as:

$$\begin{bmatrix} \mathbf{0}^T & -w_i\boldsymbol{X}_i^T & v_i\boldsymbol{X}_i^T \\ w_i\boldsymbol{X}_i^T & \mathbf{0}^T & -u_i\boldsymbol{X}_i^T \end{bmatrix} \begin{bmatrix} \boldsymbol{P}^1 \\ \boldsymbol{P}^2 \\ \boldsymbol{P}^3 \end{bmatrix} = \mathbf{0} \tag{3.14}$$

,

where P^{iT} is the i^{th} row of P, and x_i in homogeneous coordinates is

$$\boldsymbol{x}_i = \begin{bmatrix} u_i \\ v_i \\ w_i \end{bmatrix} \tag{3.15}$$

Since the matrix P has 12 entries and 11 degrees of freedom (ignoring scale),it is necessary to have 11 equations to estimate P. Since each point correspondence leads to two equations, at minimum $5\frac{1}{2}$ correspondences are required to solve for P, meaning this that only one of the equations is used for the last point, so only the x or y coordinate from the sixth point is needed, being actually six the points neededÂ [1].

For simplicity, and also due to the presence of noise in the point coordinates the over-determined solution with six correspondence using both equations for the last point is chosen. In this case the matrix is computed from minimizing the algebraic error by using the SVD decomposition.

In order to obtain the homogeneous coordinates, a normalization has to be done:

$$\boldsymbol{x}_n = \boldsymbol{T}_2 \boldsymbol{x}$$

$$\boldsymbol{X}_n = \boldsymbol{T}_3 \boldsymbol{X}$$

Once obtained the normalized points, the P_n matrix is calculated with them and, afterwards, the original P matrix is obtained denormalizing

$$\boldsymbol{P} = \boldsymbol{T}_2^{-1} \boldsymbol{P}_n \boldsymbol{T}_3$$

Where the $\boldsymbol{T}_2$ is

$$\boldsymbol{T}_2 = \begin{pmatrix} \frac{\sqrt{2}}{d_{RMS}} & 0 & -m_x \frac{\sqrt{2}}{d_{RMS}} \\ 0 & \frac{\sqrt{2}}{d_{RMS}} & -m_y \frac{\sqrt{2}}{d_{RMS}} \\ 0 & 0 & 1 \end{pmatrix} \tag{3.16}$$

and $\boldsymbol{T}_3$

$$\boldsymbol{T}_3 = \begin{pmatrix} \frac{\sqrt{3}}{d_{RMS}} & 0 & 0 & -m_x \frac{\sqrt{3}}{d_{RMS}} \\ 0 & \frac{\sqrt{3}}{d_{RMS}} & 0 & -m_y \frac{\sqrt{3}}{d_{RMS}} \\ 0 & 0 & \frac{\sqrt{3}}{d_{RMS}} & -m_z \frac{\sqrt{3}}{d_{RMS}} \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.17}$$

.

The centroid of the 2D and 3D points in the sample are moved to the origin by a translation, and then the points are scaled to make the RMS distance to the centroid 2 for 2D points, and 3 for 3D points.
and d_{RMS} is

$$d_{RMS} = \sqrt{\frac{1}{N}\left(\sum_{i=1}^{N} d^2(x_i, mean(x))\right)} \tag{3.18}$$

3.3 Evaluation of the quality of projection matrices

In order to test the quality of the $\boldsymbol{P}$ matrix, two error estimation functions are defined. The rotation transposed matrix is defined $\boldsymbol{R}^T = \boldsymbol{R}^{-1}$, therefore the rotation matrix is an orthogonal matrix, it is $\boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{I}$. For that reason the $\boldsymbol{A}$ matrix is defined as

$$\boldsymbol{A} = \boldsymbol{R}^T\boldsymbol{R} \tag{3.19}$$

The way to check the quality of the estimated $\boldsymbol{R}$ matrix then is the distance or the $\boldsymbol{A}$ matrix to the rotation matrix.

In order to normalize the resulting $\boldsymbol{A}$ matrix, it is decomposed to the eigenvalues

$$\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{-1} \tag{3.20}$$

And then the $\boldsymbol{A}$ matrix is normalized by the maximum of the eigenvalues

$$\boldsymbol{B} = \boldsymbol{A}/\max\boldsymbol{\Lambda} \tag{3.21}$$

Finally the error estimators used would be the difference between the $\boldsymbol{B}$ matrix and the identity matrix

$$norm(\boldsymbol{B},\boldsymbol{I}) = ||\boldsymbol{B} - \boldsymbol{I}|| \tag{3.22}$$

and the Frobenius norm.

$$norm_{FRO}(\boldsymbol{B},\boldsymbol{I}) = \sqrt{\sum diag\left((\boldsymbol{B} - \boldsymbol{I})(\boldsymbol{B} - \boldsymbol{I})\right)} \tag{3.23}$$

The result of these two estimators is the error, that should be closer to zero for the best results, and also both of them should give a very similar result for a good estimation.

Chapter 4

Experimental Results

In this chapter the parameters for the evaluation of the algorithmsare defined and the obatined results are explained.

The first section shows the parameter values used in the testing and which impact they have in the results. The second section describe the results and their implications of the studied model usefulness.

4.1 Testing Parameters

The camera pictures used for the testing are all of 1024 x 768, given that the intrinsic matrix values are affected by the image size. The camera used is an Olympus E-500 with a sensor CCD surface of 17.3 x 13 mm^2, and the focal length is fixed at 14 mm for all images.

$$f_x = \frac{f}{\Delta_x} \tag{4.1}$$

$$f_y = \frac{f}{\Delta_y} \tag{4.2}$$

$$\Delta_x = \frac{d_x}{w} \tag{4.3}$$

$$\Delta_y = \frac{d_y}{h} \tag{4.4}$$

where w is the image wide resolution, h the height resolution, f is the focal length and d_x and d_y are the wide and height of the CCD sensor. The center of the images is then

$$c_x = \frac{w}{2} \tag{4.5}$$

$$c_y = \frac{h}{2} \tag{4.6}$$

The theoretical intrinsic matrix for these parameters, using the equations from 4.1 to 4.6

$$K = \begin{pmatrix} 828.67 & 0 & 512 \\ 0 & 828.67 & 384 \\ 0 & 0 & 1 \end{pmatrix} \tag{4.7}$$

The intrinsic matrix calculated from the image set from Appendix B.1 by using the method explained in section 3.1 is

$$K = \begin{pmatrix} 792.24 & 0 & 498.51 \\ 0 & 792.24 & 397.217 \\ 0 & 0 & 1 \end{pmatrix} \tag{4.8}$$

The result for this calculation is relatively closer to the theoretical one, giving credibility to the result.

The chosen parameter for the equation (2.16) is $\eta = 90\%$, that means that the probability of RANSAC finding the best model is a 90%. This value is chosen in order to get the results in an acceptable period of time, specially for the images with lesser similarity with the scanner, taken from a distant position, which are expected to be worst than the closer ones. The pixel distance threshold for the RANSAC algorithm is 0.8 pixels. This parameter is chosen to get an acceptable amount of inliers.

The SIFT matching threshold for the SIFT matching algorithm is, indeed, 1.7, a little superior to the default one, 1.5, to assure a good number of matches.For the SIFT features detection and matching is used the SIFT Matlab implementation version 0.9.17 created by Andrea Vedaldi from the UCLA Vision Lab - Department of Computer Science [10].

The presented scene has been arranged in order to avoid materials with a substantial disparity within the laser reflection values and the camera gray levels, as well as keeping the scanned area closer to the scanner level and small enough to get small deformation in the map.

These parameters apply to all the results except if specified otherwise.

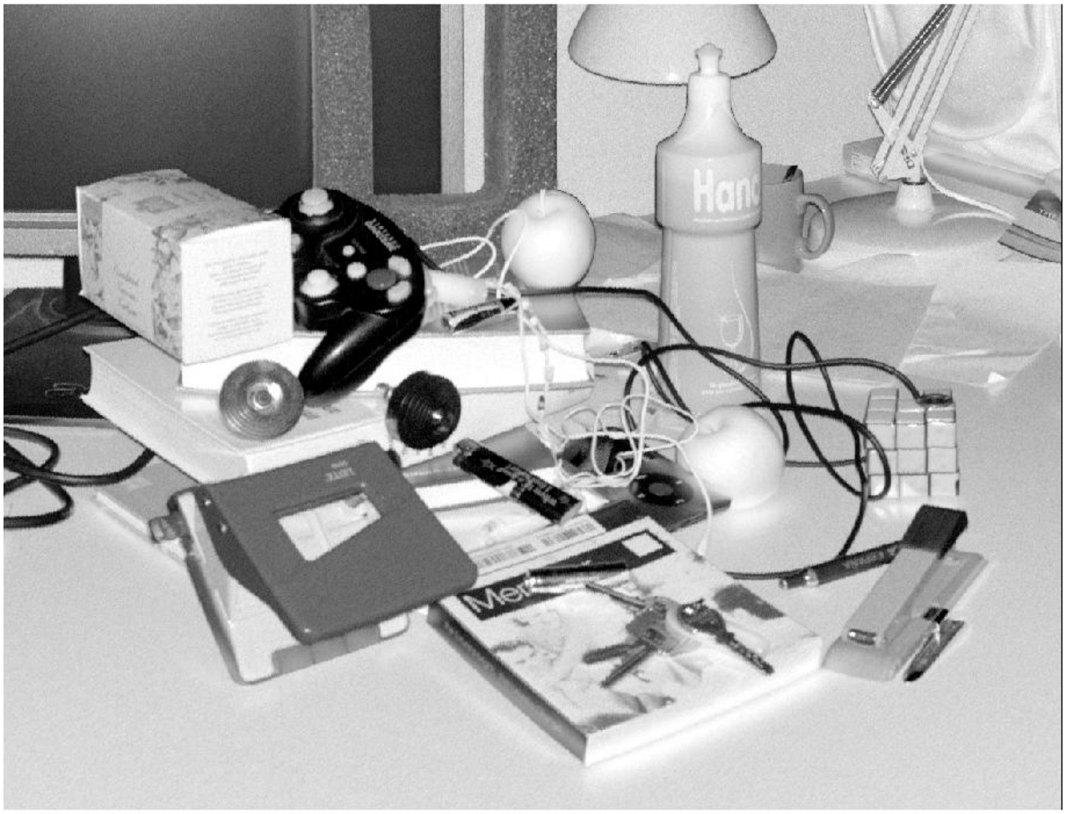


Figure 4.1: 3D scene used in the testing (except specified otherwise)

4.2 Results

The results (Table A.1), as expected, are better for the images with significant resemblance with the 3D map, with a higher number of model inliers and a lower error than images with rotation, partial sights, etc.

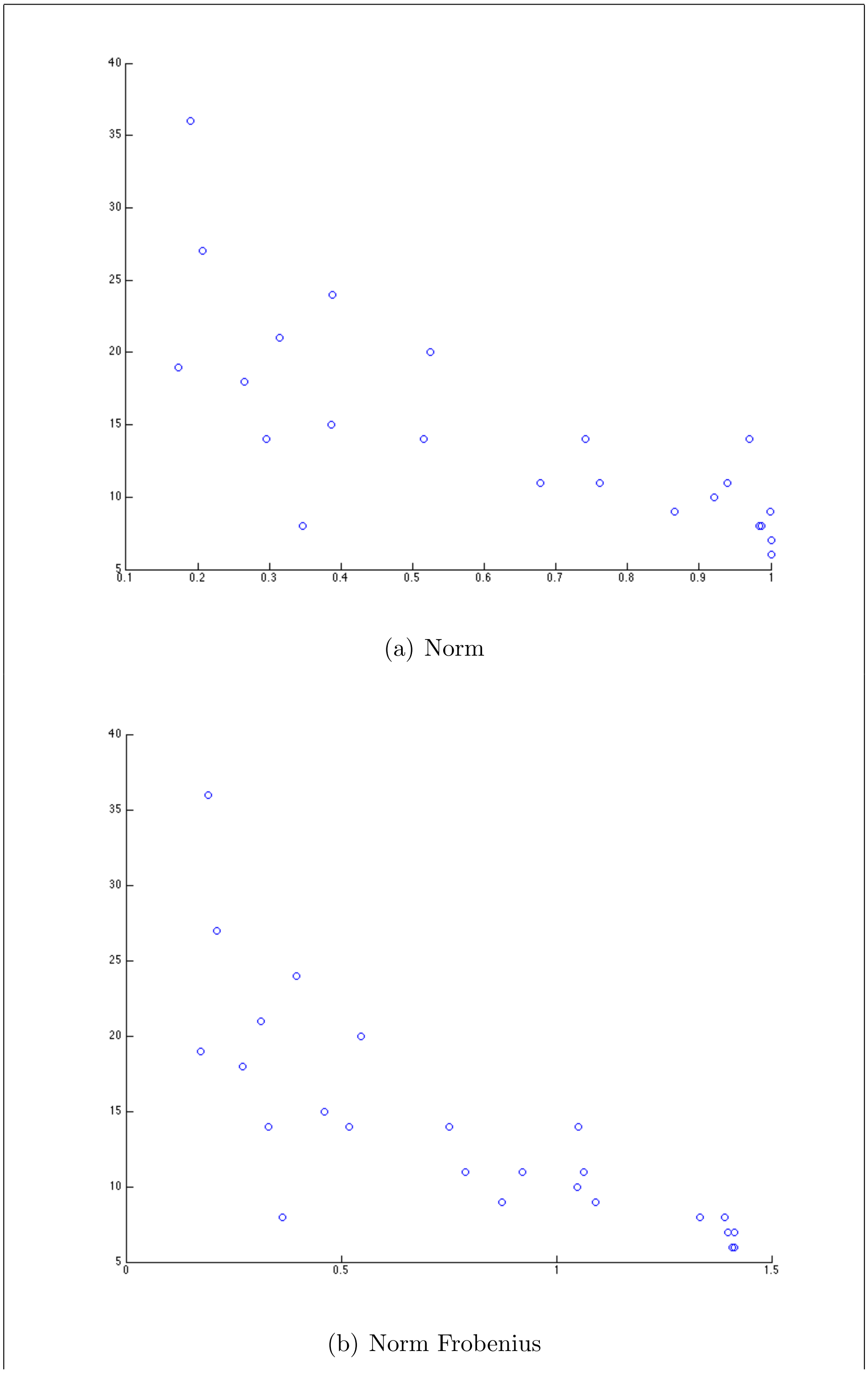
The norm and Frobenius norm columns refers to the two error estimation previously mentioned in Chapter 3. Inliers alludes to the number of inliers fits in the model for each picture. The column matches are the total amount of correspondences found after the feature matching process, and the percentage refers to the percentage of matches that are also inliers.

As can be seen in Figure 4.2, the norm error is lower for the results with a large amount of inliers. These graphics shows the The error is lower for the pictures with a higher number of inliers of the model, as well the result for both norms are more similar in these cases. By opposition, the pictures with a small amount of inliers usually have the worst results.

Independently of the quality of the image, the majority of the inliers are located in high contrast changes in the image, especially in characters.

For a more general image like Figure C.12, with a higher distortion and objects changing their reflection, the number of matches decreases, resulting in a large amount of erroneous inliers.

A reduced percentage of correct correspondences also has his impact in the processing time, since the termination of the RANSAC algorithm is determined by the number of inliers detected. This puts the processing Matlab time in a Dual Core Intel Xeon CPU at 2.66 GHz with 8 RAM GiB in the order of hours, around five or six hours in the best cases and even longer for the worst ones.



(a) Norm

(b) Norm Frobenius

Figure 4.2: Norm and Norm Frobenius versus number of inliers for all images tested.

Chapter 5

Conclusions

After the testing, there is some contrast in results. Event though some results are correct and the algorithm has proven to be able to get an acceptable projection matrix, these results are only achieved from views with great similarity to the 3D scan. With a partial view or a small rotation around the scanner point of view the quality falls to an unacceptable margin. Also, the distortion in a general image, difficulties the correct matching.

The number of matches between a 2D and a 3D images is slight and the major part of the matches are erroneous. This leads to a reduced number of inliers, compromising the quality of the estimation.

Anyway, even with these problems, it is possible to obtain acceptable results with this method if the image is quite close to the scanner point of view, with small distortion.

5.1 Future work

To improve the quality of the results some different approaches can be suggested for future work.

One of them, that was considered in the first stages of this study is to project the 3D points to a planer by a *virtual pinhole camera*, that is, as explained in the pinhole camera model section. That possibility was turned down for some reasons related to this approach:

Related to the fact of having a cloud of points, a projection could leave holes in the image within the points, affecting the quality of the image in an unaffordable way for the feature detector to get results. Related to this, the holes or the points can be projected to the same pixel, therefore it is not trivial to determine which feature will correspond to which point in the 3D cloud.

Though this problem could solved, there is still the problem of deciding which is the correct view to be projected in order to get a similar image to the camera images in a full 3D scene. Even a simple partial scan as used in this project is still problematic to focus the points in an arbitrary location, having only the 3D position of the points in Cartesian coordinates.

With the aim of keeping the solution simple, this possibility was deprecated. A future new study should have to find a solution to the related problems in order to get this approach.

Another possibility could be the calibration using a close to scanner image and a part of a complete 3D scan following this method, and then using the calibration between images to calibrate the full set of images, and then use the calibration of that picture to found the projection of every image to a 3D scan.

This method is probably simpler, but does not improve the method described in this thesis for the 3D calibration, it is just an improve the quality of the matching of the 2D images. That would require to have enough correspondences between images to be sure that all of them will be calibrated.

Appendix A

Table of Results

The table includes the following data:

- Number of the tested image
- Error of the reprojection based on the norm
- Error of the reprojection based on the Frobenius norm
- Number of inlier matches in the image
- Number of total correspondences
- Percentage of correspondences

The room images is from a different 3D scene to all the other results.

Picture number	Norm	Frobenius Norm	Viewpoint	Inliers	Matches	%
45	0.5252	0.5453	Similar	20	192	10.4
46	0.3140	0.3147	Similar	21	195	10.8
47	0.3869	0.4617	Similar	15	159	9.4
48	0.8651	0.8746	Partial view	9	122	7.4
49	0.9708	1.0512	Partial, upper	14	126	11.1
50	0.9999	1.0920	Partial	9	116	7.8
51	0.9208	1.0497	Partial	10	135	7.4
52	0.3464	0.3649	Partial	8	144	5.5
54	0.9875	1.3905	Upward	8	90	8.89
55	0.1728	0.1734	Similar	19	172	11.0
56	0.2648	0.2704	Similar	18	169	10.7
59	0.7606	0.7879	Leftward	11	141	7.8
60	0.2963	0.3302	Rightward	14	144	9.7
61	0.5153	0.5193	Downward, rightward, closer	14	120	11.7
62	0.2065	0.2122	Similar,leftwards	27	199	13.6
63	1.0000	1.4098	Upward, rightward	6	170	3.5
64	0.1904	0.1908	Slightly rightwards, closer	36	221	16.3
65	0.6778	0.9208	Upwards	11	142	7.8
67	0.3886	0.3958	Similar, slightly upwards	24	112	21.4
69	0.7405	0.7512	Leftward, upward, closer	14	126	11.1
71	1.0000	1.3991	Upward, slightly rightward	7	88	8.0
72	1.0000	1.4141	Upwards	6	68	8.8
73	0.9398	1.0641	Downward	11	97	11.3
74	0.9843	1.3330	Downward	8	87	9.2
room	1.0000	1.4138	Partial, 3D deformed at borders	7	31	22.6

Table A.1: The results in this table include the number of inliers and the error for the rotation matrix

Appendix B

Calibration Pattern Images

This is the set of images used to obtain the intrinsic matrix. The calibration pattern consist in a 9 x 7 checkerboard. The method for calculate the intrinsic matrix is explained in section 3.1.

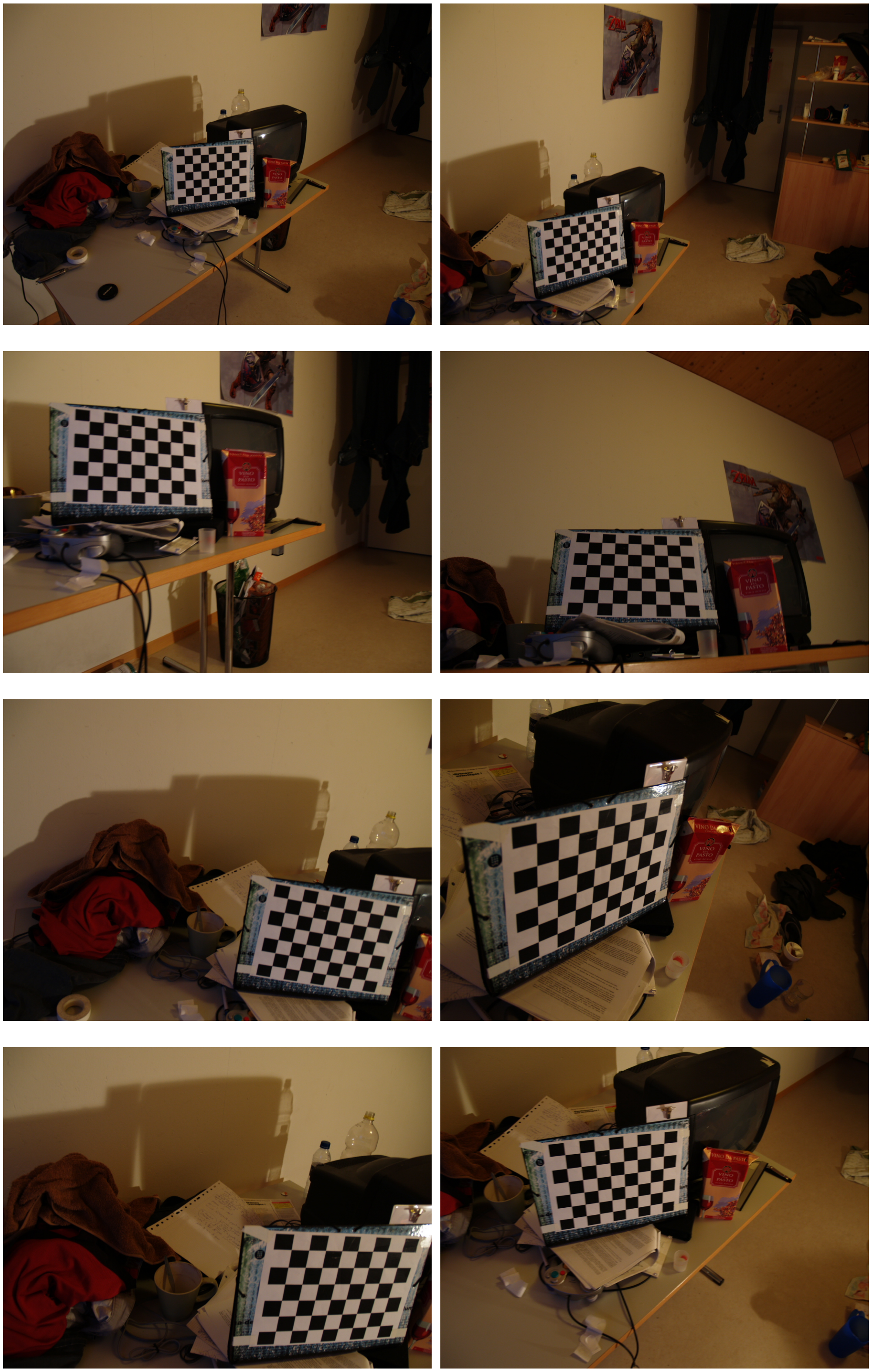
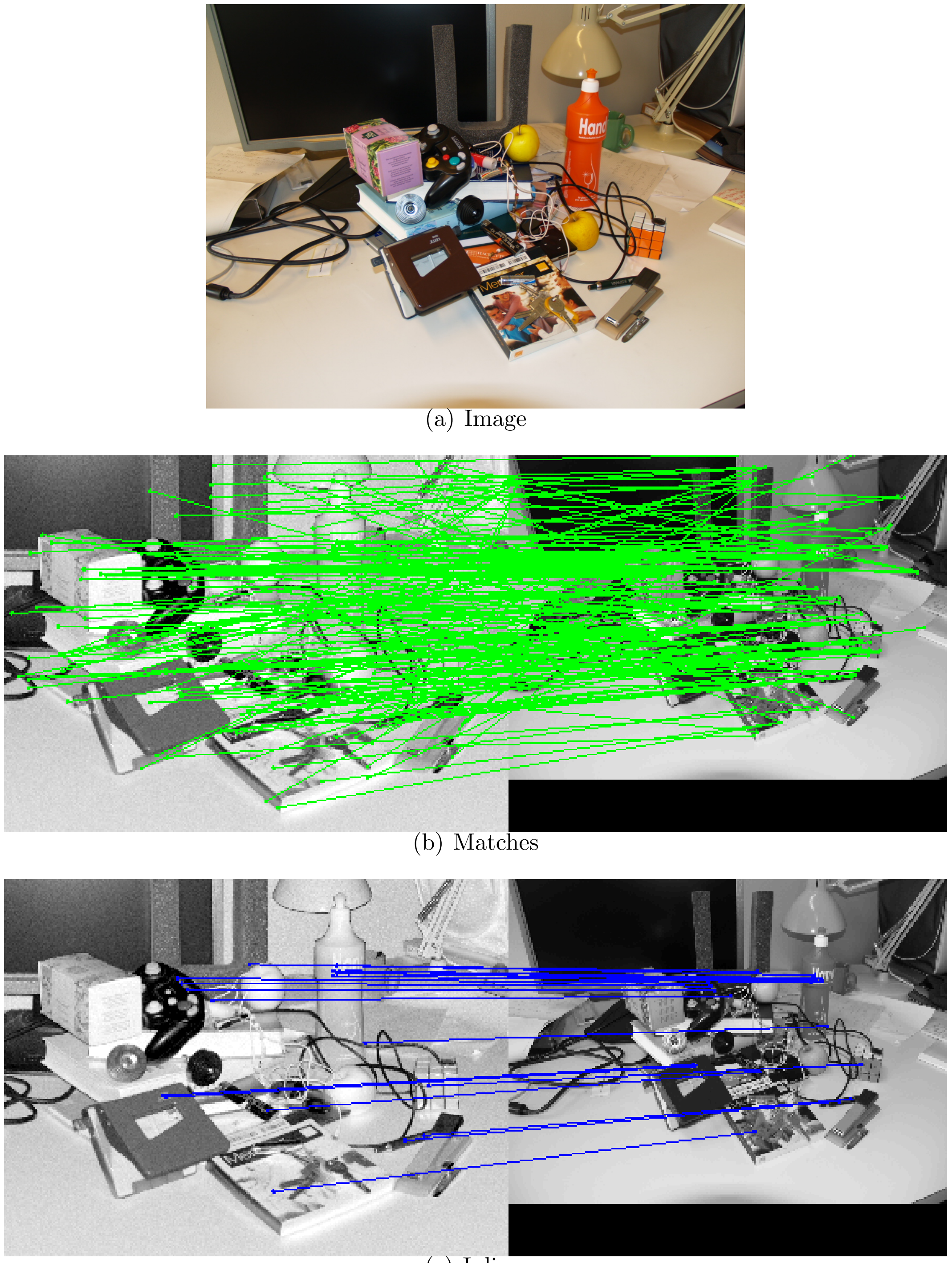


Figure B.1: Set of images with the calibration patern

Appendix C

Images

This appendix contains some representative examples of the obtained results. These figures show the original picture, the result of the matching of features and the inliers detected with RANSAC.

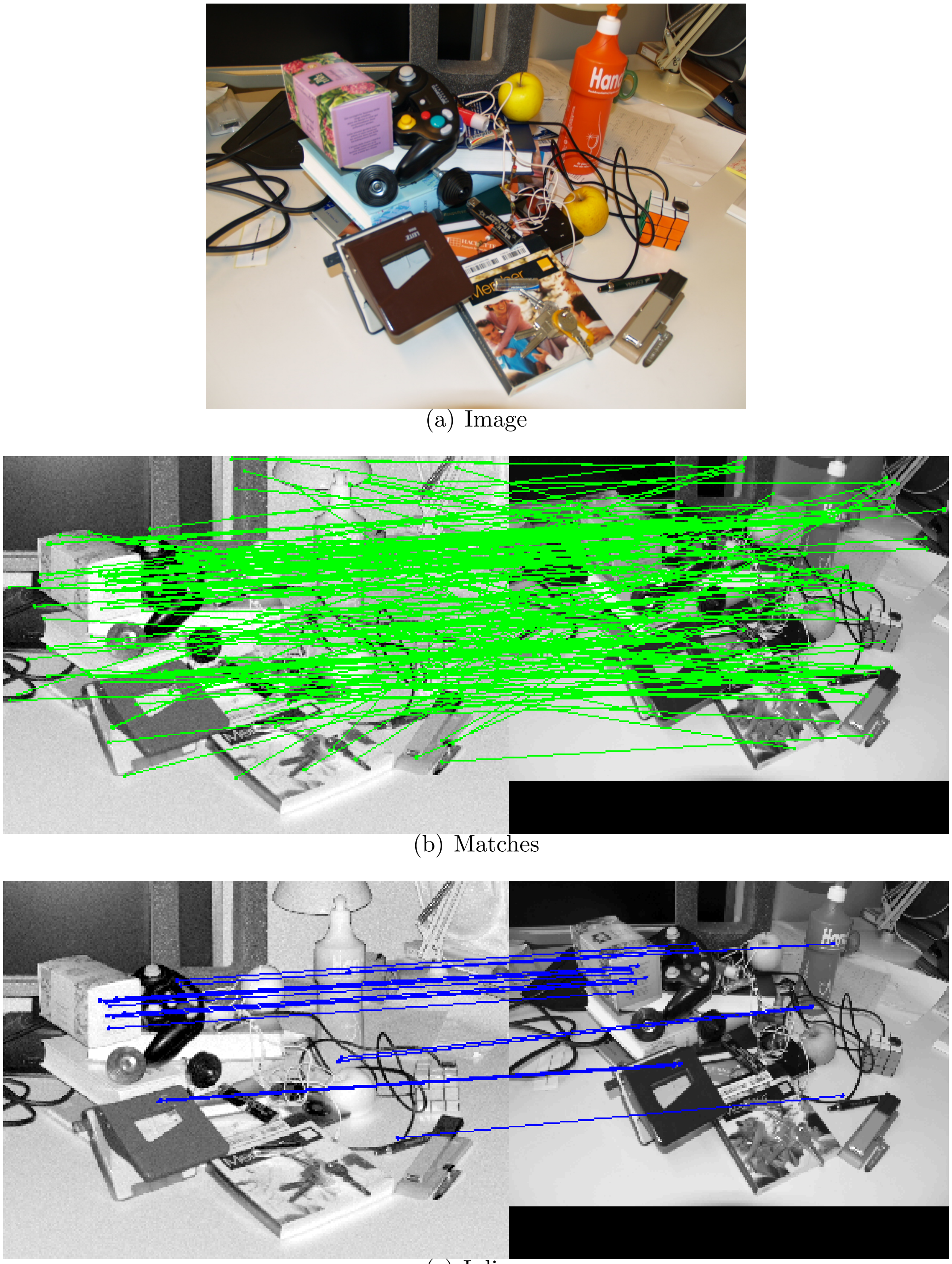


(a) Image

(b) Matches

(c) Inliers

Figure C.1: Image 45. Even not having one of the lowest errors, the number of inliers is high.

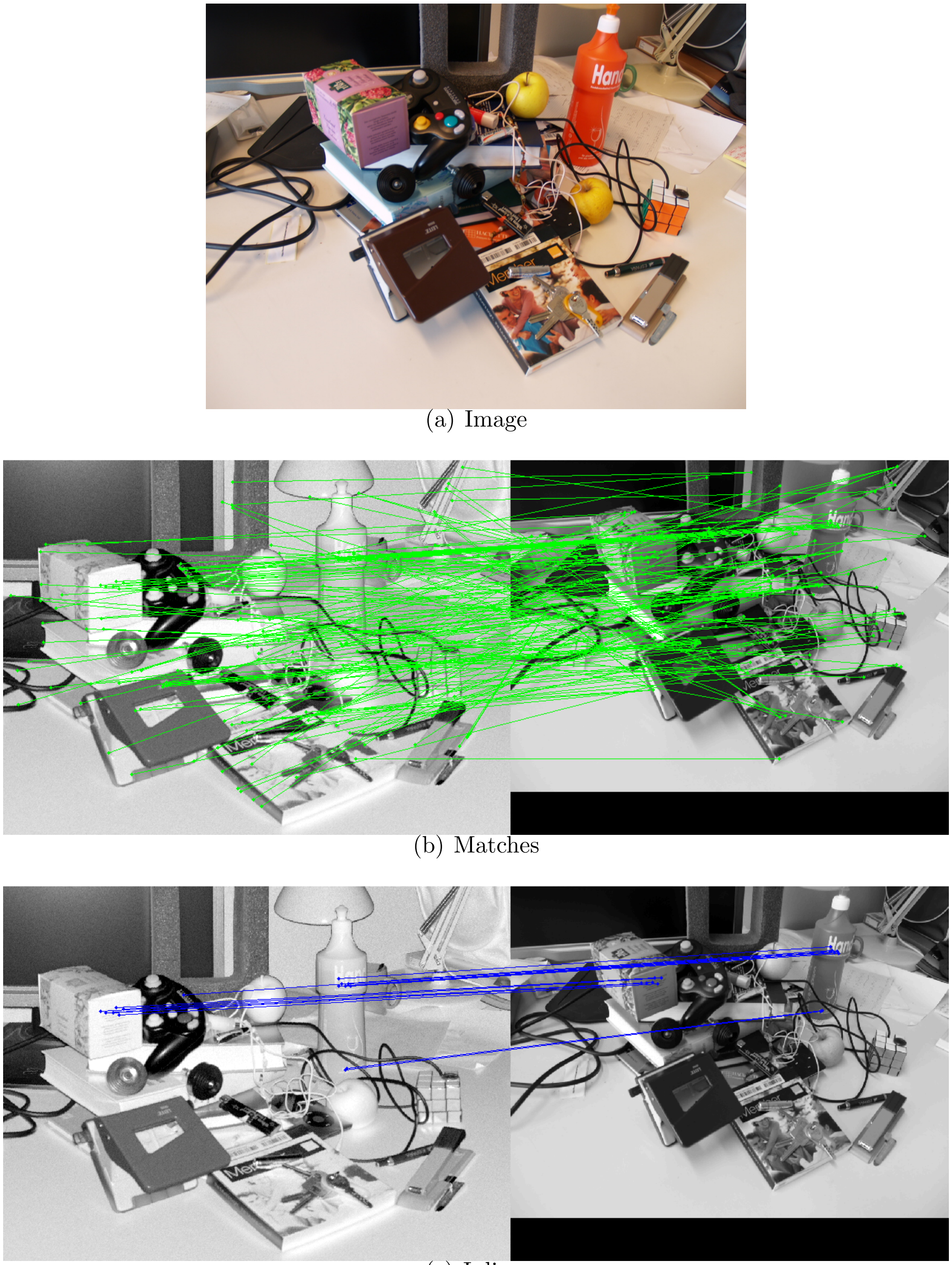


(a) Image

(b) Matches

(c) Inliers

Figure C.2: Image 46. With a similar point of view to the scanner, the results are satisfactory.

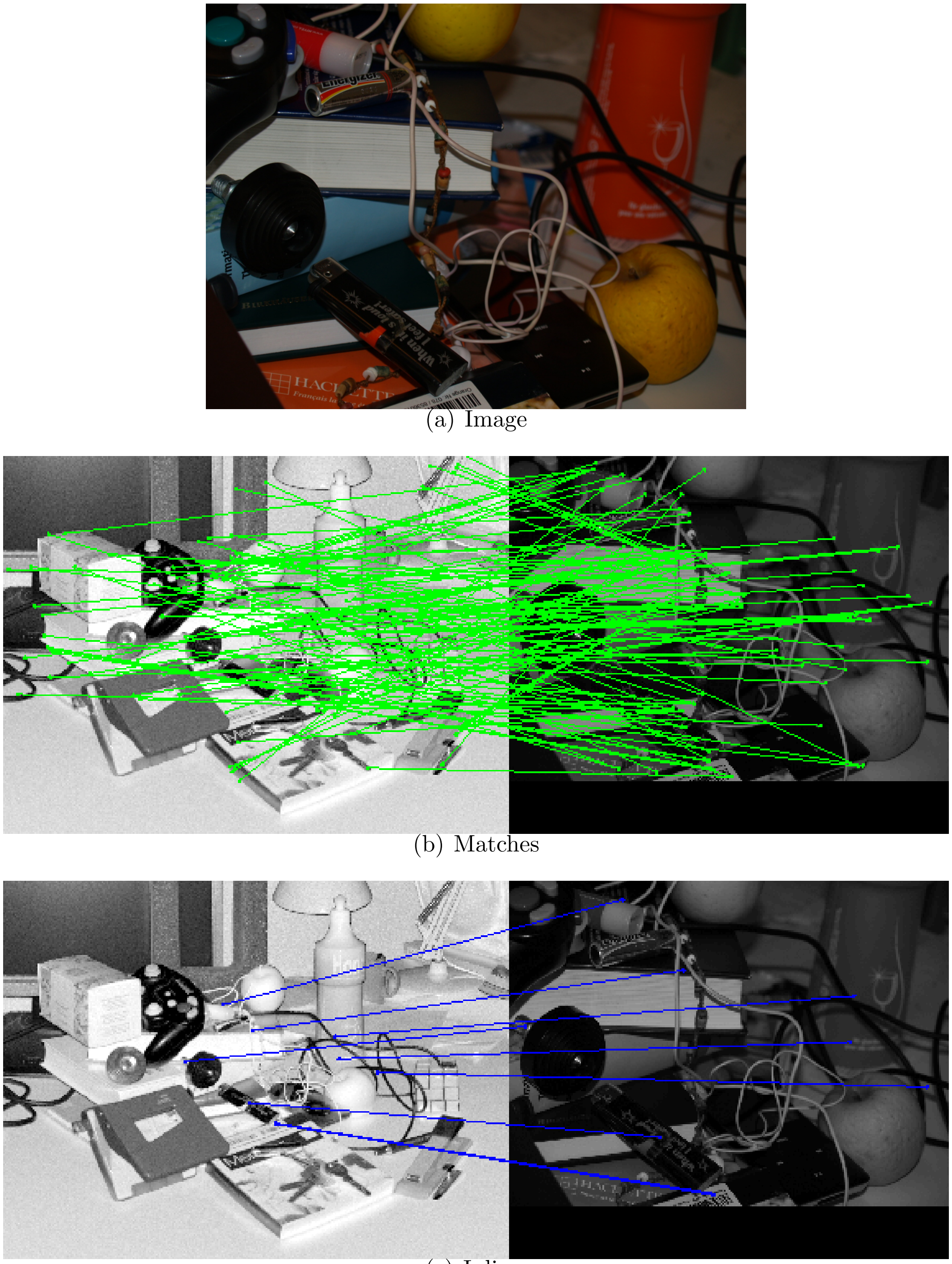


(a) Image

(b) Matches

(c) Inliers

Figure C.3: Image 47. An image similar to the 3D map, despite of being too close between thwm, the number of inliers is good.

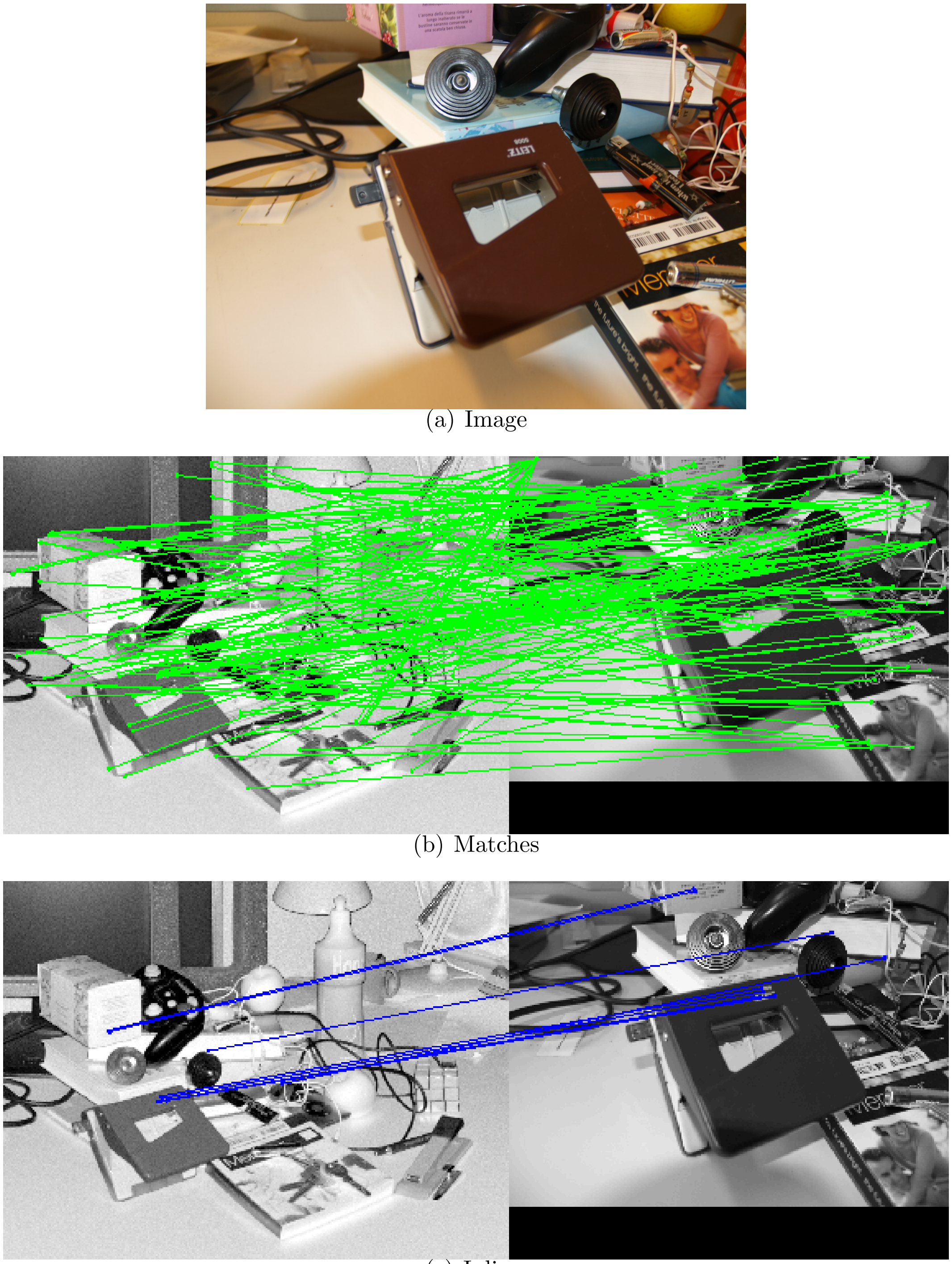


(a) Image

(b) Matches

(c) Inliers

Figure C.4: Image 48. This is an example for a partial view. There are just a few inliers and the resulting error is high.

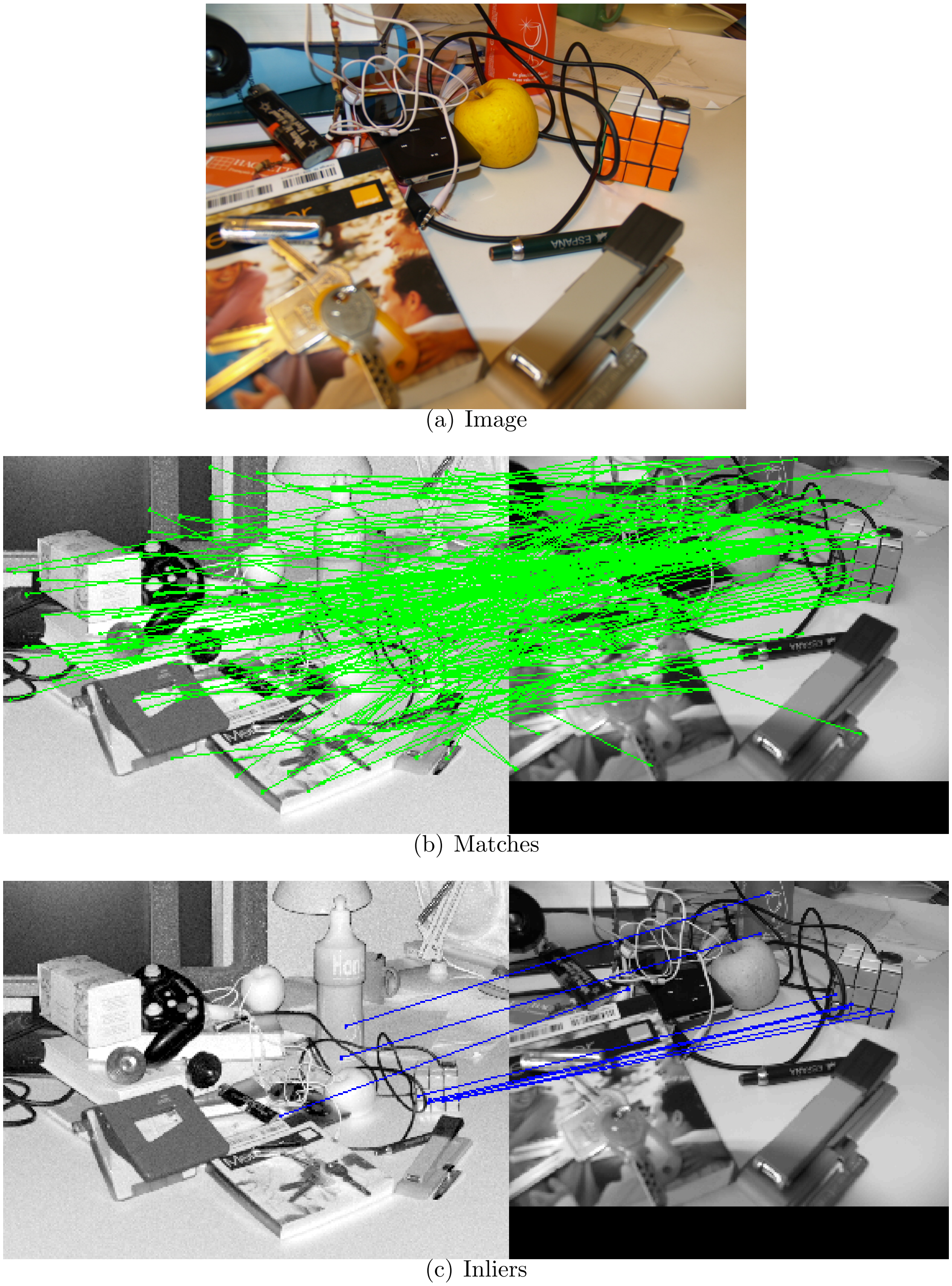


(a) Image

(b) Matches

(c) Inliers

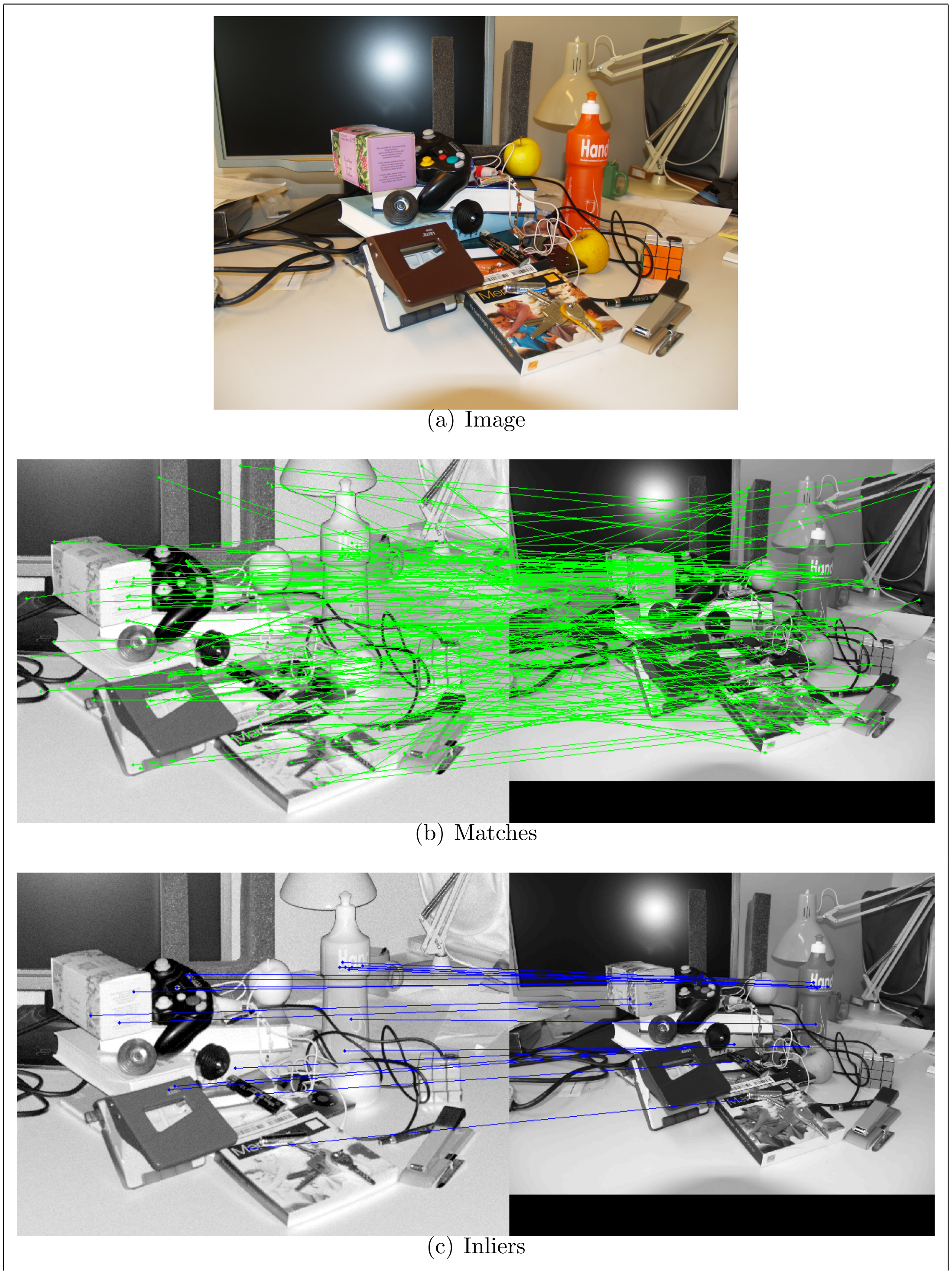
Figure C.5: Image 51. This is another example for a partial view. There are just a few inliers and the resulting error is high.


(a) Image

(b) Matches

(c) Inliers

Figure C.6: Image 52. This is another example for a partial view. Even with a few inliers and the result error is low. This case is an exceptional one.

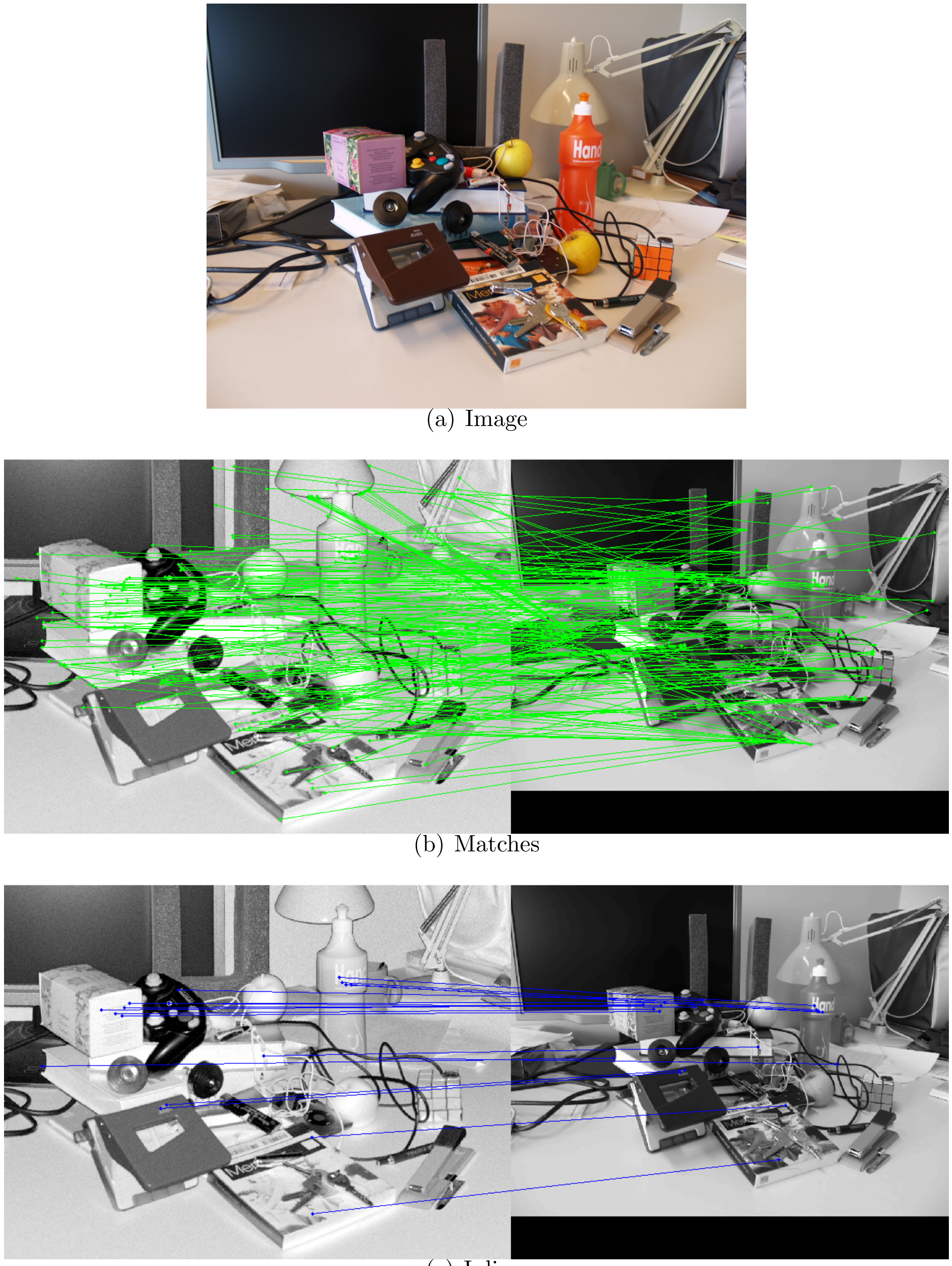


(a) Image

(b) Matches

(c) Inliers

Figure C.7: Image 55. This is a more general image. Since it is not the 3D image, there is no problem of distortion and the results are acceptable, with a fair amount of inliers.

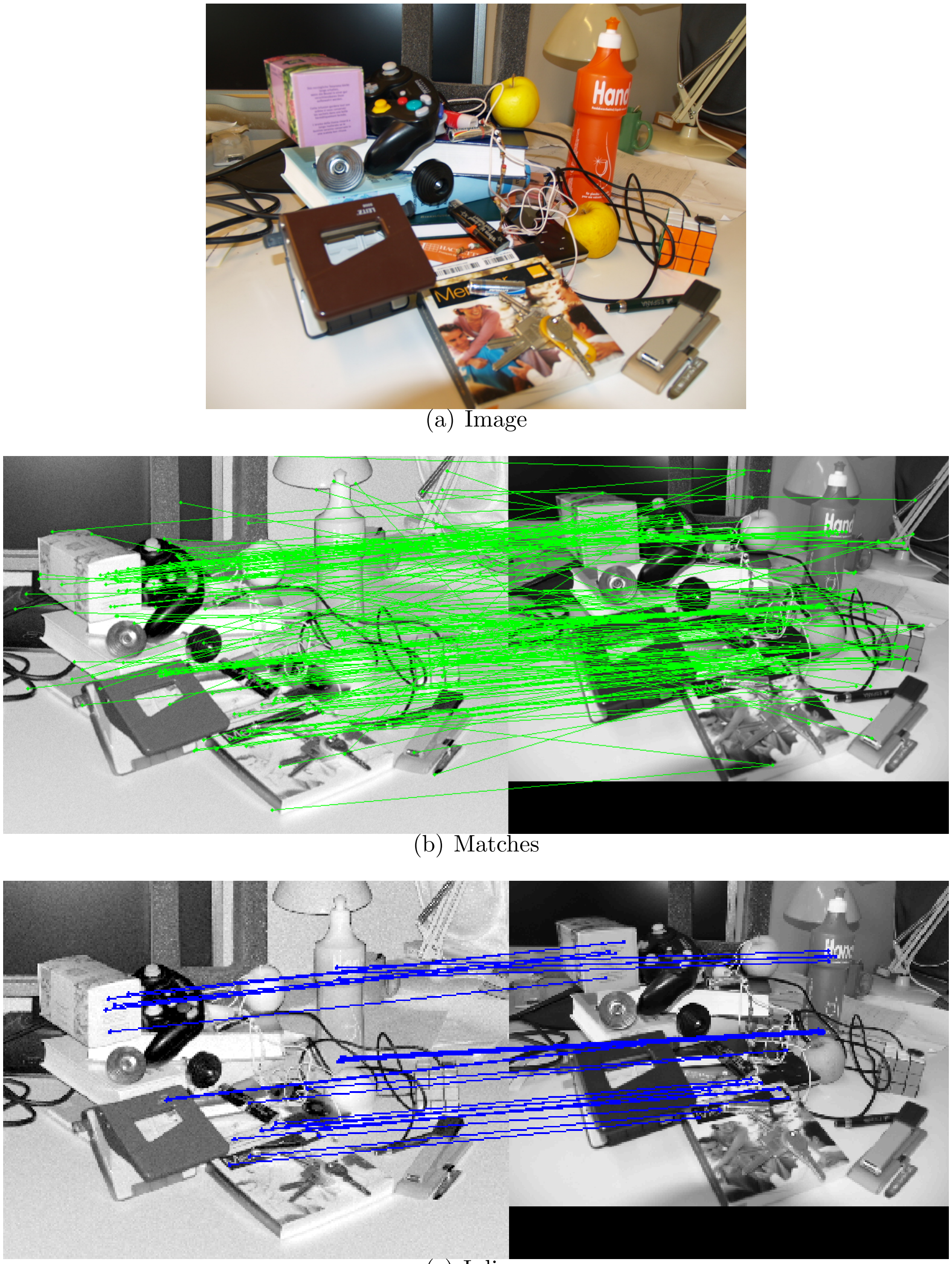


(a) Image

(b) Matches

(c) Inliers

Figure C.8: Image 56. As in the previous image, a fair amount of inliers can be found.

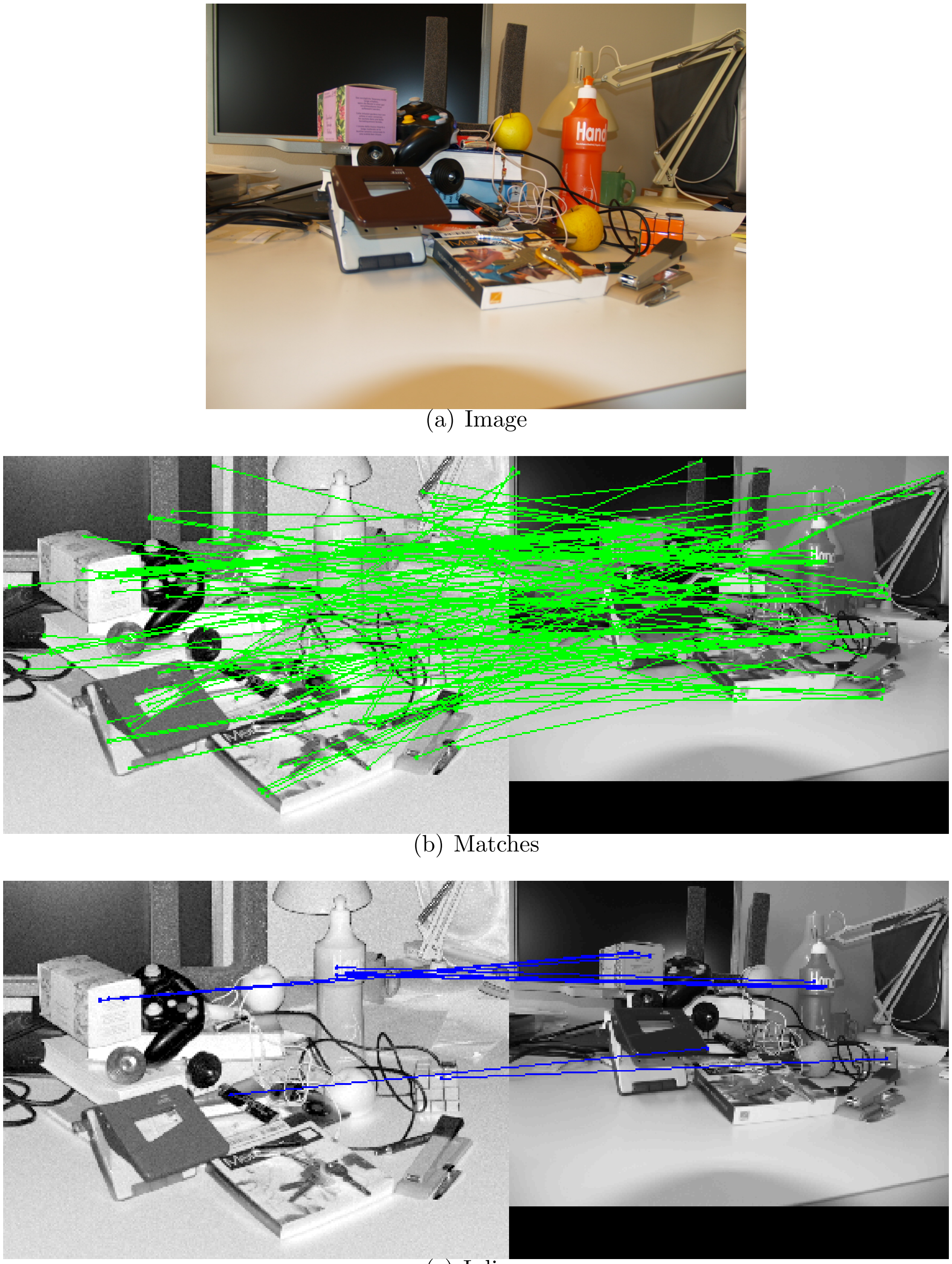


(a) Image

(b) Matches

(c) Inliers

Figure C.9: Image 64. This image has the best results of all the set in inliers. This image is very similar image to the 3D map, giving a lot of inliers.

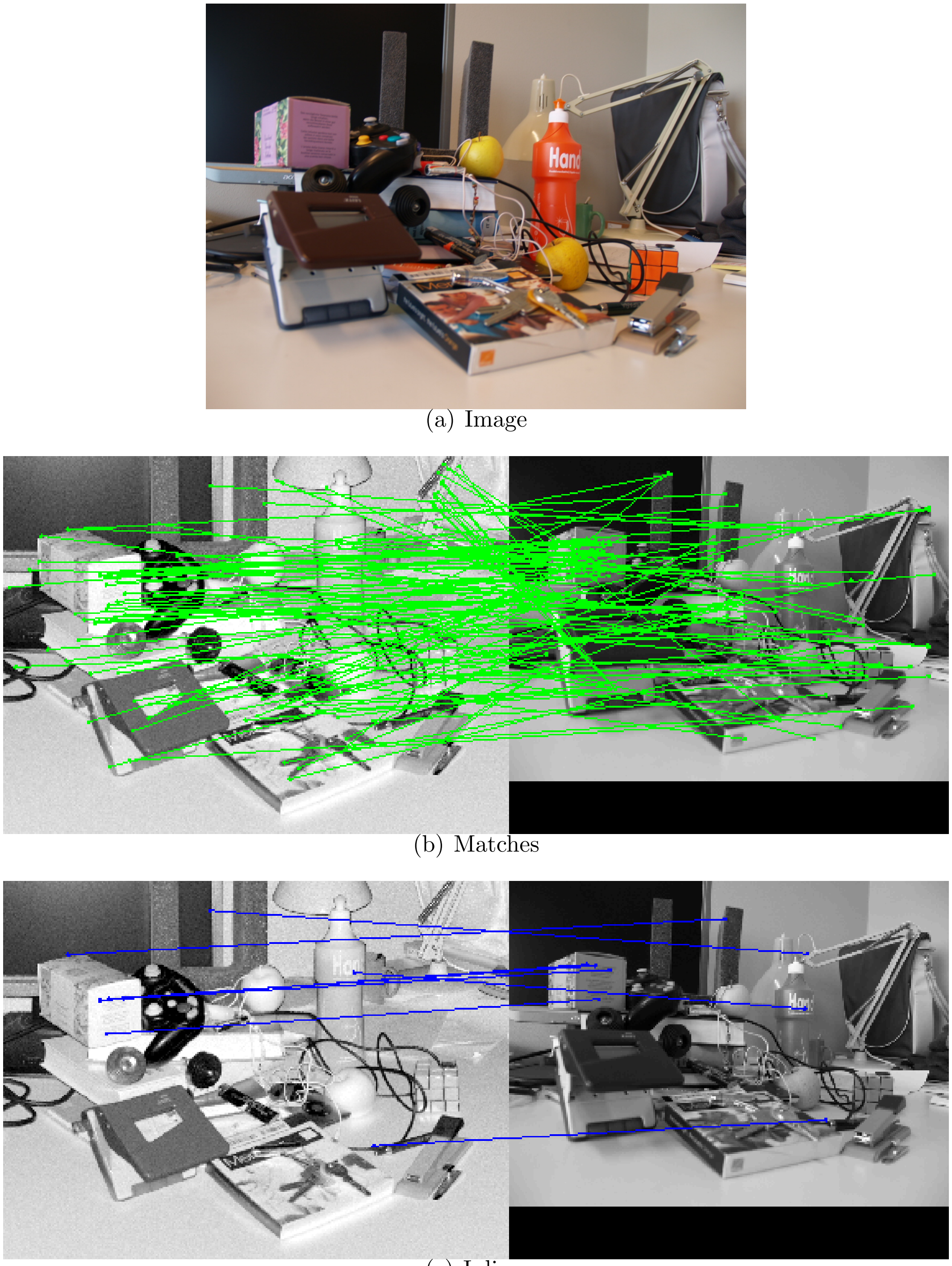


(a) Image

(b) Matches

(c) Inliers

Figure C.10: Image 73. An example from a lower view. As in many other images with a displacement to the scanner point of view, even with good matches, there is a low amount of them and the quality of the results is poor.

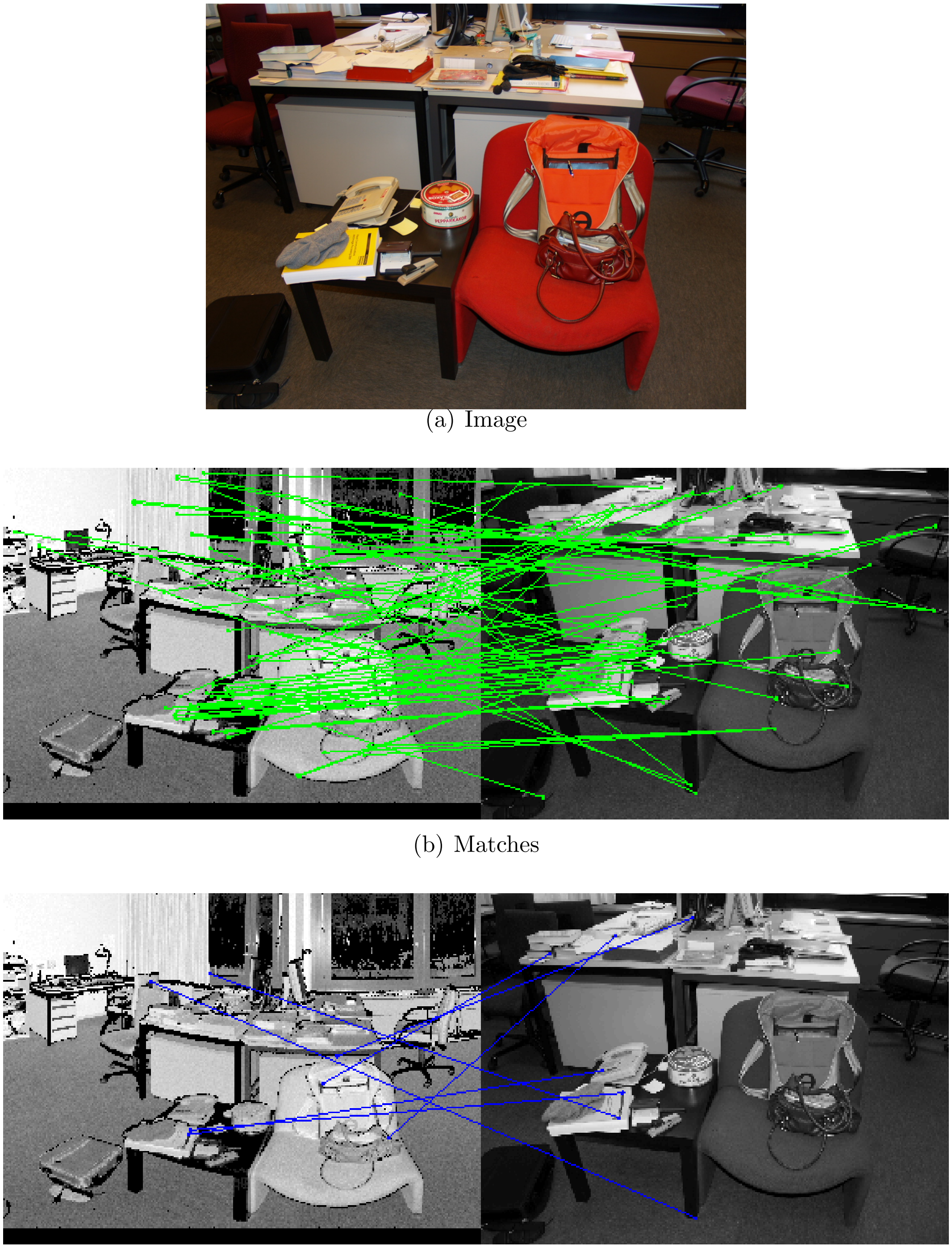


(a) Image

(b) Matches

(c) Inliers

Figure C.11: Image 74. This is an example of a bad result. The number of inliers matches is only eight and is very easy to see two bad matches, one to the top of the lamp and another one to the scanner foam rubber protection. The viewpoint is downward to the laser and the image includes objects not present in the scanner.



(a) Image

(b) Matches

(c) Inliers

Figure C.12: Image Room. The results for a more general image gives worst results compared to te images from the testing set.

Bibliography

[1] Hartley, R. and Zisserman, A. 2000. Multiple view geometry in computer vision, Cambridge University Press: Cambridge, UK.

[2] Zhang, Z., Deriche, R., Faugeras, O., and Luong, Q.T. 1995. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. Artificial Intelligence, 78:87-119.

[3] M. Fischler and R. Bolles, Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, Communications of the ACM, vol. 24, No. 6, pp. 381-385, 1981.

[4] Bernardini F. and Rushmeier H., The 3D Model Acquisition Pipeline, IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA

[5] David G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints January 5,2004 Computer Science Department University of British Columbia Vancouver, B.C., Canada

[6] Vural, E., Robust Extraction Of Sparse 3D Points From Image Sequences. Middle East Technical University, 2008.

[7] Tola, E., Multiview 3D Reconstruction Of A Scene Containing Independently Moving Objects. Middle East Technical University, 2005.

[8] Gary Bradski, Adrian Kaehler. Learning OpenCV: Computer Vision with the OpenCV Library O'Reilly Media, October 3, 2008

[9] RANSAC toolbox.
Marco Zuliani. Vision Research Lab, UCSB http://vision.ece.ucsb.edu/ zuliani/index.shtml

[10] SIFT for Matlab.
Andrea Vedaldi, University of California, Los Angeles VisionLab. http://www.vlfeat.org/ vedaldi/code/sift.html