UNIVERSITAT POLITÈCNICA DE CATALUNYA

Doctoral Programme:
AUTOMÀTICA, ROBÒTICA I VISIÓ

PhD dissertation

3D Pose Estimation in Complex Environments

Adrián Peñate Sánchez

Advisors:

Juan Andrade Cetto
Francesc Moreno Noguer

14 January, 2017

Contents

Chapter 1

Introduction

1.1 Introduction

Obtaining the 3D pose of an object in an image means identifying its position and orientationwith respect to the camera. By knowing the location and orientation of an object in the scene enables us to determine how we can interact with it in the real world. Knowing the 3D pose is a very useful tool in many different fields, we here review some of them and we further illustrate them in Fig. 1.1.

Augmented Reality Applications

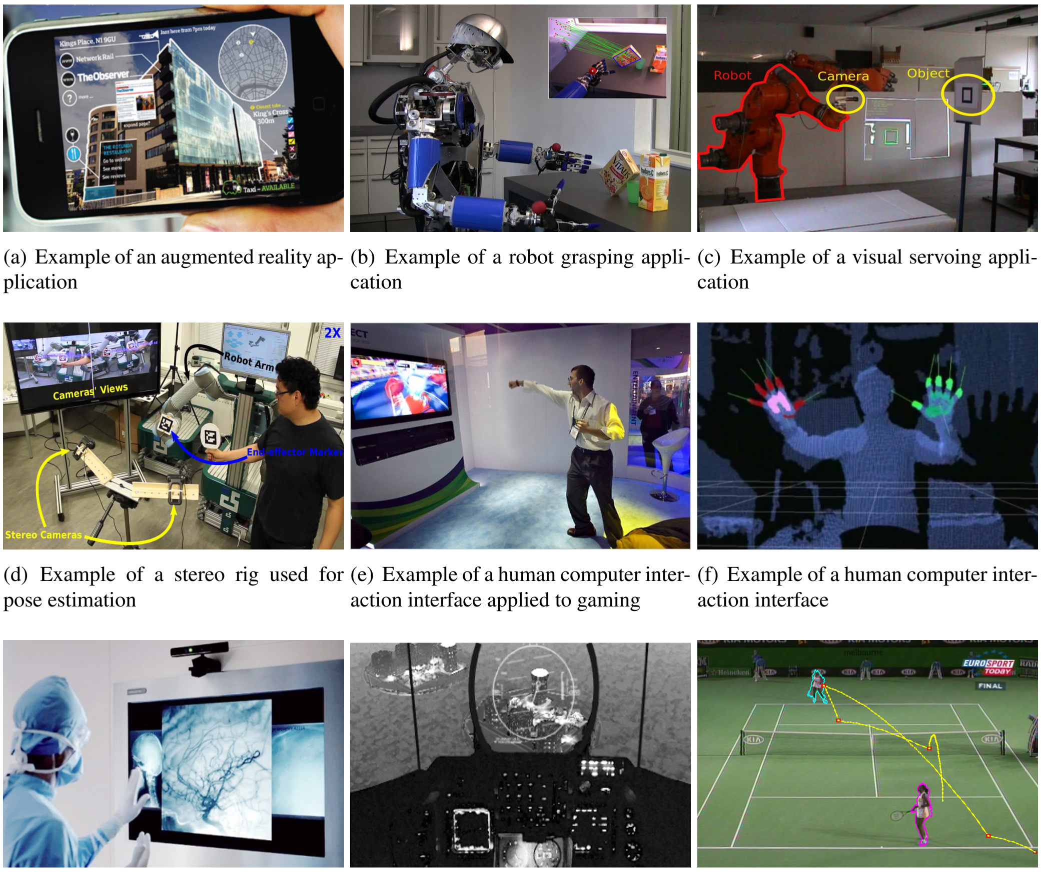
Augmented Reality has a huge potential for applications involving topics such as as medical visualization, maintenance and repair, annotation, entertainment, aircraft navigation and targeting. They all involve superposing computer generated images on real scenes, which must be done at frame-rate in online systems. 3D real-time pose estimation is therefore a critical component of most AR applications. The objects in the real and virtual worlds must be properly aligned with respect to each other, and the response time should also be as low as possible to make these applications useful.

Visual Servoing

Visual servoing involves the use of one or more cameras and a computer vision system to control the position of a device such as a robotic arm relative to a part it has to manipulate, which requires detecting, tracking, servoing, and grasping. It therefore spans computer vision, robotics, kinematics, dynamics, control and real-time systems, and is used in a rich variety of applications such as lane tracking for cars, navigation for mobile platforms, and generic object manipulation. The tracking information is required to measure the error between the current location of the robot and its reference or desired location from eye-in-hand cameras. As a consequence, the tracking algorithm must be robust, accurate, fast, and general.

Human Computer Interaction

3D pose estimation can be integrated into human computer interfaces. It can be used to continuously update the position of a hand-held object, which would serve as a 3D pointer. This object would then become an instance of what is known as a tangible interface. Such interfaces aim at replacing traditional ones by allowing users to express their wishes by manipulating familiar objects and, thus, to take advantage of their everyday experience. Eventually, this is expected to lead to more natural and intuitive interfaces. In this context, vision-based tracking is the appropriate technique for seamless interaction with physical objects.



(a) Example of an augmented reality application

(b) Example of a robot grasping application

(c) Example of a visual servoing application

(d) Example of a stereo rig used for pose estimation

(e) Example of a human computer interaction interface applied to gaming

(f) Example of a human computer interaction interface

(g) Human computer interaction interface used for medical purposes

(h) 3D tracking system for missile targeting

(i) 3D tracking system applied to understand a tennis sequence

Figure 1.1: Several examples of contexts in which 3D pose estimation is essential.

Computer Vision-Based 3D Tracking

Many other technologies besides vision have been tried to achieve 3D tracking, but they all have their weaknesses. By contrast, vision has the potential to yield non-invasive, accurate and low-cost solutions to this problem, provided that one is willing to invest the effort required to develop sufficiently robust algorithms. To offer a non-invasive solution which is as general as possible, it is desirable to rely on naturally present features, such as edges, corners, or texture.

1.2 Objectives

Although there has been remarkable progress in the pose estimation literature, there are still a number of limitations when existing approaches are to be applied in everyday applications, especially under non-controlled environments. In this thesis we seek to tackle some of these limitations, namely, computing pose for uncalibrated cameras, when features are not reliable and even when no colour information is available. We next briefly overview each of these situations.

Camera calibration is an essential ingredient in pose estimation. It basically consists on obtaining the intrinsic parameters that define the perspective camera that we are using. Without knowing these parameters one cannot solve for the rotation and translation of the object w.r.t. the camera. Usually, camera calibration has to be solved with user-assisted methods like [112]. There are pose estimation methods in the literature that solve the camera calibration problem [7, 96, 104, 48] but still camera calibration is rarely done without human intervention. For this reason we seek to further improve the case of uncalibrated pose estimation methods.

In order to solve for the 3D pose, most algorithms rely in having a set of 2D-3D correspondences. This task usually involves the extraction and matching of feature points. By matching a set of 3D points from a known model to a set of 2D points obtained through a feature point vector one can obtain the 3D pose just by solving a number of linear equations. A feature point is composed of two parts, the point of interest and a feature descriptor. The point of interest provides the coordinates of a very recognizable point in an image and the feature descriptor is the way in which we represent such point. But in many situations, feature points might not be an option due to repetitive patterns, textureless images and image degradations like blurring. To handle these contexts we will present solutions that do not require the use of features points to find an accurate pose.

Several scenarios appear when feature points cannot be used. We might be able to find points with high saliency but not be able to match them due to ambiguities in the texture. In other more difficult cases texture information might be so corrupted that one cannot find points with high saliency. The techniques needed to solve such an adverse context will not be able to rely on characteristic textures to match the known target to the scene. This techniques will instead need to hypothesize thoroughly and then validate their hypotheses against the underlying 3D structure of the scene.

Real world scenes are often subject to viewpoint changes, lighting changes, occlusions and textureless objects. In this thesis we coped with increasing difficulty in obtaining the pose for such scenes. We focused on improving the performance of pose estimation in real scenarios centering our efforts in rigid objects. The additional variability of non rigid objects or deformable objects has been out of the scope of this thesis. In particular we have identified the following four main goals to achieve in this thesis:

- Uncalibrated pose estimation from known correspondences.
- Uncalibrated pose estimation from unknown correspondences.
- Pose estimation without points of interest.
- Pose estimation without colour information.

1.3 Methodology

We first attempted to improve uncalibrated solutions to the pose estimation problem using feature points and known correspondences. That is, we assumed we were given the correct matching between points in an image and their spatial representation on the surface of an object, in this case a 3D model. Once we achieved results on this scenario we assumed a context in which feature descriptors are not reliable, thus retaining only the coordinates of the points of interest, but without knowing the one-to-one matching between them. In a further step, we attempted to solve the pose estimation problem in situations where feature points could not be detected, such as in motion blurred images. When solving for the pose without points of interest, we need to be very cautious. The quality of the images in which feature points cannot be used could be so poor that it might not be possible to distinguish the subtle differences produced by changes in the camera pose. Since, trying to cope with uncalibrated cameras, and still achieving a correct camera pose, would be a tremendous challenge, we only considered solving the calibrated case from this point forward. Finally we took up the task of obtaining a correct pose from depth images. These images can be seen as a kind of grey-scale image, the difference being that grey intensity gives us 3D structure instead of a description of the point in question. This makes finding the 3D structure easy to verify but difficult to find because points have no specific description apart from their 3D structure.

We found this sequence of tasks appropriate to achieve the goals of our work. The progression in difficulty helped us understand the actual problems of pose estimation in order to identify the specific requirements for the next step. This approach also helped the assimilation of the state-of-the-art and the acquisition of the know-how necessary to handle the task at hand. We will extensively elaborate on each one of the four main tasks separately dedicating a separate section for each. The specific state-of-the-art for each task will be defined in each section, making each topic self contained.

1.4 Contributions

There are four primary contributions in this thesis, all of them focused on the pose estimation problem:

1. We have presented a novel approach to perform uncalibrated pose estimation. Drawing inspiration in [58], we provide a new approach that shows similar efficiency and precision but that besides retrieving the pose, it estimates the camera focal length. This constitutes a new state-of-the-art solution applied to uncalibrated cameras with an arbitrarily large number of points. [78]

2. We have extended the Blind P*n*P [70] method to simultaneously estimate pose and 3D-to-2D point correspondences to an uncalibrated context. Such generalization involves finding the correct solution in higher dimensionalities, rendering the problem more complex. To overcome such problem we have provided a new formulation and cost functions. We have also circumvented the ambiguity problem that exists between translation and focal length variation in a noisy context by clustering the focal length and stopping a free gradient descent over this dimension. The provided solution has shown to be robust and capable of providing accurate pose estimates. [80]

3. We want to be able to perform accurate 3D pose estimation in low quality images. To do so we developed a new learning technique that merges 3D inputs with traditional learning approaches to robustify the classification task. The introduction of geometric priors in the learning phase allows the use of a single classifier for all viewpoints of an object. The classifier is learned in a way that takes into account the visibility of parts of the object during testing, given by the 3D model With this we are able to activate or deactivate the different parts of the object depending on if they are visible or not. This new paradigm is sufficiently robust to be able to solve pose estimation reliably even under the hardest contexts of degraded image quality. [79]

4. Finally, we have given a new solution to the 3D object pose estimation problem for non-textured point clouds. Not having texture information makes discerning a point from another a really difficult task. To perform 3D pose estimation over point clouds we integrated several weak priors and then pruned most of them robustly through geometric checks. The key issue is to do such process in an efficient manner, and we provided the most efficient approach to this day, obtaining between 10 to 100 times of speed-up against the state-of-the-art without sacrificing precision. [111]

We prioritize solutions that are simple, general and fast. Our main objective is to build enhanced approaches to the 3D pose estimation problem, which can thereafter be used in place of more traditional ones. We have compared our approach to the main state-of-the-art algorithms and shown the increase in performance both in precision and in efficiency.

1.5 Thesis overview

This thesis is structured in the following manner: four chapters describing each of our four main publications, plus an introductory chapter to lay the groundwork and present material common to several or all of our papers. Here is a summary of the chapters:

Chapter 2: Overview. This introductory chapter presents some of the tools used throughout our work, from geometric PnP algorithms, through robust matching algorithms up to learning schemes.

Chapter 3: Uncalibrated Pose Estimation from Known Point Correspondences. This chapter describes our novel solution to the uncalibrated PnP, in which we extend a previous formulation to an uncalibrated context.

Chapter 4: Uncalibrated Pose Estimation from Unknown Point Correspondences. This chapter introduces our technique to perform uncalibrated pose estimation and matching at the same time. We explain how a Kalman filter can be used to determine the candidate matches, and, by taking into account such matches estimate an error that will determine the direction in which to move in the pose space.

Chapter 5: Pose Estimation without Points of Interest. This chapter presents a new learning paradigm that performs a joint learning of an object and its pose at the same time. It also provides a geometric evaluation of the classifier by hypothesizing the pose of the object. This work is a collaboration with F. Fleuret of IDIAP Research Institute in Martigny, Switzerland.

Chapter 6: Pose Estimation without Colour Information. This chapter presents a new algorithm to perform object pose estimation in point clouds. This is a simple and efficient approach that beats the baseline by obtaining the same precision values but more than 10 times faster. This work was done at Toshiba Research Cambridge, in collaboration with C. Zach and M.T. Pham.

Chapter 7: Concluding remarks. The thesis concludes with a short summary of our efforts, an interpretation on where our work stands in a field currently undergoing significant developments, and details about how parts of the body of research presented in this thesis are linked together.

Chapter 2

Overview

2.1 Introduction

In this chapter, we will introduce all preliminary concepts and algorithms needed to understand the rest of the work presented in this thesis. We seek making this thesis as self contained as possible and for this reason we will introduce basic concepts of the field that might seem obvious but that will give a basic understanding to a reader from a different field. The included concepts will be presented in an order that will go from the basic underlying geometric definitions and algorithms to the more high level learning or probabilistic methods. This narrative makes dependencies between concepts to be incremental, thus, when approaching a new concept not requiring much knowledge of topics not yet presented. We will also give detailed descriptions and full notation for those methods on which we have contributed with new solutions.

Type	C1	C2	C3	C4	Work & citation
Geometry	✓	✓	✓	✓	Perspective Camera model
	✓	✓		✓	Horn's method [40]
PnP Algorithms	✓				EPnP [69]
		✓			BPnP [70]
Matching Algorithms	✓	✓		✓	RANSAC [26]
				✓	FLANN [71]
Features	✓				SIFT [60]
Probabilistic Algorithms		✓	✓		Kalman Filter
Structure from Motion			✓		BUNDLER [95]
Learning Algorithms			✓		AdaBoost [30]

Table 2.1: Index of the methods used in this thesis, and specification about in which chapters they are used.

2.2 Camera model

In this section we will describe the standard pinhole camera model used for this thesis. This type of model is very popular and commonly used since most modern cameras have this geometric construction,

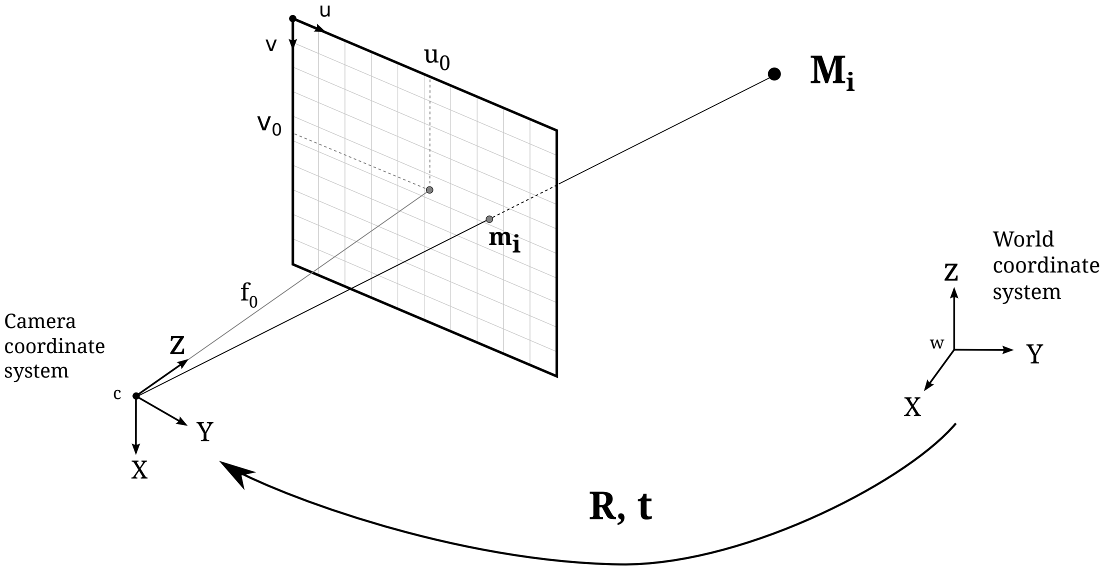


Figure 2.1: **3D Pose in the Perspective Projection Model:** Given a set of 3D points $\mathbf{M_i}$ expressed in a world reference frame, and their 2D projections $\mathbf{m_i}$ onto the image, we seek to retrieve the pose ($\mathbf{R}$ and $\mathbf{t}$) of the camera w.r.t. the world that gives rise to such projection.

therefore, all geometric notions used in this body of work apply as well as all the scientific contributions.

2.2.1 3D Pose in the Perspective Projection Model

Computing a 3D pose normally involves a set of 3D-to-2D correspondences between n reference points $\mathbf{M_1}, ..., \mathbf{M_n}$ where $\mathbf{M_i} = [X, Y, Z]^T$ are expressed in an Euclidean world coordinate system w and their pairing 2D perspective projections $\mathbf{m_1}, ..., \mathbf{m_n}$ where $\mathbf{m_i} = [u, v]^T$ are the coordinates of the point in the image plane (See Fig. 2.1).

The projection of a 3D point on the image plane can be written as:

$$s\tilde{\mathbf{m_i}} = \mathbf{P}\tilde{\mathbf{M_i}}, \tag{2.1}$$

where s is a scale factor, $\tilde{\mathbf{m_i}} = [u, v, 1]^T$ and $\tilde{\mathbf{M_i}} = [X, Y, Z, 1]^T$ are the homogeneous coordinates of points $\mathbf{m_i}$ and $\mathbf{M_i}$, and **P** is a 3 x 4 projection matrix. Since **P** is defined up to scale, it is defined by only 11 parameters.

The perspective matrix **P** can be decomposed as:

$$\mathbf{P} = \mathbf{A}[\mathbf{R}|\mathbf{t}] \tag{2.2}$$

where:

- **A** is a 3x3 matrix which contains the intrinsic camera parameters including the focal length, the scale factor and the optical centre point coordinates.

- $[\mathbf{R}|\mathbf{t}]$ is a 3x4 matrix which corresponds to the Euclidean transformation from a world coordinate system to the camera coordinate system. **R** is the rotation matrix, and **t** the translation vector.

2.2.2 Intrinsic Parameters

In the previous section we defined **A** as a matrix with the intrinsic camera calibration parameters, usually referred as camera calibration matrix. It can be written as:

$$\mathbf{A} = \begin{bmatrix} \alpha_u & s & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.3}$$

where:

- α_u and α_v are the scale factors defined in each coordinate direction u and v. These scale factors are proportional to the camera focal length, i.e. $\alpha_u = k_u f$ and $\alpha_v = k_v f$, where k_u and k_v are the total number of pixels per unit in the u and v directions.

- $\mathbf{c} = [u_0, v_0]^T$ represents the principal point coordinates. The principal point is the point in which the optical axis and the image plane intersect.

- s, referred as the skew angle, is the ratio which defines the perpendicularity of the *u* and *v* directions. In modern cameras usually this value is zero.

Often, a common approximation is to set the principal point **c** at the image center. Furthermore, in modern cameras we assume that pixels have a squared shape, which leads us to take α_u and α_v with equal values. When the intrinsic parameters are known, the camera is said to be calibrated.

2.2.3 Extrinsic Parameters

Previously, we defined $[\mathbf{R}|\mathbf{t}]$ as a 3x4 matrix which corresponds to the Euclidean transformation from a world coordinate system to the camera coordinate system. In fact, this matrix is the horizontal concatenation of the rotation matrix and the translation vector which is often referred to as the *camera pose*. (See eq. 2.4)

$$[\mathbf{R}|\mathbf{t}] = \begin{bmatrix} R_{11} & R_{12} & R_{13} & t_1 \\ R_{21} & R_{22} & R_{23} & t_2 \\ R_{31} & R_{32} & R_{33} & t_3 \end{bmatrix}, \tag{2.4}$$

Most pose estimation algorithms assume the calibration matrix **A** to be known and seek to estimate **R** and **t** through the minimization of the reprojection error:

$$\min_{\mathbf{R},\mathbf{t}} \sum_{i=1}^{n} \|u_i - \tilde{u}_i\|^2 \tag{2.5}$$

where **R** and **t** are the parameters of the Euclidean transformation that explains the motion of points from the world coordinate system w to the camera coordinate system c. A 3D point represented by the vector $\mathbf{p}_i^w$ in world coordinates will be represented by the vector $\mathbf{p}_i^c = \mathbf{R} * \mathbf{p}_i^w + \mathbf{t}$ in the camera coordinate system. From the previous formulation we can easily obtain the *camera centre* **c**, also known as *optical centre*. **c** can be recovered in the world coordinate system by satisfying $0 = \mathbf{Rc} + \mathbf{t}$, and then $\mathbf{c} = -\mathbf{R}^{-1}\mathbf{t} = -\mathbf{R}^\top\mathbf{t}$ (since **R** is a rotation matrix).

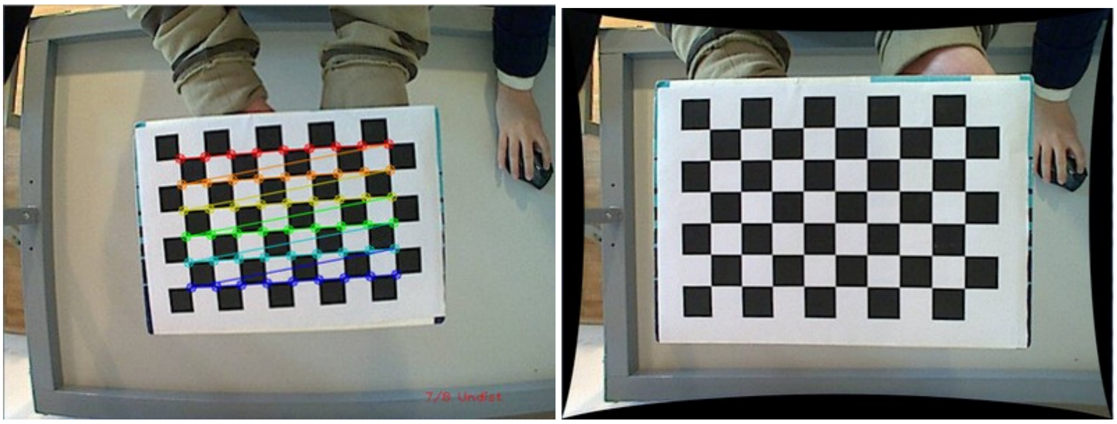


Figure 2.2: Screenshots of a camera calibration example. On the left, the pattern detection. On the right, the rectified image after the camera calibration process using the estimated intrinsic parameters. This particular case performs a full calibration of the camera parameters, including radial distortion. Depending on the specific camera used, the calibration can be simplified to assume certain effects to be negligible and thus greatly improving the complexity of the problem.

2.2.4 Camera Calibration

Most 3D pose estimation algorithms assume fixed intrinsic cameras, thus, the camera zoom has to remain constant during the whole procedure. Furthermore, there parameters are typically computed offline by processing images captured with the same camera that will then be used for estimating pose.

The standard methodology to estimate these parameters is using a calibration (reference) object with a pattern of known size. The most common objects are black-white chessboards, (See fig. 2.2), and symmetrical or asymmetrical circle patterns. The goal is to find the 2D-3D correspondences between the grid patterns centres to finally compute the projection matrix. Several toolboxes are readily available, e.g. *OpenCV* [6] or *Matlab* [64], which provide user friendly automated applications to carry on this process.

2.3 Rigid 3D-to-3D Transformation

Finding the relationship between two coordinate systems using pairs of measurements of the coordinates of a number of points in both systems is a classic photogrammetric task. In [41, 42] Horn presented closed-form solutions to the least-squares problem for three or more points. The derivation of the solution was done using both unit quaternions [41] and orthonormal matrices [42] to represent rotation, both approaches are nearly equivalent and mostly differ on the way the proof is obtained. Horn's approach finally boils down to the following:

- First we obtain the centroids υ^w and υ^c of the points in both the world coordinates and camera coordinates as $\upsilon^w = \frac{1}{n}\sum \mathbf{p}_i^w$ and $\upsilon^c = \frac{1}{n}\sum \mathbf{p}_i^c$ respectively.

- To obtain the rotation we first translate all points by their respective centroid as $\mathbf{p}_i'^w = \mathbf{p}_i^w - \upsilon^w$ and $\mathbf{p}_i'^c = \mathbf{p}_i^c - \upsilon^c$. Afterwards, we build a matrix $\mathbf{Q}$ as $\mathbf{Q} = \sum \mathbf{p}_i'^{w\top} * \mathbf{p}_i'^w$. Finally the rotation matrix $\mathbf{R}$ using the singular value decomposition of $\mathbf{Q} = \mathbf{U\Sigma V}^\top$, and then $\mathbf{R}$ as $\mathbf{R} = \mathbf{UV}^\top$. By using the SVD decomposition we are effectively performing a change in the basis of the coordinate system.

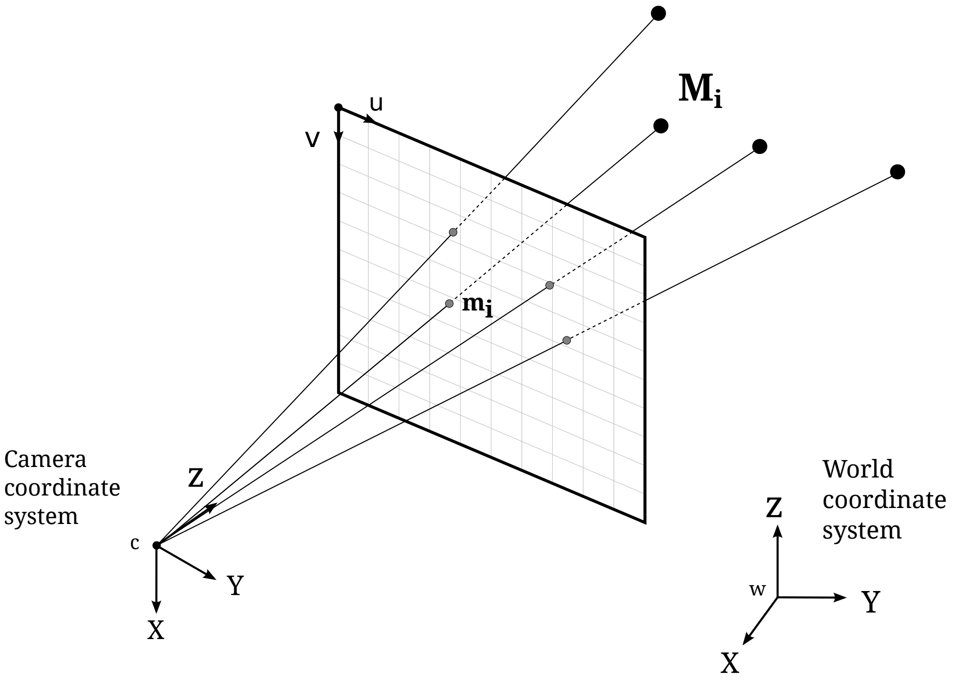


Figure 2.3: **P*n*P problem scheme:** Given a set of 3D points $\mathbf{M_i}$ expressed in a world reference frame, and their 2D projections $\mathbf{m_i}$ onto the image, we seek to retrieve the pose ($\mathbf{R}$ and $\mathbf{t}$) of the camera w.r.t. the world.

- We obtain translation as $\mathbf{t} = v^c \mathbf{R} v^w$.

These exact results are to be preferred to approximate methods based on measurements of a few selected points.

2.4 EP*n*P

We next describe a method to estimate the camera pose using a set of n 3D-to-2D correspondences and known camera parameters. Introduced by F.Moreno *et al.'s* [58], the EP*n*P algorithm solves the camera pose in an efficient way assuming that the camera parameters and a set of n correspondences between 3D points in the world coordinate system and their 2D image projections are known. This approach represents all points as a weighted sum of 4 non-coplanar virtual *control points* (See fig. 2.4). The problem is reformulated as follows:

$$\mathbf{p}_i^w = \sum_{i=1}^{4} \alpha_{ij} \mathbf{c}_j^w \tag{2.6}$$

where $\mathbf{p}_i^w = [X^w, Y^w, Z^w]^\top$ is a 3D point in world coordinates, α_{ij} are the homogeneous barycentric coordinates and $\mathbf{c}_j^w = [X^w, Y^w, Z^w]^\top$ is a 3D control point in world coordinates. Then, the 4 control

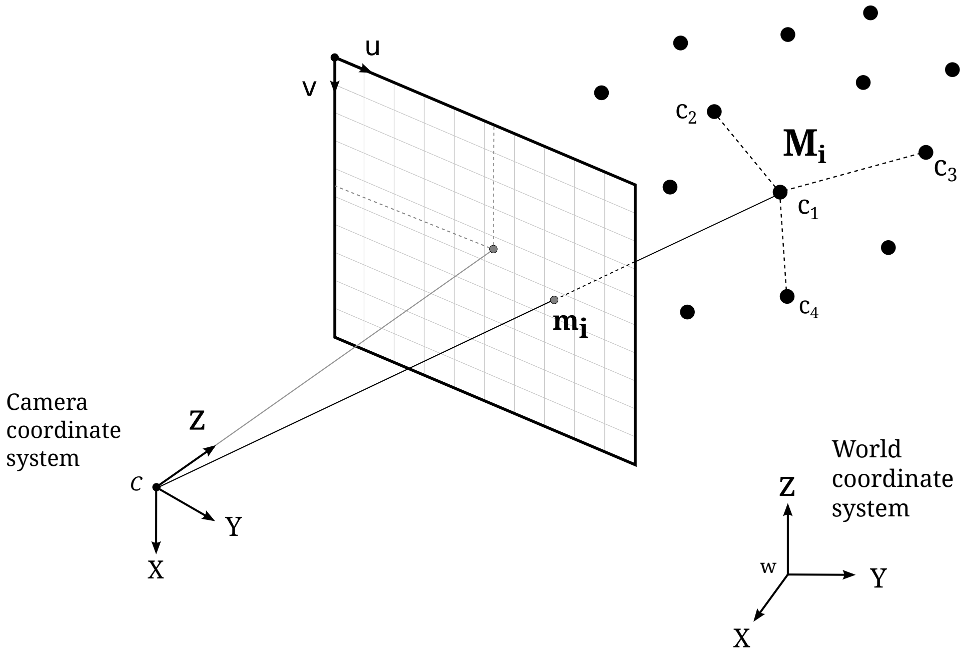


Figure 2.4: **EP*n*P formulation:** Given a set of 3D points $\mathbf{M}_i$ expressed in a world reference frame, and their 2D projections m_i onto the image, the points $\mathbf{c_{1..4}}$ form the base which represents all the set by a linear combination in order to retrieve the pose ($\mathbf{R}$ and $\mathbf{t}$) of the camera w.r.t. the world.

points in camera coordinates $\mathbf{c}_j^c$ become the unknown of the problem, giving a total of 12 unknowns.

It is needed to build a linear system in the control points reference frame:

$$w_i \begin{bmatrix} \mathbf{u}_i \\ 1 \end{bmatrix} = \mathbf{A} p_i^c = \mathbf{A} \sum_{i=1}^{4} \alpha_{ij} c_j^c, \forall i \tag{2.7}$$

where w_i are the scalar projective parameters, $\mathbf{u}_i$ are the 2D coordinates $[u_i, v_i]^T$, $\mathbf{A}$ is matrix of intrinsic parameters. This expression can be rewritten as:

$$w_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} f_u & 0 & u_c \\ 0 & f_v & v_c \\ 0 & 0 & 1 \end{bmatrix} \sum_{i=1}^{4} \alpha_{ij} \begin{bmatrix} x_j^c \\ y_j^c \\ z_j^c \end{bmatrix}, \forall i \tag{2.8}$$

From 2.8 we can obtain two linearly independent equations (for each 3D-to-2D correspondence):

$$\begin{aligned} \sum_{i=1}^{4} \alpha_{ij} f_u x_j^c + \alpha_{ij}(u_c - u_i) z_j^c &= 0, \\ \sum_{i=1}^{4} \alpha_{ij} f_v y_j^c + \alpha_{ij}(v_c - v_i) z_j^c &= 0, \end{aligned}$$

Considering all correspondences we can build the following linear system:

$$\mathbf{M}\mathbf{x} = \mathbf{0} \tag{2.9}$$

where $\mathbf{M}$ is a $2n$x12 matrix with known coefficients and $\mathbf{x} = [\mathbf{c}_1^{c\top}, \mathbf{c}_2^{c\top}, \mathbf{c}_3^{c\top}, \mathbf{c}_4^{c\top}]^\top$ is a 12-vector made of the unknowns. We then apply a linear least squares approach and obtain a system in the form $\mathbf{M}^\top\mathbf{M}\mathbf{x} = \mathbf{0}$. The solution then lies on the null space, or kernel, of $\mathbf{M}^\top\mathbf{M}$, expressed:

$$\mathbf{x} = \sum_{i=1}^{N} \beta_i \mathbf{v}_i \tag{2.10}$$

where $\mathbf{v}_i$ are the eigenvectors of $\mathbf{M}^\top\mathbf{M}$ whose eigenvalues are equal to 0. The efficiency of the EP*n*P method remains in the transformation of $\mathbf{M}$ into a small constant matrix $\mathbf{M}^\top\mathbf{M}$ of size 12x12 before computing these eigenvectors.

In Eq.2.10 it can be seen that we need to determine the size of the kernel of $\mathbf{M}^\top\mathbf{M}$, N. This is usually done taking the eigenvectors that correspond to eigenvalues that are 0. In practice the solution will be obtained only for $N = 1, ..., 4$ as proven in [58] due to the noise in the measurements of both the 2D and 3D points.

$$\begin{aligned} N = 1: \mathbf{x} &= \beta_1\mathbf{v}_1 \\ N = 2: \mathbf{x} &= \beta_1\mathbf{v}_1 + \beta_2\mathbf{v}_2 \\ N = 3: \mathbf{x} &= \beta_1\mathbf{v}_1 + \beta_2\mathbf{v}_2 + \beta_3\mathbf{v}_3 \\ N = 4: \mathbf{x} &= \beta_1\mathbf{v}_1 + \beta_2\mathbf{v}_2 + \beta_3\mathbf{v}_3 + \beta_4\mathbf{v}_4 \end{aligned}$$

In order to solve for each β, geometric constraints are added. By doing so, the method enforces that the distances between control points in the camera coordinate system remain equal to those in the world coordinate system. That is,

$$\|\mathbf{c}_i^c - \mathbf{c}_j^c\|^2 = \|\mathbf{c}_i^w - \mathbf{c}_j^w\|^2, \tag{2.11}$$

For instance, taking the case $N = 1$ we would have that

$$\|\beta_1\mathbf{v}_i - \beta_1\mathbf{v}_j\|^2 = \|\mathbf{c}_i^w - \mathbf{c}_j^w\|^2, \tag{2.12}$$

Then β_1 can be computed as follows

$$\beta_1 = \frac{\sum_{\{i,j\}\in[1;4]} \|\mathbf{v}_i - \mathbf{v}_j\| \cdot \|\mathbf{c}_i^w - \mathbf{c}_j^w\|}{\sum_{\{i,j\}\in[1;4]} \|\mathbf{v_i} - \mathbf{v_j}\|^\mathbf{2}}, \tag{2.13}$$

Once the betas are computed, one just needs to solve the previous system of equations and obtain the control points coordinates in the camera frame. Finally, it just remains to compute the coordinates of all 3D points in the camera frame reference and, as shown in [40], compute the rotation matrix **R** and the translation vector **t**, as detailed in 2.3.

2.5 RANSAC

The Random Sample Consensus method or RANSAC [26] is a non-deterministic iterative method which estimates parameters of a mathematical model from observed data producing an approximate result as the number of iterations increase.

```
RANSAC($\mathbf{pIn}$)
    INPUT:
        $\mathbf{pIn}$:   Set of 2D-3D point matches
    OUTPUT:
        $\mathbf{pOut}$: Subset of the original point matches without outliers.

$Consensus \leftarrow$ empty vector
$NConsensus \leftarrow 0$
$K \leftarrow \frac{\log 1-p}{\log 1-w^n}$ ()
{Calculate K as defined in 2.14}
for $Iterations < K$ do
    Select at random a minimum set from $\mathbf{pIn}$;
    Generate a model using the minimum set of point matches;
    $Samples \leftarrow$ point matches that confirm the model;
    $NSamples \leftarrow$ size($Samples$);
    if $NSamples > NConsensus$ then
        $Consensus \leftarrow Samples$
        $NConsensus \leftarrow NSamples$
    end if
end for
$\mathbf{pOut} \leftarrow \mathbf{Consensus}$
```

Algorithm 1: RANSAC algorithm.

In the context of camera pose estimation, it is very simple to implement since an initial guess of the parameters is not needed. From the set of correspondences, the algorithm randomly extracts small subsets of points to generate what is called the hypothesis. For each hypothesis a P*n*P approach is used to recover a camera pose which then is used to compute the reprojection error. The points in which the reprojection error is under a threshold are called inliers.

RANSAC depends on some parameters such as the tolerance error, which decides whether a point will be considered as an inlier based on the reprojection error. It also depends on the number of iterations, in [26], a formula to compute the number of iterations given a desired probability *p* that at least one of the hypothesis succeed as a consistent solution is given. The mentioned formula is the following:

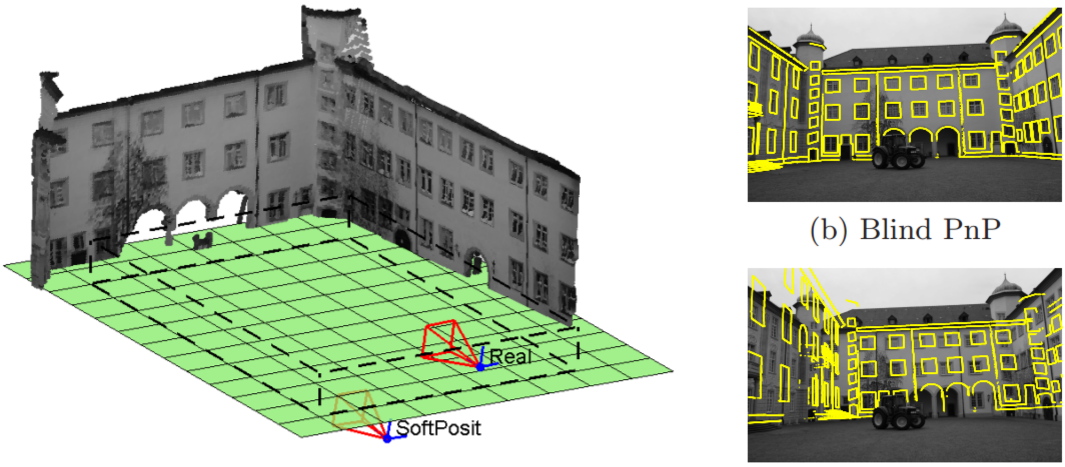
$$k = \frac{\log(1-p)}{\log(1-w^n)}, \tag{2.14}$$

where w is the expected ratio between inliers and the number of points it can always be set to a safely large value). The parameter *k* tends to increase with the size of the subsets. The step by step process RANSAC uses to remove outlier matches is detailed in Algorithm 1.

2.6 BP*n*P

In [70] the authors proposed the BP*n*P, an approach robust to occlusion, clutter, and repetitive patterns that simultaneously solves for pose and correspondences in scenes without discriminative texture. Using SoftPosit [16] as baseline, BP*n*P showed an improved behaviour against occlusion, clutter and repetitive patterns. The advantage of BP*n*P compared to RANSAC is in those cases where RANSAC has a high cost and BP*n*P yields a less computational complex solution to the problem just by introducing pose priors.

The main characteristic of the BP*n*P is to rely on the fact that prior information on the camera pose is often available. This prior can be used to eliminate many spurious solutions. For example, as depicted



(a)

(b) Blind PnP

(c) SoftPosit

Figure 2.5: Recovering the pose in a scene with repetitive patterns. (a) 3D model of the scene. The "Real" camera, and the pose recovered by SoftPosit [16] are indicated. BlindP*n*P is able to retrieve the "Real" pose. (b) Model reprojected after estimating the pose using BlindP*n*P. (c) Model reprojected after estimating the pose using SoftPosit.

in Fig. 2.5, the camera must be pointing towards the building and be above ground level. Such a prior is modeled as a Gaussian Mixture Model (GMM) and each component of the GMM is used to initialize a Kalman filter. The proposed approach explores then the space of possible correspondences within a subset of potential matches and keeps the hypothesis that yields the smallest reprojection error.

2.7 Approximate Nearest Neighbor Search: FLANN

For many computer vision and machine learning problems the most computationally expensive part corresponds to the process of finding nearest neighbour matches of high dimensional vectors. In [71], new algorithms for approximate nearest neighbour matching were proposed. For matching high dimensional features, they found two algorithms to be the most efficient: the randomized k-d forest and a the priority search k-means tree. The common architecture of both approaches exploit is depicted in Fig. 2.6. The paper also proposes a new binary feature matching algorithm consisting in searching multiple hierarchical clustering trees. By providing an efficient matching algorithm and an efficient search scheme the authors provide a state-of-the-art solution to the problem of matching binary features in large datasets. All this research has been released as an open source library called fast library for approximate nearest neighbours (FLANN), which has been incorporated into OpenCV [6] and is now one of the most popular approaches for nearest neighbour matching.

2.8 SIFT

SIFT [60], is an interest point descriptor based on multiple orientation histograms, which tolerates significant local deformations. This descriptor has been shown in [67] to be very efficient and robust to changes.

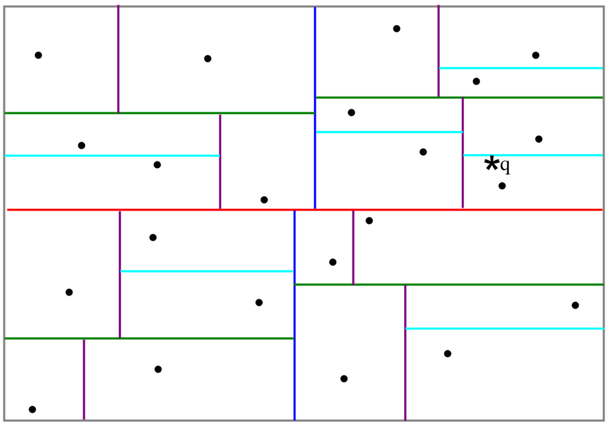


Figure 2.6: Example of randomized kd-trees. The nearest neighbour is determined by using the decision boundaries that split the space in 2 parts successively. Every time we use a decision boundary to determine were our candidate point lies we reject approximately half the points that could be matched. This reduces the computational complexity from a $O(N)$ complexity to a $O(\log N)$ complexity.

The remarkable invariance of the SIFT descriptor is achieved by a succession of carefully designed techniques. First the location and scale of the keypoints are determined precisely by interpolating the pyramid of Difference-of-Gaussians used for the detection. To achieve image rotation invariance, an orientation is also assigned to the keypoint. It is taken to be the one corresponding to a peak in the histogram of the orientation of gradients within a region around the keypoint. This method is quite stable under viewpoint changes, and achieves an accuracy of a few degrees. The image neighbourhood of the feature point is then corrected according to the estimated scale and orientation, and a local descriptor is computed on the resulting image region to achieve invariance to the remaining variations, such as illumination or out-of-plane variation. The point neighbourhood is divided into several, typically 4×4, subregions and the contents of each subregion is summarized by an eight-bin histogram of gradient orientations. A graphical description of the approach can be seen in Fig. 2.7. The keypoint descriptor becomes a vector with 128 dimensions, built by concatenating the different histograms. Finally, this vector is normalized to unit length to reduce the effects of illumination changes.

2.9 Kalman Filter

The Kalman Filter is an algorithm that uses a series of measurements observed over time, containing statistical noise and other inaccuracies, and produces estimates of unknown variables that tend to be more precise than those based on a single measurement alone. The algorithm works in a two-step process. In the prediction step, the Kalman filter produces estimates of the current state variables, along with their uncertainties. Once the outcome of the next measurement (necessarily corrupted with some amount of error, including random noise) is observed, these estimates are updated using a weighted average, with more weight being given to estimates with higher certainty. The algorithm is recursive. It can run in real time, using only the present input measurements and the previously calculated state and its uncertainty matrix. The underlying model is a Bayesian model similar to a hidden Markov model but where the state space of the latent variables is continuous and where all latent and observed variables have Gaussian distributions.

The successive states $\mathbf{s}_t \in R^n$ of a discrete-time controlled process are assumed to evolve according

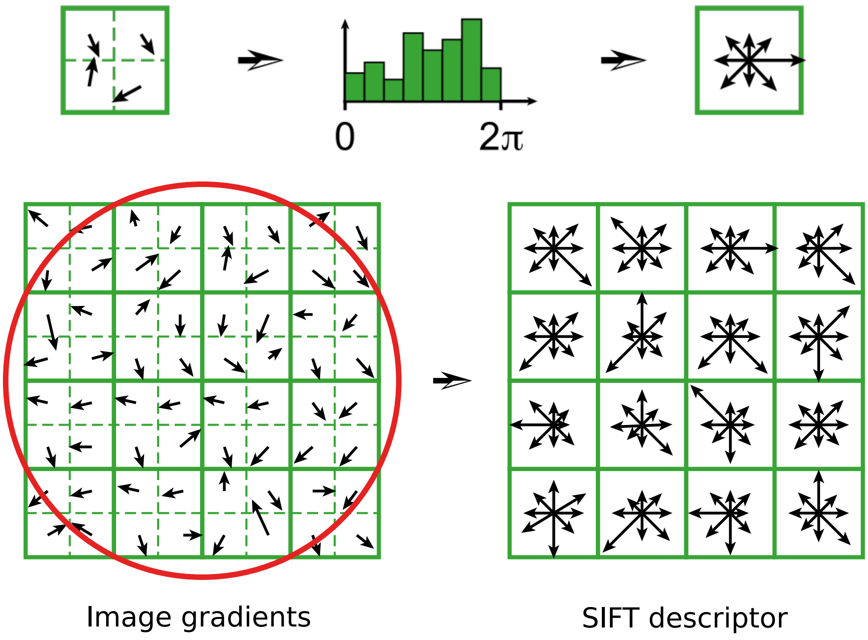


Figure 2.7: SIFT descriptors are 3-D histograms of image gradients over the spatial coordinates and gradient orientations. The image gradients are weighted by a gaussian window, indicated by the red circle on the bottom left figure. The length of the arrows corresponds to the sum of gradient magnitudes on a given direction. Released under a CC-BY-SA-3.0 license (Creative Commons), edited afterwards.

to a dynamic model written as follows

$$\mathbf{s}_t = \mathbf{K}\mathbf{s}_{t-1} + \mathbf{w}_t, \tag{2.15}$$

where K is called the *state transition matrix*, and $\mathbf{w}_t$ represents the process noise which is assumed to be normally distributed with zero mean. For instance, in the case of tracking the pose of a moving camera, the state vector will be comprised by the 6 parameters of the camera pose, plus the translational and angular velocities.

The measurements $\mathbf{z}_t$ such as the the camera pose at time t, are assumed to be related to the state $\mathbf{s}_t$ by a linear measurement model

$$\mathbf{z}_t = \mathbf{C}\mathbf{s}_t + \mathbf{v}_t, \tag{2.16}$$

where $\mathbf{v}_t$ represents the measurement noise.

At each time step, the Kalman Filter makes a fist estimation of the current state called the *a priori* state estimate $\mathbf{s}_t^-$, which is refined by incorporating the measurements to yield the *a posteriori* estimate $\mathbf{s}_t$. $\mathbf{s}_t^-$ and its covariance matrix $\mathbf{S}_t^-$, are computed during the prediction stage and can be written as

$$\begin{aligned} \mathbf{s}_t^- &= \mathbf{K}\mathbf{s}_{t-1}, & (2.17) \\ \mathbf{S}_t^- &= \mathbf{K}\mathbf{S}_{t-1}\mathbf{K}^T + \mathbf{\Lambda}_\mathbf{w}, & (2.18) \end{aligned}$$

where $\mathbf{S}_{t-1}$ is the *a posteriori* estimate error covariance for the previous time step, and $\mathbf{\Lambda}_\mathbf{w}$ is the process covariance noise that measures quality of the motion model respect to the reality. Next, the Kalman Filter

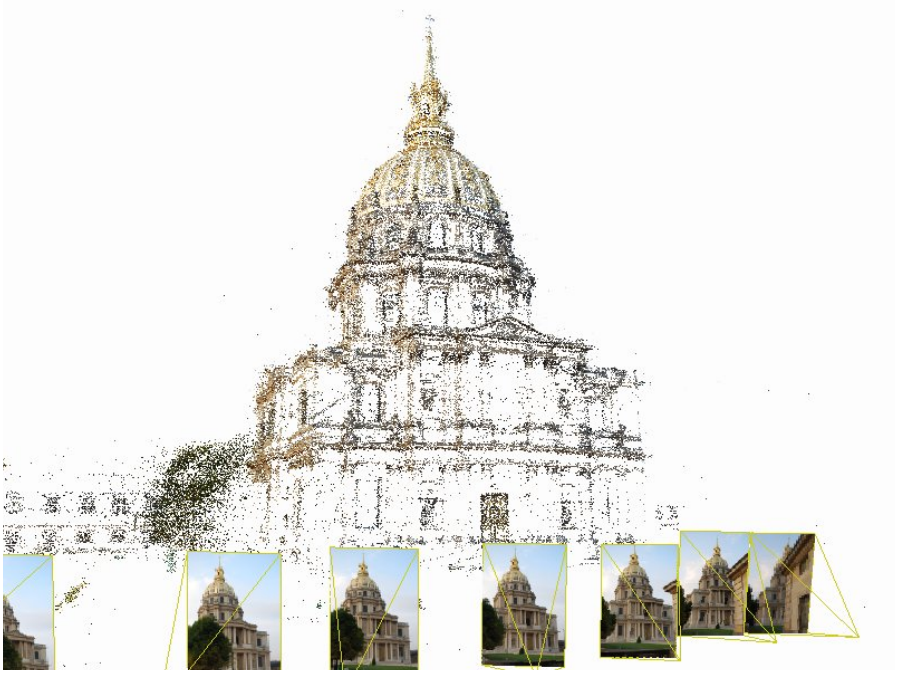


Figure 2.8: Example of a Structure from Motion pipeline.

does a "measurement update" or correction. The *a posteriori* state estimate $\mathbf{s}_t$ and its covariance matrix $\mathbf{S}_t$ are now generated by adding the measurements $\mathbf{z}_t$

$$\begin{aligned} \mathbf{s}_t &= \mathbf{s}_t^- + \mathbf{G}_t(\mathbf{z}_t - \mathbf{G}\mathbf{s}_t^-), & (2.19) \\ \mathbf{S}_t &= \mathbf{S}_t^- - \mathbf{G}_t\mathbf{C}\mathbf{S}_t^-, & (2.20) \end{aligned}$$

where the Kalman gain $\mathbf{G}_t$ is computed as

$$\mathbf{G}_t = \mathbf{S}_t^-\mathbf{C}^T(\mathbf{C}\mathbf{S}_t^-\mathbf{C}^T + \mathbf{\Lambda_v})^{-1}, \tag{2.21}$$

with $\mathbf{\Lambda_v}$ being the measurements covariance matrix.

In the context of 3D tracking, the *a priori* state estimate $\mathbf{s}_t^-$ can be used to predict the camera extrinsic parameters, therefore, the predicted measurement vector $\mathbf{z}_t^-$ is the following

$$\mathbf{z}_t^- = \mathbf{C}\mathbf{s}_t^-, \tag{2.22}$$

The uncertainty on the prediction is represented by the covariance matrix $\mathbf{\Lambda_z}$ estimated by propagating the uncertainty

$$\mathbf{\Lambda_z} = \mathbf{C}\mathbf{S}_t^-\mathbf{C}^T + \mathbf{\Lambda_v}, \tag{2.23}$$

2.10 Structure-from-Motion: BUNDLER

‘Structure-from-Motion’ (SfM) operates under the same basic assumptions as stereoscopic photogrammetry, basically that 3-D structure can be resolved from a series of overlapping images. However, it

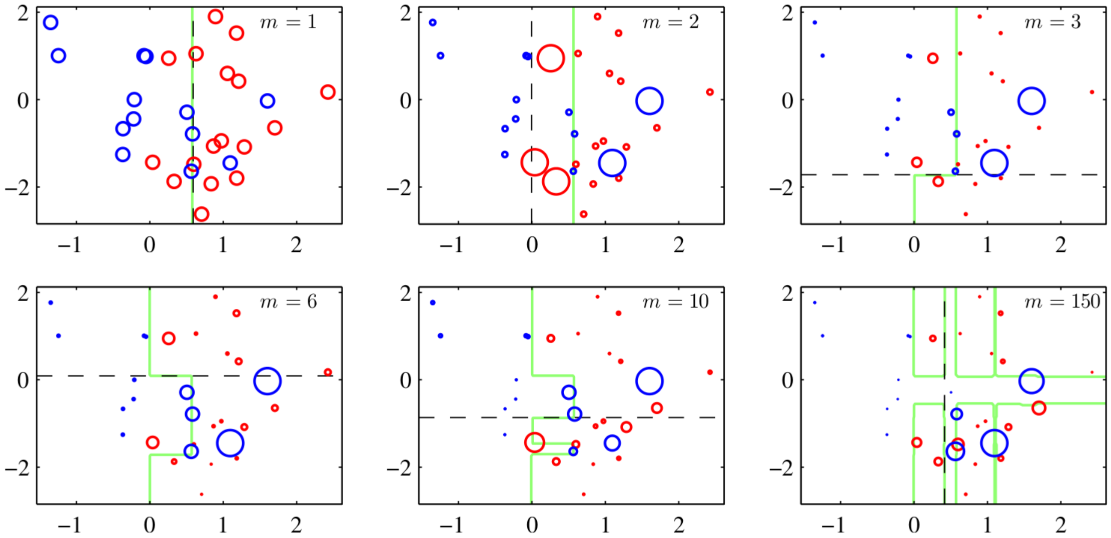


Figure 2.9: Illustration of boosting in which the base learners consist of simple thresholds applied to one or other of the axes. Each figure shows the number m of base learners trained so far, along with the decision boundary of the most recent base learner (dashed black line) and the combined decision boundary of the ensemble (solid green line). Each data point is depicted by a circle whose radius indicates the weight assigned to that data point when training the most recently added base learner. Thus, for instance, we see that points that are misclassified by the m = 1 base learner are given greater weight when training the m = 2 base learner.

differs fundamentally from conventional photogrammetry, in that the geometry of the scene, camera positions and orientation is solved automatically without the need to specify a priori, a set of targets which have known 3-D positions. Instead, these are solved simultaneously using a highly redundant, iterative bundle adjustment procedure, based on a database of features automatically extracted from a set of multiple overlapping images. In this thesis we use a solution to SfM called BUNDLER [95], it introduces most of the advances developed for SfM and at the same time makes a strong effort to give a system as general and flexible as possible. In Fig. 2.8 we can see an example of a reconstructed building and the associated images with their estimated poses.

2.11 Boosting: AdaBoost

Boosting involves training multiple models in sequence in which the error function used to train a particular model depends on the performance of the previous models. This can produce substantial improvements in performance compared to the use of a single model. Instead of averaging the predictions of a set of models, an alternative form of model combination is to select one of the models to make the prediction, in which the choice of model is a function of the input variables. Thus different models become responsible for making predictions in different regions of the input space.

Boosting can give good results even if the base classifiers have a performance that is only slightly better than random, and hence sometimes the base classifiers are known as weak learners. The most widely used form of boosting is AdaBoost, short for ‘adaptive boosting’, developed by Freund and Schapire [30]. The AdaBoost algorithm is illustrated in Fig. 2.9 using a set of 30 data points. Here each base learner consists of a threshold on one of the input variables. This simple classifier corresponds to a form of decision tree known as a ‘decision stumps’, i.e., a decision tree with a single node. Thus, each base learner classifies an input according to whether one of the input features exceeds some threshold and therefore simply partitions the space into two regions separated by a linear decision surface that is parallel to one of the axes.

Chapter 3

Uncalibrated Pose Estimation from Known Point Correspondences

3.1 Introduction

Estimating the camera pose from n 3D-to-2D point correspondences is a fundamental and well-understood problem in computer vision. Its solution is relevant to almost every application of computer vision in the era of smart phones. The most general version of the problem requires estimating the six degrees of freedom of the pose and five calibration parameters: focal length, principal point, aspect ratio and skew. This can be established with a minimum of 6 correspondences, using the well known Direct Linear Transform (DLT) algorithm [36].

There are, though, several simplifications to the problem which turn into an extensive list of different algorithms that improve the accuracy of the DLT. The most common simplification is to assume known calibration parameters. This is the so-called Perspective-n-Point problem, for which three point correspondences suffice in its minimal version [31]. There exist also iterative solutions to the over-constrained problem with $n > 3$ point correspondences [17, 39, 61] and non-iterative solutions that vary in computational complexity and accuracy from $O(n^8)$ [1] to $O(n^2)$ [25] down to $O(n)$ [58].

For the uncalibrated case, given that modern digital cameras come with square pixel size and principal point close to the image center [9, 36], the problem simplifies to the estimation of only the focal length. Solutions exist for the minimal problem with unknown focal length [7, 53, 96, 104], and for the case with unknown focal length plus unknown radial distortion [9, 10, 48, 104].

Unfortunately, in the presence of noise and mismatches, these solutions to the minimal problem become unstable and may produce unreliable pose estimates. This is commonly addressed including an extra RANSAC [26] iterative step for outlier removal, either taking minimal or non-minimal subsets [99], but at the expense of high computational load. Recent approaches have reformulated the problem as a quasi-convex optimization problem, allowing for the estimation of global minima [13, 49, 50]. Yet, while this is a very attractive idea, the iterative nature of these approaches makes them unpractical for real-time applications, unless a very small number of correspondences is considered.

In this thesis we advocate for an efficient solution that can handle an arbitrarily large point sample, thus increasing its robustness to noise. Using a large point set may be especially useful for current applications such as 3D camera tracking [57] or structure-from-motion [109], which require dealing with hundreds of noisy correspondences in real time.

The method we propose fulfills these requirements: it allows estimating pose and focal length in bounded time, and since it is a non-minimal solution, it is robust to situations with large amounts of noise

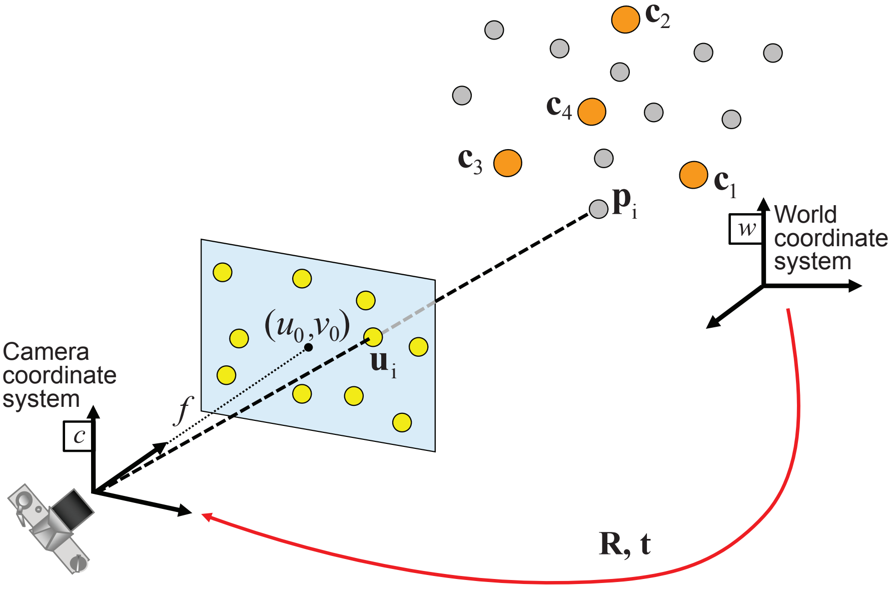


Figure 3.1: **Problem Formulation:** Given a set of correspondences between 3D points $\mathbf{p}_i$ expressed in a world reference frame, and their 2D projections $\mathbf{u}_i$ onto the image, we seek to retrieve the pose ($\mathbf{R}$ and $\mathbf{t}$) of the camera w.r.t. the world and the focal length f.

in the input data. Drawing inspiration on the EPnP algorithm [58, 69], we show that the solution of our problem belongs to the kernel of a matrix derived from the 3D-to-2D correspondences, and thus can be expressed as a linear combination of its eigenvectors. The weights of this linear combination become the unknowns of the problem, which we solve applying additional distance constraints.

However, solving also for the focal length has the effect that the linearization and relinearization techniques used in [58, 69] to estimate these weights are no longer valid. Several factors contribute to this: (1) the new polynomials that need to be considered are of degree four, in contrast to those in the EPnP that were of degree two; (2) the variables being computed differ in several orders of magnitude and small inaccuracies in the input data may propagate to large errors in the estimation; and (3) the number of possible combinations in the solution subspace explodes combinatorially for large kernel sizes. All these issues make that a naïve selection of equations for back substitution after linearization produces unreliable results. Moreover, a least squares solution of the kernel weights is also not viable since it will equally ponder constraints that involve variables with different orders of magnitude. We propose alternative solutions, which we call exhaustive linearization and exhaustive relinearization that circumvent these limitations by systematically exploring the solution subspace.

As will be shown in the results section, our method, called Uncalibrated PnP (UPnP), compares favorably in terms of accuracy to the DLT algorithm, the only closed-form solution we are aware that is applicable for an arbitrary number of correspondences. This is because the least squares solution of the DLT algorithm chooses an optimal solution only in the direction along the vector associated with the smallest singular value of the linear system of equations built from the 3D-to-2D correspondences. In contrast, our method considers all directions of the kernel of the system, which for the ideal case is of size one [89], but for noisy overconstrained systems grows in size [58]. Our method also yields better accuracy and efficiency than [49] and [50], which are algorithms that guarantee maximum error tolerance, but which are computationally expensive. In fact, the accuracy of our results is even comparable with that of the EPnP, which assumes known calibration parameters.

3.2 Problem Formulation

In this section we formulate the problem of recovering the camera pose and focal length from a set of n 3D-to-2D point correspondences. We first show that these matches yield a rank-deficient linear system, which requires additional constraints to be solved. We then introduce distance constraints that convert the original linear system into a set of polynomial equations of degree four. In Sec. 3.3 we introduce novel linearization techniques that help solve this polynomial set of equations.

3.2.1 Linear Formulation of the Problem

We assume that we are given a set of 3D-to-2D correspondences between n reference points $\mathbf{p}_1^w, \ldots, \mathbf{p}_n^w$ expressed in a world coordinate system w, and their 2D projections $\mathbf{u}_1, \ldots, \mathbf{u}_n$ in the image plane. We further assume a camera with square pixel size and with the principal point (u_0, v_0) at the center of the image, although we do not know its focal length. Under these assumptions, we formulate the problem as that of retrieving the focal length f of the camera, and the rotation $\mathbf{R}$ and translation $\mathbf{t}$, that align the world and the camera coordinate systems (see Fig. 3.1).

We will address this problem by minimizing the following objective function based on the reprojection error:

$$\underset{f,\mathbf{R},\mathbf{t}}{\text{minimize}} \sum_{i=1}^{n} \|\mathbf{u}_i - \tilde{\mathbf{u}}_i\|^2 \ , \tag{3.1}$$

where $\tilde{\mathbf{u}}_i$ is the projection of point $\mathbf{p}_i^w$:

$$k_i \begin{bmatrix} \tilde{\mathbf{u}}_i \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix} [\mathbf{R}|\mathbf{t}] \begin{bmatrix} \mathbf{p}_i^w \\ 1 \end{bmatrix} \ , \tag{3.2}$$

with k_i a scalar projective parameter.

Following [58], we rewrite each 3D point in terms of the barycentric coordinates of 4 control points. This turns the problem into that of finding the solution of a linear system of $2n$ equations in 12 unknowns.

Let $\mathbf{c}_j^w, j = 1, \ldots, 4$, be the coordinates of these four control points defining an arbitrary basis in the world coordinate system. Without loss of generality, we choose this basis to be centered at the mean of the reference points and aligned with the principal directions of the data. Each reference point $\mathbf{p}_i^w$ then becomes

$$\mathbf{p}_i^w = \sum_{j=1}^{4} a_{ij} \mathbf{c}_j^w \ . \tag{3.3}$$

The a_{ij} terms indicate the barycentric coordinates of the i-th reference point and may be computed from the position of the reference and control points in the world coordinate system, with the normalization constraint that $\sum_{j=1}^{4} a_{ij} = 1$. Note that these barycentric coordinates are independent from the coordinate system we use, i.e., the same points in the camera referential c become $\mathbf{p}_i^c = \sum a_{ij}\mathbf{c}_j^c$.

Therefore, replacing $\mathbf{R}\mathbf{p}_i^w + \mathbf{t}$ with $\mathbf{p}_i^c$ into Eq. 3.2, produces the two following perspective projection equations for each 3D-to-2D correspondence:

$$\sum_{j=1}^{4}\left(a_{ij}x_j^c + a_{ij}(u_0 - u_i)\frac{z_j^c}{f}\right) \quad = \quad 0\,, \tag{3.4}$$

$$\sum_{j=1}^{4}\left(a_{ij}y_j^c + a_{ij}(v_0 - v_i)\frac{z_j^c}{f}\right) \quad = \quad 0\,, \tag{3.5}$$

where $\mathbf{u}_i = [u_i, v_i]^\top$ and $\mathbf{c}_j^c = [x_j^c, y_j^c, z_j^c]^\top$. These equations can be jointly expressed for all the n correspondences as a linear system

$$\mathbf{M}\mathbf{x} = \mathbf{0}\,, \tag{3.6}$$

where $\mathbf{M}$ is a $2n \times 12$ matrix made of the coefficients a_{ij}, the 2D points $\mathbf{u}_i$, and the principal point; and $\mathbf{x}$ is our vector of 12 unknowns containing both the 3D coordinates of the control points in the camera reference frame and the camera focal length, dividing the z terms:

$$\mathbf{x} = [x_1^c, y_1^c, z_1^c/f, \ldots, x_4^c, y_4^c, z_4^c/f]^\top\,. \tag{3.7}$$

Note that by using the barycentric coordinates we have converted the pose estimation problem to that of estimating the position of the four control points $\mathbf{c}_i^c$ in the camera coordinate system. The two problems, though, are equivalent, since given $\mathbf{c}_i^w$ and $\mathbf{c}_i^c$, we can then apply standard techniques to compute the orientation and translation between the world and camera referentials [40].

Equation 3.6 tells us that the solution lies on the null-space of $\mathbf{M}$. We can therefore write $\mathbf{x}$ as a weighted sum of the null eigenvectors $\mathbf{v}_k$ of $\mathbf{M}^\top\mathbf{M}$, which can be computed using Singular Value Decomposition (SVD). Hence, we write

$$\mathbf{x} = \sum_{k=1}^{N}\beta_k\mathbf{v}_k\,, \tag{3.8}$$

where the weights β_k become our new unknowns and N is the rank of the kernel of $\mathbf{M}^\top\mathbf{M}$. It can be shown that, for $n \geq 6$, and with noise-free correspondences, $N = 1$. In practice, though, noise makes no eigenvalue exactly zero and the matrix $\mathbf{M}^\top\mathbf{M}$ has full rank. Nonetheless, the matrix loses rank numerically and the effective dimension of the null space increases. Thus, we have to consider the effective dimension of the kernel being greater than one, and to cope with this situation, we follow a similar strategy as in [58], and compute the solution for various values of N, picking the one that minimizes Eq. 3.1. There is no clear criterion on the value of N to choose, as this will depend on the focal length magnitude and on the amount of noise in our input data. Yet, setting $N \leq 3$ has proven adequate in all our experiments.

3.2.2 Introducing Distance Constraints

In order to solve for the weights β_k in Eq. 3.8 we add constraints that preserve the distance between control points. That is, for each pair of control points $\mathbf{c}_j$ and $\mathbf{c}_{j'}$,

$$\|\mathbf{c}_j^c - \mathbf{c}_{j'}^c\|^2 = d_{jj'}^2\,, \tag{3.9}$$

where $d_{jj'}$ is the known distance between control points $\mathbf{c}_j^w$ and $\mathbf{c}_{j'}^w$ in the world coordinate system. Rewriting $\mathbf{c}_j^c$ and $\mathbf{c}_{j'}^c$ in terms of the β_k coefficients, from Eqs. 3.7 and 3.8 we obtain

$$\mathbf{c}_j^c = \begin{bmatrix} x_j^c \\ y_j^c \\ z_j^c \end{bmatrix} = \sum_{k=1}^{N}\begin{bmatrix} \beta_k v_{k,x}^{[j]} \\ \beta_k v_{k,y}^{[j]} \\ f\beta_k v_{k,z}^{[j]} \end{bmatrix}\,, \tag{3.10}$$

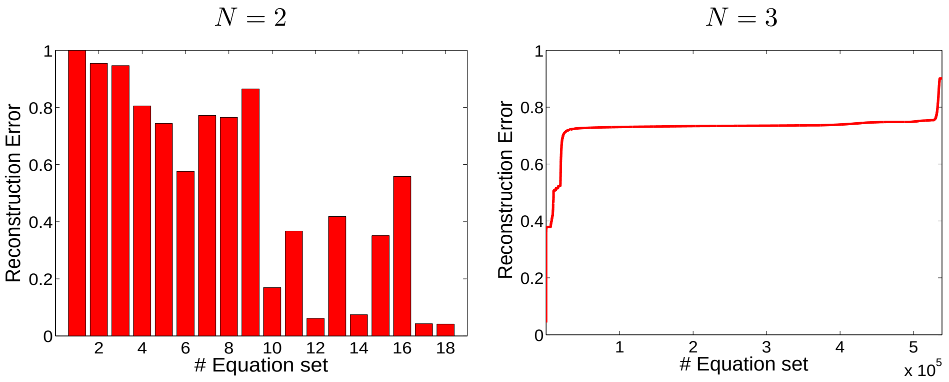


Figure 3.2: Reconstruction error for all possible equation sets. The graphs plot the mean normalized reconstruction error over 1000 different experiments with random input correspondences, and different amounts of noise. **Left.** All 18 triplet combinations for a kernel of size $N = 2$. **Right.** All 538.704 quadruplet combinations for a kernel of size $N = 3$. In this case, the reconstruction errors have been sorted in increasing order of magnitude for viewing purposes. Observe that in both cases, the selection of one set of equations from another, results in significantly different reconstruction error (and hence pose and focal length estimation).

where $\mathbf{v}_k^{[j]} = [v_{k,x}^{[j]}, v_{k,y}^{[j]}, v_{k,z}^{[j]}]^\top$ is the sub-vector of $\mathbf{v}_k$ corresponding to the coordinates of the j-th control point. Observe that the unknown focal length has been moved to the right-hand side of Eq. 3.8 and now multiplies the z component of the control points. As a consequence, applying the distance constraints between all pairs for control points will now generate 6 polynomials of degree 4, in contrast to the quadratic equations appearing in the original EPnP formulation. As we will see in the following sections this will require a substantially different approach, especially when solving the cases $N = 2$ and $N = 3$.

3.3 Exhaustive Linearization and Relinearization

In this section we introduce novel closed-form linearization techniques to solve the systems of polynomial equations which result from combining Eq. 3.8 and the six distance constrains of Eq. 3.9. We will see that a standard linearization approach is only effective to solve the case $N = 1$ (when only two variables need to be estimated), but it fails to solve the cases $N = 2$ and $N = 3$, in which we have a larger number of unknowns while the number of equations remains the same.

3.3.1 Case $N = 1$: Linearization

For the case where $N = 1$, we only need to solve for β_1 and f. This case may be solved by simply linearizing the system of equations and introducing new unknowns for the quadratic and bi-quadratic terms. In particular, we will use $\beta_{11} = \beta_1^2$, and $\beta_{ff11} = f^2\beta_1^2$. Applying the six distance constraints from Eq. 3.9 between all pairs of control points, results in a system of the form

$$\mathbf{L}\mathbf{b} = \mathbf{d} \ , \tag{3.11}$$

where $\mathbf{b} = [\beta_{11}, \beta_{ff11}]^\top$ and $\mathbf{L}$ is a 6×2 matrix built from the known elements of $\mathbf{v}_1$, and $\mathbf{d}$ is a 6-vector of squared distances between the control points. We solve this overdetermined linearized system using least squares and estimate the magnitudes of β_1 and f by back substitution:

$$\beta_1 = \sqrt{\beta_{11}} \qquad f = \sqrt{|\beta_{ff11}|/|\beta_1|} \tag{3.12}$$

Finally, we select the sign of β_1 such that after computing the pose, all the points end up placed in front of the camera.

3.3.2 Case $N = 2$: Exhaustive Linearization

For the case $N = 2$ we need to solve for β_1, β_2 and f. Applying the six distance constraints we obtain again a linear system $\mathbf{Lb} = \mathbf{d}$, where now $\mathbf{L}$ is a 6×6 matrix built from substituting the known elements of the basis $\mathbf{v}_1$ and $\mathbf{v}_2$ into Eq. 3.9. The number of unknowns becomes a six dimensional vector

$$\mathbf{b} = [\beta_{11}, \beta_{12}, \beta_{22}, \beta_{ff11}, \beta_{ff12}, \beta_{ff22}]^\top. \tag{3.13}$$

Note that the entries in $\mathbf{L}$ become quadratic expressions on the elements of the orthogonal basis vectors $\mathbf{v}_1$ and $\mathbf{v}_2$, and since these are made of control points which are by construction different from each other, $\mathbf{L}$ has full rank. Following a similar procedure as before, we can thus retrieve the vector of linear unknowns $\mathbf{b}$ computing the inverse of $\mathbf{L}$.

However, the simple backsubstitution scheme used to solve for each of the individual unknowns as in Eq. 3.12 is no longer valid. In fact, by simple observation of the vector $\mathbf{b}$ it can be seen that the individual variables may be computed, once $\mathbf{b}$ is known, applying backsubstitution over 18 different triplets, namely $(\beta_{11}, \beta_{12}, \beta_{ff11})$, $(\beta_{11}, \beta_{12}, \beta_{ff12})$, $(\beta_{11}, \beta_{12}, \beta_{ff22})$, $(\beta_{11}, \beta_{22}, \beta_{ff11})$ and so on. It turns out that in the absence of noise, all these triplets render the same solution, but when noise comes into play, each of the triplets has a different effect on the solution. This is depicted in Fig. 3.2-left, where we plot the mean reconstruction error of the solution obtained with each triplet, computed as the mean Euclidean distance between the 3D points aligned with respect to the ground truth camera coordinate system, and the same 3D points aligned using the estimated pose and focal length.

To choose the right equation set we propose what we call an *exhaustive linearization*, which is a strategy that generates and explores all possible triplets, and takes the one that minimizes the reprojection error of Eq. 3.1. Note that the number and form of each triplet is always the same, and independent of the input data. Therefore, this exploration can be efficiently executed in parallel.

To solve the monomial quadratic terms we rewrite bilinearities as logarithmic sums. That is, by applying logarithms on the absolute values of all the elements within the triplet, we can rewrite the terms β_{ij} as equations of the form $\log|\beta_{ij}| = \log|\beta_i| + \log|\beta_j|$. Doing this for all elements within the triplet produces a linear system of 3 equations and 3 unknowns that yields the magnitude of each individual variable. To determine the sign of each variable we check sign consistency with the components of $\mathbf{b}$ that have not been used, and also enforce the geometric constraints of positive focal length and 3D point location in front of the camera.

3.3.3 Case $N = 3$: Exhaustive Relinearization

For the case of $N = 3$ we need to solve for β_1, β_2, β_3 and f. Unfortunately, neither the linearization nor the exhaustive linearization techniques suffice to address this case, because the number of quadratic unknowns in the linearized system is larger than the number of equations. We have 12 linearized terms of the form β_{kl} and β_{ffkl} with k and $l \in \{1, 2, 3\}$, while the number of distance constraints remains equal to six. We solve this problem by using a relinearization technique [51] in conjunction with our exhaustive strategy described above. We call the combination of both methods as *exhaustive relinearization*.

The idea of the relinearization technique is to add constraints that enforce the algebraic nature of the elements β_{kl} and β_{ffkl}. We start by considering the following homogeneous linear system:

$$[\mathbf{L}| - \mathbf{d}] \begin{bmatrix} \mathbf{b} \\ \rho \end{bmatrix} = \mathbf{0} \quad \Rightarrow \quad \tilde{\mathbf{L}}\tilde{\mathbf{b}} = \mathbf{0} \ , \tag{3.14}$$

where $\tilde{\mathbf{L}}$ is now a 6×13 matrix, $\tilde{\mathbf{b}}$ is a 13-vector including the quadratic and biquadratic unknowns, and ρ is a scaling factor. The solution for $\tilde{\mathbf{b}}$ is then spanned by the null space of $\tilde{\mathbf{L}}$. That is,

$$\tilde{\mathbf{b}} = \sum_{i=1}^{M} \lambda_i \tilde{\mathbf{w}}_i \ , \tag{3.15}$$

where $\tilde{\mathbf{w}}_i$ are the right singular vectors of $\tilde{\mathbf{L}}$. As in the case $N = 2$, $\tilde{\mathbf{L}}$ is of rank 6 by construction, and thus $M = 7$. Finally, we solve for the λ_i-s setting $\rho = 1$ to remove the scale ambiguity, and using additional constraints coming from the commutativity of the multiplication of the β_{kl} and β_{ffkl} monomials, e.g.,

$$\beta_{klmf} = \beta_{kl}\beta_{mf} = \beta_{k'l'm'f'} \ , \tag{3.16}$$

where (k', l', m', f') represents any permutation of (k, l, m, f). After imposing these constraints, the coefficients λ_i are solved using linearization, and thus the name *relinearization*.

However, this second linearization suffers again from the problem we mentioned above for the case $N = 2$. That is, the coefficients λ_i may be retrieved from small sets of quadratic monomials $\lambda_{ij} = \lambda_i\lambda_j$, but due to noise, choosing each of these sets produces a different reprojection error, which is the function we are trying to minimize. Hence, we need to perform again an exploration of the possible minimal sets of λ_{ij} vectors. In addition, once the coefficients $\lambda_1, \ldots, \lambda_M$ have been recovered, we need to retrieve the coefficients β_1, β_2, β_3 and f by exploring the possible minimal sets of β_{kl} vectors. To filter out parasitic solutions we impose the additional constraints $\beta_{ii}\beta_{jk} = \beta_{ij}\beta_{ik}$.

Efficient Exploration of the Minimal Equation Sets

Note that the number of all possible sets of equations we have to explore grows exponentially with M. In our experiments we have observed that it is sufficient to explore only up to the 5th singular vector of $\tilde{\mathbf{L}}$, which produces 1548 different equation sets from which to retrieve the λ's, and for each of them we have 348 quadruplets from which to retrieve the β's. This yields a total of 538.704 possible combinations to explore. Exploring all possible combinations is computationally expensive (in the order of minutes on a standard PC). Fortunately, the right equation set to choose does not heavily depend on the point configuration nor the value of the focal length, more than on the algebraic combination of variables. For this reason, we devised a strategy to select off-line the best equation set from a large number (10^3) of synthetic experiments, without jeopardizing the computational efficiency of the overall method at run time.

The idea is to order the equation sets according to their weighted contribution in solving all experiments in the large training session. To do this, we run the complete algorithm over synthetic random input data and assign to each equation set Q_i, a weight inversely proportional to the cumulative reconstruction error throughout all experiments. Fig. 3.2-right illustrates the normalized error distribution for each equation ordered using this weight.

At run time, this ordering is used to test each equation set searching for the one that minimizes Eq. 3.1, as shown in Alg. 2. Only in those cases when the reprojection is still not good enough (above a threshold E_{min}) for the cases $N = 1$ and $N = 2$, we enter the case $N = 3$ and iterate over the ordered list of equation sets, compute the λ's, β's and f for each set, and use these parameters to recover the pose parameters $\mathbf{R}$ and $\mathbf{t}$. The solution is updated should it improve the reprojection error. A stopping condition is set after exploring a reduced number of equation sets or once the reprojection error falls below E_{min}.

```
EXPLORESET($\mathbf{p}$,$\mathbf{u}$,$E_1$,$E_2$)
    INPUT:
        $\mathbf{p}$:  3D points.
        $\mathbf{u}$:  image correspondences.
        $E_1$: reprojection error for the case $N = 1$.
        $E_2$: reprojection error for the case $N = 2$.
    OUTPUT:
        $\mathbf{t}^*$:  Camera translation.
        $\mathbf{R}^*$: Camera rotation.
        $f^*$:  focal length.

$E^* \leftarrow \infty$
if $E_1 > E_{\text{min}}$ and $E_2 > E_{\text{min}}$ then
    for each equation set $\mathcal{Q}_i$ in decreasing rank order do
        $(\lambda' s, \beta' s, f) \leftarrow$ EXHAUSTIVERELINEARIZATION($\mathbf{p}$,$\mathbf{u}$,$\mathcal{Q}_i$)
        $(\mathbf{R},\mathbf{t}) \leftarrow$ RECOVERPOSE($\mathbf{p}$,$\beta' s$,$f$)
        $E \leftarrow$ REPROJECTIONERROR($\mathbf{p}$,$\mathbf{R}$,$\mathbf{t}$,$f$)
        if $E < E^*$ then
            $E^* \leftarrow E$, $\mathbf{R}^* \leftarrow \mathbf{R}$, $\mathbf{t}^* \leftarrow \mathbf{t}$, $f^* \leftarrow f$
        end if
        if $i > i_{\text{max}}$ or $E^* \leq E_{\text{min}}$ then
            RETURN($\mathbf{R}^*$, $\mathbf{t}^*$, $f^*$)
        end if
    end for
end if
```

Algorithm 2: Algorithm to explore the set of equations in the case $N = 3$.

The parameter i_{max} defines the maximum number of equation sets to explore, and thus it is an upper bound in the time required by our algorithm. This parameter offers a trade-off between efficiency and optimality. While the computation time grows linearly with i_{max}, the residual error of the minimization rapidly falls after just a few iterations. In practice, as shown in the next section, by setting the maximum number of equations to validate to 500, the accuracy results are comparable to that of the calibrated case, while maintaining computational efficiency. In addition, E is usually good enough for the cases $N = 1$ or $N = 2$, preventing from having to evaluate the case $N = 3$ at all, a situation that happens roughly 80% of the time for noise levels between 1 and 3 pixels.

3.3.4 Why Using Minimal Sets of Equations?

One question that may naturally arise from our methodology is why exploring minimal sets of equations (triplets for solving the case $N = 2$ and quadruplets for the case $N = 3$). As an alternative to this, we could have also tried to take the logarithms of all the elements of the vector $\mathbf{b}$, and use least squares over the resulting overdetermined system to retrieve the variables $\log|\beta_i|$ and $\log f$. However, although this solution is faster than independently evaluating triplets or quadruplets and retaining the solution with minimum reprojection error, it is far less accurate. The reason is that the algebraic combination of variables with severe differences in order of magnitude, weights binomials that include focal length more heavily than other binomials, and a least squares solution would wrongly average such inconsistencies.

To see this effect, we compare our Exhaustive Linearization and Exhaustive Relinearization approaches, to linearization and relinearization implementations that use least squares to solve for the β's and λ's. Fig. 3.3 compares both alternatives in an experiment where pose and focal length are computed

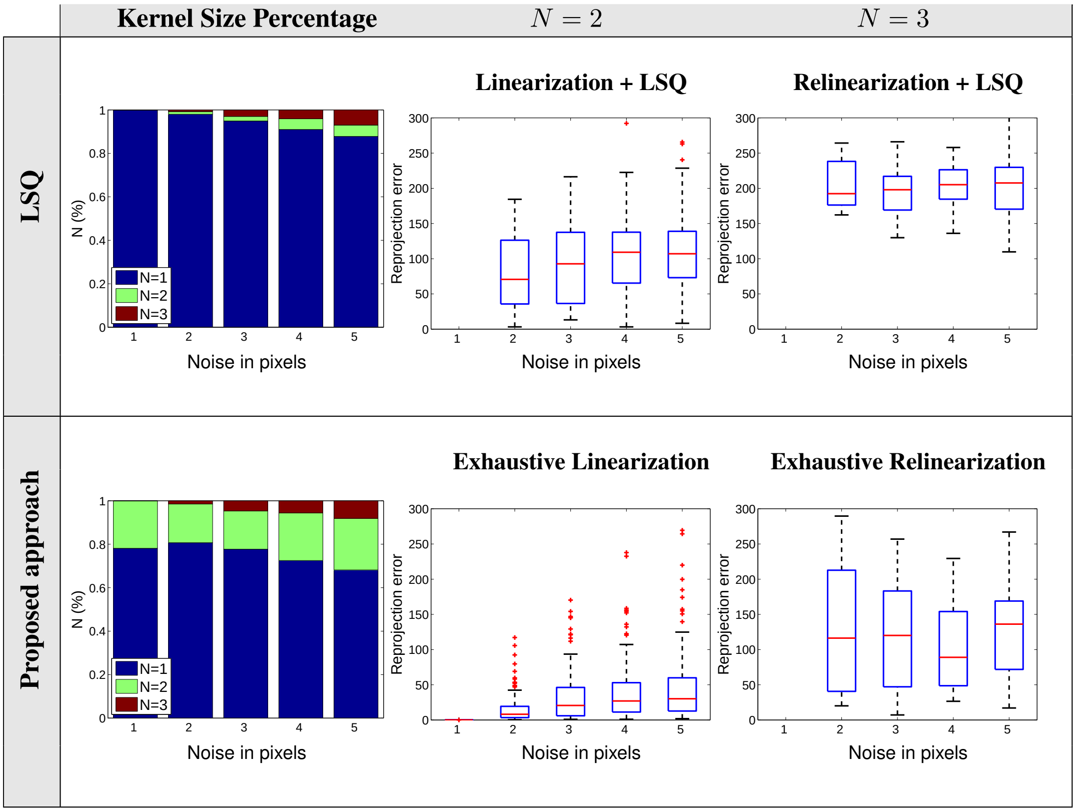


Figure 3.3: Comparison of our approach that estimates the parameters via backsubstitution over minimal equation sets, against an approach that estimates these via backsubstitution over a least squares approximation using all equations for a set of 1000 experiments and varying image noise. **Left column:** Effective number N of null eigenvalues of $\mathbf{M}^\top\mathbf{M}$. **Middle and right columns:** Reprojection error distributions for the $N = 2$ and $N = 3$ cases. The box edges in the boxplots denote first and third quartiles, red lines inside the boxes indicates medians, dashed lines represent the extent of the statistical data, and red crosses are outliers.

for $n = 6$ random 3D-to-2D correspondences with increasing amounts of noise in a 640×480 image. The leftmost plots show the effective number N of null singular values in $\mathbf{M}^\top\mathbf{M}$, i.e., the percentage of solutions in which the minimal reprojection error has been obtained for each specific value of N. Note that for $N = 1$ neither linearization nor relinearization come into play. The differences between the different methodologies can only be assessed for those cases in which $N = 2$ or $N = 3$ improve the solution obtained with $N = 1$. Both the effective number N and the distribution of the reprojection error for the cases $N = 2$ and $N = 3$ (Fig. 3.3 middle and right), indicate that our approach clearly outperforms the methods using least squares.

This result was in fact expected because the noise in the input 3D-to-2D correspondences is not homogeneously propagated through the SVD decomposition and linearization processes, and as seen in Fig. 3.2, it results in equation sets with very different accuracies. Simultaneously handling all equation sets in a least squares sense does not allow to filter out these large variations, and is only using a robust method like the algorithm we proposed in the previous section that we can optimally search for the right values for the β's and λ's.

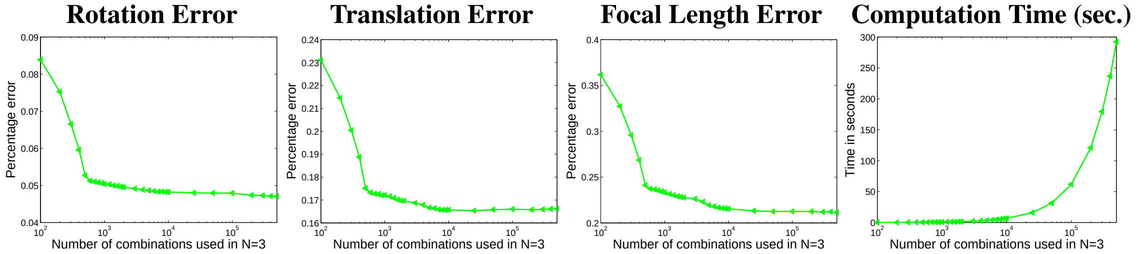


Figure 3.4: Mean rotation, translation and focal length estimation errors when $N = 3$ is selected as the best solution (we refer the reader to Sec. 3.4.1 for a precise definition of these errors), and computation time for an increasing number of equation sets. Note that the horizontal axis is plotted in logarithmic scale, and the time scales linearly with the number of equation sets. Exploring 500 equation sets is a good tradeoff between accuracy and computation time. These graphs are generated with random experiments with $n = 7$ points, and large amounts of 2D noise (at the level of $\sigma_n = 5$), in order to ensure that exploration of the case $N = 3$ was meaningful.

3.3.5 Dealing with Planar Configurations

Like the EPnP algorithm [58], our approach can be easily adapted to address situations in which the 3D points lie on a plane. For these configurations, the n 3D reference points can be spanned using only three control points –instead of four–. The 3D to 2D projection of the point correspondences may then be written as a linear system equivalent to that of Eq. 3.6, but with a different dimensionality. Now, the matrix $\mathbf{M}$ of coefficients will be $2n \times 9$, and the vector of unknowns will contain the focal length and the coordinates of only three control points, $\mathbf{x} = [x_1^c, y_1^c, z_1^c/f, \ldots, x_3^c, y_3^c, z_3^c/f]^\top$.

We will solve this homogeneous linear system by independently resolving specific dimensionalities of the kernel of $\mathbf{M}^\top\mathbf{M}$, as in the non-planar case. However, note that when using three control points, we can only define up to three constraints based on their inter-distances. These three equations will not be sufficient to solve for the six unknowns of the vector $\mathbf{b}$ in Eq. 3.13, for the case $N = 2$. As a consequence, for $N \geq 2$, we will need to make use of the additional equations provided by the extended relinearization technique explained above.

3.3.6 Iterative Refinement

Although the exhaustive linearization and relinearization techniques perform a sequential exploration of the collection of equation sets, the spirit of the whole algorithm is still non-iterative, as no initialization is required and the exploration can be performed in bounded time. We will now feed this result into a final iterative stage that will increase the accuracy in the estimation of both the camera pose and focal length at a very small additional cost.

Following [58], we iterate over the parameters β_1, β_2, β_3, and f to solve the problem

$$\underset{\beta_1,\beta_2,\beta_3,f}{\text{minimize}} \sum_{(i,j)\text{ s.t. } i<j} (\|\mathbf{c}_i^c - \mathbf{c}_j^c\|^2 - d_{ij}^2) \tag{3.17}$$

where the d_{ij}'s are the known distances between control points in the world coordinate system and, following Eq. 3.10, the $\mathbf{c}_i^c$ are expressed in terms of the β_k coefficients and focal length f. Their values are initialized to those estimated using the exhaustive linearization approaches, or to zero when they are not available. That is, when the effective rank of $\mathbf{M}^\top\mathbf{M}$ is found to be $N = 1$, then β_2 and β_3 are initialized to zero. When the rank is found to be $N = 2$, only β_3 is set to zero. We then perform the minimization using a standard Gauss-Newton optimization.

Note that the minimization is performed over the four dimensional space of the β's and f coefficients, and not over the seven dimensional space of the pose and focal length. In addition, since in general the initialization provided by the linearization approaches is usually very accurate, the optimization typically converges in about 10 iterations. Overall, the impact of this refinement on the method's computational time is of less than 5% of the total time.

3.4 Experimental Results

In this section we compare the accuracy of our algorithm with and without the final Gauss Newton optimization (we denote these cases *UPnP+GN* and *UPnP*, respectively) against the *DLT* [36], and the approaches [49] and [50], which search for a global solution. The first of these methods, denoted by *L2-L2*, is based on a branch and bound strategy that minimizes the L_2 norm of the reprojection error. The approach described in [50] shows that replacing the L_2 norm by the L_∞ norm yields a convex formulation of the problem with a unique minimum which is retrieved using second-order cone programming. In the following we will denote this method by *Linf* [1]. Note that DLT, L2-L2 and Linf retrieve the complete 3×4 projection matrix $\mathbf{P}$, while our approach separately estimates the orientation $\mathbf{R}$, translation $\mathbf{t}$ and focal length f. In order to perform a fair comparison, given $\mathbf{P}$ we will first retrieve the calibration matrix $\mathbf{A}$, using a Cholesky factorization of $\mathbf{P}_3\mathbf{P}_3^\top$, where $\mathbf{P}_3$ is the left 3×3 submatrix of $\mathbf{P}$ [112]. We will then fix the principal point to the ground truth value and estimate $\mathbf{R}$ by ortho-normalizying $\mathbf{A}^{-1}\mathbf{P}_3$. The translation vector $\mathbf{t}$ is directly estimated from the last column of $\mathbf{P}$.

We also include the results of the EPnP [69], and the EPnP with a Gauss-Newton refinement [58]. For both these approaches the true focal length is provided and obviously work better than the uncalibrated methods. We plot them here as a reference baseline.

One parameter that needs to be chosen beforehand in our algorithm, is the maximum number i_{max} of equations we want to explore for the case $N = 3$. In Fig. 3.4 we plot the pose and focal length estimation errors as a function of the number of equations, when we enforce our algorithm to compute pose with only the case $N = 3$. In all cases we obtain reasonable results in relatively short time by exploring 500 sets of equations, which only represents a very small fraction of all possible 538.704 combinations. In the experiments, we will thus evaluate each of these situations, indicating the number of explored equations. When nothing is said, we will assume that only 500 equations are evaluated.

3.4.1 Non-Planar Synthetic Experiments

For the synthetic experiments, we simulated 3D-to-2D correspondences for sets of points of different size, uniformly distributed in the cube $[-2, 2] \times [-2, 2] \times [4, 8]$, and projected onto a 640×480 image using a virtual calibrated camera with squared pixels, and principal point at $(u_0, v_0) = (320, 240)$. Image points were corrupted with Gaussian noise.

For any given ground truth camera pose, $\mathbf{R}_{\text{true}}$ and $\mathbf{t}_{\text{true}}$, focal length f_{true}, and corresponding estimates $\mathbf{R}$, $\mathbf{t}$ and f, the relative rotation error was computed as $\text{E}_{\text{rot}} = \|\mathbf{q}_{\text{true}} - \mathbf{q}\|/\|\mathbf{q}\|$, where $\mathbf{q}$ and $\mathbf{q}_{\text{true}}$ are the normalized quaternions of $\mathbf{R}$ and $\mathbf{R}_{\text{true}}$, respectively; the relative translation error was computed with $\text{E}_{\text{trans}} = \|\mathbf{t}_{\text{true}} - \mathbf{t}\|/\|\mathbf{t}\|$; and the error in the estimation of the focal length was determined by $\text{E}_f = |f_{\text{true}} - f|/f$. All errors reported in this section correspond to average errors estimated over 100 experiments with random positions of the 3D points.

The first and second rows in Fig. 3.5 show the robustness of all methods against image noise. For these experiments the 2D coordinates of the matches were corrupted with additive Gaussian noise with

[1] For the L2-L2 method we have used the implementation from the Branch and Bound Optimization toolbox available at http://www.cs.washington.edu/homes/sagarwal/code.html. The code for the Linf method has been taken from the L-infinity toolbox available at http://www.maths.lth.se/matematiklth/personal/fredrik/download.html.

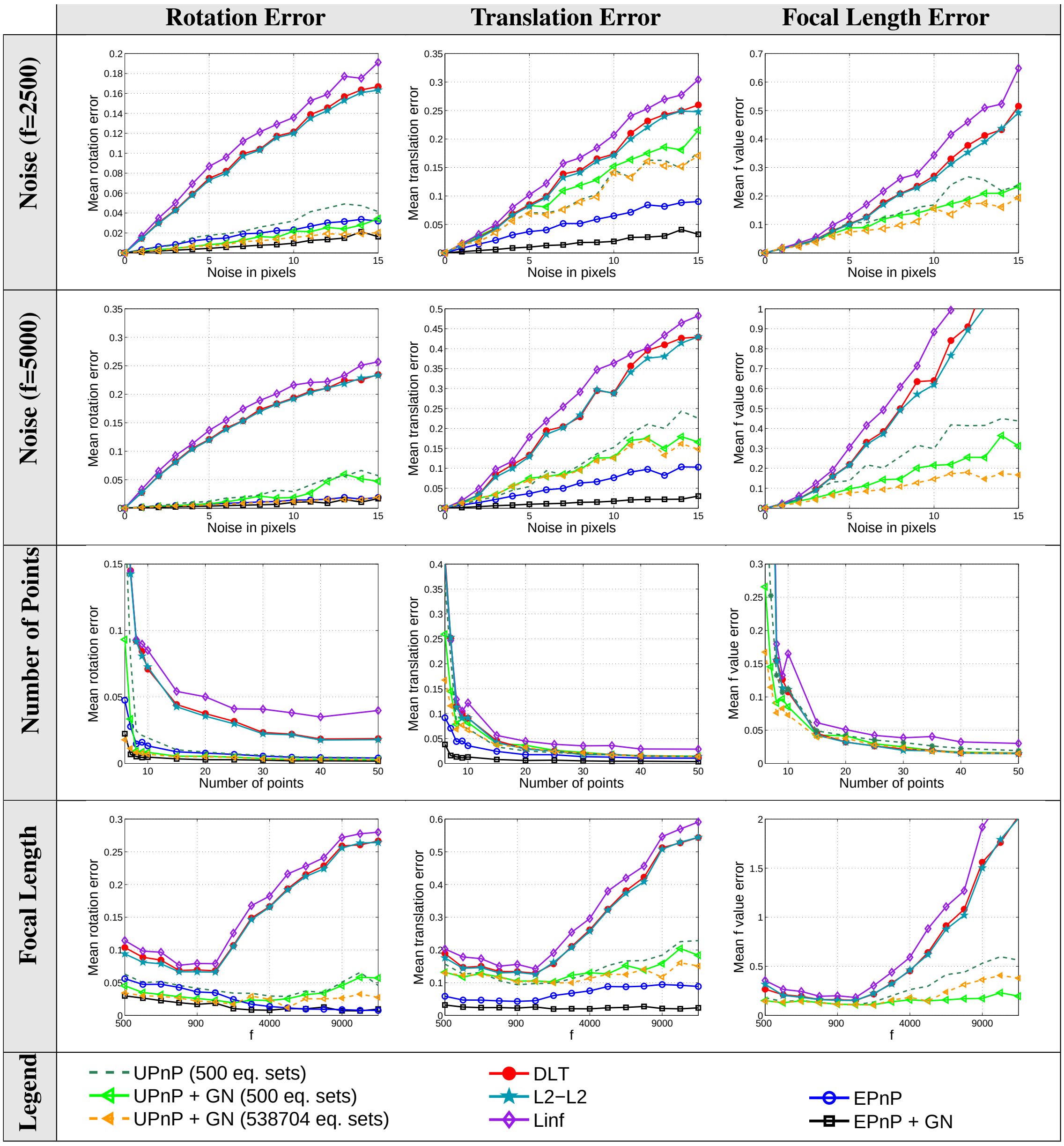


Figure 3.5: Results on synthetic data for non-planar distributions of points. **Two upper rows:** Mean rotation, translation and focal length errors for: increasing levels of image noise on 10 2D-3D correspondences, and two different focal lengths. **Third row:** increasing number of 2D-3D correspondences. **Fourth row:** increasing focal length, also for 10 point correspondences. Each tick in the plot represents the average over 100 experiments with random points.

a growing standard deviation σ up to 15 pixels, and the number of correspondences was set to $n = 10$. Observe that our approach performs consistently better than other uncalibrated approaches, and even retrieves the rotation matrices with an accuracy comparable to that of the calibrated ones. Yet, the

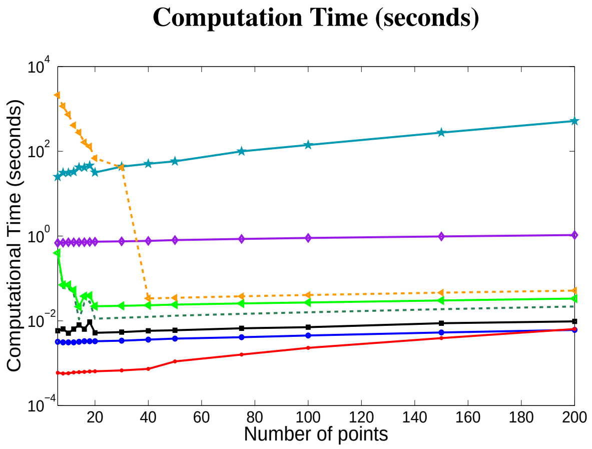


Figure 3.6: Comparison of the computation time of our method against state of the art approaches with respect to the number of input points, and for fixed values of $f = 2500$ and $\sigma_n = 5$. The color codes and line styles are the same as those used in Fig. 3.5.

translation error is larger, and responds to the fact that the ambiguity between focal length and translation cannot be perfectly solved, specially for noisy 2D-to-3D correspondences. In fact, note that not even with the final refinement using Gauss Newton optimization and considering all equation sets we were able to completely solve this ambiguity. In any case, both the translation and focal lengths estimations we obtain are remarkably more accurate than those obtained by the rest of uncalibrated methods. It is worth to note that the L2-L2 algorithm guarantees a bound with respect to the global minimum solution below a certain tolerance ϵ, which we set to 0.05. Although improved accuracies might be achievable choosing smaller tolerances, we found it prohibitive as the computational burden at $\epsilon = 0.05$ was already too high.

The third row in Fig. 3.5 shows the robustness of the method for varying sizes of the point correspondence set. Fixing the image reprojection noise at $\sigma = 5$, and varying the number of points in the set from 6 to 50, the method again outperforms the other uncalibrated methods, and turns to be very similar to the EPnP.

The last row in Fig. 3.5, plots simulation results for varying focal length values. The number of 3D-to-2D correspondences and their 2D noise are set to constant values of $n = 10$ and $\sigma = 5$, respectively. Note that while for low values of f, our UPnP method performs slightly better than other approaches, as projection becomes orthographic, the difference becomes more drastic. The UPnP algorithm remains stable whereas the accuracy of the other uncalibrated algorithms degenerates. This is because DLT, L2-L2 and Linf assume a projective camera model, which leads to failure when the camera gradually comes close to turning orthographic. In contrast, our approach can naturally handle this situation, as the effect of moving from a fully perspective to an orthographic camera is to increase the dimensionality of the kernel of $\mathbf{M}^\top\mathbf{M}$, and thus for large values of the focal length, the UPnP method automatically finds the most accurate solutions at $N = 2$ or $N = 3$.

Fig. 3.6 shows the computation time of all algorithms for an increasing number of input correspondences and fixed values of $\sigma = 5$ and $f = 2500$. All algorithms are implemented in Matlab, although the Linf and L2-L2 methods use compiled C functions. Among the uncalibrated methods, only the DLT algorithm is faster than our algorithm, although as shown in Fig. 3.5, the DLT performs comparatively very poorly in terms of accuracy. Surprisingly, our approach happens to be slower for a small number of input correspondences. This is because when the number of input points is small, the pose and focal length estimates become very sensitive to noise. This requires evaluating all kernel dimensionalities, i.e.,

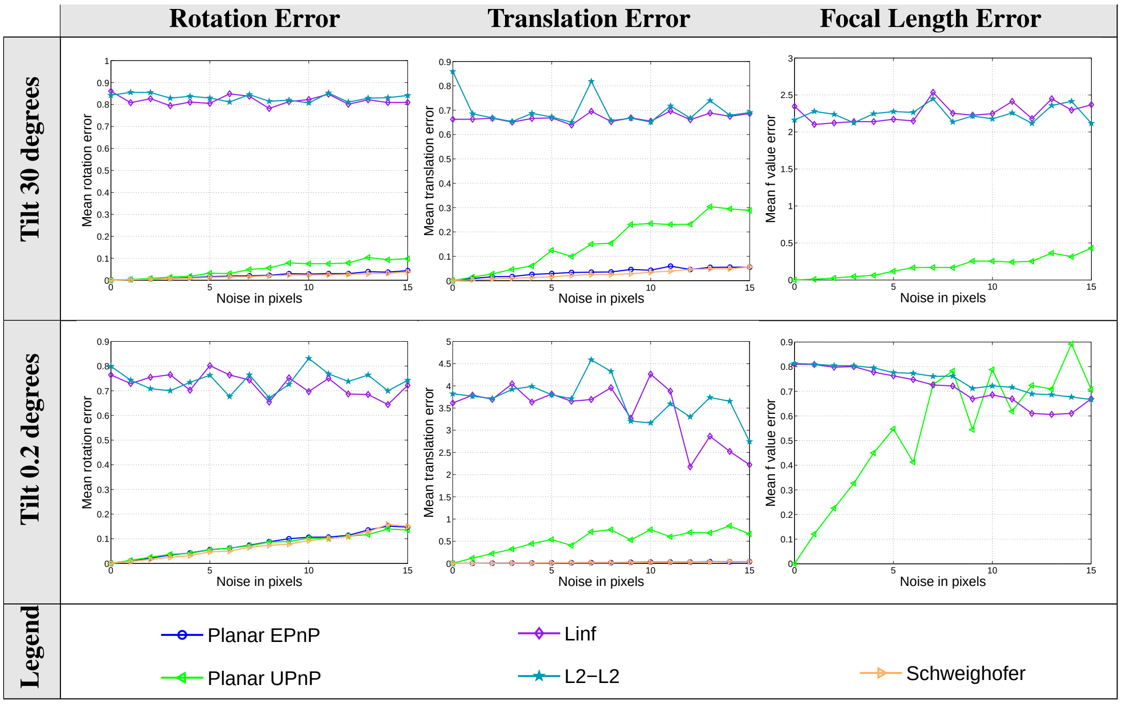


Figure 3.7: Results on synthetic data for planar distributions of points. Mean rotation, translation and focal length errors for increasing levels of image noise, and two different tilt values.

$N = 1, 2$ and 3, where the latter may be quite expensive, especially when testing all equation sets. In particular, observe that the difference in computation time of having to test 500 or all 538.304 equation sets is of more than two orders of magnitude, while the performances reported in Fig. 3.5 of both alternatives are pretty similar.

Yet, when the size of the correspondence set increases ($n \geq 9$), ambiguities and instabilities induced by noise are reduced, and small reprojection errors are generally obtained by just evaluating $N = 1$ and $N = 2$. In fact, for a large number of points, the computation time of our approach is very similar to that of the EPnP, which assumes a calibrated camera. In addition, the cost of our algorithm could be further improved by exploiting the fact that the equations sets that need to be explored are independent and known in advance, and thus, their exploration could be easily parallelized.

3.4.2 Planar Synthetic Experiments

We now present the results obtained on planar scenes. The DLT has been removed from this analysis as it is not directly applicable to planar distributions of points. By contrast, we have included the approach of Schweighofer and Pinz [90], which is a calibrated method specifically designed to handle planar scenes. Jointly with the EPnP, this method is used as a baseline to evaluate the magnitude of the error of the non-calibrated approaches.

These experiments have been performed for a constant number $n = 10$ of 3D-2D correspondences, corrupted using Gaussian noise with a standard deviation σ ranging from 0 to 15 pixels. In addition, we have considered two different situations, one in which the points lie on a quasi frontoparallel plane, and another in which this plane has a tilt of 30 degrees w.r.t to the optical axis of the camera. Fig. 3.7

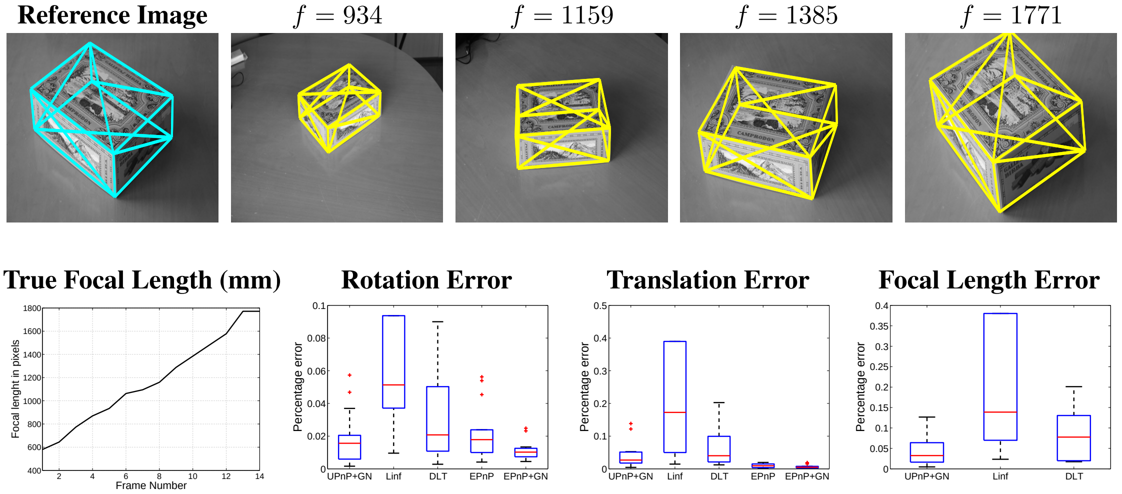


Figure 3.8: Results on a real sequence with increasing focal length. **Top.** 3D model reprojected onto the reference and input images using the pose and focal length retrieved with the UPnP. **Bottom.** Ground truth focal length and performance comparison of all methods.

UPnP+GN	Linf	DLT	EPnP	EPnP+GN
1.67	1052.04	2.72	0.15	0.13

Table 3.1: RANSAC Computation Times (seconds)

summarizes the results. Note that the pose and focal length estimates obtained using the UPnP clearly outperform those of the Linf and L2-L2 methods. The accuracy of our approach only falls when noisy input data is combined with a frontoparallel distribution of points. In this case, the ambiguity between focal length and translation is magnified and cannot be resolved by any of the non-calibrated methods. Yet, our approach yields an estimation of the rotation matrix which is almost as accurate as that of the calibrated algorithms.

3.4.3 Real Images

The method was also tested on a real image sequence taken with a Canon EOS 550D digital camera. The camera was manually moved around an object of interest with known geometry and the focal length was changed from 600 to 2000 pixels. Ground truth focal lengths were read from the *exif jpeg* image headers, and ground truth poses were computed by applying the EPnP+GN to a set of 3D-to-2D matches manually selected. We then manually registered the 3D model to one reference image, from which we extracted approximately 500 SIFT feature points. After backprojecting these 2D points onto the model we obtained a set of reference 3D points, with an associated SIFT descriptor.

At runtime, 2D feature points and their corresponding SIFT descriptors were automatically extracted from each input image, and matched to the set of reference 3D points. This provided an initial set of 3D-to-2D hypotheses. To filter out outliers, we then independently ran RANSAC using each of the algorithms until obtaining a concensus of 200 inlier correspondences. The UPnP, EPnP and DLT performed quite efficiently while Linf required a considerable additional amount of time. The L2-L2 was not applicable within a RANSAC framework as its convergence rate was even two orders of magnitude larger than that of Linf. Table 3.1 reports the mean computation time per frame required for each method.

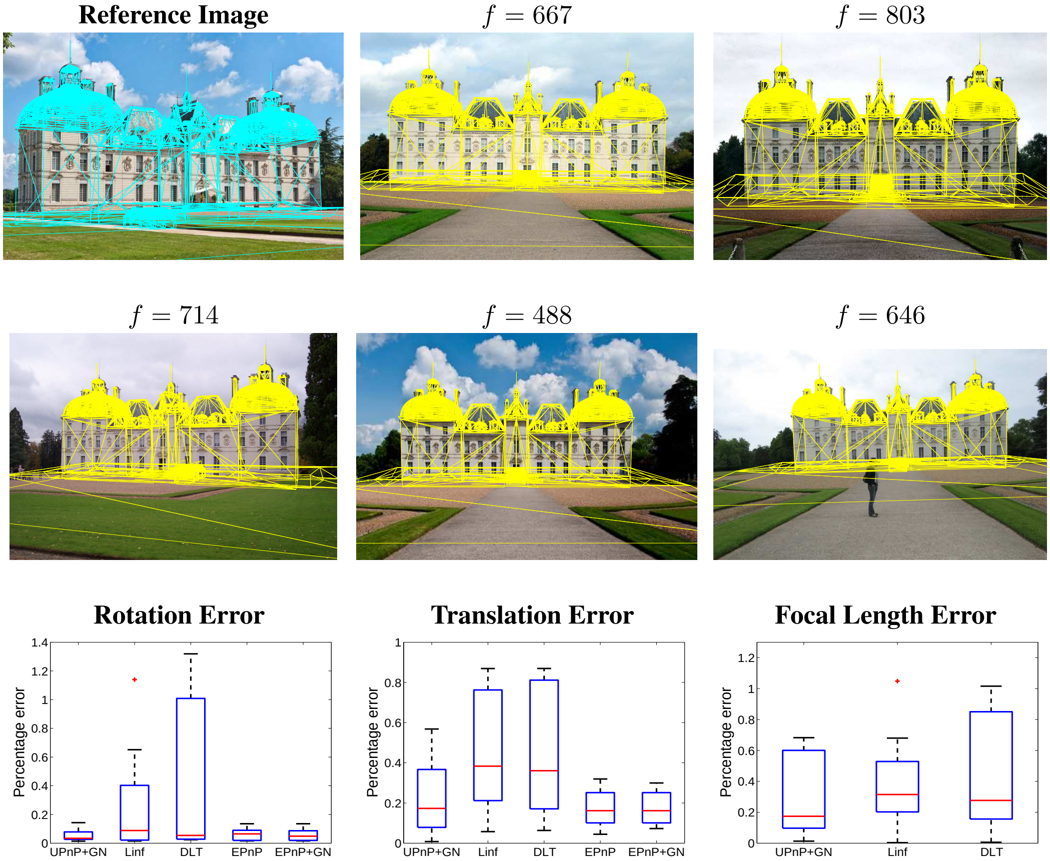


Figure 3.9: Results on a real set of images obtained from $Flickr$ with a 3D model obtained from $GoogleEarth$. **Top.** 3D model reprojected onto the reference and input images using the pose and focal length retrieved with the $UPnP$. **Bottom.** Comparing the accuracy of our approach against Linf, DLT, EPnP and EPnP+GN. The last two methods assume a calibrated camera.

The accuracies of all approaches are depicted in Fig. 3.8-bottom. The images on the top show the reprojection obtained with UPnP.

Finally, as a test case, the method was also used to register 12 images available on Flickr of the Cheverny Castle with its GoogleEarth 3D model. Feature correspondences were manually matched in both the reference and input to obtain pose ground truths. The true focal lengths were obtained from the camera settings available in the Flickr images. The test was again performed after using RANSAC to filter out mismatches between the SIFT features of the reference and input images. As shown in the box plots at the bottom of Fig. 3.9, our method compares again favorably with the DLT and Linf algorithms. Some of the reprojection results are shown in the top of the figure.

3.4.4 Comparison with Minimal Solutions

The UPnP provides an efficient solution to estimate pose and focal length from an arbitrary large number of 3D-to-2D correspondences. As discussed in Section 3.2.1, the minimum number of correspondences which are required to solve the underlying linear system of Eq. 3.6 is 6. In fact, we could even solve

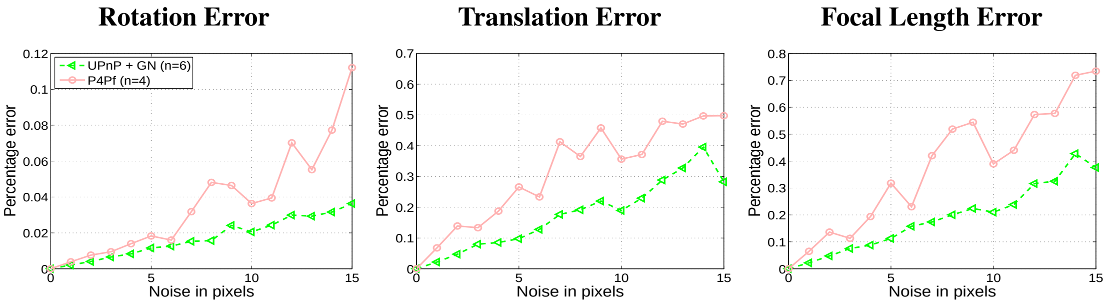


Figure 3.10: Comparison of the UPnP using 6 correspondences vs. the minimal approach proposed in [7], which estimates pose and focal length from four 3D-to-2D correspondences.

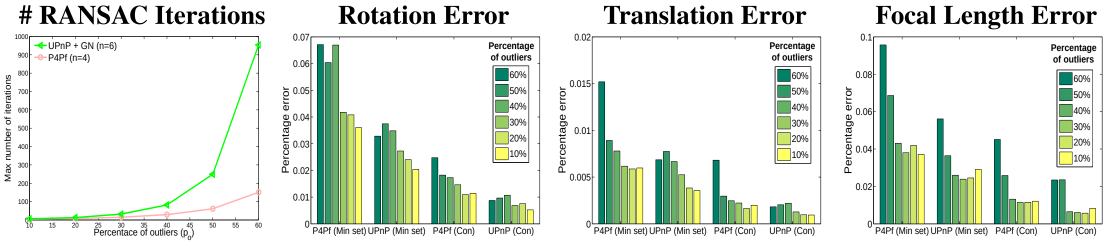


Figure 3.11: Comparison of UPnP vs P4Pf within a RANSAC scheme. **Left:** Number of iterations required to retrieve a hypothesis of n point correspondences free of outliers, where $n = 4$ for the P4Pf and $n = 6$ for the UPnP. **Other three frames:** Rotation, Translation, and Focal length errors for different levels of outliers. Min set: Errors obtained when computing pose using the best minimal subset. Con: Error after computing pose using all the correspondences within the *inlier* set. Since the P4Pf does not generalize to more than four correspondences, the error of the consensus is computed using the UPnP in both cases.

when only 5 noise-free correspondences are given, as in this case the rank of the kernel of $\mathbf{M}^\top\mathbf{M}$ would be $N = 2$. Solving the minimal case with 4 correspondences requires considering larger kernel dimensionalities, and the complexity of the exhaustive relinearization would become impractical. In order to solve the minimal case with four correspondences there exist specialized algorithms such as [7], which takes advantage of the constraints introduced by all the possible pairs of distances between 3D points. These constraints generate a system of 15 polynomial equations, solved using hidden variable or Gröbner basis methods. Note however, that this is only feasible for the minimal case, as the number of pairs of distances between points explodes with n. In Fig. 3.10 we compare the performance of [7], which we denote as P4Pf, with the UPnP for $n = 6$ and for an increasing amount of noise. As expected, P4Pf is more sensitive to noise, and by just considering two additional correspondences UPnP yields significantly more accurate results.

One advantage of taking minimal subsets is that it increases the chances of picking an *all-inliers* subset in a RANSAC-based algorithm. Yet, while this may accelerate the outlier removal process when considering noise free data, it can have an opposite effect when the data, besides containing outliers, is corrupted by noise [99]. In this case, the hypotheses fitted on minimal subsets may be severely biased, even when only containing inliers, and many true inliers may not be included in the final consensus set, leading to accuracy errors. In order to put evidence on this, we have performed an experiment where the P4Pf and the UPnP have been used within a RANSAC scheme. We have considered a set of 5000

3D-to-2D correspondences, corrupted by 2D noise with $\sigma = 5$ pixels, and different percentages p_o of outliers, going from 10 to 60%. Taking minimal subsets of size $n = 4$ for the P4Pf and $n = 6$ for the UPnP, we have then followed an hypothesize-and-test approach, until reaching a maximum number of iterations $i_{\text{max}}^{\text{ransac}}$, that ensures with a confidence level P an outlier-free hypothesis. This threshold is computed as [26]

$$i_{\text{max}}^{\text{ransac}} = \frac{\log(1 - \mathrm{P})}{\log(1 - (\mathrm{p}_i)^n)} \tag{3.18}$$

where $\mathrm{p}_i = 1 - \mathrm{p}_o$ is the percentage of inliers, and P has been set to a 98% level. Fig. 3.11-left shows this theoretical number of iterations for the different percentages of outliers. Fig. 3.11-center and right represent the rotation, translation and focal length errors for each method and level of outliers. In each of these graphs, we plot both the error computing the pose and focal length with the *best* minimal subset for each algorithm, and the error computing the pose and focal length using the whole consensus. The latter has been computed with the UPnP in both cases, as the P4Pf can only be used in the minimal case. Observe that although the P4Pf requires a smaller number RANSAC iterations, the UPnP consistently yields better estimations of the pose. In fact, there are levels of outliers for which the number of theoretical iterations is very similar in both algorithms, while the gain in accuracy is still significant in favor of the UPnP.

3.4.5 Other Methods to Solve the System of Polynomial Equations

Linearizing and relinearizing is just one way to solve multivariate systems of polynomial equations, but there are other alternatives. Of these, the reduced Gröbner basis method has become a popular alternative for the solution of minimal problems in computer vision [7, 10]. The technique uses Buchberger's algorithm which iteratively reduces each polynomial in the set by subtracting from it multiples of elements of the other polynomials in the set in such a way that power products become the smallest possible. The reduced forms in the basis are not unique, and the algorithm is very sensitive to the ordering of variables, in a similar way that ordering plays a crucial role in Gaussian elimination. Our intuition is that the relinearization technique presented in this paper is less sensitive to this ordering of variables since the final solution is not chosen from a single minimal equation set but from a careful evaluation of many sets, each chosen from a different ordering of variables. There are also considerations with regards to complexity. The size of a Gröbner basis has a double exponential growth with respect to the number of variables, and while it can still be manageable for $N = 2$, for our case $N = 3$ its application would be impractical. Lastly, state of the art implementations of the Gröbner basis method only work for rational polynomial coefficients. In our case, the polynomial coefficients are given by the terms of the eigenvectors of $\mathbf{M}^T\mathbf{M}$, which are not rational. Rationalizing these coefficients may also account for reduced numerical accuracy of the solution. This conjecture is only empirical, but sustained from our comparison to the P4Pf which uses the Gröbner basis method. We leave the solution of our polynomial system of equations using rational ideals as an issue for further research.

3.5 Conclusions

In this chapter, we have presented a fast solution to the problem of recovering the pose and focal length of a camera, given n 3D-to-2D correspondences. We have shown that our approach can be expressed as the solution of a fixed-size linear set of equations independent of the number of points, similar to the EPnP algorithm for the fully calibrated case. However, dealing with uncalibrated cameras required the introduction of new approaches to handle higher degree polynomials under noisy input data. To this end, this chapter presents the *extended linearization* and *extended relinearization* techniques, which overcome the limitations of current linearization-based approaches. An extensive evaluation of the method shows remarkable improvement when compared to competing methods, and also to algorithms for pose recovery that make use of calibrated cameras.

An unexploited advantage of the approach is that it is highly parallelizable for large kernel sizes since the sets of equations that need to be exhaustively explored are known in advance. Another alternative to speed up the process would be to use strategies such as kernel voting [8] to directly pick a solution from the set of minimal equations, based on how well they satisfy the distance constraints. This would remove the need to repetitively calculate and test reprojection error. We leave this as an unexplored venue for further research.

Chapter 4

Uncalibrated Pose Estimation from Unknown Point Correspondences

4.1 Introduction

The 3D camera pose and focal length estimation presented in the previous chapter requires to know in advance the set of 3D-to-2D correspondences between a 3D model of the object and an input image. Typically, a reference model image is registered to the 3D model using point descriptors, like SIFT [60], and by performing robust matching [26, 14], we can eliminate outliers to come up with an adequate correspondence set.

Matching methods based on RANSAC rely on having a small outlier rate. This way, the probability of obtaining a correct minimal set of points is high enough to compute a solution in fixed time. If the percentage of outliers increases, the number of RANSAC loops needed to obtain a correct set of minimal points grows exponentially. Also, as shown in [78], using minimal sets of points might not yield the best pose if there is noisy data in the system. The percentage of outliers depends on several factors: the precision recall of the point descriptor, changes in appearance, changes in point of view, repeated patterns, self occlusions, etc. We propose a matching algorithm that performs robust matching under the presence of a large percentage of outliers. Our approach is a generalization of the work in [70] to deal with uncalibrated cameras and noisy 3D information. We show experiments using 3D data obtained with a Kinect camera [92] as an example of the kind of contexts we solve.

To tackle all these issues we propose to split the initial prior distribution of the combined pose and focal length estimate into an arbitrary large number of Gaussian priors. These priors are spread within very rough bounds of where the pose and focal length are expected to be. At runtime, each of these priors is used to guide the search for the 3D-to-2D correspondences, while progressively pruning the number of potential candidate matches and refining the pose and focal length values. Repeating this process for each of the priors guarantees an exhaustive exploration of the solution space at a limited computational cost.

Experiments in both synthetic and real data will show that the proposed approach is robust to large levels of 2D and 3D noise and clutter, yielding reasonable results for outlier rates of up to 80%. This significantly outperforms competing approaches [14, 12, 50], and is comparable to methods that assume a calibrated camera [70].

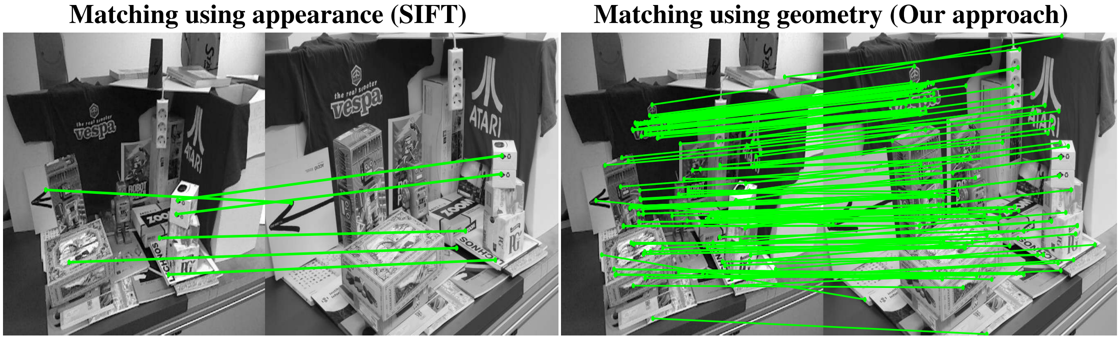

Matching using appearance (SIFT)
Matching using geometry (Our approach)

Figure 4.1: Inlier correspondences of a matching algorithm that uses appearance information (left) and our matching algorithm (right) that only uses the 3D and 2D location of the points to search for the correspondences. Matching between the intensity component of a Kinect camera and a Canon EOS 60D camera, together with the viewpoint changes and the self-occlusions, differences in terms of image noise and resolution jeopardize appearance matching producing a very small set of correct matches. In contrast, our method based purely on geometric information is able to retrieve a much larger number of correspondences, accurately computing the relative pose and focal length under such conditions.

4.2 Related Work

Pose estimation techniques that maximize image similarity, such as [11], are not applicable in our context due to their limited ability to deal with significant differences in appearance and reduced capture range, as that of Fig. 4.1. We shall therefore consider only techniques that explicitly perform matching using the geometric structure of the 2D and 3D point sets.

The robust estimation of correspondences between two sets of points has been historically solved by hypothesize and test algorithms such as RANSAC [26] and Least Median Squares [87]. They rely on a random sampling of minimal subsets to generate model hypotheses, and favor the one that best explains most of the data points. Unfortunately, in these methods, computational complexity scales exponentially with the number of model parameters and the size of the point set. Among the several variations of the original RANSAC algorithm, Guided-MLESAC [101] and PROSAC [14] avoid sampling unlikely correspondences by using appearance based scores and thus are not applicable to our problem. Similarly, GroupSAC [73] uses image segmentation to sample more efficiently the data. Other techniques of the same family such as Preemptive RANSAC [74] or ARRSAC [83] work within a limited time scenario thus increasing the probability of not reaching the best estimate. Finally, MultiGS [12] accelerates the search strategy by guiding the sampling with information from residual sorting and is able to account for multiple structures appearing in the scene.

In the absence of robust appearance information graph matching can be used, as proposed by [19]. Yet, due to its high computational cost, these methods are only applicable to small graphs. While such approaches allow global optimization, they cannot be used with large intra-image distances due to very different points of view of the scene, or when the number of outliers is excessive, which are the cases we consider in this paper.

The L_∞ technique proposed in [50], uses second-order cone programming, and guarantees optimality under the L_∞ norm, for different geometric structure and motion problems, including the camera pose estimation considered in this work. However, this particular metric is highly sensitive to outliers, as pointed out in [35]. Even when it is possible to address in part the outlier removal problem as proposed in [94], the L_∞ solution for the camera pose estimation only performs as well as the standard L_2 norm. In the experimental section, we compare our approach against [12, 14, 50] which we consider representative methods of the whole family.

Other approaches simultaneously solve for pose and correspondences purely from geometric point matching. Of these, SoftPOSIT [16] uses an iterative technique to generate correspondence candidates, but the global minimum can not be guaranteed. Our approach is inspired in the Blind PnP algorithm [70], where local optimality is alleviated introducing the scene geometry as pose priors, modeled as a Gaussian mixture model, and progressively refined by hypothesizing correspondences. Incorporating each new candidate in a Kalman filter rapidly reduces the number of potential 2D matches for each 3D point and makes it possible to search the pose space sufficiently fast for the method to be practical. More recent techniques [91] use robust estimation in a final stage to refine the pose. Unfortunately, the approach cannot be applied straightforward to the uncalibrated case due to the ambiguities between focal length and the pose translation vector.

4.3 Method

The structure of the algorithm we propose is similar in spirit to the Blind PnP algorithm [70]. In an initial stage we split the solution space into a set of Gaussian clusters. Then, we progressively explore each of these clusters to simultaneously establish the 2D-to-3D correspondences and refine the camera pose and focal length. In contrast to the original Blind PnP formulation, we are considering an uncalibrated camera, with the focal length as an additional parameter to estimate. This yields and extra degree of complexity to the problem, as there are large ambiguities between changes of the camera focal length and translations along the optical axis. In addition, the proposed approach takes into account the uncertainties in the 3D model, characteristic of range sensors such as ToF or Kinect cameras [29].

4.3.1 Problem Formulation

Let us assume we are given a reference model made up of M 3D points $\mathcal{X} = \{\mathbf{x}_i\}$, with their 2D correspondences on a reference image, and a set of N 2D points $\mathcal{U} = \{\mathbf{u}_j\}$ on an input image, acquired with an uncalibrated camera. We consider that correspondences between the 2D and 3D sets is possible since both point sets were extracted using the same interest point detector over the reference and input images. Let us denote by $\mathbf{p}$ a 6-dimensional vector, parameterizing the rotation and translation that aligns the camera with respect to the 3D point set coordinate system, and let f be the unknown camera focal length. We additionally assume a camera with square pixels, and with the principal point located at the center of the image, and hence, being the focal length the only unknown intrinsic parameter. Our goal is to retrieve the pose of the camera and its focal length. As the 3D-to-2D correspondences are unknown, they need to be retrieved together with the pose and focal length parameters. This can be formulated as an optimization problem, retrieving $\mathbf{p}$ and f such that the reprojection error between the projected 3D points $\mathbf{x}_i$ and their corresponding matches $\mathbf{u}_j$ is minimized,

$$\underset{\mathbf{p},f}{\text{minimize}} \sum_{i=1}^{M} \mathsf{Inlier}(\|\mathsf{Proj}(\mathbf{x}_i;\mathbf{p},f) - \mathsf{Match}(\mathbf{x}_i;\mathcal{U})\|) \tag{4.1}$$

where $\mathsf{Proj}(\mathbf{x}_i;\mathbf{p},f)$ returns the 2D perspective projection $\tilde{\mathbf{u}}_i$ of a 3D point $\mathbf{x}_i$ given the pose and focal length parameters; $\mathsf{Match}(\mathbf{x}_i;\mathcal{U})$ returns the $\mathbf{u}_j \in \mathcal{U}$ that is closest to $\tilde{\mathbf{u}}_i$; and

$$\mathsf{Inlier}(d) \begin{cases} d & \text{if } d < Max\_distance\_inlier \\ \mathsf{Penalty\_outlier} & \text{otherwise} \end{cases}$$

is a function that penalizes points whose reprojection error is above a $Max\_distance\_inlier$ threshold . This is to avoid local minima, otherwise, small sets of candidate matches can get a lower error than the actual solution.

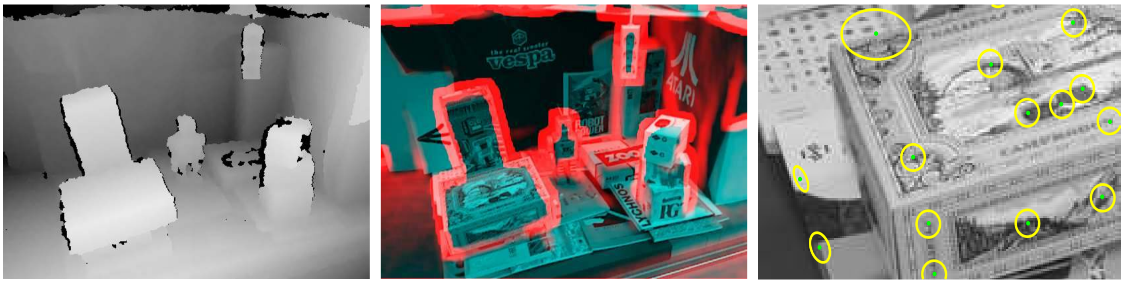


Figure 4.2: **Uncertain 3D model**. **Left:** 3D model acquired with a Kinect camera. Regions in which the 3D data is most uncertain are depth discontinuities. **Center:** We detect the uncertain regions –shown in red– computing depth covariances within local neighborhoods. **Right:** A 3D covariance is assigned to each 3D model point and propagated to the image plane. This is used to limit the area where to search for potential match candidates.

4.3.2 Modeling Uncertainty

A straight-forward way to minimize Eq. 4.1 would be to use a RANSAC-like approach [26], and repetitively hypothesize sets of four 3D-to-2D correspondences until one of them yields an estimate of $\mathbf{p}$ and f that brings the reprojection error below a certain threshold. Unfortunately, since these methods do not introduce constraints on the potential set of 2D candidates that can match each 3D point, they are only computationally tractable for a relatively small number of features. In order to make the minimization of Eq. 4.1 tractable, we follow a similar strategy as in [70], and split the solution space into an arbitrary large number of Gaussian regions. At runtime, each of these regions is explored in turn, guiding a matching process for each 3D point. By splitting the search space in these small regions, the total number of potential 2D candidates for each 3D point is significantly reduced.

Estimation of the pose and focal length priors is done in a pre-processing stage. We first define the bounds of these parameters. For the pose, we acquire several images of the 3D object at the extremal positions and orientations of the working space. This is used to build an hyper-box in the 6-dimensional pose space, which is then subsampled using Montecarlo. Expectation Maximization is run over these samples to compute the N_p Gaussian priors on the pose, defined by a set of mean poses $\mathbf{p}_k$, $k = 1, \ldots, N_p$, and a set of 6×6 covariance matrices $\Sigma_k^{\mathbf{p}}$. Similarly, the range of feasible focal lengths, is split into N_f Gaussian priors, defined by mean values f_l and the corresponding one-dimensional variances σ_l^f, $l = 1, \ldots, N_f$.

One of the key ingredients of our approach is that it lets us handle uncertain 3D models, such as those obtained from a Kinect camera. It is well known that these sensors suffer from inaccuracies, especially at depth discontinuities (see Fig. 4.2). In order to inject this uncertainty into the optimization process, each 3D model point $\mathbf{x}_i$ is assigned a covariance $\Sigma_i^{\mathbf{x}}$, computed considering the depth variations of their neighboring points. During the optimization process, those points with larger uncertainties will have smaller impact in the computation of the solution. We also assign an uncertainty $\Sigma_j^{\mathbf{u}}$ to each 2D measurement.

4.3.3 Optimization

Given the sets $\mathcal{X}$ and $\mathcal{U}$, the pose and focal length priors, and the 3D and 2D uncertainties, we proceed to the optimization of Eq. 4.1 by progressively exploring each pair of priors $\{\mathbf{p}_k, \Sigma_k^{\mathbf{p}}\}; \{f_l, \Sigma_l^f\}$, using the following steps:

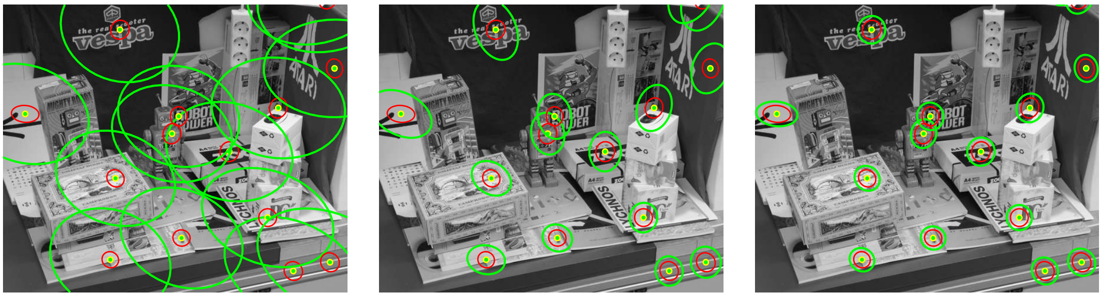


Figure 4.3: **Limiting the number of potential candidates. Left:** Search region obtained after projecting the three terms of Eq. 4.2 independently. **Center and Right:** Refinement of the search space, after establishing correspondences.

Projecting Uncertainties onto the Image Plane

To limit the number of potential 2D match candidates for each 3D point $\mathbf{x}_i$, we project them onto the image plane and compute the uncertainty in the projection assuming independent contribution from all three sources: 3D point uncertainty, pose uncertainty, and focal length uncertainty. The result is a Gaussian distribution with mean $\tilde{\mathbf{u}}_i$ and covariance $\Sigma_i^{\tilde{\mathbf{u}}}$:

$$\begin{aligned} \tilde{\mathbf{u}}_i &= \mathsf{Proj}(\mathbf{x}_i;\mathbf{p}_k,f_l) \\ \Sigma_i^{\tilde{\mathbf{u}}} &= \mathbf{J}_{\mathbf{x}}\Sigma_i^{\mathbf{x}}\mathbf{J}_{\mathbf{x}}^\top + \mathbf{J}_{\mathbf{p}}\Sigma_k^{\mathbf{p}}\mathbf{J}_{\mathbf{p}}^\top + \mathbf{J}_f\sigma_l^f\mathbf{J}_f^\top \ , \end{aligned} \tag{4.2}$$

where $\mathbf{J}_g = \frac{\partial \mathsf{Proj}(\mathbf{x}_i;\mathbf{p}_k,f_l)}{\partial g}$ is the Jacobian of the projection function with respect to each of the uncertain parameters $g = \{\mathbf{x},\mathbf{p},f\}$. Using the Gaussian distribution $\{\tilde{\mathbf{u}}_i, \Sigma_i^{\tilde{\mathbf{u}}}\}$, we can define a search region for the point $\mathbf{x}_i$, and consider as potential candidates $\mathcal{PC}(\mathbf{x}_i)$ all points $\mathbf{u}_j \in \mathcal{U}$ whose Mahalanobis distance is below a threshold $\mathsf{Max\_Mah}$, i.e:

$$\mathcal{PC}(\mathbf{x}_i) = \left\{\mathbf{u}_j \in \mathcal{U} \ \text{ s.t. } \ (\mathbf{u}_j - \tilde{\mathbf{u}}_i)^\top(\Sigma_i^{\tilde{\mathbf{u}}})^{-1}(\mathbf{u}_j - \tilde{\mathbf{u}}_i) < \mathsf{Max\_Mah}^2\right\} \cup \{\emptyset\} \tag{4.3}$$

where $\emptyset$ denotes the possibility that $\mathbf{x}_i$ is in fact an outlier and does not have a 2D image correspondence.

Local Guided Hypothesize-and-Test

Once we have defined the set of potential 2D candidates for all the 3D points, we start a hypothesize and test strategy, similar to what is done in a standard RANSAC algorithm. Yet, in contrast to RANSAC we only need to establish potential matches within local neighborhoods. In addition, after a hypothesis has been made, we use a Kalman filter formulation to shrink the size of the Gaussian regions associated to the pose and focal length, to further reduce and guide the set of potential candidates in each iteration Figure 4.3. We initialize this step choosing the least ambiguous point

$$\mathbf{x}_i^* = \arg\min_{\mathbf{x}_i \in \mathcal{X}} |\mathcal{PC}(\mathbf{x}_i)| \ , \tag{4.4}$$

i.e, the 3D point with the lowest number of potential candidates. In doing so we start with a 3D point with low uncertainty, since these are the ones with smaller search regions for potential matches, in the Mahalanobis sense. We then hypothesize the match $\{\mathbf{x}_i^*, \mathbf{u}_j^*\}$, where $\mathbf{u}_j^*$ is the 2D candidate within $\mathcal{PC}(\mathbf{x}_i^*)$ that is closest to $\tilde{\mathbf{u}}_i^*$ in terms of Mahalanobis distance. We then use standard Kalman filter equations to update the pose and focal length and reduce their associated covariances:

$$\begin{array}{ll} \mathbf{p}_k^+ = \mathbf{p}_k + \mathbf{K}_\mathbf{p}(\mathbf{u}_j^* - \tilde{\mathbf{u}}_i^*) & f_l^+ = f_l + \mathbf{K}_f(\mathbf{u}_j^* - \tilde{\mathbf{u}}_i^*) \\ \Sigma_k^{\mathbf{p},+} = (\mathbf{I} - \mathbf{K}_\mathbf{p}\mathbf{J}_\mathbf{p})\Sigma_k^\mathbf{p} & \sigma_l^{f,+} = (1 - \mathbf{K}_f\mathbf{J}_f)\sigma_l^f \end{array} .$$

The new pose, focal length and covariance matrices are used to project again the 3D points onto the image, and define new and smaller search regions.

Backtracking and Iterating over all Priors

Maintaining the strategy of choosing the 3D point with less potential matches, and the 2D point closest to its projection, we hypothesize new 3D-to-2D matches and refine the pose and focal length posteriors and their associated uncertainties. This process is repeated until the Kalman update terms become negligible, usually in less than five iterations. Upon convergence, we project the remaining 3D points onto the image and match them to the nearest 2D feature point. 3D points whose nearest neighbor distance is larger than $Max\_distance\_inlier$ are classified as outliers. Using both the inlier and outliers points, we compute the error of Eq. 4.1 and stop the algorithm for the current prior set $\{\mathbf{p}_k, \Sigma_k^\mathbf{p}\}; \{f_l, \Sigma_l^f\}$ if the error falls below a given threshold. If not, we backtrack through the list of 3D-to-2D matches to change the assignments and repeat the guided search and refinement process. When no more assignments are available, we repeat the process with a different pose and focal length prior.

Accelerating the Optimization and Final Refinement

We can further increase the efficiency of the algorithm if a wrong pose or focal length prior is detected before completing the exploration over all corresponding points. One criteria for doing this is setting a threshold to the maximum number of outliers allowed. Following a branch and bound strategy, we terminate the exploration for a specific pose and focal length prior when the minimum number of outliers reaches a specific threshold. Being conservative, we have set this threshold to $0.8M$, i.e, we accept a maximum of 80% of outliers.

Finally, at the termination of the search, our algorithm yields a set of correspondences and estimations of the focal length and pose parameters. We further refine these results by performing a final RANSAC-based step, in order to eliminate residual mismatches. Note however, that this approach does not need to establish matches, but just remove a very reduced number of incorrect ones, and thus, its computational cost is negligible.

4.4 Experimental Validation

Evaluation has been done on synthetic and real data. Synthetic results accurately evaluate our approach in comparison to state-of-the-art. Real data results provide a qualitative assessment of the method in a challenging scenario with high levels of clutter and noise.

4.4.1 Synthetic Results

We now present results on synthetic data by evaluating our algorithm in controlled experiments with known ground truth. For these experiments we compare our approach, denoted Uncalibrated BlindPnP (UBPnP), with Multi Guided Sampling (MultiGS) [12], the L_∞ method (Linf) [50] and with PROSAC [14]. We also compare it against BlindPnP [70] for which we inject the true calibration parameters, and use the results as a baseline to give significance to the reported accuracy estimates. We synthetically generated the 3D model by randomly sampling $N = 50$ points from a cube of dimensions $x \in [-1, 1]$, $y \in [-1, 1]$,

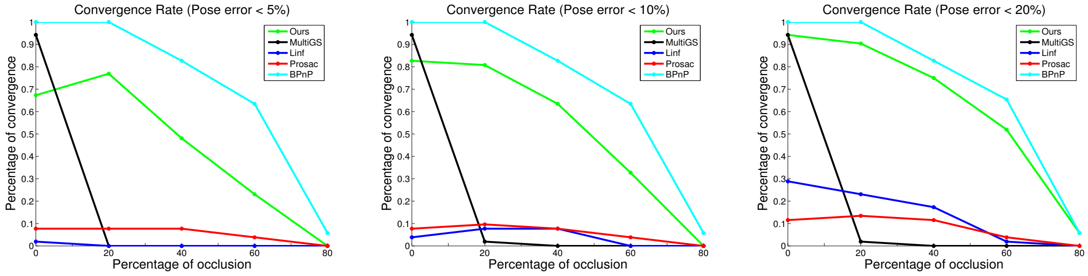


Figure 4.4: **Convergence rates obtained for increasing levels of occlusion.** The solution is considered correct if both the relative errors between the estimated pose and focal length, and their ground truth values do not exceed 5, 10 and 20% (from left to right).

$z \in [-1, 1]$, and selecting the ground truth camera pose from inside a torus surrounding the point set. The camera optical axes are chosen randomly to point anywhere on the 3D model. Then, the 2D points are produced by projecting the model onto a 640×480 image, using a calibration matrix where we allowed the focal length to vary within the interval $f = [600 - 1200]$. We performed 50 trials on each of the different setups with increasing percentages of occlusion $p_o \in \{0, 20, \ldots, 80\}$. Pose clusters are created with a 30-component Gaussian Mixture Model, with 10 additional components to cluster the focal length range.

The standard situation in which Linf, MultiGS and PROSAC are used, is one in which a set of potential matches, computed using texture information, are given in advance and their goal is to reject mismatches. In our case, since the texture is not available, we calculate the potential matches for these algorithms using only the pose and focal length clusters in which the correct solution lies. We have projected the covariances to get all the matching candidates for each 3D point and kept all the possible pairs as candidate matches. In addition, for PROSAC we replaced its score function based on appearance similarity, by a similarity function defined by the Euclidean distance between the projected point and each potential 2D candidate. Note that this criterion should be informative enough considering the points were projected from the correct pose prior.

Results comparing the convergence ratio in all the experiments are shown in Fig.4.4. This convergence ratio, represents the percentage of experiments for which the relative rotation, translation, and focal length errors are below a certain threshold (5%, 10% and 20%). These thresholds are chosen to reflect the fact that the reconstruction error is more sensitive to a correct estimate of the rotation than to a correct estimate of the translation and focal length terms. Note that the performance of our approach breaks only once occlusion levels larger than 50% are reached, being very similar to the calibrated BlindPnP. As seen in the charts, the methods built to rely on appearance, even with the given advantages, cannot perform as well as purely geometric methods when appearance cannot be used. Regarding computational time, UBPnP scales linearly the complexity of BPnP as a result of introducing an extra dimension, clustered using additional N_f Gaussian priors. This means a computation time between 10 and 1000 seconds for increasing sizes of the point set going from 20 to 80 points.

4.4.2 Results with Real Data

The technique has been tested for the registration of imagery acquired with a Canon EOS 60D camera varying the pose and focal length, to a 3D model acquired with a Kinect camera. The Kinect intensity image was used as a reference to extract the model points. Points of interest were extracted using DoG. $M = 100$ model points were computed on the reference image, and between $N = 100 - 150$ were computed on each of the 50 input images.

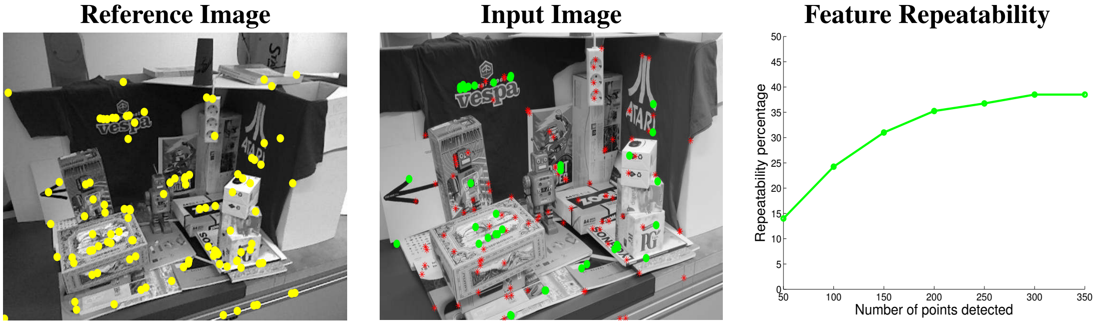


Figure 4.5: **Feature repeatability.** The image in the middle shows in green the feature points that have been consistently detected in both the reference and input images, and in red, the features that have only been detected in the input image. The graph on the right shows how the percentage of repeated features (potential matches) increases with the number of detected points. In our experiments we work with 50-100 model points, and hence, we need to be robust to percentages of up to $75-85\%$ of outliers.

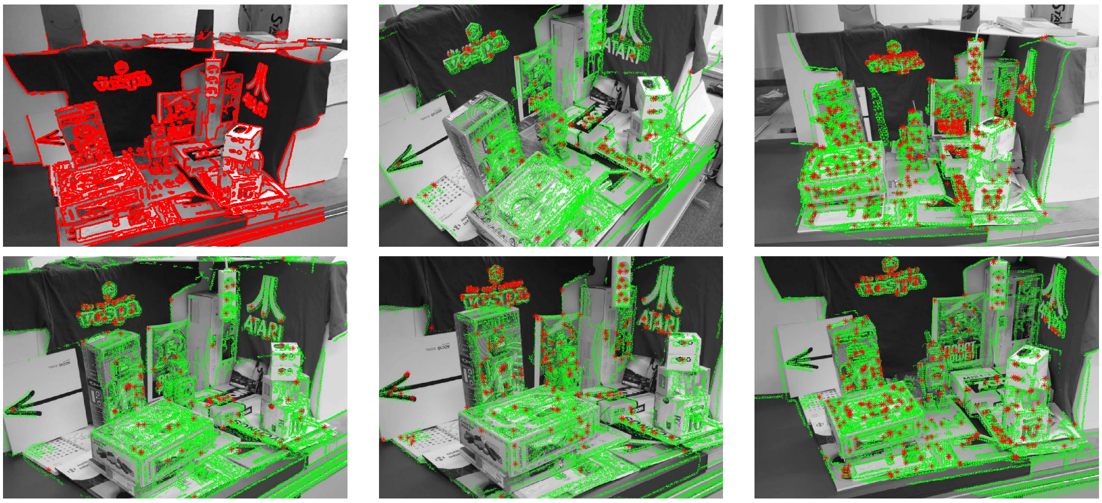


Figure 4.6: **Real experiments. Top Left:** Reference image registered to the 3D model. **Others:** Reprojection of the 3D model onto the input images after estimating pose, focal length and 3D-to-2D matches with our approach.

One important issue we had to handle is that of obtaining a minimum number of keypoints that consistently appear in the reference and input images. This problem is especially critical in our framework, as our algorithm (and also the competing approaches) can only handle, in a reasonable amount of time, sets of about 100 points. As shown in Fig. 4.5 the percentage of inliers decreases with the number of detected points, dropping from levels of about a 40% of inliers for 350 models points, to 25% for 100 points, which is the number of 3D model points we used in our experiments. This is therefore a challenging scenario to test the robustness of our approach in the presence of outliers.

To compute the pose priors we acquired several images at extremal positions and orientations of the working space and manually registered them with the 3D map. The poses of these images were used as bounds for fitting $N_p = 100$ pose priors. The $18-150$mm range of the lens was split into $N_f = 10$ Gaussian intervals. Given that the ground truth pose of the query images is not available, we evaluated the method according to the minimization of the reprojection error. Fig. 4.6 shows several images in which the boundaries of the 3D model are reprojected after computing the correspondences, pose and focal length. Even dealing with large differences in viewpoint, the reprojection results are very accurate.

4.5 Conclusion

Simultaneously estimating the camera position, orientation, focal length and establishing 3D-to-2D correspondences between model and image points, poses a challenging optimization problem which can hardly be solved without prior information. Most current approaches rely on appearance information to first solve the correspondences and then retrieve the pose and focal length while rejecting missmatches. Yet, there are many situations in which the appearance is either not available or not a reliable cue.

In the absence of appearance, we propose to use only geometric priors, which are just rough approximations of the pose and focal length solution. By progressively exploring these priors we are able to efficiently prune the potential number of 3D-to-2D matches, while reducing the uncertainty of the pose and focal length estimates. The method is shown to be highly resilient to clutter and noise on the image features and in the 3D model. The latter is especially suited for dealing with 3D models obtained from noisy range sensors, such as the Kinect or Time of Flight cameras.

Chapter 5

Pose Estimation without Points of Interest.

5.1 Introduction

In the previous two chapters we presented solutions to the pose estimation problem, first from a set of known 2D-3D correspondences, and later without knowing the a priori pairing, and computing it along the way during the optimization using geometric priors to aid feature matching search. In this chapter we explore the use of machine learning to annotate appearance features with the camera pose most common to them. In this way we combine bothe the strengths of geometry and appearance in the same pose estimation framework.

Pure geometric methods for pose estimation initially use training data to build a 3D model, and then search for the 2D-to-3D correspondences that best align interest points in the test image with the 3D model [69]. Machine learning approaches on the other hand, annotate training imagery with discrete locations in the pose manifold, and then search globally for this pose-annotated matching of appearance, without resorting to full 3D reconstruction of the object [32, 76, 108]. The advantage of the global methods is that they are less sensitive to precise localization of individual features, which makes them more robust to image degradations than local geometric methods. But in contrast, global methods are not generally robust to occlusions. In addition, they often require splitting the pose space into several classes, and train specific classifiers for each of them, limiting the precision of the estimated pose to that of the granularity between classes and losing the correlation between neighboring poses.

In our proposed approach, LETHA, (Learning on Easy data, Test on Hard), we use high-definition training images to create a 3D model of the object, and from it devise a pose-indexed feature extraction scheme that binds image quantities to the object pose. These features are then combined into strong priors for each training pose. Since we focus on one single object, the priors we build capture the variability of the appearance and generalize to test images with severe artifacts in a manner similar to that of [57, 20]. These approaches, though, again rely in the fact that similar local features appear in both training and test images. We get rid of this requirement by building specific priors for each training pose, in which we exactly know where the features should appear in the test image, hence no point of interest matching is needed. A *single* classifier, common to all the poses, is trained from these pose-indexed feature vectors. We use a new novel procedure, dubbed AbstainBoost, able to cope with incomplete feature vectors, a situation that occurs during self occlusion, allowing us to evaluate the classifier, even when some features have been wiped out or corrupted by image artifacts such as loss of resolution, motion blur or partial occlusions.

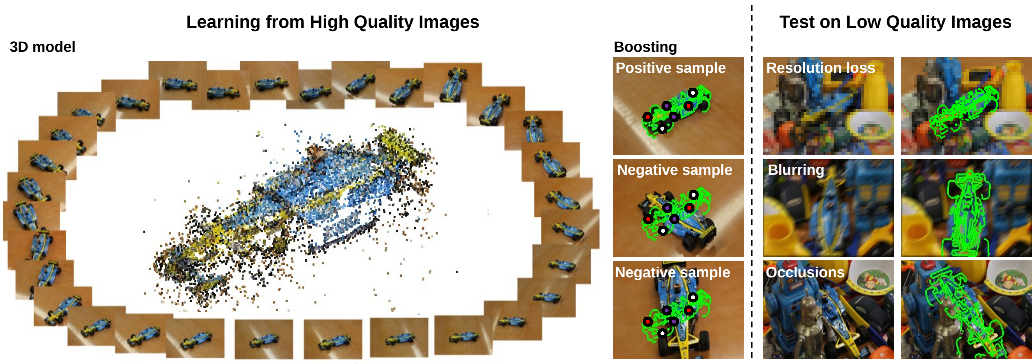


Figure 5.1: **LETHA pipeline diagram** LETHA uses high-quality images to train a classifier that is tested on low quality data. The top frame shows the built 3D model, one positive and two negative training image-pose pairs. The fact that features are indexed with pose is indicated by the same location on the three training images of the projection of point pairs, and by the object contour projected in green. Observe that detecting the F1 car in the test images at the bottom frame is even difficult for the human eye due to different artifacts. The green contour indicates the pose estimated by our approach.

During the test phase, given a low quality input image and a hypothetical pose to assess, the pose-indexed feature vector is computed similarly, without the need to detect and match points of interest, and fed to the classifier. The object's pose is estimated by measuring the classifier response for all the poses seen in the training images, and then refined by resampling around those poses with maximal classifier response. We insist on the fact that since this optimization is done by visiting multiple poses systematically, we do not need to match points of interest, or perform any type of fragile matching prior to using our predictor.

As shown in Fig. 5.4.2, our method is able to estimate the pose even in the presence of severe image artifacts such as motion blur and occlusion. As we will demonstrate in the experimental section, LETHA compares favorably against geometric approaches based on SIFT and DBRIEF features, and also against global approaches based on Bag of Features descriptors [107], GIST [75] and PCA cross-correlation [102].

The solution we propose consists on building a single classifier that will be able to handle viewpoint changes without clustering the pose space or building several classifiers for each viewpoint. To showcase the strength of handling 3D geometry in a coherent and complete manner offer we will perform instance 3D pose estimation in extremely hard images. The different kinds of hard images will be handled without retraining the classifier. This approach uses a very simple feature based in gray level intensity. In such hard images no viewpoint information could be obtained previously with the amount of detail we have obtained in this paper.

5.1.1 How we came across the idea

If we wanted to be able to solve hard scenarios we knew we could not rely only on feature points like [60] or [3]. We turned to look for previous work able to perform pose estimation without using feature points. We researched previous work on instance pose estimation methods that did not rely on feature points and found that most approaches heavily rely on shape to perform instance pose estimation [55, 54]. Boundary information is really robust against the absence of texture on the object as well as illumination changes but if an object presented symmetries it was not clear to us it would work with such good results, it would also suffer under occlusions. In our approach we ended using shape implicitly.

One of the first things we got straight was the use of a classifier, we were trying to emulate the process a child does learning a new object, so it made sense to use learning techniques. By learning the best cues for each object instance we can handle objects that are symmetrical and objects that do not have texture. Once we had decided on it we thought that we were going to have serious problems handling viewpoint variance. When classifiers are used to obtain viewpoint information usually viewpoint space needs to be clustered and a classifier is trained for each cluster [32, 76]. The solutions obtained usually have to reduce pose space to a number of clusters that normally represent rotation around the object in no more than 16 clusters to obtain good results.

We then turned to see how [65] and [77] tackled the problem. Both their approaches left behind global representations of the object to focus on using features over the object, this way the could give a pose estimate and not the closest cluster of poses. Although, both approaches just were able to handle a 2 dimensional subspace of pose by defining a manifold around the object and obtaining the viewpoint but not the full pose.

To achieve our goal we started by thinking how we could handle the changes produced by viewpoint variance. We thought that using the 3D information of the object could be the solution. By taking into account the 3D shape of the object we are able of properly handling any kind of viewpoint variance and self occlusions. We ended using Structure from Motion [95] to build a 3D model of the object from the training images. We still did not know what feature we could use to provide the most robust solution.

We thought that it would be interesting to search over the pose space directly instead of using a sliding window approach. If what we were looking for was the object in a particular pose we thought that making hypothesis on the pose space instead than in the image plane made sense. By taking candidates in the pose space and knowing the object 3D we could handle viewpoint variance to its full measure.

5.2 Related Work

The proposed method stands between instance detection and 3D pose estimation. 3D pose estimation methods may be roughly split in those techniques relying on local image features that purely use geometric relations to compute the pose; and methods that compute global descriptors of the image to estimate the pose.

Local approaches use feature point descriptors [60, 3] to estimate 2D-to-3D correspondences between one input image and one or several reference images registered to a 3D model of the object. PnP algorithms such as the EPnP [69] are then used to enforce geometric constraints and explicitly solve for the pose parameters. On top of that, robust RANSAC-based strategies [14, 70] can be used both to speed up the matching process and to filter outlier correspondences. Yet, while these methods provide very accurate results, they require both the reference and input images to be of high quality, such that local features can be reliably and repetitively extracted. As we will show in the results section, these methods are not applicable for the level of image artifacts we consider in this paper.

When texture cannot be obtained in an object we need to use shape information. [24, 37, 46, 103] are examples of what we mean by texture information not being available or reliable. In such cases resorting to boundary information. The approach proposed in [37] is an special case as it needs 3D input for the input image. This makes the approach specially suited to use with the Kinect camera [92]. An interesting evolution of [37] was presented in [43], it focuses in solving object partial occlusions by taking into account what it sees in the boundaries of the object it infers if the response of the classifier in a particular part of the object is due to an occlusion. This approach goes in the path we seek, handling hard contexts as a whole not as specific solutions for each case.

A particularly relevant way of using shape information is the Implicit Shape Model presented in [56], in this shape based method different parts of the object are projected to the plane taking into account

u	Image.
ξ	An hypothetical target pose to be tested
$\widetilde{\mathbb{R}}$	Set of real numbers extended with the value $n.a.$
T	Nb. of high quality training images.
u_t^*	t-th training image.
R^+, R^-	Nb. of positive and negative samples for Boosting.
Q	Nb. of 3D points in the model.
p_q	q-th 3D point in the object model.
$\Pi(p, \xi)$	Projection of point p in the image for object's pose ξ.
$\Psi(u, \xi)$	Pose-indexed feature vector on image u for pose ξ.
D	Feature vector size.
Φ	Trained classifier.
M	Nb. of stumps in Φ.
σ	Threshold function extended to $n.a.$
ρ_m, ω_m	m-th stump threshold and weight.

Table 5.1: LETHA Notation

3D information to improve results thanks to the coherence 3D data introduces to the shape. From the previous work a derivation was presented in [84], in this work we find the notion of sharing features between close viewpoints used in a classifier. This is an attribute that naturally emerges when using point descriptors but that needs to be introduced by design in classifiers, In [84] the authors acknowledge that close poses inherently share common aspects and create a framework for that purpose. Both approaches do not solve the problem of giving a full pose and instead resort again to viewpoint clusters. In any case [84], by using shared features across views, achieves better responses in the boundaries of each viewpoint cluster.

By contrast, approaches relying on global descriptions of the object are less sensitive to a precise localization of individual features. These methods typically use a set of training images acquired from different viewpoints to statistically model the spatial relationship of the local features, either using one single detector for all poses [44, 59, 97, 45] or a combination of various pose-specific detectors [76, 100, 108, 81]. Another alternative is to bind image features with poses during training and have them vote in the pose space [32, 33]. These approaches, though, focus on recognizing instances of a generic class and are not designed to deal with image content different from that in the training set.

Some of the approaches that compute a global descriptor of the object have hinted on the advantages of using a coherent modeling of the objects 3D shape. Both [77, 65] make use of a single pose classifier understanding that if pose is clustered, detection at the boundaries of the cluster will be inefficient and strongly dependent from granularity of the clusters.

Among the methods that compute a global descriptor of the image, we find some holistic representations that do not require extracting any kind of feature. For instance, the GIST descriptor [75] encodes sustained overall orientation of straight edges on images, rather than localized features. This descriptor is conceived more as a class descriptor than as a unique sample identifier, and is not generally robust for discriminating between poses, mainly because it is built using only 2D intensity data, disregarding visibility information of the 3D model. The same applies to the PCA cross-correlation, used in [102] as a similarity measure between tiny images. As it will be shown in the results section, considering visibility constraints in our pose-indexed feature vectors brings a remarkable advantage of our approach against such global descriptors, especially under occlusions. This is because, to account for occlusions, we devote a special treatment to missing data in our feature vector, and design a boosting mechanism able to cope with abstaining weak learners.

The concept on which we want to work, a pose classifier that uses the position of its features to infer the pose of the object, is not a new one. In [28] we could see that by taking into account body part location we could identify its position in the image plane robustly. Similarly, [21] defined subjects of interest as two volumetric ellipsoids and by taking pose indexed descriptors on them was able to detect the subjects od interest and their pose. The main drawback from both approaches is that they are only able of giving rough estimates of pose. Of course, we are solving instance pose estimation and so achieve much better results, as seen in Table 5.2; but with this work we also seek to open new paths that might help obtain greater precision in viewpoint estimation applied to class detection.

Error in degrees	mean	median	std
Our approach	13.67	5.43	21.47
Glasner et.al. [33]	36.44	12.25	55.32

Table 5.2: Results of our approach compared to those published in [33] on the Weizmann Car dataset. We have calculated our pose estimation errors in the same way [33] does. It simplifies pose error to the error in the angle between the correct view and the estimated one. This is not a fair comparison because we train each object instance 3D pose estimation and they do class detection. What this chart shows is how much room for improvement there is if we perform a search in spoce space instead of just clustering the poses.

5.3 The LETHA approach

In this section we describe an implementation of the LETHA learning paradigm, applied to the estimation of the pose of an object in low-quality images.

First, as described in Section 5.3.2, we generate a 3D point cloud model of the object, from which we derive a pose-indexed feature extraction scheme able to compensate for pose changes. Second, as described in Section 5.3.3, these features are combined into a *single* classifier common to all the poses. To handle weak learners abstaining because of self-occlusion, we use a boosting procedure, dubbed AbstainBoost. The search for the optimal pose is a coarse-to-fine process, as described in Section 5.3.5. We first visit exhaustively the poses met in the training set, and then visit more densely around the most promising hypotheses by generating synthetically perturbed poses in their neighborhoods.

5.3.1 Motivation and summary of the approach

Our overall approach consists of reformulating the estimation of the pose of the object of interest in a framework similar to the sliding-window approach for detection: we visit many "poses", and estimate for each a matching score with a *single* trained predictor. The key idea is that the extraction of the features alleviates the training of that single predictor by handling geometrical invariance.

Standard detection with a sliding window

For the sake of simplicity, consider first the sliding-window approach for face detection. Given an image u, it visits a large number R of sub-windows, each defined by a location in the image plane and a scale, and for each of these "hypothetical poses" $\{\xi_1, \dots, \xi_R\}$, it extracts a vector of features $\Psi(u, \xi)$ in the corresponding sub-window, such as the responses of linear filters translated and scaled according to ξ, and feed them to a predictor Φ, such as an SVM or a Boosted linear predictor. The response $\Phi(\Psi(u, \xi))$ should be positive if a face is present there, negative otherwise.

The central idea is that the *same* predictor Φ is used for every window. The way the feature responses are computed ensures that Φ it does not have to cope with invariance to translation or scale.

A remarkable property of this approach, as noticed in [28], is that the "windows" do not really exist. What defines the overall process is (a) a set of poses $\{\xi_1, \dots, \xi_R\}$, and (b) a procedure which to compute a

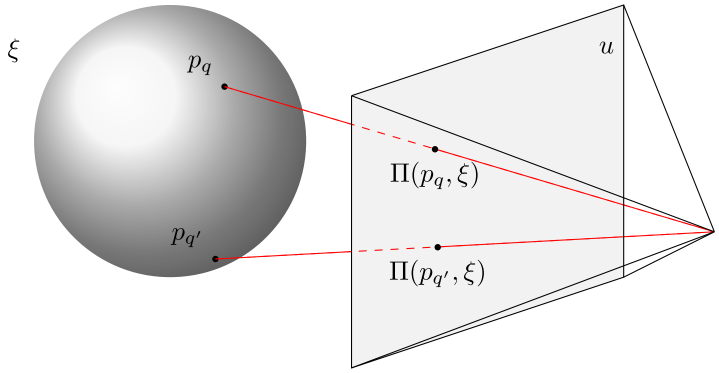


Figure 5.2: **Pose-indexed features description.** Each individual pose-indexed feature is computed as the difference between the gray levels at locations $\Pi(p_q, \xi)$ and $\Pi(p_{q'}, \xi)$, corresponding to the projections of the points p_q and $p_{q'}$ into the image plane u, for the hypothetical object pose ξ. If one of the points is not visible due to self-occlusion, the feature value is $n.a.$

“pose-indexed” feature vector $\Psi(u, \xi)$ for any image u and pose ξ, which accommodates the perturbations due to the pose. These two components are used to produce the training feature vectors to learn Φ, and the test feature vectors to use during detection.

Extension to general 6D poses

We can generalize the same approach to rigid objects, in which case the pose is 6D. For this, Ψ should extract quantities in the image at locations corresponding to 3D points fixed in the object reference frame, and projected in the image according to the hypothetical 6D pose to test.

Our algorithm goes one step further and *learns* from data both the set of poses $\{\xi_1, \dots, \xi_R\}$, and the mapping Ψ. Given high-definition training images, we estimate the camera positions from which we build the set of poses $\{\xi_1, \dots, \xi_R\}$, and a 3D model of the object from which we construct the functional Ψ. In practice, this Ψ computes quantities in the image at locations corresponding to points physically on the object. We extract features with this Ψ, and as in the standard sliding-window case, we train a single predictor Φ common to all the poses.

5.3.2 Learning pose-indexed features Ψ

The first step to learn Ψ from the high-definition training images $u_1^*, \dots, u_T^*$ is to build a 3D cloud model of the object. We use Bundler [95], a SfM system that matches SIFT key-points through iterative bundle adjustment. It generate a dense family of Q points $p_q \in \mathbb{R}^3$ laying on the object's surface, and an estimate of each training image viewpoint pose, from which we derive the object pose in the observer's referential $\xi_t^* \in \mathbb{R}^3 \times \text{SO(3)}, \quad t = 1, \dots, T.$

Then, for each gray-scale image u, and for any pose ξ, let $\Pi(p, \xi) \in \{1, \dots, W\} \times \{1, \dots, H\} \cup \{n.a.\}$ denote the projection into the image plane of u of the point p laying on the 3D model surface, with W and H being the image width and height, respectively.

This projection will take the value $n.a.$ when the point is hidden due to self-occlusion. As Bundler does not provide polygonal information to compute visibility constraints, we compute these with standard z-buffering. For any pair of point indexes $(q, q') \in \{1, \dots, Q\}^2$, we define a pose-indexed feature as the difference between the image intensities at the two projected points (see Fig. 5.2):

$$\Psi_{q+Qq'}(u, \xi) = u(\Pi(p_q, \xi)) - u(\Pi(p_{q'}, \xi)) \,, \tag{5.1}$$

which takes the value $n.a.$ if either one of the projections is $n.a.$ From these features, we define a full pose-indexed feature vector of dimension $D = Q^2$. This whole feature vector is never actually computed. During training, only a random subset of features is evaluated, and during test, the number of features actually evaluated is equal to the number $M << D$ of stumps in the predictor Φ.

5.3.3 Training Φ with AbstainBoost

We want to train a predictor Φ to evaluate from the feature vector $\Psi(u, \xi)$, whether the image / pose pair (u, ξ) is consistent. That is, whether the object of interest is visible in u with the pose ξ.

Stumps and training set

We choose as predictor a linear combination of decision stumps:

$$\Phi(\psi) = \sum_{m=1}^{M} \omega_m \, \sigma(\psi_{d_m}, \rho_m) \ , \tag{5.2}$$

where ψ_{d_m} is the d_m-th feature of the pose-indexed feature vector ψ, ρ_m is the stump threshold, and ω_m is the stump weight, and all these parameters are chosen during training.

As stated in Section 5.3.2, features may take the value $n.a.$, to account for self-occlusion. We use a thresholding function that sends the $n.a.$ values to 0:

$$\sigma(z, \rho) = \begin{cases} 0 & \text{if} \quad z = n.a. \\ -1 & \text{if} \quad z < \rho \\ 1 & \text{if} \quad z \geq \rho \end{cases} \tag{5.3}$$

To build the training set, negative samples are generated by computing, for every single training image u_t^*, the pose-indexed feature vectors using the poses from other training images which have a relative difference with ξ_t^* greater than 10%. Positive samples are obtained by computing the pose-indexed feature vector for poses around ξ_t^* (i.e. we add a 1% relative noise to each component of the pose). The number R^+ of positive samples is a constant factor of the number R^- of negative samples (i.e., 30%).

Following the parallel with the sliding-window face detection, positive samples are taken "around" the actual location of every face, and negative samples are taken "far from" any face.

5.3.4 Obtaining the optimal ρ_m

The parameter ρ_m defines for Eq. 5.3 if we consider a stump to be $true$ (1) or $false$ (-1). It is important to say that when we take a pair of points we want the difference in Eq. 5.2 to be positive. If this value is negative, we flip the points so $p_{q'}$ becomes p_q and p_q becomes $p_{q'}$. Flipping the points does not alter by any measure the stumps and it is just done to make the stumps easier to understand.

The straightforward solution to the value of ρ_m would be to make it 0. If an incorrect pair of pose/image is given to a stump the probability for Eq. 5.2 of being greater than 0 would be of 50%. On the other hand, if a correct pair of pose/image is given to a stump the probability for Eq. 5.2 of being greater than 0 would at least 51%, setting the basis to solve the problem with machine learning techniques.

To show the solution we developed we first want to reference Fig. 5.3 and to detail what appears in it. We have taken training samples of both consistent and inconsistent pose/image pairs. A pair formed by a pose and an image is considered consistent if the object appears in that pose in the image. An inconsistent pair is formed by an image and a pose that do not match together. In Fig. 5.3 we have generated random stumps on the F1 dataset and then taken all consistent pairs of images/poses in which that stump is visible.

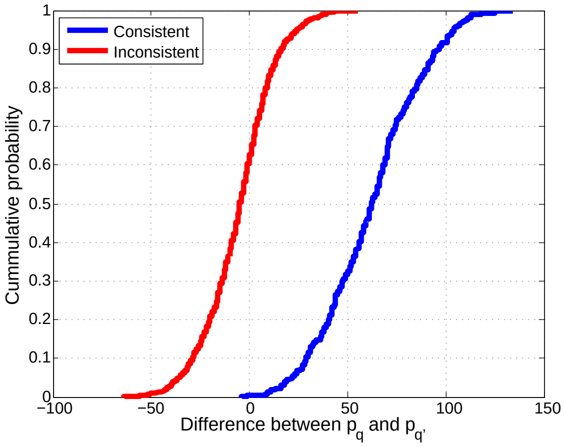


Figure 5.3: **Cumulative probability distribution of consistent and inconsistent samples for random stumps.** The response here showed is the difference of gray intensity between the two points that belong to a stump. Consistent samples are made by the image and the correct pose for the object in that image, inconsistent samples contain the same pose but incorrect images. We have selected a stump that appears in many training images to more meaningfully show the case in hand.

For those same stumps we have changed the correct image for an incorrect one to generate inconsistent samples. For those consistent and inconsistent samples we have calculated the result of Eq. 5.3 and created a cumulative distribution for each. This can be seen in Fig. 5.3.

Now that context has been established lets get back to the computation of ρ_m. The biggest gap between both probability distributions the consistent and the inconsistent samples will normally not be when ρ_m is 0. The gap between both distributions defines the strength of the stump The bigger the gap the more consistent samples will respond 1 and less inconsistent samples will respond 1. We thus find for each stump the value of ρ_m that maximizes the separation between the consistent and inconsistent data provided by the training samples.

By defining ρ_m in the way we have detailed we make each stump adapt to the characteristics of each object, we greatly improve our performance in contrast to defining ρ_m at 0, and we avoid user defined thresholds while not increasing the computational cost. By transforming regular intensity gradients into our more robust stumps we make the classifier able to solve the training samples with less weak learners.

AbstainBoost

A classical method to build a linear combination of stumps from a training set is Adaboost, which selects stumps one after another to reduce the exponential loss in a greedy manner [63]. The standard derivation of this procedure relies on all the weak learners having the same L^2 norm. If we define

$$W_\tau = \sum_{n:y_n h(\psi_n)=\tau} \exp(-y_n \Phi(\psi_n)), where y_n \in \{-1,1\} \tag{5.4}$$

is the label of the n training samples, and ψ_n the corresponding feature vector, Adaboost chooses weak learners h maximizing

$$|W_{+1} - W_{-1}| \tag{5.5}$$

However, as stated in Eq. (5.3), we have to deal with zero-valued responses, hence weak-learners of various norms. Relying on the inner product between the weak-learner's responses and the sample

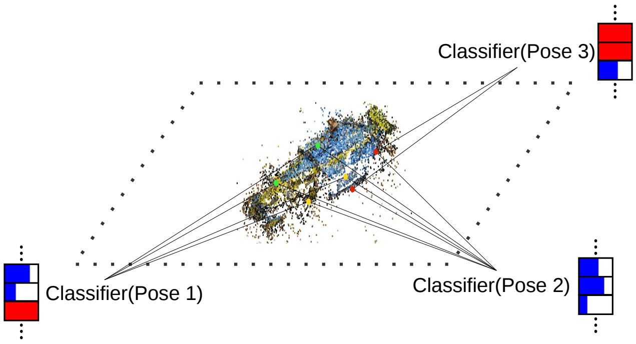


Figure 5.4: **Classifier tested on 3 different poses.** Due to the geometry of the object, the 3D points that conform the stumps are visible only from certain viewpoints. This makes that $Pose_1$ sees only the first two stumps, $Pose_2$ all the stumps and $Pose_3$ only he third stump. Next to each camera center we show the response of the classifier for that marked stumps. If the stump in the weak learner is not visible it is shown in red if it is visible we show the intensity of the response in blue. By doing this we only use the relevant information in each pose and at the same time avoid creating and training multiple classifiers for each viewpoint.

weights as an indication of "good direction" in the functional space, as Adaboost does, is incorrect and leads in practice to weak learners with fewer zero responses, even if they are often incorrect, because they have larger norm. With weak learners taking values in $\{-1, 0, 1\}$, we derived analytically (appendix in chapter 9) that the optimal weak learner – i.e. the one inducing the maximum reduction of the exponential loss when added with its optimal weight ω – is the one maximizing

$$|\sqrt{W_{+1}} - \sqrt{W_{-1}}|. \tag{5.6}$$

This is the criterion we used by the AbstainBoost procedure. As for Adaboost, we can find the best stump's threshold in a time linear with the number of samples after they have been sorted according the feature's value, and the optimal weight remains $\omega_m = \log \sqrt{W_{+1}/W_{-1}}$. AbstainBoost's derivation is similar to the GrowRule operation in the Slipper algorithm [15], with the main difference that it allows to directly select signed abstaining decision stumps, instead of being applied to a greedy construction of Boolean disjunctions.

To summarize: given the training set, and the stumps defined on Eq. 5.3, our learning procedure consists of M AbstainBoost iterations, each one sampling at random several feature indexes $1 \leq d \leq D$, and keeping the one that maximizes the abovementioned score. The corresponding stump is then added to the strong classifier, and the process is re-iterated. To clearly show what happens to weak learners when they abstain we introduced Fig. 5.4. Then figure shows a simple example of weak learners abstaining due to the different poses with which we test the classifier.

5.3.5 Coarse-to-fine pose estimation for testing

The test proceeds in a two-step "coarse-to-fine" manner, visiting first the poses seen during training, and then focusing on the best ones by visiting another set of poses generated in their neighborhoods. For each visited pose, the classifier response is computed as depicted in Figure 5.6.

More precisely the process first loops through the T training poses $\xi_1^*, \ldots, \xi_T^*$, and for each, it projects the Q object model points onto the image plane, and creates the pose-indexed feature vector $\Psi(u, \xi_t^*)$.

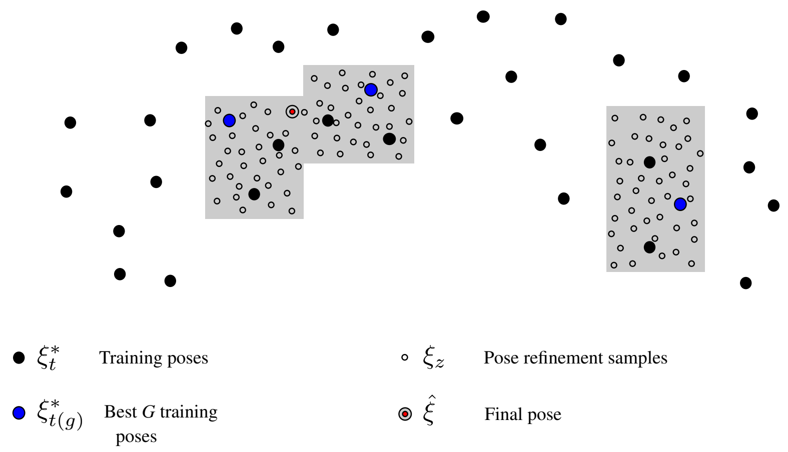

● ξ_t^* Training poses
○ ξ_z Pose refinement samples
● $\xi^*_{t(g)}$ Best G training poses
◉ $\hat{\xi}$ Final pose

Figure 5.5: **Refining the final pose.** A bounding box in pose space is computed around each of the G poses that have maximal classifier response and their two nearest neighbors. Z new poses are uniformly sampled inside these boxes, keeping as final pose that with maximum classification score.

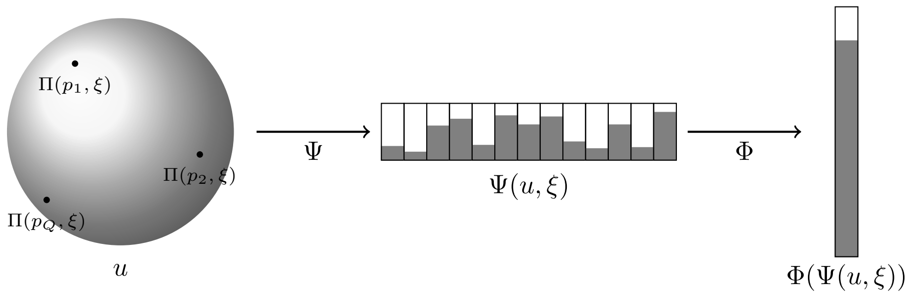


Figure 5.6: To evaluate the classifier on a test image u, for an hypothetical pose ξ, the algorithm first computes the feature vector $\Psi(u, \xi)$, and then the response of the predictor $\Phi(\Psi(u, \xi))$.

This feature vector is used to evaluate the classifier response $\Phi(\Psi(u, \xi_t^*))$, and the G poses with the highest responses are retained

In a second step, the final pose is refined by reevaluating the classifier on a set of poses generated synthetically in the neighborhoods of the best G poses retained. We first set a hyper-box around each one of the G best poses by defining a minimum and maximum value for each single component, using the pose itself and its two closest neighbors, with an additional 10% relative margin on each component (see Fig. 5.5). We sample uniformly Z poses is in each box, and evaluate the classifier for each. The final pose $\hat{\xi}$ is the one with maximum response.

Note that the core property of our algorithm is that since we perform measurements at locations *recorded on the training images*, we do not require any point of interest detection in the test phase, since we know where the image intensities should be measured for each hypothetical pose we test. This makes the algorithm appealing for low-resolution or severely corrupted images.

In addition, we will show our approach can be used in conjunction with a generic object detector [22] that provides many hypotheses about the object location. In this case, LETHA will be used to prune out false positive responses and use the remaining one to accurately estimate viewpoint.

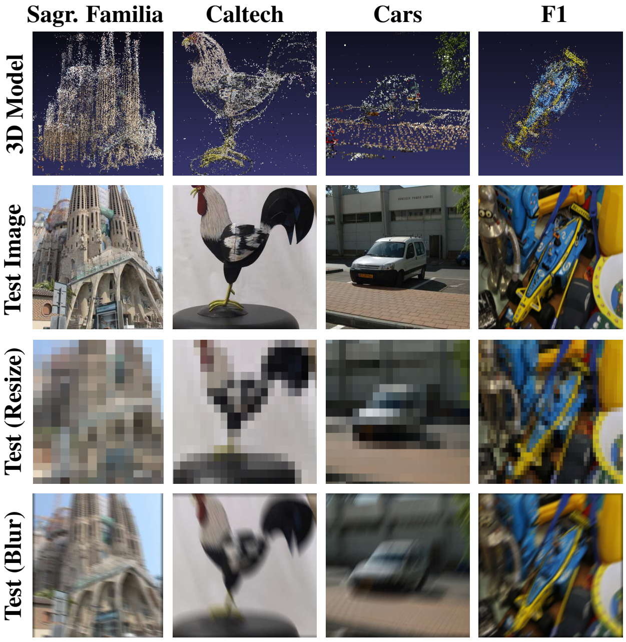


Figure 5.7: **3D model and sample images used in our tests.** The images shown correspond to one original test image for each dataset along with the maximum point of degradation for resolution loss and motion blur reached in our experiments. Note how tn the F1 dataset, image quality degradation is combined with the strong occlusions present in the testing data.

5.4 Experiments and results

We next describe our experiments: the datasets we use for the evaluation, the competing approaches, the parameters used for learning the LETHA classifier, and the results.

5.4.1 Datasets

We use four datasets for evaluation (see Figs. 5.1 and 5.7 for some examples). For all of them but the last one, half of the available images were randomly selected for training (i.e. building the 3D model and training the classifier), and the remaining ones were used for testing. The internal camera parameters for each image are known in all datasets.

• **Sagrada Familia dataset**: Composed of 478 images of resolution 718×480 of this church located in Barcelona. They were taken along two loops around the building, at street level, every meter, at different daytimes and with different weather conditions. The images taken from the first day were used for training and the images from the second day were used for testing. There are occlusions due to pedestrians, buses, and trees, and changes of daylight illumination. The strong changes in illumination can be seen in Fig. 5.11.

• **Caltech dataset**: The objects from the *CalTech Turntable* dataset [68], imaged 360 times at 2048×1536 resolution and 1 degree intervals, in a controlled environment with constant lighting and textureless background.

• **Cars dataset**: Sequence used in [32]. It is composed of 1840×1224 images of 21 different cars observed under 68 viewpoints on average. There is a small number of instances per class, which makes learning more difficult.

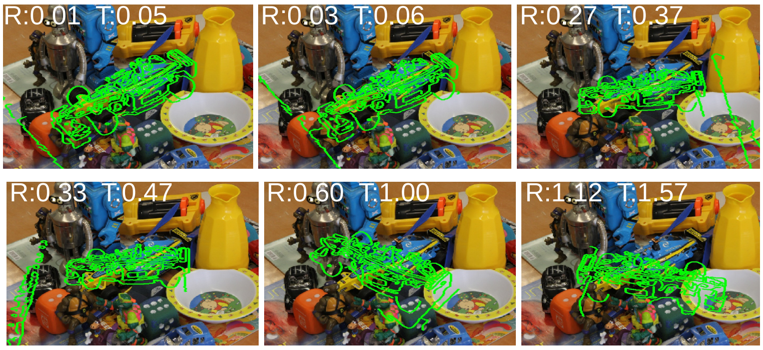


Figure 5.8: **We show examples of increasing amounts of error to ease the understanding of the result charts.** R is for rotation error and T is for translation error. The image shows the error between a test image and one of the training images. Edges from the corresponding training image are superposed on the test image to show the variation of pose. For clarity, we have used the same test image to make comparison of the error magnitudes easy. This figure is purely informative and the errors shown are not the actual errors yielded by our approach or any of the baselines.

• **F1 dataset**: This set is the only one where the training and test images were generated differently. It contains 317 calibrated training images of resolution 480×718, showing a F1 model car on an empty table, and 336 test images of the same F1 model car but in a heavy cluttered environment.

5.4.2 Baselines

Our first choice for a baseline was a purely geometric method relying on local feature point extraction with RANSAC-based geometrical pose estimation. Given a high definition 718×480 image of the Sagrada Familia dataset, and its closest pose-correspondent 143×96 image, we extracted points of interest in both images using DBRIEF [105], computed the number of inliers after matching features and estimated the pose transform using EPnP[69] with RANSAC. Due to the large amount of mismatches and localization biases of the same 3D in both images, even after 50,000 RANSAC iterations (which took 5 minutes to compute in a standard PC) the algorithm was not able to converge to a correct pose. Obviously the problem would be magnified if, for a given test image, all training images had to be evaluated, and thus we discarded RANSAC-based approaches from subsequent analysis.

The baselines we chose are state of the art methods of both geometrical and global approaches:

• **PCA-NCC**: PCA normalized cross-correlation [102] is a good candidate, as it does not require an outlier rejection stage to compare a pair of images.

• **GIST**: The Gist descriptor [75] uses information of the entire image, and as suggested by [102] it is an appropriate approach to compare very tiny images.

• **BoF**: As a representative of the methods that build a global descriptor of the image from local features we used a Bag of SIFT Features (BoF) [107].

• **DBRIEF**: We used the average confidence of individual DBRIEF matches [105]. It uses a similar scheme as ours, it trains a dictionary of features by learning the appearance changes over a 3D model. This is done to ensure robustness to 3D deformations with the same intention as LETHA.

5.4.3 Training and testing with LETHA

Using Bundler, we built the 3D models for each dataset. The size of these models ranges from about 1×10^5 points for the Sagrada Familia dataset down to 2×10^4 points in the F1 dataset. For the Cars

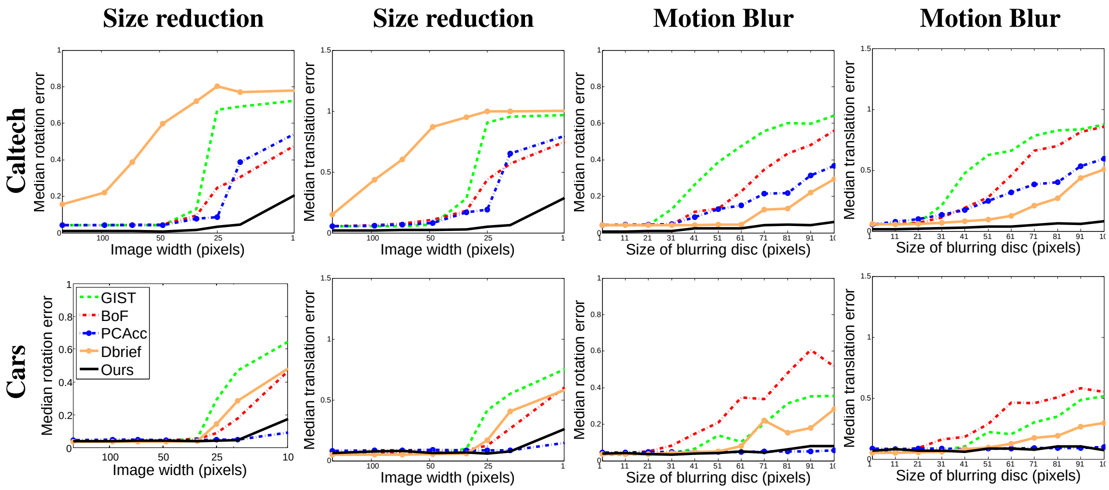


Figure 5.9: **Caltech and Weizmann datasets results.** Pose estimation error of LETHA and other approaches in experiments with severe degradations of image size and motion blur. Results over the Caltech dataset and the Weizmann car dataset.

dataset we built a different 3D model for each object class.

Training images are initially convolved by Gaussian filters with standard deviation of 2 pixels. Then, following the methodology described in Section 5.3.3, for each of the T training poses, we generate R^+ positive and R^- negative samples. Training images are initially convolved by Gaussian filters with standard deviation of 2 pixels. Then, following the methodology described in Section 5.3.3, for each of the T training poses, we generate R^+ positive and R^- negative samples, according to the procedure described in Section 5.3.3. This results, for each training image, in a total of 52 samples for the Cars dataset, 240 for the Sagrada Familia dataset, 300 for the Caltech dataset, and 412 for the F1 dataset. The full sample set used to train the predictor Φ has a size ranging from $10,000$ samples to $350,000$ samples, out of which $1/4$-th are positive samples and $3/4$-ths negative ones.

5.4.4 Results

In all experiments we describe below, we compute the pose of a corrupted test image with each of the algorithms. For all competing methods, the estimated pose will be the one of the most similar training image. For LETHA it is computed as described in Section 5.3.5. Let $\hat{\xi} = (\hat{\mathbf{q}}, \hat{\mathbf{t}})$ be that estimated pose, where $\hat{\mathbf{q}}$ is a normalized quaternion representing the rotation and $\hat{\mathbf{t}}$ the translation vector. Similarly, let $\xi_{\text{true}} = (\mathbf{q}_{\text{true}}, \mathbf{t}_{\text{true}})$ be the ground truth pose of the test image. As in [69], relative rotation and translation errors are computed as

$$\mathbf{E}_{\text{rot}} = \|\mathbf{q}_{\text{true}} - \hat{\mathbf{q}}\| / \|\hat{\mathbf{q}}\|, \tag{5.7}$$

$$\mathbf{E}_{\text{trans}} = \|\mathbf{t}_{\text{true}} - \hat{\mathbf{t}}\| / \|\hat{\mathbf{t}}\|, \tag{5.8}$$

respectively. In all discussed experiments we compute the median error of all testing images over different configurations. To show further insight on what to expect for different magnitudes of error, different cases of errors have been drawn for a test image in Fig. 5.8.

Note that LETHA refines the pose by sampling around the training poses with highest scores. Yet, given a training pose, the reduction in pose error when using this fine estimation is very small. The real

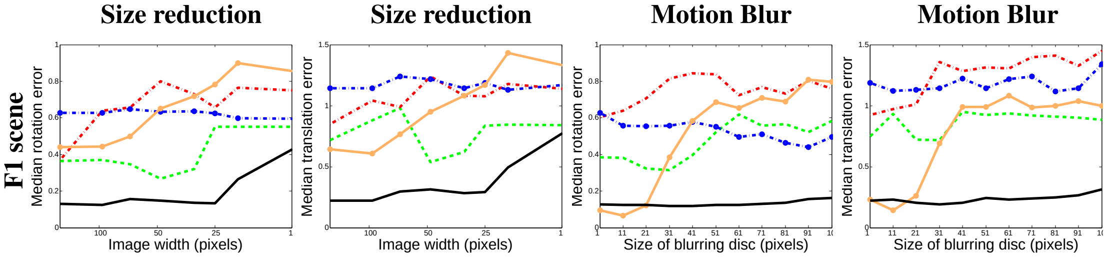


Figure 5.10: **F1 dataset results.** Pose estimation error of LETHA and other approaches in experiments with severe degradations of image size and motion blur. Charts depict the results on the F1 dataset which contains strong occlusions.

benefit of re-sampling around a few training poses, is that the final chosen pose may be the result of sampling around a training pose which initially did not have the highest score.

The list of imaging artifacts and contexts on which we have tested the performance of our approach include viewpoint variance, motion blur, resolution loss, lighting changes and occlusions. Viewpoint variance is tested on all datasets due to having images from many different points of view. Motion blur and resolution loss artifacts can be produced artificially with a high amount of resemble to artifacts naturally encountered. Producing meaningful occlusions and changes in illumination is not as trivial, for this purpose we created the Sagrada Familia and the F1 datasets. For this reason we will test viewpoint variance, motion blur and occlusions on all datasets but occlusions only in the F1 dataset and lighting variance in only the Sagrada Familia dataset. It is also important to underline that the features used and the training has only been performed once on each dataset and without changes or adjustment of any parameter.

Resolution loss and blurring

We evaluated all methods in two different situations: reduction of the image size and motion blur (see Fig. 5.7). Let us first focus on the Sagrada Familia, Caltech and Cars datasets for which the amount of occlusions produced by external objects does not exist or is relatively small.

The first three rows of Fig. 5.9 summarize these results. For the "size reduction" experiment both PCA-NCC implementations and our approach show high robustness. For instance, in the Sagrada Familia dataset this means that the algorithms are capable of finding the right pose for a test image as small as 14×10 pixels compared to the 718×480 size of a training image. The performance of our approach and of PCA-NCC degrade for larger reduction sizes. Note that PCA-NCC, despite being a relatively simple approach, takes advantage of the fact that it uses all pixels in the image. The rest of methods that either rely on combination of local features (BoF or DBRIEF), or orientations of straight edges (GIST), generally rapidly fail for moderate reductions in size, when these features are prone to disappear.

The performance in the "motion blur" experiment is similar. Both PCA-NCC and LETHA clearly outperform other techniques in the Sagrada Familia and Cars dataset. In the Caltech dataset, though, LETHA is consistently more robust than PCA-NCC. In all experiments, relative pose errors below 10% are comparable to those that would be obtained using a purely geometric method, such as the EPnP [69] when using high resolution images with no degradation.

Resolution loss and blurring + Occlusions

Let us now focus on the experiments for the F1 dataset depicted in Fig. 5.10. As shown in Fig. 5.1, the test images contain strong occlusions of the object which were not included in the training set. On top of

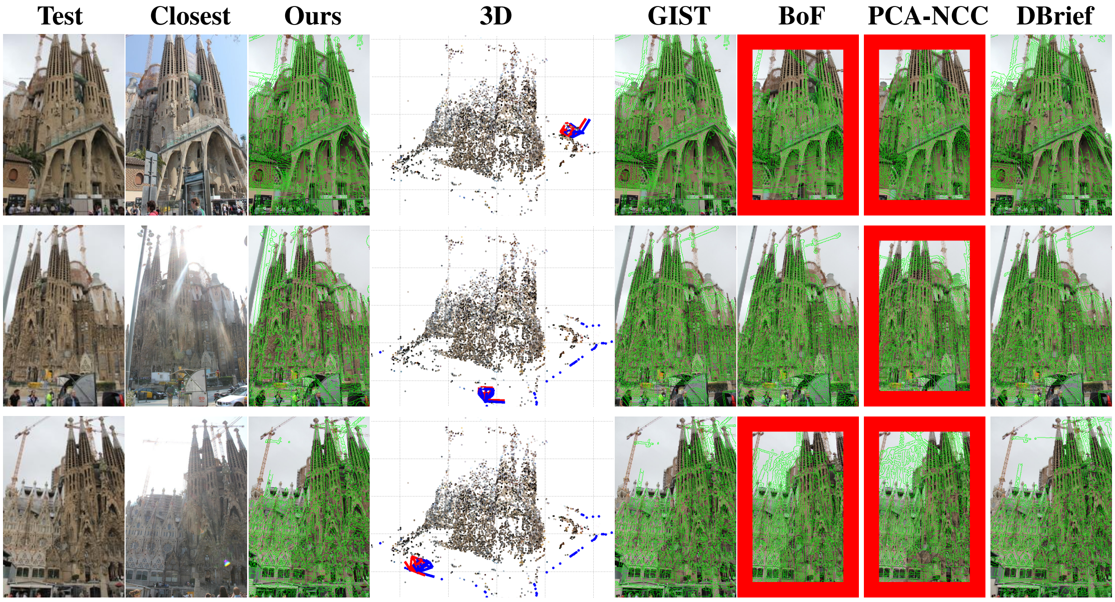


Figure 5.11: **Sagrada Familia dataset results.** Examples of the results obtained on the Sagrada Familia dataset. On the first two columns we can see the big differences between the testing images and the training images of which we show the closest. The thrid column show the obtained result with our method by reprojecting the boundaries of the closest training image to our prediction. The fourth column shows the ground truth camera in blue and in red the camera we found as prediction. The other columns show results of the baselines in the same fashion as ours. A red border indicates that more than 20% error was produced.

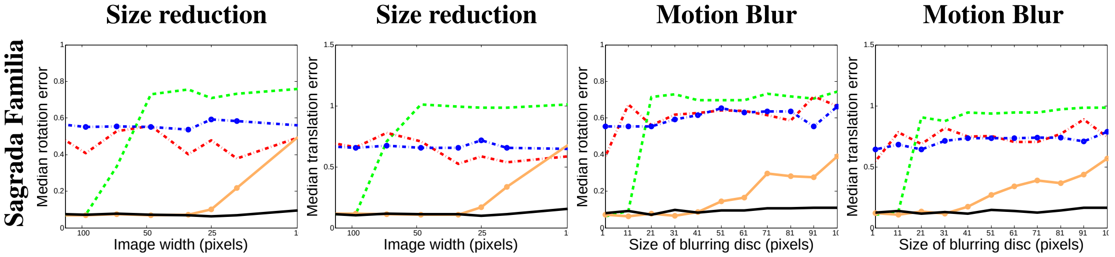


Figure 5.12: **Sagrada Familia results.** Pose estimation error of LETHA and other approaches in experiments with severe degradations of image size and motion blur. Results over the Sagrada Familia dataset which includes strong illumination changes due to images taken at different days under different meteorological conditions.

this, we also considered the image degradation artifacts used above. In this situation neither the methods that use information of the entire image (PCA-NCC and GIST), nor the methods based on local features (BoF and DBRIEF), are able to succeed. LETHA, in contrast, exploits all its properties to yield robust results. On the one hand, the fact that the location of the features for each pose is known in advance alleviates the problem of feature detection when the image is severely deteriorated. On the other hand, it also exploits the fact that for each view only local features that are not affected by self-occlusion are considered. This maximizes the chance of obtaining a large number of visible features, even when the target object is partially occluded by external objects.

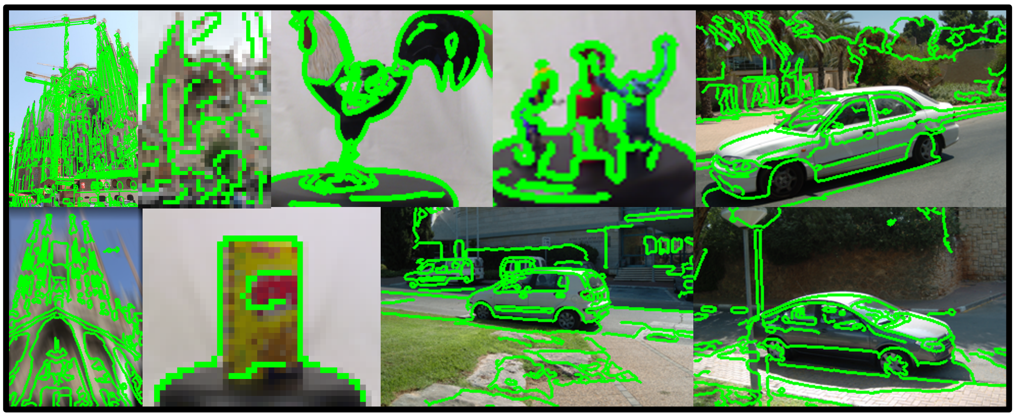


Figure 5.13: **Reprojected results for LETHA.** Pose estimation results represented as reprojected wireframes of the best matching candidate in the training data. The fact that the wireframes do not match exactly in some cases is because sampling for pose refinement produces better pose estimates than what can be obtained from the closest image in the training set. Unfortunately projecting the 3D model point cloud clutters the image in an unintuitive way.

Resolution loss and blurring + Lighting Changes

As our features use difference of intensity and not a fixed value we have a certain theoretical resistance to illumination changes. To achieve complete invariance we would need to make use of boundary information, however this would render us unable to handle blurring effects. We chose gradient on the image intensity as the best all-around information to use, this way we can handle the widest range of imaging conditions possible. To show our performance on scenes with illumination changes we have created 2 datasets of images on 2 different days, one really sunny and he other a cloudy day. Examples of both datasets can be seen in Fig. 5.11. We outperform all competing methods, as seen in Fig. 5.12, to the point in which two of them, PCA-NCC and BoF, are uncapable of handling the lighting changes. GIST and DBRIEF on the other hand are able of handling the changes but break down sooner when motion blur or image size degrade the image.

5.4.5 Computational cost

Given a test image, the time to estimate the pose for the different experiments is shown in Table 5.3. Note that all methods are about the same order of magnitude. DBRIEF is slow because we have used its MATLAB implementation. In addition, while it is fast in extracting the features, the process of matching a large number of them is slow. To end the result section we provide qualitative results of the obtained poses that we achieved with our approach in Fig. 5.13.

	GIST	BoF	PCA-NCC	DBRIEF	LETHA
Sagr. Fam	1.19	0.12	0.27	8.30	1.64
Caltech	0.89	0.10	0.34	7.90	1.56
Cars	0.67	0.19	0.38	2.09	0.82
F1	1.56	0.22	0.47	17.1	4.34

Table 5.3: **Execution time table for LETHA.** Time (seconds) required to compute the pose of an input image for all experiments and methods. Note that BoF is implemented in C while the other methods are in MATLAB

5.5 Conclusion

We have proposed a new machine learning paradigm: Learning with high-quality data to be able to test with low quality data. The rationale behind this idea is that inference is possible only from clean data, or using a strong model, and that the latter can be inferred from the former.

From this general principle, and extending the concept of pose-indexed features to be able to learn them, we have derived a novel and very efficient algorithm for the specific problem of pose estimation. As demonstrated on the F1 data-set, with sufficiently good training data, we obtain an extremely good estimate of the object pose, in very low resolution images, and with high levels of noise and occlusion.

This procedure is promising as a near-perfect solution to be used in controlled environments such as a factory. Our future work will aim at extending it to multi-target detection, richer poses, and class-level detection.

Chapter 6

Pose Estimation without Color Information.

6.1 Introduction

Since the emergence of commodity depth sensors in the past few years, recognizing objects and estimating their pose using such depth sensors is an active research topic. Several approaches demonstrate that the results for this task can be improved over methods using only color images by combining RGB and depth features (e.g. [38, 85, 98]), but in many situations color cues are either not available, not informative or unreliable. In particular, in the context of automated manufacturing and mechanical assembly, objects may need to be recognized and their pose estimated from depth data only.

In contrast to color images, depth maps are usually far less discriminative in their appearance. While a good statistical model for color images is still an open research topic, a sensible and simple prior for depth images is given by a piecewise smooth regularizer. Consequently, we do not rely on any interest point detection in depth images and evaluate features densely (or quasi-densely by subsampling) in the query image. Further, real depth sensors exhibit several shortcomings at depth discontinuities, such as occlusions and foreground flattening occurring with triangulation-based sensors (passive stereo or Kinect-type active stereo), and mixed pixels with time-of-flight sensors. Overall, many depth sensing technologies report reliable and accurate depth values only in smooth regions of the true scene geometry. Beside that, the piecewise smooth appearance of range images also implies that extracting a full 3D local coordinate frame is not repeatable, but at least estimating surface normals is rather reliable. Thus, feature extraction can be easily made invariant with respect to two degrees of freedom (i.e. the surface normal) but not reliably invariant with respect to the remaining 2D rotation in the local tangent plane. We also believe that for the same reason predicting poses directly based on feature correspondences leads to large uncertainties in the estimates, and therefore we follow [93, 5] in predicting “object coordinates” (i.e. 3D vertices on the object of interest) and computing more certain and accurate poses from multiple correspondences.

Finally, objects of interest can be occluded and only be partially visible. A sensible principle to add robustness with respect to occlusions is to employ a compositional method, i.e. to detect the object and estimate its pose by detecting and aligning smaller parts. Due to the locally ambiguous appearance of depth images, we expect a much higher false-positive rate than with color images when matching features extracted in the query images with the ones in the training database, and it will be essential to maintain several predictions of object coordinates per pixel to address the amount of false positive matches. In summary, object detection solely from depth data is facing the following challenges: (i) few salient regions in range images, (ii) unreliable depth discontinuities, and (iii) uninformative features and descriptors.

Since depth cameras report 3D geometry, and our method is based on predicting 3D object coordinates for pixels in the range image, we are able to assess the internal consistency of putative object coordinates by comparing the distance between two observed 3D points (back-projected from the depth map) and the predicted object coordinates. Grossly deviating distances indicate that at least one of the predicted object coordinates is an outlier. Thus, one can easily avoid sampling and evaluating pose hypotheses from outlier-contaminated minimal sample sets by scoring this (pairwise) consistency between predictions and observed data.

If one interprets the object coordinate hypotheses per pixel as unknown (or latent) states, then the pairwise consistency of predicted object coordinates plays the role of pairwise potentials in a graphical model. Consequently, it is natural to employ the methodology of inference in graphical models in this setting in order to rank sets of putative object coordinates by computing respective min-marginals. In contrast to the standard use of graphical models with respect to images, which usually defines a random field over the entire image. We utilize many but extremely simple graphical models whose underlying graph has exactly the size of the required minimal sample set.

Robust geometric estimation is typically addressed by data-driven random sampling in computer vision. A standard RANSAC-type approach for rigid object pose estimation would randomly draw three object coordinate hypotheses (not necessarily using a uniform distribution) and evaluate the induced pose with respect to the given data. On a high level view RANSAC generates a large number of pose hypotheses and subsequently ranks these. We reverse the direction of computation: our method considers a large number of overlapping minimal sample sets and removes the ones clearly contaminated with outliers by utilizing the consistency criterion. Since the minimal sets are overlapping, applying the consistency criterion to a pair of putative correspondences is able to discard several minimal sample sets at once. We believe that our approach is an elegant solution to generate promising sample sets for robust (pose) estimation exhibiting very few inlier correspondences.

6.2 Related work

Object detection from 3D data has been widely researched during the past decade. Initially many solutions focused on trying to solve object detection from laser scans or even from synthetically generated meshes [47, 66, 18]. However, with the popularization of RGB-D sensors since 2010 there has been an increasing demand of algorithms [38, 85, 98, 5] that operate at interactive frame rates and that are able to cope with inputs that are less reliable than laser scans. Most of the latter algorithms rely heavily on RGB data to perform detection, which prohibits the application of these methods on 3D only inputs. Several approaches [38, 85, 98] use a global description of the object (using RGB edges and depth normals), hence these methods have difficulties in handling occlusions. Brachmann et al. [5] on the other hand compute features densely for each pixel, and subsequently apply a regression forest followed by pose scoring to determine detections. Because it uses a dense description of local features it is able to address occlusions. The biggest advantage of RGB-D algorithms over methods that rely solely on 3D or depth data is their capability to deliver up to real-time performance. Further, these methods are able to cope with noisier data returned by commodity depth sensors.

Methods that utilize only 3D data as input can be based on either global or local object representations. Several proposed methods based on a global object representation employ the Hough transform [52, 82, 110]. These approaches create a set of features that are accumulated in a Hough voting space and then select the pose which gathered the largest number of votes. Like in the RGB-D case, global descriptions suffer again when strong occlusions are present. Several local descriptor-based approaches find salient points in the point cloud and then obtain invariant descriptions of the regions around them [88, 47, 2, 86].

The main problem with this approaches is that 3D information, in contrast to RGB, is usually quite uninformative (many flat surfaces or similar curves) and one cannot find a sufficient number of reliable features in many situations.

Spin images [47] are among the successful local descriptors to recognize 3D shapes. Applied on 3D shapes the spin image is a revolution histogram describing the local surface, and (due to alignment with the local surface normal) it is invariant to 1D rotations in the tangent plane. Mian et al. [66] also use the normal to obtain an invariant descriptor: they fix the local coordinate frame by using two points on the model (in additional to the normal), and fill an occupancy grid given the local coordinate frame. In this work it was also shown that the use of occupancy grids is more discriminative than the use of spin images.

Similar to [66], Drost et al. [18] fix the local coordinate frame of the shape descriptor by choosing two vertices, but their descriptor is based on the distance and the geometric relation of the normals at the chosen surface points instead of using an occupancy grid. This descriptor design makes Drost's features less discriminative than Mian's but also faster to compute. Altough Drost's features are less informative, by using all possible combinations of two points in the model, given an initial point, he was able to construct an overall more informative description. One of the most important contributions of this work is that it cannot be classified as either a local or a global description but rather as both; by taking pairs of points, Drost is able to obtain invariance to occlusion like other local approaches and at the same time by creating a feature using every point in the model it makes the approach robust to non informative objects. The true potential of Drost's feature is shown in [106], by applying several learning techniques they are able to reduce the number of features required obtaining a reduced more informative set of pair features. When compared to all previous approaches they clearly outperform all of them in both computation time and accuracy.

Although these techniques using only 3D data as input obtain very good results, they are designed to work with relatively clean data such as laser scans. The effect of noise on detection rates is not assessed in most cases. Another big drawback are the computation times, since none of these algorithms is able to perform close to real time speeds. In this chapter we show that our proposed approach is able of handling a noisy sensor while performing at several frames per second. Another challenging aspect not explicitly addressed in [47, 66, 18, 106] is handling objects with highly self-similar local shape appearance (e.g. surfaces of revolution or objects with multiple symmetries).

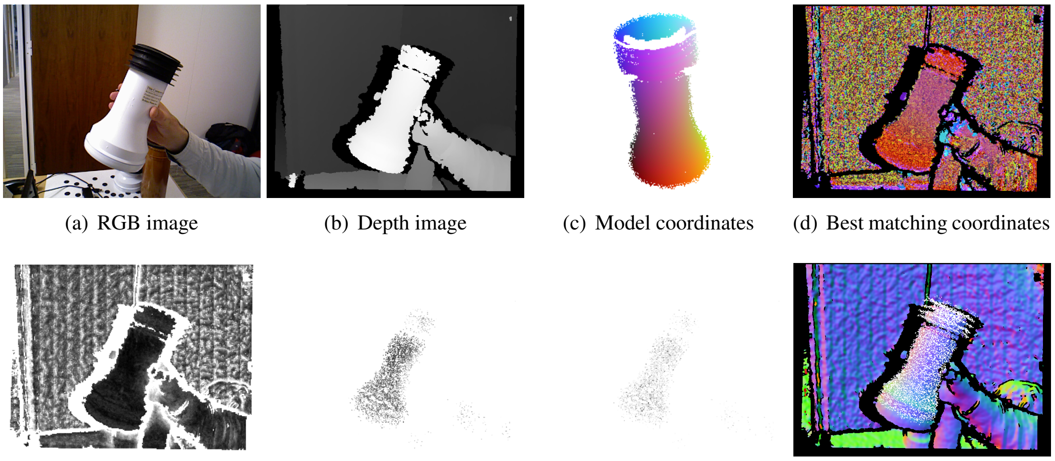
If local minima were of no concern, then estimating the pose of a rigid object given depth data amounts to registering two meshes (a given object of interest and the current depth observation), which can be solved by the ICP algorithm [4] or one of its robust variants. It is well known that ICP requires a good initial estimate to converge to a sensible solution. Ultimately, all methods to detect 3D objects in either depth-only or RGB-D data aim to provide a good initializer for an ICP-like refinement procedure.

6.3 Our Approach

Before we describe our method in detail, we provide a high-level overview (see also Fig. 6.1): at test time the algorithm maintains a set of putative matching object coordinates for each pixel in the test image (Figs. 6.1(d,e)). Instead of sampling minimal sets of correspondences required for (rigid) pose computation, the utility of pairs of correspondences is assessed by using the consistency with the observed depth data. Triplets of correspondences are ranked (Fig. 6.1(f)), and finally promising ones are evaluated using a standard geometric criterion (Fig. 6.1(g)) to determine the best-scoring object pose (Fig. 6.1(h)).

6.3.1 Descriptor Computation

Given the nature of depth maps and the problem of detecting objects that occupy only a fraction of the image, we opt for a dense (or quasi-dense) computation of descriptors in order not to rely on unstable



(a) RGB image (b) Depth image (c) Model coordinates (d) Best matching coordinates

(e) Feature (Hamming) distance (f) Self-consistency (g) Pose score (h) Superimposed model

Figure 6.1: **Method overview.** This is a depiction of the consecutive steps taken by our approach to estimate the final pose of a known object. (a) input RGB image (for illustration purpose only); (b) input depth image; (c) view on the trained CAD model with color coded object coordinates; (d) best matching object coordinates for the input to illustrate the level of false positives; (e) the corresponding minimal feature distances, which also serve as unary potentials in Eq. 6.4; (f) the smallest min-marginals Eq. 6.6 per pixel; (g) the geometric pose scores (Eq. 6.11) after pose refinement; and (h) points of the model superimposed according to the best pose estimate.

salient feature points.

A natural choice for a descriptor to represent (local) geometry is based on an implicit volumetric representation of range images and 3D surface meshes. We employ a binary occupancy grid to compute descriptors. A slightly more discriminative volumetric data structure would be a (truncated) signed distance function (TSDF), but we discard this option for efficiency reasons (proper TSDF computation is costly, and the descriptors would use several bits per voxel). We believe that using generalizations of successful gradient-based image descriptors to 3D shapes (such as 3D-SURF [52]) is not necessary, since the intensity values of the 3D image are known to be only 0 and 1 for occupancy grids (and therefore invariance to intensity transformations is unnecessary). Consequently, our descriptor is a bit string of occupancies in the vicinity of a surface point.

In order to obtain some degree of invariance with respect to viewpoint changes, the z-axis of the local coordinate frame at a surface point is aligned with the local surface normal. Given the piecewise smooth characteristic of range images, normals can be estimated relatively reliably for most pixels (after running a Wiener filter to reduce the quantization artifacts observed in triangulation-based depth sensors). For the same reason computation of the second principal direction is highly unreliable and not repeatable. Therefore we compute several descriptors at each surface point by sampling the 2D rotation in the tangential plane (we sample in 20° steps resulting in 18 descriptors per surface point).

Instead of storing a full local occupancy grid (centered at a surface point) we use a subset of voxels (512 in our implementation, i.e. our descriptors are 512 bits long). We initially utilized a conditional mutual information based feature selection method [27] to determine the most informative set of voxels, but this procedure turned out to be rather slow even with the lazy evaluation technique. The reason is that many voxels are not very discriminative, and their respective conditional mutual information is similar. By running feature selection on example training data, we observed that only voxel positions near the tangent plane are selected. Thus, we decided to randomly sample voxel positions in a box aligned with the tangent plane that has half the height than width and depth (we use 8cm × 8cm × 4cm boxes). This means, that building the descriptors from the given depth images or training meshes is very fast.

6.3.2 Matching

At test time descriptors are computed for each pixel with valid depth and estimated surface normal in the (sub-sampled) depth image, and the task is to efficiently determine the set of object coordinates with similar local shape appearance. The natural choice to quantify similarity of binary strings is the Hamming distance. We experimented with approximated nearest neighbours implementation for binary data in FLANN [71] and with a hashing based indexing data structure using orthonormal projections [34]. In our experience the performance is roughly similar for both acceleration strategies, we only report the results using FLANN below.

6.3.3 Pairwise Compatibility

The matching step returns a list of object coordinate candidates for each pixel with attached descriptors. Even without generating a pose hypothesis it is possible to assess the quality of pairs of putative correspondences by exploiting the information contained in the range image. If p and q are two pixels in the query range image, and $\hat{X}_p$ and $\hat{X}_q$ are the respective back-projected 3D points induced by the observed depth, and X_p and X_q are putative correspondences reported at p and q, then a necessary condition for $\hat{X}_p \leftrightarrow X_p$, $\hat{X}_q \leftrightarrow X_q$ being inlier correspondences is that

$$\|\hat{X}_p - \hat{X}_q\| \approx \|X_p - X_q\|. \tag{6.1}$$

If the Euclidean distance between $\hat{X}_p$ and $\hat{X}_q$ deviates substantially from the one between X_p and X_q, then X_p and X_q cannot be part of an inlier set. The exact quantification of "sufficiently large" deviations depends on the depth sensor characteristics. Note that this criterion is invariant to any hypothesized pose. It can be made stronger (more discriminative) by adding the compatibility of normal estimates as e.g. considered in [18]. In order not to introduce extra tuning parameters of how to weight the distance and normal compatibility terms, we focus on the distance based compatibility of predicted object coordinates in the following. We believe that the loss of discrimination power by excluding normal compatibility has minimal impact on the results, since the final compatibility scores are based on triplets of correspondences as described below. Thus, our scoring function to assess the compatibility between correspondences $X_p \leftrightarrow \hat{X}_p$ and $X_q \leftrightarrow \hat{X}_q$ (which will play the role of pairwise potentials in the following) is given by

$$\psi(X_p, X_q; \hat{X}_p, \hat{X}_q) \overset{\text{def}}{=} \tag{6.2}$$

$$\begin{cases} \Delta^2(X_p, X_q; \hat{X}_p, \hat{X}_q) & \text{if } |\Delta(X_p, X_q; \hat{X}_p, \hat{X}_q)| \le \sigma \\ \infty & \text{otherwise.} \end{cases} \tag{6.3}$$

with $\Delta(X_p, X_q; \hat{X}_p, \hat{X}_q) \overset{\text{def}}{=} \|\hat{X}_p - \hat{X}_q\| - \|X_p - X_q\|$. σ is the maximum noise or uncertainty level expected from the depth sensor and matching procedure. Since we densely sample the training data, the value of σ does not need to reflect the surface sampling density of training meshes. We set $\sigma = 3$mm in our experiments.

6.3.4 Minimal Sample Set Generation

Rigid pose estimation requires at least three (non-degenerate) point-to-point correspondences. Given three such correspondences, e.g. $\{\hat{X}_p \leftrightarrow X_p, \hat{X}_q \leftrightarrow X_q, \hat{X}_r \leftrightarrow X_r\}$, a Euclidean transformation and therefore pose estimate can be computed via the Kabsch algorithm or Horn's method [41]. The task at hand is to generate a promising set of three correspondences from the candidate object coordinates determined for each pixel.

Randomly sampling three putative correspondences will be inefficient, since the inlier ratio is very small as illustrated in the following example: if the object of interest is seen in about 5% of the image pixels, and 10 putative correspondences are maintained per pixel (and contain a true positive for each pixel covered by the object), the inlier ratio is 0.5%, and naive RANSAC sampling at a 95% confidence level will require more than 20 million iterations. This value is only a coarse estimate, since it is too pessimistic (e.g. by assuming a naive sampling over the full image instead of a more sophisticated sampling strategy) and too optimistic (by assuming pixels seeing the object have always a true positive correspondence) at the same time. Nevertheless, almost all random minimal sample sets will contain at least one outlier, and the pairwise compatibility criterion described in Section 6.3.3 will be crucial to efficiently determine promising sample sets.

To this end we propose to compute min-marginals via max-product belief propagation (BP) on a tree, which is actually min-sum BP since we operate on negative log-potentials, to quickly discard outlier contaminated sample sets. Let $\{p, q, r\}$ be a set of non-collinear pixels in the query image, let X_s, $s \in \{p, q, r\}$ range over the putative object coordinates, and $\phi_s(X_s)$ be a unary potential (usually based on the descriptor similary), then the negative log-likelihood energy of states (X_p, X_q, X_r) according to our graphical model is

$$\begin{aligned} E_{pqr}(X_p, X_q, X_r) &\overset{\text{def}}{=} \phi_p(X_p) + \phi_q(X_q) + \phi_r(X_r) \\ &+ \psi(X_p, X_q; \hat{X}_p, \hat{X}_q) + \psi(X_p, X_r; \hat{X}_p, \hat{X}_r). \end{aligned} \tag{6.4}$$

We use the Hamming distance between the descriptor extracted at pixel s and the ones returned by the approximate nearest neighbor search for X_s as unary potential $\phi_s(X_s)$.

Note that min-marginals, i.e. the quantities $\mu_{pqr}(X_p) \overset{\text{def}}{=} \min_{X_q, X_r} E_{pqr}(X_p, X_q, X_r)$ for each X_p can be computed via the bottom up pass of belief propagation on a tree rooted at p. In our case we only need 3 correspondences to determine a pose estimate, and therefore the tree degenerates to a chain. If the minimum sample size is larger—e.g. when computing the pose of an object subject to low-parametric and, approximately, isometric deformations—the obvious generalization of the underlying graph is a star graph.

The relevant values computed during BP are the upward messages

$$m_{q \to q}(X_p) = \min_{X_q} \left\{ \phi_q(X_q) + \psi(X_p, X_q; \hat{X}_p, \hat{X}_q) \right\} \tag{6.5}$$

sent from a leaf q to the root p. Note that the min-marginals can be expressed as

$$\begin{aligned} \mu_{pqr}(X_p) &= \min_{X_q, X_r} E_{pqr}(X_p, X_q, X_r) \\ &= \phi_p(X_p) + m_{q \to p}(X_p) + m_{r \to p}(X_p). \end{aligned} \tag{6.6}$$

Further, observe that the vector of messages $m_{q \to p} \overset{\text{def}}{=} (m_{q \to p}(X_p))_{X_p}$ can be reused in all trees containing the (directed) edge $q \to p$, leading to substantial computational savings. For certain pairwise potentials ψ the message vector computation is sub-quadratic in the number of states (i.e. putative object coordinates in our setting, see e.g. [23]), which would lead to further computational benefits. Unfortunately our choice of the pairwise potential given in Eq. 6.3 does not allow an obvious faster algorithm for message computation. Message computation does not only yield the value of the messages, $m_{q \to q}(X_p)$, but also the minimizing state

$$X^*_{q \to p}(X_p) \overset{\text{def}}{=} \arg\min_{X_q} \left\{ \phi_q(X_q) + \psi(X_p, X_q; \hat{X}_p, \hat{X}_q) \right\}, \tag{6.7}$$

which is used to quickly determine the optimal object coordinate predictions at pixels q and r given a prediction X_p at pixel p. Computation of the min-marginals $\mu_{pqr}(X_r)$ does not take into account the third edge potential between pixel q and r, $\psi(X_q, X_r; \hat{X}_q, \hat{X}_r)$. Adding this edge to the energy Eq. 6.4 would require message passing over triple cliques, which we considered to be computationally too costly at this point. Message passing would be cubic in the number of states in such setting.

We densely compute the min-marginals for each pixel in the query image, i.e. every pixel is the root, and compute messages $m_{p+\delta_k \to p}$ from pixel located at an offset δ_k, $k \in \{1, \ldots, K\}$, from p. Our choice of the set $\{\delta_k\}$ contains the 16 offsets of axis aligned and diagonal offsets at 8 and 16 pixels distance, which aims to trade off locality of predictions and numerical stability of pose estimation. For two edges $q \to p$ and $r \to p$ the predictions $(X_p, X^*_{q \to p}(X_p), X^*_{r \to p}(X_p))$ form a minimal sample set for estimating the rigid pose, and min-marginals are for all $K(K-1)/2$. Such triplets are used to rank these minimal sample sets. The method proceed with estimating and evaluating the pose for the top 2000 ranked as described in the next subsection.

6.3.5 Pose Hypotheses Evaluation

Assessing the quality of a pose hypothesis by aligning the 3D model with the range image appears to be straightforward—if the poses are affected by no or minimal noise. We do expect a substantial noise level in our pose hypotheses, and a sensible scoring function to rank the poses needs to take this into account. To this end a scoring function needs to be invariant to pose uncertainties. Since the true pose is effectively a latent variable, we can either marginalize (i.e. average) over nearby poses or maximize over the latent pose. We choose the latter option. It essentially amounts to smoothing the input, see [62] for an extensive discussion of building invariance with respect to geometric transformation. Since we do not expect or assume to obtain many pose hypotheses near the true pose, we refrain from using pose clustering or averaging approaches e.g. employed in [18, 82]. In contrast to works such as [93, 5], which refine a pose entirely based on correspondences between predicted object coordinates and observed depth geometry, we utilize a "classical" geometric approach by determining an optimal alignment between the given 3D model points and the depth map.

A proper way to assess the quality of a hypothesized pose (or any latent variable in general) is to "explain" the data given the assumptions on the sensor noise, i.e. to formulate a respective cost function that sums (integrates) over the image domain. Unfortunately, this more principled formulation is expensive to optimize. Thus, we employ—like most of the respective literature—the reverse direction of "explaining" the model for computational reasons (recall that up to 2000 pose hypotheses are considered at this stage). We implemented several methods to robustly refine the pose of a point set with respect to a depth map, including pose refinement via (robust) non-linear least squares. In our experience the following simple alternation algorithm proves to be efficient and effective:

1. Perform "projective data association" (i.e. establish the correspondence between a model point X_j and the back-projected depth $\hat{X}_j$ with both $\hat{X}_j$ and $RX_j + T$ being on the same line-of-sight),

2. and update R and T using a weighted extension of the Kabsch algorithm. The weights w_j are derived from the smooth approximation of the robust truncated quadratic kernel (see e.g. [113] for a discussion of this kernel)

$$\rho_\tau(e) \overset{\text{def}}{=} \begin{cases} \frac{e^2}{4}\left(2 - \frac{e^2}{\tau^2}\right) & \text{if } e^2 \le \tau^2 \\ \frac{\tau^2}{4} & \text{otherwise,} \end{cases} \tag{6.8}$$

$$\omega_\tau(e) \overset{\text{def}}{=} \rho'_\tau(e)/e = \max\{0, 1 - e^2/\tau^2\}, \tag{6.9}$$

and given by

$$w_j = \omega_\tau \left(\left(RX_j + T - \hat{X}_j \right) \right) . \tag{6.10}$$

The weights given in Eq. 6.10 are based on depth deviation between the transformed model point and the corresponding value in the depth map. If a deph value is missing for the projected model point, that correspondence is considered an outliers and has zero weight. τ is the inlier noise level and we use the same value as for σ which is 3mm, recall Sec. 6.3.3. Please observe that this algorithm does not optimize a single energy, a property shared with most ICP variants using projective data association. We iterate these two steps 10 times on a a random subset of 1000 model points. The final score of the pose hypothesis is evaluated on a larger subset of 10000 model points by using a robust fitting cost,

$$\sum_j \rho_\tau \left(\left(RX_j + T - \hat{X}_j \right) \right) . \tag{6.11}$$

The pose with the lowest cost is reported and visualized.

6.3.6 Implementation Notes

Computation time

The core data used in the training stage are depth images of the objects of interest together with the respective pose data. These depth maps can be generated synthetically from e.g. CAD models or captured by a depth sensor. If CAD models are rendered, the camera poses are generated randomly looking towards the object's center of gravity. At this point we do not aim to simulate the real depth sensor characteristic (e.g. noise or quantization effects), which in some cases led to missed correspondences in parts of the object (e.g. the top of the pipe in Fig. 6.1 has a substantially different appearance in rendered and real depth maps). From these depth maps we extract a target number of descriptors (typically 32k in our experiments) by selecting a random subset of valid pixels in the depth map. Random sampling is slightly biased towards pixels in the depth map with close to fronto-parallel surface patches. Thus, about 600k descriptors (32k × 18 for the sampled tangent-plane rotations) are generated and stored. No further processing takes part at training time. Consequently, the training phase is completed within seconds.

Parallel Implementation

Most steps in our approach can be trivially parallelized (including descriptor extraction, matching against the database, message passing, and pose evaluation). While we did not implement any part of the algorithm on the GPU, we made straightforward use of OpenMP-based multi-processing whenever possible. The input depth maps are 640×480 pixels, but we compute predicted object coordinates on either 320×240 or 160×120 images. Lhe latter one enables us to achieve interactive frame rates. On a dual Xeon E5-2690 system we achieve between 2 frames per second using a 320×240 resolution or up to 10 Hz using a 160×120 resolution. Nearest-neighbor descriptor matching is usually the most time consuming part (see also Fig. 6.4). We anticipate real-time performance of a GPU implementation.

6.4 Experiments

We show results on the Mian dataset [66], since it is the de facto baseline benchmark dataset for 3D object detection algorithms. We also show our own datasets recorded with the ASUS Xtion camera in order to demonstrate our algorithms ability to cope with noisy inputs. Since our 3D object detection algorithm

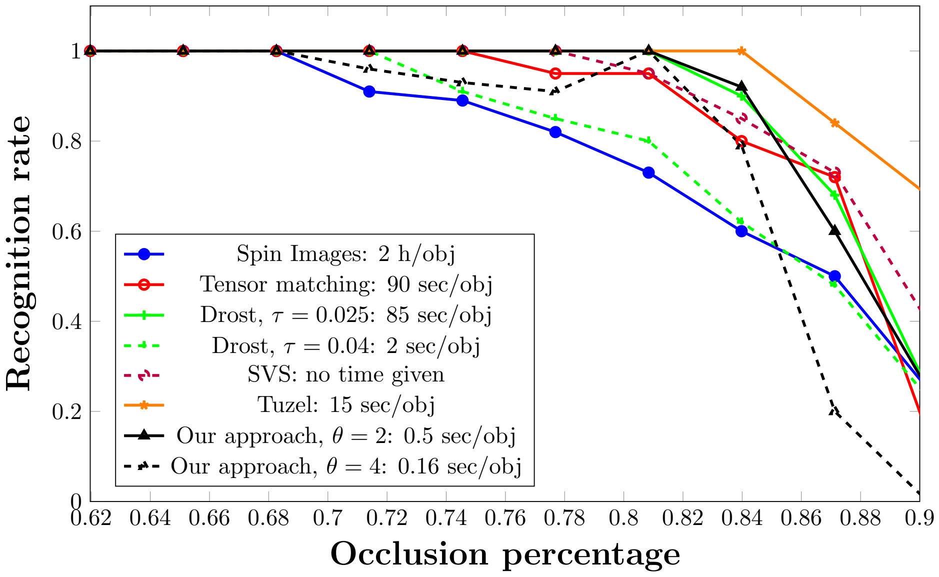


Figure 6.2: Results obtained on the Mian dataset [66]. It can be seen that our method is capable to handle occlusions of up to 81% and still give 100% of detection rates. It is also significant that the time required to detect a single object compared to the only other approaches that obtain similar or better detection rates [18, 106], is of up to 30 times less for our approach when compared with Tuzel and up to 170 times less compared to Drost.

takes depth maps as input, we converted the given meshes to range images by rendering into 640 × 480 depth maps using approximate parameters for the camera intrinsics (since the calibration parameters of the range scanner are not available). Consequently, the amount of occlusions in our depth maps may be slightly higher than in the provided meshes. We show as baseline methods the following approaches: Spin images [66], Tensor matching [66], Drost et al. [18], SVS [72] and Tuzel et al. [106].

6.4.1 Experimental Setup

The Mian dataset contains 50 scenes with 4 models on which to perform detection. Ground truth pose is provided for all instances of all objects. Apart from those 4 models another model exists that was excluded in Mian's experiments [66]; hence our approach and all baselines do not include this object. To validate a detection as valid we use the same thresholds as used in [18], we also define occlusion values in the same manner. We provide results for two different resolutions for the prediction image, 320 × 240 (downsampling factor $\theta = 2$), and 160 × 120 ($\theta = 4$). A smaller resolution of the predicted object coordinate image means faster computation, but also a lower probability of finding an inlier sample set, and consequently returning a successful detection.

6.4.2 Experimental Results

As seen in Fig. 6.2, we are able to perform to a 100% detection with up to 81% of occlusion, with higher levels of occlusion we perform similarly to the best baselines. The approach of Tuzel et al. [106] is the only approach capable of giving a significant improvement on levels of occlusion higher to 81%. Learning techniques could likely be employed to boost our results in terms of recognition rate and possibly in runtime.

The results on the Mian dataset give us a clear understanding of how our approach compares against previous work, but at the same time the data is much cleaner than depth maps obtained by current commodity sensors. Consequently, we recorded our own data using an ASUS Xtion depth sensor and ran our method for objects with available CAD models either obtained from a 3D model database, such as the toy car and the bracket, or by approximate manual 3D modeling of pipe-like structures. When creating the descriptors for the objects of interest we do not simulate any of the depth sensor characteristics such as boundary fattening and depth quantization. Thus, the 3D model to detect and the actual range images may be significantly different in their depth appearance. Fig. 6.3 depicts sample frames with the model point cloud superimposed on the input depth, rendered via its normal map. Our produced sequences differ in several aspects from the benchmark dataset [66]: the depth sensor characteristics at training time and test time do not match, the depth maps at test time are limited in quality, compared to a more expensive scanning setup, and objects themselves are less discriminative in shape.

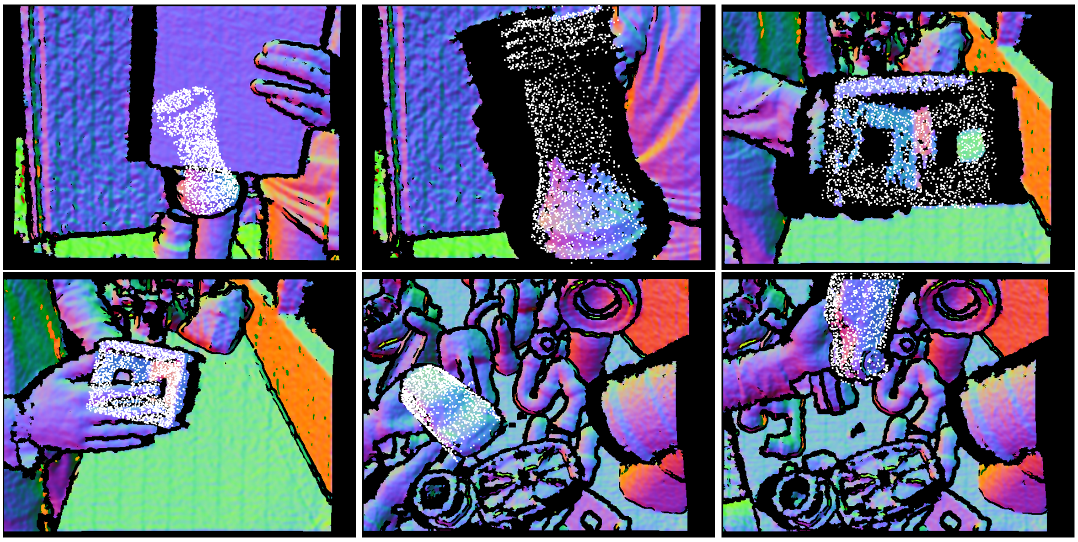


Figure 6.3: Sample frames from the ASUS Xtion sequences. The respective model point cloud is superimposed on the normal-map rendered input. Correct detections and poses can be seen despite large occlusions, missing depth data, and strong viewpoint changes.

6.4.3 Computation Time

We present results with a CPU implementation of the approach, although a GPU implementation for most steps in the algorithm is straightforward and is expected to yield real-time performance (20Hz). In Fig. 6.4 we break down the individual contributions of the various stages of our method: descriptor computation, Hamming distance based descriptor matching using FLANN, message passing for min-marginal computation, ranking/sorting according to Eq. 6.6, and final pose evaluation including ICP. The exact values vary depending on the input frame and the object of interest, but in general feature matching, i.e. nearest neighbor search, consumes a dominant fraction of the overall frame time. The matching time is typically faster for object with a highly distinctive local shape appearances than for object with redundant surface structures, since in the former case the search trees tend to be more balanced.

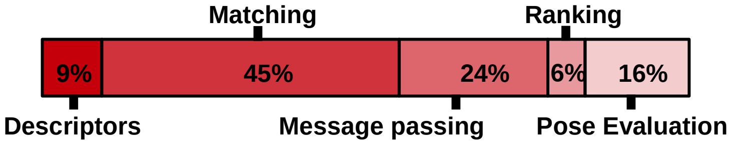


Figure 6.4: Percentage of the total time employed in each of the stages of the algorithm. We can see that by far the most expensive step is the feature matching step.

6.5 Conclusions

We have addressed the problem of 3D object detection and corresponding pose estimation, and we discussed a more efficient paradigm to solve this task while still obtaining state of the art detection rates. We believe that this work creates a new and robust framework from which to build new 3D object detection approaches. In the method described in this chapter we left out basically any learning-based technique to boost the detection performance or run-time behavior. While we argue that computationally expensive learning techniques will limit the general applicability of 3D object recognition (since adding new objects requires time-consuming retraining), we foresee that more sophisticated processing of training objects than our current one will lead to more discriminative descriptors, and therefore will be highly beneficial for this task.

Chapter 7

Concluding Remarks

The main objective of this thesis was to explore strategies to enhance state-of-the-art techniques to solve the pose estimation problem in computer vision. By analyzing a wide array of contexts and determining the characteristics that could be enhanced we have provided a new set of tools for this purpose. We have also shown throughout the various methods presented that purely geometric problems can benefit from learning techniques. In the end, we have presented four works that enhance in different ways 3D pose estimation. We will now summarize our main contributions:

1. We have presented a novel approach to solve the uncalibrated pose estimation problem. We derived new mathematical formulae to solve the perspective camera equations, by drawing inspiration from the EP*n*P [58] approach we have been able to give a novel and efficient solution to the uncalibrated context that can be applied to any number of points.

2. We have extended the BP*n*P [70] approach to the uncalibrated context. By increasing the dimensionality of the problem we made the algorithm more general but at the same time increased the complexity of the problem. To solve the inherent problems with uncalibrated pose estimation we proposed a new framework and a new distance that better handles matching between candidate points.

3. We developed a new paradigm for learning object representations from multiple views. By introducing geometric priors in the learning scheme we were able to use a single classifier for all viewpoints of an object. This avoids training multiple classifiers for each viewpoint, thus, reducing the number of training samples required. We have proven this new paradigm to be sufficiently robust to be able to solve pose estimation reliably even under the hardest contexts. This new paradigm opens numerous possibilities due to it not being dependent of any particular feature or learning algorithm.

4. We have given a new solution to object pose estimation using only depth. By extracting dense descriptors in the scene and performing an energy minimization to obtain the best candidates we can robustly calculate a 3D pose of the object in the scene that is as accurate as the state of the art while being between 10 and a 100 times faster.

The contexts on which we have worked are in some concern independent, although the task at hand is common. In any case, there are common characteristics to parts of this work:

- Both our solutions to the P*n*P and BP*n*P problems are applied to an uncalibrated context. We think that by offering better algorithms that do not depend on a constant calibration we give more flexibility to such algorithms and at the same time the possibility of performing zoom changes on-the-fly which is very useful for real case scenarios.

- Our solutions to the P*n*P and BP*n*P also share the fact that they solely rely on geometry to solve the pose estimation. Depending on texture or other cues would have made these solutions less practical.

- All our algorithms have a strong focus on offering fast and efficient solutions while at the same time matching or even improving the precision of the state-of-the-art. Now that Moore's law is starting to not hold, efficient solutions are more of an essence to the field and careful consideration has to be taken, specially when using large servers is not an option as it happens to be the case with hand-held devices.

- All our body of work assumes a 3D prior of the object or the scene. This was something critical years ago, but now with the improvement of structure from motion algorithms and the appearance of cheap consumer 3D sensors it is no longer a problem and, by taking advantage of these improvements, we can enhance most computer vision tasks.

- The work we have developed in this thesis is, to a great degree, independent of how we describe a scene. By focusing on the geometry rather than on how to describe the scene we have provided solutions that can be used in a wider array of situations and that can be used in most contexts.

Chapter 8

Publications

8.1 Publications

The following is a list of accepted publications resulting from the work performed during the elaboration of the PhD thesis . Publications are shown in chronological order.

J1. **A. PENATE-SANCHEZ**, J. ANDRADE-CETTO, F. MORENO-NOGUER. Exhaustive Linearization for Robust Camera Pose and Focal Length Estimation. In *2013 Pattern Analysis and Machine Intelligence, IEEE Transactions on*. Impact factor **5.694**.

C1. **A. PENATE-SANCHEZ**, J. ANDRADE-CETTO, F. MORENO-NOGUER. Simultaneous Pose, Focal Length and 2D-to-3D Correspondences from Noisy Observations. In *2013 British Machine Vision Conference*. Selected for Poster presentation (30% acceptance rate).

C2. A. RUBIO, M. VILLAMIZAR, L. FERRAZ, **A. PENATE-SANCHEZ**, A. SANFELIU, F. MORENO-NOGUER. Estimacion monocular y eficiente de la pose usando modelos 3D complejos. In *2014 Jornadas de Automatica*, **Best Vision Paper Award**.

C3. **A. PENATE-SANCHEZ**, F. MORENO-NOGUER, J. ANDRADE-CETTO, F. FLEURET. LETHA: Learning from High Quality Inputs for 3D Pose Estimation in Low Quality Images. In *2014 International Conference on 3D Vision*, Selected for **Oral** presentation (13.5% acceptance rate for Oral presentations).

C4. A. RUBIO, M. VILLAMIZAR, L. FERRAZ, **A. PENATE-SANCHEZ**, A. RAMISA, E. SIMO-SERRA, A. SANFELIU, F. MORENO-NOGUER. Efficient Monocular Pose Estimation for Complex 3D Models. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, (41% acceptance rate).

C5. C. ZACH, **A. PENATE-SANCHEZ**, M.T. PHAM. A Dynamic Programming Approach for Fast and Robust Object Pose Recognition from Range Images. In *2015 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. Selected for Poster presentation (**21.9%** acceptance rate).

Chapter 9

Appendix: AbstainBoost Derivation

AbstainBoost is an adaptation of AdaBoost to the case where weak learners may respond 0 and not only ± 1.

The standard way to derive Adaboost is as a gradient descent: Pick the weak learner h corresponding to the direction of maximum reduction of the loss L

$$\arg\max_h \left| \left. \frac{\partial L(f_t + \lambda h)}{\partial \lambda} \right|_{\lambda=0} \right| \tag{9.1}$$

and go "in that direction" to minimize the loss optimally

$$\lambda^* = \arg\min_\lambda \; L(f_t + \lambda h). \tag{9.2}$$

If the weak learners respond ± 1 they are all of same norm, and Equation9.1 makes sense. However, it looks obvious that for instance, $2h$ should not be preferred to h simply because it looks twice better in Equation9.1.

The first strategy would be to normalize the score of Equation (9.1) with the norm of the weak learner, hence looking for

$$\arg\max_h \left| \left. \frac{1}{||h||_2} \, \frac{\partial L(f_t + \lambda h)}{\partial \lambda} \right|_{\lambda=0} \right|. \tag{9.3}$$

However, a weak learner which responds always 0 but on a few samples where it does a perfect job would have a very good score. This is obviously a serious weakness.

Let $\{(x_n, y_n)\} \in \mathcal{X} \times \{-1, 1\}$ be a training set. We then choose h by minimizing directly

$$\begin{aligned}
C(h) &= \min_\lambda L(f + \lambda h) && (9.4)\\
&= \min_\lambda \sum_n \exp(-y_n(f(x_n) + \lambda h(x_n))) && (9.5)\\
&= \min_\lambda \sum_{n:y_n h(x_n)=0} \exp(-y_n(f(x_n) + \lambda h(x_n))) && (9.6)\\
&\quad + \sum_{n:y_n h(x_n)=1} \exp(-y_n(f(x_n) + \lambda h(x_n))) && (9.7)\\
&\quad + \sum_{n:y_n h(x_n)=-1} \exp(-y_n(f(x_n) + \lambda h(x_n))) && (9.8)\\
&= \min_\lambda \sum_{n:y_n h(x_n)=0} \exp(-y_n f(x_n)) && (9.9)\\
&\quad + \exp(\lambda) \sum_{n:y_n h(x_n)=-1} \exp(-y_n f(x_n)) && (9.10)\\
&\quad + \exp(-\lambda) \sum_{n:y_n h(x_n)=1} \exp(-y_n(f(x_n))) && (9.11)\\
&= \min_\lambda \; W_0 + W_- \exp(\lambda) + W_+ \exp(-\lambda) && (9.12)
\end{aligned}$$

where

$$\begin{aligned}
W_0 &= \sum_{n:y_n h(x_n)=0} \exp(-y_n f(x_n)) && (9.13)\\
W_+ &= \sum_{n:y_n h(x_n)=1} \exp(-y_n f(x_n)) && (9.14)\\
W_- &= \sum_{n:y_n h(x_n)=-1} \exp(-y_n(f(x_n))) && (9.15)
\end{aligned}$$

Since

$$\arg\min_\lambda L(f + \lambda h) = \log \sqrt{W_+/W_-}, \qquad (9.16)$$

we have

$$\begin{aligned}
\min_\lambda f(\lambda) &= W_0 + W_- \exp(\log \sqrt{W_+/W_-}) + W_+ \exp(-\log \sqrt{W_+/W_-}) && (9.17)\\
&= W_0 + W_- \sqrt{W_+/W_-} + W_+ \sqrt{W_-/W_+} && (9.18)\\
&= W_0 + 2\sqrt{W_+ W_-} && (9.19)
\end{aligned}$$

which leads to

$$\begin{aligned}
C(h) &= \min_\lambda L(f + \lambda h) = W_0 + 2\sqrt{W_+ W_-} && (9.20)\\
&= Z - W_+ - W_- + 2\sqrt{W_+ W_-} = Z - \left(\sqrt{W_+} - \sqrt{W_-}\right)^2 && (9.21)
\end{aligned}$$

With $Z = \sum_n \exp(-y_n f(x_n))$.

It can be seen that while Adaboost maximizes $|W_+ - W_-|$, Abstainboost maximizes $|\sqrt{W_+} - \sqrt{W_-}|$.

Bibliography

[1] A. Ansar and K. Daniilidis. Linear pose estimation from points or lines. *Transactions on Pattern Analysis and Machine Intelligence*, 25(5):578–589, 2003.

[2] P. Bariya and K. Nishino. Scale-hierarchical 3d object recognition in cluttered scenes. In *Conference on Computer Vision and Pattern Recognition*, pages 1657–1664, 2010.

[3] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *European Conference on Computer Vision*, 2006. 404–417.

[4] P. Besl and N. McKay. A method for registration of 3-D shapes. *Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[5] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother. Learning 6d object pose estimation using 3d object coordinates. In *European Conference on Computer Vision*, volume 8690, pages 536–551, 2014.

[6] G. Bradski. Opencv library. *Dr. Dobb's Journal of Software Tools*, 2000.

[7] M. Bujnak, Z. Kukelova, and T. Pajdla. A general solution to the P4P problem for camera with unknown focal length. In *Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[8] M. Bujnak, Z. Kukelova, and T. Pajdla. Robust focal length estimation by voting in multi-view scene reconstruction. In *Asian Conference on Computer Vision. Vol. 5994 of Lecture Notes in Computer Science*, 2009. 13–24.

[9] M. Bujnak, Z. Kukelova, and T. Pajdla. New efficient solution to the absolute pose problem for camera with unknown focal length and radial distortion. In *Asian Conference on Computer Vision*, 2010.

[10] M. Byrod, Z. Kukelova, K. Josephson, T. Pajdla, and K. Astrom. Fast and robust numerical solutions to minimal problems for cameras with radial distortion. In *Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[11] M. Calonder, V. Lepetit, M. Özuysal, T. Trzcinski, C. Strecha, and P. Fua. Brief: Computing a local binary descriptor very fast. *Transactions on Pattern Analysis and Machine Intelligence*, 34:1281–1298, 2012.

[12] T.J. Chin, J. Yu, and D. Suter. Accelerated Hypothesis Generation for Multi-Structure Robust Fitting. In *European Conference on Computer Vision*, 2010.

[13] K. Choi, S. Lee, and Y. Seo. A branch-and-bound algorithm for globally optimal camera pose and focal length. *Image and Vision Computing*, 28(9):1369–1376, 2010.

[14] O. Chum and J. Matas. Matching with PROSAC - progressive sample consensus. In *Conference on Computer Vision and Pattern Recognition*, pages 220–226, 2005.

[15] W.W. Cohen and Y. Singer. A simple, fast, and effective rule learner. In *American Association for Artificial Intelligence Conference*, 1999.

[16] P. David, D. DeMenthon, R. Duraiswami, and H. Samet. SoftPOSIT: Simultaneous Pose and Correspondence Determination. *International Journal of Computer Vision*, 59(3):259–284, 2004.

[17] D. DeMenthon and L.S. Davis. Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15(1-2):123–141, 1995.

[18] B. Drost, M. Ulrich, N. Navab, and S. Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *Conference on Computer Vision and Pattern Recognition*, pages 998–1005, 2010.

[19] O. Enqvist, K. Josephson, and F. Kahl. Optimal Correspondences from Pairwise Constraints. In *International Conference on Computer Vision*, 2009.

[20] G. Fanelli, J. Gall, and L. van Gool. Real time head pose estimation with random regression forests. In *Conference on Computer Vision and Pattern Recognition*, 2011. 617–624.

[21] R. Farrell, O. Oza, N. Zhang, V. I. Morariu, T. Darrell, and L.S. Davis. Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance. In *International Conference on Computer Vision*, pages 161–168, 2011.

[22] P. F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *Transactions on Pattern Analysis and Machine Intelligence*, pages 1627–1645, 2010.

[23] P. F Felzenszwalb and D.P. Huttenlocher. Efficient belief propagation for early vision. *International Journal of Computer Vision*, 70(1):41–54, 2006.

[24] V. Ferrari, T. Tuytelaars, and L. Van Gool. Object detection by contour segment networks. In *European Conference on Computer Vision*, volume 3953 of *LNCS*, pages 14–28. Elsevier, June 2006.

[25] P.D. Fiore. Efficient linear solution of exterior orientation. *Transactions on Pattern Analysis and Machine Intelligence*, 23(2):140–148, 2001.

[26] M.A Fischler and R.C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications ACM*, 24(6):381–395, 1981.

[27] F. Fleuret. Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research*, 5:1531–1555, 2004.

[28] F. Fleuret and D. Geman. Stationary features and cat detection. *Journal of Machine Learning Research*, 9:2549–2578, 2008.

[29] S. Foix, G. Alenya, J. Andrade-Cetto, and C. Torras. Object modeling using a tof camera under an uncertainty reduction approach. In *International Conference on Robotics and Automation*, pages 1306–1312, 2010.

[30] Y. Freund and R.E. Schapire. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5):771–780, 1999.

[31] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng. Complete solution classification for the perspective-three-point problem. *Transactions on Pattern Analysis and Machine Intelligence*, 25(8):930–943, 2003.

[32] D. Glasner, M. Galun, S. Alpert, R. Basri, and G. Shakhnarovich. Viewpoint-aware object detection and pose estimation. In *International Conference on Computer Vision*, 2011. 1275–1282.

[33] D. Glasner, M. Galun, S. Alpert, R. Basri, and G. Shakhnarovich. Viewpoint-aware object detection and continuous pose estimation. *Journal of Image Vision Computing*, 2013. In press, available online.

[34] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2916–2929, 2013.

[35] R. Hartley and F. Schaffalitzky. L_∞ Minimization in Geometric Reconstruction Problems. In *Conference on Computer Vision and Pattern Recognition*, 2004.

[36] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition, 2004.

[37] S. Hinterstoisser, C. Cagniart, S. Ilic, P. Sturm, N. Navab, P. Fua, and V. Lepetit. Gradient response maps for real-time detection of texture-less objects. *PAMI*, 2012.

[38] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *International Conference on Computer Vision*, 2011.

[39] R. Horaud, F. Dornaika, B.t Lamiroy, and S. Christy. Object pose: The link between weak perspective, paraperspective, and full perspective. *International Journal of Computer Vision*, 22(2):173–189, 1997.

[40] B. K. P Horn, H. M. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America*, 5(7):1127–1135, 1988.

[41] B.K.P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal Of The Optical Society America*, 4(4):629–642, 1987.

[42] B.K.P. Horn, H.M. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *Journal Of The Optical Society America*, 5(7):1127–1135, 1988.

[43] E. Hsiao and M. Hebert. Occlusion reasoning for object detection under arbitrary viewpoint. In *Conference on Computer Vision and Pattern Recognition*, pages 3146–3153, 2012.

[44] W. Hu and S.-C. Zhu. Learning a probabilistic model mixing 3D and 2D primitives for view invariant object recognition. In *Conference on Computer Vision and Pattern Recognition*, 2010. 2273–2280.

[45] C. Huang, H. Ai, Y. Li, and S. Lao. Vector boosting for rotation invariant multi-view face detection. In *International Conference on Computer Vision*, pages 446–453, 2005.

[46] K. Ikeuchi. Generating an interpretation tree from a cad model for 3d-object recognition in bin-picking tasks. *International Journal of Computer Vision*, 1(2):145–165, 1987.

[47] A.E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433–449, 1999.

[48] K. Josephson and M. Byrod. Pose estimation with radial distortion and unknown focal length. In *Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2419–2426, 2009.

[49] F. Kahl, S. Agarwal, M. Chandraker, D. Kriegman, and S. Belongie. Practical global optimization for multiview geometry. *International Journal of Computer Vision*, 79(3):271–284, 2008.

[50] F. Kahl and R. Hartley. Multiple-view geometry under the L_∞-norm. *Transactions on Pattern Analysis and Machine Intelligence*, 30(9):1603–1617, 2008.

[51] A. Kipnis and A. Shamir. Cryptanalysis of the HFE public key cryptosystem by relinearization. In *CRYPTO*, 1999. 19–30.

[52] J. Knopp, M. Prasad, G. Willems, R. Timofte, and L. Van Gool. Hough transform and 3D SURF for robust three dimensional classification. In *European Conference on Computer Vision*, pages 589–602, 2010.

[53] Z. Kukelova, M. Bujnak, and T. Pajdla. Polynomial eigenvalue solutions to the 5-pt and 6-pt relative pose problems. In *British Machine Vision Conference*, 2008.

[54] S. Kwak, W. Nam, B. Han, and J.H. Han. Learning occlusion with likelihoods for visual tracking. In Dimitris N. Metaxas, Long Quan, Alberto Sanfeliu, and Luc J. Van Gool, editors, *International Conference on Computer Vision*, pages 1551–1558. IEEE, 2011.

[55] J.F. Lalonde, D. Hoiem, A. A Efros, C. Rother, J. Winn, and A. Criminisi. Photo Clip Art. *ACM SIGGRAPH*, 26(3):3, 2007.

[56] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision*, 77(1-3):259–289, 2008.

[57] V. Lepetit and P. Fua. Keypoint recognition using randomized trees. *Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1465–1479, 2006.

[58] V. Lepetit, F. Moreno-Noguer, and P. Fua. EPnP: An accurate $O(n)$ solution to the PnP problem. *International Journal of Computer Vision*, 81(2):151–166, 2008.

[59] J. Liebelt and C. Schmid. Multi-view object class detection with a 3d geometric model. In *Conference on Computer Vision and Pattern Recognition*, 2010. 1688–1695.

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

[61] C.P. Lu, G.D. Hager, and E. Mjolsness. Fast and globally convergent pose estimation from video images. *Transactions on Pattern Analysis and Machine Intelligence*, 22(6):610–622, 2000.

[62] S. Mallat. Group invariant scattering. *Communications on Pure and Applied Mathematics*, 65(10):1331–1398, 2012.

[63] L. Mason, J. Baxter, P. Bartlett, and M. Frean. Boosting algorithms as gradient descent. In *Neural Information Processing Systems*, 2000.

[64] MATLAB. *version 7.10.0 (R2010a)*. The MathWorks Inc., Natick, Massachusetts, 2010.

[65] L. Mei, J. Liu, A. O. Hero III, and S. Savarese. Robust object pose estimation via statistical manifold modeling. In *International Conference on Computer Vision*, pages 967–974, 2011.

[66] A.S. Mian, M. Bennamoun, and R. Owens. Three-dimensional model-based object recognition and segmentation in cluttered scenes. *Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1584–1601, 2006.

[67] K.n Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *Transactions on Pattern Analysis and Machine Intelligence*, 10(27):1615–1630, 2005.

[68] P. Moreels and P. Perona. Evaluation of features detectors and descriptors based on 3D objects. *International Journal of Computer Vision*, 73(3):263–284, 2006.

[69] F. Moreno-Noguer, V. Lepetit, and P. Fua. Accurate non-iterative o(n) solution to the pnp problem. In *International Conference on Computer Vision*, 2007.

[70] F. Moreno-Noguer, V. Lepetit, and P. Fua. Pose Priors for Simultaneously Solving Alignment and Correspondence. In *European Conference on Computer Vision*, volume 2, pages 405–418, 2008.

[71] M. Muja and D.G. Lowe. Fast matching of binary features. In *CRV*, pages 404–410, 2012.

[72] H.V. Nguyen and F. Porikli. Support vector shape: A classifier-based shape representation. *Transactions on Pattern Analysis and Machine Intelligence*, 35(4):970–982, 2013.

[73] K. Ni, H. Jin, and F. Dellaert. GroupSAC: Efficient Consensus in the Presence of Groupings. In *International Conference on Computer Vision*, pages 2193–2200, 2009.

[74] D. Nistér. Preemptive RANSAC for Live Structure and Motion Estimation. In *International Conference on Computer Vision*, pages 199–206, 2003.

[75] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

[76] M. Ozuysal, V. Lepetit, and P. Fua. Pose estimation for category specific multiview object localization. In *Conference on Computer Vision and Pattern Recognition*, 2009. 778–785.

[77] N. Payet and S. Todorovic. From contours to 3D object detection and pose estimation. In *International Conference on Computer Vision*, 2011. 983–990.

[78] A. Penate-Sanchez, J. Andrade-Cetto, and F. Moreno-Noguer. Exhaustive linearization for robust camera pose and focal length estimation. *Transactions on Pattern Analysis and Machine Intelligence*, 99, 2013.

[79] A. Penate-Sanchez, F. Moreno-Noguer, J. Andrade-Cetto, and F. Fleuret. Letha: Learning from high quality inputs for 3d pose estimation in low quality images. In *3D Vision Conference*, 2014.

[80] A. Penate-Sanchez, E. Serradell, J. Andrade-Cetto, and F. Moreno-Noguer. Simultaneous pose, focal length and 2d-to-3d correspondences from noisy observations. In *British Machine Vision Conference*, 2013.

[81] B. Pepik, M. Stark, P. V. Gehler, and B. Schiele. Teaching 3d geometry to deformable part models. In *Conference on Computer Vision and Pattern Recognition*, pages 3362–3369, 2012.

[82] M.T. Pham, O.J. Woodford, F. Perbet, A. Maki, B. Stenger, and R. Cipolla. A new distance for scale-invariant 3d shape recognition and registration. In *International Conference on Computer Vision*, pages 145–152, 2011.

[83] R. Raguram, J.M. Frahm, and M. Pollefeys. A Comparative Analysis of RANSAC Techniques Leading to Adaptive Real-Time Random Sample Consensus. In *European Conference on Computer Vision*, pages 500–513, 2008.

[84] N. Razavi, J. Gall, and L. Van Gool. Backprojection revisited: Scalable multi-view object detection and similarity metrics for detections. In *European Conference on Computer Vision*, pages 620–633, Berlin, Heidelberg, 2010. Springer-Verlag.

[85] R. Rios-Cabrera and T. Tuytelaars. Discriminatively trained templates for 3d object detection: A real time scalable approach. In *International Conference on Computer Vision*, December 2013.

[86] E. Rodolà, A. Albarelli, F. Bergamasco, and A. Torsello. A scale independent selection process for 3d object recognition in cluttered scenes. *International Journal of Computer Vision*, 102(1-3):129–145, 2013.

[87] P. Rousseeuw and A. Leroy. *Robust Regression and Outlier Detection*. Wiley, 1987.

[88] R.B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (FPFH) for 3D registration. In *International Conference on Robotics and Automation*, pages 1848–1853, 2009.

[89] M. Salzmann, V. Lepetit, and P. Fua. Deformable surface tracking ambiguities. In *Conference on Computer Vision and Pattern Recognition*, 2007.

[90] G. Schweighofer and A. Pinz. Robust pose estimation from a planar target. *Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2024–2030, 2006.

[91] E. Serradell, M. Özuysal, V. Lepetit, P. Fua, and F. Moreno-Noguer. Combining Geometric and Appearance Priors for Robust Homography Estimation. In *European Conference on Computer Vision*, pages 58–72, September 2010.

[92] J. Shotton, A.Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *Conference on Computer Vision and Pattern Recognition*, pages 1297–1304, 2011.

[93] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon. Scene coordinate regression forests for camera relocalization in RGB-D images. In *Conference on Computer Vision and Pattern Recognition*, pages 2930–2937, 2013.

[94] K. Sim and R. Hartley. Removing Outliers Using the L_∞ Norm. In *Conference on Computer Vision and Pattern Recognition*, pages 485–494, 2006.

[95] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring image collections in 3D. *Transactions on Graphics*, 25:835–846, 2006.

[96] H. Stewenius, D. Nister, F. Kahl, and F. Schaffalitzky. A minimal solution for relative pose with unknown focal length. In *Conference on Computer Vision and Pattern Recognition*, pages 789–794, 2005.

[97] H. Su, M. Sun, L. Fei-Fei, and S. Savarese. Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories. In *International Conference on Computer Vision*, 2009. 213–220.

[98] A. Tejani, D. Tang, R. Kouskouridas, and T.K. Kim. Latent-class hough forests for 3d object detection and pose estimation. In *European Conference on Computer Vision*, volume 8694, pages 462–477, 2014.

[99] T. Thang-Pham, T.J. Chin, J. Yu, and D. Sutter. The random cluster model for robust geometric fitting. In *Conference on Computer Vision and Pattern Recognition*, pages 710–717, 2012.

[100] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiel, and L. Van Gool. Towards multi-view object class detection. In *Conference on Computer Vision and Pattern Recognition*, 2006. 1589–1596.

[101] B. Tordoff and D. W. Murray. Guided-MLESAC: Faster Image Transform Estimation by Using Matching Priors. *Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1523–1535, 2005.

[102] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: a large dataset for non-parametric object and scene recognition. *Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970, 2008.

[103] A. Toshev, B. Taskar, and K. Daniilidis. Shape-based object detection via boundary structure segmentation. *International Journal of Computer Vision*, 99(2):123–146, 2012.

[104] B. Triggs. Camera pose and calibration from 4 or 5 known 3d points. In *International Conference on Computer Vision*, pages 278–284, 1999.

[105] T. Trzcinski and V. Lepetit. Efficient Discriminative Projections for Compact Binary Descriptors. In *European Conference on Computer Vision*, 2012.

[106] Oncel Tuzel, Ming-Yu Liu, Yuichi Taguchi, and Arvind Raghunathan. Learning to rank 3d features. In *European Conference on Computer Vision*, pages 520–535, 2014.

[107] A. Vedaldi and B. Fulkerson. Vlfeat – an open and portable library of computer vision algorithms. In *ACM Multimedia*, 2010. 1469–1472.

[108] M. Villamizar, H. Grabner, F. Moreno-Noguer, J. Andrade-Cetto, L. Van Gool, and A. Sanfeliu. Efficient 3D object detection using multiple pose-specific classifiers. In *British Machine Vision Conference*, pages 20.1–20.10, 2011.

[109] B. Williams, G. Klein, and I. Reid. Real-time SLAM relocalisation. In *International Conference on Computer Vision*, pages 1–8, 2007.

[110] O. Woodford, M.T. Pham, A. Maki, F. Perbet, and B. Stenger. Demisting the hough transform for 3d shape recognition and registration. In *Proc. British Machine Vision Conference*, pages 32.1–32.11, 2011.

[111] C. Zach, A. Penate-Sanchez, and M.T. Pham. A dynamic programming approach for fast and robust object pose recognition from range images. June 2015.

[112] Z. Zhang. A flexible new technique for camera calibration. *Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 1998.

[113] M. Zollhöfer, M. Nießner, S. Izadi, C. Rehmann, C. Zach, M. Fisher, C. Wu, A. Fitzgibbon, C. Loop, and C. Theobalt. Real-time non-rigid reconstruction using an RGB-D Camera. *ACM TOG*, 2014.

List of Figures

List of Tables