Cecilia Maria Buarque Fredrich

A Parallel Method For Object Tracking

An algorithm adaptable to different conditions of object, movement and camera

MSc Thesis

DEPARTMENT OF ELECTRICAL ENGINEERING

Post–graduate program in Signal Processing and Control

Rio de Janeiro
June 2009

Pontifícia Universidade Católica
do Rio de Janeiro

Cecilia Maria Buarque Fredrich

A Parallel Method For Object Tracking

An algorithm adaptable to different conditions of object, movement and camera

MSc Thesis

Thesis presented to the Post–graduate Program in Signal Processing and Control of the Electrical Engineering Department at PUC–Rio in partial fulfillment of the requirements for the degree of Master of Science in Signal Processing and Control

Adviser : Prof. Raul Queiroz Feitosa
Co–Adviser: Prof. Marco Antonio Meggiolaro

Rio de Janeiro
June 2009

Pontifícia Universidade Católica
do Rio de Janeiro

Cecilia Maria Buarque Fredrich

A Parallel Method For Object Tracking

An algorithm adaptable to different conditions of object, movement and camera

Thesis presented to the Post–graduate Program in Signal Processing and Control of the Electrical Engineering Department at PUC–Rio in partial fulfillment of the requirements for the degree of Master of Science in Signal Processing and Control. Approved by the following committee:

Dr. Raul Queiroz Feitosa
Adviser
Department of Electrical Engineering — PUC–Rio

Dr. Marco Antonio Meggiolaro
Co–Adviser
Department of Mechanical Engineering — PUC–Rio

Dr. Alberto Barbosa Raposo
Computer Graphics Technology Group — TECGRAF/PUC-Rio

Dr. Luciano Vieira Dutra
Brazilian National Institute for Space Research — INPE

Prof. José Eugênio Leal
Head of the Science and Engineering Center — PUC–Rio

Rio de Janeiro — June 18, 2009

Cecilia Maria Buarque Fredrich

Cecilia Maria Buarque Fredrich earned bachelor's degrees in both Mechanical and Mechatronics Engineerings from the Pontifícia Universidade Católica do Rio de Janeiro (PUC–Rio) in 2006. She entered the postgraduate research course offered by the Department of Electrical Engineering at PUC–Rio in 2007.

Bibliographic data

Fredrich, Cecilia Maria Buarque

A Parallel Method For Object Tracking / Cecilia Maria Buarque Fredrich; adviser: Raul Queiroz Feitosa; co–adviser: Marco Antonio Meggiolaro. — Rio de Janeiro : PUC–Rio, Department of Electrical Engineering, 2009.

v., 93 f: il. ; 29,7 cm

1. MSc Thesis - Pontifícia Universidade Católica do Rio de Janeiro, Department of Electrical Engineering.

Bibliography included.

1. Electrical Engineering – Dissertation. 2. Point Matching. 3. Image Matching. 4. Object Tracking. 5. Object Recognition. 6. Normalized Cross Correlation. 7. Least Squares Matching. 8. SIFT. 9. Pose Detection. 10. Camera Calibration. 11. Parallel Processing. I. Feitosa, Raul Queiroz. II. Meggiolaro, Marco Antonio. III. Pontifícia Universidade Católica do Rio de Janeiro. Department of Electrical Engineering. IV. Title.

CDD: 621.3

Acknowledgments

To my advisors, Professors Raul Queiroz Feitosa and Marco Antonio Meggiolaro, for their insight, guidance and support in the realisation of this work and the everyday kindness and incentive throughout my Masters studies.

To Professor Mauro Speranza Neto, the project supervisor, who has been advising and supporting me since my undergraduate days.

To the CNPq and the PUC–Rio, without whose financial support this project would have never been realised.

To my colleagues at the Laboratory of Computer Vision, for their valuable comments, and those at the PUC–Rio, who made me love this place.

To the people at the Laboratory of Robotics for their constant help.

To the people at TECGRAF, who made useful test data and equipment available.

To the willing staff of the Electrical Engineering Department.

To my mother and father and all my family.

Abstract

Fredrich, Cecilia Maria Buarque; Feitosa, Raul Queiroz; Meggiolaro, Marco Antonio. **A Parallel Method For Object Tracking**. Rio de Janeiro, 2009. 93p. MSc Thesis — Department of Electrical Engineering, Pontifícia Universidade Católica do Rio de Janeiro.

Computer Vision-based techniques are powerful tools for designing efficient control systems. Cameras provide information, through which an industrial manipulator can, for instance, have its trajectory corrected with respect to the target object. Computer imagery might even be used to fully implement a controlling device for such manipulators.

This thesis proposes a real-time object tracking method robust and adaptable to different environment circumstances and scenarios, as well as to the quality of the camera available and the interest object's nature itself.

Image points are the features chosen for the object recognition procedure. The method relies on parallel processing for building the model that is best suited to the current scenario, thus dismissing heuristics for selecting the most adequate features. Although, the parallelism also contributes to the computational speed that allows real-time operation, the trade-off between algorithm complexity and quality of match is most responsible for achieving this particular goal.

Two variants were devised to cope with a number of different scenarios, as well as equipments, which accounts for the likeness of the method to succeed in a great deal of applications.

Keywords

Point Matching. Image Matching. Object Tracking. Object Recognition. Normalized Cross Correlation. Least Squares Matching. SIFT. Pose Detection. Camera Calibration. Parallel Processing.

Resumo

Fredrich, Cecilia Maria Buarque; Feitosa, Raul Queiroz; Meggiolaro, Marco Antonio. **Um Método Paralelo para Rastreamento de Objetos**. Rio de Janeiro, 2009. 93p. Dissertação de Mestrado — Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Técnicas em Visão Computacional são excelentes ferramentas para implementação de sistemas de controle eficientes. Usando câmeras, é possível corrigir a trajetória de um manipulador para que este chegue ao seu destino com maior exatidão. Imagens adquiridas através de câmeras podem, inclusive, ser dados de entrada para sistemas de controle auto-suficientes.
Esta dissertação apresenta um método paralelo para rastreamento de objetos, que opera em tempo real e é adaptável a diferentes condições de ambiente, equipamento e, também, à própria natureza do objeto em questão.
O rastreamento inicia-se com a coleta de pontos relevantes das imagens de entrada. Processamento paralelo garante que os pontos contidos no modelo do objeto sejam consistentes com aqueles coletados em determinada cena, sem necessidade de heurísticas. Apesar de o paralelismo contribuir para a velocidade de execução, esta é primordialmente conseqüência do compromisso estabelecido entre complexidade e qualidade dos algoritmos envolvidos no processo.
Duas variantes foram propostas para lidar com uma vasta gama de cenários e, também, equipamentos, tornando o método ser eficaz em diferentes aplicações.

Palavras–chave

Correspondência entre Pontos. Correspondência entre Imagens. Rastreamento de Objetos. Reconhecimento de Objetos. Correlação Cruzada Normalizada. Correspondência por Mínimos Quadrados. SIFT. Detecção de Pose. Calibração de Câmeras. Processamento Paralelo.

Contents

List of Figures

List of Tables

To see a World in a grain of sand,
And a Heaven in a wild flower,
Hold Infinity in the palm of your hand,
And Eternity in an hour.

William Blake, *Auguries of Innocence.*

1
Introduction

The majority of today's industrial robots are programmed to follow a predefined trajectory. Such controlling technique suffices when the robot is working in a fixed environment where all objects of interest are situated in a predetermined position relative to the robot's base. If, however, the robot's target position needs altering, all the trajectories have to be reprogrammed for the robot to be able to perform its tasks.

Another option is teleoperation, whereby a human operator conducts all the movements during the operation in a master-slave architecture. Since any positioning errors can be visually compensated, this configuration does not demand that the robot has a high absolute accuracy. The drawback, in this case, is the low speed and accuracy inherent to the human operator.

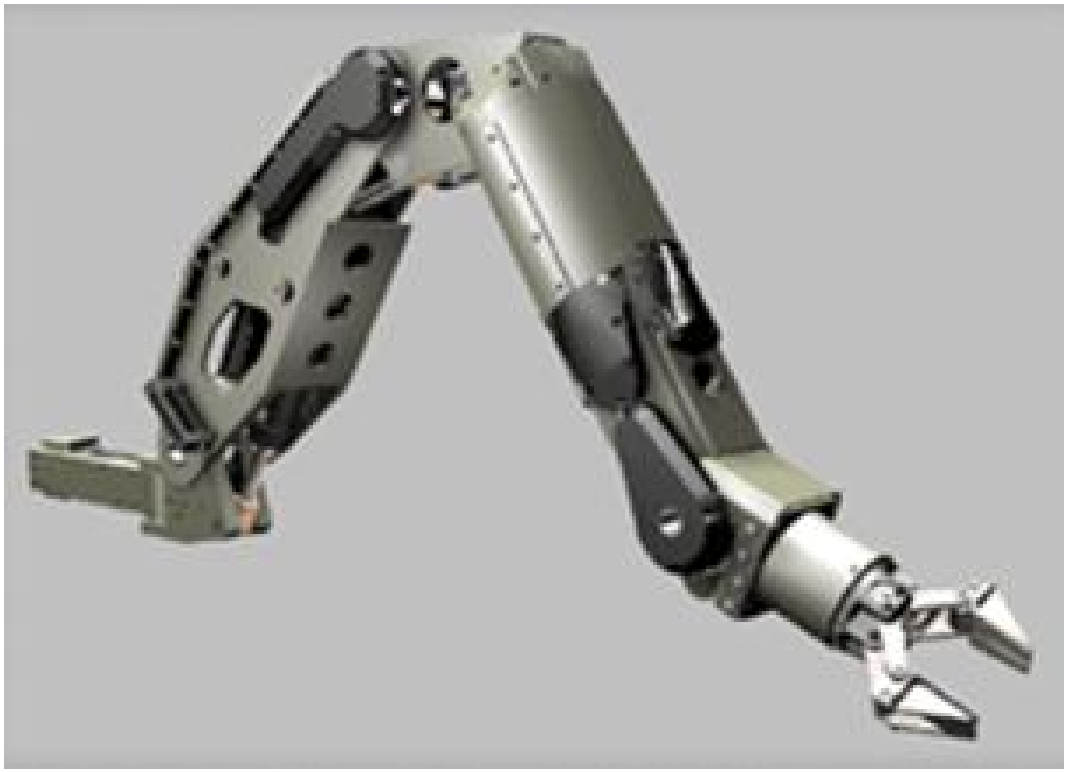

Figure 1.1: Hydraulic Manipulator Slingsby TA-40.

The manipulator considered in this research is the Slingsby TA-40, depicted in Figure 1.1, attached to a Remote Operating Vehicle (ROV), which

is brought to its working environment by the ROV operator. Every time the robot is repositioned, it needs to estimate its position and orientation relative to the work environment. The ROV operates at great depths and there are only a few sensors that can operate at such environments. This is the incentive for the use of Computer Vision to estimate the relative position of the manipulator. Through cameras, the differences between the actual and desired position of the manipulator can be estimated from pose detection, using either feature or appearance matching. This information is then sent to controllers to correct the pre-programmed trajectories. Due to their inherent higher accuracy, these methodologies can also be merged to perform in-situ calibration of the manipulator base.

This project originally addressed the recognition of an underwater target object, namely valve panels, which would be used as a secondary fine-tuning aid for a control system to correct the manipulator's trajectory. The main concern of the pose detection issue lies in the quality of the correspondences between the image pairs. Especially as the manipulator moves closer to the interest object, i.e., one of the valve panels. In such cases, if the matching lacks accuracy, the robot could fail to properly operate the valves or even collide against the panel, potentially damaging one another.

Computational efficiency is also of crucial importance, since the object tracking must be performed in real-time. Hence, the quest for a suitable trade-off between a fair enough match and an admissible processing time should be the primary aim of this research. So is the statement itself of what exactly can be considered as "fair enough" and "admissible", regarding the operation conditions.

Feature matching, namely interest points across image sequences, is the approach adopted to tackle the problem. Poor image quality and high levels of radial distortion, as well as the operation environment itself, where occlusion may occur quite often, are the greatest obstacles to obtaining a proper match.

The study comprises determining what algorithms can be used, or possibly adapted, to achieve the aforementioned trade-off, with respect to a diversity of scenarios. These vary from one another, for instance, according to the relative distance from the robot to the panel or by considering the frame sampling rate, as occlusion - or noise - might force to discard considerable pieces of the image sequence from time to time.

Since efforts are steered to both quality assessment and processing time of the matching procedure(s), a more straightforward algorithm is initially intended to implement the actual pose detection step of the control system.

The particular application concerning the valve operation by the Slingsby

TA-40 was, as mentioned, the main incentive for this research. However, the method proposed herein was not yet implemented to work with the manipulator itself. Tests were conducted in different environments, which eventually allowed to attest the method's ability to function in various conditions.

1.1 Objectives

The contributions of this thesis are manifold and can be assembled into two distinct objectives. The main purpose was to devise a novel real-time method for object tracking, based upon parallel processing and feature correspondence. The secondary goal entailed evaluating and ultimately electing point matching algorithms that could in fact make feasible the implementation of such tracking procedure.

Thus, both secondary and primary contributions, which are, respectively, the exhaustive tests on performance assessment of the matching techniques and the novel method for object tracking itself, are depicted next.

1.1.1 Secondary objective

As said, the greatest motivation of this project was the particular valve control application. Hence, the underwater environment was chosen for assessing the point matching procedures at an early stage.

At first, exhaustive tests were conducted, using underwater video sequences (some with quite poor quality), so as to evaluate a variety of matching techniques to decide which could be adopted. Later, a video sequence shot in a pool was used to stress the algorithms eventually selected as the most suitable.

Although such algorithms were to be implemented into a routine, which would function as a vision-based aid to help correct the trajectory calculated by the master control system and that would be called only eventually, the ability of working in real-time would be crucial to its success. Thus, different tests were conducted to address both accuracy and processing time.

Besides serving their purpose in this project, these experiments indeed provide insights on the performance of the algorithms tested that can be used as a knowledge tool in future related researches, as what regards how they are expected to behave under stressful conditions. However, they faded into a secondary contribution of this work, as they led to a step forward. Needless to say, the primary goal set on this research would have never been attained, unless its secondary objective was successful.

1.1.2 Primary objective

The knowledge extracted from the previous experiments allowed the implementation of a real-time object tracking method, that not only is self-sufficient, i.e., can be adopted as the control system itself, but is also adaptable to varying environment circumstances. Naturally, the method may still be used as the underwater vision-based aid discussed before. The key to its successfulness is parallel processing, as will be made clear in further chapters.

Nevertheless, once implemented, it would be desirable for the method to be capable of functioning in various conditions, other than the one that motivated its devising. Several other video sequences were used for further tests to address this matter. Being of different natures - regarding not only the environment, but also equipment and interest object - they provide ground for attesting the method's efficiency in various applications and explain the reason for proposing two variants.

1.2 Thesis outline

The remainder of this document is organised as follows. Chapter 2 presents general concepts of feature-based and appearance-based models for object tracking, a.k.a. identification or recognition, as well as a discussion of several other tracking methods, pointing the differences between their approaches and the one adopted in this work.

Chapter 3 briefly describes the point matching techniques used in the method's implementation. The method itself is presented in chapter 4, which contains the ideas underlaying its devising, as well as the details of both of its variants.

Finally, chapter 5 addresses the performance assessment of the point matching techniques and the method itself, in its two variants. Those were divided into two separate sections. The different assessment measures and their outcomes are displayed and discussed, as well as are presented a few hints on how each method's variant is best suited to a particular application. All of which followed by chapter 6's concluding remarks and directions to future works.

The appendix is divided into four chapters. Chapters A and B contain straightforward techniques of camera calibration and pose detection and present their basics. Chapter C contains the full records referring to the experiments conducted for assessing the overall performance of the point matching algorithms depicted in section 5.1.

At last, a visual aid for assessing the performance of the object tracking method is available in chapter D. The four videos show how each interest object was tracked in the respective sequences presented in section 5.2.

2
Related research

The vision-based object tracking issue derives a multitude of research lines compatible with the relevance of the matter. This chapter narrows its survey down to the main focus of this thesis: object recognition, presenting both feature-based and appearance-based mainstreams. The study of descriptors of image points and methods for object tracking from such features are also briefly introduced, as well as the camera calibration and pose detection issues.

2.1
Object identification

Object recognition is indeed a twofold issue. That is, provided there is (at least) one model of the interest object, the matter of recognising it entails *identifying* this object in an image and, then, *locating* its position in 3-D space. This section addresses the first problem.

Object identification can be perceived as determining which model in the database matches the data in the input image. In other words, it is essentially a matter of comparing input data to a model to decide whether they match and, thus, the successfulness of an identification algorithm relies a great deal on how effective is a model at describing a given object.

Namely, strategies for finding a match are divided in two basic categories, according to the model's nature: feature-based methods and appearance-based methods.

2.1.1
Feature-based methods

Features are instances extracted from numerical image data, whether related to a global property, e.g. a pixel's value, or a region having a special property, like edges. A feature must therefore be described by a collection of characteristics, so that it can be identified, assessed and compared with. Such collection is referred to as *feature descriptor*.

Thus, feature-based methods are applied to problems, where the goal is to either find instances of a given object in the input image(s); determine what object is there in a particular scene, or describe what exactly is a certain image region. Consequently, it becomes a matter of - given a model database of features and a set of features detected in an input image - implementing an algorithm for mapping the descriptors of the latter into the ones belonging to the first.

This thesis presents a method for object tracking that is based upon features, namely points. Several different descriptors are used, as will be clear from chapter 3 on.

As said, the ability of a descriptor in representing a feature is crucial for devising an effective object recognition method. Reference (20) surveys the performance of several feature descriptors. They were evaluated, compared to one another and finally ranked by the authors, who attest that the ones based upon image region perform best. Indeed, the method proposed herein uses only this kind of descriptors, with varying degrees of complexity.

As the descriptors based on the scale invariant feature transform (18), or SIFT, obtained the best results according to reference (20), this was the technique adopted for detecting points in the object tracking method proposed herein. An alternative detector, invariant to scale and affine transformations, may, however, be found in (19). For further readings, reference (21) reports a comparison of the performance of a few affine covariant region detectors, from which the authors have established a reference test set of images and performance software, so that other detectors can be evaluated in the same framework.

2.1.2 Appearance-based methods

Appearance-based identification uses images instead of features as models. The problem, thus, becomes determining whether a part of an input image is, in fact, an instance of the interest object. In such cases, the model is a set of images that represent several views of the tridimensional object and also different illumination conditions, or rather, a set of possible appearances.

Identifying an object would then be equivalent to comparing an input image to the set and deciding whether it contains an image that is similar enough to be regarded as the same object.

Appearance-based methods compare the input data directly with the model, since they are both images. At first, this may seem an obvious advantage over their counterparts, which require that features be detected and described

before the match process can actually start. However, in a great deal of applications, the database usually grows extremely large to become really representative of the interest object. In such cases, the memory usage can be unmanageable to the point that the method becomes unfeasible.

2.2 Object location

Locating objects in space equals determining their position and orientation - or rather, their 3-D translation and rotation - provided their model is known. In other words, object location is, in fact, a pose estimation problem.

The structure of the location algorithm depends on how the camera was modelled. If the projective model is adopted, it becomes the problem of handling a system of non-linear equations, which can be done by applying Newton's iterative method. In case of an affine model, the algorithm translates into solving a linear system, a quite simpler task.

Both variants are engaged in determining the extrinsic camera parameters, a major part of the camera calibration problem, which are ultimately used for estimating the object's pose.

Since the method for object recognition proposed herein uses image points as features, the matched conjugates themselves become the inputs of the standard camera calibration and pose detection routines. Straightforward techniques and basic theory on both issues are available at the appendix (chapters A and B , respectively) and (26) describes a method for calibrating high-distortion cameras. However, a deeper discussion on such matters falls outside the scope of this document.

2.3 Other works

Two fundamental issues present the greatest difficulties, as far as the problem of object recognition based on image points is concerned: processing time and exactly *what* are the most suitable points to use as both input data and model.

The first is a stumbling block for real-time applications, due to the computation cost involving the state-of-art algorithms for point detection and matching. The latter concerns the quality of the tracking algorithm more directly. Not only all object views must be modelled by a single set of image points - so that it can be identified (or tracked) at all times - but also the tracking system must account for problems regarding the environment conditions - e.g. occlusion, varying illumination patterns and view angle - and

be robust to them. This means that the input features have to be equally well representative and furthermore less subjected to a false match. Some real-time procedures are presented next, and eventually compared to the method introduced in chapter 4.

A vision-based localisation and mapping procedure, reported in (24), devised to work in unmodified dynamic environments, uses SIFT features for estimating a robot's pose, from a three-camera stereo system. Here, a major concern was to implement an algorithm for deciding which features were most suitable for being part of the object's model as the robot moved around. Additionally, there was the safeguard of relying on odometry information to predict the feature characteristics for each model point in the next frame and to eventually estimate the pose, in case no sufficient matches were available.

From the same authors, (25) describes a method for global localisation, where the robot localises itself, without any prior estimate. This is achieved by matching features in the current frame to a model map. Se et al. have proposed two different, and costly, approaches, namely the Hough Transform (8) and the Random Sample Consensus (10), or RANSAC, that proved to yield the best results and through which global localisation is performed by finding the robot pose supported by most image points.

Another method for object tracking is based on a different, alternative, descriptor. Described in (29), it uses Fern-based classification to match image points. The descriptors, named *ferns*, are non-hierarchical structures for classifying image patches. They consist of small sets of binary tests and return the probability that a patch belongs to any one of the classes that have been learned during the training phase, which consists of synthesising many views of points extracted from a training image. These responses are then combined in a Naïve Bayesian way. This is actually an open source, available at `http://cvlab.epfl.ch/software/ferns/index.php`, and, thus, it was intended as a benchmark. However, despite showing satisfactory results in face detection applications, it failed to track the objects in all four sequences used to test the method described in 4.

3
Fundamental algorithms for point matching and underlaying theory

This chapter presents a description of the three matching techniques adopted for the implementation of the proposed object tracking algorithm. The theory concerning the pose estimation, for determining the actual object location, is also discussed in B. Nevertheless, as what regards the pose detection and camera calibration issues - these last presented in A - only standard techniques are described, since investigating these matters falls outside the scope of this thesis.

The contents of this chapter suffice to fully understand the functioning of the proposed method. However, by no means are they thorough to the extent of exhausting these subjects.

As presented in chapter 4, three algorithms for matching are combined into a keypoint tracking process with a decaying degree of complexity. Namely, the normalised cross correlation (NCC), least squares matching (LSM) and the scale invariant feature transform (SIFT). The theory that underlay these algorithms is discussed next.

3.1
Normalised cross correlation

The cross correlation function measures the degree to which two signals are similar (or correlated). In image processing applications, these are matrices representing patches of each image.

Since their brightness may vary and to allow the pattern recognition to be independent of scale and offset, the images should be normalised. This is done by subtracting the mean at each position and then dividing by the product of the standard deviations of the patches, in a computation that can be perceived as the dot product of two normalised vectors.

Patches cropped around an image point are defined as area-based descriptors as what concerns the feature matching matter. The normalised cross correlation (NCC) of two windows of size W built around points p_t and p_m in the template and matching images, respectively, is then given by eq. (3-1),

whereby $I(p)$ denotes the image intensity value at location p.

$$NCC = \frac{\sum_{W}\left[\left(I_t\left(p_t\right) - \overline{I_{t_{W_{p_t}}}}\right)\cdot\left(I_m\left(p_m\right) - \overline{I_{m_{W_{pm}}}}\right)\right]}{\sqrt{\sum_{W}\left(I_t\left(p_t\right) - \overline{I_{t_{W_{p_t}}}}\right)^2 \cdot \sum_{W}\left(I_m\left(p_m\right) - \overline{I_{m_{W_{pm}}}}\right)^2}} \tag{3-1}$$

The image of NCC falls within $[-1, 1]$ and, for identical patches, it equals 1. Patches that have no similarity to one another incur a 0 correlation factor. A value of -1 would indicate two inversely correlated patches, as, e.g., a template-matching pair formed by the diapositive and the negative of an image.

Naturally, if the matching image is merely a shift transformation of the template, the perfect match is easily found by sliding the matching window around the search space, provided its wide enough to actually include the correspondent coordinate. Otherwise, the match is to be considered as the point within the search space, whose window yields the maximum correlation factor.

3.2
Least squares matching

The least squares matching (LSM) (1, 13) is another area-based technique for finding conjugate points in a pair of images. The underlaying idea is to minimise the intensity level differences between the template and matching patches. In LSM, however, both the centre and the shape of the matching window are to be determined, which is done by small incremental adjustments in an iterative process of a sub-pixel resolution. An initial guess for the location of the matching point is, therefore, needed to launch the procedure, as well as starting values for the transformation parameters.

In other words, LSM iteratively searches for the best parameter values of a pre-determined geometric transformation that maps the matching image into the template, by computing the grey level differences between the windows around the template points and the ones built and transformed around their respective conjugates. Thus, it is necessary to first define what transform is more likely to map one image into the other: projective, affine or shift. The projective transformation includes the other two, as the affine includes the shift. Nevertheless, having to adjust a larger set of parameters not only reduces the convergency rate, but can also potentially lead to stability problems. Thus, the appropriate geometric transformation for a particular problem is the one

able to do the mapping, with the required accuracy and the lowest possible complexity.

Additionally, it may be necessary to introduce a radiometric transformation T_r to model eventual differences in contrast (r_0) and brightness (r_1) between the two images. Eq. (3-2) shows the transformation applied to a given matrix M.

$$T_r(M) = r_0 \cdot M + r_1 \tag{3-2}$$

Although it is possible to add the radiometric transformation to the LSM process, this computation is generally done as a pre-process for the reasons presented above. In this work, only the affine and shift transformations were applied to the LSM routine, as will become clear through the next two chapters.

The affine transformation, defined by the six t_i terms, of a pixel $p_0 = [x_0, y_0]^T$ in the initial matching window M_0 at each iteration is given by

$$\begin{bmatrix} x_T \\ y_T \end{bmatrix} = \begin{bmatrix} t_0 & t_1 & t_2 \\ t_3 & t_4 & t_5 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} \tag{3-3}$$

The transformed window M - at each iteration step - can then be approximated by eq. (3-4), where the $\triangle t_i^j$ term would denote the adjustment of the i^{th} parameter at iteration j.

$$M \approx M_0 + \frac{\partial M}{\partial x_T} \cdot \sum_{i=0}^{2} \frac{\partial x_T}{\partial t_i} \triangle t_i + \frac{\partial M}{\partial y_T} \cdot \sum_{i=3}^{5} \frac{\partial y_T}{\partial t_i} \triangle t_i \tag{3-4}$$

The LSM procedure then searches for the parameter values that yield the residual matrix R having the minimum variance estimative. Matrix R is written as the difference between the template window T and M, as defined in eq. (3-5).

$$R = T - M \tag{3-5}$$

Merging eq. (3-4) and eq. (3-5) yields

$$R = T - M_0 - \frac{\partial M}{\partial x_T} \cdot \sum_{i=0}^{2} \frac{\partial x_T}{\partial t_i} \triangle t_i - \frac{\partial M}{\partial y_T} \cdot \sum_{i=3}^{5} \frac{\partial y_T}{\partial t_i} \triangle t_i \tag{3-6}$$

Writing eq. (3-6) for all pixel pairs in T and M yields a linear system, where the unknown $\triangle t_i$ coefficients are calculated at each iteration, until convergence, or the maximum number of iterations is reached. The final transformation coefficients are given by the sums of all $\triangle t_i$, and the conjugate point, by the centre of the matching at the last iteration. The quality of the match can be assessed by computing the variance of the residuals or, alternatively, the NCC between the template and the final matching windows.

Applying the LSM technique to the shift transformation is analogous as described, only taking $t_0 = t_1 = t_3 = t_4 = 0$. Notice that such a case would be equivalent to a sub-pixel resolution NCC. For simplicity, from hereon, both NCC and Shift-LSM will be referred as the former, whereas LSM will denote the Affine-LSM routine.

3.3 Scale invariant feature transform

The scale invariant feature transform (SIFT) (16, 18), devised by D. Lowe, is a powerful point detector that rely on an orientation-histogram-based descriptor. Such keypoints are invariant to image scaling and rotation, and partially invariant to changes in illumination and 3-D camera viewpoint. Their distinctiveness allows each of them to be robustly matched against large databases built from many images.

The steps to recognise a feature as a reliable keypoint are:

1. Scale-space extrema detection: use of a Difference-of-Gaussian (DoG) Pyramid (5) to search for features that are invariant to scale and orientation;

2. Accurate keypoint localisation: keypoints considered stable have their location scale determined down to a sub-pixel resolution;

3. Orientation assignment: orientations based on local gradient directions are assigned to each keypoint location;

4. Keypoint descriptor: built from local gradient measures at the selected scale around the keypoint.

The point matching is done by running the algorithm to identify the keypoints of an individual image and then compare each one against the SIFT feature database previously built from a set of model images.

The SIFT code that has been implemented for this work is a modified version of the one written by R. Hess, available at http://web.engr.oregonstate.edu/~hess/index.html.

The four stages of the SIFT algorithm are discussed in some detail next.

3.3.1 Scale-space extrema detection

At first, keypoint candidates are detected and roughly located using cascade filtering. A set of scales is produced by repeatedly applying the continuous scale space function (28) to the input image. This function uses the Gaussian function, given by eq. (3-7), as kernel.

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \tag{3-7}$$

That is, the space $L(x, y, \sigma)$ of the image $I(x, y, \sigma)$ is defined by

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y, \sigma) \tag{3-8}$$

The entire set of scales, or rather, the pyramid of scale-space images is searched for invariant features.

Actually, feature stability can be more efficiently assured by detecting scale-space extrema in the Difference-of-Gaussian function $D(x, y, \sigma)$, rather than in $L(x, y, \sigma)$. The DoG is defined as the difference of two adjacent scales, separated by a constant multiplicative factor k (see eq. (3-9)). Since the computation of $L(x, y, \sigma)$ is mandatory for scale-space feature description, no significant additional cost is incurred from calculating $D(x, y, \sigma)$.

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \tag{3-9}$$

The local spaces for extrema detection consist of triplets of DoG image patches, as depicted in Figure 3.1[1]. The central patch is cropped around each

[1]All figures used as visual aids in the description of the SIFT algorithm were cropped directly from (18).

point in the inner pyramid scales and the two others located at the same x-y coordinates in the upper and lower adjacent levels. If the centroid of each cubic region corresponds to its maximum or minimum, then it is a extrema location, i.e., a keypoint candidate.

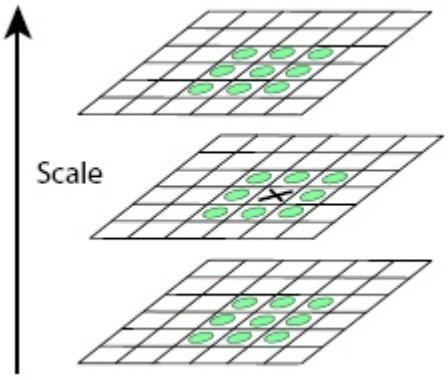

Figure 3.1: Searching region for scale-space extrema detection.

3.3.2 Accurate keypoint localisation

The keypoint candidates have their x-y localisation and scale adjusted down to sub-pixel resolution to improve stability even further. This fine tuning is done by drawing a 3-D quadratic curve calculated by Taylor expansion of $D(x, y, \sigma)$ and then shifted, so that its origin is at the candidate's location, as given by eq. (3-10), whereby $\mathbf{x} = [x, y, \sigma]^T$.

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}} + \frac{1}{2}\mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x} \tag{3-10}$$

The derivatives are approximated by finite differences, thus the samples for the calculation of each accurate keypoint position are the candidate itself and its immediate neighbours in all three dimensions (x, y and σ). In eq. (3-10), D and its derivatives are taken at the sample point that is considered to be the closest to the accurate location, i.e., the central sample point (initially, the keypoint candidate) and $\mathbf{x}$ is the offset from this point. Determining the magnitude of the fine adjustment is equivalent to finding the values of $\mathbf{x}$ that correspond to the extremum point of the curve, here denoted by $\hat{\mathbf{x}}$.

Deriving eq. (3-10) with respect to $\mathbf{x}$ and setting it to 0 yields the 3×3 system

$$\begin{bmatrix} \frac{\partial^2 D}{\partial \sigma^2} & \frac{\partial^2 D}{\partial \sigma \partial y} & \frac{\partial^2 D}{\partial \sigma \partial x} \\ \frac{\partial^2 D}{\partial \sigma \partial y} & \frac{\partial^2 D}{\partial y^2} & \frac{\partial^2 D}{\partial y \partial x} \\ \frac{\partial^2 D}{\partial \sigma \partial x} & \frac{\partial^2 D}{\partial y \partial x} & \frac{\partial^2 D}{\partial x^2} \end{bmatrix} \cdot \begin{bmatrix} \sigma \\ y \\ x \end{bmatrix} = - \begin{bmatrix} \frac{\partial D}{\partial \sigma} \\ \frac{\partial D}{\partial x} \\ \frac{\partial D}{\partial y} \end{bmatrix} \tag{3-11}$$

for whose coefficients the linear approximations by finite differences are:

$$\begin{cases} \frac{\partial D}{\partial \sigma} = & \frac{D_{i,j,k+1} - D_{i,j,k-1}}{2} \\ \frac{\partial D}{\partial y} = & \frac{D_{i,j+1,k} - D_{i,j-1,k}}{2} \\ \frac{\partial D}{\partial x} = & \frac{D_{i+1,j,k} - D_{i-1,j,k}}{2} \end{cases}$$

$$\begin{cases} \frac{\partial^2 D}{\partial \sigma^2} = & D_{i,j,k-1} - 2D_{i,j,k} + D_{i,j,k+1} \\ \frac{\partial^2 D}{\partial y^2} = & D_{i,j-1,k} - 2D_{i,j,k} + D_{i,j+1,k} \\ \frac{\partial^2 D}{\partial x^2} = & D_{i-1,j,k} - 2D_{i,j,k} + D_{i+1,j,k} \end{cases}$$

$$\begin{cases} \frac{\partial^2 D}{\partial \sigma \partial y} = & \frac{(D_{i,j+1,k+1} - D_{i,j+1,k-1}) - (D_{i,j-1,k+1} - D_{i,j-1,k-1})}{4} \\ \frac{\partial^2 D}{\partial \sigma \partial x} = & \frac{(D_{i+1,j,k+1} - D_{i+1,j,k-1}) - (D_{i-1,j,k+1} - D_{i-1,j,k-1})}{4} \\ \frac{\partial^2 D}{\partial y \partial x} = & \frac{(D_{i+1,j+1,k} - D_{i+1,j-1,k}) - (D_{i-1,j+1,k} - D_{i-1,j-1,k})}{4} \end{cases}$$

An absolute value greater than 0.5 in any direction of $\hat{\mathbf{x}}$ means that the accurate keypoint location is actually closer to another sample. In such cases, the central point must shift to the respective sample coordinates. This process is repeated until $\hat{\mathbf{x}}$ has all its values below 0.5 and the accurate position of the keypoint is then obtained by adding the adjustment $\hat{\mathbf{x}}$ to the central point location.

For stability reasons, the keypoint candidate has to pass a contrast filter. Namely, if $|D(\hat{\mathbf{x}})|$ is below an user-defined threshold (reference (18) suggests a

bound of 0.03, for normalised image pixel values), the correspondent keypoint is eliminated.

The remaining candidates have to go through yet another filter to be considered stable enough to enter the keypoint list. The DoG function is typically quite responsive to edges, even if a candidate location is fuzzy around it. Such features can be detected by using the ratio between the trace and the determinant of the Hessian matrix **H** computed at their position (see eq. (3-12)).

$$\frac{Tr(\mathbf{H})}{Det(\mathbf{H})} = \frac{(r+1)^2}{r} \tag{3-12}$$

whereby r is the ratio between the largest and the smallest eigenvalue of **H** ($r = 10$ is the ratio used on tests reported in (18)).

Candidates that pass both filters are elected SIFT features, or keypoints, and will undergo the last two steps of the algorithm.

3.3.3 Orientation assignment

Invariance to image rotation is achieved by representing the keypoints relative to their orientation. The computations are made at the $L(x, y, \sigma)$ that is closest to each keypoint scale, so that the features remain invariant to changes in this dimension. The magnitude $m(x, y)$ and orientation $\theta(x, y)$ of $\nabla L(x, y, \sigma)$ are calculated by eq. (3-13) and eq. (3-14), respectively.

$$m(x,y) = \sqrt{[L(x+1,y) - L(x-1,y)]^2 + [L(x,y+1) - L(x,y-1)]^2} \tag{3-13}$$

$$\theta(x,y) = \arctan\left(\frac{L(x+1,y) - L(x-1,y)}{L(x,y+1) - L(x,y-1)}\right) \tag{3-14}$$

An histogram of orientation is built for each keypoint and the pixels that surround it. The entries are weighted by their magnitudes and a gaussian window with a standard deviation typically 1.5 times greater than the feature scale. Each bean of the histogram covers a 10° range.

The highest peak in each histogram, plus any other that eventually is within 80% of it, are selected to be feature orientations. Since there can be

more than one peak, spin-off keypoints having the same (x, y, σ), but a different orientation, may be created. The orientations are also adjusted by fitting a parabola to the three histogram values closer to each peak.

3.3.4 Keypoint descriptor

Up to this stage, the resulting features are invariant to scale and orientation. A descriptor for the area around the keypoint must then be computed to achieve robustness regarding illumination changes, 3-D view point and even non-rigid deformations. Figure 3.2 provides an example of the entire process.

The approach for building the keypoint descriptor was inspired by the idea that had been introduced in (9). The first step is to sample the gradient magnitudes and orientations around the keypoint, whose scale defines the level of gaussian blur for the gradient image. Rotating both gradient orientations and descriptor coordinates relative to the feature orientation assures invariance to this parameter. Then, the magnitude of each sample point around the feature is weighted by a gaussian function with standard deviation equal to half the width of the descriptor window.

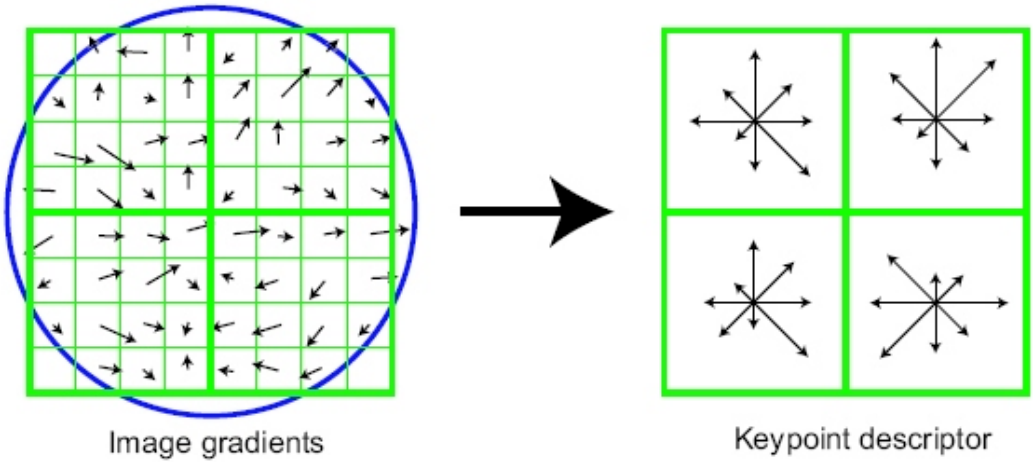

Figure 3.2: A 2×2 descriptor, on the right, computed from four 4×4 subregions, on the left, where the gaussian window is indicated by the overlaid circle. The length of each descriptor arrow corresponds to the sum of the weighted gradient magnitudes near that direction within the respective subregion.

The sample area is divided into sub-regions, each represented by an orientation histogram. Their entries are the sum of all weighted sample magnitudes within the sub-region, in the respective orientation range. Before the summation, however, these samples are multiplied by yet another weighting factor, which is inversely proportional to the gap between their orientation and the centre of their respective bins.

The histograms are assembled and then normalised to form the keypoint descriptor. Finally, to favour the contribution of orientation over magnitude, all descriptor entries are limited to a given upper bound and then re-normalised.

Experiments reported in (18) use 16×16 sample areas and 4×4 sub-regions, represented by 8-bin histograms, as well as an upper bound of 0.2.

3.3.5 Matching

Euclidean distance between two descriptors is the measure assessed to identify a conjugate pair of keypoints. The database feature within the closest distance from a given keypoint descriptor is its match, which must then be validated.

For every false match, there is likely a number of other ones within similar distances, due to the high dimensionality of the descriptor space. Therefore, a match is considered valuable, if the difference between the closest and second-closest distances is wide enough.

Exhaustive search for the conjugate pair would incur a tremendous computational cost. Thus, the match searching engine is KD-tree-based (12). An alternative procedure, the Best-Bin-First (BBF) algorithm (4), returns the closest descriptor with high probability by searching the bins in the order of their closest distance from the query location.

4
Proposed methods for keypoint tracking

This chapter introduces two algorithms devised for tracking keypoints in a video sequence. The following sections describe in details each one individually, but the general guidelines given in this preamble will help understanding their broad functioning, for they may actually be perceived as two variants of the same method, rather than different approaches to tackle the same problem.

In fact, the choice of which to adopt should be problem-driven, i.e., regarding the particular characteristics of a sequence, whose terms will be discussed in the forthcoming chapter. Both variants are a potpourri of the algorithms for point matching presented in 3 - or a variation of some of them - and the contents of each mix are what makes one brand more suitable for dealing with a specific kind of movie.

Generally, in a video sequence, two subsequent frames are converted into the image pair for tracking each keypoint position. The processing time of the matching procedure determines the frame sampling rate; that is, the faster the procedure, the closer (in time scale) the two frames.

Consecutive frames that are closer in time tend to be more similar to one another, making keypoints easier to trace. Thus, in such cases, the robustness demanded of the matching technique may drop, to a certain degree, without corrupting the tracking progress. That said, one is left with the circular trade-off dilemma depicted below, in Figure 4.1.

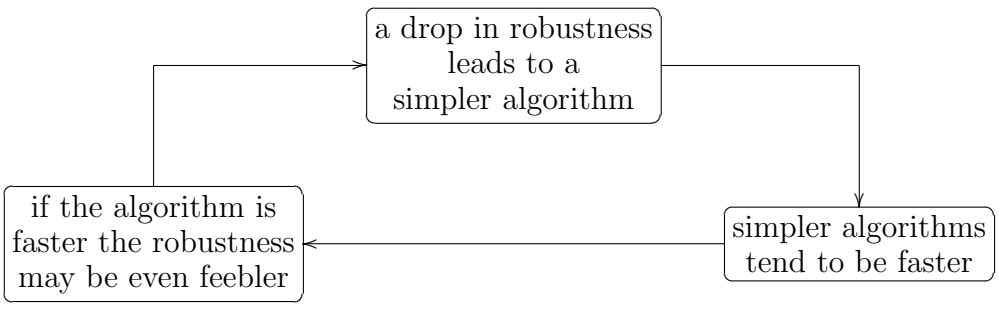

Figure 4.1: Scheme that drove the devising of the object tracking method.

Leaving the relation between robustness, speed and complexity of the matching procedures for a later discussion, there is still the matter of initialising - and maintaining - a keypoint collection, so that the tracking process may actually be carried on, to be taken under consideration. Typically, matching procedures that are less robust are also more likely to lose track of a greater number of keypoints during the process. Thus, such collection must be *updated* to either replace lost keypoints, refresh their coordinates, or even add new ones that are more suitable to the current scene.

Moreover, environmental problems, such as occlusion and change in illumination, make the selection of the initial keypoints even less trivial and in fact suggest such collection should not be statical. Which brings on the matter of how to assess the suitability of a keypoint to the current scene, as addressed in (24) and (25), for instance.

The method proposed herein makes needless to implement any heuristics in such assessment. The adequacy of the collection is automatically assured by the fact that its entries are the keypoints found as close in time as possible to the sequence frame being processed at a given instant, and thus they have a higher chance of being consistent with the upcoming scene. This is achieved by parallel processing, i.e., separate threads for the tracking and updating processes.

The block diagram depicted by Figure 4.2 illustrates the overall process throughout time.

Prior to running the tracking algorithm, an object model must be built off-line. This model consists of a keypoint list - which may be analyst-selected, to assure stability - containing the features' x-y coordinates, associated to a particular descriptor, and their correspondent world coordinates, for the object pose detection calculations, which can be performed, for instance, by a third thread. Since the SIFT algorithm was adopted at the early tracking stage in both variants, the descriptor used is exactly as described in (18). The collection must contain keypoints corresponding to all views of the object.

The updating strategy is as follows. Using the past matching sequence frames, a thread should be held responsible for carrying on the collection update. New features, in these last fames, are matched to the original model image(s) to assemble a new starting list, i.e., the keypoint collection to be processed by the next tracking thread. Finally, a mask matrix is associated to the starting list. These keypoints are the ones most likely to be consistent with the scenes from this moment on - since they were spotted at frames corresponding to immediately past time instants - therefore making them the best candidates for tracking. They must be matched against the original model,

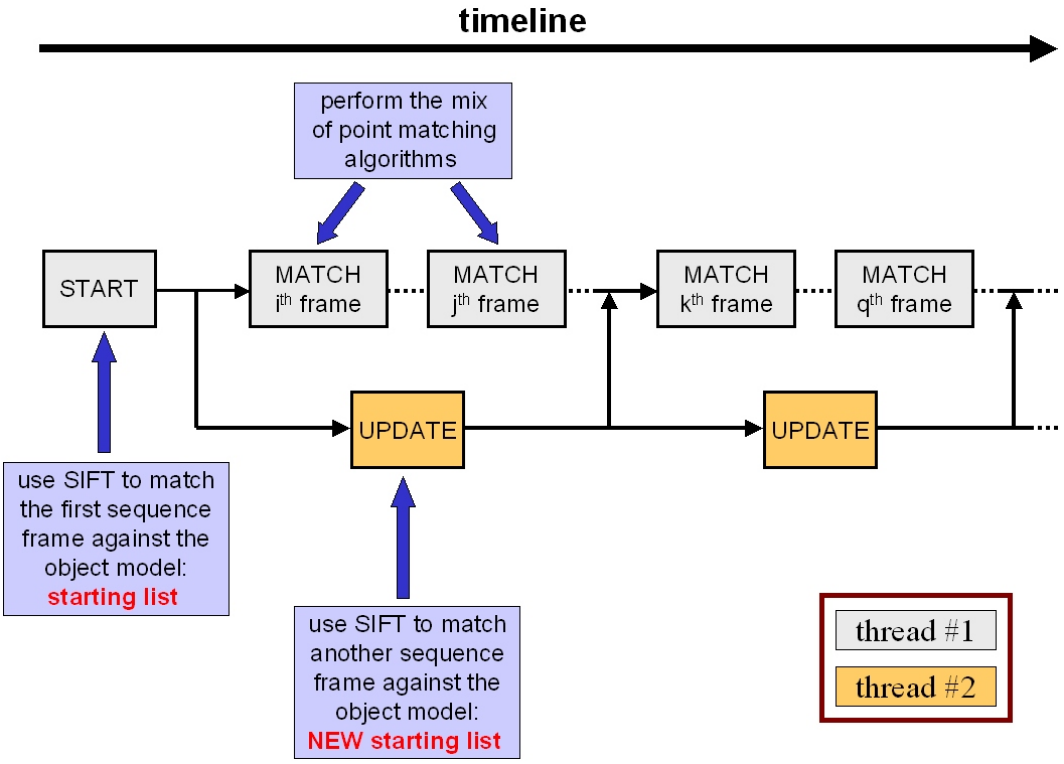

Figure 4.2: Block diagram of the object tracking process.

as to assure they fall in the area defined as being stable at the off-line training stage.

As soon as an update is ready (and the current tracking thread is idle), the keypoint coordinates corresponding to the template image, or rather, the template coordinates, are refreshed - or replaced by others - and the matching process continues. The old tracking thread dies as another update thread starts. All these new keypoints are flagged as "fit" in the mask matrix. During the matching process, whenever the algorithm fails to keep track of a feature, its template coordinates are tagged as "unfit" and cease being part of the tracking process at the subsequent frames, remaining latent until the current thread dies, but remembering that they may still be members of the next list delivered by the following update.

The keypoint updating process is kept indefinitely ongoing. The only requirement is that there must be a new set of keypoints ready, whenever the number of fit pairs falls beneath a certain threshold, which may be around the minimum points required for the 3-D reconstruction (pose detection). If, by any chance, the list is entirely lost before the update process has delivered the next one, the object will remain untraceable for that period.

Having introduced the kernel of the object tracking process, let us present in details the two variants devised for running in the matching thread.

4.1 Variant 1

As previously stated, the matching algorithms implemented for the actual object tracking that are fast enough to keep a high frame sampling rate, are also less robust, due to their simpler nature. This lack of complexity demands keypoints that are themselves both inherently easy to match and consistent with the upcoming scene, which is achieved by the updating procedure described in the previous section. Having addressed the point stability matter, let us discuss how the matching techniques presented in chapter 3 can be used to take full advantage of the keypoint collection quality.

At the very beginning of the tracking process, there is hardly any knowledge about the camera position, in relation to the pose of object model (or the keypoint collection). Hence, the point correspondence in the template-matching frame pair at this stage is rather more difficult, in relation to the ones hereafter. Therefore, a more robust procedure should be adopted.

From hereon, as to assure keypoint stability and pairing consistency, an algorithm that can not only provide a match, but also eliminate false initial matches is desirable. As the video sequence moves further in time and the camera movements have stabilised, as well as the false matches have been eliminated, the tracking process will not require a great filtering ability nor a wide image transformation coverage. Consequently, the matching algorithm as the current tracking thread advances in time can be even simpler.

Based upon the discussion above, the mix proposed for this variant is as follows. Figure 4.3 shows the object tracking scheme.

- apply the **SIFT** algorithm and match the starting frame against the model
- whenever the former step is completed, grab the incoming frame and use **LSM** to adjust the coordinates of all keypoints, using their present values, i.e., the **SIFT** output, as initial guesses and with the last processed frame - the SIFT-frame - as the template
- consecutively apply **NCC** from now on to refresh the still fit keypoints; the initial guesses and template are analogous to the LSM-step

This procedure restarts every time a collection update is released. The update processes are in fact analogous to the first step and replace it as the scheme goes further in the time line. Notice that the number of frames lost, i.e., left unprocessed during each match calculation drop significantly from stage to stage, since the procedures themselves get faster and faster and, additionally,

more keypoints become unfit (and no longer adjusted), although the latter aspect is a direct consequence of adopting this procedure rather than properly an advantage.

Chapter 5 provides quantitative information to buttress the strategy devised for this variant.

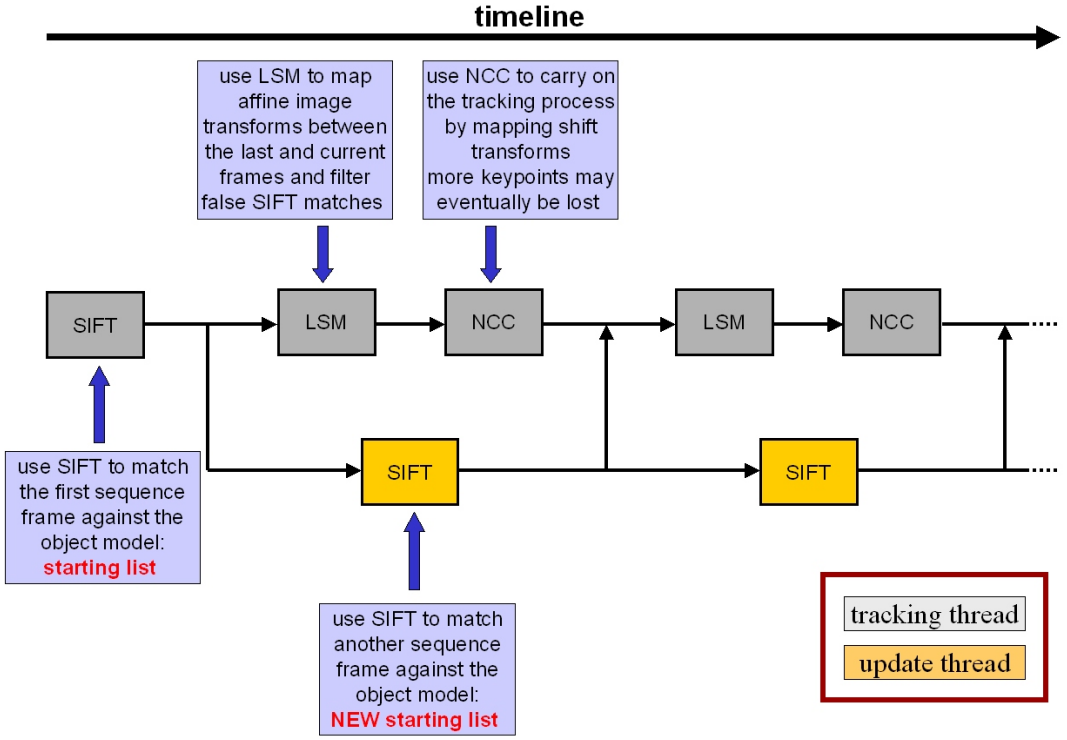

Figure 4.3: Block diagram of the object tracking method: variant 1, also called the *SLN variant*.

In the remaining of this document, variant 1 will be designated by the initials of the algorithms that compound the mix (namely, SIFT, LSM and NCC), that is, by the **SLN** abbreviation.

4.2 Variant 2

Both the LSM and the NCC algorithms have matrices of neighbouring image pixel values as descriptors. Hence, not only they behave rather poorly in cases where the geometric difference between the pairs of frames is too significant - like in high movement sequences - but they also may become quite unstable when processing severely compressed video sequences, namely those where the quantisation blocks are noticeable and a pixel's neighbours - and even location - are far from corresponding to the ones in the original uncompressed frames.

This second variant was devised to deal with such sequences and its functioning is very similar to its peer's. Here, the LSM-NCC routine is replaced by

a SIFT-based matching technique. Figure 4.4 depicts the alternative matching strategy.

Running the SIFT algorithm as implemented in the first step throughout the whole sequence would incur a massive loss of image frames, due to its processing time. Again, if the subsequent frames used as template-matching pairs are kept close enough in time to one another, it is fair to admit that a keypoint location will not wander off, in terms of absolute image coordinates, regardless the significancy of the geometric transformation it went through. Therefore, processing only a patch of the matching frame, around the coordinates of each template keypoint, should be enough to determine the new location.

Regarding this variant, the matching technique consists then in consecutively applying the SIFT algorithm to each keypoint individually, considering the image to be only a small area around it (in other words, locally applying the algorithm), until the update process is completed.

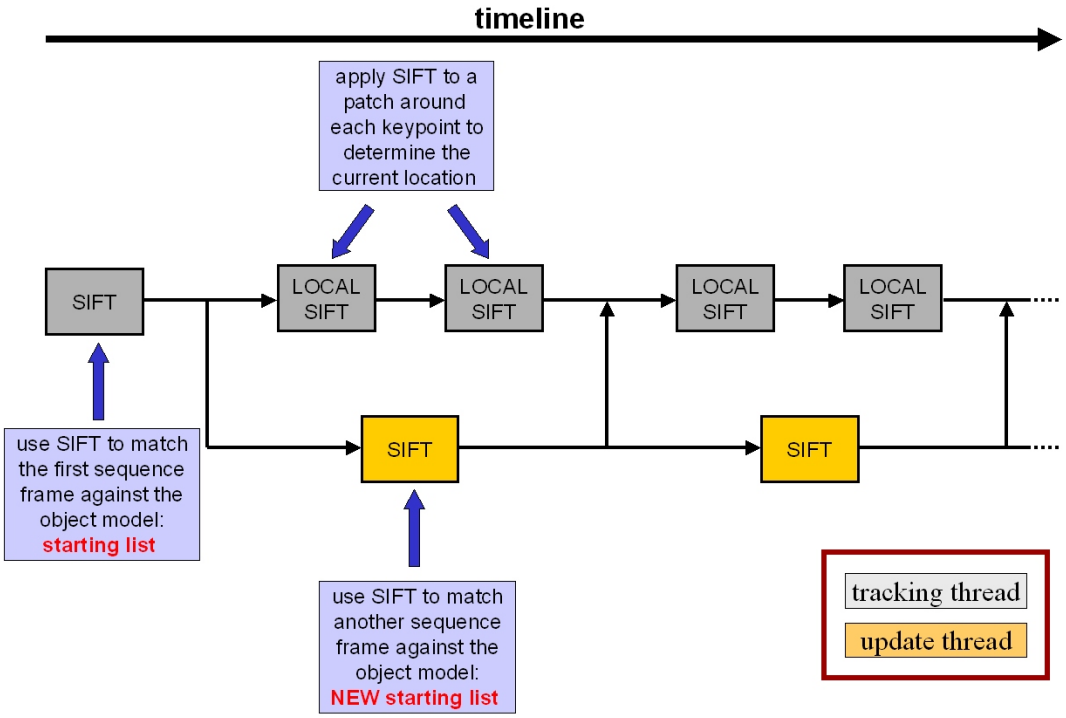

Figure 4.4: Block diagram of the object tracking method: variant 2, also called the *SLS variant*.

Evidently, this second variant can be successfully applicable to all natures of video sequences, including the ones best suited to the one presented in the previous section. However, since the SIFT algorithm, even when applied locally, is slower than both LSM and NCC procedures, the frame sampling rate would be lower, leaving more frames unprocessed. In fact, if applying both methods to the same low movement, not severely compressed, sequence and then playing

only the processed frames as a movie, the video made using the frames output by the second variant would appear rather jerky, when compared to the other.

Analogously, this variant will be referred to as the **SLS** variant from hereon.

4.3 General remarks on performance

Commenting on the overall method in terms of programming code, there are two identical structures that carry keypoint collections, one referred as the updating list and the other, as the tracking list. Whenever an update is ready, they simply switch their memory addresses, which makes the refreshing of the tracking keypoints virtually instantaneous. Moreover, since multi-core processors have become quite dominating, the parallelism of the method proposed herein is easily implemented.

In addition, the greatest stumbling block, regarding the method's performance in both variants, is the processing time of the matching procedures. Thus, as the CPU clock frequency increases, so does the quality of the results achieved by the object tracking algorithm presented in this chapter, both in number of frames processed and location accuracy, the latter being a direct consequence of the former. Moreover, faster machines also lead to an increase in the collection updating rate, hence the consistency of the keypoint lists with the changing scene escalates, as well, which further contributes to the method's accuracy.

5
Performance assessment

This chapter is organised as follows. The first section provides a discussion on the quality assessment of all the point matching algorithms addressed in chapter 3, which founded the theory behind the object tracking method previously discussed.

The last section presents the performance evaluation of the method itself. General guidelines on which variant to use, are given as well, according to the nature of a sequence, in terms of environment, video quality and compression characteristics.

All tests reported herein were conducted in a Pentium Quad-Core Q6600 Processor (2.40 GHz – 3.23 GB RAM).

5.1
Performance evaluation of the matching techniques

The quality assessment of the algorithms were made considering both accuracy, as defined in section 5.1.1, and processing time. Different databases were used to cover each aspect.

5.1.1
Accuracy measures

The database used for measuring the accuracy of both the SIFT and LSM procedures was built from a single 640×479 image, which underwent five different affine transformations. These transformations had to be applied artificially, so that the ground truth would be known.

This is actually an intensity level image correspondent to a frame of a video sequence shot underwater[1], which is the source of various test data used in the experiments reported herein. The fact that it was shot with a camera that is most likely to be used in the valve control application, described in section 1, is the main reason why the experiments resorted so much to it. Additionally, the environment is as close as possible to the real conditions of such application, when compared to the rest of the test data available.

[1]This source also engendered the "Pool" video sequence and, therefore, its properties are fully addressed in section 5.2.1.

Building the database

The five affine transformations (I-V), defined by the 3×2 matrix in eq. (5-1) and used to assess the accuracy of the point matching algorithms, are listed next.

$$\begin{bmatrix} x_{tr} \\ y_{tr} \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{5-1}$$

$$\begin{cases} I: & \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \\ II: & \begin{bmatrix} 0.5 & 0.1 & 2 \\ 0.1 & 0.5 & 2 \end{bmatrix} \\ III: & \begin{bmatrix} 1.1 & 0.1 & 2 \\ 0.1 & 1.1 & 2 \end{bmatrix} \\ IV: & \begin{bmatrix} 1.1 & 0.05 & 3 \\ 0.05 & 1.1 & 3 \end{bmatrix} \\ V: & \begin{bmatrix} 1.2 & 0.2 & 1 \\ 0.2 & 1.2 & 1 \end{bmatrix} \end{cases}$$

The results of applying each of the transformations above to the original image are shown below.

Figure 5.1: Original image.

Figure 5.2: Affine transformation I.

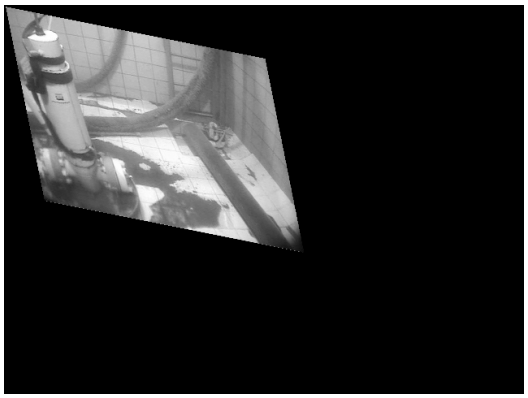

Figure 5.3: Affine transformation II.

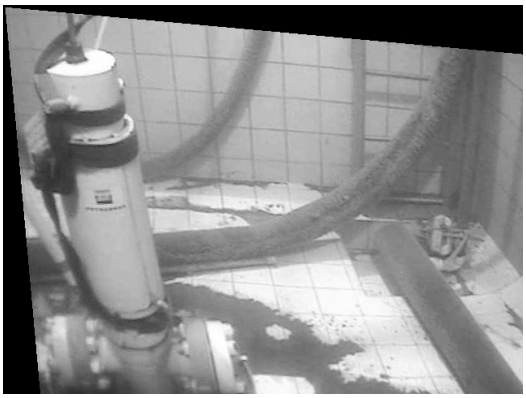

Figure 5.4: Affine transformation III.

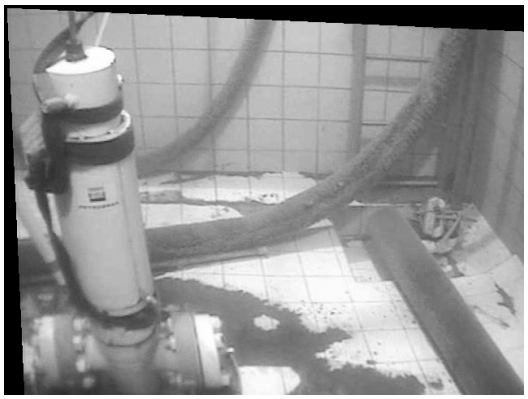

Figure 5.5: Affine transformation IV.

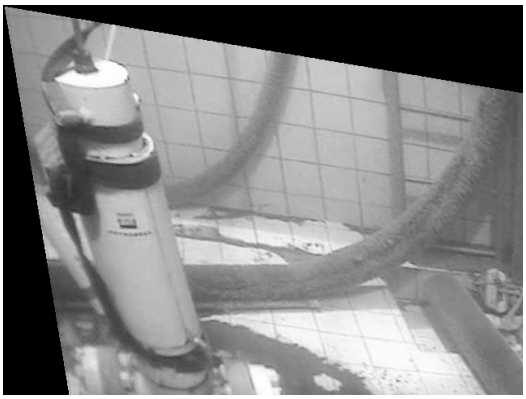

Figure 5.6: Affine transformation V.

Experiment setup

The experiments were conducted as follows:

1. the SIFT algorithm, up to the keypoint localisation stage, was applied to all six images, original plus its five affine spin-offs
2. the original image was matched against each spin-off individually (for the LSM assessment, the coordinates output by SIFT in each image were taken as the initial guesses)
3. all false matches were eliminated
4. the ground truth (GT) of the five matching procedures was obtained by computing the known affine transformations of all remaining keypoints on the original image, in each case
5. accuracy was assessed by measuring the distance, in pixels, between:
   - GT and SIFT outcome
   - GT and LSM outcome
   - SIFT and LSM outcomes

As what regards the NCC algorithms, both pixel and sub-pixel versions, tests were conducted in a similar fashion. Only, in this case, the test points were chosen manually. However, for true shift transforms are of a quite simple nature, in all cases the outcomes matched the ground truth exactly.

Results

All five transformations were joined into single tables and charts, while the three distance measures - x-y coordinates and euclidean - were kept separate.

The tables and charts referring to the fifth item listed in the experiment setup are presented consecutively and followed by a comparative discussion over the results.

	x	y	*euc.*
mean	0.34	0.32	0.51
stddev	0.42	0.51	0.63
min	0	0	0
max	4.55	6.81	8.04

Table 5.1: Difference in pixels between the SIFT outcome and the ground truth.

	x	y	*euc.*
mean	0.67	0.82	1.17
stddev	0.80	0.99	1.16
min	0.01	0.01	0.06
max	4.49	7.09	8.39

Table 5.2: Difference in pixels between the LSM outcome and the ground truth.

	x	y	*euc.*
mean	0.53	0.60	0.91
stddev	0.49	0.57	0.61
min	0.01	0	0.1
max	2.47	2.76	2.76

Table 5.3: Difference in pixels between the SIFT and the LSM outcomes.

Manually eliminating false matches, whose coordinates in the transformed image are not far from the correct location, can be rather difficult for some of the affine transformations. This may explain the higher values observed in the charts and tables that refer to the comparisons with the ground truth(Tables 5.1 and 5.2 and Figures 5.7 and 5.8). Especially considering the much lower values observed when the two algorithms were compared with one

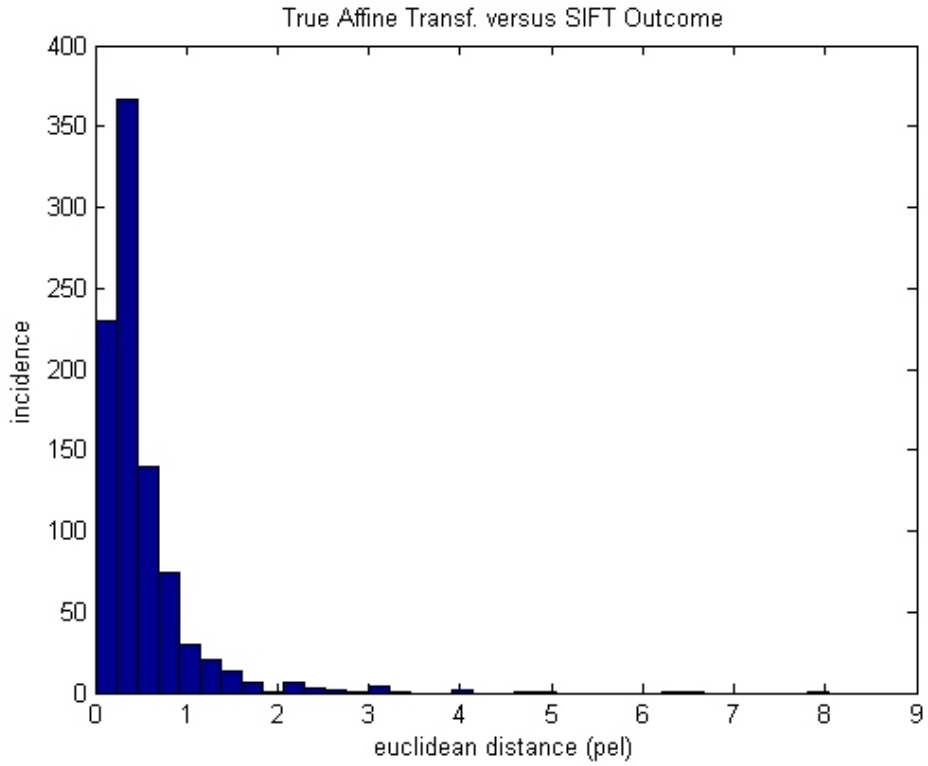

Figure 5.7: Euclidean distance between the SIFT outcome and the ground truth.

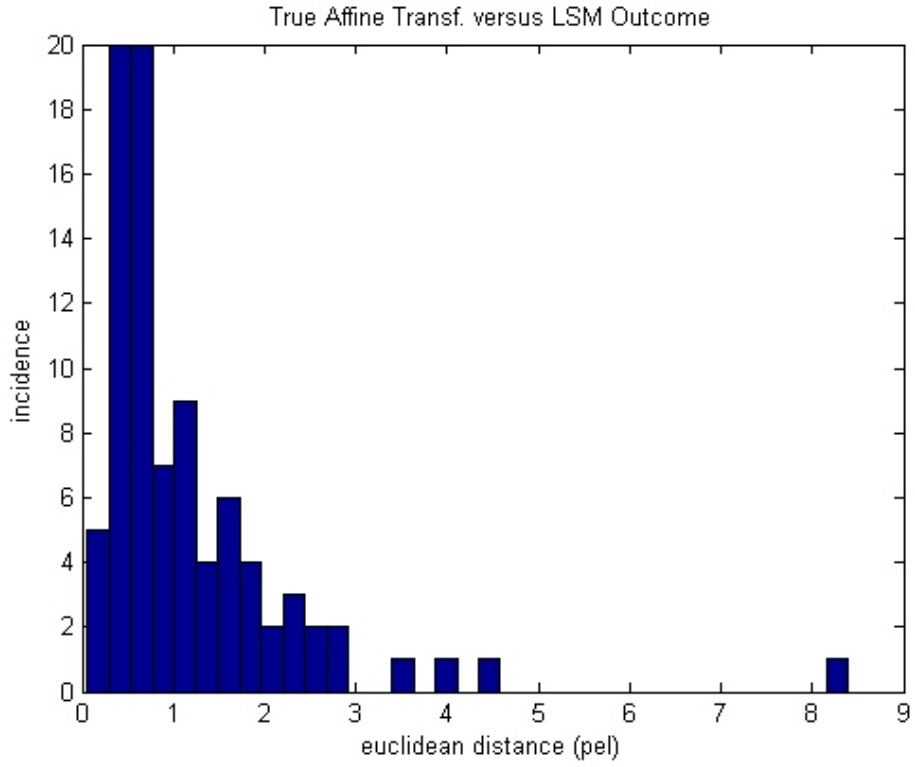

Figure 5.8: Euclidean distance between the LSM outcome and the ground truth.

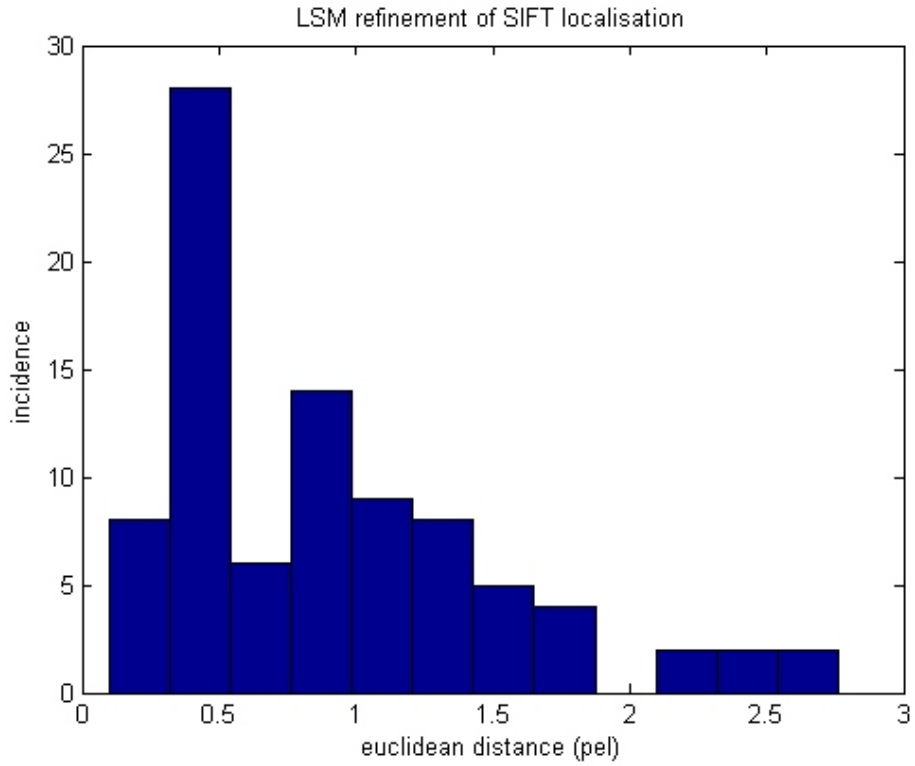

Figure 5.9: Euclidean distance between SIFT and LSM outcomes.

another (Table 5.3 and Figure 5.9). These lower values, along with the quite similar patterns of the histogram distributions seen in Figures 5.7 and 5.8, point that both procedures indeed output equivalent results.

Moreover, the procedure adopted for the ground truth computation is subjected to truncation and therefore not 100% accurate, thus adding residual error to the resulting coordinates, as indicate the high standard deviation levels. The average values, however, attest the robustness of both procedures and the disparity between their outcome and the ground truth can indeed be attributed a great deal to the residues.

5.1.2
Hit rate and processing time

The results presented herein are summarised. Complete records are available in the appendix.

This section addresses a comparative study of the SIFT and the LSM procedures, regarding their ability to match keypoints in two consecutive frames of a video sequence. Performance measures assessed were hit rate, rather than the accurate point location, and processing time.

The hit rate percentage is defined as

$$hit\ rate = \left(1 - \frac{\#\ false\ matches}{\#\ total\ matches}\right) \cdot 100 \tag{5-2}$$

Object tracking in real-time video sequences can be translated into repetitively matching a set of its keypoints in two consecutive frames throughout time. Thus, if an algorithm can not only match a reasonable number of keypoints in two views of an object - keeping a low incidence of false positives - but also run fast enough, so that not many frames are left unprocessed, then it is fit for object detection. The actual pose detection from such keypoint record is a simple straightforward procedure, provided the camera parameters and the correspondent coordinates in the model, relative to a given World system, are both known. In this case, the total of matched keypoints required is also very low (about 3 should be enough), although a greater number is desirable for robustness, as it tends to rule out the effects of small location errors or even eventual surviving false positives.

Database

The database used in this study was a 401-frame uncompressed video sequence, with a frame rate of 40.00 fps, shot underwater. Another piece was saved for the quality assessment of the method itself (section 5.2.1's "Pool" sequence).

Experiment setup

The experiment was designed as follows:

1. taking 2 consecutive frames at the time, the complete SIFT algorithm was applied to find keypoint matches between the images, resulting in 400 pairs
2. total of matches and processing time of each pair were recorded
3. the LSM algorithm was then applied, using the SIFT outcome as its template (first image) and initial guess (second image) coordinates
4. the same performance measures were recorded
5. finally, each one of the 800 (400 from SIFT and 400 from LSM) pairs was individually examined, match-by-match, to manually count all false positives, whereby the criteria were strictly visual

Filtering false matches in two consecutive frames of a video sequence is a much easier task, in comparison to the case reported in section 5.1.1, since they are very similar to one another. Thus, the false positives elimination is most likely to have been 100% accurate.

Results

Complete records of the results presented in this section are available in the appendix (chapter C).

The tests conducted show the ability of the LSM procedure in filtering false matches from the SIFT outcome. Overall, the hit rate improvement, after running LSM, was quite substantial.

Tables 5.4[2] and 5.5 present the overall performance of the SIFT and LSM algorithms, respectively. Although SIFT had an already low average of less than two mismatches per frame, the number of false matches after running LSM is nearly 10 times smaller and its worst hit rate score is just below 99%. Comparing only the worst absolute behaviours, SIFT output at most 6 false positives against LSM's 2. There was an increase of 245 frames with only correct matches (over 60%) and in only 7.5% of cases the hit rate dropped after applying LSM, even so always below 0.5% (Tables 5.8-(c) and 5.7). However, its somewhat instable nature caused a considerable reduction (in average almost 27%) of the original SIFT matches (see Table 5.6), meaning that the model must contain an either long or, preferably, stable list of points. For instance, model points near homogeneous areas should be avoided.

	mean	*stddev*	*max*	*min*	*total*
# matches	561.45	186.16	1073	99	224581
# false matches	1.49	1.26	6	0	597
hit rate	99.68	0.34	100	96.71	X
time	0.82	0.17	1.84	0.23	X

Table 5.4: Overall performance of SIFT.

	mean	*stddev*	*max*	*min*	*total*
# matches	410.42	197.50	903	9	164170
# false matches	0.16	0.39	2	0	64
hit rate	99.94	0.17	100	98.94	X
time	0.74	0.22	1.44	0.14	X

Table 5.5: Overall performance of LSM.

[2]The symbol "#", seen in Tables 5.4 to 5.12, denotes quantity.

	mean	*stddev*	*max*	*min*	*total*
# matches	-151.02	47.24	-70	-332	-60411
# false matches	-1.33	0.39	0	-6	-533
hit rate	0.26	0.17	3.29	-0.48	X

Table 5.6: Comparative table of SIFT and LSM overall performances; negative values denote a decrease from the former to the latter.

highest drop in false matches	6
highest decrease of hit rate	0.48
highest increase of hit rate	3.29

Table 5.7: Comparative table of SIFT and LSM overall performances: absolute values.

Tables 5.8-(a..c) to 5.12 provide further information on the ability of LSM to improve a SIFT outcome. They list the outcome of both algorithms in terms of mean and standard deviation, as well as direct comparison of their performance regarding the total of matches. These were built by assembling pairs of consecutive frames into groups, whose labels are depicted below. The hit rate metric was also used as a comparative measure. Tables 5.8 to 5.12, along with Table 5.13, provide the ground for the discussion presented in the remainder of section 5.1.2.

***IP*:** template-matching image pairs

***PM*:** pairs with no false matches

***HR*:** hit rate

(a) SIFT

# IP	400
# PM	95

(b) LSM

# IP	400
# PM	340

(c) LSM to SIFT

# IP	400
# increase in PM	245
# decrease in HR	30
# increase HR	275
# same HR	95

Table 5.8: Comparing SIFT and LSM performances: total of pairs.

	# PM
# matches	618.83
time	0.82

Table 5.9: SIFT pairs: comparison by mean.

	# PM
# matches	197.47
time	0.20

Table 5.10: SIFT pairs: comparison by standard deviation.

	# PM
# matches	417.94
time	0.75

Table 5.11: LSM pairs: comparison by mean.

	# PM
# matches	196.70
time	0.23

Table 5.12: LSM pairs: comparison by standard deviation.

According to Tables 5.9 and 5.11, both procedures, when considering only frames with no false positives, yielded an average number of matches even (a little) higher than the one obtained with respect to all frames (the standard deviation values also remained about the same, as show Tables 5.10 and 5.12, and are much lower than the respective mean records). This helps speaking for the robustness of the matching algorithms, since they are capable of performing with 100% accuracy, even when processing a high number of conjugate pairs.

Processing time was much longer than that obtained when embedding both algorithms to the object tracking method, but so was the list of keypoints found, which are in a considerably smaller number in practical object tracking applications. Keypoints were located in the entire images to increase the number of samples for robustness purposes. Additionally, this also allowed a more comprehensive analysis regarding the investigation of feature stability. In fact, all false matches observed in this video sequence involved points located in a restrained area of the scene. This strongly suggests that post-processing the object model created for the tracking to look for such "instability" areas may contribute a great deal to the improvement of the method's performance. Moreover, a considerably higher contrast threshold and a shorter pyramid were used, severely reducing the number of keypoints.

In addition to the aforementioned performance measures, another assessment of the keypoint location accuracy is presented next. Naturally, the ground truth, as defined in section 5.1.1, could not be computed. However,

the results provide further information about the similarity between the SIFT and LSM outcomes. This calculations were done automatically, i.e., without filtering the false positives, which nevertheless, due to the high levels observed for the hit rate, are unlikely to have contaminated the results to a degree that would invalidate the conclusions drawn. Tables 5.3 and 5.13 are analogous.

	x	y	*euc.*
mean	0.23	0.19	0.34
stddev	0.37	0.34	0.48

Table 5.13: Difference in pixels between the SIFT and the LSM outcomes: total of 164170 samples.

The *maximal* and *minimal* assessments were suppressed from Table 5.13, because of the presence of false matches. The average difference stays around 1/5 of a pixel in each direction, which may be solely attributed to truncation errors, as further suggest the much higher levels of standard deviation. Since replication of error patterns are extremely rare, especially considering that both algorithms are of complete different natures, these results offer yet another indication of the accuracy of pixel localisation achieved by adopting such procedures.

5.2 Performance evaluation of the object tracking method

This section presents the results of applying the method for object tracking described in chapter 4 to different video sequences; each having its own idiosyncrasies, regarding compression, environment and degree of movement. The method's performance quality assessment was chiefly visual-aided and, therefore, of a more qualitative nature, rather than quantitative. Numerical data were nevertheless produced and are made available, as well.

Discussions are followed by a few guidelines on which variant is more likely to yield the best results, given the characteristics of a sequence.

5.2.1 Database

Four sequences, from three different cameras, were used for evaluating the method's performance. They are:

Surgical Procedure: JPEG-compressed sequence, of an extremely high degree of movement. Environment and object blend; the human body being the scene, and the prostate the specific target to track. The

geometrical transformations the object undergoes throughout this video may be remarkably difficult to model, even between consecutive frames, due to the rubbery consistency of animal tissues.

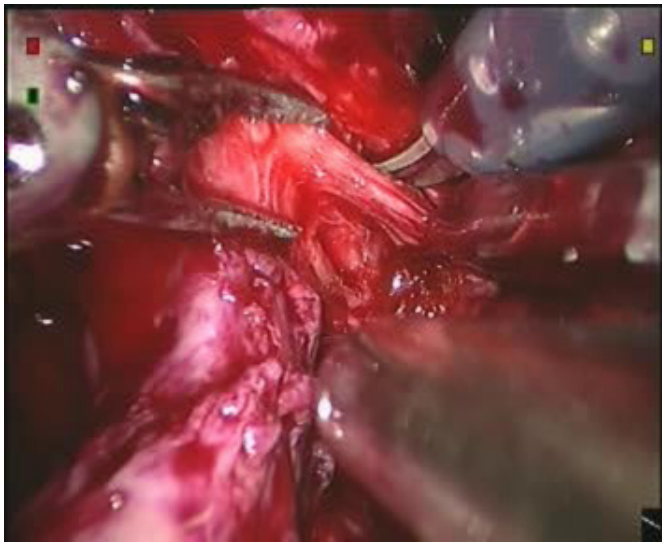

Figure 5.10: Surgical procedure: frame sample.

Fast Pre-Amp.: High movement JPEG-compressed sequence, specially shot for testing the method's behaviour. The interest object, an ancient pre-amplifier unit, placed in a dry environment, is the only one appearing in the scene. World coordinates of the keypoints are easily measurable, due to the nature of the interest object. However, the background is still not perfectly neutral and illumination changes and incidence cause it to incur perceptible alterations in contrast.

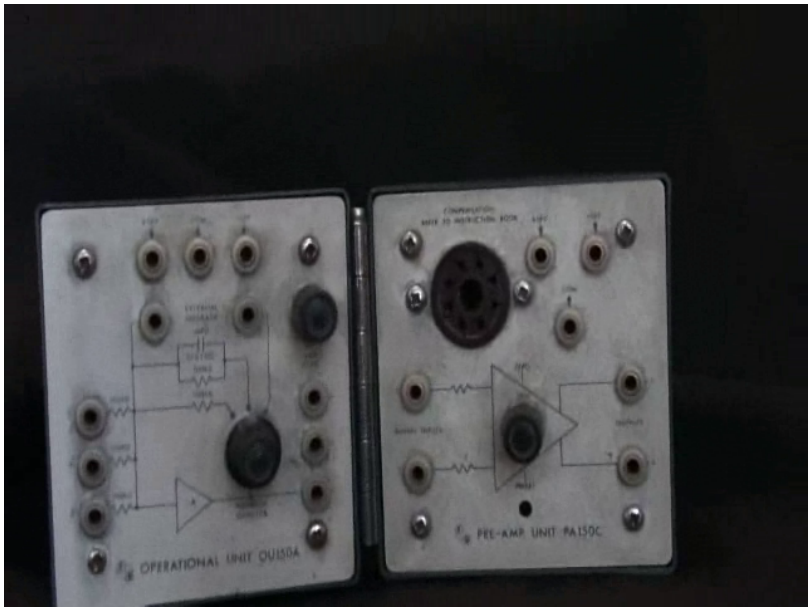

Figure 5.11: Fast Pre-Amp.: frame sample.

Slow Pre-Amp.: Very similar to the previous one, only differing as regards the degree of movement; low, is this case. The former observations made on illumination conditions apply here, as well.

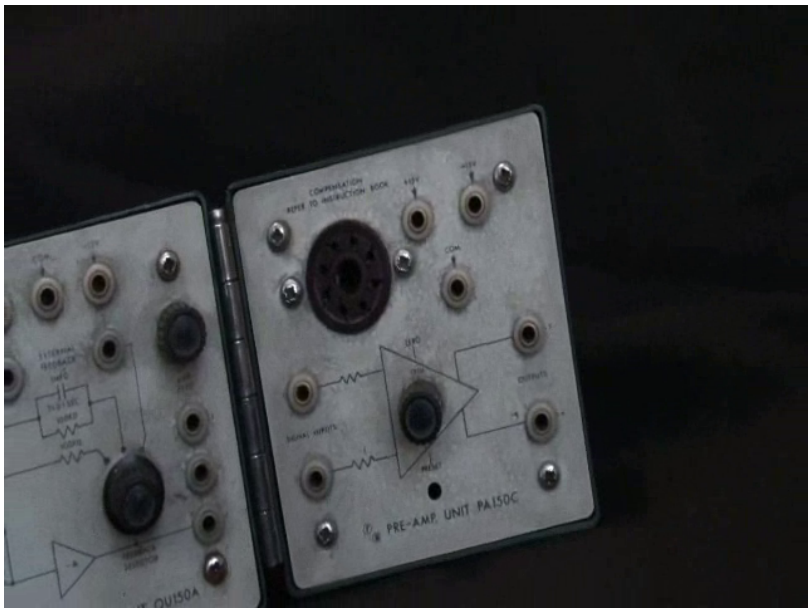

Figure 5.12: Slow Pre-Amp.: frame sample.

Pool: Low movement uncompressed sequence, shot underwater in a (rather dirty) pool. The object is a sort of tank, located in a fairly more interesting environment, i.e., full of potential keypoint candidates, to be recognised by the SIFT algorithm.

Figure 5.13: Pool: frame sample.

5.2.2 Experiment setup

The degree of movement was the main criterion for dividing the sequences among the two method variants. Applying the SLN and SLS variants to the surgical procedure and pool sequences, respectively, was a considerably straightforward choice. The geometrical changes of the former go completely against the SLN fundamentals, quite the opposite of the latter, a low movement sequence, with no compression, meaning that the relations between neighbouring pixels remained intact, which provides a more reliable keypoint descriptor for the two last procedures.

Matter resolved, two extra videos were made for testing:

- SLN's performance in cases where (JPEG) compression is present, and
- SLS's behaviour in less extreme environmental conditions.

However, the incidence of keypoints is significantly low, due to the object's own nature (i.e., its appearance that is remarkably uniform), which adds difficulty to the tracking conditions.

Thus, all four sequences present some characteristic that affects the level of difficulty for testing an object tracking procedure. Indeed, they are intended to assess the robustness of the method, ultimately speaking for the likelihood of its success in real applications.

The four sequences were processed by their respective variants, whose performance was assessed from:

- charts of the total of keypoints throughout time that show the cyclical routine of continuous drops - as frames are processed - and sudden increases - whenever an updated is released;
- histograms of the keypoints in each frame, which offer the complementary view of how many frames hold a particular number of features, and
- records of the actual keypoint locations in all processed frames, merged into an output movie, in which each one eventually lost was replaced by the last processed frame.

5.2.3 Results

This section reports the results of the performance measures previously described. A further analysis on the behaviour of the SLN variant, when applied to the "Pool" sequence, regarding the update process, is also presented.

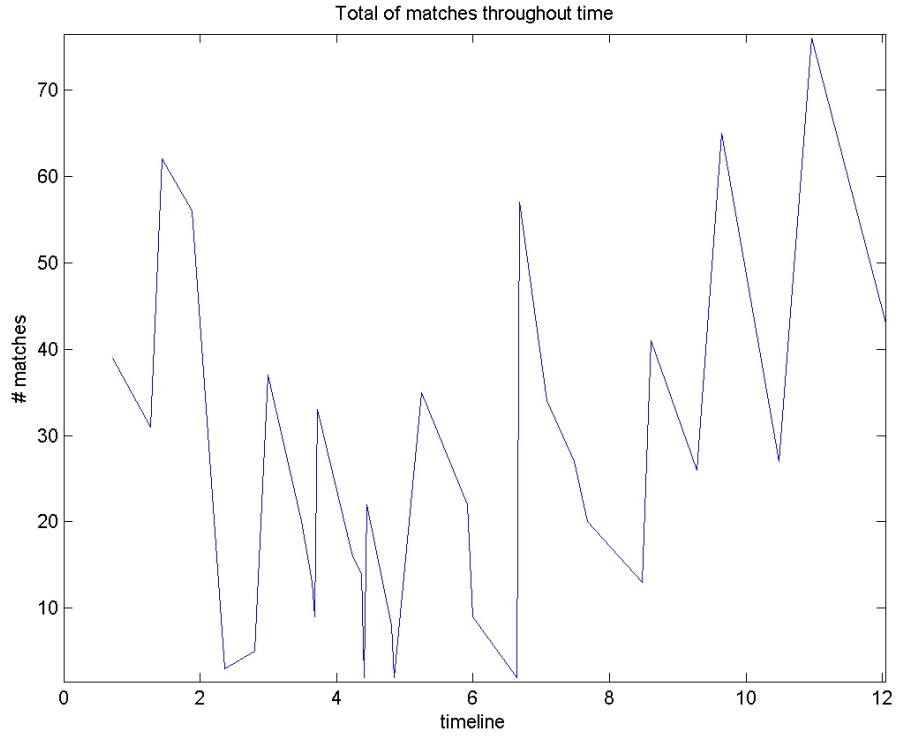

Figure 5.14: Surgical procedure: Incidence of matched keypoints throughout time (SLS variant).

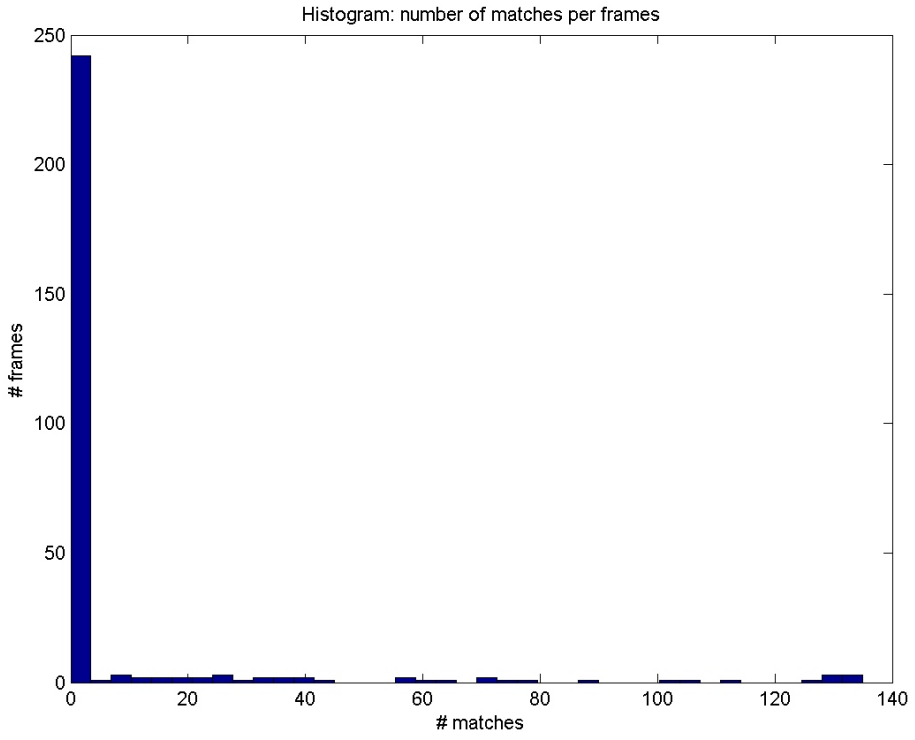

Figure 5.15: Surgical procedure: Matches per frames histogram (SLS variant).

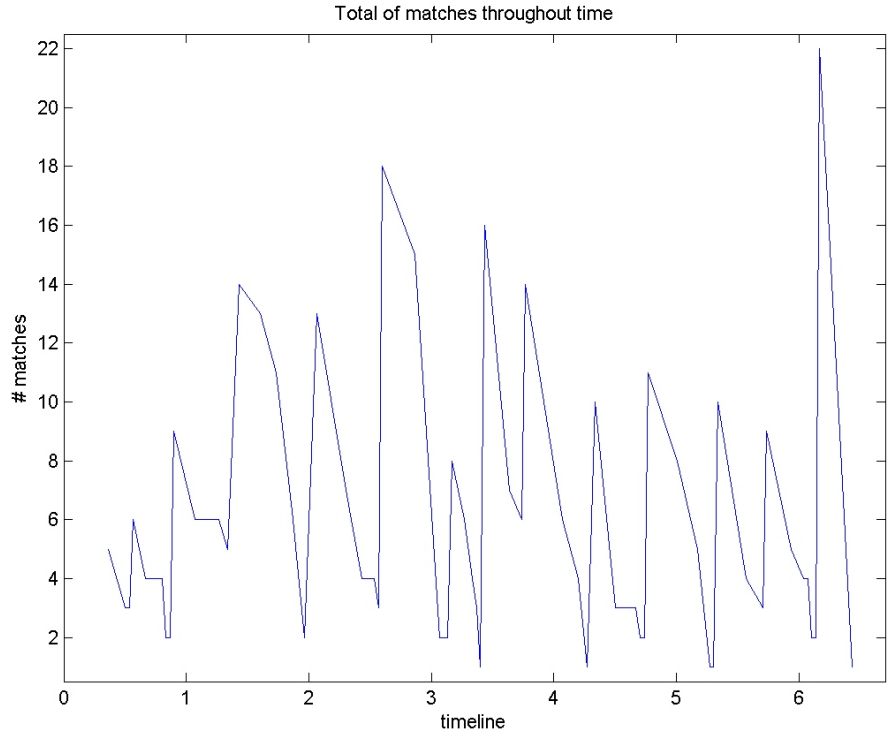

Figure 5.16: Fast Pre-amp: Incidence of matched keypoints throughout time (SLS variant).

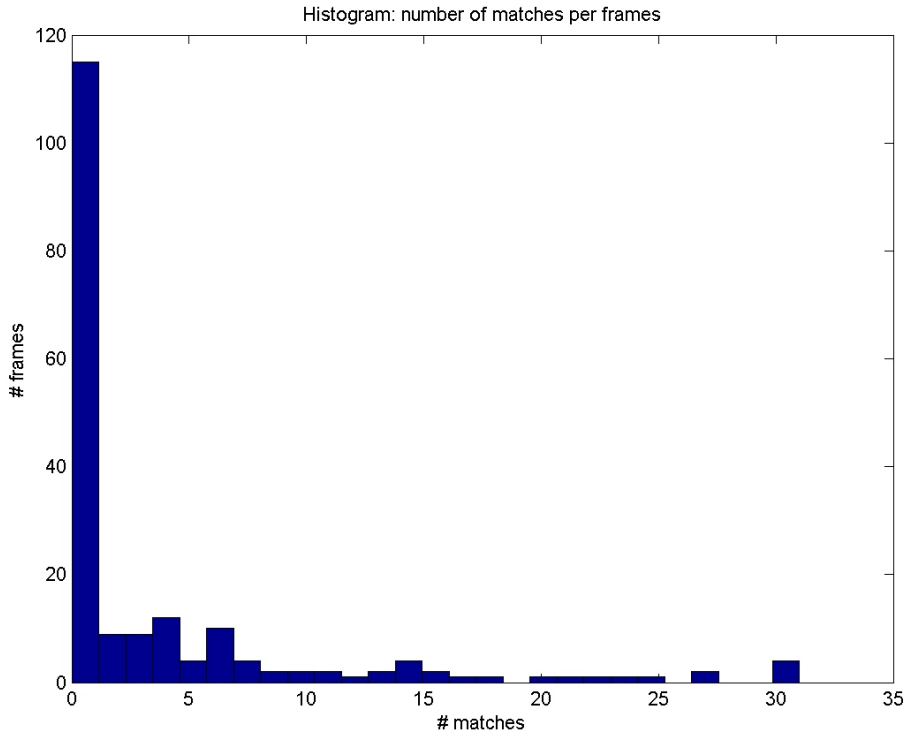

Figure 5.17: Fast Pre-amp: Matches per frames histogram (SLS variant).

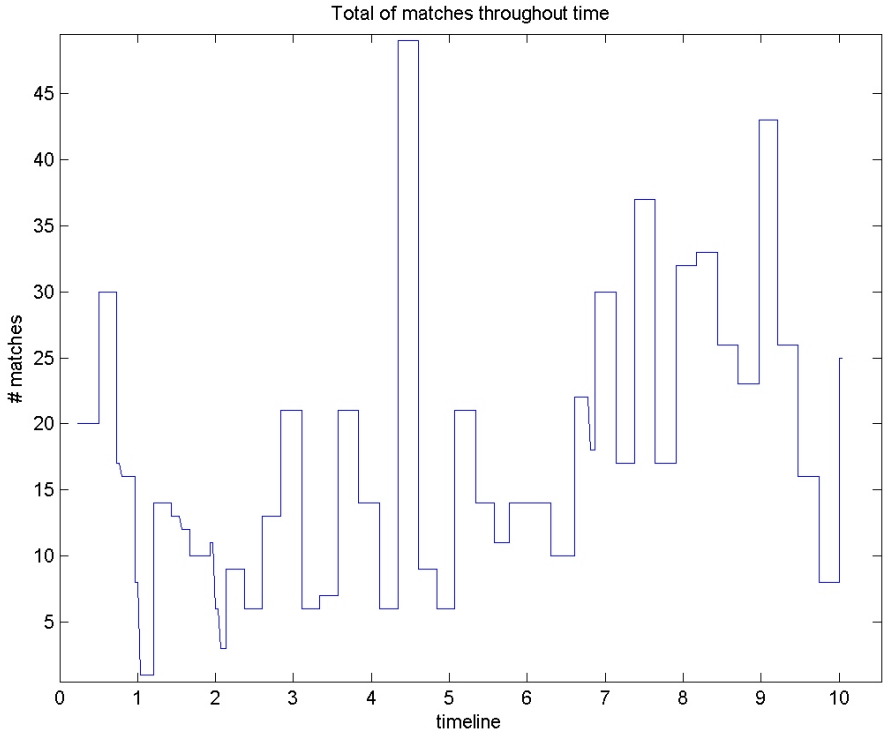

Figure 5.18: Slow Pre-amp: Incidence of matched keypoints throughout time (SLN variant).

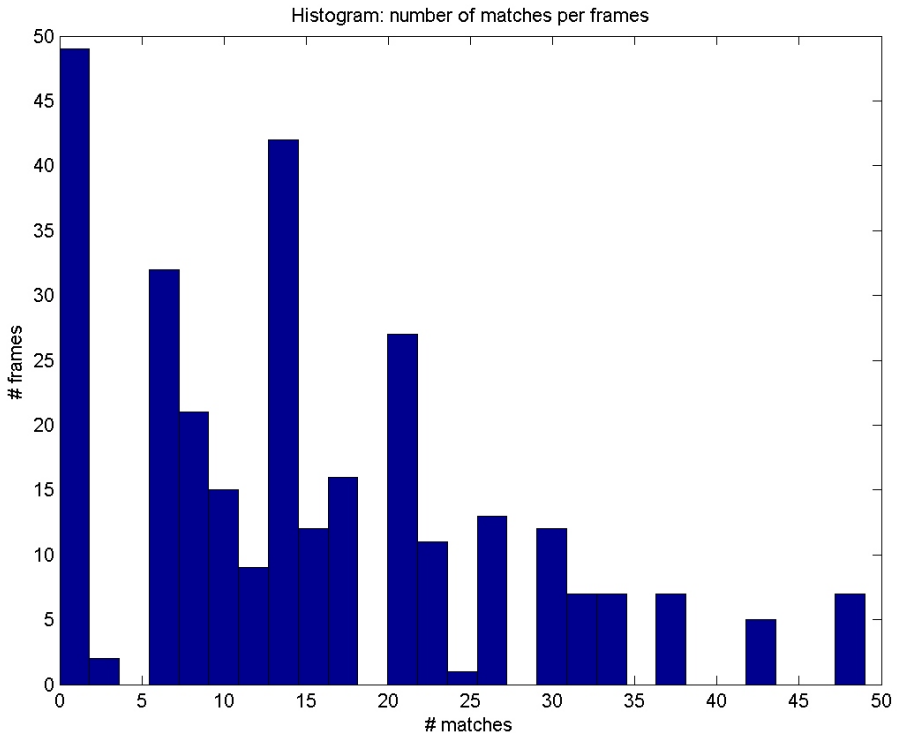

Figure 5.19: Slow Pre-amp: Matches per frames histogram (SLN variant).

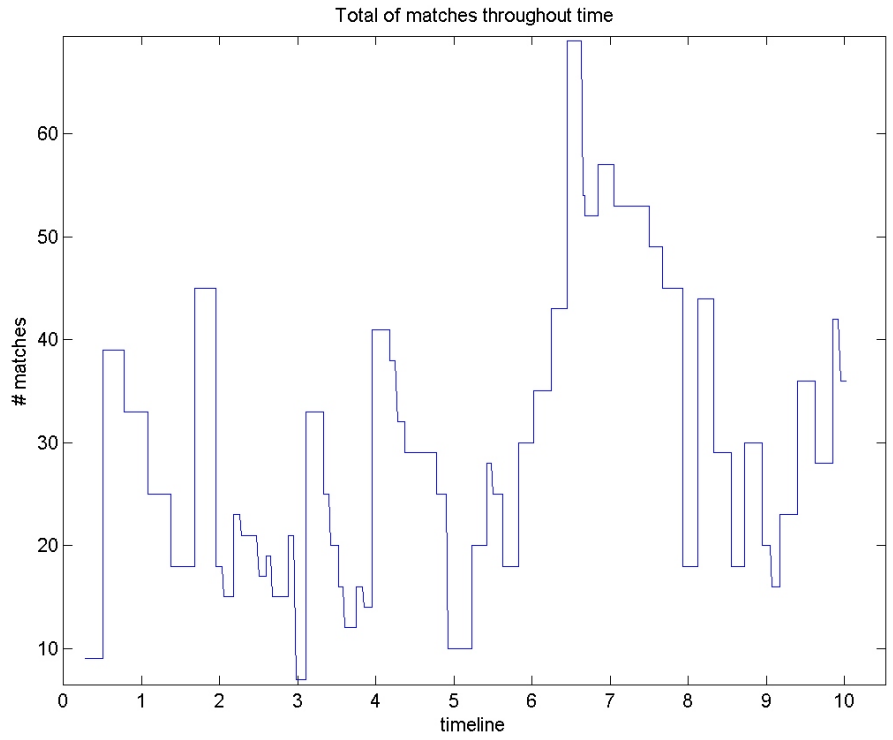

Figure 5.20: Pool: Incidence of matched keypoints throughout time (SLN variant).

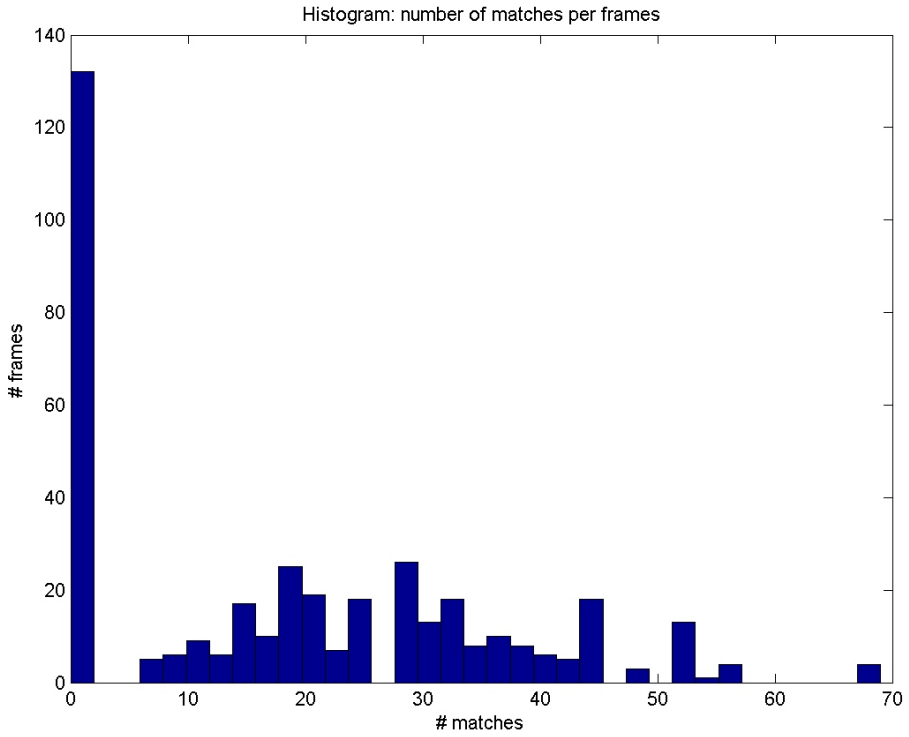

Figure 5.21: Pool: Matches per frames histogram (SLN variant).

All charts and histograms are depicted in Figures 5.14 to 5.21. Time elapsed during the starting step of the object tracking (the "START" block on Figure 4.2) was excluded from these records. The remaining unprocessed frames, however, contribute with a 0 entry in the histograms.

The SLN and SLS variants are instantly recognisable from their general response depicted in the charts (Figures 5.14, 5.16, 5.18 and 5.20). They show that SLN is able to track more keypoints, missing less frames, as indicate the series of short steady periods, during which the total of keypoints remains the same. The update and tracking processes are more easily identified on the SLS charts, for subsequent scenes - further apart in time - hardly ever have the same number of keypoints. The overall duration of a tracking chain is bound by the time needed for the next update release. However, in SLN's tracking chain ("thread #1" on Figure 4.2), each link is faster, which is crucial to the effectiveness of the area-based keypoint descriptors in this case and hence accounts for the steadier behaviour of this variant, in terms of match losses.

Comparing both SLS sequences, the lower degree movement of "Fast Pre-Amp", in comparison to "Surgical Procedure", cause the variant's behaviour to somewhat resemble the standard SLN's, despite of the average total of matched keypoints being lower, for the reasons given in section 5.2.2.

Regarding the SLN sequences, the appearance of the interest object in "Slow Pre-Amp", which hampers the matching process, helps to explain the rather more erratic behaviour of the method, depicted by the high frequency of fluctuations on Figure 5.18, when compared to ones on Figure 5.20.

The histograms show that the total of matched keypoints varies significantly among the frames, especially in the SLN sequences, which may be explained by the fact that the tracking procedure actually consists of several independent processes, started by an update event that has no memory of the past ones and whose relation to them is solely the resemblance - or lack of it - borne by the scenes such frames depict.

Different keypoints may have the same x-y coordinates, but differ in the σ dimension. This explains why the number of matches is greater than the one perceived on the images.

As visual aid for the performance evaluation, mosaics showing the different localisations of some keypoints throughout a few frames, i.e., describing how the object moves along the scene, are presented next. A change of colour denote that an update has taken place, and, in this case, kept the feature in the tracking list. Table 5.14 lists the time elapsed between the first and last frames in Figures 5.22 to 5.26, where keypoint locations were rounded for the display.

	time (s)
Surgical Procedure 1	1.40
Surgical Procedure 2	3.48
Fast Pre-Amp	1.50
Slow Pre-Amp	0.53
Pool	1.23

Table 5.14: Time elapsed during the processes of tracking the keypoints in the examples given.

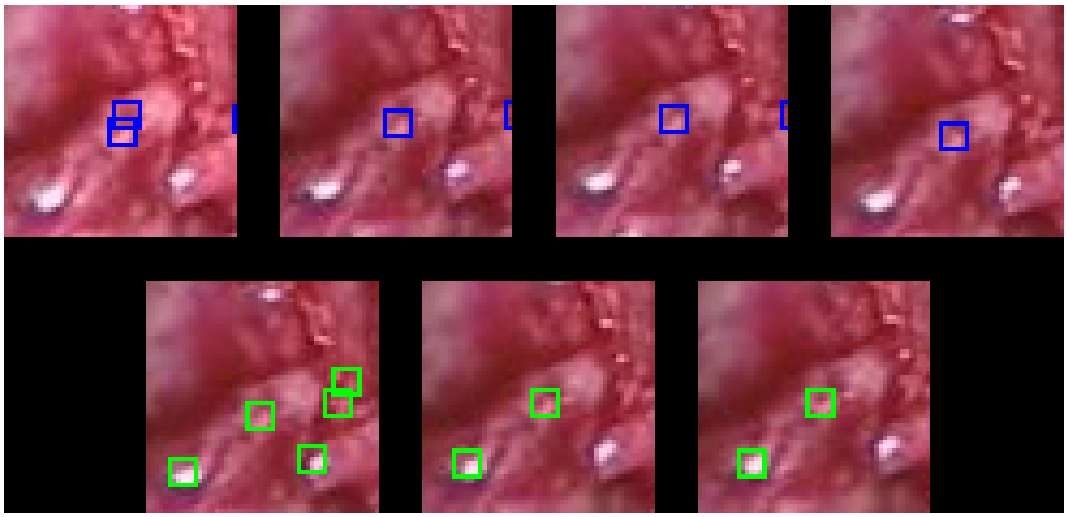

Figure 5.22: Surgical Procedure: a keypoint moving throughout time (row-wise, from top left to bottom right).

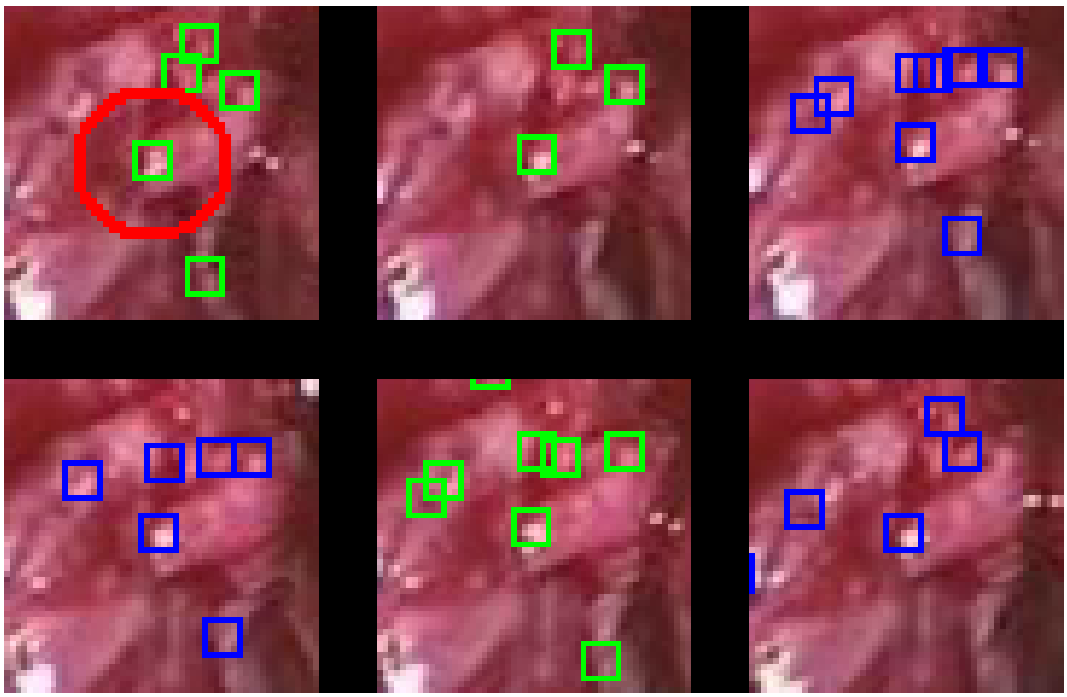

Figure 5.23: Surgical Procedure: another keypoint moving throughout time (row-wise, from top left to bottom right).

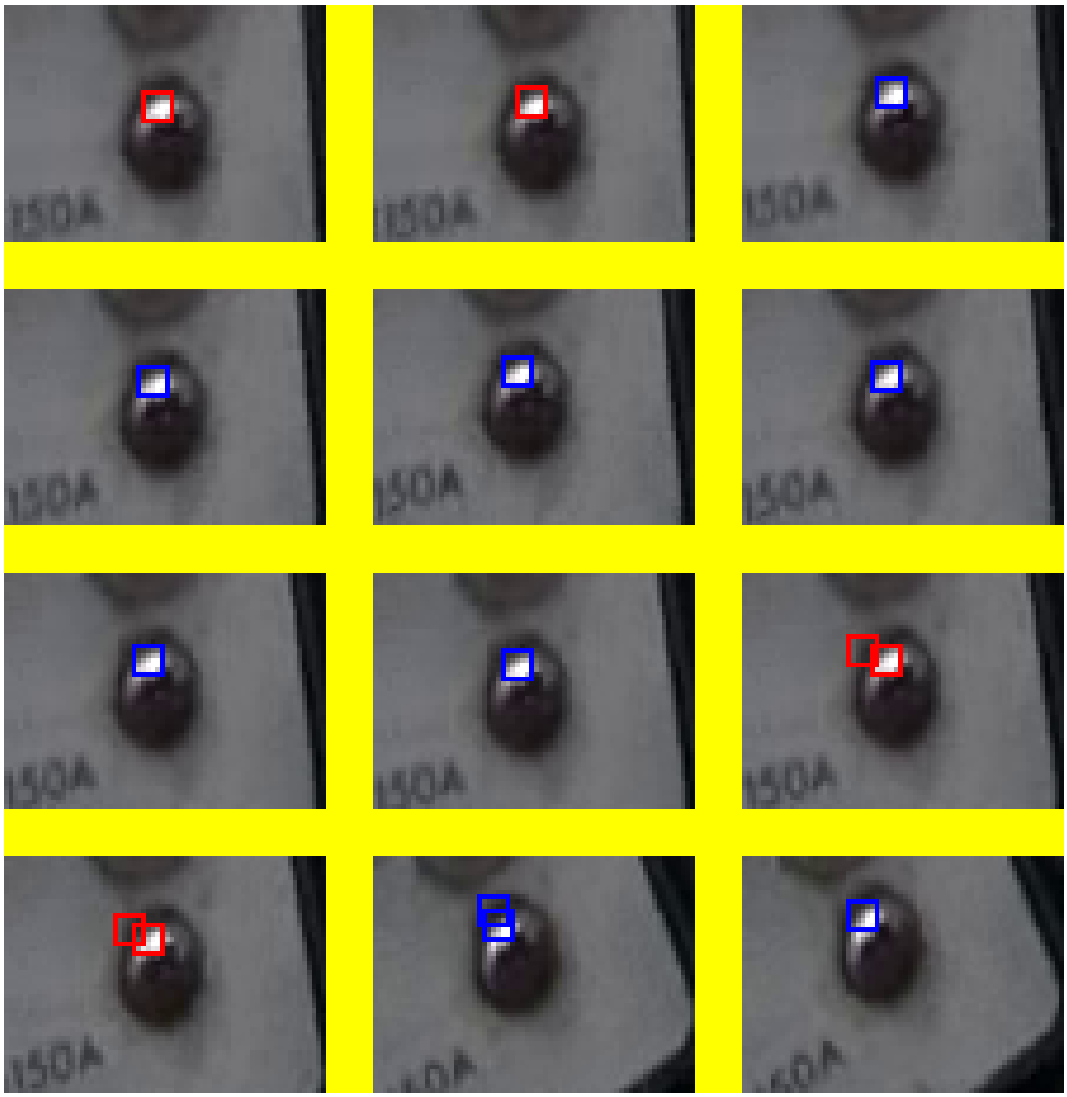

Figure 5.24: Fast Pre-Amp: a keypoint moving throughout time (row-wise, from top left to bottom right).

Exception made to the two "Surgical Procedure" figures, consecutive frames on the mosaics do not necessarily correspond to consecutive processed frames. They followed, nevertheless, the time line. The object's appearance is likely to be the main reason of the more dynamic keypoint turn out at updates in the "Surgical Procedure" sequence.

Conversely, for a sequence like "Pools", where the scene changing rates are much lower and the interest object is rigid, consecutive updated collection releases are expected to be quite similar to one another - with respect to the distribution of keypoints - since the SIFT response is deterministic. However, during some intervals, the tracking list changed quite remarkably after an update event. As said, due to its high stability standards, LSM eliminates a fair amount of the matched point entries in the list delivered by SIFT. In addition, varying environmental conditions cause changes from one frame to

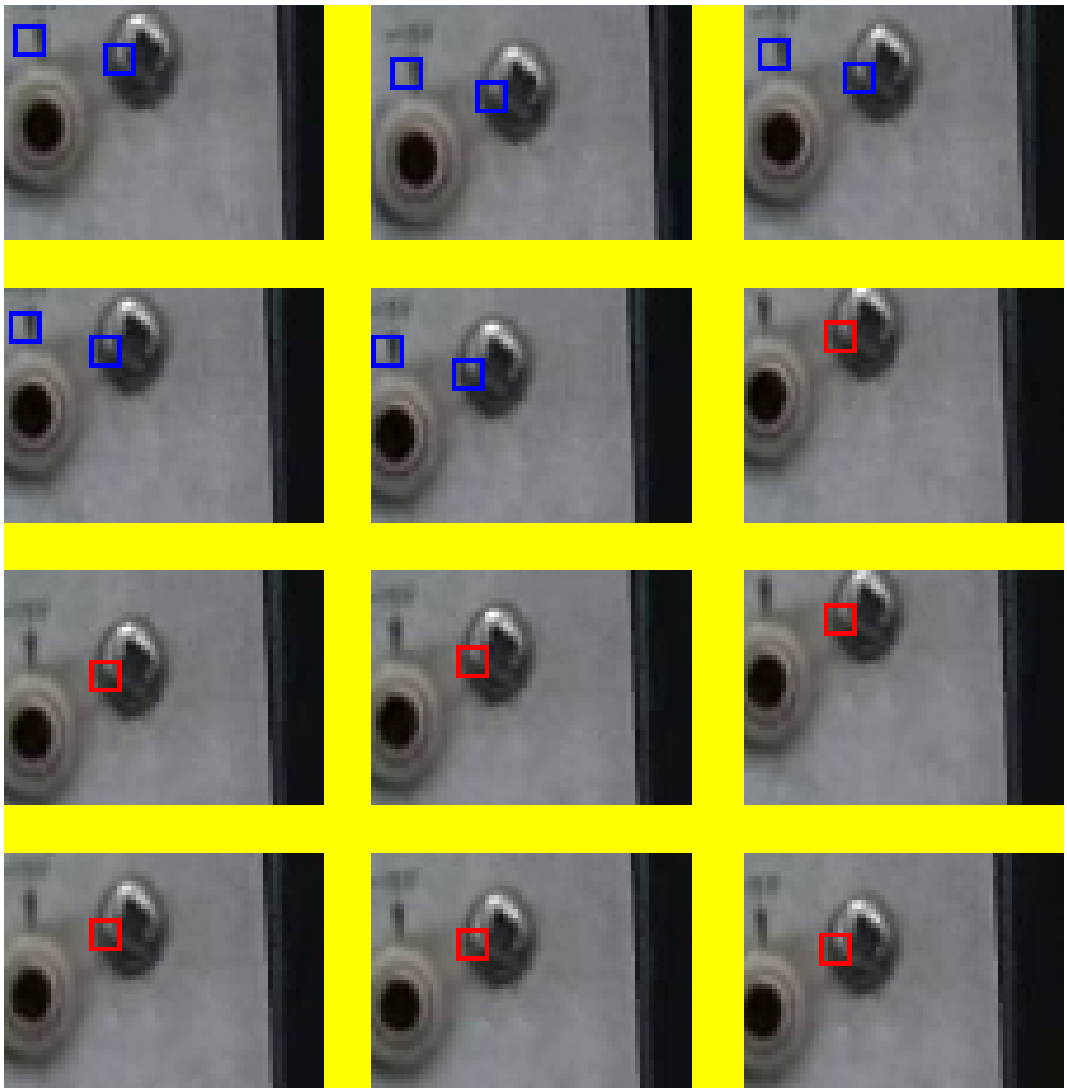

Figure 5.25: Slow Pre-Amp: a keypoint moving throughout time (row-wise, from top left to bottom right).

another, even if the scene remained stationary. This is particularly significant in a sequence shot underwater, since the medium is subjected to refraction and movement. Thus, the keypoints that meet those stability criteria may be different at each update process, even if the original starting lists are similar.

As to verify this hypothesis, six frame triplets were analysed. Figures 5.27 and 5.28 depict, respectively, three samples representing cases, in which the matched keypoints list released by the update:

- is similar to the one before, and
- changed considerably in comparison with the previous one.

Each triplet contains the SIFT outcomes - up to the keypoint localisation stage - in the following frames:

1. frame used for a given update event (left patch; keypoints in red)

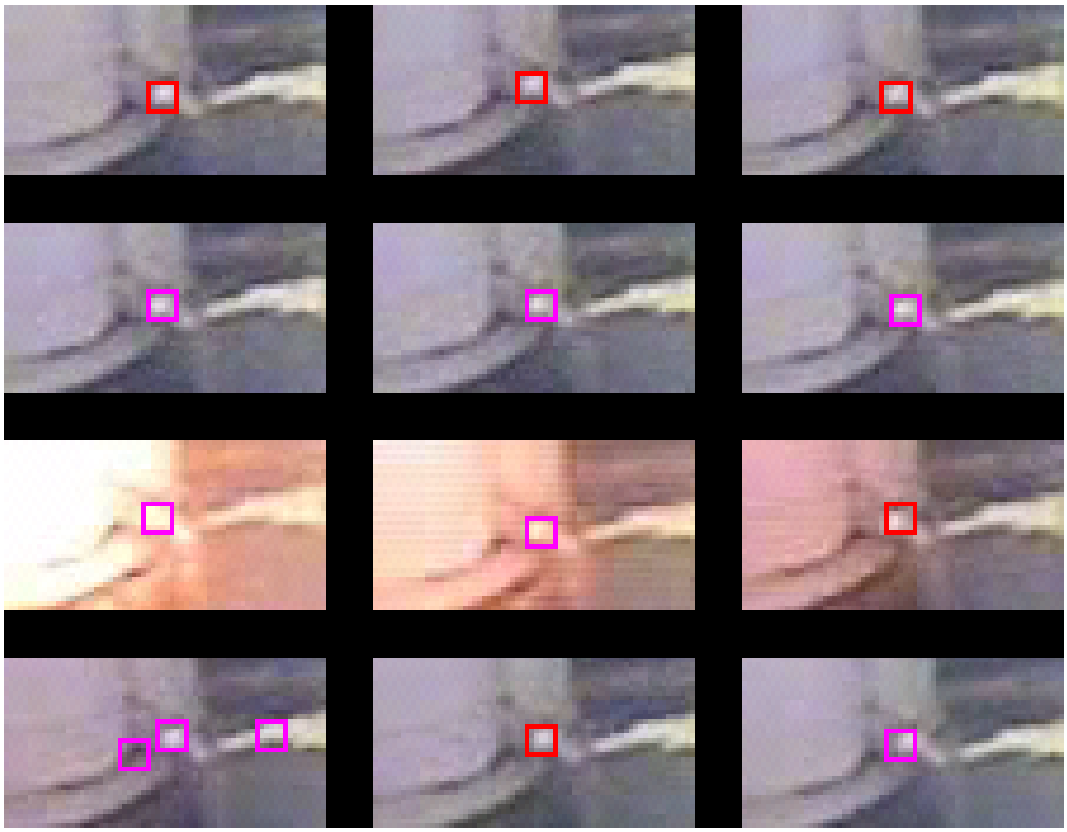

Figure 5.26: Pool: a keypoint moving throughout time (row-wise, from top left to bottom right).

2. last frame processed by the tracking thread originated by the update (central patch; keypoints in green)

3. frame used for the next update event (right patch; keypoints in blue)

All triplets yielded very similar keypoint distribution patterns - especially the last two items in each triplet, which are, in fact, closer in time - and, thus, corroborate the previous statement.

At first, in the cases of sequences processed by the SLN variant it might seem like both versions of the proposed method could actually be used interchangeably. In fact, the SLS variant would work well with all sequences. However, the former produces a greater average number of matches. Thus, the decision for one or the other is essentially a trade-off between match incidence and range of transformation mapping. There are yet other factors that contribute to the suitability of one variant over the other, a matter addressed in the next section.

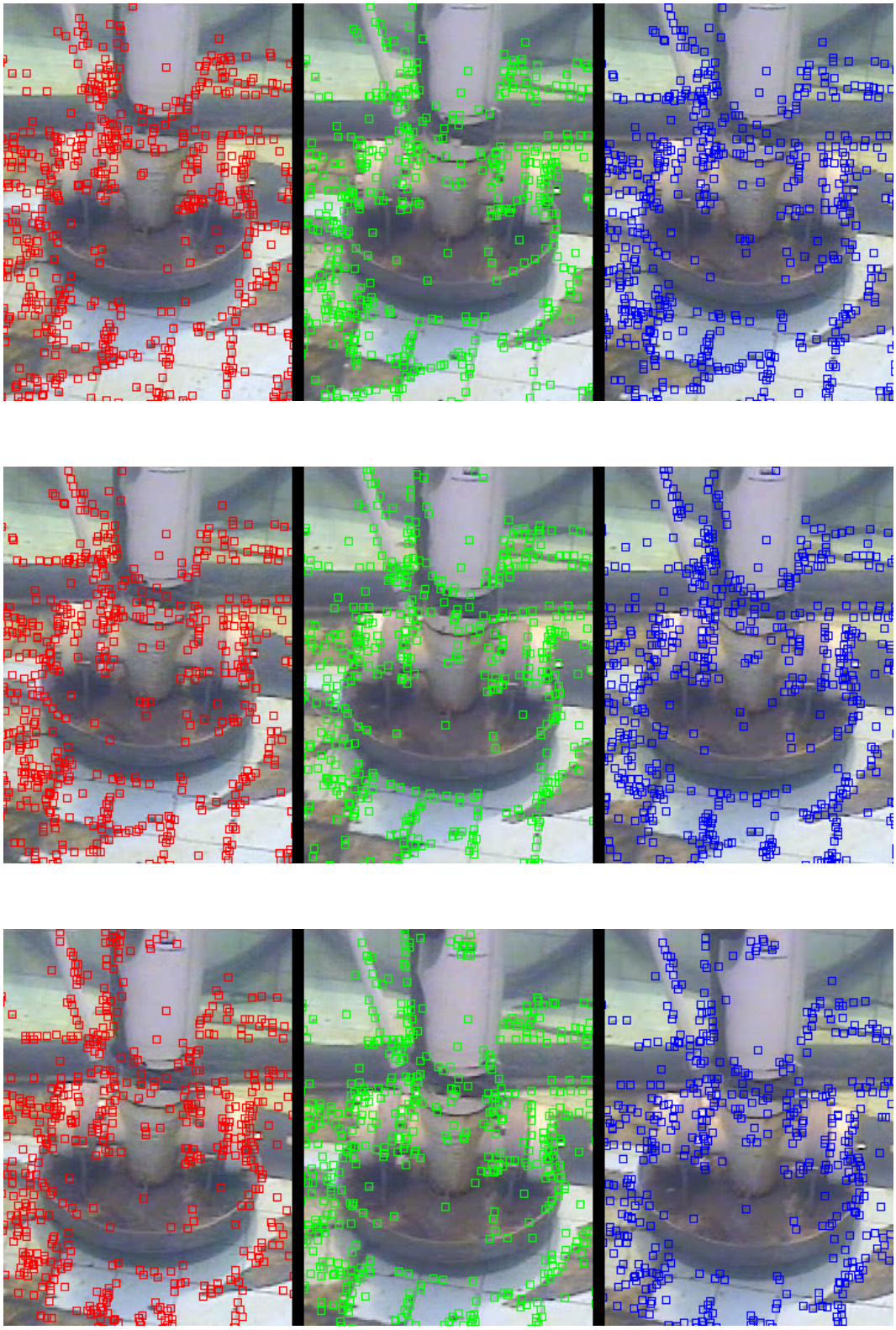

Figure 5.27: Samples that yielded similar matched keypoint lists. Left: at an update event; centre: last frame in the tracking phase; right: at the next update event.

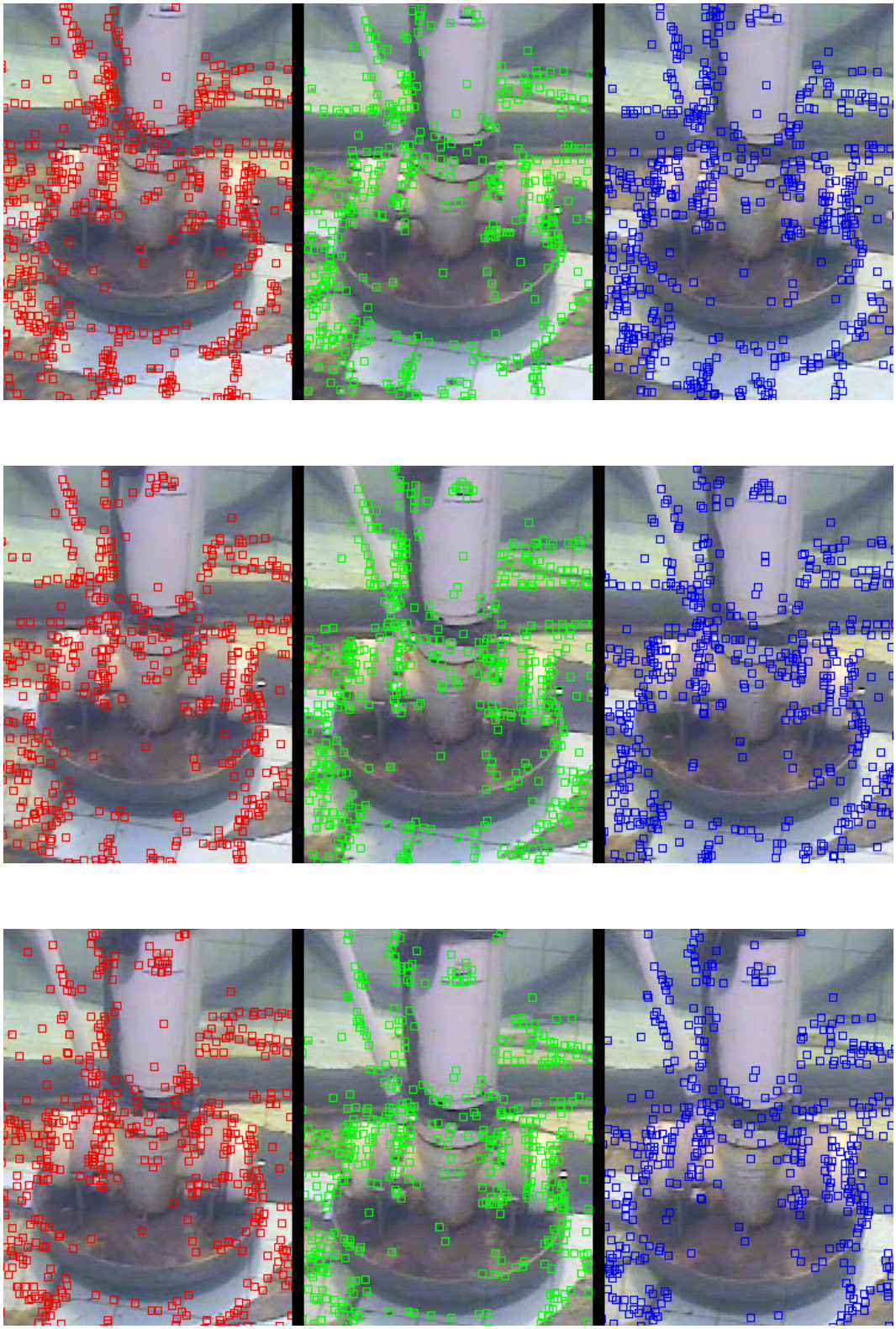

Figure 5.28: Samples that yielded significantly different matched keypoint lists. Left: at an update event; centre: last frame in the tracking phase; right: at the next update event.

5.2.4 General guidelines

In this section, the overall characteristics of the four test sequences, regarding environment, compression, nature of the interest object and degree of movement, are assembled into Table 5.15. Hopefully, it will help selecting between the two variants for a future application, given a particular set of conditions.

frame rate (fps)	25.00	29.97	29.97	40.00
compression	JPEG	JPEG	JPEG	NONE
degree of movement	EXTREM. HIGH	HIGH	LOW	LOW
environment	HUMAN BODY	DRY	DRY	UNDERWATER
interest object	DEFORMABLE	HARD	HARD	HARD
VARIANT	**SLS**	**SLS**	**SLN**	**SLN**

Table 5.15: Summary table for case-driven determination of the most suitable variant.

6
Concluding remarks

This work aimed at the development of a novel method for tracking a target object in real-time. By determining the locations of several keypoints throughout a video sequence, the object's pose can then be estimated. Exhaustive tests on point matching algorithms provided the information required for devising a method that is adaptable to different conditions of environment, the nature of the interest object, quality of the camera(s) and degree of movement. Although it was meant for self-sufficient use, the proposed algorithm may also be embedded in a control system to function as the visual-based fine tuning stage, where it would be triggered occasionally. That was, in fact, the original project.

Parallel processing automatically assures that the set of keypoints is always consistent to the current scene, thus discarding algorithms for monitoring or controlling these features' incidence. Two variants of the method were designed to increase its robustness and, thus, widen its applicability in the face of the diversity of working conditions that may be encountered. Indeed, the procedure proved to be case-driven, for the inherent qualities of some sample sequences violated the very principals, in which bed the original variant's functioning.

Future works include adding the pose detection step to a demo version. If the camera parameters and the correspondent world coordinates are known, the procedure is rather simple and only requires one camera. For analogous reasons to what exposed herein, an independent third thread should perform the pose computation, allowing the processing time of the two existing ones to remain unchanged.

Bibliography

[1] ACKERMANN, F. **Digital image correlation: performance and potential application in photogrammetry**. The Photogrammetric Record, 11(64):429–439, 1984.

[2] ARYA, S.; MOUNT, D. M. **Approximate nearest neighbor queries in fixed dimensions**. In: PROCEEDINGS OF THE FOURTH ANNUAL ACM-SIAM SYMPOSIUM ON DISCRETE ALGORITHMS (SODA'93), p. 271–280. Society for Industrial and Applied Mathematics, 1993.

[3] AYRA, S.; MOUNT, D. M.; NETANYAHU, N. S.; SILVERMAN, R. ; WU, A. Y. **An optimal algorithm for approximate nearest neighbor searching**. Journal of the ACM, 45(6):891–923, 1998.

[4] BEIS, J.; LOWE, D. G. **Shape indexing using approximate nearest-neighbour search in highdimensional spaces**. In: CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, p. 1000–1006, Puerto Rico, 1997.

[5] BURT, P. J.; ADELSON, E. **The laplacian pyramid as a compact image code**. IEEE Transactions on Communications, COM-31(4):532–540, April 1983.

[6] DANIILIDIS, K.; EKLUNDH, J.-O. **Springer Handbook of Robotics**, chapter 23, p. 543–562. Springer, Berlin Heidelberg, May 2008. 3-D Vision and Recognition.

[7] DEITEL, H. M.; DEITEL, P. J. **C++: como programar**. Bookman, Porto Alegre, 3rd edition, 2001.

[8] DUDA, R. O.; HART, P. E. **Use of the hough transformation to detect lines and curves in pictures**. Commun. ACM, 15(1):11–15, 1972.

[9] EDELMAN, S.; INTRATOR, N. ; POGGIO, T. **Complex cells and object recognition**. Preprint, http://kybele.psych.cornell.edu/ edelman/Archive/nips97.pdf, 1997.

[10] FISCHLER, M. A.; BOLLES, R. C. **Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography**. Commun. ACM, 24(6):381–395, June 1981.

[11] FORSYTH, D. A.; PONCE, J. **Computer Vision: A Modern Approach**. Prentice Hall, New Jersey, 2003.

[12] FRIEDMAN, J. H.; BENTLEY, J. L. ; FINKEL, R. A. **An algorithm for finding best matches in logarithmic expected time**. ACM Transactions on Mathematical Software, 3(3):209–226, 1977.

[13] GRUEN, A. W. **Adaptive least squares correlation: a powerful image matching technique**. South African Journal of Photogrammetry, Remote Sensing and Cartography, 14(3):175–187, April 1985.

[14] HUANG, T. S.; FAUGERAS, O. D. **Some properties of the e-matrix in two-view motion estimation**. IEEE Trans. Pattern Anal. Mach. Intell., 11(12):1310–1312, December 1989.

[15] LONGUET-HIGGINS, H. C. **A computer algorithm for reconstructing a scene from two projections**. Nature, 293:133–135, September 1981.

[16] LOWE, D. G. **Object recognition from local scale-invariant features**. In: INTERNATIONAL CONFERENCE ON COMPUTER VISION, p. 1150–1157, Corfu,Greece, 1999.

[17] LOWE, D. G. **Local feature view clustering for 3d object recognition**. In: COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, volume I, p. 682–688. IEEE, 2001.

[18] LOWE, D. G. **Distinctive image features from scale-invariant keypoints**. International Journal of Computer Vision, 60(2):91–100, January 2004.

[19] MIKOLAJCZYK, K.; SCHMID, C. **Scale and affine invariant interest point detectors**. International Journal of Computer Vision, 60(1):63–86, 2004.

[20] MIKOLAJCZYK, K.; SCHMID, C. **A performance evaluation of local descriptors**. IEEE Transactions on Pattern Analysis & Machine Intelligence, 27(10):1615–1630, 2005.

[21] MIKOLAJCZYK, K.; TUYTELAARS, T.; SCHMID, C.; ZISSERMAN, A.; MATAS, J.; SCHAFFALITZKY, F.; KADIR, T. ; VAN GOOL, L. **A comparison of affine region detectors**. International Journal of Computer Vision, 65(1/2):43–72, 2005.

[22] PRESS, W. H.; TEUKOLSKY, S. A.; VETTERLING, W. T. ; FLANNERY, B. P. **Numerical recipes : the art of scientific computing**. Cambridge University Press, New York, third edition, 2007.

[23] SCHENK, T. **Digital Photogrammetry**, volume I. TerraScience, Ohio, 1999.

[24] SE, S.; LOWE, D. ; LITTLE, J. **Vision-based mobile robot localization and mapping using scale-invariant features**. In: PROCEEDINGS OF THE IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA), p. 2051–2058, 2001.

[25] SE, S.; LOWE, D. ; LITTLE, J. **Global localization using distinctive visual features**. In: INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, p. 226–231, Lausanne, Switzerland, 2002. IROS.

[26] SHAH, S.; AGGARWAL, J. K. **Intrinsic parameter calibration procedure for a (high-distortion) fish-eye lens camera with distortion model and accuracy estimation**. Pattern Recognition, 29(11):1775–1788, 1996.

[27] TRUCCO, E.; VERRI, A. **Introductory Techniques for 3-D Computer Vision**. Prentice Hall, New Jersey, 1998.

[28] WITKIN, A. P. **Scale-space filtering**. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, p. 1019–1022, Karlsruhe, Germany, 1983.

[29] ZUYSAL, M.; FUA, P. ; LEPETIT, V. **Fast keypoint recognition in ten lines of code**. In: IN PROC. IEEE CONFERENCE ON COMPUTING VISION AND PATTERN RECOGNITION, 2007.

A
Camera model and calibration overview

This chapter describes the physical parameters that relate World and camera coordinates, as well as the derived general perspective projection equation.

A.1
Model

The parameters that relate a camera to the World are divided into intrinsic and extrinsic categories. The former models the camera's actual coordinates to the idealised system (see eq. A-8) and the latter relates the device's coordinate system to a fixed World point, determining its position and orientation.

The coordinates of a point $\mathbf{P} = [x, y, z]^T$ and its image $\mathbf{P'} = [x', y', z']^T$ are related by

$$\begin{cases} x' = z'\dfrac{x}{z} \\ y' = z'\dfrac{y}{z} \end{cases} \tag{A-1}$$

Since $\mathbf{P'}$ lies in the camera plane, its z' coordinate is equal to the focal distance f'.

If normalising the image plane, a point $\hat{\mathbf{p}} = [\hat{u}, \hat{v}, 1]^T$ in this system can by related to the World coordinates by

$$\hat{\mathbf{p}} = \frac{1}{z}\begin{bmatrix} I & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{P} \\ 1 \end{bmatrix} \tag{A-2}$$

Let us consider the pinhole camera model, which can be perceived as a light-proof box with a small hole in one side. When light from a scene passes through this pinhole, it projects an inverted image on the opposite side. For instance, cameras using small apertures act like such box.

Assuming that the physical retina of a camera is located at a distance f from the pinhole and its centre is not at the origin of the camera system, the coordinates of an image point $\mathbf{p}[u, v, 1]^T$, in pixel units, are given by eq. (A-3), where

- u_0, v_0 are the offset from the retina centre;
- α, β are the magnification parameters, defined by the ratio between the distance f and the size of a pixel in the camera frame (expressed in the same length units as f);
- θ is the angle between the image axes, introduced to account for the skew caused by eventual manufacturing errors.

$$\begin{cases} u = \alpha\dfrac{x}{z} - \alpha\cot\theta\dfrac{x}{z} + u_0 \\ v = \dfrac{\beta}{\sin\theta}\dfrac{y}{z} + v_0 \end{cases} \tag{A-3}$$

Substituting (A-3) into (A-2) yields

$$\hat{\mathbf{p}} = \underbrace{\begin{bmatrix} \alpha & -\alpha\cot\theta & u_0 \\ 0 & \dfrac{\beta}{\sin\theta} & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathcal{K}} \mathbf{p} \tag{A-4}$$

or rather,

$$\hat{\mathbf{p}} = \frac{1}{z}\mathcal{M}\mathbf{P}, \qquad \text{where} \qquad \mathcal{M} \triangleq \begin{bmatrix} \mathcal{K} & \mathbf{0} \end{bmatrix} \tag{A-5}$$

In eq. (A-5), $\mathcal{M}$ represents the perspective projection that maps $\mathbf{P}$ into $\mathbf{p}$.

Furthermore, the camera and World frames may be distinct from one another, thus $\mathcal{M}$ can be rewritten as

$$\mathcal{M} = \mathcal{K}\begin{bmatrix} \mathcal{R} & \mathcal{T} \end{bmatrix} \tag{A-6}$$

whereby $\mathcal{R}$ and $\mathcal{T}$ are, respectively, the rotation matrix and the translation vector from the World to the camera system. Eq. (A-5) still applies and, in this case, is known as the general form of the perspective projection equation.

The depth z in eq. (A-5) is not an independent parameter and, in fact, equals $\mathbf{m}_3 \cdot \mathbf{P}$, where $\mathbf{m}_3$ is the third row of $\mathcal{M}$.

Explicitly writing the projection matrix as a function of its intrinsic (α, β, u_0, v_0 and θ) and extrinsic ($\mathcal{R}$ and $\mathcal{T}$) parameters yields the following 3×4 matrix

$$\mathcal{M} = \begin{bmatrix} \alpha \mathbf{r}_1^T - \alpha \cot\theta \mathbf{r}_2^T + u_0 \mathbf{r}_3^T & \alpha \mathbf{t}_1 - \alpha \cot\theta \mathbf{t}_2 + u_0 \mathbf{t}_3 \\ \dfrac{\beta}{\sin\theta} \mathbf{r}_2^T + v_0 \mathbf{r}_3^T & \dfrac{\beta}{\sin\theta} \mathbf{t}_2 + v_0 \mathbf{t}_3 \\ r_3^T & \mathbf{t}_3 \end{bmatrix} \tag{A-7}$$

A.2 Calibration

Camera calibration is the process of estimating the parameters introduced in last section.

Camera lenses are always subjectable to some degree of radial distortion, which are generally quite significant in underwater applications, for instance. To account for such phenomenon, the general perspective projection equation becomes

$$\begin{cases} (\mathbf{m}_1 - \lambda u \mathbf{m}_3) \cdot \mathbf{P} = \mathbf{0} \\ (\mathbf{m}_2 - \lambda v \mathbf{m}_3) \cdot \mathbf{P} = \mathbf{0} \end{cases} \tag{A-8}$$

where the parameter λ is a polynomial function of the distance between the image centre and the image point $\mathbf{p}$, given by

$$\lambda = 1 + \sum_{p=1}^{q} \kappa_p d^{2p} \tag{A-9}$$

where the κ_p term stands for the radial distortion coefficients. Assuming the image centre is known, both u_0 and v_0 can be set to 0, thus the distance

d expressed as a function of $\mathbf{p}$ yields

$$d = \sqrt{\frac{u^2}{\alpha^2} + \frac{v^2}{\beta^2} + 2\frac{uv}{\alpha\beta}\cos\theta} \tag{A-10}$$

which turns the calibration matter into a non-linear system of order $q + 11$.

Alternatively, it can be broken into two steps. The first consists in solving eq. (A-5) for nine camera parameters and the second, in estimating the remaining $q + 2$ by solving a simpler non-linear system.

B
Pose detection overview

In robotics, pose detection is the ultimate goal of a vision-based position control system. Through cameras, the differences between the actual and desired position of the manipulator can be estimated from pose detection, using, e.g., point matching. This information is then sent to controllers to, for instance, correct the pre-programmed trajectories. In other words, the aim is to determine the rotation matrix $\mathcal{R}$ and the translation vector $\mathcal{T}$ that map the position of an object in an image frame into its real location in the World system.

This chapter addresses the geometric and algebraic constraints that link two views of a scene, as well as the 3-D reconstruction of a scene from a set of point projections taken from two different cameras. In addition, a brief comment on the monoscopic variant of the pose detection issue is also included.

B.1
Monoscopic pose estimation

Assuming the intrinsic camera parameters are known, the calibration matrix $\mathcal{K}$ is constant and the pose estimation issue translates into the simplified calibration problem of finding the extrinsic camera parameters, provided that the correspondent World coordinates of each point in the database are also known.

B.2
Stereoscopic pose estimation

The depth of a image point along its projection ray cannot be directly assessed from a single image. However, if (at least) two views are available, it can be calculated through triangulation, for an image point lies, necessarily, in the pane formed by the other image and the camera centres. This is called the epipolar constraint, which is represented by a 3×3 matrix, named the essential matrix $\mathcal{E}$ (15) or the fundamental matrix $\mathcal{F}$, depending on whether the intrinsic camera parameters are known or unknown.

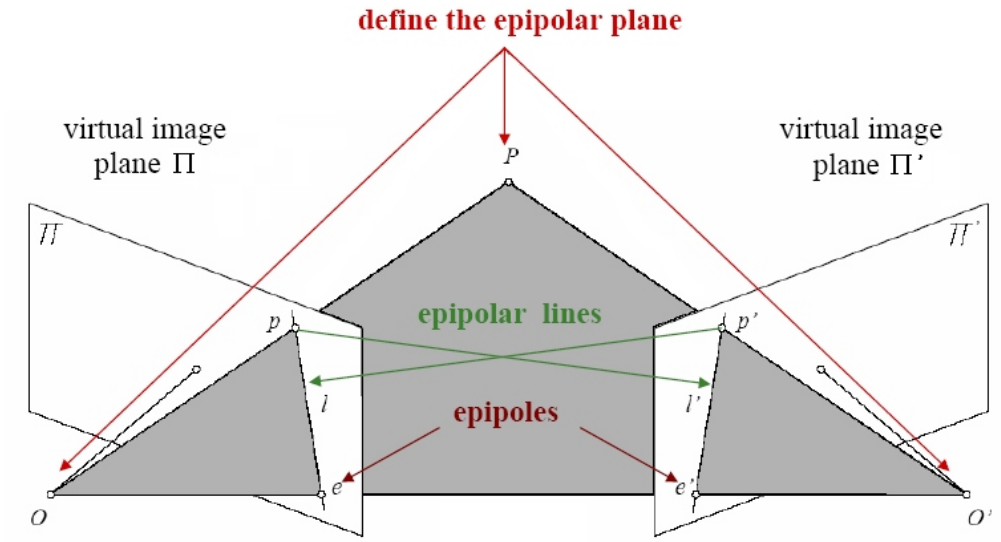

Figure B.1: Epipolar geometry.

Figure B.1 depicts the epipolar geometry. It shows that the point $\mathbf{P}$ and its images p and p', taken by a pair of cameras - i.e., a conjugate point - whose centres are O and O' all belong to the epipolar plane defined by OP and $O'P$. The epipole e' is the projection of the optical centre O of the left camera in the image taken by the right camera, and vice-versa and, since p and p' are images of the same point, $p'(p)$ lies on the epipolar line $l(l')$ associated with $p(p')$.

Again, let us assume that the intrinsic camera parameters were previously computed and are, therefore, known. In such case, the homogeneous and normalised image coordinate vectors are perfectly equivalent. That is, $\mathbf{p} = \hat{\mathbf{p}}$. It should have been made clear from Figure B.1 that

$$\overrightarrow{Op} \cdot \left[\overrightarrow{OO'} \times \overrightarrow{O'p'}\right] = 0 \tag{B-1}$$

which can be rewritten as

$$\hat{\mathbf{p}} \cdot [\mathcal{T} \times (\mathcal{R}\hat{\mathbf{p}}')] = 0 \tag{B-2}$$

The two projection matrices $\mathcal{M}$ and $\mathcal{M}'$, with respect to the left camera, are given by

$$\begin{cases} \mathcal{M} = \begin{bmatrix} I & \mathbf{0} \end{bmatrix} \\ \mathcal{M}' = \begin{bmatrix} \mathcal{R}^T & -\mathcal{R}^T\mathcal{T} \end{bmatrix} \end{cases} \tag{B-3}$$

Rewriting eq. (B-2) in terms of the essential matrix $\mathcal{E}$ yields

$$\hat{\mathbf{p}}^T \left[\mathcal{T} \times \mathcal{R}\right] \hat{\mathbf{p}}' = 0 \qquad \Longrightarrow \qquad \hat{\mathbf{p}}^T \mathcal{E} \hat{\mathbf{p}}' = 0 \tag{B-4}$$

The nine coefficients of $\mathcal{E}$ are only defined up to scale and can be parametrised by the three degrees of freedom of $\mathcal{R}$ and the two of $\mathcal{T}$.

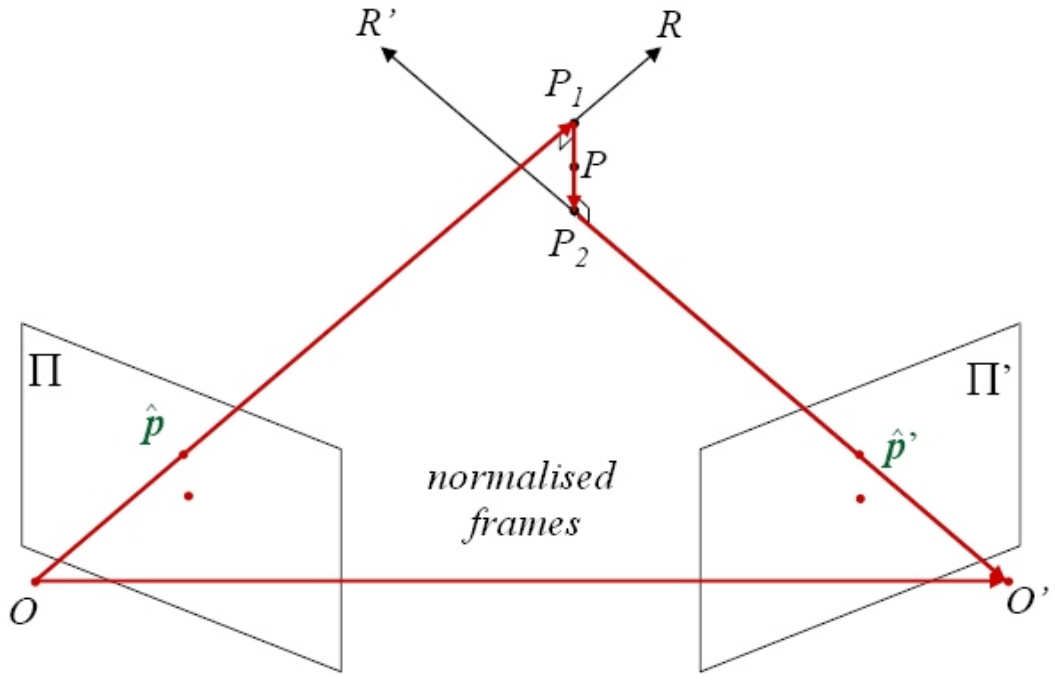

Figure B.2: 3-D reconstruction by triangulation.

The 3-D reconstruction can be performed by triangulation, as depicted in Figure B.2. In eq. (B-4), the product $\mathcal{E}\hat{\mathbf{p}}'$ represents the epipolar line l associated to p'. Furthermore, from Figure B.2

$$\overrightarrow{OP_1} + \overrightarrow{P_1P_2} + \overrightarrow{P_2O'} = \overrightarrow{OO'} \tag{B-5}$$

which can be rewritten, *with respect to the* ***left*** *image plane*, as

$$z\hat{\mathbf{p}} - d_l\left(\hat{\mathbf{p}} \times \mathcal{R}'\hat{\mathbf{p}}'\right) - z'\mathcal{R}'\hat{\mathbf{p}}' - \mathcal{T} = 0 \tag{B-6}$$

where d_l is the euclidean distance between the origin and l (see Figure B.1).

Finally, if that is the case, the World coordinates $\mathbf{P}$ of the conjugate pair can then be calculated by combining the projective projection equation (see eq. (A-5)) and eq. B-6 to yield eq. B-7.

$$\mathbf{P} = z\hat{\mathbf{p}} - d_l\frac{\left(\hat{\mathbf{p}} \times \mathcal{R}'\hat{\mathbf{p}}'\right)}{2} \tag{B-7}$$

C
Match quality assessment: full tables

This chapter contains full records of the results summarised in section 5.1.2.

C.1
SIFT records

pair	#matches	#false m.	hit rate	time
198	99	0	100	0,234

Figure C.1: SIFT: lowest incidence of matches.

pair	#matches	#false m.	hit rate	time
284	1073	0	100	0,968

Figure C.2: SIFT: highest incidence of matches.

pair	#matches	#false m.	hit rate	time	
29	419	6	98,57	0,844	
47	443	6	98,65	0,875	
142	380	6	98,42	0,766	
	414	6	98,54667	0,828333	Mean
	31,79623	0	0,116762	0,056163	Std Dev

Figure C.3: SIFT: worst hit rate behaviour.

pair	#matches	#false m.	hit rate	time
1	713	0	100	1,609
3	659	0	100	1,313
6	587	0	100	0,844
12	672	0	100	0,875
16	723	0	100	1
18	626	0	100	1,157
56	766	0	100	1,078
62	777	0	100	0,984
67	639	0	100	0,859
73	599	0	100	0,938
76	1063	0	100	1,063
82	649	0	100	0,844
88	607	0	100	0,828
91	617	0	100	0,953
97	693	0	100	0,985
106	643	0	100	0,875
108	443	0	100	0,907
113	1010	0	100	1,046
121	397	0	100	0,953
123	865	0	100	1,125
126	669	0	100	1,125
128	509	0	100	0,718
129	668	0	100	0,859
131	537	0	100	0,735
135	667	0	100	0,859
138	796	0	100	0,906
143	585	0	100	0,765
144	751	0	100	0,985
147	938	0	100	1,063
151	379	0	100	0,765
160	782	0	100	1,078
162	557	0	100	0,734
163	722	0	100	0,875
166	740	0	100	0,937
174	448	0	100	0,703
175	700	0	100	0,828
178	604	0	100	0,812
181	753	0	100	0,829
183	483	0	100	0,656
184	705	0	100	0,86
185	762	0	100	0,891
193	446	0	100	0,594
194	677	0	100	0,782
196	227	0	100	0,25
197	211	0	100	0,25
198	99	0	100	0,234
199	405	0	100	0,516

pair	#matches	#false m.	hit rate	time	
200	615	0	100	0,687	
211	437	0	100	0,61	
212	619	0	100	0,828	
214	517	0	100	0,75	
216	330	0	100	0,641	
218	714	0	100	1,125	
227	591	0	100	0,734	
228	735	0	100	0,813	
231	228	0	100	0,468	
234	694	0	100	0,875	
236	477	0	100	0,672	
241	413	0	100	0,735	
252	738	0	100	0,844	
266	427	0	100	0,594	
267	640	0	100	0,828	
269	583	0	100	0,734	
277	601	0	100	0,829	
279	739	0	100	0,813	
280	954	0	100	0,875	
282	995	0	100	0,89	
283	820	0	100	0,891	
284	1073	0	100	0,968	
286	1066	0	100	0,938	
288	1054	0	100	0,937	
290	634	0	100	0,781	
298	521	0	100	0,687	
314	485	0	100	0,718	
316	513	0	100	0,703	
322	518	0	100	0,719	
323	712	0	100	0,781	
338	485	0	100	0,687	
339	655	0	100	0,859	
344	432	0	100	0,641	
347	434	0	100	0,672	
348	741	0	100	0,891	
349	768	0	100	0,906	
351	372	0	100	0,547	
354	673	0	100	0,89	
361	889	0	100	0,937	
362	314	0	100	0,813	
363	428	0	100	0,61	
364	661	0	100	0,859	
370	688	0	100	0,812	
377	307	0	100	0,813	
384	378	0	100	0,625	
388	382	0	100	0,578	
391	720	0	100	0,89	
400	451	0	100	0,703	
	618,8316	0	100	0,821537	Mean
	197,4748	0	0	0,200059	Std Dev

Figure C.4: SIFT: best hit rate behaviour.

Seq. Pair	# Matches	# False Matches	Hit Rate	Time (s)
1	713	0	100	1,609
2	399	2	99,5	1,594
3	659	0	100	1,313
4	692	1	99,86	0,938
5	381	1	99,74	0,797
6	587	0	100	0,844
7	718	5	99,3	0,938
8	420	3	99,29	0,797
9	622	2	99,68	0,859
10	767	2	99,74	0,969
11	430	2	99,53	0,843
12	672	0	100	0,875
13	701	2	99,71	1,078
14	406	3	99,26	0,828
15	609	1	99,84	0,843
16	723	0	100	1
17	410	1	99,76	0,812
18	626	0	100	1,157
19	706	1	99,86	0,938
20	435	4	99,08	0,89
21	596	3	99,5	0,828
22	788	1	99,87	0,984
23	416	2	99,52	0,906
24	655	2	99,69	0,937
25	798	4	99,5	1
26	417	5	98,8	0,937
27	733	2	99,73	0,891
28	725	1	99,86	0,969
29	419	6	98,57	0,844
30	636	3	99,53	0,843
31	714	1	99,86	1
32	409	4	99,02	0,797
33	627	1	99,84	0,859
34	709	1	99,86	0,969
35	406	2	99,51	0,828
36	572	3	99,48	0,828
37	696	1	99,86	0,938
38	379	1	99,74	0,796
39	618	1	99,84	1,015
40	783	2	99,74	1,062
41	426	3	99,3	0,906
42	655	3	99,54	0,875
43	776	2	99,74	0,969
44	424	4	99,06	0,875
45	637	3	99,53	0,891
46	837	2	99,76	1,047
47	443	6	98,65	0,875
48	664	2	99,7	0,907
49	922	3	99,67	1,11
50	858	2	99,77	1,109
51	437	4	99,08	0,969
52	647	2	99,69	0,922
53	431	2	99,54	0,828
54	661	3	99,55	0,906
55	991	1	99,9	1,063
56	766	0	100	1,078
57	444	1	99,77	0,859
58	630	2	99,68	0,844
59	413	1	99,76	0,828
60	671	2	99,7	0,907
61	1061	2	99,81	1,079
62	777	0	100	0,984
63	439	1	99,77	1,094
64	641	1	99,84	0,891
65	891	1	99,89	1,032
66	454	3	99,34	0,875
67	639	0	100	0,859
68	410	3	99,27	0,938
69	663	1	99,85	0,89
70	1051	1	99,9	1,125
71	760	2	99,74	0,969
72	413	3	99,27	0,828
73	599	0	100	0,938
74	391	4	98,98	0,859
75	660	1	99,85	0,985
76	1063	0	100	1,063
77	826	1	99,88	1,016
78	435	1	99,77	0,922
79	605	1	99,83	0,828
80	841	2	99,76	0,984
81	397	2	99,5	1,125
82	649	0	100	0,844

Seq. Pair	# Matches	# False Matches	Hit Rate	Time (s)
83	782	2	99,74	0,922
84	404	2	99,5	0,782
85	646	1	99,85	0,844
86	772	2	99,74	0,968
87	401	1	99,75	0,906
88	607	0	100	0,828
89	828	3	99,64	0,984
90	399	1	99,75	0,797
91	617	0	100	0,953
92	708	1	99,86	0,938
93	345	1	99,71	0,813
94	605	1	99,83	0,828
95	833	1	99,88	0,984
96	439	2	99,54	0,828
97	693	0	100	0,985
98	865	2	99,77	1,109
99	448	4	99,11	0,828
100	679	3	99,56	0,969
101	835	1	99,88	1,047
102	428	4	99,07	0,906
103	693	2	99,71	0,922
104	955	3	99,69	1,094
105	460	4	99,13	0,922
106	643	0	100	0,875
107	855	4	99,53	1,078
108	443	0	100	0,907
109	670	1	99,85	0,953
110	922	3	99,67	1,062
111	433	3	99,31	1,047
112	682	3	99,56	0,875
113	1010	0	100	1,046
114	776	2	99,74	1,015
115	341	1	99,71	0,796
116	605	3	99,5	0,938
117	877	1	99,89	1,047
118	407	2	99,51	0,875
119	650	3	99,54	0,891
120	799	3	99,62	1,031
121	397	0	100	0,953
122	646	1	99,85	0,875
123	865	0	100	1,125
124	412	3	99,27	0,875
125	597	1	99,83	0,844
126	669	0	100	1,125
127	361	1	99,72	0,937
128	509	0	100	0,718
129	668	0	100	0,859
130	331	5	98,49	0,719
131	537	0	100	0,735
132	774	2	99,74	0,906
133	321	1	99,69	0,75
134	535	1	99,81	0,734
135	667	0	100	0,859
136	341	2	99,41	0,703
137	551	1	99,82	0,719
138	796	0	100	0,906
139	373	2	99,46	0,734
140	543	1	99,82	0,765
141	588	1	99,83	0,968
142	380	6	98,42	0,766
143	585	0	100	0,765
144	751	0	100	0,985
145	407	3	99,26	0,844
146	546	1	99,82	0,766
147	938	0	100	1,063
148	766	1	99,87	0,922
149	398	1	99,75	0,797
150	575	2	99,65	0,751
151	379	0	100	0,765
152	597	3	99,5	0,891
153	973	1	99,9	1,843
154	781	1	99,87	0,953
155	383	1	99,74	0,766
156	492	1	99,8	0,735
157	763	1	99,87	0,953
158	373	1	99,73	0,75
159	522	1	99,81	0,735
160	782	0	100	1,078
161	396	2	99,49	0,734
162	557	0	100	0,734
163	722	0	100	0,875
164	339	2	99,41	0,766

Seq. Pair	# Matches	# False Matches	Hit Rate	Time (s)
165	494	1	99,8	0,672
166	740	0	100	0,937
167	313	2	99,36	0,672
168	438	1	99,77	0,594
169	640	2	99,69	0,781
170	337	3	99,11	0,703
171	445	1	99,78	0,625
172	659	1	99,85	0,86
173	332	1	99,7	0,672
174	448	0	100	0,703
175	700	0	100	0,828
176	344	5	98,55	0,688
177	469	1	99,79	0,672
178	604	0	100	0,812
179	349	3	99,14	0,672
180	471	2	99,58	0,672
181	753	0	100	0,829
182	360	2	99,44	0,735
183	483	0	100	0,656
184	705	0	100	0,86
185	762	0	100	0,891
186	315	1	99,68	0,656
187	527	1	99,81	0,656
188	735	1	99,86	0,797
189	331	2	99,4	0,688
190	477	1	99,79	0,657
191	622	1	99,84	0,813
192	299	3	99	0,672
193	446	0	100	0,594
194	677	0	100	0,782
195	118	1	99,15	0,594
196	227	0	100	0,25
197	211	0	100	0,25
198	99	0	100	0,234
199	405	0	100	0,516
200	615	0	100	0,687
201	267	1	99,63	0,594
202	493	3	99,39	0,703
203	641	2	99,69	1,5
204	313	1	99,68	0,593
205	460	2	99,57	0,672
206	600	1	99,83	0,703
207	340	2	99,41	0,671
208	483	1	99,79	0,625
209	707	1	99,86	0,781
210	350	2	99,43	0,687
211	437	0	100	0,61
212	619	0	100	0,828
213	360	3	99,17	0,672
214	517	0	100	0,75
215	726	1	99,86	0,797
216	330	0	100	0,641
217	524	1	99,81	0,641
218	714	0	100	1,125
219	387	2	99,48	0,719
220	530	2	99,62	0,703
221	650	3	99,54	0,797
222	389	2	99,49	0,688
223	504	2	99,6	0,656
224	756	1	99,87	0,812
225	335	3	99,1	0,672
226	454	1	99,78	0,625
227	591	0	100	0,734
228	735	0	100	0,813
229	209	1	99,52	0,625
230	412	1	99,76	0,5
231	228	0	100	0,468
232	615	1	99,84	0,797
233	703	1	99,86	0,812
234	694	0	100	0,875
235	333	2	99,4	0,672
236	477	0	100	0,672
237	714	2	99,72	0,797
238	399	3	99,25	0,704
239	501	2	99,6	0,687
240	626	1	99,84	0,766
241	413	0	100	0,735
242	502	1	99,8	0,64
243	730	1	99,86	0,812
244	405	2	99,51	0,687
245	526	3	99,43	0,688
246	549	2	99,63	0,781

Seq. Pair	# Matches	# False Matches	Hit Rate	Time (s)
247	417	2	99,52	0,703
248	483	2	99,59	0,688
249	626	1	99,84	0,922
250	437	1	99,77	0,703
251	524	1	99,81	0,718
252	738	0	100	0,844
253	436	2	99,54	0,719
254	567	2	99,65	0,813
255	660	2	99,7	0,781
256	451	2	99,56	0,735
257	528	2	99,62	0,734
258	747	2	99,73	0,797
259	431	2	99,54	0,718
260	551	2	99,64	0,718
261	565	2	99,65	0,813
262	152	5	96,71	0,625
263	380	1	99,74	0,453
264	320	2	99,38	0,406
265	204	1	99,51	0,437
266	427	0	100	0,594
267	640	0	100	0,828
268	266	1	99,62	0,625
269	583	0	100	0,734
270	559	1	99,82	0,719
271	401	1	99,75	0,828
272	517	2	99,61	0,735
273	714	2	99,72	1,047
274	442	3	99,32	0,719
275	518	1	99,81	0,734
276	448	3	99,33	0,719
277	601	0	100	0,829
278	796	1	99,87	0,828
279	739	0	100	0,813
280	954	0	100	0,875
281	751	1	99,87	0,875
282	995	0	100	0,89
283	820	0	100	0,891
284	1073	0	100	0,968
285	884	1	99,89	0,891
286	1066	0	100	0,938
287	934	1	99,89	0,906
288	1054	0	100	0,937
289	730	1	99,86	0,812
290	634	0	100	0,781
291	354	1	99,72	0,687
292	440	2	99,55	0,594
293	634	1	99,84	0,704
294	432	2	99,54	0,656
295	511	1	99,8	0,688
296	640	1	99,84	0,765
297	464	1	99,78	0,656
298	521	0	100	0,687
299	708	1	99,86	0,781
300	454	3	99,34	0,687
301	540	2	99,63	0,703
302	717	2	99,72	0,797
303	438	4	99,09	0,719
304	536	3	99,44	0,766
305	775	3	99,61	0,859
306	457	4	99,12	0,735
307	574	2	99,65	0,75
308	709	1	99,86	0,797
309	403	2	99,5	0,75
310	524	2	99,62	0,718
311	679	1	99,85	0,797
312	840	1	99,88	1
313	392	3	99,23	0,781
314	485	0	100	0,718
315	427	2	99,53	0,734
316	513	0	100	0,703
317	718	3	99,58	0,844
318	874	1	99,89	0,906
319	411	3	99,27	0,75
320	518	1	99,81	0,688
321	422	3	99,29	0,734
322	518	0	100	0,719
323	712	0	100	0,781
324	645	1	99,84	0,844
325	382	3	99,21	0,688
326	489	1	99,8	0,703
327	715	1	99,86	0,859
328	428	1	99,77	0,828

Seq. Pair	# Matches	# False Matches	Hit Rate	Time (s)
329	510	2	99,61	0,718
330	673	2	99,7	0,812
331	380	1	99,74	0,75
332	492	1	99,8	0,688
333	731	1	99,86	0,86
334	375	5	98,67	0,718
335	468	2	99,57	0,656
336	518	3	99,42	0,75
337	383	4	98,96	0,75
338	485	0	100	0,687
339	655	0	100	0,859
340	375	4	98,93	0,797
341	454	1	99,78	0,656
342	715	1	99,86	0,828
343	345	3	99,13	0,75
344	432	0	100	0,641
345	666	2	99,7	0,813
346	333	2	99,4	0,797
347	434	0	100	0,672
348	741	0	100	0,891
349	768	0	100	0,906
350	285	4	98,6	0,781
351	372	0	100	0,547
352	375	1	99,73	0,641
353	446	2	99,55	0,656
354	673	0	100	0,89
355	883	2	99,77	0,938
356	331	3	99,09	0,735
357	424	1	99,76	0,609
358	337	2	99,41	0,688
359	430	2	99,53	0,64
360	650	2	99,69	0,797
361	889	0	100	0,937
362	314	0	100	0,813
363	428	0	100	0,61
364	661	0	100	0,859

Seq. Pair	# Matches	# False Matches	Hit Rate	Time (s)
365	316	3	99,00	0,75
366	414	2	99,52	0,626
367	733	1	99,86	0,859
368	307	2	99,35	0,735
369	393	1	99,75	0,594
370	688	0	100	0,812
371	289	2	99,31	0,734
372	411	2	99,51	0,625
373	641	2	99,69	0,812
374	289	2	99,31	0,75
375	411	4	99,03	0,625
376	689	1	99,85	0,843
377	307	0	100	0,813
378	403	3	99,26	0,641
379	678	1	99,85	0,812
380	314	2	99,36	0,766
381	400	2	99,5	0,641
382	721	1	99,86	1,063
383	329	2	99,39	0,781
384	378	0	100	0,625
385	719	2	99,72	0,891
386	816	1	99,88	1
387	325	2	99,38	0,89
388	382	0	100	0,578
389	351	2	99,43	0,687
390	432	1	99,77	0,688
391	720	0	100	0,89
392	860	1	99,88	1,016
393	317	2	99,37	0,844
394	434	1	99,77	0,671
395	756	2	99,74	0,922
396	328	5	98,48	0,875
397	476	4	99,16	0,688
398	744	1	99,87	0,875
399	357	1	99,72	0,875
400	451	0	100	0,703

Figure C.5: SIFT: full record.

C.2 LSM records

pair	#matches	#false m.	hit rate	time
198	9	0	100	0,14

Figure C.6: LSM: lowest incidence of matches.

pair	#matches	#false m.	hit rate	time
76	903	0	100	1,437

Figure C.7: LSM: highest incidence of matches.

pair	#matches	#false m.	hit rate	time	
20	220	2	99,09	0,625	
29	242	2	99,17	0,609	
246	424	2	99,55	0,703	
247	606	2	99,38	0,579	
	373	2	99,2975	0,629	Mean
	180,2406	0	0,208066	0,05289	Std Dev

Figure C.8: LSM: worst hit rate behaviour.

pair	#matches	#false m.	hit rate	time
1	548	0	100	0,875
2	253	0	100	0,547
3	526	0	100	0,828
4	537	0	100	0,844
5	237	0	100	0,516
6	453	0	100	1,125
7	548	0	100	0,875
8	285	0	100	0,547
9	482	0	100	0,781
10	612	0	100	0,954
12	528	0	100	0,844
13	564	0	100	0,891
14	221	0	100	0,578
15	472	0	100	0,765
16	576	0	100	0,922
18	487	0	100	0,781
19	542	0	100	0,859
21	451	0	100	0,766
22	631	0	100	0,984
25	644	0	100	1
26	232	0	100	0,578
28	588	0	100	0,938
30	499	0	100	0,828
31	570	0	100	0,875
32	167	0	100	0,61
33	506	0	100	0,812
34	584	0	100	0,906
35	213	0	100	0,594
36	455	0	100	0,766
37	565	0	100	0,907
38	173	0	100	0,625
39	485	0	100	0,829
40	646	0	100	1
41	213	0	100	0,625
43	654	0	100	1
44	220	0	100	0,657
45	517	0	100	0,844
46	684	0	100	1,125
47	216	0	100	0,672
48	535	0	100	0,859
49	770	0	100	1,234
50	715	0	100	1,125
51	212	0	100	0,64
52	509	0	100	0,828
53	207	0	100	0,61
54	525	0	100	0,859
56	630	0	100	0,985
58	508	0	100	0,969
60	545	0	100	0,859
61	898	0	100	1,437
62	654	0	100	0,984
63	179	0	100	0,625

pair	#matches	#false m.	hit rate	time
64	520	0	100	0,829
65	742	0	100	1,234
66	184	0	100	0,687
67	512	0	100	0,844
69	544	0	100	0,859
71	643	0	100	0,969
72	175	0	100	0,625
73	480	0	100	0,813
74	159	0	100	0,609
75	543	0	100	1
76	903	0	100	1,437
77	688	0	100	1,109
78	184	0	100	0,64
79	475	0	100	0,797
81	165	0	100	0,594
82	522	0	100	1,11
85	532	0	100	0,86
86	646	0	100	1,109
87	167	0	100	0,609
88	473	0	100	0,828
89	683	0	100	1,219
90	167	0	100	0,609
91	506	0	100	0,859
93	102	0	100	0,546
94	454	0	100	0,766
95	673	0	100	1,109
97	574	0	100	0,891
98	718	0	100	1,125
99	156	0	100	0,656
100	547	0	100	0,891
101	682	0	100	1,11
103	559	0	100	0,906
104	810	0	100	1,219
105	159	0	100	0,703
106	519	0	100	1,11
108	164	0	100	0,671
109	543	0	100	0,875
111	169	0	100	0,656
113	840	0	100	1,328
114	640	0	100	1
115	109	0	100	0,547
116	482	0	100	0,781
117	725	0	100	1,11
118	75	0	100	0,641
119	523	0	100	0,875
120	682	0	100	1,109
121	130	0	100	0,609
122	505	0	100	1,109
123	704	0	100	1,11
124	134	0	100	0,625
125	478	0	100	0,781
126	549	0	100	0,859

pair	#matches	#false m.	hit rate	time
128	425	0	100	0,687
129	548	0	100	0,86
130	91	0	100	0,5
131	414	0	100	0,891
132	632	0	100	0,969
133	85	0	100	0,468
134	388	0	100	0,657
135	522	0	100	0,813
136	101	0	100	0,5
138	647	0	100	0,985
139	105	0	100	0,578
140	431	0	100	0,703
141	472	0	100	0,75
143	449	0	100	0,734
144	594	0	100	0,922
145	132	0	100	0,625
146	412	0	100	0,735
147	743	0	100	1,219
148	609	0	100	0,984
149	114	0	100	0,641
150	421	0	100	0,703
151	97	0	100	0,546
152	429	0	100	0,718
153	760	0	100	1,219
154	619	0	100	0,953
155	90	0	100	0,578
156	357	0	100	0,641
157	573	0	100	0,891
158	108	0	100	0,547
159	384	0	100	0,875
160	595	0	100	0,922
161	106	0	100	0,578
162	426	0	100	0,687
163	569	0	100	0,875
164	137	0	100	0,515
165	380	0	100	0,625
166	570	0	100	0,875
168	330	0	100	0,563
169	498	0	100	0,765
170	104	0	100	0,5
171	323	0	100	0,547
172	509	0	100	0,781
173	79	0	100	0,5
174	346	0	100	0,593
175	558	0	100	0,859
176	125	0	100	0,516
177	344	0	100	0,594
178	457	0	100	0,735
179	120	0	100	0,531
180	353	0	100	0,703
181	604	0	100	0,937
182	122	0	100	0,516
183	359	0	100	0,625
184	566	0	100	0,875
185	610	0	100	0,922
186	99	0	100	0,469
190	346	0	100	0,594
191	489	0	100	0,765
193	333	0	100	0,531
194	521	0	100	0,797
195	28	0	100	0,156
196	142	0	100	0,265
197	131	0	100	0,25
198	9	0	100	0,14
199	306	0	100	0,5
200	472	0	100	0,735
201	115	0	100	0,406
204	139	0	100	0,453
205	341	0	100	0,688
207	146	0	100	0,5
208	356	0	100	0,609
209	558	0	100	0,859

pair	#matches	#false m.	hit rate	time
211	330	0	100	0,563
212	498	0	100	0,75
214	406	0	100	0,969
215	583	0	100	0,922
216	137	0	100	0,516
217	398	0	100	0,671
218	579	0	100	0,875
221	522	0	100	0,813
222	228	0	100	0,562
223	387	0	100	0,657
224	633	0	100	0,937
225	183	0	100	0,469
226	356	0	100	0,563
227	476	0	100	0,719
228	595	0	100	0,891
229	108	0	100	0,297
231	91	0	100	0,391
232	486	0	100	0,781
233	559	0	100	0,843
234	556	0	100	0,922
235	210	0	100	0,5
236	391	0	100	0,625
237	575	0	100	0,875
238	295	0	100	0,563
239	391	0	100	0,64
240	505	0	100	0,781
241	307	0	100	0,562
242	403	0	100	0,641
243	586	0	100	0,906
244	304	0	100	0,562
248	388	0	100	0,625
249	520	0	100	0,797
252	606	0	100	0,906
253	345	0	100	0,594
254	463	0	100	0,828
255	537	0	100	0,812
256	351	0	100	0,578
257	411	0	100	0,671
258	622	0	100	0,922
259	332	0	100	0,562
260	447	0	100	0,703
261	459	0	100	0,703
262	82	0	100	0,203
263	280	0	100	0,468
264	228	0	100	0,406
265	114	0	100	0,281
266	333	0	100	0,547
267	498	0	100	0,75
268	183	0	100	0,391
269	469	0	100	0,735
270	453	0	100	0,703
271	295	0	100	0,516
272	410	0	100	0,672
273	584	0	100	0,891
274	354	0	100	0,563
275	417	0	100	0,656
276	336	0	100	0,687
277	487	0	100	0,765
278	647	0	100	0,953
279	604	0	100	0,907
280	778	0	100	1,219
281	616	0	100	0,906
282	834	0	100	1,219
283	669	0	100	1,109
284	879	0	100	1,438
285	729	0	100	1,125
286	873	0	100	1,328
287	764	0	100	1,109
288	868	0	100	1,219
289	603	0	100	0,922
290	526	0	100	0,781
292	353	0	100	0,578

pair	#matches	#false m.	hit rate	time
293	512	0	100	0,781
296	526	0	100	0,86
297	363	0	100	0,594
298	423	0	100	0,672
299	590	0	100	0,875
300	362	0	100	0,609
301	448	0	100	0,703
302	586	0	100	0,891
303	335	0	100	0,578
304	421	0	100	0,688
305	617	0	100	0,922
306	328	0	100	0,609
307	461	0	100	0,734
308	574	0	100	0,859
309	284	0	100	0,672
311	540	0	100	0,812
312	662	0	100	0,984
313	235	0	100	0,547
314	389	0	100	0,641
315	290	0	100	0,594
316	392	0	100	0,656
317	581	0	100	0,875
318	724	0	100	1,125
319	289	0	100	0,563
320	402	0	100	0,656
322	389	0	100	0,641
323	564	0	100	1,11
325	237	0	100	0,547
327	568	0	100	0,86
328	274	0	100	0,703
329	395	0	100	0,656
332	371	0	100	0,641
334	206	0	100	0,547
335	355	0	100	0,61
336	394	0	100	0,64
338	374	0	100	0,64
339	520	0	100	0,813
340	195	0	100	0,547
341	338	0	100	0,578
342	572	0	100	0,89
343	145	0	100	0,531
344	331	0	100	0,562
345	529	0	100	0,812
346	147	0	100	0,5
347	313	0	100	0,547
348	600	0	100	0,906

pair	#matches	#false m.	hit rate	time
349	620	0	100	1,109
350	105	0	100	0,438
351	269	0	100	0,484
353	339	0	100	0,578
354	539	0	100	0,843
355	719	0	100	1,11
356	132	0	100	0,5
357	315	0	100	0,531
358	138	0	100	0,468
359	310	0	100	0,61
361	706	0	100	1,109
362	115	0	100	0,453
363	321	0	100	0,547
364	515	0	100	0,906
365	111	0	100	0,453
366	316	0	100	0,532
367	565	0	100	0,875
368	99	0	100	0,453
369	281	0	100	0,5
370	523	0	100	0,859
371	84	0	100	0,422
372	299	0	100	0,531
373	495	0	100	0,781
374	90	0	100	0,407
375	295	0	100	0,516
376	532	0	100	0,828
377	134	0	100	0,438
378	295	0	100	0,516
379	529	0	100	0,828
380	140	0	100	0,438
381	286	0	100	0,578
382	546	0	100	0,844
383	160	0	100	0,453
384	278	0	100	0,469
385	550	0	100	0,844
386	613	0	100	0,937
387	173	0	100	0,453
388	275	0	100	0,5
390	305	0	100	0,532
391	522	0	100	0,828
392	631	0	100	0,968
393	141	0	100	0,453
394	297	0	100	0,531
397	341	0	100	0,578
398	575	0	100	0,875
400	338	0	100	0,563

Figure C.9: LSM: best hit rate behaviour.

Seq. Pair	# Matches	# False Matches	Hit Rate	Time (s)
1	548	0	100	0,875
2	253	0	100	0,547
3	526	0	100	0,828
4	537	0	100	0,844
5	237	0	100	0,516
6	453	0	100	1,125
7	548	0	100	0,875
8	285	0	100	0,547
9	482	0	100	0,781
10	612	0	100	0,954
11	256	1	99,61	0,578
12	528	0	100	0,844
13	564	0	100	0,891
14	221	0	100	0,578
15	472	0	100	0,765
16	576	0	100	0,922
17	214	1	99,53	0,562
18	487	0	100	0,781
19	542	0	100	0,859
20	220	2	99,09	0,625
21	451	0	100	0,766
22	631	0	100	0,984
23	259	1	99,61	0,593
24	518	1	99,81	0,843
25	644	0	100	1
26	232	0	100	0,578
27	589	1	99,83	0,937
28	588	0	100	0,938
29	242	2	99,17	0,609
30	499	0	100	0,828
31	570	0	100	0,875
32	167	0	100	0,61
33	506	0	100	0,812
34	584	0	100	0,906
35	213	0	100	0,594
36	455	0	100	0,766
37	565	0	100	0,907
38	173	0	100	0,625
39	485	0	100	0,829
40	646	0	100	1
41	213	0	100	0,625
42	524	1	99,81	1
43	654	0	100	1
44	220	0	100	0,657
45	517	0	100	0,844
46	684	0	100	1,125
47	216	0	100	0,672
48	535	0	100	0,859
49	770	0	100	1,234
50	715	0	100	1,125
51	212	0	100	0,64
52	509	0	100	0,828
53	207	0	100	0,61
54	525	0	100	0,859
55	838	1	99,88	1,328
56	630	0	100	0,985
57	204	1	99,51	0,656
58	508	0	100	0,969
59	179	1	99,44	0,61
60	545	0	100	0,859
61	898	0	100	1,437
62	654	0	100	0,984
63	179	0	100	0,625
64	520	0	100	0,829
65	742	0	100	1,234
66	184	0	100	0,687
67	512	0	100	0,844
68	177	1	99,44	0,609
69	544	0	100	0,859
70	873	1	99,89	1,328
71	643	0	100	0,969
72	175	0	100	0,625
73	480	0	100	0,813
74	159	0	100	0,609
75	543	0	100	1
76	903	0	100	1,437
77	688	0	100	1,109
78	184	0	100	0,64
79	475	0	100	0,797
80	691	1	99,86	1,109
81	165	0	100	0,594
82	522	0	100	1,11

Seq. Pair	# Matches	# False Matches	Hit Rate	Time (s)
83	643	1	99,84	1,109
84	175	1	99,43	0,704
85	532	0	100	0,86
86	646	0	100	1,109
87	167	0	100	0,609
88	473	0	100	0,828
89	683	0	100	1,219
90	167	0	100	0,609
91	506	0	100	0,859
92	591	1	99,83	0,969
93	102	0	100	0,546
94	454	0	100	0,766
95	673	0	100	1,109
96	175	1	99,43	0,641
97	574	0	100	0,891
98	718	0	100	1,125
99	156	0	100	0,656
100	547	0	100	0,891
101	682	0	100	1,11
102	148	1	99,32	0,656
103	559	0	100	0,906
104	810	0	100	1,219
105	159	0	100	0,703
106	519	0	100	1,11
107	725	1	99,86	1,109
108	164	0	100	0,671
109	543	0	100	0,875
110	772	1	99,87	1,219
111	169	0	100	0,656
112	539	1	99,81	0,875
113	840	0	100	1,328
114	640	0	100	1
115	109	0	100	0,547
116	482	0	100	0,781
117	725	0	100	1,11
118	75	0	100	0,641
119	523	0	100	0,875
120	682	0	100	1,109
121	130	0	100	0,609
122	505	0	100	1,109
123	704	0	100	1,11
124	134	0	100	0,625
125	478	0	100	0,781
126	549	0	100	0,859
127	131	1	99,24	0,547
128	425	0	100	0,687
129	548	0	100	0,86
130	91	0	100	0,5
131	414	0	100	0,891
132	632	0	100	0,969
133	85	0	100	0,468
134	388	0	100	0,657
135	522	0	100	0,813
136	101	0	100	0,5
137	429	1	99,77	0,704
138	647	0	100	0,985
139	105	0	100	0,578
140	431	0	100	0,703
141	472	0	100	0,75
142	98	1	98,98	0,563
143	449	0	100	0,734
144	594	0	100	0,922
145	132	0	100	0,625
146	412	0	100	0,735
147	743	0	100	1,219
148	609	0	100	0,984
149	114	0	100	0,641
150	421	0	100	0,703
151	97	0	100	0,546
152	429	0	100	0,718
153	760	0	100	1,219
154	619	0	100	0,953
155	90	0	100	0,578
156	357	0	100	0,641
157	573	0	100	0,891
158	108	0	100	0,547
159	384	0	100	0,875
160	595	0	100	0,922
161	106	0	100	0,578
162	426	0	100	0,687
163	569	0	100	0,875
164	137	0	100	0,515
165	380	0	100	0,625

Seq. Pair	# Matches	# False Matches	Hit Rate	Time (s)
166	570	0	100	0,875
167	94	1	98,94	0,469
168	330	0	100	0,563
169	498	0	100	0,765
170	104	0	100	0,5
171	323	0	100	0,547
172	509	0	100	0,781
173	79	0	100	0,5
174	346	0	100	0,593
175	558	0	100	0,859
176	125	0	100	0,516
177	344	0	100	0,594
178	457	0	100	0,735
179	120	0	100	0,531
180	353	0	100	0,703
181	604	0	100	0,937
182	122	0	100	0,516
183	359	0	100	0,625
184	566	0	100	0,875
185	610	0	100	0,922
186	99	0	100	0,469
187	406	1	99,75	0,656
188	561	1	99,82	0,875
189	120	1	99,17	0,484
190	346	0	100	0,594
191	489	0	100	0,765
192	95	1	98,95	0,453
193	333	0	100	0,531
194	521	0	100	0,797
195	28	0	100	0,156
196	142	0	100	0,265
197	131	0	100	0,25
198	9	0	100	0,14
199	306	0	100	0,5
200	472	0	100	0,735
201	115	0	100	0,406
202	381	1	99,74	0,609
203	502	1	99,8	0,766
204	139	0	100	0,453
205	341	0	100	0,688
206	465	1	99,78	0,734
207	146	0	100	0,5
208	356	0	100	0,609
209	558	0	100	0,859
210	168	1	99,4	0,5
211	330	0	100	0,563
212	498	0	100	0,75
213	190	1	99,47	0,531
214	406	0	100	0,969
215	583	0	100	0,922
216	137	0	100	0,516
217	398	0	100	0,671
218	579	0	100	0,875
219	216	1	99,54	0,547
220	401	1	99,75	0,656
221	522	0	100	0,813
222	228	0	100	0,562
223	387	0	100	0,657
224	633	0	100	0,937
225	183	0	100	0,469
226	356	0	100	0,563
227	476	0	100	0,719
228	595	0	100	0,891
229	108	0	100	0,297
230	319	1	99,69	0,516
231	91	0	100	0,391
232	486	0	100	0,781
233	559	0	100	0,843
234	556	0	100	0,922
235	210	0	100	0,5
236	391	0	100	0,625
237	575	0	100	0,875
238	295	0	100	0,563
239	391	0	100	0,64
240	505	0	100	0,781
241	307	0	100	0,562
242	403	0	100	0,641
243	586	0	100	0,906
244	304	0	100	0,562
245	413	1	99,76	0,656
246	443	2	99,55	0,703
247	321	2	99,38	0,579
248	388	0	100	0,625

Seq. Pair	# Matches	# False Matches	Hit Rate	Time (s)
249	520	0	100	0,797
250	333	1	99,7	0,578
251	424	1	99,76	0,672
252	606	0	100	0,906
253	345	0	100	0,594
254	463	0	100	0,828
255	537	0	100	0,812
256	351	0	100	0,578
257	411	0	100	0,671
258	622	0	100	0,922
259	332	0	100	0,562
260	447	0	100	0,703
261	459	0	100	0,703
262	82	0	100	0,203
263	280	0	100	0,468
264	228	0	100	0,406
265	114	0	100	0,281
266	333	0	100	0,547
267	498	0	100	0,75
268	183	0	100	0,391
269	469	0	100	0,735
270	453	0	100	0,703
271	295	0	100	0,516
272	410	0	100	0,672
273	584	0	100	0,891
274	354	0	100	0,563
275	417	0	100	0,656
276	336	0	100	0,687
277	487	0	100	0,765
278	647	0	100	0,953
279	604	0	100	0,907
280	778	0	100	1,219
281	616	0	100	0,906
282	834	0	100	1,219
283	669	0	100	1,109
284	879	0	100	1,438
285	729	0	100	1,125
286	873	0	100	1,328
287	764	0	100	1,109
288	868	0	100	1,219
289	603	0	100	0,922
290	526	0	100	0,781
291	242	1	99,59	0,516
292	353	0	100	0,578
293	512	0	100	0,781
294	337	1	99,7	0,547
295	433	1	99,77	0,671
296	526	0	100	0,86
297	363	0	100	0,594
298	423	0	100	0,672
299	590	0	100	0,875
300	362	0	100	0,609
301	448	0	100	0,703
302	586	0	100	0,891
303	335	0	100	0,578
304	421	0	100	0,688
305	617	0	100	0,922
306	328	0	100	0,609
307	461	0	100	0,734
308	574	0	100	0,859
309	284	0	100	0,672
310	413	1	99,76	0,672
311	540	0	100	0,812
312	662	0	100	0,984
313	235	0	100	0,547
314	389	0	100	0,641
315	290	0	100	0,594
316	392	0	100	0,656
317	581	0	100	0,875
318	724	0	100	1,125
319	289	0	100	0,563
320	402	0	100	0,656
321	266	1	99,62	0,594
322	389	0	100	0,641
323	564	0	100	1,11
324	493	1	99,8	0,765
325	237	0	100	0,547
326	387	1	99,74	0,641
327	568	0	100	0,86
328	274	0	100	0,703
329	395	0	100	0,656
330	544	1	99,82	0,844
331	221	1	99,55	0,531

Seq. Pair	# Matches	# False Matches	Hit Rate	Time (s)
332	371	0	100	0,641
333	598	1	99,83	0,922
334	206	0	100	0,547
335	355	0	100	0,61
336	394	0	100	0,64
337	218	1	99,54	0,578
338	374	0	100	0,64
339	520	0	100	0,813
340	195	0	100	0,547
341	338	0	100	0,578
342	572	0	100	0,89
343	145	0	100	0,531
344	331	0	100	0,562
345	529	0	100	0,812
346	147	0	100	0,5
347	313	0	100	0,547
348	600	0	100	0,906
349	620	0	100	1,109
350	105	0	100	0,438
351	269	0	100	0,484
352	167	1	99,4	0,578
353	339	0	100	0,578
354	539	0	100	0,843
355	719	0	100	1,11
356	132	0	100	0,5
357	315	0	100	0,531
358	138	0	100	0,468
359	310	0	100	0,61
360	518	1	99,81	0,813
361	706	0	100	1,109
362	115	0	100	0,453
363	321	0	100	0,547
364	515	0	100	0,906
365	111	0	100	0,453
366	316	0	100	0,532

Seq. Pair	# Matches	# False Matches	Hit Rate	Time (s)
367	565	0	100	0,875
368	99	0	100	0,453
369	281	0	100	0,5
370	523	0	100	0,859
371	84	0	100	0,422
372	299	0	100	0,531
373	495	0	100	0,781
374	90	0	100	0,407
375	295	0	100	0,516
376	532	0	100	0,828
377	134	0	100	0,438
378	295	0	100	0,516
379	529	0	100	0,828
380	140	0	100	0,438
381	286	0	100	0,578
382	546	0	100	0,844
383	160	0	100	0,453
384	278	0	100	0,469
385	550	0	100	0,844
386	613	0	100	0,937
387	173	0	100	0,453
388	275	0	100	0,5
389	181	1	99,45	0,5
390	305	0	100	0,532
391	522	0	100	0,828
392	631	0	100	0,968
393	141	0	100	0,453
394	297	0	100	0,531
395	571	1	99,82	0,891
396	188	1	99,47	0,453
397	341	0	100	0,578
398	575	0	100	0,875
399	231	1	99,57	0,469
400	338	0	100	0,563

Figure C.10: LSM: full record.

C.3
Comparative records

A negative number on the following table means that the respective SIFT value was higher than LSM's.

Seq. Pair	# Matches	# False Matches	Hit Rate
1	-165	0	0
2	-146	-2	0,5
3	-133	0	0
4	-155	-1	0,14
5	-144	-1	0,26
6	-134	0	0
7	-170	-5	0,7
8	-135	-3	0,71
9	-140	-2	0,32
10	-155	-2	0,26
11	-174	-1	0,08
12	-144	0	0
13	-137	-2	0,29
14	-185	-3	0,74
15	-137	-1	0,16
16	-147	0	0
17	-196	0	-0,23
18	-139	0	0
19	-164	-1	0,14
20	-215	-2	0,01
21	-145	-3	0,5
22	-157	-1	0,13

Seq. Pair	# Matches	# False Matches	Hit Rate
23	-157	-1	0,09
24	-137	-1	0,12
25	-154	-4	0,5
26	-185	-5	1,2
27	-144	-1	0,1
28	-137	-1	0,14
29	-177	-4	0,6
30	-137	-3	0,47
31	-144	-1	0,14
32	-242	-4	0,98
33	-121	-1	0,16
34	-125	-1	0,14
35	-193	-2	0,49
36	-117	-3	0,52
37	-131	-1	0,14
38	-206	-1	0,26
39	-133	-1	0,16
40	-137	-2	0,26
41	-213	-3	0,7
42	-131	-2	0,27
43	-122	-2	0,26
44	-204	-4	0,94

Seq. Pair	# Matches	# False Matches	Hit Rate
45	-120	-3	0,47
46	-153	-2	0,24
47	-227	-6	1,35
48	-129	-2	0,3
49	-152	-3	0,33
50	-143	-2	0,23
51	-225	-4	0,92
52	-138	-2	0,31
53	-224	-2	0,46
54	-136	-3	0,45
55	-153	0	-0,02
56	-136	0	0
57	-240	0	-0,26
58	-122	-2	0,32
59	-234	0	-0,32
60	-126	-2	0,3
61	-163	-2	0,19
62	-123	0	0
63	-260	-1	0,23
64	-121	-1	0,16
65	-149	-1	0,11
66	-270	-3	0,66
67	-127	0	0
68	-233	-2	0,17
69	-119	-1	0,15
70	-178	0	-0,01
71	-117	-2	0,26
72	-238	-3	0,73
73	-119	0	0
74	-232	-4	1,02
75	-117	-1	0,15
76	-160	0	0
77	-138	-1	0,12
78	-251	-1	0,23
79	-130	-1	0,17
80	-150	-1	0,1
81	-232	-2	0,5
82	-127	0	0
83	-139	-1	0,1
84	-229	-1	-0,07
85	-114	-1	0,15
86	-126	-2	0,26
87	-234	-1	0,25
88	-134	0	0
89	-145	-3	0,36
90	-232	-1	0,25
91	-111	0	0
92	-117	0	-0,03
93	-243	-1	0,29
94	-151	-1	0,17
95	-160	-1	0,12
96	-264	-1	-0,11
97	-119	0	0
98	-147	-2	0,23
99	-292	-4	0,89
100	-132	-3	0,44
101	-153	-1	0,12
102	-280	-3	0,25
103	-134	-2	0,29
104	-145	-3	0,31
105	-301	-4	0,87
106	-124	0	0
107	-130	-3	0,33
108	-279	0	0
109	-127	-1	0,15
110	-150	-2	0,2
111	-264	-3	0,69
112	-143	-2	0,25

Seq. Pair	# Matches	# False Matches	Hit Rate
113	-170	0	0
114	-136	-2	0,26
115	-232	-1	0,29
116	-123	-3	0,5
117	-152	-1	0,11
118	-332	-2	0,49
119	-127	-3	0,46
120	-117	-3	0,38
121	-267	0	0
122	-141	-1	0,15
123	-161	0	0
124	-278	-3	0,73
125	-119	-1	0,17
126	-120	0	0
127	-230	0	-0,48
128	-84	0	0
129	-120	0	0
130	-240	-5	1,51
131	-123	0	0
132	-142	-2	0,26
133	-236	-1	0,31
134	-147	-1	0,19
135	-145	0	0
136	-240	-2	0,59
137	-122	0	-0,05
138	-149	0	0
139	-268	-2	0,54
140	-112	-1	0,18
141	-116	-1	0,17
142	-282	-5	0,56
143	-136	0	0
144	-157	0	0
145	-275	-3	0,74
146	-134	-1	0,18
147	-195	0	0
148	-157	-1	0,13
149	-284	-1	0,25
150	-154	-2	0,35
151	-282	0	0
152	-168	-3	0,5
153	-213	-1	0,1
154	-162	-1	0,13
155	-293	-1	0,26
156	-135	-1	0,2
157	-190	-1	0,13
158	-265	-1	0,27
159	-138	-1	0,19
160	-187	0	0
161	-290	-2	0,51
162	-131	0	0
163	-153	0	0
164	-202	-2	0,59
165	-114	-1	0,2
166	-170	0	0
167	-219	-1	-0,42
168	-108	-1	0,23
169	-142	-2	0,31
170	-233	-3	0,89
171	-122	-1	0,22
172	-150	-1	0,15
173	-253	-1	0,3
174	-102	0	0
175	-142	0	0
176	-219	-5	1,45
177	-125	-1	0,21
178	-147	0	0
179	-229	-3	0,86
180	-118	-2	0,42

Seq. Pair	# Matches	# False Matches	Hit Rate
181	-149	0	0
182	-238	-2	0,56
183	-124	0	0
184	-139	0	0
185	-152	0	0
186	-216	-1	0,32
187	-121	0	-0,06
188	-174	0	-0,04
189	-211	-1	-0,23
190	-131	-1	0,21
191	-133	-1	0,16
192	-204	-2	-0,05
193	-113	0	0
194	-156	0	0
195	-90	-1	0,85
196	-85	0	0
197	-80	0	0
198	-90	0	0
199	-99	0	0
200	-143	0	0
201	-152	-1	0,37
202	-112	-2	0,35
203	-139	-1	0,11
204	-174	-1	0,32
205	-119	-2	0,43
206	-135	0	-0,05
207	-194	-2	0,59
208	-127	-1	0,21
209	-149	-1	0,14
210	-182	-1	-0,03
211	-107	0	0
212	-121	0	0
213	-170	-2	0,3
214	-111	0	0
215	-143	-1	0,14
216	-193	0	0
217	-126	-1	0,19
218	-135	0	0
219	-171	-1	0,06
220	-129	-1	0,13
221	-128	-3	0,46
222	-161	-2	0,51
223	-117	-2	0,4
224	-123	-1	0,13
225	-152	-3	0,9
226	-98	-1	0,22
227	-115	0	0
228	-140	0	0
229	-101	-1	0,48
230	-93	0	-0,07
231	-137	0	0
232	-129	-1	0,16
233	-144	-1	0,14
234	-138	0	0
235	-123	-2	0,6
236	-86	0	0
237	-139	-2	0,28
238	-104	-3	0,75
239	-110	-2	0,4
240	-121	-1	0,16
241	-106	0	0
242	-99	-1	0,2
243	-144	-1	0,14
244	-101	-2	0,49
245	-113	-2	0,33
246	-106	0	-0,08
247	-96	0	-0,14
248	-95	-2	0,41

Seq. Pair	# Matches	# False Matches	Hit Rate
249	-106	-1	0,16
250	-104	0	-0,07
251	-100	0	-0,05
252	-132	0	0
253	-91	-2	0,46
254	-104	-2	0,35
255	-123	-2	0,3
256	-100	-2	0,44
257	-117	-2	0,38
258	-125	-2	0,27
259	-99	-2	0,46
260	-104	-2	0,36
261	-106	-2	0,35
262	-70	-5	3,29
263	-100	-1	0,26
264	-92	-2	0,62
265	-90	-1	0,49
266	-94	0	0
267	-142	0	0
268	-83	-1	0,38
269	-114	0	0
270	-106	-1	0,18
271	-106	-1	0,25
272	-107	-2	0,39
273	-130	-2	0,28
274	-88	-3	0,68
275	-101	-1	0,19
276	-112	-3	0,67
277	-114	0	0
278	-149	-1	0,13
279	-135	0	0
280	-176	0	0
281	-135	-1	0,13
282	-161	0	0
283	-151	0	0
284	-194	0	0
285	-155	-1	0,11
286	-193	0	0
287	-170	-1	0,11
288	-186	0	0
289	-127	-1	0,14
290	-108	0	0
291	-112	0	-0,13
292	-87	-2	0,45
293	-122	-1	0,16
294	-95	-1	0,16
295	-78	0	-0,03
296	-114	-1	0,16
297	-101	-1	0,22
298	-98	0	0
299	-118	-1	0,14
300	-92	-3	0,66
301	-92	-2	0,37
302	-131	-2	0,28
303	-103	-4	0,91
304	-115	-3	0,56
305	-158	-3	0,39
306	-129	-4	0,88
307	-113	-2	0,35
308	-135	-1	0,14
309	-119	-2	0,5
310	-111	-1	0,14
311	-139	-1	0,15
312	-178	-1	0,12
313	-157	-3	0,77
314	-96	0	0
315	-137	-2	0,47
316	-121	0	0

Seq. Pair	# Matches	# False Matches	Hit Rate
317	-137	-3	0,42
318	-150	-1	0,11
319	-122	-3	0,73
320	-116	-1	0,19
321	-156	-2	0,33
322	-129	0	0
323	-148	0	0
324	-152	0	-0,04
325	-145	-3	0,79
326	-102	0	-0,06
327	-147	-1	0,14
328	-154	-1	0,23
329	-115	-2	0,39
330	-129	-1	0,12
331	-159	0	-0,19
332	-121	-1	0,2
333	-133	0	-0,03
334	-169	-5	1,33
335	-113	-2	0,43
336	-124	-3	0,58
337	-165	-3	0,58
338	-111	0	0
339	-135	0	0
340	-180	-4	1,07
341	-116	-1	0,22
342	-143	-1	0,14
343	-200	-3	0,87
344	-101	0	0
345	-137	-2	0,3
346	-186	-2	0,6
347	-121	0	0
348	-141	0	0
349	-148	0	0
350	-180	-4	1,4
351	-103	0	0
352	-208	0	-0,33
353	-107	-2	0,45
354	-134	0	0
355	-164	-2	0,23
356	-199	-3	0,91
357	-109	-1	0,24
358	-199	-2	0,59

Seq. Pair	# Matches	# False Matches	Hit Rate
359	-120	-2	0,47
360	-132	-1	0,12
361	-183	0	0
362	-199	0	0
363	-107	0	0
364	-146	0	0
365	-205	-3	0,95
366	-98	-2	0,48
367	-168	-1	0,14
368	-208	-2	0,65
369	-112	-1	0,25
370	-165	0	0
371	-205	-2	0,69
372	-112	-2	0,49
373	-146	-2	0,31
374	-199	-2	0,69
375	-116	-4	0,97
376	-157	-1	0,15
377	-173	0	0
378	-108	-3	0,74
379	-149	-1	0,15
380	-174	-2	0,64
381	-114	-2	0,5
382	-175	-1	0,14
383	-169	-2	0,61
384	-100	0	0
385	-169	-2	0,28
386	-203	-1	0,12
387	-152	-2	0,62
388	-107	0	0
389	-170	-1	0,02
390	-127	-1	0,23
391	-198	0	0
392	-229	-1	0,12
393	-176	-2	0,63
394	-137	-1	0,23
395	-185	-1	0,08
396	-140	-4	0,99
397	-135	-4	0,84
398	-169	-1	0,13
399	-126	0	-0,15
400	-113	0	0

Figure C.11: SIFT-LSM: full record.

D
Video sequences

Video sequences used for assessing the performance of the tracking algorithms. The squares switch colour to indicate that a new database has been released.

Notice: it may take this document a while to load the movies...

D.1
Surgical procedure

Surgical procedure: SLS Variant.

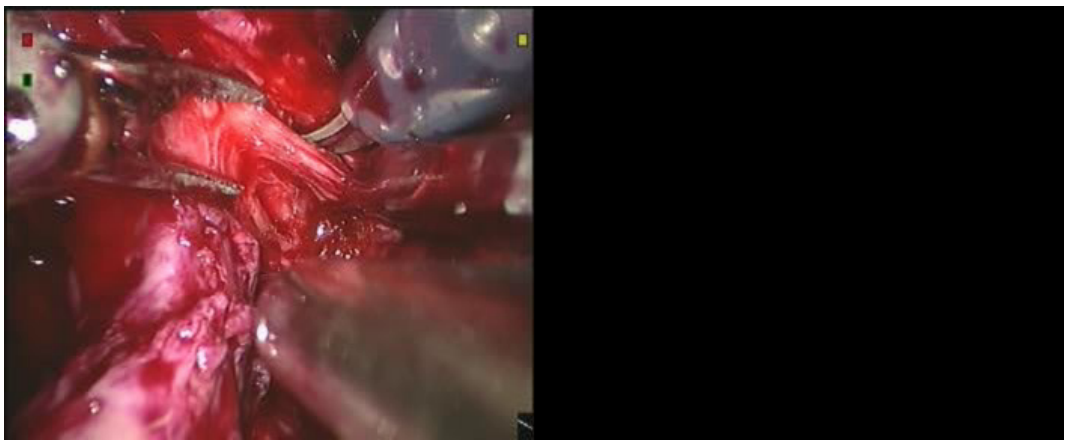

Figure D.1: Surgical procedure: click to play.

D.2
High movement pre-amp

High movement pre-amp: SLS Variant.

Figure D.2: High movement pre-amp: click to play.

D.3
Low movement pre-amp

Low movement pre-amp: SLN Variant.

Figure D.3: Low movement pre-amp: click to play.

D.4 Pool

Pool: SLN Variant.

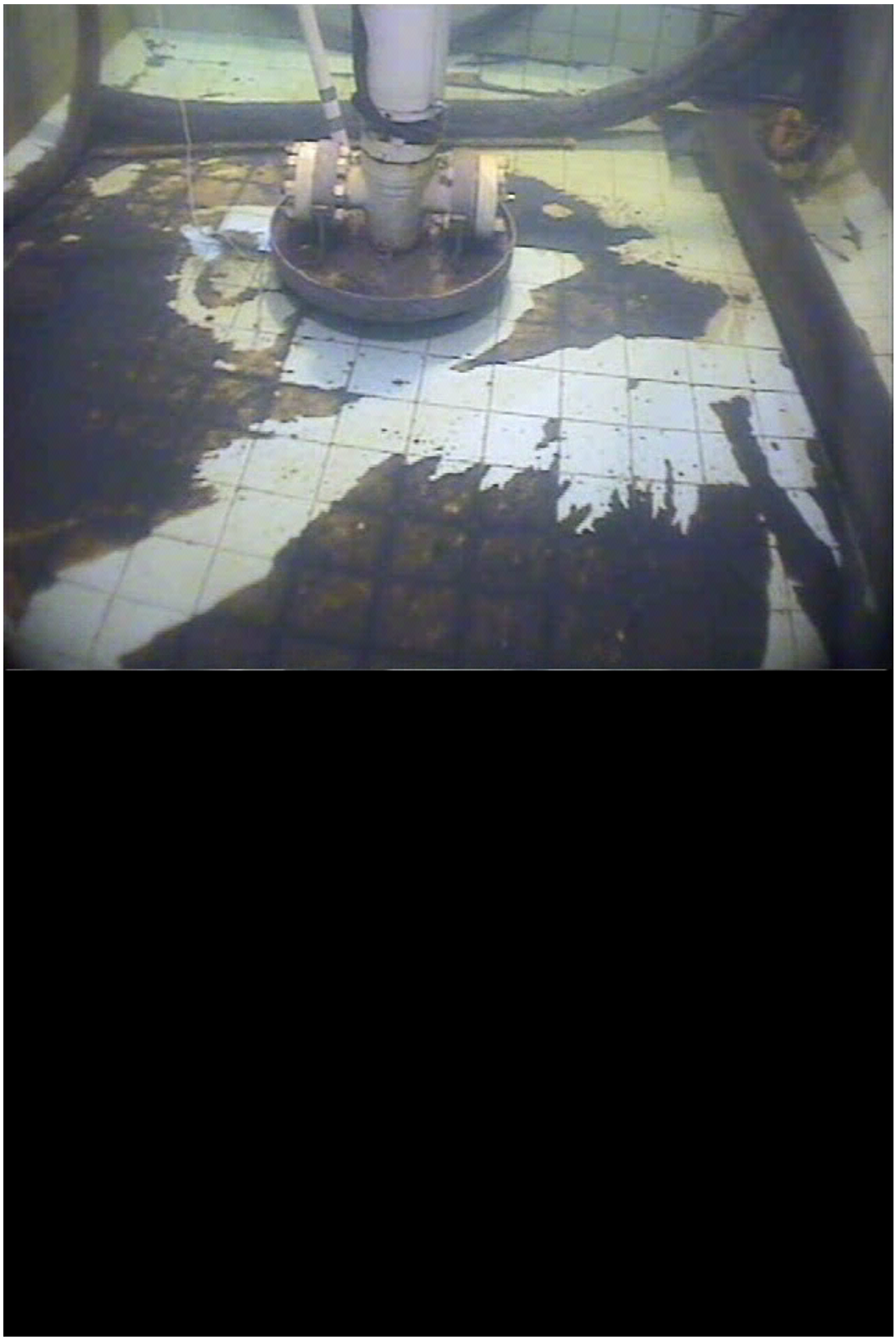

Figure D.4: Pool: click to play.