



Intestinal Content Detection in Capsule Endoscopy

Using Robust Features

Memòria del Projecte Fi de Carrera
d'Enginyeria en Informàtica
realitzat per

Harald Reingruber

i dirigit per

Fernando Vilariño

Bellaterra, 26 de Juny de 2009



El sotasignat, ***FERNANDO VILARIÑO***

Professor/a de l'Escola Tècnica Superior d'Enginyeria de la UAB,

CERTIFICA:

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per en

HARALD REINGRUBER

I per tal que consti firma la present.

Signat:

Bellaterra, 26 de Juny de 2009

Contents

Abstract	iii
1 Introduction	1
1.1 Project Overview	1
1.2 Project Objective	1
1.3 Task list	2
1.4 Project Environment	2
1.5 Hardware/Software Environment	3
1.6 Capsule Endoscopy	3
1.7 Document Structure	6
2 Robust Features in Digital Images	7
2.1 Definitions	7
2.2 SIFT	8
2.2.1 Feature detector	8
2.2.2 Feature descriptor	10
2.3 SURF	10
2.3.1 Feature detector	11
2.3.2 Feature descriptor	12
2.4 Harris-affine	13
2.4.1 Feature detector	13
2.4.2 Feature descriptor	14
2.5 Hessian-affine	14
2.5.1 Feature detector	14
2.5.2 Feature descriptor	15
2.6 Key points assumed to be detected	15
2.7 Summary	16
3 Methods and Materials	18
3.1 Bubble detector	18
3.1.1 Point detector	18
3.1.2 Bubble classifier	18
3.2 Feature Analysis Tool	19

3.2.1	Dual image viewer	19
3.2.2	Directory image browser	19
3.2.3	Feature detector and descriptor selection	19
3.2.4	Feature detection and visualization	20
3.2.5	Descriptor visualization	20
3.2.6	Parameter configuration	21
3.2.7	Feature matching	22
3.2.8	Capsule endoscopy mask	23
3.2.9	Bubble classifier	23
3.2.10	Ground truth validation	24
3.2.11	Point tool	24
3.2.12	Polygon tool	24
3.3	Implemented MATLAB programs	25
3.3.1	Feature extraction	25
3.3.2	SVM classifier training	25
3.3.3	Performance evaluation	25
3.4	Third-party libraries used	25
4	Evaluation and Results	27
4.1	Qualitative Evaluation	27
4.2	Quantitative Evaluation	28
4.2.1	Experimental Setup	29
4.2.2	Precision-Recall Graphs vs. ROC-Curves	31
5	Conclusion	35
5.1	Final conclusion	35
5.2	Project Difficulties	35
5.2.1	Motion Analysis	36
5.2.2	Ground truth Validation of Feature Detectors	36
5.3	Personal Experience of the Author:	37
	Bibliography	38

Abstract

This work covers two aspects. First, it generally compares and summarizes the similarities and differences of state of the art feature detector and descriptor and second it presents a novel approach of detecting intestinal content (in particular bubbles) in capsule endoscopy images.

Feature detectors and descriptors providing invariance to change of perspective, scale, signal-noise-ratio and lighting conditions are important and interesting topics in current research and the number of possible applications seems to be numberless. After analysing a selection of in the literature presented approaches, this work investigates in their suitability for applications information extraction in capsule endoscopy images. Eventually, a very good performing detector of intestinal content in capsule endoscopy images is presented.

A accurate detection of intestinal content is crucial for all kinds of machine learning approaches and other analysis on capsule endoscopy studies because they occlude the field of view of the capsule camera and therefore those frames need to be excluded from analysis.

As a so called “byproduct” of this investigation a graphical user interface supported Feature Analysis Tool is presented to execute and compare the discussed feature detectors and descriptor on arbitrary images, with configurable parameters and visualized their output. As well the presented bubble classifier is part of this tool and if a ground truth is available (or can also be generated using this tool) a detailed visualization of the validation result will be performed.

Chapter 1

Introduction

The following sections provide a brief overview of the project, its surroundings and how it was organized. The following chapters later on will go more into detail of the established concepts, the implementation and eventually the evaluation and interpretation of the achieved results.

1.1 Project Overview

The detection of intestinal content is a crucial task in many applications of capsule endoscopy analysis. Intestinal content (e. g. food in digestion, turbid liquid or bubbles) may occlude the capsules camera field of view, hence disturb the detection of certain events. Therefore its necessary accurately detect frames containing intestinal content to be able to exclude them from further analysis.

This work focusses on the detection of bubbles in capsule endoscopy frames, and presents a novel approach of detecting bubbles by the use of state of the art feature detectors and descriptors combined with a machine learning classification method. In order to

1.2 Project Objective

The two main objectives of this work are:

- First, analysing and qualitatively evaluate the characteristics of 4 different feature detectors and their appropriate descriptors. This work focusses on *Scale Invariant Feature Transform* (SIFT) [4], *Speeded Up Robust Features* (SURF) [2], *Hessian-affine* and *Harris-affine* [7] feature detectors.
- Second, it should be analysed if this feature detectors are feasible for detecting events in capsule endoscopy images, in particular intestine

content like bubbles, and evaluate the detection performance qualitatively and quantitatively.

For repeatability of the concluded results and testing the provided approaches on arbitrary data, a Graphical User Interface (GUI) which offers configurable parameters shall be provided. The final goal is to present a proof of concept, to demonstrate if the implemented ideas hold in general and how well they perform. A future line of research could be implementing the concepts in an “application ready” state.

1.3 Task list

The following list is an overview of the task which have been successfully implemented during this project:

- Investigation in literature of state of the art feature detector and descriptor techniques.
- Identification of availability of third-party libraries which can be used for this investigation.
- Development of a common interface for the different feature detection and description libraries.
- Analysis of the different characteristics of the feature detectors and descriptors and development of a flexible Graphical User Interface with configurable parameters for support of this task.
- Analysis of the images of a capsule endoscopy sample study and identification of good features to detect.
- Setup of a training and validation data set for the training of a Support Vector Machine classifier to classify bubble or non-bubble descriptors.
- Evaluation and visualization of the classifier performance.

1.4 Project Environment

This work originated during the authors exchange semester called European Region Action Scheme for the Mobility of University Students (ERASMUS) programme, a European student exchange programme, at the *Computer Vision Center* (Centre de Visó per Computador) of the Universitat Autònoma de Barcelona. The project was supervised by *Fernando Luis Vilariño Freire*, Ph.D. who always knew the right questions to ask for helping me finding the solution to present technical or conceptional problems and I am very grateful

for him showing satisfaction of achieved interim result, while encouraging me striving for more at the same time.

The Computer Vision Center (CVC) is a notable research center investigating in many different state of the art issues of computer vision and image processing. As medical imaging is an important area of modern computer vision investigations, one branch of the CVC is devoted to medical imaging projects with the aim of innovate and further develop software solutions in this area.

The project was scheduled for a time period of four month (March to June) and positioned as a side project of current investigations on machine learning approaches for intestinal motility assessment with capsule endoscopy (comp. [15]). The result of this investigation should provide a further tool for deriving information or detecting scenes of interest in capsule endoscopy analysis.

1.5 Hardware/Software Environment

Hardware environment: The following hardware was used to execute the implemented software tools:

- 1 GHz 32-bit (x86) processor
- 1 GB of system memory
- 200 MB available hard disk space

Software environment: The following software was used to to execute the implemented software tools:

- Windows XP
- Matlab R2007
- Cygwin (including the module libpng12)

1.6 Capsule Endoscopy¹

Capsule endoscopy is a novel imaging technique which allows the visualization of the whole intestinal tract. The capsule endoscopy procedure consists of the ingestion of a capsule, with a complete visualization device attached to it, which registers a video movie of its trip throughout the gut. This video is emitted by radio frequency and recorded into an external device carried by

¹comp. the introduction in [15]

the patient. Once the study is finished, the final record can be easily downloaded into a PC with the appropriate software for its posterior analysis by the physicians.

Recently, several works have tested the performance of capsule endoscopy in multiple clinical studies. Some of these clinical scenarios include intestinal polyposis and the diagnosis of small bowel tumors, obscure digestive tract bleeding, Crohn's disease and small bowel transplant surveillance. Some authors have analyzed the validity of capsule endoscopy for detection of small bowel polyps in hereditary polyposis syndromes, concluding the clinical value of the methodology for the familial adenomatous polyposis and that capsule endoscopy could be used as a first line surveillance procedure for Peutz-Jeghers syndrome. Following a different line of research, other studies have focused their efforts on the evaluation of the clinical effectiveness of capsule endoscopy in the management of patients with obscure digestive tract bleeding, concluding that capsule endoscopy seems best suited for patients with obscure gastrointestinal bleeding who have undergone inconclusive standard evaluations and in whom the distal small bowel (the portion beyond the reach of a push enteroscope) needs to be visualized. In this direction, comparative studies have been published showing the main advantages and drawbacks of capsule endoscopy in comparison with push enteroscopy, sonde enteroscopy and intraoperative endoscopy in this kind of pathologies. Capsule endoscopy has also demonstrated to be highly effective for diagnosing several pathologies associated with Crohn's disease which are frequently missed by conventional tests. Other researchers have reported capsule endoscopy to be a helpful tool for post-transplant surveillance of small intestine. Capsule endoscopy displayed post-transplant changes in the villi that ranged from blunted white villi seen at day 20 to normal villi observed at 6 months. A more exhaustive review and summary are referenced in [15].

Device description Capsule endoscopy was first developed and introduced by *Given Imaging Limited* under the trade mark of M2A Given Diagnostic Imaging System – M2A is the acronym for “mouth to anus” – being cleared for marketing for the first time through the U.S. government on August 1, 2001. This technology is performed by means of three main components: the capsule, the registration device and the proprietary data analysis software.

The capsule is an ingestible device equipped with all the suitable technology for image acquisition, including illumination lamps and radio frequency emission. Figure 1.1 shows a graphical scheme of the capsule together with the distribution of its components in scale. It consists of an external envelope with a transparent dome front sizing 11x30 mm (1), which contains a lens holder (2) with one lens (3), four illuminating leds (4), a complementary metal oxide silicon (CMOS) image sensor (5), a battery (6), an



Figure 1.1: The M2A© camera (left) and the camera components (right) [15].

application-specific integrated circuit (ASIC) transmitter (7), and a micro-antenna (8). The field of view of the lens spans 140-degree, very similar to that of standard endoscopy. The illuminating lamps are low consume white-light emitting diodes (LED). The video images are transmitted using UHF-band radiotelemetry to aerials taped to the body which allow image capture, and the signal strength is used to calculate the position of the capsule in the body. Synchronous switching of the LEDs, the CMOS sensor and the ASIC transmitter minimize power consumption, which lets the emission of high-quality images at a frame ratio of 2 frames per second during 6 hours. The capsule is completely disposable and does not need to be recovered after use, being expelled by the body 10 to 72 hours after ingestion.

The registration device consists of a set of aerial sensors for the RF signal reception, connected to a CPU with a hard disc for data storage. The registration device is carried by the patient fastened into a belt, altogether with a battery for power supply. The aerial sensors are taped to the body of the patient, forming an antenna array which collects the signal transmitted by the capsule and sends it to the receiver. The received data is subsequently processed and stored in the data storage by the CPU.

The proprietary software is installed on a PC workstation. It allows the physicians to retrieve the data from the recorder and to transfer it to the workstation for additional processing and visualization on the display. The performed study can be stored independently on the workstation hard disk or be copied on a CD, a DVD or any other storage device, being ready for visualization and annotation on any computer in which the displaying software has been previously installed.

1.7 Document Structure

This document is structured into 5 chapters. After this introduction into the topic and the objectives of this work a summary of feature detectors and descriptors analysed during this investigation will follow. Later on the methods and materials implemented will be described and after the presentation of the results and their evaluation the final conclusion will be stated.

Chapter 2

Robust Features in Digital Images

This chapter provides a detailed description of the used feature detectors and their appropriate descriptors. It will summarize what kind of region properties are crucial regarding the different approaches of feature detection. The following sections provide a definition of feature, feature detection, and feature description; a glimpse of the SIFT, SURF, Hessian-affine, and Harris-affine feature detector process steps and how their descriptors are derived.

2.1 Definitions

Feature and feature detection: Generally speaking there is no exact definition what a feature is, but in terms of digital image processing a feature often signifies an “interesting” part of an image, hence it strives to detect local images structures which, as an example, could be used to detect and maybe match in other images. As there is a big diversity of possible applications there are also different types of image features which can be detected. The following list will note the most important image features in digital image processing:

- Edges
- Corners
- Blobs (also called regions of interest)

The feature detection process is the process calculating the exact location of this features, and permitting only the most robust feature points to be detected. Robustness, depending on the actually chosen approach, can signify invariance to noise, change in perspective, change in scale and change in contrast respectively lighting conditions. Rejecting non-robust feature points is usually implemented by the use of a threshold.

Remark that corner and blob features are often generalized by the term *interest points*, that's why in literature often the term interest point detector is used instead of feature detector.

Feature description: As soon a feature is detected, hence the location determined, the feature can be described. The feature description process is also often referred to as feature extraction. Therefore characteristics of the local surroundings of the detected point are extracted and the information combined it into a feature vector, which denotes the eventual feature descriptor.

As feature detection and feature description are not necessarily connected, usually a feature detector can be combined with an arbitrary feature descriptor and vice versa. Although, sometimes it might be useful to use a combination presented together to take advantage of synergies in their calculation, hence saving calculation time. The calculation of the integral image in the SURF detector and descriptor is a good example for this.

2.2 SIFT

This section summarizes the approach of feature detection and description presented in [4] at which it is referred to for more detail.

2.2.1 Feature detector

Base of the SIFT detector is the Difference of Gaussian (DoG) function. DoG operates in scale-space, hence enables scale-invariance. The following subsections enumerate the different steps in the key point detection process.

Detection of scale-space extrema

A image pyramid based on the DoG function is generated where each point is defined by a 3 dimensional vector with the elements x, y, σ . A single DoG image (respectively a single level of the DoG image pyramid) $D(x, y, \sigma)$ is given by

$$D(x, y, \sigma) = L(x, y, k_i\sigma) - L(x, y, k_j\sigma)$$

where $L(x, y, k\sigma)$ is the convolution of the original image $I(x, y)$ with the Gaussian filter kernel $G(x, y, k\sigma)$ at scale $k\sigma$, which concludes

$$L(x, y, k\sigma) = G(x, y, k\sigma) * I(x, y)$$

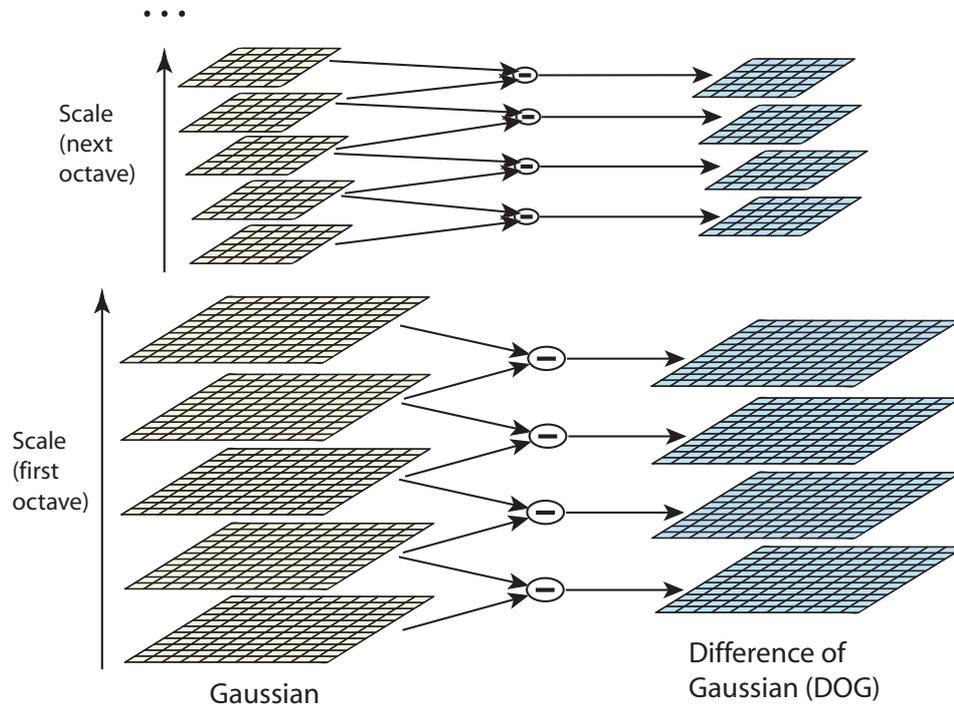


Figure 2.1: The left part shows the Gaussian filtered images with increasing σ and on the right side the differences of each two adjacent Gaussian images are displayed [4].

Hence, a DoG image between the scales $k_i\sigma$ and $k_j\sigma$ is the difference of the Gaussian filtered images at scales $k_i\sigma$ and $k_j\sigma$. The computation of the DoG image pyramid is illustrated in Figure 2.1.

Each value of the DoG pyramid is later compared with its 8 neighbour values of the same scale level (σ) and with the 9 neighbour values of the scale below and above the current scale (illustrated in Figure 2.2). If this value is a maximum or a minimum in this range it is selected as key point candidate.

These extrema represent blobs at the scale corresponding to their actual size.

Eliminating low contrast extrema

Extrema with low-contrast are very sensitive to noise and therefore considered unstable. First, the location and value of the extrema of each candidate key point will be interpolated. This interpolated DoG value can be used to reject these unstable extrema if the absolute value is below a certain threshold. A threshold of 0.03 is used in the experiments implemented in [4].

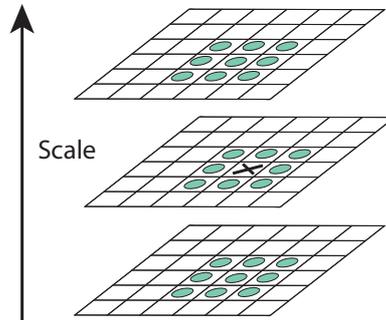


Figure 2.2: The maxima and minima of the DoG image pyramid are detected by comparing each value to its 26 neighbours in the 3×3 regions of the current and adjacent scales [4].

Eliminating edge responses

The DoG function deliver also high response along edges. This is considered unstable because the location along edges is only poorly determined (e. g. non-geometrical edges originated from perspective borders).

This edge response can be rejected by calculating the ratio between the curvature across the edge, which is high, and the perpendicular curvature along the edge, which is low. The eigenvalues of the Hessian matrix are proportional to the principal curvatures, hence can be used to calculate the ratio. All candidate key points with a ratio *exceeding* a certain threshold will be rejected. 10 is used as the edge-ratio threshold in [4].

2.2.2 Feature descriptor

Base of the descriptor is the gradient magnitude and orientation at the region around each key point. The gradients of the area will be computed for a 16×16 sample array corresponding to the detected scale and orientation of each point. Then the gradients are weighted by a Gaussian window to decrease the influence of gradients far from the center.

Afterwards this gradients are accumulated in a orientation histogram with 8 histogram bins (representing 8 orientations) for each region of 4×4 subregions. The computation of the SIFT feature descriptor is illustrated in Figure 2.3.

2.3 SURF

This section summarizes the feature detection and description approach presented in [2] at which it is referred to for more detail.

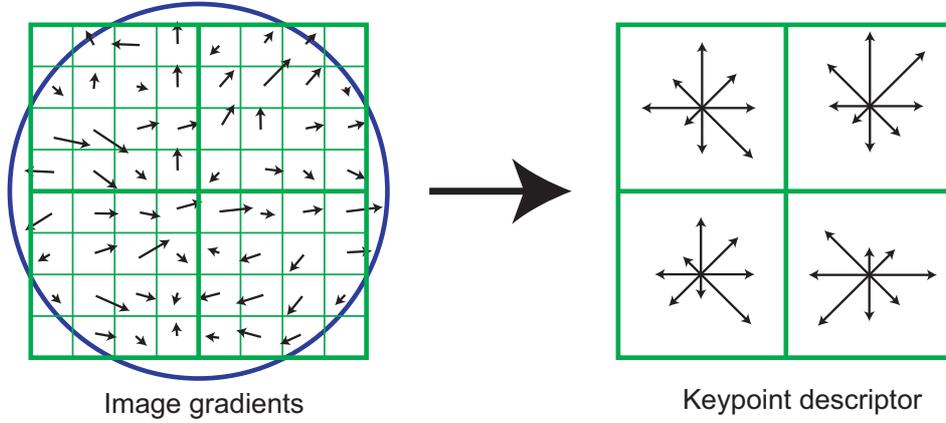


Figure 2.3: The left part displays the image gradients surrounding the detected point. These gradients weighted by a Gaussian window (indicated by the circle) and are accumulated in a gradient orientation histogram for each of 4×4 subregions (right part). This figure shows only a 2×2 subregions descriptor computed from a 8×8 array of samples, whereas the default SIFT implementation uses 4×4 subregions computed from a 16×16 sample array. [4]

2.3.1 Feature detector

The SURF feature detector uses the determinant of the Hessian-Matrix for detecting points of interest. Given a point $\mathbf{x} = (x, y)$ in an image I , the Hessian matrix $\mathcal{H}(\mathbf{x}, \sigma)$ in \mathbf{x} at scale σ is defined as

$$\mathcal{H}(\mathbf{x}, \sigma) = \begin{bmatrix} L_{xx}(\mathbf{x}, \sigma) & L_{xy}(\mathbf{x}, \sigma) \\ L_{xy}(\mathbf{x}, \sigma) & L_{yy}(\mathbf{x}, \sigma) \end{bmatrix}$$

where $L_{xx}(\mathbf{x}, \sigma)$ is the convolution of the Gaussian second order derivative $\frac{\partial^2}{\partial x^2}g(\sigma)$ with the image I in point \mathbf{x} , and similarly for $L_{xy}(\mathbf{x}, \sigma)$ and $L_{yy}(\mathbf{x}, \sigma)$.

As Gaussian filters for the estimation of the second partial derivatives of the Hessian-Matrix are non ideal in any case, the Fast-Hessian detector pushes the simplification of the approximation even further and uses box filters. Because of this simplification, the approximation of the Hessian-Matrix can be very performant implemented using integral images. Hessian-Matrix based detectors are used for detecting blob-like structures in images. The comparison between Gaussian second order partial derivatives and the approximation using box filters are illustrated in Figure 2.4.

For detecting extrema in scale space, the Fast-Hessian detector utilizes the fact that by using integral images, the filter masks for calculating Haar-wavelet responses may be of arbitrary size with equal calculation cost. Therefore the filter masks are increased in size gradually, each filter mask size corresponding to a certain sigma in scale space.

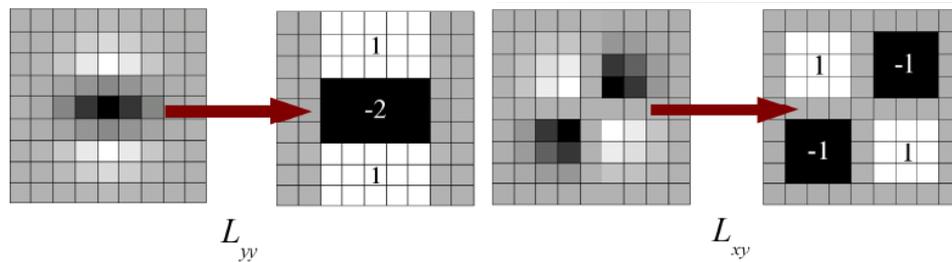


Figure 2.4: The two images on the left side display the Gaussian second order partial derivatives in y -direction and xy -direction. On the right side the approximations using box filters are displayed [2].

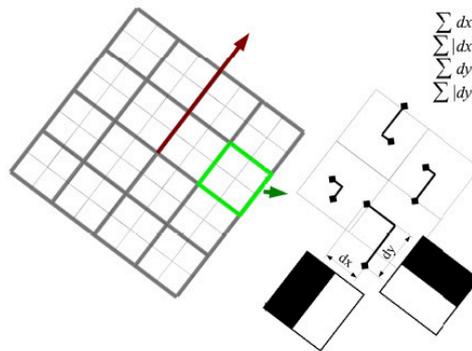


Figure 2.5: The SURF descriptor is made up by 4×4 subregions (left) where each contain the sum of wavelet responses and the sum of absolute wavelet responses in x and y direction (right). *Source:* <http://www.cse.cuhk.edu.hk/~lyu/seminar/06fall/Wyman.ppt>

Blob responses (the absolute value of the Hessian-Determinant) lower than the threshold parameter are rejected.

2.3.2 Feature descriptor

Similar as SIFT the SURF descriptor uses a 4×4 subregion around the detected key point. Then it calculates the Haar wavelet responses for 5×5 regularly spaced sample points within each subregion in x and y direction. For each subregion a descriptor consisting of the sum of wavelet responses in each direction and the sum of their absolute values. Hence, each subregion has a four dimensional vector consisting of $\mathbf{v} = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$ which describes the dominant orientation and results in a 64 element feature vector for each key point. The structure of the 4×4 array and the concept of the subregion descriptor are visualized in Figure 2.5 and Figure 2.6 provides more insight in the significance of the descriptor entries.

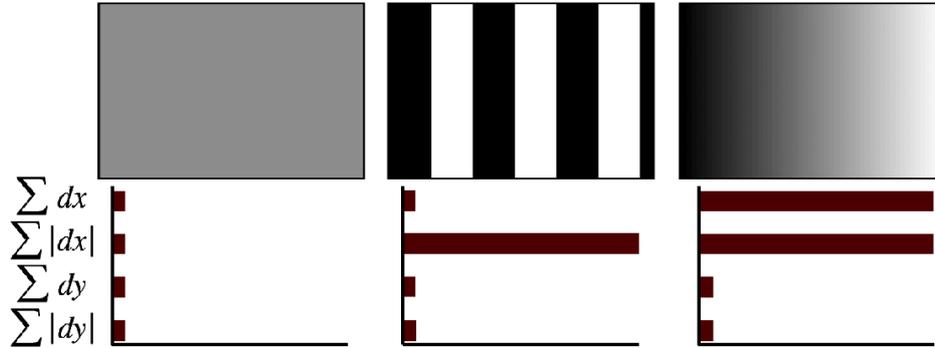


Figure 2.6: The descriptor entries of a subregion represent the underlying intensity pattern. In case of a homogeneous region (left) all values are low. In presence of frequencies in x -direction (middle), the value of $\sum |d_x|$ is high, but all others remain low. If the intensity is gradually increasing (right) in x -direction, both values $\sum d_x$ and $\sum |d_x|$ are high [2].

Again, the integral image (which was already calculated during the detection step) can be used for a performant calculation of the Haar wavelet responses.

2.4 Harris-affine

This section summarizes the Harris-affine feature detection and description approach presented in [7] at which it is referred to for more detail.

2.4.1 Feature detector

The Harris-affine feature detector is basically a Harris corner detector extended by scale and affine invariance. The Harris corner measure uses the second moment matrix (also called autocorrelation matrix or structure tensor) which is constructed of the partial derivatives in x and y direction.

Given a point $\mathbf{x} = (x, y)$ in an image I , the second moment matrix $\mathcal{A}(\mathbf{x})$ in \mathbf{x} is defined as

$$\mathcal{A}(\mathbf{x}) = \sum_{x,y} w(x, y) \begin{bmatrix} I_x^2(\mathbf{x}) & I_x I_y(\mathbf{x}) \\ I_x I_y(\mathbf{x}) & I_y^2(\mathbf{x}) \end{bmatrix}$$

where I_x and I_y are the respective derivatives (of pixel intensity) in the x and y direction. The off-diagonal entries are the product of I_x and I_y , while the diagonal entries are squares of the respective derivatives.

The applied weighting function $w(x, y)$ is typically a circular Gaussian to average the values while increasing the weight of central values at the same time.

The eigenvalues of the second moment matrix describes how the autocorrelation measure change in space, hence it's principal curvature. The Harris corner detector detects points locally maximizing this eigenvalues, but as the calculation of eigenvalues is very costly (e. g. with the singular value decomposition method) it rather uses an approximation based on the determinant and the trace of the second moment matrix:

$$R = \det(A) - \alpha \text{trace}^2(A) = \lambda_1 \lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$$

where α is a constant. Corner points have large, positive eigenvalues and therefore a large Harris measure.

This principal idea is further expanded by introducing the Gaussian scale-space dimension by choosing the scale $\sigma_I^{(k+1)}$ that maximizes the Laplacian-of-Gaussians (LoG) over a predefined range of neighboring scales, and applying affine shape adaption. Points with a Harris measure under a certain Threshold are rejected.

Furthermore the implementation of [7] rejects points on edges, which are detected by the use of a Canny edge detector.

More details can be found in [7].

2.4.2 Feature descriptor

In this study, the Harris-affine feature detector is combined with the SIFT feature descriptor. For more information see the description given in section 2.2.

2.5 Hessian-affine

This section summarizes the Hessian-affine feature detection and description approach presented in [7] at which it is referred to for more detail.

2.5.1 Feature detector

Whereas the Harris-affine detector is based on the second moment matrix, the Hessian-affine detector relies on the Hessian matrix (as the name supposes) which is constructed of the second-order partial derivatives.

Given a point $\mathbf{x} = (x, y)$ in an image I , the Hessian matrix $\mathcal{H}(\mathbf{x})$ in \mathbf{x} is defined as

$$\mathcal{H}(\mathbf{x}) = \begin{bmatrix} L_{xx}(\mathbf{x}) & L_{xy}(\mathbf{x}) \\ L_{xy}(\mathbf{x}) & L_{yy}(\mathbf{x}) \end{bmatrix}$$

where $L_{xx}(\mathbf{x})$ and $L_{yy}(\mathbf{x})$ are second partial derivative in the x and y direction and $L_{xy}(\mathbf{x})$ is the mixed partial second derivative in the x and y directions.

Interest points are points which maximizes simultaneously the determinant and the trace of the Hessian matrix (at each scale):

$$\begin{aligned}\det &= \sigma_I^2(L_{xx}L_{yy}(\mathbf{x}) - L_{xy}^2(\mathbf{x})) \\ \text{trace} &= \sigma_I(L_{xx} + L_{yy})\end{aligned}$$

where σ_I is the Gaussian scale factor.

As the Hessian-affine detector uses the Hessian matrix for key point detection, it also detects Blob-like structures in images.

The introduction of scale, affine invariance and rejection of points located on edges works similar as for the Harris-affine detector, see [7] for more details.

2.5.2 Feature descriptor

As mentioned already for the Harris-affine detector, the Hessian-affine feature detector is combined with the SIFT feature descriptor. For more information see the description given in section 2.2.

2.6 Key points assumed to be detected

SIFT: As mentioned in section 2.2, the SIFT feature detector delivers key points which are:

- representing blobs
- of arbitrary scale i. e. size
- of high contrast (depending on threshold)
- not part of edges (depending on edge-ratio threshold)

Remark that because of the DoG function, extrema in higher scale level can be detected with insensitivity regarding to high-frequent noise.

SURF: As mentioned in section 2.3, the SURF feature detector delivers key points which are:

- representing blobs
- of arbitrary scale i. e. size
- high Hessian determinant (depending on threshold)

Table 2.1: Feature detector overview

	<i>SIFT</i>	<i>SURF</i>	<i>Harris-affine</i>	<i>Hessian-affine</i>
Structure detecting:	Blobs	Blobs	Corners	Blobs
Detector base:	DoG extrema	Hessian determinant	Second moment matrix det. and trace	Hessian det. and trace

Harris-affine: As mentioned in section 2.4, the Harris-affine feature detector delivers key points which are:

- representing corners
- of arbitrary scale i. e. size
- high Harris measure (depending on threshold)
- not part of edges (depending on high and low canny threshold)

Hessian-affine: As mentioned in section 2.5, the Hessian-affine feature detector delivers key points which are:

- representing blobs
- of arbitrary scale i. e. size
- high Hessian determinant and trace (depending on threshold)
- not part of edges (depending on high and low canny threshold)

2.7 Summary

The table 2.1 provides an overview of the similarities and differences of the presented feature detectors and 2.2 provides an overview of the similarities and differences of the presented feature descriptors.

Table 2.2: Feature descriptor overview

	<i>SIFT</i>	<i>SURF</i>
Descriptor base:	Local gradients	Haar-wavelet responses
Default dimension:	128	64
Subregions:	4x4	4x4
Subregion feature:	Gradient orientation histogram	Sums of wavelet responses in x/y direction

Chapter 3

Methods and Materials

This chapter describes the software and scripts implemented during this project and the third-party libraries which have been discovered and identified to be suitable for the purpose of this project. The main product of this work are the implemented bubble detector and the Feature Analysis Tool, supporting a Graphical User Interface (GUI). The following sections provide more details.

3.1 Bubble detector

During the analysis of the capsule endoscopy sample study the author noticed that the presented interest point detector are responding very well when bubbles are present in the images. As the presence of intestinal content – like bubbles – is absolutely disturbing for further machine learning applications on capsule endoscopy data sets, the accurate detection of them is a crucial step in this applications. Therefore, a Support Vector Machines (SVM) classifier has been implemented for detecting them.

3.1.1 Point detector

All of the feature detectors respond generally well to the presence of bubbles, but a qualitative perceptual analysis showed that the Harris-affine feature detector delivers the most accurate results. The Harris corner measure seems ideal for the detection of the specular reflection of the bubbles, that's why the Harris-affine detector is the tool of choice for our classifier.

3.1.2 Bubble classifier

The descriptors of the detected points are later used as the input of the bubble classifier. To establish a accurate and generalizable classifier, a good training set is crucial. In order to set up a good training set for the SVM

classifier the capsule endoscopy sample study all frames containing approximately more than 90 % of bubble descriptors, from the almost 50,000 frames of the sample study. The Support Vector Machines should be robust enough to deliver a performant classifier in spite of about 10 % falsely labeled non-bubble descriptors in the selected bubble frames. To complete the training set a equal amount of non-bubble frames are randomly selected from the sample study. As the feature vector for each sample is of high dimensionality (128 dimensions for SIFT and 64 dimension for SURF) a great amount of training samples are necessary to train a good performing classifier. This bubble classifier is also part of the Feature Analysis Tool which will be introduced in the following section.

The details about the classifier training and testing will be stated in chapter 4.

3.2 Feature Analysis Tool

The in MATLAB implemented Feature Analysis Tool is a GUI which provides a general feature detector and descriptor analysis interface for the discussed detectors and descriptors, as well as special functions for capsule endoscopy images like the bubble classifier. This enables very easy repeatability of in this work presented figures and results, as well as testing and analysing different capsule endoscopy data or arbitrary images.

3.2.1 Dual image viewer

To provide a nice possibility of matching the descriptors of two images, both images can be loaded and visualized at the same time and configured separately (see Figure 3.1). This enables to step-by-step approximate the desired thresholds of the chosen detector for each image before matching them. It can be selected to which of the two axes the following operations are applied to, or if they shall be applied on both axes.

3.2.2 Directory image browser

For analysing a big amount of data the possibility is offered to select a directory as default image directory, and pre-fetch the file list. Later the images of the directory can be stepped through comfortably with the keyboard cursor. Click “Select Img Dir...” for choosing a default image directory and click “Fetch Img Dir” to pre-fetch the directory file list.

3.2.3 Feature detector and descriptor selection

Following combinations of detectors and descriptors can be selected (see Figure 3.2):

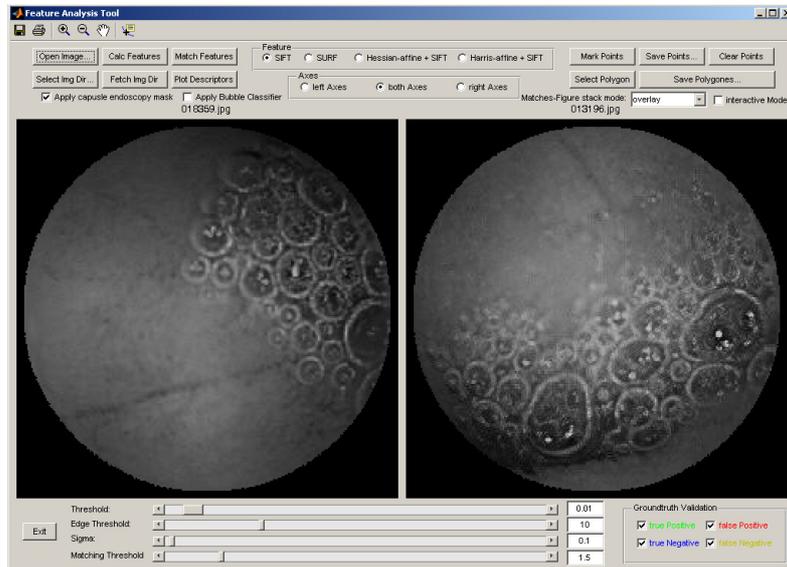


Figure 3.1: Feature Analysis Tool: dual image viewer

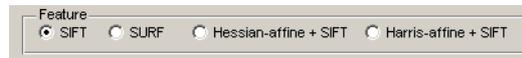


Figure 3.2: Feature detector and descriptor selection

- SIFT detector and descriptor
- SURF detector and descriptor
- Hessian-affine detector and SIFT descriptor
- Harris-affine detector and SIFT descriptor

3.2.4 Feature detection and visualization

By clicking “Calc Features” features will be detected and their descriptors calculated. Figure 3.3 illustrates an example how the detected points are visualized. The interest point detection can be configured with different parameters, and additionally the capsule endoscopy mask or bubble classifier can be applied.

3.2.5 Descriptor visualization

By clicking the “Plot Descriptors” button a visualization of the SIFT descriptors will be performed (see Figure 3.4). For SIFT descriptor visualization it’s

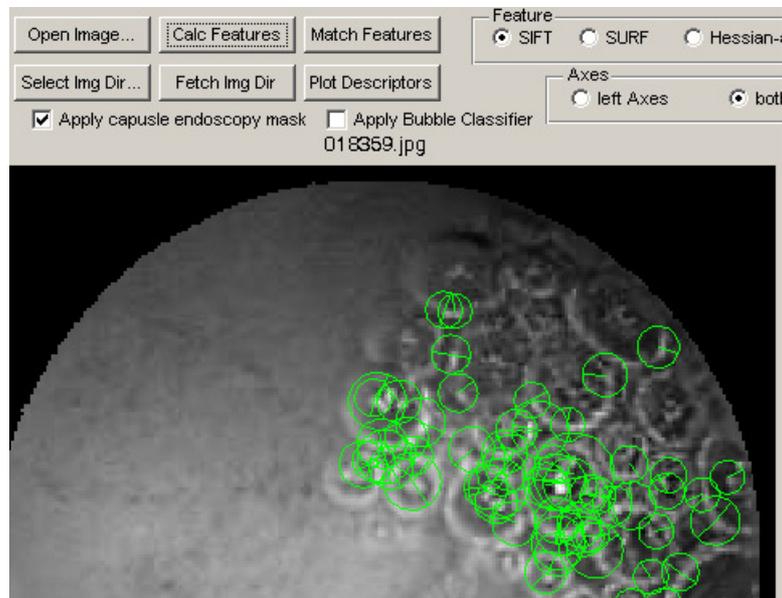


Figure 3.3: The detected points will be visualized as circles with radius of their calculated scale (σ). The line from the center displays the orientation of the detected point.

recommended to use a high threshold in order to generate only a few interest points. Otherwise the active image will get very cluttered and the descriptors visualized are not distinguishable. This feature works only with SIFT descriptors, it will deliver undefined results if SURF is activated.

3.2.6 Parameter configuration

Three detection parameters and one matching parameter can be configured (see 3.5).

Threshold: This parameter configures the threshold of the currently active feature detection algorithm.

Edge Threshold: This parameter configures the SIFT edge ratio threshold, and for Hessian-affine and Harris-affine the upper Canny edge detector threshold. This parameter is irrelevant for the SURF detector.

Sigma: This parameter configures the Sigma (σ) of the Gaussian filter kernel for smoothing the image before applying the feature detection algorithm. This parameter can be useful to reduce noise in images.

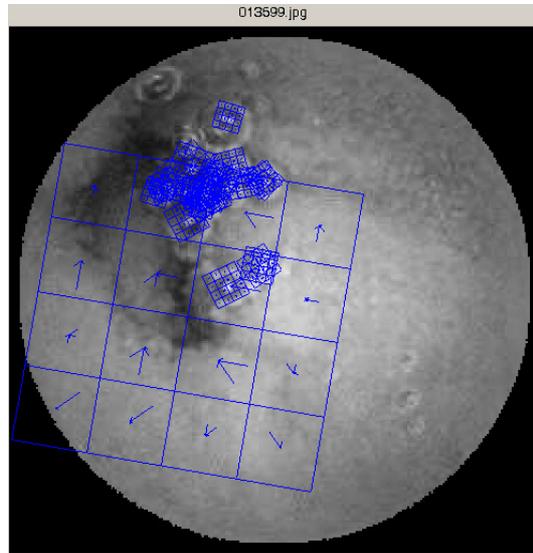


Figure 3.4: SIFT descriptor visualization

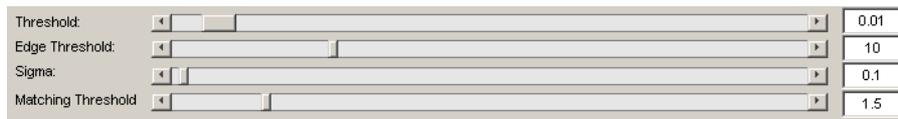


Figure 3.5: Parameter configuration

Matching Threshold: This parameter configures the distance threshold for the descriptor matching algorithm. The matching algorithm of the SIFT implementation [4] is used here, where the threshold is a ratio value between the closest neighbour and the second closest neighbour. If the ratio is below the chosen threshold the will not be matched with another point.

3.2.7 Feature matching

By clicking “Match Features”, the (before calculated) descriptors of both images will be matched against the ones from the other image. The result will be displayed in a new figure (see Figure 3.6). Depending what is selected in the “Matches-Figure stack mode” drop-down menu, if *overlay* is selected, the two images are linearly combined and the matching points plotted with points and lines. The options *horizontal*, *vertical* and *diagonal* are selected, the two images are composited like the selection and again the matching points are plotted as line and points.

If the checkbox “interactive Mode” is activated, the detected points in the Matches-Figure can be highlighted by clicking on the points (on the points,

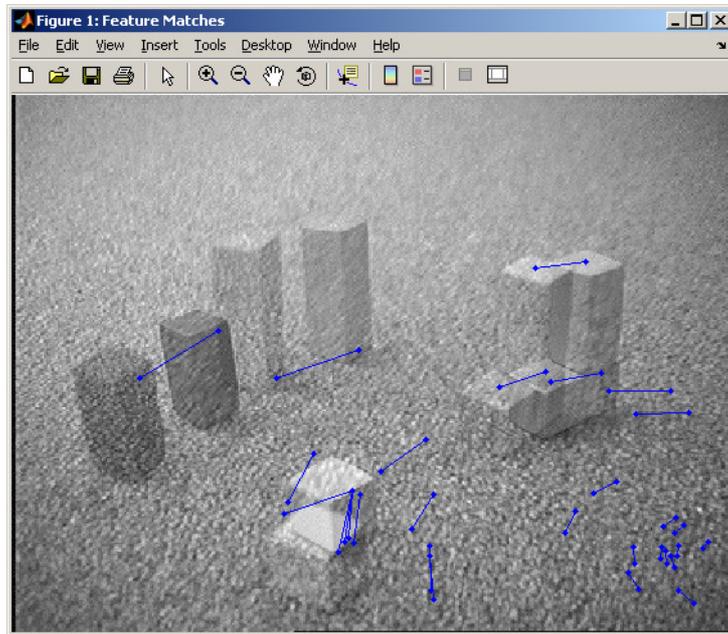


Figure 3.6: This example shows the result of the feature matching in overlay mode, where the two images are linearly combined. The blue lines represent the calculated matches.

clicking on the connected lines is effectless). This feature can be useful for very cluttered results. The interactive Mode will be terminated by pressing Enter on the keyboard.

3.2.8 Capsule endoscopy mask

For capsule endoscopy images only the inner circular region of the 255x255 px image is significant. By activating this checkbox, a mask is applied for rejecting all calculated interest point outside the mask.

3.2.9 Bubble classifier

By activating this checkbox, the bubble classifier will be applied when the “Calc Feature” button will be pressed. Bubble points will be plotted green, and non-bubble points will be plotted in red (see Figure 3.7).

If there exists a MAT-file with the image filename plus “.mat”, containing a polygon(s) representing the bubble region, the ground truth validation will be executed, depending on which layers of the ground truth validation are activated.

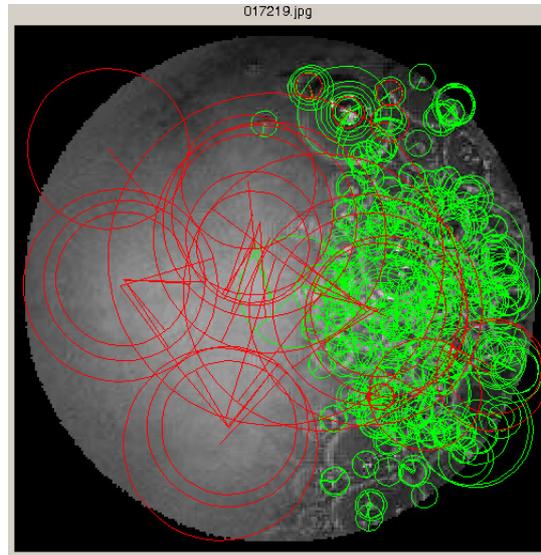


Figure 3.7: Result of the applied bubble classifier. Descriptors classified as bubbles are displayed in *green* and points classified as non-bubbles in *red*.

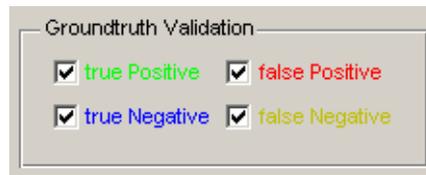


Figure 3.8: Ground truth evaluation layers

3.2.10 Ground truth validation

This four checkboxes activate and deactivate the layers of the ground truth validation (see Figure 3.8). If the ground truth validation will be executed, the classified points will be plotted in the color of the corresponding checkbox label.

3.2.11 Point tool

This tool enables selecting a arbitrary number of points in the current image, and save their coordinates to a MAT-file. The selection tool will be terminated by pressing Enter on the keyboard.

3.2.12 Polygon tool

This tool enables drawing a arbitrary number of polygons in the current image, and save their coordinates to a MAT-file.

3.3 Implemented MATLAB programs

The feature extraction, training set construction and SVM classifier training steps are implemented in several MATLAB scripts. Also the performance evaluation was implemented as a script, so it's repeatability is ensured.

3.3.1 Feature extraction

The feature extraction step for the training and test set was separated from the classifier training step (instead of extracting the features on demand) to keep the computation time of the classifier training at a minimum. Due to this, the training can be repeated with different parameters without the necessity of extracting the features all over again.

Details on the training and test set generation will be stated in chapter 4.

3.3.2 SVM classifier training

The most challenging part of the classifier training was to handle the balance of on the one hand providing the training with enough samples to generate a generalizable and accurate classifier, but on the other hand keeping the training computation time at a tolerable amount. The author decided to use all available bubble frames from the sample study, but sub-sampling them in a factor 1:10. This would provide a better coverage regarding the statistical distribution of bubble descriptors in case bubble descriptors of one and the same frame are somehow correlated. Therefore, it seems to be a good idea using all the frames but only a sub-sampled part of their descriptors, instead of selecting only a part of all the bubble frames and using all their descriptors.

3.3.3 Performance evaluation

A comprehensive evaluation of the classifier performance is the final achievement of this work. The author decided that the evaluated numbers are more significant if they are evaluated separately on bubble frames and non-bubble frames. In this manner, better insight about the accuracy of the classifier on frames without containing bubbles (which are usually the great majority) is provided.

The results of the evaluation will be summarized in chapter 4.

3.4 Third-party libraries used

This section gives an overview of the discovered third-party libraries which have been identified to be suitable for the purpose of this project.

SIFT for Matlab [14]: This is a MATLAB/C implementation of the SIFT detector and descriptor. It is customizable and features a decomposition of the algorithm in several reusable M and MEX files. This implementation produces interest points and descriptors which are very similar to David Lowe's [5] implementation.

<http://www.vlfeat.org/~vedaldi/code/sift.html>

SURFmex [13]: SURFmex is a MATLAB interface for the Speeded Up Robust Features interest point detector, written by Petter Strandmark. The original C++ library is available from the authors' webpage [1].

This is a very simple MATLAB library for Windows, written in C and C++, which provides an interface to the SURF DLL and a simple routine to match descriptors.

<http://www.vision.ee.ethz.ch/~surf/>

Harris & Hessian [6] Scale & affine invariant feature detectors used in Mikolajczyk CVPR06 and CVPR08 for object class recognition. Efficient implementation of both, detectors and descriptors. Currently only sift descriptor was tested with the detectors but the other descriptors should work as well. Package contains a PCA basis for projections on fewer number of dimensions. Run with PCA projection and take as many SIFT dimensions as you wish. Run without options for help. Includes windows executable which requires cygwin.

<http://www.robots.ox.ac.uk/~vgg/research/affine/detectors.html#binaries>

OSU SVM Toolbox for MATLAB [11] OSU SVM is a Support Vector Machine (SVM) toolbox for the MATLAB numerical environment. The toolbox is used to create models for regression and classification using support vector machines.

<http://sourceforge.net/projects/svm/>

Precision-Recall and ROC Curves [12] Calculate and plot P/R and ROC curves for binary classification tasks.

<http://www.mathworks.com/matlabcentral/fileexchange/21528>

Chapter 4

Evaluation and Results

In the following sections the results of the performance evaluation of the presented classifier are explained.

4.1 Qualitative Evaluation

To qualitatively evaluate the bubble detector 2 test images (which were not part of the classifier training) are used to demonstrate the accuracy of the classifier based on the different feature detectors. See Figure 4.1, Figure 4.2, Figure 4.3 and Figure 4.4. Correct detected bubbles are displayed in green, and falsely claimed bubbles (false positives) are displayed in red.

This experiments clearly displays that the accuracy (also from the localization of the bubbles) of the classifier based on the *Harris-affine* feature detector outmatches the other ones.

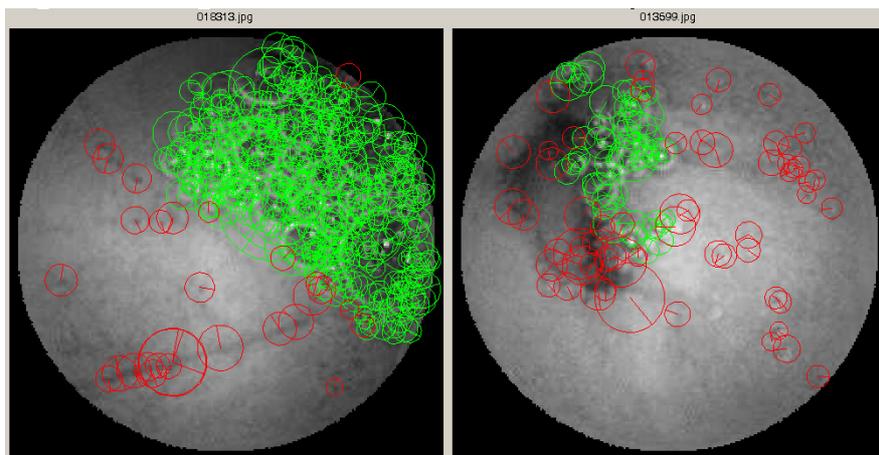


Figure 4.1: Validation of two bubble frames using the SIFT detector based classifier.

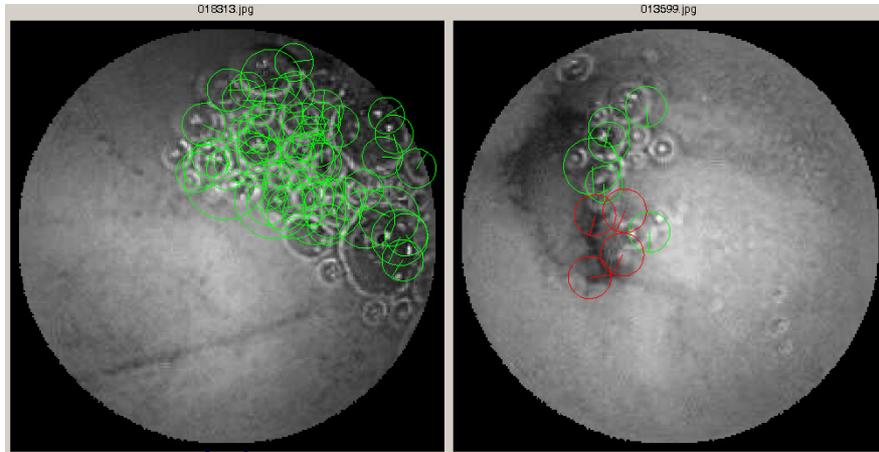


Figure 4.2: Validation of two bubble frames using the SURF detector based classifier.

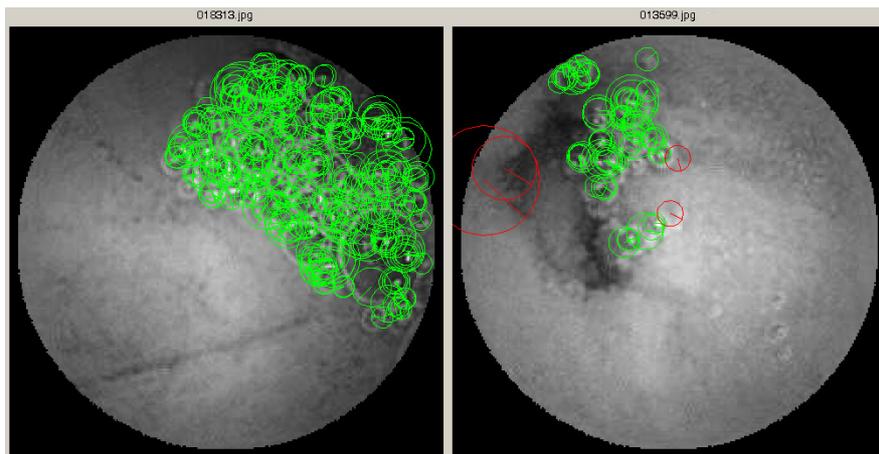


Figure 4.3: Validation of two bubble frames using the Harris-affine detector based classifier.

4.2 Quantitative Evaluation

This experiment evaluates the success-rate, respectively verifies the feasibility of a Support Vector Machines (SVM) classifier which was trained to detect bubbles in capsule endoscopy frames. The sample points used for classification are detected by four considered feature detector of this work and the sample vector is composed by extracting the appropriate descriptor (SIFT or SURF) from those sample points.

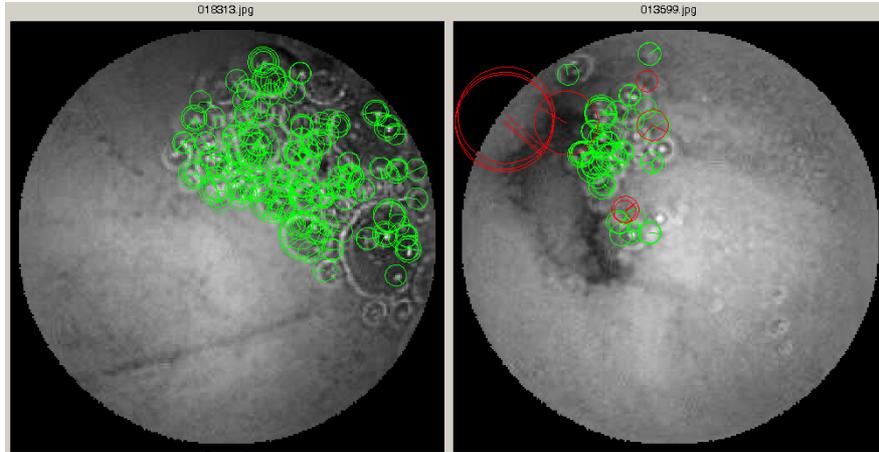


Figure 4.4: Validation of two bubble frames using the Hessian-affine detector based classifier.

4.2.1 Experimental Setup

In order to present the accuracy of the bubble classifier it is evaluated with a cross-validation approach, which means that the set of data is separated into subsets, one subset used for the training of the classifier and the other one used for validating the classifier respectively the generalization ability (performance on unseen data).

Take into account that the objective of this project is primarily a proof of concept. In this experiment only the images of a single patient study has been used for the analysis. For a “application ready” state it would be necessary using data of several patient studies of different patients. Also increasing the number of training and test frames and performing the cross-validation on several different training and test sets is recommended.

Training set: These sample points were detected by using the considered feature detector. Of these points feature descriptor vectors were extracted for the later use for the training of a SVM classifier. The bubble frames represent 90 frames, manually selected from a capsule endoscopy study, which show the most dense content of bubbles. The 90 non-bubble frames are randomly selected from the same study, and frames containing bubbles were manually removed. As less key points are detected in non-bubble frames and for bubble frames a high percentage of real bubble descriptors is desired different thresholds are used for the extraction of bubble and non-bubble frames. Table 4.1 contains the figures to this data set. The number of samples is given by the number of frames and in brackets the number of points. The number of bubble samples is obtained under the assumption that all the points in a bubble frame are associated to bubbles and all the points in a non-bubble

Table 4.1: Training set

	<i>SIFT</i>	<i>SURF</i>	<i>Harris-affine</i>	<i>Hessian-affine</i>
Number of samples:	180 (7960)	180 (2218)	180 (11337)	180 (12611)
Bubble samples:	90 (3843)	90 (1133)	90 (5608)	90 (5946)
Non-bubble samples:	90 (4117)	90 (1085)	90 (5729)	90 (6665)
Threshold bubble frames:	0.02	2	200	120
Threshold non-bubble frames:	0.007	0.4	10	20

Table 4.2: SVM parameters used in the classifier training.

<i>Support Vector Machines parameters</i>	
Kernel type:	Radial Basis Functions (RBF)
Degree:	1
Gamma:	0.01
Coefficient:	1
C:	1
Epsilon:	0.001
SVM Type:	c-SVC
loss-tolerance:	0.1
shrinking:	0

frame are associated to non-bubbles. Table 4.2 lists the parameters used for the training of the SVM classifier.

Test set (Bubble frames): These sample points are again obtained using the considered feature detector and their appropriate descriptor as the feature vector. Table 4.3 contains the figures to this data set. Number of samples are given by the number of frames and in brackets the number of points. The number of bubble samples is obtained under the assumption that all the points within a manually selected groundtruth area are associated to bubbles and all the points outside this area are associated to non-bubbles-

Test set (Non-Bubble frames): These sample points are again obtained using the considered feature detector and their appropriate descriptor as the feature vector. Table 4.4 contains the figures to this data set. Number of samples are given by the number of frames and in brackets the number of points under the assumption that all the points of non-bubble frames are

Table 4.3: Test set (Bubble frames).

	<i>SIFT</i>	<i>SURF</i>	<i>Harris-affine</i>	<i>Hessian-affine</i>
Number of samples:	50 (29918)	50 (8180)	50 (32986)	50 (61031)
Bubble samples:	22391	5352	21935	44524
Non-Bubble samples:	7527	2828	11051	16507
Threshold	0.007	0.4	10	20

Table 4.4: Test set (Non-Bubble frames).

	<i>SIFT</i>	<i>SURF</i>	<i>Harris-affine</i>	<i>Hessian-affine</i>
Non-Bubble samples:	50 (21407)	50 (5588)	50 (30418)	50 (34070)
Threshold	0.007	0.4	10	20

associated to non-bubbles.

Result: Table 4.5 shows the evaluation of the classification on the test set of bubble frames and table 4.6 shows the evaluation of the classification on the test set of non-bubble frames.

4.2.2 Precision-Recall Graphs vs. ROC-Curves

To get more insight in the performance of the classifiers, the Precision-Recall Graphs and ROC-Curves of the Classifiers using the 4 different feature detectors are provided in this section.

See Figure 4.5, Figure 4.6, Figure 4.7 and Figure 4.8.

Table 4.5: Result of classification on bubble frames test set.

	<i>SIFT</i>	<i>SURF</i>
Number of samples:	29918 (100.00%)	8180 (100.00%)
True positive:	15110 (50.50%)	3500 (42.79%)
True negative:	3555 (11.88%)	1518 (18.56%)
False positive:	3972 (13.28%)	1310 (16.01%)
False negative:	7281 (24.34%)	1852 (22.64%)
Precision:	0.7918	0.7277
Recall/Sensitivity:	0.6748	0.6540
Specificity:	0.4723	0.5368

	<i>Harris-affine</i>	<i>Hessian-affine</i>
Number of samples:	32986 (100.00%)	61031 (100.00%)
True positive:	16851 (51.09%)	35847 (58.74%)
True negative:	6669 (20.22%)	8936 (14.64%)
False positive:	4382 (13.28%)	7571 (12.41%)
False negative:	5084 (15.41%)	8677 (14.22%)
Precision:	0.7936	0.8256
Recall/Sensitivity:	0.7682	0.8051
Specificity:	0.6035	0.5413

Table 4.6: Result of classification on non-bubble frames test set.

	<i>SIFT</i>	<i>SURF</i>
Number of samples:	21407 (100.00%)	5588 (100.00%)
True negative:	18798 (87.81%)	5237 (93.72%)
False positive:	2609 (12.19%)	351 (6.28%)

	<i>Harris-affine</i>	<i>Hessian-affine</i>
Number of samples:	30418 (100.00%)	34070 (100.00%)
True negative:	27954 (91.90%)	31079 (91.22%)
False positive:	2464 (8.10%)	2991 (8.78%)

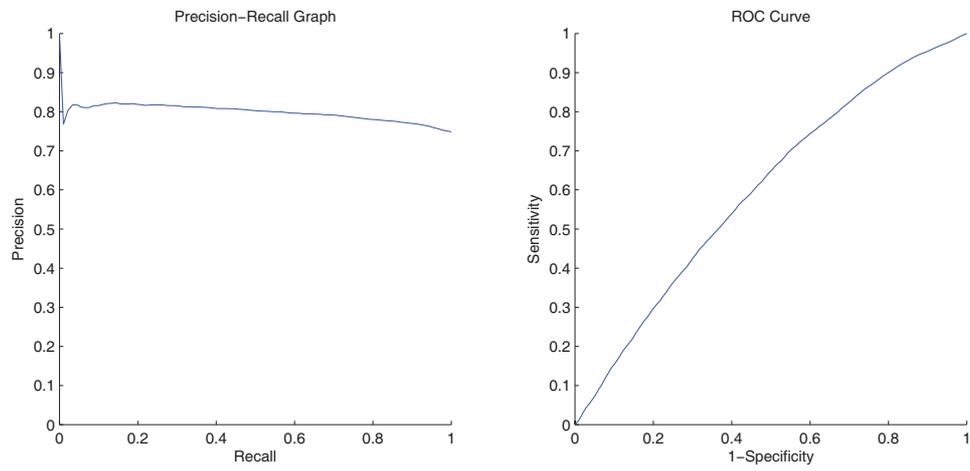


Figure 4.5: PR-graph vs. ROC-curve: Bubble classifier using SIFT detector

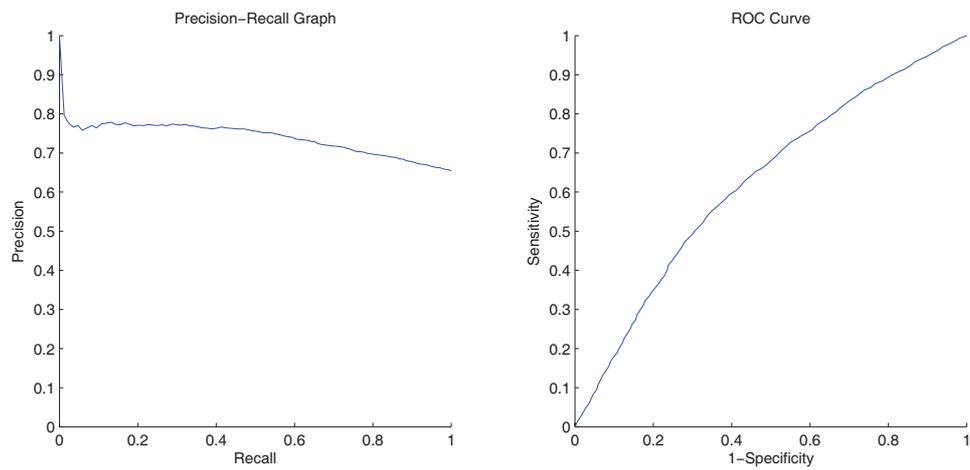


Figure 4.6: PR-graph vs. ROC-curve: Bubble classifier using SURF detector

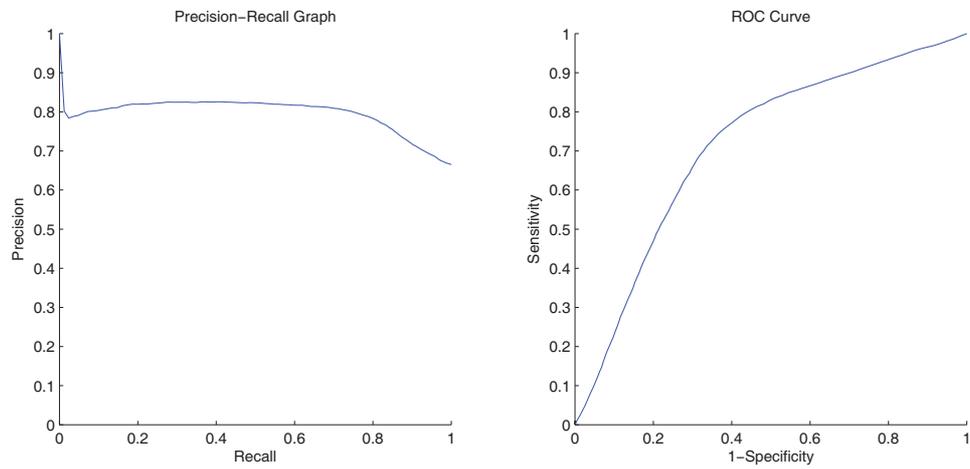


Figure 4.7: PR-graph vs. ROC-curve: Bubble classifier using Harris-affine detector

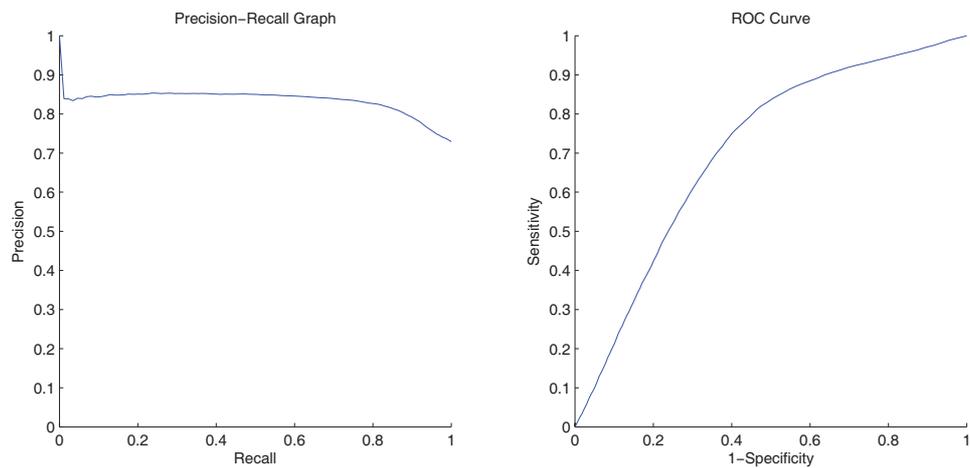


Figure 4.8: PR-graph vs. ROC-curve: Bubble classifier using Hessian-affine detector

Chapter 5

Conclusion

This chapter provides a short interpretation of the result and a final conclusion.

5.1 Final conclusion

Finally, it is concluded that the *Harris-affine* feature detector is the best choice as a basis for the bubble classifier. A precision of 0.79 with a recall of 0.76 and 91.9 % correctness on the classification of non-bubble frame descriptors is a very good result which only can be outmatched by the Hessian-affine detector. Additionally, the PR-Graphs and ROC-Curves confirm the good overall performance of the classification using this detector (except again, Hessian-affine outmatches the Harris-affine detector).

But it has to be emphasized that the performed evaluation is only test of the performance of the classifier on the input descriptors. In the qualitative evaluation it has been shown that the detection of bubble points is more accurate using the Harris-affine detector.

In a nutshell, the combination of the Harris-affine feature detector and the SVM classifier leads to a very good performing bubble detector.

5.2 Project Difficulties

As typical for research projects the initial objectives are not always certain if they are achievable with the available resources. During the authors investigations occurred as well several unexpected discoveries which led to the necessity of adapting the initial goals. The following sections are a brief summary of occurred difficulties during this work.

5.2.1 Motion Analysis

It was also tested deriving motion analysis by matching features between adjacent frames in capsule endoscopy videos. After deeper investigation and assessment of feature detection in capsule endoscopy frames the author concluded that regular frames containing the intestine wall and maybe the lumen (opening or closing) provide too little (or almost none) features to track, and if there are some they are not robust hence not useful for matching.

5.2.2 Ground truth Validation of Feature Detectors

For providing a representable evaluation of the feature detectors taken into account, different approaches of ground truth validation has been considered. The author experienced that establishing a ground truth is not as simple as it might appear, and it was concluded that it is feasible but exceeds the possibilities of this project and not scope of this project. Although some approaches appear a little less extensive but exhibit severe drawbacks. Therefore a qualitative analysis of the used feature detectors was considered sufficient. The following paragraphs will provide examples of considered ground truth validation approaches.

Ground truth based on image homography: In [8, 9] ground truth is established by selecting manually a homography in two images which differ only by a affine transformation. The two big drawbacks of this approach are that this is only feasible for planar scenes (or scenes which can be considered planar because of great distance between camera and object) and because of the necessity of selecting the homography manually it's hardly possible to evaluate the performance on a big test image set for statistical robustness (as displayed in [10] and [3]).

Ground truth based on trifocal geometry: In [3] a trifocal tensor is used to reconstruct the affine transformation in 3D space and therefore providing ground truth where each point in one image should appear in the transformed. To deduct the trifocal tensor corresponding image points of the original and the transformed images are necessary. As shown in [10] because of this the evaluation result based on trifocal geometry is biased and also only feasible (without enormous effort) for a handful test images.

Ground truth based on epipolar geometry: [10] overcomes the mentioned drawbacks by using stereoscopic vision and epipolar geometry, which means reconstructing the 3D coordinates of the objects used for testing. In [10] 100 3D objects are used for testing, placed on an automatic turn table and photographed by two high resolution cameras every 5 degree, with different focal length and varying lightning conditions.

As this evaluation is based on using geometric constraints only and feasible on a large-scale test set (as the process is fully automatable) this approach delivers the most accurate results but is also the approach with the highest effort for establishing the test data and implementing the evaluation method.

5.3 Personal Experience of the Author:

Well, without the necessity to exaggerate I really can say this project was a great experience in many aspects. I learned a lot and improved my practical skills regarding computer vision concepts like feature detection, description and matching (of course) but also got some insight in concepts of stereoscopy and epipolar-geometry and other CV topics which challenged me in a positive way during my work.

Furthermore, the practice in structured scientific investigation will be a highly valuable experience for my hopefully as interesting challenges in the future.

Finally I can say my exchange semester here in Barcelona was enriching for me technically and scientifically but also personally from what I learned by getting to know the people, costumes and traditions here in Barcelona, Catalunya and España. A valuable enlargement of my horizon.

Die Grenzen meiner Sprache bedeuten die Grenzen meiner Welt.
Ludwig Wittgenstein
österreichischer Philosoph

The limits of my language are the limits of my world.
Ludwig Wittgenstein
Austrian philosopher

Visca el Barça i visca Catalunya!

Bibliography

- [1] Bay, H., T. Tuytelaars, and L. Van Gool: *SURF*. <http://www.vision.ee.ethz.ch/~surf/>, 2006.
- [2] Bay, H., T. Tuytelaars, and L. Van Gool: *Surf: Speeded up robust features*. In *European Conference on Computer Vision*, pp. 404–417, 2006.
- [3] Fraundorfer, F. and H. Bischof: *Evaluation of local detectors on non-planar scenes*. In *Proceedings 28th workshop of the Austrian Association for Pattern Recognition*, pp. 125–132, 2004.
- [4] Lowe, D.G.: *Distinctive image features from scale-invariant keypoints*. *International Journal of Computer Vision*, 60:91–110, 2004.
- [5] Lowe, D.G.: *Demo software: SIFT keypoint detector*. <http://www.cs.ubc.ca/~lowe/keypoints/>, 2005.
- [6] Mikolajczyk, K.: *Scale & affine invariant feature detectors*. <http://www.robots.ox.ac.uk/~vgg/research/affine/detectors.html#binaries>, 2006.
- [7] Mikolajczyk, K. and C. Schmid: *Scale & affine invariant interest point detectors*. *International Journal Computer Vision*, 60(1):63–86, October 2004.
- [8] Mikolajczyk, K. and C. Schmid: *A performance evaluation of local descriptors*. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 27(10):1615–1630, 2005.
- [9] Mikolajczyk, K., T. Tinne, C. Schmid, A. Zisserman, J. Matas, F. Schafalitzky, T. Kadir, and L. Van Gool: *A comparison of affine region detectors*. *International Journal of Computer Vision*, 65(1/2):43–72, 2005.
- [10] Moreels, P. and P. Perona: *Evaluation of features detectors and descriptors based on 3d objects*. *International Journal of Computer Vision*, pp. 800–807, 2005.
- [11] *OSU SVM toolbox for MATLAB*. <http://sourceforge.net/projects/svm/>, 2003.

- [12] Schroedl, S.: *Precision-Recall and ROC curves*. <http://www.mathworks.com/matlabcentral/fileexchange/21528>, 2008.
- [13] Strandmark, P.: *SURFmex*. <http://www.maths.lth.se/matematiklth/personal/petter/surfmex.php>, 2008.
- [14] Vedaldi, A.: *SIFT for Matlab*. <http://www.vlfeat.org/~vedaldi/code/sift.html>, 2006.
- [15] Vilariño Freire, F.L.: *A Machine Learning Approach for Intestinal Motility Assessment with Capsule Endoscopy*. PhD thesis, Universitat Autònoma de Barcelona, Centre de Visió per Computador, Barcelona, June 2006.