Sequential image analysis for computer-aided wireless endoscopy

Michal Drozdzal

Sequential image analysis for computer-aided wireless endoscopy

Michal Drozdzal

Department of Applied Mathematics and Analysis

Universitat de Barcelona

A thesis submitted for the degree of

Doctor in Mathematics (PhD)

Doctoral advisor

Dr. Petia Ivanova Radeva

December 2013

Abstract

Wireless Capsule Endoscopy (WCE) is a technique for inner-visualization of the entire small intestine and, thus, offers an interesting perspective on intestinal motility. The two major drawbacks of this technique are: 1) huge amount of data acquired by WCE makes the motility analysis tedious and 2) since the capsule is the first tool that offers complete inner-visualization of the small intestine, the exact importance of the observed events is still an open issue. Therefore, in this thesis, a novel computer-aided system for intestinal motility analysis is presented. The goal of the system is to provide an easily-comprehensible visual description of motility-related intestinal events to a physician. In order to do so, several tools based either on computer vision concepts or on machine learning techniques are presented.

A method for transforming 3D video signal to a holistic image of intestinal motility, called motility bar, is proposed. The method calculates the optimal mapping from video into image from the intestinal motility point of view.

To characterize intestinal motility, methods for automatic extraction of motility information from WCE are presented. Two of them are based on the motility bar and two of them are based on frame-per-frame analysis. In particular, four algorithms dealing with the problems of intestinal contraction detection, lumen size estimation, intestinal content characterization and wrinkle frame detection are proposed and validated.

The results of the algorithms are converted into sequential features using an online statistical test. This test is designed to work with multivariate data streams. To this end, we propose a novel formulation of concentration inequality that is introduced into a robust adaptive windowing algorithm for multivariate data streams. The algorithm is used to obtain robust representation of segments with constant intestinal motility activity. The obtained sequential features are shown to be discriminative in the problem of abnormal motility characterization.

Finally, we tackle the problem of efficient labeling. To this end, we incorporate active learning concepts to the problems present in WCE data and propose two approaches. The first one is based the concepts of sequential learning and the second one adapts the partition-based active learning to an error-free labeling scheme.

All these steps are sufficient to provide an extensive visual description of intestinal motility that can be used by an expert as decision support system.

Dla Ewy i Andrzeja, moich rodziców.

Acknowledgements

First of all, I would like to express appreciation and thanks to my advisor Dr. Petia Radeva for the opportunity she gave me and for all the support she has been giving me during all these years. Thank you Petia!

I would like to express my gratitude to the members of *University of Barcelona* working in the intestine project Dr. Jordi Vitrià and Dr. Santi Seguí. Their ideas, energy and commitment have been important components of all the work in this thesis.

Spacial thanks to Dr. Fernando Azpiroz and Dr. Carolina Malagelada from *Hospital General de la Vall d'Hebron* for taking care of the medical part of this project and for showing me how difficult it is to explain well new ideas.

More broadly, I would like to mention the members of the research groups from *BCN Perceptual Computing Lab* and *Computer Vision Centre* who have been sharing with me their knowledge and expertise. Many thanks to Francesco, Santi and Pierluigi for all the discussions we had and for the time we spent together.

This thesis wouldn't be possible without the participation of GivenImaging that provided the founding and gave us the opportunity to work on the global scale.

Thanks to my family, my parents, my sisters and my grandfathers for all the inspiration and the support I have received on my way up to here. Finally, last but not least, to Adriana. Thank you all!

The best way of finding out the difficulties of doing something is to try to do it.

David Maar, Vison

Contents

List of Figures

List of Tables

1

Introduction

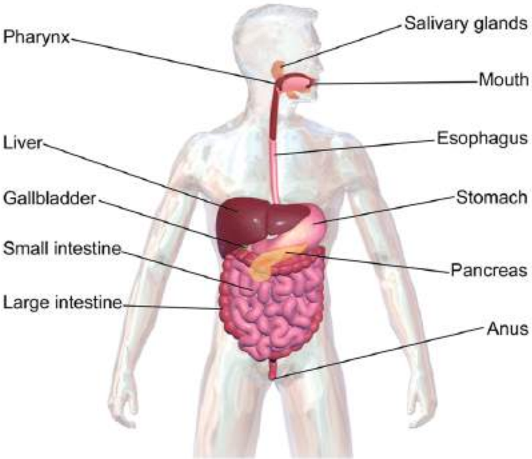


Figure 1.1: Human digestive system (*Graphics from Wikipedia*).

1.1 Intestinal motility

The digestive system is responsible for breaking down large food molecules into smaller ones and absorbing the resulting nutrients into the bloodstream. In humans, the digestive system is composed of various organs that can be splitted into two groups: 1) gastrointestinal tract, composed of mouth, esophagus, stomach, small intestine, large intestine and anus and 2) accessory organs of digestion including salivary glands, liver, gallbladder and pancreas. All these organs are presented in Figure 1.1.

The gastrointestinal tract includes all parts of the digestive system through which the food is physically passing by and is where the digestion takes place. Different organs have different function in the digestion process. First, the food enters the mouth, where it is chewed and where the saliva is released. Saliva cleans the oral cavity, moistens the food, and contains digestive enzymes that are used to start the food breaking down process. Next, the food is moved to the esophagus, a narrow muscular tube, that connects the mouth with the stomach. Stomach is made of thick muscular walls which, with the help of enzymes, reduces the size of the food into small particles that are then passed by to the small intestine. Food leaves the stomach in a semi-liquid form that is known as chyme. Small intestine is the organ, where the majority of the food digestion and absorption takes place and where the chyme is mixed with three different digestive juices: bile, pancreatic juice and intestinal juice. After that, the food enters the large intestine, where the digestion is retained long enough to allow fermentation which breaks down some of the substances that remain after the processing in the small intestine. The result of the

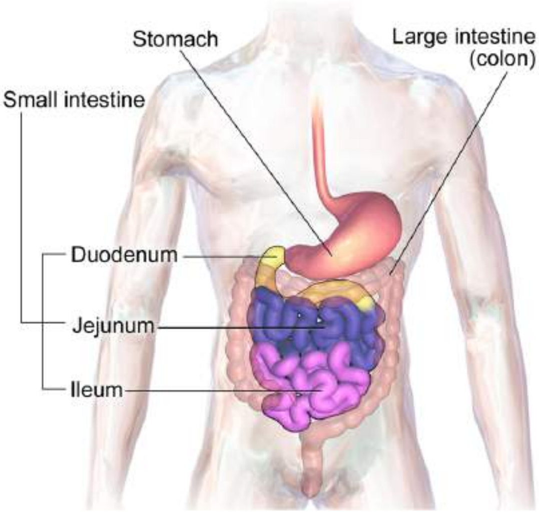


Figure 1.2: Small intestine (*Graphics from Wikipedia*).

large intestine work are feces that are stored in the rectum for a certain amount of time. Finally, the feces are released finishing the digestive process.

The small intestine (also called small bowel) is a part of the gastrointestinal tract that connects the stomach with the large intestine (see Figure 1.2). The length of the small intestine in an adult human is variable and, depending on the conditions, can measure from 3 to 8 meters. The following three parts of the small intestine can be distinguished:

- Duodenum, where the mixing of digestive juices from: 1) the pancreas and 2) the liver (called bile) with chyme happens. This process results in breaking down the chyme into molecules that can be easily absorbed by the remaining parts of the small intestine.

- Jejunum, where products of previous digestion (e. g. sugars, amino acids, and fatty acids) are absorbed into the bloodstream.

- Ileum absorbs mainly vitamin B12 and bile acids as well as any other remaining nutrients.

According to (2), the primary motor functions of the small intestine are to: 1) mix, agitate and propel the ingesta at rates that allow efficient absorption of food components; 2) keep the lumen clean in the interdigestive state; and 3) rapidly propel luminal contents over long distances without regard for digestion or absorbtion in situations such as vomiting, and mass movements. The small intestine pushes the ingesta through by means of a physiological mechanism called motility. In general, the intestinal motility can be categorized as follows (3): 1) Peristalsis - synchronized movement of the intestinal wall responsible for moving the ingesta in one direction. 2) Segmentation - unsynchronized movement of the intestinal wall where the muscles squeeze more or less independently of each other; this has the effect of mixing the contents but not moving them up or down. The movement of the intestinal wall is called contraction. The peristalsis and the segmentation are regulated by three types of contractions (2):

- *Rhythmic phasic contractions* that consist of brief periods of both relaxation and contraction. These contractions cause mixing and slow propulsion of ingesta.
- *Ultrapropulsive contractions* that are used to move rapidly the intestinal content without regard for digestion or absorption. These contractions are two to four times larger in amplitude and four to six times longer in duration than phasic contractions. Moreover, since the goal of these contractions is to clean the intestine rapidly, they propagate uninterruptedly over long distances of the small intestine.
- *Tonic contractions* that are maintained for a longer period of time (from several minutes to several hours). The precise role of these contractions in the digestion process has not been established (2).

During fasting, the small bowel exhibits a cyclic activity pattern, alternating phases of quiescence with phases of intense biliopancreatic secretion into the duodenum associated with forceful propagating contractions. These contractions push the content in caudad direction, clearing residues from the gut. These phases of intense motor and secretory activity occur on average every $100min$ (4). The association of high concentration of biliopancreatic secretion with wall contractions results on a foamy appearance of contents, visually recognized by the presence of abundant bubbles. Ingestion of a meal interrupts this fasting cyclic activity pattern and induces a more homogeneous secretory and motor activity in order to digest the meal. Regardless of the characteristics and amount of food ingested, the stomach delivers into the small intestine a homogeneous chime with particles of less than $1mm$, at a steady rate adjusted to the intestinal processing capability. In fact, the small bowel controls gastric emptying and biliopancreatic secretion by a complex net of feedback mechanisms. As a consequence, postprandial intestinal content consists of a mixture of homogenized nutrients and biliopancreatic secretion in a proportion related to the types of foodstuffs in the meal. Since surfactive agents are diluted into the mixture, the appearance of the chime is turbid without bubbles (5).

The motility process is the result of the integrated activity of nerves, muscles and hormones. Abnormalities in any of these components or in their integration can result in different motility dysfunctions (6, 7). The accurate assessment of the small bowel motility constitutes one of the most relevant issues in gastroenterology. Intestinal motility dysfunctions appear when the organ looses its ability to coordinate muscular activity, manifesting an abnormal contractile activity (e.g. spasms or intestinal paralysis) (8). In a broad sense, any alteration in the transit of foods and secretions into the small intestine tube may be considered a motility disorder[1] (8). In the case of the small bowel, motor dysfunctions may cause mild and severe clinical syndromes, such as intestinal pseudo-obstruction, reduced tolerance to feeding and inability to maintain normal body weight. These motility disorders of the small bowel are mainly caused by improper activity of the muscular layer of the intestine or of the neural control system (9).

Currently, the main source of information, and the only one which leads to a diagnosis of small intestine motility disorders is manometry (10). The diagnosis is based on the measure of intestinal wall pressure change in a fixed part of the small intestine. However, this technique has several drawbacks: 1) it is highly invasive causing discomfort to the patient; 2) it does not offer the visualization of the small intestine; 3) only a portion of the small bowel can be evaluated and 4) the performance of this test is limited to referral centers around the world due to its complexity and the difficulty in the interpretation of results.

1.2 Wireless Capsule Endoscopy

The wireless device, named PillCam SB2, measures $11mm \times 26mm$ and weights less than 4 grams, it has a camera with $156°$ field of view, a battery, a wireless system and 3 optical lenses. The frame rate is 2 frames per second and its image resolution is 256×256 pixels (11). This capsule has been designed to capture images from the small intestine. The camera moves freely inside the intestine - upwards, backwards and with respect to all three rotational axis, (see scheme represented in Figure 1.3) -. The capsule travels through the small intestine (from the beginning of the small intestine - source, to the end of the small intestine - sink) by means of: a) the small intestine motor activity and b) the gravity. These are the only two factors that

[1] It is important to note that the motility dysfunctions are just one part of all digestive problems (like: bleeding, polyps or tumors)

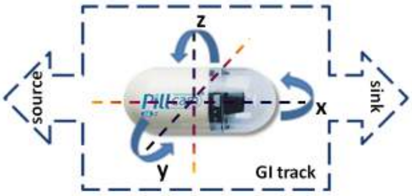


Figure 1.3: Scheme representing possible camera movements.

control the capsule movement, velocity and direction. Generally, the capsule moves forward into the sink, but it is also possible that, for some period of time, the capsule travels backward. The capsule transit time is variable and can vary from minutes to more than 8 hours.

The capsule has rapidly gained recognition within the gastroenterology community thanks to its two main advantages: 1) it offers the inner visualization of the entire small intestine and 2) it obtains the images in a minimally invasive manner reducing the patient preparation time and her/his discomfort. In contrast, standard techniques of gastrointestinal tract examination like manometry or gastroendoscopy are more invasive and produce patient's discomfort or even the need of hospitalization.

However, procedures based on the capsule present several limitations (12). First, the time needed by the physician to analyse the entire video: the capsule emits images at a rate of two frames per second for over 8 hours, that can result in more than 57.600 images for a single study. Second, the device has no therapeutic capability, i. e., if any lesion that needs treatment is discovered, some additional tests must be done with standard procedures as endoscopy, radiology or surgical techniques. Finally, there is a difficulty in discerning the exact location of the visualized lesion.

The Wireless Capsule Endoscopy (WCE) video can be visualized by using the software provided by the capsule manufacturer. The software interface for the analysis of WCE data is shown in Figure 1.4, where three types of information are presented: 1) video field presenting frame view (Figure 1.4 - top), 2) an approximation of the capsule position inside the gastrointestinal tract (Figure 1.4 - bottom left) and 3) a color bar representing the whole video, where each column in the stripe represents a mean color intensity in one-minute frame period (Figure 1.4 - bottom right).

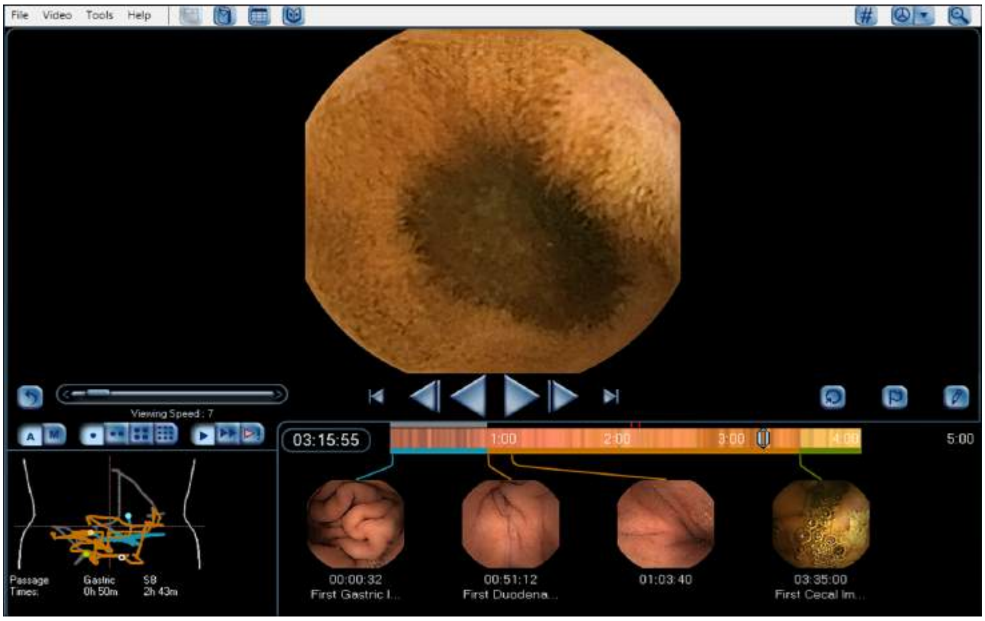
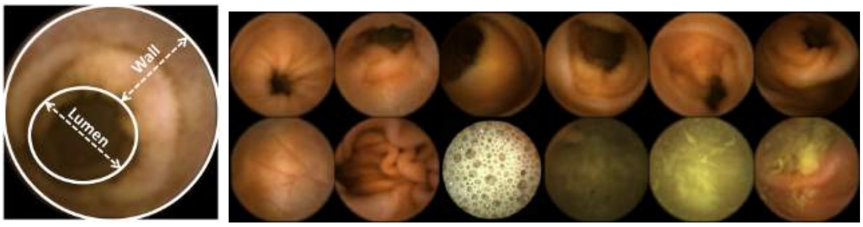


Figure 1.4: Rapid Reader interface by GivenImaging (1). The interface presents: (up) video field presenting frame view, (bottom-left) an approximation of the capsule position inside the gastrointestinal tract and (bottom-right) color bar representing the whole video, each element in the stripe represents a mean color in one-minute frame period.



(a)

(b)

Figure 1.5: (a) Lumen and wall. (b) Different frames acquired by the WCE capsule, first row presents different lumen position due to camera/intestine movement, second row presents (from left to right) two examples of intestinal wall and four examples of intestinal content.

1.2.1 What do we see with Wireless Capsule Endoscopy?

The image from WCE show the lumen and the intestinal wall, see Figure 1.5(a). Since, both the capsule and the intestine can be constantly moving, the lumen can be seen only partially or might be out of the file of view of the camera. Moreover, the file of view of the WCE is often partially or completely occluded by *intestinal content* (intestinal juices and food in digestion, see Figure 1.5(b)).

1.2.1.1 Intestinal content

Previous studies in endoluminal image analysis (13, 14) have shown that intestinal content may exhibit two appearance paradigms, namely: *bubbles* or *turbid* material. Bubble formation depends on the presence of agents that reduce the surface tension, analogous to a detergent. In normal conditions this activity is due to the presence of biliopancreatic secretions, responsible for the solubilization and subsequent digestion of fat. In contrast, turbid appearance reflects the presence of chyme, that is, the meal transformed by the processes of gastric and partial intestinal digestion. In this context, the type of content depends, in normal conditions, on the characteristics and time elapsed since the last meal (5).

The presence of these types of content patterns along the small bowel reflects the propulsion of nutrients and secretions, as well as the degree of digestion, which differs at various levels of the intestine as a function of the digestion progress. Furthermore, abnormal digestive function may affect this process and might modify the pattern distribution of the intestinal contents.

In a normal video, with standard clinical patient preparation, between 5% and 40% of the video frames contain intestinal content and its degree, in a single frame, can vary from covering a small area of the image to completely occluding the intestinal wall and the lumen. Intestinal content frames are presented in a high variability of colors and textures within a video. The colors and the textures are highly correlated with the food ingested by the patients. Generally, according to the visual appearance of these images, *turbid* can be easily differentiated from *bubbles*.

Turbid is usually presented as a region with homogeneous texture (see Figure1.6). The predominant colors presented in turbid frames vary from brown to yellow, however, sometimes it can also be presented in less common colors such as green or red.

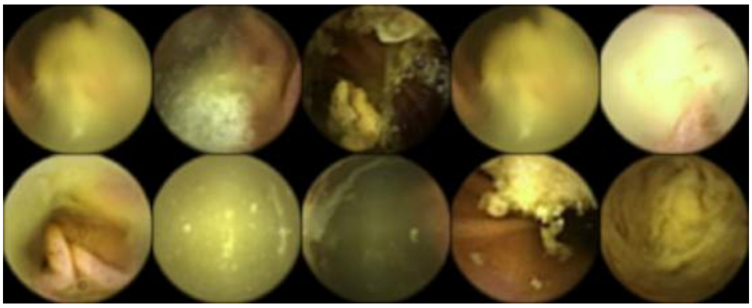


Figure 1.6: An example of turbid frames.

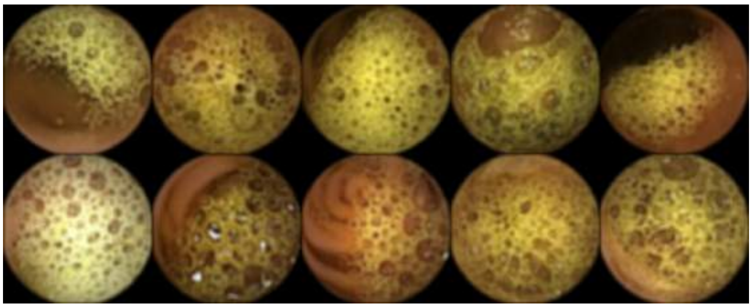


Figure 1.7: An example of bubble frames.

Bubbles are displayed in the image as a well-defined texture (see Figure1.7). This texture is characterized by several ellipsoidal blobs that can vary in size. The predominant colors of the bubbles are: white, yellow and green. However, sometimes bubbles can be practically transparent and the only visible part is the bubble's contour.

1.2.1.2 Intestinal events

Intestinal events (like contractions) describe the behaviour of the lumen/wall over some short period of time. Thus, in order to analyse the intestinal motility, one should analyse a group of consecutive frames. This movement, in the WCE video, can be characterized into two categories of events: 1) *contraction* - movement of intestinal wall/lumen, and, 2) *static* - paralyzed intestine. Moreover, if the intestine is paralyzed with an open lumen we speak about *tunnel* sequence.

Contractions In WCE, the intestinal contractions are visualized as a sequence representing, first, the closing of the lumen from the resting position, and then, the opening of the lumen

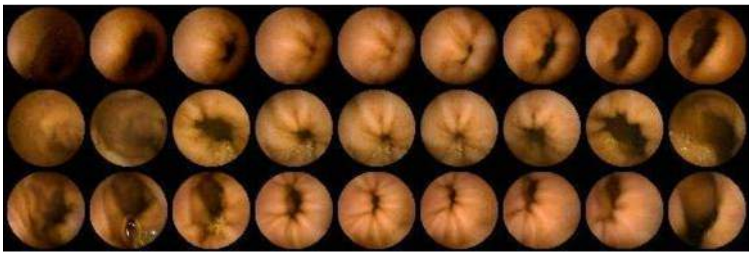


Figure 1.8: Samples of intestinal contractions. Each row represents a set of frames depicting one contraction.

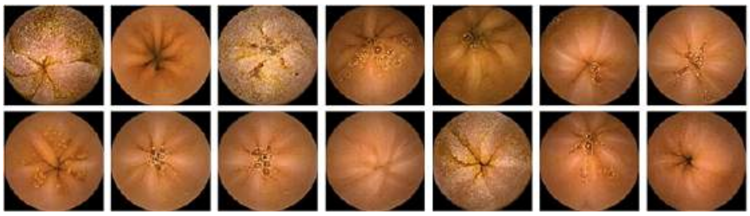


Figure 1.9: Images corresponding to central frames of contractions clearly show a star-like pattern produced by the strong pressure of nerves when closing the intestinal wall.

to the resting position again. The main visual features to characterize these events are: 1) the changing lumen area, and, 2) the presence of characteristic wrinkle-like structure in central frames of the sequence (see Figure 1.8). The duration of the contraction event is variable, depending on the type of contraction.

Wrinkles are seen as a star-like folds of the intestinal wall (see Figure 1.9). Usually, the wrinkle pattern is observed in the central frames of the intestinal contraction where strong pressure is produced by the nervous system.

Static periods and tunnel Static periods represent lack of the intestinal activity and are seen, when the WCE capsule is not moving and the small intestine is paralyzed. Examples of some static periods are shown in Figure 1.10. The duration of the static sequences is variable.

Tunnel is visualized as a static period with open and relaxed lumen, examples of the tunnel periods are shown in Figure 1.11. The duration of the tunnel sequences is similar to the one of static sequence.

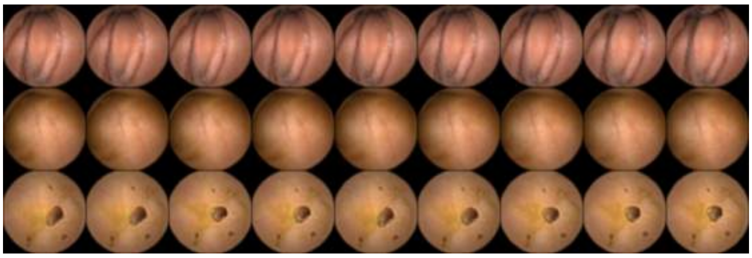


Figure 1.10: An example of static sequences, each line represents one sequence.

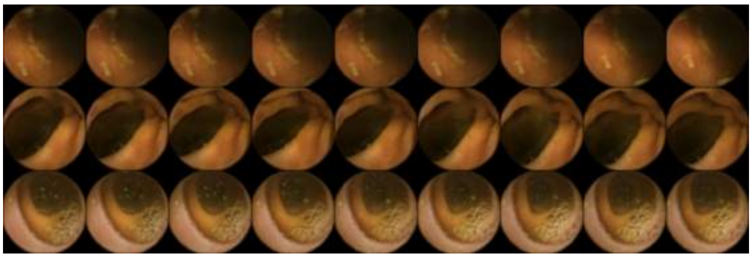


Figure 1.11: An example of tunnel sequences, each line represents one sequence.

1.2.2 Challenges in Wireless Capsule Endoscopy

While designing the system for intestinal motility analysis by means of WCE, the following characteristics should be considered:

Complex appearance of intestinal events The camera moves freely inside the intestine (upwards, backwards and with respect to all three rotational axis). Hence, it is very difficult (or even impossible) to determine the exact capsule position or orientation. The image from WCE can show either the whole lumen or only a part of it, or the intestinal wall. Moreover, the field of view of the WCE is often partially or completely occluded by intestinal content.

Complex interpretation of intestinal scene The intestinal movement can be characterized into three categories: 1) *contraction* 2) *static*, and, 3) *intestinal content*. Although the importance of these events in motility disorders diagnosis has been proven (9), the exact medical importance of each of the event and the relations between them are still open issues.

Large number of images Having available significant amount of data allows detailed description of physiological events. Meanwhile, a huge amount of the data (up to 60000 frames, with each color frame being of the size of 256 x 256 pixels that is about 87 gigabit of information per a single study) requires a long time (up to several hours) for video visualization and for diagnosing a study by the physician.

1.3 Contributions of the thesis

In this thesis, a novel computer-aided system for intestinal motility analysis is presented. The system is based on sequential feature analysis. The goal of the system is to provide an easily-comprehensible visual description of motility-related intestinal events to a physician. In order to do so, several tools based either on computer vision concepts or on machine learning techniques are presented. The contributions of the thesis can be summarized in the following items:

- *Motility bar: a novel representation of intestinal motility.* A method for transforming 3D video signal to a holistic image of intestinal motility is proposed. The method is based on Dynamic Programming and calculates the optimal mapping from video into image from the intestinal motility point of view. The motility bar is validated, showing that the motility information presented on it is very similar to the motility information presented in WCE video. Moreover, it is shown that the motility bar reduces significantly the time needed for visual inspection of motility information.

- *Automatic feature extraction.* Four methods for automatic extraction of motility information from WCE are presented. Two of them are based on the motility bar and two of them are based on frame-per-frame analysis.

  - *Motility bar based features.* First, it is shown how intestinal events are visible in the motility bar. Second, a method for contractions detection is presented and validated. The method is based on Gabor-like filters that detect contractions in the motility bar at different time scales. The results of the different filters are jointed into a single signal representing contractions positions in the motility bar. Second information extracted from the motility bar is the lumen size. The detector is based on the assumption that the lumen is a dark region in the image.
  - *Frame based features.* Two frame-based detectors are introduced. First, a system for detecting intestinal content is presented. This method is able to differentiate between two types of the intestinal content: 1) turbid and 2) bubbles. Moreover,

the method is able to quantify the amount of intestinal content inside a single frame and thus, in the whole WCE video. Second, a method for wrinkle frames detection based on mid-level image descriptors is presented and validated. This method has shown to set-up new state-of-the-art result in the wrinkle frame detection problem.

- *Sequential feature analysis.* We propose a novel formulation of concentration inequality for multivariate data stream. This formulation is sensitive to permutations of vector components and is introduced into a robust adaptive windowing algorithm for multivariate data streams. The algorithm is used to obtain robust representation of segments of constant means. We refer to these segments as sequential features. The algorithm is visually validated in the WCE problems and different sequential features are obtained: 1) color segmentation, 2) joint contraction-lumen analysis and 3) intestinal content. To measure the clinical importance of the sequential features, a set of videos of healthy volunteers and severe intestinal dysmotility patients is collected. Using this database, we show that the sequential features are discriminative to detect subjects with abnormal motility.

- *Efficient labeling systems.* We adapt active learning concepts to the problems present in WCE data. In particular, we address the problem of intestinal content frame labeling and propose two labeling systems.

  - *A system for efficient labeling of WCE frames.* This system is based on concepts from sequential learning and discovers the samples of interest from the point of view of the model that is being constructed. To discover the samples of interest Locality-Sensitive Hashing is used, while the problem of sampling is addressed with criteria from active learning. Finally, this process is incorporated into an online learning setting.

  - *A system for error-free labeling of WCE frames.* The concepts of partition-based active learning are adapted to an error-free labeling scheme. In this scheme, an expert visually revise all data labels. In order to reduce user's effort, the algorithm gives a label proposal. This proposal is based on the system knowledge gained during the labeling process.

This thesis is organized as follows. In Chapter 2, the background on computer-aided systems in WCE is presented. Chapter 3 introduces motility bar, a holistic view on intestinal motility in a single image. In Chapter 4, the methods for automatic features extraction from WCE are presented. In particular 4 methods are revealed: contraction detection, lumen perimeter estimation, intestinal content detection and wrinkle frames detection. Chapter 5 introduces the concepts of multivariate data stream analysis and applies these concepts to sequential analysis of features obtained in Chapter 4. In Chapter 6, the clinical importance of obtained sequential features is presented. In Chapter 7, the concepts of active learning are introduced and two applications for efficient WCE frame labeling are presented and validated. Conclusions and future work end the thesis.

2

Background

2.1 Computer-aided systems in WCE video analysis

Wireless capsule endoscopy has undertaken a relevant boost in recent years and technological advances have been proposed both in hardware and software areas (15) making WCE a widely spread clinical routine. This growth has been generally caused by the interest of the community in developing computer-aided decision support systems (CADSS) (16). These systems have been designed to detect and/or classify abnormalities and thus assist a medical expert in improving the accuracy of medical diagnosis (16).

While revising the literature on CADSS one can notice that all the approaches, first, extract some features form images and, second, apply some decision rule.

Feature extraction Once an image (or a set of images) is obtained, the CADSSs codify the visible information into numerical features. Depending on the problem that is being tackled, one can use one of the following descriptors: textural features -e.g. Local Binary Pattern (LBP) (17)-, color-based features -e. g. Color Histogram-, gradient-based features -e. g. Histogram of Gradients (HoG) (18)- , blob detectors -e. g. Laplacian of Gaussian (LoG)-, bag-of-visual-words features or points of interest -e. g. SIFT points (19)-. Sometimes, some higher level features are needed to describe the image. In order to extract those features, one needs to apply some algorithm beforehand -e. g. edge detection, shape fitting or segmentation algorithms (20)-.

Decision rule Once the features are extracted, the decision can be made. The decision can be based either on some simple thresholding approach or can incorporate some classification algorithms -e. g. Support Vector Machines (SVM) (21), or Neural Networks (NN) (22)-. Since the parameters of decision making algorithm should be trained beforehand, the CADSSs require the collection of training, validation and testing sets.

In case of CADSSs for WCE, researchers have focused their efforts on trying to deal with the inherent drawbacks associated to the video screening stage: the long time needed for visualization and the potential subjectivity of the observer due to fatigue.

The main effort in CADSSs for WCE has been put on the detection of bleeding and lesions (e. g. polyps, ulcers) (16). Bleeding is usually detected by performing an analysis of color (usually in Hue-Saturation-Intensity space). The method in (23) uses chromaticity moment with a NN classifier, in (24) the adaptive color histogram together with a SVM classifier is used, while the algorithm in (25) uses color spectrum transformation with threshold value to detect bleeding regions in WCE frames.

Color information is sufficient for bleeding detection, still, for lesion detection some reacher descriptors should be used. For polyp detection several methods have been proposed: (26) integrates color and texture information using LBP, (27) uses Gabor filters based segmentation enhanced with edge detector, whereas (28) uses curvature information to obtain polyp segments. In (29), wavelet based LBP are used to detect tumors and bag-of-words approach based on LBP and SIFT features are exploited in (30) to detect ulcers. In (31), pixel brightness and image texture descriptors together with nonlinear classifier are used to detect celiac disease. Finally, MPEG-7 descriptors for color, texture and edge are tested for Chron's disease detection in (32).

Another line of CADSSs is focused on differentiation of the diverse organs of the intestinal tract like esophagus, stomach, duodenum, jejunum-ileum and cecum. In (33), the textural features captured by Gabor filters are exploited in the problem of duodenum discrimination. The locations of the esogastric junction, pylorus, and ileo-cecal valve are estimated with an algorithm based on MPEG-7 visual descriptors in (34), while the color change pattern is exploited to detect different organs in (35).

Regarding the problem of WCE video visualization techniques, the researchers have focused their efforts on video compaction. This process results in eliminating/compacting similar frames (36, 37, 38) and/or in applying variable sampling rate at acquisition step (39). Video compaction by elimination of frames of the video in which the capsule/intestine is paralyzed permits to reduce the experts skimming process and, thus, the time needed for video visualization. The methods in (36, 37, 38) automatically control display rates of the WCE by estimating the color distribution change between two consecutive frames. In (39), a model of deformable ring has been proposed to estimate the capsule/intestine motion between two consecutive frames.

Some authors have addressed the problem of intestinal content detection (also referred to as non informative frames detection). In (14), the authors have presented a method for detecting bubble-like shape of intestinal juices based on Gabor filters, while in (6) color histograms together with a SVM classifier are used to detect intestinal content. In (13), a three-stage cascade to detect informative frames has been proposed: in the first stage, color information (histogram and color moments with a SVM classifier) is used to characterize turbid, in the second stage, texture segmentation (Gauss Laguerre Transform segmentation) is applied to characterize bubbles, and, finally, a threshold on the segmented regions is applied in order to detect informative frames.

In the area of intestinal motility some work on detection and characterization of specific events of intestinal motility has been done, e.g. 1) contractions and wrinkles detection (6, 7, 40, 41, 42) or 2) static and tunnel sequence detection (43).

Contractions and wrinkles In (7, 40), a three-stage cascade is proposed: first, an edge detector is used to find sequences of WCE video with possible contractions, second, the similarity between light-intensity histograms of frames is evaluated to eliminate non-contractions and, finally, the presence of wrinkle-like pattern (described using the edge direction histogram) in the central frames of the sequence is used to detect real contractions. In (6), a sequential design is proposed, based on: 1) textural features (co-occurrence matrix of gray level image and LBP), 2) color features (mean lumen color)and blob features (LoG filter). This information is extracted for 9 consecutive frames and is classified with a SVM classifier. Several publications have focused only on the detection of wrinkle frames. In (41), a detector of tonic contractions based on wrinkle information has been proposed. The method uses general linear radial patterns as features. An alternative method has been proposed in (42), where a structure tensor matrix is used to derive an image descriptor. In both works (41, 42), the WCE frame is divided into 4 different quadrants positioned with respect to the lumen center, the features are computed for each quadrant and classified with SVM.

Static and tunnel In the literature, static sequences have been characterized by color distribution change between consecutive frames (36, 37, 38, 43). Tunnel detection is performed analyzing the lumen change in a sequence of consecutive frames. In (43), the lumen is characterized using LoG filters and classified with SVM.

Let us make some remarks on the training sets used in WCE. It is important to note that all papers working with lesions, intestinal content or event detection use the ground truth visually established by an expert (16). Since data collection for WCE is challenging, the size of the data sets used for training and testing is rather small. It might be difficult to make a strong statement on CADSS applicability using such small amount of data (16). One can find out in (16) that the majority of WCE papers uses less than 10 video cases to conclude on the algorithm performance.

In (9), the authors approach the evaluation of motility from WCE images. This paper is an extension of works on specific events characterization making the analysis of the small intestine motility more complex. The method first extracts several motility descriptors (each descriptor represents a specific intestinal event e. g. number of contractions, video coverage with intestinal content, etc.) from WCE videos. Then, it combines the extracted characteristics into one feature vector for each video, and finally, draws a conclusion on the small intestine motility using a non-linear two class SVM classifier. In this study, the authors use a database composed of 36 abnormal motility patients and 50 healthy subjects.

3

Data representation for intestinal motility characterization

3.1 Introduction

Using WCE signal, two types of data representation can be used: 1) frame view and 2) longitudinal view. The frame view gives information about a slice of intestine (e.g. lumen/wall appearance, see Figure 3.1(left)), while the longitudinal view, usually perpendicular to the intestinal tube, shows the desired segment of the video (e.g. lumen/wall change in time - motility, see Figure 3.1(right)). The longitudinal view is obtained from the WCE video by ”cutting”[1], frame by frame, a line of pixels from an WCE image. Note that these cuts do not have to be fixed at the same angle/position, they can adapt to the lumen position (e. g. compensating the camera rotation).

In this chapter, a novel technique called Adaptive Cut (Acut) longitudinal view for data representation of WCE videos is presented. The chapter addresses the problem of choosing the optimal sequence of cuts through consecutive video frames in order to represent the intestinal motility. Due to the free movement of the capsule inside the intestine, the lumen is not always present in the center of the frame. Therefore, a straight-forward approach to cut through a fixed angle (e.g. the vertical line) of the pixels of the image would lead to losing motility information (for an example of a fixed cut see Figure 3.2(top)). The Acut methodology is based on an optimization problem that maximizes the probability of passing through the lumen in order to preserve motility information in the new visualization scheme (for an example of adaptive cut see Figure 3.2(bottom)). The main advantages of the proposed method are:

- It reduces the information provided by the WCE video, transforming 3D video signal into 2D image and, thus, permits to evaluate the intestinal motility at glance.

- It preserves the motility information by applying adaptive cuts that maintain, where possible, the lumen/wall information.

Since the Adaptive Cut longitudinal view shows motility information from WCE video, throughout the thesis we will refer to it as motility bar. In the current Chapter, we will refer to it as Adaptive Cut longitudinal view.

[1] Throughout the thesis, we will refer with the word ”cut” to a straight line of pixels that passes through the center of the frame (diameter).

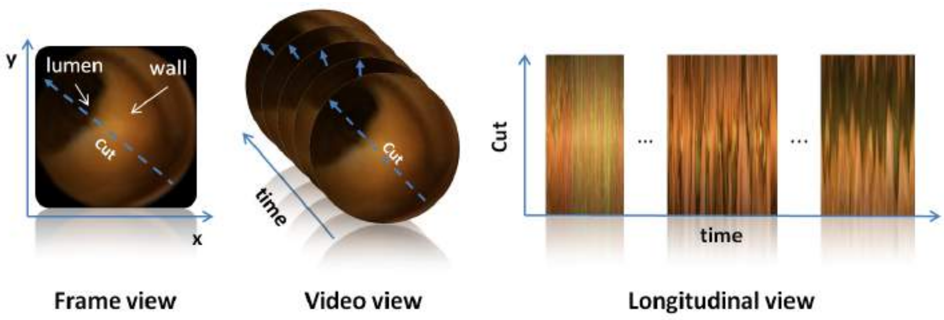


Figure 3.1: Different views that can be obtained from the WCE video, from left to right: 1) single frame view, 2) video view and 3) longitudinal view of adaptive cuts.

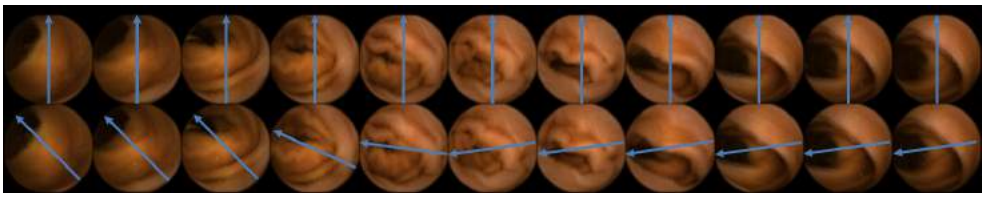


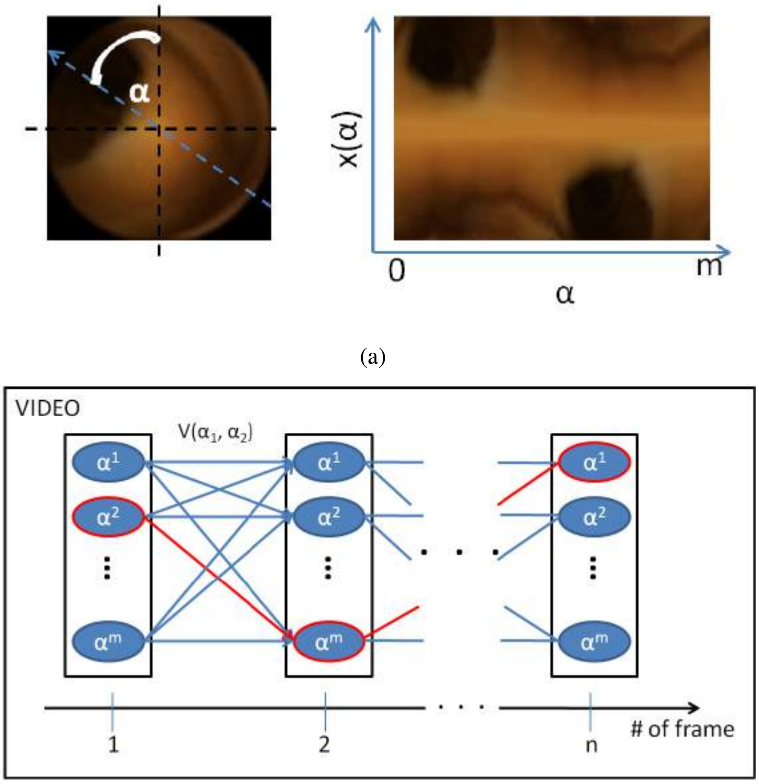
Figure 3.2: Examples of fixed cuts (first row) and adaptive cuts (second row).

3.2 Motility bar creation

In this section, the algorithm for adaptive longitudinal cuts is presented. The word adaptive, in this context, means that the algorithm searches for an optimal path through all video frames, considering for each frame a set of all possible cuts.

The WCE video can be seen as a chain of n frames. Each frame i has m possible cuts $\alpha_i \in \mathbf{\Omega} = \{1, \cdots, m\}$, where the angle α_i denotes the angle between the vertical line passing through the center of the frame and the line representing the cut (see Figure 3.3(a)). Moreover, let us define the cost of passing from cut α_i in frame i to cut α_{i+1} in frame $i+1$ as $V(\alpha_i, \alpha_{i+1})$.

The problem of adaptive longitudinal view construction can be seen as an optimization problem, where two constrains are introduced: 1) the lumen visibility and 2) the smoothness of the view. The first term ensures that the cut passes through the lumen and, thus, ensures its visualization in the longitudinal view, while the smoothness term is important to avoid sudden changes between consecutive frames and, thus, maintain the interpretability of the view. This task can be reformulated as a problem of finding an optimal path (presented in red in the graph in Figure 3.3(b)). The cost of a candidate solution $(\alpha_1, \cdots, \alpha_n)$ to the problem can be defined



(a)

(b)

Figure 3.3: a) An illustration of the cut angle α_i. Left-hand-side, ith image, right-hand-side, all possible cuts through the original image. b) A graph illustration of Dynamic Programming problem in WCE video.

according to the following cost equation (44):

$$E(\alpha_1, \cdots, \alpha_n) = \sum_{i=1}^{n} D(\alpha_i) + \sum_{i=2}^{n} V(\alpha_{i-1}, \alpha_i). \tag{3.1}$$

The terms $D(\alpha_i)$ are used to ensure that the cut in the ith image passes through the lumen, while the $V(\alpha_{i-1}, \alpha_i)$ ensures that the change among angles α_{i-1} and α_i is smooth (this term captures the cost of change between two consecutive frames, $i - 1$ and i). The best solution is the one that passes through all video frames and has the minimal cost.

Because of the size of the WCE video (n up to 60000 frames), we propose to use Dynamic Programming (DP) in order to find the minimum of the function described by Equation (3.1) and to obtain the angles for the image cuts to be used in the longitudinal view of the WCE video.

3.2.1 Dynamic Programming

Dynamic Programming (DP) is broadly used in discrete optimization problems. DP finds the global optimum to the given problem. The basic idea of DP is to decompose a problem into a set of subproblems which can be efficiently solved in a recursive way (44). Hence, the difference with respect to classical recursive methods is memoization (storing solutions to already solved subproblems).

Let the table $B(\alpha_i)$ denote the cost of the best assignment of angle cuts to the frame i, with the constraint that ith frame has label α_i. The size of the table B is $n \times m$, where m is the cardinality of the set $\mathbf{\Omega}$ and n the number of frames. The table $B(\alpha_i)$ can be filled in increasing i by using the following recursive equations:

$$B(\alpha_1) = D(\alpha_1),$$

$$B(\alpha_i) = D(\alpha_i) + min_{\alpha_{i-1}}(B(\alpha_{i-1}) + V(\alpha_{i-1}, \alpha_i)) \tag{3.2}$$

The subproblems are defined as follows. For the first frame, $B(\alpha_1)$ is the cost of assigning the angle α_1 to the first frame. For every other frame, $B(\alpha_i)$ is the cost of assigning the angle α_i plus the minimal transition cost from $i-1$th to the ith frame $min_{\alpha_{i-1}}(B(\alpha_{i-1})+V(\alpha_{i-1}, \alpha_i))$.

In order to avoid recalculating the solutions to sub-problems, a matrix T is filled in while calculating the tables $B(\alpha_i)$. The matrix T stores optimal solutions to sub-problems (thanks to this book-keeping, each sub-problem is calculated only once). Each row of the matrix has a size of m and stores the ”best way” to get to the ith solution from the $i-1$th solution. Each time a new value is added to $B(\alpha_i)$, T is updated according to the rule:

$$T(\alpha_i) = \underset{\alpha_{i-1}}{\operatorname{argmin}}(B(\alpha_{i-1}) + V(\alpha_{i-1}, \alpha_i)) \tag{3.3}$$

As a result, the matrix T stores the indices of the nodes belonging to the optimal problem solution. Finally, the overall solution is tracked back $\alpha_{i-1} = T(\alpha_i)$ starting at $i = n$ resulting in the sequence of optimal cuts $(\alpha_1, \cdots, \alpha_n)$ through all frames in the video. This sequence of optimal cuts can be seen as the path of minimal cost through the frames in the video.

3.2.2 Lumen visibility

The lumen in the WCE image is seen as a dark blob often surrounded by the intestinal wall. The image cut that passes through the intestinal lumen can be characterized in terms of mean and variance of the light intensity. In order to ensure the lumen visibility, the algorithm looks for cut with high variance and low mean value (mean and variance are calculated using the pixels that compose the cut). High variance σ^2 ensures that the cut preserves maximal information of the frame maintaining, where possible, the lumen/wall information. Low mean value μ ensures that the cut passes through the dark area of the image. Note that the dark area of the image presents a lumen with high probability.

Let $x(\alpha_i)$ denote the vector of the pixels from the image cut localized in the angle α_i and passing through the center of the image (see Figure 3.3(a)), the lumen visibility cost D can be defined as follows:

$$D(\alpha_i) = 1/(\sigma(x(\alpha_i)) + 1) + \mu(x(\alpha_i)) \tag{3.4}$$

where it is assumed that the values of the vector $x(\alpha_i)$ are in the range $[0, 1]$.

3.2.3 Smoothness

In order to ensure the longitudinal view smoothness, a term that controls the changes between the angles of consecutive frames is introduced. The smoothness is restricted by two factors: 1) angle change $V'(\alpha_{i-1}, \alpha_i) = 180° - |180° - |\alpha_{i-1} - \alpha_i||$ and 2) similarity between two consecutive cuts $V''(\alpha_{i-1}, \alpha_i) = \|x(\alpha_{i-1}) - x(\alpha_i)\|_2$. The final smoothness term V is defined as follows:

$$V(\alpha_{i-1}, \alpha_i) = \beta(V'(\alpha_{i-1}, \alpha_i)/\gamma_1)^2 + (1 - \beta)(V''(\alpha_{i-1}, \alpha_i)/\gamma_2)^2 \tag{3.5}$$

where quadratic terms in V' and V'' are introduced in order to penalize sudden changes, γ_1 and γ_1 are normalization terms, and $\beta \in [0, 1]$ is a parameter controlling the weight between change of angles and similarity of cuts in consecutive frames.

3.2.4 Computational issues

Let m denote the number of possible cuts and n denote the number of frames. At each iteration the algorithm calculates: 1) m means of pixels in cut; 2) m variances of pixels in cut; 3) m^2 angle differences between cuts in consecutive frames and 4) m^2 similarities between cuts in consecutive frames. So, the computational complexity of the algorithm is $O(m^2 n)$.

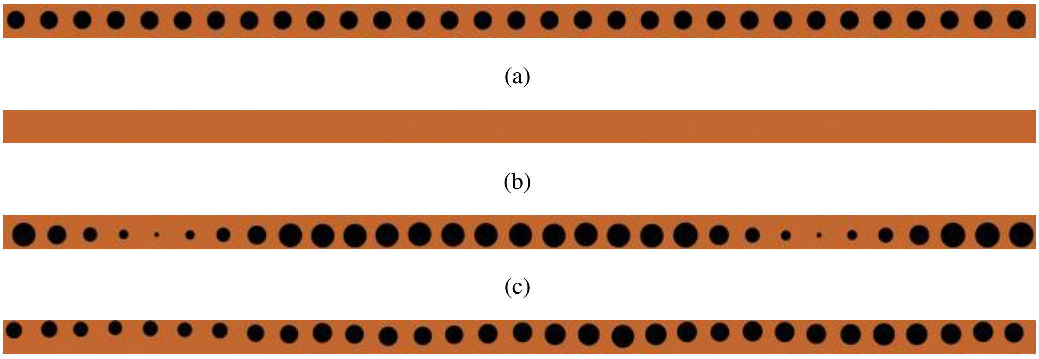
We implemented the algorithm in Matlab and ran the code on 2.6GHz Intel Xenon machine with 16 GB of memory. The running time for one video of 9679 frames was of 1468 seconds. The memory used: 1) cost matrix storing 90 cuts for every frame of double precision: 6.8 MB and 2) vector with 1 uint8 index for each frame: 76 kB.

3.3 Validation

We tested our algorithm on synthetic data and on WCE data. The WCE data was obtained using the SB2 capsule endoscopy camera developed by Given Imaging, Ltd., Israel (1). All videos were conducted at Digestive Diseases Department, Hospital General ”Vall d’Hebron” in Barcelona, Spain.

During the validation three types of cuts for longitudinal view were tested:

1. ($Acut$) - The adaptive cuts described in section 3.2,
2. ($Acut^-$) - The modification of the proposed algorithm by removing the smoothing term V. In this way Equation (3.1) becomes $E(\alpha_1, \cdots, \alpha_n) = \sum_{i=1}^{n} D(\alpha_i)$. This is done to test the influence of the smoothing term in the energy function,
3. ($Fcut$) Longitudinal view with fixed cut.



(a)

(b)

(c)

(d)

Figure 3.4: Some examples of the synthetically created intestinal events. From the top: a) tunnel, b) static, c) contractions and d) undefined movement.

3.3.1 Synthetic data

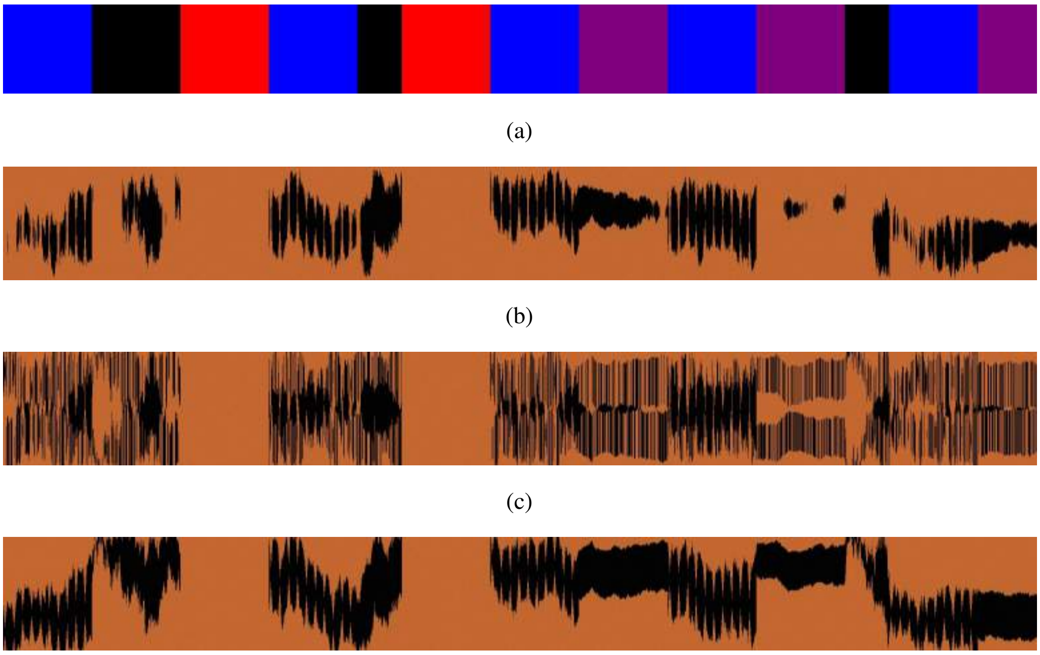
In this experiment, a synthetic video of 40.000 frames was created with a frame rate of 2 *fps*. On an uniform background, a blob is placed. The blob position was changed on consecutive frames depending on the intestinal event. The following intestinal events were used to create the synthetic video: $\{$*tunnel*, *static*, *contraction*, *undefined movement*$\}$. In order to make the video more realistic, the events order and duration were defined with random number generator.

The following definitions of the specific events were used to generate the sequence:

- *Tunnel* - defined as a sequence of paralyzed intestine with open lumen. Lumen size is defined as highly constant (± 2 pixels difference in diameter between two consecutive frames) and highly open (larger than 70 pixel in diameter). An example of the tunnel sequence is presented in Figure 3.4(a).

- *Static* - defined as a sequence of paralyzed intestine with closed lumen (no lumen observed in the frame). An example of static sequence is presented in Figure 3.4(b).

- *Contraction* - defined as a sequence of frames with presence of intestinal contraction defined as symmetric pattern of open-close-open lumen with duration of 9 frames and a fixed frequency of 6 contractions per minute. The lumen size of the central frame of the intestinal contraction was defined as 10% of the initial lumen size. An example of contraction sequence is presented in Figure 3.4(c).



(a)

(b)

(c)

(d)

Figure 3.5: An example of three longitudinal views generated from the synthetic data. From the top: a) ground truth (blue - contraction sequence, black - undefined movement sequence, red - static sequence, purple - tunnel sequence), b) $Fcut$, c) $Acut^{-}$ and d) $Acut$.

$Fcut$	$Acut^{-}$	$Acut$
73%	85%	92%

Table 3.1: Table presenting the video segmentation score for different cuts. The number represents the segmentation accuracy.

- *Undefined movement* - defined as an irregular movement of the lumen size (± 30 pixels variation between consecutive frames) and capsule (± 30 pixels variation between consecutive frames). An example of the undefined movement sequence is presented in Figure 3.4(d).

Some examples of the longitudinal view obtained using different cuts are presented in Figure 3.5. Note that the $Fcut$ loses the blob information and leads to miss-interpretation of the events (e. g. tunnel sequence, Figure 3.5(a) vs. Figure 3.5(b)). Applying the adaptive cut without the smoothing term $Acut^{-}$ leads to good lumen detection, but the view is uninterpretable (here, only the static sequence can be visually detected). The adaptive cut $Acut$ presents well both lumen information and view smoothness.

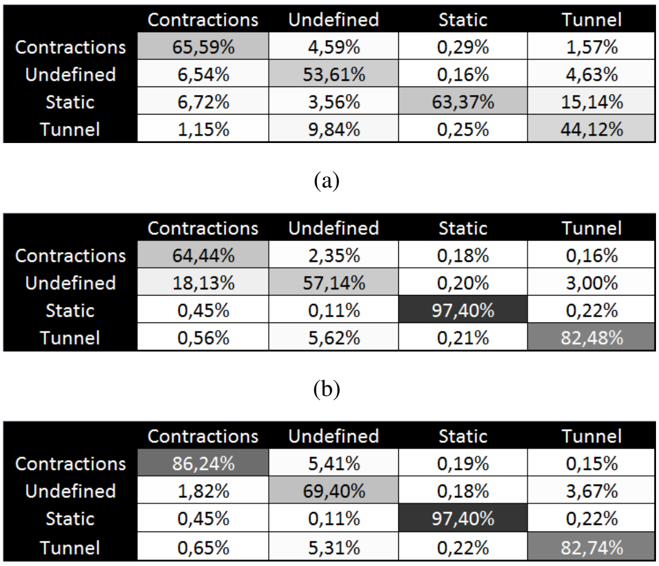
The synthetic video was presented to an expert asking them to recognize and to mark the beginning and the end of the intestinal event sequences. The results were evaluated using the Jaccard index (45) (1 means perfect video segmentation and 0 means no coincidence between visually detected sequences and ground truth).

The overall results are presented in Table 3.1. As it can be seen, the longitudinal view obtained by using $Acut$ method achieves 92% and outperforms $Fcut$ and $Acut^{-}$ methods (73% and 85%, accordingly). Analyzing the confusion matrices presented in Figure 3.6, the following conclusions can be drawn:

- In $Fcut$, the tunnel sequences are frequently confused with static sequences, this happens when the blob is placed out of the cutting plane (see Figure 3.6(a)).

- Applying the $Acut^{-}$ improves the lumen detection reducing the confusion between tunnel and static sequences. The contraction detection rate is still small due to the lack of smoothness, sequences are often confused with undefined sequences (see Figure 3.6(b)).

- In $Acut$, the biggest confusion is caused by the undefined sequence. The expert has a problem in distinguishing between the undefined and the contraction sequences and



	Contractions	Undefined	Static	Tunnel
Contractions	65,59%	4,59%	0,29%	1,57%
Undefined	6,54%	53,61%	0,16%	4,63%
Static	6,72%	3,56%	63,37%	15,14%
Tunnel	1,15%	9,84%	0,25%	44,12%

(a)

	Contractions	Undefined	Static	Tunnel
Contractions	64,44%	2,35%	0,18%	0,16%
Undefined	18,13%	57,14%	0,20%	3,00%
Static	0,45%	0,11%	97,40%	0,22%
Tunnel	0,56%	5,62%	0,21%	82,48%

(b)

	Contractions	Undefined	Static	Tunnel
Contractions	86,24%	5,41%	0,19%	0,15%
Undefined	1,82%	69,40%	0,18%	3,67%
Static	0,45%	0,11%	97,40%	0,22%
Tunnel	0,65%	5,31%	0,22%	82,74%

(c)

Figure 3.6: Confusions matrices presenting Jaccard index obtained using the synthetic data. a) $Fcut$, b) $Acut^-$ and c) $Acut$.

	$Acut$	$Acut^-$	$Fcut$
detection rate	87%	94%	55%

Table 3.2: Table presenting results on lumen detection using different image cuts. Numbers represent the detection rate.

between the undefined and tunnel sequences. Note that, the confusion is small, less than 6% (see Figure 3.6(c)).

3.3.2 Lumen detection

In this part of the validation, we evaluate the lumen detection using different cuts. Here, correct detection means that the cut passes through the intestinal lumen. The lumen was manually segmented in 24740 frames from WCE video. As expected, applying the smoothing term reduces blob detection rate by 7%, resulting in an overall score of 87% of the blobs detected. This is a good score when compared to fixed cut that loses the lumen every two frames. Results are presented in Table 3.2.

3.3.3 Real data

As a first step of validation on real data, a qualitative inspection of the videos obtained with the *Acut* and the *Fcut* is done. Figure 3.7 shows several examples of applying the *Acut* and the *Fcut* to the WCE videos. The difference between cuts is clearly visible when analyzing the lumen. More precisely, the difference (*Acut* vs. *Fcut*) can be summarized as follows:

- Figure 3.7(a) vs. Figure 3.7(b) - the lumen is better followed by the *Acut*, where the reconstruction of the tunnel is clearly visible, while in the case of *Fcut* it can be observed that the intestinal wall is present in some parts of the intestinal tunnel.
- Figure 3.7(c) vs. Figure 3.7(d) - *Acut* presents the lumen in the whole segment, while *Fcut* presents the lumen only in some parts. The presence of the lumen in the longitudinal view facilitates the interpretation of the motility information.
- Figure 3.7(e) vs. Figure 3.7(f) - larger radius of tunnel is visible in *Acut* than in *Fcut*. Moreover, the *Acut* is able to follow the radius while it changes the position in the WCE video.
- Figure 3.7(g) vs. Figure 3.7(h) - *Acut* presents well the open-close-open lumen pattern, while in *Fcut* this pattern is not clearly visible.

	Contraction	**Static**	**Tunnel**	**Turbid**
Lumen presence	Yes	No or small lumen	Yes	Can be occluded
Intestine tissue presence	Yes	Yes	Yes	Can be occluded
Colors seen	Orange (tissue) and dark brown/black (lumen)	Orange (tissue)	Orange (tissue) and dark brown/black (lumen)	Light green to dark brown (turbid), and optionally orange (tissue)
Lumen/tissue movement	Yes	No	No or small changes	Can be occluded
Open-close-open lumen pattern	Yes (visible as ”stripes”)	No	No	No

Table 3.3: Table pointing out the main visual aspects of different sequences seen on longitudinal view.

In the next part of the validation, the *Acut* longitudinal view (see Figure 3.8) and the original WCE video (using Rapid Viewer interface see Figure 1.4) were presented to an expert

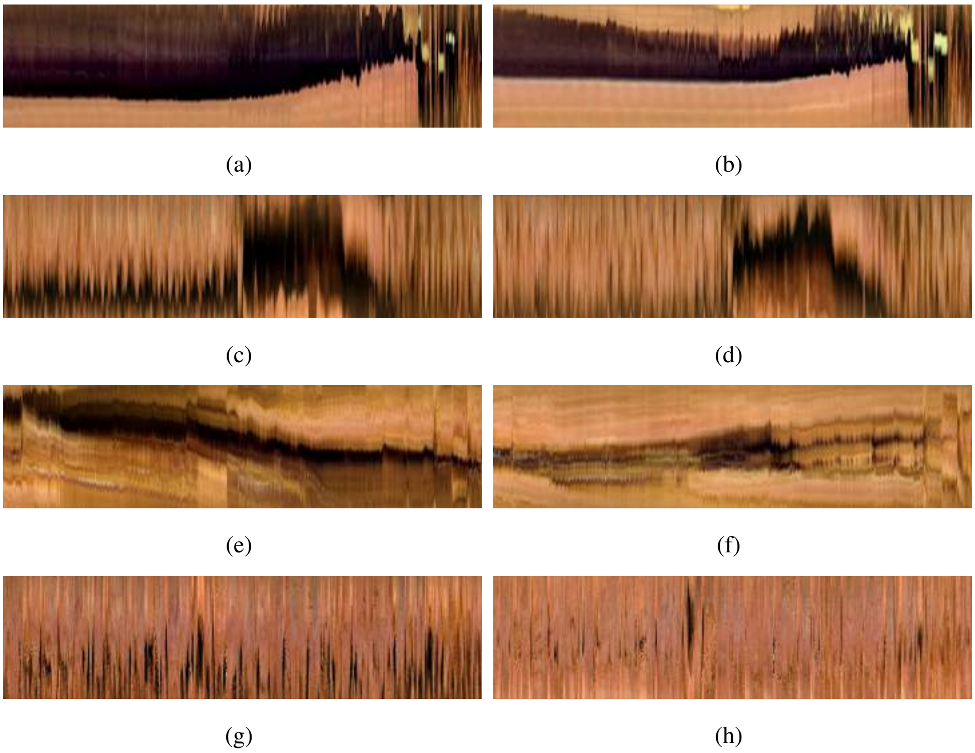


Figure 3.7: Visual comparison of different cuts (a, c, e, g) - $Acut$, (b, d, f, h) - $Fcut$.

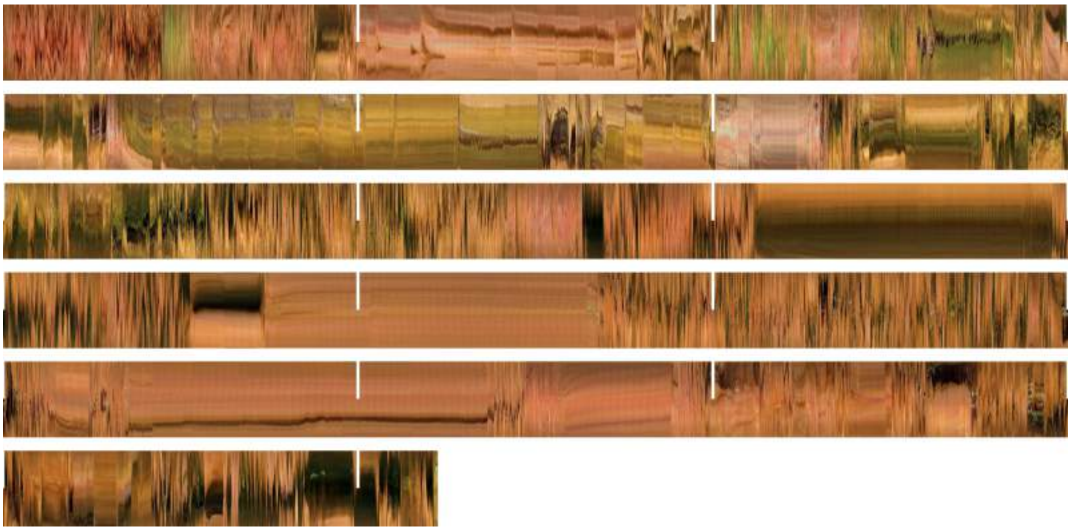


Figure 3.8: An example of longitudinal view presenting the whole small intestine from duodenum to cecum. Each white stripe marks 10 minutes of the video duration.

asking them to mark the beginning and the end of the following sequences: $\{$*tunnel*, *static*, *contraction*, *turbid* and *undefined movement* $\}$. Table 3.3 points out the main visual aspects of different segments seen in longitudinal view.

Some examples of sequences present in longitudinal view can be seen in Figure 3.9:

- Figure 3.9(a) and Figure 3.9(b) show static sequences that are seen as homogeneous images of intestinal wall (tissue),
- Figure 3.9(c) and Figure 3.9(d) show tunnel sequences where both intestinal lumen and wall are static and clearly visible,
- Figure 3.9(e) and Figure 3.9(f) show turbid sequences where the green color of intestinal content occluding lumen/wall can be seen,
- Figure 3.9(g) and Figure 3.9(h) show contraction sequences with the periodical changes in the visible lumen size.

Pay attention to the turbid mixing contractions in Figure 3.9(e). Also, note the variety in contraction rhythm in, Figure 3.9(g) and Figure 3.9(h).

In the first step, the expert labeled[1] five types of event sequences in a video view. In the second part, the expert labeled the sequences using longitudinal view ($Acut$). The similarity between different annotations in video view and in longitudinal view ($Acut$) is 81%. The time needed by an expert for visual inspection of the data of duration of 164 minutes was 80 minutes in video display vs. 18 minutes in the longitudinal view. The longitudinal view reduces the inspection time by 62 minutes, that is a reduction of 444%.

For the obtained annotations, the confusion matrix presenting the overlapping between two annotations was calculated. Analyzing the confusion matrix presented in Figure 3.10, the following conclusions can be drawn:

- Turbid sequences are coinciding quite well between both annotations. The biggest confusion 7% is caused by the tunnel sequence in the longitudinal view. This is due to the opaque turbid that only slightly hinders the lumen/wall and is difficult to be perceived in longitudinal view.

[1] The experts knew how the Acut works, and knew that the task was being timed.

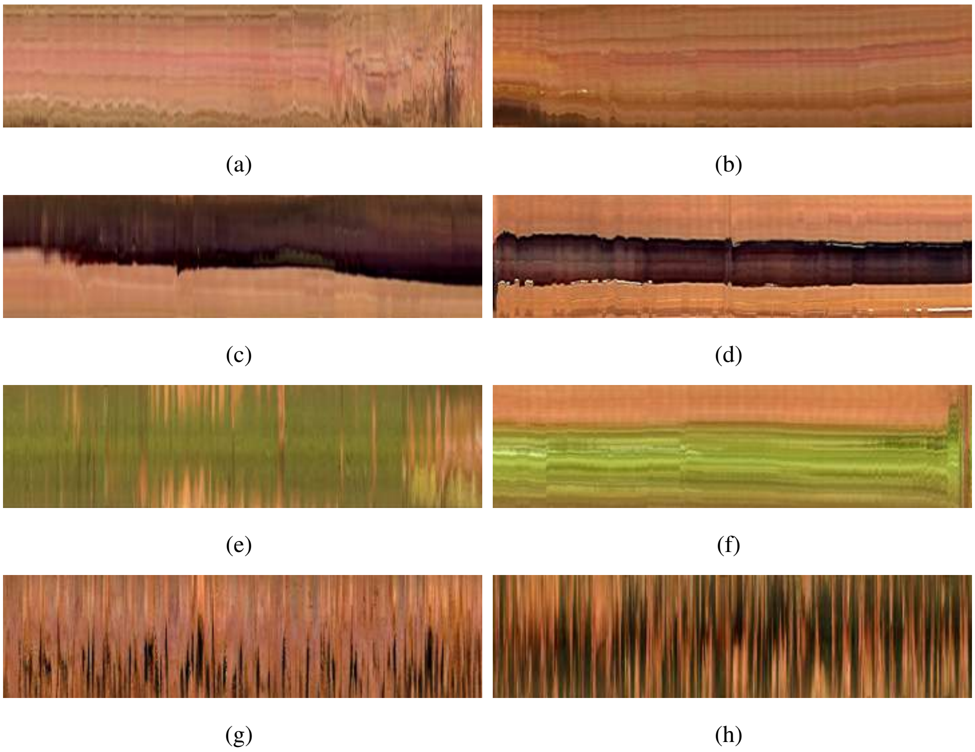
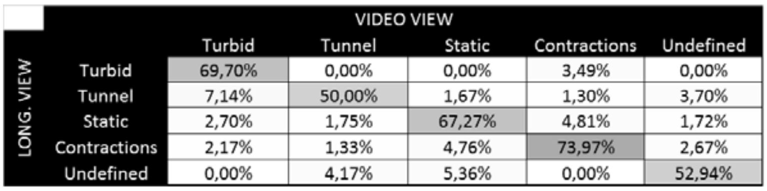


Figure 3.9: Examples of longitudinal views presenting different intestinal events (a - b) static, (c - d) tunnel, (e - f) turbid and (g - h) contractions.



		VIDEO VIEW				
		Turbid	Tunnel	Static	Contractions	Undefined
LONG. VIEW	Turbid	69,70%	0,00%	0,00%	3,49%	0,00%
	Tunnel	7,14%	50,00%	1,67%	1,30%	3,70%
	Static	2,70%	1,75%	67,27%	4,81%	1,72%
	Contractions	2,17%	1,33%	4,76%	73,97%	2,67%
	Undefined	0,00%	4,17%	5,36%	0,00%	52,94%

Figure 3.10: Confusion matrix for annotation comparison between video based annotations and longitudinal view based annotations. The numbers represent the Jaccard index.

- The 50% coincidence in tunnel between annotations in the longitudinal view and video view is caused by: 1) opaque turbid that is difficult to detect in longitudinal view and 2) the camera rotation together with small intestine wall oscillations make it difficult to spot the tunnel sequence in video view.

- Static sequences coincide well in both annotations with the score of 67%. The biggest confusion is between static and undefined movement sequence.

- Contractions have the highest score of coincidence of all the annotated events 74%.

To check the intra-observer variability, another user labeled the events in the longitudinal view image. The obtained kappa score was: 0.80, with 0.84 observer agreement and 0.21 random agreement. Analyzing the confusions/errors of the annotations, the following observations can be made:

- Turbid vs. contractions - one expert in the presence of turbid and contractions preferred to annotate turbid, the other annotated contractions.

- Undefined - this label is used whenever an expert is not sure about the considered event. In many cases, it is used in borderline situations.

3.4 Discussion

In this chapter, a fast and efficient algorithm for constructing an adaptive longitudinal view for motility analysis has been presented. It allows compact display and fast inspection of motility data acquired with WCE. To the best of our knowledge, no previous longitudinal visualization technique has been proposed for motility analysis. The algorithm adapts the frame cut angle to the lumen position by minimizing cost function. This problem formulation permits to apply a Dynamic Programming framework to efficiently find the global minimum to the proposed cost function. Experimental results on both: 1) synthetic data and 2) WCE data show that the proposed algorithm preserves well lumen/wall separation allowing to inspect the intestine motility in details. The annotations obtained by using adaptive longitudinal view coincide well with the ones obtained by using video view. Moreover, the time needed for visual inspection is four times faster in longitudinal view than in Rapid Reader interface.

The proposed method has some limitations that are worth mentioning. First of all, the cut passes always through the central point of the frame, this set-up reduces the space of possible solutions. As a result, the obtained solution does not have to be the solution containing the maximal lumen region. Second limitation is that the first term of the proposed energy function (based on basic statistics of the cut) may not infer lumen regions, it only ensures that the cut passes through dark part of the frame as it is assumed that the lumen is a dark blob. Examples of the frames where this assumption might fail are: frames with dark food content or water bubbles. This fact is not critical since usually frames with food content or water bubbles tend to have a significant coverage and thus any cut has the same clinical interest during inspection and analysis.

4

Automatic feature extraction

4.1 Introduction

In Chapter 1, we have described intestinal motility events like: *contractions*, *static* and *intestinal content*. In this Chapter, we propose automatic methods for the detection and quantification of such motility events. To this end, we use computer vision techniques to quantify the motility information perceived by an expert while screening the WCE video (or motility bar). In Figure 4.1, the semantic of intestinal motility is shown. Following the intestinal events, we propose to use three main categories to describe intestinal motility: 1) occluded field of view (intestinal content), 2) intestinal activity (contractions) and 3) lack of intestinal activity (static and tunnel periods). For each of the categories, we define some descriptors e.g. frames representing bubbles, contraction density or lumen size (see Figure 4.1).

In the current Chapter, semantic descriptors are converted into numerical features (in particular we introduce: contraction detector, lumen size estimator, intestinal content detector and wrinkle frame detector). The methods, where possible, are based on the motility bar. Some events like: turbid and bubbles or wrinkles can not be perceived in the motility bar. Therefore in these cases, we develop systems based on frame-per-frame analysis. First, we describe algorithms that are based on the motility bar and, second, the ones that are based on frame analysis.

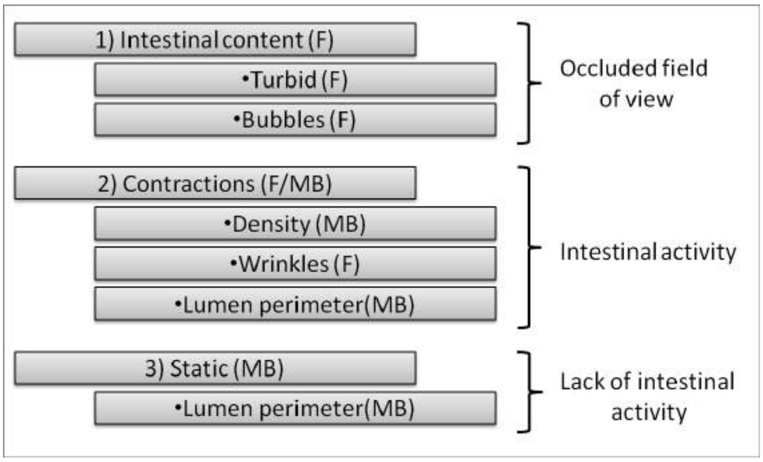


Figure 4.1: Three main categories describing intestinal motility. For each event we propose descriptors that are converted into numeric features using automatic detection systems. (F) means frame based detection, (MB) means motility bar based detection.

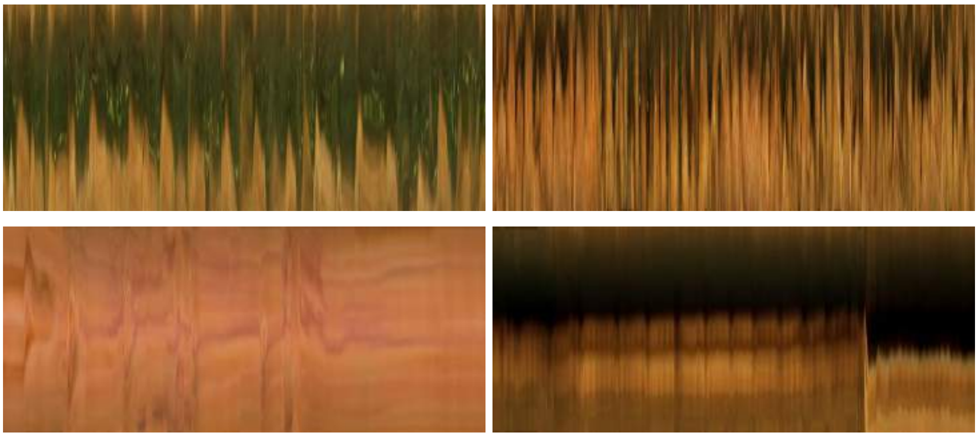


Figure 4.2: Some examples of different visual patterns that can be seen in motility bar.

4.2 Motility-bar-based features

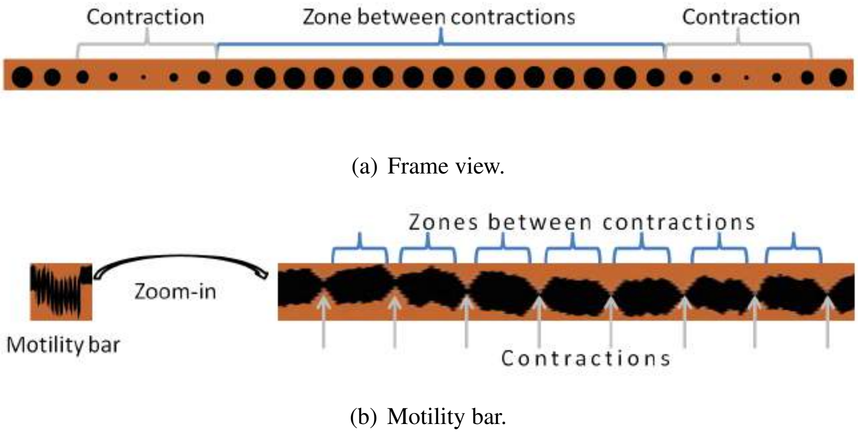
Visual inspection of intestinal motility patterns shown in the motility bar provides a clue that different segments can be generally characterized by two phenomena: contractile velocity and lumen size. Examples of different pieces of the motility bar with different contractile velocity and different lumen size are shown in Figure 4.2.

In this section, we propose systems for automatic analysis of the motility bar. In particular, we propose methods for: 1) characterization of intestinal contractions (also referred to as intestinal oscillations) and 2) estimation of lumen perimeter.

4.2.1 Contractile activity characterization

In this subsection, we propose the methods for automatic analysis of contractile patterns that are seen in the motility bar. In order to characterize the contractile events, we detect the valleys in the motility bar.

When analyzing examples of motility bar (e . g. see first line of Figure 4.2), one can conclude that the contractions are generally grouped, meaning that, it is unusual to see one isolated contraction. We refer to such group of contractions as contractile sequence. In order to fully understand the concept of contraction in the motility bar, let us present a synthetic example. As mentioned before, contraction can be described (in frame view) as open-close-open lumen sequence (see Figure 4.3(a)). Moreover, let as define the zone between two contractions as the sequence of open lumen between two wall closures (see Figure 4.3(a)). The same event can be



(a) Frame view.

(b) Motility bar.

Figure 4.3: An example of image representing contractile phenomena. The same event is shown in the frame view (sub-figure a) and the motility bar (sub-figure b).

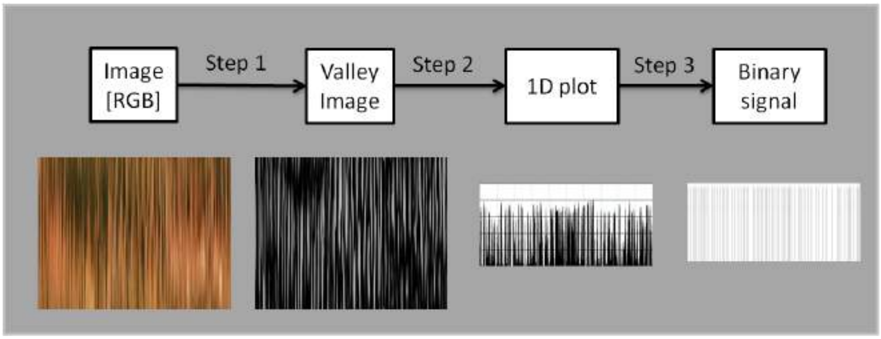


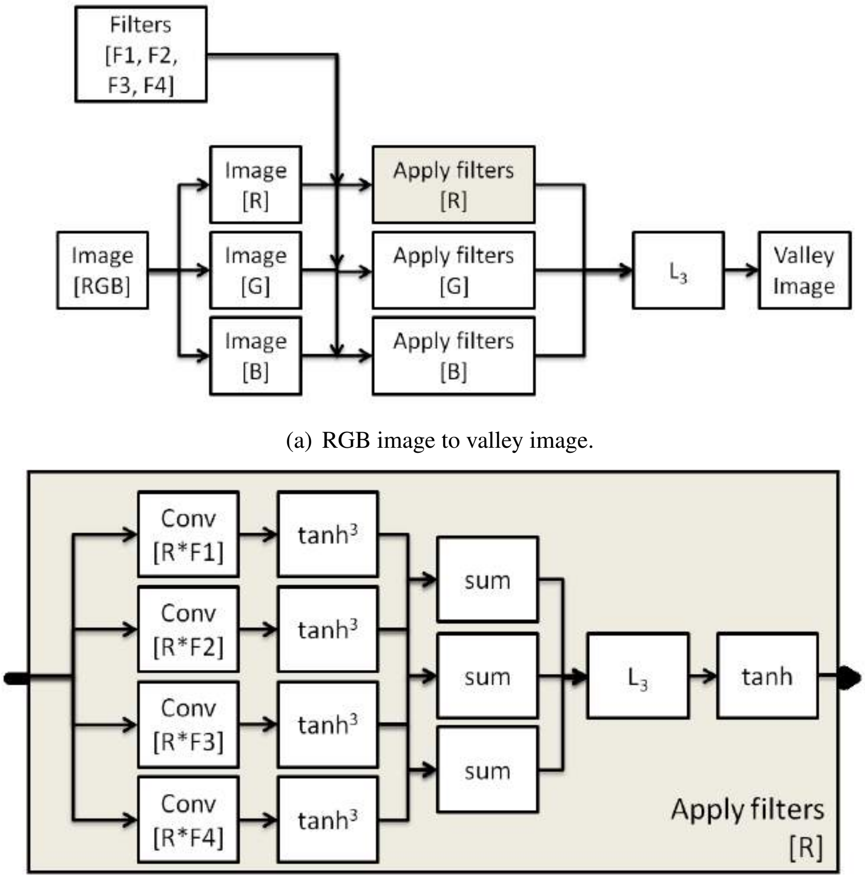
Figure 4.4: Contractile detection system pipeline.

represented in motility bar (see Figure 4.3(b), where a contractile sequence is presented). In order to estimate the density of the contractile activity one can measure either contractions or zones between contractions. In our set-up we count the number of zones separating contractions (instead of counting contractions) and refer to one separation zone as valley.

4.2.1.1 Methodology

The pipeline for analysis of contractile movements in motility bar is presented in Figure 4.4. The input to the method is a color image, the output is a binary signal with value 1 where the contractile oscillation is detected. To do so, the method performs 3 basic steps:

- **Step 1**: Apply a set of Gabor-like filters in order to detect valleys in the image.



(a) RGB image to valley image.

(b) Apply filter module.

Figure 4.5: Diagrams showing the steps of method for valley detection in the motility bar.

- **Step 2**: Convert the valley image into 1D signal representing valley positions.
- **Step 3**: Detect peaks representing contractions in valleys positions signal.

Step 1. From RGB image to valley image. The first step of the method is the detection of valleys in the motility bar. The architecture of the system is shown in Figure 4.5(a). We process separately R, G and B channels calculating the filter response for each one. Finally, we join the responses using L3 norm. Note that L3 norm favors high values.

There are two inputs to the method: an RGB image and a set of filters. We use 4 filters that correspond to the detection of valleys at different time scales (different contractile velocities). The filters that use correspond to second order derivative of gaussian filter and are defined in the direction of time axis of the motility bar, so that they look only for vertical valleys. All

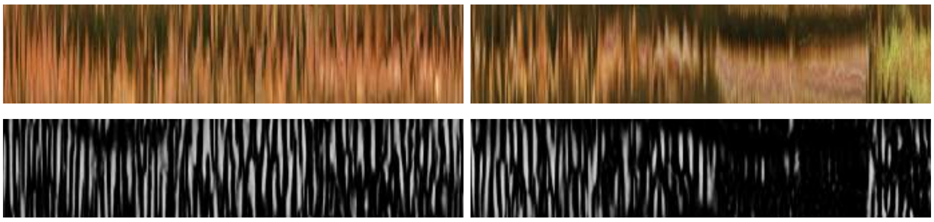


Figure 4.6: An example of the detection of valleys in the motility bar. First line shows two images of the motility bar, second line shows the detection of valleys in the motility bar (white intensity).

filters have a height of 10 pixels and are normalized to have mean 0 and energy 1. The filters can be characterized by scale parameter (the duration of the crest). We use the ones that detect valleys at the following scales:

- **Filter 1**: 10 frames, 5 seconds,
- **Filter 2**: 16 frames, 8 seconds,
- **Filter 3**: 22 frames, 11 seconds,
- **Filter 4**: 28 frames, 14 seconds,

Figure 4.5(b) shows how the filters are applied to each one of the channels. The first step of the method is a convolution of image with the filter. In the second step, we apply hyperbolic tangent non-linearity in order to normalize the filters responses into the range [-1, 1]. Since we are interested in valleys (and not in ridges), we set all negative responses to 0 (in the literature this step is referred to as rectified linear unit). In order to boost the high filter responses we elevate the results to the power of 3. We follow by summing up the result on neighbouring scales and joining the results using L3 norm.

Some examples of valley images are presented in Figure 4.6. As it can be seen, the method detects well the valleys of duration ranged between 10 and 28 frames (pixels).

Step 2. Valley image to 1D signal. In the second step, the valley image is converted to 1D signal. To do so, a percentile is calculated for each vertical line of the motility bar. Since we are not interested in shallow (not well marked) contractions, a percentile 75 is applied so that a restriction on the oscillation size is imposed. This restriction is important since some small oscillations are not contractions and can represent external oscillatory movements e. g. respiratory oscillations. The results are shown on the Figure 4.7.

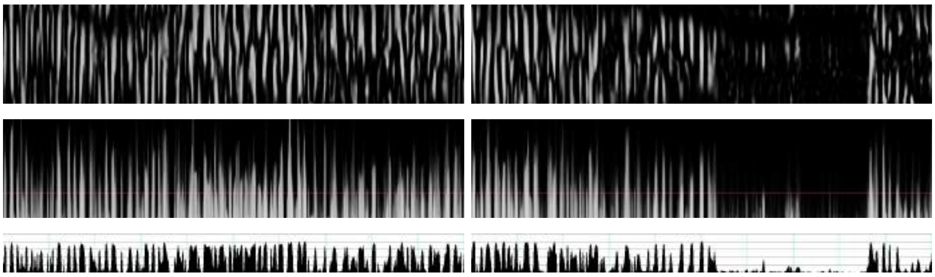


Figure 4.7: From valley image to 1D signal. The first line shows a valley image, the second line shows the result of sorting of each vertical line, the third line shows the result of applying percentile 75 to each vertical line of the valley image.

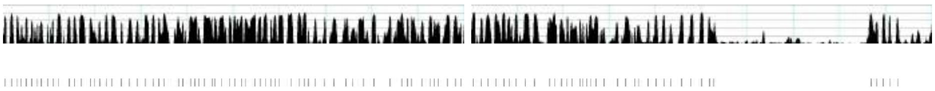


Figure 4.8: 1D signal to the binary image.

Step 3. 1D signal to binary image. In the last step of the method, the 1D signal is converted into a binary signal representing the detected contractile movements. In order to convert the signal to 1D signal the local maxima are detected. We impose 2 restrictions on local maxima: the minimum height of the peak, and the distance between two neighbouring local maxima (used to avoid two detections that separated by small amount of time). The result of the peak detection is shown in Figure 4.8. As it can be seen, strong peaks are correctly detected while low peaks are not classified as contractions.

Finally, the binary signal with contractile information is used to estimate the density of contractions. The density is estimated using 1 minute sliding window. The results are presented in Figure 4.9 (bottom line). The maximum value of density plot represents 12 contractions per minute, the minimal value indicates 0 contractions per minute.

4.2.1.2 Validation

The result of contractile density estimation for the whole video is shown in Figure 4.10. As it can be seen, the zones of the video with no intestinal movement have low density score (e. g. see the 8th line of Figure 4.10). By further analysis of the results, one can observe that the zones where the contractile movement is 'frequent' have high density score (e.g. see the central part of the second line of Figure 4.10). Moreover, it can be seen that the method detects oscillations with the presence of intestinal content (mixing contractions). For en example of

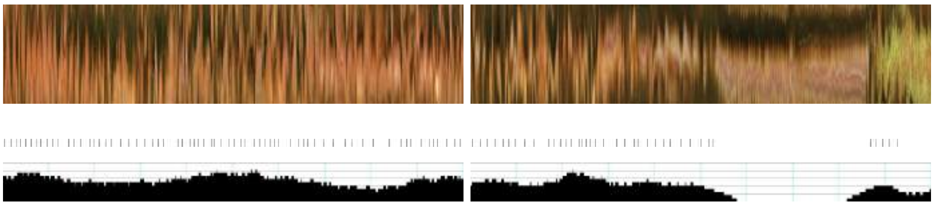


Figure 4.9: An example of the result. The image is reduced to binary signal representing contractions.

such contractions see the second line of Figure 4.10, where the intestinal content is mixed with variable contractile strength (oscillation height).

We implemented the method in Matlab and ran it on Intel I5-2520 CPU machine. The time needed to obtain contractile density estimation for a motility bar build from 28000 frames was approx. 30 seconds.

4.2.2 Lumen perimeter estimation

In WCE image, the lumen is visually characterized as a region with low illumination intensity since the capsula light is not reflected from the intestinal wall. Thus, the detection of the lumen should be based on the analysis of light intensity (for some examples of intestinal lumen see dark regions of the sequences in Figure 4.2).

4.2.2.1 Methodology

Lumen perimeter estimation algorithm is three-fold:

- **Step 1**: Reduce the noise in color distribution by applying mean-shift clustering.
- **Step 2**: Convert image to intensity image and apply threshold in order to obtain the regions of lumen.
- **Step 3**: Apply morphological operations in order to obtain smooth regions of intestinal lumen.

In order to reduce the noise in the color feature space, we use mean-shift method (46). Figure 4.11 shows the result of applying the mean-shift method in the RGB feature space. As it can be observed, the colors of WCE occupy only small subspace of RGB cube. The size of the mean-shift bandwidth is set so that the visual perception of the lumen is not modified.

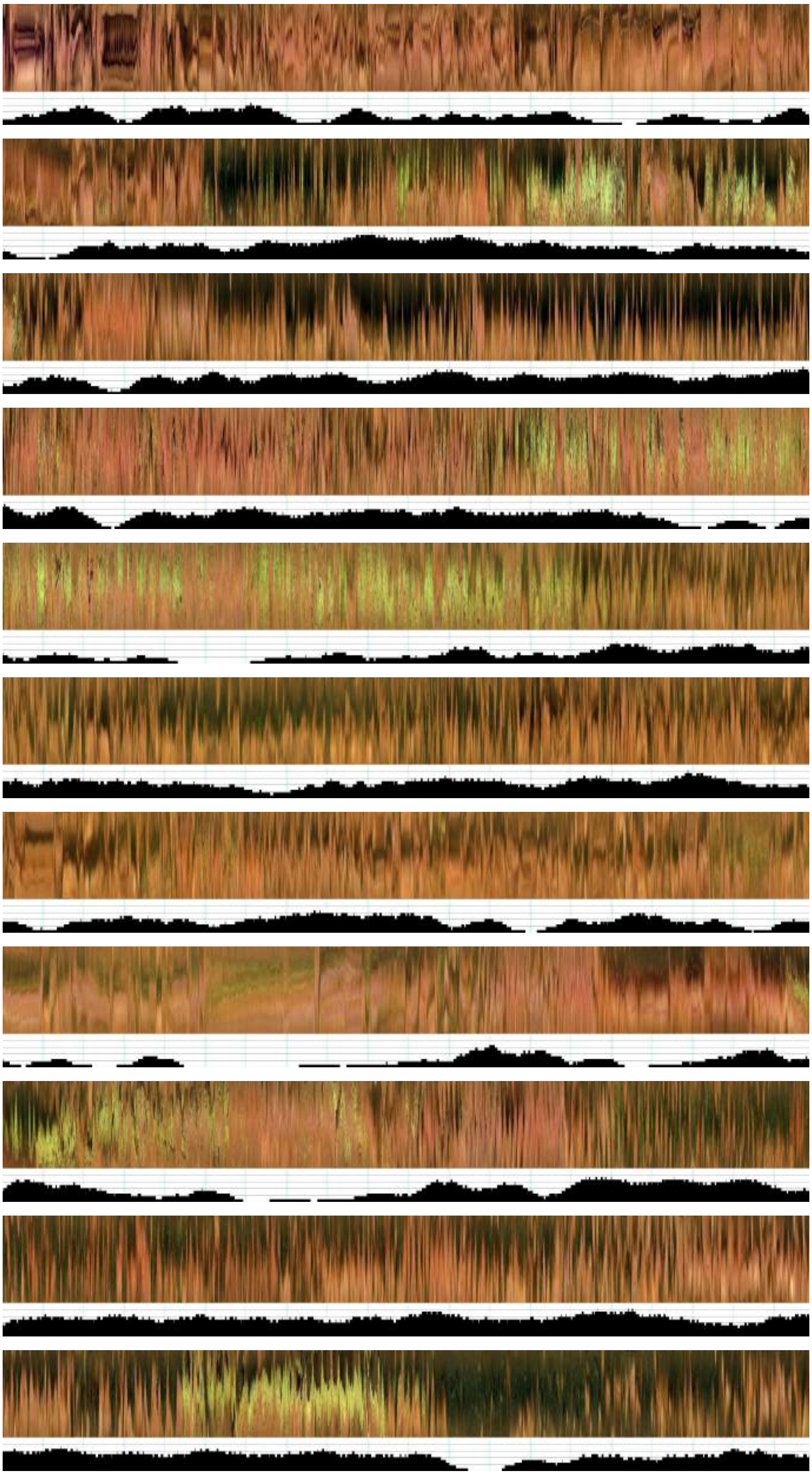


Figure 4.10: The result of contraction density estimation for a whole video of WCE.

Figure 4.12 shows the resulting image of applying the mean-shift method to the motility bar. It can be seen that all the contours of the lumen are well preserved.

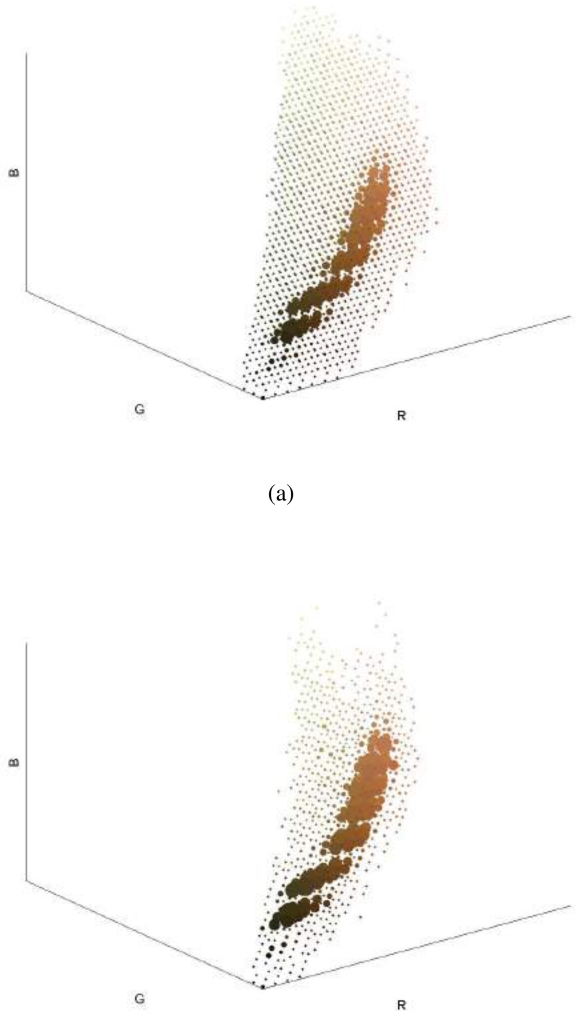
In the second step, the image is transformed to a gray scale image and a threshold is applied. The gray scale image has values in the range of $[0, 1]$ and the threshold is fixed to 0.3. As the result, the image is converted to black and white with black representing the lumen and white representing no lumen regions. In order to remove small regions of tunnel, morphological operations of opening and closing are applied. Figure 4.13 shows the results of lumen segmentation from the motility bar (black area). As it can be seen, the lumen segments coincide well with their original position in the motility bar.

Finally, we convert the segmentation results to a numerical value. The value represents the percentage of each cut that is occupied by intestinal lumen. This value is calculated for each line of the motility bar. Figure 4.14 shows the results for the lumen size estimation (bottom line), where the maximum value means that 100% of the motility bar represents lumen.

4.2.2.2 Validation

The result of the lumen size estimation for the whole video is shown in Figure 4.15. As it can be seen, the zones of the video with no visible lumen have low lumen size value (e. g. see the 5th line of Figure 4.15). By further analysis of the results, one can observe, that the zones where the lumen size occupies the whole motility bar shows high lumen size value (e.g. see the central part of the last line of Figure 4.15).

We implemented the method in Matlab and ran it on Intel I5-2520 CPU machine. The time needed to obtain lumen size estimation for a motility bar build from 28000 frames was approx. 110 seconds.



(a)

(b)

Figure 4.11: Density plots of colors for a motility bar. The size of dots reflects the density of space. a) before applying mean-shift filtering, b) after applying mean-shift filtering.

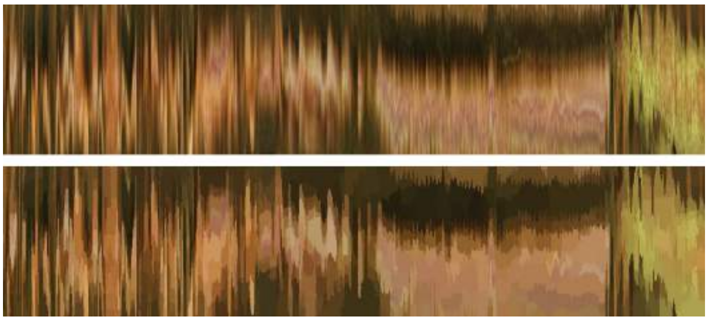


Figure 4.12: An example of motility bar. Top image - original motility bar, bottom image - motility bar after mean-shift.

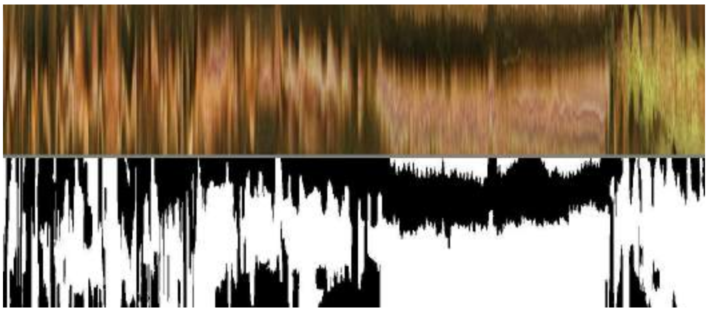


Figure 4.13: An example of motility bar segmentation. Top image - original motility bar, bottom image - segmentation of motility bar. The segmented lumen is shown in black color.

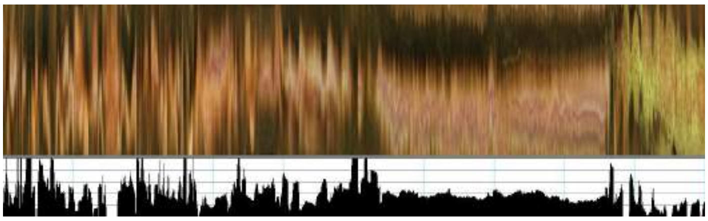


Figure 4.14: An example of perimeter estimation. Top image original motility bar, bottom image a bar plot representing an estimation of the percentage of each line of motility bar occupied by lumen.

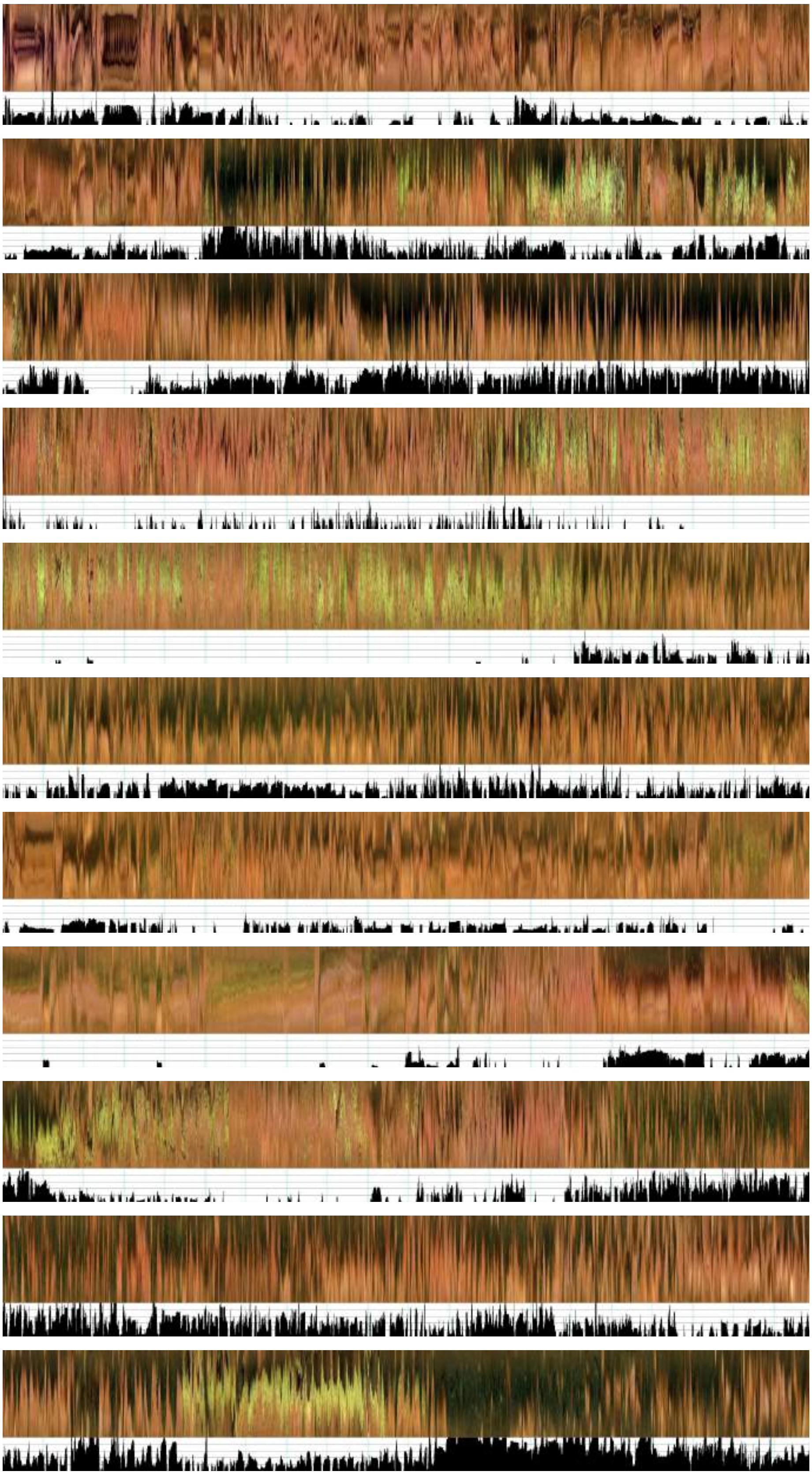


Figure 4.15: The result of perimeter estimation for a whole video of WCE.

4.3 Frame-based features

In this section, we propose algorithms for: 1) automatic detection and characterization of intestinal content frames, and 2) automatic detection of frames with wrinkle structure. The methods presented in this section are based on frame analysis.

4.3.1 Intestinal content frames detection

The proposed system is divided into two consecutive steps: 1) detection and 2) segmentation. The aim of the detection step is to target the segmentation of the intestinal content only in the frames where the intestinal content is present, reducing the computational cost. The advantage of the second step is two-fold: 1) it provides information about the percentage of the image covered by intestinal content and 2) it allows accurate measurement of the amount of turbid and bubbles. The complete system scheme is presented in Figure 4.16.

4.3.1.1 Methodology

Image Classification. In the first step, the frames with intestinal content are found. As explained in Section 1.2, there are two different types of intestinal content: turbid, which is characterized by color information, and bubbles, which are characterized by texture. In order to detect both types of intestinal content, two feature descriptors are used for each frame: a color histogram and a texture descriptor. These image features are merged to train a SVM classifier (47).

Color Features. We represent each frame by its color histogram. In order to reduce the image complexity, a color quantization is performed. As the result, the possible colors are reduced from 16 million colors to 64 colors. Typically, color quantization is performed dividing the original color space into smaller subregions of equal size (13). In WCE videos, only a subset of colors is observed and most of the observed colors in WCE are concentrated in a small region of the RGB space (e. g. see Figure 4.11). This information can be used to reduce the dimensionality of color representation with minimum visual loss to the 64 colors. We refer to this color representation as *Intes Color Map*. This map was created using a set of 80 WCE videos. The three-dimensional RGB data representing all observed colors were assigned into 64 clusters using k-means technique (48). In order to evaluate the quantization, mean errors

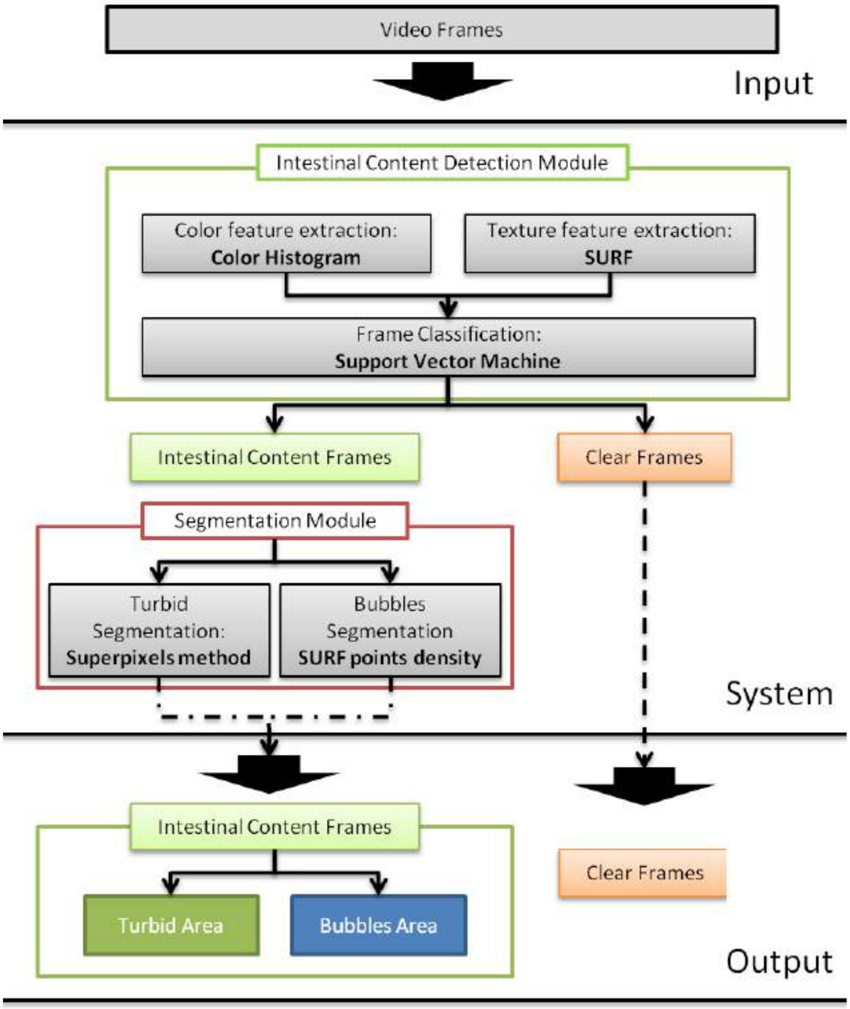

Video Frames
Input
Intestinal Content Detection Module
Color feature extraction: Color Histogram
Texture feature extraction: SURF
Frame Classification: Support Vector Machine
Intestinal Content Frames
Clear Frames
Segmentation Module
Turbid Segmentation: Superpixels method
Bubbles Segmentation SURF points density
System
Intestinal Content Frames
Turbid Area
Bubbles Area
Clear Frames
Output

Figure 4.16: Architecture of system for detection and segmentation of intestinal content.

were calculated: 1) 10.02 ($std = 9.98$) for *Intes Color Map* and 2) 33.33 ($std = 19.23$) for uniform quantization. The errors were defined as mean distance to the nearest centroid.

Textural Features. We propose to use Speeded-Up Robust Feature (SURF) (49) as bubble frame descriptor. This method is a fast, scale- and rotation-invariant blob detector and descriptor. The SURF method can be applied to the problem of bubble detection since bubbles are blob-like structures. We represent each frame by the number of blobs detected and our assumption is that in non-bubble image small number of blobs are present. The SURF method uses an integer approximation of the determinant of Hessian matrix as blob detector and the detections are referred to as SURF points or as points of interest.

Classification. In order to classify the frames into classes $\{$*intestinal content, clear*$\}$, both color and textural features are considered. This is done by simply expanding the color histogram by one extra bin representing the number of points of interest. Therefore, we obtain a 65 dimensional feature vector, which is classified using a Support Vector Machine (SVM) classifier (47). Since our features are represented by histograms, we use the Histogram Intersection Kernel (50) defined as follows:

$$K_{int}(z, z') = \sum_{j}^{m} \min(z_j, z'_j) \tag{4.1}$$

where $z = \{z_1, .., z_m\}$ and $z' = \{z'_1, .., z'_m\}$ are the histograms with $m - 1$ bins representing color information and one bin representing the number of points of interest.

Image Segmentation. Each frame classified as *intestinal content* is further processed by segmentation module and, as a result, the regions of image are labeled as *clear*, *turbid* or *bubbles*. In order to obtain this segmentation two approaches are presented 1) color-based intestinal content segmentation and 2) texture-based bubble region segmentation.

Color-based intestinal content segmentation. In order to obtain the exact area covered by intestinal content in the image (which includes both turbid and bubbles), each pixel should be labeled as *intestinal content/clear* frame. We use the *Superpixel* method (51) to obtain homogeneous regions. *Superpixels* are obtained using Normalized Cuts (NCuts) (52). NCuts is an algorithm, which employs spectral clustering to exploit pairwise brightness, color and texture affinities among pixels. Rather than focusing on local features and their consistencies in the images, the aim of the NCuts consists of extracting a global image impression. The number of *Superpixels* is an input parameter and can be set using cross-validation. Each *Superpixel* region is classified using a linear SVM classifier. As a descriptor of each region, we use the mean intensity of the pixels for each channel R, G and B (see Algorithm 1).

Texture-based bubble region segmentation. Finally, bubble regions are detected. To do so, the density of bubbles in a frame is estimated. Areas with high bubble density are considered to be bubble regions. Density is estimated using kernel density method. Let s be a

Algorithm 1 Algorithm for intestinal content segmentation

```
Input: image $I$ and number of regions $N$
  Compute $N$ regions $R_N$ using NCuts method.
  for $i = 1$ to $N$ do
    Compute mean values of $R_i$ in R, G and B channels, $f_i = [f_i^r, f_i^g, f_i^b]$.
    Classify $f_i$ using linear SVM and assign boolean value representing $\{intestinal\ content, clear\}$ to all pixels in $R_i$
  end for
Output: Binary image representing segmented regions $\{intestinal\ content, clear\}$.
```

location in the image I and $p_{1..n}$ be the locations of the interest points detected by SURF. The density estimation using the kernel method is given by:

$$\hat{f}_k(s) = \frac{1}{\sigma_k(s)} \sum_{i=1}^{n} \frac{1}{h^2} k\left(\frac{s - p_i}{h}\right) \tag{4.2}$$

where $\sigma_k(s)$ is the correction for edge effects for location s, k is the kernel and h is the bandwidth. We use the quadratic kernel proposed in (53):

$$k(\mathbf{u}) = \frac{3}{\pi}(1 - \mathbf{u}^t\mathbf{u})^2 \qquad \mathbf{u}^T\mathbf{u} \leq 1 \tag{4.3}$$

When this kernel function is introduced in Formula (4.2) and $\sigma_k(s)$ is fixed to 1, the following density estimate function is obtained:

$$\hat{f}_k(s) = \sum_{d_i \leq h} \frac{3}{\pi h^2}\left(1 - \frac{d_i^2}{h^2}\right)^2 \tag{4.4}$$

where d_i is the distance between location s and SURF points p_i. Finally, given the bubble density image, the bubble area is defined as the region where the density value is higher than a threshold thr_b.

Final labeling. According to the results of two proposed segmentation methods, the system output is defined as:

- *Bubbles*: All image pixels that belong to a bubble area.
- *Turbid*: The set of image pixels considered as intestinal content and not considered as bubbles.
- *Clear*: All other image pixels (not bubbles and not turbid).

Table 4.1: List of videos with the corresponding number of clear and intestinal content frames.

Video	#clear/IC frames	Video	#clear/IC frames	Video	#clear/IC frames	Video	#clear/IC frames	Video	#clear/IC frames
Video1	1327 / 672	Video11	993 / 1006	Video21	615 / 1225	Video31	1857 / 143	Video41	1788 / 212
Video2	1451 / 549	Video12	992 / 353	Video22	1815 / 140	Video32	1766 / 234	Video42	988 / 1012
Video3	1205 / 795	Video13	1369 / 470	Video23	200 / 1453	Video33	1892 / 108	Video43	1769 /232
Video4	1008 / 992	Video14	1184 / 499	Video24	457 / 1535	Video34	1921 / 79	Video44	1886 / 114
Video5	1135 / 865	Video15	434 / 1389	Video25	1383 / 298	Video35	1517 / 483	Video45	1406 / 594
Video6	1530 / 434	Video16	502 / 1375	Video26	1904 / 85	Video36	1993 / 7	Video46	1307 / 693
Video7	1346 / 453	Video17	1418 / 192	Video27	1390 / 527	Video37	1998 / 2	Video47	1041 / 959
Video8	1203 / 797	Video18	750 / 738	Video28	709 / 847	Video38	1631 / 369	Video48	1120 / 880
Video9	1337 / 613	Video19	1556 / 302	Video29	223 / 1763	Video39	1974 / 26	Video49	1743 / 257
Video10	624 / 1261	Video20	1288 / 476	Video30	828 / 867	Video40	1762 / 238	Video50	1714 / 286

4.3.1.2 Validation

In this section, we present the experimental results of the proposed system for automatic characterization of intestinal content frames. First, we describe the data set and the evaluation procedure and then, we show the qualitative and quantitative results of all parts of the proposed system. More precisely, we present the validation of:

- SURF detector for bubble frames detection;
- Intestinal content detector;
- Intestinal content segmentation.

Database. The data set was obtained using the SB2 capsule endoscopy camera developed by Given Imaging, Ltd., Israel (1). All cases were conducted in the same conditions at the Digestive Diseases Department, Hospital General "Vall d'Hebron" in Barcelona, Spain (9). For the experimental setup, a set of 50 studies from different subjects was used. For each video, the duodenum and cecum entrance were marked by a medical expert. A random set of frames from each video was selected and then labeled as $\{intestinal\ content, clear\}$. The number of frames per video is a number between 1000 and 2000 depending on the video length. These frames represent between 5% to 10% of the video frames from the duodenum until the the cecum. Table 4.1 shows the list of videos used in the experiments, indicating the number of frames from each class. As it can be observed, there is a high variability in terms of percentage of intestinal content in videos: there are some videos which practically do not present intestinal content (video 36 and 37) and there are others where intestinal content is present in more than 80% of the frames (video 29).

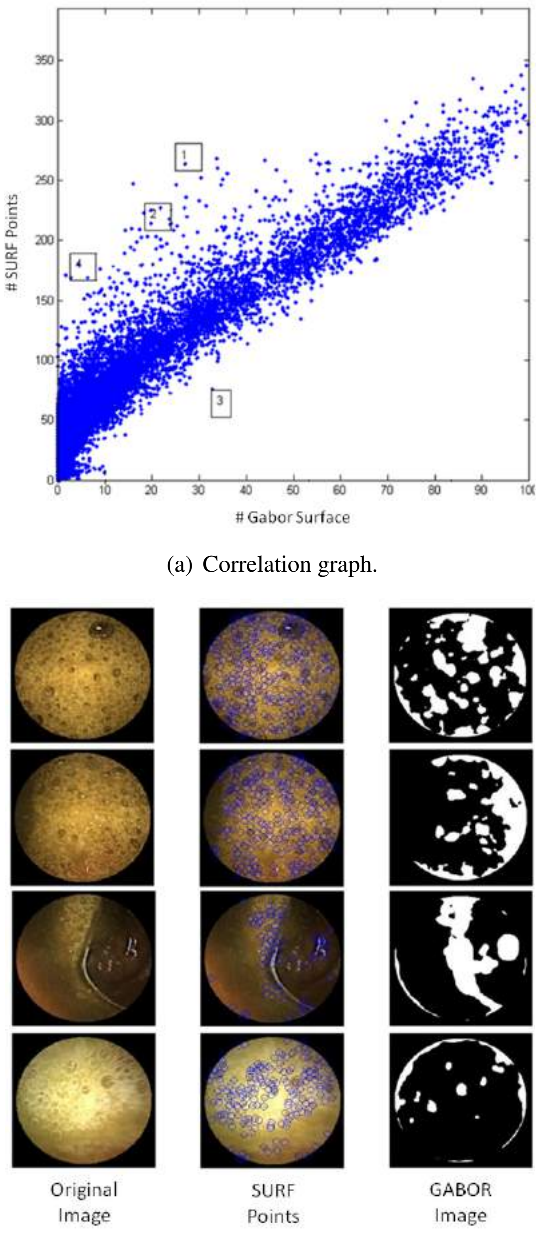
SURF detector validation for bubble frames detection. In this experiment, we compare the results of the SURF method with the method from (14), which estimates the bubble area using Gabor filters. A threshold for the SURF method thr_surf is fixed by cross-validation. In Figure 4.17, we present a scatter plot showing the correlation graph between the output of both methods: the number of Surf points and the surface area estimated by Gabor filters. Pearson correlation coefficient r is used to evaluate the output of both methods. The obtained value ($r = 0.95$) indicates that the methods are highly correlated. As it can be seen, there are only some samples which present a significant difference between the methods. In the same figure, we show four images (marked with blue square) where the methods present low correlation. As it can be seen, the qualitative analysis of these outliers shows that the proposed method performs better than the Gabor filter, for the case of blurred bubbles.

Intestinal Content Classification. In this paragraph, we evaluate the system for intestinal content detection. In order to assess the method, a leave-one-video-out validation method is used. The following measurements are used to evaluate the classifier:

- Accuracy (A) = $\frac{TP+TN}{TP+FP+TN+FN}$
- Sensitivity (S) = $\frac{TP}{TP+FN}$
- Specificity (K) = $\frac{TN}{TN+FP}$
- Precision (P) = $\frac{TP}{TP+FP}$

where TP = true positive, TN = true negative, FP = false positive and FN = false negative. The frames with intestinal content are considered as positive samples and the clear frames as negative cases.

We compare the results of our system with two methods proposed in (13): 1) *Color Moment Features* and, 2) *HSV 64 bin color Histogram* features. Additionally, we test a simplified version of our proposed system that uses only 64 bin color histogram (without texture information). The results are presented in Table 4.2 where the mean value and the standard deviation of the different methods are presented. We can see that the proposed method achieves the best results, outperforming other methods in all measurements (accuracy, sensitivity, specificity and precision). The box plots of accuracy are presented in Figure 4.18, where it can be seen that the proposed method has the smallest variance.



(a) Correlation graph.

(b) Outliers of the comparison between the methods.

Figure 4.17: The correlation between Gabor and SURF methods, both applied to the detection and segmentation of bubble frames. Figure a) shows the correlation graph ($r = 0.95$) between the two methods. Each point in the graph represents single frame. Ordinate axis represents the % of frame surface covered by bubbles following the Gabor method. Abscise axis represents the number of SURF points detected in that frame. With numbers 1, 2, 3 and 4 some outliers have been marked. The outliers and the output of both methods are shown in Figure b).

Table 4.2: Accuracy of intestinal content detection methods

	Accuracy	Sensitivity	Specificity	Precision
Color Moments	83.6 ± 10.9%	54.4 ± 24.1%	92.3 ± 10.5%	73.4 ± 26.9%
HSV 64bin	89.9 ± 7.8%	73.7 ± 22.6%	93.1 ± 9.5%	83.8 ± 21.8%
IntesColorMap	91.2 ± 6.9%	78.3 ± 17.7%	92.8 ± 8.1%	82.5 ± 18.4%
IntesColorMap + Bubbles	91.6 ± 6.6%	80.1 ± 16.7%	93.1 ±7.9%	83.0 ± 18.2%

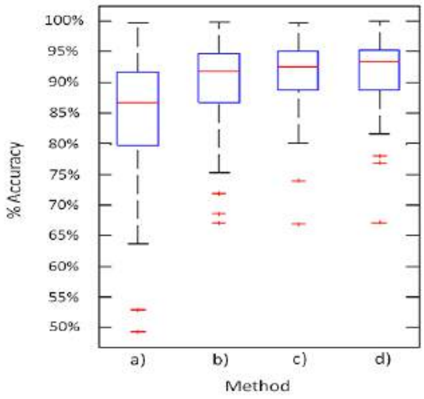


Figure 4.18: Box plots of intestinal content classification results using different sets of features: a) Color Moment Features b) HSV histogram with 64 bins c) *Intes Color Map* histogram with 64 bins and d) *Intes Color Map* histogram with 64 bins + bubbles. On each box, the central mark is the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually.

We implemented the method in Matlab and ran it on Intel I5-2520 CPU machine. The time needed to obtain intestinal content frames for a video of 28000 frames was approx. 1000 seconds.

Study of reliability. A classifier is reliable when the training data used in the model construction process represents well the future data. In order to ensure that our approach is accurate and consistent with respect to new videos, analysis of the training set is done. In this analysis, the 50 WCE videos are used (for details, see Table 4.1).

In the experiments presented in this section, two questions are tackled. First, we would like to address the problem of the turbid variability between different subjects. Second, we would like to determine the minimum number of videos that builds a reliable classifier.

The results on turbid variability between different subjects are presented in Figure 4.19 and Figure 4.20. Figure 4.19 represents the results of training the classifiers with one video and testing the classifier accuracy with remaining 49 videos. The experiment was repeated 50 times (once for each video in the training set). As it can be seen, the variability of the accuracy is quite high. For instance, there are some videos (#10, #21 and #38) that, when used for training the classifier, give good and generalizable models for the remaining 49 videos. For these videos the median accuracy is high and the variance is small. These videos contain high variability of both intestinal content and clear frames that well represent the images in the remaining videos. On the other hand, some models obtained by using videos #36, #37 or #39 are not able to generalize the results in the remaining set of videos, the median accuracy is low and the variance is high. There are two possible justifications to this observation: 1) those videos are very homogenous and they do not offer enough information to learn a good classifier, or 2) these videos contain information (e.g. strange turbid color) that is not seen in other videos (they can be outlier videos). Those videos are interesting because they can provide new information about the data distribution that might be helpful in the discriminative model construction.

A second experiment, designed to study the variability between different subjects, shows how well a video is represented by a set of videos (see Figure 4.20). Here, each video is classified using 49 classifiers trained with the data from the remaining videos, the experiment was repeated 50 times, once for each video. As it can be seen, there are some videos (#33 and #37) that are well classified by all trained models. They have high median accuracy and small variance. These are homogenous videos with frequently appearing images in the other videos. On the other hand, there are some videos (#23 and #29) which are misclassified by the majority of classifiers. These videos contain frames with color and texture that are not frequently observed in WCE video.

Usually, the larger training set, the better classification results. However, it is important to remember that data acquisition and labeling have an associated cost limiting the size of the training set. To evaluate the influence of the training set size on the classifier accuracy, a set of 10 test videos were randomly selected from a pool of 50 videos and were classified using different training sets sizes (from 1 to 40 videos). At each iteration of the test one video is added to the training set. The results are presented in Figure 4.21, where a learning curve for each video is presented. We can observe that, when the training set contains 30 or more videos, the classifier accuracy stabilizes. Moreover, we see that for some videos a high accuracy is achieved using a small training set size. These results show an asymptotically convergent learning curve, which appears to assess the validity of the size of our data set.

Segmentation. Finally, the segmentation methods are evaluated. In order to evaluate the methods, a set of 350 images was selected. These images were manually selected by an expert

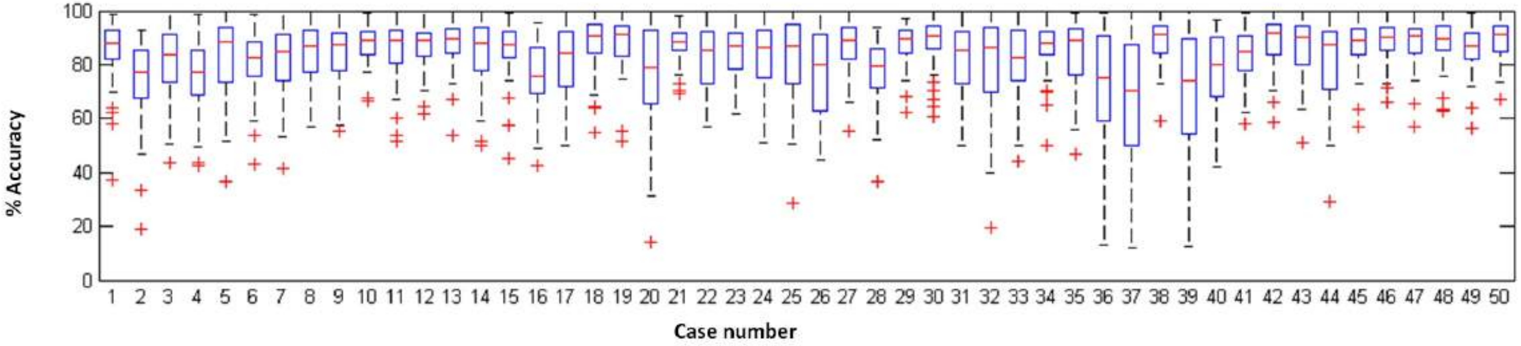


Figure 4.19: Each boxplot represents the accuracy obtained by testing a classifier learned by the video of x-axis with all other 49 videos in our dataset.

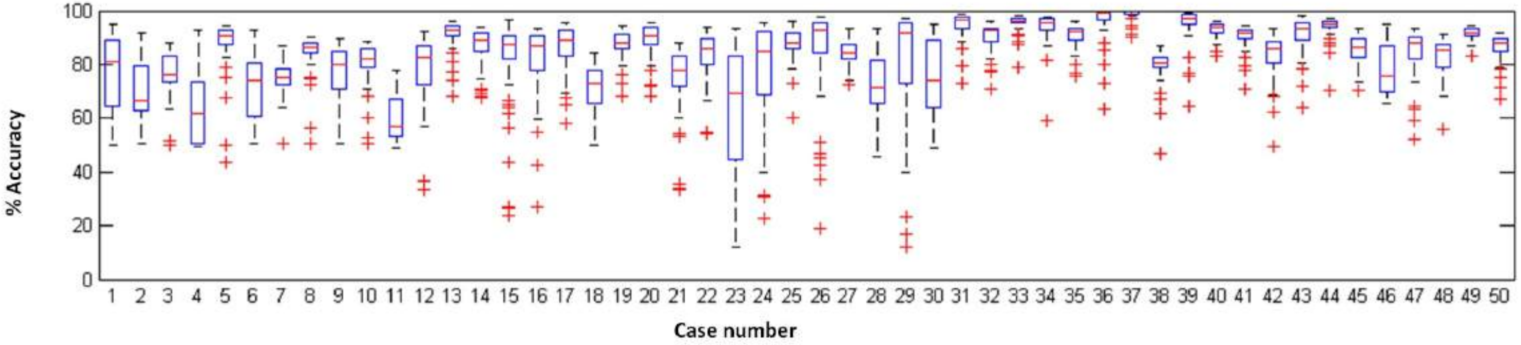


Figure 4.20: Each boxplot represents the accuracy obtained by testing the video represented by x-axis using 49 different classifiers learned using one different video in each case.

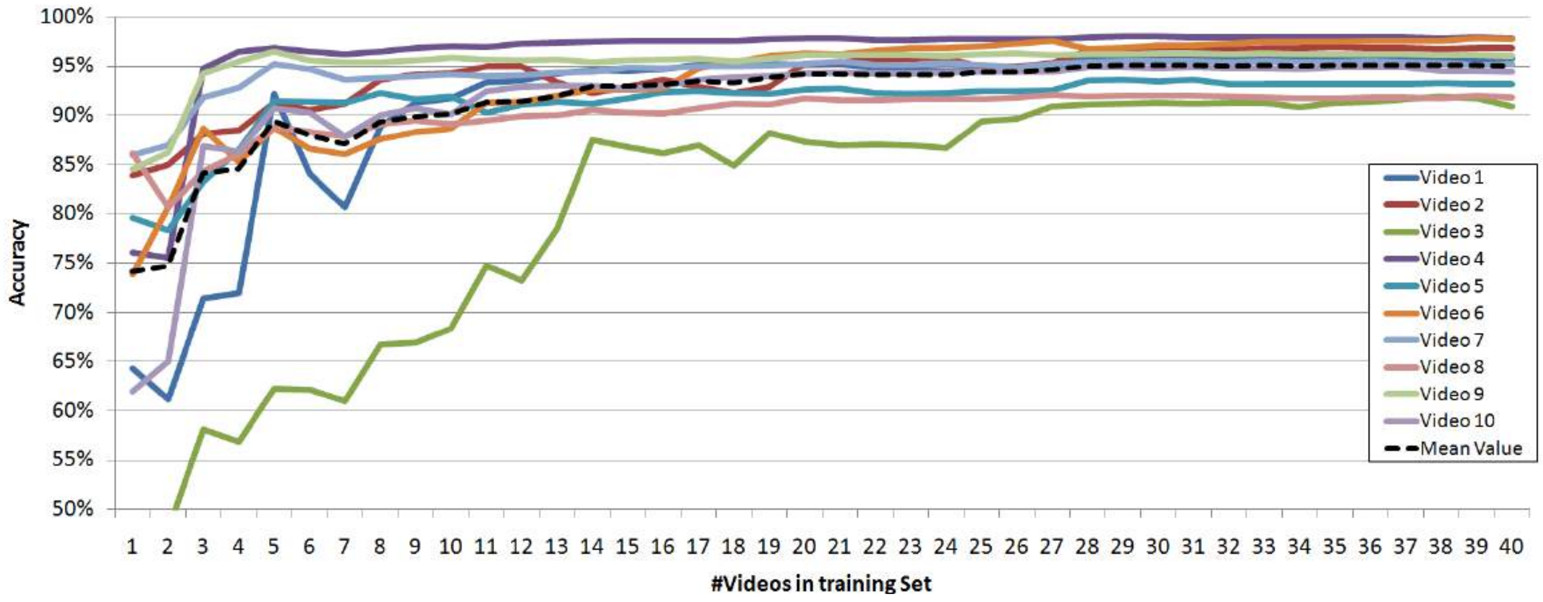


Figure 4.21: System accuracy for 10 videos from different subjects. X-axis represents the number of videos in the training set.

with the purpose of selecting a set of frames with high variability in terms of content percentage, texture and color. The manual annotation of regions were done by three different experts assigning to each region of the image one of the following labels: *clear*, *turbid* and *bubbles*.

The manual segmentation of intestinal content is a complex task. This complexity sometimes leads to an ambiguity between annotations from the same/different experts. The uncertainty of the experts arises from the variability of intestinal content. Frequently, the limits between the intestinal content and the intestinal wall or the lumen are questionable while a clear contour is not preserved. Moreover, the variability is higher in case of semi-transparent intestinal content. In order to evaluate intra-observer variability, the overlapping area between the annotations of the three experts was calculated. Table 4.3 summarizes intra-observer variability. As it can be observed, the observer variability on bubbles regions is low, presenting an overlap higher than 99% between the experts. However, turbid regions present variability between annotations of 10%.

Table 4.3: Intestinal content segmentation: User-variability

	Overlapping Area		
	Expert 1&2	Expert 1&3	Expert 2&3
Turbid	91.86%	89.71%	92.63%
Bubbles	99.04%	99.41%	98.30%
Intestinal Content (Bubbles + Turbid)	91.22%	89.39%	91.99%

Validation of color-based segmentation. The number of *Superpixel* regions was set to 60 and the regularization parameter of SVM classifier was set to 1 after cross-validation. In Figure 4.22, qualitative results of 9 different images for color-based intestinal content segmentation are presented. First column shows WCE frame, second column shows the *Superpixel* regions and third column shows the regions classified as intestinal content. As it can be observed, both turbid and bubble regions are classified as intestinal content regions in most of the images.

Validation of texture-based segmentation. The bubble segmentation method has two parameters: the kernel bandwidth $h = 50$ and the threshold $thr_b = 0.1$. The qualitative results of the method are presented in Figure 4.23. For each evaluated frame, the following information is presented: 1) the original image, 2) the output of the density estimation method and 3) the final binary output. As it can be seen, only the image regions containing bubbles are detected.

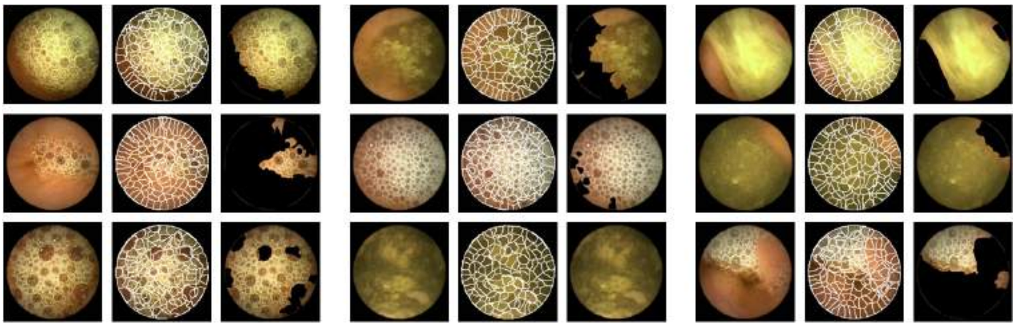


Figure 4.22: Qualitative results of color-based intestinal content segmentation obtained from 9 random images.

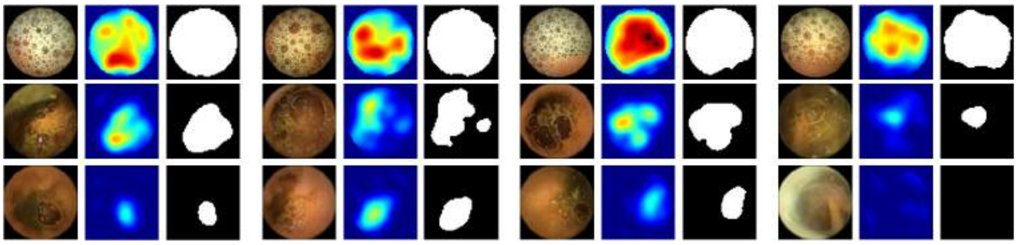


Figure 4.23: Qualitative results of texture-based bubble region segmentation obtained from 12 random images.

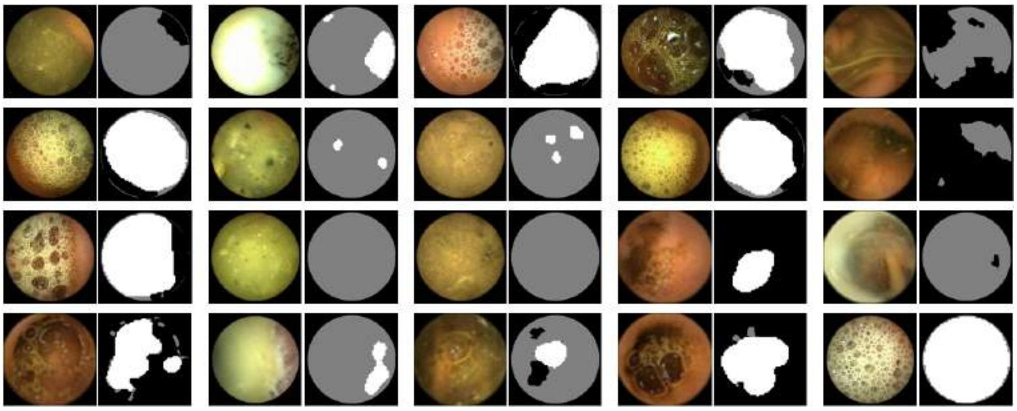


Figure 4.24: Results of intestinal content segmentation for 20 random images. Black areas mean clear region, white mean region with bubbles and grey areas mean turbid regions.

Overall system results. In Figure 4.24, we present the overall qualitative results. A set of 20 random images from the test set is shown. For each image, we present the segmentation output with the associated labels {*clear*, *turbid* and *bubble*}. The labels are represented with the following colors {*black*, *gray* and *white*} respectively. In this figure, we can observe that by using both segmentation systems, we are able to automatically differentiate between *clear*, *turbid* and *bubble* regions.

Finally, in Table 4.4 the quantitative results are presented. This table summarizes the overlapping area between the annotations of the 3 experts and the result of the segmentation (*turbid*, *bubbles* and *intestinal content*). We can appreciate that the bubble segmentation method outperforms the results of the turbid segmentation.

Table 4.4: Intestinal content segmentation results

	Overlap Area		
	Expert 1	Expert 2	Expert 3
Turbid	78.04%	81.60%	79.16%
Bubbles	92.43%	92.25%	92.60%
Intestinal Content (Turbid + Bubbles)	81.71%	85.28%	82.88%

4.3.2 Wrinkle frames detection

The second frame based detector deals with the problem of wrinkle frame classification. Given a WCE frame, the proposed method to detect wrinkle structures is divided into the following four steps:

1. In the first step, we compute, for each image pixel, a matrix that represents the predominant curvature directions in the pixel neighborhood by using the Hessian matrix. This descriptor provides clear and discriminant information about wall folds.

2. In the second step, the eigen-decomposition is applied to the Hessian matrix to compute its eigenvalues (λ_1, λ_2) and the corresponding eigenvectors, $(\mathbf{e}_1, \mathbf{e}_2)$. Let λ_1 be the eigenvalue with the highest absolute value (see Figure 4.25(b)). We construct a set of local histograms describing the orientation distribution of $\mathbf{e_2}$ eigenvector in a similar way as it is done with the well-known Histogram of Gradients (HoG) (18) (see Figure 4.25(c)).

3. Then, a mid-level image descriptor is obtained by transforming the set of local histograms into a graph and computing a centrality measure of each node (see Figure 4.25(d)).

4. Finally, a Structured Output Support Vector Machine classifier trained with mid-level features is applied, by following a sliding window approach to detect the presence of wrinkle structures in the image.

In the following subsections, we give details of each step of our approach.

4.3.2.1 Methodology

Feature Extraction: Hessian Matrix. In order to detect the intestinal wrinkles, we need a descriptor that is able to discriminate the shape of these specific image features.

The Hessian Matrix is a matrix derived from the second order derivatives of the image that summarizes the predominant directions of the local curvatures and their magnitudes in a neighborhood of a point. The Hessian matrix, $Hess_\sigma$ of an image I is a symmetric 2×2 defined as:

$$Hess_\sigma(p) = \begin{bmatrix} G(\sigma) * I_{xx}(p) & G(\sigma) * I_{xy}(p) \\ G(\sigma) * I_{xy}(p) & G(\sigma) * I_{yy}(p) \end{bmatrix} \tag{4.5}$$

where I_{xx}, I_{xy}, I_{yy} are the second order partial derivatives of the image I with respect to x and y coordinates, $p = (x, y)$ is an image point, $*$ is the convolution operator and $G(\sigma)$ is the

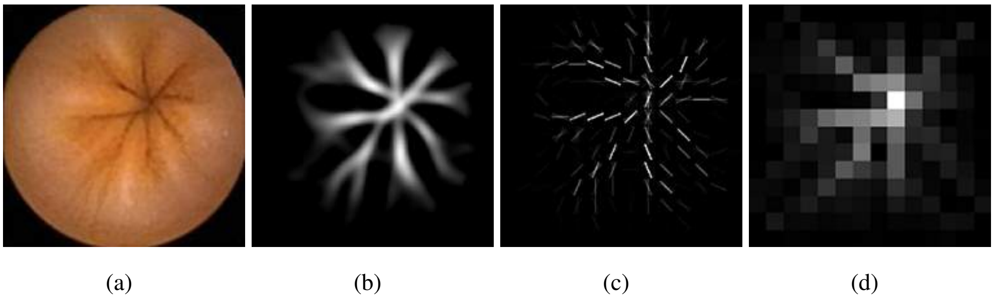


Figure 4.25: An example representing the important steps of the proposed methodology: a) Original image; b) $\max(0, \lambda_1)$; c) Histogram of oriented features computed from $(\lambda_1, \mathbf{e_2})$; d) Centrality descriptor calculated by transforming the histogram of oriented features into a graph (darker cells indicate low centrality values and lighter cells indicate high centrality values).

Gaussian function. Particularly interesting are the eigenvalues (λ_1, λ_2) and the eigenvectors $(\mathbf{e}_1, \mathbf{e}_2)$ of the Hessian matrix. Let λ_1 be the largest eigenvalue by absolute value, $|\lambda_1| > |\lambda_2|$. $|\lambda_1|$ shows the strength of the local image curvature and its corresponding eigenvector $\mathbf{e}_1$ is aligned with the dominant curvature direction of the image within a window defined by σ. The second eigenvector is orthogonal to the dominant curvature direction, generally pointing towards the direction of the least curvature.

Wrinkle structures can be associated to image valleys (54). For this reason, λ_1 represents at every image pixel a *wrinkleness* measure that can be used to detect foldings of the intestinal wall. In order to select these points, we consider for every pixel the map represented by $\max(0, \lambda_1)$. An example that illustrates this procedure is presented in Figure 4.26. In Figure 4.26(a), it is shown that the considered map defines tubular image structures at scale σ and can be used as an indicator of wrinkle presence. In fact, we have observed that pixels corresponding to low curvature regions do not carry any interesting information for wrinkle detection. For this reason, we apply an adaptive threshold t fixed by cross-validation that selects the 30% of image pixels with the highest values of $\max(0, \lambda_1)$. In Figure 4.26(b), it is shown that the second eigenvector $\mathbf{e}_2$ is aligned with the wrinkle direction, and consequently, it points towards the closed lumen.

Feature Representation: Histogram of Features. The computation of the Hessian-based descriptor on an image produces a two-dimensional vector for every pixel (see Figure 4.26(b)). In order to reduce the dimensionality of this representation and also to increase its invariance to scale and position variations, we decompose the image into a set of M small squared cells and compute a histogram over orientation bins. The angle of $\mathbf{e_2}$ is used to vote on the corresponding orientation bin. The vote is weighted by λ_1. Votes are accumulated over all pixels within

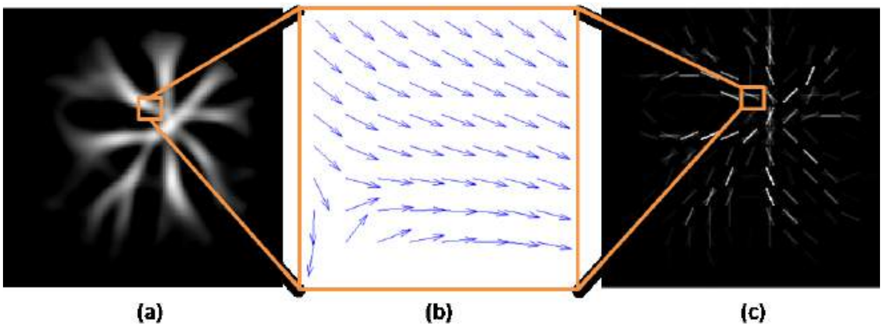


Figure 4.26: From image to histogram: a) max$(0, \lambda_1)$; b) Orientations of the second eigenvector of one selected cell; c) Histogram of oriented features computed from $(\lambda_1, \mathbf{e_2})$;.

each cell. Following the classical HoG, an image descriptor is then built by concatenating the values of the bins of all histograms, getting a high-dimensional vector $H = (\mathbf{h}_1, \dots, \mathbf{h}_M)$ that represents the image, where $\mathbf{h}$ is a cell histogram. This image representation is shown in Figure 4.26(c).

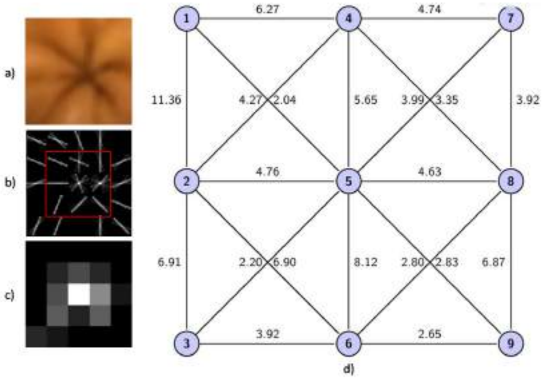


Figure 4.27: An example of centrality measure calculation for an image (a). A histogram of oriented features (b) can be transformed into a graph (d) by defining a node for each cell and a set of edges connecting all neighbor cells. Each edge can be labeled with a weight value representing the connectivity degree of the pair of nodes (i, j) it links, which can be computed from the histograms of their corresponding cells, $\mathbf{h_i}$ and $\mathbf{h_j}$. Then, a centrality measure (c) can be computed on this weighted graph structure.

Mid-level descriptor. If we consider $(\lambda_1, \mathbf{e_2})$ to be a low-level image descriptor, a mid-level descriptor would be a continuous or discrete numeric measurement obtained after a global analysis of the interactions between the values of $(\lambda_1, \mathbf{e_2})$ in the whole image. In our case, we are interested in a discriminant descriptor to characterize the prototypical star-like pattern that represents wrinkle frames. To this end, we define a new mid-level image descriptor, that we call *image centrality*, which is based on the betweenness centrality measure that was originally proposed to analyze social networks (55). If we consider the graph where each image cell corresponds to a node and links are only defined for pairs of neighboring cells, then, this descriptor defines for each image cell a robust measure of its centrality in terms of its probability to occur on a randomly chosen shortest path between two randomly chosen cells. The following subsections describe how the histogram is transformed into a graph and how centrality of each graph node can be computed.

From histograms to graph. The graph can be formally defined in the following way. Let M be the number of image cells resulting from dividing the image I into cells of $n \times n$ pixels. Let $H = (\mathbf{h}_1, \ldots, \mathbf{h}_M)$ be the histogram of oriented features computed from $(\lambda_1, \mathbf{e_2})$. Let $\mathbf{h}_i(\alpha)$ be the value of the orientation bin of cell i corresponding to the votes of $\mathbf{e}_2$ oriented at α degrees. Then, we can build a weighted graph $\mathcal{G} = (V, E, \mathbf{A})$ by considering a set of M nodes V, where each node $v_i \in V$ corresponds to cell i of I, a set of edges E, where edge $e_{i,k} \in E$ connects two neighboring cells i, k with corresponding histograms $\mathbf{h}_i$ and $\mathbf{h}_k$, and a matrix $\mathbf{A}$ of weights. We have considered the set of 8-connected cells of the image.

Let i and k be two neighboring image cells. Let β be the orientation of the vector extending from the center position of i to the center position of k. Then, the cost of the edge $e_{i,k}$ is assigned by taking into account the relationship between the value of β and the values of $\mathbf{h}_i$ and $\mathbf{h}_k$. When considering 8-connected cells, the values of β (in degrees) can be $(0, 45, 90, 135, 180, 225, 270, 315)$. Taking into account that we have considered 8 orientation bins for each orientation histogram, corresponding to $0, 22.5, 45, 67.5, 90, 112.5, 135$ and 157.5 degrees, the cost of $e_{i,k}$ can be defined as:

$$\begin{aligned} e_{i,k} &= \frac{1}{4}(\mathbf{h}_i(\beta) + \mathbf{h}_k(\beta)) \\ &+\frac{1}{8}(\mathbf{h}_i(\beta - 22.5) + \mathbf{h}_k(\beta + 22.5)) \\ &+\frac{1}{8}(\mathbf{h}_i(\beta + 22.5) + \mathbf{h}_k(\beta - 22.5)) \end{aligned} \tag{4.6}$$

All angle operations are defined (mod 180).

This is a symmetric measure ($e_{i,k} = e_{k,i}$) that assigns high values to neighboring cells which present curvature fields (represented by their histograms) that are similar to β and low values (or even zero) to neighboring cells which present curvature fields with orientations different from β.

We show an example of the graph corresponding to the 3×3 cells of the central part of a wrinkle image in Figure 4.27.

Centrality descriptor. Given an image I and its corresponding weighted undirected graph $\mathcal{G}$, the main assumption of our method is that the image cell that corresponds to the closed lumen is an important node of $\mathcal{G}$ when considering the shortest paths between all pairs of nodes in $\mathcal{G}$. Recall that, from our graph definition, the shortest paths will lie on regions of the image with a high curvature and will be aligned with the direction of the least curvature. Then, due to the wrinkle structure, *the node corresponding to the closed lumen position will have a high probability to occur on the shortest path between two randomly chosen nodes of the image*.

This assumption advises for an image descriptor based on the concept of shortest paths in a graph, and more specifically for a measure of the importance of a node based on the number of shortest paths it belongs to. In the literature, we can find several proposals regarding the measurement of the importance of a node based on this idea, all representing different approaches to the concept of *node centrality*.

Closeness centrality (56) is the inverse of the average shortest-path distance from the vertex to any other vertex in the graph. It can be viewed as the efficiency of each individual vertex in spreading information to all other vertices. Graph centrality was introduced implicitly in (57) to identify the *center* of a network by using only the maximum value of the shortest-path distances. Stress centrality was introduced in (58) to measure how much *work* is done by each vertex in a communication network. It assumes that the set of paths used for communication as the set of shortest paths. Finally, betweenness centrality (55) is the most important one and it constitutes a fundamental measurement concept that was originally proposed for the analysis of social networks. In its first definition, betweenness centrality for undirected graphs was derived from the column totals of a single matrix of numbers of pairwise dependencies of each point on every other point in terms of mediating access in reaching third points. This measure was proposed to identify important nodes in the network, going beyond the simplest degree centrality measure, which was defined as the number of links incident upon a node.

More formally, the four centrality measures can be defined as follows:

1. Closeness centrality: $C_1(v) = \frac{1}{\sum_{t \in V} d_{\mathcal{G}}(v,t)}$.
2. Graph centrality: $C_2(v) = \frac{1}{\max_{t \in V} d_{\mathcal{G}}(v,t)}$.
3. Stress centrality: $C_3(v) = \sum_{s \neq v \neq t \in V} \sigma_{st}(v)$.
4. Betweenness centrality: $C_4(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$.

The parameter σ_{st} is the number of shortest paths from node v_s to node v_t, $d_{\mathcal{G}}(s,t)$ is the distance between nodes v_s and v_t (i.e. the length of the shortest path from node v_s to node v_t) and $\sigma_{st}(v)$ denotes the number of shortest paths from s to t that some $v \in V$ lies on.

The resulting n^2-dimensional vector, C, stores the centrality measure of all graph vertices. This vector can be seen as a mid-level descriptor that represents the importance of a region of the image with regard to the shortest paths that run by following the field defined by $\mathbf{e}_2$. From this point of view, it contains global information that cannot be captured by any means using local operators. In our application, this vector represents fairly well the wrinkle structures and its maximum value component is located in the position of the closed lumen. Figure 4.28 shows different centrality measures. It can be seen that the betweenness centrality is the measure that best aligns with the wrinkle structures and for this reason, it is our choice for representing mid-level visual information.

A naive implementation of betweenness centrality would result in an algorithm complexity of $\Theta(|V|^3)$, where $|V|$ is the number of nodes of $\mathcal{G}$, which would make the computation of this measurement for large graphs prohibitive. An algorithm for the calculation of the betweenness centrality that runs in $\mathcal{O}(|V|\, m)$, where m is the number of edges, was proposed in (59). This algorithm allows a very fast computation of the betweenness centrality measure of all image cells.

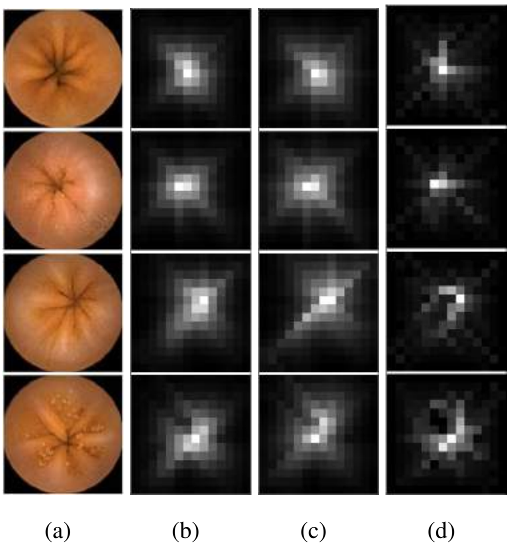


Figure 4.28: Centrality measures for different images: (a) Original image, (b) Closeness centrality, (c) Graph centrality and (d) Betweenness centrality. Stress centrality is not shown because in most of the cases, given our specific graph structure, it is equivalent to (d).

Automatic detection of wrinkle frames. The last step of the method deals with the detection of a wrinkle pattern in a WCE video. To this end, we propose to learn a linear classifier from a set of positive and negative examples and, then, to apply this classifier to image frames by using a sliding window that scans the image cells looking for the presence of a wrinkle pattern. In the following paragraphs, we first introduce the Structured Output Support Vector Machines, then, we formulate the wrinkle model learning problem and, finally, we define an algorithm for wrinkle frame detection.

Structured Output Support Vector Machines. Support Vector Machines (SVM) (60) are widely used to solve linear classifier problems in binary data. However, in their classical formulation, they are not easily applicable to multiclass problems. Structured Output Support Vector Machines (SO-SVM)(61, 62) is a recently proposed extension to SVM that is able to deal even with problems with infinite number of classes.

Let $d(\mathbf{x},\mathbf{y}), \mathbf{x} \in \mathbb{R}^{D_1}, \mathbf{y} \in \mathbb{R}^{D_2}$ be an unknown true data distribution, where in our case, $\mathbf{x}$ represents an image and $\mathbf{y}$ its corresponding label. Let $\{(\mathbf{x}_1,\mathbf{y}_1),\ldots,(\mathbf{x}_N,\mathbf{y}_N)\}$ be a set of N i.i.d. samples from $d(\mathbf{x},\mathbf{y})$. Let $\Delta : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ be a loss function that represents the price we are willing to pay by predicting an estimated value instead of the true value for an instance of data. Let $\phi(\mathbf{x},\mathbf{y}) : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}^D$ be a problem-dependent feature function that measures the correspondence between a data sample and a label. Finally, let $f(\mathbf{x}) = \arg\max_{\mathbf{y}\in\mathcal{Y}} \langle \mathbf{w}, \phi(\mathbf{x},\mathbf{y})\rangle$ be a decision rule that assigns a label to a data sample. Then, the objective of a linear learning method can be defined as:

- Find the weight vector $\mathbf{w}^*$ that minimizes the expected loss: $\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim d(\mathbf{x},\mathbf{y})}\{\Delta(\mathbf{y}, f(\mathbf{x}))\}$.

In SO-SVM, the optimization problem of finding the weight vector $\mathbf{w}^*$ is defined as:

$$\min_{\mathbf{w},\{\xi_i\}} \frac{1}{2}||\mathbf{w}||^2 + \frac{C}{N}\sum_{i=1}^{N}\xi_i$$

s.t. for $i = 1,\ldots,N$:

$$\Delta(\mathbf{y}_i,\mathbf{y}) + \langle \mathbf{w}, \phi(\mathbf{x}_i,\mathbf{y})\rangle - \langle \mathbf{w}, \phi(\mathbf{x}_i,\mathbf{y}_i)\rangle \leq \xi_i,$$

for all $\mathbf{y} \in \mathcal{Y}\backslash\{\mathbf{y}^n\}$.

This is an optimization problem with $N|\mathcal{Y}|$ linear constraints and a convex, differentiable objective function that can be solved with off-the-shelf optimization software by following the iterative Algorithm 2, proposed in (62).

When using Algorithm 2 for solving a specific problem, one only needs to define the following functions:

Algorithm 2 for training a SO-SVM

```
Input: $\{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_N, \mathbf{y}_N)\}, C, \epsilon$
$\mathcal{W} \leftarrow \emptyset$, $\mathbf{w} \leftarrow 0$, $\xi_i \leftarrow 0$ for all $i = 1, \ldots, N$..
repeat
  for $i = 1$ to $N$ do
    $\mathbf{y}^* = \arg\max_{\mathbf{y} \in \mathcal{Y}} \{\Delta(\mathbf{y}_i, \mathbf{y}) + \mathbf{w}[\phi(\mathbf{x}_i, \mathbf{y}) - \phi(\mathbf{x}_i, \mathbf{y}_i)]\}$
    if $\mathbf{w}[\phi(\mathbf{x}_i, \mathbf{y}_i) - \phi(\mathbf{x}_i, \mathbf{y}^*)] < \Delta(\mathbf{y}_i, \mathbf{y}^*) - \xi_i - \epsilon$ then
      $\mathcal{W} \leftarrow \mathcal{W} \cup \{\mathbf{w}[\phi(\mathbf{x}_\mathbf{i}, \mathbf{y}_\mathbf{i}) - \phi(\mathbf{x}_\mathbf{i}, \mathbf{y}^*)] \geq \boldsymbol{\Delta}(\mathbf{y}_\mathbf{i}, \mathbf{y}^*) - \xi_\mathbf{i}\}$
      $(\mathbf{w}, \xi_\mathbf{i}) \leftarrow \min_{\mathbf{w}, \xi_\mathbf{i}} \frac{\mathbf{1}}{\mathbf{2}}||\mathbf{w}||^\mathbf{2} + \frac{\mathbf{C}}{\mathbf{N}} \sum_{\mathbf{i=1}}^{\mathbf{N}} \xi_\mathbf{i}$ s.t. $\mathcal{W}$
    end if
  end for
until $\mathcal{W}$ does not change during the iteration
Output: $\mathbf{w}$
```

1. The feature function $\phi(\mathbf{x}, \mathbf{y})$.

2. The loss function $\Delta(\mathbf{y}, \mathbf{y}')$.

3. The constraint generation function:

$$\arg\max_{\mathbf{y} \in \mathcal{Y}} \{\Delta(\mathbf{y}_i, \mathbf{y}) + \mathbf{w}[\phi(\mathbf{x}_\mathbf{i}, \mathbf{y}) - \phi(\mathbf{x}_\mathbf{i}, \mathbf{y}_\mathbf{i})]\}.$$

Learning problem formulation. We formulate our problem as to learn a localization function that predicts the bounding box of a wrinkle structure, centered on the lumen position, that is, $f(\mathbf{x}) : \{\text{all images} \rightarrow \text{all image squared bounding boxes}\}$. We are given a set of training pairs $\{(\mathbf{x}_1, \mathbf{y}_1) \ldots (\mathbf{x}_N, \mathbf{y}_N)\}$, where $\mathbf{x}_i$ are images and $\mathbf{y}_i$ consist of a binary label o indicating whether an object is present, and a four dimensional vector $(x_{tl}, y_{tl}, x_{br}, y_{br})$ indicating the top-left and bottom-right coordinates of the bounding box within the image: $\mathbf{y}_i \in \{(o, x_{tl}, y_{tl}, x_{br}, y_{br}) | o \in \{+1, -1\}, (x_{tl}, y_{tl}, x_{br}, y_{br}) \in \Re^4\}$. The objective is to learn a mapping f that generalizes from given examples.

To define the feature function $\phi(\mathbf{x}, \mathbf{y})$, we note that we have two different kinds of labels which combined define a class: a binary label indicating whether or not the bounding box contains a wrinkle structure, and four numerical labels representing the bounding box coordinates. Bounding box coordinates are clearly irrelevant for learning a good mapping f, because wrinkle patterns can be found at any image position but the binary label defines a partition of the input space that should be represented in the feature function. Following the model proposed in (63), we define $\phi(\mathbf{x}, \mathbf{y})$ as a $2M$-dimensional vector $\phi(\mathbf{x}, \mathbf{y}) = (x_1, \ldots, x_M, 0, \ldots, 0)$ when

$\mathbf{x}$ is a positive example (o = +1) and $\phi(\mathbf{x}, \mathbf{y}) = (0, \ldots, 0, x_1, \ldots, x_M)$ when $\mathbf{x}$ is a negative example (o = -1). Consequently, $\mathbf{w}$ must be also a $2M$-dimensional vector.

This feature representation induces the simultaneous learning of two weight vectors: a weight vector $\mathbf{w}_+ = (w_1, \ldots, w_M)$ for positive examples and a weight vector $\mathbf{w}_- = (w_{M+1}, \ldots, w_{2M})$ for negative examples. This scheme, in spite of increasing the dimensionality of $\mathbf{w}$, has been shown useful for training Structured Output Support Vector Machines (63).

In SO-SVM, the loss function $\Delta(\mathbf{y}, \mathbf{y}')$ plays similar role as the margin in classical SVM. It measures how far a prediction $\mathbf{y}'$ is from a true label $\mathbf{y}$. Let $\mathbf{y}|_{bb}$ be the set of image pixels included in the bounding box represented in $\mathbf{y}$. Then, given a prediction $\mathbf{y}'$ and a true label $\mathbf{y}$, we defined the following loss function:

$$\begin{cases} \Delta(\mathbf{y}, \mathbf{y}') = 1 - A, \text{ iff both examples are positive and } \mathbf{y}|_{bb} \cap \mathbf{y}'|_{bb} \neq \emptyset, \\ \Delta(\mathbf{y}, \mathbf{y}') = 0, \text{ iff } \mathbf{y} = \mathbf{y}', \\ \Delta(\mathbf{y}, \mathbf{y}') = 1, \text{ otherwise,} \end{cases}$$

where A is the Jaccard coefficient (45):

$$A = \frac{\text{Area of } (\mathbf{y}|_{bb} \cap \mathbf{y}'|_{bb})}{\text{Area of } (\mathbf{y}|_{bb} \cup \mathbf{y}'|_{bb})},$$

The constraint generation function identifies which is the most incorrect output $\mathbf{y}'$ that the current model still considers to be compatible with a sample $\mathbf{x}_i$. In our case, this process consists of finding in the image a bounding box $\mathbf{y}'$ such that $\langle \mathbf{w}, \phi(\mathbf{x}_i, \mathbf{y}') \rangle > \langle \mathbf{w}, \phi(\mathbf{x}_i, \mathbf{y}_i) \rangle$. The cost of this function is linear with respect to the number of different bounding boxes considered for each image.

Wrinkle frame detection. Finally, given a video frame $\mathbf{x_i}$ and $\mathbf{w}^*$ (computed by the SO-SVM), the detection process follows the steps presented in Algorithm 3. The dimension of the wrinkle model is directly related to the size (or number of cells) of the samples we used to train the model. In our case, positive and negative samples are half of the size of a WCE frame in order to have a more localized response in the image.

Lines (1-7) of the algorithm show the computation of the values of the betweenness matrix C. The algorithm uses a sliding window approach in order to evaluate the response of the model $\mathbf{w}^*$ at every location of the frame $\mathbf{x_i}$ (lines (8-14)). If at least one evaluation reports a positive detection, the image is considered a wrinkle frame.

Note that up to now we have been considering features (low- and mid-level) that were derived from the frame intensity. We can get an improvement on the performance of this algo-

Algorithm 3 Wrinkle frame detection

```
Input: A video frame $\mathbf{x}_i$ and a wrinkle frame model $\mathbf{w}^*$.
Compute for each pixel the Hessian matrix $Hess$.
Compute for each pixel $\lambda_1, \lambda_2, \mathbf{e}_1$ and $\mathbf{e}_2$.
Compute an adaptive threshold value $t$ and apply it to the image by selecting those pixels
    where $\lambda_1 > t$.
Divide the image in $n \times n$ non-overlapping cells.
Compute the histogram of features $H$ corresponding to the $\mathbf{e}_2$ direction for each cell.
Build the graph $\mathcal{G}$ from $H$.
Compute the $n \times n$ vector $C$ corresponding to the betweenness centrality of each image
    cell.
Define a $m \times m$ window $W$ such that $m \leq n$.
for every possible position of $W$ on the image do
    Build a vector $\phi(\mathbf{x}_i, \mathbf{y}_i)$ by considering the centrality values of all the image cells inside
        $W$.
    Compute $\mathbf{y}^*_{jk} = \arg\max_{\mathbf{y} \in \mathcal{Y}} \langle \mathbf{w}^*, \phi(\mathbf{x}_i, \mathbf{y}_i) \rangle$
end for
if there is a $\mathbf{y}^*_{jk} \neq (0,0,0,0)$ then
Output:    $\mathbf{y}_i = \mathbf{y}^*_{jk}$.
else
Output:    $\mathbf{y}_i = (0,0,0,0)$.
end if
```

rithm by adding color features, since color is an important visual cue that can improve wrinkle detection. Color information can be easily added by concatenating a few values representing the color inside the window hypothesis to the betweenness matrix C.

4.3.2.2 Validation

In this section, we perform an evaluation of the proposed method. We compare the mid-level features based on betweenness centrality to several low-level images features such as the Histogram of Gradients (HoG) and the Histogram of Features (HoF) based on the Hessian using a standard linear SVM (64). We show that the best results are obtained for the betweenness centrality descriptor. Moreover, we show that the color information is an important cue for wrinkle frame detection. Finally, we compare a standard linear SVM to the linear SO-SVM. Parameters in both classifiers were tuned using a cross-validation.

Database. In order to validate the proposed system, a training and a test set were created using different videos obtained with a PillCam SB2 capsule provided by Given Imaging Ltd. Both, the training and the test set were collected and labeled by experts at the original 256×256 image resolution. The training set consists of 1000 wrinkle frames and 1000 non wrinkles frames from 4 videos. The lumen center was manually labeled in all training wrinkle images. For each positive sample in the training set, 4 partial and 1 full wrinkle windows were considered, where partial windows mean a window with partial overlapping with the ground truth. Both full and partial windows consist of 128x128 image pixels with the corresponding label defined as the bounding box coordinates. Negative samples consist of 128x128 pixel image windows located at random locations of negative samples. The test set consists of 1500 wrinkle frames and 2500 non wrinkle frames from 5 videos (not considered in the training set). All negative frames, from both training and test set, were obtained by a random subset of non-wrinkle frames.

Measurements. In order to compare different wrinkle descriptors and different classifiers the following measures are used:

- AUC : Area under Precision/Recall curve
- Accuracy (A) = $\frac{TP+TN}{TP+FP+TN+FN}$
- Precision (P) = $\frac{TP}{TP+FP}$

- Recall (R) = $\frac{TP}{TP+FN}$

where TP = true positive, TN = true negative, FP = false positive and FN = false negative. The frames with wrinkles are considered the positive samples.

Validation. Table 4.5 presents the obtained results when using four different image descriptors and the sliding window approach:

- the standard HoG descriptor computed from image gradients,
- a histogram descriptor built from the $\mathbf{e_2}$ values of the Hessian matrix HoF,
- the proposed centrality descriptor C,
- the concatenation of the betweenness centrality and the color information C_c. Color information has been defined as a 3-dimensional vector representing the mean RGB color of each cell.

Since wrinkles are tubular structures, intuitively they should be better represented by the Hessian matrix than by the distribution of gradient vectors on the image. This hypothesis is confirmed by analyzing the results presented in Table 4.5. By comparing AUC value of HoG and HoF, an improvement of more than 20% can be observed. By further analysis of the obtained results, it can be seen that the proposed mid-level centrality descriptor outperforms the low-level information coded in the histogram by increasing AUC from 87.78% to 91.91% and the accuracy from 83.83% to 87.76%. This result confirms that the relation between different image cells that is coded in the mid-level centrality vector C is useful for wrinkle detection. Finally, results confirm that color information is an important cue. The inclusion of color information provides further improvement in AUC from 91.91% to 94.71%.

The precision/recall curves presented in Figure 4.29 show that the centrality descriptor obtains a better compromise between precision and recall. The inclusion of the color information increases the performance of the detector by allowing the discrimination of frames with food content that sometimes resemble wrinkle structures.

Finally, we consider to use a linear SO-SVM instead of the linear SVM (see Table 4.5 and Figure 4.29). The main difference between these two approaches is the consideration of *partial windows* during the learning process. These hypotheses, that correspond to bounding boxes that intersect the true bounding box in a wrinkle frame, have a clear regularization effect on the learned decision function that allows a better generalization to unseen samples.

A qualitative evaluation is presented in Figure 4.30, 4.31(a) and 4.31(b) showing, respectively, True Positive (TP) detections, False Negative (FN) detections and False Positive (FP)

Table 4.5: Classification performance using different descriptors 1) Standard HoG, 2) Histogram of Features HoF, 3) Betweenness Centrality C and 4) Betweenness Centrality plus color information C_c. In the validation, two types of linear classifiers were used: Support Vector Machines (SVM) and Structured Output Support Vector Machines (SO-SVM).

Descriptor Classifier	HoG SVM	HoF SVM	C SVM	C_c SVM	C_c $SO-SVM$
AUC	64.19	87.78	91.91	94.76	**96.67**
Accuracy	67.40	83.82	87.76	88.43	**89.54**
Precision	53.18	70.30	78.95	87.12	**90.27**
Recall	54.14	70.75	83.52	91.18	**93.32**

detections. All figures present both, the original frame and a visualization of its corresponding centrality descriptor. As it can be seen in Figure 4.30, the centrality descriptor for most TP samples shows the star-like shape and the cell related to the closed lumen is the one with highest centrality value. On the other hand, we can see in Figure 4.31(a) that most of FN present very smooth folds of intestinal wall and a completely closed lumen. These issues make it difficult to properly characterize the frame with the centrality descriptor, since in most of these cases the star-like shape is not observed. Finally, we can see in Figure 4.31(b) that some of the FP are difficult to be labeled, for instance, images from the third row (second and third image from left) present several folds. Moreover, the majority of FP contains some intestinal content hinders the lumen visibility, and so, the proper image classification.

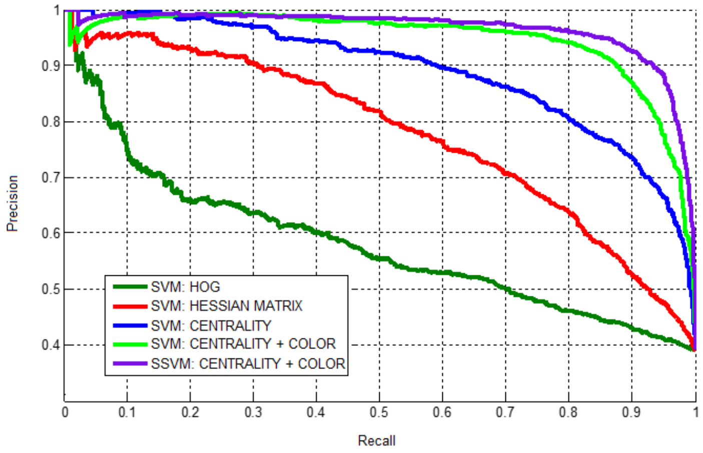


Figure 4.29: Precision/Recall curves for wrinkle frames detection.

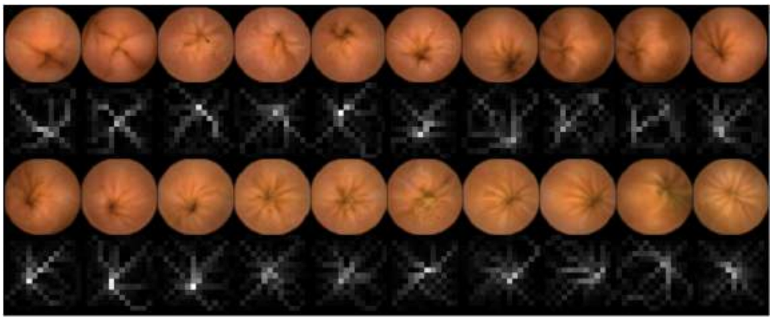
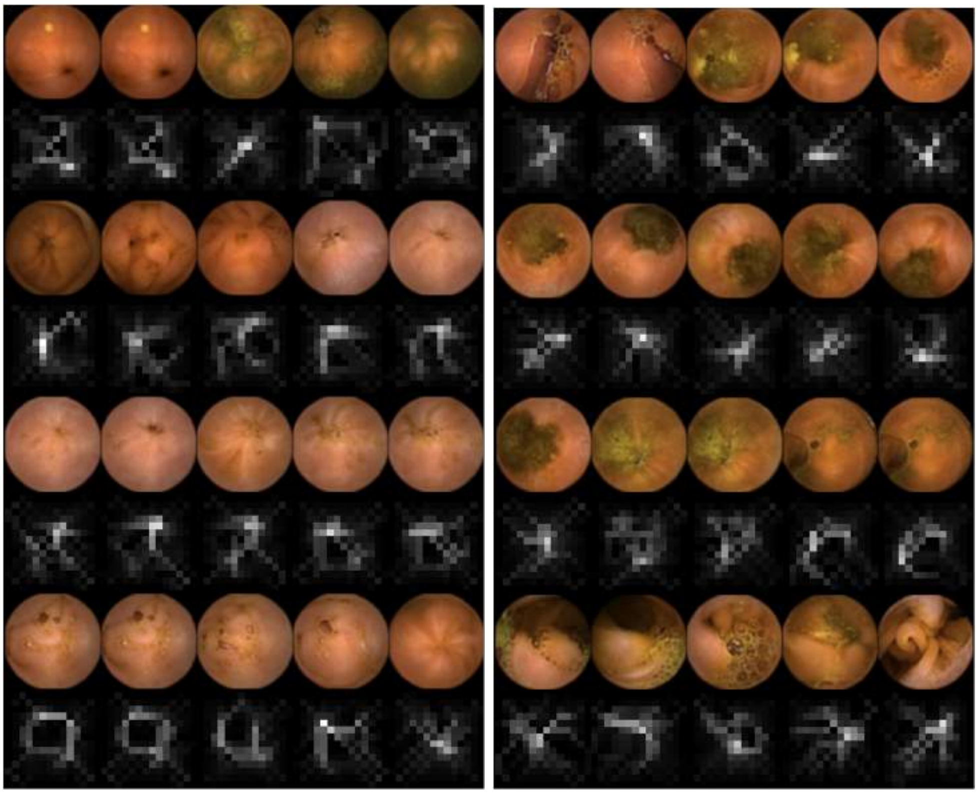


Figure 4.30: True Positive detections of wrinkle frames.



(a)

(b)

Figure 4.31: Visual evaluation of the results: (a) False Negative detections, (b) False Positive detections.

For a physician, it is important to see how wrinkle frames are distributed along a small intestine. To this end, we display in Figure 4.32 the regions where our system detects a high percentage of wrinkle frames by using the motility described in chapter 3. Below the mosaic image, some random frames classified as wrinkles are visualized. The corresponding video

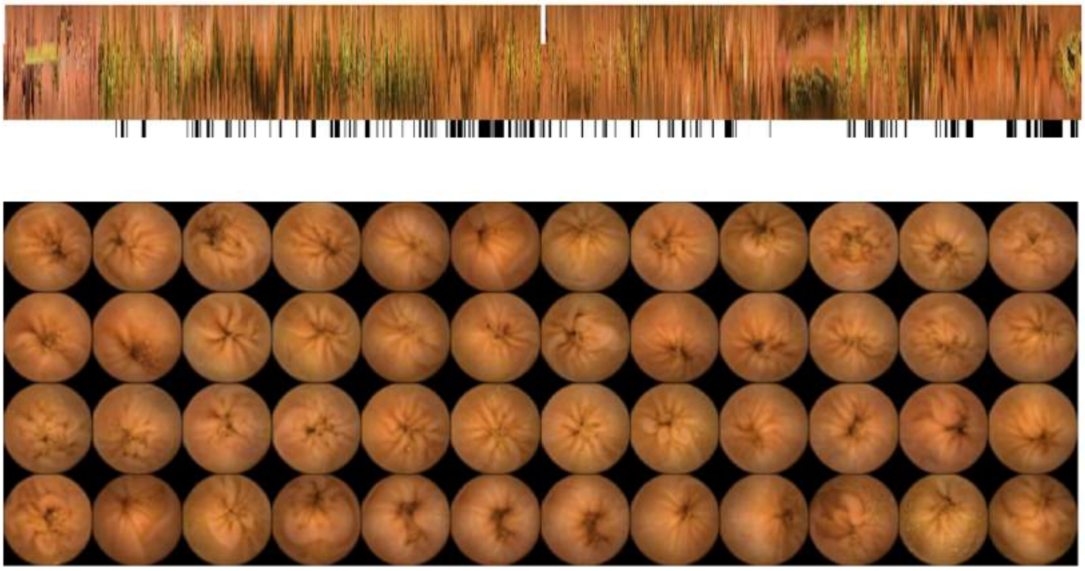


Figure 4.32: The image on the top shows the results of the proposed wrinkle detector together with the motility bar. Black vertical bars correspond to the video segments with high density of frames detected as wrinkles. The bottom image shows some random frames detected as wrinkles by the system.

segment is 20 minutes long, which means 2.400 frames. We implemented the method in Matlab and ran it on Intel I5-2520 CPU machine. The time needed to obtain wrinkle score for this video segment was approx. 30 minutes.

4.4 Discussion

In the current Chapter, we have presented methods for automatic analysis of the motility information that can be perceived in WCE video. We have proposed two motility bar based methods: 1) contraction density estimation and 2) intestinal perimeter estimation. Moreover, two methods for frame per frame analysis have been proposed: 1) intestinal content detection and 2) wrinkle frame detection.

4.4.1 Motility-bar-based features

Contractile density estimation We have proposed and visually evaluated an automatic method for the detection of contractions using the motility bar image. The proposed method is three-fold. In the first step, vertical valleys are detected. The detection is based on Gabor-like filters that are convolved with the image at different scales. The convolution results are combined into one image representing the detected valleys. In the second step, the valley image is transformed into one-dimensional signal using a restriction on oscillation size (percentile 75). In the third step, local maxima of the one dimensional signal are detected. The local maxima correspond to contractions. Finally, the density of detected contractions is calculated. Visual analysis of the results suggests that the contractile oscillations are well detected and the estimation of density reflects the visual information that is present in the motility bar.

Lumen size estimation We have proposed and visually evaluated an automatic method to measure the lumen size in the motility bar. The method is based on the assumption that the lumen is visible as a dark region in the image and applies a threshold in order to obtain its segmentation. In order to reduce small regions detected as lumen, morphological operations are applied. Visual analysis of the results shows that the method correctly estimates the lumen size in the motility bar.

4.4.2 Frame-based features

Intestinal Content We have proposed and evaluated an automatic system for categorization and segmentation of intestinal content frames for WCE. The three main contributions of this method are: 1) development and validation of an automatic system for intestinal content detector; 2) development and validation of a segmentation method for detection of bubbles and turbid media in WCE images; and 3) definition of a new image feature of WCE: area covered by each kind of intestinal content. The presented method is divided into two steps. In the first step, the frames with intestinal content are detected using a color and textural feature and a Linear SVM classifier. In the second step of the system, the intestinal content frames are segmented and the image regions of bubbles and turbid media are obtained. The evaluation of the proposed system, using a large data set, shows that the presented method outperforms the results of the state-of-the-art. Moreover, we observe that, regarding the intestinal content variability in terms of color and texture, a large data set is needed to ensure the generalization of the method, and in this sense, our experiments confirm the statistical robustness of the presented outcomes. Finally, qualitative and quantitative results of segmentation method present good performance when discriminating intestinal content in bubbles and turbid.

Wrinkles We have presented a new image descriptor for the classification of WCE wrinkle frames. The proposed image descriptor is based on image centrality descriptor, which is based on the histogram of oriented features extracted from the Hessian matrix of an image. This mid-level descriptor integrates global image information that is useful to detect star-like shape patterns. The detection process is based on a model learned by using a Structural Output Support Vector Machine approach. This approach not only uses positive and negative samples but also samples that correspond to partial hypotheses. This inclusion produces better detection models. The detection process is performed by a sliding window procedure that scans the image looking for a positive label. This allows to train more accurate models that can be applied in a multiscale architecture in order to get better localization. A second advantage of this approach is that there is no need to detect the lumen center previously to the classification. The low complexity of all involved algorithms allows for near real time processing of WCE frames. The validation, carried out on a large database, shows that the proposed descriptor successfully detects this particular event in WCE videos, outperforming previous methods and defining a new state of the art for this problem. Regarding future work, one can note that wrinkle detection can be extended to multiples scales by applying the same algorithm to a pyramid of the image representing different image resolutions. This would allow the detection of closed lumens with short wrinkles around them.

5

Sequential feature analysis

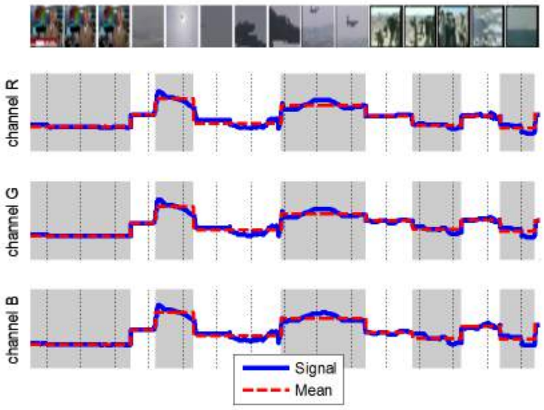


Figure 5.1: An example of mean change detection in video data. The first line, represents some samples from the video stream. For every frame, the R, G and B values are calculated (blue line). Using our approach the video segments with constant means are detected and marked with red line.

5.1 Introduction

Up to now, we have introduced a representation of intestinal motility (motility bar) and we have provided automatic methods for describing the basic information presented in the motility bar (contractions density and lumen perimeter estimator) as well as detectors based on WCE frame analysis (turbid, bubble and wrinkles detector). In the current Chapter, we show how to move from the analysis based on individual frames into sequential analysis. To this end, we introduce a robust method for the analysis of multivariate data streams. The method detects segments of constant norm of mean values obtained from the k-dimensional data stream (the method can be seen as change point detector). For an exemplary result of constant mean segment detection in three dimensional color signal see Figure 5.1.

In this Chapter, first, we present the concepts of streaming data analysis, second, introduce the algorithm for robust analysis of the norm of the means of a multivariate signal and, finally, show the results of applying the method to WCE data. In particular, we show how the method works for the detection of constant color in the motility bar, a joint analysis of lumen size and contraction density, and the detection of turbid segments. The huge amount of data represented by WCE and theirs sequential nature justify our assumption done in this chapter: WCE data are streaming data.

5.2 Robust method for data stream analysis

Being able to accurately handle streaming data whose nature is changing over time is one of the core problems in data mining, pattern recognition and machine learning. This is a challenging problem, since the streaming data are constantly arriving and, generally, the labels are not available at the moment a new sample arrives. Moreover, streaming data analysis algorithms must deal with limited memory availability (much less than the possibly infinite input size) and limited processing time per item (in order to ensure the memory requirement).

The following three problems should be considered when handling streaming data (65): 1) detecting when a significant change occurs in the distribution of the data stream; 2) deciding which examples must be kept in memory and which ones to forget (this step permits the optimization of the memory used for data stream analysis); and 3) revising the actual model whenever a significant change has been detected. Having a set of robust methods to deal with these problems is the first requirement to work on streaming data analysis.

Since the streaming data algorithms have no direct access to underlying data distribution but to samples drawn from this distribution, they should conclude on the distribution changes (or drifts) analysing only the seen samples. A difficult problem in handling distribution changes is distinguishing between true distribution changes and noise. Algorithms can misleadingly treat noise as distribution changes. In the literature, this problem is overcome by the use of robust statistical hypothesis tests (66), confidence intervals (67) or concentration inequalities (65, 68). Concentration inequalities (69) provide probability bounds on how a random variable deviates from its expectation. Hoeffding's inequality (70) and Bernstein's inequality (71) are examples of concentration inequalities.

Recently in (65), a non-parametrical method called *AdWin* was proposed. In (72), *AdWin* is referred to as reference algorithm for sequence-based adaptive sliding window. The method is based on analyzing the content of a sliding window, whose size is not fixed a priori, but is recomputed online according to the rate of change of the contents of the window itself. As a result of the algorithm, the window increases its length when the data are stationary and shrinks automatically when the change takes place. The criteria to grow/shrink the sliding window are based on concentration inequalities (Hoeffding's inequality). As a result, (65) proposes rigorous guarantees on the performance of the algorithm. Moreover, the algorithm uses a variant of the exponential histogram technique (73) to compress the window. Thus, in order to keep a window of size W, it uses $O(\log W)$ memory and $O(\log W)$ processing time per item.

Unfortunately, the bounds derived in (65) are only valid for one dimensional data streams, which makes the method difficult to apply to computer vision problems where descriptors have a dimensionality higher than one, such as color histogram change detection in video sequences or structural changes in scenes represented by histograms of oriented gradients.

In this chapter, we present a new approach to apply an *AdWin*-like algorithm to multivariate data. To this end, we propose a generalization of the bounds proposed in (65) to derive a new algorithm to detect the change of the mean in multidimensional data streams. The proposed method monitors the k-dimensional means inside an adaptive window and determines whenever there is a partition in two sets inside the current window, whose k-dimensional means are significantly different. In order to detect a significant change between the two k-dimensional means, the algorithm uses the norm of the mean. The algorithm provides rigorous guarantees on the confidence of the detected change and can be applied to any data distribution. Thus, unlike (65), our bounds can be used not only to 1-D data streams, but to any k-dimensional data stream. Moreover, the algorithm can be applied with any norm L_p for $p = 1, 2, ..., n$.

5.3 Important inequalities and bounds

5.3.1 Hoeffding's inequality and bound

In order to be able to robustly detect changes in a data stream in an unsupervised way, it is necessary to have a bound on the difference between the estimated change measure and its real value given a confidence parameter. In the case of one dimensional data, the Hoeffding's bound, which can be derived from the classical Hoeffding's inequality (70), can be readily used. This inequality measures the distance between the empirical mean and the expected value of several observations of a random variable.

Theorem 5.3.1 (Hoeffding's inequality, 1963)**.** *Consider a real-valued random variable x whose range is R. Suppose we have made n independent observations of this variable $x_1, ..., x_n$ and computed their empirical mean $\widehat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i$. Let μ be the expected value of x. Then, for any $\epsilon > 0$,*

$$Pr\left(|\widehat{\mu} - \mu| > \epsilon\right) \leq 2\exp\left(\frac{-2n\epsilon^2}{R^2}\right). \tag{5.1}$$

From the inequality the following bound can be derived:

Definition 1 (Hoeffding's bound)**.** *If we draw n samples from x, then with probability at least $1 - \delta$, the expected value of x is within ϵ of $\widehat{\mu}$, where:*

$$\epsilon = R\sqrt{\frac{\ln(2/\delta)}{2n}}. \tag{5.2}$$

Note that Inequality (5.1) is true regardless of the desired confidence value $1 - \delta$, but in order to use it as a bound, a confidence value δ must be defined.

5.3.2 Concentration inequalities for multivariate data streams

Let us introduce the inequality for the norm of k-dimensional means:

Theorem 5.3.2 (Inequality theorem for the norm of k-dimensional means)**.** *Consider a real-valued k-dimensional vector of random variables $\overrightarrow{x} = [x_1, x_2, ..., x_k]^T$. Without loss of generality, each element x_i of the vector $\overrightarrow{x}$ is defined within a range $R = [0, 1]$. Suppose that we have made n independent observations of this variable $X = (\overrightarrow{x}_1, ..., \overrightarrow{x}_n)$. Let $Y = ||\frac{1}{n}\sum_{i=1}^{n} X_{i,j} - E(X_j)||_p$ where $||\ ||_p$ is the p-norm of the vector. Then, for any $\epsilon > 0$,*

$$Pr\left(Y > \epsilon\right) \leq 2k \exp\left(\frac{-2n\epsilon^2}{k^{2/p}}\right). \tag{5.3}$$

Proof. Let us define $\overrightarrow{y} = |\frac{1}{n}\sum_{i=1}^{n} X_{i,j} - E(X_j)|$. Then,

$$Pr\left(||\overrightarrow{y}||_p > \epsilon\right) = Pr\left(\left(\sum_{j=1}^{k}(y_j)^p\right)^{\frac{1}{p}} > \epsilon\right) \tag{5.4}$$

Since $\epsilon > 0$ and $\sum_{j=1}^{k}(y_j)^p > 0$, we can rewrite (5.4) as:

$$\begin{aligned} Pr\left(\sum_{j=1}^{k}(y_j)^p > \epsilon^p\right) &= Pr\left((y_1)^p + (y_2)^p + ... + (y_k)^p > \epsilon^p\right) \leq \\ \leq Pr\left((y_1)^p > \frac{\epsilon^p}{k}\right) &+ Pr\left((y_2)^p > \frac{\epsilon^p}{k}\right) + ... + Pr\left((y_k)^p > \frac{\epsilon^p}{k}\right) \end{aligned} \tag{5.5}$$

To see the correctness of Inequality (5.5) note that $\{\sum_{j=1}^{k}(y_j) > a\} \leq \bigcup_{j=1}^{k}\{y_j > a/k\}$. Now using (5.1), we have:

$$Pr\left((y_j)^p > \tfrac{\epsilon^p}{k}\right) = Pr\left(y_j > \tfrac{\epsilon}{k^{1/p}}\right) = \tag{5.6}$$

$$Pr\left(|\tfrac{1}{n}\textstyle\sum_{i=1}^{n} X_{i,j} - E(X_j)| > \tfrac{\epsilon}{k^{1/p}}\right) \leq 2\exp\left(\tfrac{-2n\epsilon^2}{k^{2/p}}\right) \tag{5.7}$$

Finally, by joining the results of (5.5) and (5.6), we obtain:

$$Pr\left(||\frac{1}{n}\sum_{i=1}^{n} X_{i,j} - E(X_j)||_p > \epsilon\right) \leq 2k \exp\left(\frac{-2n\epsilon^2}{k^{2/p}}\right) \tag{5.8}$$

□

For the sake of completeness, we introduce the inequality for the mean of the norms that can be easily derived from Hoeffding's inequality.

Theorem 5.3.3 (Inequality theorem for the mean of the norms)**.** *Consider a real-valued k-dimensional vector of random variables $\overrightarrow{x} = [x_1, x_2, ..., x_k]^T$. Without loss of generality, each element x_i of the vector $\overrightarrow{x}$ is defined within a range $R = [0, 1]$. Let $L = ||\overrightarrow{x}||_p$ be a random variable defined as p-norm of $\overrightarrow{x}$. Suppose that we have made n independent observations of this variable $L_1, ..., L_n$ and computed their empirical mean $\widehat{L} = \frac{1}{n}\sum_{i=1}^{n} L_i$. Let $E(L)$ be the expected value of L. Then, for any $\epsilon > 0$,*

$$Pr\left(|\widehat{L} - E(L)| > \epsilon\right) \leq 2\exp\left(\frac{-2n\epsilon^2}{k^{2/p}}\right). \tag{5.9}$$

Note that the definitions of Inequality (5.3) and Inequality (5.9) are different. To highlight the impact of this difference, let us see the following: Given 2 samples from a 2-dimensional data stream $\overrightarrow{x_1} = [1, 0]^T$ and $\overrightarrow{x_2} = [0, 1]^T$, let us assume that the expectation $E(\overrightarrow{x}) = [1, 0]^T$ and evaluate the left-hand side of both inequalities. Inequality (5.3) evaluates $Pr(||[\frac{1}{2}, \frac{1}{2}]^T - [1, 0]^T||_p > \epsilon)$, while Inequality (5.9) is equivalent to evaluate $Pr(|\frac{1}{2} - 1| > \epsilon)$. As it can be noted Inequality (5.3) is sensitive to the permutations of the components of vector $\overrightarrow{x}$, while Inequality (5.9) is insensitive to the components permutation.

From Inequalities (5.9) and (5.3), the following bounds can be derived:

Definition 2 (Bound for the norm of k-dimensional means)**.** *If we draw n samples from $\overrightarrow{x}$, then with probability at least $1 - \delta$, the vector of expected values of X_j is within ϵ of $\frac{1}{n}\sum_{i=1}^{n} X_{i,j}$ under the p-norm, where:*

$$\epsilon = k^{1/p}\left(\frac{1}{2n}\ln\frac{2k}{\delta}\right)^{\frac{1}{2}}. \tag{5.10}$$

Definition 3 (Bound for the mean of the norms)**.** *If we draw n samples from $\overrightarrow{x}$, then with probability at least $1 - \delta$, the expected value of $L = ||\overrightarrow{x}||_p$ is within ϵ of $E(L)$, where:*

$$\epsilon = k^{1/p}\left(\frac{1}{2n}\ln\frac{2}{\delta}\right)^{\frac{1}{2}}. \tag{5.11}$$

Remember that bound from Formula (5.11) (contrary to Formula (5.10)) is not sensitive to the permutations among vector components. Since in computer vision the permutation is important (e. g. for RGB color [1, 0, 0] is red color, while [0, 0, 1] is blue color), in our algorithm we use the bound from Formula (5.10) based on Inequality (5.3).

Let us illustrate the behavior of ϵ from Formula (5.10) for different values of data dimensionality k and norm p (see Figure 5.2, the figure is plotted in a logarithmic scale). Note that for $k = 1$, the plot presents the Hoeffding's bound and that for this k, the algorithm is independent of the chosen metric p. Given a confidence value δ, the higher the dimension k is, the more samples n the bound needs in order to reach the same value of ϵ. The higher norm is used (p value), the less important the dimensionality k becomes.

5.4 Adaptive windowing algorithm for multivariate data streams

Once the inequality for the norm of k-dimensional means has been introduced, we explain how it can be used to detect changes in the norms of the means of multivariate data.

Let us consider a multivariate data stream $\overrightarrow{x_1}, \overrightarrow{x_2}, ..., \overrightarrow{x_t}, ...$, where each observation $\overrightarrow{x_i} = [x_i^1, x_i^2, ..., x_i^k]^T$, $i = 1, 2, ..., t, ...$ is a k-dimensional vector. Each sample $\overrightarrow{x_i}$ is generated according to some multivariate distribution D_i. Each $x_i^j \in [0, 1]$. Let $\overrightarrow{\mu} = [\mu^1, \mu^2, ..., \mu^k]^T$ be the expected value of $\overrightarrow{x}$ and $\overrightarrow{\widehat{\mu}_W} = [\widehat{\mu}_W^1, \widehat{\mu}_W^2, ..., \widehat{\mu}_W^k]^T$ be the estimated mean within the window W. Nothing else is known about the data stream.

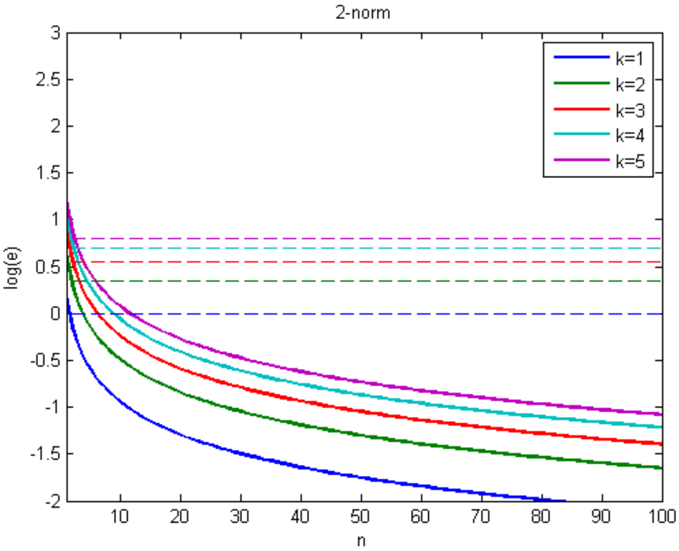
We begin with a window W. At each step t of the algorithm, we add next data from the data stream (increasing the size of window W by one). We form all bi-partitions $W_1.W_0$ of window W and calculate the means of the bi-partitions. As soon as the p-norm of differences between the calculated means of a bi-partition is greater or equal to ϵ_{cut}, we can reduce the size of the window W by dropping the oldest stream elements and reducing the size of W.

Note that one of the advantages of the algorithm is that it has only one parameter: the confidence value $\delta \in (0, 1)$. Algorithm 4 shows the adaptive windowing algorithm for a k-dimensional data stream.

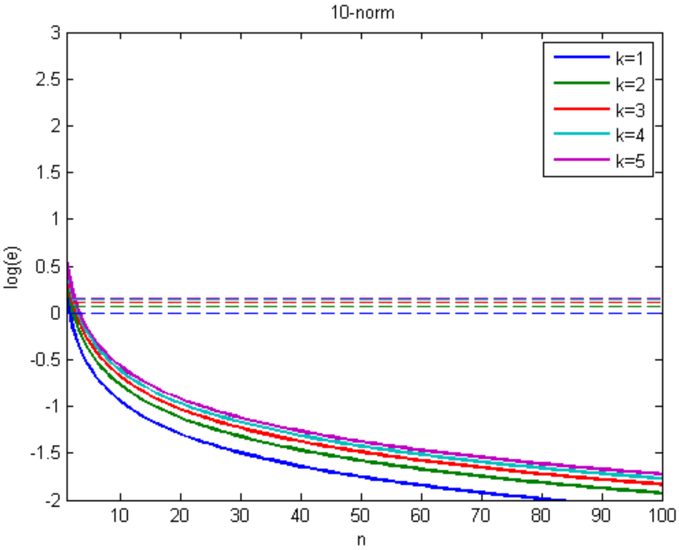
Algorithm 4 Adaptive windowing algorithm for k-dimensional data stream.

```
Input: multivariate data stream $\overrightarrow{x_1}, \overrightarrow{x_2}, ..., \overrightarrow{x_t}, ...$
Input: $\delta$ parameter
Initialize window $W$
for each $t > 0$ do
$W \leftarrow W \cup \{\overrightarrow{x_t}\}$ {add $\overrightarrow{x_t}$ to the head of $W$}
while exists a split of $W$ into $W = W_0 \cdot W_1$ such that $||\overrightarrow{\widehat{\mu}_{W_0}} - \overrightarrow{\widehat{\mu}_{W_1}}||_p \geq \epsilon_{cut}$ do
    Drop elements of $W_0$ from the tail of $W$
end while
Output: $\overrightarrow{\widehat{\mu}_W}$ and $\overrightarrow{\widehat{\mu}_{W_0}}$
```

A natural question arises: what guarantees can be given on the algorithm performance? In particular, it is interesting to know with what probability the algorithm can mistakenly reduce the window size whenever the $||\overrightarrow{\mu}||_p$ remains constant (False positive bounds) and with what probability the algorithm will split the current window W into two subwindows $W_0.W_1$ once



(a)



(b)

Figure 5.2: Analysis of the bound ϵ in Formula (5.10). Plots are shown in a logarithmic scale. Y-axis represents the number of samples n. a) $\delta = 0.1$ and $p = 2$, b) $\delta = 0.1$ and $p = 10$. Dashed lines present the maximum value of the p-norm for a given dimension k.

a significant change has been detected in $||\overrightarrow{\mu_{W_0}} - \overrightarrow{\mu_{W_1}}||_p$ (False negative bounds).

The following guarantees can be proved [1]:

Theorem 1: False positive bounds. If $||\overrightarrow{\mu}||_p$ remains almost constant within W, the probability that the algorithm shrinks the window at this step is at most δ.

Theorem 2: False negative bounds. Suppose that for some partition of W into two parts $W_0.W_1$ (where W_1 contains the most recent items), we have $||\overrightarrow{\mu_{W_0}} - \overrightarrow{\mu_{W_1}}||_p > 2\epsilon_{cut}$. Then, with probability $1 - \delta$, the algorithm shrinks W to W_1 or shorter.

The algorithm is based on the idea that if the means have changed sufficiently, the data come from the distributions centered around different means. (65) used the Hoeffding's inequality in order to derive a bound for the empirical means difference of two windows. We generalize this bound to any k-dimensional feature space and p-norm applying the inequality for multivariate data, see Formula (5.10). Let n, n_0 and n_1 denote the sizes of the window W, W_0 and W_1. Then, the ϵ_{cut} value we obtain is the following:

$$\epsilon_{cut} = k^{1/p} \left(\frac{1}{2m} \ln \frac{4}{k\delta'} \right)^{\frac{1}{2}} \tag{5.12}$$

where $\delta' = \frac{\delta}{n}$ and $m = \frac{1}{1/n_0 + 1/n_1}$. The bound is derived in Appendix A.1.

Similarly as in (65), it is possible to provide a more sensitive value for ϵ_{cut} assuming normal distributions of data. In this case, it can be easily proven that ϵ_{cut} has the following form:

$$\epsilon_{cut} = k^{1/p} \left(\left(\frac{2}{m} \sigma_W^2 \ln \frac{2}{k\delta'} \right)^{\frac{1}{2}} + \frac{2}{3m} \ln \frac{2}{k\delta'} \right) \tag{5.13}$$

where σ_W^2 is the observed variance of the p-norm of the elements in the window W.

A comparison of ϵ_{cut} value from Formula (5.12) and from Formula (5.13) is performed and illustrated in Figure 5.3. Since Formula (5.13) assumes normal distributions of data and includes a term of the observed variance, the ϵ_{cut} becomes tighter than the one from Formula (5.12) (assuming that the variance is small).

5.5 Validation

In this section, we present some applications of the multivariate data stream analysis algorithm to WCE data. We start with color analysis (3D RGB vector), then we show some results for 2D lumen-contraction analysis and, finally we move to 1D signal of turbid density. For more results of video description using sequential feature analysis, please refer to Appendix B.

[1] The proof of the theorems is given in Appendix A.1

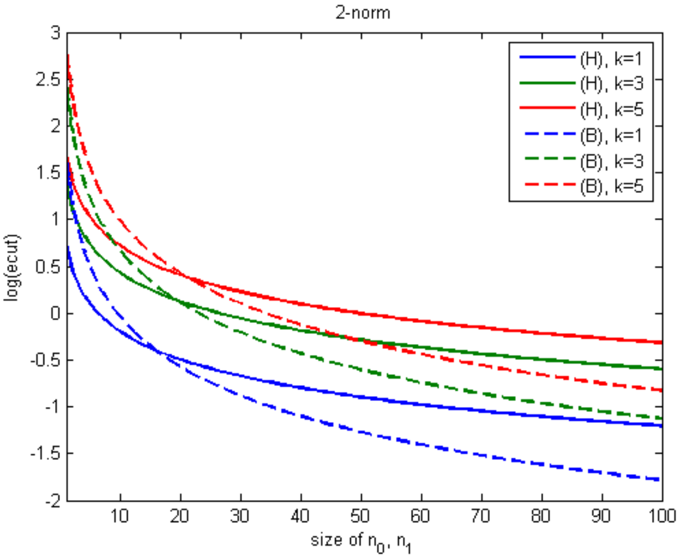


Figure 5.3: Comparison of the ϵ_{cut} value from Formula (5.12) (continuous line) and from Formula (5.13) (dashed line). Variance was fixed to 0.01 and $p = 2$. Plots are shown in logarithmic scale. Y-axis represents the number of samples $n_0 = n_1$.

5.5.1 Color analysis

In the first experiment, we apply our method to estimate the change in color signal. To this end, we calculate the mean color for each vertical line of the motility bar and represent it with a vector of three values R, G and B. Next, we apply our algorithm to obtain the points in which color changes. The results are shown in Figure 5.4. The video is displayed in 10 lines (each line represents the information from 2400 frames of WCE). In each line, 3 images are shown (from top to bottom): 1) motility bar, 2) mean color of vertical line in motility bar, and, 3) the result of our method. The following parameters were used $\delta = 0.1$ and $p = 2$. As it can be seen, the result of our method describes well the mean color information of the motility bar.

5.5.2 Joint contraction-lumen analysis

In the next experiment, we evaluate the lumen size and the contraction density. We set the following parameters: $\delta = 0.1$ and $p = 2$. As a contraction density descriptor, we use the one from Section 4.2.1 and as lumen size descriptor, we use the one from Section 4.2.2. Results are presented in Figure 5.5. The video is displayed in 10 lines (each line represents the information from 2400 frames of WCE). Each line is composed of 3 images (from top to bottom): motility bar, contraction density information and lumen size information. The color legend is displayed

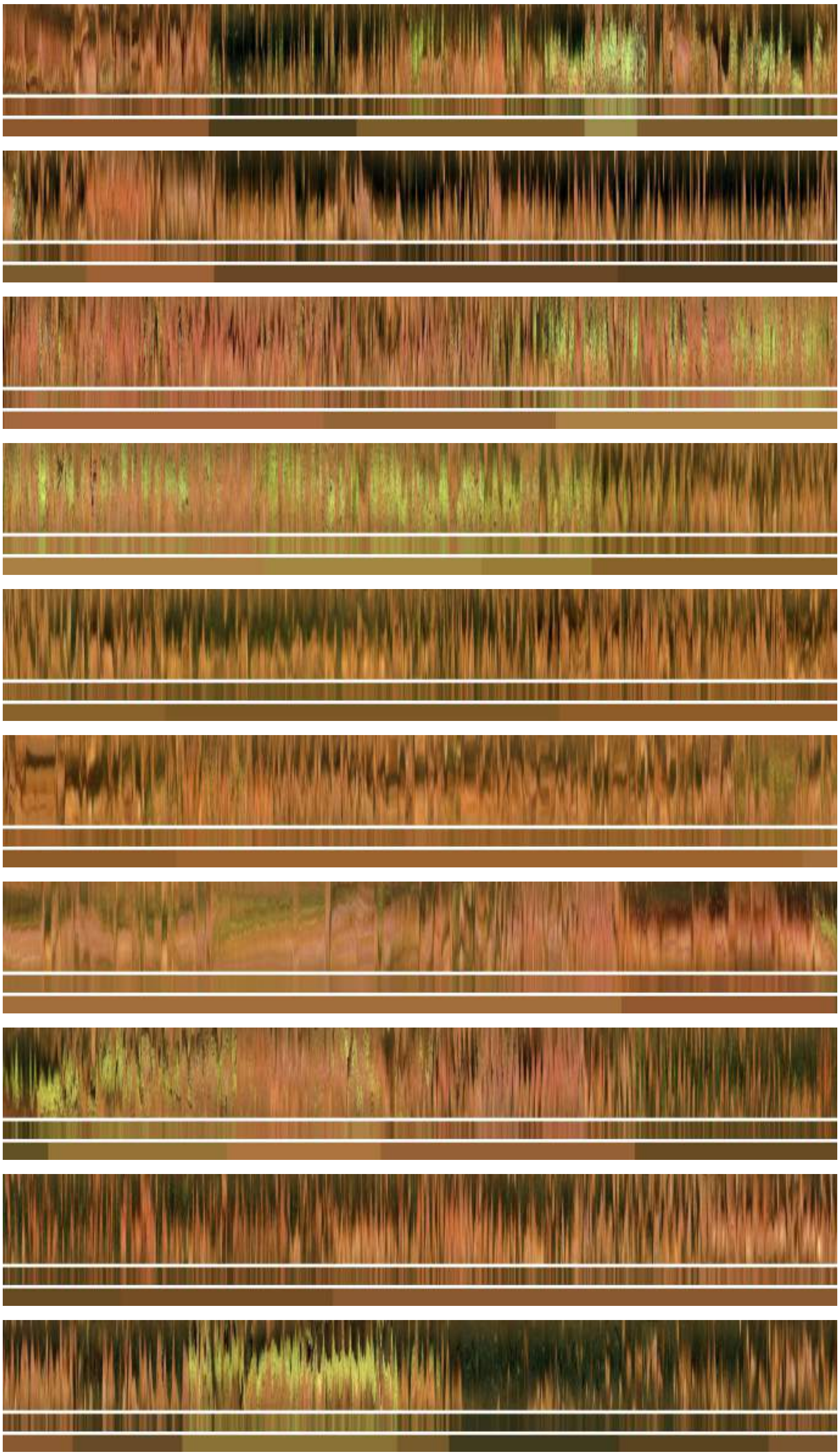


Figure 5.4: An example of color analysis based on the multivariate stream analysis.

in Figure 5.5(a). As it can be seen, the color coding adapts well to the information displayed in the motility bar, for both descriptors: contraction density and lumen size.

We implemented the method in Matlab and ran it on Intel I5-2520 CPU machine. The time needed to obtain constant motility segments for a motility bar build from 28000 frames was approx. 40 seconds.

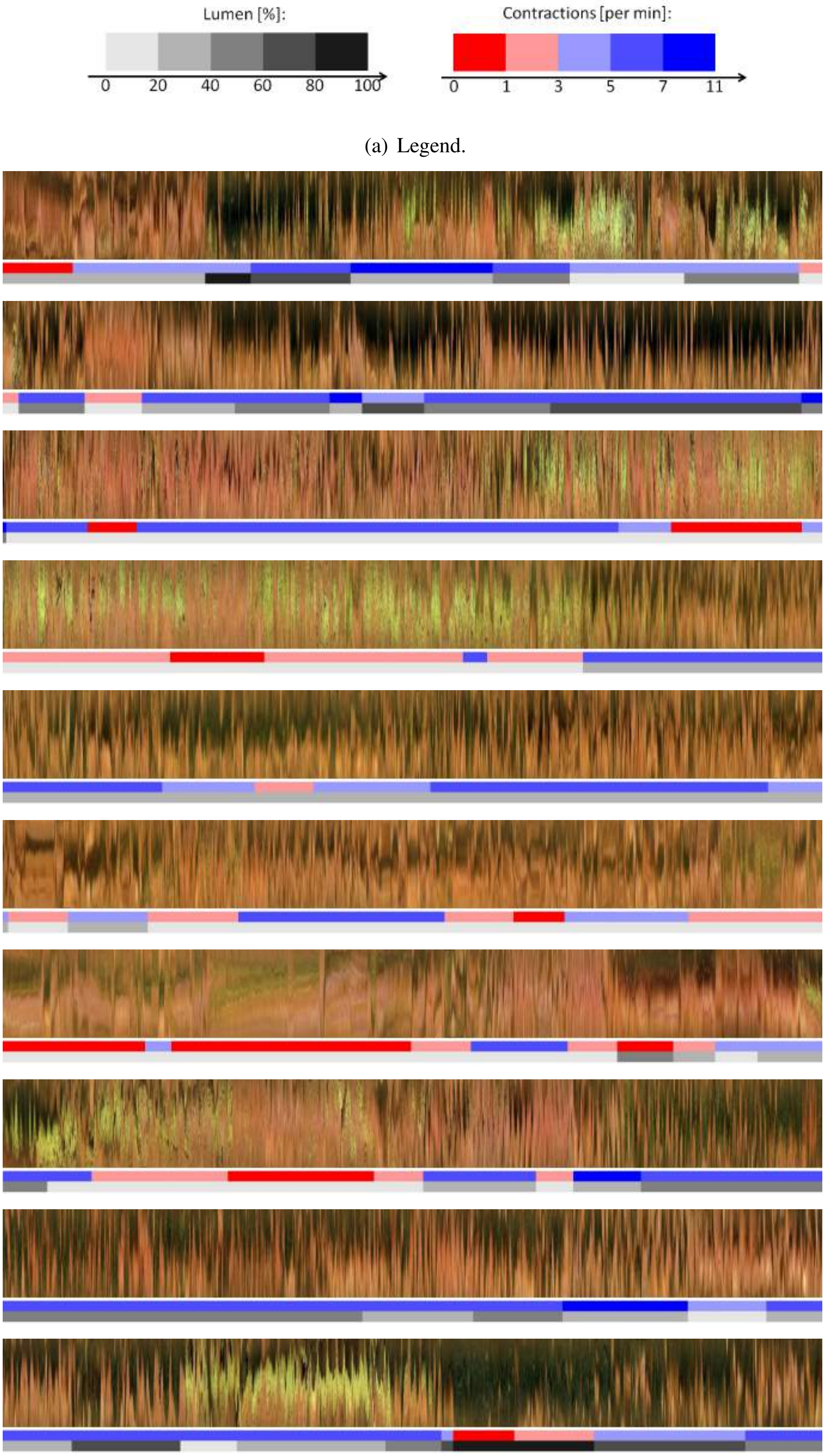
Tunnel segments Once we have obtained the segments in the motility bar in terms of: 1) the contractile activity and 2) the lumen size, we can define tunnel sequences. Tunnel sequences are static sequences with an open lumen. Using the result of joint lumen-contraction analysis, one can easily define these sequences. In Figure 5.6, some examples of tunnel sequences are shown. We have defined tunnel sequences as the ones with the lumen size larger than 50% of the motility bar and a contraction density smaller than 1 contraction per minute.

5.5.3 Intestinal content analysis

In the last experiment, the algorithm is applied to the intestinal content. We use the detection results of the system presented in Section 4.3.1. In order to be able to apply the sequential analysis, similarly as for contractions, we estimate the density of the detections using 1 minute sliding window. In case of intestinal content, we differentiate two sequences the one occupied by intestinal content and the one that is not occupied by intestinal content (clear). To distinguish between the two, we fix a threshold to 30% meaning that, if more than 30% of frames inside one-minute sliding window are detected as intestinal content, an intestinal content sequence is defined. Otherwise, we detect clear sequence. The results for one video are shown in Figure 5.7. The video is displayed in 10 lines,with 2400 frames each. Each line showws (from top to bottom): 1) motility bar and 2) intestinal content information. Intestinal content is marked in green. As it can be seen, the binary description fits well to the information displayed in the motility bar.

5.6 Discussion

In this chapter, we have introduced a method for the analysis of the WCE data using a sequential feature analysis approach. We defined a sequence as a zone of video, where a norm of k-dimensional means is constant. We proposed a new definition of concentration-like inequality for the norm of k-dimensional means. Using this inequality, a generalization of *AdWin* algorithm to multivariate data streams was provided. We showed the utility of the tool using different signals obtained from WCE video, in particular, we analysed: 3D color information, 2D contraction-lumen information and 1D intestinal content signal. The visual analysis of the



(a) Legend.

Figure 5.5: An example of joint contraction-lumen analysis based on the multivariate stream analysis.

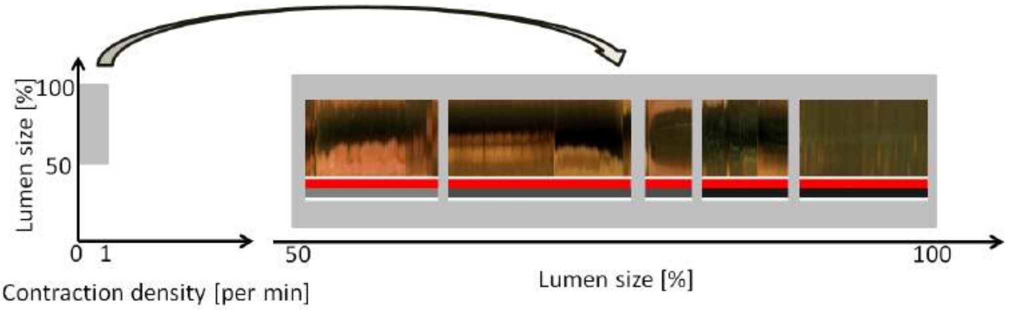


Figure 5.6: Tunnel detection. On the left-hand side, the part of the space where tunnel segments are found. On the right-hand side, some examples of tunnel sequences with different lumen size.

results shows that the segments of constant mean adapt well to the information provided in motility bar. More results on the sequential feature analysis are shown in Appendix B.

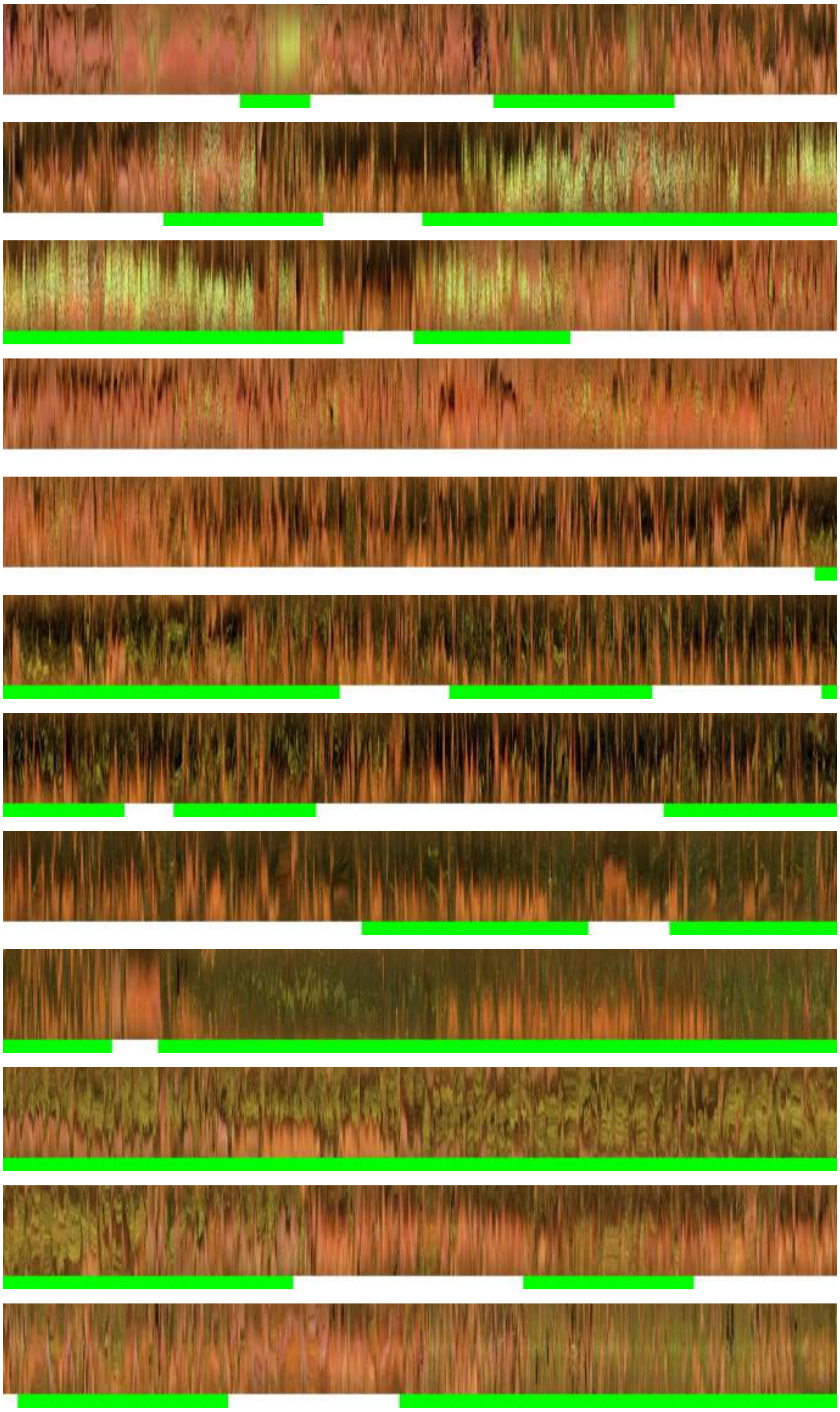


Figure 5.7: An example of intestinal content analysis based on the multivariate stream analysis.

6

Clinical importance of features

6.1 Introduction

In the previous chapter, we have shown how to obtain sequential features of different motility phenomena, in particular: contraction density, lumen perimeter and turbid. Our next goal is to find features that are useful to build a normality model of intestinal motility. Moreover, we expect the features to be discriminative, indicating subjects with abnormal motility behavior. In this Chapter, we show how well our features are fitted to this problem.

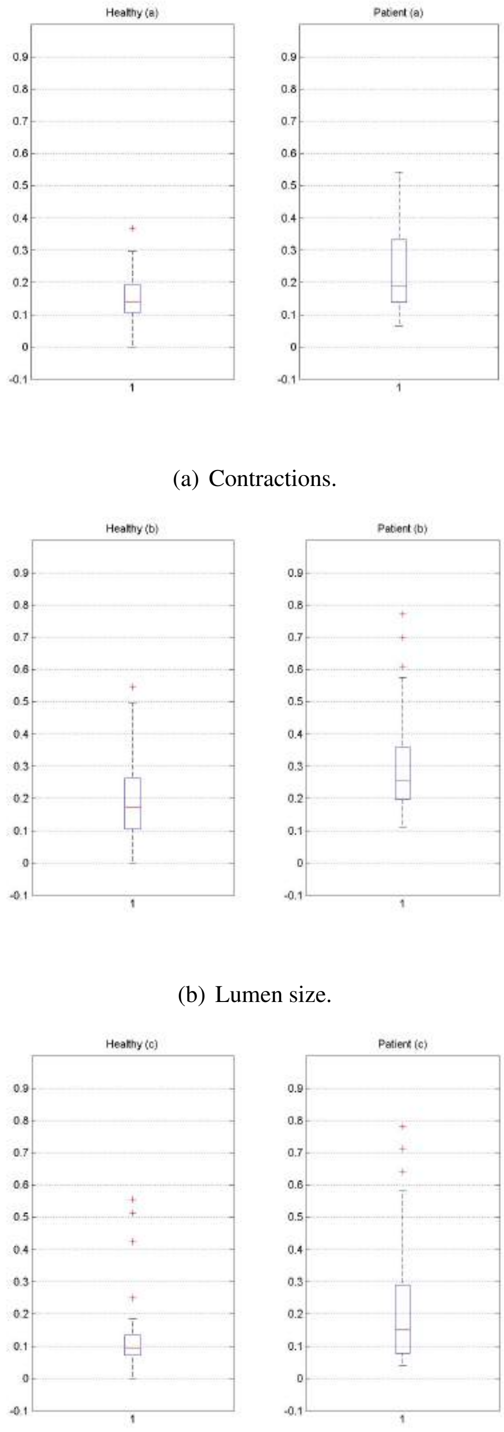
6.2 Methodology

In order to evaluate the features, we use a training set composed of 85 WCE videos of healthy volunteers and a test set of 45 WCE videos of severe intestinal dysmotility patients, 40 WCE videos of healthy volunteers subjects.

Each video is represented as a histogram of features. In particular, we represent a video with three different histograms: 1) contractile activity, 2) lumen size and 3) turbid distribution. Fore each feature, we use a training set to learn a normality model. To this aim, we simply pick the case that has minimal median distance to all healthy subjects. Since each subject is represented as a histogram, histogram intersection distance is used. Next, we evaluate the distances to the normality histogram in the test set and evaluate the following hypotheses: 1) abnormal motility patients have the same distance to the normal contractile density distribution as healthy cases, 2) abnormal motility patients have the same distance to the normal lumen distribution as healthy cases and, 3) abnormal motility patients have the same distance to the normal intestinal content distribution as healthy cases. To ensure statistical significance of the result, a statistical test is performed.

6.3 Results

The results, for both populations (healthy subjects and severe intestinal dysmotility patients), are presented in Figure 6.1. As it can be seen, the distribution of healthy subjects is different than the one of severe intestinal dysmotility patients for all three features: contractions, lumen size and intestinal content distributions. Moreover, a two-tailed t-test has shown that the samples come from different populations giving the following results: 1) contractions $p = 2.4192e - 04$, 2) lumen size $p = 8.4908e - 04$, and 3) intestinal content 0.0052. One can conclude that the proposed features are discriminative between healthy subjects and severe intestinal dysmotility patients.



(a) Contractions.

(b) Lumen size.

(c) Intestinal content.

Figure 6.1: Boxplots of different features: (a) contractions, (b) lumen size and (c) intestinal content. A healthy population is shown on the left while severe intestinal dysmotility patients are shown on the right.

7

Image labeling systems

7.1 Introduction

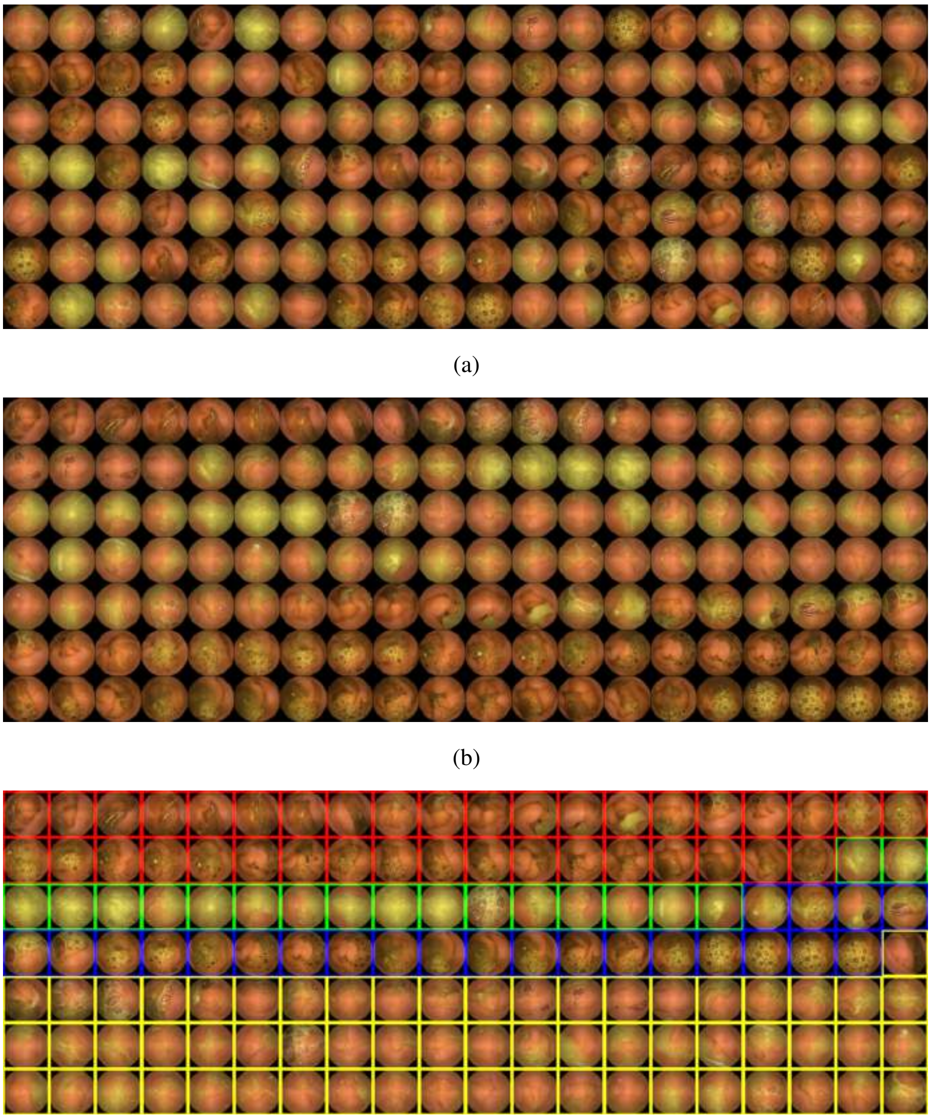
In order to build classifiers for WCE data (such as the ones presented in Section 4.3.1 or Section 4.3.2), one needs to collect a large data set of labeled images. Robust classifiers can be built when the training set is representative of the data population. Therefore, to overcome this problem, one needs to construct a wide labeled training set. It is well known that labeling is a human activity domain [1]. Hence, expert knowledge, time and effort are needed to label the data, making the whole process highly expensive. Moreover, when spending long time on data labeling, an oracle/expert gets tired and errors can be easily introduced, thus, as a result, the labeling becomes inconsistent.

The objective of efficient labeling algorithms is to minimize the oracle effort. This effort can be minimized with the help of computer-aided systems. First, the system should come up with a proposal of the label. This proposal is based on the knowledge gained by the system during the labeling process. Then, the human operator faces two possible decisions: to accept the system proposal or to change the sample label. In practice, these two options have a non symmetric cost for the human operator: accepting the model proposal can be efficiently implemented with a low cognitive load for the operator assuming it by default, while changing a label has a larger cost consisting in manual intervention while labeling the frame. This cost can be evaluated by the number of interventions with the system during the labeling process (74) (e. g. number of ”clicks”).

Hence, the system that minimizes the oracle effort should address two issues: 1) which rule the system should follow when giving the proposition of the label and, 2) how the data should be organized while being displayed to the user. The first question has been tackled with active learning techniques (for revision of active learning techniques, see Section 7.2). Let us briefly comment on the second question. A natural way to display the data is driven by data similarity and not by randomness (see Figure 7.1(a)). It is convenient to display similar samples together. If data was obtained sequentially, in time, it can be assumed that data acquired in instance i and $i + 1$ are similar (see Figure 7.1(b)). In case, when the data have not been sampled from a sequential process or when the samples come from highly dynamic events, the similarity can be defined in some feature space. In this case, similar images can be grouped into cluster-structure (see Figure 7.1(c) for an example of data grouping in color histogram feature space). The assumption done here is that, in some well defined feature spaces, it is more probable that similar samples share the same labels than samples far away in the feature space.

In this chapter, we introduce two applications that deal with efficient data labeling. One is based on concepts of sequential learning and discovers samples of interest from the point

[1] Here, it is assumed that humans are highly accurate while labeling the data.



(a)

(b)

(c)

Figure 7.1: An example of image ordering in case of Wireless Capsule Endoscopy data for two class classification problem: clear frames vs. intestinal content frames: a) random order, b) sequential in time, c) order according to the frames similarity with respect to the color features (color mark indicates different clusters obtained with k-means algorithm - similar images).

of view of the model that is being constructed (e. g. samples not represented in the current training set). This application takes advantage of the fact that the WCE data are sequential in time. The other application deals with the problem of error-free labeling, where all labels have to be revised by an expert. We refer to the former one as interactive labeling application and to the latter one as error-free [1] labeling application. Error-free labeling is a batch approach that looks for similarity between WCE frames. More in detail, the proposed applications can be described as follows:

Interactive Labeling: If we consider training as a sequential process (in time), the training problem can be partially overcome by the integration of online learning methodologies and an ”intelligent” reduction of the samples to be added into the training set. Given a training set at a given time, an ”intelligent” system should add to the training set only those data samples that are not represented, or are under-represented, in the previous version of the set. In other words, the training set should be enlarged by those new data samples that enrich the representability of the classification models while avoiding unnecessary sample redundance.

Error-free Labeling: In this set-up, we consider batch setting for applications, where all training data must be checked by the oracle to ensure the correctness of the labeling. In order to ensure that all data are correctly labeled, the oracle has to revise visually all the elements in the data set. A new element is labeled only if it has been seen by the oracle. This is not a case of direct application of active learning techniques, as they are focused on maximizing classifier performance in the test set and not on minimizing the oracle effort, when labeling the whole training set. So, active learning techniques are not designed for efficient error-free labeling and, thus, a need for labeling applications appears that minimizes the effort of the user, while providing error-free labeling.

7.2 Background: Active learning

The majority of the work that is similar in spirit to our set-up is based on active sampling techniques for active learning.

In active learning, the learner interactively chooses which data points to label with the hope of minimizing the number of required labels. Most of these strategies have been based on the assumption that a good heuristic for minimizing the number of samples to label is to perform

[1] The term error-free labeling refers to the fact that all data and their labels proposals are revised by an expert, it is a difference with Active Learning where only some samples are being displayed to the oracle. Clearly it is possible that an expert will miss-label some data according to the criteria of different expert.

an *efficient search through the classifier hypothesis space* (75). The querying process is guided by the principle that by selecting those labels that will shrink the set of classifiers (consistent with the labels seen so far) as fast as possible, the number of queries will be minimized. This hypothesis is used by a variety of specific algorithms, but in all cases the process includes a classifier fitting step after each query (that can involve a single sample or a small batch of samples) (76, 77, 78). This fact severely limits the applicability of these methods to small scale problems or to problems which can be solved by using learning methods that can operate in pseudo-linear or better time.

Most of the proposed methods for active learning follow the algorithmic scheme represented by Algorithm 5 based on fitting a classifier and selecting data points according to some classifier-based criterion/a.

Algorithm 5 Classifier-based Active Leaning

```
Input: A set of unlabeled, indexed data samples $\mathbf{X} = \{\vec{x_i}\}_{i=1,\dots,n}$.
Input: A budget $m$ representing the maximum number of queries we can afford.
Input: A number $s$ representing the number of samples to be queried at each active learning
    step.
1: Sample $s$ points at random from the set of unlabeled data and query their labels.
2: $i \leftarrow s$
3: while $i \leq m$ do
4:     Fit a classifier (or a set of classifiers) to the data labeled so far.
5:     Select $s$ points from the set of unlabeled data by following a specific sampling strat-
       egy and query their labels.
6:     $i \leftarrow i + s$
7: end while
Output: the classifier trained with the labeled data set.
```

Active learning methods can be discriminated by considering the sampling strategy they follow: the `Select` procedure in step 5 of Algorithm 5. For example, *density sampling* methods sample from maximal-density unlabeled regions (79, 80). On the other hand, *uncertainty sampling* methods sample the regions where the trained classifier is least certain (81). The combination of density and uncertainty criteria has also been explored and it is called *representative sampling*. This approach explores the clustering structure of uncertain samples for selecting the most representative ones (82). Using a different heuristic, *instability sampling* approaches are based on sampling from those regions that maximally change the classifier decision boundary (83).

In order to increase the efficiency of these sampling strategies, several research lines have been proposed. One of the most successful lines is to take benefit from the use of ensemble learning methods (84). For example, *Query-by-committee* (85) selects samples that cause maximal disagreement amongst an ensemble of hypotheses. Another interesting line has been the development of hybrid methods, such as the method presented in (86), where the parameters selection strategy can be adaptively updated after each actively sampled point.

Most of these methods share a common characteristic: the need of fitting a classifier (step 4 of Algorithm 5) every time a new sample (or a small batch of samples) is queried in order to implement the sampling criterion. This means that a type of the classifier (or a set of classifiers) should be chosen *a priori*. Moreover, in the case where the type of the classifier is to be changed, the whole process of active learning should be repeated, since the optimal labels for one classifier types do not have to be optimal for different classifier type (independently of the sampling strategy).

The computational complexity of the classifier becomes a problem, when dealing with a large number of samples. State-of-the-art algorithms such as Support Vector Machines involve inverting a kernel matrix, which has complexity of $O(n^3)$. This complexity, which can be affordable for training a single classifier, becomes a problem when repeatedly applied to large scale datasets. In these cases, the use of online learning techniques becomes mandatory.

Unfortunately, many active learning algorithms suffer from the problem of inconsistency, where (even with infinite number of samples) the obtained result can be far from the optimal one (87, 88, 89). Why active learning algorithms can be inconsistent? Suppose that data has some build-in order and, thus, can be represented by some structure (e. g. partitions or clusters). Note that the active learner has no previous knowledge on this data structure. Therefore, the goal of the active learner is two-fold: to actively discover the data structure (data structure exploration) and to actively learn the optimal discrimination between different classes/partitions (data structure exploitation) (89). In this context, (87) introduces the term of ”missing-cluster” effect referring to unexplored parts of the input data space (some elements of the exploited data structure can be missing). This effect is observed, when the active learner focuses too much on finding the optimal discrimination between classes and does not pay enough attention to the structure discovery step.

Partition-based active learning. An approach to active learning and active discovery that does not require any classifier training step and does not suffer from the problem of inconsistency has been proposed in (88), inspired by the pioneering work of (90). This algorithm belongs to a novel class of methods that aim *to exploit the cluster structure in the data*. The intuition behind this approach is simple: if there is a distribution of the data in pure and separable clusters, we only need to label one sample per cluster to get a good labeling because the rest of the members of the cluster can be implicitly labeled with the same label. Under this assumption, the problem is reduced to active search for an optimal partition of the data and the most critical part is how to find pure clusters as fast as possible by efficiently exploring the data set.

One of the main advantages of the hierarchical sampling method for active learning (88) is that it is not based on retraining multiple times a classifier, but on a sampling process that makes only one assumption on data distribution: the data are distance-clusterizable in the Euclidean space. The classifier is trained once the exploration of the data structure has produced a satisfactory solution or once the available labeling budget for the problem is spent. The other advantage of the algorithm is that this sampling strategy provides a bound on the empirical labeling error giving, at any time, the interval in which the true labeling error lies with high probability. This property of the sampling algorithm allows for labeling complexity analysis, indicating the number of queries to be seen in order to obtain a given labeling accuracy.

The hierarchical sampling method is a specific case of a more general paradigm called *partition-based sampling* (in statistics, also referred as *cluster-based sampling*), characterized by the use of a data structure that represents all possible data clusters or partitions. In (88), the chosen data structure is a hierarchical clustering tree based on the Ward's method (91). This tree, which is computed offline and in an unsupervised way, is a static data structure that defines the search space for the learner. The learner can "navigate" among all possible prunings of the tree by using active sampling strategy. Given a labeling budget, it has been shown that this active sampling strategy is clearly better than the random sampling strategy (which is a necessary condition for any active learning methodology) in the presence of label-aligned data clusters, and it is not worse (except, maybe, for some pathological cases) than random sampling, when these structures are not present in the data.

7.3 Interactive Labeling

First, we describe an interactive labeling system that allows, in an efficient way, 1) to detect frames that are not represented in the training set, 2) to obtain statistics of intestinal content and clear frames that are not represented in the training set, and 3) to iteratively increase the representability of the training set in an ”intelligent” way by reducing significantly the number of clicks related to manual labeling. Note that in this set-up the oracle does not revise all labels.

To this aim, we propose the following algorithm for interactive labeling of a set of new images optimizing the user feedback (see Algorithm 6).

Algorithm 6 Interactive labeling algorithm.

```
Input: A set of labeled data $L$
Input: $M_1$ a discriminative model trained on $L$
Input: $U$ a set of unlabeled data
Input: $C$ a criterion to select the most informative frames from $U$
Select the subset of samples $N = \{x_j^N\}$ from $U$ such that they are considered as under-represented by the labeled set $L$.
Evaluate the subset $N$ with $M_1$, assigning a label $l_j$ to every data sample $x_j$ from $N$.
i=1;
while there are elements in $N$ do
    Evaluate the elements of $N$ with respect to the criterion $C$ and get the set of $n$ most informative samples $I \subset N$ (with the purpose of minimizing the expected number of expert clicks).
    Delete the elements of $I$ from $N$.
    Present the samples from $I$ to the user with their associated label.
    Get the user feedback (nothing for samples with correct labels, one click for each wrongly classified sample).
    Update $L$ by adding the elements of $I$ and theirs user-corrected label.
    Perform an online training step for $M_i$ by adding the elements of $I$ to the model, getting $M_{i+1}$.
    Evaluate the elements of $N$ with $M_{i+1}$, assigning a label $l_j$ to every data sample $x_j$ from $N$.
    $i = i + 1$;
end while
Output: Updated set of labeled data and updated model
```

The critical step in order to minimize the number of clicks is to choose a good criterion for Step 6, since it represents the main strategy of choosing the order of presentation of the samples to be labeled by the user.

To this end, we studied three different sorting policies for the elements of N. These policies are based on the following criteria: 1) to choose those elements that are far from the training data L and far from the boundary defined by M_i, 2) to choose those elements that belong to the most dense regions of N, and 3) to choose the elements in a random way.

More specifically, we define them in the following way:

Criterion 1 (C1) *Distance of data to the classifier boundary and training data.* In this criterion, two measurements are combined: 1) The data are sorted from the farthest to the nearest distance with respect to the classifier boundary. This scheme assumes that the classifier, while proposing labels, will commit errors with higher probability for the samples that are far from the boundary than for the ones that are relatively close to the boundary. 2) The data are sorted from the farthest to the nearest ones with respect to the training data. This scheme assumes that the classifier, while proposing labels, will commit errors with higher probability for the samples that are far from the training set than for the data that are relatively close to the known data. A final sorting is performed in the data by adding the ranking indices of the two previously described schemes.

Criterion 2 (C2) *Data density.* Each sample is sorted decreasingly with respect to a data density measure in its environment. This scheme assumes that the classifier should learn more quickly if we first provide samples from the zones with higher density. Data density can easily be computed as the mean distance to the k-nearest neighbors of the sample.

Criterion 3 (C3) *Random order.* The order of presentation of the samples is randomly determined.

7.3.1 Methodology

The goal of the interactive labeling system is two-fold: 1) to detect, for each new video, the set of frames that are not represented in the training set, and 2) to label those frames with minimal user effort. To this end, we propose a system design with two main components (see Figure 7.2):

1. A data density estimation method that allows fast local estimation of the density and distance of a data sample to other examples, e.g. from the training set (see Step 3 of the algorithm for interactive labeling).

2. An online discriminative classifier which allows to sequentially update the classification model M_i (see Step 2, 10 and 11 of the algorithm for interactive labeling).

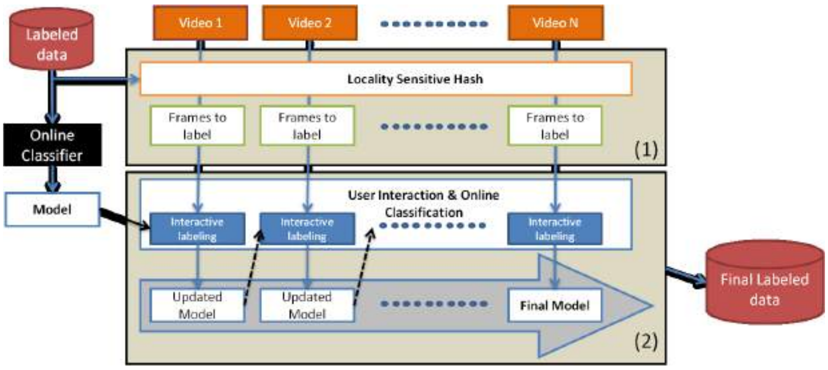


Figure 7.2: The interactive labeling system architecture with its two main components: 1) Detection of frames not represented in the training set and 2) Labeling of frames and model enlarging using an online classifier method.

Fast Density Estimation As previously commented, the local density of a data sample x_i with respect to a data set can be easily estimated by computing the mean distance from x_i to its k-nearest neighbors. The simplest solution to this problem is to compute the distance from the sample x_i to every sample in the data set, keeping track of the ”k-best so far”. Note that this algorithm has a running time of $O(nd)$, where n is the cardinality of the data set and d is the dimensionality of samples.

Because of the excessive computational complexity of this method for large data sets, we need a flexible method that allows from one side effective measurements of characteristics of large data and, from the other side, introducing new unseen data into the training set for enlarging the data representation. An example of such flexible method is the Locality Sensitive Hashing (LSH) approach (92). LSH allows to quickly find a similar sample in a large data set. The basic idea of the method is to insert similar samples into a bucket of a hash table. As each hash table is created using random projections over the space, several tables can be used to ensure an optimal result (92). Another advantage of LSH is the ability to measure the density of the data in a given space vicinity by analyzing the number of samples inside the buckets (see Figure 7.3). In order to evaluate if the new sample improves the representation of the data set, the space density of the training set is estimated. If the new sample is in a dense part of the space, then, the sample is considered redundant and, thus, it is not used to improve the model. Otherwise, the sample is used to enlarge the training set.

The density D of the sample x is estimated according to the formula:

$$D(x, T_r) = \sum_{i=1}^{M} ||B_i||, \tag{7.1}$$

where M is the number of hash tables, $||B_i||$ is the number of elements in the bucket where the

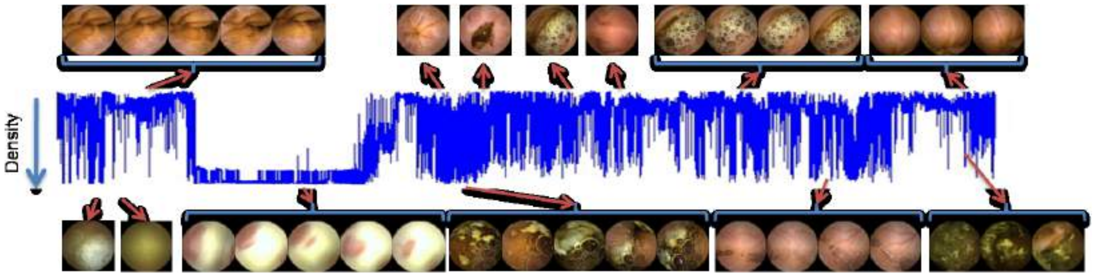


Figure 7.3: Example of training set density estimation for a test video using LSH. The images show the zones of high and low density with respect to given labeled set L.

new element x is assigned and T_r represents the training set. The subset of non-represented samples in the training set $N = \{x_1^*, ..., x_m^*\}$ from new unlabeled data U is defined as:

$$N = \{\forall x \in U : D(x, T_r) < T\}, \tag{7.2}$$

where T represents a fixed threshold.

Note that Formula (7.2) expresses the condition that the new samples fall in buckets with low density of the training set. That is, if the new sample is in a dense part of the space, then, the sample is not considered to improve the model. Otherwise, the sample is used to enlarge the training set.

Online Classifier Taking into account that our classifier must be retrained with thousands of images/feature vectors of up to 256 components, using an online classifier is a must. Online classifiers are able to update the model in a sequential way, so, if needed, the classifier can constantly learn from new data, improving the quality of label proposal process. In order to optimize the learning process, the data are sorted according to the previously described criteria. A kernel-based online Perceptron classifier (93) is used because of its simplicity and efficiency. As mentioned in chapter 4, the main information used to detect intestinal content frames is the color. In order to reduce the dimensionality of the data, each image is quantized into 256 colors. As a result, each frame is represented by 256 color histogram. The score for a given sample takes this form:

$$S(x) = \sum_{j=1}^{K} \alpha_j K(v_i, x), \tag{7.3}$$

where $\{v_1, \alpha_1), ..., (v_k, \alpha_k)\}$ is the set of training vectors with their corresponding estimated weights $(\alpha_1, ..., \alpha_k)$ by the learning algorithm, when minimizing the cumulative hinge-loss suffered over a sequence of examples and $K()$ is a kernel function (in our case, we apply Radial Basis Function).

7.3.2 Validation

We test the algorithm in the problem of intestinal content labeling. For our experiments, we consider a set of 40 videos obtained using the WCE device. 10 videos are used to build the initial classification model M_1, and the other 10 to evaluate the proposed interactive labeling system. In the test, the 10 videos are sequentially processed. If needed, at each iteration, the training set could be increased by a new set of frames that improves the data representation. Additionally, the validation set of 20 videos is used in order to evaluate the error of the final intestinal content/clear frames classifier.

In the experiments, we show that: 1) the proposed system reduces the effort needed for data labeling, 2) the first criterion *Criterion 1 (C1): Distance of data to the classifier boundary and training data* gives the best results, 3) the global performance of intestinal content/clear frames classifier is improving, while enlarging the training set, and 4) the LSH optimizes the computation process.

Table 7.1 shows that all three proposed schemes reduce the number of clicks. Even using random order improves a lot with respect to the naive approach. This phenomenon can be explained by the fact that the colors in a given video are similar. From the results, it can be concluded that *Criterion 1* appears to be the best sorting criterion for interactive labeling. Intuitively, the samples that are far from the boundary are classified with high confidence. However, when dealing with the frames that are not similar to the ones in the training set (and are far from the ones in the training set), the classifier confidence can be erroneous since the

Table 7.1: Results of different interactive labeling criteria.

				#clicks	
Video	#frames	#strange frames	Criterion_1	Criterion_2	Criterion_3
Video1	35847	4687	103	103	147
Video2	51906	10145	211	213	316
Video3	52777	5771	270	270	376
Video4	56423	13022	86	90	151
Video5	55156	7599	68	68	131
Video6	33590	17160	381	389	617
Video7	17141	1072	8	8	39
Video8	26661	5437	88	97	151
Video9	14767	1006	28	28	76
Video10	22740	1993	63	63	110
Average clicks per video	-	-	1.5%	1.5%	2.9%

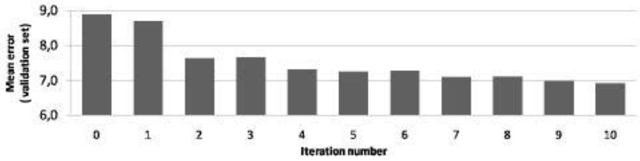
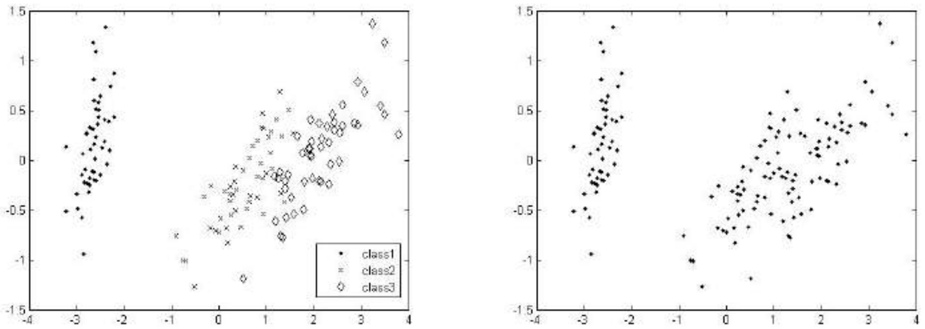


Figure 7.4: Mean error on validation set of 20 WCE videos.

samples can come from an undiscovered cluster (e. g. missing-cluster effect (87)). Therefore, when introducing examples where the classifier wrongly assigns the label (user needs to switch the label), it is highly probable that the boundary changes adapting to the newly discovered cluster.

Introducing new data into the training set improves the final classifier performance and reduces the error by 2% after 10 iterations of the algorithm (where each iteration corresponds to a newly introduced video) (Figure 7.4). Furthermore, the LSH in average reduces the number of frames to check by more than 80%. This means that tested videos have about 20% of the frames that are ”strange”. While inserting new frames into the classifier model, it can be seen that at each iteration some data that are not represented in the training set are found. The conclusion that can be drawn is that, in order to create a good training set for intestinal content/clear frame classification, the number of 20 WCE videos is not enough.



(a)

(b)

Figure 7.5: An example of data distribution in 2D (data from IRIS dataset after applying PCA). a) data with all labels uncovered, b) data with unknown labels.

7.4 Efficient error-free active labeling

Efficient error-free active labeling, contrary to Interactive Labeling: 1) is based on batch setting, and 2) provides a scheme for efficient revising of all batch elements by an expert.

The algorithm looks for an optimal labeled-aligned cluster structure. This is important, since, if we knew all the labels, our problem would become trivial (see Figure 7.5(a)): we could present the data to the labeler in an optimal cluster-based organization to minimize the effort. For example: if a cluster is pure (only one class), we can label it (all elements inside the cluster) using only one intervention (e. g. one click). But at the beginning of the labeling process, the labels are unknown (see Figure 7.5(b))! The problem of finding optimal label-aligned cluster structure can be seen as joint cluster *exploration* and *exploitation* task. *Exploration* is responsible for a discovery of data structure while *exploitation* is in charge of finding an optimal discrimination between classes in the data structure (89).

In our algorithm, the user decides whenever it is more convenient to explore or to exploit the current data structure. In order to represent the data structure, we use tree representation of hierarchical clustering. The navigation through all possible space partitions represented by the hierarchical clustering tree is based on the strategy of partition-based active learning.

7.4.1 Partition-based active learning

In this subsection, we formalize the definition of partition-based active learning. The paradigm can be described in its most general terms in the following way: Let $\mathbf{X} = \{\vec{x_1}, ..., \vec{x_n}\}$ be the set of unlabeled data points, $\mathbf{C}^k$ be some partition set of $\mathbf{X}$ and C^k_r be an element of this partition containing some data from $\mathbf{X}$ such that for a given partition space (search set of the algorithm) $\mathfrak{C} = \{\mathbf{C}^0, ..., \mathbf{C}^u\}$ the following statement holds:

$$\forall \mathbf{C}^k \in \mathfrak{C} : \bigcup_{C_r^k \in \mathbf{C}^k} C_r^k = \mathbf{X}$$

The objective is to *efficiently search through the space of partitions* for an optimal partition $\mathbf{C}^{opt}$ of $\mathbf{X}$ such that all data points belonging to any element $C_r^k \in \mathbf{C}^{opt}$ can be automatically labeled with high confidence by using the labeled data. Its basic steps are described by Algorithm 7.

The algorithm has four associated procedures that need some explanation: `Initialize`, `Select`, `Bound` and `Search`.

The procedure `Initialize` selects some initial partition $\mathbf{C}^0$ from the partition space $\mathfrak{C}$. In the case of tree representation (e.g. Ward clustering), the algorithm should start with the root partition, a partition of one element containing all data points.

The procedure `Bound` returns the conservative mislabeling error estimate within a given element C_r^k of the current partition $\mathbf{C}^k$ given the data points seen so far. This conservative estimate indicates the confidence of the learner on the data points labeling after seeing only some data points using an estimator. If a majority label is used as an estimator, this procedure can be implemented in an elegant way, by using the bounds derived from the tails of the binomial or multinomial distribution (88).

Suppose there are η possible labels and that their proportions in a given element C_r^k of the partition are $p_{C_r^k,l}$ for $l = 1, \dots, \eta$. Then, after labeling all the samples, the error induced by assigning all points in C_r^k to its majority label is $\epsilon_{C_r^k} = 1 - max_l(p_{C_r^k,l})$. At any iteration k of the algorithm, we can associate with each element C_r^k of the partition $\mathbf{C}^k$ an empirical estimate of the labeling error $\epsilon_{C_r^k}$ and a confidence interval within which we expect the true error $p_{C_r^k,l}$ to lie:
$[p^{LB}_{C_r^k,l}, p^{UB}_{C_r^k,l}] = [max(p_{C_r^k,l} - \frac{1}{n_{C_r^k}} - \sqrt{\frac{p_{C_r^k,l}(1-p_{C_r^k,l})}{n_{C_r^k}}}, 0), min(p_{C_r^k,l} + \frac{1}{n_{C_r^k}} + \sqrt{\frac{p_{C_r^k,l}(1-p_{C_r^k,l})}{n_{C_r^k}}}, 1)]$,
where $n_{C_r^k}$ is the number of points sampled from C_r^k up to that moment. In this way, we have defined a statistical measure about the error introduced if we automatically label the samples of an element C_r^k after seeing n labels. The conservative estimate of this error is defined as `Bound(`C_r^k`)=`$1 - p^{LB}_{C_r^k,l}$ (88).

The procedure `Select` determines which element C_r^k of the current partition $\mathbf{C}^k$ is sampled. There are several alternatives to implement this procedure but the most evident choice, taking into account the objective of minimizing the number of queries, is to focus the selection process on those regions of the space that are still under-sampled. A simple implementation

Algorithm 7 Partition-based Active Leaning

```
Input: A budget $m$ representing the maximum number of queries we can afford.
Input: A structure $\mathfrak{C}$ to represent space of partitions.
Input: A number $s$ representing the number of samples to be queried at each active learning step.
Initialize $\mathbf{C}^0$ from $\mathfrak{C}$
$i \leftarrow 0$ {number of seen data points}
$k \leftarrow 0$ {iteration of the algorithm}
while $i < m$ do
    $j \leftarrow 0$
    while $j < s$ do
        Select an element $C_r^k$ from $\mathbf{C}^k$.
        Sample a random point $x$ from $C_r^k$ and quiery its label $l$.
        $j \leftarrow j + 1$
    end while
    $i \leftarrow i + s$
    for each $C_r^k \in \mathbf{C}^k$ do
        Compute Bound($C_r^k$) {an estimate of the mislabeling error of data points within $C_r^k$}.
    end for
    Search for a new better partition $\mathbf{C}^{k+1}$ in $\mathfrak{C}$
    $k \leftarrow k + 1$
end while
for each $C_r^k \in \mathbf{C}^k$ do
    Assign to each data point in $C_r^k$ a label based on the majority label of the points in $C_r^k$.
end for
Output: the labeled set to train a classifier.
```

of this idea is to choose an element C_r^k with a probability proportional to $\omega_{C_r^k}$ $\texttt{Bound}(C_r^k)$, where $\omega_{C_r^k}$ is the fraction of the dataset covered by element C_r^k (proportional to the number of data points inside this element). The term $\texttt{Bound}(C_r^k)$ reduces the labeling effort in those regions that can be automatically labeled with high confidence, while the factor $\omega_{C_r^k}$ enforces exploration of the large elements of the data structure.

The most critical part of the algorithm is the definition of a searching strategy for finding $\mathbf{C}^{opt}$. After seeing sufficient number of samples, the algorithm must be able to generate a new and probably better partition $\mathbf{C}^{k+1}$ from $\mathbf{C}^k$. In this framework, we can define the following order relation between partitions: A partition $\mathbf{C}^{k+1}$ is better than a partition $\mathbf{C}^k$ if $\sum_{C_r^{k+1} \in \mathbf{C}^{k+1}} \omega_{C_r^{k+1}} \texttt{Bound}(C_r^{k+1}) < \sum_{C_r^k \in \mathbf{C}^k} \omega_{C_r^k} \texttt{Bound}(C_r^k)$, that is, if the fraction of the dataset that can be automatically labeled with confidence in $\mathbf{C}^{k+1}$ is larger than the one in $\mathbf{C}^k$.

A simple but efficient alternative when implementing *Partition-based Active Learning* used in (88), is to consider a reduced partition space: instead of considering all possible partitions [1] of $\mathbf{X}$, they only consider the subspace formed by the partitions defined by a given pruning of the tree representing a hierarchical clustering of $\mathbf{X}$. In this case, the navigation strategy can also be simplified and the $\texttt{Search}$ procedure in Algorithm 7 consists of selecting a good pruning that can estimate the majority label with high confidence of the pre-calculated hierarchical clustering tree (see (88), for more details).

7.4.2 Methodology

Figure 7.6 shows the application flowchart. The core of our approach is based on the partition-based active learning. In order to build an application for efficient label-free labeling, the following elements should be considered:

1. An engine responsible for *exploration* and *exploitation* of the data structure (e. g. partition space).

2. A strategy for choosing the elements to be displayed, for both, individual data elements due to structure *exploration* step and a group of elements representing some part of the data structure to enable to the user the possibility of the data structure *exploitation*.

3. An interface to show the data according to displaying strategy and to accept user interactions with the system.

[1] The number of possibilities, a set of n elements can be partitioned into nonempty subsets is represented by the Bell number and is denoted B_n. For example, $B_{10} = 115975$.

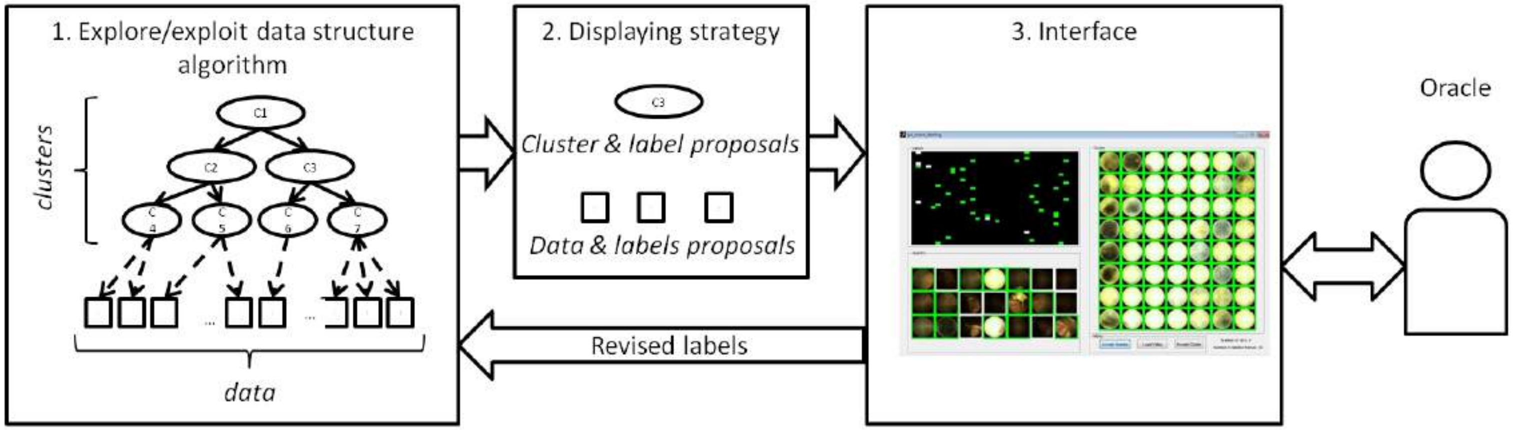
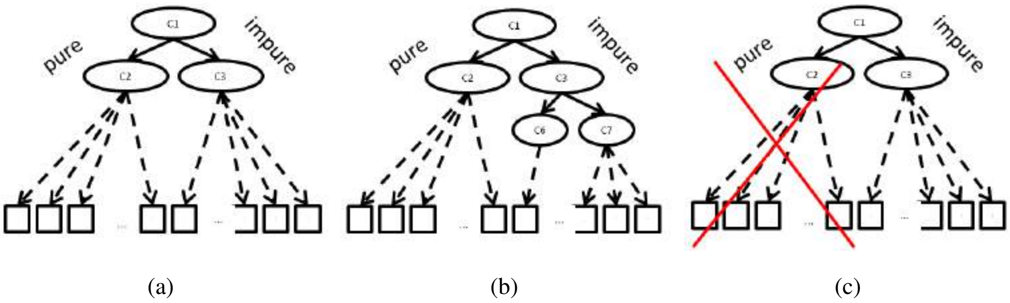


Figure 7.6: Efficient error-free active labeling application flowchart.



(a) (b) (c)

Figure 7.7: An illustration of the possible actions on data structure. a) hypothetical data structure with one pure and one impure cluster, b) *exploration* - the user decides to label samples from impure cluster, as an effect the algorithm descends in the hierarchical structure and uncovers new clusters, and c) *exploitation* - the user decides to revise the labels in a pure cluster, and as an effect all data from this cluster are labeled.

Algorithm for data structure exploration/explotation. Algorithm 8 shows the basic steps of the error-free active labeling algorithm. Since the Algorithm is based on partition-based active learning, we only comment the differences with respect to Algorithm 7.

The procedure `Display` sends the data samples with label proposals to the display and waits for the user interaction. After executing this procedure, the algorithm receives a group of revised labels that can be used either for structure exploitation or for exploration.

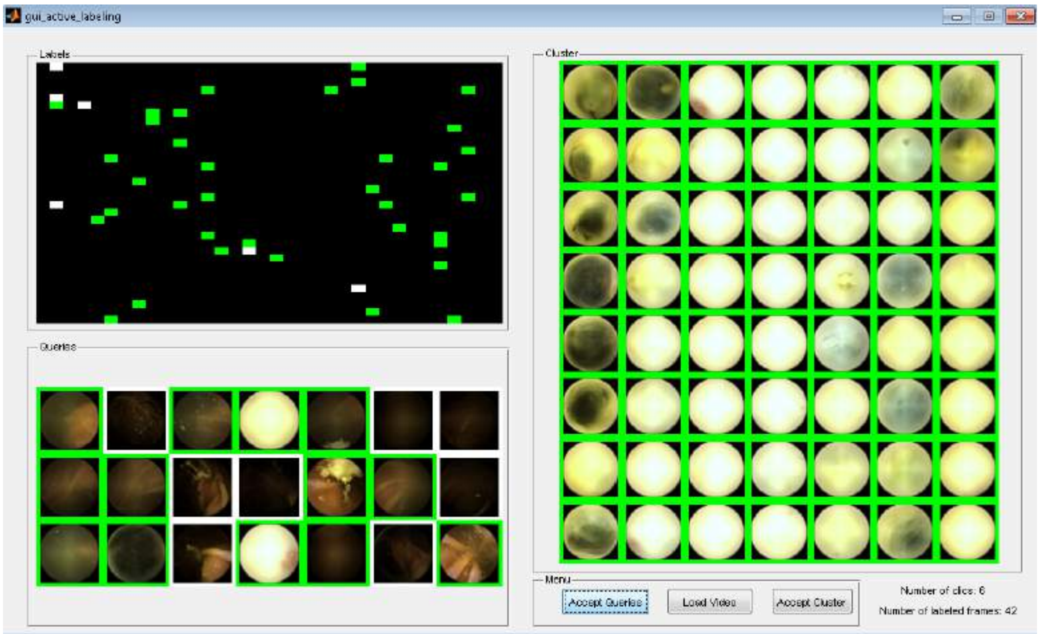
At each step of the *exploration*/*exploitation* of data structure, the algorithm produces three outcomes: 1) the samples from impure clusters with the label proposals $\{\mathbf{x}, \hat{\mathbf{L}}\}$, 2) the current clustering structure grouping the similar data samples and their label proposals $\{\mathbf{C}, \hat{\mathbf{L}}\}$ and 3) the purity measure of each cluster $\{\mathbf{C}, \mathtt{Bound(C)}\}$. The question that arises is how to present all this information to the user (see Figure 7.7).

It is straight-forward to present to the oracle the samples from impure clusters $\{\mathbf{x}, \hat{\mathbf{L}}\}$ and it is a necessity in the data structure *exploration* (see Figure 7.7(b)). If the user decides to label those samples, the algorithm will learn about true labels $\{\mathbf{x}, \mathbf{L}\}$. This results in dividing the current cluster structure into more refined one.

When it comes to the clusters $\{\mathbf{C}, \hat{\mathbf{L}}\}$, it is unfeasible to display all the partitions at once. In this case, two different criteria can be applied: one that maximizes the information gain and other that minimizes the oracle's effort. The first one means to display the most impure cluster from the current data structure and to ask the user to correct the labels. But this displaying strategy is contrary to the problem set-up, while it is not minimizing the oracle's effort. Moreover, the algorithm already provides to the oracle some samples from impure clusters due to

Algorithm 8 *Exploration/exploitation* algorithm

```
Input: A data structure $\mathbf{C}$ to represent any partition of $\mathbf{X}$.
Input: A number $s$ representing the number of samples to be queried at each algorithm step.
$\mathbf{C}^0 \leftarrow \mathbf{X}$
Bound($\mathbf{C}^0$)
$\hat{\mathbf{L}}(\mathbf{X}) \leftarrow 1$ {Arbitrary label for label proposal list.}
$\mathbf{L}(\mathbf{X}) \leftarrow empty$ {Revised labels list.}
$i \leftarrow 0$
$k \leftarrow 0$
while unseen labels do
    $j \leftarrow 0$
    $\mathbf{x} \leftarrow empty$ {List of data points to query.}
    while $j < s$ do
        Select an element $C$ from $\mathbf{C}^k$.
        Sample a random point $p$ from $C$.
        $\mathbf{x} \leftarrow \mathbf{x} \cup p$
        $j \leftarrow j + 1$
    end while
    Find the purest cluster $C^p$ in $\mathbf{C}^k$
    Display samples $\{\mathbf{x}, \hat{\mathbf{L}}\}$ and cluster with label proposals $\{C^p, \hat{\mathbf{L}}\}$ and get true labels
    in case of exploration $\mathbf{L}(\mathbf{x})$ or exploitation $\mathbf{L}(C^p)$.
    $i \leftarrow i + s$
    if explore then
        for each $C \in \mathbf{C}^k$ do
            Compute Bound($C$), a conservative estimate of the mislabeling error within $C$.
            Update label proposals $\hat{\mathbf{L}}(C)$
        end for
        Search for a new better partition $\mathbf{C}^{k+1}$
    else if exploit then
        $\mathbf{C}^{k+1} \leftarrow \mathbf{C}^k / C^p$
    end if
    $k \leftarrow k + 1$
end while
Output: the labeled set $\{\mathbf{X}, \mathbf{L}\}$ to train a classifier.
```



Number of clics: 6
Number of labeled frames: 42

Figure 7.8: An example of interface used to display images during the labeling process of WCE video. (left-top) Assigned labels (green intestinal content frames, white clear frames), (left-down) file displaying samples from all clusters with the label proposal, (right) file showing the purest cluster with label proposal.

data structure exploration step.

The second approach is to present to the user the purest cluster and its labels proposals $\hat{\mathbf{L}}(C^p)$ and asking them to correct, hopefully, a few samples and accept the whole cluster. Once the oracle accepts the cluster, all the samples are assigned a correct label and this part of the space is no longer sampled by the algorithm. If the data can be represented by the label-aligned data structure with large clusters then, with a few (or non) oracle interventions, a whole cluster can be labeled (see Figure 7.7(c)). Moreover, if the data can be divided in a few such clusters, the labeling process becomes very efficient and cheap in terms of oracle effort. This is a step of the algorithm that exploits the current data structure.

To be able to jointly *explore/exploit* the data structure and minimize the oracle's effort, both data, 1) samples from impure clusters and 2) samples form the purest cluster, should be presented in parallel to the user. In this set-up, the user decides whenever it is convenient to exploit the current data structure or to continue exploring to get finer cluster-label alignment.

Interface for interactive labeling of endoscopic frames. The interface is shown in Figure 7.8. The interface is composed of 3 fields, one to display the data labels that have been revised by the oracle, one for displaying the samples from impure clusters and one to display the samples from the purest cluster. The user can choose in which field they are willing to interact. If the purest cluster is homogenous, it is favorable to change a few (or none) labels and to accept a large number of labeled samples. Otherwise, the oracle should interact in the field of samples from impure clusters and wait for a pure cluster to appear. Once the oracle has revised the labels in the field of samples from impure clusters (or in the purest cluster field), they should press the button ”accept queries” (or cluster), so that the algorithm comes-up with new data and their labels proposal.

With respect to the interface two questions remain open:

- Number of images to be displayed in each filed.
- Optimal resolution of the image.

These questions are not treated in this chapter, while the answers should be adjusted individually to the data set that is being labeled and to the screen resolution. The general remark is that the parts of the image that are being subject to labeling should be well visible to the user. The user should not spend too much time on visual inspection of a single image and should be able to quickly spot the discriminative (in context of labeling) parts of the image such as: color, structure, shape etc.

7.4.3 Validation

We evaluate the following scenarios:

1. *Random order* - a label has to be provided for each frame individually, the number of oracle interventions is proportional to the number of samples.
2. *Sequential (in time) order* - the label is activated and lasts until it is changed, the number of oracle interventions depends on the dynamics of the process that is being observed. If the process is slow in time then the number of interventions is proportional to the number of classes in the data. If the process is highly dynamic then the number of interventions is proportional to the number of samples.
3. *Proximity in the feature space (hierarchical clustering)* - the data are organized into clusters in the feature space. The number of the oracle interventions depends on the cluster structure. If some large fairly-pure clusters are present, the number of interventions is proportional to the number of clusters. If the data can not be organized into fairly-pure label-aligned clusters, the number of interventions is proportional to the number of samples.

To evaluate the scenarios, data from clear vs. intestinal content frames problem of WCE have been used. The scenarios 2 and 3 are further analyzed. In case of scenario 1, it is assumed that the number of oracle intervention is equal to the number of samples.

In the evaluation, one WCE video of 55156 frames was sub-sampled every 50th frame producing a string of images of length 1104 frames. First, the video was presented to the expert according to the Scenario 2 asking them to label the informative and non-informative frames. Each sample was displayed in a sequential order with a label proposal, if the proposal was incorrect the user could change the label. In this scenario of reviewing and labeling of all the frames, the oracle needed 57 clicks.

Second, the data were presented to the same user using the application presented in Section 7.4.2 askingthem to label the informative and non-informative frames. In the field of samples from impure clusters, 27 frames were displayed. In this scenario the user needed 45 interventions to revise and label 1104 frames of WCE (giving an improvement of 27% with respect to scenario 2).

In order to fully appreciate the utility of the application, we tested the proposed labeling scheme in the task of face database creation. The database contains labeled examples of facial and not-facial images and the goal is to separate the true faces from the false detections. In order to do so, each face detection image is represented by using the Histogram of Gradients (HoG) and the data set is grouped in hierarchical structure. In the experiment, we use 1064 images (of faces and no-faces). At the beginning, all images are assigned to a face class. Using our approach, the user was able to review all labels and get perfect labeling with only 87 clicks.

7.5 Discussion

In this Chapter, two applications for efficient labeling have been presented. One that is based on the concepts of online labeling and the other one on an efficient error-free labeling application.

Interactive Labeling: The methodology is based on two steps: 1) the detection of frames that enrich the training set representation and, thus, should be labeled, and 2) the interactive labeling system that allows to reduce the user effort, in the labeling process, using an online classifier, which sequentially learns and improves the model for the label proposals. The detection of frames that enlarge the data representation has been performed using LSH. The LSH method allows a fast processing for getting efficient results for data density estimation. Three different sorting polices are defined and evaluated for the online classification: 1) *Distance of data to the classifier boundary and training data*, 2) *Data density*, and 3) *Random order*. It is shown that by using adapted sorting criteria for the data, we can improve the label proposal process and, in this way, reduce the expert effort.

Finally, we have observed that enlarging the initial training set with the non-represented frames from unlabeled videos, we achieve an improvement of the classification performance.

Efficient error-free active labeling: The application is based on data similarity in the feature space. This method actively *explores* the data in order to find the best label-aligned clustering and *exploits* it to reduce the oracle effort. At each step of the method, the oracle can decide if it is more convenient to go for data exploitation (the displayed cluster is fairly pure) or for further data structure exploration. The algorithm for each data sample presents a label proposal, based on majority label estimation in the current cluster. The error-free labeling is guaranteed by the fact that all data and their label proposals are visually revised by an expert. Thanks to the clustering structure, this revision can be done in an efficient way reducing significantly the time of constructing a wide set of training samples. This strategy has been compared to the sequential (in time) ordering of the data that should be used, when the data come from steady (or even static) process. On the other hand, the strategy based on proximity in the feature space is favorable for the data, where some large fairly-pure clusters are expected to be found.

8

Conclusions and future work

8.1 Conclusions

In this thesis, a novel computer-aided system for intestinal motility analysis has been presented. The system is based on sequential feature analysis and provides to the user an easily-comprehensible visual description of motility related intestinal events. To this purpose, several tools based either on computer vision concepts or on machine learning techniques have been presented. The conclusions of the thesis can be summarized in the following points:

- *Motility bar: a novel representation of intestinal motility.* A new method for transforming 3D video signal into a holistic image of intestinal motility has been proposed. The method, based on a Dynamic Programming framework, calculates the optimal (from the intestinal motility point of view) mapping from video data into image representation. The motility bar has been validated, showing that the motility information presented on it is very similar to the motility information presented in a WCE video. Moreover, it has been shown that the motility bar reduces significantly the time needed for visual inspection of motility information.

- *Automatic feature extraction.* Four methods for automatic extraction of motility information from WCE have been presented. Two of them are based on the motility bar and two of them are based on frame-per-frame analysis.

  - *Motility bar based features.* A method for contractions detection has been presented and validated. The method is based on Gabor-like filters that detect contractions in the motility bar using different time scales. The results of different filters are joined into a single signal representing contraction positions in the motility bar. The second kind of information extracted from the motility bar has been the lumen perimeter. The detector is based on the assumption that the lumen is a dark region in the image.
  - *Frame based features.* Two frame-based detectors have been introduced. First, a system for detecting intestinal content has presented. This method is able to differentiate between two types of intestinal content: 1) turbid and 2) bubbles. Moreover, the method is able to quantify the amount of intestinal content inside a single frame and, thus, in the whole WCE video. Second, a method for wrinkle frames detection based on mid-level image descriptor has been presented and validated. This method has shown to set-up new state-of-the-art results in the wrinkle frame detection problem.

- *Sequential feature analysis.* A novel formulation of concentration inequality for multi-variate data stream has been introduced. This formulation is sensitive to permutations of vector components. The formulation has been introduced into a robust mean change detection algorithm for data streams. The algorithm has been used to obtain robust representation of segments of constant means (sequential features). The algorithm has been visually validated in the WCE analysis. The following problems have been tested: 1) color segmentation, 2) joint contraction-lumen analysis and 3) intestinal content.

- *Clinical importance of features.* To measure the clinical importance of the sequential features, a set of videos of healthy volunteers and severe intestinal dysmotility patients has been collected. Using this database, we have shown that our sequential features are discriminative to detect subjects with abnormal motility.

- *Efficient labeling systems.* Finally, the problem of intestinal content frames labeling has been addressed and two labeling systems have been proposed.

  - *A system for efficient labeling of WCE frames.* This system is based on concepts from sequential learning and discovers the samples of interest from the point of view of the model that is being constructed. To discover the samples of interest LSH is used, while the problem of sampling is addressed with a criteria from active learning. Finally, this process is incorporated into an online learning setting.
  - *A system for error-free labeling of WCE frames.* The concepts of partition-based active learning have been adapted to an error-free labeling scheme. In this scheme, an expert visually revises all data labels. In order to reduce user effort, the algorithm comes up with a label proposal. This proposal is based on the system knowledge gained during the labeling process.

All these steps are sufficient to provide an extensive visual description of intestinal motility that can be used by an expert as decision support system.

8.2 Future work

Further investigation can be devoted to the topics presented in this thesis.

- *Motility bar: a novel representation of intestinal motility.* The motility bar offers a novel, holistic view into the small intestine motility. It opens new investigation lines in the intestinal motility analysis, allowing to see the events that are difficult to spot by using traditional WCE video analysis. The proposed algorithm could be further improved by placing the lumen position in the center of the longitudinal view. This is not an easy task mainly because of the free capsule movement inside the intestine and due to the fact that in a lot of frames the lumen is not visible. Placing the lumen in the center of the view could be an important step to obtain 3D reconstruction and display of small intestine that could be of high interest to detect lesions and achieve image-guided interventions.

- *Automatic feature extraction.* New motility features should be explored, especially interesting is to deepen into the analysis of intestinal motility that is present in the motility bar. The motility bar presents very rich motility information and in this thesis only small portion of this information has been exploited. For example, new features could be discovered, while analysing more in detail the contractile patterns. In particular, it might be interesting to measure the contractile strength directly in the motility bar (and not only the contractile frequency). Other new features could be discovered, while incorporating the information from manometry into the contractile analysis. For example, one could look for motility-like abnormal patterns in motility bar.

- *Sequential feature analysis.* Since the algorithm for multivariate data stream analysis is a generic one, it can easily be applied to problems not related to Wireless Capsule Endoscopy. As future work, we would like to test it in problems like background adaptation or object detection. Moreover, it would be interesting to evaluate the multivariate concentration-like inequality definition in problems not related to adaptive windowing in streaming data.

- *Clinical importance of features.* The data sets in which the sequential features have been tested are rather small. It would be interesting to validate the features in larger data sets.

- *Efficient labeling systems.* It would be interesting to see how the proposed labeling systems works with not WCE related problems. The main limitation of error-free active labeling is that it is only suited to work with a batch setting. Thus, as a future work, we would like to adapt it to work in streaming setting by introducing the possibility to adapt hierarchical clustering structure to incoming data.

9

Publications

The work presented in this thesis has been partially published in several computer vision and medical journals, book chapters, conferences and partially filed for IP protection.

9.1 Journals

Detection of wrinkle frames in endoluminal videos using betweenness centrality measures for images; Santi Seguí, Michal Drozdzal, Ekaterina Zaytseva, Carolina Malagelada, Fernando Azpiroz, Petia Radeva, and Jordi Vitrià; IEEE Journal of Biomedical and Health Informatics; Accepted.

Adaptable image cuts for motility inspection using WCE; Michal Drozdzal, Santi Seguí, Jordi Vitrià, Carolina Malagelada, Fernando Azpiroz, Petia Radeva; Computerized Medical Imaging and Graphics; Volume 37; Issue 1; January 2013; Pages: 72-80.

Categorization and Segmentation of Intestinal Content Frames for Wireless Capsule Endoscopy; Santi Seguí, Michal Drozdzal, Fernando Vilarino, Carolina Malagelada, Fernando Azpiroz, Petia Radeva, Jordi Vitrià; Information Technology in Biomedicine, IEEE Transactions on; Volume:16; Number:6; November 2012; Pages: 1341-1352.

Functional gut disorders or disordered gut function? Small bowel dysmotility evidenced by an original technique; Carolina Malagelada, Santi Seguí, Sara Mendez, Michal Drozdzal, Jordi Vitrià, Petia Radeva, Javier Santos, Anna Accarino, Juan R. Malagelada, Fernando Azpiroz; Neurogastroenterology & Motility; Volume: 24; Issiue:3; March 2012; Pages: 223-228.

9.2 Book chapters

An Application for Efficient Error-Free Labeling of Medical Images; Michal Drozdzal, Santi Seguí, Petia Radeva, Carolina Malagelada, Fernando Azpiroz, Jordi Vitrià; Multimodal Interaction in Image and Video Applications; Springer Berlin Heidelberg; Year: 2013; Pages: 1-16.

9.3 Patents

SYSTEM AND METHOD FOR AUTOMATIC DETECTION OF IN VIVO CONTRACTION VIDEO SEQUENCES; Inventors: Michal Drozdzal, Santiago Seguí Mesquida, Petia Radeva, Jordi Vitrià, Laura Igual-Muñoz, Carolina Malagelada, Fernando Azpiroz;

Patent publication number: 13228287; Publication date: 2011/9/8; Filed in countries: USP.

SYSTEM AND METHOD FOR SYNCHRONIZING IMAGE SEQUENCES CAPTURED IN-VIVO FOR AUTOMATIC COMPARISON; Inventors: Michal Drozdzal, Santiago Seguí Mesquida, Petia Radeva, Jordi Vitrià, Laura Igual-Muñoz, Carolina Malagelada, Fernando Azpiroz; Patent publication number: 13154544; Publication date: 2011/6/7; Filed in countries: USP.

SYSTEM AND METHOD FOR DISPLAYING MOTILITY EVENTS IN AN IN VIVO IMAGE STREAM; Inventors: Michal Drozdzal, Santiago Seguí Mesquida, Petia Radeva, Jordi Vitrià, Laura Igual-Muñoz, Carolina Malagelada, Fernando Azpiroz; Patent publication number: 2013114361; Publication date: 2013/8/9; Filed in countries: USP.

9.4 Conferences

A new image centrality descriptor for wrinkle frame detection in WCE videos; Santi Seguí, Michal Drozdzal, Ekaterina Zaytseva, Carolina Malagelada, Fernando Azpiroz, Petia Radeva, Jordi Vitrià; 30th IAPR International Conference on Machine Vision Applications; Kyoto, Japan 2013.

Active labeling: Application to wireless endoscopy analysis; Petia Radeva, Michal Drozdzal, Santi Seguí, Carolina Malagelada, Fernando Azpiroz, Jordi Vitrià; International Conference on High Performance Computing and Simulation (HPCS 2012); Pages: 174-181.

Interactive labeling of WCE images; Michal Drozdzal, Santi Seguí, Carolina Malagelada, Fernando Azpiroz, Jordi Vitrià, Petia Radeva; Lecture Notes in Computer Science; IbPRIA 2011; Pages: 143-150.

Aligning endoluminal scene sequences in wireless capsule endoscopy; Michal Drozdzal, Laura Igual, Jordi Vitrià, Carolina Malagelada, Fernando Azpiroz, Petia Radeva; Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on; Pages: 117-124.

9.5 Medical Conferences

Analysis of endoluminal images typifies abnormal intestinal motor behaviour in patients with functional bowel disorders; Carolina Malagelada, Michal Drozdzal, Santi Seguí, Javier Santos, Anna Accarino, Juan R. Malagelada, Jordi Vitrià, Petia Radeva, Fernando

Azpiroz; United European Gastroenterology Week; Berlin, Octubre 2013; United European Gastroenterology Journal 2013; Supplement 1 (1S) A109.

Distinctive Dysmotility Patterns Demonstrated by Computer Vision Analysis of Capsule Endoscopy Images in Patients With Functional Gut Disorders; Carolina Malagelada, Santi Seguí, Michal Drozdzal, Sara Mendez Soriano, Anna Accarino, Javier Santos, Juan R. Malagelada, Jordi Vitrià, Fernando Azpiroz; DDW 2012; Gastroenterology Vol. 142, Issue 5, Supplement 1: S-825-S-826

Appendices

Appendix A

Proofs

In this Apendix the proofs for the equations provided in Section 5.4 are provided.

A.1 False positive and false negative bounds

At every iteration of the algorithm, we have bounded the false positive and false negative rates. The proof is largely based on the one presented in (65) (the proof can also be found in (94)):

Let x be a real-valued random variable. Assume that x is bounded, $x \in [0, 1]$. Let the μ be the expected value of x and let the $\widehat{\mu}$ be the empirical mean of n independently drawn observations $x^1, x^2, ..., x^n$.

Theorem A.1.1 (False positive rate bound)**.** *If* $||\overrightarrow{\mu}||_p$ *remains constant within* W*, the probability that the algorithm shrinks the window at this step is at most* δ/n*.*

Proof. Let W be a window of the data stream that splits in a bi-partition $W_0.W_1$. Assume $||\overrightarrow{\mu_{W_0}}|| = ||\overrightarrow{\mu_{W_1}}|| = ||\overrightarrow{\mu_W}||$ as a null hypothesis. If for any partition of W we have a probability at most δ/n that the algorithm decides to shrink W to W_1, this is equivalent to $Pr\left[||\overrightarrow{\hat{\mu}_{W_1}} - \overrightarrow{\hat{\mu}_{W_0}}|| \geq \epsilon_{cut}\right] \leq \delta/n$.

Since there are at most n partitions W_0W_1, the claim follows by the union bound. Note that, for every real number $l \in (0, 1)$, $||\overrightarrow{\hat{\mu}_{W_1}} - \overrightarrow{\hat{\mu}_{W_0}}||$ can be decomposed as: $Pr\left[||\overrightarrow{\hat{\mu}_{W_1}} - \overrightarrow{\hat{\mu}_{W_0}}|| \geq \epsilon_{cut}\right] \leq Pr\left[||\overrightarrow{\hat{\mu}_{W_1}} - \overrightarrow{\mu_W}|| \geq l\epsilon_{cut}\right] + Pr\left[||\overrightarrow{\mu_W} - \overrightarrow{\hat{\mu}_{W_0}}|| \geq (1-l)\epsilon_{cut}\right]$.

Applying the norm of k-dimensional means bound, we have that:
$Pr\left[||\overrightarrow{\hat{\mu}_{W_1}} - \overrightarrow{\hat{\mu}_{W_0}}|| \geq \epsilon_{cut}\right] \leq 2k \exp\left(\frac{-2n_0(l\epsilon_{cut})^2}{k^{2/p}}\right) + 2k \exp\left(\frac{-2n_1((1-l)\epsilon_{cut})^2}{k^{2/p}}\right)$.

To approximately minimize the sum, we choose the value of l that makes both probabilities equal, i.e. such that $\sqrt{n_0}l\epsilon_{cut} = \sqrt{n_1}(1-l)\epsilon_{cut}$, which is $l = \sqrt{n_1}/(\sqrt{n_0} + \sqrt{n_1})$. For this l, we have: $\frac{-2}{k^{2/p}}(\frac{\sqrt{n_0 n_1}}{(\sqrt{n_0}+\sqrt{n_1})}\epsilon_{cut})^2 \leq (\frac{-2}{k^{2/p}}\frac{n_0 n_1}{(n_0+n_1)})\epsilon_{cut}^2 = \frac{-2}{k^{2/p}}m\epsilon_{cut}^2$, where $m = \frac{n_0 n_1}{n_0+n_1}$.

Therefore, in order to have $Pr\left[||\overrightarrow{\hat{\mu}_{W_1}} - \overrightarrow{\hat{\mu}_{W_0}}|| \geq \epsilon_{cut}\right] \leq \delta/n$,
it suffices to have $4k \exp\left(\frac{-2}{k^{2/p}}m\epsilon_{cut}^2\right) \leq \frac{\delta}{n}$, which is satisfied by $\epsilon_{cut} = k^{1/p}\left(\frac{1}{2m}\ln\frac{4kn}{\delta}\right)^{\frac{1}{2}}$.
□

Theorem A.1.2 (False negative rate bound)**.** *Suppose that for some partition of* W *in two parts* $W_0.W_1$*, (where* W_1 *contains the most recent items), we have* $||\overrightarrow{\mu}_{W_1} - \overrightarrow{\mu}_{W_0}|| > 2\epsilon_{cut}$*. Then, with probability* $1-\delta$*, the algorithm shrinks* W *to* W_1*, or shorter.*

Proof. Let us assume that $||\overrightarrow{\mu_{W_1}} - \overrightarrow{\mu_{W_0}}|| > 2\epsilon_{cut}$. We want to show that $Pr\left[||\overrightarrow{\hat{\mu}_{W_1}} - \overrightarrow{\hat{\mu}_{W_0}}|| \leq \epsilon_{cut}\right] \leq \delta$, which means that with probability at least $1-\delta$, a change is detected. As before, for any $l \in (0, 1)$, we can decompose $||\overrightarrow{\hat{\mu}_{W_1}} - \overrightarrow{\hat{\mu}_{W_0}}|| \leq \epsilon_{cut}$: $Pr\left[(||\overrightarrow{\hat{\mu}_{W_1}} - \overrightarrow{\hat{\mu}_{W_0}}|| \leq \epsilon_{cut}\right] \leq Pr\left[(||\overrightarrow{\hat{\mu}_{W_0}} - \overrightarrow{\mu_{W_0}}|| \geq l\epsilon_{cut}) \cup (||\overrightarrow{\hat{\mu}_{W_1}} - \overrightarrow{\mu_{W_1}}|| \geq (1-l)\epsilon_{cut})\right]$
$\leq Pr\left[||\overrightarrow{\hat{\mu}_{W_0}} - \overrightarrow{\mu_{W_0}}|| \geq l\epsilon_{cut}\right] + Pr\left[||\overrightarrow{\hat{\mu}_{W_1}} - \overrightarrow{\mu_{W_1}}|| \geq (1-l)\epsilon_{cut}\right]$.

Observe that if: $||\overrightarrow{\hat{\mu}_{W_1}} - \overrightarrow{\hat{\mu}_{W_0}}|| \leq \epsilon_{cut}$, $||\overrightarrow{\hat{\mu}_{W_0}} - \overrightarrow{\mu_{W_0}}|| \leq l\epsilon_{cut}$ and $||\overrightarrow{\hat{\mu}_{W_1}} - \overrightarrow{\mu_{W_1}}|| \leq (1-l)\epsilon_{cut}$ hold, by the triangle inequality, we have:

$$||\overrightarrow{\mu_{W_1}} - \overrightarrow{\mu_{W_0}}|| \leq ||\overrightarrow{\hat{\mu}_{W_1}} - \overrightarrow{\hat{\mu}_{W_0}}|| + \epsilon_{cut} \leq 2\epsilon_{cut},$$

that is contradicting the hypothesis. Then, we get:

$$Pr\left[||\overrightarrow{\hat{\mu}_{W_1}} - \overrightarrow{\hat{\mu}_{W_0}}|| \leq \epsilon_{cut}\right] \leq 2k \exp\left(\frac{-2n_0(l\epsilon)^2}{k^{2/p}}\right) + 2k \exp\left(\frac{-2n_1((1-l)\epsilon)^2}{k^{2/p}}\right).$$

We can choose l as before and follow the steps as in the previous proof to show that $Pr\left[||\overrightarrow{\hat{\mu}_{W_1}} - \overrightarrow{\hat{\mu}_{W_0}}|| \leq \epsilon_{cut}\right] \leq 4k \exp\left(\frac{-2}{k^{2/p}} m\epsilon_{cut}^2\right) \leq \frac{\delta}{n} \leq \delta$ as desired. □

Appendix B

Additional results

In this Appendix, we show results of the algorithm presented in Section 5.4. We show the results for 5 different videos of WCE. Each line of image shows one hour of WCE video. The image starts when the capsule enters into the small bowel (SB entrance mark) and ends when the capsule reaches the small bowel exit (SB exit mark). The distance between two white vertical lines is 10 minutes. We show two images for each video: first one, representing the mean color change in the motility bar (Figures B.2, B.4, B.6, B.8, B.10), and, second one, representing the intestinal motility information: contraction density, lumen size, and intestinal content information (Figures B.3, B.5, B.7, B.9, B.11). The legend for the motility information images is shown in Figure B.1.

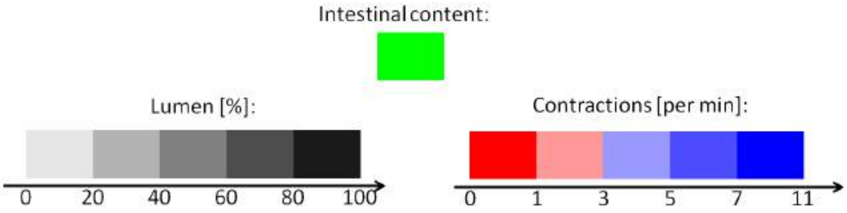


Figure B.1: Legend for the images with motility information.

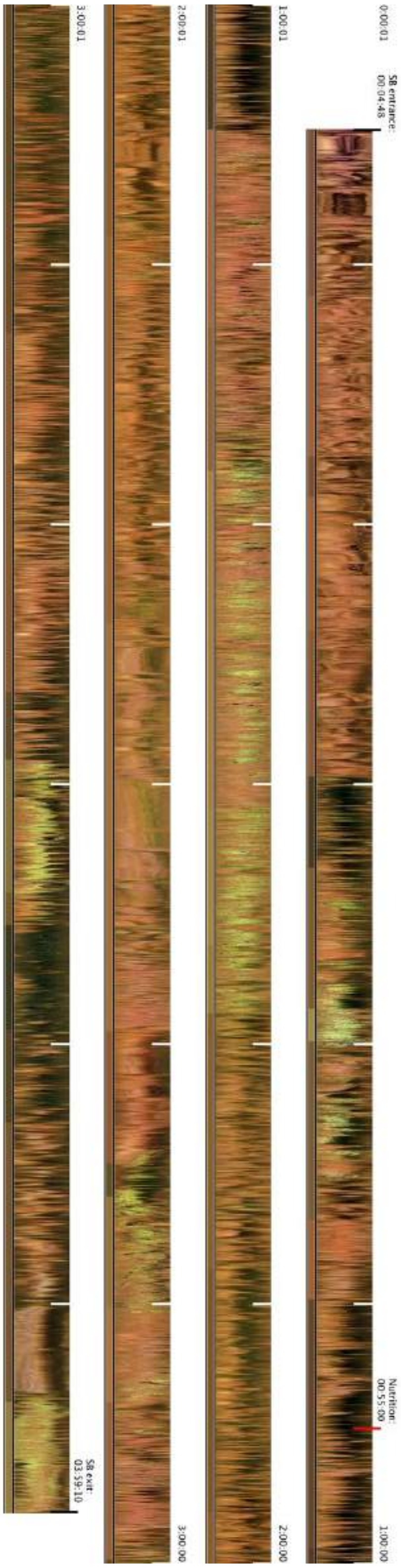


Figure B.2: Video 1. Mean color change.

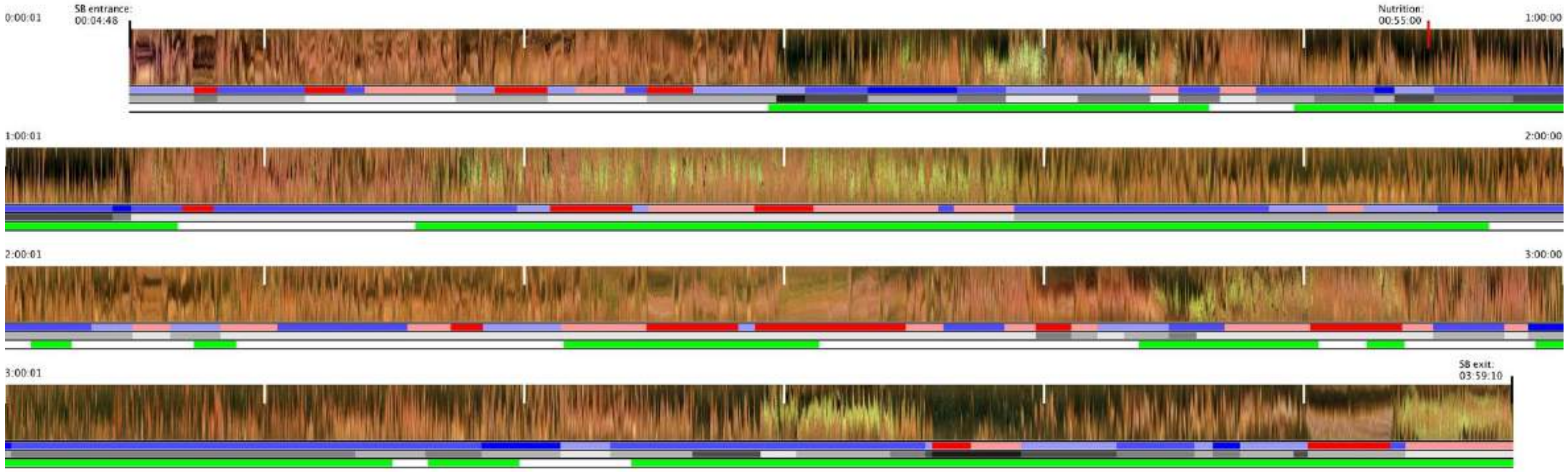


Figure B.3: Video 1. Motility descriptors.

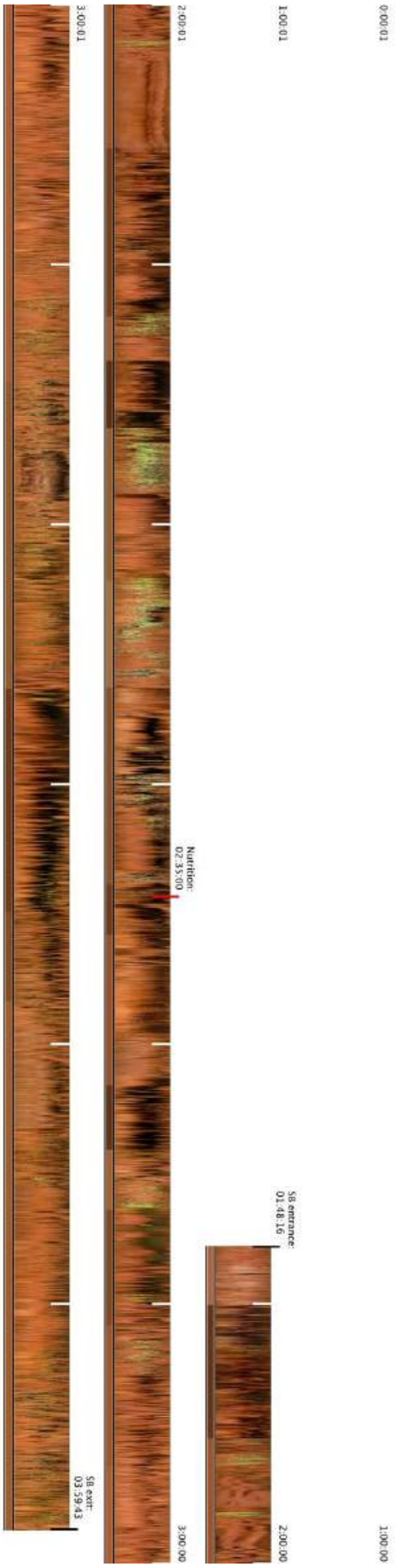


Figure B.4: Video 2. Mean color change.

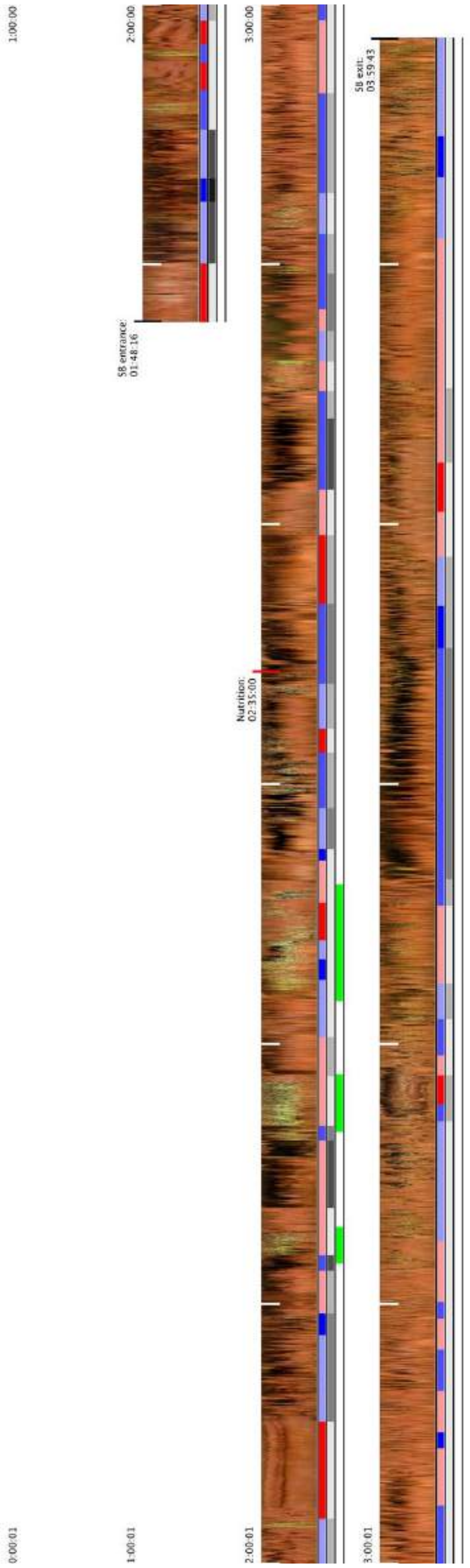


Figure B.5: Video 2. Motility descriptors.

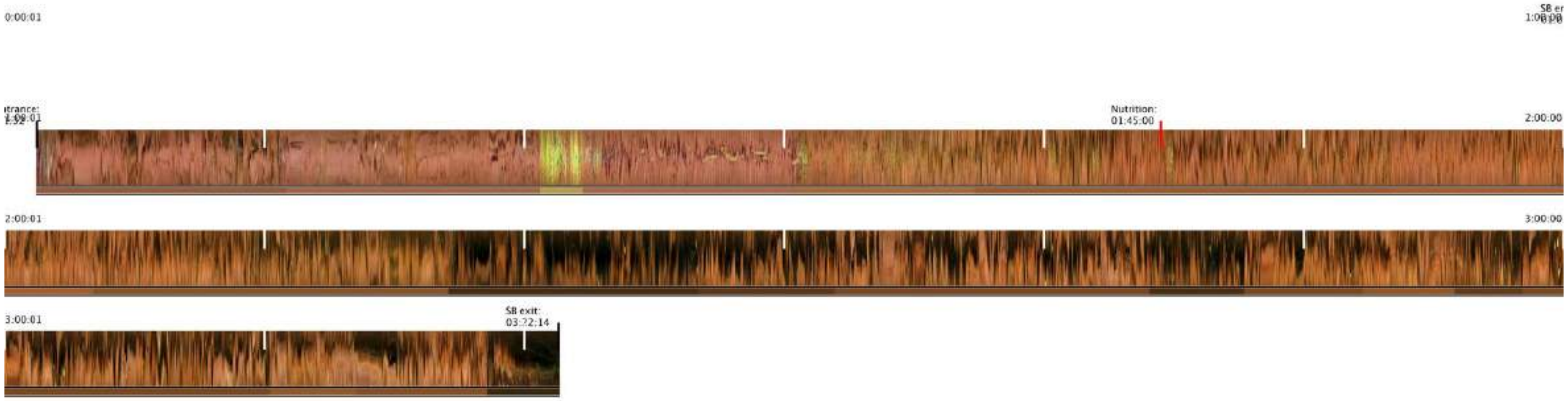


Figure B.6: Video 3. Mean color change.

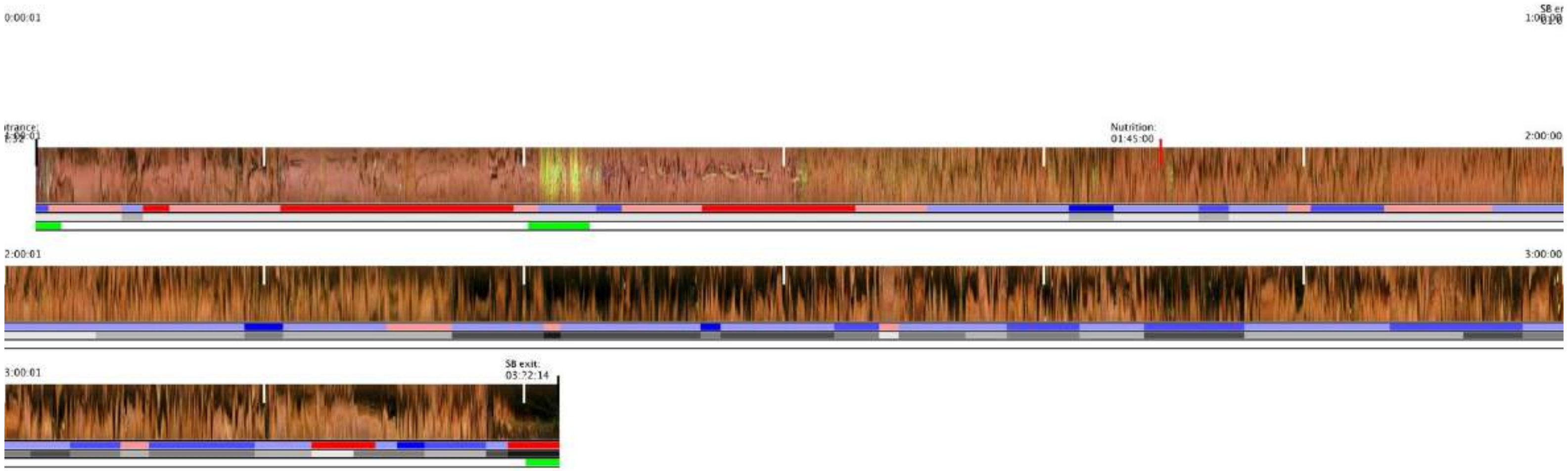


Figure B.7: Video 3. Motility descriptors.

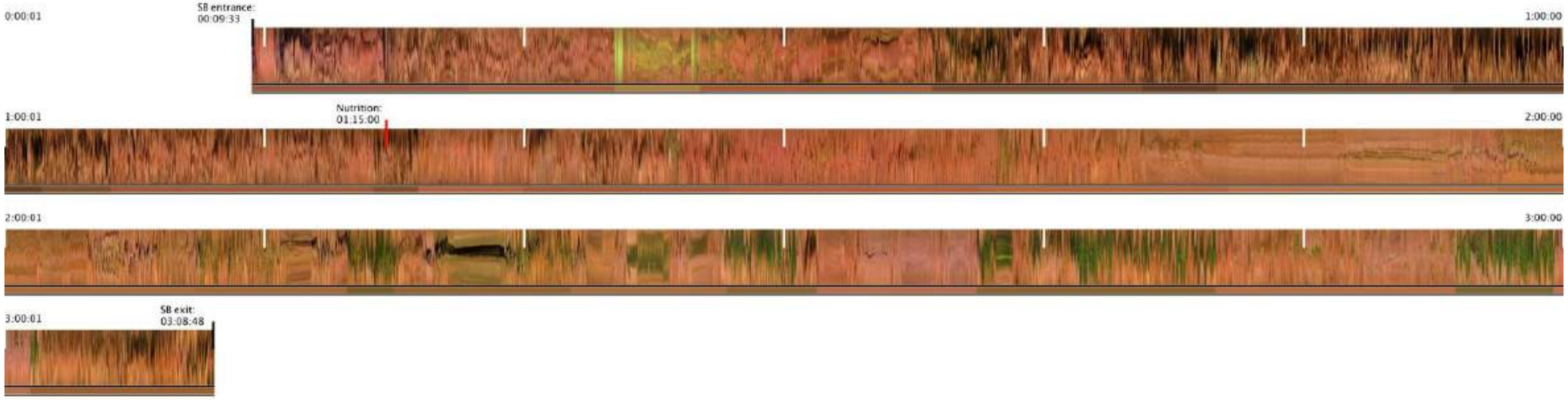


Figure B.8: Video 4. Mean color change.

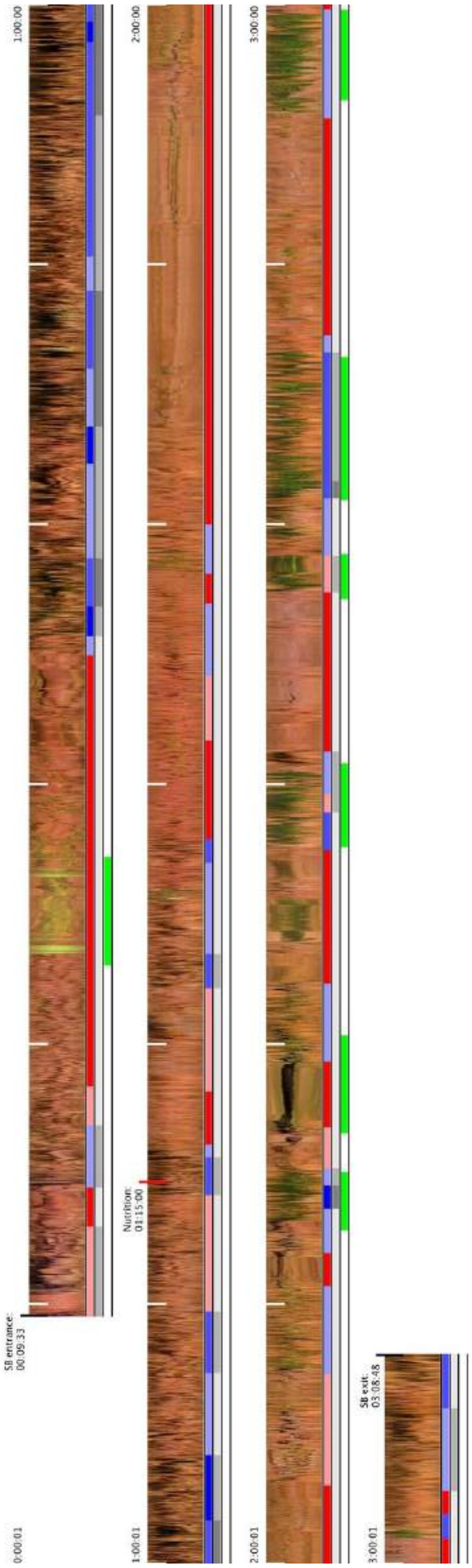


Figure B.9: Video 4. Motility descriptors.

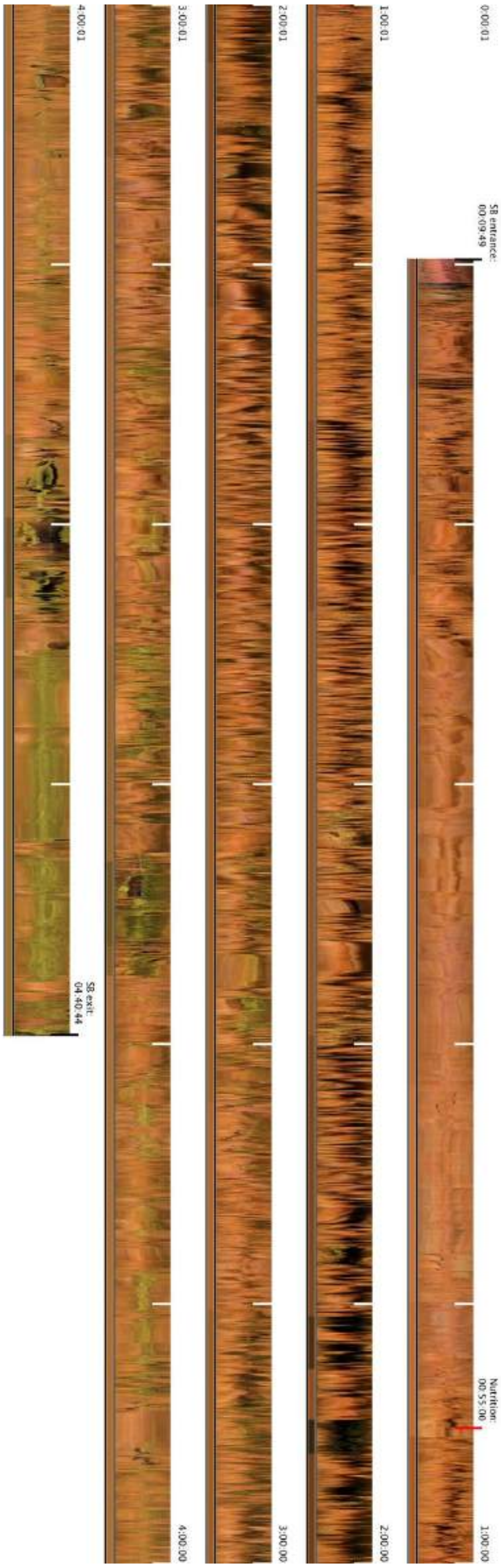


Figure B.10: Video 5. Mean color change.

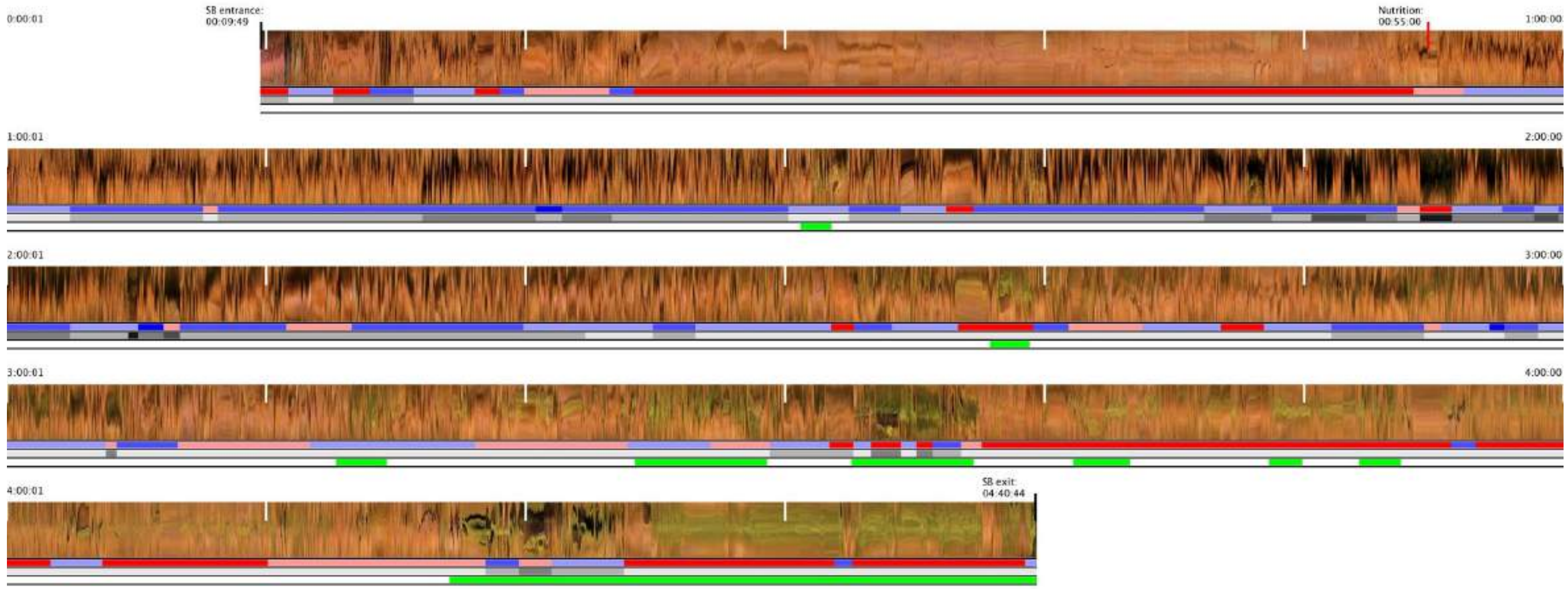


Figure B.11: Video 5. Motility descriptors.

Bibliography

[1] GIVEN IMAGING. http://www.givenimaging.com/, 2013. ix, 7, 27, 56

[2] SUSHIL K. SARNA. *Myoelectrocal and contractile activities of the gastrointestinal tract*. BC Decker Inc., 2002. 3, 4

[3] WILLIAM E. WHITEHEAD. **Gastrointestinal Motility Disorders of the Small Intestine, Large Intestine, Rectum, and Pelvic Floor**. *International Foundation for Functional Gastrointestinal Disorders*, **IFFGD Fact Sheet No. 162**, 2001. 4

[4] JUAN-R. MALAGELADA AND FERNANDO AZPIROZ. *Determinants of gastric emptying and transit in the small intestine*. John Wiley & Sons, Inc., 2010. 4

[5] K.E. BARRETT, L.R. JOHNSON, F.K. GHISHAN, J.L. MERCHANT, AND H.M. SAID. *Physiology of the Gastrointestinal Tract*. Number v. 2 in Physiology of the Gastrointestinal Tract. Elsevier Science, 2006. 5, 8

[6] FERNANDO VILARINO, PANAGIOTA SPYRIDONOS, FOSCA DEIORIO, JORDI VITRIA, FERNANDO AZPIROZ, AND PETIA RADEVA. **Intestinal motility assessment with video capsule endoscopy: automatic annotation of phasic intestinal contractions.** *IEEE Transactions on Medical Imaging*, **29**(2):246–59, 2010. 5, 17, 18

[7] HAI VU, TOMIO ECHIGO, RYUSUKE SAGAWA, KEIKO YAGI, MASATSUGU SHIBA, KAZUHIDE HIGUCHI, TETSUO ARAKAWA, AND YASUSHI YAGI. **Detection of contractions in adaptive transit time of the small bowel from wireless capsule endoscopy videos.** *Comp. in Bio. and Med.*, **39**(1):16–26, 2009. 5, 18

[8] MEDSCAPE. http://emedicine.medscape.com/article/179937-overview, 2011. 5

[9] CAROLINA MALAGELADA, FOSCA DE IORIO, FERNANDO AZPIROZ, ANNA ACCARINO, SANTI SEGUÍ, PETIA RADEVA, AND JUAN-R MALAGELADA. **New insight**

into intestinal motor function via noninvasive endoluminal image analysis. *Gastroenterology*, **135**(4):1155–62, 2008. 5, 11, 19, 56

[10] E. M. Quigley. **Gastric and Small Intestinal Motility in Health and Disease**. *Gastroenterology Clinics of North America*, **25**:113–145, 1996. 5

[11] Y.C. Metzger, S.N. Adler, A.B. Shitrit, B. Koslowsky, and I. Bjarnason. **Comparison of a new PillCam SB2 video capsule versus the standard PillCam SB for detection of small bowel disease**. *Reports in Medical Imaging*, **2**:7–11, 2009. 5

[12] Michal Mackiewicz. *Capsule Endoscopy - State of the Technology and Computer Vision Tools After the First Decade, New Techniques in Gastrointestinal Endoscopy*. Oliviu Pascu and Andrada Seicean (Ed.), 2011. 6

[13] M.K. Bashar, T. Kitasaka, Y. Suenaga, Y. Mekada, and K. Mori. **Automatic detection of informative frames from wireless capsule endoscopy images**. *Medical Image Analysis*, **14**(3):449 – 470, 2010. 8, 17, 52, 57

[14] Fernando Vilariño, Panagiota Spyridonos, Oriol Pujol, Jordi Vitrià, and Petia Radeva. **Automatic Detection of Intestinal Juices in Wireless Capsule Video Endoscopy**. In *18th Inter. Conf. on Pattern Recognition (ICPR)*, pages 20–24, 2006. 8, 17, 57

[15] A. Moglia, A. Menciassi, and Paolo. Dario. **Recent Patents on Wireless Capsule Endoscopy**. *Recent Patents on Biomedical Engineering*, **1**:24–33, 2008. 16

[16] Michael Liedlgruber and Andreas Uhl. **Computer-aided Decision Support Systems for Endoscopy in the Gastrointestinal Tract: A Review**. *IEEE Reviews in Biomedical Engineering*, **4**:73–88, 2012. 16, 18

[17] Ojala T., Pietikäinen M., and Mäenpää T. **Multiresolution gray-scale and rotation invariant texture classification with Local Binary Patterns.** 2002. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(7):971 - 987. 16

[18] Navneet Dalal and Bill Triggs. **Histograms of Oriented Gradients for Human Detection**. In Cordelia Schmid, Stefano Soatto, and Carlo Tomasi, editors, *International Conference on Computer Vision & Pattern Recognition*, **2**, pages 886–893, June 2005. 16, 66

[19] David G. Lowe. **Distinctive Image Features from Scale-Invariant Keypoints**. *Int. J. Comput. Vision*, **60**(2):91–110, November 2004. 16

[20] RICHARD SZELISKI. *Computer Vision: Algorithms and Applications*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010. 16

[21] CHRISTOPHER J. C. BURGES. **A Tutorial on Support Vector Machines for Pattern Recognition**. *Data Min. Knowl. Discov.*, **2**(2):121–167, June 1998. 16

[22] CHRISTOPHER M. BISHOP. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. 16

[23] BAOPU LI AND MAX Q. H. MENG. **Computer-based detection of bleeding and ulcer in wireless capsule endoscopy images by chromaticity moments**. *Comput. Biol. Med.*, **39**(2):141–147, February 2009. 16

[24] MICHAL W. MACKIEWICZ, MARK FISHER, AND CRAWFORD JAMIESON. **Bleeding detection in wireless capsule endoscopy using adaptive color histogram model and support vector classification**. In *Proc. SPIE*, **6914**, pages 69140R–69140R–12, 2008. 16

[25] Y. SUB JUNG, Y. HO KIM, D. HA LEE, AND J. HYO KIM. **Active Blood Detection in a High Resolution Capsule Endoscopy using Color Spectrum Transformation**. In *Proceedings of International Conference on BioMedical Engineering and Informatics*, pages 859–862, 2008. 17

[26] QIAN ZHAO AND M.Q.-H. MENG. **Polyp detection in wireless capsule endoscopy images using novel color texture features**. In *Intelligent Control and Automation (WCICA), 2011 9th World Congress on*, pages 948–952, 2011. 17

[27] A. KARARGYRIS AND N. BOURBAKIS. **Detection of Small Bowel Polyps and Ulcers in Wireless Capsule Endoscopy Videos**. *Biomedical Engineering, IEEE Transactions on*, **58**(10):2777–2786, 2011. 17

[28] ISABEL N. FIGUEIREDO, SURYA PRASATH, YEN-HSI R. TSAI, AND PEDRO N. FIGUEIREDO. **Automatic detection and segmentation of colonic polyps in wireless capsule images**. In *ICES REPORT 10-36, The Institute for Computational Engineering and Sciences, The University of Texas at Austin*, September 2010. 17

[29] BAOPU LI AND M.Q.-H. MENG. **Small bowel tumor detection for wireless capsule endoscopy images using textural features and support vector machine**. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 498–503, 2009. 17

[30] Lecheng Yu, P.C. Yuen, and Jianhuang Lai. **Ulcer detection in wireless capsule endoscopy images**. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 45–48, 2012. 17

[31] Edward J. Ciaccio, Christina A. Tennyson, Suzanne K. Lewis, Suneeta Krishnareddy, Govind Bhagat, and Peter H. R. Green. **Distinguishing patients with celiac disease by quantitative analysis of videocapsule endoscopy images**. *Comput. Methods Prog. Biomed.*, **100**:39–48, October 2010. 17

[32] Rajesh Kumar, Qian Zhao, S. Seshamani, G. Mullin, G. Hager, and T. Dassopoulos. **Assessment of Crohns Disease Lesions in Wireless Capsule Endoscopy Images**. *Biomedical Engineering, IEEE Transactions on*, **59**(2):355–362, 2012. 17

[33] L. Igual, J. Vitrià, F. Vilariño, S. Seguí, C. Malagelada, F. Azpiroz, and P. Radeva. **Automatic Discrimination of Duodenum in Wireless Capsule Video Endoscopy**. **22**, pages 1536–1539, 2008. 17

[34] J. Pa Sa Cunha, M. Coimbra, P. Campos, and J. Ma Soares. **Automated Topographic Segmentation and Transit Time Estimation in Endoscopic Capsule Exams**. *IEEE Transactions on Medical Imaging*, **27**(1):19–27, 2008. 17

[35] JeongKyu Lee, Jung-Hwan Oh, Subodh Kumar Shah, Xiaohui Yuan, and Shou Jiang Tang. **Automatic Classification of Digestive Organs in Wireless Capsule Endoscopy Videos**. In *Proceedings of the 2007 ACM symposium on Applied computing*, pages 1041–1045, 2007. 17

[36] Hai Vu, Ryusuke Sagawa, Yasushi Yagi, Tomio Echigo, Masatsugu Shiba, Kazuhide Higuchi, Tetsuo Arakawa, and Keiko Yagi. **Evaluating the control of the adaptive display rate for video capsule endoscopy diagnosis**. In *Proceedings of the 2008 IEEE International Conference on Robotics and Biomimetics*, 2009. 17, 18

[37] Vu Hai, Tomio Echigo, Ryusuke Sagawa, Keiko Yagi, Masatsugu Shiba, Kazuhide Higuchi, Tetsuo Arakawa, and Yasushi Yagi. **Adaptive Control of Video Display for Diagnostic Assistance by Analysis of Capsule Endoscopic Images**. In *Proceedings of the 18th International Conference on Pattern Recognition - Volume 03*, ICPR '06, pages 980–983, 2006. 17, 18

[38] Y. Yagi, H. Vu, T. Echigo, R. Sagawa, K. Yagi, M. Shiba, K. Higuchi, and T. Arakawa. **A diagnosis support system for capsule endoscopy**. *Inflammopharmacology*, **5**(2):78–83, 2007. 17, 18

[39] Piotr M. Szczypinski, Ram D. Sriram, Parupudi V.J. Sriram, and D. Nageshwar Reddy. **A model of deformable rings for interpretation of wireless capsule endoscopic videos**. *Medical Image Analysis*, **13**(2):312 – 324, 2009. Includes Special Section on Functional Imaging and Modelling of the Heart. 17

[40] Hai Vu, Tomio Echigo, Ryusuke Sagawa, Keiko Yagi, Masatsugu Shiba, Kazuhide Higuchi, Tetsuo Arakawa, and Yasushi Yagi. **Contraction detection in small bowel from an image sequence of wireless capsule endoscopy**. In *Proceedings of the 10th international conference on Medical image computing and computer-assisted intervention - Volume Part I*, MICCAI'07, pages 775–783, Berlin, Heidelberg, 2007. Springer-Verlag. 18

[41] Fernando Vilariño, Panagiota Spyridonos, Jordi Vitrià, Carolina Malagelada, and Petia Radeva. **Linear Radial Patterns Characterization for Automatic Detection of Tonic Intestinal Contractions**. In *CIARP*, pages 178–187, 2006. 18

[42] Panagiota Spyridonos, Fernando Vilariño, Jordi Vitrià, Fernando Azpiroz, and Petia Radeva. **Anisotropic Feature Extraction from Endoluminal Images for Detection of Intestinal Contractions**. In *MICCAI (2)*, pages 161–168, 2006. 18

[43] Santi Seguí. *Contributions to the diagnosis of intestinal motility by automatic image analysis*. PhD thesis, Universitat de Barcelona, June 2011. 18

[44] Pedro Felzenszwalb and Ramin Zabih. **Dynamic Programming and Graph Algorithms in Computer Vision**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **33**:721–740, 2011. 25

[45] Paul Jaccard. **Étude comparative de la distribution florale dans une portion des Alpes et des Jura**. *Bulletin del la Société Vaudoise des Sciences Naturelles*, **37**:547–579, 1901. 30, 74

[46] Dorin Comaniciu, Peter Meer, and Senior Member. **Mean shift: A robust approach toward feature space analysis**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**:603–619, 2002. 46

[47] V. N. Vapnik. **An overview of statistical learning theory**. *Neural Networks, IEEE Transactions on*, pages 988–999, 1999. 52, 54

[48] S. LLOYD. **Least squares quantization in PCM**. *Information Theory, IEEE Transactions on*, **28**(2):129 – 137, March 1982. 52

[49] H. BAY, A. ESS, T. TUYTELAARS, AND L. VAN GOOL. **Speeded-up Robust Features (SURF)**. *Computer Vision and Image Understanding (CVIU)*, **110**(3):346–359, June 2008. 53

[50] S. MAJI, A.C. BERG, AND J. MALIK. **Classification using intersection kernel support vector machines is efficient**. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conf. on*, pages 1 –8, June 2008. 54

[51] X. REN AND J. MALIK. **Learning a classification model for segmentation**. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 10 –17 vol.1, 10 2003. 54

[52] JIANBO SHI AND J. MALIK. **Normalized cuts and image segmentation**. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **22**(8):888 –905, August 2000. 54

[53] T. C. BAILEY AND A. C. GATRELL. *Interactive Spatial Data Analysis*. Longman Scientific and Technical, London, 1995. 55

[54] ROBERT M HARALICK. **Ridges and valleys on digital images**. *Computer Vision, Graphics, and Image Processing*, **22**(1):28–38, 1983. 67

[55] LINTON C FREEMAN. **A set of measures of centrality based on betweenness**. *Sociometry*, pages 35–41, 1977. 69, 70

[56] GERT SABIDUSSI. **The centrality index of a graph**. *Psychometrika*, **31**(4):581–603, 1966. 70

[57] PER HAGE AND FRANK HARARY. **Eccentricity and centrality in networks**. *Social networks*, **17**(1):57–63, 1995. 70

[58] ALFONSO SHIMBEL. **Structural parameters of communication networks**. *The bulletin of mathematical biophysics*, **15**(4):501–507, 1953. 70

[59] ULRIK BRANDES. **A faster algorithm for betweenness centrality***. *Journal of Mathematical Sociology*, **25**(2):163–177, 2001. 71

[60] V VAPNIK. **The Nature of Statistical Learning Theory**. *Data mining and knowledge discovery*, pages 1–47, 6. 72

[61] Thorsten Joachims, Thomas Hofmann, Yisong Yue, and Chun-Nam Yu. **Predicting structured objects with support vector machines**. *Communications of the ACM*, **52**(11):97–104, 2009. 72

[62] Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, Yasemin Altun, and Yoram Singer. **Large margin methods for structured and interdependent output variables**. *Journal of Machine Learning Research*, **6**(2):1453, 2006. 72

[63] Koby Crammer and Yoram Singer. **On the algorithmic implementation of multiclass kernel-based vector machines**. *J. Mach. Learn. Res.*, **2**:265–292, March 2002. 73, 74

[64] Chih-Chung Chang and Chih-Jen Lin. **LIBSVM: A library for support vector machines**. *ACM Transactions on Intelligent Systems and Technology*, **2**:27:1–27:27, 2011. 76

[65] Albert Bifet and Ricard Gavalda. **Learning from time-changing data with adaptive windowing**. In *In SIAM International Conference on Data Mining*, 2007. 85, 86, 91, 138

[66] Ludmila I. Kuncheva. **Change detection in streaming multivariate data using likelihood detectors**. *IEEE Transactions on Knowledge and Data Engineering*, **25**, 2013. 85

[67] Cesare Alippi, Giacomo Boracchi, and Manuel Roveri. **Change detection tests using the ICI rule.** *Neural Networks (IJCNN), The 2010 International Joint Conference on*, 2010. 85

[68] Anton Dries and Ulrich Rückert. **Adaptive concept drift detection**. *Stat. Anal. Data Min.*, **2**(5-6):311–327, December 2009. 85

[69] Pascal Massart. **Some applications of concentration inequalities to statistics**. *Annales-Faculte des Sciences Toulouse Mathematiques*, **9**(2), 2000. 85

[70] Wassily Hoeffding. **Probability Inequalities for Sums of Bounded Random Variables**. *Journal of the American Statistical Association*, **58**(301):13–30, March 1963. 85, 86

[71] Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári. **Tuning Bandit Algorithms in Stochastic Environments**. In *Proceedings of the 18th international con-*

ference on Algorithmic Learning Theory, ALT '07, pages 150–165, Berlin, Heidelberg, 2007. Springer-Verlag. 85

[72] JOÃO GAMA. **A survey on learning from data streams: current and future trends**. *Progress in AI*, **1**(1):45–55, 2012. 85

[73] MAYUR DATAR, ARISTIDES GIONIS, PIOTR INDYK, AND RAJEEV MOTWANI. **Maintaining Stream Statistics over Sliding Windows**. *SIAM J. Comput.*, **31**(6):1794–1813, June 2002. 85

[74] MICHAL DROZDZAL, SANTI SEGUI, CAROLINA MALAGELADA, FERNANDO AZPIROZ, JORDI VITRIA, AND PETIA RADEVA. **Interactive labeling of WCE image**. In *Iberian Conference on Pattern Recognition and Image Analysis*, IbPRIA '11, pages 143–150, 2011. 104

[75] SANJOY DASGUPTA. **Two faces of active learning**. *Theoretical Computer Science*, **In Press, Corrected Proof**, 2010. 107

[76] MARIA-FLORINA BALCAN, ALINA BEYGELZIMER, AND JOHN LANGFORD. **Agnostic active learning**. *J. Comput. Syst. Sci.*, **75**(1):78–89, 2009. 107

[77] DAVID A. COHN, LES E. ATLAS, AND RICHARD E. LADNER. **Improving Generalization with Active Learning**. *Machine Learning*, **15**(2):201–221, 1994. 107

[78] Y. FREUND, H. SEBASTIAN SEUNG, ELI SHAMIR, AND NAFTALI TISHBY. **Selective Sampling Using the Query by Committee Algorithm**. *Machine Learning*, **28**(2-3):133–168, 1997. 107

[79] ANDREW MCCALLUM AND KAMAL NIGAM. **Employing EM and Pool-Based Active Learning for Text Classification**. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 350–358, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. 107

[80] HIEU T. NGUYEN AND ARNOLD SMEULDERS. **Active learning using pre-clustering**. In *Proceedings of the twenty-first international conference on Machine learning*, ICML '04, pages 623–630, New York, NY, USA, 2004. ACM. 107

[81] SIMON TONG AND DAPHNE KOLLER. **Support vector machine active learning with applications to text classification**. *J. Mach. Learn. Res.*, **2**:45–66, March 2002. 107

[82] ZHAO XU, KAI YU, VOLKER TRESP, XIAOWEI XU, AND JIZHI WANG. **Representative sampling for text classification using support vector machines**. In *Proceedings of*

the 25th European conference on IR research, ECIR'03, pages 393–407, Berlin, Heidelberg, 2003. Springer-Verlag. 107

[83] Kenneth Dwyer and Robert Holte. **Decision Tree Instability and Active Learning**. In *Proceedings of the 18th European conference on Machine Learning*, ECML '07, pages 128–139, Berlin, Heidelberg, 2007. Springer-Verlag. 107

[84] Ludmila I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004. 108

[85] Yoav Freund, H. Sebastian Seung, Eli Shamir, and Naftali Tishby. **Selective Sampling Using the Query by Committee Algorithm**. *Mach. Learn.*, **28**:133–168, September 1997. 108

[86] Pinar Donmez, Jaime G. Carbonell, and Paul N. Bennett. **Dual Strategy Active Learning**. In *Proceedings of the 18th European conference on Machine Learning*, ECML '07, pages 116–127, Berlin, Heidelberg, 2007. Springer-Verlag. 108

[87] Hinrich Schütze, Emre Velipasaoglu, and Jan O. Pedersen. **Performance thresholding in practical text classification**. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, CIKM '06, pages 662–671, New York, NY, USA, 2006. ACM. 108, 115

[88] S. Dasgupta and D. Hsu. **Hierarchical sampling for active learning**. In *Proceedings of the 25th international conference on Machine learning*, 2008. 108, 109, 117, 119

[89] T. Hospedales, S. Gong, and T. Xiang. **Finding Rare Classes: Active Learning with Generative and Discriminative Models**. *Knowledge and Data Engineering, IEEE Transactions on*, **PP**(99):1, 2011. 108, 116

[90] Xiaojin Zhu, John Lafferty, and Zoubin Ghahramani. **Combining Active Learning and Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions**. In *ICML 2003 workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, pages 58–65, 2003. 108

[91] Joe H. Ward, Jr. **Hierarchical Grouping to Optimize an Objective Function**. *Journal of the American Statistical Association*, **58**(301):236–244, 1963. 109

[92] Aristides Gionis, Piotr Indyk, and Rajeev Motwani. **Similarity Search in High Dimensions via Hashing**. In *Proc. of the 25th ICVLDB*, VLDB '99, pages 518–529, 1999. 112

[93] FRANCESCO ORABONA, JOSEPH KESHET, AND BARBARA CAPUTO. **The projectron: a bounded kernel-based Perceptron**. In *Proc. of the 25th ICML*, ICML '08, pages 720–727, 2008. 113

[94] ALBERT BIFET. *Adaptive learning and mining for data streams and frequent patterns*. PhD thesis, Universitat Politecnica de Catalunya, April 2009. 138