Detection of Circular Content Area in Endoscopic Videos for Efficient Encoding and Improved Content Analysis

Bernd Münzer

bernd@itec.aau.at

Klaus Schöffmann

ks@itec.uni-klu.ac.at

Laszlo Böszörmenyi

laszlo@itec.aau.at

Institute of Information Technology
University Klagenfurt
Technical Report No TR/ITEC/12/2.03
November 2012

Abstract

The actual content of endoscopic videos is typically limited to a circular area in the center of the image due to the inherent characteristics of the camera. This area is surrounded by a dark border that fills up the remainder of the rectangular image and is subject to noise. The position and size of the circle is not standardized and usually varies over time. In this paper a robust algorithm is presented that (1) classifies which parts of an endoscopic video feature a circular content area and (2) determines its exact position and size, if present. This information is useful for improving video encoding efficiency, limiting further analysis steps to the relevant area and saving ink when printing still images on findings. Our evaluation shows that the proposed method is very fast, reliable and robust. Moreover, it indicates that by exploiting this information for video encoding a considerable bitrate reduction is possible with the same visual quality.

Keywords: Circle Detection, Endoscopy, Video Analysis, Video Encoding

1 Introduction

Endoscopy is a modern, minimally invasive technique for screenings and operations in various regions of the human body, including colon, abdomen and joints. A small camera is inserted through a natural or artificial orifice and produces a video stream that is shown to the endoscopist on a screen. In recent years, it became common to record this video stream for documentation purposes and for retrospective analysis.

A typical characteristic of such endoscopic videos is the fact that the actual content is limited to a circular area in the center of the image. The pixels outside this circle are black or at least rather dark to a certain degree and contain no useful information at all, as illustrated in Figure 1(a). The reason for this is as follows: Every camera lens basically projects light on a circular area that is called image circle. Usually, cameras have an image sensor that is smaller than the image circle and hence only consider a rectangular subarea of it. For widescreen aspect ratio (16:9) that means that about 50% of the gathered light is not captured by the sensor. But in the case of endoscopes it is more desirable to capture the full image circle without losing information. Therefore the image sensor is designed to be larger than the image circle. This has the drawback of unused border regions.

The problem is that if content-based image analysis (e.g. the calculation of histograms) is to be carried out on the image to detect anomalies or operation instruments these irrelevant pixels can severely impair the result. Moreover, the area outside the image circle does not form a perfectly homogeneous black area but is subject to intensive noise. Thus, if the video is recorded, a certain amount of bandwidth has to be wasted to encode it. Therefore, we propose a new robust algorithm to exactly determine the parameters of the featured circle (centre coordinates and radius). We can use this information to concentrate further content based analysis on the relevant pixels inside the circle. Moreover, we can overlay the outer pixels with a pure black mask that can be encoded more efficiently. This mask could also have any other color. For example, it could make sense to use a white overlay mask for images in printed findings.

In this context another domain specific aspect has to be considered. Some endoscopes offer a zoom function that can be used by the endoscopist to magnify the video image, causing the circle to grow out of the image. Therefore, our circle detection algorithm does not only have to detect the position and size of the circle, the even more critical issue is to detect if there is a circle at all. If the image is zoomed, meaning that the relevant area covers the whole image, it must be guaranteed that no false circle is detected because this would lead to a severe information loss when overlaying the outside area with a black mask or ignoring it for analysis. On the other side, it is less fatal if an actually existing but ambiguous circle is not detected.

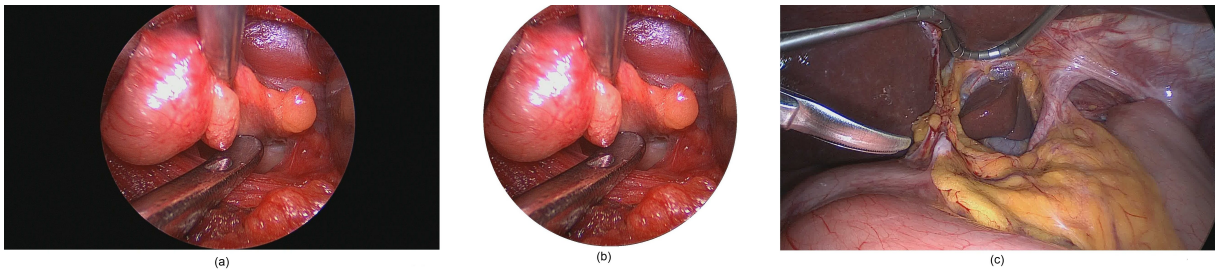

Figure 1: (a) Original frame with a black border, (b) the same frame with a white border overlay, (c) a zoomed frame

Figure 1(a) illustrates how a frame with a circular content area typically looks like. As can be seen, about half of the pixels belong to the dark border. This ratio is especially high for videos with a widescreen aspect ratio (16:9) which is the common format for HD (High Definition) videos. Figure 1(b) demonstrates the appearance of this frame with an overlaid white mask that could be used for printed findings to save ink. Figure 1(c) shows an example of a zoomed frame.

The benefits of performing a circle detection can be summarized as follows:

1. **Basis for subsequent analysis:** Further content-based analysis methods (e.g. the calculation of histograms) only have to operate on the actually relevant area of the image. This reduces processing time and increases accuracy.
2. **Video encoding efficiency:** The noisy data outside the detected circle can be replaced by a homogeneous black mask which can be encoded more efficiently, especially with skipped macroblocks as featured by H.264/AVC.
3. **Printing ink savings:** Single still images of endoscopic procedures are often extracted and used for printed findings. Since the black area outside the circle covers a considerably portion of such images (for widescreen images about 50% of all pixels), a lot of printing ink can be saved if these areas are overlaid with a white mask.
4. **More focused visualization:** Video summarization or browsing interfaces can be designed in an optimized way that minimizes wasted screen space by only showing the relevant content area.

The remainder of the paper is structured as follows: Chapter 2 gives a short overview of related work. Chapter 3 describes the proposed algorithm in detail. Chapter 4 presents an evaluation of the proposed method. Chapter 5 evaluates the impact of using the proposed method for improving coding efficiency. Chapter 6 concludes the paper and gives an outlook to future work.

2 Related work

Current research in the field of endoscopic image/video analysis is mainly focused on the classification of polyps, lesions and tumors in the context of CAD (computer aided diagnosis) for colonoscopy [9] and real-time analysis of laparoscopic video streams during operations for robotic endoscope and instrument guidance and augmented reality [4][17]. Two key techniques in this field are 3D reconstruction [6][15] and instrument recognition and tracking [11][19]. Further research topics that are frequently addressed are image enhancement and pre-processing (e.g. distortion correction [5] and specular reflection removal [2][18]) and assessment of the quality of colonoscopic examinations [7][13].

Many of these publications presumably could benefit from an accurate detection of the circular content area to narrow down analysis to the actually relevant parts of an image. The most common approach is to simply analyse the whole image, although especially global image features get biased due to the irrelevant border pixels and analysis algorithms have to work on a larger area than necessary. Some authors ignore all dark pixels below a certain threshold (e.g. [16]). The disadvantage of this straightforward approach is that it also ignores dark pixels that occur inside the circle and should be taken into account for the analysis, e.g. parts of operation instruments. Moreover, it does not consider the varying border brightness and heavy border noise. Another idea is to crop the image at empirical boundaries so that only the inner square of the circle is used and the dark border is eliminated [1]. This obviously has the drawback that useful information is disregarded. A further approach uses a predefined static mask for the content area [20], but this does not take into account the unsteady nature of the circle in terms of position and size and the possibility of a zoom. To the best of our knowledge, the idea of exactly determining the parameters of the circle has not been addressed yet.

A possible alternative to our approach could be the well-established generalized Hough transform [8] or one of its many derivatives. This technique is even able to find multiple circles on arbitrary positions in the image, but that is not needed in our scenario. Besides, the hough transform is not very efficient in terms of processing speed due to its brute-force nature. Our experiments also revealed that its accuracy is not very high and the circle parameters are only estimated approximately. Therefore we present a fast and robust algorithm that exploits the domain specific knowledge that an image features no or exactly one circle with a roughly known size approximately in the center of the image.

3 Circle detection algorithm

Our proposed algorithm for circle detection consists of a number of steps that are performed for each frame of the video. It is not sufficient to detect the circle for one frame and use the result for the whole video because the circle size and position are rather unsteady and can change at any time because of strong camera motion (especially rotation) and/or zoom (further technical details of endoscopes and endoscopic videos can be found in [10]).

The basic idea is to calculate the circle from three edge points on the edge image. But in order to be robust against outliers, a larger number of edge points is determined. In a filtering step outliers are detected and discarded. If the respective frame is a zoomed frame the distribution of the edge points does not conform to the typical pattern featured in frames with a circle but is rather random. This fact is exploited to detect the zoomed frames.

It is very important to prevent false positives, i.e. a detected circle in a frame that actually does not contain a circle. Therefore, the algorithm is designed and parameterized in a rather defensive way that rather risks missing some ambiguous circles for the sake of preventing false positives. Each step has a built-in plausibility check to detect frames without a circle and immediately terminate the detection process by classifying the frame as zoomed frame. A frame is only classified as circle frame if all the phases are passed. The sequence of steps is illustrated in Figure 2. The individual steps are described in detail below.

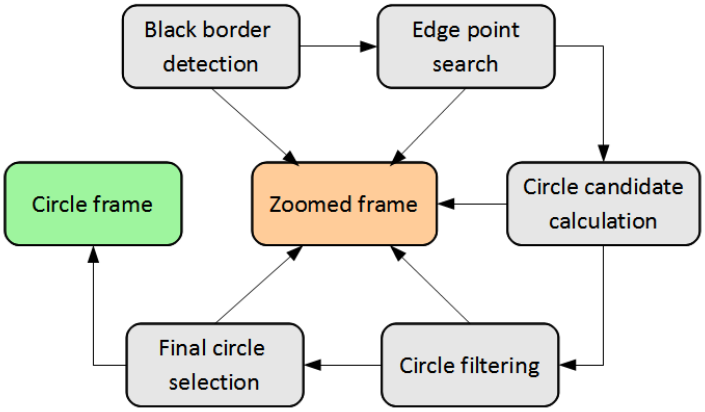

Figure 2: Phases of the circle detection algorithm

3.1 Black border detection

In the first place, we heuristically try to find out if the frame is likely to have a black border at all. If this is not the case we skip the subsequent phases and immediately classify it as zoomed frame. This substantially reduces the probability of detecting a false circle whilst improving performance in terms of processing speed as a side effect.

A square sample is taken from the image center and compared to two sample stripes from the left and right boundary in terms of average intensity and variance. In a frame with a black border, the intensity and variance of the outer samples are rather low and in any case lower than the center samples'. If this constraint is violated, i.e. the intensity or variance of an outer sample is higher than a threshold or the center sample, the frame definitely has no black border and is classified accordingly. Otherwise, the intensity and variance values are aggregated to a value between 0 and 1 which indicates the probability of the frame to have a black border. If it is higher than a threshold, the actual circle detection is performed by proceeding with the subsequent steps.

This phase does not guarantee that every zoomed frame is detected straightaway. For example, it is possible that a zoomed frame has a rather dark and homogenous periphery which is misinterpreted as black border. Moreover, the thresholds have to be rather tolerant because endoscopic videos from different sources may have different overall intensity and contrast and a different amount of noise. For that reason, the further analysis steps can not assume that every frame that passes the black border detection does contain a circle. Occasional misclassifications have to be corrected by further mechanisms in the following phases.

3.2 Edge point search

The actual circle detection is performed only if the previous phase indicates that the frame has a black border. At first, the Canny edge detector [3] is used to find the contour of the circle. The parameters of the Canny edge detector are set empirically according to experiments. The idea is to calculate a circle from three arbitrary points on this contour.

However, the resulting edge image is not perfectly reliable because it can have gaps due to dark areas at the periphery of the circle in the original image. Furthermore, heavy noise can cause edges to be detected outside the circle. Thus, in order to increase the robustness of the algorithm a number of circle candidates is calculated.

We determine n seed lines which are evenly distributed over the height of the edge image. For each of these lines the first pixel from the left and from the right corresponding to an edge in the edge image is considered as edge point. So we obtain up to $2 * n$ edge points. The number may be smaller in case there is no edge on one or more lines of the edge image. If the ratio of found edge points to the maximal possible number of edge points is below a threshold, we conclude that this frame is a zoomed frame.

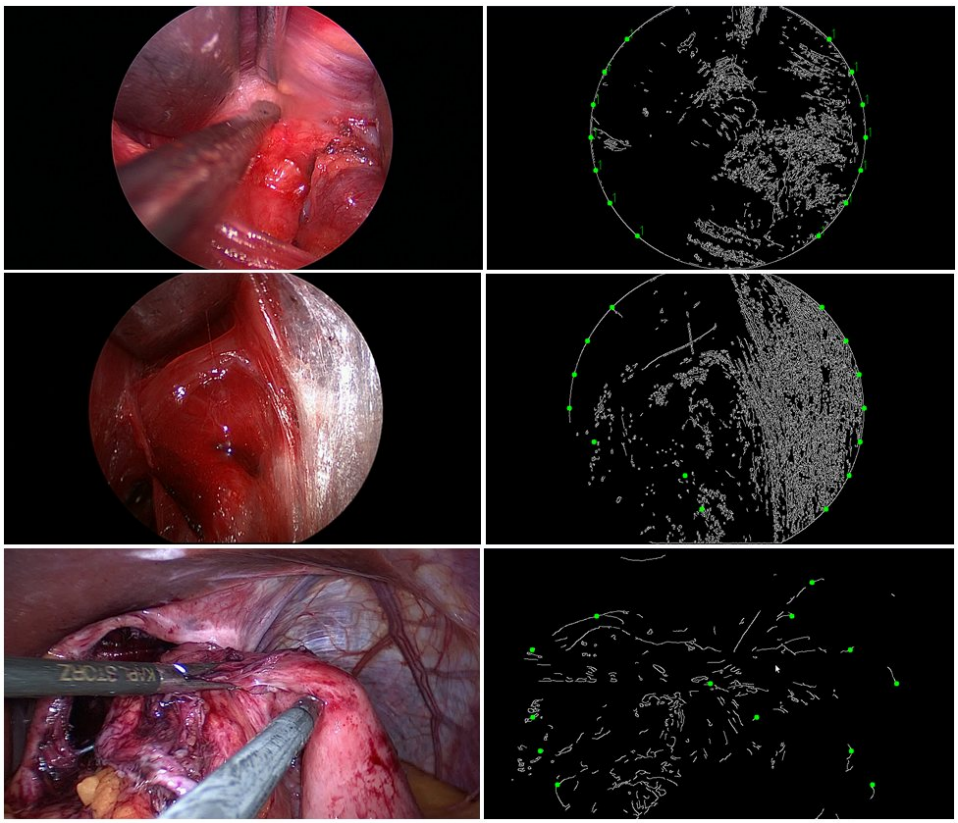

Figure 3: Examples for edge point search

In an optimal case, all of the edge points lie on the contour of the circle but this can not be taken for granted. Outlier points may lie on any arbitrary position and thus would produce a totally wrong circle which makes it necessary to identify and discard them. What is more, if the analyzed frame actually is a zoomed image that slipped through the black border detection, the points occur at random positions and produce arbitrary circles.

Figure 3 illustrates the edge point search with the parameter n set to 7, i.e. we consider 7 lines of the edge image. The first image shows an optimal case where the edge image features a perfect circle. All edge points lie on this circle and all possible circles that can be calculated from these points would be correct. The second image represents a harder example with three outliers due to gaps in the edge image. These gaps result from a rather dark area at the lower left of the circle which is in strong contrast to a very bright area on the right. The third image depicts a zoomed frame. In this case the edge points are distributed randomly over the image. In fact, this frame would already have been classified as zoomed frame by the black border detection. Nevertheless, we cannot rely on the latter and have to ensure that the edge point based circle calculation fails for such frames.

3.3 Circle candidate calculation

The next step is to calculate a number of circle candidates. Each possible combination of three edge points yields one circle candidate. The calculation uses basic geometry: the three edge points A, B and C always form a triangle which is encompassed by the sought circle. For two sides of this triangle we determine the perpendicular bisector, i.e., the orthogonal line that crosses the side at its midpoint. The intersection of the perpendicular bisectors corresponds to the circle centre. The radius is equal to the distance of the centre to any of the three edge points. This process is illustrated in Figure 4.

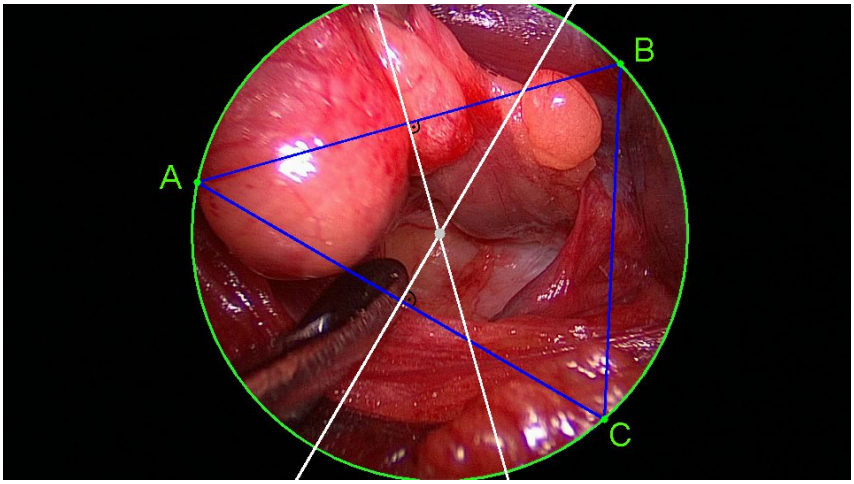

Figure 4: Calculation of a circle candidate from three edge points

3.4 Edge point and circle filtering

Each of the circle candidates is checked for plausibility. In order to be regarded as plausible the distance between the circle centre and the centre of the image must be smaller than a threshold. However, this threshold must not be too restrictive because the actual position of the circle in the image is not standardized and only roughly at the centre. As a further criterion, the radius has to be greater than $\frac{h}{2} * p_1$ and less than $\frac{w}{2} * p_2$, where h is the height of the image, w is the width of the image and p_1 and p_2 are parameters which specify the allowed size of the circle. If a circle is plausible, it is retained in the list of circle candidates, otherwise it is discarded.

Figure 5 depicts three different circle candidates for the second frame of Figure 3. a) and b) represent unplausible candidates based on two correct edge points and one outlier. c) is a plausible candidate formed by three correct edge points. d) illustrates a circle candidate of the zoomed frame which is plausible by chance in this special example. Nevertheless, the number of unplausible circles obviously is much higher so we finally can conclude that this candidate is not correct.

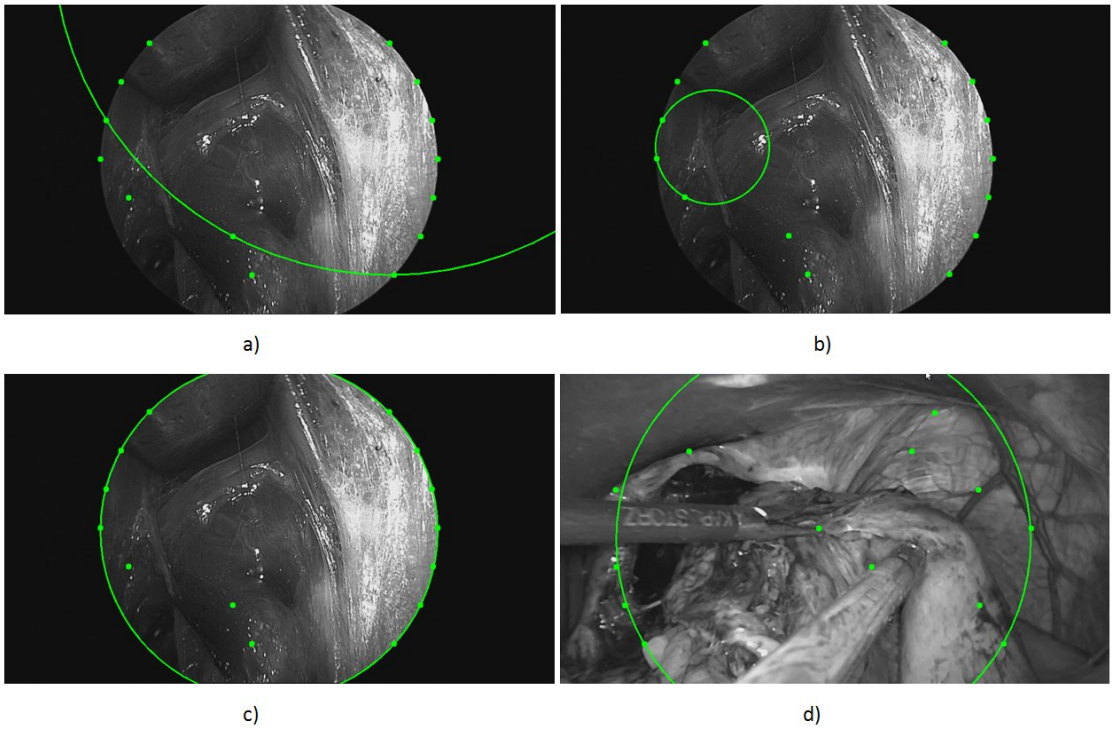

Figure 5: Examples for circle candidates

Edge points are filtered as follows: For each edge point e we calculate a confidence value $c(e) = cp(e)/cc(e)$ where $cp(e)$ is the number of plausible circle candidates based on e and $cc(e)$ is the total number of circle candidates based on e. If $c(e)$ is considerably smaller than the median confidence of all edge points or smaller than an absolute threshold, e is regarded as invalid. All circle candidates that are based on one or more invalid edge points are also discarded.

In case of a zoomed frame, the random distribution of edge points leads to a high number of unplausible circle candidates and further to very low confidence values for the edge points rendering them invalid. If the ratio of remaining plausible circle candidates to the original total number of circle candidates is below another threshold, the frame is classified as zoomed frame.

Figure 6 depicts the results of the filtering step for the second and third example frame from Figure 3. For the first example frame no edge points have to be filtered because all of them are on the circle and therefore have a plausibiltiy of 1. In the figure, red points represent invalidated edge points while green points represent valid edge points. The number beside the points denotes the respective plausibility.

3.5 Final circle selection

If there are remaining circle candidates that passed the filtering step, this final phase is supposed to select the best of these candidates. Each is compared to the edge image and the candidate with the highest match is selected as final circle, provided that the match is above a threshold. Otherwise, the frame is classified as zoomed frame.

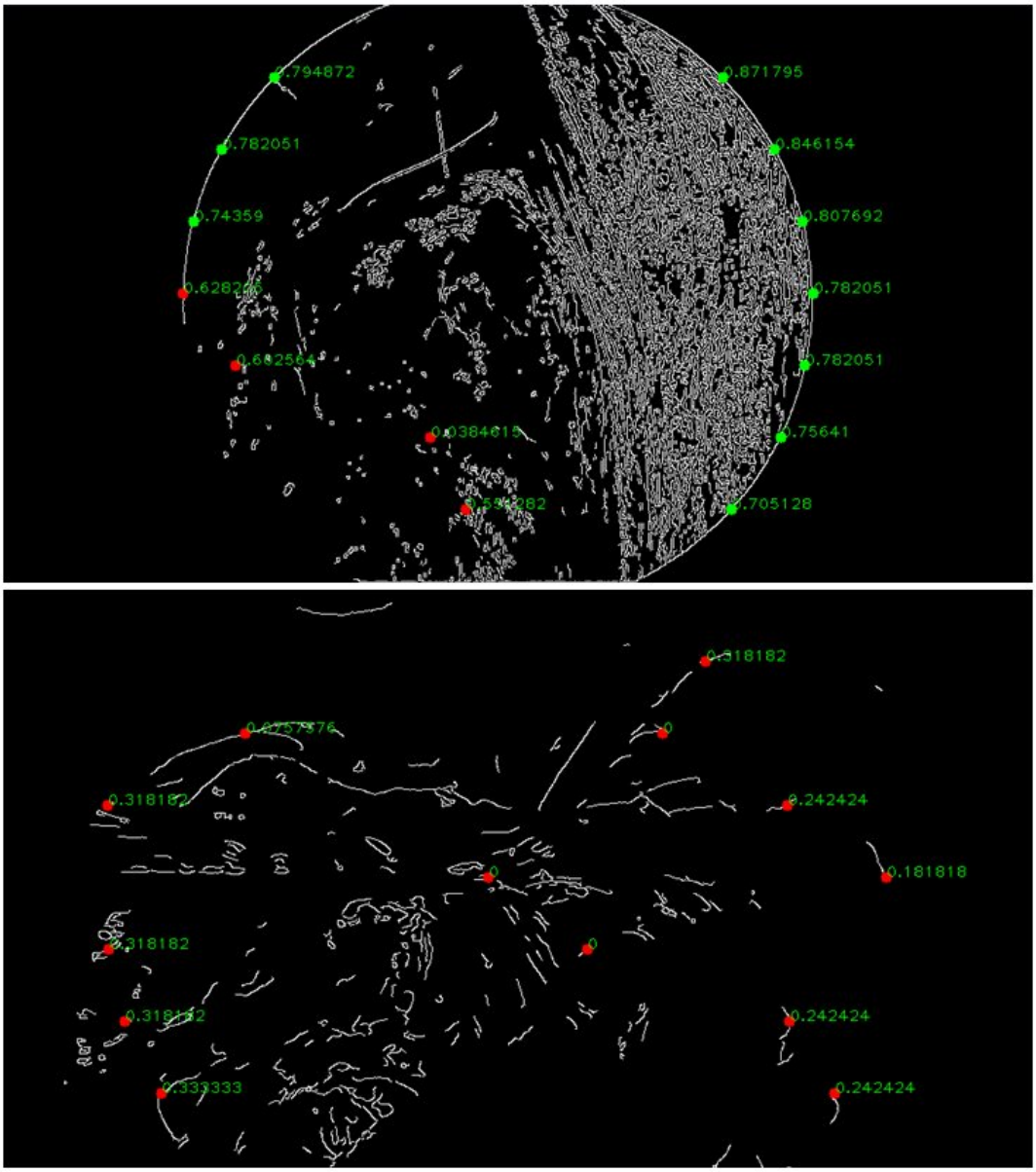

Figure 6: Filtered edge points

3.6 Outlier correction

After all individual frames have been analyzed the sequence of frame classifications and detected circles is examined. In the unlikely case that some single frames have been classified wrongly, they are corrected by a temporal filtering. Each sequence of equally classified frames (circle or zoomed frame) is checked to have a minimum length (e.g., 100 frames). If there is a shorter sequence, the corresponding frame classifications are modified accordingly. For example, if there are 10 frames with a detected circle amidst 100 zoomed frames, these circles are ignored. Analogously, a minority of zoomed frames amidst a majority of circle frames get classified as circle frames. Their circle parameters are computed as an average of the neighbor frames' circles. Additionally, each circle is compared to the circles of preceding and subsequent frames to find and smooth outliers in terms of position and size.

4 Evaluation

For evaluation, we used our proposed algorithm to analyze 65 distinct High Definition (HD) video files of laparoscopic surgery. This testset has a total length of about five hours, which corresponds to 439,608 individual frames. 62% of these frames contain a circle while the remaining 38% represent zoomed images. The ground truth (i.e., the classification between circle or zoomed image and in the former case the exact position and radius of the circle) has been annotated manually.

4.1 Accuracy of the circle detection

The evaluation shows very good results in terms of classification accuracy. Our proposed algorithm achieves a precision of 99.9992%, a sensitivity of 97.7%, a specificity of 99.9988% and an accuracy of 98.57%. For the calculation of these metrics circle frames are considered as positives, zoomed frames as negatives.The detailled results are summarized in Table 1.

		Classification	
Ground truth		Circle	Zoomed
Circle	271161	264863	6298
Zoomed	168447	2	168445

Table 1: Circle detection accuracy

As can be seen in the table, exactly 2 frames yielded a false positive while all the other frames that were classified as circle frame indeed have a circle. A closer investigation revealed that both frames belong to a transition scene where a zoom is performed and the ground truth annotation has a slightly different scene boundary. So when omitting these definitely non critical false positives, precision and specificity reach 100%.

About 2% of the circle frames (6,298 frames) were missed, i.e., no circle was detected although it exists. The closer investigation revealed that in all such cases this was due to very dark scenes of poor quality showing the environment outside the patient where the circular shape is hardly visible even for the human eye. So if we only take into account the relevant scenes inside the patient, also the sensitivity and consequently also the accuracy is 100%. Filtering out-of patient frames is subject of further study and out of scope of this paper.

Additionally, the parameters of the detected circles were compared to the circles annotated in the ground truth. This comparison shows that the average deviation is 0.16% for the x coordinate, 0.44% for the y coordinate and 0.25% for the radius. For a full HD video (1920x1080) this corresponds to an average error of 1-2 pixels which is neglectable. Anyway the accuracy of this result is rather vague because of ground truth limitations regarding the precisely accurate annotation of each single frame. As the actual circle position is not perfectly static but slightly moving it would be necessary to manually annotate every single frame which is not feasible. However, a more important aspect is the fact that not a single detected circle had a radius with a deviation of more than 2% from the ground truth so we can state that our algorithm is also highly accurate in terms of circle parameters.

4.2 Runtime

For evaluating the performance of the algorithm in terms of runtime we executed it on an off-the-shelf Intel Xeon 2.4 GHz processor and measured the processing time for the individual phases. Our implementation uses the OpenCV library for standard operations like edge detection and image filtering. We did not take into account the time for decoding the video but only the net processing time for a frame in grayscale color space. As the circle detection is meant to be a preprocessing step for further analysis algorithms decoding will be necessary anyway. What is more, intermediate results like the edge image may also be

required by these algorithms and can be reused. We logged the processing times for each frame of the 65 videos of our testset and calculated the average for each phase. The results are summarized in Table 2. The processing time of the final outlier correction phase is neglected because it is not executed for every frame but only once per video and takes less than 10 ms, depending on the length of the video.

	Default configuration		No scaling		No PCC	No BBD
Processing step	**Circle**	**Zoomed**	**Circle**	**Zoomed**	**Circle**	**Zoomed**
Image scaling	1,680 ms	1,209 ms	0 ms	0 ms	1,700 ms	1,145 ms
Black border detection (BBD)	1,810 ms	1,287 ms	10,513 ms	19,561 ms	1,824 ms	0 ms
Edge detection	3,823 ms	1,762 ms	36,386 ms	24,992 ms	3,838 ms	6,244 ms
Preceding circle check (PCC)	0,018 ms	0,005 ms	0,034 ms	0,005 ms	0 ms	0,017 ms
Edge point search	0,001 ms	0,006 ms	0,007 ms	0,010 ms	0,017 ms	0,020 ms
Circle candidate calculation	0,003 ms	0,008 ms	0,019 ms	0,011 ms	0,071 ms	0,028 ms
Circle filtering	0,001 ms	0,000 ms	0,007 ms	0,000 ms	0,026 ms	0,000 ms
Final circle selection	0,05 ms	0,000 ms	0,708 ms	0,000 ms	1,265 ms	0,001 ms
Total	**7,39 ms**	**4,28 ms**	**47,67 ms**	**44,58 ms**	**8,74 ms**	**7,45 ms**
	135 fps	**234 fps**	**21 fps**	**22 fps**	**114 fps**	**134 fps**

Table 2: Runtime

It turned out that the greatest part of the processing time is consumed by the preprocessing steps, especially the edge detection. The phase with the second longest processing time is the black border detection. This is mainly due to a blur filter that is applied in this phase. Compared to these preprocessing steps the runtime of the actual detection algorithm is almost negligible. In our default configuration, we additionally scale down every frame and operate on this version. This way, we can process about 135 frames per second (fps). In the case of zoomed frames the algorithm is even faster (234 fps). This is because most of the frames are already classified in the black border detection phase and the subsequent phases including the expensive edge detection are skipped. This leads to a shorter average processing time for these phases.

We also ran our code with minor modifications to show the effect of some key phases of the detection process. For example, if we do not scale down the frames but operate on full resolution frames, the analysis rate drops to about 20 fps which is a significant slowdown. Especially the edge detection is much slower (36 ms as opposed to 3,8 ms). Also the black border detection is very expensive for full frames because of the blur filter. The saved processing time more than compensates for the time needed for downscaling. Moreover, our experiments showed that the accuracy of the analysis result is not noticeably impaired by downscaling the frame for analysis purposes.

Another advantage of downscaling the original frame is that the runtime is roughly constant for different resolutions. However, we do not scale to one specific target resolution. Instead, we divide the dimensions by the greatest possible multiple of two so that the width does not fall below a minimum which is empirically defined as 480 pixels. That means that videos with a resolution of 1920x1080 are scaled to 480x270 and videos with a resolution of 1280x720 are scaled to 640x360. Our testset also contains SD (Standard Definition) videos with a resolution of 720x576 and a PAR (pixel aspect ratio) of 64:45. That means that each pixel has to be displayed as a rectangle instead of a square. So, to be applicable to our algorithm, we first have to convert these frames to the corresponding resolution with a PAR of 1. In this case this resolution is 1024x576. We then scale the frame to 512x288.

In the default configuration, the times of all the phases after the preceding circle check (PCC) are extremely low. This is because of the following optimization that has been implemented in our algorithm: If a circle is detected for frame n, we check if it also matches the edge image of frame $n + 1$. If this is the case we "reuse" the circle and proceed with the next frame instead of calculating edge points and circle candidates. Because the position and size of the circle often stay constant for a certain period, this optimization increases the fps by 18%. In Figure 2 the column "No PCC" shows the processing times if the preceding circle check is disabled, i.e. all phases are executed for each frame. It can be seen that

especially the final circle selection is rather time consuming because all remaining circle candidates are compared to the edge image to find the best one while with the PCC enabled in many cases only one comparison is necessary.

The last column of Table 2 represents another modification regarding the omission of the black border detection (BBD). For circle frames it is obvious that the average processing time is exactly reduced by the processing time of this phase which is about 24%. However, the average processing time for zoomed frames increases significantly (by 74%). This is mainly due to the expensive edge detection that has to be executed for each frame while with the BBD enabled it is skipped most of the time.

The runtime evaluation shows that the circle detection algorithm is very efficient and can also be applied in real time scenarios. For example it could be used at recording time to directly encode the border area outside the circle with a homogenous black mask to reduce the encoding bitrate. In post-processing applications it can be used as efficient pre-processing step, especially if the required operations (scaling, blur filtering, edge detection) have to be conducted anyway and can be reused. A potential for further performance optimization could be to only analyze every n^{th} frame and only take into account the intermediate frames if the parameters of the circle change.

4.3 Robustness

The algorithm must guarantee that no false positives occur because of possible information loss. So we performed an additional analysis run, without the black border detection phase, to verify that the edge point filtering is restrictive enough to detect zoomed frames. Additionally, we switched off the outlier correction phase to check if all zoomed frames are detected correctly without exception. This run yielded the same results in terms of false positives. The only difference was that the number of false negatives was slightly higher due to the missing outlier correction. So we can state that our proposed method is highly reliable in terms of classification accuracy even without the pre- and post-processing phase.

To further prove the robustness of the algorithm we tested it on the TRECVID 2012 general video data set [14] consisting of 8,263 video files with an overall length of 200 hours. In 208 of these videos one or more circles were found. In most cases this is due to short scenes showing various circular shapes (moon, planet, coin, clock, logo, telescope view etc.) with a black background and can therefore be regarded as correct. In 9 videos these correctly detected circle scenes where long enough to pass the outlier correction phase and were incorporated into the final result. In only 21 videos very few false positives (1 - 44, on average 9 frames) occurred due to a rectangular black border and an unfavorable color distribution that is very unlikely to occur in endoscopic videos. In absolute numbers, 194 false positives were detected in total which corresponds to 0.001% of all frames. However, all of them were filtered out by the outlier correction phase so finally not a single false positive was incorporated into the result.

5 Encoding efficiency

Our original incentive behind the idea of circle detection was the presumption that if we do not encode the noise in the black border of endoscopic videos we can considerably improve the encoding efficiency, i.e. we can encode the video in the same quality with less bits. What is more, we believe that the video playback has a more pleasant and less distracting appearance to the observer if the border area features one homogenous color.

Figure 7 depicts example patches from the border region and the corresponding histograms to illustrate the intensity of the noise. If a video is recorded, the encoder misinterprets this noise as detail information and tries to preserve it. This causes a certain ratio of the available bitrate to be wasted to encode completely irrelevant information caused by sensor errors. Figure 8(b) illustrates the macroblock (MB) types chosen by an H.264/AVC encoder for an example frame which actually has a rather low noise intensity. Nevertheless, many of the MBs in the border regions are coded as intra MB or predictive MB which implies the useless coding of coefficients and prediction information.

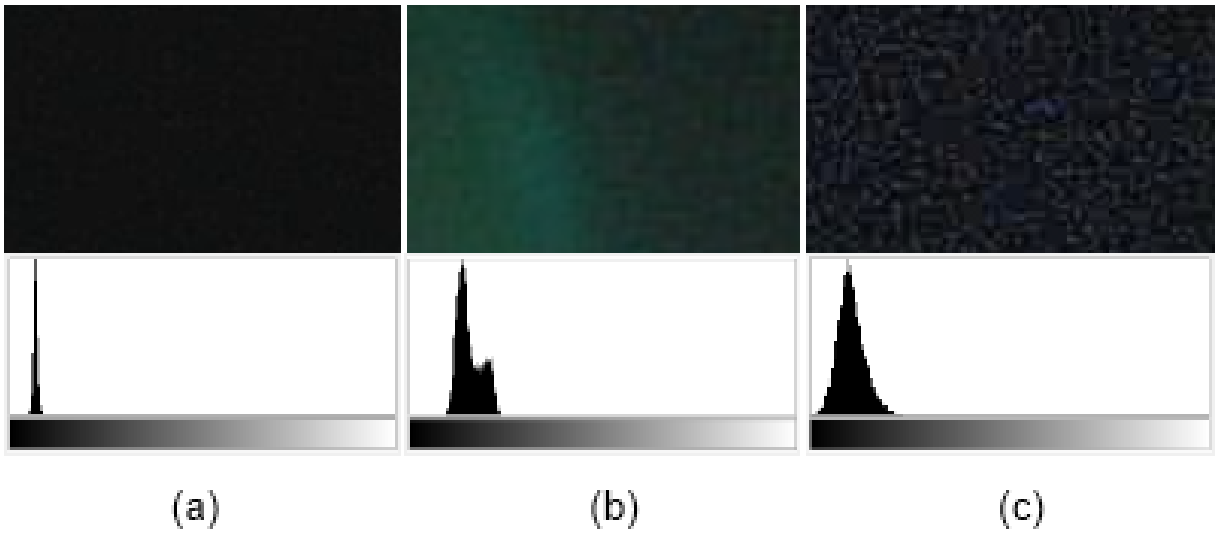

Figure 7: Example patches from border regions with histograms. (a) in-patient (HD), (b) in-patient (SD), (c) out-of-patient (HD)

5.1 Black border overlay

In order to improve the encoding efficiency of endoscopic videos, we superimpose the noisy border regions with a pure black mask which we call border overlay. The underlying conjecture is that this modification causes H.264/AVC encoders to choose skipped MBs instead of intracoded and predictive MBs for the border regions. Thus, no image sample information, residual coefficients, MB partitions etc. have to be coded and a lower overall bitrate can be achieved for the same visual quality of the content area. The border overlay is created by modifying each individual frame according to equation 1.

$$I(x,y) = \begin{cases} 0 & \text{if } x \leq x_c - b(y) \vee x \geq x_c + b(y) \\ I(x,y) & \text{otherwise} \end{cases} \tag{1}$$

with

$$b(y) = \begin{cases} \sqrt{(r*\varepsilon)^2 - (y_c - y)^2} & \text{if } |y_c - y| < r*\varepsilon \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $I(x,y)$ is the pixel at position (x,y) of the frame, x_c, y_c and r represent the parameters of the circle (center coordinates and radius) and ε is the tolerance margin coefficient which can be used to slightly extend the content area. This makes sense in endoscopy domains where the viewing direction of the endoscope is indicated by a small spike at the border of the circular content area and should be preserved. In the following experiments we used an ε of 1.

5.2 Evaluation method

To evaluate the impact of the black border overlay on encoding efficiency, we used four representative video data sets with a total length of 224 minutes. They consist of 138 video files which have been randomly obtained from cooperating hospitals. All videos have originally been encoded with the MPEG-2 video codec in constant bitrate mode, which is commonly used in many endoscopic video capturing systems for historical reasons. The data sets are summarized in Table 3.

Data set	Resolution	Bitrate	Framerate	Files	Length
ip (HD)	1920x1080, 1280x720	20 Mb/s, 12 Mb/s	25 fps	20	74 min.
ip (SD)	720x576, 720x544	7 Mb/s,	25 fps	69	53 min.
ip (CIF)	384x288	7 Mb/s,	25 fps	31	27 min.
oop (HD)	1920x1080, 1280x720	20 Mb/s, 12 Mb/s	25 fps	18	70 min.

Table 3: Video data sets

The data sets contain typical endoscopic videos in different resolutions. Our main data set consists of HD (High Definition) videos because HD systems are currently establishing as common practice and presumably will be widely used in the near future. Nevertheless, even if HD is used during surgery, due to economic restrictions, for recording most surgeons use SD (Standard Definition) or even CIF (Common Interchange Format) resolution. Recordings in SD quality are particularly relevant for this evaluation because existing endoscopic video archives mainly consist of such videos. They feature noticeably more border noise than HD videos. Moreover, the border area sometimes has a green or red tinge, which wastes bits that could better be used for the content area.

The first three data sets show actual operation scenes which take place inside the patient (ip). However, as our HD data set has been obtained randomly from a cooperating hospital it quite authentically reflects the way endoscopic videos are recorded and stored in practice. We observed that it also contains a considerable proportion of frames showing the environment outside the patient (oop) instead of an actual intervention. Out-of-patient frames are extremely noisy, apparently due to the ambient light of the operation room. They carry no relevant information at all and in a perfect world should be filtered out completely (we will address this issue in future work). Nevertheless, they are often captured unintentionally and therefore are currently included in many stored recordings. Hence, we split up the ip and oop components of the HD data set to separate files and evaluated the encoding efficiency separately. The difference in terms of border noise is illustrated in Figure 7.

We used the well-established x264 encoder (version 0.118.x) in conjunction with the ffmpeg API (version 0.7.13) to transcode the original MPEG-2 videos of our test sets to H.264/AVC. We chose this codec because it is one of the most popular state-of-the-art codecs and capable of encoding homogenous areas very efficiently with skipped macroblocks. As rate control mechanism we chose crf (constant rate factor) because it is the best suited mechanism for our application as we are not interested in a constant bitrate or fixed filesize but in constant quality. The crf rate control accepts a crf value in the range from 0 to 51 where a lower value involves a lower quantization and thus yields a higher quality [12].

For each original MPEG-2 video file, we used x264 (Main profile with default parameters) to encode one H.264 version without modification and one with a black border overlay. We performed this step with crf values from 18 to 28 which is a "subjectively sane range" according to the ffmpeg encoding guide[1]. We exploit the fact that the crf rate control provides the same visual quality of the content area for both versions of each crf value, but uses a different overall bitrate due to the modified border regions in the overlay version.

To ensure that this assumption holds and a file size reduction does not involve a quality degradation we additionally checked the PSNR (peak signal to noise ratio) of the two transcoded versions. As reference we used a H.264 version encoded with crf 1 which is practically lossless compared to the original MPEG-2 video. For the PSNR calculation we only considered the circular content area because otherwise the overlay would be judged as quality degradation. The content areas can not be expected to be completely identical because the rate control mechanism behaves slightly different due to the modified global image content. This is also the reason for minor differences in the MB decisions inside the content area (see Figure 8). We found that the average PSNR difference between the two versions is 0.17 dB (between 0.14 dB for crf 28 and 0.22 dB for crf 18). This slight difference is neglectable so we can state that the content areas of both versions have the same visual quality. Based on that, it is straightforward to measure the gain in coding efficiency by looking at the filesize difference of the two versions for each crf value.

[1] http://ffmpeg.org/trac/ffmpeg/wiki/x264EncodingGuide

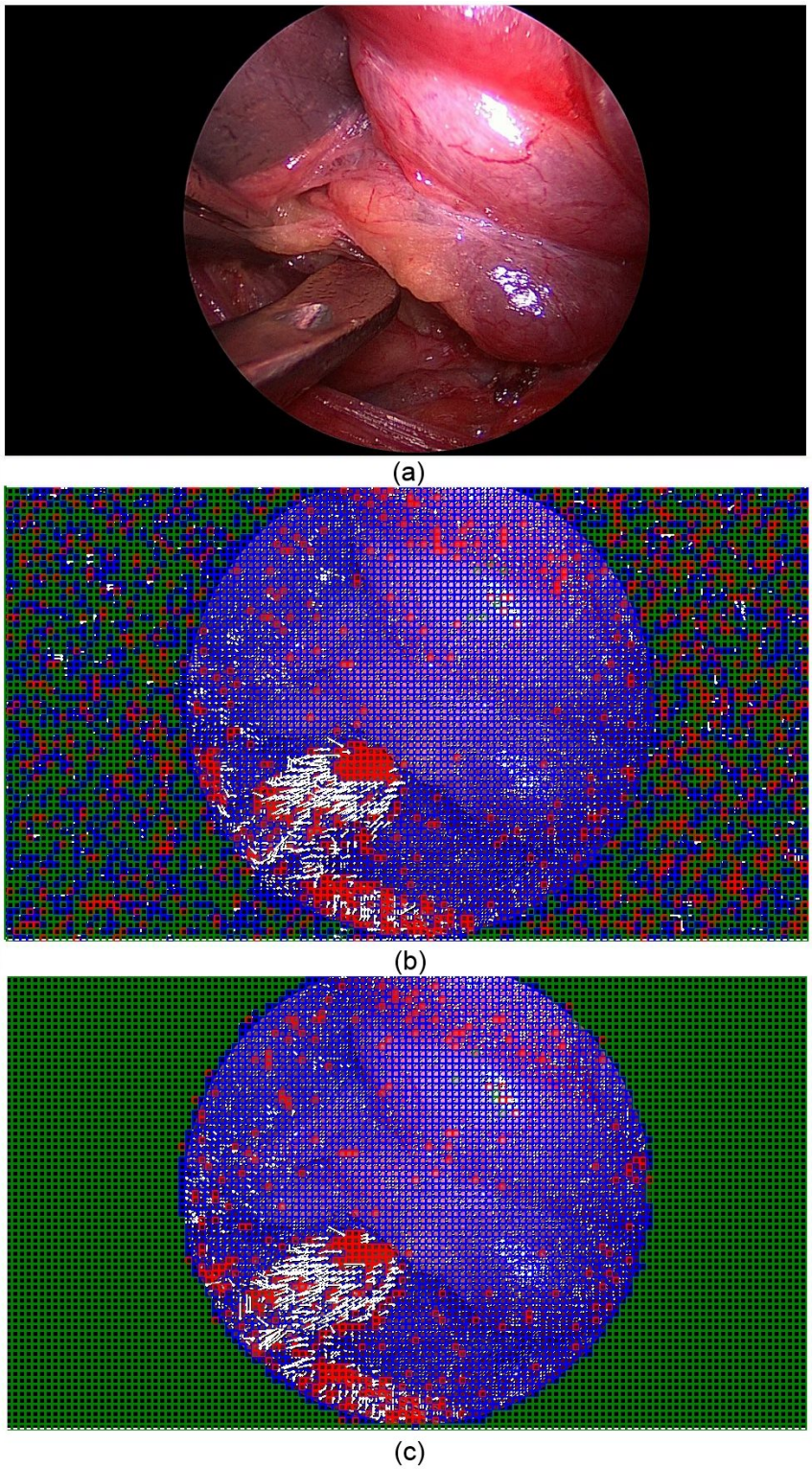

Figure 8: Visualization of the macroblock analysis of a representative example frame. (a) shows the original image, (b) shows the macroblock types of the unmodified version, (c) shows the macroblock types of the overlay version. The color codes are as follows: green = skipped MB (very low bitrate consumption), red = intra-coded MB (high bitrate consumption), blue = predicted MB (medium bitrate consumption), white = motion vector

5.3 Experimental results

The detailed results of our experiments are summarized in Table 4 and illustrated in Figure 9. The values in Table 4 denote the weighted average filesize (or bitrate) reduction between the unmodified and the overlay version of all videos of the respective data set for a given crf value, i.e., for example the value of 6.74 means that the filesize has been reduced by 6.74%. Formally, they have been calculated according to equation (3).

$$R(d,q) = \frac{\sum_{i=1}^{n_d} (1 - S'_{d,i,q}/S_{d,i,q}) * L_{d,i}}{\sum_{i=1}^{n_d} L_{d,i}} \tag{3}$$

$R(d,q)$ is the bitrate reduction for data set d and crf value (quality) q,
n_d is the number of video files of data set d,
$S_{d,i,q}$ is the filesize of video i of data set d with crf q encoded without modification,
$S'_{d,i,q}$ is the filesize of video i of data set d with crf q encoded with a black border overlay,
$L_{d,i}$ is the (temporal) length of video i of data set d.

	Average bitrate reduction			
crf	ip (HD)	ip (SD)	ip (CIF)	oop (HD)
18	6.74	26.28	33.05	48.20
19	6.71	25.12	28.45	48.01
20	6.61	22.90	22.32	48.04
21	6.43	20.08	16.26	48.19
22	6.16	16.32	10.92	48.02
23	5.91	12.71	7.11	47.92
24	5.57	9.64	4.25	47.82
25	5.15	6.83	1.54	47.46
26	4.98	4.87	0.43	47.02
27	4.57	3.16	-0.65	46.21
28	3.79	2.03	-1.15	44.27

Table 4: Coding efficiency improvement

For the HD in-patient videos only a modest gain in encoding efficiency of about 6% can be observed. There is no significant difference between the two different HD resolutions (1920x1080 and 1280x720). On the contrary, for the out-of-patient videos the border overlay tremendously reduces the bitrate by nearly 50%. For both data sets we can only observe a slight increase for lower crf values. The result for the HD in-patient videos can be explained by the fact that their original quality is already high. On the other side, the extremely high reduction rate for HD out-of-patient videos correlates to the ratio of border pixels to total pixels, which is in the range from 48-53% in our testset. This means that without a border overlay the encoder uses roughly the same number of bits for the noisy border region as for the content area.

For the low resolution videos the result is very different. Both SD and CIF resolution show a strong increase of encoding efficiency with higher quality. For crf 18 (which produces high quality with practically no visual loss compared to the original signal) the gain is about one third of the filesize, which is a substantial improvement. This result is a consequence of the lower quality compared to the HD in-patient videos which leads to a higher amount of border noise. What makes this result even more remarkable is the fact that the low resolution videos have an aspect ratio of 5:4 or 4:3 while the HD videos have a 16:9 widescreen aspect ratio. That means that the ratio of border pixels to total pixels is only about 30-40% while for the HD videos it is about 50%.

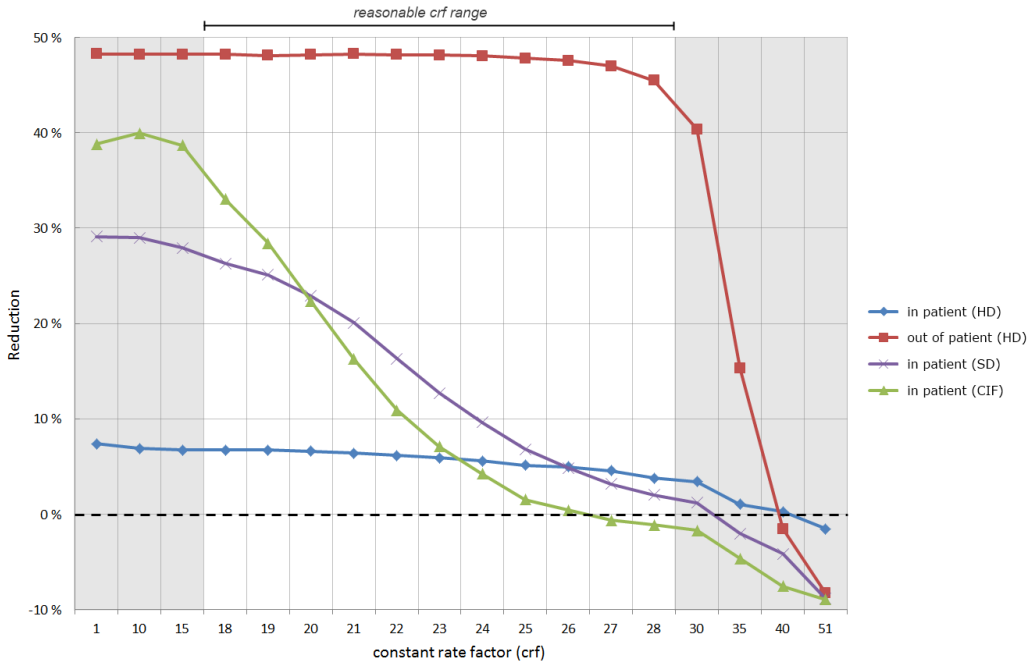

Figure 9: Bitrate reduction for different crf values

The gain in encoding efficiency highly depends on the crf factor. The reason is that the higher the crf value, the more noise is filtered out by the stronger quantization itself, so the difference between the overlay version and the non-overlay version becomes smaller. Thus, the tradeoff of a low crf value that produces high quality for the content area is that it also retains the border noise if no overlay is used.

Another interesting observation is that the curves of SD and CIF are similar, but shifted along the crf axis. This indicates that the resolution plays a role in the choice of an appropriate crf value to the effect that videos with lower resolution need a lower crf value to reach a certain quality. A possible explanation could be that for a higher resolution, the noise is represented by more pixels and therefore can withstand a higher quantization in the encoding process which always operates on blocks of the same size, regardless of the resolution.

In Figure 9 we also illustrate the filesize difference for crf values outside the reasonable range (with a gray background and with larger steps). It shows that crf values below 18 do not give a much better result, although they produce much larger files. The greatest improvement can be noticed for the CIF videos which reinforces our oberservation regarding the crf shift for different resolutions. Additionally, we see that for the HD out-of-patient videos the reduction rate only starts to slump between crf 30 and 35. Also for the HD in-patient videos the largest decrease can be observed in this range. Interestingly, with the highest possible crf value of 51 all data sets yield a negative reduction value. That means that the version with the black border overlay is larger than the unmodified version. However, this counterintuitive result has no practical relevance because the video quality in this range is anyway far from being acceptable in practice.

6 Conclusion and future work

In this paper we have presented a novel robust algorithm for differentiating between frames of endoscopic videos that feature the typical circular content area and frames that show a zoomed image. In the former case, the exact parameters of the circle are determined. This information is an important input for

further sophisticated content based analysis algorithms to narrow down analysis to the relevant parts of the image. Moreover it can be used for improving encoding efficiency, for economic printing of findings in terms of ink consumption and for optimizing content visualization in summaries etc.

The evaluation shows that our algorithm is highly accurate and reliable, especially in terms of avoiding false positives. Furthermore, it can also be executed in realtime scenarios. In offline applications it can be used as efficient pre-processing step, especially if the required pre-processing operations (scaling, blur filtering, edge detection) have to be conducted anyway and can be reused.

We also evaluated the impact on coding efficiency of superimposing the noisy border regions by a homogenous (black) border overlay. We demonstrated that this modification causes encoders to predominantly chose skipped macroblocks for the border regions. Thus, the overall bitrate can be reduced considerably without degrading the quality of the circular content area. In case of a limited constant bitrate the visual quality of the content area could even be enhanced because a higher ratio of the bitrate is available for the content area. As a convenient side effect, the decoding time is reduced as well because for the skipped MBs no coefficients and no motion vectors have to be decoded. Moreover, denoised borders have a more pleasing visual appearance and are less distracting. However, the extent of the bitrate reduction depends on the intensity of the border noise as well as on the degree of compression.

The most significant gain in encoding efficiency (around 50%) was observed for out-of-patient video segments which typically feature very heavy border noise. These irrelevant segments can congest medical video archives because the heavy border noise is misinterpreted as detail and therefore encoded with a high bitrate. In our future research, we plan to address the problem of automatic out-of-patient detection separately and more in-depth.

Additionally, the proposed method will be used as a pre-processing step for further analysis algorithms that we are currently working on. We also plan to evaluate to what extent the consideration of the actual content area improves the performance of these techniques. Furthermore, it should be investigated and quantified to what extent printing ink could be saved if the border region is ignored for printed findings. In this context, it also would be interesting to see if physicians accept this modified appearance of images. Additional future work may include the evaluation of the algorithm with further testsets, an automatic adaptation of parameters (thresholds etc.) to a given training set and minor performance optimizations.

References

[1] L. Alexandre, N. Nobre, and J. Casteleiro. Color and position versus texture features for endoscopic polyp detection. In *International Conference on BioMedical Engineering and Informatics, 2008. BMEI 2008*, volume 2, pages 38–42, May 2008.

[2] M. Arnold, A. Ghosh, S. Ameling, and G. Lacey. Automatic segmentation and inpainting of specular highlights for endoscopic imaging. *EURASIP Journal on Image and Video Processing*, 2010:1–12, 2010.

[3] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, Nov. 1986.

[4] C. Doignon, F. Nageotte, and M. de Mathelin. Segmentation and guidance of multiple rigid objects for intra-operative endoscopic vision. *Dynamical Vision*, pages 314–327, 2007.

[5] J. P. Helferty, C. Zhang, G. McLennan, and W. E. Higgins. Videoendoscopic distortion correction and its application to virtual guidance of endoscopy. *Medical Imaging, IEEE Transactions on*, 20(7):605–617, 2001.

[6] M. Hu, G. Penney, M. Figl, P. Edwards, F. Bello, R. Casula, D. Rueckert, and D. Hawkes. Reconstruction of a 3D surface from video that is robust to missing data and outliers: Application to minimally invasive surgery using stereo and mono endoscopes. *Medical Image Analysis*, 16(3):597–611, Apr. 2012.

[7] S. Hwang, J. H. Oh, J. K. Lee, Y. Cao, W. Tavanapong, D. Liu, J. Wong, and P. C. de Groen. Automatic measurement of quality metrics for colonoscopy videos. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 912–921, 2005.

[8] J. Illingworth and J. Kittler. A survey of the hough transform. *Computer Vision, Graphics, and Image Processing*, 44(1):87–116, Oct. 1988.

[9] M. Liedlgruber and A. Uhl. Computer-aided decision support systems for endoscopy in the gastrointestinal tract: A review. *Biomedical Engineering, IEEE Reviews in*, 4:73–88, 2011.

[10] M. Lux, O. Marques, K. Schöffmann, L. Böszörmenyi, and G. Lajtai. A novel tool for summarization of arthroscopic videos. *Multimedia Tools and Applications*, 46(2-3):521–544, Sept. 2009.

[11] S. J. McKenna, H. N. Charif, and T. Frank. Towards video understanding of laparoscopic surgery: Instrument tracking. In *Proc. of Image and Vision Computing, New Zealand*, 2005.

[12] L. Merritt and R. Vanam. Improved rate control and motion estimation for h.264 encoder. In *IEEE International Conference on Image Processing, 2007. ICIP 2007*, volume 5, pages V –309 –V –312, Oct. 2007.

[13] J. Oh, S. Hwang, Y. Cao, W. Tavanapong, D. Liu, J. Wong, and P. de Groen. Measuring objective quality of colonoscopy. *IEEE Transactions on Biomedical Engineering*, 56(9):2190–2196, Sept. 2009.

[14] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and trecvid. In *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, New York, NY, USA, 2006. ACM Press.

[15] P. Sánchez-González, F. Gayá, A. Cano, and E. Gómez. Segmentation and 3D reconstruction approaches for the design of laparoscopic augmented reality environments. *Biomedical Simulation*, page 127–134, 2008.

[16] S. R. Stanek, W. Tavanapong, J. S. Wong, J. Oh, and P. C. de Groen. Automatic real-time capture and segmentation of endoscopy video. volume 6919, pages 69190X–69190X–10. SPIE, 2008.

[17] C. Staub, G. Panin, A. Knoll, and R. Bauernschmitt. Visual instrument guidance in minimally invasive robot surgery. *International Journal On Advances in Life Sciences*, 2(3 and 4):103–114, 2011.

[18] D. Stoyanov and G. Z. Yang. Removing specular reflection components for robotic assisted laparoscopic surgery. In *IEEE International Conference on Image Processing, 2005. ICIP 2005*, volume 3, pages III – 632–5, Sept. 2005.

[19] S. Voros, J.-A. Long, and P. Cinquin. Automatic detection of instruments in laparoscopic images: A first step towards high-level command of robotic endoscopic holders. *The International Journal of Robotics Research*, 26(11-12):1173–1190, Nov. 2007.

[20] R. Yao, Y. Wu, W. Yang, X. Lin, S. Chen, and S. Zhang. Specular reflection detection on gastroscopic images. In *2010 4th International Conference on Bioinformatics and Biomedical Engineering (iCBBE)*, pages 1–4, June 2010.

Institute of Information Technology
University Klagenfurt
Universitaetsstr. 65-67
A-9020 Klagenfurt
Austria

http://www-itec.uni-klu.ac.at