university of groningen

faculty of science and engineering

Summarisation of Endoscopy Video

Bachelor thesis

University of Groningen

SVCG Group

In collaboration with ZiuZ Visual Intelligence

Author:
Nico Stegeman

Supervisors:
André Sobiecki
Alexandru C. Telea

July 2017

Abstract

Capsule endoscopy is a diagnosis tool for viewing the small intestine. It is a small camera in a pill which films a video of around 8 to 12 hours. The goal of this bachelor thesis is to research and implement a method to summarise the capsule endoscopy video. To achieve this goal the algorithm cluster the video frames by finding local maxima in the neighbouring distance graph created from the colour histograms of every frame in the video. The algorithm then generates a key-frame for every cluster by calculating the centroid frame in the cluster.

The second goal of this bachelor thesis is to visualise the resulting summary. This is done by mapping every cluster based on the size and quality of the key-frame onto a timeline. This map can then be used to navigate through the key-frames generated by the algorithm. To further visualise the distance between key-frames the program shows a strip of key-frames with the key-frame of the current cluster at the centre.

Contents

Chapter 1

Introduction

1.1 Capsule Endoscopy

Capsule endoscopy is a medical diagnosis tool which is used to record a video of the patients digestive tract. The method consists of two devices. The first is a capsule pill with a camera and a light inside. This pill is swallowed by the patient and sends its images to an external receiver. This external receiver is worn by the patient. The capsule endoscopy is used for capturing images of the intestines which are not reachable with normal endoscopy methods. Capsule endoscopy is used for the following procedures:

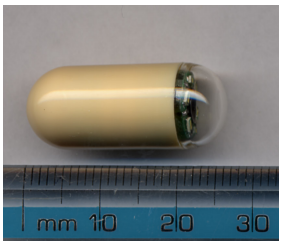

Figure 1.1: Image of a Endoscopy Capsule

- Find the cause of gastrointestinal bleeding.
- Diagnose inflammatory bowel diseases.
- Diagnose cancer.
- Diagnose celiac disease.
- Screen for polyps
- Do follow-up testing after other imaging tests.

One of the problems with capsule endoscopy is that the resulting video is 8 to 12 hours long. It takes a doctor about 2 hours to review the video. The main goal of the project will be to reduce the amount of time that is needed when reviewing capsule endoscopy videos. To achieve this goal this paper will examine if and how we can create a strip of the timeline that show a condensed view of the video. Thus presenting a summary of the original video. From this we set two research questions:

1. How can we summarise a long gastrointestinal video in a strip/timeline-like metaphor?

2. How can we represent groups of related frames by single 'summary' frames?

In chapter 2 the research that has been done on this topic is discussed. In chapter 3 the implementation of the algorithm is examined. In chapter 4 the result of experimenting with the parameters of the algorithm are examined and a qualitative examination of the user interface is done. Chapter 5 concludes the thesis.

1.2 Requirements

Part of the thesis will be a software project which will implement an algorithm that summarises an endoscopy video. The program will also visualize the results in such a way that we try to answer the first research question. The software project has a few requirement that must be met.

1. The product should be able to calculate a summary in under 15 minutes on consumer grade hardware. The products main purpose is saving time for the doctors in analysing the endoscopy video. Therefore the algorithm should not take too long.

2. There should be a balance between the accuracy and verbosity of the summarisation. The product should generate the least amount of frames as possible. If the resulting product generates too many frames the doctor might as well review the original video. On the other hand the key-frames should be as representative of their cluster as possible. This will be harder if there are fewer key-frames.

3. The algorithm should be nearly automatic. The users of the product are doctors, who are not trained computing scientists or mathematicians. Thus the product has to be as user friendly as possible.

4. There should be an overview of all clusters. To answer the first research question we will have to visualise all cluster on one screen.

5. Clusters should be easy to navigate. Simple navigation plays a big part in cutting down review time for the user.

6. Clusters should be linked to the original video. The user must be able to easily access the input video frames of every cluster.

Chapter 2

Related Work

Video summarisation is an indexing and search technique for large amounts of video data. As the name suggest the technique outputs a short summary of the video to give the user a visual abstract of the video sequence. A good video abstract tries to maximize the amount of information retained in the summary while removing as much noise as possible [1]. Automatic video summarisation is often done with clustering algorithms where the algorithm either extract a key-frame for every cluster or a moving images (video skims). A cluster in a clustering algorithm refers to a group of objects that are more similar to each other then to those objects in the other clusters (groups).

In my research I came across four algorithms that were interesting. They are all algorithms designed to summarise capsule endoscopy. In this chapter I will describe the algorithms and compare them with each other. The algorithms will be compared at the hand of several attributes.

1. Reproducibility: Here the algorithm will be evaluated on how difficult the implementation of the algorithm will be.

2. Parameters: Here the parameters of the algorithm are discussed and compared.

3. Results: Here the result of the research paper are discussed and compared. Three of the four papers use the same semi-Hausdorff algorithm[2] to calculate their fidelity metric. The fidelity of a key-frame measures how representative a key-frame is of its cluster.

4. Performance: Here the time complexity of the algorithms are discussed and compared.

All the algorithms are clustering algorithms. A clustering algorithm divides all the data points in your data set into several cluster. These algorithm try to place similar data points into the same cluster. With video summarisation the algorithm also tries to generate one or several key-frames which are representative of their cluster. The papers do not discus the ways these algorithms may be

visualised, but the algorithms discussed all use a single key-frame to represent a cluster which is linked to a frame in the input video.

2.1 K-means

This section analyses a simple but interesting method for summarising endoscopy video by using an algorithm influenced by the k-means clustering algorithm [3]. The algorithm is a combination of finding local maximum and k-means. It creates a distance graph of the video frames. These distances are the chi-squared distance between two colour histograms of the frames. The colour histograms are in CIE lab colour space and use a 4x8x8 bin distribution. Then it creates cluster boundaries by searching for local maximum. It finds these local maximum with these properties.

1. The distance is the maximum distance over $2k$ symmetric neighbours distance comparisons.

2. The distance is n times greater then the second maximum.

The algorithm extracts a key-frame by picking the nearest frame to the cluster mean.

Reproducibility

The algorithm is not difficult to implement. The program needs a color histogram creator that turns a frame into a color histograms. It will need a frame comparator which returns the distance between two color histograms, an algorithm that implements the cluster boundary search and finally a key-frame extraction algorithm.

Parameters

There are two parameters that depend on user input. These are k and n from the cluster boundary detection.

Results

The quality of the algorithm is measured by its fidelity and compression ratio when used on test clips. The fidelity is a measure of the quality of the key-frames calculated using a a semi-Hausdorff algorithm[2]. It is supposed to measure the quality of a key-frame.

The average fidelity was 0.987 and the compression ratio was 0.901. This was with a k of 2 and a n of 2.5.

Performance

The boundary search can be done almost linear in the number of frames. The time complexity is $(2k) * n - 1$ where k is a user given constant and n is the number of frames.

2.2 Texture feature extraction

Another alternative to calculate frame distance and cluster the videos [4] is discussed in this section. The method here uses affinity propagation to cluster the video. Affinity propagation calculates centre points of clusters with two matrices of size $N \times N$ where N is the size of the data set. These matrices are updated in a iterative function which stops after x number of times.

The first matrix is the responsibility matrix R has the values $R(i, k)$. This values in this matrix quantify how suitable frame k as a centre for i. The second matrix is is the availability matrix A containing the values $A(i, k)$ that represent how appropriate it is for frame i to choose k as it's centre. The matrices are updated according the following equations where $S(i, k)$ is a similarity function:

$$R(i,k) \leftarrow S(i,k) - \max_{k' \neq k}\{A(i,k') + S(i,k')\} \tag{2.1}$$

$$A(i,k) \leftarrow min(0, R(k,k) + \sum_{i' \notin \{i,k\}} \max(0, R(i',k))) \tag{2.2}$$

$$A(k,k) \leftarrow \sum_{i' \neq k} \max(0, R(i',k)) \tag{2.3}$$

Affinity propagation needs a similarity function which is usually $-||x_i - x_k||^2$ where x_i is a quantification of frame i and x_k a quantification of frame k. The frames are quantified using two methods. The first detects colour feature. It does this using a spatial pyramid histogram in the HSI colour space. A spatial pyramid histogram has L levels every level a subdivision takes place and then histograms are created for each cell. The second method is Gaussian filtered local binary pattern. There a Gaussian filter is used to suppress noise. Then a feature histogram is created with the local binary pattern algorithm.

Reproducibility

This algorithm is a bit harder to implement. The challenge is in the local binary pattern and that the authors have omitted a lot of details when discussing their implementation.

Parameters

The pyramid histogram has has two dependant values L the max subdivision level and N the number of histogram bins. The local binary pattern has two

dependant values P and R. P is the number of sample points and R is the radius of the sample circle. Finally the affinity propagation algorithm is dependant on the maximum number of iterations.

Results

The same metrics to define quality and quantity as the k-means algorithm are used in this paper [2]. The fidelity is on average 0.912 and the compression ratio is 0.881.

Performance

Calculating the distance between two by the two methods are dependant on the frame size. The time it takes to calculate the affinity propagation is $I * N^2$ where I is the amount of iterations and N the size of the data.

2.3 Relation matrix rank

In this section we will analyse a method that uses a relation matrix rank algorithm to extract the key-frames[5]. In this algorithm segmentation is done twice. First the video is segmented by colour. The image is translated to HSV colour space. Then for both the H and S channel we create a histogram. For the H histogram we take the two highest colour bins and for the S histogram the highest bin and they are grouped as a vector. Clip boundaries are where this vector is differs between two frames.

Next a block edge directivity descriptor algorithm (BEDD) is used to create a texture feature vector from which the distance between two frames can be calculated. The BEDD method divides the image into 12 blocks then for every block is converted into grayscale. Then the edge directivity descriptor (EDD) technique is used to extract a 4-dimensional texture feature vector (TFV) for each grayscale-block. Then a 48-dimensional TVF is constructed by concatenation of the TFV of each block. The EDD uses vertical and horizontal Sobel operators to extract the TFV[6].

The distance between two frames is calculated with the following equation:

$$D(f_i, f_j) = \sum_{k=1}^{48}(1 - \frac{|TFV_{48}^i(k) - TFV_{48}^j(k)|}{TFV_{48}^i(k) + TFV_{48}^j(k)}) \tag{2.4}$$

Where f_i and f_j are frames with index i and j, TFV_{48}^i is the 48-dimensional TVF of the frame i. Then boundaries for the clusters are placed when a distance is higher then the threshold set by the user.

At last a relation rank matrix algorithm is used to calculate the most fitting key-frame. The relation rank matrix algorithm uses a matrix which contains the distances between all frames in the cluster (M_k) and a vector containing the key-frame relevance of all frames in the cluster (R_k) where k is the cluster the

matrix and vector belong to. The algorithm then iterates until the relevance vector does not change between iterations. The next iteration is calculated by the following equation where l is the current iteration:

$$R_k(l+1) = M_K R_k(l) \tag{2.5}$$

Reproducibility

The hardest part to reproduce will be the BEDD algorithm. This seems to be a complex texture feature extraction algorithm. The rest of the algorithm will not be hard to implement.

Parameters

The whole methods is dependant on one user variable. This is the threshold value in the second segmentation step.

Results

The paper uses the same metrics to define quality and quantity as the k-means algorithm [2]. The average compression ratio of their method is 0.779. The fidelity with random key-frame selection is 0.976 and with the relation matrix rank calculation 0.984.

Performance

The clustering is linear in the number of frames. The algorithm that calculates the key-frame is iterative until a threshold is reached so its complexity is less fixed.

2.4 Unsupervised learning and feature discrimination

This paper only details the clustering part of the algorithm. It calculates cluster centres using relevance weights. It learns the optimal relevance weight for each feature subset within each cluster with an iterative algorithm. It also tries to find the optimal number of clusters unsupervised [7].

Reproducibility

The proposed algorithm is very complex and the article does not detail how the frame features are calculated. This will make it hard to reproduce.

Parameters

There are four user variables that must be taken into account k, M, η, m. Where k is the number of feature subsets. M is the maximum number of clusters. η is the importance of second term over the first. And m is a fuzzier.

Results

This paper compares its result with three other cluster algorithms. No compression ratio is given.

Performance

The performance is not fixed. The algorithm convergences to a stable center. Every iteration all frames will have to be compared to each other so each iteration has a complexity of n^2 where n is the number of frames. This is will make it a very slow algorithm.

2.5 Comparison

In this section I am comparing the different algorithms based on four criteria. These criteria are the reproducibility of the algorithms, the user defined parameters of the algorithms, the results of the algorithms and the performance of the algorithms. A simple summation into +, ± or - is given in table 2.1.

attribute	k-means	Texture feature extraction	Relation matrix rank	Unsupervised learning
Reproducibility	+	±	±	-
Parameters	+	-	+	±
Results	+	-	±	?
Performance	+	-	±	-

Table 2.1: Algorithm comparison table.

Reproducibility

Implementation wise the algorithms can be roughly divided into three parts. These three parts are feature-distance calculation, clustering and key-frame extraction.

For feature-distance calculation implementation wise the k-means and relation matrix rank algorithms are the most interesting. They are relatively simple to implement while at the same time resistant to noise.

For the clustering algorithms the simplest to implement are the relation rank matrix algorithm and k-means. These algorithms go linear through the video frames and add boundaries with on simple calculation based on the distance. The texture feature extraction and unsupervised learning algorithm calculate cluster centres using a form of weighting that updates every iteration. The

unsupervised learning algorithm uses a very complex algorithm to calculate these centres.

Key-frame extraction is part of the unsupervised learning and texture feature extraction algorithms. For the k-means and relation matrix rank algorithms they are separate from the clustering. The k-means uses a simple algorithm to extract the key-frame, but the relation matrix rank algorithm uses more complex relevance algorithm to calculate the key-frame of the cluster.

Parameters

All of the algorithms are dependent on user defined constants. These parameters will be compared with each other while keeping our user group in mind. These algorithms will have to be usable by doctors who have no background in computing science or mathematics.

The relation matrix rank algorithm has the fewest parameters with only one user defined constant. And the texture feature extraction and unsupervised learning the most with 4 parameters. The effects of the parameters of these algorithms are also hard to predict by someone who hasn't done a full study of the algorithms. That will make them hard to use for our user. The parameters of the k-means and relation matrix rank algorithm will be easy to explain and experiment with.

Results

Three of the four papers use the same metrics to test their algorithms. The unsupervised naming algorithm uses different performance measures. The problem with this paper is that it does not explain how it calculates these measures for their technique and the techniques they compare themselves with. All the other papers use a fidelity and compression measure.

From these result the k-means and relation matrix rank algorithms seems to retain the best fidelity of their key-frames. The k-means and texture feature extraction algorithms get the best compression ratio.

An interesting thing to note is that all of the result from the algorithms are about compression rates around the ten to twenty percentage. All of them seem to assume that the summary video is its own product instead of an navigation tool for the input video. Therefore an interesting experiment would be to evaluate what happens when the compression ratio gets even higher.

Performance

The time complexities of these algorithms with exception of the k-means algorithm are not fixed unless an maximum amount of iteration constant is added to these algorithms. With k-means being linear it will be the fastest of all the algorithms. On second place will be the relation matrix rank algorithm this algorithm clusters linearly in time, but the key-frame extraction is iterative until

a threshold is reached. The texture feature extraction and unsupervised learning algorithm have a $)(n^2)$ time complexion every iteration. For all videos it is also important to note that they are also influenced by the size of the frames. This was left out of the comparisons because the frame size is constant within a video and all are influenced in the same way.

The memory requirements of texture feature and unsupervised learning are larger then the other algorithms. They need $O(n^2)$ memory for the matrices in the cluster calculation. The k-means and relation matrix rank will need between $O(1)$ and $O(n)$ for clustering. The memory complexity depends on our implementation if we store all distance calculations then we do not have to recalculate them when extracting key-frames or when the user changes one of the constants of the algorithm.

2.6 Conclusion

From this comparison I have decided to not implement the texture feature and unsupervised learning algorithms. The texture feature algorithm seems slow, memory intensive and dependant on too many constants. The algorithm also gets worse results then the k-means and relation matrix rank algorithms in fidelity. The unsupervised learning algorithm is very complex to implement and the paper describing the algorithm does not describe what the experiment result were that they measured. The paper also does not provide the compression ratio of the algorithm.

The relation matrix rank algorithm was not implemented in this project. The colour histograms clustering that is the first part of the algorithm did not preform as advertised by the paper describing it. In the test videos it on it's own had a compression rate around 0.7. That means that with that pass alone the compression already lower then the result presented in the article. This pass also has no parameters so it can not be tweaked by the user. The k-means algorithm was implemented instead.

Chapter 3

Methods

The current chapter will describe my implementation of the k-means based algorithm. In the section Overview an overview of the implementation is given. The sections Histograms, Distance, Clustering, Key-frame Computation and Key-frame Fidelity explain the corresponding steps in the program pipeline. In the final section Complexity the time complexity of the implementation is discussed in detail.

3.1 Overview

The algorithm consists of five different steps in a pipeline like shown in figure 3.1.

- First it creates an array of color histograms.
- Then it will calculate an array containing the distance between all consecutive frames.
- After that it will create cluster.
- Then for every cluster it will calculate a key frame.
- And finally it will calculate the fidelity of every key frame.

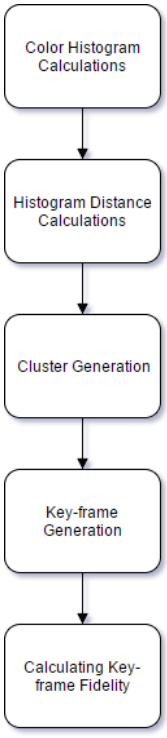

Figure 3.1: Program pipeline

3.2 Histograms

The colour histograms are the CIE lab colour space. This colour space is used, because it separates the colours from the light intensity. The histogram has an 4x8x8 bin distribution which result in a 20 bin histogram. For every frame a histogram is created and stored in an array. The histograms are stored because calculating the histograms is the slowest

part of the algorithm. When stored these can be reused when calculating key frame fidelity and when changing parameters.

3.3 Distance

When working with a data set the distance is a similarity metric between two data points. This part of the program creates an array of distances between all consecutive frames. The distance between consecutive frames is calculated from the CIE lab colour space histograms using the Chi-squared test described in equation 3.1.

$$d(H1, H2) = \sum \frac{(H1 - H2)^2}{H1} \tag{3.1}$$

Where $H1$ and $H2$ are colour histograms in CIE lab colour space as described in section 3.2 of two consecutive frames. Again the values are stored in memory because these don't change when the parameters are changed.

3.4 Clustering

Clustering is a method of grouping data points in a data set together based on their similarity. The algorithm used here is based on finding local maxima. The clustering is done based on the distance array. The algorithm loops through all calculated distances. At every distance point it will look to $2 \times k$ symmetrical neighbours (k neighbours to the left and k to the right) and get the maximum value of these neighbours. If the current distance is n times larger then the maximum of its neighbours ($n \geq 1$), then a boundary is placed between the current two frames that are compared. Where k and n are user specified parameters.

3.5 Key-frame Computation

A key-frame is a frame that represents a cluster. During development two versions were implemented. The first used calculated a mean colour histogram and picked the frame nearest in distance to the mean histogram. The resulting key-frames were not of a good quality. The second version calculated the mean distance of the consecutive frames in the cluster and picked the frame closest to this number.

3.6 Key-frame Fidelity

The key-frame fidelity is a metric that expresses the quality of a key-frame by comparing it to the other frames of the cluster. The key frame fidelity is the max

distance of the distances between the key frame and it's cluster. The equation **??** is used to calculate the fidelity

$$fidelity(KF_k, F) = \max_i distance(F(i), KF_k) i = s_k, ..e_k \tag{3.2}$$

Where KF_k is the key frame of cluster k, F is the set of frames in the video, s_k is the starting frame of the cluster k, e_k the final frame of the cluster k and distance the function calculation the distance between two frames as described in section 3.3.

The key-frame fidelity is calculated so that the user can view the quality of the key-frames relative to the other key-frames generated.

The distance values of the chi-squared function are not normalized. This is not a problem when trying to visualize the relative quality of the cluster. When trying to compare the quality of frames between frames it becomes a bigger problem. Now the different quality values become meaningless when compared to other videos. This made it hard to do a concrete evaluation of the quality of the key-frames.

To solve the problem of plain distance values not being comparable between videos, the key-frame quality is normalized by the standard deviation of distance values in the video. The advantage of this approach is that the mean distance of the video is the zero line of the quality video. That means that when the quality value of a key-frame is below zero it is a better key-frame for all cluster frames then picking any random frame in the video. The down-side is that it takes a lot of time to calculate the mean and standard deviation distance of a video. Because the normalization is only needed for the experiments and not for the users of the final product this is not a big issue.

3.7 Complexity

In this section we are breaking down the time complexity of the algorithm. The number of frames is n.

- The complexity of the histogram calculation is $O(n \times w \times h)$, where w is the pixel width and h the pixel height of the frames.
- The distance calculation is $O(n)$. The number of histogram bins is fixed in the program and the program has to calculate $n - 1$ distance values.
- The clustering calculation is $O(n * k)$. Where k is the parameter k. The clustering look at $2 \times k$ neighbours at every distance value.
- The key frame calculation is $O(n)$. For every cluster the algorithm calculates the mean distance and the nearest distance value to the frame.
- The key frame fidelity is in $O(n^2)$. Calculating the max distance for every key frame is linearly. The calculating of the video mean and standard deviation is quadratic in time. This means that it takes a long time to

calculate a full video. On the plus side the mean and standard deviation calculation is only present for the experiments are are not needed for the algorithm.

So in conclusion the time complexity of the algorithm is $O(n \times w \times h \times k)$ and for the experiments it is $O(n \times w \times h \times k + n^2)$.

Chapter 4

Results

4.1 Quantitative Evaluation

The hard way to test is the use actual users, but they were not available so we tested it ourselves from the perspective of doctors. For the experiments we have four test video's all with different symptoms. The test videos have a resolution of 320x320 and are shot at 3 frames per second.

- Angiodysplasie: This video has a running time of 11 hours and 59 minutes and 125608 frames.
- Bleeding: This video has a running time of 8 hours and 58 minutes and 68939 frames.
- Polyp: This video has a running time of 12 hours and 00 minutes and 124970 frames.
- Ulchera: This video has a running time of 11 hours and 38 minutes and 121934 frames.

For the experiments we run the algorithm on all test video and we test different values for the parameters k and n. For these videos we will test the algorithm on the quantity and quality of the key frames. The quantity will be measured in the amount of frames divided by the total amount of frames in the video. For the quality we will measure the mean key frame fidelities. Then we take the average key frame quality and normalize it by standard deviation. Because the ideal distance between two frames is 0 the lower the distance the better the quality of the key frame.

With the experiments described in this section we will try to answer two key questions. The first one is: Can the algorithm generate a summary video while keeping key-frame quality high. A doctor must be able to review the summary without losing information important to the diagnosis. The second question is: Can we estimate the parameter values for which the fewest amount of frames are generated with the highest quality.

Compression

On the y axis of the graphs is the compression rate of the algorithm. This is the amount of frames in the video divided by the amount of frames generated by the algorithm. On the x-axis is the parameter n. The different lines represent different values of k.

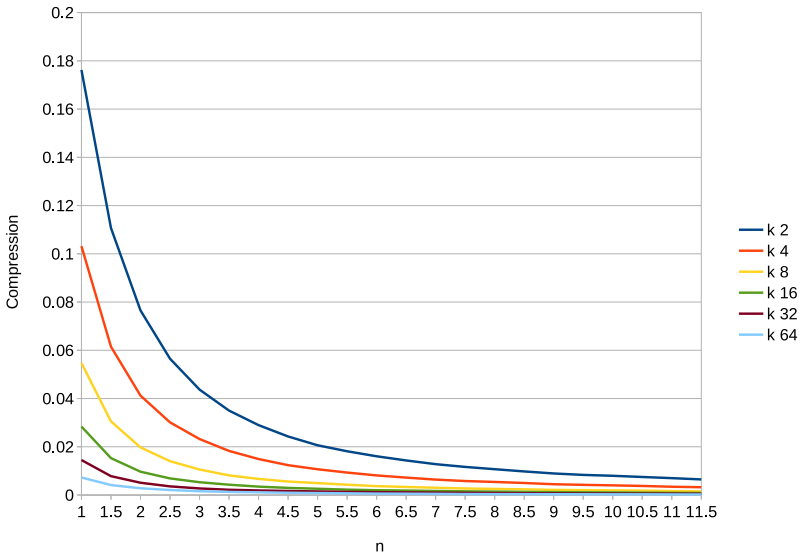

Figure 4.1: Compression graph of angiodysplasie video. Where the y-axis is the compression rate, the x-axis the parameter n and the different lines the parameter k.

As can be seen on the graph in figure 4.1 the compression greatly increases at low n values but the effect decreases as n becomes higher. The same is can be said for the k value. In the experiments the value is double every time but the increase in compression steadily becomes smaller. This is not only the case for the Angiodysplasie video, but for all test videos. The quantity graphs of the other videos are almost the same. As can be seen in the graphs in figures 4.2, 4.3, 4.4.

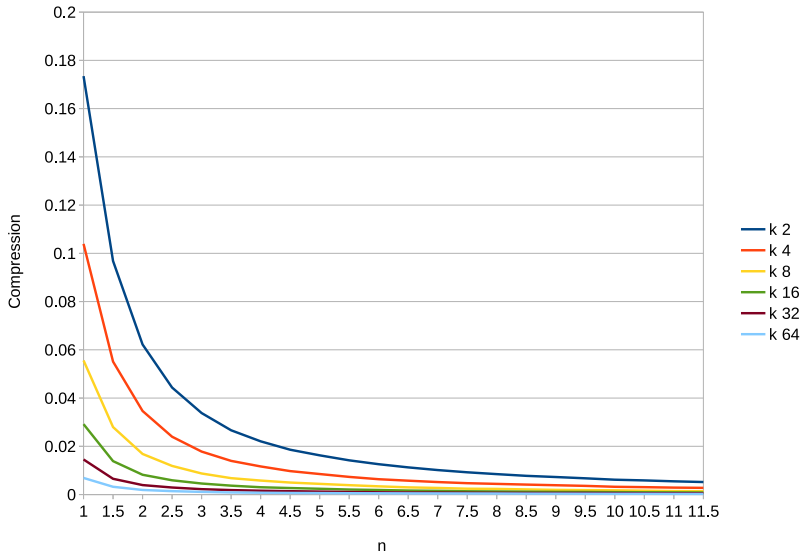

Figure 4.2: Compression graph of bleeding video. Where the y-axis is the compression rate, the x-axis the parameter n and the different lines the parameter k.

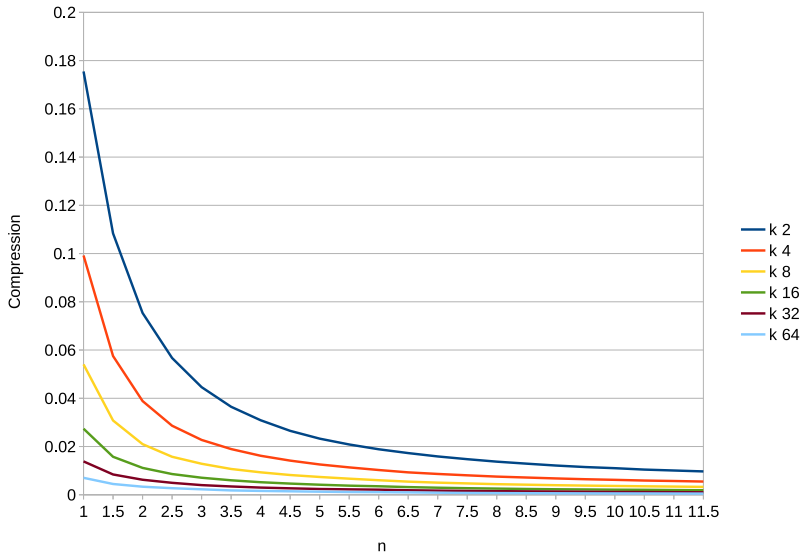

Figure 4.3: Compression graph of polyp video. Where the y-axis is the compression rate, the x-axis the parameter n and the different lines the parameter k.

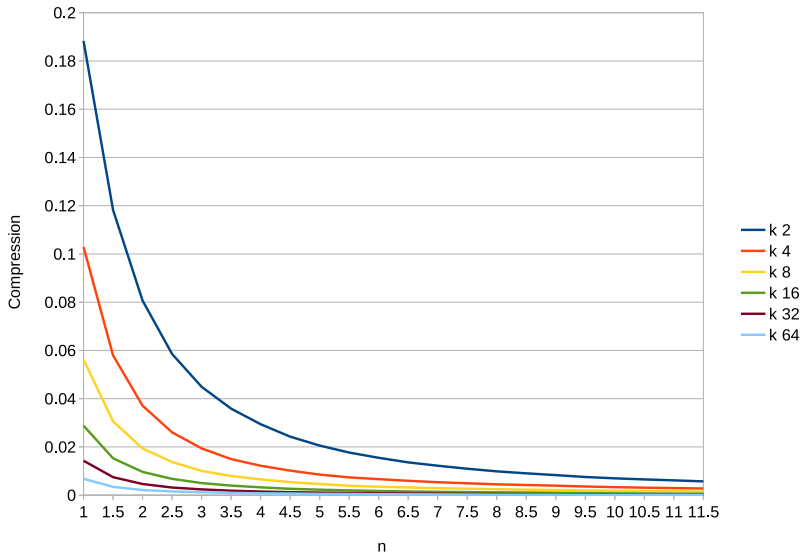

Figure 4.4: Compression graph of ulchera video. Where the y-axis is the compression rate, the x-axis the parameter n and the different lines the parameter k.

Key-frame Fidelity

On the y axis of the graphs is the normalized average quality of the key-frames. The quality of every key-frame is calculated by the algorithm and the average is on the y-axis. On the x-axis is the parameter n. The different lines represent different values of k. As can be seen on graph in figure 4.5 and the other fidelity graphs in figures 4.6, 4.7, 4.8 the quality of the key-frames decreases almost linearly for the lower k values. The higher k values are a bit more erratic in the key frame. There the average key-frame quality increases sometimes when increasing n. This may indicate that the key-frame calculation of the algorithm is not optimal.

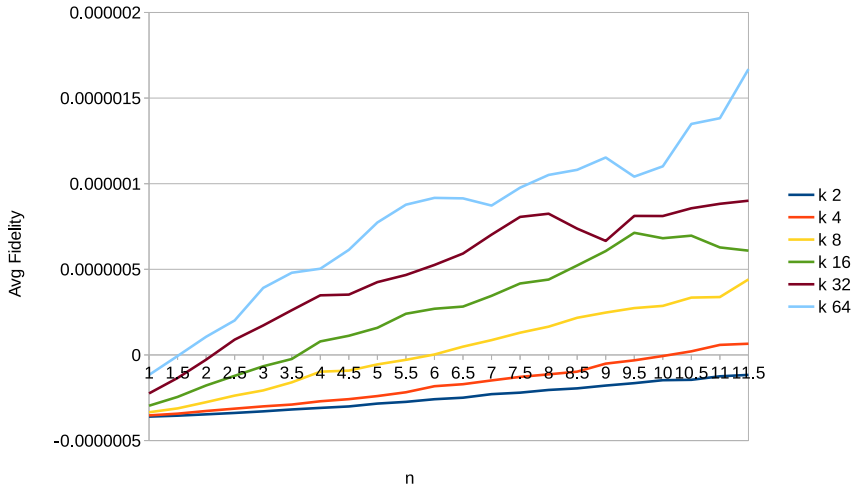

Figure 4.5: Fidelity graph of Angiodysplasie video. Where the y-axis is the average fidelity of the key-frames, the x-axis the parameter n and the different lines the parameter k.

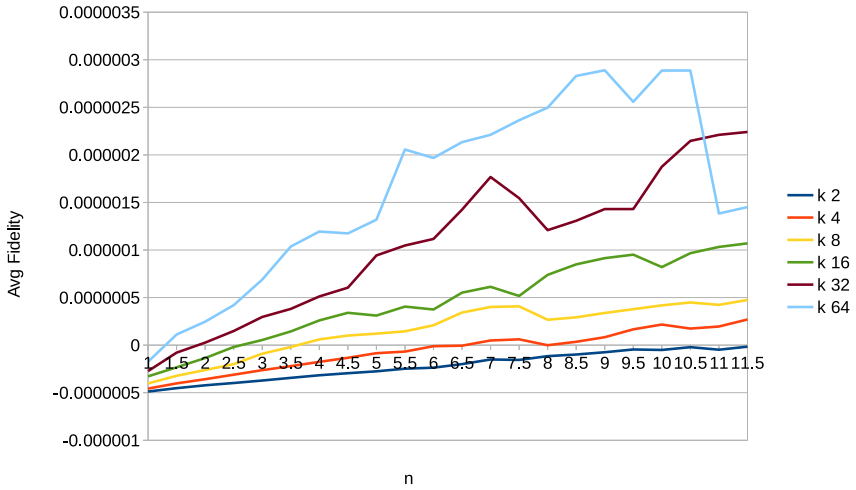

Figure 4.6: Fidelity graph of Bleeding video. Where the y-axis is the average fidelity of the key-frames, the x-axis the parameter n and the different lines the parameter k.

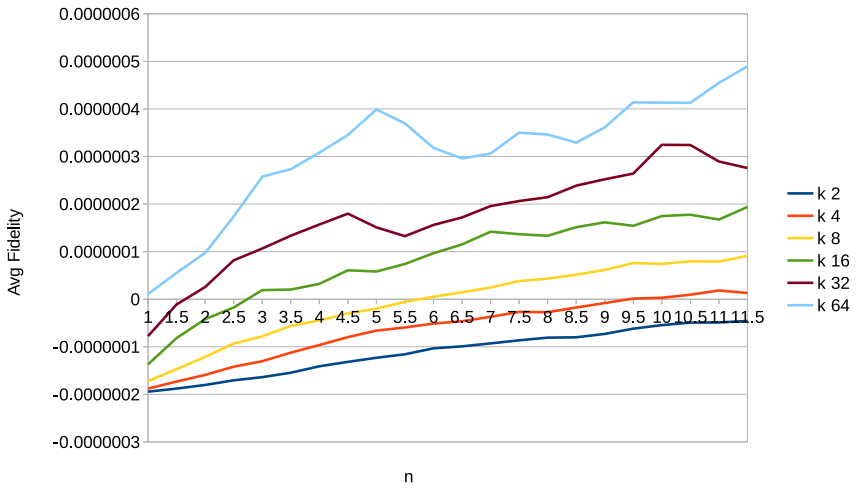

Figure 4.7: Quality graph of Polyp video. Where the y-axis is the average fidelity of the key-frames, the x-axis the parameter n and the different lines the parameter k.

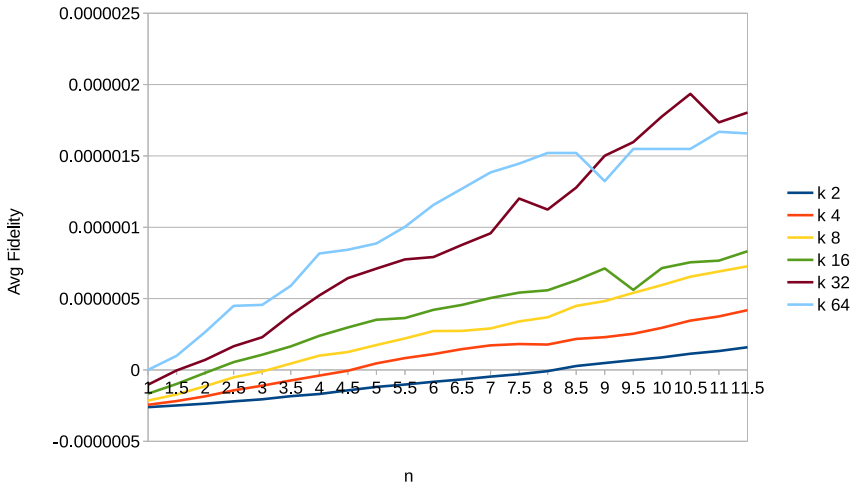

Figure 4.8: Fidelity graph of Ulchera video. Where the y-axis is the average fidelity of the key-frames, the x-axis the parameter n and the different lines the parameter k.

General Discussion

When increasing the parameters k and n the compression rate increases similar to a exponential decay function. The quality of the key-frames decreases almost linearly in the parameter n and k. With a slight favour towards the parameter n. This suggest a diminishing return in compression compared to the key-frame quality.

One of the interesting aspect of working with an algorithm like this is to make them usable to there target user. A doctor would most likely prefer pre set zoom levels to messing with mathematical parameters. So we try to define six levels of increasing compression. Then we pick the parameters that have the best average key-frame quality at a certain amount of generated frames. The parameters that are found are in table 4.1. In this table the compression values are a estimation, because they differ slightly between input videos. From the experiments with the parameters of the algorithm we found that increasing the n parameter decreases the quality slightly less then the k parameter. Therefore the zoom level also favour lower k values slightly.

Compression	k	n
0,06	2	2.5
0,03	2	4
0,008	4	6
0,004	8	5.5
0,0008	16	11.5
0,0002	64	10

Table 4.1: Table outlining zoom levels.

4.2 Qualitative Evaluation

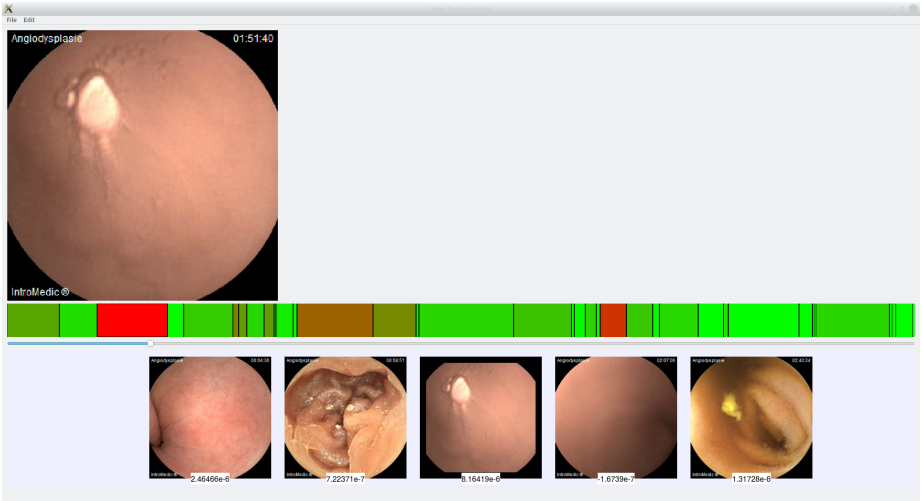

Figure 4.9: Screenshot of program.

Figure 4.9 shows the interface of the program. In the top bar of the program we have two options file and edit. In file we can open a new input video and save the summary to video. In edit we can change the parameters of the algorithm.

The next element below the bar is the current frame of the input video. The horizontal slider shows at which position in the input video we are. Between the video frame and the slider is the cluster map. Below the slider is the key-frame strip.

Cluster Map

Figure 4.10: The cluster map

The cluster map purpose is to give an overview of the clustering result. Here it has to be clear how large the clusters are and what the quality of the key-frames of the cluster are.

The cluster map shows the size en relative key-frame quality of all clusters (see figure 4.10). The key-frame quality is relative to the quality of all clusters. The worst key-frame is in red and the best in green. All the other quality values are linearly interpolated between these colours in rgb colour space. This gives a quick overview of quality of the key-frames.

The size of a cluster is displayed by the length of the cluster on the map. Cluster borders have a black line between them to emphasize the start of a new cluster. The length of the cluster on the cluster map also correspondents to the position on the slider. The cluster the viewed video frames is in, is shown by drawing a full black border around the cluster on the cluster map. This means that both the border around the cluster and the slider indicate which cluster on the cluster map is the current cluster of the viewed frame.

Key-frame Strip

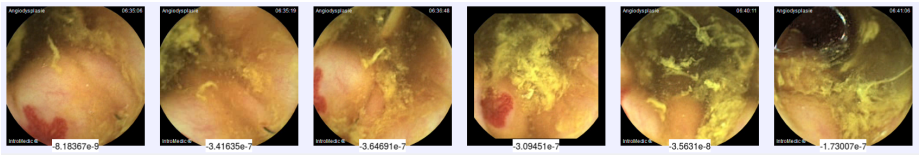

Figure 4.11: The cluster timeline

The purpose of the key-frame strip is to show give an indication of the quality of the key-frame. The tool is meant to be used together with the cluster map which gives an indication of the size of the cluster and quality relative to the other key-frames.

The key-frame strip shows the key-frame of the current cluster and the neighbouring key-frame as seen in figure 4.11. The key-frame of the current cluster if highlighted by drawing a black bar around the frame. The neighbouring key-frames are shown to give a visual representation of the distance to the neighbours.

Interaction

The program is also meant to be a navigational tool for Capsule Endoscopy. Therefore it should be easy to navigate to all key-frames and navigate to all input video frames. To accomplish this the program has three methods to navigate through the input video. The slider, the cluster map and the key-frame strip.

The slider represents all frames in the input video. It navigates through the input video like a slider of a normal video player. The user can also click on the clusters in the cluster map. If a user clicks on a cluster k in the cluster map, the program will navigate to the key-frame of cluster k. The last method to navigate is with the key-frame strip. The key-frame strip shows several frames of the input video. If the user clicks on one of these frames the program will navigate to the clicked frame.

Chapter 5

Conclusion

The goal of the thesis is to answer two questions. The first is "How can we summarise a long gastrointestinal video in a strip/timeline-like metaphor?". While testing with real users was beyond our resources, we can from our own testing conclude that we can summarise a gastrointestinal video in a timeline and strip-like metaphor. The timeline overview is visualised by the cluster map. On this map one can quickly get an overview of the broad regions of the gastrointestinal track when generating a small amount of cluster. When generating a large amount of clusters this map shows the parts of the video where there is not a lot of difference between frames and the parts of the video where there is a lot of difference. The key-frame strip at the bottom of the program also gives a quick overview of the current and surrounding clusters.

The second question was "How can we represent groups of related frames by single 'summary' frames". First the algorithm has to calculate the related frames by grouping the frames into clusters. Searching for the local maxima in distance between neighbouring frames seems to generate a stable amount of clusters across all test video. From this we can conclude that the method is a decent and predictable method of generating related frames. The related frames are then represented by a single centroid frame.

A large part of answering the first research question is done by implementing the program itself. So in the conclusion we will go over the requirements and how they were met.

1. **The product speed should be beneath 15 minutes on consumer grade hardware.** By picking a algorithm that has a linear time complexity in the number of frames. On a consumer grade dual core processor the algorithm takes around 2.5 minutes. Most the the speed is dependant on the frame size and the amount of frames. So when the camera quality of the capsule endoscopy increases the time it takes to calculate a result will greatly increase.

2. **There should be a balance between the accuracy and verbosity of the summarisation.** The accuracy of the key-frames is hard to measure

in any real statistic. But from the normalised average key-frame fidelity most of the parameter settings are below the video mean. And the decrease in quality is very linear. But that still does not give us a hard metric by which to measure key-frame quality.

The amount of frames generated is based on the parameters. But at $k = 2$ and $n = 1$ which are low settings the compression rate is still around 18%. Which means that even at the lowest compression the doctor still only has to go through five times fewer frames. The limiting factor in this is that if we increase compression the quality of the key-frames decreases a lot more then the increase in compression.

3. **There should be an overview of all clusters.** The cluster map gives a decent overview of all of the cluster while not overloading the user with information. The limiting factor here is that the key-frame quality is relative to the other key-frames and not a constant between all possible result and input video.

4. **Clusters should be easy to navigate.** The video can be navigated in two ways. The first is dragging the time slider where the user navigates trough the original video. The second is by navigating to the key-frames of cluster by clicking on the cluster map or the key-frame strip. The clustering therefore achieves its goal in making the navigation of the input video a lot easier.

5. **Clusters should be linked to the original video.** When navigating trough the original video the current cluster is displayed and when navigating to key-frames the timeline slider is set to the position of that frame.

5.1 Future Extension

For future research several parts can still be examined. The interface has not been tested by doctors yet. An interesting experiment would be to try different ways to visualise the clusters and the key-frames. Then let doctors test those and evaluate which options have the best usability.

For the algorithm it would be interesting to experiment with calculating distances based on other features. Interesting variations would be extracting texture feature vectors based on shapes in the frames. There are several methods for extracting a texture feature vector. Like the BEDD method discussed in the Related Works chapter.

Key-frame calculation is also an interesting part of the algorithm to experiment with. Here methods like relation matrix rank method are interesting to compare to the the k-means method.

A limiting factor when judging the strength of the key-frame calculation is that there is no real hard metric to measure key-frame quality. An interesting

research question would be to define a key-frame quality metric that works well with capsule endoscopy.

On a different note an other interesting way to approach video summarisation is to use machine learning to recognize patterns and divide the video in cluster based on those patterns.

Bibliography

[1] Z. E. khattabi, Y. Tabii, and A. Benkaddour", ""video summarization: Techniques and applications"," *"World Academy of Science, Engineering and Technology International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol:9, No:4"*, "2015".

[2] C. Gianluigi and S. Raimondo", ""an innovative algorithm for key frame extraction in video summarization"," *"Journal of Real-Time Image Processing"*, "2006".

[3] B. Li, M. Q.-H. Meng, and Q. Zhao, "Wireless capsule endoscopy video summary," in *IEEE International Conference on Robotics and Biomimetics*, 2010.

[4] B. Li, H. Jin, C. Yang, and G. Xu, "A novel color textural feature towards capsule endoscopy video summary," in *IEEE International Conference on Information and Automation Lijiang*, 2015.

[5] J. S. HUO and Y. X. ZOU", ""an advanced wee video summary using relation matrix rank"," in *"Proceedings of the IEEE-EMBS International Conference on Biomedical and Health Informatics"*, "2012".

[6] F. Tomita and S. Tsuji", *"Computer Analysis of Visual Textures"*. " London Kluwer", "1990".

[7] M. M. B. Ismail, O. Bchir, and A. Z. Emam", ""endoscopy video summarization based on unsupervised learning and feature discrimination"," in *"Visual Communications and Image Processing (VCIP)"*, "2013".