ÇANKAYA UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

Project Report
Version 1

CENG 407
Innovative System Design and Development I

P*2017-22*
Content Based Video Segmentation

Berk Can ÖZÜTEMİZ
201311049
Ece NALÇACI
201411040
Veli Engin ÖZTÜRK
201311048

Advisor
Prof.Dr. *Erdoğan DOĞDU*
Co-Advisor
Assist. Prof. Dr. Roya CHOUPANI

Table of Contents

Abstract

A video is a sequence of images named frames. Our project aims to divide a video into segments (shots) wherein the scene or camera position does not change significantly and our project will consider both visual and semantic contents of the frames. The frames are shot in a very short time interval from each other. Therefore, the consecutive frames are very similar in content. However, with changes in scene, story, etc. the frame contents may change very quickly. For instance, a video may start with an indoor scene, and then it may switch to some outdoor scene. The first step in video indexing is video segmentation into shots where each shot is manually or automatically labeled. These labels are later used for retrieving the videos or video segments which are looked for in a query. Multimedia Computing gained traction over the last two decades which makes it fairly new. While we mention that Multimedia Computing is new, it is highly complex because it investigates human visual system, networks and devices at the same time. Our challenge is to learn current coding standards, dealing with major changes which do not necessarily correspond to the end of a video segment while minor changes in the frames can be handled easily.

Key words:

Video, Video Segmentation, Content Based Video Segmentation

Özet:

Bir video, çerçeve adında görüntü dizilerinden oluşur. Projemiz, bir videoyu sahne veya kamera konumunun belirgin bir şekilde değişmediği bölümlere (çekimler) ayırmayı ve görsel ve semantik içeriklerini de dikkate alarak parçalamayı amaçlıyor . Çerçeveler birbirinden çok kısa bir zaman aralığında çekilir. Bu nedenle ardışık çerçeveler içerik açısından çok benzer. Bununla birlikte, sahnedeki değişiklikler, hikaye vb. çerçevelerin içeriği çok hızlı değişebilir. Örneğin, bir video kapalı bir sahne ile başlayabilir ve daha sonra bazı açık hava sahnelerine geçebilir. Video indekslemenin ilk adımı, her atışın manuel veya otomatik olarak etiketlendiği çekimlere video bölütlemesidir. Bu etiketler daha sonra, bir sorguda aranan videoları veya video bölümlerini almak için kullanılır. Çoklu Ortam Hesaplama, son yirmi yılda oldukça yeni kılan bir çekiş sağlamıştır. Çoklu Ortam Hesaplama'nın yeni olduğunu belirtmekle birlikte, insan görsel sistemi, ağları ve cihazları aynı anda araştırdığı için oldukça karmaşıktır. Zorluklarımız, mevcut kodlama standartlarını öğrenmek ve bir video parçasının sonuna karşılık gelmeyen önemli değişiklikleri ele almak ve çerçevelerdeki küçük değişiklikleri kolayca ele alınabilmektir.

Anahtar Kelimeler:

Video, Video Parçalama, İçerik Tabanlı Video Parçalaması

1. Introduction

1.1 Problem Statement

With the advancement in technology, multimedia is being widely used. Multimedia is the field concerned with the computer-controlled integration of text, graphics, drawings, still and moving images (Video), animation, audio, and any other media where every type of information can be represented, stored, transmitted and processed digitally. Multimedia Computing focuses on processing the data being created by the millions of people all around the world. Even with the current advancements in technology, Multimedia Computing is becoming more important every day because of the rapid growth of data. Most of the multimedia data such as videos, stored in compressed form. Video compression is that a video may be considered as a stream of frames taken consecutively at very short intervals and is viewed and understood by some end user(s) [1]. Many video codec (compression /decompression) industry standards and so many proprietary algorithms are available to make it practical to store and transmit video in digital form [2]. Due to the growth in multimedia information, an effective video indexing and video retrieval is necessary. Video segmentation is a way of dividing a video into meaningful segments. This can be achieved when effective video segmentation tools and algorithms are available. MPEG-compressed videos are mostly used by researchers for video segmentation. Shot boundary detection, color histogram characteristics, DC-images, motion vector and motion compensation, threshold-based detector, etc. are mostly used for video segmentation [3]. Video segmentation is a way of dividing a video into meaningful segments. Our aim is to implement a software which can segment a video based on its content after learning why and how the data is compressed, how a video file is compressed, how a video is segmented into shots and what are the latest approaches for content-based video segmentation.

1.2 Motivation

We are a group of senior students in computer engineering department who are interested in multimedia computing. As a group, we have taken the course of introduction to multimedia computing for a better understanding in multimedia field. We aimed to use the knowledge we obtained from the introduction to multimedia computing in this project. We have chosen MATLAB productive software environment which all of the members of the group are already familiar to develop our project because many mathematical instruments are needed for our project and we want to see the first results very fast.

1.3 Solution Statement

The fast growth of multimedia usage is increasing the need of better compression algorithms moreover the compressed data needs to be adaptable to different networks and devices. In order to predict frames, motion estimation and motion compensation will be performed. Cut scene detection will be performed. If the difference between consecutive frames are more than the rest of the frames, shots will be segmented. Temporal video segmentation will be performed based on DCT coefficients, based on DC terms, based on DC terms and MB coding mode, based on DCT coefficients, MB Coding Mode and MVs, based on MB coding mode and MVs, MB coding mode and bit rate information H.264 will be used as the video codec.

1.4 Contribution

Video segmentation software can be used in many applications. Content selection allows time and space to be reduced during the retrieval process. One of the advantage of our project is that, content based video segmentation allows to summarize a sport event, to skim a video, to mine the patterns of a video. Video surveillance can be done with content based video segmentation where the segmentation result can be used to identify a pattern. The software also helps to achieve a better quality by coding the most relevant objects at higher quality.

2. Literature Search

2.1 Video Coding

Moreover than a century, humankind always wanted to transmit visual images to a remote location. According to its definition [4], video is an electronic medium for the recording, copying, playback, broadcasting, and display of moving visual media. Digital videos are stored in a large amount of space or bandwidth, which makes them hard to distribute. Video coding is the process of compressing and decompressing videos. Video compression is used in moving digital images. According to Mrak et al. [5], distribution of high-resolution video content to a large number of users requires transmission of extremely large amounts of data. Video codecs has made it possible to reduce the size of content so that it can be stored and transmitted by compressing and decompressing digital videos. Video compression techniques generally require a transform process. A high quality video can not be visualized with many details in a low quality displaying device. Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) are used to separate important data from less important data [6]. Ahmet et al. [7] introduced block-based discrete cosine transform (DCT) in 1974. Sikora emphasizes that

DCT approach, which has been standardized with H.263, MPEG-1/2/4 and H.264 have further improved compression efficiency because of more complex decoders and encoders [8]. In most standards coding schemes, i.e., lossy JPEG, each 8x8 block of coefficients is quantized and coded into variables or fixed length binary code words [9]. DCT is used because of its high decorrelation performance and there are many fast DCT algorithms that is used for implementations. However, DCT coding will bring block-artefact distortion problems in low bit rates. Research has been made to make video compression algorithms more efficient, flexible and more robust against bit and packet errors. Most of the research focuses on recognition of meaningful semantic information in images for advanced motion prediction so the video coding research has moved from DCT to DWT. DWT coding provides improved coding compared to DCT. Shi and Sun proved that DWT enables in combination with embedded quantizers alongside with excellent compression efficiency so-called fine-granularity embedded coding functionalities fully integrated into the coder [10]. DWT is usually implemented in JPEG 2000 and MPEG-4 as a frame-based approach applied to entire images [11] and DWT enables block partitions [10]. DWT helps achieving high coding efficiency and quality scalability. However, DWT has not successfully been applied to video coding due to several difficulties [10] such as mismatches between block sizes reduce the inter-frame coding efficiency. We aim to use DWT in our project because according to Aziz and Dolly [12], DWT avoids blocking artifact, DWT allows good localization both in time and spatial frequency domain, DWT has inherent scaling and has better identification of which data is more relevant to human perception, which enables higher compression ratio, DWT has higher flexibility because wavelength function can be chosen freely.

2.2 Data Redundancy

If some parts of data are stored repeatedly, or can be derived from other parts, the data is said to be redundant. Even with the advancements in technology streaming video with adaptive quality is still not perfect. Videos include consequent frames (images). At least 16 frames per second are needed for a smooth video and generally, 30-35 frames per second are used in high quality videos. One frame of a high quality video can have 3112 rows and 4096 columns and each color pixel needs 24 bits. Considering, 30 frames are stored for 1 second video, that 1 second video takes more than 1 Gbytes which is more than the capacity of the CD. Storing and transmitting un-compressed video over the Internet (video on demand) is almost impossible. This is why data compression is needed. There are two approaches to video compression; Intra frame compression, which considers each frame of a video as a still image, and Inter-frame compression, which uses temporal redundancy predictions.

2.2.1 Intra-Frame Compression

Intra-frame compression is used to compress a still picture. There is only one frame and the data within that frame is analyzed for compression. If the individual frames are saved as separate files, they are called image sequences. An intra-frame codec is bunched as one file. There are redundancy types available in still images which are visual, spatial and stochastic redundancy. Visual redundancy helps to store less color information for each pixel because human's visual system is more sensitive to brightness than color. Spatial redundancy is used to reduce the stored data size because the pixels that are near to each other have similar colors. Stochastic redundancy is used to store less bits when a pixel color is used more than other colors.

2.2.2 Inter-Frame Compression

Inter-frame coding is the process of video compression using motion estimation. An image is two-dimensional. It has rows and columns of pixels. An intra-frame codec compresses two-dimensionally. An inter-frame codec compresses three-dimensionally. Inter-frame compression uses temporal redundancy predictions. The frames of a video are very similar if the time interval between them is short. Given the nature of any video sequence, the numerical values of the luminance/brightness and chrominance/color of all the pixels in a given frame is either exactly similar or at least near similar to those values in the previous frames. The objects in the first frame are slightly moved to a new place. Estimating the motion before comparing is the solution to the object displacement. Assuming a continuous video has 30 frames per second, time travel between two neighboring frame is approximately 33.3 milliseconds and similarity between them is so strong. By taking advantages of similarity and correlation within frames, we can predict next frame from its neighboring frames with the help of temporal redundancy method [3]. According to Liyin et al., the key to high performance of video compression lies in an efficient reduction of the temporal redundancy [13]. Although it decreases the data size perceptibly, it adds the complexity of motion detection [14]. The process of temporal redundancy has based on image is divided into small fixed (H.261, MPEG-1) [15] or variable sized blocks (H.263 [17], MPEG4 [15], H.264 [17]). Motion detection then is performed by finding the best match for the block in a window in the first frame centered at the block's position in the second frame [18]. Block motion estimation is adopted in many video compression standards from H.261 to H.264 to remove the temporal redundancy within the successive frames.

2.3 Motion Estimation and Motion Compensation

Until here, we mentioned redundancy types being used to compress data based on characteristics of still images. Motions of objects in the video should also be considered to compress a video. Motion estimation is the process of determining motion vectors that describe the transformation from one 2D image to another, usually from adjacent frames in video sequence [12]. Motion estimation and motion compensation aims to remove inter frame redundancy and it is core of current video coding standards, also it is used to find the most similar area in the reference frame to a block in the current frame. Short time interval between consecutive frames means that the contents are very close to each other and the only difference may be encoded and transmitted [19]. Aziz and Dolly [12] proposed a motion estimation and motion compensation video compression method using DCT and DWT which states a motion is block is chosen and observed pixel by pixel to understand the similarity. The most similar area is found in the reference frame using the motion vector. The more similarity ratio, the more compression rate. For each block, a vector called motion vector, shows the displacement needed to reach the most similar region in the reference frame. The compressed video can be decompressed by motion vector and the compressed video. The difference of image compression from video compression is the motion compensation. Motion compensation is algorithmic techniques used to predict a frame in a video. A reference picture is chosen, this picture can be from the previous pictures or from the future pictures to transform to the current picture. When images can be accurately synthesized from previously transmitted mages, compression efficiency can be improved [20].

2.4 Current Status of Video Coding Standards

The fast growth of multimedia usage is increasing the need of better compression algorithms moreover the compressed data needs to be adaptable to different networks and devices. Below you can find the list of organizations and current standards [3] for video coding. JPEG is used for only still images, but it is still possible to compress a video by only removing intra frame redundancy. It is not effective, but it is useful for some applications. The international standards organization (ISO) and the international telecommunications union

(ITU) have developed a series of standards that have shaped the development of the media industry. Popular ISO coding standards include MPEG- 1, MPEG-2 and MPEG-4. ITU-T has published the H.26x line of coding standards including H.261, H.262, H.263 and H.263+. We aim to use H.264 standard in our project.

Table 1. Video / Image Coding Standard

Video/Image Coding Standard		
Name	**Year**	**Major Features**
JPEG	1992	For still image coding, DCT based
JPEG2000	2000	For still image coding, DWT based
H.261	1990	For videoconferencing, 64 kbits/s-1.92 Mbits/s
MPEG-1	1991	For CD-ROM, 1.5 Mbits/s
MPEG-2 (H.262)	1994	For DTV/DVD, 2-15 Mbits/s; for ATSC HDTV, 19.2 Mbits/s; most extensively used
H.263	1995	For very low bit rate coding, below 64 kbits/s
MPEG-4 Part 2	1999	For multimedia, content based coding, its simple profile and advanced simple profile is applied to mobile video and streaming
H264/AVC (MPEG-4 Part 10)	2005	For many applications with significant improved coding performance over MPEG-2 and MPEG-4 Part 2
VC-1	2005	For many applications, coding performance close to H.264
RealVideo	1997	For many applications, coding performance similar to MPEG-4 Part 2
MPEG-7	2000	Content description and indexing
MPEG-21	2002	Multimedia framework

2.4.1 MPEG Standards

Motion Picture Experts Group (MPEG) is the most commonly used standard for video coding. MPEG defines three frame types; I Frame, B Frame and P Frame. The first frame is treated as a still image and this frame is called I frame. I frames are encoded independently. I frame is the reference frame for the first P frame. In B frames, each block is compared to its previous and next frames because the interval time is short. B frames give smaller coefficients than P frames because B frames takes one difference into consideration. B frames are not used as reference frames. In P frames, the frame is split into macro blocks. The pixel colors are converted into YCbCr. Macro blocks are down sampled. For each block after down sampling, the most similar region in the previous frame is found. The difference of the block with the most similar region is computed and DCT transformed. The DCT coefficients are quantized, and entropy is encoded. P frame's first previous frame is I frame which is independently compressed. Frames of a video are grouped as one. I frame is followed by some P or B frames. The sequence can change according to the compression. This is called a Group Of Pictures. The MPEG standards has different parts, which have different properties: MPEG-1, MPEG-2, MPEG-3 and MPEG-4. MPEG-1 is optimized for coding of video and for digital storage media. MPEG-1 video compression is built on the same technique that is used in JPEG, it also contains schemes for well-organized coding of a video sequence. With MPEG-4, video compression can support lower bandwidth consuming applications. MPEG-4 initiative is addressing generic, integrated video communication. The difference of MPEG-2 from MPEG-4 does not associate to video coding or surveillance applications. Encoding is implied with key frames via JPEG algorithm and the motion changes between these key frames. Compression ratios above 100:1 are common. MPEG-3 was fused into MPEG-2 and it no longer exists. The fundamental method is to predict motion from frame to frame in the temporal direction, and then to use discrete cosine transforms to categorize the redundancy.

2.4.2 H26+ Standards

The video coding standards (H.261, H.263. H.26+) that was proposed by ITU have been targeted at real-time, point-to- point communications. H.261 has and continues to be used in many video conferencing applications. H.261 has two type of frames; I frames and P frames. I frame provides an access point. P frames uses predicted frames from previous frames. H.263 has developed in need of performance and video coding techniques. H.263 provided better compression performance, better quality and more flexibility by making improvements to H.261. H.262 or also known as MPEG-2 Part 2 is used for encoding the compressed video and audio data multiplexed with signaling information in a serial bit stream. H.262 provides support for interlaced videos. H.262 is not an optimal standard for low bit rates but it is better than MPEG-1. H.264/AVC was finalized in March 2003 and approved by the ITU-T in May 2003 [21]. H.264 is used for high definition videos. H.264 standard enhanced compression ratio and established video representation between conversational and non-conversational applications. Nemþiü et. al. claims that [22] H.264 has improved coding efficiency by a variable block-size motion compensation with the block size as small as 4x4 pixels, motion vectors over picture boundaries, multiple reference picture motion, multiple reference picture motion compensation, in the loop de-blocking filtering, quarter sample motion vector accuracy and enhanced entropy coding. H.264 has improved robustness to data errors and losses during transmissions over different networks. Three profiles are supported in the H.264 standard. The Baseline profile is suitable for conversational applications such as video-conferencing. The Extended profile has additional tools and can be used for video streaming applications. The Main profile is suitable for video broadcast and video storage. H.264 has improved rate-distortion efficiency relative to existing standards [23]. The more redundancy, the higher the quality at any given bit rate so H.264 streams include three types of frames; I frames, P frames and B frames. I frames don't use information from any other frames because they are also known as key frames. I frames have the highest quality but least efficient for compression. P frames are more efficient than I frames but less efficient than B frames. P frames are also called predicted frames because they use information from previous I or P frames. B frames are produced by both information from previous or next frame. H.264 uses YCbCr sampling as in prior standards. A picture is partitioned into fixed-size macroblocks of 16x16 luma components and 8x8 samples of the two chroma components. These macroblocks are processed in so called slices whereas a slice is usually a group of macroblocks processed in raster scan order [21]. All macroblocks are predicted and the resulting prediction is encoded using transform coding. The transformation is applied to 4x4 blocks, and instead of a 4x4 DCT, a separable integer transform with similar properties as a 4x4 DCT is used. Entropy coding is applied. H.264/AVC defines an adaptive in-loop-deblocking filter, where the strength of filtering is controlled by the values of several syntax elements [24].

2.5 Content Based Video Segmentation

Multimedia information systems generate and use large collections of video data. In order to provide these needs, appropriate index, search, browse and retrieve relevant material is necessary. Several content based retrieval systems for organizing and managing video databases have been proposed [25].

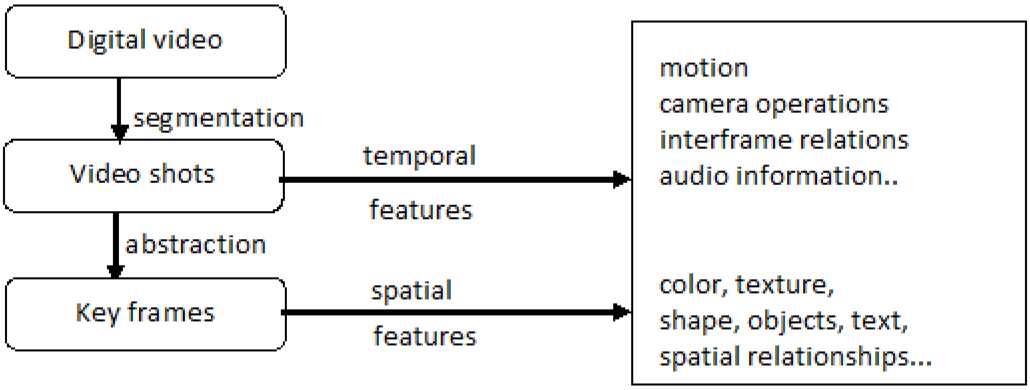

Figure 1. Content-based retrieval of video databases

As shown in Fig.1, in order to represent digital video sequences in automatic annotations, temporal video segmentation is used. Temporal video segmentation aims to divide the video stream into a set of meaningful and manageable segments called shots that are used for indexing. Each shot is represented by selecting key frames and indexed by extracting spatial and temporal features. The retrieval is based on the similarity between the feature vector and already stored video features. According to Koprinska and Carrato [26], a shot is defined as an unbroken sequence of frames taken from one camera. Shot transitions can be abrupt or gradual. Abrupt transitions, which are often called cuts occur in a single frame when stopping and restarting the camera. Gradual transitions are the combination between cinematic effects that are applied to abrupt transitions artificially. A fade out is means there is a slow decrease in brightness resulting in a black frame. A fade in is a gradual increase in intensity from a black image. Dissolves indicate that the first shot is getting dimmer and second shot is getting brighter. Gradual transitions are more difficult to detect than cut because they have to be distinguished from camera operations. Also the movement of the objects can cause false positives.

Ananger and Little [27] presented a survey in video indexing, including some techniques for temporal video segmentation mainly in uncompressed domain. Idris and Panchanathan [28] surveyed methods for content-based indexing in image and video databases focusing on feature extraction. A review of video parsing is presented but it mainly includes methods that operate on uncompressed domain and detect cuts. Early work focus on cut detection, while more recent techniques deal with the harder problem which is gradual transitions detection. Nowadays video is stored in compressed format and methods that can operate directly on the encoded stream are necessary. Working in the compressed domain helps to perform removes the need to perform decoding and encoding, reduce the computational complexity, saves decompression time, makes operations faster due to the lower data rate and the encoded video stream already contains motion vectors and block averages that are suitable for temporal video segmentation.

Table 2. Six groups of approaches for temporal video segmentation in compressed domain based on the information used

Group						
Information Used	[29]	[30]	[31]	[32]	[33]	[34]
DCT coefficients	✓			✓		
DC terms		✓	✓			
MB coding mode			✓	✓	✓	✓
MVs			✓	✓	✓	
Bit-rate						✓

As shown in table 2, several algorithms for temporal video segmentation in the compressed domain have been reported. This algorithms can be divided into six groups; segmentation based on DCT, DC terms, macroblock (MB) coding mode and motion vectors (MV), DCT coefficients, MB coding mode and MVs and MB coding mode and bit rate information.

2.5.1 Temporal Video Segmentation Based on DCT coefficients

Arman et al. [29] proposed a technique for cut detection based on the DCT coefficients of I frames in compressed domain. For each frame a subset of the DCT coefficients of a subset of the blocks is selected in order to construct a vector which represent the frame from the video sequence in the DCT space. Temporal video segmentation based on DCT coefficients can only be applied to I frames of the compressed video because they are the frames fully encoded with DCT. This enables the processing time to reduce. However, temporal resolution is low and the loss of resolution between I frames false positives can occur and the classification accuracy can decrease. This algorithms cannot handle gradual transitions or false positives.

2.5.2 Temporal Video Segmentation based on DC terms

DC terms are directly related to the pixel domain. DC terms reconstructs P and B frames. Yeo and Liu [30] propose a method where DC-images are created and compared. DC-images are spatially reduced versions of the original images. B and P frames are estimated using the motion vectors and DCT coefficients of the previous I frames. The reconstruction technique is computationally expensive. Shots are detected by color histogram comparison of DC term images which are formed by the DC terms of the DCT coefficients for a frame. Histogram based techniques may miss shot transition if the luminance distribution of the frames do not change significantly. The temporal resolution is low hence the exact shot boundaries cannot be labeled.

2.5.3 Temporal Video Segmentation Based on DC Terms and MB Coding Mode

Meng et al. [31] propose a shot boundaries detection algorithm based on the DC terms and the type of MB coding. DC components only for P frames are reconstructed. Gradual transitions are detected by calculating the variance of the DC term sequence for I and P frames. If there is a cut on a P frame, the encoder cannot use many MBs from the previous frame for motion compensation and as a result many MBs will be coded intra. On the other hand, if there is a cut on a B frame, the encoding will be mainly backward. There can be cut on an I frame only if the intensity variance of the frames during a shot is stable or a false positive occur due to motion.

2.5.4 Temporal Video Segmentation Based on DCT Coefficients, MB Coding Mode and MVs

Zhang et al. [32] proposed a two pass approach. First the regions of potential transitions, camera operations and object motions are located by applying the pair-wise DCT coefficients comparison of I frames. The goal of the second pass is to refine and confirm the break points detected by the first pass. By checking the number of MVs for the selected areas, the exact cut locations are detected. Gradual transitions are found by an adaptation of the twin comparison algorithm utilizing the DCT differences of I frames.

2.5.5 Temporal Video Segmentation Based on MB Coding Mode and MVs

In [33] cuts, fades and dissolves are detected only using MVs from P and B frames and information about MB coding mode. The system follows a two- pass scheme and has a hybrid rule-based/neural structure. During the rough scan peaks in the number of intra coded MBs in

P frames are detected. They can be sharp or gradual with specific shape and are good indicators of abrupt and gradual transitions, respectively.

2.5.6 Temporal Video Segmentation Based on MB Coding Mode and Bit-rate Information

According to Feng et.al[34], a simple and effective is approach is proposed although it is limited only to cut detection. This approach uses the bit-rate information at MB level and the number of various motion predicted MBs. A large change in bit-rate between two consecutive I or P frames indicates a cut between them. The number of backward predicted MBs is used for detecting cuts on B frames. The algorithm is able to locate the exact cut locations. First it locates a suspected cut between two I frames then between P frames of the group of pictures and finally it locates B frames if necessary.

3. Software Requirements Specification

3.1 Introduction

3.1.1 Purpose

The purpose of this document is to give general description about Content Based Video Segmentation. The purpose of Content Based Video Segmentation is to divide a video into segments (shots) wherein the scene or camera position does not change significantly and our project will consider both visual and semantic contents of the frames. This document includes detailed information about requirements, attributes, demands, system constraints, user characteristics, software and hardware requirements, interfaces, project development methodologies and other needed elements of the project.

3.1.2 Scope of Project

Nowadays, on the television screens, the tablets and the monitors of the mobile phones can reach 1920x1080p (Full HD) resolution, we see that this resolution sets the today's standards. However, the development in the technology is trying to add new standards and superior possible visuals to technological devices. Thinking that there are 2 million pixels on each frame of each video, and 24 bites that allow each pixel to get 16 million different colors data redundancy must be considered because a great deal of data is tried to be handled. This long video footage and videos that contain many data are made up of multiple scenes. Short time interval between consecutive frames means that the contents should be very close to each other and only the difference may be encoded and transmitted [1]. Li and Ngan stated that advanced multimedia applications led to content-related functionalities such as searching and retrieving meaningful objects, detecting, analyzing and understanding the scenes to allow user to access and manipulate the multimedia content with more flexibility [2]. Content based means that the search will analyze the actual content of the video. With the development in multimedia computing and available bandwidth, people are in a demand of video retrieval systems. Content selection allows time and space to be reduced during the retrieval process. Video segmentation can be used in many applications. Content based video segmentation allows to summarize a sport event, to skim a video, to mine the patterns of a video. Video surveillance can be done with content based video segmentation where the segmentation result can be used to identify a pattern. Content based video segmentation is also used in videoconferencing, it helps to achieve a better quality by coding the most relevant objects at higher quality. Content Based Video Segmentation project will segment video into meaningful shots which allows the users to watch a specific part of the video they want.

3.1.3 Glossary

Table 3 Glossary of SRS

Term	Definition
End user	Person who receives the segmented the video
Stakeholders	Any person who has contribution in the project.
MPEG	Video Coding Standard
Publish	Adding, creating, viewing, reading, changing, updating, removing and deleting the content
System	Content Based Video Segmentation
H.264	Video Compression Standard

3.1.4 References

[1] M. Tolun, R. Choupani, S. Wong "Video Coding and Transcoding: A Review". Retrieved November, 2017 Available: http://www.ce.ewi.tudelft.nl/choupani/

[2] H. Li and K. Ngan, "Automatic video segmentation and tracking for content-based applications", *IEEE Communications Magazine*, vol. 45, no. 1, pp. 27-33, 2007.

3.1.5 Overview of Document

The second part of the document describes functionalities of the Content Based Video Segmentation. Informal requirements are described and it is a context for technical requirement specification in the Requirement Specification chapter.

Specific requirements chapter mentions the system's performance, functional, non-functional analysis and attributes of the Content Based Video Segmentation software.

3.2 Overall Description

3.2.1 Product Perspective

Content Based Video Segmentation software aims to divide a video into segments (shots) wherein the scene or camera position does not change significantly and our project will consider both visual and semantic contents of the frames. It also deals with major changes which do not necessarily correspond to the end of a video segment while minor changes in the frames can be handled easily. Our project aims to divide the video into meaningful shots automatically.

3.2.2 Development Methodology

For developing the project, we have planned to use Scrum which is an agile software development methodology. Agile development methodology minimizes the risk by developing software in short time boxes called iterations. The scope of the next release can be quickly determined in agile development methodology. Scrum is time boxed and each time box, which is called a sprint, has a fixed length and the length of every sprint should be equal to each other. Sprints can last one to two weeks. At the end of each sprint, stakeholders and team members meet to plan next steps. Scrum teams should try to become a feature team, able to produce a workable product to deliver to customer after each sprint [35]. Scrum allows more transparency because project is visible. The accountability of the team is increased. It is easy to accommodate changes and cost savings are increased. In Scrum development methodology, there are specific roles such as product owner, scrum master and a scrum team. Product owner focuses on requirements and prioritize all the work. Scrum master helps the team to make their possible work. Scrum master doesn't have authority over the team but can organize meetings. Scrum team consists of developers who complete the set of work together. Scrum team has daily stand up meetings to share work intel which takes about 10 to 15 minutes. End user is the main actor.

End user decides which video to be segmented, choose the duration of the video which will be segmented (interval of the video time), test the results of different segmentation methods and list the data set.

3.2.3 User Characteristics

End user is the main actor. End user decides which video to be segmented, choose the duration of the video which will be segmented (interval of the video time), test the results of different segmentation methods and list the data set.

3.3 REQUIREMENTS SPECIFICATION

3.3.1 External Interface Requirements

3.3.1.1 User Interfaces

The user interface will work as a desktop application.

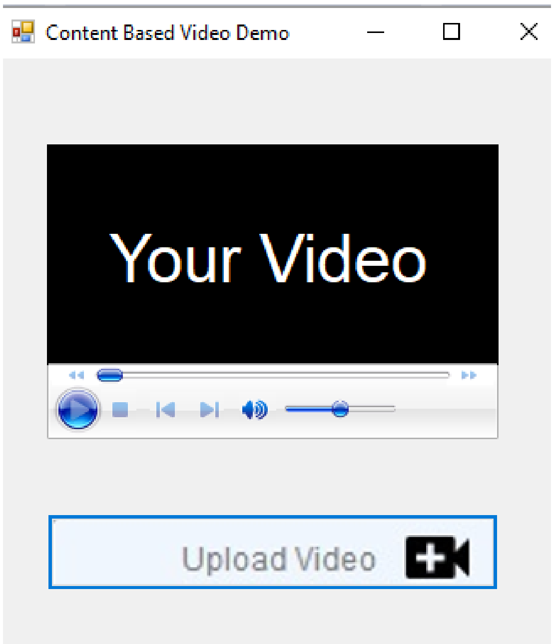

Figure 2: Uploading a video

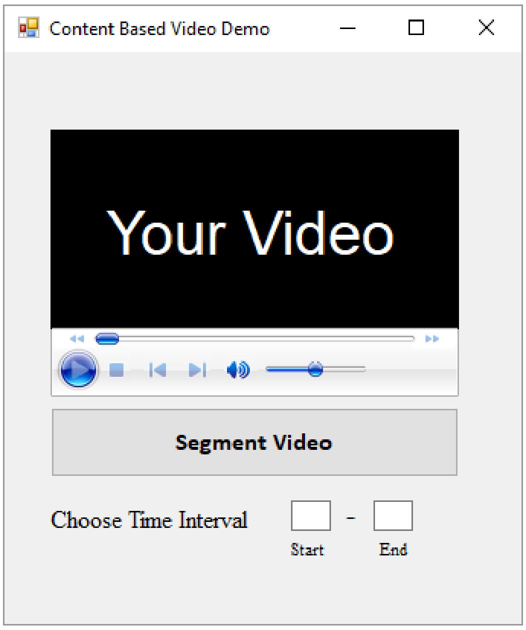

Figure 3: Choosing time interval

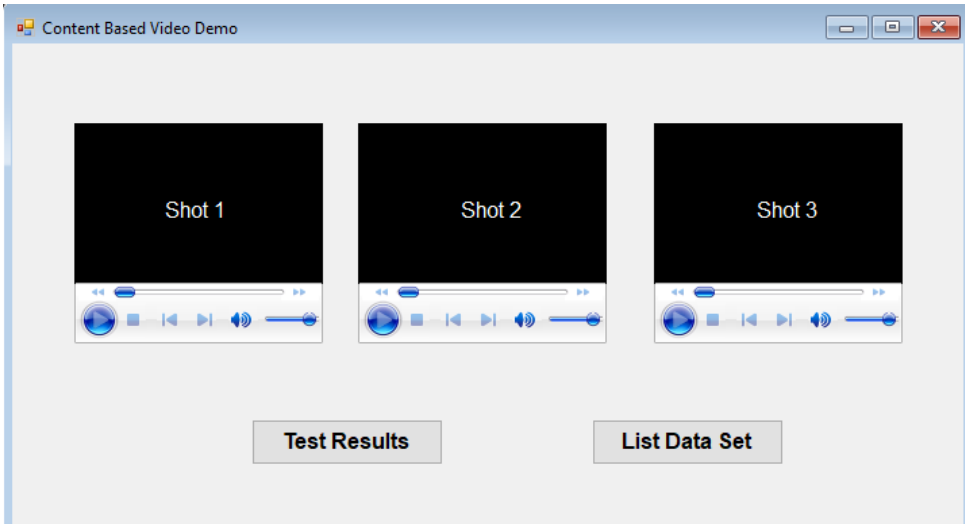

Figure 4: Segmented shots

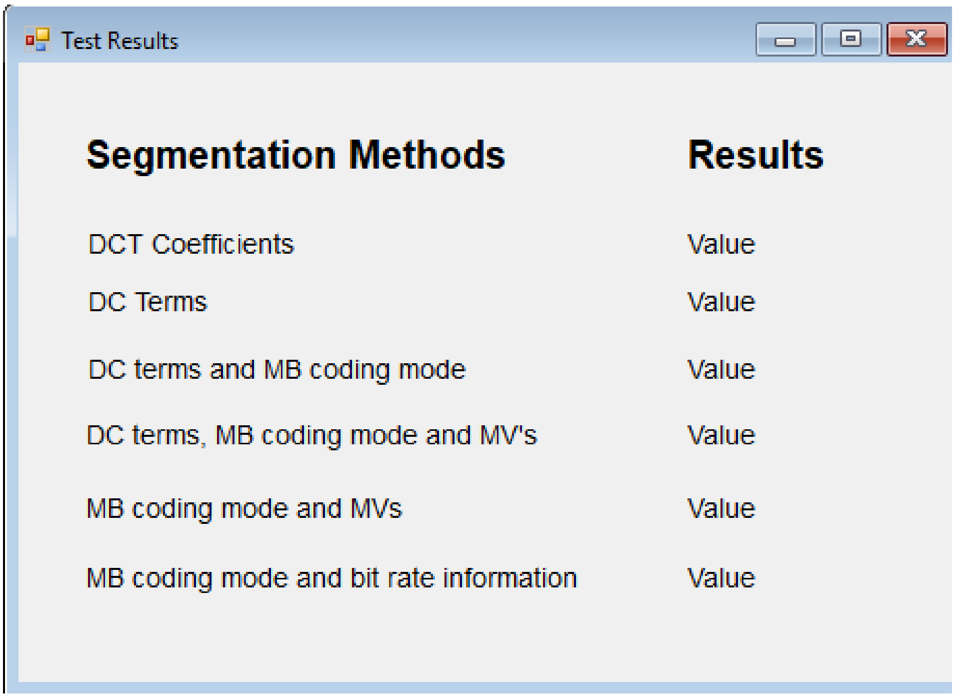

Figure 5: Test Results

3.3.1.2 Hardware Interfaces

Content Based Video Segmentation will run on desktop and all operating systems.

3.3.1.3 Software Interfaces

Codecs are used to obtain perfect quality video encoding. We prefer to use MPEG-2 and H.264. End user can use any kind of media player (such as Windows Media Player and VLC media player) and any operating systems.

3.3.1.4 Communications Interfaces

There are no external communications interface requirements.

3.3.2 Functional Requirements

3.3.2.1 Content Based Video Segmentation Use Case

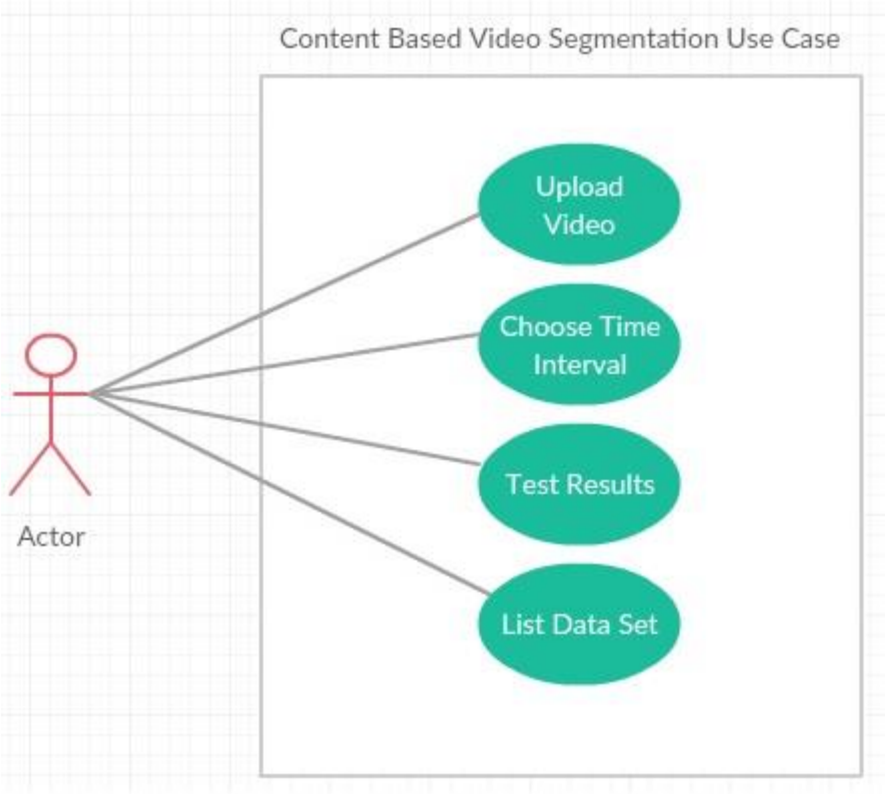

Figure 6: Content Based Video Segmentation Use Case

In Figure 5 it is shown that end user can upload a video, choose shot length, segment video, test results, and list data sets.

3.3.2.2 Upload Video Use Case

<u>Use Case:</u>

- Upload Video

Diagram:

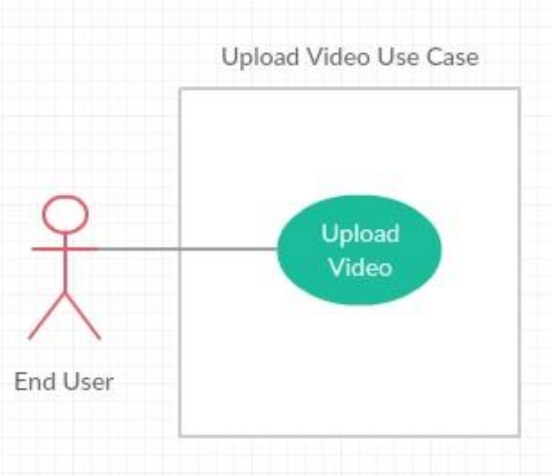

Figure 7: Upload Video Use Case

Brief Description:

In Upload Video Use Case diagram (Figure 6) explains one of the basic operations end user can do. End user can upload any video he or she wishes to the desktop application.

Initial Step by Step Description:

1. End user shall upload any video to start segmentation process.

2. End user shall test the result of the algorithms by comparing the results to temporal video segmentation based on DCT coefficients, based on DC terms, based on DC terms and MB coding mode, based on DCT coefficients, MB Coding Mode and MVs, based on MB coding mode and MVs, MB coding mode and bit rate information. End user shall determine the best algorithm by testing results.

3.3.2.3 Choose Time Interval Use Case

Use Case:

- Choose Time Interval

Diagram:

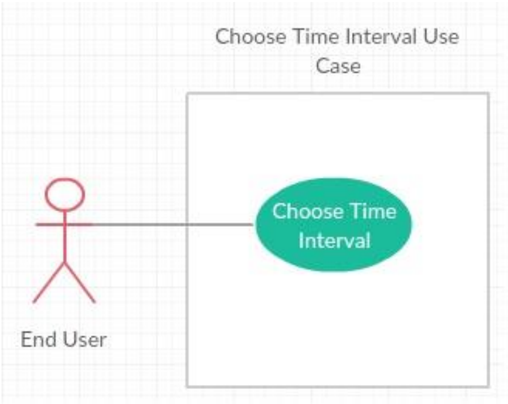

Figure 8: Choose Time Interval Use Case

Brief Description:

In Choose Time Interval diagram (Figure 8) explains one of the basic operations end user can do. End user can choose time interval.

Initial Step by Step Description:

1. User shall choose time interval in order to indicate the start and end time of the video to be content based segmented.

3.3.2.4 Test Results Use Case

Use Case:

- Test Results

Diagram:

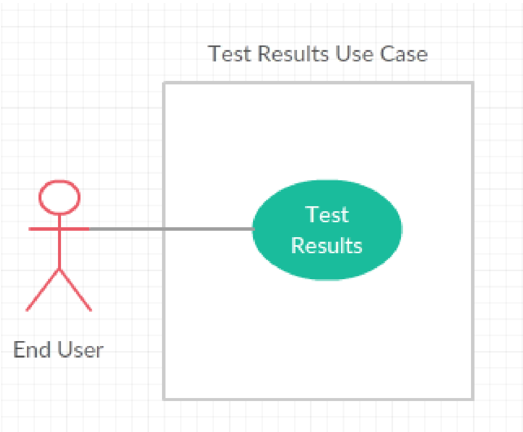

Figure 9: Test Results Use Case

Brief Description:

In Test Results Use Case diagram (Figure 9) explains one of the basic operations end user can do. End user can test results and compare the results with other segmentation methods.

Initial Step by Step Description:

1. End user shall test the result of the algorithms by comparing the results to temporal video segmentation based on DCT coefficients, based on DC terms, based on DC terms and MB coding mode, based on DCT coefficients, MB Coding Mode and MVs, based on MB coding mode and MVs, MB coding mode and bit rate information. End user shall determine the best algorithm by testing results.

3.3.2.5 List Data Set Use Case

Use Case:

- List Data

Diagram:

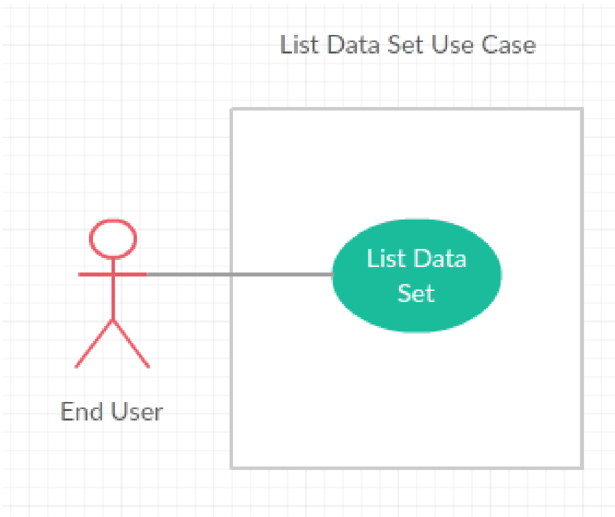

Figure 10: List Data Set Use Case

Brief Description:

In List Data Set Use Case diagram (Figure 10) explains one of the basic operations end user can do. End user can list data set, choose one of them and use it in different methods to compare.

<u>Initial Step by Step Description:</u>

1 .End user shall list the data set, shall choose one of the data and shall compare with the other methods to compare and shall check the accuracy of the result.

3.3.3 Non-Functional Requirements

3.3.3.1 Time

Processing speed is important. System should cause minimum time delay while presented to end user after content-based video segmentation. When the software is segmenting the video according to its content, time consumption should be minimum.

3.3.3.2 Image Quality

Video segmentation in compressed domain must maintain contents and visual quality of the frames. Segmenting videos based on content can cause pixel corruptions. If we use traditional decode-process-encode approach, lossy algorithms will degrade quality in every decode process, so that system works on compressed domain. That is why the system does segmenting on compressed domain to protect image quality.

3.3.3.3 Compatibility with H.26+ Standards

Compressed video frames need to be able to adapt to different networks and devices. In addition to this we need to comply with video coding standards such as H.263 and H.264. H.264 brought flexibility to the application, efficiency to coding and robustness to changes [36].

3.3.3.4 Compatibility with MPEG Standards

For broadcasting videos, MPEG 2 and MPEG-4 are widely used for networks with low bit rates. MPEG provides better video quality and higher resolution. Moreover, MPEG-2 and MPEG-4 standards possess more complex decoding algorithms. MPEG-4 supports video compression in lower bandwidth consuming applications [37].

3.3.3.5 Performance

The performance will be determined according to the number of shots.

4. Software Design Description

4.1 Introduction

4.1.1 Purpose

The purpose of this Software Design Document is providing the details of project titled as “Content Based Video Segmentation”.

The purpose of Content Based Video Segmentation is to divide a video into segments (shots) automatically wherein the scene or camera position does not change significantly and our project will consider both visual and semantic contents of the frames. Short time interval between consecutive frames means that the contents should be very close to each other and only the difference may be encoded and transmitted [38]. Video segmentation can be used in many applications. Content selection allows time and space to be reduced during the retrieval process. Content based video segmentation allows to summarize a sport event, to skim a video, to mine the patterns of a video. Video surveillance can be done with content based video segmentation where the segmentation result can be used to identify a pattern. Content based video segmentation is also used in videoconferencing, it helps to achieve a better quality by coding the most relevant objects at higher quality. Content Based Video Segmentation project will segment video into meaningful shots which allows the users to watch a specific part of the video they want.

In order to provide a better comprehension, this Software Design Document includes various diagrams such as activity diagram and block diagram.

4.1.2 Scope

This document contains a complete description of the design of Content Based Video Segmentation.

MATLAB is a high-performance language for technical computing. MATLAB stands for matrix laboratory. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation [39]. MATLAB can be used to develop algorithms, modeling, prototyping, scientific and engineering graphics, application development, graphical user interface building, data analysis, visualization and computation. MATLAB provides integration of other programming languages. MATLAB can be called with C/C++, Java, Fortran and Phyton languages.

Coding part of the project is will be written with C language. C has ready access to the hardware when needed, the compiler is portable, the library is standard and applications can be optimized by hand-coding procedures. The reasons why we have chosen C as our programming language are all the members of the group have knowledge of C programming language and C is one of the programming languages which can be used in MATLAB. C/C++ code can be converted into MATLAB algorithms by using MATLAB Coder [40].

End user shall navigate through graphical user interface. End user shall upload a video, choose time interval and the video will be content based segmented during the time interval which end user specified. Motion estimation and motion compensation will be performed. Cut scene detection will be performed. If the difference between consecutive frames are more than the rest of the frames, shots will be segmented. The user shall test result with other segmentation methods and shall list data to check accuracy of the results. Temporal video segmentation will be performed based on DCT coefficients, based on DC terms, based on DC terms and MB coding mode, based on DCT coefficients, MB Coding Mode and MVs, based on MB coding mode and MVs, MB coding mode and bit rate information H.264 will be used as the video codec. H.264 is preferred because H.264 brought flexibility to the application, efficiency to coding and robustness to changes [41].

4.1.3 Glossary

Table 4 Glossary of SDD

Term	Definition
Block Diagram	The type of schema which the components in the system are displayed in blocks.
SDD	Software Design Document.
End User	Person who receives the segmented the video.
DCT	Discrete Cosine Transform
MB	Macro Block
MV	Motion Vector
H.264	Video Coding Standard

4.1.4 Overview of document

The remaining chapters and their contents are listed below.

Section 2 is the Architectural Design which describes the project development phase. Also it contains architecture design of the Content Based Video Segmentation which describes actors, exceptions, basic sequences, priorities, pre-conditions and post conditions. Additionally, this section includes activity diagram of the project.

Section 3 is Use Case Realization. In this section, a block diagram of the system, which is designed according to use cases in SRS document, is displayed and explained.

4.2 Architecture design

4.2.1 Content Based Video Segmentation Design Approach

For developing the project, we have planned to use Scrum which is an agile software development methodology. Agile development methodology minimizes the risk by developing software in short time boxes called iterations. The scope of the next release can be quickly determined in agile development methodology. Scrum is time boxed and each time box, which is called a sprint, has a fixed length and the length of every sprint should be equal to each other. Sprints can last one to two weeks. At the end of each sprint, stakeholders and team members meet to plan next steps. Scrum teams should try to become a feature team, able to produce a workable product to deliver to customer after each sprint [42]. Scrum allows more transparency because project is visible. The accountability of the team is increased. It is easy to accommodate changes and cost savings are increased. In Scrum development methodology, there are specific roles such as product owner, scrum master and a scrum team. Product owner focuses on requirements and prioritize all the work. Scrum master helps the team to make their possible work. Scrum master doesn’t have authority over the team but can organize meetings. Scrum team consists of developers who complete the set of work together. Scrum team has daily stand up meetings to share work intel which takes about 10 to 15 minutes. By taking into consideration of these facts, Scrum is the most suitable methodology for the project.

4.2.2 Architecture Design of Simulation

4.2.2.1 Upload Video

Summary: This system is used by end user. End user can upload can any video according to his or her choice by clicking upload video button.

Actor: End User

Precondition: End user must run the program.

Basic Sequence:

1. User shall open the program.
2. User shall click the “Upload Video” button.
3. User can choose any video from his/her documents to upload.

Exception: None

Post Conditions: None

Priority: Low

4.2.2.2 Choose Time Interval

Summary: End user can choose the time interval of the video he/she uploaded to the system to be content based segmented.

Actor: End User

Precondition: End user must upload a video.

Basic Sequence:

1. End user can choose the time interval.
2. End user can state the start and end time of the video which he/she wants to be segmented.
3. End user can choose the whole video if he/she doesn’t want to state a time interval.
4. End user can click “Segment Video” button to start the segmentation process.

Exception: None

Post Conditions: None

Priority: High

4.2.2.3 Test Results

Summary: End user can compare the results of different segmentation methods by clicking “Test Results” button.

Actor: End User

Precondition: End user must upload a video and choose the time interval to be segmented and must click to "Segment Video" button.

Basic Sequence:

1. End user can view the shots which are automatically content based segmented.
2. End user can click "Test Results" button to view the results of other segmentation methods.
3. End user can view temporal video segmentation based on DCT coefficients, based on DC terms, based on DC terms and MB coding mode, based on DCT coefficients, MB Coding Mode and MVs, based on MB coding mode and MVs, MB coding mode and bit rate information.
4. End user can determine the best algorithm by comparing the results.

Exception: None.

Post Conditions: None.

Priority: Medium.

4.2.2.4 List Data Set

Summary: End user can list data set, choose one of them and use it in different methods to compare by clicking "List Data Set" button.

Actor: End User

Precondition: User must click "Segment Video" button.

Basic Sequence:

1. End user can list data set.
2. End user can choose data set.
3. End user can use the data set to compare the results with other segmentation methods.
4. End user can check the accuracy of the results.

Exception: None.

Post Conditions: None.

Priority*:* Medium**.**

4.2.3 Activity Diagram

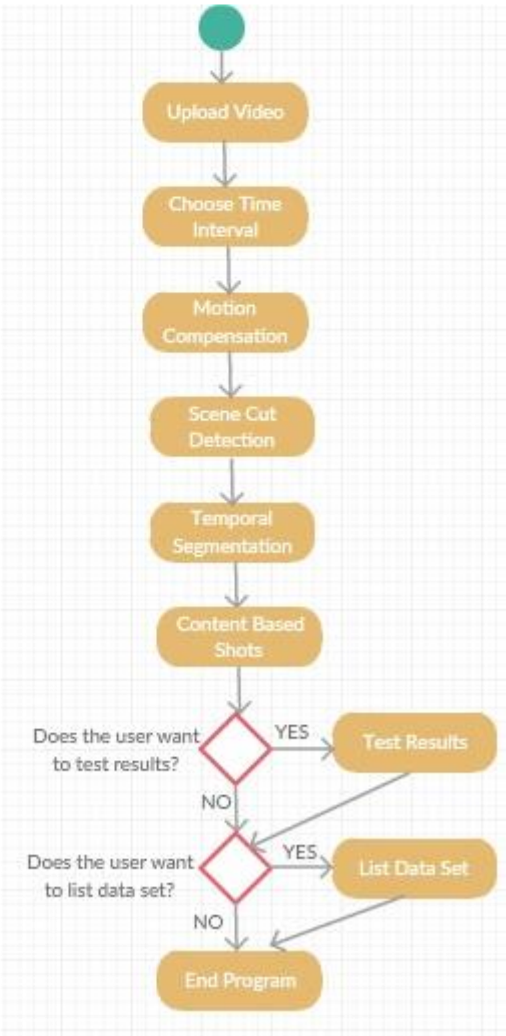

Figure 11: Activity Diagram of Content Based Video Segmentation

Figure 11 shows how the Content Based Video Segmentation works as an activity diagram. The end user uploads the video which he/she wants to segment. End user decides the time interval which he/she wants the video to be content based segmented. Motion compensation and motion estimation are performed. Scene cut detection is performed. Temporal segmentation is performed. If the user wants to test the result with other segmentation methods, the results from other segmentation methods will be shown to user. If the user wants to use the data list to check accuracy, the results of data list will be shown to user.

4.3 USE CASE REALIZATIONS

Content Based Video Segmentation Project

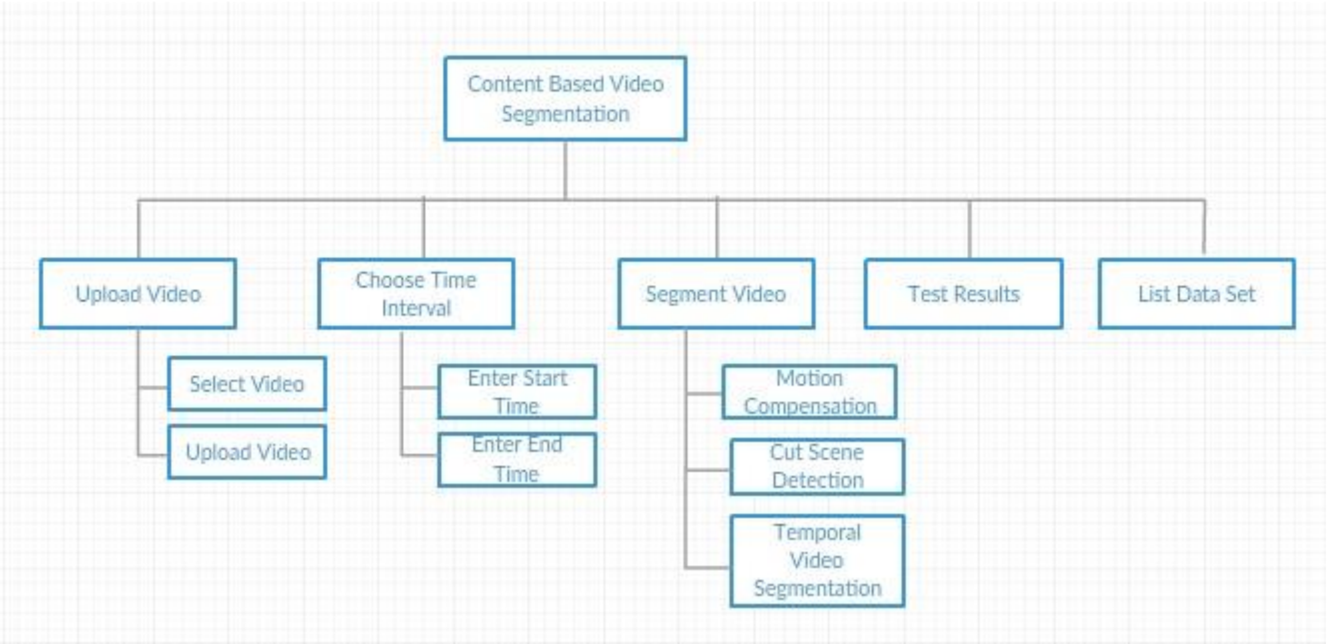

Figure 12: Functional Decomposition Diagram of Content Based Video Segmentation

4.3.1 Brief Description of Figure 12

Functions of Content Based Video Segmentation are shown in figure 12. All designed systems of the project are displayed in the block diagram in the figure. There are five main components of the system which have their own sub-systems.

4.3.1.1 Upload Video

Upload video is responsible for the user to upload a video in order to segment the video. It has two sub-systems which are select video and upload video. User can select and upload a video of his/her choice.

4.3.1.2 Choose Time Interval

Choose Time Interval is responsible for the user to choose time interval in order to indicate the start and end time of the video to be content based segmented.

4.3.1.3 Segment Video

Segment Video is responsible for segmenting the video. Temporal video segmentation based on DC Terms and MB coding mode method is used. In this method, only DC components for P frames are reconstructed. Gradual transitions are detected by calculating the variance of the DC term sequence for I and P frames and by looking for parabolic shapes in this curve. Cuts are detected by the computation of the following three ratios:

$$R_f = \frac{\text{forward}}{\text{backward}},\ R_p = \frac{\text{intra}}{\text{forward}},\ R_b = \frac{\text{backward}}{\text{forward}}$$

where, intra, forward and backward are the number of MBs in the current frame. If there is a cut on P frame, the encoder can't use MBs from previous frame for motion compensations and MBs is be coded intra. If there is a cut on B frame, encoding will be backward.

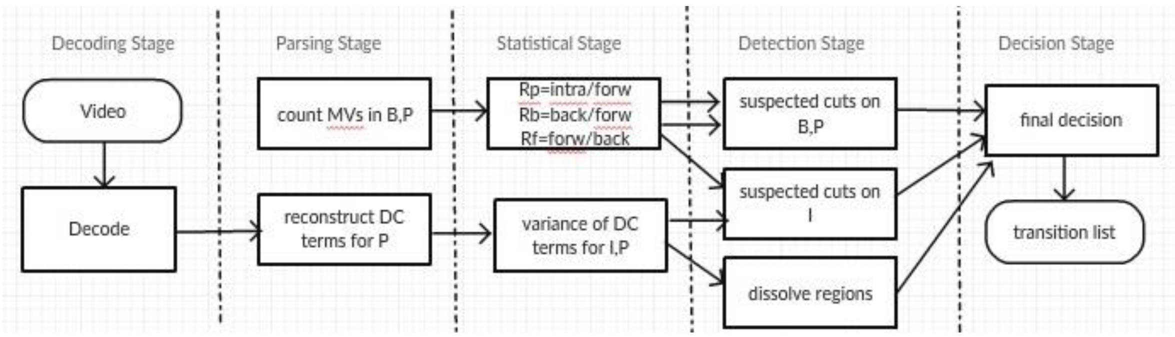

Figure 13: Temporal Video Segmentation based on DC Terms and MB coding mode

4.3.1.4 Test Results

End user shall test the result of the algorithms by comparing the results to temporal video segmentation based on DCT coefficients, based on DC terms, based on DC terms and MB coding mode, based on DCT coefficients, MB Coding Mode and MVs, based on MB coding mode and MVs, MB coding mode and bit rate information. End user shall determine the best algorithm by testing results.

4.3.1.5 List Data Set

End user shall list the data set, shall choose on of the data and shall compare with the other methods to compare and shall check the accuracy of the result.

4.4 USER INTERFACE DESIGN

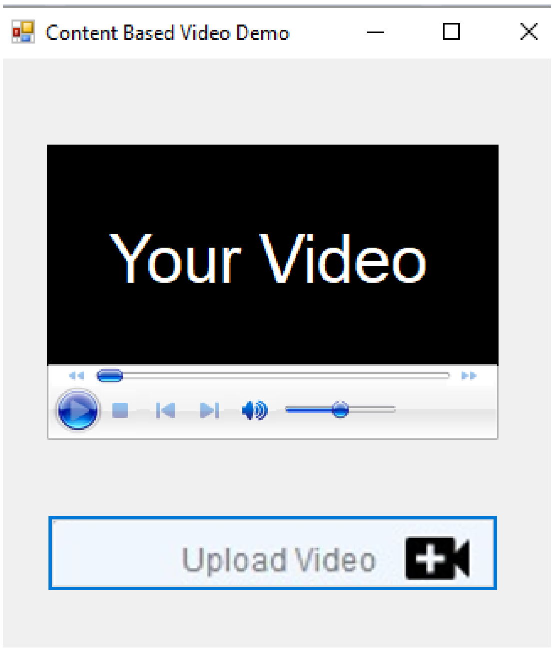

Figure 14: Uploading a video

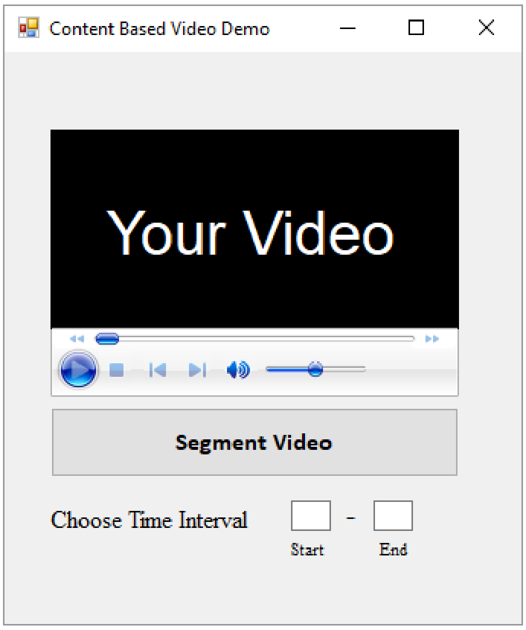

Figure 15: Choosing time interval

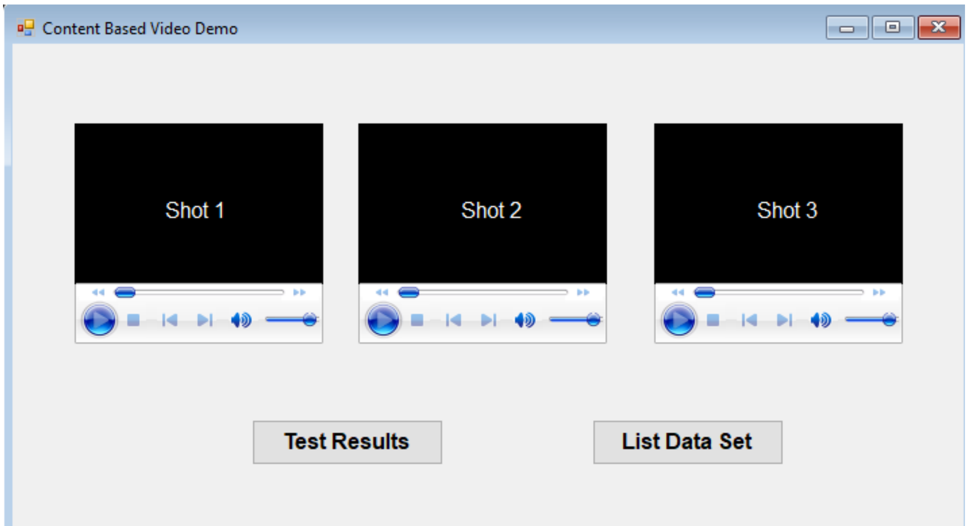

Figure 16: Segmented shots

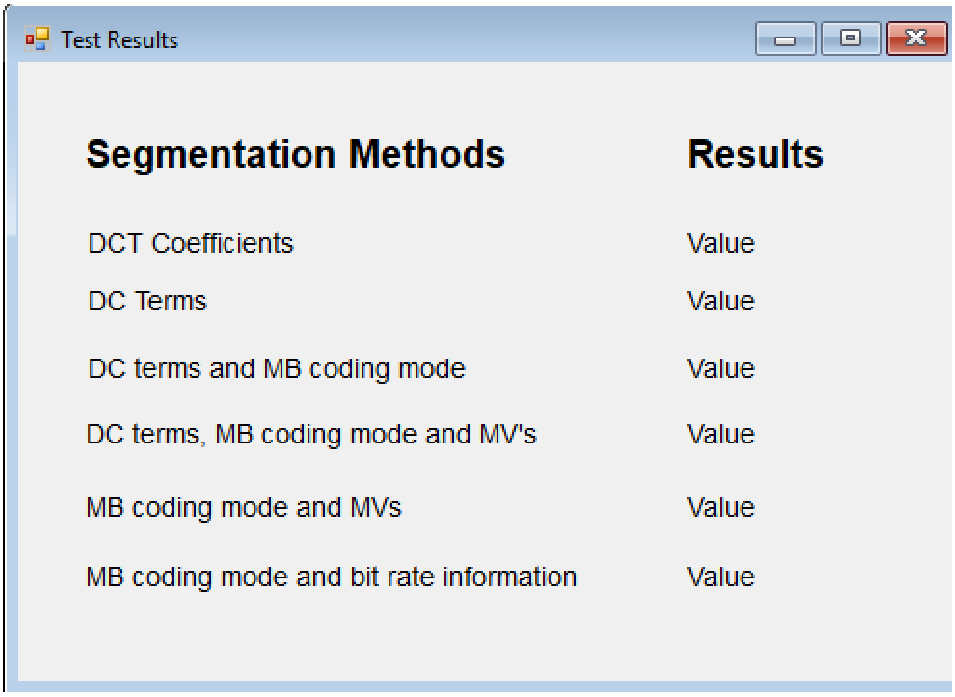

Figure 17: Test Results

5. Conclusions

This document includes wide information about our project that titled as "Content Based Video Segmentation". Our project aims to divide a video into segments (shots) wherein the scene or camera position does not change significantly and our project considers both visual and semantic contents of the frames.

To develop this project, we have made a lot of research about video coding and its standards. We have analyzed similar projects, and tried to understand what features have made them effective. We have gained a lot of information about video segmentation and how to develop a project that segment video content based. After research, we have received requirements from our advisor and co-advisor. Upon these requirements, the SRS document is prepared. After requirements are specified, design of the developing product is prepared and this design is explained in the SDD document. During this period, we designed the architecture of the project.

Video segmentation can be used in many applications. Content selection allows time and space to be reduce, during the retrieval process. Some of the advantages of Content based video segmentation can be considered as summarizing a sport event, skimming a video and mining the patterns of a video. In addition, video surveillance can be done with content based video segmentation where the segmentation result can be used to identify a pattern. Content based video segmentation is also used in videoconferencing, it helps to achieve a better quality by coding the most relevant objects at higher quality.

Acknowledgement

We are grateful for guidance we have received from Prof. Dr. Erdoğan DOĞDU and Assist. Prof. Dr. Roya CHOUPANI. The help we received from them was a great asset to improve this project and ourselves.

References

[1] Y. Wang, S.F. Chang, A. C. Loui, "Subjective Preference of Spatio- Temporal Rate in Video Adaptation Using Multi-Dimensional Scalable Coding," Proceedings of IEEE International Conference on Multimedia and Expo, Taipei, Taiwan, June 2004.

[2] Suganya.G, Mahesh.K, "A Survey: Various Techniques of Video Compression", *International Journal of Engineering Trends and technology*, vol.7, no. 2, 2014.

[3] Thounaojam, D., Trivedi, A., Manglem, K. and Roy, S. (2014). In: Intelligent Computing, Networking, and Informatics, Advances in Intelligent Systems and Computing. [online] Available at:https://www.researchgate.net/publication/ 283532444_A_Survey_on_Video _Segmentation [Accessed 3 Nov. 2017].

[4] "Video - HiDef Audio and Video", Hidefnj.com. [Online]. Available: http://hidefnj.com/video.html.

[Accessed: 19- Nov- 2017].
[5] M. Mrak, M. Naccari, R. Weerakkody, A. Dias, S. Schwarz and S. Blasi, "Video Coding - BBC R&D", Bbc.co.uk. [Online]. Available: http://www.bbc.co.uk/rd/projects/video-coding. [Accessed: 19- Nov- 2017].
[6] E.C. Reed and J.S. Lim, ”Optimal multidimensional bit-rate control for video communications”, IEEE Transaction on Image Processing, vol. 11, no. 8, pp. 873–885, August 2002.
[7] N. Ahmed, T. Natrajan, and K. R. Rao,“Discrete cosine transform,” *IEEE Trans. Comput.*, vol. C-23, no. 1, pp. 90-93, Dec. 1984.
[8] T. Sikora, "Trends and Perspectives in Image and Video Coding", Proceedings of the IEEE, vol. 93, no. 1, pp. 6-17, 2005.
[9] A. Smolic, T. Sikora, and J.-R. Ohm, “Long-term global motion estimation and its application for sprite coding, content description, and segmentation,”IEEE Trans. Circuits Syst. Video Technol., vol. 9, no.8, pp. 1227–1242, Dec. 1999.
[10] Yun Q. Shi, Huifang Sun, Image and video compression for multimedia engineering-fundamentals, algorithms, and standards, CRC Press, Boca Raton London New York Washington, D.C., 2000
[11] F. Pereira and T. Ebrahimi, The MPEG-4 Book. Englewood Cliffs, NJ: Prentice-Hall, 2002.
[12] T. Aziz and D. Dolly, "Motion Estimation and Motion Compensated Video Compression Using DCT And DWT", International Journal of Emerging Technology and Advanced Engineering, vol. 2, no. 12, 2012.
[13] X. Liyin, S. Xiuqin and Z. Shun, "A Review of Motion Estimation Algorithms for Video Compression", in 2010 International Conference on Computer Application and System Modeling, Taiyuan, China, 2017.
[14] Huifang Sun, Yun Q. Shi, Image and Video Compression for Multimedia Engineering Fundamentals, Algorithms, and Standards. CRC Press, 2nd Edition ed., Vol. ;.2007
[15] Public MPEG documents, http : //www.chiariglione.org/mpeg/workingdocuments.ht
[16] ”Video coding for low bit rate communication”, International Telecommunication
[17] Union-Telecommunication (ITU-T), Geneva, Switzerland, Recommendation H.263, 1998
[18] X. Wei, Y. Sun, Y. Zhang. I. Ahmad, "Video Transcoding: An Overview of Various Techniques and Research Issues," *IEEE TRANSACTIONS ON MULTIMEDIA*, vol. 7, no. 5, pp. 793-804, 200.

[19] Mehmet Tolun, Roya Choupani, and Stephan Wong "Video Coding and Transcoding: A Review". Retrieved November, 2017 Available: http://www.ce.ewi.tudelft.nl/choupani/

[20] I. Nahhas and M. Drahansky, "Analysis of Block Matching Algorithm with Fast Computational and Winner-Update Strategies", *International Journal of Signal Processing, and Pattern Recognition,* vol.6, no.3, 2013

[21] Ostermann, J.; Bormans J.; List P.; Marpe D.; Narroschke M.; Pereira F.; Stockhammer T.; Wedi T.: Video coding with H.264/AVC

[22] O. Nemþiü, M. Vranješ and S. Drlje, "Comparison of H.264/AVC and MPEG-4 Part 2 Coded Video", in 49th International Symposium ELMAR-2007, Zadar, Croatia, 2017.

[23] Marpe, D. and Wiegand, T. and Sullivan, G.J. (2006): The H.264/MPEG4 advanced video coding standard and its applications: IEEE Communications Magazine, 44(8), pp. 134-143.

[24] Wiegand, T; Sullivan, G.J; Bjontegaard, G; Luthra, A. (2003): Overview of the H.264/AVC video coding standard: IEEE Transactions on circuits and systems for video technology, 13(7), pp. 560-576.

[25] W. Niblack, X. Zhu, J.L. Hafner, T. Breuer, D.B. Ponceleon, D. Petkovic, M.D. Flickner, E. Upfal, S.I. Nin, S. Sull, B.E. Dom, B.-L. Yeo, S. Srinivasan, D. Zivkovic, M. Penner, Updates to the QBIC system, in: Proceedings of IS&T/SPIE Conference on Storage and Retrieval for Image and Video Databases VI, Vol. 3312, 1997, pp.150-161.

[26] I. Koprinska and S. Carrato, "Temporal video segmentation: A survey", Signal Processing: Image Communication, vol. 16, no. 5, pp. 477-500, 2001.

[27] G. Ahanger and T. Little, "A Survey of Technologies for Parsing and Indexing Digital Video1", Journal of Visual Communication and Image Representation, vol. 7, no. 1, pp. 28-43, 1996.

[28] F. Idris and S. Panchanathan, "Review of Image and Video Indexing Techniques", Journal of Visual Communication and Image Representation, vol. 8, no. 2, pp. 146-166, 1997.

[29] Arman, A. Hsu, M.-Y. Chiu, Image processing on compressed data for large video databases, in: Proceedings of First ACM International Conference on Multimedia, 1993, pp. 267-272.

[30] B. Yeo and B. Liu, "Rapid scene analysis on compressed video", IEEE Transactions on Circuits and Systems for Video Technology, vol. 5, no. 6, pp. 533-544, 1995.

[31] J. Maeda, Method for extracting camera operations to describe sub-scenes in video sequences, in: Proceedings of IS&T/SPIE Conference on Digital Video Compress, San Jose, Vol. 2187, 1994, pp. 56-67.

[32] H. Zhang, C. Low and S. Smoliar, "Video parsing and browsing using compressed data", Multimedia Tools and Applications, vol. 1, no. 1, pp. 89-111, 1995.

[33] Koprinska, S. Carrato, Detecting and classifying video shot boundaries in MPEG compressed sequences, in: Proceedings of IX European Signal Processing Conference (EUSIPCO), Rhodes, 1998, pp. 1729-1732.

[34] Feng, K.-T. Lo, H. Mehrpour, Scene change detection algorithm for MPEG video sequence, in: Proceedings of International Conference on Image Processing (ICIP'96), Lausanne, 1996.

[35] M. James, "An Empirical Framework For Learning (Not a Methodology)", *Scrummethodology.com*, 2017. [Online]. Available: http://scrummethodology.com/. [Accessed: 10- Dec- 2017].

[36] O. Nemþiü, M. Vranješ and S. Drlje, "Comparison of H.264/AVC and MPEG-4 Part 2 Coded Video", in 49th International Symposium ELMAR-2007, Zadar, Croatia, 2017.

[37] T.Sheela, T.Muthumanickam A. Nagappan" A LFSR based Binary Numeral System Using CMOS VLSI" International Journal of VLSI and Embedded Systems, ISSN:2249–6556,Vol 05,Article 12210;January2014

[38] M. Tolun, R. Choupani, S. Wong "Video Coding and Transcoding: A Review". Retrieved November, 2017 Available: http://www.ce.ewi.tudelft.nl/choupani/

[39] "MATLAB - MathWorks", *Mathworks.com*, 2017. [Online]. Available: https://www.mathworks.com/products/matlab.html. [Accessed: 24- Dec- 2017].

[40] "MATLAB Coder", *Mathworks.com*, 2017. [Online]. Available: https://www.mathworks.com/products/matlab-coder.html. [Accessed: 24- Dec- 2017].

[41] O. Nemþiü, M. Vranješ and S. Drlje, "Comparison of H.264/AVC and MPEG-4 Part 2 Coded Video", in 49th International Symposium ELMAR-2007, Zadar, Croatia, 2017.

[42] M. James, "An Empirical Framework For Learning (Not a Methodology)", *Scrummethodology.com*, 2017. [Online]. Available: http://scrummethodology.com/. [Accessed: 10- Dec- 2017].

Appendices

All images used in the document does not reflect actual product. User interface screens could change in the future. These images are used only for draft project.