Automatic Generation of Effective Video Summaries

Johannes Sasongko

Bachelor of Information Technology

Submitted in fulfilment of the requirements for the degree of
Master of Information Technology (Research)

Information Systems Discipline
Faculty of Science and Technology
Queensland University of Technology
January 2011

Keywords

Key frame, multimedia information retrieval, scene segmentation, shot boundary detection, video summarisation, video tagging.

Abstract

As the popularity of video as an information medium rises, the amount of video content that we produce and archive keeps growing. This creates a demand for shorter representations of videos in order to assist the task of video retrieval. The traditional solution is to let humans watch these videos and write textual summaries based on what they saw. This summarisation process, however, is time-consuming. Moreover, a lot of useful audio-visual information contained in the original video can be lost. Video summarisation aims to turn a full-length video into a more concise version that preserves as much information as possible. The problem of video summarisation is to minimise the trade-off between how concise and how representative a summary is. There are also usability concerns that need to be addressed in a video summarisation scheme.

To solve these problems, this research aims to create an automatic video summarisation framework that combines and improves on existing video summarisation techniques, with the focus on practicality and user satisfaction. We also investigate the need for different summarisation strategies in different kinds of videos, for example news, sports, or TV series. Finally, we develop a video summarisation system based on the framework, which is validated by subjective and objective evaluation.

The evaluation results shows that the proposed framework is effective for creating video skims, producing high user satisfaction rate and having reasonably low computing requirement. We also demonstrate that the techniques presented in this research can be used for visualising video summaries in the form web pages showing various useful information, both from the video itself and from external sources.

Table of Contents

List of Figures

List of Tables

Statement of Original Authorship

The work contained in this thesis has not been previously submitted to meet requirements for an award at this or any other higher education institution. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made.

Signature: _____

Date: _____

Acknowledgements

I would like to thank my supervisor for being an endless source of motivation; my close family members, who stood behind every decision I made; and colleagues who acted as intellectual sparring partners, keeping me sharp through dull moments.

This research was carried out as part of the activities of, and funded by, the Smart Services Cooperative Research Centre (CRC) through the Australian Government's CRC Programme (Department of Innovation, Industry, Science and Research).

Chapter 1: Introduction

1.1. BACKGROUND

Due to the widespread availability of high-speed Internet, videos have become a common information medium in the World Wide Web. YouTube, possibly the largest public online video archive, had around 101.9 million unique viewers from the USA alone during the month of January 2009, with an average of 62.5 videos per viewer (comScore, Inc., 2009). In January the following year, these numbers climbed to 136.5 million viewers and 93.9 videos per viewer (comScore, Inc., 2010).

The increasing supply and demand of video content, however, creates an interesting challenge for both producers and consumers: while the number of videos produced and stored can grow at a rapid rate, the amount of time that we have to watch these videos is limited. For example, YouTube Inc. (2010) reported that 24 hours' worth of video content was being uploaded every minute to YouTube, as of March 2010. This illustrates our inability as individuals to keep up with the vast amount of video data around us.

This is the root of our interest in video summarisation: if we can somehow "compress" the information contained in videos, we will be able to understand more videos without actually watching them in their entirety.

In traditional media, a video summary would consist of a textual description of the video, created manually by human writers. These textual summaries are stored in databases, and thus can be searched, viewed, edited, and so on. The Internet has changed this somewhat, with many recent movies having interactive websites and video trailers. However, the creation of such summaries is still largely a manual process.

An automatic video summarisation system aims to condense the information found in a full-length video into a shorter representation. However, there is usually a trade-off between how concise and how representative a summary's content is. The problem of video summarisation is to minimise this trade-off, creating a concise

version of a video which is still a close representation of the original contents. There are also usability concerns that need to be addressed in a video summarisation scheme; for example, how easily understandable the summary is, how long it takes to find a particular information from the summary, and so on.

1.2. AIMS

The main aim of this research is to develop an automatic video summarisation framework that produces effective video summaries. The research looks at various existing video summarisation techniques, and builds upon these techniques by combining and improving them. The research also investigates the need for different summarisation strategies in different kinds of videos, for example news, sports, or TV series. Finally, we develop a video summarisation system based on the framework, which is validated by subjective and objective evaluation.

1.3. SIGNIFICANCE

As the previous sections explain, this research works on the problem of creating effective video summaries. It creates a new video summarisation framework based on ranking video segments by their perceived importance in the original video. It also presents some feature extraction and feature combination techniques to improve the quality of the video summaries. The application of the system on different video types highlights some important considerations while working on specific types of data.

Another contribution of this research is in the development of visualisation techniques that are suitable for presenting video summaries in web pages. It shows how image thumbnails, tags, and contextual information can be added to the visualisation in order to improve the quality of summaries.

The framework produced in this research will be useful for general-purpose applications and is appropriate for usage in, for example, summarising movies, home videos, and news videos. The framework is not intended for special-purpose video summarisation such as security/surveillance due to the difference in focus of such

applications (video summarisation in the surveillance field focuses more on the detection of certain suspicious events than on user satisfaction).

One of the main problems of video summarisation is in finding segments that are important or interesting, and eliminating segments that are neither important nor interesting to viewers. In order to solve this, a scoring scheme will be used to rate the importance of each video segment, and segments with higher scores will have priority for inclusion in the video summary. In addition, junk filtering and duplicate filtering will also be employed to reduce the amount of unnecessary segments. Interesting segments will be "assembled" into the final video summary, taking into consideration the implications of various techniques (video speed-up, rate of change, etc.) on the pleasantness of the output.

Some examples of practical applications where an effective video summarisation system will improve productivity are:

- Video browsing, to allow users to quickly see the contents of videos.
- Video searching, letting users preview search results.
- Video editing, giving editors an easier time managing large numbers of video recordings.
- Casual viewing, for users who are not interested in viewing full videos given their time limitations.

1.4. PUBLICATIONS

Parts of this research have been published and presented in one workshop and one conference; both are international refereed publications:

Sasongko, J., Rohr, C., & Tjondronegoro, D. (2008). Efficient generation of pleasant video summaries. In *Proceedings of the TRECVid Video Summarization Workshop* (pp. 119–123).

Sasongko, J. & Tjondronegoro, D. (2010). Automatic visualization of story clusters in TV series summary. In *Advances in multimedia modeling*, Lecture Notes In Computer Science, 5916 (pp. 656–661).

1.5. THESIS OUTLINE

Chapter 2 provides an overall view of the current state of video summarisation. The chapter begins by explaining elements of a generic video summarisation. It then explores several domain-specific summarisation methods that have been used in the past. Finally, it details some important techniques that are used in video summarisation, as well as the criteria for evaluating video summaries.

Chapter 3 provides an in-depth explanation of the video summarisation framework. It details all of the steps involved in the framework, including optional steps that may be used depending on the user's intention. It also describes the web visualisation methods that have been developed for this research and how they provide valuable information to users.

Chapter 4 presents the outputs of this research. The first part of this chapter describes in detail the results of the evaluation for the video skim creation, which was performed as part of the TRECVid 2008 event. The chapter also shows two alternative visualisations in the form of web pages: the first is a simple interactive video browser, and the second is a story-based video visualisation method.

Chapter 5 concludes this thesis by summarising its contributions, and discusses the potential future work that can be based on this research.

Chapter 2: Video Summarisation Techniques

2.1. INTRODUCTION

Video summarisation deals with the creation of shorter representations of videos that help humans obtain information from these videos more easily (faster, in a more comfortable manner, and so on). The main motivation for the development of video summarisation systems is because managing a large amount of raw video requires considerable time and effort. Video summarisation helps to solve this "information overload" by emphasising important information contained in videos and de-emphasising or eliminating less important details.

A number of different video summarisation techniques have been developed, with different points of view and objectives. Based on the form of the summary output, existing techniques are classified into two categories (Truong & Venkatesh, 2007). The first category produces a set of *keyframes*, which are static images representing the contents of a video. This, however, cannot convey information present in the audio and motion of the original video. Techniques in the second category produce a *video skim* for a given video; in other words, it creates a significantly shorter summary video that humans can view in order to gain understanding of the contents of the original video. This research focuses on the second category, that is, creating video skims. However, many specific video summarisation techniques can be used for both, and we demonstrate some possible applications and/or combinations in later chapters.

2.2. GENERIC VIDEO SUMMARISATION METHODS

The simplest form of video summarisation is to speed-up a video according to the target length (Hauptmann et al., 2007). Instead of trying to extract any kind of context from videos, this method simply uses a higher frame sampling on the videos. For example, for a 25 frame-per-second video, given 2% target summary length (1/50 of the original video), it is sampled at 25/50 = 0.5 frame per second, and the resulting summary will appear 50 times faster than the original. Although this

produces a high ground truth inclusion score, the method is said to "seriously degrade coherence, causing discomfort to the viewer" (Truong & Venkatesh, 2007). However, the idea of speeding up certain parts of a video summary exists in more sophisticated algorithms (Chen, Cooper, & Adcock, 2007; Lie & Lai, 2004) which do not simply speed up the whole video, but rather select important video segments and adjust their speed-up rates.

Most of the recent summarisation approaches use similar ideas as Ma, Lu, Zhang, and Li's (2002) *user attention modelling* technique to automatically find important segments in a video. User attention modelling predicts how much attention humans give to each point in a video based on various video features such as motion, faces, camera effects, and speech. Models for these features are generated and combined to produce the final *attention curve*, which is inspected for local maxima to obtain important points in the video.

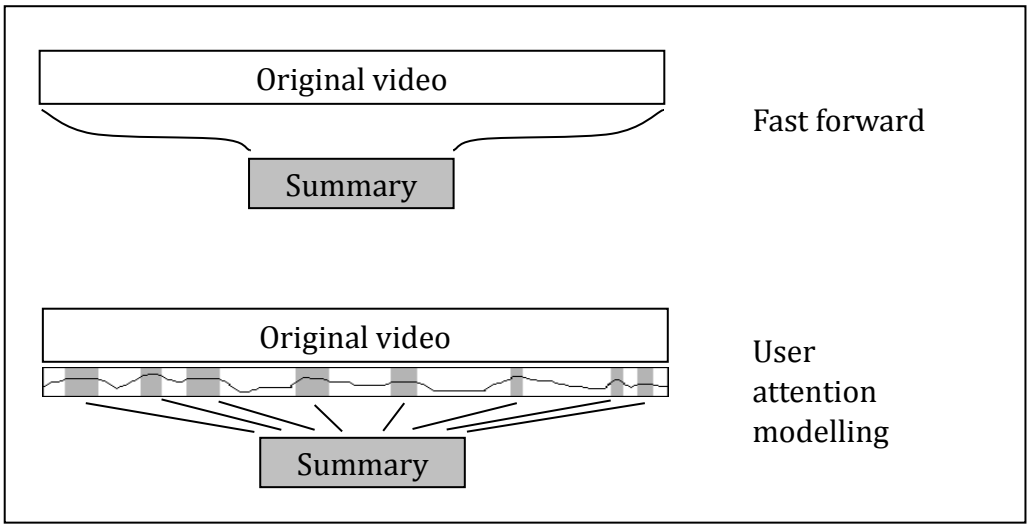

Figure 2.1. Fast forward and user attention modelling techniques

While most research has focused on creating video summaries in a fully automated way, Yu, Ma, Nahrstedt, and Zhang (2003) presented a unique video summarisation scheme that works semi-automatically, with human involvement in the ranking stage. The core of this method is a shot ranking algorithm called ShotRank that is based on a log of users' viewing behaviour. The motivation for ShotRank is that, as an *engaged user* (someone with sufficient interest, experience, time, etc.) browses a video, their viewing behaviour reflects their evaluation of parts of the video. The more times a shot is viewed, the relative importance of this shot gets higher. The algorithm also tries to predict the authoritativeness of each user by

comparing their viewing behaviour with previous users'. If a user does not view shots that have been ranked high based on previous iterations of the algorithm, his relative value is lowered, and thus his behaviour will have less effect on the subsequent ranking of the shots. Due to the semi-automated nature of this method, however, its application is quite limited and is not suitable for many purposes which require a fully automated summarisation system.

Another new idea for a semi-automatic approach to video summarisation involves monitoring human physiological responses to videos and creating video summaries based on these data (Money & Agius, 2008a). Physiological responses useful for this purpose include electro-dermal response, respiratory rate, blood volume pulse, heart rate, and respiration amplitude. However, the interpretation of this information for the purpose of creating the actual video summaries has not been explored.

Some studies have experimented on adding presentation features that help viewers to understand summary videos better. Timelines, for example, are often used to show the time position of the current frame in the original video (Beran et al., 2008; Chen, Adcock, & Cooper, 2008; Dumont & Mérialdo, 2008). Other methods include presenting in a four-pane split screen (Dumont & Mérialdo, 2007), showing key frames from the current shot on one side of the screen (Dumont & Mérialdo, 2008), and listing scenes and actors at the beginning of the summary (Wang et al., 2007).

2.3. VIDEO SUMMARISATION IN MULTIPLE DOMAINS

Video summarisation is often seen as a domain-specific problem. There are some applications where this is not practical, for example in a video retrieval system where the videos can be of any type. However, due to the varying structures and objectives of different types of videos, there are some optimisations that can be performed on top of a generic video summarisation framework to make it work better on specific video types. The rest of this section presents several domain-specific techniques that are useful for summarising videos of their respective types.

2.3.1. SPORTS

In sports videos, segments are classified as either *play* or *break*. Breaks consist of segments where the game is in pause or where the video does not show the game in action. Examples of breaks include player close-ups, replays, and spectator shots. Plays, on the other hand, occur when the game is happening. Because breaks contain repeated information, or information that is less important, they can be removed to arrive at a summarised version of the video (Ekin & Tekalp, 2003). However, Tjondronegoro, Chen, and Pham (2004) argue that breaks are also important, because they may contain information not present in play segments. For example, breaks may contain close-up shots of an important segment from different angles, exposing viewers to information that is not apparent from the play segments.

Some research has been done on detecting specific events in news videos. Pan, van Beek, and Sezan (2001) detect slow-motion replay sequences using machine learning on the transition effects that occur before and after such replay sequences. The idea is that slow-motion replays always occur in the sequence: normal segment – visual effect – replay – visual effect – normal segment.

To complement play-break transition detection, Tjondronegoro, Chen, and Pham (2004) propose a simple whistle sound detection method by calculating the sound energy that is within the whistle's frequency range. However, because the whistles used between various types of sports are not uniform, the frequency ranges need to be trained individually depending on the type of the sports being analysed; the authors have identified the typical whistle frequency ranges for soccer, swimming, rugby, basketball, and netball. In the same paper, they also propose methods to locate interesting events in a sports video: excitement detection (based on the assumption that, during exciting events, the crowd's and commentator's voices become louder, faster, and higher in pitch) and text detection (using Hough transform to detect the text boxes).

2.3.2. NEWS

An important step in news video parsing is the segmentation of a news video into news stories. One way to do this is by the detection of anchorperson shots

(Chua, Chang, Chaisorn, & Hsu, 2004). Different anchorperson detection algorithms have been proposed in the past, but methods based on shot similarity clustering has been shown to work well (De Santo, Percannella, Sansone, & Vento, 2004; Smeaton et al., 2004).

Another method to determine story boundaries is based on the speech transcript. Pickering, Wong, & Rüger (2003) use a hybrid method based on the assumption that story boundaries always occur at shot boundaries; the shots are then merged based on the similarity of the textual content.

A novel presentation technique for news video summarisation was proposed by Lie and Lai (2004), where anchorpersons' voice is placed over non-anchorperson shots. This effectively removes anchorperson shots, which are generally not visually interesting, but keeps most of the news information from these anchorperson shots intact (via the audio track).

2.3.3. MUSIC VIDEO

Music videos are significantly different from other types of videos because they are more audio-oriented, and the visual aspects of the videos can be considered secondary.

Shao et al. (2006) presented an automatic music video summarisation system with a separation between the music and video summary generators. They argue that audio-visual synchronicity in a music video is less important than in other types of videos. The music summary generator extracts several features from the audio track: linear prediction coefficients (LPC), LPC-derived cepstrum coefficients, zero-crossing rates, and mel-frequency ceptral coefficients. These features are given to a support vector machine in order to classify the vocal and instrumental parts of the audio. The music summary is created by clustering sound frames and finding repeated segments. These repeated segments are the most representative segments of the music, and are thus included in the summary. On the other hand, the video summary generator uses a similar clustering method, but works on video shots instead of audio frames. In each cluster, the shot with maximum length is taken as the representative of that cluster. Representative shots with less than 1.5 seconds length are removed, and the remaining shots are shortened to 1.5 seconds. These shots are

then aligned with the audio summary so that the audio and video clips are not entirely unsynchronised.

2.3.4. RUSHES

Rushes are film recordings that are raw / unedited. In terms of content, it includes multiple takes of the same scene, noise, blank frames, and other types of "junk" due to its unedited nature.

The TRECVid 2007 event evaluated 22 automatic rushes summarisation systems (Over, Smeaton, & Kelly, 2007). For this evaluation, the target length of each video summary was set to 4% of the original video length. The resulting summaries were evaluated based on completeness (how much information is retained), ease of understanding, redundancy, evaluation time, summary length (in relation to the 4% target length), and creation time. In 2008, 31 teams participated in the TRECVid rushes summarisation task (Over, Smeaton, & Awad, 2008). The target summary length was reduced to 2% of the original length, and a new "amount of junk" evaluation measure was introduced.

Rushes are a very specific type of video, containing a lot of repetitions from multiple takes of the same scene. In order to find and remove redundant shots, researchers have used clustering on the set of shot key frames, where only one shot from each cluster is retained in the video summary (Wang et al., 2007; Xie et al., 2004). Blank frames, colour bars, and clapboard shots can also be removed to save some space in the generated summaries.

Over, Smeaton, & Kelly (2007) note that assessors seem to prefer summaries that only contain parts of the original video played at normal speed. Lower "ease of use" scores were given to summaries that are played at high speed, consist only of static slide shows, or contain multiple simultaneous images. Although not mentioned explicitly in the paper, it seems that systems which try to show more of the "important" or "interesting" shots (as opposed to simply all shots) obtain better scores in the evaluation.

2.3.5. MEETING RECORDINGS

A summarisation system for meeting recordings is presented by Erol, Lee, and Hull (2003). Visually, the system detects activity by detecting local luminance changes between frames. Audio activity is detected by measuring the sound amplitude and sound localisation (i.e. where the sound originates from). The assumption is that, if the sound origin changes quickly, a discussion is happening. Audio transcripts are also processed to detect the occurrence of interesting keywords. The locations of interesting segments from these three modalities (audio, visual, and textual) are then combined to create the video summary.

Compared to other kinds of videos, meeting videos present a unique case which turns out to have some advantages for the purpose of summarisation. For example, speaker identification is normally performed using voice recognition, which is complicated. In meeting videos, participants can simply be identified by their relative position to the microphone, because they do not normally move within a meeting session. Motion is also usually limited to significant events, such as people joining the meeting or someone doing a presentation.

2.3.6. HOME VIDEO

Hua, Lu, and Zhang (2003) present an automatic home video editing system which consists of three stages:

1. *Content analysis:* In this stage, the video and music are analysed to find their features. A viewer attention curve is built based on the importance of each point in the video. Sub-shot segmentation and sentence detection are performed, and the music is segmented into clips based on the presence of strong beats.

2. *Content selection:* This stage uses the results of the content analysis to choose video segments and music clips to be included in the summary. First, low-quality segments are removed from the video. Second, interesting sub-shots are selected based on the attention curve and the speech recognition. Third, the sub-shots are aligned with selected music clips to ensure that sub-shot transitions occur at music beats.

3. *Composition:* Transition effects are applied to the selected sub-shots. The type of transition effect between two sub-shots is determined by their similarity. Afterwards, the video is ready for rendering.

2.4. VIDEO SEGMENTATION

At the most basic level, a video consists of a number of frames. Between the video level and frame level, a video can be segmented in several different conceptual levels.

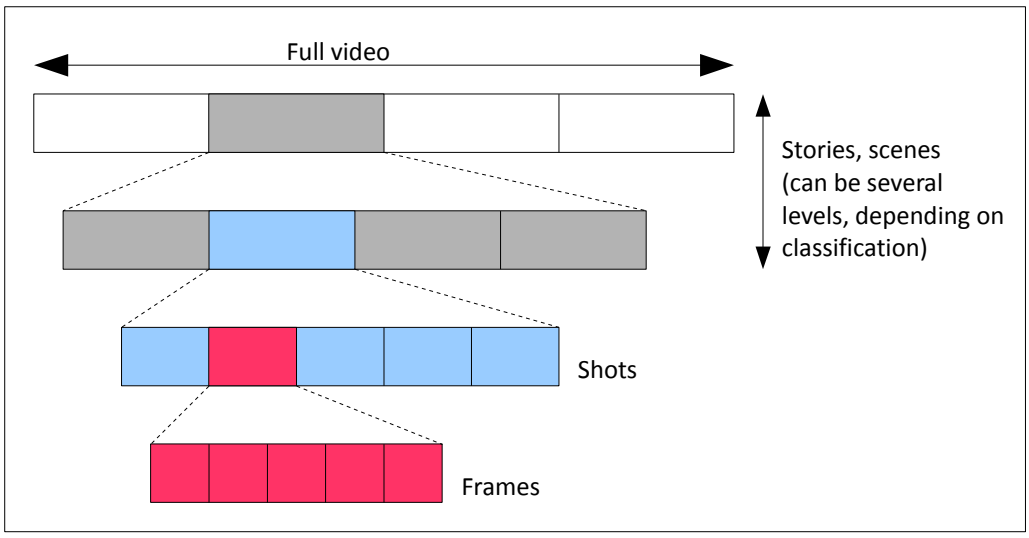

Figure 2.2. Different levels of video segmentation

Below the video level, there is an arbitrary number of levels that represent different application- and domain-specific segmentations normally described as *stories* or *scenes*. For example, a news broadcast may contain a number of news items (1 level); meanwhile, a tennis match consists of sets, each consisting of games, which in turn consists of points (3 levels). Due to the dependence on domain knowledge, there is no specific set of segmentation that can work in all types of videos.

Below the stories/scenes level, however, there is one level of segmentation that can be applied to any video. A shot is a continuous recording from one camera angle, and consists of adjacent frames in the video. In order to limit their problem space and processing time, many video indexing and retrieval applications work on shots as the lowest level (instead of frames). To do this, the video needs to be segmented into

shots first.

Figure 2.3. News video example.
The video consists of 4 stories, and each story consists of several shots (three sample shot key frames are shown here for each story).

Segmenting a video into shots is an important step in trying to understand the video. Shot boundary detection is used on many summarisation techniques because it simplifies the summarisation process by allowing each shot to be treated as a unit represented by a few *key frames* (often only one frame for one shot).

Shot boundaries are signified by shot transitions. There are two kinds of shot transitions: abrupt and gradual. Abrupt shot changes are easy to detect, because if two consecutive frames are visually very different they can be classified as belonging to two different shots; a common and effective way to do this is by comparing the histograms of the frames. On the other hand, detecting gradual shot changes needs to take into account the shot transition effects that are applied to the video, because simple frame-by-frame comparison often fails to detect these gradual transitions.

There have been some literature discussing the two types of shot transitions. Lupatini, Saraceno, and Leonardi (1998) provide a comparison of several shot boundary detection algorithms that are based on global colour histogram, motion, or contour. Their results show that algorithms based on global colour histogram perform better than motion- and contour-based algorithms.

Koprinska and Carrato (2001) present a more complete classification of shot segmentation algorithms, although they did not perform detailed evaluation on the different methods. In their classification system, shot segmentation algorithms are

broadly categorised based on whether they work on compressed or uncompressed information. Techniques that work on uncompressed video are further categorised into (1) pair-wise pixel comparison; (2) block-based comparison; (3) histogram comparison (global and local); (4) clustering-based segmentation; (5) feature-based segmentation; and (6) model-driven segmentation. On the other hand, segmentation techniques that directly work on compressed video data are categorised based on the particular types of features used: (1) discrete cosine transform (DCT) coefficient; (2) DC terms; (3) DC terms, macroblock coding mode, and motion vectors; (4) macroblock coding mode and motion vectors; and (5) macroblock coding mode and bitrate.

Further processing can be done to refine the shot boundary detection results and reduce the number of false positives (non-boundaries detected as boundaries) and false negatives (boundaries not detected). For example, it is possible to remove false shot boundaries that happen due to temporary noise such as camera flash, by checking that each detected shot is longer than a specified time period (Bertini, Del Bimbo, & Pala, 2001). Shots that last only for a short period of time can be considered noise and be eliminated, and the two shots around it are merged if they are similar.

2.5. RATING SEGMENTS USING INTERNAL FEATURES

To obtain the important segments of a video, it is necessary to detect some information from the video that, when combined together, determine the importance of a particular point / segment in the video. Money and Agius (2008b) classify the sources of this information into *internal* and *external*. Internal information is obtained from within the video, while external information comes from outside sources, for example user tagging or annotations. This research focuses information sources that are internal to the video; therefore, discussion on external video information has been omitted.

A video consists of three distinct dimensions: visual, audio, and textual. Each of these dimensions has a number of features that can be extracted in order to produce a meaningful summary of the video.

2.5.1. VISUAL DIMENSION

The most distinguishing feature of video from traditional media such as text and sound recording is the inclusion of moving images. The visual dimension of a video is rich in information and has many extractable features. While the extraction of some of these features leverage existing image processing techniques, other features can only be extracted using more information than that is available in single images.

Motion among a sequence of images can be determined using optical flow algorithms. Optical flow algorithms measure the motion speed and direction of points from the original image to the target image. In general, object motion is signified by motion in parts of the image, while camera motion involves motion of all points in the image. It is worth noting that some video encoding schemes such as MPEG may embed motion information in the video file, which eliminates the need to re-calculate it (Smith & Kanade, 1997; Yu, Kankanhalli, & Mulhen, 2003).

Colour, intensity, and orientation can be used to detect interesting regions within an image. The research in this area is based on imitating human response to the composition of these features, where certain patterns cause more attention to be given to a region in the image (Itti, Koch, & Niebur, 1998).

Visual effects are sometimes used to delimit specific segments of a video. For example, Pan, van Beek, and Sezan (2001) use Hidden Markov Models to train and detect transition effects that occur before and after slow-motion replays in sports video.

In a video summarisation system, *repetition* is usually not desired, and only one shot from a set of repeated shots should be included in a video summary. However, repetition may also signify an important video segment, for example in the case of sports video replays (Tjondronegoro, Chen, & Pham, 2004). A method often utilised to detect repeated shots is shot clustering (Wang et al., 2007; Xie et al., 2004), which groups together shots that are visually similar. On a higher level, repetition of scenes (or shot sequences) can be detected by considering it to be a local alignment problem (Wang et al., 2007). The image feature most commonly used for shot clustering is the colour histogram of shot key frames, although *colour moments*

has been shown to better represent the similarity between images in an image searching scenario (Stricker & Orengo, 1995).

2.5.2. AUDIO DIMENSION

Despite normally being considered a secondary dimension in videos, audio tracks do contain some important information that can be used in video summarisation. In some applications, the *occurrence of specific sounds* can be an important cue as to the content of a video segment. Usually machine learning on low-level audio features is applied in order to detect excitement or some significant event. For example, Tjondronegoro, Chen, & Pham (2004) train their sports summarisation system to detect the whistle sounds from a few different types of sports, in order to locate potentially interesting segments. Conversely, in Chen, Cooper, and Adcock's (2007) work, the existence of sound from clapboard being clapped determines that a segment is a clapboard shot, which is considered junk and excluded from the final summary.

Among the various excitement measurement techniques, *loudness* and *pitch* are some of the most popular. Loudness (audio energy) is used to detect interesting segments of videos, with the assumption that louder sounds attract more attention and are thus more interesting to viewers (Ma, Lu, Zhang, & Li, 2002). Similarly, higher pitch (audio frequency) may also indicate excited speech, such as in a sports commentary when a significant event occurs (Tjondronegoro, Chen, & Pham, 2004).

Although less obvious, the *type of sound* present at each point in a video also contain useful information. Silence detection can be used to locate pauses in speech in order to determine excited speech, that is, speech with short pauses (Tjondronegoro, Chen, & Pham, 2004), and significantly long silence may indicate lack of interesting action. The presence or absence of speech and music has also been used as a basis to determine the attention level placed on a certain audio segment (Ma, Lu, Zhang, & Li, 2002).

2.5.3. TEXTUAL DIMENSION

Despite videos being an audio-visual media, some textual information can be extracted out of them. Visually, Optical Character Recognition (OCR) can be applied

to find texts that are shown in the video frames. This is useful for recognising the contents of a video scene, such as a location or a person (Bertini, Del Bimbo, & Nunziati, 2006; Sato, Kanade, Hughes, & Smith, 1998). Closed caption text provided by the video delivery system can also be utilised to create video summaries (Agnihotri, Devara, McGee, & Dimitrova, 2001; Bagga, Hu, Zhong, & Ramesh, 2002). In place of closed caption text, Automatic Speech Recognition (ASR) can be performed on the audio track in order to transform speech into text.

For example, a simple text-based video retrieval method was presented by Xu, Chia, Yi, & Rajan (2006), where speech recognition is used to obtain textual information similar to that of subtitles. The resulting text is scanned for certain keywords that signify specific emotions. However, the use of this technique for video summarisation is limited, due to relying only on finding predefined words.

Once textual information has been extracted out of a video, *important keywords* can be detected by calculating the *term frequency–inverse document frequency* (tf-idf) measure of each word (Smith & Kanade, 1997). TF-IDF is defined as the frequency of a word in a scene, divided by the frequency of the same word in a standard text. In simpler terms, it tries to find words that do not normally occur often in a standard text but happens to occur often enough in a given portion of text. This measure gives a good indication of important keywords that are present in a video. A video summarisation method based on this was proposed by Yi, Rajan, & Chia (2005), whereby tf-idf vectors are created for each segment in the subtitles. Important keywords in the video are detected by clustering the tf-idf vectors. The segments where these keywords occur are used for the video summary. This method, however, does not take into account the visual aspect of the video.

2.6. EVALUATION OF VIDEO SUMMARIES

Until recently, video summarisation systems were evaluated independently of each other. Most evaluation methods involve some user studies, asking a number of volunteers to rate the video summaries. However, the lack of a set criteria to rate videos results in difficulties in directly comparing the performance of multiple summarisation systems due to the different evaluation methods and datasets.

TRECVid, an "international benchmarking activity to encourage research in video information retrieval" (Smeaton, Over, & Kraaij, 2006), introduced a video summarisation task in 2007. This event answered concerns regarding evaluation of video summarisation systems by providing a set of videos to be summarised by researchers, and evaluating the outputs based on specific guidelines. Although these evaluation guidelines were designed with one video type in mind (unedited movie recordings), they can be useful for evaluating other kinds of videos as well. The following list outlines the evaluation criteria that were used in TRECVid 2007 (Over, Smeaton, & Kelly, 2007).

- *Ground truth inclusion:* Ground truth inclusion refers to the amount of useful information from the original video that is retained in the summary. This is measured by human judges comparing the summary video with a predetermined list of ground truth (for example, whether the picture of a certain object is present in the summary video).

- *Ease of understanding:* Ease of understanding here refers to the generated video summaries, instead of the summarisation system. As this is largely a subjective measure, it can only be evaluated by user survey.

- *Redundancy:* The presence of redundant information in a summary video is not desirable, because a summary should ideally be as short as possible, while redundant shots occupy space that could otherwise be used to cover more ground truth. For example, a scene of a dialogue between two people contains many shots that are visually nearly identical and may be removed.

- *Evaluation time:* Evaluation time is the time that a judge takes to evaluate a video for its ground truth inclusion.

- *Summary length:* The purpose of video summarisation is to minimise the time needed to understand a video without losing too much information. Therefore, the length of the video summary is an important variable in determining the validity of the summary and the system that generates it.

- *Creation time:* In a realistic setting, the creation time of a video summary is relevant in determining the quality of a summarisation system. However, the

TRECVid 2007 evaluation does not emphasise summary creation times, other than mentioning the median value and showing that there are only few systems that take exceptionally longer times. This is made more difficult by the fact that participants run their systems in different machines with different capabilities.

In the TRECVid 2008 video summarisation task (Over, Smeaton, & Awad, 2008), the *ease of understanding* measure was removed, and two new measures were added:

- *Tempo and rhythm*: This measure is intended to capture users' satisfaction of the tempo and rhythm of the summary videos.

- *Amount of junk:* The unedited nature of the videos in the TRECVid dataset leads to some amount of "junk" that need to be removed from summaries. There are three types of junk present in the dataset: blank frames (completely black or completely white), colour bars (vertical bars in different colours), and clapper boards (boards shown at the beginning of shots to indicate, for example, the scene and take numbers).

An analysis of the TRECVid 2008 results is provided in section 4.1 *Video Skim Creation*.

2.7. SUMMARY OF LITERATURE REVIEW

The purpose of video summarisation is to allow people to obtain information from videos in a more efficient way. The need for video summarisation increases with the amount of video that are recorded and archived, and consequently several video summarisation systems have been developed.

Video summarisation is a common need among several application domains, and can be approached as a domain-dependent problem. However, this is sometimes not practical when dealing with videos of unknown type.

A popularly used summarisation method is to locate interesting segments or events in a video and copy them to the summary video. The detection of interesting

segments is based on lower-level features of the video, which can be categorised into visual, audio, and textual features. Existing techniques still largely rely on visual information, and often require computationally expensive operations that limit their practical usefulness.

Comparing several video summarisation techniques is difficult without common evaluation method and data. TRECVid 2007 and 2008 was the most recent evaluation event with has a specific task on video summarisation, providing a specific method and data for evaluating video skims.

Chapter 3: Framework for Automatic Video Summarisation

3.1. INTRODUCTION

Our video summarisation framework operates on two different types of information: visual and audio-textual. The visual information contained in a video are processed through several image/video processing techniques and the output is a video skim, that is, a short video. The audio-textual information uses some text data mining techniques to create a web visualisation. However, these two processes also exchange data with each other to produce better outputs.

Figure 3.1 shows the steps involved in this framework.

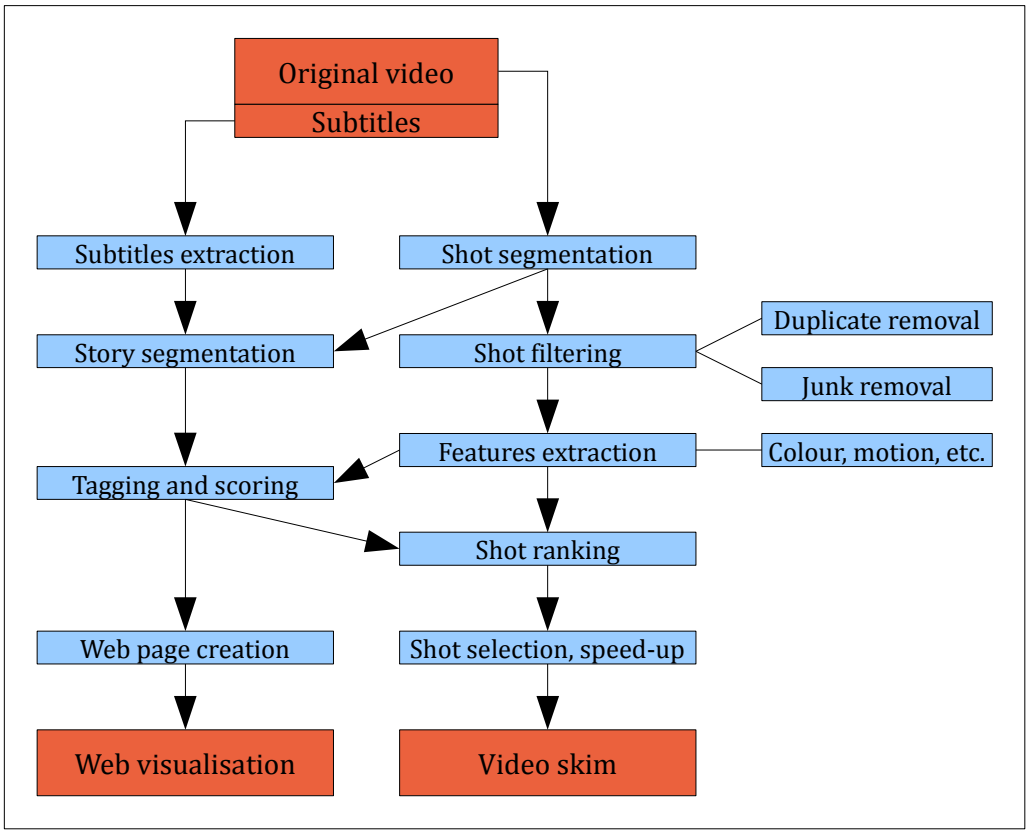

Figure 3.1. Summarisation framework

In the video skim creation process, the original video is split into distinct camera shots. Each shot is processed through a set of feature extractors, and scored according to its level of importance based on the extracted features. Once a score has

been assigned to each shot, the highest-scoring shots are selected as candidates for inclusion in the final video summary.

To create the web visualisation, subtitles from the video are extracted and split into story units based on the keyframes obtained from shot boundary detection. These subtitles are than tagged and scored to obtain important tags in the whole video and in each story. The web page is then created to show these tags, as well as still images from the video.

One important design consideration is that each of these steps have adjustable parameters that influence the final output. The framework allows users to modify these parameters according to their preference. The framework is also designed to be "plugged in" with different techniques in order to produce the finished video summaries. The next few sections explain critical parts of the framework, namely the shot segmentation, segment filtering, keyword detection (as part of tagging), and shot scoring. The last section in this chapter discusses some of the steps that are only relevant to the creation of our web visualisation, namely story segmentation, tag ranking, and creating the visualisation.

3.2. SHOT SEGMENTATION

A shot can be described as one continuous recording from a camera. For example, in a video showing a conversation between two people, the video may be cut into a sequence of shots going back and forth between two camera angles.

Because shot segmentation works on individual frames level, it is potentially one of the most time-consuming tasks in our framework. Previous works such as Gao and Tang (2002) focus on the accuracy of the detection and perform complex shot transition modelling in order to detect various types of abrupt and gradual shot transitions. However, these calculations are expensive in terms of processing time; consequently, we decided to limit the time complexity of the shot boundary detection by using a simple frame-by-frame comparison technique which, in Koprinska and Carrato's (2001) classification, is listed as the *segmentation of uncompressed video based on global histogram comparison* technique.

To detect shot boundaries, *firstly*, three histograms for each frame are calculated, one for each colour component (hue, saturation, value). *Secondly*, for each pair of consecutive frames, their colour histograms are compared by calculating the *chi-square* values (Patel & Sethi, 1996) for each of the three colour components:

$$\chi^2 = \sum_{i=1}^{n} \frac{\left(H_1(i) - H_2(i)\right)^2}{H_1(i) + H_2(i)} \quad , \tag{3.1}$$

where H_1 and H_2 are the histograms of the two frames, and n is the number of histogram bins. *Thirdly*, the three chi-square values are combined into the final histogram difference value:

$$d = a\chi^2_{hue} + b\chi^2_{sat} + c\chi^2_{val} \quad . \tag{3.2}$$

In this work, we set a to 4, b to 2, and c to 1 based on manual testing. This is in accordance with existing research which indicates that the use of chi-square test on global colour histogram, emphasising on the hue part of the colours, is effective for finding shot boundaries (Lupatini, Saraceno, & Leonardi, 1998). *Finally*, this final histogram difference value is compared to a set threshold obtained through experiments.

In order to speed up the process, frames are sampled at a lower rate than the original video frame rate. After the shot units are extracted, keyframes are selected automatically to visually represent each shot. To save processing time, this is done using a simple method whereby for every shot, the system selects the frame at the N-seconds mark into the shot as the keyframe. Any shot that is shorter than N seconds is deemed too short and not significant enough to be used in the summary. In this work, N is arbitrarily set to 2. This number can be changed within reasonable range with no noticeable difference on the summary output; the important thing to keep in mind is that N defines the minimum short length allowed, so it cannot be set too high.

3.3. SEGMENT FILTERING

Segment filtering means selecting segments that are known to be undesirable and removing them from the list of candidate segments. We classify segment filtering into *junk filtering* and *duplicate filtering*.

3.3.1. JUNK FILTERING

Junk filtering refers to the removal of known "bad" patterns. For example, our dataset contains blank and colour bar frames (Figure 3.2); these are artefacts from the recording process that have not been edited out. However, they also often occur in videos recorded from television when there are problems with the signal reception.

Figure 3.2. Sample junk shots

These types of junk shots can be removed with a simple visual similarity comparison. For example, blank frames and colour bars exhibit unique colour histograms, making them very easy to detect. Histograms of known blank and colour bar frames are compared with each candidate shot's key frame histogram. If any of them matches, the candidate shot is rejected.

Other junk shots may contain images of a certain kind of object, which is difficult to detect using global frame features. For example, in the TRECVid 2007 and 2008 datasets, the rushes videos contained junk shots in the form of clapboards (Figure 3.3). Some participants removed these clapboard segments by comparing the SIFT descriptors of the frames and trained clapboard images, and found some improvement in the "less amount of junk" measure (Christel et al., 2008). Another method used the audio track in order to find clapboard sounds and to remove frames surrounding the occurrence of the sound (Chen, Cooper, & Adcock, 2007).

Figure 3.3. Sample clapboard segments

To remove clapboards segments in our dataset, we exploit some of the properties of our shot slicing algorithm. Due to the shot boundary detection, a clapboard segment is either detected as a separate shot, or integrated into the next shot. The first case causes the shot to fail the length threshold, because clapboard

segments are very short. If the clapboard segment is instead integrated into the following shot, they are usually eliminated by the slicing process, which only takes the middle portion of shots. The full shot slicing algorithm is described in detail in Section 3.6.

3.3.2. DUPLICATE FILTERING

Shot clustering is used to detect retakes / duplicate shots. In order to find duplicate shots, all shots from the original video are clustered based on their similarity. Figure 3.4 shows some sample shot clusters.

Our clustering method uses the histogram difference of shot keyframes (calculated using the chi-square test as explained in Section 3.2) as the distance metric. There is some evidence that colour moments work better for measuring similarity between images (Stricker & Orengo, 1995), but it has higher computational requirements and the benefit for video summarisation is unclear; we have therefore decided to keep using colour histograms.

Figure 3.4. Example of duplicate shots.
Each image represents one shot, while each line represents one cluster of duplicate shots.

From each cluster, the longest shot is taken as a candidate shot (the shot that is used for all further processing). This is based on the observation that the longer a shot is, the more likely it is to be important. Although not always accurate, this approach is chosen because it is computationally inexpensive.

3.4. AUTOMATIC KEYWORD DETECTION

Keywords or tags for a particular video can be detected from its subtitles. Digital broadcast TV often includes subtitles, either through live captioning (e.g. during live sports event) or from post-production (e.g. for delayed news). For many recent movies or TV series, the subtitles are available in their DVD releases, and can be extracted using programs such as SubRip[1] or Avidemux[2].

Subtitle texts are associated with video shots based on their timestamps, and a database of words appearing in the subtitle texts is then built. Stop word removal is used to filter out common words that are not suitable as keywords. The words are also reduced to stem form by applying the Porter stemming algorithm (Porter, 1980).

3.4.1. SPEECH TRANSCRIPTION

If the subtitles of a particular video are not available, speech transcription technologies can be used to extract this information. However, speech transcription is still an unsolved research topic. As a result, the accuracy of video subtitles obtained from speech transcription tends to be considerably lower than manually-written subtitles.

For testing purposes, we tried an existing a commercial product called Adobe Soundbooth CS4[3], which includes an automatic speech transcription module. The following was part of the output from a cricket video:

> fv fv fv fv fv fv fv who who what are the highlights of this penchant legislatures Cup match at the MCG Beach with the US trade and the West Indies debate citing pledged that the Sete on Tuesday the same two teams took the field for this match at the MCG ….

Often, named entities (persons, locations, organisations, etc.) are detected correctly. An example found here is “West Indies”. However, sometimes they are wildly off the mark, as in the “US trade” case, which actually should be “Australia”. As another example, we found this phrase while testing speech transcription in a news video regarding Iraq:

1 http://zuggy.wz.cz/dvd.php
2 http://fixounet.free.fr/avidemux/
3 http://www.adobe.com/products/soundbooth/

security personnel in Nebraska be carrying that for months ….

The actual phrase was “security personnel in Iraq have been carrying them for months”. If used in a tagging application, this would cause the news article to be mistakenly tagged with *Nebraska* instead of *Iraq*.

Although we are sure there has been significant research effort in the area of speech transcription, they are out of the scope of this research. Instead, we decided to focus on only dealing with videos with readily available subtitles. We will only assert that speech transcription *can* be used in our framework, if the user considers the limitations of current transcription systems acceptable.

3.4.2. OPTICAL CHARACTER RECOGNITION (OCR)

In some cases, named entities can also be detected by using optical character recognition on superimposed text, which is often present in videos that have gone through post-processing. For example, sports videos often show player names, while news videos often show interviewee names or news topics.

The method we chose for performing OCR is the open-source Tesseract OCR engine[4]. An objective comparison between Tesseract and several other OCR engines shows that it performs relatively well (Rice, Jenkins, & Nartker, 1995).

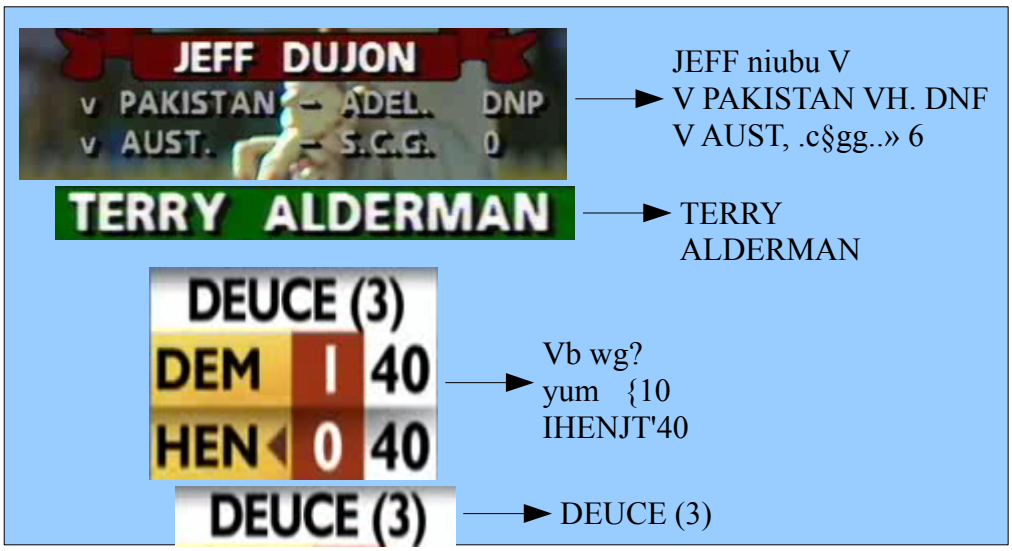

Figure 3.5. Sample OCR results

4 http://code.google.com/p/tesseract-ocr/

Figure 3.5 shows some sample output from the OCR. While the results look promising for localised text with plain background, it did not perform well on more complicated backgrounds. Therefore, it can only be used if the text background is mostly solid colour, and if the text location is known beforehand.

3.5. SEGMENT SCORING

In order to produce a meaningful ranking for each segment, features from the video must first be detected and given numeric values. These features may include, among other things: number of people, amount of motion, presence of speech, etc. Based on the features detected, a score will be given to each shot, and the shots will be ranked based on this score.

From the visual information contained in a video, we calculate the score of each shot based on a number of features, namely:

1. A set of numbers of faces (F) detected at 20-frame intervals along the shot;
2. A set of magnitude values of the motion (M), calculated on 20-frame sub-segments of the shot;
3. The length of the segment (L), in number of frames;
4. The number of shots that are in the same cluster as the shot, which corresponds to the retake frequency (R).

The following formula is used to calculate the final score:

$$Score = \frac{\text{mean}(F)}{\text{stddev}(F)+0.1} + \frac{\text{mean}(M)}{\text{stddev}(M)+0.1} + \log(L+1) + \log(\min(R, 10)) \quad . \quad (3.3)$$

For the face (F) and motion (M) measures, we divide the mean of the 20 frames with their standard deviations to emphasise shots with rapid changes. The length (L) and retake (R) measures are scaled logarithmically in order to de-emphasise their large values.

Additional measures can be added into this score. For example, we can

increase the score for shots containing certain keywords.

3.6. SATISFYING TIME CONSTRAINTS

In most cases, even after removing junk and redundant shots, the remaining shots still would not fit into the target summary length. Algorithms used to solve this fall into one or a combination of these categories:

- *Remove lower-ranked shots:* While this is a very useful technique, the risk of removing important information increases with the number of shots that have to be removed.

- *Speed up the shots:* The usefulness of this technique is limited by the maximum speed-up ratio that humans can tolerate before the video becomes difficult to understand and not pleasant to watch.

- *Sample a limited number of frames from each shot:* This includes techniques such as taking a number of frames from the middle of each shot, or taking a number of frames distributed among several sections of each shot (e.g. beginning, middle, and end of shot). This technique has similar implications to the first technique: the fewer frames can be taken from each shot, the more information is lost.

In order to minimise information loss and maximise the pleasantness of the summaries, we combine these three techniques using the following algorithm.

1. "Slice" the middle *MaxLength* of each shot, of the whole shot if it is shorter than *MaxLength*. For our TRECVid evaluation, *MaxLength* is set to 60 frames.

2. Sort the list of shots by their scores.

3. If all slices fit into the summary of length *T* with a maximum speed-up rate of *MaxSU*, the output video is generated, containing all the slices at the calculated speed-up rate.

4. Otherwise, start removing slices with lower importance scores until

the remaining slices fit into *T* with at most *MaxSU* speed-up rate.

In this algorithm, the target length (*T*) is set based on user requirement. Maximum slice length (*MaxLength*) determines how much of each shot is taken for the summary video, while maximum speed-up (*MaxSU*) determines the highest speed-up ratio allowed for the whole video summary; note that all slices have the same speed-up value. The latter two variables control the consistency in the final output video and are aimed to improve the "pleasant rhythm/tempo" evaluation measure.

Some examples of the first step in the algorithm (shot slicing):

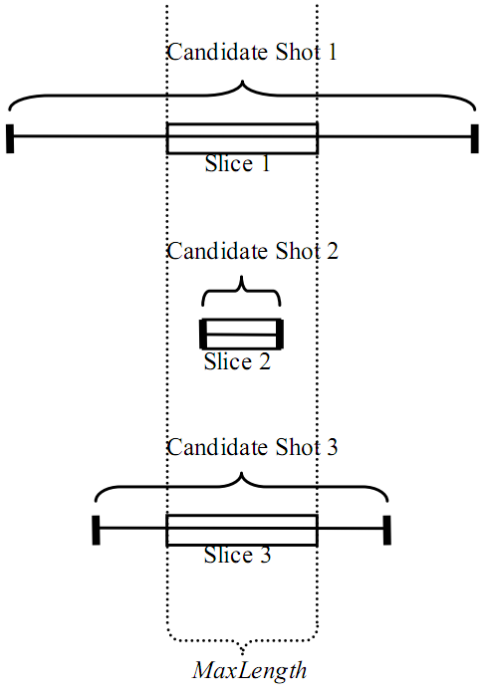

Figure 3.6. Examples of shot slicing

Steps 2–4 of the algorithm can be expressed in the following pseudocode.

```
function TimeFit(S, T, MaxSU) {
        // S is an array of slices, sorted by descending score.
        // T is the target video length.
        // MaxSU is the maximum speed-up allowed.

        loop until S is empty {
                L = total length of S

                // Case 1: The slices fit into T.
                if L ≤ T: return S

                // Case 2: The slices fit into T after limited speed-up.
                // su is the speed-up required for the slices to fit into T.
```

```
                    su = T / L
                    if su ≤ MaxSU: return SpeedUp(S, su)

                    // Case 3: Cannot fit all slices.
                    Remove last element of S
                }
            }
```

3.7. CREATING WEB VISUALISATION

In this section, we present a hybrid visualisation method for summarising a video. This method combines the visual-based information in the form of keyframes extracted from the video, as well as textual-based information in the form of keywords taken from the video subtitles. The visualisation shows shots from story clusters within the video, combined with a tag cloud of keywords for each cluster and for the whole video.

In order to show the keywords within the whole video or a particular cluster, we chose to visualise them as a tag cloud of the highest-scored keywords, sorted alphabetically. The size of the keyword text in the output is scaled based on the score. Therefore, higher-valued keywords are shown in larger font sizes.

Cluster keyframes are shown in thumbnail size below the keywords tag cloud. Each thumbnail is accompanied by a timestamp indicating where the shot appears in the video. When the user clicks on a thumbnail, the full-size picture is displayed.

Combined together, the keywords tag cloud and image thumbnails give users a visual and textual overview of stories and themes within the a video.

3.7.1. STORY SEGMENTATION

To segment the video into stories, we extend the clustering algorithm from Section 3.3.2 to resemble the time-constrained hierarchical clustering approach proposed by Yeung & Yeo (1996). Two shots are linked into one cluster if they satisfy these two criteria: (1) the histogram difference between the shots fall below a set threshold determined from experiments; and (2) the shots occur within a set time difference of each other, ensuring that shots far apart in the video are not accidentally clustered together.

Each of the resulting clusters shows a particular story, for example, a conversation. For the purpose of building a web visualisation, we filter out story clusters that are too short (noise); clusters that are less than 15 seconds in length are removed. This leaves the clusters that cover significant parts of the video.

3.7.2. TAG RANKING ALGORITHM

The score for a particular tag/keyword within the whole video depends on: (1) uniqueness of keyword in the video; (2) uniqueness of keyword in the language. These observations result in equation 3.1.

$$score_t = \frac{n_t}{n} \times \log \frac{1}{F_t} \quad , \tag{3.4}$$

where n_t is the occurrence of term t in the episode; n is the occurrence of all terms in the episode; and F_t is the frequency of t in spoken context. Leech, Rayson, & Wilson (2001) provide a list of word frequencies in spoken English.

Keyword scores for each story cluster are calculated similarly, except we use a measure like tf-idf in order to compare the word frequency within the cluster with the word frequency in the whole episode. This increases the value of unique keywords within the particular cluster. This tf-idf value is then combined with the inverse word frequency in spoken English. We define the score of a particular term in a cluster as:

$$score_{clust,t} = \frac{n_{clust,t}}{n_{clust}} \times \log \frac{C}{C_t} \times \log \frac{1}{F_t} \quad , \tag{3.5}$$

where $n_{clust,t}$ is the occurrence of term t in the cluster *clust*; n_{clust} is the occurrence of all terms in *clust*; C is the number of clusters in the episode; C_t is the number of clusters containing t; and F_t is the frequency of t in spoken English.

3.7.3. WEB VIDEO BROWSER

By combining the web visualisation with its original video, we can come up with a unique video viewer that allows browsing within the video itself. This video browser would, for example, allow users to click on a keyframe thumbnail to view the represented video shot. Another possibility is to display contextual information (e.g. tags, images, articles, advertising) for each segment as it is playing; when the

video goes to another segment, the contextual information also changes.

Figure 3.7 shows a mock-up of a video browser application for mobile devices, which displays contextual information related to the current segment. An "interest graph" is used as the seek bar to show occurrence distribution of all tags throughout the video.

Figure 3.7. Mock-up of video browser showing contextual information

The technology to embed videos in a web page has existed through numerous video player plugins or, more recently, through the Adobe Flash platform[5]. The HTML5 draft[6], which is partially implemented in modern browsers, specifies a new `video` element that can also be used for this purpose.

Section 4.2 shows a web video browser prototype that we produced for a demonstration, featuring tags and clickable key frames.

5 http://www.adobe.com/flashplatform/
6 http://www.w3.org/TR/html5/

Chapter 4: Results and Discussion

4.1. VIDEO SKIM CREATION

To evaluate our summarisation framework, we participated in the TRECVid 2008 Video Summarisation task. TRECVid is an "international benchmarking activity to encourage research in video information retrieval" (Smeaton, Over, & Kraaij, 2006). In 2007, a *video summarisation* task was introduced into TRECVid. This event answered concerns regarding evaluation of video summarisation systems by providing a set of videos to be summarised by researchers, and evaluating the outputs based on specific guidelines.

The test dataset provided consists of 39 *rushes* videos, each approximately 10–40 minutes long (over 17 hours in total). Rushes are raw film recordings that are still in their original, unedited state. They contain many so-called "junk" shots, mainly artefacts from the recording stage. The techniques we use for filtering out these junk shots are explained in Section 3.3.1.

The evaluation was performed by human judges employed by the TRECVid organisers on seven measures: ground truth inclusion, tempo and rhythm, amount of junk, redundancy, evaluation time, summary length, and creation time. A detailed description of the measures used in the evaluation is available in TRECVid's summary paper (Over, Smeaton, & Awad, 2008) and in Section 2.6 of this thesis.

While observing our evaluation results and those of other participants, we identified three major patterns in the objectives of the different submissions, as shown on Figure 4.1:

1. Pattern 1: Short length, high pleasantness;
2. Pattern 2: Medium length, high pleasantness, medium ground truth inclusion;
3. Pattern 3: High ground truth inclusion.

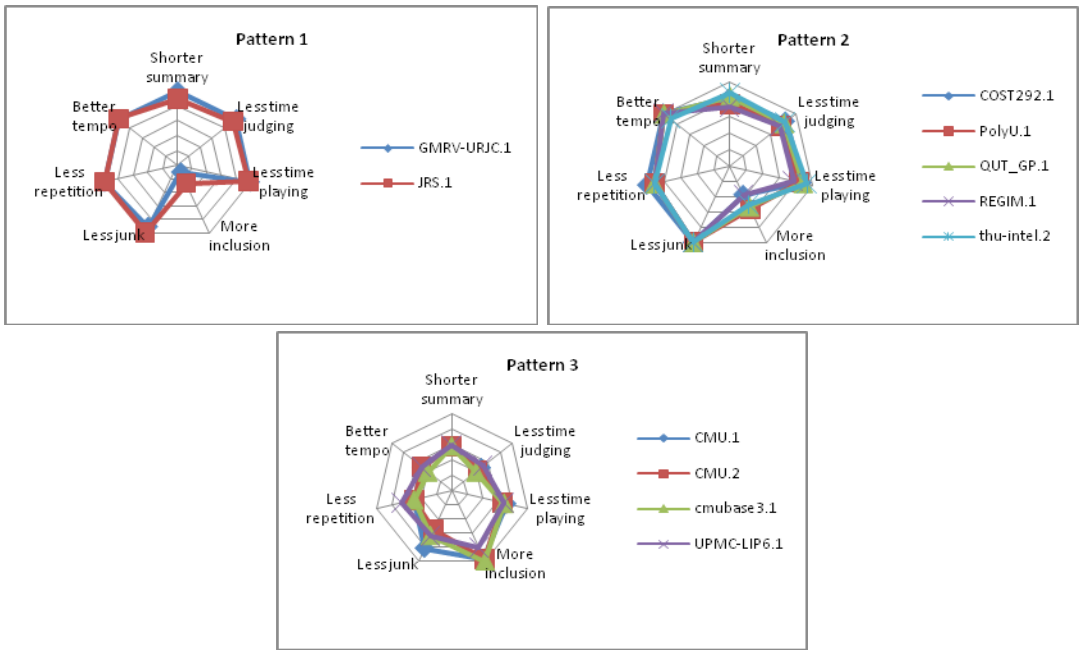

Figure 4.1. Three patterns in the TRECVid evaluation results. Note that the axes do not scale in the same way; they are only meant to show participants' scores relative to each other.

Our algorithm (labelled QUT_GP.1) falls into Pattern 2, which maximizes the three pleasantness (user satisfaction) measures—better tempo (TE), less repetition (RE), less junk (JU)—without sacrificing too much ground truth inclusion. In line with our aim of creating pleasant video summaries, our system succeeds in obtaining high scores in these three measures that we consider represent the pleasantness of the summaries, as shown on Table 4.1.

Rank	Systems	Pleasantness (TE+RE+JU) / 3
1	COST292.1, JRS.1	3.6667
2	PolyU.1, **QUT_GP.1**, REGIM.1	3.5567
3	GMRV-URJC.1	3.5533

Table 4.1. Systems with top three pleasantness scores

The "shorter summary" and "more inclusion" measures seem to be opposites of each other; short summaries yield less ground truth inclusion, while more ground truth inclusion is possible given longer summaries. Figure 4.1 shows this relationship: systems producing short summaries tend to neglect ground truth inclusion (Pattern 1), while systems that focus on inclusion produce long summaries and are less pleasant (Pattern 3). As with other algorithms in Pattern 2, we position ourselves in the middle of both extremes, producing short summaries with reasonable

ground truth inclusion (see Figure 4.2).

Parameters in our algorithm can be modified in order to achieve results more similar to the first and third patterns. The maximum video speed-up (*MaxSU*) can be increased to increase the ground truth inclusion at the cost of pleasantness (tempo). The maximum slice length (*MaxLength*) can also be decreased to obtain the same effect. If ground truth inclusion is not important, the maximum summary length (*T*) can be reduced, and the results will be closer to pattern 1. This shows the flexibility of our algorithm, as these different parameters can be tweaked depending on preference.

In terms of efficiency, our system ranked eighth in the average summary creation time (see Figure 4.2), which is the best among the 6 systems with highest pleasantness scores mentioned in Table 4.1, even though it was running on medium-end laptop computers. However, this result should be taken with caution because the processing times are self-reported by each participant, and there was no standard hardware nor measurement method specified. The machines running our code consist of an Intel Core 2 Duo 1.83 GHz with 2 GB RAM running Windows Vista (for shot segmentation, clustering, filtering, and scoring) and an Intel Core 2 Duo 2.16 GHz with 2 GB RAM running Mac OSX (for shot ranking, time fitting, and video writing). Note that the two machines were not running in parallel, and we do not take into account the post-processing time to compress the video files.

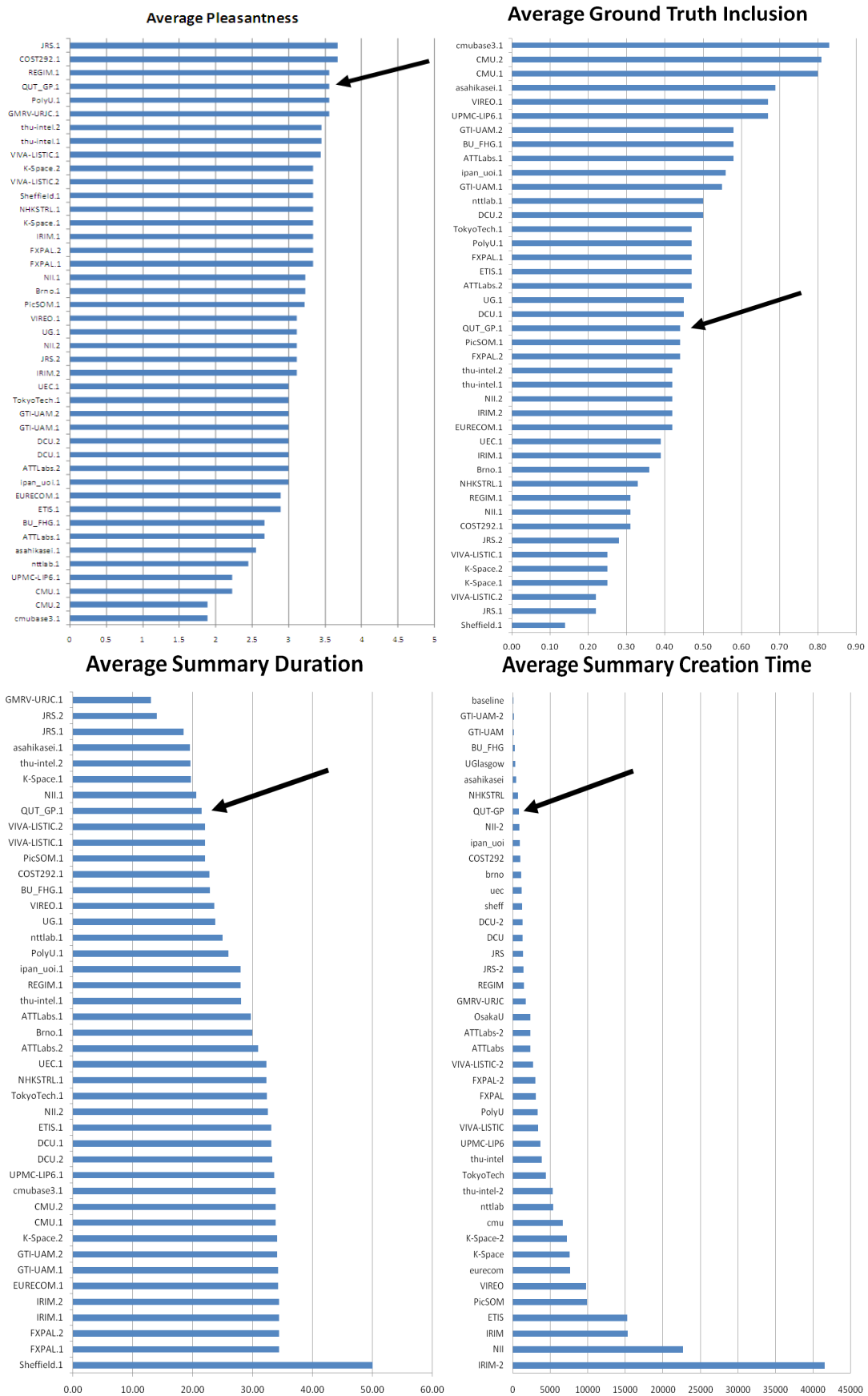

Figure 4.2. TRECVid 2008 evaluation results

4.2. WEB VIDEO BROWSER

To demonstrate the web video browser mentioned in Section 3.7.3, we created a simple HTML-based video player that displays information related to persons shown in a video, and allows the user to skip to parts of the video where they are mentioned. Note that this was developed demonstration purposes, and no evaluation was performed on the results.

Tags in the video browser are obtained by submitting the video subtitles to the Calais web service[7] and parsing the output for *Person* named entities. These tags are then correlated with the original subtitles to obtain the timestamps where they are mentioned in the video. This allows us to create a thumbnail for each name occurrence and let the user jump to that point by clicking on the thumbnail. In addition, we also display the first paragraph of each person's Wikipedia[8] article (which, in the Wikipedia style, usually contains a summary of the article).

Figure 4.3. Web video browser prototype for news and sports videos

7 http://www.opencalais.com/
8 Http://www.wikipedia.org/

4.3. STORY-BASED WEB VISUALISATION

This section details objective evaluation results on the alternative visualisation method presented in Section 3.7. The web visualisation method was applied on four popular TV series. The particular series and episodes used in this experiment were selected arbitrarily in order to demonstrate the generality of our method.

4.3.1. DESCRIPTION OF DATASET

The ground truth used in the experiment is partly based on the "episode recaps" found on TV.com[9]. Using this, we determined stories contained in the videos. These stories are then matched with the stories we obtained in the summary.

The first video that we used for experiment is from the series "Doctor Who". This video is characterised by a straightforward plot, with no side stories or flashbacks. The recording in this video uses a lot of close-up shots. The second video comes from "Battlestar Galactica", which has several parallel plots with characteristically distinct environment backgrounds, taking place in two different planets and a space ship. There are also several flashbacks. The third video is from the series "Desperate Housewives". This video also has several parallel plots happening around the same time at various locations. There is a recurring flashback that is shown a few times. Compared to the other videos, this video is a lot more visually diverse and is shot with more kinds of backgrounds. The last video in the experiment dataset comes from "Terminator". The plot in this video is rather complex and involves three timelines: the "past", the "present", and the "future". These three timelines are shown interspersed with each other, giving the impression of flashbacks. Unlike flashbacks in the second and third videos, however, the flashbacks in this video are central to the whole episode storyline and are not shown in a distinct (saturated) visual style.

4.3.2. RESULTS AND SAMPLE OUTPUT.

The evaluation on this technique was performed objectively on the tagging and story segmentation results. To evaluate the tagging module, we manually compared the automatically extracted tags with the actual story and note their similarities. To

9 http://www.tv.com/

evaluate the story segmentation module, we compared our result with the manually-created TV.com summaries and measure their overlap.

Figure 4.4(a) shows keywords from the Doctor Who episode. This episode depicts characters watching the *death* of *planet Earth* due to *heat* from the sun. *Humans* and a*liens* are watching from space, and the plot involves someone tampering with the *sunfilter* of the spaceship windows (causing them to *descend*), endangering the ship *guests*. The story cluster shows the exchange of *gifts* between *guests*. One of the characters gave "the *air* of [his] *lungs*", while another guest gave "the gift of bodily *salivas*". The *Jolco* keyword shown prominently here is a name.

Figure 4.4(b) shows tags obtained from the Battlestar Galactica video. In this series, the *fleet* refers to a number of space ships that the Galactica ship protects, and *Cylons* are a type of humanoid robots featured in the series. *Chief*, *Cally*, and *Cottle* are names of some of the ship's crew. The episode plot is about the Galactica "*jumping*" to the wrong location. The story cluster in depicts characters *Helo* and *Kara* (code-named *Starbuck*) talking about a *Cylon* robot named *Sharon* who previously *lied* to them.

Figure 4.4(c) shows tags from the Desperate Housewives video. These include names of important characters including *Bree*, *Ian*, *Jane*, *Mike*, *Monique*, *Orson*, and *Zach*. Other relevant tags include *remember*, *memory* (a character has amnesia and lost his memory), and *date* (several couples in the episode are dating). The story cluster in Fig 8 shows a flashback of Mike coming from back a *hardware* store to fix a *leaking sink*. Other tags are closely related to this story and mentioned in conversations, e.g. *damage*, *pipe*, *seeping*, *wash*, and *water*.

Figure 4.4(d) comes from the TV series Terminator. Some personal names such as *Roger*, *David*, *Lauren*, and *Sarah* are picked up as keywords. The plot involves a *cyborg* from the *future* coming to *kill* an unborn *baby* who has *immunity* against a certain disease. The story cluster shows a conversation within the episode, with the topic of *cyborgs* and how one of the characters has a "not exactly *legal*" dealing with a *cybernetics company*. The topic of *birdhouses* comes up during small talk.

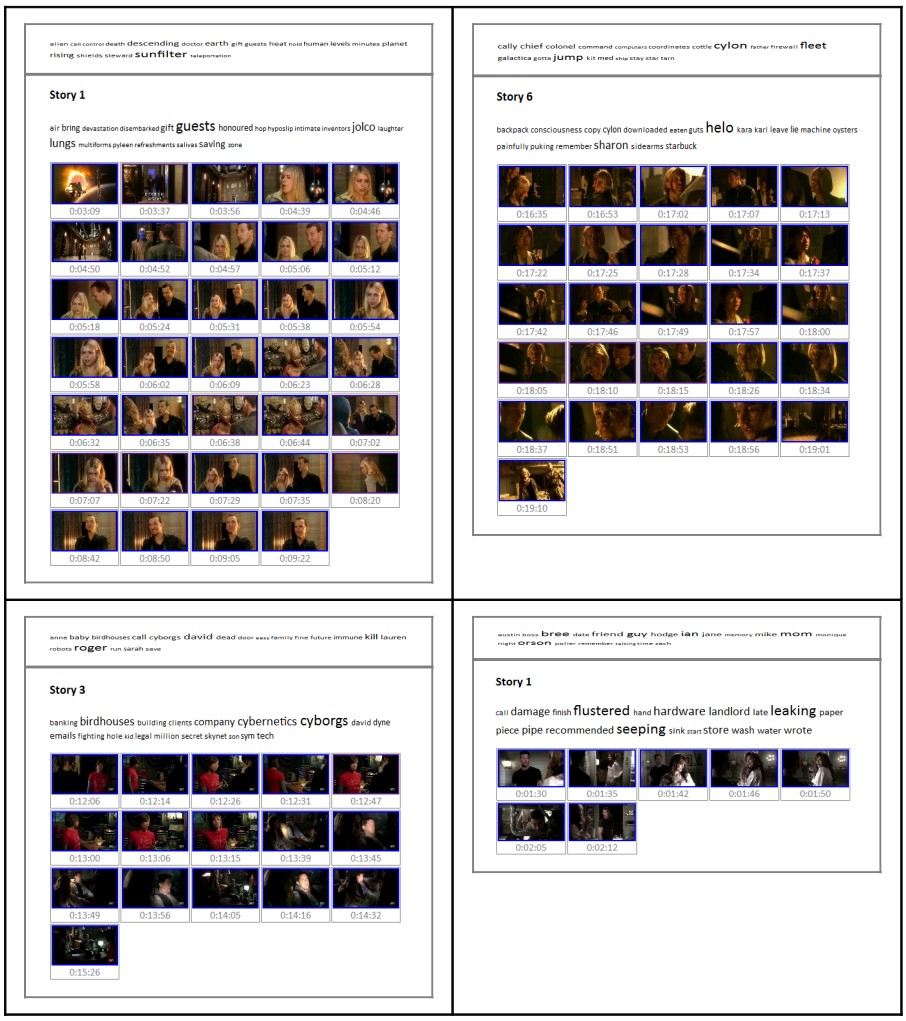

Figure 4.4. *Keywords and sample story clusters from four videos. Clockwise from top left: (a) Doctor Who; (b) Battlestar Galactica; (c) Desperate Housewives; (d) Terminator.*

As can be observed from the description above and Figure 4.4, the tags picked up by the system correspond well to the story topics. The images shown below the tags represent shots contained in each story. Comparing this story segmentation output with the textual description from TV.com, we obtain the following result.

Video	Actual	Found	Accurate	Precision	Recall	F_1
Doctor Who	18	23	14	77.78%	60.87%	68.29%
Battlestar Galactica	28	17	15	53.57%	88.24%	66.67%
Desperate Housewives	26	22	16	61.54%	72.73%	66.67%
Terminator	23	17	15	65.22%	88.24%	75.00%
Average				64.53%	77.52%	69.16%

Table 4.2. *Accuracy of the story clustering method on the test videos*

Table 4.2 shows that the simple keyframe-based story clustering described in

Section 3.7.1 is able to achieve around 70% F_1 measure. The worst result in terms of precision is the Battlestar Galactica video. This is because some of the stories take place in similarly saturated background, which causes their histograms to be more uniform than they should be; this, in turn, means that some stories are mixed together in the output.

Chapter 5: Conclusion

5.1. CONTRIBUTIONS OF THIS RESEARCH

Video summarisation can potentially overcome the problem of information overload due to the large number of videos available to us. A major part of video summarisation deals with the definition of important events in a video. This thesis has presented several existing techniques to solve this problem, as well as new ones that have been developed and integrated into the video summarisation framework.

An implementation of the video summarisation framework was tested during the TRECVid 2008 event and was shown to perform really well in the user satisfaction / pleasantness measures (low junk, low duplicates, enjoyable tempo). It is reasonably accurate (average content inclusion) while producing summary videos of medium length and being very fast in terms of processing speed. Important factors in this success are the duplicate removal process, and the shot ranking and time fitting algorithms.

In addition to creating video skims, we have shown that the framework is capable of being used to produce a different visualisation in the form of web pages. In this type of visualisation, the main aim is to convey story lines contained within a video to the user. The main parts of this visualisation are sets of keyframe clusters and keyword tag clouds for each cluster and for the whole episode. Besides the potential application in an website, this visualisation can also be useful for browsing personal video libraries or for commercial video archiving. The method can be easily adapted for any type of videos. Because the story clustering is independent of the subtitles, the method is still useful for visualising videos where the complete subtitles are not available or only available in low quality, for example due to live captioning.

5.2. FUTURE WORK

The advantages brought by an automatic video summarisation system are obvious, but real-world adoption of such system has been rare. This thesis has been a

step forward in the effectiveness and efficiency of summarisation techniques, and further improvements in these areas will hopefully help this adoption rate.

In terms of important events detection, there are still features that have not been developed and tested in this framework, for example audio pitch and intensity, or user-created content. The latter is an interesting research direction by itself, one that has not been explored much.

An expansion of this work would be to create a summary of multiple videos, for example one whole season of a TV series. This will provide better input to the tagging and will give an opportunity for more interesting visualisation features, because the system can pick up common topics and entities (e.g. person, location, organisation) throughout the videos.

The web visualisation method would also benefit from using a better scene segmentation method instead of simple clustering. While the story clustering algorithm described here works quite well, sometimes shots in one story exhibit distinct histogram patterns, which the clustering method often cannot take into account.

Bibliography

Agnihotri, L., Devara, K. V., McGee, T., & Dimitrova, N. (2001). Summarization of video programs based on closed captions. In *Proceedings of SPIE, 4615* (pp. 599–607).

Bagga, A., Hu, J., Zhong, J., & Ramesh, G. (2002). Multi-source combined-media video tracking for summarization. In *Proceedings of the 16th International Conference on Pattern Recognition, 2* (pp. 818–821).

Beran, V., Hradiš, M., Zemčík, P., Herout, A., & Řezníček, I. (2008). Video summarization at Brno University of Technology. In *Proceedings of the TRECVid Video Summarization Workshop* (pp. 31–34).

Bertini, M., Del Bimbo, A., & Nunziati, W. (2006). Automatic detection of player's identity in soccer videos using faces and text cues. In *Proceedings of the 14th Annual ACM International Conference on Multimedia* (pp. 663–666).

Bertini, M., Del Bimbo, A., & Pala, P. (2001). Content-based indexing and retrieval of TV news. *Pattern Recognition Letters, 22*(5), 503–516.

Chen, F., Adcock, J., Cooper, M. (2008). A simplified approach to rushes summarization. In *Proceedings of the TRECVid Video Summarization Workshop* (pp. 60–64).

Chen, F., Cooper, M., & Adcock, J. (2007). Video summarization preserving dynamic content. In *Proceedings of the International Workshop on TRECVID Video Summarization* (pp. 40–44).

Christel, M. G., Hauptmann, A. G., Lin, W. H., Chen, M. Y., Yang, J., Maher, B., & Baron, R. V. (2008). Exploring the Utility of Fast-Forward Surrogates for BBC Rushes. In *Proceedings of the TRECVid Video Summarization Workshop* (pp. 35–39).

Chua, T. S., Chang, S. F., Chaisorn, L., & Hsu, W. (2004). Story boundary detection in large broadcast news video archives: Techniques, experience and trends. In *Proceedings of the 12th Annual ACM International Conference on Multimedia* (pp. 656–659).

comScore. (2009). *YouTube surpasses 100 million U.S. viewers for the first time*. Retrieved February 11, 2010, from http://www.comscore.com/Press_Events/Press_Releases/2009/3/YouTube_Surpasses_100_Million_US_Viewers

comScore. (2010). *comScore releases January 2010 U.S. online video rankings*. Retrieved February 11, 2010, from http://www.comscore.com/layout/set/popup/Press_Events/Press_Releases/2010/3/comScore_Releases_January_2010_U.S._Online_Video_Rankings

De Santo, M., Percannella, G., Sansone, C., & Vento, M. (2004). A comparison of unsupervised shot classification algorithms for news video segmentation. In *Lecture Notes in Computer Science, 3138* (pp. 233–241).

Dumont, E., & Mérialdo, B. (2007). Split-screen dynamically accelerated video summaries. In *Proceedings of the TRECVid Video Summarization Workshop* (pp. 55–59).

Dumont, E., & Mérialdo, B. (2008). Sequence alignment for redundancy removal in video rushes summarization. In *Proceedings of the TRECVid Video Summarization Workshop* (pp. 55–59).

Ekin, A., & Tekalp, M. (2003). Generic play-break event detection for summarization and hierarchical sports video analysis. In *Proceedings of the 2003 International Conference on Multimedia and Expo, 1* (pp. 169–172).

Erol, B., Lee, D. S., & Hull, J. (2003). Multimodal summarization of meeting recordings. In *Proceedings of the 2003 International Conference on Multimedia and Expo, 3* (pp. 25–28).

Gao, X., & Tang, X. (2002). Unsupervised video-shot segmentation and model-free anchorperson detection for news video story parsing. *IEEE Transactions on Circuits and Systems for Video Technology, 12*(9), 765–776.

Hauptmann, A. G., Christel, M. G., Lin, W. H., Maher, B., Yang, J., Baron, R. V., & Xiang, G. (2007). Clever clustering vs. simple speed-up for summarizing rushes. In *Proceedings of the International Workshop on TRECVID Video Summarization* (pp. 20–24).

Hua, X., Lu, L., & Zhang, H. (2003). AVE: automated home video editing. In *Proceedings of the Eleventh ACM International Conference on Multimedia* (pp. 490–497).

Itti, L., Koch, C., & Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 20*(11), 1254–1259.

Koprinska, I., & Carrato, S. (2001). Temporal video segmentation: A survey. *Signal Processing: Image Communication*, 16(5), 477–500.

Leech, G., Rayson, P., Wilson, A. (2001). Word frequencies in written and spoken English: Based on the British National Corpus. London: Longman.

Lie, W., & Lai, C. (2005). News video summarization based on spatial and motion feature analysis. In *Advances in multimedia information processing - PCM 2004*, Lecture Notes in Computer Science, 3331 (pp. 246–255).

Lupatini, G., Saraceno, C., & Leonardi, R. (1998). Scene break detection: A comparison. In *Proceedings of the Eighth International Workshop on Research Issues in Data Engineering: Continuous-Media Databases and Applications* (pp. 34–41).

Ma, Y. F., Lu, L., Zhang, H. J, & Li, M. (2002). A user attention model for video summarization. In *Proceedings of the Tenth ACM International Conference on Multimedia* (pp. 533–542).

Money, A. G., & Agius, H. (2008a). Feasibility of personalized affective video summaries. In *Affect and emotion in human-computer interaction*, Lecture Notes in Computer Science, 4868 (pp. 194–208).

Money, A. G., & Agius, H. (2008b). Video summarisation: A conceptual framework and survey of the state of the art. *Journal of Visual Communication and Image Representation*, *19*(2), 121–143.

Over, P., Smeaton, A. F., & Kelly, P. (2007). The TRECVID 2007 BBC rushes summarization evaluation pilot. In *Proceedings of the International Workshop on TRECVID Video Summarization* (pp. 1–15).

Over, P., Smeaton, A. F., & Awad, G. (2008). The TRECVid 2008 BBC rushes summarization evaluation. In *Proceedings of the TRECVid Video Summarization Workshop* (pp. 1–20).

Pan, H., van Beek, P., & Sezan, M. (2001). Detection of slow-motion replay segments in sports video for highlights generation. In *Proceedings of the 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing, 3* (pp. 1649–1652).

Patel, N. V., & Sethi, I. K. (1996). Compressed video processing for cut detection. *IEE Proceedings – Vision, Image and Signal Processing, 153*(6), 315–323.

Pickering, M. J., Wong, L., & Rüger, S. M. (2003). ANSES: summarisation of news video. In *Proceedings of the 2nd international conference in image and video retrieval* (pp 425–434).

Porter, M. F. (1980). An Algorithm for Suffix Stripping. *Program: Electronic Library and Information Systems, 14*(3), 130–137.

Rice, S. V., Jenkins, F. R., & Nartker, T. A. (1995). *The fourth annual test of OCR accuracy*. Retrieved March 12, 2010, from http://www.isri.unlv.edu/downloads/AT-1995.pdf

Sato, T., Kanade, T., Hughes, E. K., & Smith, M. (1998). Video OCR for digital news archive. In *Proceedings of the IEEE International Workshop on Content-Based Access of Image and Video Database* (pp. 52–60).

Shao, X., Xu, C., Maddage, N.C., Tian, Q., Kankanhalli, M. S., & Jin, J. S. (2006). Automatic summarization of music videos. *ACM Transactions on Multimedia Computing, Communications, and Applications, 2*(2), 127–148.

Smeaton, A. F., Gurrin, C., Lee, H., McDonald, K., Murphy, N., O'Connor, N. E., et al. (2004). The Físchlár-News-Stories system: personalised access to an archive of TV news. In *Proceedings of RIAO 2004 - Coupling Approaches, Coupling Media and Coupling Languages for Information Retrieval*.

Smeaton, A. F., Over, P., & Kraaij, W. (2006). Evaluation campaigns and TRECVid. In *Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval* (pp. 321–330).

Smith, M., & Kanade, T. (1997). Video skimming and characterization through the combination of image and language understanding techniques. In *Proceedings of the 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 775–781).

Stricker, M. A., & Orengo, M. (1995). Similarity of color images. In *Proceedings of SPIE, 2420* (pp. 381–392).

Truong, B. T., & Venkatesh, S. (2007). Video abstraction: A systematic review and classification. *ACM Transactions on Multimedia Computing, Communications, and Applications, 3*(1), 1–37.

Tjondronegoro, D., Chen, Y. P. P., & Pham, B. (2004). Highlights for more complete sports video summarization. *IEEE Multimedia, 11*(4), 22–37.

Wang, T., Gao, Y., Li, J., Wang, P. P., Tong, X., Hu, W., Zhang, Y., Li, J. (2007). THU-ICRC at rush summarization of TRECVID 2007. In *Proceedings of the International Workshop on TRECVID Video Summarization* (pp. 79–83).

Xie, Y. X., Luan, X. D., Lao, S. Y., Wu, L. D., Xiao, P., & Wen, J. (2004). EDU: A model of video summarization. In *Image and video retrieval*, Lecture Notes in Computer Science, 3115 (pp. 106–114).

Xu, M., Chia, L. T., Yi, H., Rajan, D. (2006). Affective content detection in sitcom using subtitle and audio. In *The 12th International Multi-Media Modelling Conference Proceedings* (pp. 129–134).

Yeung, M.M., Yeo, B.L. (1996). Time-constrained clustering for segmentation of video into storyunits. In: *Proceedings of the 13th International Conference on Pattern Recognition, 3*, (pp. 375–380).

Yi, H., Rajan, D., Chia, L. T. (2005). Semantic video indexing and summarization using subtitles. In *Advances in Multimedia Information Processing - PCM 2004*, Lecture Notes in Computer Science, 3331 (pp. 634–641).

YouTube Inc. (2010). *Oops pow surprise...24 hours of video all up in your eyes!* Retrieved March 29, 2010, from http://youtube-global.blogspot.com/2010/03/oops-pow-surprise24-hours-of-video-all.html

Yu, B., Ma, W. Y., Nahrstedt, K., & Zhang, H. J. (2003). Video summarization based on user log enhanced link analysis. In *Proceedings of the Eleventh ACM International Conference on Multimedia* (pp. 382–391).

Yu, J. C. S., Kankanhalli, M. S., & Mulhen, P. (2003). Semantic video summarization in compressed domain MPEG video. In *Proceedings of the 2003 International Conference on Multimedia and Expo, 3* (pp. 329–332).