This is a preliminary version of an article published in
Proc. of IS&T/SPIE Electronic Imaging (EI), pp. 417 – 428, San José, CA, January 2004
by Stephan Kopf, Thomas Haenselmann, Dirk Farin, Wolfgang Effelsberg

Automatic Generation of Summaries for the Web

Stephan Kopf, Thomas Haenselmann, Dirk Farin and Wolfgang Effelsberg

Dept. of Computer Science IV, University of Mannheim, Germany

ABSTRACT

Many TV broadcasters and film archives are planning to make their collections available on the Web. However, a major problem with large film archives is the fact that it is difficult to search the content visually. A video summary is a sequence of video clips extracted from a longer video. Much shorter than the original, the summary preserves its essential messages. Hence, video summaries may speed up the search significantly.

Videos that have full horizontal and vertical resolution will usually not be accepted on the Web, since the bandwidth required to transfer the video is generally very high. If the resolution of a video is reduced in an intelligent way, its content can still be understood. We introduce a new algorithm that reduces the resolution while preserving as much of the semantics as possible.

In the MoCA (movie content analysis) project at the University of Mannheim we developed the video summarization component and tested it on a large collection of films. In this paper we discuss the particular challenges which the reduction of the video length poses, and report empirical results from the use of our summarization tool.

Keywords: Video summarization, region-of-interest, skimming, video content analysis

1. INTRODUCTION

The number and volume of digital video libraries is growing rapidly. TV broadcasters and other private and public film archives are digitizing their film collections. Local users of the archives have the opportunity to access the material, but it is also often desirable to make the content available to the public at large via the Web.

Since a major problem with large film archives is the difficulty in visually searching their content, additional metadata information for each film is stored. Relevant films can be found by searching the index of metadata information. Typically, the result of a query is a list of key frames with some textual information. Furthermore, it would be desirable to have short *video summaries* that contain the essence of a longer film. A video summary is a short video clip that has been extracted from a longer video. Much shorter than the original, the summary preserves its essential messages. A summary does not change the presentation medium; image and audio information is available to the user.

Without a reduction in resolution, the bandwidth required to transfer the video is very high. If the resolution of a video is reduced in an intelligent way, its content can still be understood. We introduce a new algorithm that reduces the resolution while preserving as much of the semantics as possible.

Another area that would benefit from automatically generated low-resolution videos is the transmission on mobile devices (PDAs or mobile phones). Many of these support the playback of videos, and wireless LAN is available in many places. The algorithm in this paper offers the possibility to generate low-resolution videos or video summaries. It has been optimized for mobile devices.

The principle behind our new approach is to create high-quality video summaries even at a very low image resolution. To reduce the size, we scale the video and/or select a region (window) within the video. We combine four methods to select the most relevant region:

- Regions that contain high-level semantic information should be selected, e. g., text regions, faces, people, and moving objects.
- Irrelevant regions should not be part of the summary. E. g., many frames in digitized videos have a small border with black pixels or noise.
- The selected region in a frame is scaled to the size of the final summary. It is possible that due to its small size the content in a scaled frame can no longer be recognized. If this is the case, a different region should be selected.

- The position and size of the regions is not fixed in consecutive frames. A virtual camera motion may increase the visible information in the shot.

The remainder of this paper is organized as follows: Section 2 describes related work in the area of video presentation, video summarization and the detection of relevant regions in images. Section 3 gives an overview of our video summarization application. Sections 4 and 5 describe the automatic computation of features and the detection of the most relevant region in a frame. The selection of relevant shots and the generation of the summary is presented in 6. We then present the results and the outlook in Section 7 and 8.

2. RELATED WORK

Many tools have been developed to generate a compact representation of a long video. This process is usually called *video summarization, video skimming* or *video abstracting.* Most approaches either analyze visual features alone, extract key frames, or calculate background mosaic images on a per-shot basis. Many applications allow quick navigation based on the key frames; in response to clicking on a key frame, they play the corresponding shot in the video.

The MoCA (movie content analysis) abstracting tool was one of the first tools to generate moving summaries from feature films automatically.[1] Since the system was initially developed to generate trailers of feature films, a major component was the detection of events of particular relevance such as explosions, gunfire or dialogs.

The Informedia Digital Video Library project[2] at the Carnegie Mellon University has developed two applications to visualize video content. The first one provides an interface to generate and display so-called video skims. [3] Important words are identified in the textual transcript of the audio. Text and face recognition algorithms detect relevant frames. Video skims are generated based on the results of the automatic analysis. Additionally, an interface for browsing video collections has been introduced, a collage in which information from multiple video sources is summarized and presented. [4,5]

A simple approach to reducing the length of a video is to increase the frame rate and thus speed up the playback process (time compression).[6] IBM's CueVideo system uses this approach and modifies the time scale of the audio signal. [7]

Lienhart[8] describes an approach to video summarization tailored especially to home videos. Text segmentation and recognition algorithms are used to identify the date and time inserted into the frames by the camcorder. Hierarchical clusters are built with shots based on the recording time. A heuristic selects shots based on these clusters without actually analyzing the content of the home video.

Numerous other methods have been proposed, e.g., a comic-book style of presentation to arrange the key frames [9,10] or summaries based on background mosaic images.[11] A method to summarize and present videos by analyzing the underlying story structure was proposed very early by Yeung et al..[12]

Web pages, PDAs, or mobile phones require a special presentation of images and videos due to their limited resolution. Fan et al.[13,14] have introduced a selective attention model that gives priority to the semantically most important regions in an image, where bandwidth or computing power are limited. None of these approaches addresses the selection of the most relevant regions in videos for presentation in terms of a video summary with a reduced resolution.

3. SYSTEM OVERVIEW

The general approach to generating a video summary is to analyze the video, identify its specific features and use a heuristic to select the most relevant shots for the summary. In our approach we analyze syntactic information (e. g., camera motion, color distribution, contrast) and semantic information (e. g., text, faces, people, or moving objects) to locate relevant shots.

If a summary is to be made available on the Web, videos that have full horizontal and vertical resolution will usually not be accepted due to limited bandwidth. Much information may be lost if the video summary is scaled to a smaller size. Although the selection of the shots is based on semantic information, it is possible that due to its reduced size this information will no longer be available. E. g., to be legible, a text must have a minimum character size.

The principle behind our new approach is to create high-quality video summaries even at a very low image resolution. Before a video summary can be generated, two selections have to be made: shots must be selected (temporal selection) and the image size must be decreased (spatial selection). To reduce the image size, we scale the video and/or select a region (window) within the video.

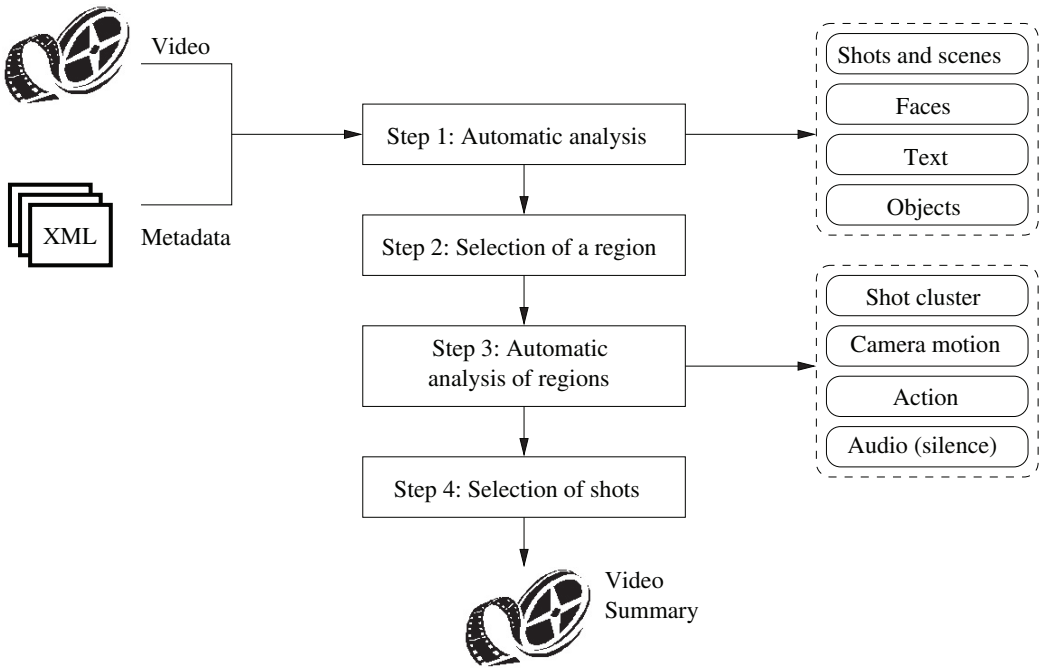

Figure 1. Overview of the video summarization process.

The position of the selected region is not fixed in a shot, and even an artificial camera motion within a larger frame of the original video is possible. For example, if a person is shown in a shot, it might be useful to begin with the full video and then in the summary zoom to the person or focus on the face of the person. Details will still be visible in the zoomed version despite the lower resolution of the video.

A video summary can be generated in four steps (see Figure 1). The first step extracts shot boundaries and higher semantic information from the video. As in most other systems, shots and scenes define the basic structure of a video. A shot is defined as a continuous camera recording, whereas a scene is an aggregation of consecutive shots that have some properties in common, usually the same physical location.

Although it is infeasible to understand the full semantics of an image or shot, it is possible to detect specific features. We have developed special modules to detect frontal faces, text regions and specific moving objects (e. g., cars or walking people).

High-level semantic information is extracted to enable an identification of the most relevant regions first. The second step detects the position and size of the most relevant region within a frame, so that the visible information based on text, faces and objects in each frame is maximized (see Section 6). To avoid jitter or jumps between consecutive frames the detected regions are aggregated at the shot level. The region may follow a continuous motion (pan/tilt) or scaling operation (zoom) within the larger frame of the original video.

In step three, the selected regions are analyzed and additional semantic and syntactic information is calculated for them. We have developed a grouping mechanism that identifies visually similar shots. Shots with a high visual similarity are grouped into the same *cluster.* The size of a cluster – defined as the number of frames of all shots within it – indicates its relevance.

Another criterion that evaluates the relevance of a shot is *action intensity:* The more motion we find in a shot, the more of it we need in the summary. Motion can be either object motion or camera motion. We automatically detect camera motion, moving objects and general action intensity, and then use these features in the synthesis phase to determine the relevance of shots and scenes.

Once shot selection is complete, the low-resolution summary in either MPEG-I or MPEG-II format can be created. The digital video is stored in a database or on the Web in order to be available for other users.

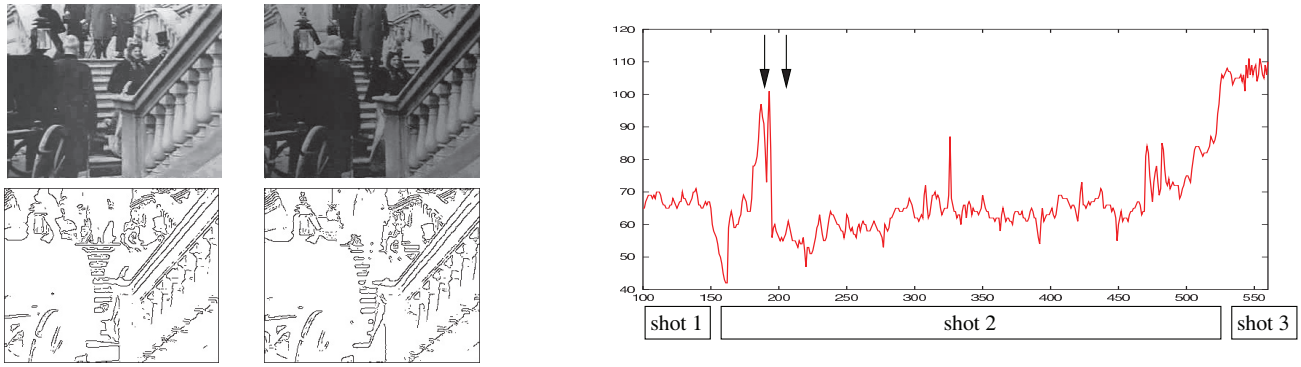

Figure 2. Example of a shot with great changes in the lighting. Top left: Two frames of a shot with average luminance values of 90 and 60. Bottom left: The edge images of these two frames used as input for the ECR. Right: Average luminance of the frames of a shot. The position of the two frames is marked with arrows. Although the histogram difference is very high due to the changes in the luminance (candidate for a hard cut), the ECR-values do not signify a hard cut.

4. FEATURE EXTRACTION

4.1. Shot Boundary Detection

Shots define the basic structure of a film and constitute the basis for the detection of most other semantic features like moving objects or faces. Usually, over 90 % of the transitions in videos are hard cuts. The amount of fades and dissolves is less than 10 %, whereas other transitions, such as wipes occur rarely.

Our shot boundary detection algorithm identifies *hard cuts, fades* and *dissolves.* We combine histograms with edge information and camera motion in order to detect shot boundaries.

We use quantized color histograms to compare consecutive frames. The distance $D_{i,j}$ of frames i and j is defined as the sum of the absolute differences of corresponding histogram bins. In a first step, possible candidates for hard cuts are identified. A hard cut between frames i and $i+1$ is detected if

$$D_{i,i+1} > 2 \cdot \mu \cdot \max\{D_{j,j+1}, j : 1 \leq |i-j| \leq 5\}, \tag{1}$$

where μ is the average histogram difference of all neighboring frames in this video. A hard cut is detected if the histogram difference is significantly larger than the maximum histogram difference in the five-frame neighborhood of the analyzed frame. We use the five-frame neighborhood since short-term changes in frames, such as flashlights or single-frame errors, should not be identified as hard cuts.

In order to improve the cut detection reliability we also compute the edge change ratio (ECR [15]) between adjacent candidate frames. The ECR analyzes the number of edge pixels which appear (incoming edges) or disappear (outgoing edges) in two consecutive frames. The ECR is the normalized sum of outgoing and incoming edge pixels. Many edge pixels change at hard cuts, but luminance changes (e. g., turn on the light) do not affect the ECR significantly.

Our detection of *fade-ins* and *fade-outs* is based on the standard deviation of luminance values for each frame: if the standard deviation decreases from frame to frame and the final frames are close to monochrome frames, we qualify the sequence as a fade-out. We validate a fade-out by also computing the ECR: The number of edges decreases in a fade-out, with many outgoing and no incoming edges.

A *dissolve* has characteristics similar to those of a fade. The standard deviation of the gray-scale values of the pixels in the middle of a dissolve is significantly lower than that at the beginning or end of it. As the significant edges disappear in the first part of the dissolve, the number of outgoing edges increases. In the second half of a dissolve the number of incoming edges is much higher than the number of the outgoing edges.

If a fast horizontal or vertical camera operation occurs (pan or tilt), the images are often blurred. The blurring causes the standard deviation and number of edges to decrease. When the movement stops, the values increase again. To avoid classifying fast camera movements as dissolves, we analyze the camera motion and explicitly eliminate fast camera movements. Figure 2 depicts two frames of a shot with significant changes in the lighting.

4.2. Face Detection

Persons are very important in most types of videos. Close-up views of the faces of main actors are important in feature films, whereas documentaries often feature sports persons, politicians, etc. Face areas are one of the semantic features used for the specification of interesting regions in a frame.

Rowley, Baluja and Kanade[16] have developed a famous, very reliable face recognition algorithm based on a neural network. The algorithm detects about 90 % of the frontal faces in a video. Non-face areas (i.e., false hits) are rare. We have implemented the face detector and trained the network with more than $7,500$ faces. We were able to reproduce the good detection results and have extended the algorithm to detect slightly tilted faces (+/-30 degrees).

A second processing step tracks the faces within a shot. The tracking allows us to find single skipped faces and removes most of the false hits (mis-classified face regions). The tracking analyzes all detected faces in a shot. If one face was detected, the position and size of the face is estimated for the next frame by the global camera motion. The tracking increases the reliability of the face detection algorithm with only a very small increase of computation time.

4.3. Text Recognition

Artificial text in videos has some special properties:

- horizontal alignment,
- significant luminance difference between text and background,
- the character size is within a certain range,
- usually monochrome text,
- text is visible in consecutive frames,
- a horizontal or vertical motion of text is possible.

Our text detection algorithm detects candidate text regions first, and validates these regions in the following steps. The first step analyzes the DCT coefficients of the macro-blocks (strong frequencies in vertical, horizontal and diagonal coefficients). The bounding rectangle of connected blocks, that can be tracked through consecutive frames for at least one second, are marked as text area.

The second step detects the exact boundaries of the text regions. The derivative in x-direction is summed for each horizontal line of the rectangle. Two significant peaks of the summed values indicate the exact position of the text line, e. g., the base and top line in a text with large capitals. The analysis of the text color validates the detected text.

4.4. Recognition of Moving Objects

Moving objects deliver additional semantic information. If the same moving object is visible in many shots, it should also be visible in the summary. The number of moving objects in a video is also an indicator for motion intensity. A film of a car race or a tennis match repeatedly shows moving cars or tennis players. The selection algorithm in Section 6 will assign a high priority to shots containing these identified moving objects.

Our object recognition algorithm consists of two components, a segmentation module and a classification module. Figure 3 depicts the main recognition steps. The motion of the camera is estimated in a first step. The parameters of motion estimation are used to construct a background image for the entire shot. During construction of the background, foreground objects are removed by means of temporal filtering. Object segmentation is then performed by evaluating differences between the current frame and the constructed background. To reduce the effect of incorrectly detected object areas, a tracking algorithm is applied to the object masks. Only objects that can be tracked through several frames of the shot are kept for further processing.

The classification module analyzes the segmented object masks. For each mask, an efficient shape-based representation is calculated (*contour description*).[17] A curvature scale space (CSS) image is used to describe a contour. The CSS technique is based on the idea of curve evolution and provides a multi-scale representation of the curvature zero crossings of a closed planar contour. The CSS-method is one method in MPEG-7 to describe shapes. The matching process compares

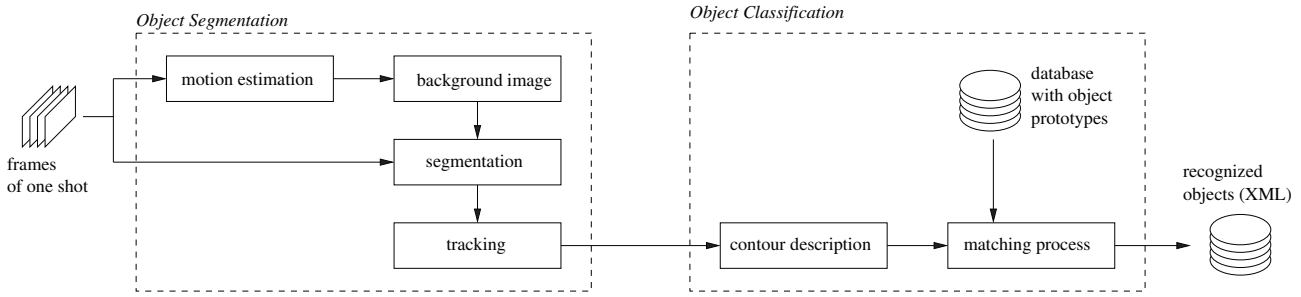

Figure 3. Overview of the object recognition process.

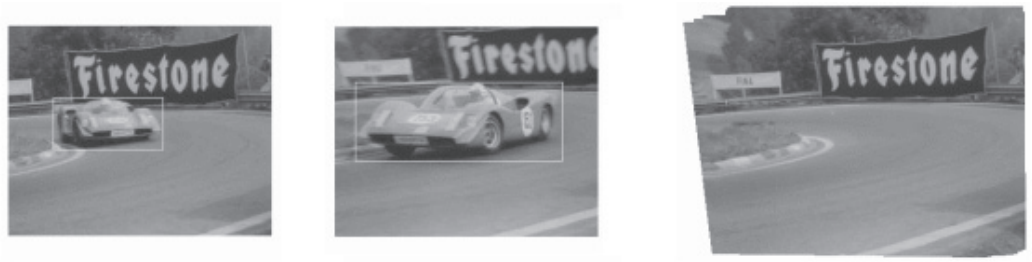

Figure 4. Left: The images show two shots of a scene. The automatically segmented and classified objects are marked in these frames. Right: Automatically constructed background image.

these contour descriptions to pre-calculated object descriptions stored in a database. The matching results for a number of consecutive frames are aggregated. This adds reliability to the approach since unrecognizable single object views occurring in the video are insignificant with respect to the entire sequence.

Figure 4 depicts two sample frames from a shot of a historical car race and the automatically constructed background image. The segmented and classified object (*car*) is marked with a rectangle. A detailed description of the segmentation and classification algorithm can be found in.[17,18]

5. SELECTION OF A REGION

Before a video summary can be generated that is optimized for small displays, two selections have to be made: shots must be selected (temporal selection) and the image size must be decreased (spatial selection). To achieve the spatial reduction it is possible, either to scale the frame or crop parts there or to combine both methods. An advantage of scaling is that all parts of the full frame are still visible. Relevant parts may be lost if a border is cropped. On the other hand it is possible that the content in a scaled frame can no longer be recognized. E. g., if text is scaled down too much, it will no longer be legible. Scaling also reduces the possibility to recognize other content like people or objects and many details may be lost.

We describe in the following section our approach which finds the best compromise between scaling and cropping. Therefore we define a measurement based on the semantic features text, faces, people and objects, that evaluates the information of a region in a frame.

5.1. Information value of regions

Each semantic feature (text, faces, people, and moving objects) represents important information in our terms. The *information value* of a region is defined as the summarized values of its semantic features. The goal is to find the position and size of the region in a frame, such that the information value of this region is maximized. We define the following design goals:

- the information value of a region is maximized,
- the size of the region is larger than the requested screen/window size of the video summary, and

Figure 5. Sample frame with three automatically detected feature regions. Eight possible combinations of these regions are analyzed in order to find the maximum information value.

- the aspect ratio of the region must match the (smaller) viewing window.

We assume that the information of a semantic feature is proportional to the size of its bounding box. The size in the summary depends on the scaling factor and the size of the cropped border (a large border reduces the scaling factor). If the size of a feature drops below a certain threshold, the information it contains is no longer relevant. E, g., it is not possible to read a text if the character size is smaller than a certain value (lower threshold). If this is the case, the information of this text region is set to zero. In addition to a lower threshold, an upper threshold may be required. A very large text does not increase the amount of information, so the size of the characters should be kept within a certain range.

A third condition influences the size of the cropped borders. If parts of features in the cropped frame are no longer visible (e. g., some characters in a line of text), this feature will be ignored. Based on these three conditions, the information value of a text region $V \in [0;1]$ with the text height h is defined as:

$$V = \begin{cases} \frac{h_{max}}{h} & h > h_{max}, \\ \frac{h}{h_{max}} & h \geq h_{min} \ \wedge \ h \leq h_{max}, \\ 0 & h < h_{min}. \end{cases} \tag{2}$$

h_{min} and h_{max} define thresholds for a minimum and a maximum character height. These values depend on user preferences and the hardware used. On a standard PC a text height of 15/40 pixels for h_{min}/h_{max} worked well.

The information value of the other semantic features (faces, people, objects) also depends on their size. The heuristics are similar although an upper threshold that limits the size of an area is not required, and the information value is proportional to its size. Note that it is only possible to downscale the size of a video. As with text areas, the information value is also set to zero for partly visible feature regions. To calculate the information values of faces, people, and objects, in Equation 2 the high threshold (h_{max}) is set to the frame height and h_{min} is set to 25/50/30 for faces/people and objects.

Figure 5 depicts an example containing three automatically detected feature regions. The scaled images on the right side were generated both without cropping (top) and with a large cropped border (bottom). Without cropping the borders, it is very difficult to read the text. The information value is much higher and the visual result much better if a combination of scaling and cropping is used.

The summarized information value V_{sum} aggregates the information values of all features in a region. We have implemented a fast algorithm that calculates the maximum of V_{sum} and detects the size and position of this region. The calculation of the best region is very fast for a limited number of feature regions.

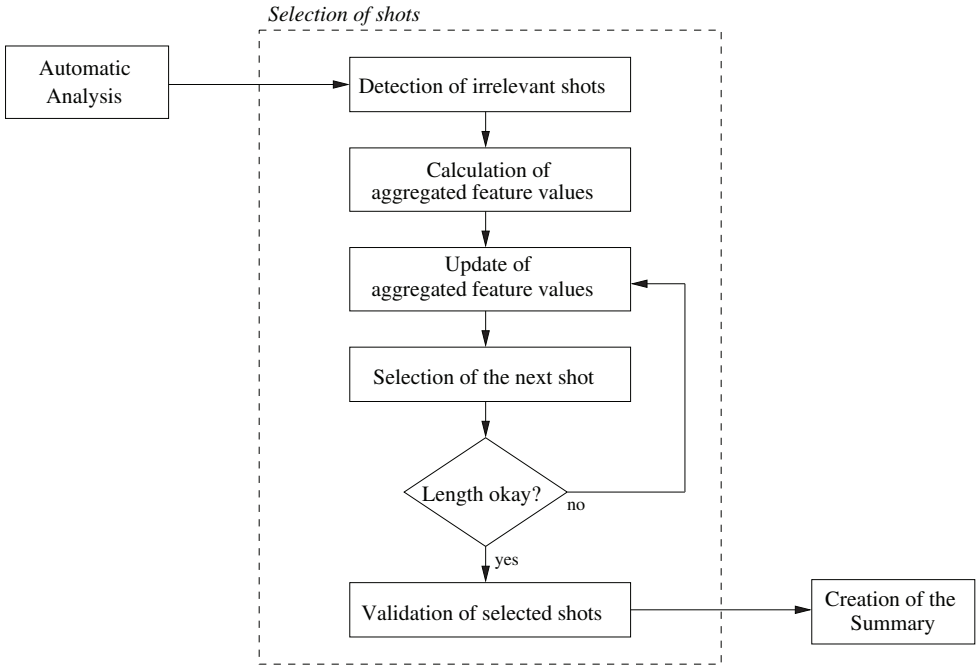

Figure 6. Overview of the algorithm to select shots.

The border of the optimized region must match those of the detected features. If a region selects only a part of a feature, the value of this feature will be ignored and will not increase V_{sum}. On the other hand, we will have a very high scaling factor if a large region is selected. The information values of all features will drop and V_{sum} will not be maximized. It is easy to select the best position and size of a region: All combinations of features are analyzed. The current region is defined as the bounding box of the features actually selected. The full frame will be used if no feature was selected. The information value V_{sum} is calculated for the scaled region and the maximum of V_{sum} and the position and size of the region are stored. If the aspect ratio does not fit or if the selected region is smaller than the expected size of the web video, the size of the selected region will be enlarged.

A large number of features entails significantly increased computational effort. Due to the fact that the probability of recognizing many features in just one frame is very low, the calculation complexity of this algorithm is no problem in the case of real videos.

5.2. Aggregation of regions

Although we have found the region that maximizes the information value of a frame, it is probably not the best selection for the summary. Jitter effects or fast changes of the camera are very unpleasant, while a continuous artificial camera motion is acceptable. The positions of the corners of the selected regions are smoothed with a Gaussian kernel until the camera motion is continuous. E.g., if one large object moves from left to right in the image, the selected region should represent a continuous horizontal camera motion (pan) to follow the feature. We note that a similar recapturing is often done manually when wide-screen cinema films are edited for television.

6. SELECTION OF SHOTS

To select shots additional features in the detected regions are analyzed. If we apply to a shot in this section, we always refer to the selected and scaled region of a shot. Figure 6 depicts the main steps in the selection process. In the first step, irrelevant shots are identified. All shots less than three seconds are removed as are shots with a very low contrast (e. g., monochrome frame).

6.1. Aggregated Feature Values

We calculate aggregated feature values in order to make the different features comparable. Otherwise it would be difficult to make a selection based on face information (position, size, rotation angle) or camera motion information (type of camera operation, motion speed). The aggregated feature value characterizes a feature at the shot level. Each aggregated feature value is normalized to the interval $[0, 1]$.

Most aggregated feature values are initialized only once and no modification is required during the selection process (static features). Other feature values, however, depend on previously selected shots (dynamic features). These are updated whenever a new shot is selected. For each feature and shot an aggregated value is calculated in a first step.

Static features

The aggregated *face* value is the normalized quotient of face pixels to all pixels in the selected region. Two medium-sized faces or one large face would be similarly relevant. The average value of all regions in a shot is stored as the aggregated face value.

Our moving object classification algorithm detects planes, boats, cars and people. The aggregated value for *moving objects* is determined by the number of recognized objects in a shot, the size of the objects and the reliability of the recognition. Moreover, the relevance of a recognized object depends on the objects in the other shots. E. g., if many cars can be recognized, they are very relevant for the video and should also be part of the summary. We gain an additional information from recognized objects. If it is possible to recognize an object in a shot, we known that the quality of that shot is high. A background image cannot be constructed with blurred frames, and noise in the images prevents an exact segmentation.

A zoom-out, pan or tilt introduces a location, where the subsequent act takes place. Typically the countryside, a building or a room is recorded. A zoom-in directs the viewer's attention to the person or object in the center of the zoom. The aggregated value for *camera operations* is a function based on the type of operation (a zoom-in is the most significant), the length of the motion vectors, and the duration of the operation. A static camera at the end of a shot increases the value as well.

The *action* value is the normalized sum of the two values: the average length of the motion vectors as a measure of the motion intensity in a frame and the average pixel difference between two consecutive frames. The aggregated action value is the average of these values of all frames in a shot.

Since it is very hard to recognize the content of shots with a very low contrast, we analyze the contrast to prevent the selection of these shots. The aggregated *contrast* value is the average contrast of all frames in a shot.

Dynamic features

The aggregated values that have been described so far are initialized once and the values do not change (static features). It is necessary to update the values of *shot clusters, scenes* and *position values,* whenever a new shot is selected.

The *relevance of a cluster* C_i, which stores all visually similar shots, depends on the length of all shots in this cluster i:

$$C_i = \frac{L_i}{max\{L_j\}} \cdot \frac{1}{1 + S_i}, \qquad j = 1 \ldots N.$$

L_i is the summarized length of all shots of cluster i. S_i is the number of shots already selected from this cluster and N is the total number of clusters.

To better understand the content of a *scene,* at least two consecutive shots should be selected. Consecutive shots reduce the likelihood of broken sentences. The aggregated scene values are calculated in several steps: first, the values are initialized with an average value. This value will be reduced, if two or more shots in a scene have been selected. If only one shot has been selected, the values of neighboring shots will be increased. The heuristic prefers the selection of two consecutive shots in a scene.

A major goal of a video summary is to give an overview of the full video. It is necessary to select shots from all parts of the video. A summary of a feature film may target a different goal so as not to reveal the thrilling end of a film. The $distance$ value tries to distribute the selected shots over the full length of the video. The value calculates the distance from one shot to the next one selected. This distance is normalized to the interval $[0, 1]$.

6.2. Selection of Shots

The selection process uses the aggregated feature values. The summarized relevance R_i of a shot i is defined as

$$R_i = \alpha_i \cdot F_i, \qquad \sum \alpha_i = 1.$$

We used fixed weights ($\alpha_i = \alpha_j, \ \forall i, j$) in our implementation, however, a user can define customized weights.

The selection algorithm is an iterative process, as depicted in Figure 6. Once the feature values have been calculated, the shot with the maximum summarized relevance R_i can be selected for the summary. The algorithm stops, when the summary has reached the desired length. Otherwise the dynamic feature values will be updated and the next shot will be selected.

6.3. Validation of Selected Shots

The presentation of the selected shots is very important for the acceptance of the video summary. Some constraints must be regarded in order to avoid disturbing effects. The most important constraints are:

- Subsequent camera operations (e.g., a zoom-in followed by a pan followed by a zooms-out) should be avoided. Two shots with significant camera operations should be separated by at least one shot with a static camera.
- At least two shots should be selected from a scene. These shots should be consecutive.
- The audio track should be cut in areas of silence.
- The average level of action in the summary should not be significantly higher than the level of action in the full video. Especially in films with a great deal of action a validation of the action intensity is required. Otherwise it is highly probable that nearly all shots selected for the summary will have a very high action intensity.
- The length of the summary is similar to the length specified by the user.

If constraints are violated, the result may be the removal, addition or replacement of several shots.This depends on the current length of the summary. All constraints are checked iteratively until all violations have been resolved.

The length of the summary can be defined by the user as an absolute or as a relative value. If no length is specified, it will be set to a predefined value that is dependent on the length of the original film.

The audio is very important for the acceptance of video summaries. Speech and music should not be cut at random positions. We set the final cuts at silent areas, even if this involved the addition or removal of several frames.

6.4. Creation of the summary

The final step selects the transitions between the shots and creates the summary. The transitions in both summary and film should be similar. E.g., if the film uses many dissolves, these should be chosen as the transitions for the summary, too.

In addition to the image resolution in the final summary, the user can modify the framerate and bitrate. E.g., if a user wants to create MPEG-I summaries in QCIF resolution from high-resolution MPEG-II videos, he can specify the required parameters and the summary will be generated.

We have two options when selecting the visual region for the summary: We can select either the original frame or the region that maximizes the information value. Both regions must be scaled to fit to the final resolution of the summary. We have included a third option that adds a virtual camera motion and combines both regions. This artificial camera motion highlights the detected features in the region. E. g., if a single person is visible in a shot, it is a good heuristic to show the frame in full size first and then zoom in on the face of the person. In the case of a static camera, it is possible to add any kind of motion. Otherwise it is only possible to increase the speed of the motion. We must bear in mind that two consecutive shots should not have a significant camera motion. In a last step, the small-screen version of the video summary is encoded based on the selected regions and shots.

7. RESULTS

The shot boundary detection algorithm is very reliable due to the combination of different approaches (histograms, edges and camera operation). We have analyzed its reliability in random selected films with a total length of more than one hour. More than 91 % of all cut-boundaries can be detected with 2 % false hits. A simple histogram-based approach detects 70 % at an error rate of 10 %. Our approach is very reliable even in the case of noisy or damaged films.

The estimation of the camera operation is very precise. Otherwise the object recognition – that requires exact background images – would fail. Errors occur if large foreground objects are present or if background images are blurred.

Our face detection system locates more than 90 % of the frontal faces with a minimum height of at least 25 pixels. The detection of moving objects is much more fault-prone. The recognition rate in shots with one car or one person is acceptable (about 40 %). It is much lower for planes or boats due to the changing background (water) or very few edges in the background (e.g., sky with some clouds). The segmentation of many objects failed, but almost no wrong classification occurs. The probability of detecting an object is very low if:

- more than one object moves in the shot,
- the object is very large,
- the background is blurred or noisy,
- the luminance changes, or
- the object is partially occluded.

The detection of the most relevant region works very well. It was much easier to understand the content of a video summary given our new approach and the average information value was significantly increased.

An example of a summary of a historical documentation from 1947 is depicted in Figure 7. Sample frames of three shots have been selected to visualize the selection of regions. The top row in Figure 7 depicts scaled frames of the video, whereas the frames in the bottom row were generated with our new approach.

Text regions were detected in the first shot. The small text in the lower part (timecode) is ignored due to its small size. In the second shot a virtual zoom-in was generated to visualize more details in the last frames of the shot. Our face detection algorithm could not locate the man in the last shot due to its beard and glasses. Although no relevant region was selected, a small part of the border was classified as irrelevant region and cropped (black pixels and noise).

Figure 7. Selected frames of a historical documentation from 1947. Top: Sample frames of an automatically generated video summary. The video has been scaled to 172x144 pixels without analyzing the information value. Bottom: The information value was used to locate the best position to crop the frames. In the second shot an artificial camera motion (zoom-in) was generated automatically.

8. CONCLUSIONS AND OUTLOOK

In the *ECHO* (European Chronicles Online) project a software system was developed that stores and manages large collections of historical films for the preservation of cultural heritage. Four major national film archives (Instituto Luce (Italy), Memoriav (Switzerland), Netherlands Audiovisual Archive (the Netherlands) and Institut National de l'Audiovisuel (France)) stored several hundred thousand hours of historical film material in their archives. Video summaries generated with our tools facilitate the work of the historians and archivists.

Many national archives will make parts of their collections available on the Web. Our new approach offers a possibility to generate small-resolution video summaries without losing too much relevant information.

During the last year we have received feedback from our partners in the ECHO project and conducted some local tests. Two major problems were reported. Shots that did not show any relevant information were selected for the summary. A very low contrast was the one feature common to all these shots. Therefore we added the contrast measure.

The second problem concerns the audio track of the summaries. It is very disturbing if a sentence or music is interrupted. A reliable recognition of words is nearly impossible, and the end of sentences cannot be detected. A possible solution might be to fade-in and fade-out the audio. Additional research is required in this area.

REFERENCES

1. R. Lienhart, S. Pfeiffer, and W. Effelsberg, "Video abstracting," *Communications of the ACM* , pp. 55–62, 1997.
2. H. D. Wactlar, "Informedia – search and summarization in the video medium," in *Proceedings of Imagina*, 2000.
3. M. G. Christel, A. G. Hauptmann, A. S. Warmack, and S. A. Crosby, "Adjustable filmstrips and skims as abstractions for a digital video library," in *Proc. of the IEEE Advances in Digital Libraries Conference*, pp. 98–104, 1999.
4. M. G. Christel, A. G. Hauptmann, H. D. Wactlar, and T. D. Ng, "Collages as dynamic summaries for news video," in *Proceedings of the 2002 ACM workshops on Multimedia*, pp. 561–569, ACM Press, 2002.
5. T. D. Ng, H. D. Wactlar, A. G. Hauptmann, and M. G. Christel, "Collages as dynamic summaries of mined video content for intelligent multimedia knowledge management," in *AAAI Spring Symposium Series on Intelligent Multimedia Knowledge Management*, 2003.
6. N. Omoigui, L. He, A. Gupta, J. Grudin, and E. Sanocki, "Time-compression: systems concerns, usage, and benefits," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 136–143, ACM Press, 1999.
7. A. Amir, D. Ponceleon, B. Blanchard, D. Petkovic, S. Srinivasan, and G. Cohen, "Using audio time scale modification for video browsing," in *IEEE 33rd Hawaii International Conference on System Sciences*, pp. 254–261, IEEE, 2000.
8. R. Lienhart, "Dynamic video summarization of home video," in *Proceedings of the SPIE, Storage and Retrieval for Media Databases 2000*, **3972**, SPIE, 2000.
9. J. Boreczky, A. Girgensohn, G. Golovchinsky, and S. Uchihashi, "An interactive comic book presentation for exploring video," in *CHI 2000 Conference Proceedings*, pp. 185–192, ACM Press, 2000.
10. S. Uchihashi, J. Foote, A. Girgensohn, and J. Boreczky, "Video manga: Generating semantically meaningful video summaries," in *Proceedings ACM Multimedia*, pp. 383–392, ACM Press, 1999.
11. A. Aner and J. R. Kender, "Video summaries through mosaic-based shot and scene clustering," in *Proc. European Conference on Computer Vision*, 2002.
12. M. M. Yeung, B.-L. Yeo, and B. Liu, "Extracting story units from long programs for video browsing and navigation," in *Proc. IEEE International Conference on Multimedia Computing and Systems*, pp. 296–305, 1996.
13. X. Fan, X. Xie, W. Ma, H. Zhang, and H. Zhou, "Visual attention based image browsing on mobile devices," in *Int. Conf. on Multimedia and Expo (ICME 03)*, IEEE, (Baltimore, USA), July 2003.
14. L.-Q. Chen, X. Xie, X. Fan, W.-Y. Ma, H.-J. Zhang, and H.-Q. Zhou, "A visual attention model for adapting images on small displays," *ACM Multimedia Systems Journal* **9**(4), pp. p353–364, 2003.
15. R. Zabih, J. Miller, and K. Mai, "A feature-based algorithm for detecting and classifying scene breaks," in *Proceedings ACM International Conference on Multimedia*, pp. 189–200, ACM Press, 1995.
16. H. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**, pp. 23–38, 1998.
17. S. Richter, G. Kühne, and O. Schuster, "Contour-based classification of video objects," in *Proceedings of SPIE, Storage and Retrieval for Media Databases*, **4315**, pp. 608–618, SPIE, (Bellingham, Washington), January 2001.
18. D. Farin, T. Haenselmann, S. Kopf, G. Kühne, and W. Effelsberg, "Segmentation and classification of moving video objects," in *Handbook of Video Databases*, B. Furht and O. Marques, eds., CRC Press, 2003.