Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2007, Article ID 19496, 14 pages
doi:10.1155/2007/19496

Research Article

Compact Visualisation of Video Summaries

Janko Ćalić and Neill W. Campbell

Department of Computer Science, Faculty of Engineering, University of Bristol, Bristol BS8 1UB, UK

Received 31 August 2006; Revised 22 December 2006; Accepted 2 February 2007

Recommended by Ebroul Izquierdo

This paper presents a system for compact and intuitive video summarisation aimed at both high-end professional production environments and small-screen portable devices. To represent large amounts of information in the form of a video key-frame summary, this paper studies the narrative grammar of comics, and using its universal and intuitive rules, lays out visual summaries in an efficient and user-centered way. In addition, the system exploits visual attention modelling and rapid serial visual presentation to generate highly compact summaries on mobile devices. A robust real-time algorithm for key-frame extraction is presented. The system ranks importance of key-frame sizes in the final layout by balancing the dominant visual representability and discovery of unanticipated content utilising a specific cost function and an unsupervised robust spectral clustering technique. A final layout is created using an optimisation algorithm based on dynamic programming. Algorithm efficiency and robustness are demonstrated by comparing the results with a manually labelled ground truth and with optimal panelling solutions.

1. INTRODUCTION

The conventional paradigm to bridge the *semantic gap* between low-level information extracted from the digital videos and the user's need to meaningfully interact with large multimedia databases in an intuitive way is to learn and model the way different users link perceived stimuli and their meaning [1]. This widespread approach attempts to uncover the underpinning processes of human visual understanding and thus often fails to achieve reliable results, unless it targets a narrow application context or only a certain type of the video content. The work presented in this paper makes a shift towards more user-centered summarisation and browsing of large video collections by augmenting user's interaction with the content rather than learning the way users create related semantics.

In order to create an effortless and intuitive interaction with the overwhelming extent of information embedded in video archives, we propose two systems for generation of compact video summaries in two different scenarios. The first system targets high-end users such as broadcasting production professionals, exploiting the universally familiar narrative structure of comics to generate easily readable visual summaries. In case of browsing video archives in a mobile application scenario, visual summary is generated using a model of human visual attention. The extracted salient information from the attention model is exploited to lay out an optimal presentation of the content on a device with a small size display, whether it is a mobile phone, handheld PC, or PDA.

Being defined as "spatially juxtaposed images in deliberate sequence intended to convey information" [2], comics are the most prevalent medium that expresses meaning through a sequence of spatially structured images. Exploiting this concept, the proposed system follows the narrative structure of comics, linking the temporal flow of video sequence with the spatial position of panels in a comic strip. This approach differentiates our work from the typical reverse storyboarding [3, 4] or video summarisation approaches. There have been attempts to utilise the form of comics as a medium for visual summarisation of videos [5, 6]. Here, the layout algorithm optimises the ratio of white space left and approximation error of the frame importance function. However, the optimisation algorithm utilises a full search method, which becomes impractical for larger layouts.

This work brings a real-time capability to video summarisation by introducing a solution based on dynamic programming and proving that the adopted suboptimal approach achieves practically optimal layout results. Not only does it improve the processing time of the summarisation task, but it enables new functionalities of visualisation for large-scale video archives, such as runtime interaction,

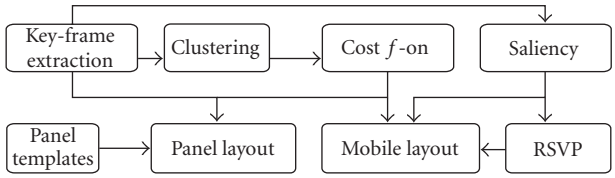


Figure 1: Block scheme of the video summarisation system. For the both layout modules, key-frame sizes are being estimated. The saliency based cropping is applied only in case of mobile scenario.

scalability and relevance feedback. In addition, the presented algorithm applies a new approach to the estimation of key-frame sizes in the final layout by exploiting a spectral clustering methodology coupled with a specific cost function that balances between good content representability and discovery of unanticipated content. Finally, by exploiting visual attention in the small screen scenario, displayed key frames are intelligently cropped, displaying only the most salient image regions. This completely unsupervised algorithm demonstrated high precision and recall values when compared with hand-labelled ground truth.

The initial step of key-frame extraction, presented in Section 3, utilises underlying production rules to extract the best visual representative of a shot in an efficient manner [7]. In order to rank the importance of key frames in the final visual layout, a specific cost function that relies on a novel robust image clustering method is presented in Section 4. Two optimisation techniques that generate a layout of panels in comic-like fashion are described in Section 5. The first technique finds an optimal solution for a given cost function, while the second suboptimal method utilises dynamic programming to efficiently generate the summary [8]. In order to adapt the summaries to small screen devices, visual attention modelling [9] is used to estimate the most salient regions of extracted key-frames, as given in Section 6. The number of salient regions is defined by the desired response time, determined from the required speed of rapid serial visual presentation (RSVP) [10]. Finally, the results of the algorithms presented are evaluated in Section 7 by comparing achieved output with a manually labelled ground truth and benchmarking the optimal against a suboptimal panelling solution. The following section outlines the architecture of the overall system.

2. SYSTEM DESCRIPTION

The proposed system for video summarisation comprises two main modules: (i) *panel layout* and (ii) *mobile layout*, as depicted in Figure 1. The *panel layout* module generates video summaries for computer screens and exploits information from the key-frame extraction module, estimation of the layout cost function and the panel template generator. On the other hand, *mobile layout* module uses key-frame saliency maps and the timing defined by the visual attention model and the RSVP trigger. This module generates a sequence of compact summaries comprising the most salient key-frame regions. In order to generate the visual summary, a set of the most representative frames is generated from the analysed video sequence. It relies on the precalculated information on shot boundary locations that is retrieved from an existing indexed metadata database. The shot-detection module utilises block-based correlation coefficients and histogram differences to measure the visual content similarity between frame pairs [11]. Shot boundary candidates are labelled by thresholding χ^2 global colour histogram frame differences, while in the second pass, a more detailed analysis is applied to all candidates below a certain predetermined threshold. Developed as a part of a joint project [12] that analyses raw footage of wildlife rushes, this algorithm achieves a higher recall and precision compared with the conventional shot-detection techniques. Once the shot boundaries are determined, a single key frame is extracted from each shot to represent its content in the best possible way.

In the second stage, a specific layout cost function is assigned to each key frame to rank the importance of the key frame in the final layout. In order to calculate the cost function, key frames are initially clustered using a robust, unsupervised spectral clustering technique.

For the high-end summaries, comic-like panel templates are laid out in the final visual summary using an efficient optimisation algorithm based on dynamic programming. In this scenario, the aspect ratio of images is fixed to the source aspect ratio and therefore there are no attempts to crop or reshape them for the final layout.

However, in order to produce highly compact summaries for the mobile devices, salient image regions are extracted using a human visual attention model. A single screen summary is generated by laying the extracted salient regions on a screen. Utilising the RSVP approach, layouts are displayed sequentially to the user until the end of presented video sequence is reached.

3. KEY-FRAME EXTRACTION

In order to generate the visual summary, a set of the most representative frames is generated from the analysed video sequence. Initially, video data is subsampled in both space and time to achieve real-time processing capability. Spatial complexity reduction is achieved by representing an 8×8 block with its average pixel value, generating a low-resolution representation of video frames known as the *DC sequence*. By doing this, the decoding process is minimised since the DC sequence can be efficiently extracted from an MPEG compressed video stream [13]. In the temporal dimension, key frame candidates are determined either by uniform sampling every nth frame or after a cumulative pixelwise prediction error between two adjacent candidate frames reaches a predefined threshold. The latter approach distorts the time in a nonlinear fashion and thus loses the notion of real motion required by the camera work classification module. Therefore, a temporal decimation with the constant factor of $n = 5$ is applied.

Having generated the low complexity data representation with dimensions $W \times H$, a dense optical flow $\vec{F}(x, y)$ of the DC sequence is estimated efficiently using the Lucas-Kanade image registration technique [14]. In order to apply model

Table 1: Camera work categories and corresponding error threshold values.

	Zoom		Pan		Tilt	
	In	Out	Left	Right	Up	Down
$\mathrm{Th}_{\mathrm{cw}}$	<-1.2	>1.2	<-0.7	>0.7	<-0.8	>0.8

fitting of optical flow data to a priori generated camera work models (i.e., *zoom*, *tilt*, and *pan*), specific transformations are applied to the optical flow $F^i(x, y)$ for each frame i, as given in (1):

$$\begin{aligned}\Phi_z^i(x,y) &= \operatorname{sgn}\left(x-\frac{W}{2}\right)F_x^i(x,y)+\operatorname{sgn}\left(y-\frac{H}{2}\right)F_y^i(x,y),\\ M_z^i(x,y) &= \Phi_z^i(x,y)\cdot\omega(x,y),\\ M_p^i(x,y) &= F_x^i(x,y)\cdot\omega(x,y),\\ M_t^i(x,y) &= F_y^i(x,y)\cdot\omega(x,y).\end{aligned} \tag{1}$$

Weighting coefficients $\omega(x, y)$ favour influence of the optical flow in image boundary regions in order to detect camera work rather than a moving object, typically positioned in the centre of the frame. As shown in (2), the weighting coefficients are calculated as an inverted ecliptic Gaussian aligned to the frame center, with spatial variances determined empirically as $\sigma_x = 0.4 \cdot W$ and $\sigma_y = 0.4 \cdot H$:

$$\omega(x,y) = 1 - e^{-((x-W/2)^2/\sigma_x+(y-H/2)^2/\sigma_y)}. \tag{2}$$

The measure of optical flow data fitness for a given camera work model is calculated as a normalised sum of $M_{\mathrm{cw}}^i(x, y)$ for each type of camera work (cw): zoom (z), pan (p), and tilt (t), as given in (3). If the absolute value of fitness function becomes larger than the empirically predefined threshold $\mathrm{Th}_{\mathrm{cw}}$, the frame i is labelled with one of the six camera work categories, as given in Table 1:

$$\Psi_{\mathrm{cw}}^i = \frac{1}{wh}\sum_{x=1}^{W}\sum_{y=1}^{H} M_{\mathrm{cw}}^i(x,y), \quad \text{where } \mathrm{cw}\in\{z,p,t\}. \tag{3}$$

Finally, the binary labels of camera work classes are denoised using morphological operators retaining the persistent areas with camera motion while removing short or intermittent global motion artefacts.

Once the shot regions are labelled with appropriate camera work, only the regions with a static camera (i.e., no camera work labelled) are taken into account in selection of the most representative key-frame candidates. This approach was adopted after consulting the views of video production professionals as well as inspection of manually labelled ground truth. The conclusions were that: (i) since the cameraman tends to focus on the main object of interest using a static camera, the high-level information will be conveyed by the key frame in regions with no camera work labels, (ii) chances to have artefacts like motion and out-of-focus blur are minimised in those regions.

Subsequently, frames closest to the centre of mass of the frame candidates' representation in a multidimensional feature space are specifically ranked to generate the list of region representatives. The algorithm for key-frame selection is as follows:

(1) select $N_{\mathrm{ist}} \geq N_{kf}$ candidates from static regions,
(2) calculate feature matrices for all candidates,
(3) loop through all candidates:
  (a) rank them by L_2 distance to all unrepresented frames of the analysed shot in ascending order;
  (b) select the first candidate and label its neighbouring frames as represented;
  (c) select the last candidate and label its neighbouring frames as represented;
(4) export N_{kf} selected key frames as a prioritised list.

The feature vector used to represent key-frame candidates is an $18 \times 3 \times 3$ HSV colour histogram, extracted from the DC sequence representation for reasons of algorithm efficiency. As an output, the algorithm returns a sorted list of N_{kf} frames and the first frame in the list is used as the key frame in the final video summary. In addition to the single key frame representation, this algorithm generates a video skim for each shot in the video sequence. Depending on application type, length of the skim can be either predefined (N_{kf} = const.) or adaptive, driven by the number of static camera regions and maximum distance allowed during the ranking process. By alternately selecting the first and the last frame from the ranked list, a balance between the best representability and discovery of unanticipated content is achieved.

4. ESTIMATION OF FRAME SIZES

As mentioned before, our aim is to generate an intuitive and easily readable video summary by conveying the significance of a shot from analysed video sequences by the size of its key-frame representative. Any cost function that evaluates the significance is highly dependent upon the application. In our case, the objective is to create a summary of archived video footage for production professionals. Therefore, the summary should clearly present visual content that is dominant throughout the analysed section of the video, as well as to highlight some cutaways and unanticipated content, essential for the creative process of production.

More generally speaking, being essentially a problem of high-level understanding of any type of analysed content, the summarisation task requires a balance between the process that duly favours dominant information and the discovery of the content that is poorly, if at all, represented by the summary. Keeping this balance is important especially in case of visual summarisation, where introduction of unanticipated visual stimuli can dramatically change the conveyed meaning of represented content. In a series of experiments conducted to indicate the usefulness and effectiveness of film editing [15], Russian filmmaker Lev Kuleshov (circa 1918) demonstrated that juxtaposing an identical shot with different appendices induces completely different meaning of the shot in audiences. In other words, the conveyed meaning is created by relation and variance between representing

elements of visual content. This idea of emphasizing difference, complexity, and non-self-identity rather than favouring commonality and simplicity and seeking unifying principles is well established in linguistics and philosophy of meaning through theory of *deconstruction*, forged by French philosopher Derrida in the 1960s [16].

In the case of video summarisation, the estimation of frame importance (in our case frame size) in the final video summary layout is dependant upon the underlying structure of available content. Thus, the algorithm needs to uncover the inherent structure of the dataset and by following the discovered relations evaluate the frame importance. By balancing the two opposing representability criteria, the overall experience of visual summary and the meaning conveyed will be significantly improved.

4.1. Frame grouping

In order to generate the cost function $C(i)$, $i = 1, \ldots, N$ where $C(i) \in [0, 1]$ that represents the desired frame size in the final layout, the key frames are initially grouped into perceptually similar clusters. The feature vector used in the process of grouping is the same HSV colour histogram used for key-frame extraction appended with the pixel values of the DC sequence frame representation in order to maintain essential spatial information.

Being capable of analysing inherent characteristics of the data and coping very well with high nonlinearity of clusters, a *spectral clustering* approach was adopted as method for robust frame grouping [17]. The choice of the spectral clustering approach comes as a result of test runs of standard clustering techniques on wildlife rushes data. The centeroid-based methods like K-means failed to achieve acceptable results since the number of existing clusters had to be defined *a-priori* and these algorithms break down in presence of nonlinear cluster shapes [18].

In order to avoid data-dependent parametrization required by bipartitioning approaches like N-cut [19], we have adopted the K-way spectral clustering approach with unsupervised estimation of number of clusters present in the data.

The initial step in the spectral clustering technique is to calculate the *affinity matrix* $W_{N\times N}$, a square matrix that describes a pairwise similarity between data points, as given in (4):

$$W(i, j) = e^{-\|x_i^2 - x_j^2\|/2\cdot\sigma^2}. \tag{4}$$

Instead of manually setting the scaling parameter σ, Zelnic and Perona [20] introduced a locally scaled affinity matrix, where each element of the data set has been assigned a local scale σ_i, calculated as median of $\kappa = 7$ neighbouring distances of element i so that the affinity matrix becomes

$$W_{\text{loc}}(i, j) = e^{-\|x_i^2 - x_j^2\|/2\cdot\sigma_i\cdot\sigma_j}. \tag{5}$$

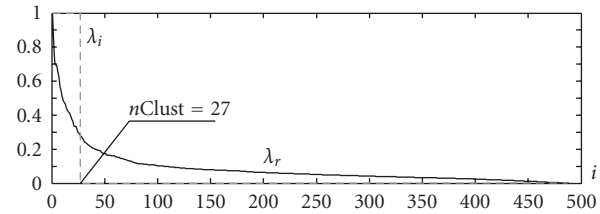


Figure 2: Sorted eigenvalues of affinity matrix with estimated number of data clusters nClust in the ideal case (λ_i) and a real case (λ_r). By clustering eigenvalues in two groups, the number of eigenvalues with value 1 in the ideal case can be estimated.

After calculating the locally scaled affinity matrix W_{loc}, a generalised eigen-system given in 6 is solved:

$$(D - W)y = \lambda D y. \tag{6}$$

Here, D is known as a *degree matrix*, as given in (7):

$$D(i, i) = \sum_j W_{\text{loc}}(i, j). \tag{7}$$

K-way spectral clustering partitions the data into K clusters at once by utilising information from eigenvectors of the affinity matrix. The major drawback of this algorithm is that the number of clusters has to be known *a-priori*. There have been a few algorithms proposed that estimate the number of groups by analysing eigenvalues of the affinity matrix. By analysing the ideal case of cluster separation, Ng et al. [21] show that the eigenvalue of the Laplacian matrix $L = D - W$ with the highest intensity (in the ideal case it is 1) is repeated exactly k times, where k is a number of well-separated clusters in the data. However, in the presence of noise, when clusters are not clearly separated, the eigenvalues deviate from the extreme values of 1 and 0. Thus, counting the eigenvalues that are close to 1 becomes unreliable. Based on a similar idea, Polito and Perona in [22] detect a location of a drop in the magnitude of the eigenvalues in order to estimate k, but the algorithm still lacks the robustness that is required in our case.

Here, a novel algorithm to robustly estimate the number of clusters in the data is proposed. It follows the idea that if the clusters are well separated, there will be two groups of eigenvalues: one converging towards 1 (high values) and another towards 0 (low values). In the real case, convergence to those extreme values will deteriorate, but there will be two opposite tendencies and thus two groups in the eigenvalue set. In order to reliably separate these two groups, we have applied K-means clustering on sorted eigenvalues, where $K = 2$ and initial locations of cluster centers are set to 1 for high-value cluster and 0 to low-value cluster. After clustering, the size of a high-value cluster gives a reliable estimate of the number of clusters k in analysed dataset, as depicted in Figure 2. This approach is similar to the automatic thresholding procedure introduced by Ridler and Calvard [23] designed to optimize the conversion of a bimodal multiple gray level picture to a binary picture. Since the bimodal tendency of the eigenvalues has been proven by Ng et al. in [21], this algorithm robustly estimates the split of the eigenvalues in an

optimal fashion, regardless of the continuous nature of values in a real noisy affinity matrix (see Figure 2).

Following the approach presented by Ng et al. in [21], a Laplacian matrix $L = D - W$ (see (6)) is initially generated from the locally scaled affinity matrix W_{loc} with its diagonal set to zero $\widehat{W}_{\text{loc}}(i,i) = 0$. The formula for calculating the Lapacian matrix normalised by row and column degree is given in (8):

$$L(i,j) = \frac{\widehat{W}_{\text{loc}}(i,j)}{(D(i,i)\cdot D(j,j))}. \tag{8}$$

After solving the eigen system for all eigenvectors eV of L, the number of clusters k is estimated following the aforementioned algorithm. The first k eigenvectors $eV(i)$, $i = 1,\ldots,k$ form a matrix $X_{N\times k}(i,j)$. This matrix is renormalised for each row to have unit length, as given in (9):

$$\hat{X}(i,j) = \frac{X(i,j)}{\sqrt{\sum_{j=1}^{k} X(i,j)^2}}. \tag{9}$$

Finally, by treating each column of $\hat{X}$ as a point in $\mathbb{R}^k$, N vectors are clustered into k groups using the K-means algorithm. The original point i is assigned to cluster j if the vector $\hat{X}(i)$ was assigned to the cluster j.

This clustering algorithm is used as the first step in revealing the underlying structure of the key-frame dataset. The following section describes in detail the algorithm for calculation of the cost function.

4.2. Cost function

To represent the dominant content in the selected section of video, each cluster is represented with a frame closest to the centre of the cluster. Therefore the highest cost function $C(i,d,\sigma_i) = 1$ is assigned for $d = 0$, where d is the distance of the key frame closest to the centre of cluster and σ_i is i^{th} frame's cluster variance. Other members of the cluster are given values (see Figure 3):

$$C(i,d,\sigma_i) = \alpha\cdot\left(1 - e^{-d^2/2\sigma_i^2}\right)\cdot h_{\max}. \tag{10}$$

The cost function is scaled to have a maximum value $h_{\max}$ in order to be normalised to available frame sizes. Parameter α can take values $\alpha \in [0,1]$, and in our case is chosen empirically to be 0.7. In Figure 3, a range of different cost dependency curves are depicted for values $\alpha \in \{0.5,\ldots,1.0\}$ and $h_{\max} = 1$. The value of α controls the balance between the importance of the cluster centre and the outliers.

By doing this, cluster outliers (i.e., cutaways, establishing shots, etc.) are presented as more important and attract more attention of the user than key frames concentrated around the cluster centre. This grouping around the cluster centres is due to common repetitions of similar content in raw video rushes, often adjacent in time. To avoid the repetition of content in the final summary, a set of similar frames is represented by a larger representative, while the others are assigned a lower cost function value.

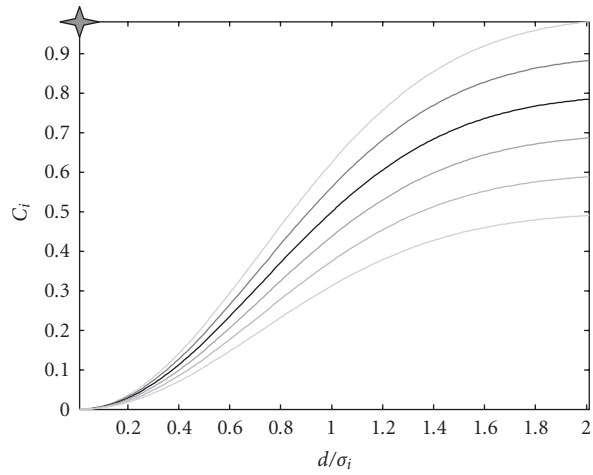


Figure 3: Cost function dependency on distance from the cluster centre for values of parameter $\alpha \in [0.5, 1.0]$.

5. PANELLING

Given the requirement that aspect ratio of key frames in the final layout has to be the same as aspect ratio of the source video frames, the number of possible spatial combinations of frame layouts will be restricted and the frame size ratios have to be rational numbers (e.g., $1:2, 1:3, 2:3$). In addition, following the model of a typical comic strip narrative form, a constraint of spatial layout dependance on time flow is introduced. In our case, the time flow of video sequence is reflected by ordering the frames in left-to-right and top-to-bottom fashion. Excluding this rule would impede the browsing process.

Two page layout algorithms are presented. The first algorithm searches for all possible combinations of page layout and finds an optimal solution for a given cost function. However, processing time requirements make this algorithm unfeasible if the number of frames to be laid out on a single page exceeds a certain threshold. Therefore, a novel suboptimal algorithm is introduced. It utilises dynamic programming (DP) to find the best solution in very short time. Results presented in Section 7 show that the error introduced by the suboptimal model can be disregarded. Firstly, an algorithm that generates panel templates following the narrative structure of comics is presented, followed by detailed descriptions of layout algorithms.

5.1. Panel generator

Following the definition of the art of comics as a sequential art [24] where space does the same as time does for film [2], this work intuitively transforms the temporal dimension of videos into spatial dimension of the final summary by following the well-known rules of comics' narrative structure.

The *panel* is a basic spatial unit of comics as a medium and it distinguishes an ordered pictorial sequence conveying information from a random set of images laid out on a page, that is, it enables closure. Closure is a phenomenon of observing the parts and perceiving the whole. Therefore,

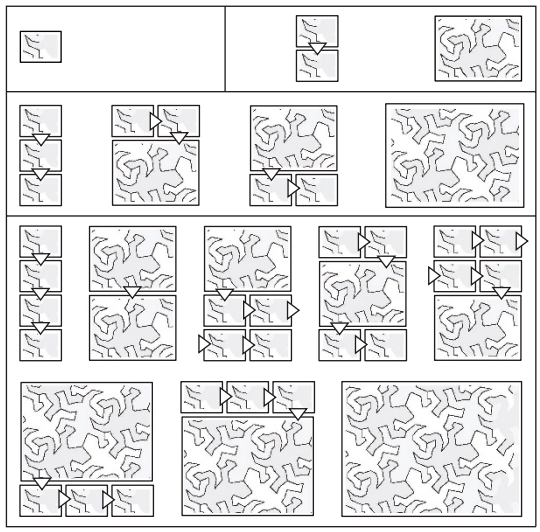


Figure 4: Panel templates for panel heights 1 to 4. Arrows show the temporal sequence of images for each template, adopted from the narrative structure in comics.

in order to achieve an intuitive perception of the comic-like video summary as a whole, panels in the summary layout need to follow basic rules of comics' narrative structure (e.g., time flows from left to right, and from top to bottom).

Therefore, a specific algorithm that generates a set of available panel templates is developed. It creates templates as vectors of integers x_i of normalised image sizes ordered in time. Panel templates are grouped by panel heights, since all panels in a row need to have the same height. The algorithm generates all possible panel vectors x_i, for all $h \in \{1,\ldots,h_{\max}\} \wedge w \in \{1,\ldots,h\}$ and checks if they fit the following requirements:

(1) $h \cdot w = \sum_{\forall i} x_i^2$,
(2) the panel cannot be divided vertically in two.

The final output is a set of available panel templates for given panel heights, stored as an XML file. Examples of panel templates, for panel heights 1–4, are depicted in Figure 4. Paneling module loads required panel templates as well as the cost function and key frames from the database and produces a final page layout, as presented in Section 5.3.

5.2. Optimal solution using full search

In addition to the requirements for a page layout, the optimal layout solution needs to fit exactly into a predefined page width with a fixed number of images per page. This requirement enables objective comparison of layout algorithms, since the DP solution generates layout with adaptive page width and number of frames per page.

As a result of these requirements, for a given maximal row height $h_{\max}$, a set of available panel templates is generated as described before. For a given page height h, page width w, and number of images per page N, distribution of frame sizes depends on the cost function $C(i)$, $i = 1 \ldots N$. An algorithm for calculation of the cost function is described in Section 4.

The main task is to find a frame layout that optimally follows the values of the cost function only using available panel templates. Each panel template generates a vector of frame sizes, that approximates the cost function values of corresponding frames. Precision of this approximation depends upon the maximum size of a frame, defined by the maximum height of the panel $h_{\max}$ which gives granularity of the solution. For a given $h_{\max}$, a set of panel templates is generated (see Figure 4), assigning a vector of frame sizes to each template.

The page-panelling algorithm is divided into two stages: (i) distribution of row heights and (ii) distribution of panels for each row. Since the second stage always finds an optimal solution, the final page layout is determined by finding a minimum approximation error for a given set of row height distributions.

In both parts of the algorithm, the search space is generated by the partitioning of an integer (h or w) into its summands. Since the order of the summands is relevant, it is the case of *composition* of an integer n into all possible k parts, in the form [25]:

$$n = r_1 + r_2 + \cdots + r_k, \qquad r_i \geq 0, \quad i = 1,\ldots,k. \tag{11}$$

Due to a large number of possible compositions (see (12)), an efficient iterative algorithm described in [26] is used to generate all possible solutions:

$$N_{\text{compositions}} = \binom{n+k-1}{n}. \tag{12}$$

In order to find an optimal composition of page height h into k rows with heights $h(i)$, $i = 1,\ldots,k$, for every possible $k \in [h/h_{\max}, h]$, a number of frames per row $\eta(i)$, $i = 1,\ldots,k$ is calculated to satisfy the condition of even spread of the cost function throughout the rows:

$$\forall i, \quad \sum_{j=1}^{\eta(i)} C(j) = \frac{1}{k} \sum_{l=1}^{N} C(l). \tag{13}$$

For each distribution of rows $\eta(i)$, $i = 1,\ldots,k$ and a given page width w, each row is laid out to minimise the difference between the achieved vector of frame sizes and the corresponding part of the cost function $C(i)$. For each composition of $\eta(i)$, a set of possible combinations of panel templates is generated. The vector of template widths used to compose a row has to fit the given composition, as well as the total number of used frames has to be $\eta(i)$. For all layouts that fulfill these conditions, the one that generates a vector of frame sizes with minimal approximation error to the corresponding part of the cost function is used to generate the row layout. The final result is the complete page layout $\Theta(i)$ with the minimal overall approximation error Δ, where Δ is calculated as given in (14):

$$\Delta = \sum_{\forall i} C(i) - \Theta(i). \tag{14}$$

5.3. Suboptimal solution using dynamic programming

There have been numerous attempts to solve the problem of discrete optimisation for spatio-temporal resources. In our case, we need to optimally utilise the available two-dimensional space given required sizes of images. However, unlike many well-studied problems like stock cutting or bin packing [27, 28], there is a nonlinear transformation layer of panel templates between the error function and available resources. In addition, the majority of proposed algorithms are based on heuristics and do not offer an optimal solution.

Therefore, we propose a suboptimal solution using dynamic programming and we will show that the deviation of achieved results from the optimal solution can be practically disregarded. Dynamic programming finds an optimal solution to an optimisation problem $\min \varepsilon(x_1, x_2, \ldots, x_n)$ when not all variables in the evaluation function are interrelated simultaneously:

$$\varepsilon = \varepsilon_1(x_1, x_2) + \varepsilon_2(x_2, x_3) + \cdots + \varepsilon_{n-1}(x_{n-1}, x_n). \quad (15)$$

In this case, solution to the problem can be found as an iterative optimisation defined in (16) and (17), with initialisation $f_0(x_1) = 0$:

$$\min \varepsilon(x_1, x_2, \ldots, x_n) = \min f_{n-1}(x_n), \quad (16)$$

$$f_{j-1}(x_j) = \min \left[f_{j-2}(x_{j-1}) + \varepsilon_{j-1}(x_{j-1}, x_j) \right]. \quad (17)$$

The adopted model claims that optimisation of the overall page layout error, given in (14), is equivalent to optimisation of the sum of independent error functions of two adjacent panels x_{j-1} and x_j, where

$$\varepsilon_{j-1}(x_{j-1}, x_j) = \sum_{i \in \{x_{j-1} \cup x_j\}} .(C(i) - \Theta(i))^2. \quad (18)$$

Although the dependency between nonadjacent panels is precisely and uniquely defined through the hierarchy of the DP solution tree, strictly speaking the claim about the independency of sums from (15) is incorrect. The reason for that is a limiting factor that each row layout has to fit to required page width w, and therefore, width of the last panel in a row is directly dependent upon the sum of widths of previously used panels. If the task would have been to lay out a single row until we run out of frames, regardless of its final width, the proposed solution would be optimal. Nevertheless, by introducing specific corrections to the error function $\varepsilon_{j-1}(x_{j-1}, x_j)$ the suboptimal solution often achieves optimal results.

The proposed suboptimal panelling algorithm comprises the following procedural steps:

(1) load all available panel templates x_i,
(2) for each pair of adjacent panels:
  (a) if panel heights are not equal, penalise;
  (b) determine corresponding cost function values $C(i)$;
  (c) form the error function table $\varepsilon_{j-1}(x_{j-1}, x_j)$ as given in (18);
  (d) find optimal $f_{j-1}(x_j)$ and save it;
(3) if all branches reached row width w, roll back through optimal $f_{j-1}(x_j)$ and save the row solution,
(4) if page height reached, display the page. Else, go to the beginning.

Formulation of the error function table $\varepsilon_{j-1}(x_{j-1}, x_j)$ in a specific case when panel reaches the page width w, the following corrections are introduced:

(1) if current width $w_{\text{curr}} > w$, penalise all but empty panels,
(2) if current width $w_{\text{curr}} = w$, return standard error function, but set it to 0 if the panel is empty,
(3) if current width $w_{\text{curr}} < w$, empty frames are penalised and error function is recalculated for the row resized to fit required width w, as given in (19):

$$\varepsilon_{j-1}(x_{j-1}, x_j) = \sum_i \left(C(i) - \frac{w_{\text{curr}}}{w} \cdot \Theta(i) \right)^2. \quad (19)$$

In this context, penalising means assigning the biggest possible error value to $\varepsilon_{j-1}(x_{j-1}, x_j)$ and w is the required page width. Typically, normalised dimensions of the page, its width w and height h, are determined from the cost function and two values set by the user: expected number of frames per page $\mathcal{N}$ and page aspect ratio $\mathcal{R}$, as given in (20):

$$w = \sqrt{\frac{1}{\mathcal{R}} \sum_{i=1}^{\mathcal{N}} C(i)^2}, \qquad h = \mathcal{R} \cdot w. \quad (20)$$

This procedure generates a set of sequential displays without any screen size limitation. In other words, this algorithm targets application where the video summary is being displayed on a computer screen or is being printed as a page in video archive catalogue. In case of the small screen devices, such as mobile phones or PDAs, this approach is not feasible. The following section introduces an adaptation of the video summarisation algorithm to small screen displays.

6. ADAPTING THE LAYOUT TO MOBILE DEVICES

For the video summarisation perspective, the main limitation of mobile devices is in its small screen resolution, which is often smaller than the original size of a single key frame to be displayed. Therefore, a highly compact presentation is required in order to enable browsing of the video archives on a mobile device. This is achieved by displaying the most salient regions of a key frame determined by the visual attention modelling. In addition, knowing that on a screen a mobile device can display only a few images, we need to introduce a scheme to sequentially present the whole content to the user.

6.1. Rapid serial visual presentation

In order to visually present a summary of the whole video sequence to the user, this work follows the idea of rapid serial

visual presentation (RSVP), a technique that displays visual information using a limited space in which each piece of information is displayed briefly in sequential order [29]. The RSVP method proved to be especially interesting for video summarisation [30]. We adopt the RSVP method that generates a spatial layout of presented content together with the temporal sequencing. The proposed technique combines the timing of the RSVP with the reaction time of the visual attention model to generate easily readable spatial layout of presented content in a novel and efficient way.

In a summary of work on RSVP interfaces [29] the two main RSVP methods are defined: (i) one as a temporal sequencing of single images where each successive image displaces the previous one, a paradigmatic case of video fast-forwarding or channel flipping called *keyhole* mode, and (ii) the more interesting techniques that combine some sort of spatial layout of images with the temporal sequencing. There are four elaborated variants: carousel mode, collage mode, floating mode, and shelf mode. These all incorporate some form of spatio-temporal layout of the image frames that add additional movement or displacement of the image content as the presentation proceeds. In three of these four modes (carousel, floating, and shelf), the images that are upcoming in the sequence are revealed in the background before moving to a more foreground position (or vice versa). In the collage mode, the images appear and disappear as the focus position cycles around the space [10].

Here, we have adopted the sequential display of spatially combined images, where the temporal sequencing is being driven by the time needed to attend the most salient displayed regions, while the spatial layout is determined by optimal utilisation of the display area.

6.2. Visual attention model

Having extracted key frames from video data, salient image regions are determined in order to optimise available display space and show the most important image parts. In order to achieve this, a model of bottom-up salient region selection is employed [31]. This salient region selection algorithm estimates the approximate extent of attended visual objects and simulates the deployment of spatial attention in a biologically realistic model of object recognition in the cortex [32]. In our case, this model determines the visual attention path for a given key frame and automatically selects regions that can be visually attended in a limited time interval.

Initially, a set of *early* visual features, comprising normalised maps of multiscale center-surround differences in colour, intensity, and orientation space, is extracted for each key frame, as presented in [19]. A winner-take-all (WTA) neural network scans the saliency map for the most salient location and returns the location's coordinates. Finally, inhibition of return is applied to a disc-shaped region of fixed radius around the attended location in the saliency map. Further iterations of the WTA network generate a cascade of successively attended locations in order of decreasing saliency.

Knowing the cascade of attended regions and reaction time needed to attend them, a predefined parameter $\mathcal{T}_{\max}$ selects a set of N most important salient regions R_i, $i = 1, \ldots, N$ if $\mathcal{T}_N < \mathcal{T}_{\max}$. In other words, we select the salient regions that can be attended in a fixed time interval $\mathcal{T}_{\max}$. Afterwards, a Gaussian distribution is fitted to a union set of the saliency regions $\mathbb{R} = \bigcup_{i=1}^{N} R_i$, as given in (21):

$$\Gamma_j(x, y) = e^{-((x-\mu_{xj}/\sigma_{xj})^2+(y-\mu_{yj}/\sigma_{yj})^2)}. \tag{21}$$

The Gaussian parameters $\langle \mu_{xj}, \sigma_{xj}, \mu_{yj}, \sigma_{yj} \rangle$ are determined for each key-frame j defining the location and size of their most important parts. This information is later utilised in the layout algorithm. The RSVP timing is calculated as a sum of time intervals $\mathcal{T}_{\max}$ for all key frames in the layout.

6.3. Layout algorithm

After determining the Gaussian parameters $\langle \mu_{xj}, \sigma_{xj}, \mu_{yj}, \sigma_{yj} \rangle$ of the most relevant image region for each key-frame j, the objective is to lay out selected salient image parts in an optimal way for a given display size.

There have been numerous attempts to solve the problem of discrete optimisation for spatio-temporal resources [27]. In our case, we need to utilise the available two-dimensional space given the sizes of salient image regions. However, unlike many well-studied problems like stock cutting or bin packing [28], there is a requirement to fit the salient image regions into a predefined area in a given order. In addition, the majority of proposed algorithms are based on heuristics and do not offer an optimal solution.

Therefore, we propose an optimal solution using dynamic programming that is a modification of the algorithm given in Section 5. Just as before, we claim that optimisation of the overall layout error is equivalent to optimisation of the sum of independent error functions of two adjacent images x_{j-1} and x_j. In our case, the error function is defined as a sum of parts of Gaussians that fell outside of display boundaries (h, w) in a given layout. Knowing the overall sum of Gaussians, given in (22), and the sum of the parts within the display boundaries, given in (23), the error function for two adjacent images is defined in (24):

$$\gamma_j = \sum \sum_{\forall x,y} \Gamma_j(x, y) = \pi \sigma_x^j \sigma_y^j, \tag{22}$$

$$\delta_j = \sum_{x=1}^{w} \sum_{y=1}^{h} \Gamma_j(x, y), \tag{23}$$

$$\varepsilon_{j-1}(x_{j-1}, x_j) = \gamma_j + \gamma_{j-1} - \delta_j - \delta_{j-1}. \tag{24}$$

The search domain for each pair of Gaussians $\{\Gamma_j, \Gamma_{j+1}\}$ comprises uniformly quantised locations of the secondary Gaussian Γ_{j+1} rotated around the primary Gaussian Γ_j. The distance between the centres of Γ_j and Γ_{j+1} is quantised so that the ellipses $\mathrm{E}_j := \{\Gamma_j = \text{const.}\}$ have their semiaxes as follows:

$$\begin{aligned} a_j &= \sqrt{2} \cdot \mathcal{K} \cdot \sigma_x, \\ b_j &= \sqrt{2} \cdot \mathcal{K} \cdot \sigma_y, \\ \mathcal{K} &\in \{\mathcal{K}_{\text{opt}} - 1, \mathcal{K}_{\text{opt}}, \mathcal{K}_{\text{opt}} + 1\}. \end{aligned} \tag{25}$$

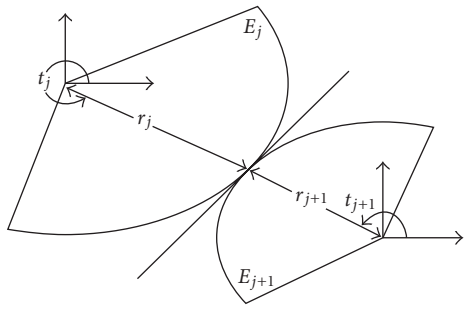


Figure 5: Definition of the search domain parameters. The relative position of centre of the secondary ellipse is determined from the condition that the two tangents coincide.

The optimal value $\mathcal{K}_{\text{opt}}$ is determined from hand-labelled ground truth, as explained in detail in Section 7.

Locus of the centre of the ellipse $\mathrm{E}_{j+1}(x, y)$ relative to the centre of $\mathrm{E}_j(x, y)$, as depicted in Figure 5, is derived from the condition that the two ellipses touch, that is their tangents coincide:

$$\begin{aligned} x_r(t_j, \mathcal{K}) &= a_j \cdot \cos(t_j) + a_{j+1} \cdot \cos(t_{j+1}), \\ y_r(t_j, \mathcal{K}) &= b_j \cdot \sin(t_j) + b_{j+1} \cdot \sin(t_{j+1}), \\ t_{j+1} &= \arctan\left(\frac{a_j \cdot b_{j+1}}{a_{j+1} \cdot b_j} \tan(t_j)\right). \end{aligned} \tag{26}$$

The rotation angle $t \in [-3\pi/4, \pi/4]$ is uniformly quantised into 9 values, eliminating the possibility of positioning new salient region above or to the left of the previous one.

The dependency between nonadjacent images is precisely and uniquely defined through the hierarchy of the DP solution tree and there is no limitation of the boundary effect described in detail in [33]. Therefore, the solution to the discrete optimisation of layout driven by parameters $\langle \mu_{xj}, \sigma_{xj}, \mu_{yj}, \sigma_{yj} \rangle$ and the display size (h, w) is practically optimal.

The proposed layout algorithm comprises the following procedural steps:

(1) determine Gaussian parameters $\langle \mu_{xj}, \sigma_{xj}, \mu_{yj}, \sigma_{yj} \rangle$ for all images,
(2) for each pair of adjacent images:
  (a) determine corresponding cost function values $C(i)$;
  (b) form the error function table $\varepsilon_{j-1}(x_{j-1}, x_j)$ as given in (18);
  (c) find optimal $f_{j-1}(x_j)$ and save it;
(3) if all DP tree branches exploited all available images, roll back through the path with minimal overall cost function f.

This procedure finds the optimal fit for saliency regions described by a Gaussian with parameters $\langle \mu_{xj}, \sigma_{xj}, \mu_{yj}, \sigma_{yj} \rangle$. The final step is to determine the rectangular boundaries for image cropping given the optimal fit. This is done by finding the intersection of each pair of Gaussian surfaces Γ_1, Γ_2, and a plane Ψ (see Figure 6) through their centre points normal to xy plane, defined by (27):

$$\Psi: \quad y = \mu_{y1} + (x - \mu_{x1}) \frac{\mu_{y2} - \mu_{y1}}{\mu_{x2} - \mu_{x1}}. \tag{27}$$

The intersection $\Gamma_1 \cap \Gamma_2 \cap \Psi$ is the minimum value on the shortest path between two centres on a surface $\Gamma_1 \cup \Gamma_2$. The optimal cropping is calculated for all N images on the page, generating $N(N-1)/2$ possible cropping rectangles. The cropping that maximises the value of overall sum within display boundaries Ω, given in (28), is applied:

$$\Omega = \sum_{j=1}^{N} \sum_{x=1}^{w} \sum_{y=1}^{h} \Gamma_j(x, y). \tag{28}$$

Finally, the source images are cropped, laid out, and displayed on the screen. A number of generated layouts is presented in the following section.

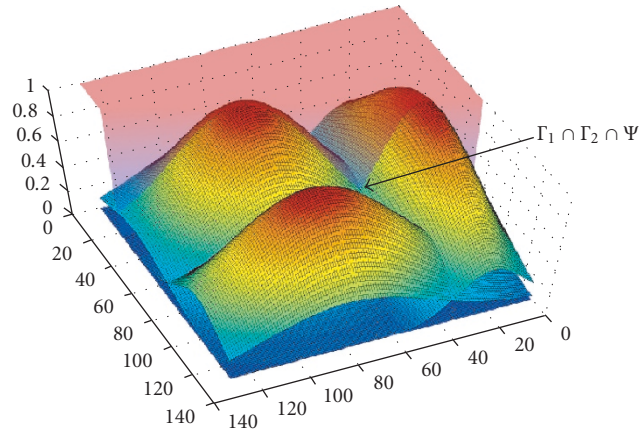


Figure 6: Locating the cropping points at intersection of two Gaussian surfaces Γ_1 and Γ_2 and the Ψ plane defined by two center points (μ_1^x, μ_1^y) and (μ_2^x, μ_2^y).

Table 2: Key-frame extraction evaluation results compared to hand labelling ground truth.

	$\mathbb{X}$	$\mathbb{G}$	$\mathbb{N}$	$\mathbb{B}$	$\mathbb{S}$	$\mathrm{Pr}_1[\%]$	$\mathrm{Pr}_2[\%]$
T01000.mov	52	121	23	14	210	93.5	93.3
T01002.mov	10	129	32	42	213	73.6	80.3

7. RESULTS

The experiments were conducted on a large video archive of wildlife rushes, a collection available as a part of the ICBR project [34]. Approximately 12000 hours of digitised footage have been indexed with shot boundary metadata used by the key-frame extraction module. First of all, we present evaluation of the key-frame extraction algorithm, followed by experimental results of both layout algorithms.

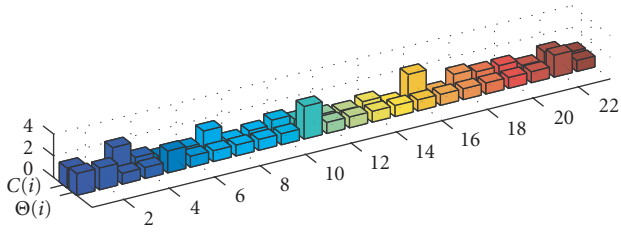


Figure 7: An example of row layout $\Theta(i)$ generated by the DP algorithm, compared to the cost function $C(i)$.

Table 3: Approximation error Δ as function of maximum row height $h_{\max}$ and number of frames on a page $\mathcal{N}$, expressed in [%].

$h_{\max} \setminus \mathcal{N}$	40	80	120	160	200	240
1	6.40	3.92	3.42	2.81	2.58	2.34
2	2.16	1.83	1.65	1.61	1.39	1.46
3	2.24	2.02	1.81	1.53	1.32	1.43
4	2.67	2.17	1.68	1.65	1.31	1.28

7.1. Evaluation of key-frame extraction

The evaluation of the key-frame extraction algorithm is undertaken by comparing achieved results to the hand-labelled ground truth. Two video clips with approximately 90 minutes of wildlife rushes from the ICBR database were labelled by a production professional, annotating the good ($\mathbb{G}$), bad ($\mathbb{B}$), and excellent ($\mathbb{X}$) regions for a potential location of the key frame. In order to numerically evaluate the quality of the first version, two precision measures were defined as follows:

$$\begin{aligned} \Pr_{1,2} &= \frac{D_{1,2}}{(D_{1,2} + \mathbb{B})}, \\ D_1 &= 2 * \mathbb{X} + \mathbb{G} - \mathbb{N}, \\ D_2 &= \mathbb{X} + \mathbb{G} + \mathbb{N}. \end{aligned} \tag{29}$$

The value D_1 incorporates the higher importance of excellent detections and penalise detections that fell into the unlabelled regions ($\mathbb{N}$), while D_2 takes into account only the fraction of key-frame locations that did not fall within regions labelled as bad. The precision results for the two hand-labelled tapes with $\mathbb{S}$ shots are given in Table 2.

7.2. Panelling results

In order to evaluate the results of the DP suboptimal panelling algorithm, results are compared against the optimal solution, described in Section 5.2. An example of a single-row layout approximation is depicted in Figure 7, comparing the desired cost function $C(i)$ with achieved values of frame sizes $\Theta(i)$.

Results in Table 3 show dependency of the approximation error defined in (30) for two main algorithm parameters: maximum row height $h_{\max}$ and number of frames on a page $\mathcal{N}$:

$$\Delta = \frac{1}{\mathcal{N} \cdot h_{\max}} \sqrt{\sum_{i=1}^{\mathcal{N}} \left(C(i) - \Theta(i)\right)^2}. \tag{30}$$

As expected, error generally drops as both $h_{\max}$ and $\mathcal{N}$ rise. By having more choices of size combinations for panel templates with larger $h_{\max}$, the cost function can be approximated more accurately. In addition, the effect of higher approximation error due to the fixed page width, that results in suboptimal solution of the DP algorithm, has less impact as number of frames per page $\mathcal{N}$, and thus page width w, rises. On the other hand, the approximation error rises with $h_{\max}$ for lower values of $\mathcal{N}$, due to a strong boundary effect explained in Section 5.3.

The first three columns of Table 4 show approximation error of the optimal method, while the other three columns show absolute difference between errors of the optimal and suboptimal solutions. Due to a high complexity of the optimal algorithm, only page layouts with up to 120 frames per page have been calculated. As stated in Section 5.3, the overall error due to the suboptimal model is on average smaller than

Table 4: Approximation error Δ using optimal algorithm for given $h_{\max}$ and $\mathcal{N}$, expressed in [%].

	Δ_{optimal}			$\lvert \Delta_D P - \Delta_{\text{optimal}} \rvert$		
$h_{\max} \setminus \mathcal{N}$	40	80	120	40	80	120
1	6.40	3.92	3.42	0.00	0.00	0.00
2	1.87	1.57	1.45	0.29	0.26	0.20
3	2.05	1.34	1.81	0.19	0.68	0.00
4	2.21	1.62	1.60	0.39	0.55	0.08

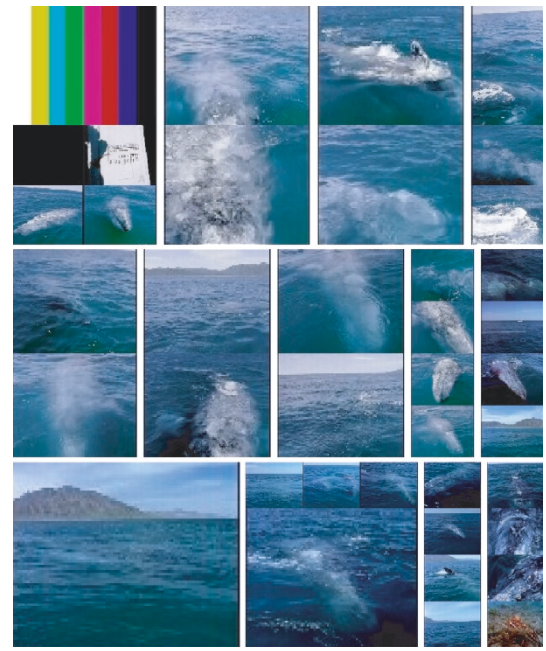


Figure 8: A page layout for parameters $\mathcal{N} = 40$ and $\mathcal{R} = 1.2$.

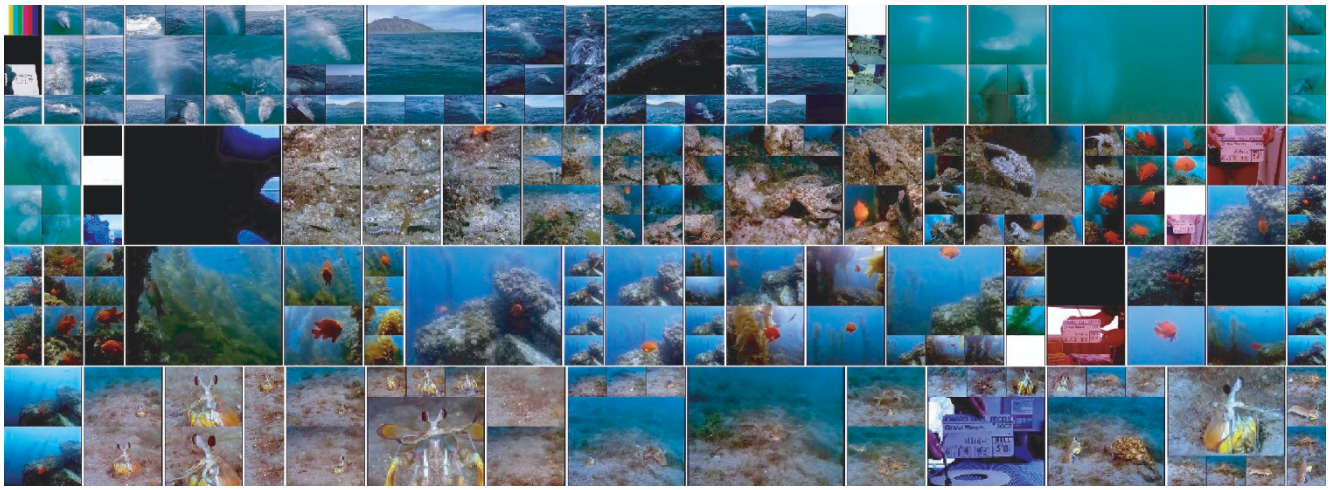


FIGURE 9: A page layout for parameters $\mathcal{N} = 250$ and $\mathcal{R} = 1/3$.

0.5% of the value of cost function. Therefore, the error can be disregarded and this result shows that incomparably faster suboptimal solution achieves practically even results with the optimal method. The optimal algorithm lays out 120 frames on a page in approximately 30 minutes, while the suboptimal algorithm does it in a fraction of a second on a standard desktop PC (Pentium 4, 2.4 GHz with 1 GB RAM).

An example of a page layout with parameters $\mathcal{N} = 40$ and $\mathcal{R} = 1.2$ is depicted in Figure 8. It shows how intuitive and swift the browsing of a 10 minute sequence can be. An hour of the raw video material is summarised in 7 comprehensible pages, with a possibility for a production professional to flip through the content of an archive and get familiar with the available content, from the desired topical takes to unexpected creative flashes and cutaways. In addition, summary of a 1 hour long tape with parameters $\mathcal{N} = 250$ and $\mathcal{R} = 1/3$ is presented in Figure 9. It demonstrates how quickly an end-user, whether professional or not, can become familiar with not only the major characteristics, but as well some important details about the content of the tape.

7.3. Results of the layout for mobile devices

In order to evaluate results of the salient region extraction, a set of key frames from wildlife videos is hand labelled with the most representative regions. By doing this, an objective measure of high-level abstraction and representation of analysed content is determined. Firstly, a collection of approximately 200 key frames extracted randomly from a large wildlife video archive is sequentially presented to an expert user. The user marks a single rectangular image region, that he/she considers as the most relevant and would most appropriately represent the image in the final layout.

These results were compared to automatically determined cropping region $\langle \mu_{xj} \pm \mathcal{K} \cdot \sigma_{xj}, \mu_{yj} \pm \mathcal{K} \cdot \sigma_{yj} \rangle$, where $\mathcal{K} \in [0,4]$. As depicted in Figure 10, if the hand-labelled ground truth was marked as A_{grt} and the automatically estimated salient image regions A_{est}, the overlapping area-labelled TP was considered as true positive detection, FN as

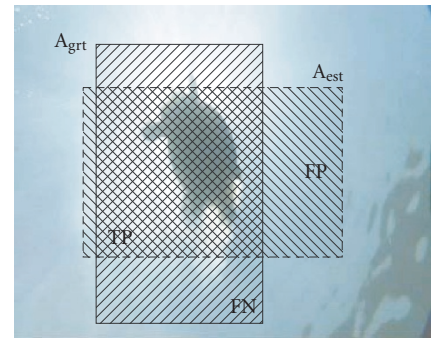


FIGURE 10: Comparison of the hand-labelled ground truth A_{grt} and automatically estimated salient image regions A_{est}.

false negative, while FP counts as the false positive detection. Thus, we can define measures of precision P and recall R as

$$R = \frac{TP}{TP + FN}, \qquad P = \frac{TP}{TP + FP}. \tag{31}$$

Both measures are calculated for values of $\mathcal{K} \in [0,4]$, and the resulting ROC curve is given in Figure 11. At the knee point of the graph in Figure 11, a balance of false positives and false negatives is achieved. For values of $\mathcal{K} < \mathcal{K}_{opt}$, some salient regions are left outside the estimated cropped region, while for $\mathcal{K} > \mathcal{K}_{opt}$ more nonsalient image regions are included.

To determine the optimal value of the parameter $\mathcal{K}$, we located the maximum of the F-measure (or harmonic mean) of the precision and recall values $F = 2 \cdot R \cdot P/(R + P)$, as depicted in Figure 12. As expected, the optimal value $\mathcal{K}_{opt} = 2.06$ is located exactly at the knee point of the ROC curve. The corresponding values of the precision and recall are $P(\mathcal{K}_{opt}) = 0.75$ and $R(\mathcal{K}_{opt}) = 0.89$, respectively.

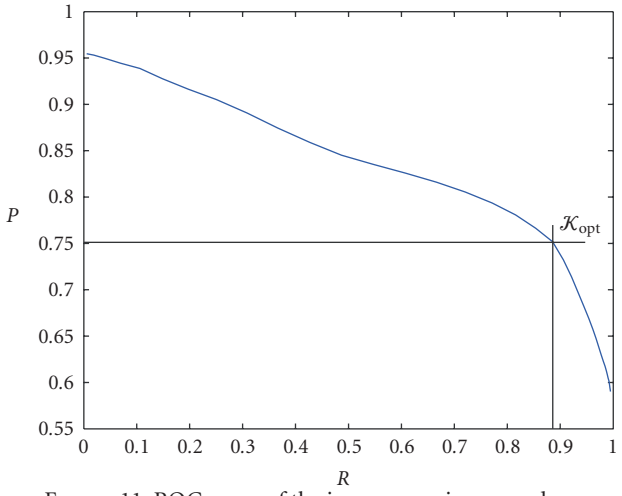


Figure 11: ROC curve of the image cropping procedure.

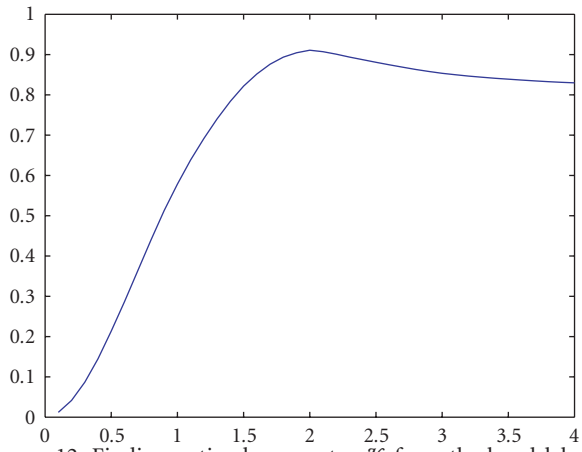


Figure 12: Finding optimal parameter $\mathcal{K}$ from the hand-labelled ground truth.

Figure 13: Examples of key-frame cropping using visual attention modelling for the optimal value of the parameter $\mathcal{K}$. The rectangular region displayed is $\langle \mu_{xj} \pm \mathcal{K}_{\text{opt}} \cdot \sigma_{xj}, \mu_{yj} \pm \mathcal{K}_{\text{opt}} \cdot \sigma_{yj} \rangle$.

(a) (b)

Figure 14: Final screen layouts of salient key-frame regions. High density of semantically important information is observable, thus delivering relevant visuals in a highly compact way.

In Figure 13 a set of source key frames with overlayed rectangular cropping regions $\langle \mu_{xj} \pm \mathcal{K}_{\text{opt}} \cdot \sigma_{xj}, \mu_{yj} \pm \mathcal{K}_{\text{opt}} \cdot \sigma_{yj} \rangle$ are depicted. The maximal time reaction parameter $\mathcal{T}_{\max}$ is empirically set to 300 milliseconds. The results show excellent selection of regions chosen by the attention model, especially knowing that the algorithm has been designed to operate in a fully unsupervised manner.

Two examples of the final layouts adapted for small screen devices are depicted in Figure 14. One can observe that the layouts do convey major features of the content and its semantics, whilst maintaining dense packing of the images on a small size screen. This method runs in real-time on a standard PC configuration, allowing for live production of summaries while the video is being played or broadcasted.

8. CONCLUSION

This paper presents a video summarisation and browsing algorithm that produces compact summaries for both high-end production environments as well as mobile devices. The system exploits the narrative structure of comics using well-known intuitive rules, creating visual summaries in an efficient and user-centered way.

A robust real-time key-frame extraction algorithm exploits production rules to select the best visual representation for a given shot. The results are evaluated by comparing them with the ground truth that was manually labelled by production professionals. The importance of key frames in the final layout is prioritised utilising a novel approach that balances the dominant visual representability and discovery of unanticipated content utilising a specific cost function and an unsupervised robust spectral clustering technique. Frames with higher importance are displayed larger in the final layout.

The final layout targeting high-end production tools are generated using an efficient optimisation algorithm based on dynamic programming. The results of the optimisation are benchmarked against the optimal solution and proved to be practically identical. From the results presented, one can observe that the approximation error introduced by the suboptimal solution is insignificant, whilst the processing is much faster, enabling real-time interaction with a long video sequence. A summary of an hour long video, comprising 250 shots, can be browsed swiftly and easily, while the creative process of finding interesting as well as representative content is significantly augmented using the comic-like layout.

On the other hand, the system targeting mobile users applies intelligent image cropping driven by the visual attention model. This algorithm shows excellent results in conveying

semantics as well as appearance of the summarised content. Finally, the layout algorithm that utilises dynamic programming achieves a high density of relevant information displayed on a small size screen. The results show high precision and recall of the image cropping results evaluated against hand-labelled ground truth.

Future work will be directed towards an extension of the summarisation algorithm towards interactive representation of visual content. Having the potential to create layouts on various types of displays and a fast system response, this algorithm could be used for interactive search and browsing of large video and image collections. In addition, a set of high-level rules of comics grammar [2] will be learned and exploited to improve representation of time in such a space constrained environment.

ACKNOWLEDGMENT

This work was supported by the ICBR project within the 3C Research, Digital Media and Communications Innovation Centre (http://www.3cresearch.co.uk).

REFERENCES

[1] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1349–1380, 2000.

[2] S. McCloud, *Understanding Comics*, Tundra, Northampton, UK, 1993.

[3] R. D. Dony, J. W. Mateer, and J. A. Robinson, "Techniques for automated reverse storyboarding," *IEE Proceedings: Vision, Image and Signal Processing*, vol. 152, no. 4, pp. 425–436, 2005.

[4] J. P. Collomosse, D. Rowntree, and P. M. Hall, "Video analysis for cartoon-like special effects," in *Proceedings of the 14th British Machine Vision Conference (BMVC '03)*, pp. 749–758, Norwich, UK, September 2003.

[5] S. Uchihashi, J. Foote, A. Girgensohn, and J. Boreczky, "Video manga: generating semantically meaningful video summaries," in *Proceedings of the 7th ACM International Multimedia Conference & Exhibition (MULTIMEDIA '99)*, pp. 383–392, Orlando, Fla, USA, October-November 1999.

[6] A. Girgensohn, "A fast layout algorithm for visual video summaries," in *Proceedings of IEEE International Conference on Multimedia and Expo (ICME '03)*, vol. 2, pp. 77–80, Baltimore, Md, USA, July 2003.

[7] J. Ćalić and B. Thomas, "Spatial analysis in key-frame extraction using video segmentation," in *Proceedings of the 5th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS '04)*, Lisboa, Portugal, April 2004.

[8] R. E. Bellman and S. E. Dreyfus, *Applied Dynamic Programming*, Princeton University Press, Princeton, NJ, USA, 1962.

[9] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[10] O. de Bruijn and R. Spence, "Rapid serial visual presentation: a space-time trade-off in information presentation," in *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '00)*, pp. 189–192, Palermo, Italy, May 2000.

[11] S. Porter, M. Mirmehdi, and B. Thomas, "Temporal video segmentation and classification of edit effects," *Image and Vision Computing*, vol. 21, no. 13-14, pp. 1097–1106, 2003.

[12] J. Ćalić, N. W. Campbell, M. Mirmehdi, et al., "ICBR: multimedia management system for intelligent content based retrieval," in *Proceedings of the 3rd International Conference on Image and Video Retrieval (CIVR '04)*, vol. 3115 of *Lecture Notes in Computer Science*, pp. 601–609, Springer, Dublin, Ireland, July 2004.

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

[14] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81)*, pp. 674–679, Vancouver, BC, Canada, August 1981.

[15] L. V. Kuleshov and R. Levaco, *Kuleshov on film: writings by Lev Kuleshov*, University of California Press, Berkeley, Calif, USA, 1974.

[16] J. Derrida, *Of Grammatology*, Johns Hopkins University Press, Baltimore, Md, USA, 1997.

[17] F. R. K. Chung, *Spectral Graph Theory*, vol. 92 of *CBMS Regional Conference Series in Mathematics*, American Mathematical Society, Providence, RI, USA, 1997.

[18] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 316–323, 1999.

[19] J. Shi and J. Malik, "Normalized cuts and image segmentation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '97)*, pp. 731–737, San Juan, Puerto Rico, USA, June 1997.

[20] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Proceedings of the 18th Annual Conference on Neural Information Processing Systems (NIPS '04)*, Vancouver, BC, Canada, December 2004.

[21] M. J. A. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: analysis and an algorithm," in *Advances in Neural Information Processing Systems 14 (NIPS '01)*, pp. 849–856, Vancouver, BC, Canada, December 2001.

[22] M. Polito and P. Perona, "Grouping and dimensionality reduction by locally linear embedding," in *Advances in Neural Information Processing Systems 14 (NIPS '01)*, pp. 1255–1262, Vancouver, BC, Canada, December 2001.

[23] T. W. Ridler and S. Calvard, "Picture thresholding using an iterative selection method," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 8, no. 8, pp. 629–632, 1978.

[24] W. Eisner, *Comics and Sequential Art*, Poorhouse, Tamarac, Fla, USA, 2001.

[25] G. E. Andrews, *The Theory of Partitions*, vol. 2 of *Encyclopedia of Mathematics and Its Applications*, Addison-Wesley, Reading, Mass, USA, 1976.

[26] A. Nijenhuis and H. S. Wilf, *Combinatorial Algorithms: For Computers and Calculators*, Computer Science and Applied Mathematics, Academic Press, New York, NY, USA, 2nd edition, 1978.

[27] A. Lodi, S. Martello, and M. Monaci, "Two-dimensional packing problems: a survey," *European Journal of Operational Research*, vol. 141, no. 2, pp. 241–252, 2002.

[28] P. E. Sweeney and E. R. Paternoster, "Cutting and packing problems: a categorized, application-orientated research bib-

liography," *Journal of the Operational Research Society*, vol. 43, no. 7, pp. 691–706, 1992.

[29] R. Spence, "Rapid, serial and visual: a presentation technique with potential," *Information Visualization*, vol. 1, no. 1, pp. 13–19, 2002.

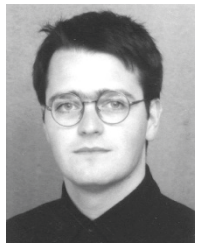
[30] T. Tse, G. Marchionini, W. Ding, L. Slaughter, and A. Komlodi, "Dynamic key frame presentation techniques for augmenting video browsing," in *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '98)*, pp. 185–194, L'Aquila, Italy, May 1998.

[31] L. Itti and C. Koch, "Computational modelling of visual attention," *Nature Reviews Neuroscience*, vol. 2, no. 3, pp. 194–203, 2001.

[32] D. Walther, *Interactions of visual attention and object recognition: computational modeling, algorithms, and psychophysics*, Ph.D. thesis, California Institute of Technology, Pasadena, Calif, USA, 2006.

[33] J. Calic, D. Gibson, and N. W. Campbell, "Efficient Layout of Comic-Like Video Summaries," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 7, pp. 931–936, 2007.

[34] J. Ćalić, N. W. Campbell, A. Calway, et al., "Towards intelligent content based retrieval of wildlife videos," in *Proceedings of the 6th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS '05)*, Montreux, Switzerland, April 2005.



Janko Ćalić is a Research Associate and Assistant Lecturer at the Department of Computer Science, University of Bristol as well as the Visiting Lecturer at the School of Engineering and Mathematical Sciences at City University, London. He had been previously with Multimedia and Computer Vision Lab at Queen Mary, University of London where he was awarded his Ph.D. title. He has been an Active Member of various international projects and networks of excellence in areas of media management. His research interests include content-based video indexing and retrieval, computational media aesthetics and multimedia repurposing.

Neill W. Campbell is a Senior Lecturer in the Department of Computer Science (Faculty of Engineering), University of Bristol. He received both his B.Eng. in computer systems engineering and Ph.D. in computer vision from the University of Bristol. He has published over 50 journal and conference papers in the fields of computer vision and computer animation. His particular interests are in the searching, understanding, and classification of large film archives, repurposing video for computer animation and in synthesising and understanding facial expressions.