

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
27 July 2006 (27.07.2006)

PCT

(10) International Publication Number
WO 2006/076760 A1

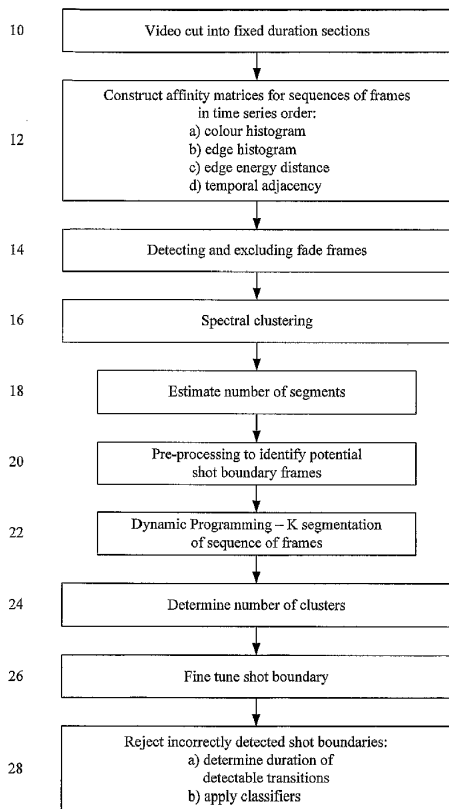
- (51) International Patent Classification:
G06K 9/00 (2006.01) G06F 17/30 (2006.01)
- (21) International Application Number:
PCT/AU2006/000012
- (22) International Filing Date: 6 January 2006 (06.01.2006)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
2005900278 24 January 2005 (24.01.2005) AU
2005901525 29 March 2005 (29.03.2005) AU
- (71) Applicant (for all designated States except US): NATIONAL ICT AUSTRALIA LIMITED [AU/AU]; Bay 15 Locomotive Workshop, Australian Technology Park, Garden Street Eveleigh, NSW (AU).
- (72) Inventors; and
- (75) Inventors/Applicants (for US only): YU, Zhenghua [AU/AU]; Bay 15 Locomotive Workshop, Australian Technology Park, Garden Street Everleigh, NSW (AU).

- VISHWANATHAN, Swaminathan, Venkata, Narayana [AU/AU]; Bay 15 Locomotive Workshop, Australian Technology Park, Garden Street Everleigh, NSW (AU).
- SMOLA, Alex [AU/DE]; Bay 15 Locomotive Workshop, Australian Technology Park Garden Street, Eveleigh, NSW (AU).
- (74) Agent: F B RICE & CO; Level 23, 44 Market Street, Sydney, NSW 2000 (AU).
- (81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.
- (84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH,

[Continued on next page]

WO 2006/076760 A1

(54) Title: SEQUENTIAL DATA SEGMENTATION



(57) Abstract: The invention concerns sequential data segmentation using spectral clustering. Attributes of data samples are extracted to construct affinity matrices for sequences of samples in sequential order. K-segmentation is then applied to a representation of the sequences of samples derived from the affinity matrices to identify K-segments. Each K-segment is comprised of representations of samples in sequential order. In this way data samples are not viewed as independent and this prevents data samples from the same true segment being assigned to different clusters. The invention respects the sequential semantics of the data and ensures that adjacent samples are consistently assigned to the same cluster. An example of sequential data segmentation is the segmentation of sequential video frames into separate shots. The invention also concerns a computer system and computer software able to perform sequential data segmentation.



GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— with international search report

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Title

SEQUENTIAL DATA SEGMENTATION

Technical Field

5 The invention concerns sequential data segmentation using spectral clustering. Aspects of the invention concern a method, system and computer software able to perform sequential data segmentation.

Background Art

10 Sequential data, is perhaps the most frequently encountered type of data existing in nature. Examples include speech, video, climatological or industrial processes, financial markets and biometrics engineering. For instance, EEG, MEG, PCG and ECG, also DNA sequences, context recognition, computer graphics and video sequences, to name just a few.

15 Segmenting sequential data into homogeneous sections is an essential initial step in processing sequential data.

 Taking as an example video content management systems that are used help to index, search and retrieve video. A shot is the most basic semantic structure of video and is a sequence of frames bounded by transitions to adjacent sequences of frames.

20 Examples of transitions are cuts, dissolves, wipes or flashes. After video shots are segmented, indexing and analysis of the segments can be performed. For example, security video can be isolated into segments interesting to surveillance staff and made accessible for further analysis.

 Known automated video shot segmentation systems are based on heuristics or
25 specialized methods to detect each of the different transition types in a video. Similarly most of the techniques for automated sequential data segmentation rely on specialized domain knowledge about the problem.

 Video shot segmentation can be performed based on a graph of the video and shot boundaries identified from the graph. A weighted undirected graph $G = (S, E)$
30 can be constructed where the vertices of the graph $S = \{s_1, \dots, s_n\}$ represent the n video frames in the video sequence, and an edge is formed between every pair of vertices. The weight of each edge, $A(i, j)$, represents the similarity between frames i and j . Finding shot boundaries is equivalent to finding cuts in graph G while respecting the temporal semantics of the video sequence.

35 Spectral clustering refers to a group of clustering methods that solve the normalised cut problem by calculating the first few leading eigenvectors of a graph Laplacian derived from an affinity matrix and perform clustering based on the

eigenvectors. Each element of the affinity matrix denotes the similarity between frames. A spectral clustering method has been proposed in A. Y. Ng, M. Jordan, and Y. Weiss, "On Spectral Clustering: Analysis and an Algorithm," in NIPS 14, 2002 ("Ng et al"). The algorithm is summarized here. Given a set of points $S = \{s_1, \dots, s_n\}$ in

5 R^l that we want to cluster into k subsets the following is performed:

1. Form the affinity matrix $A \in R^{n \times n}$ defined by $A_{ij} = \exp \frac{d^2(i,j)}{2\sigma^2}$ if $i \neq j$, and $A_{ii} = 0$. $d(i, j)$ denotes the Euclidean distance between s_i and s_j .

2. Define D to be the diagonal matrix where $D_{ii} = \sum_j A_{ij}$, and construct the normalised graph Laplacian given by $L = D^{-1/2} A D^{-1/2}$.

10 3. Find x_1, x_2, \dots, x_k , the k largest eigenvectors of L , and form the matrix $X = [x_1, x_2, \dots, x_k] \in R^{n \times k}$ by stacking the eigenvectors in columns.

4. Form the matrix Y from X by renormalizing each of X 's rows to have unit length (i.e. $Y_{ij} = X_{ij} / (\sum_j X_{ij}^2)^{1/2}$).

15 5. Treating each row of Y as a point in R^k , cluster them into k clusters via K-means.

6. Finally, assign the original point s_i to cluster j if and only if row i of the matrix Y was assigned to cluster j .

Summary of the Invention

20 The invention provides sequential data segmentation using spectral clustering. Data samples are extracted to construct affinity matrices for sequences of samples in sequential order. K-segmentation is then applied to a representation of the sequences of samples derived from the affinity matrices to identify K-segments, and each segment is comprised of representations of samples in sequential order.

25 Spectral clustering methods such as those that integrate K-means view all data samples as independent and assign those samples to clusters individually. However, in the case of sequential data all samples between two segment boundaries are not independent as they are part of an ordered series. K-means is not optimal for sequential data segmentation as data samples from the same segment may be assigned to different
30 clusters. To overcome this limitation we impose sequential ordering on samples when constructing the affinity matrices. The invention respects the sequential semantics of the data and ensures that adjacent samples are consistently assigned to the same cluster.

Elements in a row or a column of the affinity matrices represent data samples in sequential order. A graph Laplacian may be derived from the affinity matrices and then

the leading eigenvectors of the Laplacian are solved. Further, the graph Laplacian may be normalised before solving for the eigenvectors.

K-segmentation may be applied to the normalised eigenvectors representing the sequences of data samples in sequential order, such as the eigenvectors of the Laplacian
5 of the affinity matrices.

K-segmentation may comprise identifying homogenous segments in the representation of the sequences of data samples. An estimation of the number of segments K is used to initially identify K homogenous segments.

Alternatively, K-segmentation may comprise identifying significant transitions
10 in the representation of sequences of data points.

Spectral clustering may involve solving for the largest eigenvectors of the normalised graph Laplacian and performing clustering based on the eigenvectors. An alternative is to use eigenvectors of the graph Laplacian directly. The largest eigenvectors are stacked into columns to form matrix X such that the renormalisation of
15 the rows of X gives matrix Y . Section S is represented in the space spanned by the rows of Y i.e. by n samples $\{y_1, \dots, y_n\} \in R^k$.

K-segmentation is then performed on section S to ensure that each sample of a segment is consistently assigned to the same cluster. For instance, this can be done by using dynamic programming. K-segmentation can be conducted on Y to find a
20 sequence $y_{s_1} y_{s_2} \dots y_{s_k}$ of k segments such that $y_{s_1} y_{s_2} \dots y_{s_k} = \{y_1, \dots, y_n\}$ and each y_{s_i} is not empty. The k segments are found by considering the homogeneity of the segments.

Attributes of the sample data may be extracted to construct the affinity matrix. In the case of video shot segmentation, the extracted attributes of the data samples includes color attributes, edge attributes, edge energy distance values between
25 sequential samples and the temporal adjacency of samples. The attribute of random impulse noise may also be included in the affinity matrix.

The edge attributes may incorporate one or more local, semi-global or global edge histograms. For instance, an image of a single frame is divided into sub-images and edges of the sub-images are categorised to create multiple local edge histograms.
30 From the local edge histogram one global edge histogram is calculated and multiple semi-global histograms are calculated. Alternatively, edge detection may utilize Sobel edge detector templates and the sub-samples that overlap.

The extracted attributes of the sample data may also be based on the temporal adjacency of the samples. Utilising both local and global information increases the
35 accuracy of the segmentation when compared to most existing systems.

Before K-segmentation is applied, pre-processing may be applied to identify potential boundaries and to reduce the computational complexity of the K-segmentation.

5 K-segmentation may be applied based on an estimation of the number of segments.

Following K-segmentation, the number of segments is determined by rejecting segments that do not meet a predetermined threshold, such as rejecting a segment if the normalized conductance of any of its boundaries is less than a predetermined threshold. This rejects the weaker segments that may result in the false identification of a
10 segment.

The boundary of a segment may be fine tuned by considering the sample at the identified boundary of a segment and a predetermined number of adjacent samples and selecting the most suitable sample.

15 An incorrect boundary of a segment may be detected and rejected by extracting key features of the boundary and assessing those features using Support Vector machines classification. The key features may be extracted from the transition of the cut values over time.

In a further aspect the invention provides a system for the segmentation of sequential data using spectral clustering.

20 In yet a further aspect the invention provides computer software able to perform the method described above.

Brief Description of the Drawings

25 An example of the invention will now be described with reference to the accompanying drawings, in which:

Fig. 1 is flow chart of the method of performing shot segmentation;

Fig. 2 schematically shows the dividing of images into sub-images and blocks;

Fig. 3 schematically shows the dividing of sub-images into blocks that are also shifted by half a block size in comparison to the known MPEG-7 method;

30 Fig. 4 schematically shows the finite state machine used to determine the number of segments;

Fig. 5 is a schematic diagram of the segmentation of the time series S into k segments;

35 Fig. 6 is a graph plotting the cut value over time used to estimate the number of segments; and

Fig. 7 is a further graph plotting the cut value over time used to reject incorrectly detected shot boundaries.

Best Modes of the Invention

5 A method to automatically identify shots in a video sequence will now be described with reference to the flow diagram of Fig. 1.

Initially, a long video is cut 10 into fixed duration sections with overlaps between adjacent sections. For example, for a sports video a fixed duration section may be 400 frames with an additional 100 frames for overlaps. For news video a fixed
10 duration section may be 240 frames with 80 frames for overlaps. The fixed duration sections are the basic computing elements upon which shot segmentation is performed on.

Next, frame data are extracted to construct 12 affinity matrices for sequences of frames in time series order. Given N frames in a fixed duration section, affinity matrix
15 $A \in R^{n \times n}$ is defined by A_{ij} representing the similarity between frames i and j if $i \neq j$, and $A_{ii} = 0$. Subscripts i and j are both in the natural time series order of frames. For example, row k of affinity matrix $A \{A_{ik}, i = 1 \dots N\}$ represents the affinities of a sequence of frames in ascending order of time $\{1 \dots N\}$ to frame k . The affinity matrices of all the frames in the fixed duration section are constructed based on (a) color
20 histogram, (b) edge histogram, (c) edge energy distances and (d) temporal adjacency distances between the frames. Then the affinity matrices are adjusted for (e) random impulse noise.

(a) A color histogram is calculated in HSV color space. H is quantized into 16 bins, S and V are quantized into 4 bins each, so in total there are 256 bins (see
25 Manjunath B S, Salembier P, Sikora T (ed), *Introduction to MPEG-7, Multimedia Content Description Interface*, John Wiley & Sons, Ltd, 2002, "Manjunath et al"). The distance metric used is chi-squared distance.

(b) Following MPEG-7, an edge histogram is constructed with 80 bins (see Manjunath et al). Images are divided into 4 x 4 sub-images 50 as shown in Fig. 2.
30 Each sub-image is divided into a fixed number of blocks 52. Then each image block is partitioned into 2x2 block of pixels 54. Edges in the sub-images are categorized into five types: vertical, horizontal, 45° diagonal, 135° diagonal and non-directional edges where each 2x2 block is considered a pixel. The following simple templates are used in edge detection:

$$35 \quad \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}, \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} * \sqrt{2}, \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} * \sqrt{2}, \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} * \sqrt{2}.$$

Therefore each image is represented by 80 local edge histogram bins.

A global edge histogram and 65 semi-global edge histograms are computed from the 80 local histogram bins. For the global edge histogram, the five types of edge distributions for all sub-images are accumulated. For the semi-global edge histograms, subsets of sub-images are grouped.

L1 norm of the distance of local, semi-global and global histograms between two frames is adopted as the distance function. The distance of the global histogram difference is multiplied by five given the number of bins of the global histogram is much smaller than that of local and semi-global histograms. Other norms or distances may also be used based on domain knowledge about the problem on hand.

However, the following shortcomings are identified in the original MPEG-7 edge histogram descriptor. Firstly, the edge detectors are based on simple 2x2 templates which do not characterize edges well. Secondly, when partitioning sub-images into blocks 52, the blocks are not overlapped. As a result, small movement of camera or objects may lead to large variation of edge values, which is not desirable.

To address these shortcomings two improvements have been made. Firstly, the edge detection templates are replaced with Sobel edge detector which is more accurate at detecting edges. The new templates are:

$$20 \quad \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}, \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}.$$

Since we use only four directions, the local edge histogram has 64 bins instead of 80.

Secondly, when partitioning subimages into blocks, we use not only non-overlapping blocks (the same as MPEG-7), but also blocks whose origins are shifted by $\frac{1}{2}$ a block size in both horizontal and vertical directions compared to the non-overlapping blocks. This is shown in Fig. 3 where Fig. 3(a) schematically shows the blocks are divided according to the MPEG-7 method (16 non-overlapping blocks in total) and Fig. 3(b) shows the proposed method of partitioning blocks (16 non-overlapping blocks and 9 blocks shifted in both the horizontal and vertical directions; 25 blocks in total; all the blocks are of equal size).

(c) The following is a description of the edge energy distance calculation. Dissolve and fade are two major types of gradual transitions. In dissolve, video frames from two shots are gradually mixed together by (usually) linear addition

$$f=a*t+b*(1-t),$$

where a and b are two shots, t is the mixing ratio which is a variable of time, and f is the output. As every frame during a dissolve transition is the mixture of two frames, they usually contain weaker edges than frames before and after the dissolve. Therefore statistics on edge energy provides valuable clue to detect dissolves. The method
5 described here is applicable to fades as well, however we use a more specialized fade detector to be described later.

Without losing generality, we incorporate edge energy statistics to assist the detection of gradual transitions.

When performing edge detection (see description of (b) edge histogram above),
10 the outputs after applying edge detection can be denoted as e_h, e_v, e_45, e_135 respectively. The edge value of the 2*2 block of pixels is calculated as:

$$\text{Edge_value} = \max(e_h, e_v, e_{45}, e_{135}).$$

The square root edge value is calculated as:

$$\text{Edge_sqrt_value} = \sqrt{e_h^2 + e_v^2 + e_{45}^2 + e_{135}^2}.$$

15 The following four statistics about the edge values are calculated for each video frame:

$$\text{mean_edge_value} = \frac{\sum(\text{edge_value}[i])}{\text{number_of_edges}}$$

$$\text{mean_edge_sqrt_value} = \frac{\sum(\text{edge_sqrt_value}[i])}{\text{number_of_edges}}$$

$$\text{std_edge_value} = \frac{\sum((\text{edge_value}[i] - \text{mean_edge_value})^2)}{\text{number_of_edges}}$$

20
$$\text{std_edge_sqrt_value} = \frac{\sum((\text{edge_sqrt_value}[i] - \text{mean_edge_sqrt_value})^2)}{\text{number_of_edges}}$$

Although the four of these statistics all characterize the "edgeness" of the video frame and can be used to calculate the edge energy distance between two frames, empirically we have found std_edge_sqrt_value provides the best performance.

Given two video frames i and j, and the respective std_edge_sqrt_values, the
25 edge_energy_distance (EED) between these two frames is defined as:

$$\text{EED}[i,j] = \text{abs}(\min(\text{thres}, \text{std_edge_sqrt_value}[i]) - \min(\text{thres}, \text{std_edge_sqrt_value}[j])).$$

There is a threshold used to saturate std_edge_sqrt_value as large values tend to be noisy and useless (gradual transitions will mainly lead to small edge values).

(d) Temporal adjacency is then integrated into the final affinity calculation so
30 that the calculated affinity between frames incorporates both color and edge histograms, edge energy distance, and temporal adjacency.

Finally, the affinity between frames is calculated. $A_{ij} = A_{ij}^h A_{ij}^t$ if $i \neq j$, and

$A_{ii} = 0$ where $A_{ij}^t = \exp\left(-\frac{d_i^2(i,j)}{2\sigma_i^2}\right)$ represents the affinity due to temporal adjacency, $d_i(i, j)$ is the difference in frame numbers between frames i and j, and

$A_{ij}^e = \exp \frac{d_{EED}^2(i,j)}{2\sigma_{EED}^2}$ represents the affinity due to edge energy distance $d_{EED}(i, j)$, and

$A_{ij}^h = \exp \frac{d_{ch}^2(i,j)d_{eh}^2(i,j)}{2\sigma_h^2}$ represents the affinity due to color and edge histograms as described in (a) and (b). For example, $\sigma_k = 56469$ and $\sigma_t = 5$ were derived through experimentation. The value of σ_t determines the influence of frames far away from the

5 current one.

(e) Here we consider that the affinity matrix is being constructed in the presence of random impulse noise. Unfortunately video signals are usually corrupted with noise. Here we only consider noise that affects the global characteristics of a frame or few frames. Examples of such noise include flashes, blur (due to auto-
10 focusing), and sudden movement of irrelevant foreground objects close to the camera. We generalize the noise as random impulse noise (in temporal domain).

If we view frames before and after the noise, these frames usually are similar. In general, we assume that frames close by in time should be more similar than frames that are further apart. When translated into the calculation of the affinity matrix, we
15 can assume that the affinity between frames i and $i+t$ should be greater than both the affinity between frames i and $i+t+k$ where $k>0$, and the affinity between frames $i-l$ and $i+t$ where $l>0$. Consequently we apply the following adjustment when constructing the affinity matrix:

Given frames i and $i+t$, for any pair of k and l where $T>k>0$ and $T>l>0$, if
20 $A_{i,i+t} < A_{i-l,i+t+k} \exp \frac{(l+k)^2}{2\sigma_t^2}$, then make $A_{i,i+t} = A_{i-l,i+t+k} \exp \frac{(l+k)^2}{2\sigma_t^2}$. Here T is a threshold which limits the duration to check the affinities. T should be determined according to the characteristics of the video signal. In our experiments we used $T=5$ which was sufficient to handle most noise. This generic solution does not depend on the exact cause of noises.

25 Next, fades are detected and excluded from further calculations 14. While fades can be detected using our generic shot segmentation method, since we have detected edges already it is more straightforward to detect fades using the edge information. Simple fades are just black frames; fancier ones may use blurring of non-black frames. In both cases they can be characterized by low edge values.

30 The fade detector has two modules:

- For the first module (fade detector 1) if the percentage of detected edges among all pixels in the current frame $<$ threshold 1 and the maximum edge value $<$ threshold 2, then a fade frame is detected.

- For the second module (fade detector 2), as described above each image is previously divided into 16 sub-images to calculate the edge values. For each sub-image, the following conditions are tested to determine if the standard deviation of its edge value $< \max(\text{thres1}, \text{mean_edge_value_of_the_subimage} * \text{const1})$, and if the maximum edge value $< \max(\text{thres2}, \text{mean_edge_value_of_the_subimage} * \text{const2})$. If all sub-images tested meet these conditions, then a fade frame is found.

When detecting a fade, the four corner sub-images are ignored as they are likely to contain station logos which may exist during fades.

- 10 Detected fade frames are excluded in further calculations.

Next, spectral clustering is performed 16. Let us define D as the diagonal matrix where $D_{ii} = \sum_j A_{ij}$, and construct the normalised graph Laplacian $L = D^{-1/2} A D^{-1/2}$. Next, x_1, x_2, \dots, x_k k largest eigenvectors of L are found. The matrix $X = [x_1, x_2, \dots, x_k] \in R^{n \times k}$ is then formed by stacking the eigenvectors in columns. The matrix Y from X is formed by renormalizing each of X 's rows to have unit length (i.e. $Y_{ij} = X_{ij} / (\sum_j X_{ij}^2)^{1/2}$).

Next the number of segments is estimated 18. A fixed number of segments (such as $k=6$) can be used. This is sufficient for most video sequences, however there are video sequences that contains a lot of transitions (mostly cuts). Using fixed $k=6$ in this case will lead to false deletion of transitions.

Therefore we have developed a method to estimate the number of cuts N_{est} in each section (e.g., 240 frames). $N_{\text{est}}+6$ is then used as the initial number of segments (in stead of 6).

Given an affinity matrix $A[i,j]$, firstly we define two vectors which describe the sum of affinities to frames before and after the current one:

$$bw[i] = \sum_{j=i-T}^{i-1} A[i, j]$$

$$fw[i] = \sum_{j=i+1}^{i+T} A[i, j]$$

T is a threshold that limits the number of frames we check the affinity of. Transitions from large bw/fw values to small values indicate the existence of cut transition, and vice versa.

Referring now to Fig. 4, a three-state machine is defined. Peak state can be entered from any state if $bw[i] > \text{thres2}$ and $fw[i-1] > \text{thres2}$, and bottom state can be entered from any state if $bw[i] < \text{thres1}$ and $fw[i-1] < \text{thres1}$, where i is the current frame number, and thres1 and thres2 are thresholds (0.25 and 1.75 used in experiments). The

estimated number of segments N_{est} equals to the average of the number of state transitions from peak to bottom and from bottom to peak.

Next, pre-processing is applied for dynamic programming. Directly applying dynamic programming in spectral segmentation without pre-processing has two drawbacks:

(a) Dynamic programming may not identify exactly the shot boundaries with the least cross segment similarities; and

(b) The computational complexity of dynamic programming is quite high. Therefore a pre-processing step is applied before dynamic programming to isolate the potential cut points.

Pre-processing may, for instance, consist of edge finding. Let $y(i)$ be a point in R^k whose coordinates are taken from row i of the Y matrix, i.e., $y(i) = [Y_{i1} \ Y_{i2} \ \dots \ Y_{ik}]^T$ which represents data sample i . Firstly, first degree derivative of $y(i)$ is calculated $g'(i) = |y(i) * h_1|$, where $h_1 = [1 \ 0 \ -1]^T$ and * denotes convolution. Then second degree derivative is calculated $g''(i) = g'(i) * h_2$, where $h_2 = [-1 \ 1]^T$.

Edges may be detected through finding zero crossing of $g''(i)$. However there exist lots of zero value points in $g''(i)$ as the data may be noisy. Therefore edges are detected as the mean of a local maximum/minimum pair in order to detect strong edges only. We also only detect one candidate edge point in a neighbourhood of T samples where T is a variable which may be adjusted depending on the application (for example $T=10$ was adopted in some experiments).

Suppose $g''(l)$ is a local maximum ($\pm T/2$ samples) and $g''(l) > edge_thres/2$ where $edge_thres$ is a threshold, then try to find local minimum $g''(m)$ between $l+1$ and $l+T$. If $g''(m) < -edge_thres/2$, then an edge is located at $(l+m)/2$. Edge points detected in this way constitute candidate points for dynamic programming.

Suppose there are p candidates, then the complexity of dynamic programming is reduced from $O(kn^2)$ to $O(kp^2)$. In experiments N values of 400 and 240 were used, while p is usually < 10 .

Dynamic programming is then applied on candidate points determined by pre-processing. Here spectral K-segmentation is performed on frame sequences in the fixed duration sections rather than each single frame. For example, a time series S consists of n samples $\{s_1, \dots, s_n\}$. Similar to the notations in J. Himberg, K. Korpiaho, H. Mannila, J. Tikanmäki, "Time series segmentation for context recognition," Proc. 2001 IEEE Int. Conf. on Data Mining, pp. 203 – 210, 2001 ("Hinberg et al"), $ss(a, b)$ is

defined as a segment of the time series S, ie, the consecutive samples $\{s_a, \dots, s_b\}$ where $a \leq b$.

As mentioned before, k-segmentation of S is a sequence $ss_1 ss_2 \dots ss_k$ of k segments such that $ss_1 ss_2 \dots ss_k = \{s_1, \dots, s_n\}$ and each ss_i is not empty. Equivalently the task is to find k-1 segment boundaries c_1, c_2, \dots, c_{k-1} , $1 < c_1 < c_2 < \dots < c_{k-1} < n$, where $ss_1 = ss(1, c_1)$, $ss_2 = ss(c_1 + 1, c_2)$, ..., $ss_k = ss(c_{k-1} + 1, n)$. $c_0 = 1$, and $c_k = n$. The k segments are schematically shown in Fig. 5. The number k could be a pre-determined value (eg, k=6) or the estimated number of segments (eg, k=N_est+6).

Spectral K-segmentation of the sequences of frames is performed in the space spanned by the first few eigenvectors of the constructed normalised graph Laplacian. Following spectral clustering step above, the same time series can be represented in the space spanned by rows of matrix Y by n samples $\{y_1, \dots, y_n\} \in R^k$. K-segmentation of time series S can be conducted on Y, ie, to find a sequence $ys_1 ys_2 \dots ys_k$ of k segments such that $ys_1 ys_2 \dots ys_k = \{y_1, \dots, y_n\}$ and each ys_i is not empty. Segmentation in Y space instead of S space is justified because the rows of Y will form tight clusters around k well separated points while such cluster structure may not be clear in S space (Ng et al).

A cost function is defined to illustrate the internal homogeneity of all segments $cost = \frac{1}{n} \sum_{i=1}^k \sum_{j=c_{i-1}+1}^{c_i} |y(j) - \mu_i|^2$ where μ_i is the mean vector of all the data y in segment $ys_i = yg(c_{i-1} + 1, c_i)$. Depending on the application other forms of cost function may be used as well.

The problem is to find the K segment boundaries c_i that minimize the cost, therefore it is an instance of K-segmentation problem. As suggested in Hinberg et al, K-segmentation problem can be solved optimally using dynamic programming. However we apply dynamic programming to the sequence of data in the eigenspace rather than the space of the raw input data. The complexity of dynamic programming is $O(kn^2)$ since the cost of segmentation can be calculated in linear time.

Next, the number of clusters is automatically determined. This is performed through thresholding of normalized conductance. This is achieved by starting from a large number of k potential segments (k=6 used in experiments which can be determined through experience, however see step 22 above that can be used to estimate the number of cuts) and then rejecting weak segments based on the criteria below.

If c_i is shot boundary between segments P and Q, then normalized conductance of cut c_i is defined as $\phi(C_i) = \frac{\sum_{i \in P, j \in Q} A_{ij}}{\min(\frac{a(P)}{|P|}, \frac{a(Q)}{|Q|})}$ where $a(P) = \sum_{i \in P, j \in P} A_{ij}$, and $|P|$ is

the number of vertices in segment P.

However, vertices near a true shot boundary usually have small edge weights as well, so they should be excluded in the normalized conductance calculation. Define $P^* = ss(c_{i-1} + \beta, c_i - \beta)$, and $Q^* = ss(c_i + \beta, c_{i+1} - \beta)$.

$$\phi^*(C_i) = \frac{\sum_{i \in P^*, j \in Q^*} A_{ij}}{\min(\frac{a(P^*)}{|P^*|}, \frac{a(Q^*)}{|Q^*|})}$$

$$\beta = \text{min_shot_length} / 4$$

Then shot boundary c_i is rejected if $\phi^*(C_i)$ is less than a threshold.

Next, the detected shot boundary point is fine-tuned in a 3-frame window $\{c_{i-1}, c_i, c_{i+1}\}$ by finding the vertex that produces the smallest ϕ^* and using that vertex as the shot boundary. Here the number three is a parameter that may be varied depending on the application.

Finally, the incorrectly detected shot boundaries are rejected using Support Vector Machines (SVM) classification. The method of determining the number of clusters can be improved for gradual transitions. Graph cut/spectral clustering only finds single cut points, and may cut a gradual transition twice, one at the beginning and one at the end of the transition.

We view the rejection of incorrectly detected shot boundaries as a supervised classification problem and we adopt SVMs as the classifier. The key step is to represent the curve/time series using features of extracted feature points for SVM.

Cut value is the sum of affinities across the current frame, which is a terminology inherited from graph cut methods. We plot the cut value curve over time. Fig. 6 shows a typical cut value curve of gradual transition. Frames in the middle of the transition may have higher cut values, and there are two bottoms in the transition. Spectral clustering may pick up one or two of the two bottoms shown in the curve, depending on the transition.

The strategy is to identify key points out of a potential gradual transition, extract features of the key points and use these features for SVM classification. However if the duration of the transition is short (≤ 5 frames), then there is no need to detect key points. Instead we just use frames near the transition for SVM classification.

Therefore we have defined two classifiers, one for short transition (≤ 5 frames), and one for long transition (> 5 frames).

However, since spectral clustering only finds the "cut" point (the point with the minimum cut value), we need to (a) determine the duration of the detected transitions first, then (b) apply the classifiers to reject incorrectly detected transitions.

(a) In order to determine the duration of the detected transitions several steps must be performed.

(i) Starting from the cut point B in Fig. 6, try to find the entry and exit points (A and C). To find entry point A, starting from the cut point B, search backwards to find either:

- the first frame i with $bw[i] < thres10$ and $bw[i-1] > thres11$. Thres10 and thres11 typically are $2.5e-5$ and 1, respectively. This condition typically represents sharp cuts; or
- the frame i with $fw[i]$ greater than thres1 and the difference between the edge value of frames i and $i-1$ is less than another threshold. Frame i may be further extended in the backwards direction by up to 3 frames if the difference between $fw[i-1]$ and $fw[i]$ is greater than thres3. Thres1 and thres3 typically are 3.75 and 0.125, respectively. This condition typically represents gradual transitions.

Similarly the exit point C can be found. Region A-B-C is called a detected transition region (DTR).

(ii) As gradual transitions may have two bottoms in the cut value curve, and sometimes only one of them has small cut value which can be detected by spectral clustering as a transition, we need to search for the first curves with a dip (i.e., similar to A-B-C) in the left and right-hand sides of the current transition region (A-B-C). The regions are found by locating the peak point first, then the nearest entry (transition from peak to bottom) point, then finding the nearest bottom point, then finding the next exit point similar to the process described in step 1. As an example, in Fig. 6, points D, E, F and G can be found in this way. Region E-F-G is called candidate transition region (CTR).

(iii) As a true gradual transition may be broken up into one DTR plus one CTR, or two DTRs, it is necessary to check whether we need to merge any DTR with a neighbouring CTR or DTR. As an example, the appearance of several DTRs and CTRs is shown in the graph of Fig. 7.

There are following sub-steps to merge transitions.

- Sub-step 1: for every DTR, check the probabilities that it should be merged with neighbouring DTR/CTR to both sides of the current DTR, and store the probabilities
- 5 • Sub-step 2: examine all the probabilities calculated. Starting from the highest probability one, merge the corresponding DTR/CTR if the probability is higher than a threshold. Repeat the process until all probabilities have been examined.

The probability to merge DTR/CTR is calculated via SVM. The SVM features used in the SVM classifier include features about the seven feature points A, B, C, D, 10 E, F, and G representing the first entry, first bottom, first exit, middle peak, second entry, second bottom, and second exit. Detailed features are:

- The relative location of the seven feature points expressed as relative distances
- The following features of all the seven feature points:
 - 15 cut value,
 - fw value,
 - bw value,
 - mean edge value,
 - mean luminance value and
 - 20 standard deviation of luminance value

of the whole image and four sub-image columns and four sub-image rows.

After this step, we have detected (and properly merged) transitions to be further classified as true or false transitions by SVMs. Two SVM classifiers are used depending on the duration of the transition.

25 In the SVM for short transitions, the following features are extracted for SVM classification:

- the duration of the detected transition
- the edge value of the beginning and the end frames of the detected transition
- the cut value, fw value, bw value, mean edge value, mean luminance value, 30 standard deviation of luminance value of seven neighbouring frames centered around the frame within the detected transition with the smallest cut value.
- features about the transition detected before the current transition and the frames between the previous and the current transitions (we use BT to represent these frames below), including: minimum cut value of BT, the distances between the 35 minimum cut location of BT and the beginning of the current transition and the end of the previous transition, maximum cut value of BT, the distances between

- the maximum cut location of BT and the beginning of the current transition and the end of the previous transition, the distance between the previous and the current transitions, the duration of the previous transition, the cut value of the end frame of the previous transition, pre_conti_peak, mean_eh_pre, std_eh_pre, mean_pp_pre, std_pp_pre. Pre_conti_peak counts the length of continuous peak frames (greater than a threshold) starting from the beginning of the current transition. Mean_eh_pre and std_eh_pre are the mean and the standard deviation of edge values between the current and the previous transitions. Mean_pp_pre and std_pp_pre are the mean and the standard deviation of cut values between the current and the previous transitions.
- features about the transition detected after the current transition and the frames between the next and the current transitions (we use BT to represent these frames), including: minimum cut value of BT, the distances between the minimum cut location of BT and the end of the current transition and the beginning of the next transition, maximum cut value of BT, the distances between the maximum cut location of BT and the end of the current transition and the beginning of the next transition, the distance between the next and the current transitions, the duration of the next transition, the cut value of the beginning frame of the next transition, post_conti_peak, mean_eh_post, std_eh_post, mean_pp_post, std_pp_post. Post_conti_peak counts the length of continuous peak frames (greater than a threshold) starting from the end of the current transition. Mean_eh_post and std_eh_post are the mean and the standard deviation of edge values between the current and the next transitions. Mean_pp_post and std_pp_post are the mean and the standard deviation of cut values between the current and the next transitions.

For the SVM for long transitions, the features used include those similar to short transition features, together with additional features. Given a detected transition, the local maximum and local minimum points in cut value curve are detected. Suppose pre and post represent the frame numbers of the beginning and the end of the transition, up to seven feature points are selected according to the rules below:

- if peak_num=0 and bottom_num=0, then use the following 5 feature point frames: pre, post, and 3 frames that equally divide post-pre
- if peak_num>0 and bottom_num=0, then use the following 5 feature point frames: pre, post, largest_peak_frame, the average of pre and largest_peak_frame, and the average of post and largest_peak_frame

- if peak_num=0 and bottom_num>0, then use the following 5 feature point frames: pre, post, smallest_bottom_frame, the average of pre and smallest_bottom_frame, and the average of post and smallest_bottom_frame
- 5 • if bottom_num=1 and peak_num>0, then use the following 5 feature point frames: pre, post, bottom_frame, largest_peak_frame, and the average of largest_peak_frame and either pre or post depending on where bottom_frame is located (the other side is chosen)
- 10 • if bottom_num>1, then use the following 5 feature point frames: pre, post, two bottom_frames with the smallest cut value, and the frame between the two bottom frames with the largest cut value

Two additional peak/bottom frames may be selected if they are not selected among the five frames mentioned above.

Having chosen the seven feature frames, a large number of features similar to 15 features used for short transition are used, namely:

- the duration of the detected transition
- the edge value of the beginning and the end frames of the detected transition
- the cut value, fw value, bw value, mean edge value, mean luminance value, standard deviation of luminance value of the 7 feature frames (if 20 less than 7 frames are chosen, non-informative features will be included).
- features about the transition detected before the current transition and the frames between the previous and the current transitions (the same as those for short transitions)
- 25 • features about the transition detected after the current transition and the frames between the next and the current transitions (the same as those for short transitions)

and the additional features include:

- type of the feature frames (eg, peak, bottom, others)
- 30 • relative distance between the feature frames.

This method of identifying segments is sequential data can be performed by a computer that is able to accept data input of sequential data and to store the sequential data. The computer then performs on a processor the steps outlined in Fig. 1 to produce an output of the identified K-segments. For example, the computer may accept 35 digital video frames and displays the identified shot segments on a display means, such

as a monitor. The computer may also create or update an index document that indexes and is used to navigate to the shot segments.

The parameters of the algorithm were derived using three TV sequences, each with about 10000 frames: two news segments from TV channels CCTV-4 and CCTV-9, and one sport (soccer) segment from TV channel Oriental TV.

The performance of the proposed algorithm was evaluated using a complete new set of TV video sequences. In total, two hours of news and one hour of sport video sequences were used in the test. The break down is as follows:

Category	Channel	Length (minutes)	Number of shot boundaries
News	CCTV-1	56	802
News	CCTV-4	17	120
News	Phoenix TV	47	557
Sport	Oriental TV	60	353

10

All the test sequences were never used in algorithm development. Moreover, we included two completely new TV channels: CCTV-1 and Phoenix TV. In the case of CCTV-4, the development set and the test set were captured at two days. In the case of Oriental TV Sport, the development set and the test set were from two different matches captured at two different days. We observed that news from CCTV-1, CCTV-4 and CCTV-9 exhibited different editing characteristics and there is little overlap of content among them so we regard them as different channels.

15

The test results are reported in the following table. Machine segmented shot boundaries were checked against manually segmented ground truth. As evident in the table, the proposed algorithm achieved high precision and recall. If breaking down the performance on different channels, we can see the performance is consistent over all four channels. We also calculated the different performance regarding to the transition type: cut and gradual transition (transition of >1 frames). The performance on gradual transition is slightly lower, mainly due to some special editing effects.

20

25

Test case	Precision	Recall	Total number of shot boundaries
All	96.4%	94.8%	1832
News CCTV1	98.4%	93.6%	802
News CCTV4	94.1%	93.3%	120
News Phoenix	97.8%	93.9%	557
Sport Oriental	91.2%	99.4%	353

Cut		96.3%	1324
Gradual transition		90.9%	508

Although the invention has been described with reference to video shot segmentation it should be appreciated that it is not limited to that application. The invention can be applied to any time series or sequential data.

5 It will be appreciated by persons skilled in the art that numerous variations and/or modifications may be made to the invention as shown in the specific embodiments without departing from the spirit or scope of the invention as broadly described. The present embodiments are, therefore, to be considered in all respects as illustrative and not restrictive.

CLAIMS:

1. Sequential data segmentation using spectral clustering wherein attributes of data samples are extracted to construct affinity matrices for sequences of samples in sequential order, and K-segmentation is then applied to a representation of the sequences of samples derived from the affinity matrices to identify K-segments, and
5 each K-segment is comprised of representations of samples in sequential order.
2. Sequential data segmentation using spectral clustering according to claim 1, wherein elements in a row or a column of the affinity matrices represent data samples in sequential order.
10
3. Sequential data segmentation using spectral clustering according to claim 1 or 2, wherein a graph Laplacian is derived from the affinity matrices and then the leading eigenvectors of the Laplacian are solved.
- 15 4. Sequential data segmentation using spectral clustering according to claim 3, wherein the graph Laplacian is normalized.
5. Sequential data segmentation using spectral clustering according to claim 2 or 3, wherein K-segmentation is applied to the normalised eigenvector space representing
20 the sequences of data samples.
6. Sequential data segmentation using spectral clustering according to any one of the preceding claims, wherein K-segmentation comprises identifying homogenous segments in the representation of sequences of data samples.
25
7. Sequential data segmentation using spectral clustering according to claim 6, wherein an estimation of the number of segments K is used to initially identify K homogenous segments.
- 30 8. Sequential data segmentation using spectral clustering according to any one of the preceding claims 1 to 7, wherein K-segmentation comprises identifying significant transitions in the representation of sequences of data samples.

9. Sequential data segmentation using spectral clustering according to any one of the preceding claims, wherein a data sequence is initially divided into sections of a specific number of samples.
- 5 10. Sequential data segmentation using spectral clustering according to claim 9, wherein sequential data segmentation is applied to each section.
11. Sequential data segmentation using spectral clustering according to claim 9 or 10, wherein for a section S that consists of n samples $\{s_1, \dots, s_n\}$, $ss(a, b)$ is defined as a
10 sequence of samples in sequential order such that $\{s_a, \dots, s_b\}$ where $a \leq b$.
12. Sequential data segmentation using spectral clustering according to claim 9, 10 or 11 and limited by claim 3, wherein spectral clustering comprises performing clustering based on the largest eigenvectors.
- 15 13. Sequential data segmentation using spectral clustering according to claim 12, wherein the largest eigenvectors are stacked into columns to form matrix X such that the renormalisation of the rows of X gives matrix Y .
- 20 14. Sequential data segmentation using spectral clustering according to claim 13, wherein section S is represented in the space spanned by the rows of Y by n frames $\{y_1, \dots, y_n\} \in R^k$.
- 25 15. Sequential data segmentation using spectral clustering according to claim 13 or 14, wherein K -segmentation can be conducted on Y to find a sequence $y_{s_1} y_{s_2} \dots y_{s_k}$ of k segments such that $y_{s_1} y_{s_2} \dots y_{s_k} = \{y_1, \dots, y_n\}$ and each y_{s_i} is not empty.
- 30 16. Sequential data segmentation using spectral clustering according to any one of the preceding claims, wherein the extracted attributes of data samples includes the color attributes of the samples.
17. Sequential data segmentation using spectral clustering according to any one of the preceding claims, wherein the extracted attributes of data samples includes the edge attributes of the samples.

18. Sequential data segmentation using spectral clustering according to claim 17, wherein the edge attributes of the samples is determined by analysing overlapping sub-samples and using edge detection templates.
- 5 19. Sequential data segmentation using spectral clustering according to any one of the preceding claims, wherein the extracted attribute of data sample is based on edge energy distance values between sequential couples of samples.
20. Sequential data segmentation using spectral clustering according to any one of
10 the preceding claims, wherein the extracted attribute of data sample is based on the sequential adjacency of the samples.
21. Sequential data segmentation using spectral clustering according to any one of
15 the preceding claims, wherein the extracted attributes of the data samples includes random impulse noise.
22. Sequential data segmentation using spectral clustering according to any one of the preceding claims, wherein before K-segmentation is applied, pre-processing is applied to identify potential boundaries and to reduce the computational complexity of
20 the K-segmentation of the sequences of data samples.
23. Sequential data segmentation using spectral clustering according to any one of the preceding claims, wherein following K-segmentation, the number of segments is determined by rejecting segments that do not meet a predetermined threshold.
25
24. Sequential data segmentation using spectral clustering according to claim 23, wherein segments are rejected if the normalized conductance of at least one of a segment's boundaries is less than the predetermined threshold.
- 30 25. Sequential data segmentation using spectral clustering according to claim 24, where the normalized conductance is calculated from all data in the segments.
26. Sequential data segmentation using spectral clustering according to claim 24, where the normalized conductance is calculated from selected data in the segments.
35

27. Sequential data segmentation using spectral clustering according to any one of the preceding claims, wherein a boundary of a segment is fine tuned by considering the sample at the identified boundary of a segment and a predetermined number of adjacent samples and selecting the most suitable sample to be the boundary.

5

28. Sequential data segmentation using spectral clustering according to any one of the preceding claims, wherein an incorrect boundary of a segment is detected and rejected by extracting key features of the boundary and assessing those features using Support Vector Machines classification.

10

29. Sequential data segmentation using spectral clustering according to claim 28, wherein the key features are extracted from the transition of the cut values over time.

30. Sequential data segmentation using spectral clustering according to any one of the preceding claims, wherein the data is comprised of samples that are video frames, and each K-segment is a video shot.

15

31. A computer system for performing sequential data segmentation using spectral clustering according to anyone of the preceding claims.

20

32. Computer software able to perform the sequential data segmentation using spectral clustering according to any one of the claims 1 to 30.

1/7

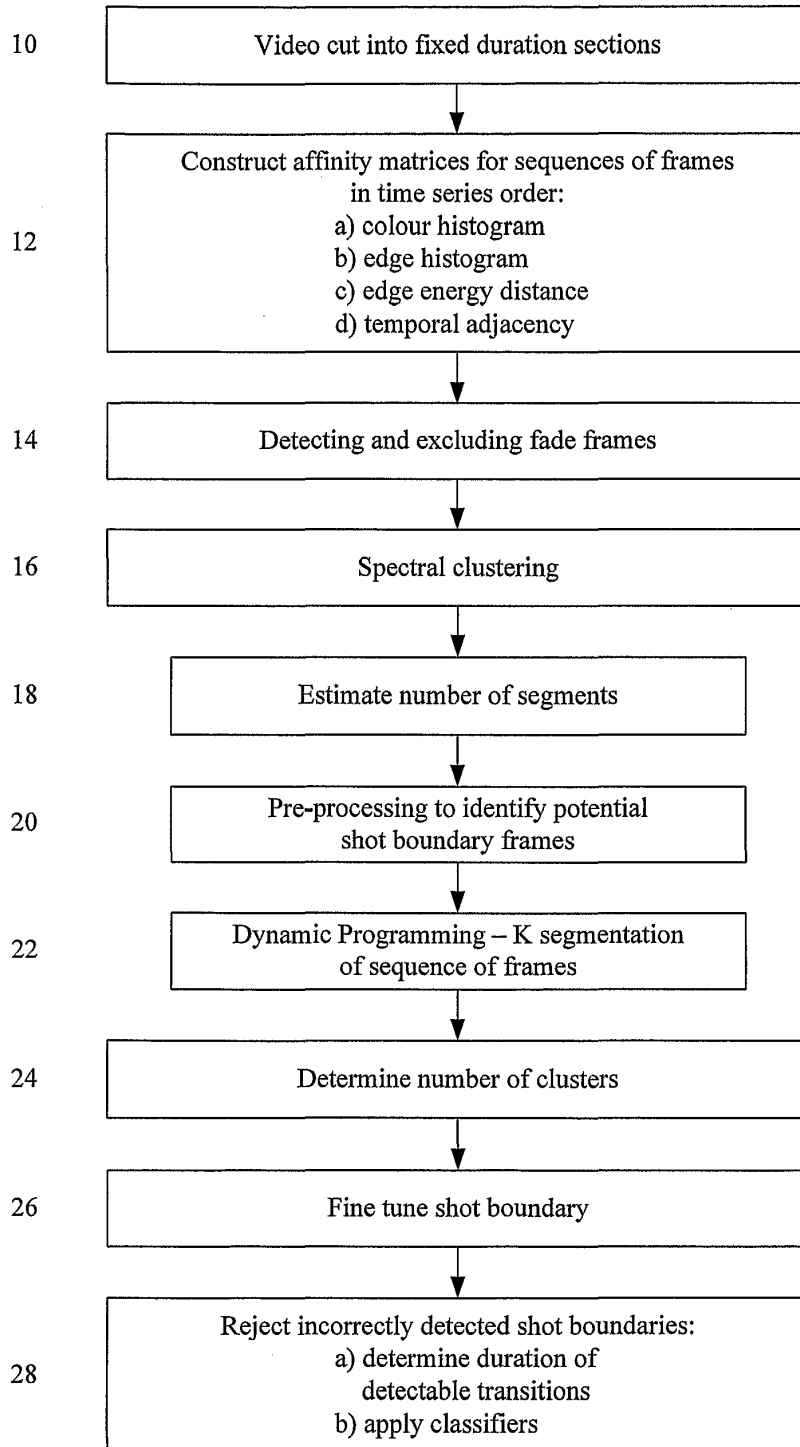


Fig. 1

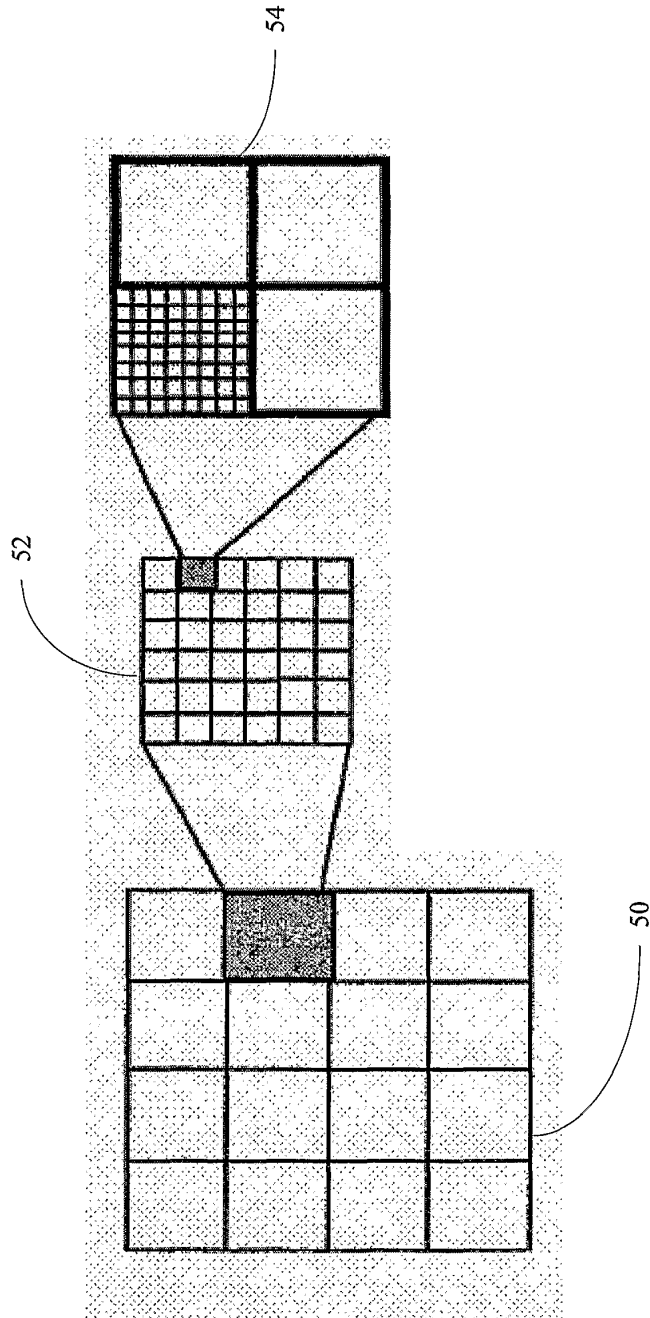
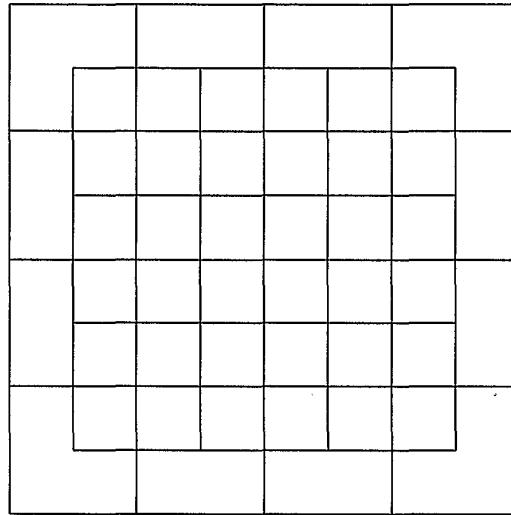
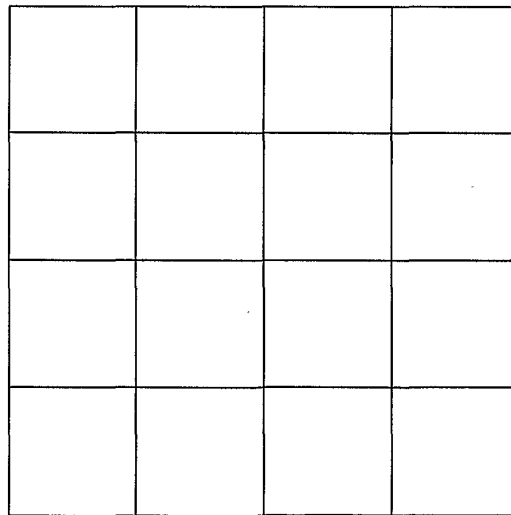


Fig. 2



(b)



(a)

Fig. 3

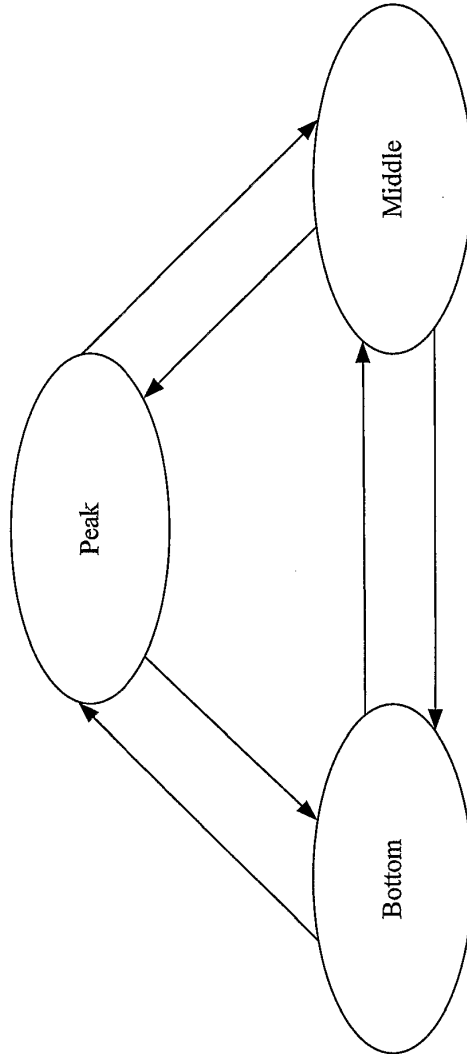
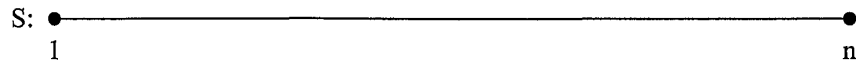


Fig. 4

Before segmentation



After segmentation

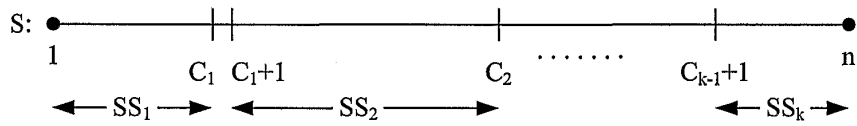


Fig. 5

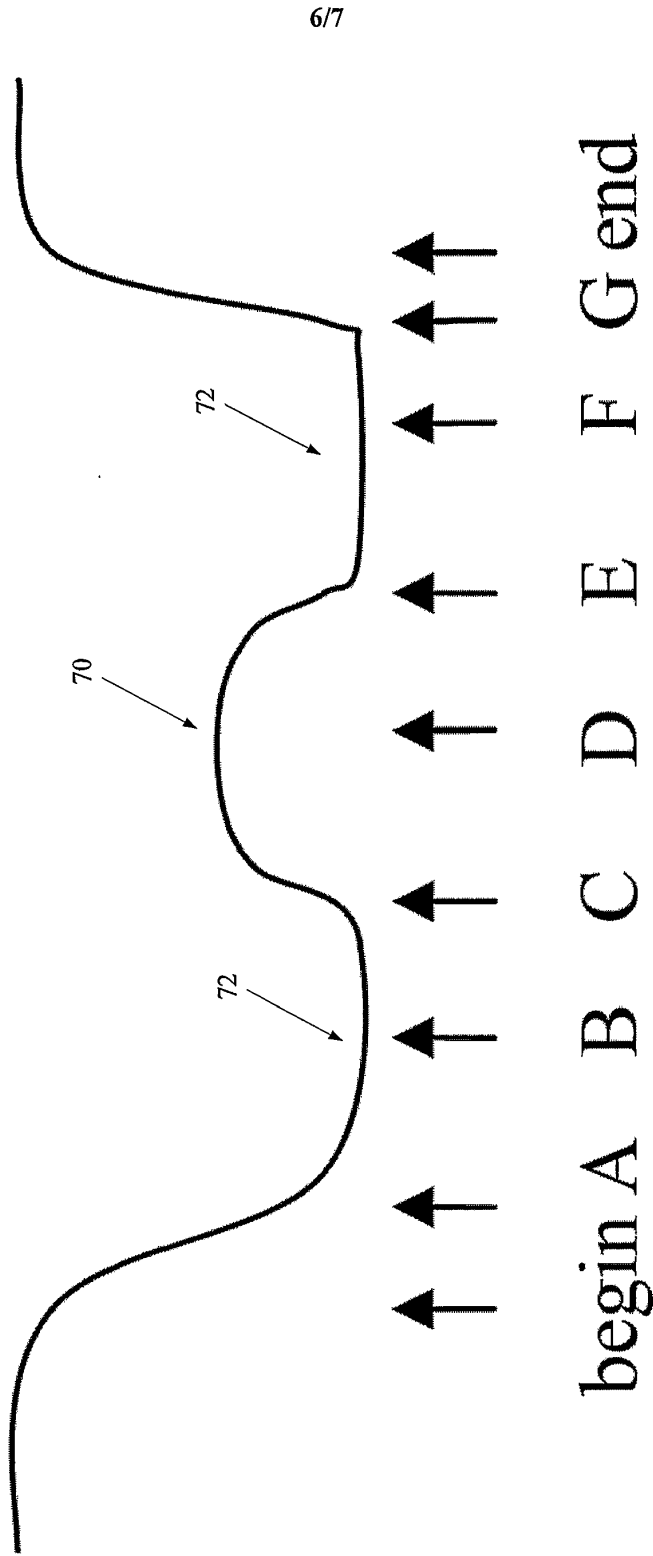


Fig. 6

7/7

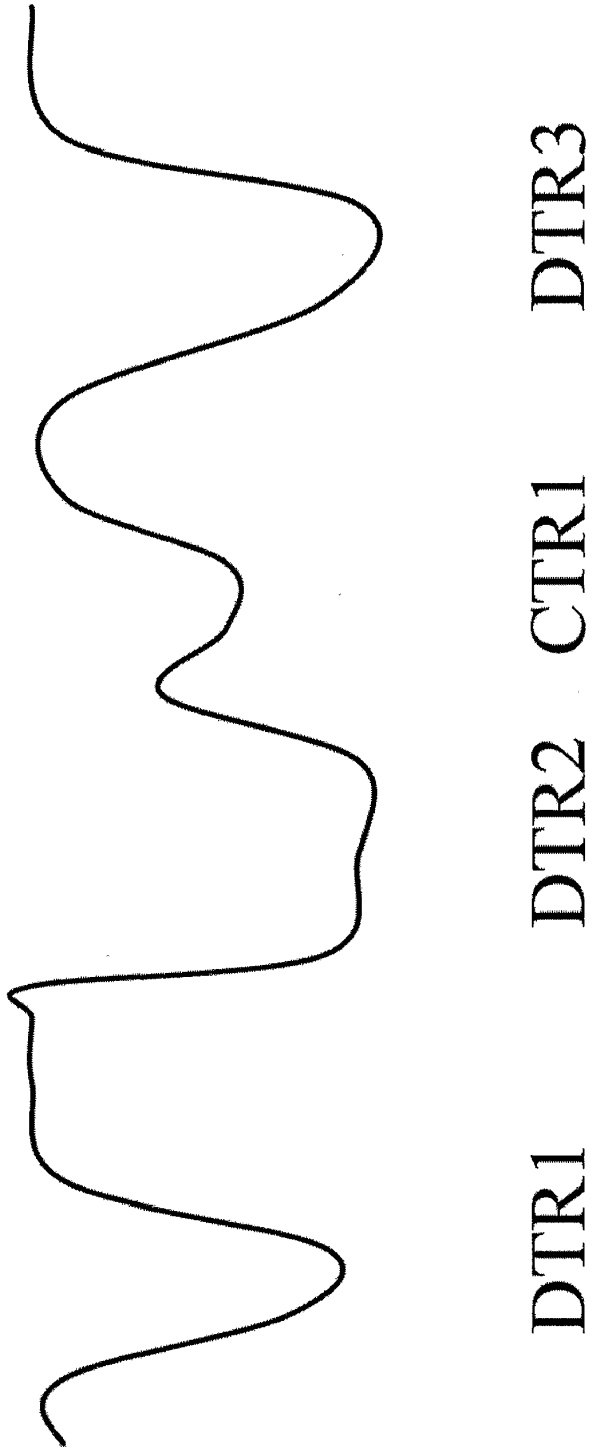


Fig. 7

INTERNATIONAL SEARCH REPORT

International application No.
PCT/AU2006/000012

A. CLASSIFICATION OF SUBJECT MATTER Int. Cl. G06K 9/00 (2006.01) G06F 17/30 (2006.01) According to International Patent Classification (IPC) or to both national classification and IPC					
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) INSPEC, IEEE, Google Scholar, DWPI, PCT, USPTO (spectral clustering, video, segmentation, etc.)					
C. DOCUMENTS CONSIDERED TO BE RELEVANT					
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.			
X	J.-M. Odobez et al, <i>Spectral Structuring of Home Videos</i> Proc. International Conference on Image and Video Retrieval, pp. 310-20, July 2003	1-32			
<input type="checkbox"/> Further documents are listed in the continuation of Box C <input type="checkbox"/> See patent family annex					
<table style="width: 100%; border: none;"> <tr> <td style="width: 33%; border: none;"> * Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed </td> <td style="width: 33%; border: none;"> "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family </td> <td style="width: 33%; border: none;"></td> </tr> </table>			* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family	
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family				
Date of the actual completion of the international search 18 January 2006	Date of mailing of the international search report 23 JAN 2006				
Name and mailing address of the ISA/AU AUSTRALIAN PATENT OFFICE PO BOX 200, WODEN ACT 2606, AUSTRALIA E-mail address: pct@ipaaustralia.gov.au Facsimile No. (02) 6285 3929		Authorized officer Matthew Hollingworth Telephone No : (02) 6283 2024			