Using control charts for on-line video summarisation

Clare E. Matthews[*], *Paria* Yousefi, and *Ludmila I.* Kuncheva

School of Computer Science, Bangor University, Bangor, Gwynedd, UK

Abstract. Many existing methods for video summarisation are not suitable for on-line applications, where computational and memory constraints mean that feature extraction and frame selection must be simple and efficient. Our proposed method uses RGB moments to represent frames, and a control-chart procedure to identify shots from which keyframes are then selected. The new method produces summaries of higher quality than two state-of-the-art on-line video summarisation methods identified as the best among nine such methods in our previous study. The summary quality is measured against an objective ideal for synthetic data sets, and compared to user-generated summaries of real videos.

1 Introduction

Lightweight, wearable devices allow consumers to capture a continuous stream of frames that provides a record of their daily activities [1]. Processing frames on-the-fly, to select a condensed set of frames that accurately represents the full content of the video, can greatly increase the duration over which such devices can operate. Methods for on-line video summarisation can be used for this process. As processing and memory resources are limited, the traditional high-level feature extraction from frames, e.g., through convolutional neural networks (CNN) [2], or methods requiring storage of all frames [3] may be infeasible. Similarly, elaborate summary selection methods may not be applicable on-line.

In our previous work [4], we proposed a taxonomy of on-line video summarisation methods. We described nine existing methods within the terms of the taxonomy, and compared them experimentally. These experiments highlighted the need for on-line methods to be robust to changes in parameter values. For example, parameters dependent on properties such as total video length are not suitable. The methods investigated are as follows: Shot boundary detection method (SBD) [5], Zero-mean normalised cross-correlation (ZNCC) [6], Diversity promotion (DIV) [2], Submodular convex optimisation (SCX) [7], Minimum sparse reconstruction (MSR) [8], Gaussian mixture model (GMM) [9], Histogram intersection (HIST) [10], Merged Gaussian mixture models (MGMM) [11], and Sufficient content change (SCC) [12]. We found that the SCX and MGMM methods consistently outperformed the others.

[*] Corresponding author: c.e.matthews@bangor.ac.uk

We use our previous findings to propose here a new on-line video summarisation method that meets the requirements of low computational complexity for feature extraction and summary selection, and with parameters that are relatively robust to different video type.

Despite the large number of video summarisation methods available, and the growing number of on-line methods, the evaluation, and therefore comparison, of methods remains a challenge. We compare our new method against the SCX and MGMM methods by running experiments on both synthetic and real data sets. For the synthetic data, an objectively "best" solution is available. For the real data, we choose a video data base where user-selected keyframe summaries are available, and can be used as ground truth.

The rest of the paper is organized as follows. Section 2 describes the classification system, and Section 3 introduces the new method. The experiments are presented in Section 4, and the conclusion, in Section 5.

2 Classification of methods

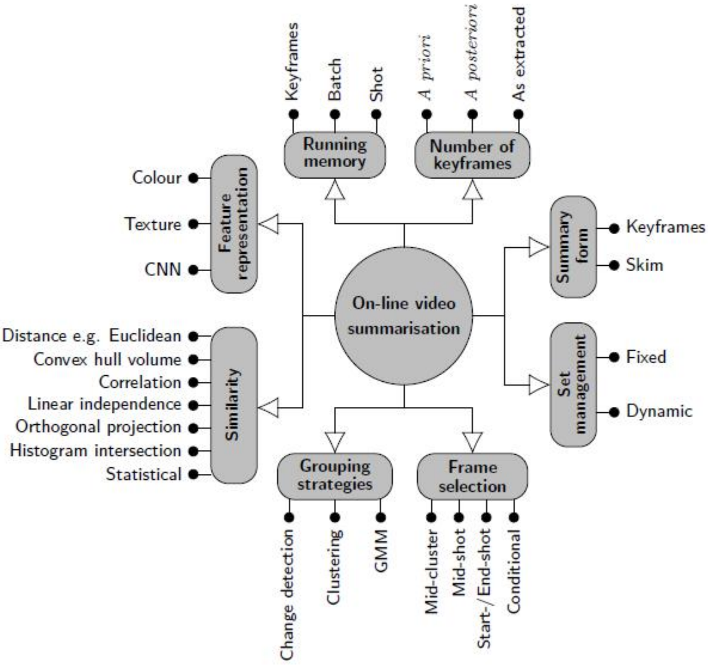

Fig. 1. A classification of on-line video summarisation methods.

The classification for on-line video summarisation methods proposed in [4] is reproduced in figure 1. This classification is adapted from that of Truong Venkatesh [12] for general video summarisation. Eight key components of an on-line summarisation method are identified:

- **Feature representation.** Each frame of the video is represented by an n-dimensional vector in some feature space, $x \in \Re^n$. Simple features may describe the colours within an image [5, 6, 9, 10], or the structure and texture [8]. Features derived from convolutional neural networks (CNN) [2, 7] are relatively complex.
- **Similarity.** A measure of similarity is used to compare the feature vectors of frames. Such measures include the distance between vectors in the feature space [7, 9], the volume of the convex hull of a set of frames [2], the degree of correlation

between frames [6], the intersection of feature vectors [10], the linear independence between batches of frames [5], and the orthogonal projection of a feature vector onto a set of existing features [8]. Statistical methods are also used, e.g. to test equivalence of sets of frames [11].

- **Grouping strategies.** Using the similarity measures described above, frames are grouped together. Groupings may be time-aware, e.g. time-contiguous shots identified using change-detection [5, 6, 10, 12], or independent of time, e.g. clustering of frames within the feature space [2, 7, 8, 13] or assigning frames to components of a Gaussian mixture model [9, 11].
- **Frame selection.** Different approaches exist for selecting a keyframe from a group of frames, e.g. choose the frame most central within a cluster [2, 7, 11], frames at the start / end [5, 12] or middle [6, 10] of a shot, or alternatively, frames that satisfy some condition, such as exceeding a ``difference" threshold [8, 10].
- **Set management.** The keyframe set may be fixed, i.e. once a keyframe is selected it cannot be removed from the set [5, 6, 7, 8, 9, 10, 12], or dynamic, i.e. a keyframe may be removed at some later point in the processing [2, 11].
- **Summary form.** Frames selected as a summary form either a static keyframe set [2, 5, 6, 7, 8, 10, 11, 12], or dynamic video skim [6, 9, 13, 3, 14].
- **Number of keyframes.** The number of keyframes in a summary is often variable and *as extracted*, determined by the algorithm and characteristics of the video [5, 6, 7, 8, 9, 10, 11, 12]. Alternatively, the number of frames can be defined *a priori* [2], or post-processing of the selected frames can reduce the set size *a posteriori*, to a pre-defined number of frames [2].
- **Running memory.** The memory required to run the summarisation is an important feature for on-line applications. Some methods need only keep the keyframe set in memory [2, 8, 9, 12], others process frames in batches and must therefore keep the full batch in memory [5, 7, 11]. Similarly, methods based on identifying shots within the video, may require a full shot to be stored in memory to then select the desired keyframe from the shot [6, 10].

3 Control-charts for on-line video summarisation

Here we propose a method that uses the statistical process of control-charts to identify shots from a streaming video. Control-charts [15] monitor a quantity of interest to detect when a process moves out of control. The mean, μ, of the quantity is used as a baseline value, and the process deemed to be “in control” while observations remain within a specified limit from the mean, typically three standard deviations, σ.

Algorithm 1: On-line control chart method

Input: Data stream $X = \{x_1,...,x_N\}$, $x_i \in \Re^L$, minimum shot length m, buffer size B, threshold for keyframe similarity θ.

Output: Selected set of keyframes $P \subset X$.

```
// Initialisation
$P \leftarrow \emptyset$
$j \leftarrow 1$ // Shot number
$S_j \leftarrow \{x_1,...,x_B\}$ // First shot
for $i \leftarrow [2,...,B]$ do
        $d_i \leftarrow d(x_i,x_{i-1})$ // Euclidean distance
```

```
    $\mu \leftarrow mean(d_2,\ldots,d_B)$
    $\sigma \leftarrow std(d_2,\ldots,d_B)$
// Process video frame-by-frame
    for $i \leftarrow [B+1,\ldots,N]$ do
        $d_i \leftarrow d(x_i, x_{i-1})$
        if $d_i < \mu + 3\sigma$ then
        // No new shot detected
            $[\mu,\sigma] \leftarrow$ update $\mu$ & $\sigma$ with $d_i$
            $S_j \leftarrow S_j \cup X(i)$
        else
        // New shot detected
            if $|S_j| > m$
            // Shot sufficiently long
                $p_j \leftarrow$ selectkeyframe$(S_j)$
                $\delta \leftarrow$ keyframediff$(p_j, p_{j-1})$
                if $\delta < \theta$ then
                    // Shots too similar: merge
                    $S_j \leftarrow S_{j-1} \cup S_j$
                    // Remove last keyframe from set
                    $P \leftarrow P(1{:}\mathrm{end}-1)$
                    $p_j \leftarrow$ selectkeyframe$(S_j)$
                $P \leftarrow P \cup p_j$
                $j \leftarrow j+1$
            else
            // Shot too short: ignore
                $S_j \leftarrow \emptyset$
____________________________________________________________
Function $f$ = selectkeyframe($Y$)
// Select the frame closest to the mean
    $f \leftarrow \underset{x \in Y}{\mathrm{argmin}}\, d(x, \bar{Y})$
____________________________________________________________
Function $\delta$ = keyframediff($f_1$,$f_2$)
// Compare 16-bin Hue histograms of frames $f_1$ and $f_2$
    $h_i$ = hist16(hue($f_i$)) // Normalised 16-bin Hue histogram
    $\delta = \sum_{j=1}^{16} |h_1(j) - h_2(j)|$
```

3.1 Control-chart method (CC)

Assuming that each frame is represented as a point in some L-dimensional space, we take the Euclidean distance, d, between consecutive frames as the process to be monitored. A distance $d > \mu + 3\sigma$ defines a shot boundary. Once a full shot has been identified, a keyframe is selected as the frame closest to the centre of the cluster defined by the shot.

Potential issues with such a method are that: (1) consecutive shots identified by the algorithm may be too similar to warrant separate keyframes, and (2) short transitions may be identified as shots, but are not important to the summary. We address these issues as follows:

- Define a measure of similarity between frames, as follows [16]. Use the HSV representation of the frames to obtain 16-bin histograms of the hue value (H). If the Minkowski distance between the normalised histograms is less than a threshold of 0.5, the frames are similar.
- After identifying a shot and selecting the representative keyframe, we compare this frame with the previous keyframe (if available). If the two consecutive keyframes are similar according to the above measure, we assume that a shot boundary has been falsely identified. The boundary is removed, and the two shots are merged. A new keyframe is selected from the combined shot to replace the two keyframes from the individual shots.
- We define an empirical constant to state the minimum shot length. If a shot contains fewer frames, the shot is ignored and no keyframe is selected.

The CC method requires three parameters: a pre-defined threshold θ for classifying keyframes as similar, a minimum shot length *m*, and initial buffer size *B* for calculating the starting mean and standard deviation. If we assume that the number of frames per second will be constant across videos, and that the duration required for a shot to be of interest is largely independent of video content, the optimal value for *m* should be consistent across videos. We select two seconds to be the minimum duration of a shot for it to be of interest. The full control-chart method is given in Algorithm 1.

3.2 Feature representation

Our control-chart method may be used with any feature space. For an on-line application, feature extraction must be a computationally inexpensive process. Therefore, relatively complex features, such as those derived from CNN, are not feasible.

To select a feature space for testing the algorithm, we implement the extraction of a number of different features, including those used by existing on-line summarisation methods. Table 1 shows the time taken to extract the different features for a video containing 3,266 frames. The extraction time for the RGB moments is substantially shorter than HSV histograms, even when a relatively small number of bins are used for the histograms. We therefore select RGB moments as the feature space to use in the CC method.

The RGB moments are a 54-dimensional feature space; the mean and standard deviation of the three colour channels for the nine sub-images created from a uniform 3-by-3 grid.

Table 1. Average time to extract features for the VSUMM video #21, and methods that use the features.

Feature	Time to extract (s)
RGB moments of 9 blocks (CC)	25
HSV histogram - [8, 4, 4] bins for Hue, Saturation, Value (ZNCC, SBD, HIST)	85
CENTRIST 252-dimensional structural histogram (MSR) [17]	522
MPEG-7 colour layout descriptor (GMM) [18]	1,546
Penultimate layer of VGG CNN (DIV) [19]	> 1.5hr

4 Method testing and evaluation

Here we compare the results for the proposed CC method with the two existing methods, SCX and MGMM, found to perform best in our previous comparison study [4].

4.1 Synthetic data

We first consider the performance of the three methods on seven synthetic data sets. The first data set follows the example of Elhamifar et al. [20]. The data consists of three clusters in 2-dimensional space as illustrated in figure 2 (#1). Each point represents a frame in the video. The three clusters come in succession but the points within each cluster are generated independently from a standard normal distribution. The order of the points in the stream is indicated by a line joining every pair of consecutive points. The time tag is represented as the grey intensity. Earlier points are plotted with a lighter shade. The “ideal” selected set is shown with red target markers. In addition to the two dimensions plotted, two noise dimensions are added (from the distribution $N(0, 0.5)$). Data sets #2 - #5 are also shown in figure 2. Again, each data set contains an additional two noise dimensions. Data sets #6 and #7 follow a similar structure but with more dimensions, six and eight, respectively.

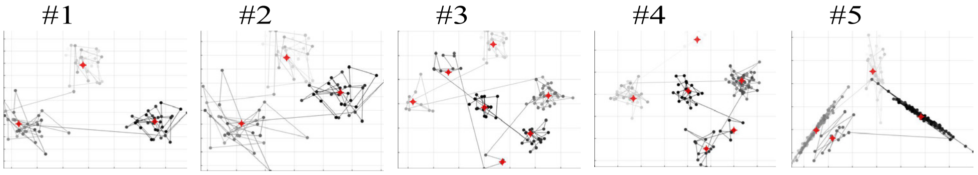

Fig. 2. Synthetic data sets #1 - #5. The time tag is represented as the grey intensity. Earlier points are plotted with a lighter shade. The “ideal” selected sets are shown with red target markers.

Using synthetic data allows an objective assessment of the summaries produced. If the video is already segmented into units (events, shots, scenes, etc.), the selected frames must allow for distinguishing between the units with the highest possible accuracy [21]. Therefore we use three complementary measures of the quality of the summary of synthetic data sets:

Cardinality : $$K = |P| \tag{1}$$

Approximation error : $$J = \sum_{i=1}^{N} d\left(x_i, p_i^*\right) \tag{2}$$

Accuracy : $$A = 1 - nn(P) \tag{3}$$

where $X = (x_1, \ldots, x_N)$ is the sequence of video frames, N is the total number of frames in the video, $P = \{p_1, \ldots, p_K\}$ is the selected set of keyframes, p_i^* is the keyframe closest to frame x_i, d is the Euclidean distance, and $1 - nn(P)$ is the resubstitution classification accuracy in classifying X using P as the reference set. To obtain a good summary, we strive to maximise A while minimising J and K.

We train the method parameters on 50 randomly generated data sets following the distribution of data set #1. Solutions are evaluated as follows:

- Find the Pareto set for the three criteria A, K and J.
- Exclude any results in the Pareto set with $K > 10$. This step removes the solution that selects *all* frames as keyframes, giving perfect accuracy and no error.

- Select the summary with the best accuracy. Where multiple summaries tie, select that with the fewest frames, and use the approximation error to split any remaining ties.

Taking the 50 optimal parameter sets as a cluster, the set closest to the cluster centre is chosen as the tuned method parameters.

The methods are then tested on 300 randomly generated data sets, 50 from each of the remaining six data set patterns, using the parameters tuned on data set #1. For each data set the accuracy, cardinality and approximation error are calculated for each method. The methods are then ranked. Four paired-sample t-tests are performed, comparing the accuracy and approximation error for our proposed CC method against the two existing methods.

Table 2. Results of paired-sample t-tests comparing the accuracy (A) and approximation error (J) for the CC method summaries and the summaries generated by the MGMM and SCX methods. The confidence interval for the difference is shown for significant results (at the 0.05 significance level).

Method	Test	P-value	Confidence interval
MGMM	$A_{\mathrm{CC}} - A_{\mathrm{MGMM}}$	$1e^{-5}$	[0.02, 0.04]
	$J_{\mathrm{CC}} - J_{\mathrm{MGMM}}$	$6e^{-4}$	[-1.7, -0.4]
SCX	$A_{\mathrm{CC}} - A_{\mathrm{SCX}}$	0.7	-
	$J_{\mathrm{CC}} - J_{\mathrm{SCX}}$	$3e^{-23}$	[-4.0, -2.7]

Table 2 shows the results of the paired-sample t-tests. At the 0.05 level, there is no significant difference between the accuracy values for the CC and SCX methods (i.e. the difference has a zero mean). All other tests find a significant difference. The confidence intervals for the mean differences are less than zero for J, implying that the error tends to be less for the CC method, and greater than zero for A, implying that the accuracy tends to be greater for the CC method. The CC method summaries tend to rank best according to our three criteria; an average of 1.4 across the 300 experiments, compared to the existing methods that have average ranks of 2.2 and 2.3 for the MGMM and SCX methods, respectively.

4.2 VSUMM videos

The methods are tested on 50 real videos from the VSUMM collection[1] [16]. Whereas the summaries of the synthetic data can be assessed in relation to a “correct” result, there is no such objective assessment available for real videos; what constitutes a good summary is somewhat subjective. The VSUMM collection includes a database of five user-selected summaries for each video. These summaries can be used as a ground-truth, to compare method-generated summaries against. Following the approach of De Avila et. al [16], the match between two summaries is described by an F-measure calculated using the 16-bin histograms of the hue values of selected keyframes, as explained in Section 3.1.

Parameters for each method are tuned on video #21. We select the parameters that produce the summary with the highest average F-measure when compared with the five user ground-truth summaries. These parameters are used to run the methods on the other 49 videos.

Figure 3 shows the F-measure (averaged across the five user summaries) versus the number of keyframes selected by each method for the VSUMM videos. Each point on the plot corresponds to a video. The ideal summary has a high F-measure, and low number of frames. Points in the upper-left corner of the plots shown in figure 3 therefore represent the better summaries. The points for all methods are plotted with grey colour on all plots. The

[1] https://sites.google.com/site/vsummsite

points of the method in the title of the subplot are shown with black markers. The CC method generates a higher proportion of good summaries than the existing two methods. As an illustration of these results, figure 4 shows the summary of video #47 produced by the CC method, compared to the summary from user #1. All five frames in the user summary are matched in the CC method summary.

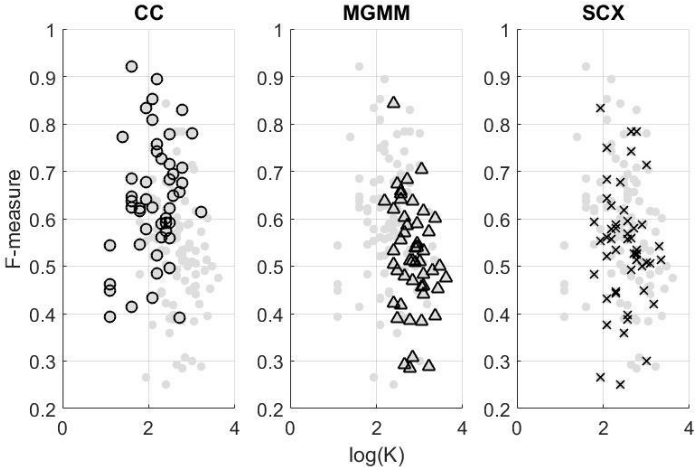

Fig. 3. Number of keyframes (K) and F-measure averaged over five user ground-truths, for summaries of the 50 VSUMM videos. Filled, grey circles show the results for all three methods, with the points for the named method highlighted in black.

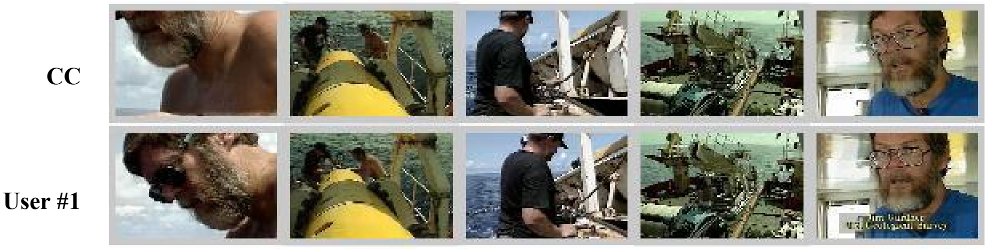

Fig. 4. Comparison of VSUMM video #47 summaries from user #1 and the CC method. The matches have been calculated using the 16-bin histogram method with threshold 0.5 [16].

5 Conclusion

Control-charts are a simple and effective approach to on-line video summarisation. Our proposed CC method performs well in comparison to existing methods, both on small synthetic data sets and real videos. On-line methods require computationally inexpensive feature spaces. The CC method uses RGB moments, which are significantly faster to extract than the features used by some of the existing on-line methods. Feature extraction time can be improved further by working with compressed images. However, further work is required to assess the effect on summary quality.

The videos used for testing have well-defined shots, providing a relatively easy summarisation task. The performance of the new method may be different on other types of video, e.g. where the shots are less clearly defined or the variability within shots is greater. Examples of such type of data are egocentric videos and lifelogging photo streams. Performance on longer videos must also be considered. For the application of wearable devices, it may be necessary to introduce a restriction on the number of keyframes that can be selected.

Similarly, when shots can potentially become very long, or consecutive shots very similar, a more dynamic approach to sampling, and the shot detection and similarity thresholds may be beneficial, and will be investigated in future work.

Acknowledgment

This work was supported by project RPG-2015-188 funded by The Leverhulme Trust, UK

References

1. Betancourt A, Morerio P, Regazzoni C S and Rauterberg M 2014 CoRR **abs/1409.1484v1** (*Preprint* arXiv:1409.1484v1) URL https://arxiv.org/abs/1409.1484v1
2. Anirudh R, Masroor A and Turaga P 2016 *IEEE International Conference on Image Processing (ICIP2016)* pp 3329–3333
3. Lan S, Panda R, Zhu Q and Roy-Chowdhury A K 2018 *IEEE/CVF Computer Vision and Pattern Recognition*
4. Matthews C E, Kuncheva L I and Yousefi P 2018 SUBMITTED: *Machine Vision and Applications*
5. Abd-Almageed W 2008 *IEEE 15th International Conference on Image Processing (ICIP 2008)* pp 3200–3203
6. Almeida J, Leite N J and Torres R d S 2013 *Journal of Visual Communication and Image Representation* **24** 729–738 URL http://dx.doi.org/10.1016/j.jvcir.2012.01.009
7. Elhamifar E and Kaluza M C D P 2017 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR2017)* pp 1818–1826
8. Mei S, Guan G, Wang Z, Wan S, He M and Feng D D 2015 *Pattern Recognition* **48** 522–533
9. Ou S H, Lee C H, Somayazulu V S, Chen Y K and Chien S Y 2015 *IEEE Journal of Selected Topics in Signal Processing* **9** 165–179
10. Rasheed Z and Shah M 2003 *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition* **2** pp 343–343
11. Song M and Wang H 2005 SPIE 5803, *Intelligent Computing: Theory and Applications III* **5803** pp 174–184
12. Truong B T and Venkatesh S 2007 *ACM transactions on multimedia computing, communications, and applications (TOMM)* **3** 3 URL https://dl.acm.org/citation.cfm?doid=1198302.1198305
13. Zhao B and Xing E P 2014 *IEEE Conference on Computer Vision and Pattern Recognition*
14. Sharghi A, Gong B and Shah M 2016 *European Conference on Computer Vision*
15. Shewhart W A 1931 Economic control of quality of manufactured product (Van Nostrand Company)
16. de Avila S E F, Lopes A P B, da Luz Jr A and de Albuquerque Araújo A 2011 *Pattern Recognition Letters* **32** 56–68

17. Wu J and Rehg J M 2011 *IEEE Transactions on Pattern Analysis and Machine Intelligence* **33** 1489–1501
18. Kasutani E and Yamada A 2001 *IEEE International Conference on Image Processing* pp 674–677
19. Simonyan K and Zisserman A 2014 Very deep convolutional networks for large-scale image recognition arXiv:1409.1556
20. Elhamifar E, Sapiro G and Sastry S S *2016 IEEE transactions on pattern analysis and machine intelligence* **38** 2182–2197
21. Kuncheva L I, Yousefi P and Almeida J 2018 *Journal of Visual Communication and Image Representation* **52** 118–130 URL https://doi.org/10.1016/j.jvcir.2018.02.010