On the Evaluation of Video Keyframe Summaries using User Ground Truth

Ludmila I Kuncheva*, Iain A D Gunn†, Paria Yousefi‡

*School of Computer Science, Bangor University, Dean Street, Bangor, Gwynedd, Wales LL57 1UT, UK
†Present address: Department of Computer Science, Middlesex University, The Burroughs, London NW4 4BT
‡Email addresses: l.i.kuncheva@bangor.ac.uk (Ludmila I Kuncheva), i.gunn@mdx.ac.uk (Iain A D Gunn), paria.yousefi@bangor.ac.uk (Paria Yousefi)

Abstract

Given the great interest in creating keyframe summaries from video, it is surprising how little has been done to formalise their evaluation and comparison. User studies are often carried out to demonstrate that a proposed method generates a more appealing summary than one or two rival methods. But larger comparison studies cannot feasibly use such user surveys. Here we propose a discrimination capacity measure as a formal way to quantify the improvement over the uniform baseline, assuming that one or more ground truth summaries are available. Using the VSUMM video collection, we examine 10 video feature types, including CNN and SURF, and 6 methods for matching frames from two summaries. Our results indicate that a simple frame representation through hue histograms suffices for the purposes of comparing keyframe summaries. We subsequently propose a formal protocol for comparing summaries when ground truth is available.

Keywords: video summarization / summarisation, keyframe selection, evaluation of summaries, F-measure, discrimination capacity.

1. Introduction

Keyframe selection is aimed at summarising video data [43]. The summary should be compact, but also diverse and informative for the viewer.

While the literature abounds with methods for keyframe selection, surprisingly little has been done towards developing a formal evaluation protocol. The need for such a protocol is widely acknowledged [45, 10, 12, 21, 35, 27, 35, 34]. However, at present authors often develop a bespoke experimental set-up in which their proposed method for keyframe selection compares favourably to just one or two alternative methods. (This is beginning to change, as recently, keyframe summaries obtained through different methods have been collated in a publicly available benchmark repository [8].) The measures of quality of the keyframe summaries are typically not commensurable across different studies. A particular problem with current evaluation techniques is the lack of comparison to baseline methods. User studies usually demonstrate some percentage improvement achieved by the proposed method against another method, with respect to a chosen criterion such as informativeness or enjoyability. The percentage improvement varies considerably from one study to the next, and it is difficult to assign meaning, let alone statistical significance, to these percentages if different quality measures are used each time. This raises the question of whether the degree of improvement in the summary justifies the effort involved in the design of the new summarisation method.

The uncertainty is amplified by the lack of large-scale comparisons between keyframe selection methods over large video repositories. The major obstacle in such an endeavour has been the fact that the evaluation of a keyframe summary requires human input at some level, and user studies are expensive. This difficulty can be addressed by obtaining human-made summaries, which we shall call ground truth summaries. That is, we ask humans to perform the task which the summary algorithms aim to automate, as opposed to asking them to evaluate the automatic summaries directly. Future automatic keyframe summaries can then be evaluated by matching against the existing ground truth collection, rather than requiring a fresh user survey.

Suppose that \( \gamma(A, B(\theta)) \) is a measure of how close keyframe summaries \( A \) and \( B \) are, and \( \theta \) is a set of parameters of \( \gamma \). High values of \( \gamma \) are desirable if one of the summaries is seeking to approximate the other. Let \( K \) be the evaluated summary, \( GT \) be a ground truth summary, and \( U \) be a uniform summary (i.e., a set of frames selected from the video at a constant interval) of the video of interest. Our idea relies on the premise that summaries obtained from purposely designed methods are closer to the user preferences (larger \( \gamma(K, GT(\theta)) \)) than the (context-blind) uniform keyframe selection is.

We are interested in proposing a good \( \gamma \). Alongside proposing a form for \( \gamma \), we will seek a parameter set \( \theta \) which tends to maximise \( \gamma(K, GT(\theta)) \) over a suitable selection of ground truths \( GT \) and automatic summaries \( K \).

In this paper we propose a generic protocol for comparing keyframe summaries with a set of ground truth summaries. The function \( \gamma \) we seek will compare the number of successful pairings between elements of the keyframe set under evaluation,
and elements of the ground truth. There are four key questions to be answered about this function. The first three questions will determine the elements of the parameter set $\Theta$:

1. **Features.** What features should be used to describe the keyframes?
2. **Metric.** What metric should be used to give distance between a pair of frames in the feature space?
3. **Matching.** How are the frames paired between the two summaries?

The fourth question pertains to the form of $\gamma$ itself:

4. **Similarity.** Given a number of pairings between two keyframes sets, and the sizes of the two sets, what value do we assign to the similarity $\gamma$ of the sets?

To address point 1), we examine the most widely used sets of features for representing keyframes. Typically, these features are colour-based (e.g., histograms of the hue value), summarising colour values for either the whole image or a grid-like split of the image into 2-by-2 and 3-by-3 subimages. We include in the comparison the RGB and HSV spaces, and other standard, though less popular, colour space representations. We also take the last fully-connected layer of the ConvNet (Convolutional Network) pretrained model as a fixed feature extractor for our data, as well as SURF features.

For point 2), we consider the Euclidean and Manhattan distances for the feature spaces apart from SURF. For the SURF representation, we apply the associated method for matching relevant points between two images.

For point 3), we describe and evaluate six approaches taken from the literature on keyframe evaluation. We believe that this is the first study which summarises and evaluates together these approaches.

Finally, (for point 4) we propose the F-measure as $\gamma$ because of its symmetry, limits, and interpretability.

To determine the empirical answers of questions 1-3, we carry out an experimental study on the 50 videos from the VSU MM project together with the automatic keyframe summaries and user ground-truth summaries provided.

The rest of the paper is organised as follows. Section 2 gives a broad overview of existing evaluation practices, including those unrelated to ground-truth methods. Sections 3 and 4 describe related work, and our experimental choices, in regard to questions 1) and 2) above, respectively. Section 5 gives the equivalent discussion of existing work and experimental choices for questions 3) and 4). Sections 6 and 7 give the proposed protocol and its experimental evaluation using the VSU MM video repository, as well as a discussion on the findings. Section 8 outlines our conclusions and further research directions.

---

2. Evaluation approaches

A diagram summarising the most used evaluation approaches is shown in Figure 1.

Category 1, “Descriptive” evaluation, pertains to some of the earliest publications in this area, which made no formal evaluation of their keyframe summaries, but simply displayed example outputs and argued for their plausibility (e.g., [42], [47], [51]). The field has since developed, giving rise to a great number of alternative summarisation algorithms as well as tools for quantitatively comparing their outputs. In the light of these developments, this “descriptive” (non-)evaluation must be considered an obsolete practice.

All forms of quantitative evaluation which involve no user input are grouped together under category 2. Subcategory 2A includes those evaluation approaches, such as the Shot Reconstruction Degree [29], which evaluate the keyframe selection by interpolating a reconstruction of the original video from it, and comparing this reconstruction to the original video. To evaluate a keyframe selection in this way is to evaluate it, essentially, as a form of compression. However, there is no guarantee that the frame set which gives rise to the best reconstruction is the one which a human user would choose. Indeed, this is the very reason that summarisation is considered as a distinct task from compression.

Subcategory 2B includes the Fidelity measure of Chang et al. [5]. This involves using a semi-Hausdorff measure to calculate a distance from the keyframe set to the set of frames of the original video. The Hausdorff and semi-Hausdorff distances are generic means of calculating a degree of similarity or distance between two sets. They are mathematically convenient and familiar from their applications in other fields, but may be inappropriate for the problem of evaluating keyframe sets, due to their strong sensitivity to the distance of the worst-case (most distant) element.

Subcategory 2C characterises methods which use a high-level feature regarded as measuring “quality” to gauge the keyframe summary. Often this feature is used itself to select the keyframes, so there is no active comparison process with the whole video. Examples from this category include snapshot detection [50] where the main objective of the keyframe selection is to be like a well-composed photograph taken by a human.

All forms of evaluation which involve user input in one way or another are grouped together under category 3. Subcategory 3A contains those methods in which users are asked to assess the automatic keyframe summary. Subcategories 3B and 3C contain those methods in which users generate a summary of their own, which is then used as a ground-truth against which the output of the algorithm is automatically compared.

Following Truong and Venkatesh [43], we distinguish between a “direct” ground truth, in which the user makes a keyframe summary, and an “indirect” ground truth, in which the user summarises the semantic content which the output keyframe summary should cover. The former subcategory, 3C, includes the keyframe-matching method developed by de Avila et al. [8], which has gained some popularity (e.g., [11], [13]).
1. Descriptive: Visual inspection
2. Automatic
   A. No ground truth
   B. Indirect (semantic) ground truth:
      a. Criterion-based: Single keyframe summary
      b. Comparing two keyframe summaries
      c. Ranking multiple keyframe summaries
3. User-based
   EVALUATION of keyframe summaries
   a. Criterion-based: Direct ground truth:
      Keyframe summaries obtained from users
   b. Comparing two keyframe summaries
   c. Ranking multiple keyframe summaries

Figure 1: Evaluation approaches for keyframe video summaries. The box indicates the approach giving rise to the protocol proposed here.

3. Features

The feature spaces used for evaluating keyframe summaries are typically quite different from the feature spaces used by the various algorithms for selecting the keyframes for their summaries. Selection methods often rely on sophisticated and context-involved features such as the presence of people, objects \[30\] \[6\] \[23\] \[22\], landscapes, motion \[48\] \[11\] \[45\], or famous landmarks \[16\], and/or the use of a visual thesaurus \[41\]. High-level descriptors gauging the quality of the video frames have also been proposed, for example, “aesthetics”, “attention”, “saliency” and “interestingness” \[16\].

For judging the similarity between a user keyframe collection and a candidate keyframe collection, however, low-level, context-blind features are usually applied. Feature spaces of this type include colour histograms, and edge and texture features \[9\] \[48\] \[7\] \[47\] \[38\] \[42\] \[8\] \[37\] \[26\].

In this study we look for a suitable feature representation of the keyframes among the alternatives listed below. Most of the feature sets are defined by splitting the image into 3-by-3 equal-sized subimages before extracting features from each sub-image: this is the meaning of the _9blocks_ suffix.

1. **RGB_9blocks.** The mean and standard deviations of the red (R), green (G) and blue (B) channels for each sub-image. (6 features per sub-image)
2. **HSV_9blocks.** The mean and standard deviations of HSV (6 features per sub-image) \[8\] \[53\].
3. **CHR_9blocks.** The mean and standard deviations of Chrominance components \(C_1\) and \(C_2\) (4 features per sub-image) calculated as \[47\]:
   \[
   C_1 = \frac{R}{q}, \quad C_2 = \frac{G - B}{q}, \quad q = \sqrt{R^2 + G^2 + B^2},
   \]
4. **OHT_9blocks.** The mean and standard deviations of features \(I_1, I_2'\) and \(I_3'\) of Ohta space (6 features per sub-image) calculated as \[37\]:
   \[
   I_1 = \frac{1}{3}(R + G + B), \quad I_2' = R - B, \quad I_3' = \frac{1}{2}(2G - R - B)
   \]

The following hue histogram feature spaces (H-histograms) were also investigated \[15\] \[29\] \[48\] \[8\] \[17\] \[53\] \[44\] \[12\] \[52\]:
5. **H8_9blocks.** A histogram of the hue (H) values of the HSV space with 8 bins (8 features) for each of the sub-images of a split of the image into a 3-by-3 grid.
6. **H16_1block.** H-histogram with 16 bins for the whole image.
7. **H16_4blocks.** H-histogram with 16 bins for a 2-by-2 split of the image in sub-images.
8. **H16_9blocks.** H-histogram with 16 bins for a 3-by-3 split of the image in sub-images.
9. **H32_1block.** H-histogram with 32 bins for the whole image.
The values of each histogram were scaled so that the sum was one.

Next we considered:

10. CNN. The fully connected layer of a pre-trained CNN was used as a 4096-dimensional feature space [40].

11. SURF. SURF features were extracted which could match relevant points between two images [2] [39] [19].

4. Similarity between two keyframes

Similarity between two images (keyframes) can be calculated in many ways. For example, one could evaluate the proportion of matched SIFT keypoints [27], or similarities between visual word histograms [24] [41]. However, more general and efficient similarity measures can be used if the images are represented as points in an n-dimensional feature space. This will be our approach with the first 10 of the feature sets specified above. The SURF approach will be our example of an alternative approach that does not attempt to embed the keyframes in \[ \mathbb{R}^p \].

Here we treat the collections of keyframes as unordered. We use the Manhattan distance [5] and the Euclidean distance on each of the feature spaces 1–10.

For the SURF features, we use the following procedure: 1. Identify the keypoints in frame 1 (total number \( n_1 \)), and find how many have been matched in frame 2 (say, \( k_1 \)). 2. Identify the keypoints in frame 2 (\( n_2 \)) and the number of matched keypoints in frame 1 (\( k_2 \)). Calculate the similarity between the two images as the proportion \( \frac{2\min(k_1, k_2)}{n_1 + n_2} \). For the sake of consistency, we will use instead a distance between two frames \( f_a \) and \( f_b \) calculated as

\[
 d(f_a, f_b) = 1 - \frac{2\min(k_1, k_2)}{n_1 + n_2}.
\]

5. Similarity between two sets of keyframes

To evaluate an automatic keyframe summary, we can compare its match to a ground-truth summary by counting the number of paired frames and taking into consideration the total number of frames in each summary [8] [11] [13] [32].

We assume that a distance measure \( d \) between two frames has been already chosen, as discussed in section 3. Two frames \( f_a \) and \( f_b \) are sufficiently similar to be called a match if \( d(f_a, f_b) < \theta \), where \( \theta \) is a chosen threshold.

Let \( K_1 \) and \( K_2 \) be two sets of keyframes. We are interested in a measure of closeness between the two sets, \( M(K_1, K_2) \). The following two questions must be answered: How do we count the number of matches \( m \) between \( K_1 \) and \( K_2 \)? Once \( m \) has been found, how do we use it to calculate \( \gamma \)? (These are questions 3) and 4) of the introduction.)

5.1. Finding the number of matches

Denote the cardinalities of the two summaries by \( N_1 = |K_1| \) and \( N_2 = |K_2| \). Construct a distance matrix \( D_{N_1 \times N_2} \) where entry \( d_{i,j} \) is the distance between frames \( i \in K_1 \) and \( j \in K_2 \). Denote the number of matches returned by \( m \). Apart from the Mahmoud algorithm below, all algorithms take as input \( D \) and \( \theta \), and return \( m \).

Here we examine six pairing (matching) algorithms:

1. Naïve Matching (no elimination). This algorithm is surprisingly popular [31] although it has an obvious flaw. If the candidate summary \( K_1 \) consists of nearly identical frames which happen to be close to one frame from the ground truth summary \( K_2 \), then the number of matches will be perfect, \( m = N_1 \), for an arbitrary \( N_1 \). Such a candidate summary, however, will be quite inadequate: it is neither concise nor representative. Algorithm 1 relies on the presumption that \( K_1 \) is a reasonable summary containing diverse frames.

2. Greedy Matching. This algorithm is widely used but is quite conservative.

3. Hungarian Matching. [21] The Hungarian algorithm will identify \( \min[N_1, N_2] \) pairs such that the sum of the distances of the paired frames is minimum. A thresholded matching can be naïvely formed from this minimal complete matching by simply removing all pairings over the threshold distance \( \theta \). Thus, close matches could be missed in an attempt to minimise the total distance.

4. Mahmoud algorithm. [31] For this algorithm, the frames are arranged in temporal order and the matches are checked and eliminated accordingly. Apart from the temporal ordering, the algorithm is identical to the Greedy Matching.

5. Kannappan algorithm. [20] An interesting alternative approach to the matching problem is put forward by Kannappan et al. [20]. In their approach, a keyframe from the candidate set and a keyframe from the ground truth are matched only if each is the other’s best possible match: Algorithm 5. In their implementation, the set of matched pairs is subsequently thresholded using a different concept of pairwise frame distance from that used to form the matches. We have modified this procedure to make it the equivalent of the de Avila et al. thresholding, by using the same distance metric for thresholding as for finding the pairings.

6. Maximal Matching. The greatest possible value of \( m \) is given by a maximal unweighted matching in which frames less than distance \( \theta \) apart can be paired. Such a matching is given by the Hopcroft-Karp algorithm [49]. We will use instead the convenient alternative Algorithm 5 in which we find the lowest-weight complete matching on a binary matrix \( D' \) obtained by thresholding \( D \). Entry \( d'_{i,j} \) in \( D' \) has value 0 if \( d_{i,j} < \theta \), and 1 otherwise. After the optimal assignment is found through the Hungarian algorithm, the number of matches is determined by counting how many of the matched pairs are at distance less than \( \theta \).
Algorithm 1: Naïve Matching
1  \( m \leftarrow 0. \)
2  \( \text{for } i = 1, \ldots, N_1 \text{ do} \)
3  \[ \text{If any } d_{ij} < \theta, j = 1, \ldots, N_2, \text{ increment the number of matches, } m \leftarrow m + 1. \]

Algorithm 2: Greedy Matching
1  \( m \leftarrow 0. \) Find the smallest distance \( d_{\text{min}} = \min D. \)
2  \( \text{while } d_{\text{min}} < \theta \text{ do} \)
3  \[ \text{Increment the number of matches, } m \leftarrow m + 1. \]
4  \[ \text{Remove the row and the column of the matched elements from } D. \]
5  \[ \text{Find the smallest distance from the remaining matrix } d_{\text{min}} = \min D. \]

Algorithm 3: Hungarian Matching
1  \( \text{Apply the Hungarian assignment algorithm to } D. \)
2  \( \text{Identify the matched pairs of frames } (i, j), \text{ and retrieve the distances } d_{ij} \text{ from } D. \)
3  \( \text{Assign to } m \text{ the number of these distances which are smaller than } \theta. \)

Algorithm 4: Algorithm of Mahmoud [31]

Input: Keyframe summaries \( K_1 \) and \( K_2 \) arranged in temporal order, and threshold \( \theta. \)

Output: Number of matches \( m. \)
1  \( m \leftarrow 0. \)
2  \( \text{for } i \in K_1 \text{ do} \)
3  \[ \text{for } j \in K_2 \text{ do} \]
4  \[ \text{if } d_{ij} < \theta, \text{ then} \]
5  \[ \text{Increment the number of pairings, } m \leftarrow m + 1. \]
6  \[ \text{Remove } i \text{ from } K_1 \text{ and } j \text{ from } K_2. \]
7  \[ \text{Break.} \]

Algorithm 5: Algorithm of Kannappan et al. [20]
1  \( \text{Initialise a set of pairings } M \leftarrow \emptyset. \)
2  \( \text{for each frame } i \in K_1 \text{ do} \)
3  \[ \text{for each frame } j \in K_2 \text{ do} \]
4  \[ \text{if } j' = \arg \min_{k \in K_1} d(i, k) \text{ and } \]
5  \[ \quad i' = \arg \min_{k \in K_1} d(k, j) \text{ then} \]
6  \[ \quad \text{Add the pair to the matching set } M \leftarrow M \cup \{(i', j')\}. \]
7  \( m \leftarrow |M|. \)

Algorithm 6: Maximal matching algorithm
1  \( \text{Construct matrix } D' \text{ of the same size as } D \text{ such that } d'_{ij} = 0 \)
2  \( \text{iff } d_{ij} < \theta, \text{ and } d'_{ij} = 1, \text{ otherwise.} \)
3  \( \text{Apply the Hungarian assignment algorithm to } D'. \)
4  \( \text{Identify the matched pairs of frames } (i, j), \text{ and retrieve the distances } d_{ij} \text{ from } D. \)
5  \( \text{Assign to } m \text{ the number of these distances which are smaller than } \theta. \)

A common drawback of these algorithms is the lack of guidance in choosing the threshold value \( \theta. \) This value has an immediate impact on the number of matches, and subsequently on the value of the measure \( \gamma(K_1, K_2). \) Different values may be appropriate for different feature spaces and metrics. While the L1 distance between distributions, such as elements of histogram feature spaces, is bounded between 0 and 2, the same is not true for other feature spaces. For histogram spaces, \( \theta = 0.5 \) has been empirically found useful [8] but it is not clear what theoretical meaning this value might have. Setting an interpretable threshold in other feature spaces is even less intuitive.

For our experiments, we will use a range of thresholds from 0.01 up to 0.7 for the Manhattan metric. For the Euclidean metric, we will scale the threshold relative to the distribution of all pairwise distances between frames in the video. The thresholds will be percentiles of this distribution, from the 0.01th up to the 3rd percentile. For the SURF metric, we will vary the threshold between 0.01 and 0.4.

5.2. Calculating the similarity between keyframe summaries using the number of matches

Interpreting the number of pairings \( m \) returned by their Greedy Matching algorithm, de Avila et al. [8] use a pair of measures called respectively “Accuracy rate” (\( \text{CUS}_A \)) and “Error rate” (\( \text{CUS}_E \)), both designed to express how well \( K_1 \) (candidate summary) matches \( K_2 \) (ground truth), but not the other way around:

\[
\text{CUS}_A = \frac{m}{|K_2|},
\]

where \(|\zeta|\) denotes the cardinality of set \( \zeta \), and

\[
\text{CUS}_E = \frac{|K_1| - m}{|K_2|}.
\]

The problem with these measures is that the upper limit of \( \text{CUS}_E \) depends on \(|K_1|\).

Alternatively, given a number of matches \( m \), the similarity between \( K_1 \) and \( K_2 \) can be quantified using the F-measure, whose advantage is that it is symmetric on its two arguments [13]. Without loss of generality, choose \( K_1 \) for calcu-
We have chosen to use this rating the Recall, and which we call “discrimination capacity” as the dissimilarity measure in Section 5.2.

For the sake of generality, our protocol is bound by minimal imports [18, 13], we decided not to include a fusing procedure, in order to maintain simplicity and transferability. Such a procedure could be designed in many different ways, and there is little to guide the choice. We opt for calculating the overall assessment of $K$ as the average of the measures of interest between $K$ and the ground truth summaries. For example, let $G = \{K_1, \ldots, K_N\}$ be a collection of ground truth summaries obtained from $L$ users. Let $U(k)$ be a uniform summary with $k$ keyframes. We calculate $C_U$, the average of $C_U$ for $K$ and $G$, as

$$C_U = \frac{1}{L} \sum_{l=1}^{L} c_U(K, U, K_l),$$

This value measures how much better $K$ is, compared to a uniform keyframe summary of the same cardinality, in matching the users’ views. Note that, for a given $K$ and $G$, $C_U$ depends on the choices we make for the parameters in $\Theta$: features, metric, pairing algorithm and threshold. Therefore we will be looking for a set of parameters which maximises $C_U$ across a range of videos and summarisation algorithms for obtaining $K$.

6. Proposed evaluation protocol

We have reviewed the approaches and methods to answer the four questions in the Introduction: (1) Features in Section 3; (2) Metric in Section 4; (3) Matching in Section 5.1 and (4) Similarity Measure in Section 5.2.

The foundational idea for our experiments is that a good measure for similarity between keyframe summaries should distinguish as clearly as possible between content-blind baseline methods such uniform summaries on the one hand, and a sophisticated algorithmic summary, on the other hand. To estimate how well a measure distinguishes between baseline designs and bespoke selection methods, we propose the quantity which we call “discrimination capacity” as the difference:

$$c_U \equiv c_U(K, U, GT) = \gamma(K, GT) - \gamma(U, GT),$$  

where $GT$ is a ground truth summary, $K$ is a keyframe summary obtained by an algorithmic method, and $U$ is a baseline summary, which in our case will be the Uniform summary of the same cardinality as $K$. Large values of $c_U$ will signify good choices of parameters $\Theta$: features, metrics, algorithms, and thresholds which could be recommended for the practical implementation of the proposed protocol as a tool for the evaluation of future algorithms.

For the sake of generality, our protocol is bound by minimal restrictions:

1. We are not concerned with how the keyframes in $K$ are obtained. For example, the video could be split into shots or used in its entirety; low-level visual features or high-level semantic features could be used, etc.

2. Both $K$ and the ground truth summaries are sets of keyframes. This means that the frames are not ranked by importance, nor are they arranged in a temporal order.

Weighing the arguments for and against fusing a possible set of available ground truth summaries into a single summary [18, 13], we decided not to include a fusing procedure.
Table 1: An example of calculation of $C_U$ for the VSUMM1 keyframe selection method, video #22, feature space #6 (H16 ,lblock), the Hungarian Matching method, Manhattan distance, and threshold 0.5. The $F$-values are shown in the table; the bottom row contains the terms in (3); the values for user #2, marked with * are the ones in Figure 2.

| User #   | 1       | 2*     | 3       | 4       | 5       | $C_U$   |
|---------|---------|--------|---------|---------|---------|---------|
| $F$ (VSU MM1 , user) | 0.5000  | 0.7500*| 0.6667  | 0.2857  | 0.4444  |         |
| $F$ (Uniform, user)   | 0.5000  | 0.2500*| 0.2222  | 0.2857  | 0.4444  |         |

Term 0 | 0.5000* | 0.4444 | 0 | 0 | 0.1889

The shape of the line graph in relation to the threshold is expected to be convex with lower values for smaller and larger thresholds. For small thresholds, there will be very few matches, hence the $F$-values will be low for both the candidate summary and the uniform summary, hence the difference $C_U$ will be small. For large values of the threshold, a large number of matches will be detected in both comparisons, both $F$-values will be high, and the difference $C_U$ will be small again. The best results (larger $C_U$) are offered by the Manhattan distance. The peak for the Manhattan distance is between $\theta = 0.3$ and $\theta = 0.5$. For the Euclidean distance, there are two different types of curves. Some peak quite early, at $\theta$ between 0 and 0.5, while others stay stable. The SURF feature curves exhibit consistent and stable patterns which will be analysed later. From these findings, we favour the Manhattan distance for our proposed protocol, and will use this distance for the following evaluation of the feature spaces.

7.3. Evaluation of feature spaces

We look for a feature space which maximises the desirable quantity $C_U$. As the Manhattan distance gave the best results in the previous section, we will consider only this metric here. Figure 4 shows the results for the 10 feature spaces. Each subplot corresponds to one feature space. As in Figure 3 (b), the horizontal axis is the threshold used with the Manhattan distance, and the vertical axis is $C_U$. This time, all curves corresponding to the respective feature space are highlighted in black (30 such curves for each feature space: 6 matching methods, 5 summarisation methods).

Our results show that the simple colour spaces (1-4) are not useful in this context. The hue histograms, on the other hand, give the best results. The feature space with the largest $C_U$ is $H32\_lblock$. This is somewhat surprising because the expected winner was either CNN or SURF, being high-level features. This result hints to the possibility that spending a lot of computational effort for calculating highly sophisticated properties of images may be unjustified in some cases. Thus, we propose to use $H32\_lblock$ for the purposes of automatic evaluation of keyframe summaries when ground truth is available.

7.4. Evaluation of matching algorithms

The results for this part are shown in Figure 5. The format is the same as in Figure 4. The lines plotted in black are the ones corresponding to the matching method in the title of the subplot.

It can be seen that, for Euclidean and Manhattan distance, the Naive matching is slightly inferior to the rest of the matching
methods. This is to be expected, as the Naive labelling method may result in a large number of false positive matches for both the uniform summary and the summary of interest. This will smear the difference between the F-values, leading to low $C_U$. The remaining 5 methods are not substantially different. Interestingly, the conservative matching methods - Greedy and Mahmoud, do not work well with the SURF features. Note that here we view all the results together, both good and bad. Further analyses show that the variability in the $C_U$ for each matching method is not due to feature spaces but to summarisation method. The best such method, VSUMM1, corresponds to the highest curves.

Based on these results, we can recommend any of the three matching methods: Hungarian (minimal-weight complete matching followed by thresholding); Kannappan (The algorithm of Kannappan et al. [20]); and Hopcroft-Karp (The Hopcroft-Karp algorithm or any equivalent algorithm returning a maximal unweighted matching from the sub-threshold pairings). Of these, Kannappan has the lowest computational complexity $O(n^2)$ compared with $O(n^3)$ for Hungarian, and with the maximal-matching method whose worst-case is $O(n^{2.5})$ if implemented as the Hopcroft-Karp algorithm, or $O(n^2)$ if implemented as algorithm 6. Hence we include the algorithm of Kannappan et al. in our proposed protocol.

7.5. The proposed protocol, with example application

Several authors (e.g. [4] [14] [33]) have followed the choice of feature space, metric, algorithm, and threshold pioneered by de Avila et al. [8]. These choices seem to have had no previously published theoretical or experimental basis. The choice of H16_1block feature space, and threshold value $\theta = 0.5$ is reasonable, though the finer-grained H32_1block feature space outperforms it on average.

We propose the use of the following:

- Feature space: 32-bin hue histogram $H32_1block$ (normalised to sum 1),

- Distance for comparison of two frames represented as a point in the 32-dimensional space: Manhattan distance,

- Threshold for accepting that two frames are a match: $\theta = 0.3$.

- Matching (pairing) algorithm to determine the number of matches between two summaries: Kannappan algorithm,

- Measure of similarity between two keyframe summaries: F-measure.

Finally, in order to allow for a fair comparison between different summarisation algorithms, we propose the use of $C_U$ as defined in equation (3). Suppose that there are two algorithmic methods giving summaries $P$ and $Q$, respectively. One of them may have a larger F-value for its match to the ground truth (GT) only by virtue of the number of keyframes within. To guard against this, $C_U$ evaluates by how much an algorithm improves over a uniform summary of the same cardinality. Therefore, instead of comparing $F(P, GT)$ with $F(Q, GT)$, we propose to compare

$$C_U(P) = F(P, GT) - F(U(|P|), GT)$$

with

$$C_U(Q) = F(Q, GT) - F(U(|Q|), GT),$$

where $U(k)$ is a uniform summary with $k$ frames.

If the two rival keyframe summaries $P$ and $Q$ are of the same cardinality, their relative merit can be evaluated by $F(P, GT)$ and $F(Q, GT)$, but the question will remain whether they improve at all on a uniform (or another) baseline.

We now illustrate how the protocol can be used in practice.

Figures 6 to 10 show the summaries by the 5 algorithmic methods: DT, OV, STIMO, VSUMM1, and VSUMM2, together with the corresponding uniform summary of the same cardinality (the bottom plots). The matches are highlighted with a dark-blue frame. The images in the summaries are arranged so that the matching ones are on the left (recall that we treat the summary as a set, and not as a time sequence). The matches are calculated using the choices of methods and parameters of our proposed protocol. Table 2 shows the numerical results for

---

8MATLAB code is provided in GitHub.
the five methods, assuming that the only available ground truth is the summary of user #3. (Both the video and the user were chosen at random.)

While in this example the overall ranking of the five summarisation methods is the same according to $F(K, GT)$ and $C_U$, this will not in general be the case. Methods with higher $C_U$ should be preferred. The F-value alone may lead to false claim of matching the ground truth, especially if $F(U(|K|), GT)$ happens to be high. In some cases $C_U$ is negative, which casts a doubt on the validity of the algorithm producing the keyframe summary $K$.

8. Conclusion

We have experimentally investigated a range of choices for different components of a protocol for evaluating the outputs of keyframe-extraction algorithms. A new measure called “discrimination capacity” $C_U$ is proposed, which evaluates by how much a given summary improves on the uniform keyframe summary of the same cardinality. Using $C_U$ and the VSUMM video collection, we offer empirical recommendations, and propose a full protocol for comparison of keyframe summaries, listed at the start of sub-section 7.5.

We discovered that the most acclaimed feature spaces such as CNN and SURF are not the best choices for our protocol. A 32-bin hue histogram feature space fared better than the high-level features. Our study also contains a comprehensive collection of algorithms for matching (pairing) two summaries of different cardinalities. These algorithms did not make a profound difference to the output, therefore we chose a simple, yet efficient matching algorithm published recently [20].

Our future work will include looking into semantic comparisons between frames and summaries in addition to matching based solely on visual appearance. Combinations thereof as well as incorporating the time tag in the comparisons will be explored.

Acknowledgment

This work was done under project RPG-2015-188 funded by The Leverhulme Trust, UK.

References

[1] Aherne, F. J., Thacker, N. A., Rockett, P. L., 1998. The bhattacharyya metric as an absolute similarity measure for frequency coded data. Kybernetika 34 (4), 363–368.
[2] Apostolidis, E., Mezaris, V., 2014. Fast shot segmentation combining global and local visual descriptors. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP. No. May. pp. 6583–6587.
[3] Bay, H., Ess, A., Tuytelaars, T., Gool, L. V., 2008. Speeded-up robust features (SURF). Computer Vision and Image Understanding 110 (3), 346–359.
[4] Cahuina, E. J. C., Chavez, G. C., 2013. A new method for static video summarization using local descriptors and video temporal segmentation. In: Graphics, Patterns and Images (SIBGRAPI), 2013 26th SIBGRAPI-Conference on. IEEE, pp. 226–233.
[5] Chang, H. S., Sull, S., Lee, S. U., 1999. Efficient video indexing scheme for content-based retrieval. IEEE Transactions on Circuits and Systems for Video Technology 9 (8), 1269–1279.
[6] Chao, G. C., Tsai, Y. P., Jeng, S. K., 2010. Augmented keyframe. Journal of Visual Communication and Image Representation 21 (7), 682–692.
Figure 5: Visualisation of the $C_U$ for the 6 matching methods.

Figure 6: Proposed protocol for video #22, **DT summarisation method**, user #3 as a single ground truth.

Figure 7: Proposed protocol for video #22, **OV summarisation method**, user #3 as a single ground truth.
Figure 8: Proposed protocol for video #22, **STIMO summarisation method**, user #3 as a single ground truth.

(a) STIMO summary: 3 matches

(b) Uniform summary $U(7)$: one match

Figure 9: Proposed protocol for video #22, **VSUMM1 summarisation method**, user #3 as a single ground truth.

(a) VSUMM1 summary: 3 matches

(b) Uniform summary $U(4)$: one match

Figure 10: Proposed protocol for video #22, **VSUMM2 summarisation method**, user #3 as a single ground truth.

(a) VSUMM2 summary: 3 matches

(b) Uniform summary $U(4)$: one match

[1] Cooper, M., Foote, J., 2005. Discriminative techniques for key frame selection. In: Proceedings of the IEEE International Multimedia and Expo Workshops (ICME). Vol. 2. pp. 0–3.

[2] De Avila, S. E. F., Lopes, A. P. B., Da Luz, A., De Albuquerque Araújo, A., 2011. VSUMM: A mechanism designed to produce static video summaries and a novel evaluation method. Pattern Recognition Letters 32 (1), 56–68.

[3] Doherty, A. R., Byrne, D., Smeaton, A. F., Jones, G. J. F., Hughes, M., 2008. Investigating keyframe selection methods in the novel domain of passively captured visual lifelogs. In: Proceedings of the 2008 International Conference on Content-based Image and Video Retrieval (CIVR). pp. 259–268.

[4] Ejaz, N., Mehmood, I., 2013. Efficient visual attention based framework for extracting key frames from videos. Signal Processing: Image Communication 28 (1), 34–44.

[5] Ejaz, N., Tarig, T. B., Baik, S. W., 2012. Adaptive key frame extraction for video summarization using an aggregation mechanism. Journal of Visual Communication and Image Representation 23 (7), 1031–1040.

[6] Furini, M., Geraci, F., Montangero, M., Pellegrini, M., 2010. STIMO : STILL and MOVing Video Storyboard for the Web Scenario. Multimedia Tools and Applications 46 (1), 47–69.

[7] Furini, M., Geraci, F., Montangero, M., Pellegrini, M., 2010. STIMO : STILL and MOVing Video Storyboard for the Web Scenario. Multimedia Tools and Applications 46 (1), 47–69.

[8] Gong, B., Chao, W.-L., Grauman, K., Scha, F., 2014. Diverse sequential subset selection for supervised video summarization. In: Advances in Neural Information Processing Systems 27 (NIPS2014). Curran Associates, Inc., pp. 2069–2077.

[9] Gong, B., Chao, W.-L., Grauman, K., Sha, F., 2014. Diverse sequential subset selection for supervised video summarization. In: Advances in Neural Information Processing Systems. pp. 2069–2077.

[10] Gong, Y., Liu, X., 2003. Video summarization and retrieval using singular value decomposition. Multimedia Systems 9 (2), 157–168.

[11] Grygi, M., Grabner, H., Riemenschneider, H., Van, L., 2014. Creating summaries from user videos. In: Proceedings of the European Conference on Computer Vision (ECCV) 2014. Vol. 8695 LNCS. pp. 505–520.

[12] Hanjalic, A., Zhang, H., Dec 1999. An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis. IEEE Transactions on Circuits and Systems for Video Technology 9 (8), 1280–1289.

[13] Huang, M., Mahajan, A. B., DeMenthon, D. F., 2004. Automatic performance evaluation for video summarization. Multimedia Systems 9 (2), 157–168.

[14] Jinda-Apiraksa, A., Machajdik, J., Sablatnig, R., 2013. A Keyframe Selection of Lifelog Image Sequences. Proceedings of MVA 2013 IAPR International Conference on Machine Vision Applications, 33–36.

[15] Kannappan, S., Liu, Y., Tiddeman, B., 2016. A pertinent evaluation of automatic video summary. In: Proceedings of the 23rd International Conference on Pattern Recognition. IEEE, pp. 2240–2245.

[16] Khosla, A., Hamid, R., Lin, C. J., Sundaresan, N., 2013. Large-Scale Video Summarization Using Web-Image Priors. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2013), 2698–2705.

[17] Lee, Y. J., Grauman, K., 2015. Predicting important objects for egocentric video summarization. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1346–1353.

[18] Lee, Y. J., Grauman, K., 2015. Predicting important objects for egocentric video summarization. International Journal of Computer Vision 114 (1), 38–55.

[19] Li, Y., Merialdo, B., 2010. VERT: automatic evaluation of video summaries. In: Proceedings of the 18th ACM International Conference on Multimedia. pp. 851–854.

[20] Lidon, A., Boliatos, M., Dimiccoli, M., Radeva, P., Garolera, M., i Nieto, X. G., 2015. Semantic summarization of egocentric photo stream events. arXiv:1511.00438

[21] Lin, W. H., Hauptmann, A., 2006. Structuring continuous video recordings of everyday life using time-constrained clustering. In: Proceedings of SPIE. Vol. 6073. pp. 111–119.

[22] Liu, G., Wen, X., Zheng, W., He, P., 2009. Shot boundary detection and keyframe extraction based on scale invariant feature transform. In: Proceedings of the 8th IEEE/ACIS International Conference on Computer and Information Science (ICIS). pp. 1126–1130.

[23] Liu, T., Zhang, H. J., Qi, F., 2005. A novel video key-frame-extraction algorithm based on perceived motion energy model. IEEE Transactions
