February 03, 2016

METHOD FOR IMPLEMENTING AUTOMATED FUSION OF MULTIPLE AUDIOVISUAL RECORDINGS

Jason Chang

Krzysztof Kulewski

Clayton Michael Ritcher

METHOD FOR IMPLEMENTING AUTOMATED FUSION OF MULTIPLE AUDIOVISUAL RECORDINGS

ABSTRACT

A system and method for implementing automated fusion of multiple recordings of an event by using video processing is disclosed. The application determines the best camera angle from the available recordings for viewing important parts of the event and allows the user to automatically generate a fused version from the several recordings. The method includes video synchronization, tracking and clustering tracked objects in the video, followed by optimum video feed inference. The method greatly reduces time and eliminates manual effort, as processes such as identifying the best quality among different versions available, sequencing various clips correctly, and fusing them seamlessly are all automated.

BACKGROUND

Generally, we consume a lot of time recording videos using low-cost recording devices such as smart phones. Such recordings are ubiquitous, and the number of videos recorded continues to grow rapidly. Usually, different people record videos of the same event (concert, wedding, recital, etc.). However, these recordings may fail to capture the event optimally because a user has to watch the event and record each video separately and repeatedly switch back and forth between the event and the recording. Thus, each recording may not capture the best camera angle at all times or the most important part of the event. Further, no single recording may capture the entire event. Fusing the audiovisual pieces together could produce a meaningful video, but manual fusion consumes a lot of time. Additionally, videos of events may need to be fused quickly, such as for live sports coverage, and hence may require a team of

professionals. However, such a solution is not feasible for all events. Therefore, there is need for an automated method for fusing audiovisual pieces of an event to produce a consolidated video.

DESCRIPTION

A system and method for implementing automated fusion of multiple recordings of an event by using video processing software or application is disclosed. The application determines the best camera angle from the available recordings for viewing important parts of the event and allows the user to automatically generate a fused version from the several recordings.

The application for fusing video recordings of the same event uses three steps:

A. Video synchronization

B. Object of interest tracking and clustering

C. Optimum video feed inference

Video synchronization in step A, involves computing the temporal offsets between the videos using pairwise spectral cross-correlation between the audio streams and belief propagation to eliminate outliers.

Tracking and clustering the object of interest in step B involves using the steps as illustrated below:

1. Downsampling the videos by synchronizing to a certain number of frames per second
2. Using object detection on objects of interest in each video (e.g. faces, animals, cars, etc.).
3. For each of the detected objects, assigning a track, where the track is a group representing a single continuous sequence of detections of the same object determined by frame-to-frame object detection as shown in FIG. 1.

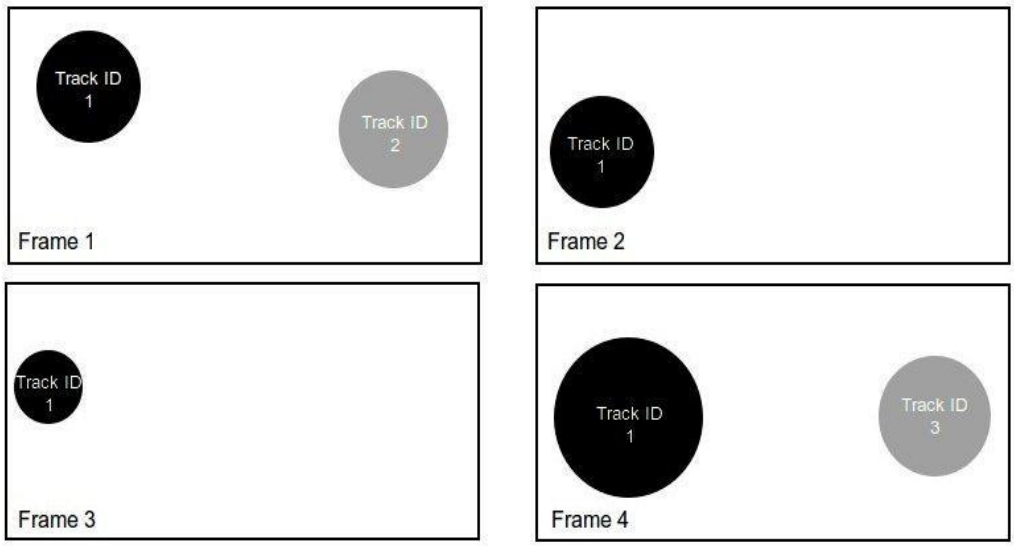

FIG. 1: Object of interest tracking

In frame 1, the black and grey circles appear for the first time, and are assigned track IDs 1 and 2, respectively. In frame 2, the black circle has moved a bit but is detected near its last detection, so it is again assigned track ID 1. The grey circle is not detected, and thus not assigned a track ID. In frame 3, the black circle is again detected. Its size change does not affect its track ID assignment, so it is given track ID 1 again. In frame 4, the black circle is assigned track ID 1 once more. Although the grey circle is detected near where it was last seen, the detections are far enough apart in time that the black circle is assigned a new track ID, 3. The tracking and clustering further involves three more steps:

4. Recording the size of each object detected relative to the video frame and estimating the pose angles with reference to the camera.
5. Merging the track groups across all videos into larger groups or clusters and in the final step calculating a popularity of each cluster. In merging the track groups across all videos into clusters, each cluster represents a group of detections of the same real-world object (e.g. same person's face, same animal, etc.). The merging is done by determining the

similarity between two track groups, using object recognition. If the similarity is above a threshold, the track groups are merged together. This is repeated until no pair of groups has a similarity measure above the threshold.

6. The popularity of each cluster is calculated as the percentage of frames in all the videos that the object appears in.

Optimum video feed inference in step C involves further processing as follows:

1) Selecting the detected object, from each video frame that has the highest popularity score.
2) For these detected objects, at each synchronized time step, concatenating their feature vectors (relative size, popularity score, and pose angles) together into one observation of all the video streams. If no object was detected in a particular frame, popularity score could be 0.
3) For each feature type (i.e. relative size, popularity score, etc.), defining two continuous probability distributions as mixtures of Gaussians. The first distribution that should be defined is the probability distribution of the feature given that the video stream is the best current video stream. This should have a high probability for values of the feature that correlate to a good video. For example, this distribution for the popularity score might be centered at 100%, because an object in the best current stream is likely to be a popular object.

An example of the probability distribution for the popularity score is shown in FIG. 2. The distribution shown is a Gaussian distribution centered at 1, or 100%. The plot is zoomed to show the values between 0 and 1 because the popularity score cannot take on other values. Because the popularity score can only be between 0 and 1, the Gaussian should be truncated at these values and normalized to integrate to one within this range.

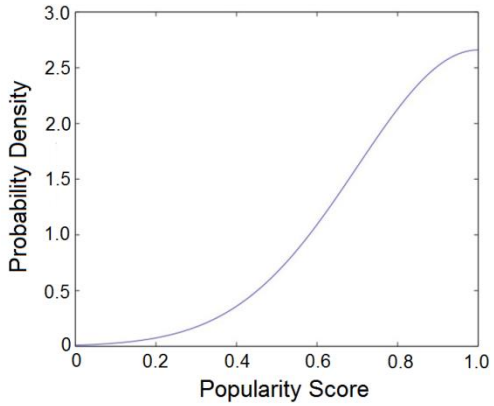

FIG. 2: Popularity score vs. probability density of the best current video stream

The other distribution that needs to be defined is the probability distribution of the feature given that the feature is not from the current best video stream, or "other than" the best. This should be highest at values suggesting that a video stream is not good for a particular object. FIG. 3 shows an example of what this distribution might look like for the popularity score. The distribution in the image is a Gaussian distribution centered at 0.

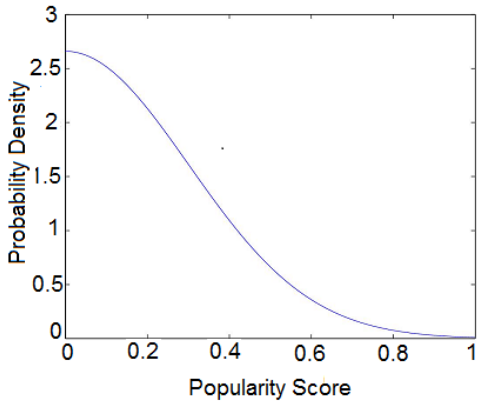

FIG. 3: Popularity score vs. probability density of "other than" the best current video stream

A final operation relating to optimum video feed inference in step C involves further processing as follows using the probability distributions as shown in FIG. 2 and FIG. 3 through formulating a Hidden Markov Model (HMM) representation of the problem as follows:

- The observed state of the HMM, **y**(t), is the concatenated list of feature vectors from 2) at time t. Each individual feature can be referred to as $\mathbf{y}_{ij}(t)$, where i is the video stream index, and j is the index of the feature in the individual stream's feature vector.
- The hidden state of the HMM, **x**(t), is the best video stream at time t.
- The distributions described in 3), then, $p(\mathbf{y}_{ij}(t) \mid \mathbf{x}(t) = i)$ and $p(\mathbf{y}_{ij}(t) \mid \mathbf{x}(t) \neq i)$, respectively. These distributions are assumed to be conditionally independent. Letting there be *n* video streams and *m* features for each frame of each stream, the emission probability, $p(\mathbf{y}(t) \mid \mathbf{x}(t) = k)$, equals $\prod_{□=0}^{□-1} \prod_{□=0}^{□-1} p(\mathbf{y}_{ij}(t) \mid \mathbf{x}(t) = k)$, using the correct distributions for when i = k and i ≠ k.
- The transition probability distribution of transitioning from hidden state k, $p(\mathbf{x}(t) \mid \mathbf{x}(t - 1) = k)$, is 0 for **x**(t) = any video stream that is empty at time t. Otherwise, the distribution is uniform over **x**(t), with a peak at **x**(t) = k. In another aspect, the model is formulated so that there is 0 probability of transitioning to a hidden state (best stream) that is empty. Also, the probability of transitioning to a non-empty state is the same, except for the probability of staying in the same state, which is higher.
- The starting state probability distribution, $p(\mathbf{x}(0))$, is uniform over all video streams that exist at t = 0.

- Use the Viterbi Decoding algorithm with the HMM formulated as above to obtain the hidden state, or best video stream, at each time step.

The system and method disclosed provides for automated process for fusing multiple video recordings of an event, thereby producing a single optimized video covering the event. The

method greatly reduces time and eliminates manual effort, as processes such as identifying the best quality among different versions available, sequencing various clips correctly, and fusing them seamlessly are all automated.

In situations in which the systems and methods discussed herein may collect personal information about users, or may make use of personal information (e.g., photos, videos, user data), users are provided with one or more opportunities to control how information is collected about the user and used in one or more described features. A user is provided with control over whether programs or features collect user data (e.g., recognition of a user's face in a photo or video, information about a user's social network, user characteristics (age, gender, profession, etc.), social actions or activities, a user's preferences, content created or submitted by a user, a user's current geographic location, etc.). A user is provided with control over whether programs or features collect user information about that particular user or other users relevant to the program or feature. Each user for which personal information is to be collected is presented with one or more options to allow control over the information collection relevant to that user, to provide permission or authorization as to whether the information is collected and as to which portions of the information are to be collected. For example, users can be provided with one or more control options over a communication network. In addition, certain data may be treated in one or more ways before it is stored or used, so that personally identifiable information is removed. For example, a user's identity may be treated so that no personally identifiable information can be determined for the user, or a user's geographic location may be generalized to a larger region so that a particular location of a user cannot be determined.