MoViMash: Online Mobile Video Mashup

Ashish Kumar
Dept. of Computer Science
IIT Ropar
Email: 2016csb1033@iitrpr.ac.in

Siddharth Nahar
Dept. of Computer Science
IIT Ropar Email: 2016csb1043@iitrpr.ac.in

Rahul Lahane
Dept. of Computer Science
IIT Ropar
Email: 2016csb10xx@iitrpr.ac.in

Anurag Singh Mehta
Dept. of Computer Science
IIT Ropar Email: 2016csb1111@iitrpr.ac.in

***Abstract*—With the proliferation of mobile video cameras, it is becoming eas- ier for users to capture videos of live performances and socially share them with friends and public. As someone who is attending such live performance has limited mobility, video cameras are restricted to being able to capture only a few limited angles which produces a monotonous clip. At such per- formances, however, multiple video clips can be captured by differ- ent users, likely from different angles and distances. These videos can be combined to produce a more interesting and representative mashup of the live performances for broadcasting and sharing. The earlier works select video shots merely based on the quality of cur- rently available videos. Shot transition and shot length distributions are learned from professionally edited videos. Further, we introduce view quality assessment in the frame- work to filter out shaky, occluded, and tilted videos. To the best of our knowledge, this is the first attempt to incorporate history- based diversity measurement, state-based video editing rules, and view quality in automated video mashup generations. Experimen- tal results have been provided to demonstrate the effectiveness of MoViMash framework.**

Categories and Subject Descriptors I.2.10 [Vision and Scene Understanding]: Video Analysis

General Terms: Algorithms, Design.

Keywords: Mobile Video, Virtual Director, Video Mashup.

I. Introduction

The estimate was that camera phone supply would increase to 1.14 billion by the end of 2011.[1]. Furthermore, a survey of over 2,500 respondents by Photobucket reveals that 45spon- dents use mobile devices to shoot video at least once weekly during the summer of 2011, validating the significant increase in the amount of mobile video uploaded to Photobuckets video shar- ing website (14 in Summer 2011 compared to December 2010) [2].

Proliferation of such mobile devices with video capture capa- bility has enabled users to capture video of their life events such as concerts, parades, outdoor performances, etc, and socially share them with friends and public as it happens. Videos recorded by a single user at such events are shot from a limited range of an- gles and distances from the performance stage, as an attendee typi- cally has limited mobility (e.g., constraint by seating arrangement). The recorded video can be monotonous and uninteresting. Further- more, videos recorded are typically short (in the order of minutes or tens of minutes), due to tired arms or power constraint of mobile devices. There are, however, likely to have more than one users recording the same performance from different angles at the same time, especially at a well-attended performance.

These recorded and shared video clips of the same per- formance can be cut and joined together to produce a new mashup video, similar to how a TV director of a live TV show would switch be- tween different cameras to produce the show. Generation of a video mashup can be cast as a video selection problem: given a set of video clips capturing the same performance event, automatically select one of the video clips at any one time instance to be included in the output mashup video.

In this paper, we introduce MoViMash, our approach to solve the above video selection problem. MoViMash aims to produce mashup video from a set of mobile devices that is interesting and pleasing to watch, and uses a combinations of content-analysis, state-based transitions, history-based diver- sity, and learning from human editors to achieve this goal.

We now provide an overview of how MoViMash works in the usual setting of live performances, shown in Figure 1. There is generally a staging area and an audience area where the audiences either sit or stand to watch the performance, and record the perfor- mance with a mobile device. This setting poses a few challenges to video mashup.

Since the videos are recorded with a hand-held mobile device, from the audience area, and likely by non-professional, there is no guarantee on the view quality. The videos can be shaky or tilted.

When MoViMash needs to decide which video to select, it first filters out the videos with bad views currently from further consideration for selection. To achieve this, MoVi- Mash analyzes the video to determine the current shakiness, the tilt angle, and the level of occlusion in the video. Note that shakiness and tilt angle can be obtained from easily sensory data of mobile device when available.

The shooting angle of the remaining videos are then clas- sified as either center,persist. To this end, MoViMash tries to imitate a professional videoeditor, by using a finite state machine, whose transition probabili- ties are learned from analyzing professionally edited videos of the same type of event. The rationale behind the inclusion of learn- ing is that,

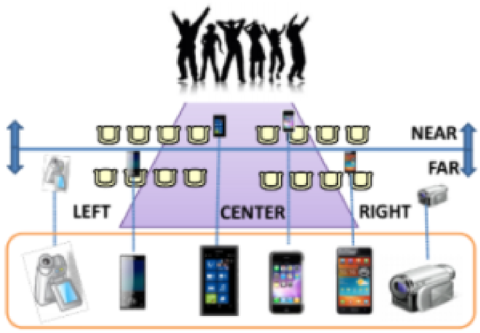


Fig. 1. A general performance scenario

we have observed that there are no generic editing rules that can be precisely defined to work with all types of events. The video editors make fine decisions such as shot lengths and transi- tions based on their experience which is hard to enumerate.

The videos from the selected class are further ranked based on the video quality and diversity values to make the final selection. To consider video quality, MoViMash favors video with low blurri- ness, low blockiness (good compression), good contrast, and good illumination in each video. To consider diversity, MoViMash stores a history of recent video selections and favors videos with dissimi- lar views with recent selections.

We have developed MoViMashs algorithm such that it is online and only depends on history information. As such, even though it is not our main goal in this paper, MoViMash can be applied to mashup of live video feeds from mobile devices.

We now briefly compare MoViMash to existing work to high- light the contribution of this paper. There has been few works on video selection in a lecture broadcast and video streaming [21] [6] and video conferencing [3]. Also, the diversity is only calculated based on the comparison of the last image of the current shot and first image of the next shot. It does not consider the history of video selection and the time for which a particular camera is selected. Further, video editing rules, which are subtle in the case of live performances, are not considered.

Contributions. We now summarize our contributions in this pa- per as follows:

- We propose a state-based approach for shot selection that in- corporates the selection history in the decision process. Earlier methods select shots based on only currently available videos.
- We include view quality in the framework to filter out the bad views that are occluded, tilted, or shaky. Earlier methods only considered video quality.
- We build a comprehensive model to calculate diversity that considers both previously selected videos and shot lengths.
- We propose a learning-based approach where the shot tran- sition probabilities and shot lengths are learned from profes- sionally edited videos.

Organization. The rest of the paper is organized as follows. We provide a review of earlier work in Section 2. In Section 3 we describe proposed mashup framework. We evaluate our system in Section 4. The conclusions are provided in Section 5.

II. Previous Work

There has been only few works on online camera selection. In most of these works, videos are mainly selected to show the speak- ers. In the work by Machnicki and Rowe [9], an online lecture webcast system is presented in which the cameras that are focusing on speaker and the presentation (the screen) are selected iteratively until anybody from audience asks question. When audience ask question, the camera that is focusing the person asking question is selected. tection is not a reliable basis to select videos.

Al-Hames et al. [3] extends the camera selection work to include the motion features. We do not use motion features in our frame- work because both performers and audience generate continuous motion. Also, the movement of the mobile camera can inject erro- neous motion in the video, which is aesthetically appealing. Yu et al. [20] propose to customize the camera selection and shot lengths based on user preferences. At every lecture webcast receiving site, the user can give score to the videos and specify rules for shot lengths. While such an interactive selection of cameras is useful for educational scenarios, people may find it annoying and stress- ful for performances, particularly when the number of videos is large.

A camera selection method for sports video broadcast is pro- posed by Wang et al. [16]. The authors assume one main cam- era and other sub cameras. The empirical main camera duration is found to be from 4 to 24 seconds, and sub camera duration is found to be 1.5 to 8 seconds. They select a sub cam- era based on the clar- ity of the view, determined using motion features. In our work, along with shakiness of the videos, we also calculate view qual- ity in terms of occlusion and rotation; and video quality in terms of contrast, blur, illumination, and blockiness. We also include explicit measurement of diversity in the framework. Engstrom et al. [8] discuss automatic camera selection for broadcast in a sports event capture scenario. The work mainly promotes collaborative video production, i.e., video recorded by production team as well as the consumers.

In other media production applications, the shots are se- lected to convey the story to the audience. For instance, de Lima et al. [7] propose a method to automatically select shots from multiple cam- eras for storytelling, according to the rules provided by the director. These methods are not useful for us as live performances generally do not have any story.

Recently, there has been works on creating video mashups from given set of videos. In one of the most recent works [14], Shrestha et al. select the cameras based on video quality. Although the au- thors refer to term diversity paper, the authors rely on the future video for current shot selection. While this approach is fine for combining stored videos, it is not suitable for live applications such as broadcasting and live sharing.

TABLE I
A COMPARISON OF PREVIOUS WORK

Work	Online	Diversity	Learning	Video Quality	View Quality	Scenario
Machnicki et al. [9]	Yes	No	No	No	No	Lecture webcast
Cutler et al. [6]	No	No	No	No	No	Meeting
Al-Hames et al. [3]	Yes	No	No	No	No	Meeting
Yu et al. [20]	Yes	No	No	No	No	Lecture webcast
Zhang et al. [21]	Yes	No	No	No	No	Lecture webcast
Wang et al. [16]	Yes	No	No	No	No	Sports webcast
Engstrom et al. [8]	Yes	No	No	No	No	Sports webcast
Lima et al. [7]	No	No	No	No	No	Storyline
Ranjan et al. [12]	Yes	No	No	No	No	Meeting
Shrestha et al. [14]	No	No	No	No	No	Live Performances
Proposed[14]	Yes	Yes	Yes	Yes	Yes	Live Performances

We have provided a comparison of the related work in Table 1. The works have been compared with respect to the following as- pects: (1) can the method be applied online (a method that uses fu- ture information cannot be applied online)? (2) is selection history- based diversity considered? (3) is learning incorporated? (4) is video quality (clarity, contrast etc.) considered? (5) is view quality (view occlusion, tilted view etc.) considered? and (6) what is the underlying application scenario? It can be easily seen that the proposed method is the first attempt to consider history based diversity through learning for online video selection for live performances.

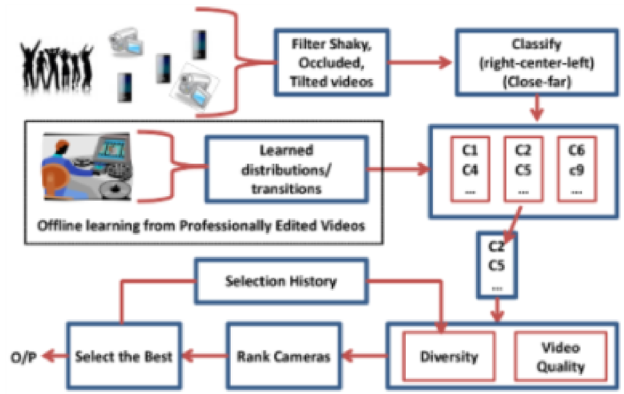


Fig. 2. The virtual director framework

III. MOVIMASH FRAMEWORK

In this section, we first enumerate the design principles that we have followed in the development of MoViMash and then describe the framework. After an overview of MoViMash, we focus on in- dividual components.

A. Design Principles

The end goal of the MoViMash is to produce a mashup that users like. To achieve this goal, we have followed a set of design princi- ples as follows:

- **Video Quality:** In our discussion, video quality includes sharpness, contrast, illumination, and blockiness (due to video compression). A good image quality gives pleasing experi- ence to the viewers [10]. Therefore, in our framework we give priority to good quality videos.
- **View Quality:** A video that is captured by a tilted camera (rotated around horizontal axis) may have very good video quality, yet, users generally do not like tilted views. Similarly, a view in which a person or object is occluding stage area (blocking performance view) may be annoying to the user. Therefore view quality is also important. We measure view quality in terms of occlusion, tilt, and shakiness.
- **Diversity:** While static cameras always record videos from same perspective, mobile users generally shoot videos from a number of views and diverse perspectives. We take this opportunity to include more diversified views in the mashup. Both temporal and spatial aspects of diversity are considered in the proposed framework.
- **Learning:** When professionals edit the videos, they make many decisions based on their experience. Such decisions include shooting angle, distance from the stage, and shot length. It is, however, difficult to precisely state this experience in terms of hard-coded rules. Therefore, in this work, we learn the shot transitions and lengths from professionally edited videos.

The above mentioned design principles are met in our framework through various quality metrics and video selection/filtering phases, as described in the following section.

B. Framework

At every time instant, we have a number of videos to choose from. Once we have chosen the video, we also need to decide when to switch to another video. Hence, there are two main questions involved here that need to be answered for combining videos: (1) which video to select? (2) when to switch to another video? We use a three-phase method to select the video while the length is determined based on learned editing rules and overall quality score of the selected video.

Figure 2 shows the block diagram of overall framework. The proposed framework consists of one offline learning phase and three online selection phases namely filtering, classification, and selec- tion. At any given time, the following steps are taken to select the most suitable video at current instant:

1) **Filtering:** In the filtering step, we determine videos that are unusable by comparing occlusion, shakiness, and tilt scores against empirically determined thresholds. The remaining videos are passed to the classification stage.
2) **Classification:** The selected cameras are classified as one of right, center, and left according to the capturing angle. Further, according to the viewing distance from the stage, they are classified as near or far.
3) **Class Prediction:** According to the class of currently selected video, and class transition probabilities learned from professionally edited videos, a most suitable class is pre- dicted and videos from that class are selected for further con- sideration.
4) **Video Selection:** The classified cameras are further ranked with respect to a combined score of video quality, diversity, and shakiness. The video with highest score is selected.
5) **Shot Length:** The length of the video is selected based on learned distributions and video quality. A higher quality video is generally selected for longer time.

While filtering and selection phase ensure view and video qual- ity, the classification and diversity ensure that we select videos recorded with different angles and viewing distances to provide a complete and interesting coverage of the performance. We now describe each component of the framework in detail.

C. View Quality

The view quality is measured in terms of three characteristics: occlusion, shakiness, and camera tilt. The details of measurement of each of these quantities is given below.

1) Occlusion: For both a stand mounted camera and a mobile camera, there is always a chance of view occlusion. At crowded places, people sometime do not notice the cameras recording the video and oc- clude the performance view. Even if people notice the cameras, they stand in front of or walk across the cameras, because the main purpose of the performances is to entertain the audience who are present at the venue rather than video recording. Therefore, we de- tect the videos which are recorded by occluded cameras and filter them out.

Occlusion detection methods are popular in the field of object tracking [13], [19]. There methods employ various appearance mod- els to seamlessly track multiple objects. In this case, the occlusion occurs when an object is hidden behind another. In live perfor- mances, this could be intentionally done by the performers, i.e., one performer coming in front of other. We are more interested in detecting the audience blocking the view. Therefore, those works are not applicable here.

We have developed an edge density based method to detect videos with occluded views. The method is based on the assumption that the objects that occlude the performance area will result in lower edge density than the performance area. Therefore, the non-occluded area of the image, which is far from the camera, will result in more dense edge points than the occluded area. To differentiate between homogeneous regions of the stage area, which could also have less edge density, and occluded area; we perform connected compo- nents on the edge image. Following are the steps of the occlusion detection in a given image I:

1) Edge Detection: In the first step, we calculate the presence of an edge at each pixel location. Let I e be the resulting binary edge image:

$$I^e(x,y) = \begin{cases} 1 & \text{if edge is detected at pixel} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

2) Edge Density: We convolve the edge image with a square matrix W with all of its elements unity:

$$I^d = I^e \odot \; W \tag{2}$$

3) Labeling the Patches: The image is now divided into patches of block size b b. Each patch is labeled as 1 if the sum of edge densities is less that a threshold, else it is labeled as 0.

$$I^p(x',y') = \begin{cases} 1 & \text{if the sum of edge densities in the} \\ & \text{patch (x', y') is greater than threshold} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

The output of the operation gives the density of edges around each pixel.

4) Connected Components: There can be homogeneous regions in the non-occluded area as well. These regions, however, are generally small. Therefore, connected components operation is performed to find the size of largest group of connected patches with label 1, which corresponds to occluded region.
5) Occlusion Score: To calculate the final occlusion score S o ,we first calculate the fractional occluded region f in the connected components output image, i.e.,

$$f = \frac{\text{No of 1 patches}}{\text{Total number of patches}} \tag{4}$$

We also observed that generally the dynamic range of f is very small. Therefore, we expand its range with an exponential function to calculate the final score S o :

$$S^o = 1 - e^{-f} \tag{5}$$

2) Tilt: In this work, we define tilt as the rotation of the camera around horizontal axis. Users generally do not like the videos recorded by tilted cameras. Therefore, we detect the tilted camera views and filter them. Here we use the heuristic that for a horizontally placed camera, most of the lines in the view are horizontal,

while an inclined has non-horizontal lines. The following steps are taken to calculation tilt:

- Line Detection: We use Hough transform to detect the straight line in the image. Let l l'_i be the length of the i^{th} line and o'_i the angle with respect to the horizontal line.

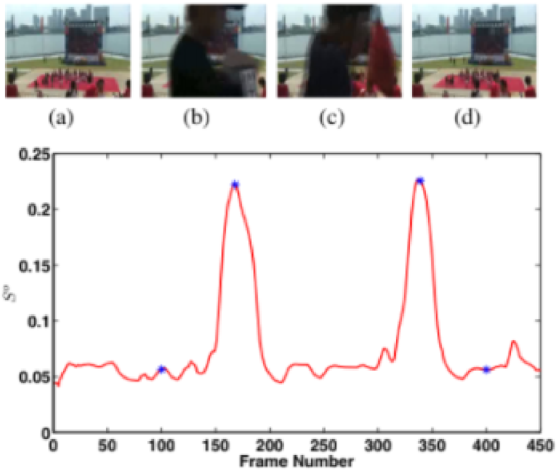


Fig. 3. Occlusion detection. Figures (a)-(d) show the frames 100, 168, 339, and 400 of the test video respectively. Figure (e) shows the corresponding occlusion score

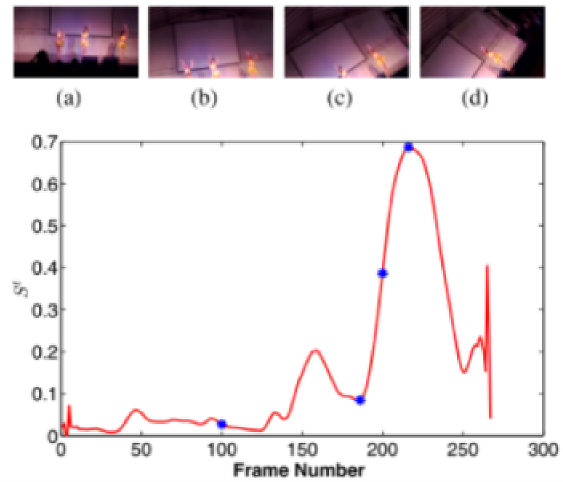


Fig. 4. Tilt results. Figures (a)-(d) show the frames 100, 186, 200, and 286 of the test video respectively. Figure (e) shows the corresponding tilt score

- Angle Restriction: We assume that the maximum tilt a camera can have is less than /4 and any line with the inclination above this angle is noise and not considered in calculation. Let the resulting orientation of l'_i line be o_i .
- The final tilt score S^t is calculated as absolute of the mean weighted orientation and normalized by /4:

$$S^t = \frac{abs\left(\frac{1}{N^t}\sum_{i=1}^{N^l} o_i \ * \ l_i\right)}{\pi/4} \tag{6}$$

The resulting occlusion scores for an example video sequence are shown in the Figure 3. The sequence shows a person walking across a camera, which is recording an outdoor performance. We can see that as the person . We found that for a patch size of 20*15 pixels, videos with occlusion score more than 0.2 are very bad, so these are filtered in the framework.

3) Shakiness: Shakiness is calculated based on the method described in [4]. In this method, the pixel values are projected on horizontal and vertical axes. The horizontal and vertical projections are matched across the frames for calculating camera motion. The final value of shakiness score, S s , the normalization value is 300; for any value above 300, S s is saturated to 1.

D. Learning

As mentioned earlier in Section 1, it is difficult to precisely enu- merate all the rules which professional editors follow in selecting a video and its corresponding shot shooting angle, shoot- ing distance, and shot length. Following are the steps taken in the process of learning:

- At first, we divide the video into a sequence of shots and record shot length.
- Each shot is classified as right ($\mathcal{R}$), left ($\mathcal{L}$), or center (C) based on shooting angle (Figure 1).
- Depending on the distance of the recording device from the stage, the videos are further classified as near ($\mathcal{N}$) or far $\mathcal{F}$) (Figure 1).
- Based on both classifications, we define six states (also re- ferred as classes in the paper) in which a video can be at any time instant, i.e., ($\mathcal{CN}$) , ($\mathcal{CF}$), ($\mathcal{RN}$) , ($\mathcal{RF}$), ($\mathcal{LN}$) , and ($\mathcal{LF}$) .
- From the sequence of the shots, we calculate the state transi- tion probabilities for the above described six states.
- We now feed the transition probabilities (transition ma- trix) along with shot lengths (emission matrix) to an hidden Markov model (HMM). The HMM generates a sequence of shot states and their corresponding lengths.

$$\begin{array}{c} \\ \mathcal{CN} \\ \mathcal{CF} \\ \mathcal{RN} \\ \mathcal{RF} \\ \mathcal{LN} \\ \mathcal{LF} \end{array}
\begin{array}{c}
\begin{array}{cccccc} \mathcal{CN} & \mathcal{CF} & \mathcal{RN} & \mathcal{RF} & \mathcal{LN} & \mathcal{LF} \end{array} \\
\begin{pmatrix}
0 & 0.4 & 0.2 & 0.1 & 0.2 & 0.1 \\
0.6 & 0 & 0.1 & 0.1 & 0.1 & 0.1 \\
0.5 & 0.1 & 0 & 0.1 & 0.2 & 0.1 \\
0.2 & 0.2 & 0.4 & 0 & 0.1 & 0.1 \\
0.4 & 0.2 & 0.2 & 0.1 & 0 & 0.1 \\
0.2 & 0.2 & 0.1 & 0.1 & 0.4 & 0
\end{pmatrix}
\end{array} \tag{7}$$

$$\begin{array}{c} \\ \mathcal{CN} \\ \mathcal{CF} \\ \mathcal{RN} \\ \mathcal{RF} \\ \mathcal{LN} \\ \mathcal{LF} \end{array}
\begin{array}{c}
\begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\
\begin{pmatrix}
1/31 & 2/31 & 4/31 & 7/31 & 7/31 & 6/31 & 4/31 \\
3/12 & 4/12 & 2/12 & 1/12 & 1/12 & 1/12 & 0 \\
2/15 & 3/15 & 4/15 & 3/15 & 2/15 & 1/15 & 0 \\
3/10 & 4/10 & 2/10 & 1/10 & 0 & 0 & 0 \\
2/15 & 3/15 & 4/15 & 3/15 & 2/15 & 1/15 & 0 \\
3/10 & 4/10 & 2/10 & 1/10 & 0 & 0 & 0
\end{pmatrix}
\end{array} \tag{8}$$

We use affine transformation to classify the video, giving an ac- curacy of 77% on our dataset. However, since learning is one time job, we performed manual classification of shots during the learning phase to get accurate statistics. Equation 7 shows the learned transition matrix while Equation

8 emission matrix. We have carefully selected five videos (live group dances with length of videos ranging from 210 to 300 seconds), which are profession- ally edited and aired on television. We downloaded these videos from YouTube.

These videos include concerts by professional bands and per- formance at the Academy Awards ceremony. We observed that in dance videos, the shot lengths are relatively smaller (average around 2.3 seconds) compared to solo singing videos (average around 3.5 seconds). This finding implies that the learning dataset should comply with the type of performance for mashup. We also observed that the average shot lengths for all five dance videos ranged between 2.2 seconds to 2.4 seconds, showing little varia- tions, which shows that a particular type of events have similar pattern of transitions and shot lengths which can be learned and applied to create online mashup.

E. Video Quality

We can have different quality videos because of the limitation of recording devices, varied camera positioning, lighting conditions, camera angle, and video recording skills of the person. To produce aesthetically beautiful video, it is important to consider the quality of the videos.We are considering the following aspects to obtain video quality score:

- **Blockiness:** The blocking effect mainly comes due to poor quality of data compression. To measure blockiness, we take current image as sample and calculate its compression qual- ity using the method described in [18]. The method generates a score that takes a value between 1 and 10 (10 represents the best quality, 1 the worst). We normalize the score between 0 and 1. Let S b be the blockiness score.
- **Blur:** The video can be blurred due to many reasons such as out-of-focus recording, camera movement etc. We are cal- culating blur based on the method described in [5]. Let S br be the blur score which varies between 0 to 1 (0 represents blurred and 1 sharp).
- **Illumination:** There can be videos that are recorded in poor lighting conditions. The purpose of including this metric in quality measurement is to avoid selecting dark videos. The illumination score for the image S im (with width N w and height N h) is calculated as average gray value, normalized by 255.

$$S^{im} = \frac{1}{255}\sqrt{\frac{1}{N_w * N_h}\sum_{x=0}^{N_w}\sum_{y=0}^{N_h} I(x,y)} \tag{9}$$

- **Contrast:** It has also been found in the literature that an im- age with good contrast is appreciated by the viewers [10]. Therefore, contrast is also chosen as one of the metrics. The contrast score S c is calculated as standard deviation of the pixel intensities.

$$S^c = \frac{1}{255}\frac{1}{N_w * N_h}\sum_{x=0}^{N_w}\sum_{y=0}^{N_h}(I(x,y) - \overline{I})^2 \tag{10}$$

- **Burned Pixels:** It has been identified that pixels that are close to 255 or 0 are generally not informative [14]. If N b is the number of such pixels, the quality score representing burnt pixels is calculated as follows:

$$S^{bp} = \begin{cases} 1 - N^b/(0.25 * N^i) & \text{if } N^b/(0.25 * N^i) < 1 \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

  where N^i is the total number of pixels in the image. In this case, a value of 1 represents best quality, i.e., no burnt pixels; while a value of 0 means at least 25% pixels are burnt.

The individual quality scores are multiplied to calculate overall video quality score S_q, i.e.,

$$S^q = S^b \times S^{br} \times S^{im} \times S^c \times S^{bp} \tag{12}$$

We have chosen to multiply the individual scores because we want to give priority to the videos that are good in all aspects.

F. Diversity

The aspect of diversity is included in the framework by calculat- ing the similarity of the views of the videos cameras that have been selected so far. The history is stored as set of chronologically order tuples, i.e.,

$$\mathcal{H} = \left\{\left(I_j^h, \Delta_j\right)|1 \leq j \leq N^v\right\} \tag{13}$$

where N^v is the number videos selected in the recent past. Each tuple has the following two entries:

- I^h - Snapshot from the selected cameras at the time of selec- tion.
- Δ - The time for which the particular camera is selected. It is normalized between 0 to 1 by dividing each video duration by the total time over which history is stored.

Let V S be the view similarity matrix:

$$V\,S = \left\{v^{ij}|1 \leq i \leq n1 \leq j \leq N^v; \forall i = j, v^{ij} = 1\right\} \tag{14}$$

where n is number of cameras, and $\sqsubseteq_\rangle|$ is the view similarity measure between current frame from the $\rangle^{\sqcup\langle}$ video and $|^{\sqcup\langle}$ frame of the history. The overall steps of diversity calculation are as follows:

1) Determine the view similarity matrix V S by comparing cur- rent frame with the frames stored in the history, i.e.,

$$v^{ij} = Diff(I_i^c, I_j^h) \tag{15}$$

   where I_i^c is the current frame of i^{th} camera, I_j^h h is the j^{th} frame of the history, and Diff can be any function to calcu- late view similarity. We are using SSIM [17] for this purpose

2) For the given content, the user interest decreases with time over which the user watches same or similar content. Hence, the diversity score of the i th video,

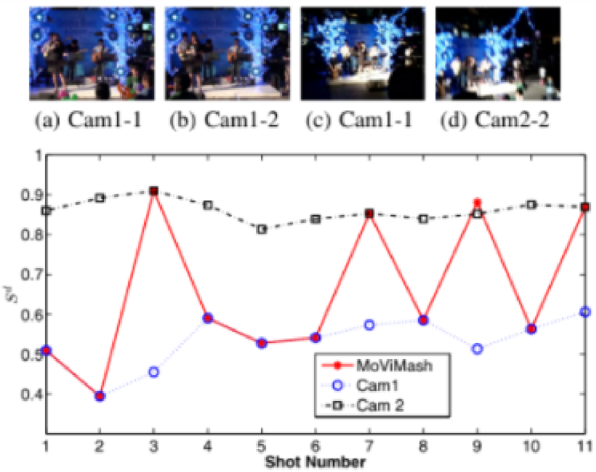


Fig. 5. Figure 5: Diversity value for two candidate videos and final mashup

i.e., S d is calculated for each of the current videos as follows:

$$S^d = \sum_{j=1}^{N^v} v^{ij} * \Delta_j \tag{16}$$

3) Store the viewing time of the previous video and the current frame of the selected video in H. If we choose a scheme where each camera is selected only for fixed amount of time, we may just store the current frame of the selected video.

The diversity scores for two candidate videos (Cam 1, Cam 2) and final mashup created using MoViMash for a performance (P3 in Table 2) are shown in the Figure 5. Although we are showing diversity for only two videos for clarity, there were five candidate videos in total. We can see that whenever a video gets selected, its diversity generally reduces, e.g., diversity of Cam 1 after Shot 8 and diversity of Cam 2 after Shot 3. At Shot 4, Cam 1 gets selected despite low diversity because its video quality is much better than others (Figure 5.a-b) with a stable view. The diversity of Cam 2 decreases even though it is not selected. It is because during this time, its view is similar to Cam 1, resulting in large (near 1) value of v 12 (Equation 15). At Shot 9, a third (other than Cam 1 and Cam 2) video gets selected until Cam 1s diversity increases enough so that it gets selected again. In summary, the metric S^d is able to capture and spatial and temporal diversity of videos.

G. Final Ranking

For all the videos of the selected class, we have three metrics: video quality score, diversity score, and shakiness score. We cal- culated weighted sum of these values to calculate final score S^f:

$$S^f = \alpha_1 S^q + \alpha_2 S^d + \alpha_3 (1 - S^s) \tag{17}$$

where α_1 , α_2 , and α_3 are weighting coefficients and α_1 + α_2 + α_3 = 1. In the experiments, we are using α_1 + α_2 + α_3

TABLE II
DETAILS OF THE DATASET

Performance	Type	Recordings	Duration(m)	Frame	Resolution
P1	Group Dance	12	4’	30	720*480
P2	Group Dance	12	3’50”	30	No720*480
P3	Solo Singing	5	2’50”	25	640*480

which gives equal weightage to the quality, diversity, and stabil- ity of the videos. Nevertheless, these coefficients should be deter- mined based on the type of performance. For instance, in a hip-hop video mashup we can give less weightage to shakiness for better di- versity and quality. A shaky video, however, can be annoying if the performance has smooth movements such as a tango performance. We therefore need to keep 3 higher in this case. Furthermore, if the videos are generally bad in the quality, we can give set high value for 1 . The shot from the video with the highest score is selected at the current time instant.

H. Length Section

To determine the switching instant, we are using a method which combines the learning based prediction to obey the editing rules and the superiority (in terms of overall quality) of the currently selected video. As discussed in the learning section (Section 3.4) For example the center videos are generally selected for longer duration while the left and right videos for relatively smaller duration.

Suppose the length predicted for the current class of the videos is L_e . To accommodate the quality of the selected video in length calculation, we define bonus length L_b . Suppose S_1^f is the final combined score of the best camera and S_2^f of the second best camera. Now the length for the currently selected video, L_s , is determined as follows:

$$\mathcal{L}^s = (1 - \varsigma)\mathcal{L}^e + \varsigma \mathcal{L}^b \upsilon \tag{18}$$

where v is the difference of the scores, i.e., v = S_1^f - S_2^f and ζ is weighting coefficient. In our experiments, we found the empirical values of $\mathcal{L}_b$ = 25 and ζ = 0.1 works well. A higher value of ζ will increase impact of the bonus length L b on the selected shot length. In this way, ζ can be manipulated to override the prediction made by learned statistics to select longer or shorter shots of given quality score $\mathcal{S}_f$.

In general cases, camera switching only takes place after the se- lected length of time. MoViMash, however, performs continuous check on occlusion and shakiness every second, and whenever the value goes above threshold (same as the one used in the filtering step), video selection is triggered.

IV. Evaluation

The main goal of the experiments is to show that the proposed framework produces a mashup with better view quality and diver- sity than earlier works. In addition, we also compare

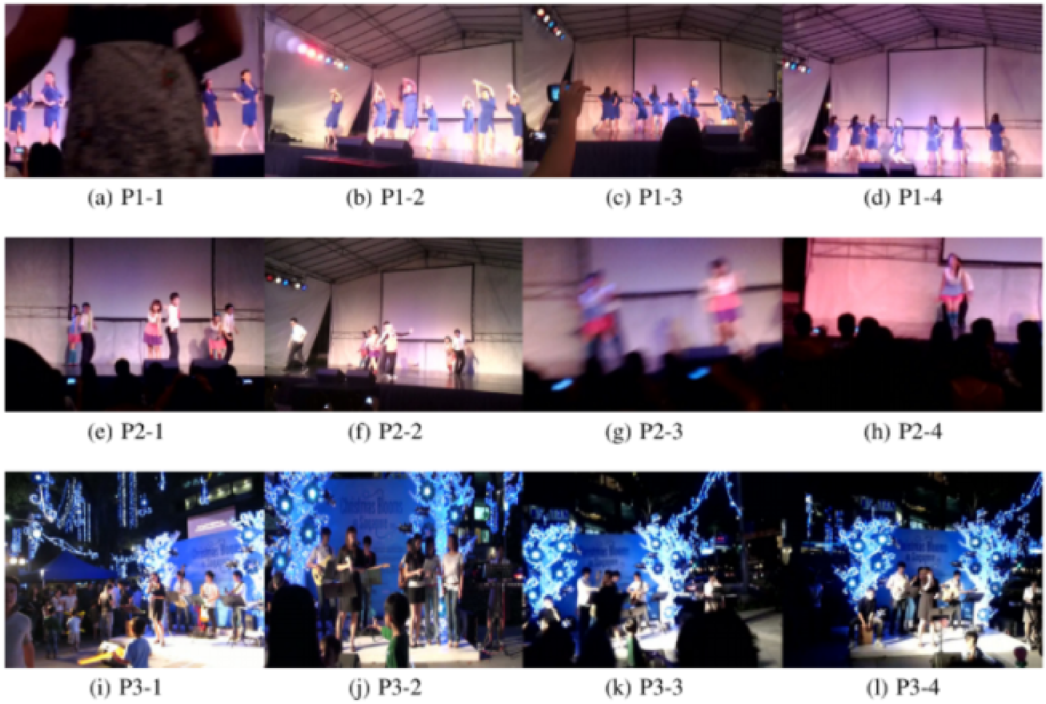


Fig. 6. Caption

our result with human-edited versions of mashups. The dataset consists of video recordings of four performances. For each performance, we cre- ate three mashups: (1) using proposed framework (MoViMash) (2) based on ranking average of shakiness, diversity, and video quality only (VQ-Only) (3) by human editor with editing experience (H- Edited). Users are asked to rate the quality of all three versions of mashup.

A. Dataset

The main application of the proposed framework is to combine the videos recorded by mobile phone users. Therefore, we went to three public performances and handed over smart phones to the au- diences for recording. The audience were given a scenario where they were recording the video for sharing with their friends who were not present at the performance. All the performances hap- pened during the night time. The video clips are converted to a common resolution and synchronized before generating the mashups. In this work, we synchronized the videos manually as our main fo- cus is on video selection. The issue of automatic synchronization is being researched separately [15]. The details of the performances and video recordings are given in the Table 2. Figure 6 shows se- lected frames for each performance.

B. User Study

In the user study, we ask the users to watch the mashups created by the three methods and rate them accordingly. A total of seven- teen users participated in the study, with age range from 20 to 30. The users were mainly graduate students, males and females. The three methods used to create the

TABLE III
USER STUDY QUESTIONS

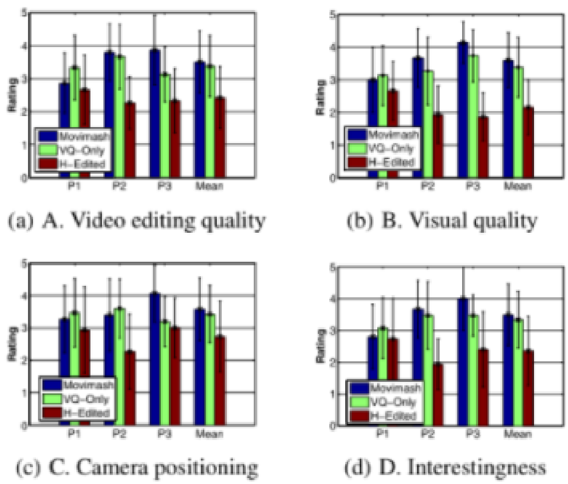
A	Quality of video editing
B	Visual quality of video
C	Camera positioning in the video
D	Interestingness (does video editing makes an interesting overview of the performance?)
E	Overall quality (how likely you will recommend the video to a friend?)

performances to reduce the bias the users may have due to particular presentation order.

The user study was conducted online with relevant instructions explained beforehand. It is told to the users that the main purpose of this user study is to evaluate quality of video editing. Users were asked to rate three distinct performance (P1, P2 and P3). The three mashups (generated by MoViMash, VQ-Only and H-Edited) are juxtaposed on a website with questions below them. The users were allowed to replay the videos while answering the questions. The users rate the videos with a rating ranged between 1 to 5, where 1 represent worst and 5 being the best. The list of questions asked are given in Table 3.

C. Result Analysis

The average responses of the users, along with their standard deviations, are plotted in Figure 7 and Figure 8. We observed re- sponses of the users for Questions A,B,C,D to be similar and with slight variation for Question E.



(a) A. Video editing quality (b) B. Visual quality

(c) C. Camera positioning (d) D. Interestingness

Fig. 7. User responses for questions A, B, C, and D

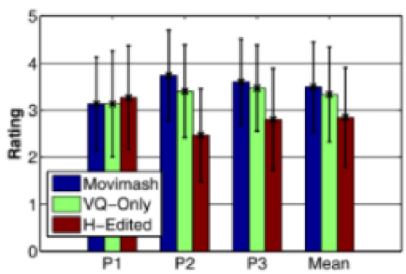


Fig. 8. User response on overall quality of video

1) Analysis of Questions A, B, C, and D: We observed that majority of users responded homogeneously for all four of them. The responses indicated that the ratings were based on some distinct characteristics of the video. Based on additional comments provided by the users, we infer that most of user ratings were based on the quality of the selected video and coverage of stage areas.

For P2 and P3, the users preferred the MoViMash created mashup. For both the performances, users found the shot transitions to be smooth and pleasing. We relate this response of the users to our class-based learning for predicting the shot transitions. The users also remarked that MoViMash produced videos had less occlusions in comparison with VQ-Only based mashup.

While users liked MoViMash created mashups of P2 and P3, they preferred VQ-Only mashup for P1. By analyzing the record- ings and user comments, we found that P1 differs from P2 and P3 in a common aspect. While P2 and P3 had a large number of similar quality videos to select from at each time instant, the videos record- ings of P1 were skewed with respect to quality. There were 2 to 3 videos which were stable and had very high video quality, whereas all other videos were relatively bad in quality. Since MoViMash attempts to maximize diversity along with video quality through class-based shot selection, sometimes poor quality videos are selected.

On the other hand, P2 and P3 had many similar quality videos to choose from, which allowed MoViMash to select videos of dif- ferent classes with smooth transitions. Therefore, in the scenarios where the quality of videos is skewed with only few good quality videos, our MoViMash should be we envision that in future the number of video recording will increase. An increased number of recordings will ensure that there are sufficient number of reason- ably good quality videos to ensure diversity of shots and smooth transitions.

shake, illumination variation etc. are generally negligible in these high quality videos. Therefore, they are generally not comfortable with the videos captured by mobile devices. Further, it would be very difficult for a human editor to evaluate and precisely compare quality of videos, particularly when the number of videos is large. The performance of video editors can get even worse when they have to make video selection in real-time. This finding makes a strong case for automated mashup creation for live applications of video sharing and broadcasting.

2) Overall Video Quality: User ratings for overall video quality are shown in the Figure 8. We can see that on average, MoViMash outperforms other meth- ods in overall video quality. Users generally liked the quality of videos created by MoViMash for all three performances. Further, the user ratings related to MoViMash had the least standard devi- ation among all three mashups. This result implies that the users had highly correlated opinion about the superiority of MoViMash. According to the comments users provided to justify their overall video quality ratings, they liked following things about MoViMash: complete coverage (from many angles), smooth shot transitions, less occlusion, and balanced camera selection. Yet, there might be instances where the funny behavior by crowd, in- teresting gestures and expressions by audiences, etc. It is hard for the proposed automatic system to identify these instances; and a human can be introduced in the selection to improve the mashup quality in such scenarios.

It is important to note that although many users liked VQ-Only created videos more than MoViMash, they did not specify any con- crete aspect they liked about the video except selection of less shaky videos. Thus, for the given dataset, even though VQ-Only method is also able to produce videos with reasonable good quality, it will fail in many real scenarios as it does not have any provision for view quality and smooth shot transitions.

D. Discussion

The MoViMash framework takes automatic mashup creation meth- ods closer to the human editors. It, however, adds to the processing cost. The enhanced diversity model requires more memory for stor- ing the history and more number of pixels in the image. There- fore, individual components of the framework could be enabled or disabled from the stand-mounted cameras, shakiness calculation can be omitted.

Furthermore, to make the system scalable, we can process downscaled images as the video resolution is not critical for the current system components.

V. CONCLUSIONS & FUTURE WORK

In this paper, we have proposed and validated an online video mashup creation framework, MoViMash, for videos recorded by mobile devices. Based on the experiments and user study, we make following conclusions:

- MoViMash creates better quality mashups in comparison to human editor, and other methods that are mainly based on video quality.
- The proposed framework is able to filter occluded and tilted views providing better viewing experience to the users.
- Human editors are not comfortable in editing videos that are recorded by mobile devices, particularly when there are large number of videos with varying quality.
- Proposed diversity model is able to incorporate both tempo- ral and spatial aspects of the selection history.

In the future, we want to extensively evaluate individual com- ponents of the system with respect to end-to-end system delay and output mashup quality. As current to insert artificial zoom and pan in the mashup using zoomable video techniques [11]. Therefore, in the future we want to study the impact of the number of videos on the quality of the final mashup.

VI. ACKNOWLEDGMENT

This research is conducted under the NExT Search Center, sup- ported by the Singapore National Research Foundation and the In- teractive Digital Media R&D Program Office of Media Develop- ment Authority under research grant WBS:R-252-300-001-490.

REFERENCES

[1] http://www.bloomberg.com/news/2011-04-07/high-resolution-cameras-will-drive-mobile-phone-shipments-above-1-billion.html.

[2] http://www.businesswire.com/news/home/20110829005068/en/photobucket-survey-video-uploads-mobile-devices-rise.

[3] M. Al-Hames, B. Hörnler, C. Scheuermann, and G. Rigoll. Using audio, visual, and lexical features in a multi-modal virtual meeting director. In *International Workshop on Machine Learning for Multimodal Interaction*, pages 63–74. Springer, 2006.

[4] M. Campanella, H. Weda, and M. Barbieri. Edit while watching: home video editing made easy. In *Multimedia Content Access: Algorithms and Systems*, volume 6506, page 65060L. International Society for Optics and Photonics, 2007.

[5] F. Crete, T. Dolmiere, P. Ladret, and M. Nicolas. The blur effect: perception and estimation with a new no-reference perceptual blur metric. In *Human vision and electronic imaging XII*, volume 6492, page 64920I. International Society for Optics and Photonics, 2007.

[6] R. Cutler, Y. Rui, A. Gupta, J. J. Cadiz, I. Tashev, L.-w. He, A. Colburn, Z. Zhang, Z. Liu, and S. Silverberg. Distributed meetings: A meeting capture and broadcasting system. In *Proceedings of the tenth ACM international conference on Multimedia*, pages 503–512. ACM, 2002.

[7] E. E. de Lima, C. T. Pozzer, M. C. d'Ornellas, A. E. Ciarlini, B. Feijó, and A. L. Furtado. Virtual cinematography director for interactive storytelling. In *Proceedings of the International Conference on Advances in Computer Enterntainment Technology*, pages 263–270. ACM, 2009.

[8] A. Engstrom, M. Esbjornsson, O. Juhlin, and M. Perry. Producing collaborative video: developing an interactive user experience for mobile tv. In *Proceedings of the 1st international conference on Designing interactive user experiences for TV and video*, pages 115–124. ACM, 2008.

[9] E. Machnicki and L. A. Rowe. Virtual director: Automating a webcast. In *Multimedia Computing and Networking 2002*, volume 4673, pages 208–226. International Society for Optics and Photonics, 2001.

[10] A. K. Moorthy, P. Obrador, and N. Oliver. Towards computational models of the visual aesthetic appeal of consumer videos. In *European Conference on Computer Vision*, pages 1–14. Springer, 2010.

[11] N. Quang Minh Khiem, G. Ravindra, A. Carlier, and W. T. Ooi. Supporting zoomable video streams with dynamic region-of-interest cropping. In *Proceedings of the first annual ACM SIGMM conference on Multimedia systems*, pages 259–270. ACM, 2010.

[12] A. Ranjan, R. Henrikson, J. Birnholtz, R. Balakrishnan, and D. Lee. Automatic camera control using unobtrusive vision and audio tracking. In *Proceedings of Graphics Interface 2010*, pages 47–54. Canadian Information Processing Society, 2010.

[13] A. Senior, A. Hampapur, Y.-L. Tian, L. Brown, S. Pankanti, and R. Bolle. Appearance models for occlusion handling. *Image and Vision Computing*, 24(11):1233–1243, 2006.

[14] P. Shrestha, H. Weda, M. Barbieri, E. H. Aarts, et al. Automatic mashup generation from multiple-camera concert recordings. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 541–550. ACM, 2010.

[15] P. Shrstha, M. Barbieri, and H. Weda. Synchronization of multi-camera video recordings based on audio. In *Proceedings of the 15th ACM international conference on Multimedia*, pages 545–548. ACM, 2007.

[16] J. Wang, C. Xu, E. Chng, H. Lu, and Q. Tian. Automatic composition of broadcast sports video. *Multimedia Systems*, 14(4):179–193, 2008.

[17] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[18] Z. Wang, H. R. Sheikh, and A. C. Bovik. No-reference perceptual quality assessment of jpeg compressed images. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 1, pages I–I. IEEE, 2002.

[19] T. Yang, Q. Pan, J. Li, and S. Z. Li. Real-time multiple objects tracking with occlusion handling in dynamic scenes. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 970–975. IEEE, 2005.

[20] B. Yu, C. Zhang, Y. Rui, and K. Nahrstedt. A three-layer virtual director model for supporting automated multi-site distributed education. In *Multimedia and Expo, 2006 IEEE International Conference on*, pages 637–640. IEEE, 2006.

[21] C. Zhang, Y. Rui, J. Crawford, and L.-W. He. An automated end-to-end lecture capture and broadcasting system. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 4(1):6, 2008.