Automatic Mashup Generation from Multiple-camera Concert Recordings

Prarthana Shrestha, Peter H.N. de With
Eindhoven University of Technology
5600MB Eindhoven, The Netherlands
{P.Shrestha, P.H.N.de.With}@tue.nl

Hans Weda, Mauro Barbieri,
Emile H.L. Aarts
Philips Research Europe
5656AE Eindhoven, The Netherlands
{Mauro.Barbieri, Hans.Weda,
Emile.Aarts}@philips.com

ABSTRACT

A large number of videos are captured and shared by the audience from musical concerts. However, such recordings are typically perceived as boring mainly because of their limited view, poor visual quality and incomplete coverage. It is our objective to enrich the viewing experience of these recordings by exploiting the abundance of content from multiple sources. In this paper, we propose a novel *Virtual Director* system that automatically combines the most desirable segments from different recordings resulting in a single video stream, called *mashup*. We start by eliciting requirements from focus groups, interviewing professional video editors and consulting film grammar literature. We design a formal model for automatic mashup generation based on maximizing the degree of fulfillment of the requirements. Various audio-visual content analysis techniques are used to determine how well the requirements are satisfied by a recording. To validate the system, we compare our mashups with two other mashups: manually created by a professional video editor and machine generated by random segment selection. The mashups are evaluated in terms of visual quality, content diversity and pleasantness by 40 subjects. The results show that our mashups and the manual mashups are perceived as comparable, while both of them are significantly higher than the random mashups in all three terms.

Categories and Subject Descriptors

H.1.2 [**Information Systems**]: User/Machine Systems—*Human factors*; I.2.10 [**Computing Methodologies**]: Vision and Scene Understanding—*Video analysis*; I.4.9 [**Computing Methodologies**]: Image Processing and Computer Vision—*Applications*

General Terms

Design, Algorithms, Human Factors

Keywords

multiple-camera recordings, mashups, user evaluation

MM'10, October 25–29, 2010, Firenze, Italy.

1. INTRODUCTION

During concerts, it has become common for audiences to capture videos using mobile phones, camcorders and digital-still cameras. Some of these videos are uploaded to the Internet contributing to a huge amount of such non-professional recordings. For example, the search phrase "ratm bombtrack pinkpop 2008" submitted to [1] on date 23-02-2010 returned 34 recordings, ranging from 32 seconds to 8 minutes in duration. The recordings, captured simultaneously at the same event and partially overlapping in time, are called *multiple-camera* or *multi-cam recordings*.

The multi-cam recordings provide coverage of the same time and event from different angles, however, they do not provide a nice viewing experience. Watching all these recordings individually takes a long time and it is likely to become boring due to the limited view of a camera, similarity in the content and incomplete coverage. Furthermore, the recordings are likely to contain ill-lit, unstable, and ill-framed images as they are generally captured by hand-held cameras under poor lighting conditions.

It is our objective to enrich the viewing experience of these recordings by exploiting the abundance of content from multiple sources. We propose a novel system called *Virtual Director* that automatically analyzes, selects, and combines audio-visual segments from multi-cam recordings in a single video stream, called *mashup*. Figure 1 illustrates the generation of a mashup by the Virtual Director system. Unlike a summary, which is a temporally condensed form of a recording, a mashup consists of different camera views interleaved in a single video. Depending on the availability of the recordings, a mashup can represent a concert in the same time flow and duration as happened in reality.

The Virtual Director system is meant for non-professionals who have access to multi-cam recordings and like to combine the contents from different recordings. For example, amateur videographers to enhance their personal recording and general video audience to get entertained. The automatic generation of mashup allows a large number of available recordings to be included. In a mashup, the presence of multiple views reduces visual monotony of a single camera recording. Similarly, the signal quality of a mashup can be raised by selecting high quality segments from the available recordings, thereby addressing the typical shortcomings of the non-professional recordings.

1.1 Related work

Previous work on combining multi-cam recordings can be

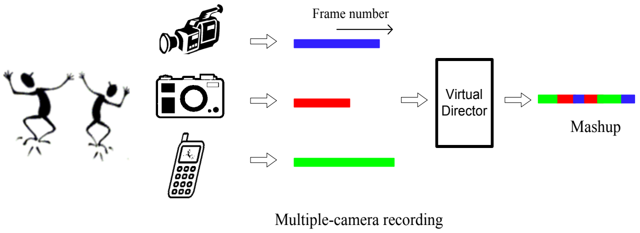


Figure 1: Illustration of a mashup generation system using concurrent recordings from different capturing devices.

broadly classified according to its application purposes into three categories: *video summarization, object reconstruction*, and *video editing*.

A summarization method for multi-cam recordings is presented in [2] for a surveillance system covering a wide area, such as a university campus. The video scenes are assigned an importance score according to the presence of objects of importance, such as humans and cars. Then the high scoring scenes are summarized using 3D graphics. Similarly, multiple perspectives obtained from different recordings are used to create panoramas [3], wide screen movies, and 3D objects [4]. Such reconstructions utilize geometric properties among cameras and objects. In [5, 6], real time video editing systems are presented for multi-cam setups in lecture-hall and meeting-room scenarios, respectively. In both systems a camera is selected based on content, such as speaker recognition and face detection, and user preference.

In the aforementioned works, multi-cam recordings are used in various ways for different applications. No prior work is found on automatic mashup generation from multiple cameras captured by non-professional users, where the environment is uncontrolled and there are no constraints on the number of cameras or their movement.

1.2 Work overview

In this paper, we describe the Virtual Director system for generating mashups from multi-cam recordings of musical concerts, captured by the audience. We start by eliciting a list of requirements obtained from focus groups, by interviewing professional video editors, and by consulting film grammar literature. We propose a formal model for mashup generation, which is based on maximizing the degree of fulfillment of the requirements. The Virtual Director system analyzes the audio-visual features of the recordings and generates a mashup employing the proposed model. The generated mashups are evaluated by means of a user study involving comparisons against two other mashups: manually created by a professional video editor and machine-generated by randomly selecting segments from the recordings. The results show that the perceived quality of a mashup generated by the random method is lower than the mashups created manually and by the Virtual Director, while the perceived quality of the mashups generated by Virtual Director and manual methods are similar.

2. MASHUP GENERATION

2.1 Requirements description

A concert video mashup is aimed to enrich the video experience. An explorative study is conducted to understand the user requirements for a mashup. The study involves focus-group meetings with 18 typical video camera users, interviews of three professional editors, and literature on film grammar and media perception, such as [7] and [8]. The following paragraphs present a list of the requirements obtained from the explorative study.

Requirement 1. **(Synchronization)** The audio and visual streams used in a mashup should be continuous in time. The time delay between audio and video causes lip sync problems and a delay between two consecutive videos causes either repetition or gaps. Therefore, for a complete and smooth coverage of a concert, it is required to find the time displacement among recordings and synchronize them in a common time-line.

Requirement 2. **(Image quality)** A good signal quality is desirable in a video for clarity and pleasure of watching. Since non-professional concert videos are generally captured with hand-held cameras, under poor lighting conditions, it is difficult to continuously capture a high-quality recording. A desired high-quality mashup can be achieved by selecting good quality segments from the multi-cam recordings.

Requirement 3. **(Diversity)** A mashup should offer variety in content and dynamic video experience. For example, if a recording segment contains a close up view of an artist and other two segments from different recordings contain a view towards the same artist and a view towards the audience, a mashup with diversity contains one segment with the artist and the next with the audience. The diversity in a mashup increases the information content and enriches the visual experience.

Requirement 4. **(User preference)** A user may have different personal preferences over different recordings. For example, when a user wants to enhance his own recording by using other recordings, he may prefer to have more of his own recording in the mashup. Therefore, users should be able provide their preference to each of the recordings.

Requirement 5. **(Suitable cut point)** In professional music video editing, the multi-cam recordings are cut into segments according to their visual content and the change in audio tempo. Cuts made at suitable times create an aesthetically pleasing transition among segments. For example, if a cut is made during a camera motion, the viewer perceives it as an abrupt break. Therefore, in a mashup, the segments should be cut in appropriate instants to give smooth transitions among the different recordings.

Requirement 6. **(Semantics)** A concert video is considered more desirable if the audio and the video content match the context, such as close-up view of an artist while singing, faces of the audience while cheering. These features add information and meaning to the content. Therefore a mashup video should contain segments based on semantically meaningful information.

Requirement 7. **(Suitable segment duration)** A video segment becomes incomprehensible if it is too short and becomes boring if it is too long. Therefore, in a mashup the video segments from a camera should be longer than a minimum value ($d_{\min}$) and shorter than a maximum value ($d_{\max}$). In professional music videos, the duration of a segment depends on the music genre.

Requirement 8. **(Completeness)** When a user chooses the recordings to generate a mashup, it is natural to expect all of them to appear in the mashup. In general, a mashup can provide better coverage of a concert by including segments from different cameras because they provide multiple perspectives and more information. Therefore, it is required that all recordings should be represented in a mashup.

Besides the requirements listed above, additional requirements were elicited from the study. The requirements include adding special effects, inserting texts and still-images; applying audio recorded from a professional recording; normalizing color discrepancies among mashup segments; and editing an automatic mashup. These requirements can be applied in a post-processing step on an automatically generated mashup and are not covered in this paper.

2.2 Formal model

In this section we present a formal model for mashup generation, which addresses the different requirements listed in the previous section. The model is used in implementing the Virtual Director system.

2.2.1 Mashup definition

A *mashup* M is an ordered sequence of non-overlapping segments S_i from a set of multi-cam recordings R_j, which consist of time-continuous audio and video frames captured concurrently at the same occasion. The mashup M is specified as:

$$M = (S_1, \ldots, S_l) \ , \tag{1}$$

where l is the total number of segments in a mashup. In a mashup two consecutive segments are acquired from different recordings, hence

$$\forall S \in M \ \exists j, k : S_i \in R_j, S_{i+1} \in R_k, j \neq k \ . \tag{2}$$

The duration of a mashup is determined by the sum of the durations of the individual segments:

$$d(M) = \sum_{i=1}^{l} d(S_i) \ .$$

It is our objective that our mashups satisfy the requirements elicited in Section 2.1 so that the mashups are perceived as a high quality or enriched video.

2.2.2 Mashup generation as optimization problem

The mashup generation problem consists of selecting segments from a multi-cam recording, while satisfying the set of requirements described in Section 2. This problem can be solved by using different approaches such as *rule-based* methods and *optimization* techniques. In the first approach, as described in [9], rule bases are developed, which imitate the mashup generation procedure followed by an expert. For example, if a candidate segment satisfies the diversity requirement, then the method checks if the requirement on image quality is satisfied, else discards the candidate segment. In the optimization-based approach, the overall mashup quality is represented by an *objective function*, which combines the requirements to be addressed in a mashup. The segments for a mashup are selected such that the function is maximized. The rule-based approach is useful in applications where rules can be established from available domain knowledge. However, in the case of mashup generation, most of the requirements represent user preferences rather than strict conditions. Therefore, we select an optimization-based approach given certain conditions.

Requirement 1 (synchronization) represents a strict condition to be fulfilled by a multi-cam recording to be included in a mashup. Therefore, this requirement is addressed prior to optimizing other requirements.

Requirements 2–6 provide user preferences such that the degree of their fulfillment corresponds to the overall mashup quality. We represent these requirements in an objective function $\mathrm{MS}(M)$. For example, Requirement 2 (image quality) is represented by a function $Q(M)$, which gives a score based on the image quality of a mashup. Similarly, Requirements 3 (diversity), 5 (suitable cut-point), 4 (user preference), and 6 (suitable semantics) are represented by functions $\delta(M)$, $C(M)$, $U(M)$, and $\lambda(M)$, respectively.

Requirements 7 (suitable segment duration) and 8 (completeness) are treated as *constraints* that should be complied by the objective function.

The requirements included in the objective function influence the quality of a mashup but their priority order and effectiveness are not known. Therefore, we use a linear approach to combine the functions $Q(M)$, $\delta(M)$, $C(M)$, $U(M)$ and $\lambda(M)$. The objective function can be formalized as:

$$\mathrm{MS}(M) = a_1Q(M)+a_2\delta(M)+a_4C(M)+a_3U(M)+a_5\lambda(M) \ . \tag{3}$$

The coefficients $a_1 - a_5$ are used to weigh the contributions of the different requirements. They allow flexible generation of the mashups by changing the weights of the requirements. The values of the individual functions, such as $Q(M)$ and $\delta(M)$, are computed by averaging the corresponding values of the segments, such as $Q(S)$ and $\delta(S)$, present in a mashup. The following paragraphs describe the modeling of the requirements included in the objective function.

The **image quality** of a video segment is determined by analyzing different low-level video features within frame, such as brightness and between frames such as motion. The image quality of a frame is given by a function $q(f) \to [0, 1]$. For a video segment of a recording, $S = (f_x, \ldots, f_y)$, the image quality $Q(S)$ is represented as the mean quality of the frames present in the segment:

$$Q(S) = \frac{1}{y - x + 1} \sum_{i=x}^{y} q(f_i) \ . \tag{4}$$

The **diversity** in a mashup is measured by the visual distance between two consecutive segments, which is computed as image distance between the last frame of the first segment and the first frame of the second segment. If the function $\psi(f_i, f_j)$ measures the visual distance between two frames, the diversity between two consecutive segments $\delta(S_i, S_{i+1})$ is given by:

$$\delta(S_i, S_{i+1}) = \psi(f_y^j, f_v^k), \text{where}$$
$$S_i = (f_x^j, ..., f_y^j) : f^j \in R_j, \ S_{i+1} = (f_v^k, ..., f_w^k) : f^k \in R_k \ .$$

The **cut-point suitability** of a frame is given by a function $\theta(f) \to [0, 1]$ computed according to the change in audio and visual content along the time. If f_x and f_y are the first and last frames of a segment S, the cut-point suitability score is computed by averaging the suitability scores corresponding to its first and the last frame.

$$C(S) = \frac{\theta(f_x) + \theta(f_y)}{2} \ . \tag{5}$$

The **user preference** score of a recording provided by a user is represented by the function, $u(R) \rightarrow [0, 1]$. If no preference is given by a user, the same score is assigned to all the recordings in a multi-cam recording. The preference score of a segment $U(S)$ is given by:

$$\forall S \in R_j,\ U(S) = u(R_j) \ . \tag{6}$$

The **semantic suitability** can be measured according to the semantic match between audio and video content. The concepts in the audio domain such as guitar, solo, cheering, silence can be linked to the concepts in the video domain such as stage, guitarist, singer, audience. The strength of the link can be used as a measure of their semantic suitability. For example, an audio concept guitar is linked by a higher value to a video concept guitarist than audience.

The computation of semantic suitability score involves advanced audio and video content analysis. We have tested a method, described in [10], to detect audio concepts such as noise, music and silence, but the results are not reliable enough for being applied in mashup generation. State-of-the-art techniques used for audio-video concept detection in concerts, such as [11, 12], show that the problem is highly content dependent and there are too many possible concepts to address in a mashup. Due to the non-availability of a reliable solution, the semantic suitability requirement is not implemented in our system.

The **suitable segment duration** and **completeness** are conditions for the objective function. They are measured in a binary scale representing whether the requirements are satisfied or not. According to Requirement 7, a mashup segment should be longer than a minimum value ($d_{\min}$) and shorter than a maximum value ($d_{\max}$). The values of $d_{\min}$ and $d_{\max}$ are adapted to the audio genre, as commonly used in the professional music-video editing community. For example, the pop music is assigned from 3 seconds to 7 seconds. The constraint is modeled as:

$$\forall S_i \in M : d_{\min} \leq d(S_i) \leq d_{\max} \ . \tag{7}$$

According to Requirement 8, a mashup requires to include segments from all the synchronized recordings of a multi-cam recording. If there are N recordings in a multi-cam recording, the completeness constraint is modeled as:

$$\forall j \in [1, \ldots, N], \exists S_i \in M : S_i \in R_j \ . \tag{8}$$

There might be cases of multi-cam recordings where it is impossible to satisfy the constraints while generating an optimal mashup. For example, the Requirement 7 (suitable segment duration) cannot be satisfied when there is only one recording available for a time duration longer than $d_{\max}$. Similarly, Requirement 8 (completeness) cannot be satisfied when there are too many recordings to fit in the duration of a mashup. In case the constraints cannot satisfied, the system can ask the user for input or continue without satisfying the requirement.

2.3 Virtual Director system

In the previous section, we have modeled the mashup generation problem as an optimization problem, where given a set of synchronized recordings, some requirements are treated as constraints and others as maximization parameters. In this section, we present our Virtual Director system, which applies the described model to generate the mashup. The system is applicable to any multi-cam recordings independent of the number of cameras or the number of camera-takes, which corresponds to the number of times a camera starts and stops capturing during an event. However, for reasons of simplicity in explaining, we consider that every recording in a multi-cam recording consists of one camera-take.

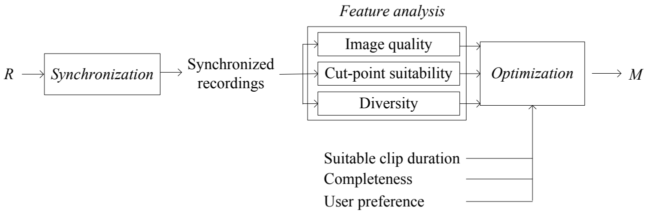


Figure 2: Schematic representation of the proposed Virtual Director system.

2.3.1 Overview

The Virtual Director consists of three processing steps, where each step addresses certain requirements that help in addressing other requirements in successive steps. The three steps are: *synchronization, feature analysis* and *optimization.* The schematic representation of the Virtual Director system is presented in Figure 2.

The first step, *synchronization*, consists of fulfilling Requirement 1. In the next step, *feature-analysis*, audio and video features are extracted and analyzed, to provide numeric values for the functions representing image quality, diversity and cut point suitability given in Equations (4) and (5), respectively. In the final step, *optimization*, we develop an algorithm to maximize the objective function given in Equation (3) and satisfy the constraints. The following sections describe the mentioned three steps.

2.3.2 Synchronization

The Virtual Director system uses an automated synchronization technique to find the synchronization time-offset between the multi-cam recordings based on their audio content. The idea is that during a concert, multiple cameras record the ‘same’ audio at least for a short duration even though they might be pointing at different objects. However, the cameras also record local audio and ambient noise, which makes the raw audio signals difficult to match.

We apply a synchronization method based on audio fingerprinting as described in [13]. Audio-fingerprints are extracted from the recordings and compared to find a match. When multiple matches are found, a voting algorithm is used to compute the most reliable synchronization offset. The method requires a minimum of 3 seconds of common audio between the recordings. It is robust against signal degradations and computes synchronization offsets with a high precision of ± 11.6 ms.

2.3.3 Feature analysis

The Virtual Director system uses audio-visual feature analysis techniques to estimate the degree of fulfillment of the

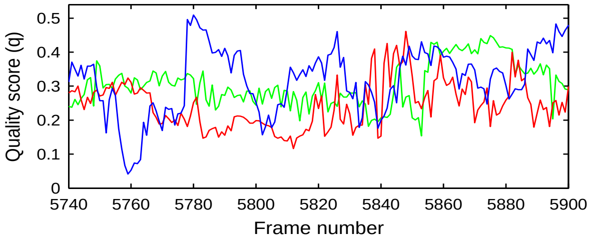


Figure 3: Quality score of recordings from a three-camera recording, given by different colors, in a common time represented by the frame numbers.

Requirements 2 (image quality), 3 (diversity), and 5 (cut-point suitability). The degrees of fulfillment of the requirements are measured in terms of numerical values, called *scores*, where a higher score corresponds to a higher degree of fulfillment. In the following paragraphs, we describe the methods used in feature extraction and computation of the scores.

Image quality estimation: Quality metrics known from video compression like mean square error or peak signal to noise ratio are not applicable to our multi-cam recordings. This is because there is no information available about the actual scene or the camera settings that can be used as a reference for estimating the signal quality. Therefore, we employ a *no-reference*, also called *blind* quality assessment method, which estimates the image quality based on objective measures of different features that influence the perception of quality [14, 15].

We employ the following quality factors: *blockiness, blurriness, brightness,* and *shakiness* to compute the image quality. These metrics address the shortcomings of handheld non-professional cameras, which typically have small lenses, low-cost sensors, and embedded compression with limited quality. The techniques used in estimating the factors are inspired by the prior works in blockiness measurement [14], blurriness measurement [15], and shakiness measurement [16]. The quality factors are given by values between 0 and 1. A multiplicative method is applied to compute the image quality of a frame such that if $\beta_i(f)$ represents a quality factor, the image quality is given by:

$$q(f) = \prod^{i} \beta_i(f). \qquad (9)$$

Figure 3 shows the quality scores of three recordings in a common time line. The quality scores of the recordings vary along the time such that at different intervals segments from different recordings become desirable to be included in a mashup. The performance of these techniques in the context of our concert recordings have been validated by subjective evaluation.

Cut-point suitability estimation: A frame is more favorable as a cut-point if it represents a change in the audio or video. Since a synchronized multiple-camera recording contains the same audio content, the cut-point suitability scores corresponding to audio are valid for all the recordings, where the scores from the video is dependent on individual cameras. In order to find the cut-point suitability score based on video, we use the method described in [16]. The method is based on the speed of the camera motion and change in brightness. A video frame is considered more suitable as a cut-point if it corresponds to the start or end of a camera motion, but not while the camera is moving at a high speed. Similarly, the higher the amount of change in the brightness values along the video frames the higher the cut-point suitability of the frames.

To find the cut-point suitability score based on audio we use tempo or speed of the music. We have tested an algorithm, described in [17] to detect tempo in our test recordings captured by multiple cameras. Since the algorithm was designed for studio-recorded high-quality audio, it failed to provide satisfactory beat and tempo detection on real life non-professional concert audio. Therefore, for our test recordings we manually annotated the cut-points by listening to the recorded music. The manual annotations represent perceptual changes, which last at least for three seconds. Among the synchronized multi-cam recordings, we chose a recording with the available highest quality audio for annotating the audio cut-points. The manual annotation could be avoided by using a high quality recording such as the soundboard recordings from concerts or by designing a robust detector for the beginning and end of a sound or a musical note.

Based on our experimental results in the concert recordings, the cut-point suitability score is calculated as the maximum of the scores from the audio and video, both ranging between 0 and 1. If the cut-point suitability scores of audio and video are represented as c_a and c_v, respectively, the cut-point suitability score is calculated as:

$$\theta(f) = \max(c_a, c_v). \qquad (10)$$

Diversity estimation: In order to fulfill Requirement 3 (diversity), we compute a diversity score between two segments based on their visual difference. The visual difference is measured in terms of image distance between two frames corresponding to two segments. The method is applied extensively for image clustering.

The method measures the distance between two images based on the differences in their corresponding features. The features used in calculating the image distance are *Luma, edges* and MPEG-7 descriptors [18]: *color hue, dominant color, color structure, color layout.* The image distance between two images is computed as a linear combination of the distances between the features. If $\alpha(f)$ represents an image feature of a frame and $\psi(\alpha(f_j), \alpha(f_k))$ represents the distance between two corresponding features from two frames, then the image distance is given by:

$$\begin{aligned} \psi(f_j, f_k) &= \sum_{i=1}^{m} w_i \; \psi\left(\alpha_i(f_j), \alpha_i(f_k)\right), \quad (11) \\ \text{where,} \quad \sum_{i=1}^{m} w_i &= 1 \,, \end{aligned}$$

and m is the total number of features. The weights w indicate the contributions of the different features and their values are chosen on the experimental results obtained from large set of test images.

2.3.4 Optimization

As introduced in Section 2.2.2, the Virtual Director applies an optimization based approach to generate a mashup such that the objective function is maximized and the constraints are satisfied. The values of the image quality score, diversity score and cut-point suitability score are obtained from feature analysis, described in Section 2.3.3.

Given the synchronized multi-cam recordings, the mashup generation problem can be formulated as to maximize:

$$\mathrm{MS}(M) = a_1 Q(M) + a_2 C(M) + a_3 \delta(M) + a_4 U(M) \ , \quad (12)$$

subject to:

$$\forall S \in M : d_{\min} \leq d(S_i) \leq d_{\max} \ , \quad (13)$$
$$\forall j \in [1, \ldots, N], \exists S_i \in M : S_i \in R_j \ . \quad (14)$$

The following paragraphs describe the approach we followed in optimizing the described mashup generation problem.

Normalization: The values of the quality, cut-point suitability and diversity scores represented in $Q(M)$, $C(M)$, and $\delta(M)$ are obtained from feature analysis described in Section 2.3.3. The scores range between 0 and 1, where a higher score represents a higher degree of fulfillment. However, the mean and standard deviation of the three scores are different. As a consequence, the contribution of the different scores in the objective function, which is a linear combination of the scores, is biased by the score type rather than the degree of fulfillment of the requirement. For example, if the mean value of the image quality score is always higher than that of the diversity score, while the standard deviation values of both scores are low, then combining the two scores will always lead to a bigger contribution of the image quality score. Therefore, we have used *Z-score* [19] to normalize scores for every set of multi-cam recording. The resulting scores are given by a common average scale of zero and standard deviation of unity.

The user preference score, ranging between 0 and 1, is assigned by a user to each of the recordings, where a higher score value represents more preference for the recording. If no user input is given, all the recordings are assigned, by default, the score value of 0.5 indicating that the user preferences for all the recordings are equal.

Optimization technique: Optimization techniques are used in problems similar to mashup generation, such as *knapsack problem* and *video summarization*. The knapsack problem requires selecting the best choice of items that can fit into one bag [20] and video summarization requires to select a set of segments from a given video to represent the video in a temporally condensed form [21]. They are typically solved with the complexity in polynomial times by dynamic programming, greedy approach [22] and local search methods. However, the mashup generation problem has additional requirements such as maintaining time continuity between consecutive segments and satisfying the completeness constraint. Similarly, the search space in case of multi-cam recording becomes extremely large compared to a single stream video summarization.

We choose a greedy approach to solve our mashup generation problem, as it is simple to address multiple requirements. The idea is that if an optimal choice is made for every segment, the optimal overall mashup quality is achieved. Therefore, to select a mashup segment, only the segments of the recordings in a given interval are considered disregarding their global characteristics. We have developed an algorithm, called *first-fit* to generate mashups, based on the model formalized in Equations (12) – (14).

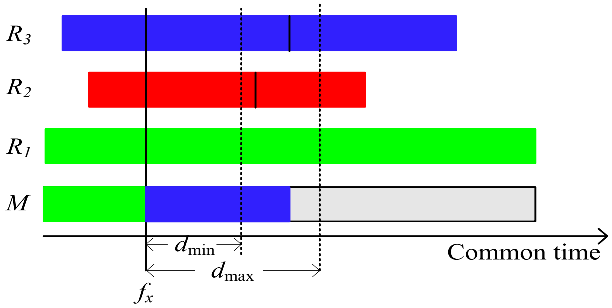


Figure 4: Synchronized multiple-cam recordings ($R_1 - R_3$), represented by different colors, and a mashup (M) in a common time. The duration of a mashup segment ranges between $d_{\min}$ and $d_{\max}$ and two consecutive segments are selected from different recordings.

First-fit algorithm: The algorithm is developed to select segments such that an optimal quality mashup is generated. Firstly, the candidate segments are determined by checking the availability of the recordings for duration $d_{\max}$ from the starting frame of the mashup segment. If there is more than one available recording and the previous mashup segment belongs to one of the available recordings, then the recording corresponding to the previous segment does not qualify for a candidate segment. The length of a candidate segment is selected by choosing a frame with the highest cut-point suitability score, provided the candidate segment is shorter than $d_{\max}$ and longer than $d_{\min}$ Equation (13).

When there is only one candidate segment, the segment is selected as a mashup segment without further calculation. If both the current and previous segments belong to the same recording, then we merge the two segments. In cases when two segments are merged, the duration constraint, given in Equation (13) may be violated. In practice, when no other recordings are available, we allow mashups to contain segments longer than $d_{\max}$. If there is more than one candidate segment, then each of them is evaluated according to the parameters given in the objective function given in Equation (12). The candidate segment corresponding to the highest score is selected as a mashup segment. The search process can be initiated from any point in the common timeline, which may require searching for segments in both forward and backward directions. The algorithm for the backward search requires searching for segments in the duration $d_{\max}$ before the starting frame f_x.

Figure 4 shows selection mashup segments from three hypothetical recordings ($R_1 - R_3$), represented in a common time-line. The first mashup segment is selected from R_1. For the next mashup segment beginning at f_x, the candidate segments are selected from R_2 and R_3. The last frame of the candidate segments, shown by the solid lines on the recordings, is selected according to the highest cut-point suitability score of the frames located in the interval shown by two dotted lines, which ensures that the candidate segments are longer than $d_{\min}$ and shorter than $d_{\max}$. The candidate segments are evaluated according to the parameters given in the objective function. The highest scoring candidate segment, belonging to R_3, is selected to be included in the mashup.

In order to satisfy the completeness constraint, given in

Equation (14), an additional condition is employed in the first-fit algorithm. During the initialization phase of the algorithm, a video frame from each of the recordings, located at least $d_{\max}$ before the last frame of the recording, is set as flagged. It is ensured that no two flags are within the distance given by $d_{\max}$ to avoid two candidate segments containing the flags. The flag is set first in the shortest recording, followed by the longer recordings to give priority to the shorter recordings. A flag is reset when a segment from the corresponding recording is included in the mashup. During the search of a mashup segment, if a candidate segment is encountered with a set flag, this segment is selected without further evaluation. In this way, we ensure that all the given recordings contribute at least one segment to the mashup.

There may be instances where it is impossible to fulfill this constraint due to the input recordings in a multi-cam recording as described in Section 2.2.2. In practice, for all the multiple-camera recordings used in our test, we have been able to satisfy the completeness criteria. In the present implementation, if a constraint is not satisfied, the mashup is considered invalid. A better approach would be to involve users in the mashup generation process such that if the constraint is not met, the users are notified. Depending on the user response, the mashup is created ignoring the criteria or recalculated.

3. MASHUP EVALUATION

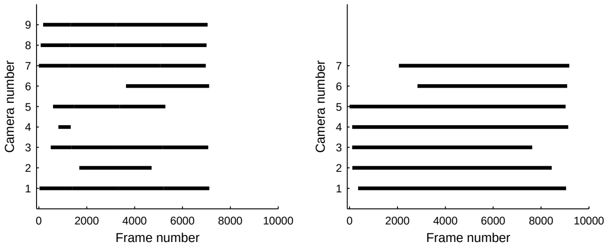
3.1 Test Design

In order to evaluate how well the mashup requirements are satisfied by the first-fit algorithm of the Virtual Director system, we compare the quality of the mashups generated by the first-fit algorithm with two other methods: *naive* algorithm and *manual*. In the following sections we describe the methods and their mashup qualities compared to that of the first-fit algorithm.

3.1.1 Naive mashup

The naive algorithm is designed to generate a mashup that fulfills the constraints given in Equations (13), and (14) derived from Requirements 7 (suitable segment duration) and 8 (completeness), respectively. No other requirements are considered during the mashup generation.

The naive algorithm generates a mashup as follows. The starting point of the segment selection is always the very first frame on the common time-line. The available recordings are searched within the given maximum segment duration $d_{\max}$. If there is more than one available recording for the candidate segments and the previous mashup segment belongs to one of the available recordings, then the recording corresponding to the previous segment does not qualify to be included as a candidate segment. If there is only one available recording, it is selected as the mashup segment. However, if multiple recordings are available, one of the recordings is selected randomly. Once a recording is selected, the last frame of the segment is selected randomly among the frames that are located between $d_{\min}$ and $d_{\max}$ from the starting frame of the segment, such that the segments are within the suitable segment duration, given in Equation (13). The algorithm assures that a mashup contains at least one segment from each of the recordings, to satisfy completeness constraint, given in Equation (14), by using flags as in the first-fit mashup generation algorithm.



(a) (b)

Figure 5: Two multi-cam recordings used as test set for evaluating *naive*, *manual* and *first-fit* mashups. The recordings are represented by the bold dark horizontal lines in a common time. The camera numbers represent the index of the recordings.

Table 1: Test set for mashup evaluation. The camera numbers of concerts C1 and C2 correspond to the recordings, which are also shown in Figure 5.

Concert	Camera #	Venue	Genre
C1 (Figure 5(a))	1, 3, 7, 8, 9	Indoor	Pop
C2 (Figure 5(b))	1, 2, 3, 4, 5	Outdoor	Rock
C3	1, 2, 3, 4	Indoor	Pop

3.1.2 Manual mashup

The manual mashups are created by a professional video editor. The synchronized multi-cam recordings were provided to the editor, who was asked to create a mashup without adding any special effects and following the time-line of the content. It took approximately 16 hours to create 3 mashups from the given test set, using commercially available multi-cam editing software [23]. The considerable time and effort required for creating manual mashups forced us to limit the size of the test set.

3.1.3 Test set

In order to compare the quality of the mashups generated by the different methods, we used 3 multi-cam recordings, which are captured during concerts by non-professionals and shared in [1]. Each of the multi-cam recordings contained 4 to 5 recordings with both audio and video streams. Figure 5 shows the duration and the time overlap among all the available recordings from the two concerts used in the test set. Table 1 shows the details of the test set, which are also made available online [24]. In all concerts the duration of the recordings is between 2.4 and 5.6 minutes and their frame rate is of 25 frames per second. The video resolution is 320×240 pixels.

The first-fit and naive algorithms use minimum and maximum segment durations of 3 and 7 seconds, respectively. The durations are based on a common practice in amateur video editing with rock-pop music genre. The image quality, diversity and video cut-point suitability scores are measured according to the methods described in Section 2.3.3.

To have an objective comparison among the algorithms, no user preference scores are provided to the recordings in the first-fit algorithm. Therefore, we ignore the preference score from our implementation of the objective function. The coefficients $a_1 - a_3$ in the objective function, given in Equation (12), are set to be equal to $\frac{1}{3}$.

The analysis of the mashups generated by the different methods according to the objective function shows that the overall scores of the first-fit mashups are at least 10 times higher than the naive mashups and also slightly higher than the manual mashups. The lowest score of the naive mashups is expected as they address fewer mashup requirements. However, we expect that the manual mashup should score the highest as human editor can better understand the requirements than a model. Further analysis of the manual mashups show that the weights applied to requirements are inconsistent in the three concerts. Since both the manual and first-fit algorithms may have addressed different requirements or with different importance, it is difficult to compare the overall mashup quality between the two methods in a limited test set by means of an objective evaluation. Therefore, we conducted an user test to measure the end user satisfaction provided by mashups generated by the different methods.

3.2 Hypotheses and operationalization

The goal of the user test is to compare the perceived quality of mashups generated by different methods. Ideally, we expect the perceived quality of a mashup to be in the ascending order: naive, first-fit, and manual. Therefore, the formulated hypotheses for the test are:

H1: The perceived quality of a mashup generated by the first-fit algorithm is higher than that of one generated by the naive algorithm.

H2: The perceived quality of a manually made mashup is higher than that of one generated by the naive algorithm.

H3: The perceived quality of a manually made mashup is higher than that of one generated by the first-fit algorithm.

Perceived quality of a mashup is an abstract concept and we need to define it in terms of measurable factors (operationalization). According to the mashup requirements described in Section 2.1, we operationalized the perceived mashup quality into three factors: *diversity, visual quality,* and *pleasantness.* A *high-quality* mashup should score high on all these factors. The factors are further divided into different parameters based on the keywords used to associate them by the participants of the focus-group meetings.

Diversity in a mashup is represented in terms of the following parameters: *atmosphere, overview*, and *content variety.* Atmosphere signifies the mood during the concert such as dull, enjoyable and wild. Overview signifies the physical settings, such as location, audience size, and content variety signifies richness in the content.

Visual quality depends on different criteria such as edge blur, brightness and noise, which are difficult to differentiate and evaluate for a general user. Therefore, we select two easily perceivable parameters: *image quality* and *camera stability.* They represent the spatial and temporal visual quality of a mashup.

Pleasantness is an experience we aim to achieve while watching a mashup. We represent pleasantness in terms of the following parameters: *boring, overall goodness*, and *entertaining.* A boring mashup is considered the opposite of entertaining, which could be due to the failure of many requirements such as image quality, diversity, suitable cut-point and suitable segment duration. An overall goodness signifies that the mashup fulfills the requirements at a satisfactory level.

3.3 Experiment

3.3.1 Procedure

The test contains two independent variables: *algorithm*, which refers to the naive, manual, and first-fit methods for generating mashups, and *content*, which refers to the concerts C1, C2 and C3, as shown in Table 1.

The test is designed as *full-factorial within-subject* such that all the nine mashups (3 algorithms × 3 concerts) are evaluated by every participant. The advantages of this design are that all the main effects of the variables and their interactions can be estimated with a limited number of participants and their interpersonal differences do not influence the evaluation. The presentation order of the mashups is arranged such that different participants view the mashups in different order, while no two consecutive mashups are shown from the same concert.

A questionnaire is designed to measure the mashup quality in terms of diversity, visual quality and pleasantness. The participants are presented with 9 statements corresponding to the operationalized parameters described in Section 3.2. The statements are: 'I got a good impression of the concert atmosphere from the mashup video'; 'The different viewpoints shown in the video gave me a rich overview of the concert'; 'I got disturbed by the lack of camera stability'; 'I found this video entertaining'; 'I think there was enough variety in the content'; 'I found the video boring to watch'; 'The cameras in the video were stable'; 'The image quality in the video was very bad'; 'Overall, I think the video was good'. We arranged the order of the statements such that questions that can be interpreted as very similar, do not appear consecutively to avoid answering on memory. Some statements look similar, but are in fact included to understand both the subjective and objective measure of a factor, such as such as shakiness in a video and feeling disturbed due to shakiness. After watching a mashup, the participants *rate* it by indicating the level of agreement or disagreement to the statements. We use a seven-point Likert scale, used extensively in perceptual tests, to measure the level of agreement.

Additionally, the perception of a mashup quality may be influenced by factors such as: participants' age and gender; preference toward an artist or genre; frequency of concert visits and watching concert videos. To check the effects of these factors on the mashup quality, related questions are asked to the participants.

3.3.2 Participants

Forty participants (17 female, 23 male) volunteered in the test. The average age of the participants is 27 (min: 22, max: 34). The average number of times they had been to a concert in the last two years is 3 (min: 1, max: 6). Among the participants, the frequency of watching concert videos is distributed as: 10% never, 45% rarely (less often than once a year), 32.5% monthly, 10% weekly, and 2.5% daily.

3.4 Results

The user ratings on the nine statements corresponding to a mashup are analyzed to measure the perception of the quality parameters: *diversity, visual quality* and *pleasant-*

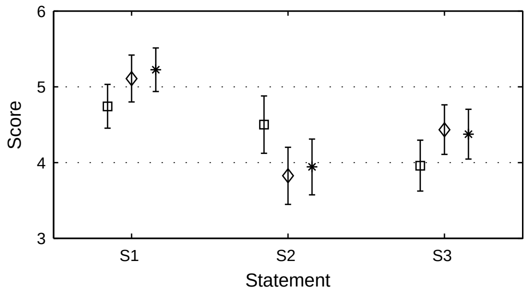


Figure 6: Mean scores across the *content* for naive (□), manual (◇) and first-fit (∗) algorithms on the statements: 'I got a good impression of the concert atmosphere from the mashup video' (S1),'the image quality in the video was very bad' (S2) and 'overall, I think the video was good' (S3). Error bars show confidence intervals of 95% of the mean value.

ness described in Section 3.2. The correlation measurement, using standard Cronbach's α, among the statements corresponding to a quality parameter, results in $\alpha > 0.81$ in all three cases. This high value of α suggests that the statements agree on the measure of the parameters. Therefore, we present the analysis of three representative quality factors measured by the following statements: (S1) 'I got a good impression of the concert atmosphere from the mashup video', (S2) 'the image quality in the video was very bad' and (S3) 'overall, I think the video was good'.

Figure 6 shows the average response value or mean score of the *algorithm* across all the participants and *content*. The confidence intervals are presented graphically as an error bar on the mean score. The intervals indicate the reliability of a mean score, such that if the test is repeated with other participants from the same target group, there is 95% probability that the mean score will remain within the interval.

The mean scores for the statement S1 in Figure 6 shows that the first-fit mashups are slightly higher than the manual mashups, which are higher than the naive mashups. From the ANOVA analysis, a significant main effect is found for *algorithm* ($F = 3.119,\ p = 0.045$) and also for *content* ($F = 6.398,\ p = 0.002$). A Tukey test on *algorithm* shows that there is a significant difference between the scores of the naive and first-fit, while the scores of the manual is not significantly different from both. The slight, but non-significant, higher mean score of the first-fit mashups over the manual mashups can be explained by the opinion of some of the participants, that the manual mashups are mainly focused on the artists, which limited their perception of the concert atmosphere. Similarly, a Tukey test on content shows that C2 scores significantly higher than C1 and C3. Since C2 was held on a large open space during daylight hours, the recordings contain different views like stage, audience, and display boards, the recording provides more variety in content than the two other indoor concerts containing mainly the stage.

The mean scores for the statement S2 in Figure 6 show that the naive mashups are higher than the other two mashups, whose values are about the same. From the ANOVA analysis, a significant main effect is found for *algorithm* ($F = 7.833,\ p < 0.001$) and for *content* ($F = 16.051,\ p < 0.001$). A Tukey test on *algorithm* shows that naive is significantly different from the other two algorithms. This result is expected as the naive algorithm does not take image quality into account. Similarly, a Tukey test on *content* shows that C3 is significantly different from C1 and C2, which corresponds with the low image quality of the recordings of C3.

The mean scores for the statement S3 in Figure 6 show that the manual and the first-fit mahsups are about the same, while both appear to be higher than naive mashups. From the ANOVA analysis, no significant main effect is found for *algorithm* ($F = 2.271,\ p = 0.071$). However, a significant main effect is found for *content* ($F = 8.993,\ p < 0.001$). A Tukey test on *content* shows that concert C1 is significantly different than C3. The result indicates that the perception of pleasantness depends more on the content than the algorithm.

The analysis of the control questions, such as age and liking the artist, described in Section 3.3.1 shows that there are additional variables to *algorithm* and *content* that influence a mashup quality. The perception of diversity is found to be influenced by the age factor, such that younger participants (25 and younger) found that the mashups contained less diversity than older participants (31 and older). The expectation of the younger group corresponds with the current trend in professionally produced music videos, where the density of shot-cuts and fast transitions has increased tremendously [7]. Similarly, it is found that the perception of visual quality is influenced by the frequency of watching concert videos. People who watch concert videos on weekly or daily basis are more critical towards image quality than people who rarely watch such videos. The perception of pleasantness is found to be dependent on liking an artist (or genre). People who like an artist (or genre) find the mashups containing the artist (or genre) more pleasant than people who do not like one.

3.5 Discussion

The test results show that the perceived quality, in terms of diversity, visual quality and pleasantness, of a mashup generated by the naive algorithm is consistently lower than that of the ones generated by the manual and first-fit algorithms. Therefore, the hypothesis **H1** and **H2**, described in Section 3.2 are confirmed. Between the first-fit and manual mashups, the first-fit scores slightly higher in diversity but slightly lower in visual quality. The pleasantness scores of both algorithms are very similar. Therefore, hypothesis **H3** is not confirmed.

The perception of a mashup quality is highly dependent on the content. The camera recordings with multiple view angles, variety in content and good visual quality allow the manual and first-fit algorithms to generate mashups that are perceived as significantly higher in quality than that of the ones generated by the naive algorithm.

4. CONCLUSION

In this paper we have presented an automated mashup generation system for multi-cam recordings captured during musical concerts by non-professionals. Our objective is to enrich the viewing experience of such recordings, which are generally incomplete, low-quality and boring to watch. We have elicited the requirements for a concert video mashups from 18 users involving professional video-editors, amateurs and multimedia researchers, based on focus-group meetings and interviews. We have proposed a formal model for mashup generation, that represents different requirements in a global objective function. We have developed a Virtual Director system that synchronizes the multi-cam recordings, measures the degree of fulfilment of the requirements and generates a mashup using an algorithm, called *first-fit*, which maximizes the proposed objective function. We have evaluated the overall quality of the mashups generated by the Virtual Director system using the first-fit algorithm with respect to the ones generated by two other methods: *naive* and *manual*. The test set includes mashups from three typical concerts, each containing 4–5 camera recordings. The analysis of the mashups generated by the different methods, according to the objective function, shows that the overall scores of the first-fit mashups are at least 10 times higher than the naive mashups and also slightly higher than the manual mashups. However, the objective evaluation of the mashups cannot be validated in the given test set. The size of the test set is too limited due to the complex and time consuming process of creating manual mashups. To assess the end-user satisfaction, we have conducted a user test with 40 subjects. The participants have rated the mashups via a questionnaire, which is designed to evaluate the mashup quality in terms of three aspects: *diversity*, *visual quality* and *pleasantness*. The results show that the naive mashups score consistently and significantly lower than the other mashups in all the aspects. In comparison to the manual mashups, the first-fit mashups scores slightly higher in diversity but slightly lower in visual quality, while both of them score similar in pleasantness. Therefore, we conclude that the perceived quality of a mashup generated by the naive method is lower than the first-fit and manual, while the perceived quality of mashups generated by the first-fit and manual methods are similar.

5. ACKNOWLEDGEMENTS

The work was carried out at Philips Research Europe, Eindhoven, with partial funding of the Dutch BSIK research program MultimediaN. We are thankful to Martin McKinney, Jaap Haitma, Janto Skowronek, Marc Peters and Marco Campanella for providing softwares for audio-visual feature analysis.

6. REFERENCES

[1] Online, "YouTube," `http://www.youtube.com/`.

[2] T. Hata, T. Hirose, and K. Tanaka, "Skimming multiple perspective video using tempo-spatial importance measures," in *VDB 5: Proc. of the 5th Working Conf. on Visual Database Systems*, 2000, pp. 219–238.

[3] H. S. Sawhney, S. Hsu, and R Kumar, "Robust video mosaicing through topology inference and local to global alignment," *Book Series Lecture Notes in Computer Science., Vol. 1407*, 1998.

[4] S. N. Sinha and M. Pollefeys, "Visual-hull reconstruction from uncalibrated and unsynchronized video streams," in *Proc. of the 3D Data Processing, Visualization, and Transmission*, 2004, pp. 349–356.

[5] F. Lampi, S. Kopf, M. Benz, and W. Effelsberg, "A virtual camera team for lecture recording," in *IEEE Multimedia, Vol. 15, Num. 3*, 2008, pp. 58–61.

[6] S. Stanislav, "Multi camera automatic video editing," in *Proc. of ICCVG 2004*, 2004, pp. 935–945.

[7] B. Reeves and C. Nass, *The media equation*, CSLI Publications, Cambridge University Press, 1996.

[8] H. Zettl, *Sight Sound Motion : Applied Media Aesthetics*, Wadsworth Publishing, 2004.

[9] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, 2002.

[10] J. Breebaart and M. McKinney, "Features for audio and music classification," in *Proc. of Int. Symp. on Music Information Retrieval*, 2003.

[11] C. G. M. Snoek, M. Worring, A. W. M. Smeulders, and B. Freiburg, "The role of visual content and style for concert video indexing," in *Proc. of the Int. Conf. on Multimedia & Expo (ICME)*, 2007, pp. 252–255.

[12] U. S. Naci and A. Hanjalic, "Intelligent browsing of concert videos," in *Proc. of the 15th ACM Int. Conf. on Multimedia*, 2007, pp. 150–151.

[13] P. Shrestha, H. Weda, and M. Barbieri, "Synchronization of multi-camera video recordings based on audio," in *Proc. of the 15th ACM Int. Conf. on Multimedia*, 2007, pp. 545–548.

[14] Z. Wang, H. R. Sheikh, and A. C. Bovik, "No-reference perceptual quality assessment of jpeg compressed images," in *Proc. of Int. Conf. on Image Processing, Vol. 1*, 2002, pp. 477–480.

[15] E. Ong et al., "A no-reference quality metric for measuring image blur," in *Proc. of 7th Int. Symp. on Signal Processing and Its Applications, Vol. 1*, 2003, pp. 469 – 472.

[16] M. Campanella, H. Weda, and M. Barbieri, "Edit while watching: home video editing made easy," in *Proc. of the IS&T/SPIE Conf. on Multimedia Content Access, Vol. 6506*, 2007, pp. 65060L–1 – 65060L–10.

[17] J. E. Schrader, *Detecting and Interpreting Musical Note Onsets in Polyphonic Music*, M.Sc. Thesis, Eindhoven University, The Netherlands, 2003.

[18] MPEG, "ISO/IEC 15938-8 Multimedia content description interface-part 8: extraction and use of MPEG-7 descriptors," 2002.

[19] R. J. Larsen and M. L. Marx, *An Introduction to Mathematical Statistics and Its Applications*, Prentice Hall, 3rd edition, 2000.

[20] S. Martello and P. Toth, *Knapsack problems: algorithms and computer implementations*, J. Wiley and Sons, 1990.

[21] M. Barbieri, *Automatic summarization of narrative video*, Ph.D. Thesis, Eindhoven University, 2007.

[22] H. Cormen, C. H. Leiserson, R. L. Rivest, and C. Stein, *Introduction To Algorithms*, MIT Press, 2001.

[23] Online, "Adobe Premiere Pro," `http://www.adobe.com/products/premiere/`.

[24] YouTube, "Test data-set used in mashup quality evaluation," 2009, `http://www.youtube.com/AutomaticMashup#play/uploads`.