3D Interest Maps From Simultaneous Video Recordings

Axel Carlier
Université de Toulouse
axel.carlier@irit.fr

Lilian Calvet
Simula Research Laboratory
lcalvet@simula.no

Duong T. D. Nguyen and Wei Tsang Ooi
National University of Singapore
{nguyend1, ooiwt}@comp.nus.edu.sg

Pierre Gurdjos and Vincent Charvillat
Université de Toulouse
{gurdjos,vcharvillat}@irit.fr

ABSTRACT

We consider an emerging situation where multiple cameras are filming the same event simultaneously from a diverse set of angles. The captured videos provide us with the multiple view geometry and an understanding of the 3D structure of the scene. We further extend this understanding by introducing the concept of *3D interest map* in this paper. As most users naturally film what they find interesting from their respective viewpoints, the 3D structure can be annotated with the level of interest, naturally crowdsourced from the users. A 3D interest map can be understood as an extension of saliency maps in the 3D space that captures the semantics of the scene. We evaluate the idea of 3D interest maps on two real datasets, taken from the environment or the cameras that are equipped enough to have an estimation of the poses of cameras and a reasonable synchronization between them. We study two aspects of the 3D interest maps in our evaluation. First, by projecting them into 2D, we compare them to state-of-the-art saliency maps. Second, to demonstrate the usefulness of the 3D interest maps, we apply them to a video mashup system that automatically produces an edited video from one of the datasets.

Categories and Subject Descriptors

H.5.1 [**Multimedia Information Systems**]: Video

1. INTRODUCTION

The proliferation of video capturing devices has lead to an explosion of user-generated video content. In this paper, we consider the case of public performances (music, dance, sport, magic, theater etc.) with a spectating crowd that film the performance with either their mobile devices or a static camera. This behavior is particularly interesting as multiple videos of the same scene are captured from multiple diverse viewpoints.

Figure 1 shows an example of the scenario we consider, where an open air performance has six cameras surrounding the stage at ground level (labeled 1 to 6) and one camera filming the performance from above with two zoom levels (labeled 7 and 8). We can treat the scene being filmed as a type of 3D visual content, with the videos of this scene filmed from different viewpoints containing, collectively, a large part of this 3D content. If we know the parameters of the cameras filming the scene, the 3D structure can be partially reconstructed. But what is more interesting is that what users choose to film in a scene signals the interestingness of the region in 3D space that is being filmed. When a videographer selects his zoom level or his preferred framing, he implicitly reveals the interestingness of the scene he films. For example in Figure 1, the user who is shooting with the top camera (at zoom level 7 represented by the red rectangle) is obviously interested in the stage, because he zoomed in towards it. By generalization, several videographers collectively reveal the most interesting 3D parts of the scene where many view frustums intersect. Thus, it is feasible to label the reconstructed 3D structure with a level of interest. This idea led us to introduce a new concept called *3D interest map* in this paper.

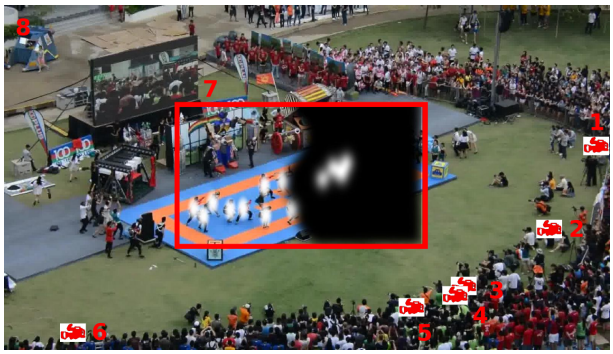

Figure 1: A dance performance from RagAndFlag dataset. Cameras 1 to 6 are located around the stage at ground level. Camera 7 shoots this picture (zoom position 8) and occasionally zoom into the red rectangle (zoom position 7). A 3D interest map is drawn in white in the 3D space and also in the 2D image space after re-projection of the 3D map (with black background).

Intuitively, a *3D interest map* represents some hot spots as illustrated in white in Figure 1 and explained by a distribution of interest with several modes or peaks (adjusted to the locations of human performers in this example). Note that the 3D interest maps can also been re-projected into 2D frames and become a standard (2D) saliency maps (as shown on the black and white region of Figure 1), but with the additional advantage that salient regions are consistent between videos filmed from different angles. Estimating such a multi-modal distribution from several synchronized videos is the main contribution of this work.

The constructed 3D interest maps are useful in many applications. We focus on the application of automated video authoring in this paper. In particular, we use 3D interest maps as additional in-

MM '14 Orlando, USA
ACM 978-1-4503-3063-3/14/11 ...$15.00.
http://dx.doi.org/10.1145/2647868.2654947.

put to a video mashup system that automatically switches between input video streams to produce a single temporally coherent video. The use of 3D interest maps allows the algorithm to exploit interestingness information for close up shots and to generate virtual camera movements. Furthermore, the availability of information *across* videos allows using the system to maintain coherence when it cuts from one camera to another. We show that 3D interest maps lead to better mashup videos. Our improvement of video mashups using 3D interest maps is the second contribution of this paper.

The paper is organized into seven sections. We first review the related work in Section 2. We then formally define the concept of 3D interest maps in Section 3, before explaining our algorithm to compute 3D interest maps in Section 4. Section 5 demonstrate the usefulness of 3D interest maps by applying them to a video mashup system. The experiments and evaluation results are presented in Section 6. Finally, the paper concludes in Section 7.

2. RELATED WORK

Our literature review is organized into three subsections. We first position our 3D interest maps in relation to other attention models. Then, we explain how the proposed approach relates to crowdsourcing, before reviewing several systems similar to ours.

Salient Region Detection. Salient regions or objects from any given image or video direct human attention by definition. In their recent survey [3], Borji and Itti discuss the most important visual attention models that compute saliency maps. In his pioneer work, Itti [15] devised the first bottom-up visual saliency model using differences across multi-scale image features. More recently, Goferman et al. [12] combine local low-level information, higher level features, and a more global spatial context to detect salient objects. This bottom-up detector, along with a few others, have been shown to be effective [4]. We use this detector in our work as a reference for some of our experiments. Top-down models for saliency detection can be even more effective in specific environments, but they usually require a costly learning stage [2].

The temporal dimension is a key factor when detecting dynamic salient regions in video sequences. For a given frame, the most salient regions should be selected from both the current image saliency map and the previous detections [17]. Further, it is natural to predict salient regions from motion maps in a video.

In the context of 3D visual content, Dittrich et al. [9] propose an approach for saliency detection in stereoscopic videos based on color, motion, and disparity. The last factor is used as regions that pops up in the scene tend to catch users' attention.

This existing work aims to determine the salient regions from low-level features of visual content. A salient region, however, does not necessary imply that it is interesting or important, two attributes that require an understanding of the semantics of the content and the users' intentions and are difficult to infer from features alone. Our approach, however, accounts for user intention which are implicitly indicated by the scene they are filming.

Crowdsourcing users' interest. To address the semantic gap, several crowdsourcing-based methods have been proposed to identify interesting regions in images and videos. The idea is that, by polling (either implicitly or explicitly) from multiple users what they find interesting, we can discover the interestingness of a region without needing to explicitly infer the semantic of the content.

In the context of games with a purpose (GWAPs) and human computation, Huang et al. [14] developed a collaborative game, called Photoshoot, for extracting image ROIs from shoot points that serve in the role of gaze points. Crowdsourcing users' interest can be also done more implicitly. As an example, Xie et al. [31] considered a mobile use-case with image browsers, where users tend to zoom and scroll to view interesting regions in detail on a small screen. By crowdsourcing these actions, a ROI extraction model can produce user interest maps. This approach, using implicit feedback while users are browsing still images, has been extended to infer moving ROIs based on video browsing behaviors [5, 6]. In this work, the users naturally zoom into the video and reveal ROIs. In our work, we consider that users around the scene are also implicitly interacting with it by filming the scene; a user filming a 3D region is implicitly designating this region as interesting.

Applications based on simultaneous video capture.

Multiple videos captured simultaneously can be automatically edited to produce a new *mashup* video similarly to how a human director would switch between different cameras to produce a show. Shrestha et al. [26] address this problem, and select the cameras by maximizing video quality. They also explain in [25] how to synchronize multiple cameras from the available multimedia content. Saini et al. [23] expand this effort on video mashups with live applications in mind. The system decides which camera to switch to without any reference to future information. Their improved model jointly maximizes the video signal quality, but also the diversity and quality of the view angles, as well as some aesthetic criteria that include shot lengths and distances from the stage. We explain in Section 5 how 3D interest maps make it possible to improve these mashups by introducing automatic zooming and panning on the frame, as well as smoothing the transitions between shots, thanks to 3D information that relates the content of one video with another.

3. WHAT IS A 3D INTEREST MAP?

Understanding and predicting user interest is a key issue in many multimedia applications. In order to efficiently compress, store, retrieve, transmit, recommend, display any media item, it is helpful to estimate users' level of interest in it.

Beyond saliency. In some situations, user interest and attention are guided by a task to be performed; in other free-viewing situations, a user may subjectively gaze at most interesting items. In the latter case, a saliency map associated with an image often serves as a gaze predictor and can be interpreted as a 2D map with a type of "interest level" (preferably normalized, between 0 and 1) assigned to each pixel. Unfortunately 2D saliency models still fall short of predicting semantically interesting objects perfectly. With 3D content in mind, one can formally define a 3D saliency map by generalization, and determine saliency at a given 3D voxel by considering how different a given voxel is from its surroundings in color, orientation, depth etc. Similar to many 2D saliency models, this generalized 3D saliency estimation does not integrate any semantics and is a poor interest predictor for a 3D scene like the one depicted in Figure 2. In this example, a semantic definition of the interest (in 3D) would probably assign the highest interest level to the voxels located on the soloist in front of the band, an intermediate interest value to voxels located on others musicians, and low interest to voxels located on the background buildings. We devised our 3D interest maps with this objective in mind.

Our 3D interest maps are innovative for several reasons. First, we infer 3D interest information from the multiple view geometry of simultaneous video recordings: it is an original 3D generalization of 2D saliency maps. Indeed, a 3D interest map is richer than several saliency maps computed from the original video recordings: as shown later, the correspondences between salient areas seen in range views are naturally established when re-projecting the components of the 3D interest maps. Another important difference with traditional saliency models is that we do not need to predict where the spectators would gaze – we can simply observe it. We can see a

user's camera as a third eye that he focuses towards the parts of the scene that he is most interested in. Our 3D interest levels are somehow "crowdsourced" from many videographers and we assume that the semantically important portions of the scene can be statistically revealed by analyzing the selected view frustums (originated by zoom, pan and framing decisions). If many videographers focus towards "the most interesting" musician (e.g., the soloist) shown in Figure 2, a bit of semantically motivated interest is revealed.

A formal approach to 3D interest maps. We define a 3D interest map with respect to a given time instant, and hence we consider a set of J images corresponding to synchronized video frames[1] taken from J cameras capturing the same scene from different angles at this instant. We assume that the camera projection matrices $\mathtt{P}^j$, $j = 1..J$, with respect to some Euclidean projective representation, are given.

We also assume that we have the 2D regions of interest (ROI) in the J range views at our disposal. Note that the simplest user's ROI in a view can be drawn from the central fixation assumption [30], in which it corresponds to an elliptic region centered at the principal point (*i.e.*, at the point where the optical axis meets the image plane). Such elliptic regions are shown in blue in Figure 2 for two cameras. Although this assumption will be relaxed later, it makes sense as it seems natural that the user tends to shoot by steering the camera optical axis towards the scene hot spots.

Figure 2: Central focus assumption on the BrassBand dataset.

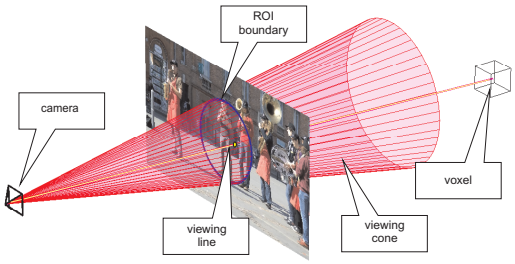

Figure 3: Viewing cone back-projecting a 2D ROI (in blue) and viewing line back-projecting a 2D point (in yellow); see text.

The key ingredient for defining a 3D interest map is the notion of viewing cones. A *viewing cone* is the back-projection of a 2D ROI, *i.e.*, the generalized cone in 3D space that is the assemblage of 3D lines passing through the camera center and every pixel on the 2D ROI boundary (see Figure 3). For the k-th elliptic ROI ($k = 1..K$) in view j, referred to as E_k^j, the corresponding viewing cone can be defined in the projective 3D space by its order-4 symmetric homogeneous matrix [13, p199]

$$\mathtt{\Lambda}(E_k^j) = (\mathtt{P}^j)^T \mathtt{E}_{jk} \mathtt{P}^j \tag{1}$$

[1]Both the synchronization and the calibration are assumed to be, at least approximately, available for the J cameras.

where $\mathtt{E}_k^j$ is the order-3 symmetric homogeneous matrix of the ellipse [13, p30] describing the boundary of E_k^j, and $\mathtt{P}^j$ is the projection matrix of camera j. In the situation illustrated by Figure 2, we have one elliptic ROI centered in each view with $J = K$. In the more general situation, multiple ROIs in a view are allowed (as explained in the next section).

We now partition the 3D scene into a set of voxels $\mathcal{S} = \{v_i\}_{i \in I}$. The intersection of viewing cones in 3D (as suggested by Figure 3) reveals a region where the voxels are seen by many cameras and can be considered as interesting for that reason. Hence, we can introduce a *measure of interest* of a voxel v_i by looking at how often it is seen. If we let $\mathcal{E}$ be the set of all 2D ROIs and $\mathcal{E}(v_i) \subset \mathcal{E}$ be a subset that depends on v_i, this 3D interest level is related to the 2D ROIs through the following definition:

$$Int(v_i) = \frac{1}{|\mathcal{E}|} \sum_{E \in \mathcal{E}(v_i)} \frac{\mathrm{Vol}(\mathtt{\Lambda}(E) \cap v_i)}{\mathrm{Vol}(v_i)} \tag{2}$$

where $\mathrm{Vol}(a)$ stands for the 3D volume of the considered 3D regions in argument a. As the quantity $\mathrm{Vol}(\mathtt{\Lambda}(E) \cap v_i)$ computes the volume of intersection of v_i with the viewing cone through E, the measure (2) is maximum when $\mathcal{E}(v_i) = \mathcal{E}$ and v_i is *entirely* included in all ROIs of $\mathcal{E}$, giving $Int(v_i) = 1$. It is minimum when v_i is not included in any ROI, giving $Int(v_i) = 0$. The intermediate values measure how often a given voxel is partially or completely seen.

We will say that the measure of interest (2) is (only) *geometrically consistent* when $\mathcal{E}(v_i) = \mathcal{E}$ and is (both geometrically and) *photometrically consistent* when $\mathcal{E}(v_i) \subset \mathcal{E}$ is the subset in which v_i is (at least partially) visible. In our context, a voxel v_i is said to be *visible with respect to a set of ROIs* $\mathcal{E}' \subset \mathcal{E}$ if there are pixels in each ROI of $\mathcal{E}'$ such that (*i*) all their back-projections as 3D *viewing lines* cut v_i (see Figure 3) and (*ii*) the neighborhoods of all these pixels are photometrically consistent in the views. It is straightforward to compute a geometrically consistent measure of interest, by intersecting all the cones associated with $\mathcal{E}$; it is the information brought by the users that contributed and no sophisticated computer vision algorithm is required. Nevertheless, due to many inter-voxels occlusions, only a subset of viewing cones associated with $\mathcal{E}$ are really seeing a visible voxel, which motivates the notion of photometrically consistent measure of interest. In Figure 6, the red voxel that is located at the intersection of the frustums of the three cameras is only visible (i.e., not occluded) in one frustum. Even if seen by all cameras, this red voxel is clearly not the most interesting since in fact only one user is seeing it. The method to determine $\mathcal{E}(v_i)$, the subset of 2D ROIs in which v_i is visible, will be tackled in the next section.

At this step, it is natural to consider the distribution of interest among the voxels through a histogram and subsequently introduce a normalized interest measure derived from (2) as

$$\widetilde{Int}(v_i) = Int(v_i) / \sum_{i \in I} Int(v_i). \tag{3}$$

Definition 1 *A **3D interest map** is the limit form of the 3D histogram with voxels as bins (diagonal length $\Delta l \to 0$) with respect to the normalized measure of interest (3).*

In other words, if the voxels decrease in size and become infinitesimally small, the normalized measure of interest (3) behaves like a continuous probability density function.

Finally, we will model our 3D interest map (in our implementation and due to its generality) as a 3D Gaussian Mixture Model (3D

GMM) with a satisfying number G of mixed components:

$$\widetilde{Int}(v) = \sum_{g=1}^{G} w_g \mathcal{N}(v; \boldsymbol{\mu}_g, \Sigma_g) \quad (4)$$

where v is now a 3D point (seen as an infinitesimally small voxel), $\mathcal{N}$ is the 3D Gaussian density, and w_g, $\boldsymbol{\mu}_g$, Σ_g are the weights and parameters of the 3D mixture. We detail the estimation of these parameters in the next section.

A GMM is flexible enough to approximate closely any multimodal density of 3D interest. Using GMM, the estimated 3D interest map can also be re-projected in the image spaces to play the role of J dependent 2D interest maps associated with the J cameras (see also the right part of the red rectangle in Figure 1).

Figure 4: 2D elliptic ROIs, as computed in Step (1) of our algorithm, on the RagAndFlag dataset.

4. 3D INTEREST MAP COMPUTATION

In this section, we present a method to build a 3D interest map using several synchronized videos shooting the same scene. The proposed algorithm (see Algorithm 1) is broken down in two successive estimations: a Gaussian mixture is first computed to produce a geometrically consistent 3D interest map (Steps (1)-(3)). The intuition behind this first step is to triangulate interesting 3D regions (Figure 5) from automatically detected 2D regions of interest (Figure 3). At this step, the 3D interest map (i.e., the Gaussian mixture) is coarse. The map is then refined into a final photometrically consistent one (Steps (4)-(6)). These last steps are made possible thanks to the constraints brought the coarse intermediate 3D interest map.

Algorithm 1 Computation of a GMM-based 3D interest map.

(1) Compute K_j ellipses of interest E_k^j for each view j
(2) Compute the intersection of visual cones
(3) Estimate a geometrically consistent 3D GMM
(4) Re-project it to get 2D GMM in image spaces
(5) Compute 2D Masks serving as PMVS input
(6) Estimate the final photometrically consistent 3D GMM from PMVS output

Detection of Ellipses (1). In this step, we look for 2D regions that are likely to contain interesting objects. These regions will then be used, as an implementation of Equation 2, to produce the cones as explained in the previous section. Many algorithms exist for such a task, from saliency maps to object detectors (face or body detectors would fit our use case).

At this stage, we do not need a high precision on these ellipses of interest, since they are primarily a way of reducing the complexity of the following steps. We therefore choose to use off-the-shelf OpenCV blob tracker, which uses standard algorithms for foreground detection (we use a temporal window of 15 frames) followed by blobs detection and tracking, considering the connected components of the foreground mask. In other words, we rely mostly on the apparent motion to detect the ellipses of interest, which is coherent with our use case of live performances. An example of the output of this algorithm can be seen on Figure 4. Note that the detected ellipses are not matched between views.

It is possible that the apparent motion is too low to detect ellipses based on the previously explained algorithm. In that case, we build on our assumption that users naturally tend to focus their camera towards objects of interest. We therefore consider an ellipse of area A_c centered on the camera frame, as shown on Figure 2. If the cumulated area of the ellipses of interest detected during Step (1) of our algorithm is less than $A_c/2$, then we empirically consider that there is insufficient apparent motion on the scene and switch to the central focus assumption.

Intersection of Visual Cones (2). At this stage, only a geometrically consistent 3D interest map can be computed. Indeed, we have no information about the visibility of a detected 2D ellipse number k in view j ($\mathtt{E}_k^j$ in Section 3's formalism) in other views $j' \neq j$. We use Equation (2) with $\mathcal{E}(v_i) = \mathcal{E}$ for defining a first coarse 3D interest map. In other words, the basic idea is to intersect cone-pairs $\mathtt{\Lambda}(\mathtt{E}_{k_1}^j)$ and $\mathtt{\Lambda}(\mathtt{E}_{k_2}^{j'})$ for all pairs of views (j, j') and for all pairs of 2D ROIs $(\mathtt{E}_{k_1}^j, \mathtt{E}_{k_2}^{j'})$ in these views.

The algorithm for intersecting two visual cones, related to cameras j and j' and 2D ROIs $\mathtt{E}_{k_1}^j$ and $\mathtt{E}_{k_2}^{j'}$, is as follows:

1. In image j, generate M random points inside the ellipse $\mathtt{E}_{k_1}^j$ and back-project each of these points p as a 3D line $\mathtt{L}_p^j$, through p and the camera center.

2. Compute the two intersection points where line $\mathtt{L}_p^j$ meets the viewing cone $\mathtt{\Lambda}(\mathtt{E}_{k_2}^{j'})$ associated with image j'. This step can be achieved very efficiently when formulated with projective geometry[2].

3. If such intersection points exist, discretize the line segment, whose endpoints are these two points, into T points so that image j yields MT 3D points.

4. Repeat 1-3 by switching the roles of cameras j and j'.

Now, given a sequence of J views, we can apply the above algorithm for each of the $\frac{1}{2}J(J-1)$ distinct image-pairs and each pair of detected ellipses that can be formed in the image-pairs. Indeed, Figure 4 shows a pair of images for which 5 (left image) and 4 ellipses were detected, which means that, in this particular example, 20 cones intersections are computed. Note that all cones do not necessarily intersect, so 20 cones intersections is an upper bound. The result of this step is a cloud of 3D points with denser regions where the interest is locally high.

Figure 5 illustrates this step on our BrassBand dataset, for which there is little movement in the scene. As a consequence, cones are generated based on the central focus assumption. Thus, there is one cone per camera. We can see on the figure the 3D points (in blue) that are generated in the cones intersection.

3D GMM Estimation (3). Based on the obtained 3D data points, we now estimate the first interest density (a 3D GMM) given by Equation (4). Since we do not know the number G of Gaussian modes and we have no initial parameter estimates, we can not rely

[2]Let $\mathbf{x} \in \mathbb{R}^3$ be the homogeneous vector of an image point x. The 3D line, back-projection of x, can be given by a 3D point function $\mathbf{X} : \mathbb{R} \to \mathbb{R}^4$ defined by $\mathbf{X}(\mu) = ((\mu\mathbf{x} - \mathbf{p}_4)^T \mathtt{M}^{-T}, 1)^T$, using the projection matrix decomposition $\mathtt{P} = [\mathtt{M} \mid \mathbf{p}_4]$ ([13, p162]). The line cuts a cone $\mathtt{\Lambda}$ at two points $\mathbf{X}(\mu_1)$ and $\mathbf{X}(\mu_2)$ where μ_1 and μ_2 are the two real roots of the quadratic polynomial $\mathbf{X}(\mu)^T \mathtt{\Lambda} \mathbf{X}(\mu)$.

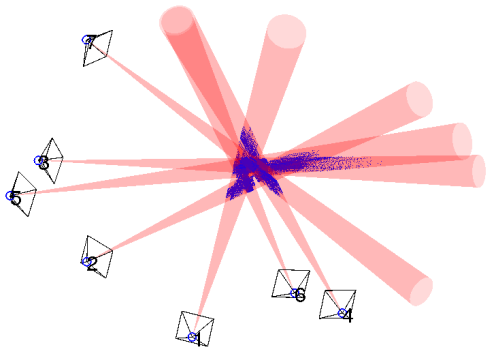

Figure 5: Intersection of visual cones (BrassBand dataset).

on the standard Expectation-Maximization technique for this step. We use Mean-Shift density estimator [7] that is devised to iteratively detect the unknown distribution modes along with the data points that belong to the cluster associated with each mode. The hard assignment between data points and modes given by Mean-Shift then allows us to trivially estimate each weight w_g (ratio of points assigned to the jth cluster vs. total number of data points) and each parameter: both $\boldsymbol{\mu}_g$ and Σ_g are estimated with maximum likelihood formulas.

Mean-Shift clustering is only controlled by one scale parameter: the bandwidth. Theoretically, the bandwidth compromises between the bias and variance of the estimator. In practice, it can be understood as the distance allowed between peaks of the distribution. At the current level (Step (3) in algorithm 1), however, the data points located at the intersection of the visual cones (see also Figure 5) do not fit with any recognizable shape (e.g., human performers). It is discriminative enough to select a relatively large bandwidth to detect the modes of our first 3D interest map: we typically choose $h = 2m$ in our experiments.

Re-projection to 2D GMM (4). Using a GMM in 3D allows the 3D interest maps to be re-projected in the J views. Such re-projection helps with the visualization and the evaluation of the 3D interest maps in the experimental section. Figure 9, second row, shows an example. Note that the 3D multi-modal distribution also appears as a multi-modal density in 2D. Flandin and Chaumette [10] define a model providing, for every 3D point of the scene, the probability that it belongs to an object of interest. They introduce several propagating rules for Gaussian uncertainties, such as subspace projection and back-projection of a distribution from 2D to 3D. Their fourth rule can be reused in our case to characterize the 2D GMM in image spaces resulting from the perspective projection of our 3D interest maps. Modeling the interest with Gaussians allows us to switch back and forth between 3D and 2D.

PMVS Masks Computation (5). We now aim at making the 3D interest map more discriminative. As said before, we used Equation (2) with $\mathcal{E}(v_i) = \mathcal{E}$, leading to a first coarse estimate. To refine it, we must determine whether a voxel is visible or not in a view. In Figure 6, we already noticed (cf. Section 3) that the red voxel that is located at the intersection of the viewing cones is the most interesting according to a *geometrically consistent measure of interest* but it should not be considered as the most interesting with a *photometrically consistent measure*: its red color is not occluded in only one view. The blue voxel is more interesting (seen from several views with photometrically consistent blue projections). In our implementation, this consistency is imposed by a constrained 3D reconstruction, achieved efficiently with multi-view stereo software such as PMVS [11] that is devised to propagate photometric consistency.

Given the extreme difficulty of wide-baseline multiple-view correspondences in our set-up, we use the re-projected GMM to produce 2D masks that will constrain and guide the 3D reconstruction algorithm. For each image, we binarize the 2D GMM (i.e., the 3D interest maps re-projection) in order for the mask to contain 95% of the information from the GMM. Results of this step can be visualized on the third row in Figure 9. As shown in the experiments, PMVS outputs denser 3D reconstructed points in the most interesting regions (e.g., around the human performers) thanks to the 2D masks guidance. The role of the "crowdsourced" masks is quantitatively evaluated in Section 6.

Final Estimation from PMVS Output (6). The PMVS software takes as input the camera projection matrices along with the masks computed during the previous step, and outputs a set of reconstructed 3D points. These points respect the photometric consistency. Note that we need to provide a parameter to PMVS, called the level, which specifies the resolution at which the input images will be processed. We set the level to 4, which means the image resolution considered is 120×68. This choice significantly speeds-up the computation as well as limits the influence of a bad synchronization on the quality of our results. Similarly to Step (4) of the algorithm, we then estimate the parameters of a 3D GMM to model the final 3D interest map. In our public event use case, if we ideally aim at estimating a separate mode for each dancer or singer, the bandwidth for the Mean-Shift clustering should be selected as $h = 60cm$ to typically separate two humans standing next to each other. This bandwidth parameter controls the granularity of the 3D interest map. The re-projection of our refined 3D interest map is shown on the fourth row in Figure 9.

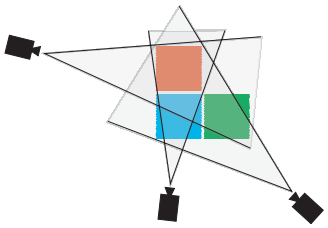

Figure 6: The red voxel has the highest geometrically consistent measure of interest (included in the three viewing cones) but a low photometrically consistent measure (only visible in one).

5. APPLICATION TO VIDEO MASHUP

We now demonstrate the power of 3D interest map, by applying the information we obtained to video mashups.

5.1 Video Mashup

The goal of video mashup is to automatically create a temporally continuous video of an event by splicing together different clips from simultaneously recorded videos of the event, usually shot from a diverse set of angles. A video mashup system aims to replace a human director, and therefore a successful video mashup system should produce videos that are interesting to watch, logically coherent, and cover the important scene that happens during the event.

One of the state-of-the-art systems for video mashup is Jiku Director [21], which focuses on mashing up videos captured with handheld mobile devices. Jiku Director uses an online algorithm [23] to produce the mashup, following a shot-change pattern learnt from professional, human-edited, video. The system also filters out bad quality videos through simple feature analysis and tries to increase

the diversity of shots in the mashup by not selecting clips from the same devices when possible.

Jiku Director, however, is limited in two ways. First, it always uses the video clips recorded by users "as-is" without any cropping and scaling, nor introducing any camera movement. As a result, the camera shots in the resulting video are limited to the movement made by the user while recording the video. Furthermore, since the videos are recorded by audiences of an event, the shots are most of type of long/medium shots. The resulting mashup is lacking in diversity of shot types. Second, when Jiku Director introduces a cut, its only consideration is the shot angle (viewing direction of the camera) and shot length (distance of the camera from the stage), without considering the subjects or narrative.

One way to overcome the first limitation is to analyze salient features from each video independently and model the importance of the content. Jiku Director could then introduce virtual camera movement (zooming effects and dolly shots) by cropping out the region-of-interests (ROIs) from the videos. This approach is similar to that adopted by Carlier et al. [5]. This approach, however, could not overcome the second limitation, as there is no way to associate an object of interest that appears in one video clip with another object of interest that appears in another clips.

The proposed 3D interest map allows us to identify such association for objects that appears in the same time instant across two video clips. Furthermore, using the objects' bounding boxes and interest levels, we can add camera movements to pan from one object to another, and to zoom in for close-up shots of the objects, producing better mashups. The rest of this section describes how we apply 3D interest map to a new video mashup algorithm.

5.2 Computing Bounding Boxes

The first step of our mashup algorithm is to compute bounding boxes that contains the most interesting regions of the videos. Since our 3D interest maps are modeled by GMM, one bounding box for each Gaussian in the mixture can be produced easily.

The mashup application, however, needs stable bounding boxes that are tracked over time. The algorithm described in Section 4, however, applies at every time instant without considering temporal consistency. Therefore, to produce bounding boxes for the mashup applications, we change the last step of our algorithm (clustering a 3D point cloud) and compute the bounding boxes in the 4D space (3D and time). We augment the 3D points with a fourth coordinate that corresponds to their frame index, and apply the Mean-Shift algorithm to this 4D point cloud. We compute a bounding box for each cluster that contains all points of the cluster for its entire duration. The bounding boxes are stable over time by construction and can be mapped in all videos. An example of this output can be seen on Figure 7. Note that all bounding boxes are constrained to be of the same aspect ratio as the original video.

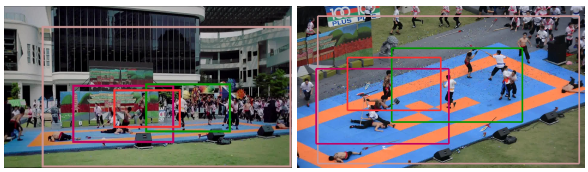

Figure 7: Bounding boxes computed from our 3D interest maps. Bounding boxes of the same color are corresponding to the same object of interest.

5.3 Mashup Heuristics

With the set of bounding boxes, we now present a simple algorithm for video mashups to illustrate the power of 3D interest map.

Note that 3D interest map cannot be computed online, but as we will later compare our results with Jiku Director's, which is an online algorithm, we choose to design our mashup algorithm to be online as well, with no knowledge of future frames beyond the current selected shots.

Our algorithm takes, as input, the set of object of interests computed from 3D interest map. For each object appearing in each frame, we have its bounding box, x-y coordinate and depth of its centroid in each frame. The output of our algorithm will be a sequence of *shots* (t, j, B, m), where t is the beginning time of the shot, j is the camera to choose the shot from, B is the set of regions of interest to use in this shot, and m is the virtual camera movement in this shot.

We use the same decision matrix used by Jiku Director to decide Camera j in each shot. Once j is fixed, we determine t, B, and m as follows. We first define a *scene* as a sequence of shots with logical coherence. For instance, a scene can start with a wide angle shot, followed by a zoom in shot to an object x, followed by a close up shot of x. We limit the length of the scene to containing no more than N_s shots (we use $N_s = 3$).

To determine the current shot starting at time t, the algorithm first needs to decide if the current shot belongs to the same scene as previous shot, or starts a new scene. If current scene has reached its limit in number of shots, we simply start a new scene. Otherwise, we try to see if the current shot fits.

Let L_{max} be the maximum possible shot length (we use $L_{max} = 7s$). To determine the current shot, we look at the bounding boxes and objects for the next L_{max} time in the video shot by camera j.

We prioritize the bounding box of each object by their importance in the scene. Each object i has a *camera independent importance*, $I(i,t)$, which is computed from $\sum_j v_i^j(t)/d_i^j(t)$, where $v_i^j(t)$ is 1 if object i appears in Camera j at time t; 0 otherwise, and $d_i^j(t)$ is the depth of the object i from the camera plane of camera j at time t, normalized to 0 and 1. The intuition here is that an object that appears frequently in the collective video clips and closer to the camera is the more important than those that appears fewer number of times/far away from cameras.

In each video clip, we compute the *camera dependent importance* of an object, $I_j(i,t)$ as $I(i,t)/d_i^j(t)$. The importance of the bounding box is then the sum of the camera dependent importance of all objects in the bounding box. Note that even though each bounding box is associated with an object, other objects may appear in the bounding box. Naturally, a bounding box that includes multiple important objects becomes important.

Now, we look at the most important bounding boxes in j at each time instance for the next L_{max} seconds. If at the start of the L_{max} seconds, the same bounding boxes is used as the previous shot, we fit the current shot into the same scene, and set B to these bounding boxes. We may stop the current shot at the time when these bounding boxes disappear. If a bounding box different from that of previous shot become the most important, we introduce a new scene, setting B to the new most important bounding box.

It remains to determine m for the current shot. The choice of m can be either: wide angle shot, close up, move the camera (zoom in, zoom out, or pan). The algorithm follows the following principles: (i) there should not be frequent camera movement, and (ii) the choice of m should lead to "fluid" shots: two consecutive close up shots should have the same object; zooming into an object should be followed by either a close up shot that include that object or a pan shot moving from that object to something else; a zoom out shot should be followed by a wide-angle shot; and a pan shot mov-

ing from an source object to a target object should be followed by a zoom out shot from the target, or a close up shot of the target. In Figure 8, a video transition from our heuristics illustrates how the 3D informations can be used for continuity editing.

Figure 8: A video transition keeping the same object(s) of interest in focus between two successive shots from two different cameras.

6. EXPERIMENTS

To evaluate our proposed 3D interest maps and its application to video mashup system, we first introduce two useful datasets and briefly explain our experimental set-up. The interest maps obtained from our experiments are then compared against saliency maps in the 2D image space (a comparison in 3D is difficult to visualize). Finally, we report the major results from our study to evaluate our improvements to video mashups.

6.1 Dataset

We present, in this subsection, the data we have worked on and that we use to evaluate our approach.

RagAndFlag. This dataset consists of videos filmed with seven cameras simultaneously (see Figure 1). This dataset presents a challenging scenario where the cameras have very diverse viewing angles and the videos depict a highly dynamic outdoor scene with variable lightning and about 50 moving objects (dancers, see also [22]). We use this dataset to test our mashup application.

BrassBand. This dataset depicts a more static scene with eight musicians close to each other, moving occasionally. Seven cameras are spread around the scene: three are fixed on a tripod and four are handheld (shot with smartphones) with little movement.

6.2 3D Interest Map

In this section, we evaluate the accuracy of the reconstructed 3D interest maps.

Our experiments use a radius of one fifth of the focal length to produce the cones-based 3D interest map. The parameters M and T are set to 10. We calibrate the cameras for each dataset using specific patterns (orange squares in **RagAndFlag**, concentric circles in **BrassBand**) on the planar stages. Knowing the plane-to-image homography associated with a given plane in the scene, we can (under standard assumptions of square pixels and centered principal point) compute the focal length [13] and then the camera pose [27]. The average re-projection error of our calibration is 2.2 pixels on 1920×1080 images, which is acceptable for our application. We synchronize the videos using an algorithm from Shrestha et. al [16] and observe an average error of 7.25 frames (0.25 sec).

Figure 9 shows examples of our results on the **RagAndFlag** dataset. In this figure, original images are introduced in the top row; the output of Steps (1)-(3) from our algorithm (i.e., a geometrically consistent version of the 3D interest maps computed from the cones intersection) is shown in the second row. This version will be referred as *cones maps* in the following paragraphs. The third row shows the masks that are inferred from the cones maps, as described in Step (4) of our algorithm. The forth row shows the final output of our algorithm, the 3D interest maps. We manually created ground truth masks to evaluate the 3D interest maps,

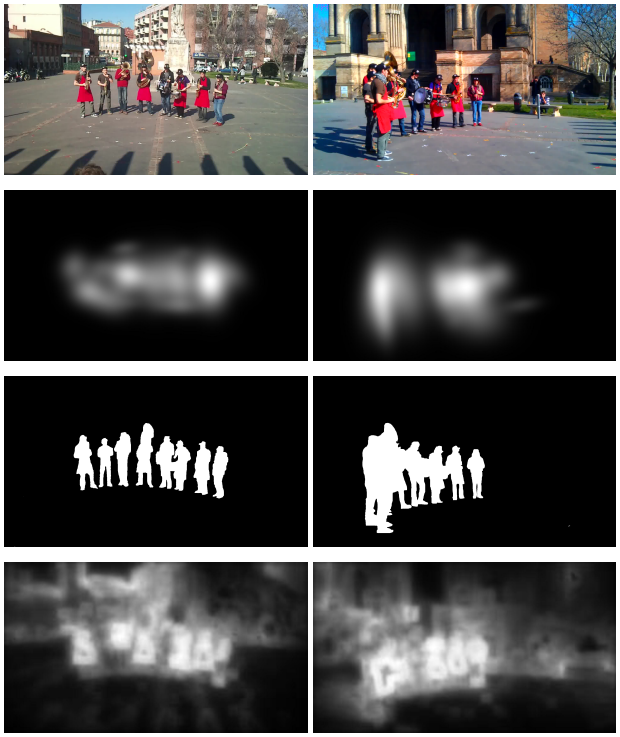

Figure 10: Results on the BrassBand dataset. For each row, from top to bottom: (i) original images; (ii) our 3D interest maps; (iii) manually generated ground truth binary masks; and (iv) state-of-the-art saliency map, as computed by Goferman [12].

and these ground truth masks are exposed on the fifth row. Finally results from a state-of-the-art saliency algorithm by Goferman et. al [12] are shown on the sixth row.

Similar results are presented for **BrassBand** in Figure 10. Due to space limitations, we only present original images, our final 3D interest maps, Goferman saliency, and the ground truth masks in these figures.

We can observe on all these figures that our 3D interest maps are very precise: almost all regions that are highlighted as salient are recognized as part of some objects of interest.

We can see also on Figure 9 that cones maps (2nd row) are more precise than Goferman's saliency maps (last row). This is the reason why we do not simply binarize state-of-the-art saliency maps to produce the 2D masks. Though the recall from saliency maps is quite high, the precision is insufficient to efficiently serve as a mask for PMVS.

In order to quantitatively evaluate our results, we follow the methodology from [4] and estimate a precision-recall (PR) curve by varying a threshold on the interest values and generating a binary saliency map for each value of the threshold. These experiments have been conducted on four types of interest maps: our final 3D interest maps, Goferman saliency, cones maps (see above) and a version of our interest maps we call *No Masks*. *No Masks* is computed by applying the PMVS algorithm to our data without inputting the masks based on the cones intersection, and estimating 3D interest maps from the set of points obtained from PMVS. We evaluate this version to prove the importance of Steps (1) to (5) in our algorithm.

We created ground truth masks for 12 instants in our datasets, and resulting PR curves are displayed in Figure 11. Our 3D interest

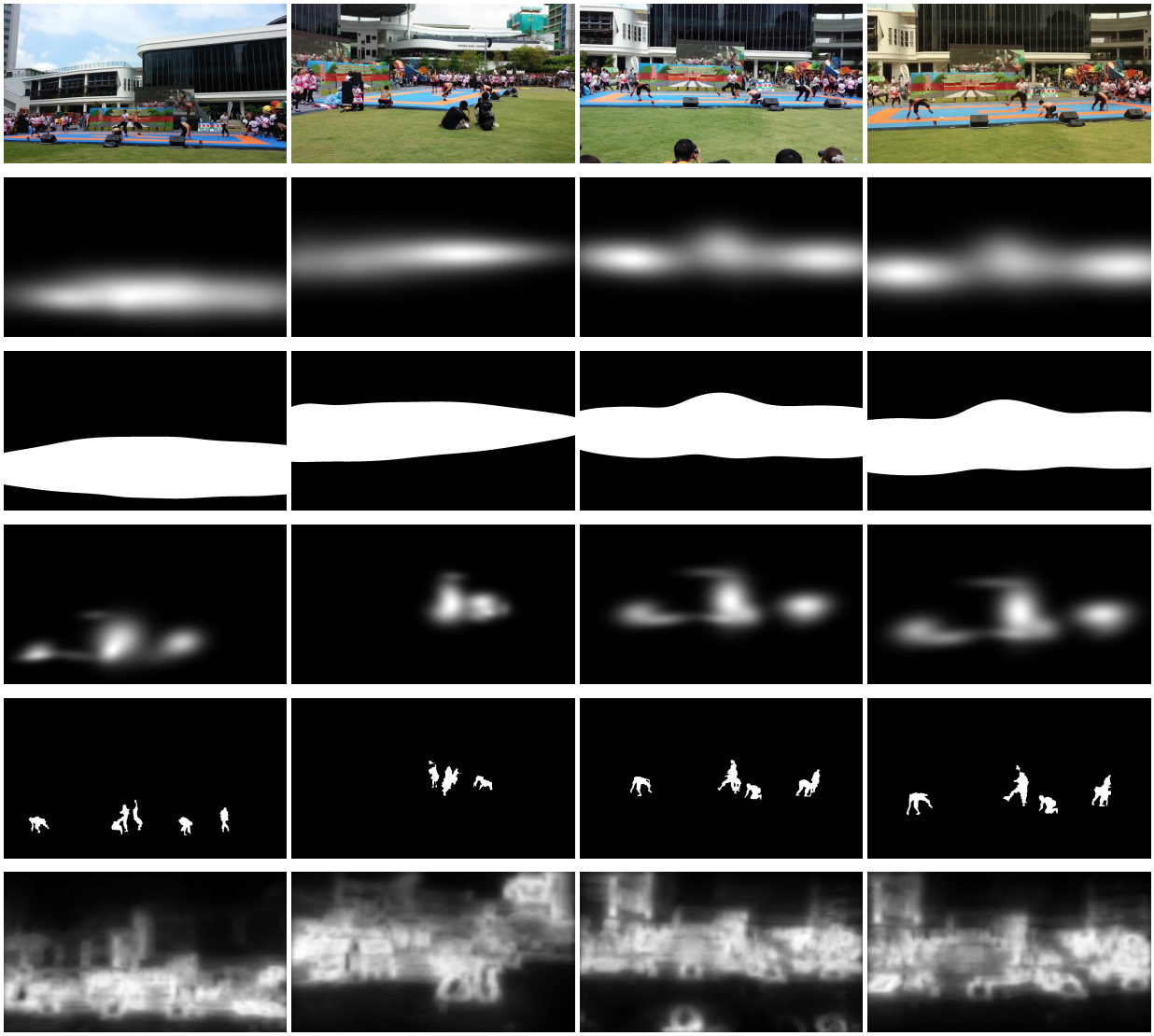

Figure 9: Results on the RagAndFlag dataset. For each row, from top to bottom: (i) original images; (ii) re-projected 3D interest maps computed by intersecting viewpoint frustums (*cones maps*); (iii) masks used as an input to PMVS; (iv) our final results; (v) manually generated ground truth binary masks; and (vi) state-of-the-art saliency map, computed using Goferman's method [12]

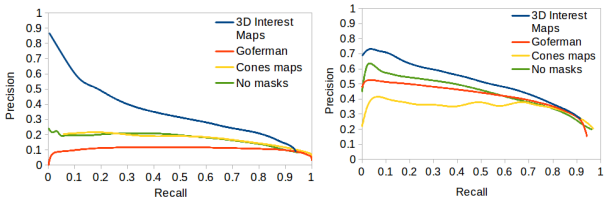

Figure 11: Averaged precision-recall curves on RagAndFlag (left) and BrassBand (right)

maps spectacularly outperforms Goferman's algorithm on the **RagAndFlag** dataset, which can be explained by the small size of the regions of interest (see Figure 1: the crowd is far from the scene) and the high variety in contrast of many background elements (the mat, the buildings, etc.). Our 3D interest maps is also better than Goferman in the **BrassBand** dataset, but the saliency detection is more performant in this case. Indeed the objects of interest are very salient in the images from **BrassBand**, since most of the musicians are wearing red colors.

Results on *cones* and *no masks* are also not as good as our 3D interest maps, which validates the importance of both steps in our approach. It is interesting, however, to note that cones maps is more precise than Goferman on the curves. This validates our choice to use cones maps to compute the PMVS masks (Step (5) in our algorithm) instead of just using a saliency map, because the more precise the masks are, the less noisy the reconstructed points are, and therefore the more precise our 3D interest maps are. In addition, it takes less than a second to compute cones maps whereas Goferman's saliency takes 5 seconds per image on GPU.

6.3 Evaluation of Mashup Algorithm

We now present the methodology and results of our user study to evaluate the video mashups produced from our algorithm.

Dataset. We pick the Rag&Flag dataset as the input to our algorithm, as it has the most dynamic content. We use a 1-minute

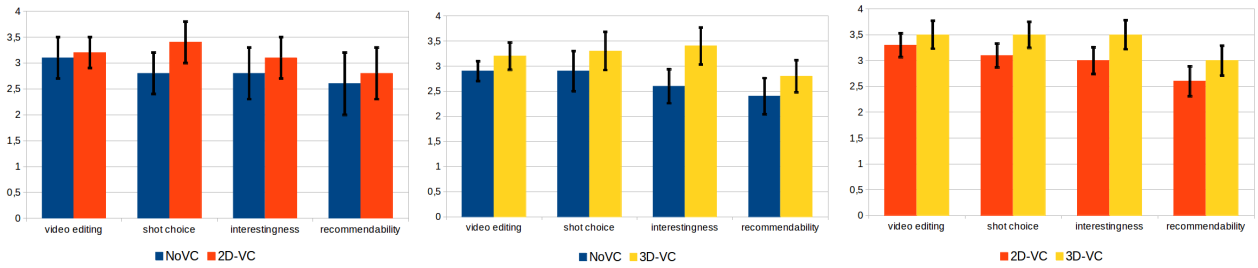

Figure 12: Average responses of the users for the three scenarios: NoVC vs. 2D-VC, NoVC vs. 3D-VC and 2D-VC vs. 3D-VC

sequence from the 7-minute video, to limit the length of the user study, and let users be able to watch the videos several times.

Benchmark. We choose to compare our output with two other methods. The first is the output from the MoviMash algorithm [23]. As MoviMash does not generate virtual camera movements, we denoted the output from this video as NoVC. We also included the output from a version of our algorithm that makes use of 2D saliency maps to generate zoom and pan shots [20]. We use the same saliency detector than the one from step (1) in algorithm 1, by consistency. We track the obtained ROIs with the same technique than the one from paragraph 5.2 except that it is applied in reduced (2D and time) dimension. The difference between this 2D version and ours is that: (i) the saliency information is computed for each video individually (without considering 3D scene information from other videos), and (ii) no interest information across different videos is available. We denoted the output from this video as 2D-VC. The output from our algorithm using 3D interest map is denoted as 3D-VC. Note that the three versions NoVC, 2D-VC, 3D-VC can be downloaded as supplemental material.

Users. We recruited 30 users (10 female, 20 male, age between 21-43) to participate in the user studies. All users have limited experience with video editing and production except for one.

Methodology. The user study is conducted online. Users are presented with the instructions and are told that "the main purpose of the user study is to rate the videos according to the quality of video editing". We created two Web pages for our purpose, each of them displaying two videos. On one of the page, NoVC is displayed along with either 2D-VC or 3D-VC (the choice is random). On the other page, 2D-VC, and 3D-VC are displayed. The order of display of the videos is random on both Web pages, to eliminate bias. The users are allowed to replay the video as many times as they want. A list of rating criteria are presented to the user below the video, where they are asked to rate the videos with a rating from 1 (worst) to 5 (best) according to the criteria of (i) the quality of video editing, (ii) the choice of camera shot, (iii) the interestingness of the video, and (iv) the likelihood that the user would recommend the video to a friend.

Results. The average responses of the users, along with their 95% confidence interval, are plotted in Figure 12.

The first result that is very clear is that users prefer the versions with virtual camera movements (2D-VC, 3D-VC) rather than the output from MoviMash (NoVC). The choice of camera shot (ii) is the criteria where the difference is the most obvious, which is understandable as variety in the choice of shots makes the video more enjoyable to watch.

The interestingness of the video (iii), meaning the capacity of a video to display the interesting subjects in a scene, is also significantly higher in 3D-VC than in NoVC. This proves that the 3D interest maps indeed help the mashup algorithm displaying the interesting regions in the videos. Note that 2D-VC also outperforms NoVC with respect to this criterion but that the difference is less significant. Indeed since the 2D interest maps used to compute this version are mostly based on apparent motion, the 2D-VC mashup has often multiple choices of moving regions to display and does not necessarily focus on the prominent region of interest. The right diagram of figure 12 seems to indicate that 3D-VC performs better than 2D-VC, however the results can not be proven statistically significant. Indeed, to non-expert users, the two videos can look similar because the shots dynamics are the same (zoom in, zoom out, pan).

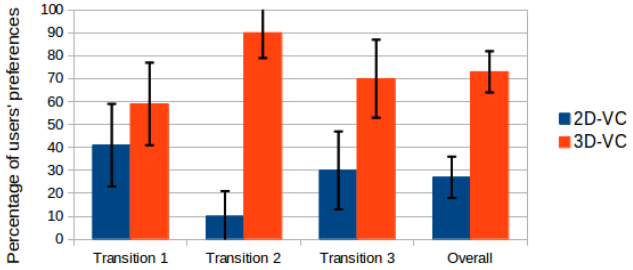

Figure 13: Second user study (99% confidence intervals)

Therefore we conducted another, much shorter, user study. We asked a different pool of 45 users to look at three pairs of very short videos, each depicting a transition between 2 shots (e.g. figure 8). Each pair consisted in the transition computed by 3D-VC and by 2D-VC. We displayed the three pairs in random order, and asked users which transition they preferred. Unlike the results presented in figure 12, this study allows us to clearly (statistically significantly) state that the transitions based on the 3D interest maps are better than the ones based on the 2D interest maps (see figure 13).

7. DISCUSSION AND CONCLUSION

This paper introduces *3D interest maps*, a new concept that associates each 3D point in a scene with its level of interest and can be viewed as extension of saliency maps to 3D. We formally define a 3D interest map and explain how we compute it. We show that our 3D interest maps are more precise in locating interesting objects than state-of-the-art saliency algorithms. A key aspect of our 3D interest maps is that they allow mapping of interesting regions across different videos that depict the same scene at the same time instance. This property is useful for many applications, such as video mashup systems.

The main challenges and limitations of our work are obviously the strong assumptions about estimation of the camera poses and synchronization between them. In our experiments, cameras were calibrated from a single view in order to avoid matching features

between views, as we benefited from the existence of geometric patterns lying on the stage ground. In many man-made environments, it might be worthwhile to calibrate the camera by automatically detecting vanishing points corresponding to orthogonal directions [8, 29]. Multiple tags and projected patterns (e.g., multiple tags or projections on the walls behind the stage) [19] also have clear potentials for solving this problem. Alternatively, we may also rethink our work following uncalibrated approaches. The authors of [1] exploit sparse feature flows in-between neighbouring images/cameras and structure these visual data using distances and angles. In our case, the idea would be to replace the 3D space where interest levels are estimated with descriptor spaces where relationships between videos crowd-sourced from different cameras could also be established.

Regarding future work, we are convinced, among other authors, that calibration might become a real-time commodity using both the smartphone cameras and inertial measurement units (IMU) as well as, in the very near future, depth sensors (e.g., as in Google's Project Tango). In any of the above-mentioned approaches, some information from the IMU (namely the gravity vector) and Exif data (namely the focal length) can be very helpful, as it can basically provide the vanishing point of the vertical direction or, equivalently, the vanishing line of the ground plane [18]. In some recent work [28], an inertial tracker provides camera poses which are subsequently refined by visual tracking. And last but not least, pose estimation by relocalization [24] form another promising, yet still challenging, paradigm.

3D interest maps is built from video clips filmed by users, where the interest levels of the scene emerges naturally from the users' shot framing. The quality of the interest map can only be improved with more videos.

Acknowledgments. This research is supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office.

8. REFERENCES

[1] A. Arpa, L. Ballan, R. Sukthankar, G. Taubin, M. Pollefeys, and R. Raskar. Crowdcam: Instantaneous navigation of crowd images using angled graph. In *3DTV-Conference*, pages 422–429, 2013.

[2] A. Borji, M. N. Ahmadabadi, and B. N. Araabi. Cost-sensitive learning of top-down modulation for attentional control. *MVA*, 22(1):61–76, Jan. 2011.

[3] A. Borji and L. Itti. State-of-the-art in visual attention modeling. *IEEE Tr. PAMI*, 35(1):185–207, 2013.

[4] A. Borji, D. N. Sihite, and L. Itti. Salient object detection: A benchmark. In *ECCV*, pages 414–429, 2012.

[5] A. Carlier, V. Charvillat, W. T. Ooi, R. Grigoras, and G. Morin. Crowdsourced automatic zoom and scroll for video retargeting. In *ACM Multimedia*, pages 201–210, 2010.

[6] A. Carlier, G. Ravindra, V. Charvillat, and W. T. Ooi. Combining content-based analysis and crowdsourcing to improve user interaction with zoomable video. In *ACM Multimedia*, pages 43–52, 2011.

[7] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Tr. PAMI*, 24(5):603–619, 2002.

[8] A. Criminisi, I. D. Reid, and A. Zisserman. Single view metrology. In *IJCV*, pages 123–148, 2000.

[9] T. Dittrich, S. Kopf, P. Schaber, B. Guthier, and W. Effelsberg. Saliency detection for stereoscopic video. In *MMSys*, 2013.

[10] G. Flandin and F. Chaumette. Visual data fusion for objects localization by active vision. In *ECCV*, pages 312–326, 2002.

[11] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Tr. PAMI*, 32(8):1362–1376, 2010.

[12] S. Goferman, L. Zelnik-Manor, and A. Tal. Context-aware saliency detection. In *CVPR*, pages 2376–2383, 2010.

[13] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[14] T.-H. Huang, K.-Y. Cheng, and Y.-Y. Chuang. A collaborative benchmark for region of interest detection algorithms. In *CVPR*, pages 296–303, 2009.

[15] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Tr. PAMI*, 20(11):1254–1259, 1998.

[16] Y. Ke, D. Hoiem, and R. Sukthankar. Computer vision for music identification. In *CVPR*, pages 597–604, 2005.

[17] F. Liu and M. Gleicher. Video retargeting: automating pan and scan. In *ACM Multimedia*, pages 241–250, 2006.

[18] J. Lobo and J. Dias. Vision and inertial sensor cooperation using gravity as a vertical reference. In *IEEE Tr. PAMI*, pages 1597–1608, 2003.

[19] I. Martynov, J.-K. Kamarainen, and L. Lensu. Projector calibration by "inverse camera calibration". In *SCIA*, pages 536–544. 2011.

[20] D.-T.-D. Nguyen, A. Carlier, W. T. Ooi, and V. Charvillat. Jiku director 2.0: A mobile video mashup system with zoom and pan using motion maps. In *ACM Multimedia*, 2014.

[21] D.-T.-D. Nguyen, M. Saini, V.-T. Nguyen, and W. T. Ooi. Jiku director: a mobile video mashup system. In *ACM Multimedia*, pages 477–478, 2013.

[22] M. Saini, S. P. Venkatagiri, W. T. Ooi, and M. C. Chan. The jiku mobile video dataset. In *MMSys*, pages 108–113, 2013.

[23] M. K. Saini, R. Gadde, S. Yan, and W. T. Ooi. Movimash: online mobile video mashup. In *ACM Multimedia*, pages 139–148, 2012.

[24] T. Sattler, B. Leibe, and L. Kobbelt. Improving image-based localization by active correspondence search. In *ECCV*, pages 752–765. 2012.

[25] P. Shrestha, M. Barbieri, H. Weda, and D. Sekulovski. Synchronization of multiple camera videos using audio visual features. *IEEE Tr. Multimedia*, 12(1):79–92, 2010.

[26] P. Shrestha, P. H. N. de With, H. Weda, M. Barbieri, and E. H. L. Aarts. Automatic mashup generation from multiple-camera concert recordings. In *ACM Multimedia*, pages 541–550, 2010.

[27] P. F. Sturm. Algorithms for plane-based pose estimation. In *CVPR*, pages 1706–1711, 2000.

[28] P. Tanskanen, K. Kolev, L. Meier, F. Camposeco, O. Saurer, and M. Pollefeys. Live metric 3d reconstruction on mobile phones. In *ICCV*, pages 65–72, 2013.

[29] J.-P. Tardif. Non-iterative approach for fast and accurate vanishing point detection. In *ICCV*, pages 1250–1257, 2009.

[30] B. W. Tatler. The central fixation bias in scene viewing: selecting an optimal viewing position independently of motor biases and image feature distributions. *J. of Vision*, 7(14):1–17, 2007.

[31] X. Xie, H. Liu, S. Goumaz, and W.-Y. Ma. Learning user interest for image browsing on small-form-factor devices. In *CHI*, pages 671–680, 2005.