

Organizing Videos Streams for Clustering and Estimation of Popular Scenes

Sebastiano Battiato¹, Giovanni M. Farinella¹, Filippo L.M. Milotta^{1,2}(✉),
Alessandro Ortis^{1,2}, Filippo Stanco¹, Valeria D’Amico², Luca Adesso²,
and Giovanni Torrisi²

¹ Department of Mathematics and Computer Science,

University of Catania, Viale A. Doria, 6, 95125 Catania, Italy

{`battiato,gfarinella,milotta,ortis,fstanco`}@dmi.unict.it

² JOL WAVE, Telecom Italia, Viale A. Doria, 6, 95125 Catania, Italy

{`valeria1.damico,luca.adesso,giovanni.torrisi`}@telecomitalia.it

Abstract. The huge diffusion of mobile devices with embedded cameras has opened new challenges in the context of the automatic understanding of video streams acquired by multiple users during events, such as sport matches, expos, concerts. Among the other goals there is the interpretation of which visual contents are the most relevant and popular (i.e., where users look). The popularity of a visual content is an important cue exploitable in several fields that include the estimation of the mood of the crowds attending to an event, the estimation of the interest of parts of a cultural heritage, etc. In live social events people capture and share videos which are related to the event. The popularity of a visual content can be obtained through the “visual consensus” among multiple video streams acquired by the different users devices. In this paper we address the problem of detecting and summarizing the “popular scenes” captured by users with a mobile camera during events. For this purpose, we have developed a framework called RECfusion in which the key popular scenes of multiple streams are identified over time. The proposed system is able to generate a video which captures the interests of the crowd starting from a set of the videos by considering scene content popularity. The frames composing the final popular video are automatically selected from the different video streams by considering the scene recorded by the highest number of users’ devices (i.e., the most popular scene).

Keywords: Video analysis · Clustering · Social cameras · Scene understanding

1 Introduction

During a social event, the audience typically uses its personal devices to record video clips related to the most interesting moments of the event. As a result, several videos will be related to the same visual contents, and this redundancy can be exploited to infer the most interesting moments of the event over time,

according to the people interests on the observed scenes. The issue of crowd-popularity estimation through automatic video processing is not trivial due to the variability of the visual contents observed by multiple devices: different points of view, pose and scale of the objects, lighting conditions and occlusions. The differences between device models should be also taken into account, since they imply different characteristics of the lens, color filter arrays, resolution and so on. For instance, even using two devices with similar (or equal) sensors the colors recorded will not necessarily be the same because devices responses are processed with different non-linear transformations due to the differences on the Imaging Generation Pipelines (IGPs). They can vary from device to device and even on an per-image basis [1, 2].

We propose a system called RECfusion to estimate the popularity of scenes related to multiple video streams. The streams are analyzed with the aim to create a continuous video flow, obtained by mixing the several input channels, taking into account the most popular scenes over time to reflect the interests of the crowd. Then, the clusters of the different scenes are tracked over time. This allows to have not only the most popular scene at each time, but also the other scenes of interest and give the possibility to introduce a scenes story log allowing the user to select the scene of interest among all the detected ones.

The reminder of the paper is structured as follows: in Sect. 2 we discuss related studies about crowd-saliency inference from multi-device videos. In Sect. 3 an overview of the RECfusion framework is given together with the description of its three main modules: intraflow analysis, interflow analysis and cluster tracking. In Sect. 4 a proper dataset is introduced, whereas in Sect. 5 we report the experimental settings and the results. We conclude the paper with a final discussion and hints for possible future works in Sect. 6.

2 Related Works

Different papers about crowd-saliency inference from multi-device videos have been proposed in literature in the past. The works in [3, 4] exploit Structure from Motion (SfM) to estimate a 3D reconstruction of the scene and the pose of employed devices. Hoshen et al. [5] uses egocentric video streams considering a single camera model acquired by different participants to create a single popular video of an event. However, in [3–5] the number of the different popular scenes and the number of the devices are known a priori. Saini et al. [6] developed the framework MoViMash with the purpose of replicate the behavior of a movie director: the system learns from a labeled sets of video frames “how to” and “when” perform transitions between different views. However, this technique is hardly adaptable for a real-time context, since for each different recorded scene a proper learning phase should be tuned to. ViComp is another framework similar to MoViMash [7]. In ViComp the final output video consists in a combination of several video streams from multiple sources. The combination is obtained by selecting high quality video segments according to their audio-visual ranking scores. It selects the best video stream among a pool of available ones basing

on degradation and noise caused by video compression [8] and estimated camera pan and tilt [9].

The aforementioned approaches achieve significant results but, compared to them, our approach (RECFusion) does not need any prior knowledge or training stage and is able to combine videos from an unknown number and types of recording devices. RECFusion is a framework with a popularity-based video selection approach: it clusters the video streams and selects the best video stream from each cluster exploiting clustering metrics.

3 RECFusion System Overview

RECFusion is a framework designed for automatic video curation driven by the popularity of the scenes acquired by multiple devices. Given a set of video streams as input, the framework can group these video streams according to the viewed similarity and popularity of the scenes over time, then it automatically suggests a video stream to be used as output acting like a “virtual director”. With the aim to mitigate the aforementioned differences in the color representation of the devices, due their different IGPs, the video frames are pre-processed by an equalization algorithm. This step helps the further computations that compares frames captured by different devices [1, 10–12]. After this normalization, the system extracts an image representation from each frame. The algorithm takes a frame as input and returns a descriptor. The aim is to have a descriptor that maximize the differences between semantically different frames and minimize the differences between semantically similar ones. In [1] a definition of light conditions (and almost devices) independent representation is given. The method is based upon the observation that changes of light conditions or device directly change the RGB values of the frame, while order of sensors response remains the same. Finally, equalization of RGB channels, as described in [1], is performed. After the normalization of the color domain, the video streams are analysed in our approach in three phases (Fig. 1), detailed in the followings.

3.1 Intraflow Analysis

The intraflow analysis segments the sequence of frames of a single video stream (Fig. 1(a)). During intraflow analysis the frames of each video are processed comparing their visual contents. For each frame of the video flow, we extract keypoints using the SIFT detection algorithm [13]. The set of the extracted SIFT features represents a template for the acquired scene. In this way, the comparison between frames could be done as the comparison between SIFT templates. When the comparison between the current frame end the reference template generates a sensible variation of features (i.e., low matching score), then the algorithm refreshes the reference template and splits the video producing a new segment. To make the matching more reliable, we reject the matchings where the keypoints are too far in terms of spatial coordinates by assuming smooth transition between consecutive frames [14]. For major stability, a new template can be defined only

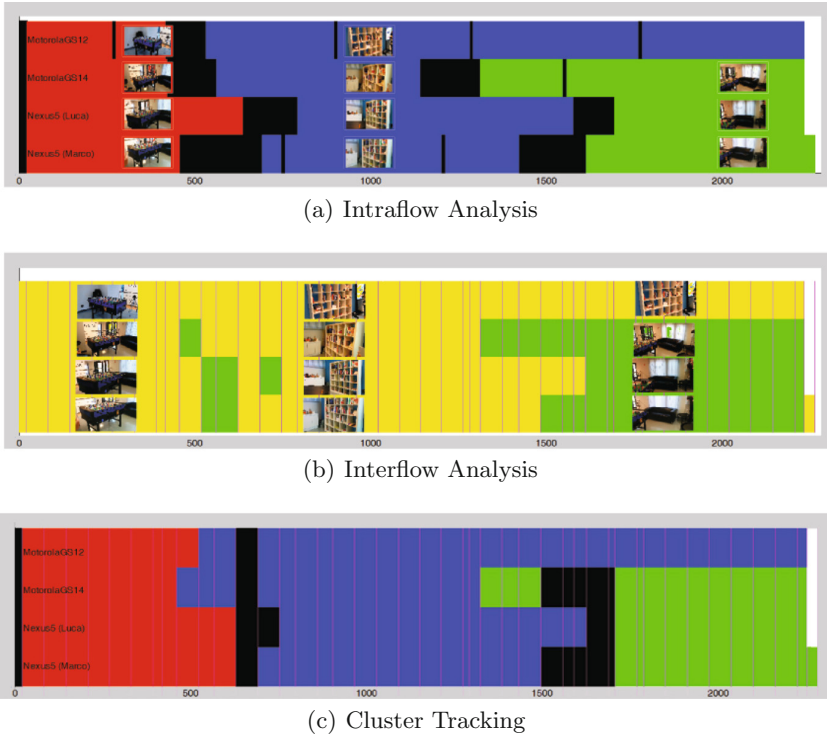


Fig. 1. RECfusion results applied on Foosball dataset. The chronograms show the results of the three main steps of RECfusion (intraflow analysis, interflow analysis and cluster tracking). Foosball dataset is composed by 4 video streams having a duration of ~ 2300 frames (~ 90 s). Each video stream is represented as a row in the chronograms. Vertical red lines mark the end of time-slots. (a) Intraframe analysis: red, blue and green frames are respectively the first, second and third scene of each video stream. Noisy frames are depicted in black. (b) Interframe analysis: yellow and green clusters are respectively the first and second cluster of each time-slot. (c) Cluster tracking: red, blue and green clusters are respectively the first, second and third cluster of the whole video set. Noisy clusters are depicted in black. (Color figure online)

if it has a duration greater than 2s, otherwise it is considered as noise. In other words, a template is considered a stable template if the number of matching SIFTs do not change too much in time. A backward checking is required in order to understand if a new defined template regards a new scene or it is related to a previously observed one. The algorithm compares the new defined template with the past ones, starting from the last found template. Each reference template is labeled with a *SceneCounter* and all video frames achieving a robust match are labeled with the same *SceneCounter*. Note that all the frames required to decide if a template should be considered as a new or an updated one are labeled as a transition interval.

3.2 Interflow Analysis

The interflow analysis is computed for each time-slot. It segments video frames labeled by intraflow analysis and assigns a *ClusterCounter* with respect to all the video streams in that specific time-slot (Fig. 1(b)). We want to group together the devices that are looking at the same scene over time. The descriptor used in the interflow analysis is based on weighted color histograms [15]. In this context the device invariance should be granted as well as possible. For this reason we firstly apply an histogram equalization, as suggested in [1]. The equalization is followed by a quantization of the color space (8 colors for each channel). The weights are obtained by using a gradient map as suggested in [15]. The gradient map is useful to highlight the structures of the objects that appear in the scene, making more robust the descriptor.

The different scenes obtained with the intraflow analysis could be considered as nodes of a complete graph in which arcs are weighted with the interflow distances between the scenes acquired by the devices. The clustering procedure selects a frame among the unclustered frames and assigns it to the most similar cluster. We used an average linkage approach to compare a frame with a cluster: the distance between a frame and a cluster is given by the average distance between the frame and all the elements within the cluster [14].

3.3 Cluster Tracking

To understand the meaning of the Cluster Tracking module we have to step back to intraflow analysis. The intraflow analysis segments the sequence of frames of a single video stream, and assigns a *SceneCounter* to each segmented scene. However, frames taken by two different video streams but labeled with the same *SceneCounter* can represent different scenes, since *SceneCounters* are discriminative only within a single video stream. The interflow analysis segments video frames in a time-slot and assigns a *ClusterCounter* to the scenes of the video streams. Interflow analysis exploits the *SceneCounters* and the set of SIFT features templates from intraflow analysis. Similarly to *SceneCounters*, the *ClusterCounters* are to be considered only within a single time-slot. Therefore, we developed a cluster tracking procedure in order to track the clusters representing the same scene in every video stream and time-slot (Fig. 1(c)). In [16] a Graphical User Interface implementing the cluster tracking typical video player commands (like Start, Pause, Stop, ...) is described (Fig. 2).

We propose a cluster tracking procedure based on a voting routine that combines the results of the intraflow and interflow analyses. Once interflow procedure has assigned a *ClusterCounter* to several *SceneCounters*, this set of scenes will characterize the same cluster also in further time-slots, so cluster tracking procedure an unique *LoggedCluster_{ID}* to this set of scenes. Differently from the *ClusterCounters*, the *LoggedCluster_{ID}s* are intended to be always discriminative. Cluster tracking procedure tracks the clusters in each time-slot assigning them *TrackedCluster_{ID}s* equals to the most similar *LoggedCluster_{ID}*. In order to define the most similar *LoggedCluster_{ID}*, cluster tracking procedure

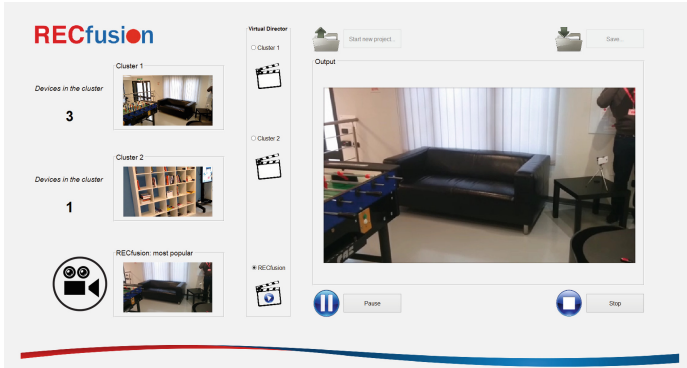


Fig. 2. RECFusion Graphical User Interface showing the Cluster Tracking framework. On the left, active clusters with respective amount of recording devices and automatically suggested video stream (called *RECFusion: most popular*) are shown. User can browse the Virtual Director panel to dynamically change the active video stream. On the right side, active video stream with classic video player commands is shown.

requires an initialization phase (at first time-slot). In this phase, the assigned $LoggedClusterIDs$ are equals to the $ClusterCounters$. Then, from the second time-slot on, the clusters will be associated to an existent $LoggedClusterID$ or to a new one, depending on a voting routine. The same routine is also used to track the $LoggedClusterIDs$ with proper $TrackedClusterIDs$.

The voting routine can be divided into 2 phases: casting of vote and voting decision. In the former phase, for each time-slot, each scene votes with three different possible values: $TrackedClusterID$ at the previous time-slot, $LoggedClusterID$ or unlogged scene (V_N), if the scene is *Noise*, already logged or unlogged, respectively. Once all the votes are casted in a time-slot, then we look for a non ambiguous voting decision (i.e., a majority is found). Majority of unlogged scenes is not admitted, so in this case we simply remove these votes from the voting decision. Depending on the reached decision, new $LoggedClusterIDs$ might be instantiated, while $TrackedClusterIDs$ at current time-slot is eventually updated. We will compare the new proposed method with respect to a cluster tracking method based on a threshold T_{CT} [16]. This threshold was used as an hyperparameter to decide whenever to create a new $LoggedClusterID$ or not. The issue with this threshold employed in [16] is that its value should be fine tuned for each video set in order to achieve the best results in cluster tracking procedure.

4 Datasets

To perform experiments we have used the RECFusion dataset [14] which is publicly available at the following URL: <http://iplab.dmi.unict.it/recfusionICIAP17>. This dataset is made up of three video sets:

1. *Foosball*: indoor context, some people appear in the scene. The number of contributing devices for this video set is 4, with an average number of frames per video stream of 2250 (44 time-slots). There are three main subjects in this video set: a foosball, a couch and a bookcase.
2. *Meeting*: indoor context, two people appear in the scene. The number of contributing devices for this video set is 5, with an average number of frames per video stream of 2895 (60 time-slots). There are two main subjects in this video set (the two guys).
3. *S. Agata*: outdoor context, lots of people appear in the scene. The number of contributing devices for this video set is 7, with an average number of frames per video stream 1258 (34 time-slots). There are two main subjects in this video set: the reliquary of S. Agata and the facade of a church.

In the experiments we exploit also a video set from the dataset used in Ballan et al. [17]. This dataset is called *Magician*. It is related to an indoor context, where one person appear in the foreground. The number of contributing devices for this video set is 6, with a fixed number of 3800 frames per video stream (77 time-slots). There are two main points of view in this video set: one above and one in front of the magician. We have chosen *Magician* video set because it is slightly different from the videos currently in RECFusion dataset. In *Magician* all the video streams are focused on a single target and are acquired as a “casual multi-view video collection” [17]. This means that backgrounds in the video streams are very different from each other and that severe camera motion could often appear. The casually filmed events represent a challenging scenario for detector like SIFT (exploited in our intraflow analysis, see Sect. 3.1), so we add *Magician* video set to our tests in order to stress and evaluate scene analysis and cluster tracking performances. We have also compared the obtained results with the benchmark dataset proposed in Hoshen et al. [5]. This dataset has been acquired with wearable devices and, like *Magician* video set, it is challenging since every video is strongly affected by motion.

5 Experimental Settings and Results

We select the last instant of time for every time-slot as the representative of that interval. Validation are made exploiting the Ground Truth related to these representative frames. To evaluate the performances of the proposed method, we compute the two quality measures described in [14]. Specifically, for each clustering step we consider:

- P_r : ground truth popularity value (number of cameras looking at the most popular scene) obtained from manual labelling;
- P_a : popularity score computed by the system (number of the elements in the popular cluster);
- P_g : number of the correct videos in the popular cluster (i.e., the number of inliers in the popular cluster).

From the above defined scores, the weighted mean of the ratios P_a/P_r and P_g/P_r over all the clustering steps are computed. The ratio P_a/P_r provides a score for the popularity estimation, whereas the ratio P_g/P_r verifies the visual content of the videos in the popular cluster and provides a measure of the quality of the popular cluster. Note that P_a/P_r is a score: when is lower than 1 it means that system is under-estimating the popularity of the cluster, while, conversely, if it is higher than 1 it results in an over-estimation.

Table 1. Validation results of popularity estimation.

Scenario	Devices	Models	P_a/P_r	P_g/P_r
Foosball	4	2	1.02	1
Meeting	2	2	1.01	0.99
Meeting	4	4	0.99	0.95
Meeting	5	5	0.89	0.76
SAGata	7	6	1.05	1
Magician	6	6	0.73	0.73
Concert [5]	3	1	1.06	1
Lecture [5]	3	1	1.05	0.86
Seminar [5]	3	1	0.62	0.62

The results of the comparison between the tested video sets are shown in Table 1. The first five rows are related to RECFusion dataset, whereas the last three rows are related to the dataset proposed in [5]. Although the constantly head motion of the wearable recording devices in videos from [5], the framework reaches good results and seems to be promising room for improvement in the field of wearable devices. Conversely, we found a drop in the performances when there is a severe difference of scale between videos in a video set. Indeed, we exploited *Meeting* video set to evaluate the drawback in performances when there are high differences between resolution of devices. We compared three cases, with 2, 4 and all the 5 devices in *Meeting* video set, respectively. Other analysis outputs could be found at the following URL <http://iplab.dmi.unict.it/recfusionICIAP17>.

In the new proposed procedure we removed the threshold T_{CT} , used as an hyperparameter to decide whenever to create a new logged-cluster or not. In [16] the value of T_{CT} was empirically set equals to 0.15 founding the best overall value between True Positive Rate, True Negative Rate and Accuracy of clustering tracking procedure on RECFusion dataset. In Fig. 3 a comparison between the average values of TPR (*True Positive Rate, or Recall*), TNR (*True Negative Rate, or Specificity*) and ACC (*Accuracy*) of RECFusion dataset and *Magician* video set whit several values of T_{CT} is shown. As can be seen, the value of T_{CT} equals to 0.15 is not the best value to be used by cluster tracking procedure, while $T_{CT} = 0.5$ should be used instead. For this reason we proposed the new

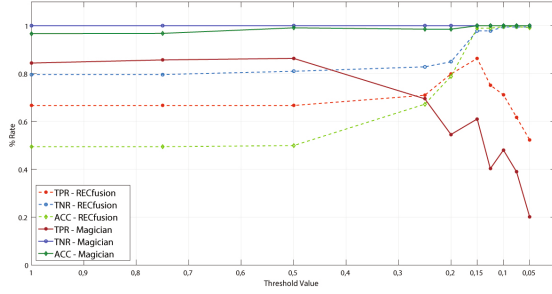


Fig. 3. A comparison of TPR (True Positive Rate, or Recall), TNR (True Negative Rate, or Specificity) and ACC (Accuracy) between RECfusion_dataset.2015 and *Magician* video set cluster tracking validations using the threshold-based procedure from [16]. As can be seen, *Magician* requires a fine tuned threshold to increase TPR , TNR and ACC values.

threshold independent cluster tracking procedure described in Sect. 3.3. We computed TPR , TNR and ACC values for each video set described in Sect. 4 and compared them with the results obtained in [16]. The comparative validation results are shown in Table 2.

Table 2. Validation results between cluster tracking procedure threshold-based and vote-based.

DS	Scene	TPR (Recall)		TNR (Specificity)		ACC (Accuracy)	
		[16]	Proposed	[16]	Proposed	[16]	Proposed
Foosball	1	0,91	0,92	0,70	1,00	0,69	1,00
	2	0,69	0,97	0,98	0,91	0,99	0,97
	3	0,41	0,74	1,00	1,00	0,50	1,00
	Mean	0,67	0,87	0,89	0,97	0,73	0,99
Meeting	1	0,99	1,00	1,00	1,00	1,00	1,00
	2	0,80	1,00	0,95	0,93	0,83	0,67
	3	0,43	0,50	1,00	1,00	0,70	1,00
	Mean	0,74	0,83	0,98	0,98	0,84	0,89
S.Agata	1	0,71	1,00	1,00	1,00	1,00	1,00
	2	0,87	0,97	0,49	0,14	0,80	0,68
	3	0,48	0,00	1,00	1,00	0,60	0,00
	Mean	0,69	0,66	0,83	0,71	0,80	0,56
Magician	1	0,73	1,00	1,00	1,00	1,00	1,00
	2	0,45	0,56	1,00	1,00	0,98	0,91
	Mean	0,59	0,78	1,00	1,00	0,99	0,96

These results show that the proposed vote-based cluster tracking procedure reaches TPR values much higher than the threshold-based procedure, while results on TNR and ACC are comparable between the two procedures. Just in the *Meeting* video set the proposed vote-based procedure is slightly outperformed: this is a limitation of the procedure. Indeed, cluster tracking procedure relies on intraflow analysis, so if the latter defines N scenes, then the former is able to distinguish at most N scenes. Hence, differently by threshold-based procedure used in [16], that can generate a bunch of small sparse clusters if T_{CT} is not fine tuned, in this case only a limited number of clusters is tracked. In *Meeting* video set two people are recorded and there are only two distinguished clusters focusing on each one of them. Sometimes interflow analysis generates a cluster containing both of the two people. This is treated by the cluster tracking vote-based procedure as *Noise*, since intraflow analysis has never labeled a scene in which the people are recorded together.

A final remark is about *Magician* video set. We added it to our dataset in order to evaluate scene analysis and cluster tracking performances in a video collection with a single scene, where all the user are focused on the same target and videos are affected by severe camera motion. Cluster tracking results with threshold-based procedure from [16] are really bad, indeed we got the worst average performance on this video set (Table 2). On the other hand, the proposed vote-based procedure reached good values of TPR , further assessing the soundness of this new cluster tracking approach. The output videos showing the result of cluster tracking vote-based procedure could be found at the following <http://iplab.dmi.unict.it/recfusionICIAP17>.

6 Conclusion and Future Works

In this paper we described RECFusion, a framework designed for automatic video curation driven by the popularity of the scenes acquired by multiple devices. Given a set of video streams as input, the framework can group these video streams by means of similarity and popularity, then it automatically suggests a video stream to be used as output, acting like a “virtual director”. We compared RECFusion intraflow and interflow analysis validations with Hoshen [5]. We have added a video set from Ballan et al. [17] to our RECFusion dataset showing that RECFusion is capable of recognize and track the scenes of a video collection even if there is a single scene, where all the user are focused on the same target and videos are affected by severe camera motion. We proposed a novel and alternative vote-based cluster tracking procedure and compared it with the one, threshold-based, described in [16]. From this comparison we found that vote-based procedure reaches very good results totally automatic and independently by a hyperparameter fine tuning phase, but with the tradeoff of be unable to create and track an unlimited number of clusters. As future works and possible applications, we are planning to augment the framework with features specifically focused on Assistive Technology or Security issues (i.e., highlight/track bad behaviour in the life style, log the visited places, search something or someone that appears in the scene).

Acknowledgments. This work has been performed in collaboration with Telecom Italia JOL WAVE in the project FIR2014-UNICT-DFA17D.

References

1. Finlayson, G., Hordley, S., Schaefer, G., Tian, G.Y.: Illuminant and device invariant colour using histogram equalisation. *Pattern Recogn.* **38**(2), 179–190 (2005)
2. Finlayson, G., Schaefer, G.: Colour indexing across devices and viewing conditions. In: *International Workshop on Content-Based Multimedia Indexing* (2001)
3. Arev, I., Park, H.S., Sheikh, Y., Hodgins, J., Shamir, A.: Automatic editing of footage from multiple social cameras. *ACM Trans. Graph.* **33**, 81 (2014)
4. Park, H.S., Jain, E., Sheikh, Y.: 3D social saliency from head-mounted cameras. In: *Advances in Neural Information Processing Systems*, pp. 431–439 (2012)
5. Hoshen, Y., Ben-Artzi, G., Peleg, S.: Wisdom of the crowd in egocentric video curation. In: *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 587–593 (2014)
6. Saini, M.K., Gadde, R., Yan, S., Ooi, W.T.: Movimash: online mobile video mashup. In: *ACM International Conference on Multimedia*, pp. 139–148 (2012)
7. Bano, S., Cavallaro, A.: ViComp: composition of user-generated videos. *Multimedia Tools Appl.* **75**, 7187–7210 (2016)
8. Mittal, A., Moorthy, A.K., Bovik, A.C.: No-reference image quality assessment in the spatial domain. *IEEE Trans. Image Process.* **21**(12), 4695–4708 (2012)
9. Nagasaka, A., Miyatake, T.: Real-time video mosaics using luminance-projection correlation. *Trans. Inst. Electron. Inf. Commun. Eng.* **82**(10), 1572–1580 (1999). <http://ci.nii.ac.jp/naid/110003183527/en/>. ISSN 09151923
10. Farinella, G.M., Ravi, D., Tomaselli, V., Guarnera, M., Battiato, S.: Representing scenes for real-time context classification on mobile devices. *Pattern Recogn.* **48**(4), 1086–1100 (2015)
11. Farinella, G.M., Battiato, S.: Scene classification in compressed and constrained domain. *Comput. Vis.* **5**(5), 320–334 (2011)
12. Naccari, F., Battiato, S., Bruna, A., Capra, A., Castorina, A.: Natural scenes classification for color enhancement. *IEEE Trans. Consum. Electron.* **51**(1), 234–239 (2005)
13. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **60**(2), 91–110 (2004)
14. Ortis, A., Farinella, G.M., D’Amico, V., Adesso, L., Torrisi, G., Battiato, S.: Refusion: automatic video curation driven by visual content popularity. In: *ACM Multimedia, MM 2015*, pp. 1179–1182. ACM (2015)
15. Domke, J., Aloimonos, Y.: Deformation and viewpoint invariant color histograms. In: *British Machine Vision Conference*, pp. 509–518 (2006)
16. Milotta, F.L.M., Battiato, S., Stanco, F., D’Amico, V., Torrisi, G., Adesso, L.: RECFusion: automatic scene clustering and tracking in video from multiple sources. In: *EI - Mobile Devices and Multimedia: Enabling Technologies, Algorithms, and Applications* (2016)
17. Ballan, L., Brostow, G.J., Puwein, J., Pollefeys, M.: Unstructured video-based rendering: interactive exploration of casually captured videos. In: *ACM Transactions on Graphics*, pp. 1–11 (2010)