DOI: 10.2352/ISSN.2470-1173.2016.7MOBMU-292

RECfusion: Automatic Scene Clustering and Tracking in Videos from Multiple Sources

Filippo L.M. Milotta[1], *Sebastiano Battiato*[1], *Filippo Stanco*[1], *Valeria D'Amico*[2], *Giovanni Torrisi*[2], *Luca Addesso*[2]
[1] *Department of Mathematics and Computer Science, University of Catania, Italy*
[2] *Telecom Italia JOL WAVE, Catania, Italy*

Abstract

RECfusion is a framework devoted to the automatic processing of video data from many devices, as smartphones, tablets, webcams, surveillance cameras, etc., where all devices are thought to be connected into a 4G LTE network. Exploiting this mobile ultra-broadband connection the communication paradigm between users in the social media context can be augmented: in events like concerts, feasts, expos and so on, users become either producers than fruitors of video data. RECfusion analyzes video streams from several devices and infers semantics performing scene understanding. Key scenes are identified with relation on each video stream and all the other ones; then the system generates a video rendered from a mixage of the selected video streams. In ref. [1] a system based upon visual content popularity has been already implemented in RECfusion. In this work we propose an extension for RECfusion: a novel automatic video cluster tracking algorithm able to identify the different scenes in the gathered video streams selecting for each of them the best recording device.

Keywords

Data Mining; Scene Understanding; Computer Vision; Prosumer Multimedia Stream; Cloud-based Application; Mobile Ultra-broadband Network.

Introduction

The automatic processing of video data from many devices, as smartphones, tablets, webcams, surveillance cameras, etc., where all these devices are thought to be connected into an ultra-broadband network, like a 4G LTE, is not a trivial issue. In the real-time context, big data in the video domain have to be processed and smart data mining should infers the right semantic from video streams. Related application scenario exploiting multi-video semantic retrieval could be multiple: for instance, the information inferred from several devices could be useful in the cultural heritage restoration scenario, since retrieved data could be used to perform interpolation on missing data [4–7]. Another useful application could be implemented with the purpose of augment assistive technology devices, introducing data-processing from multiple sensors, like the depth-cameras mounted on Electronic Travel Aid systems [8–10]. Furthermore, the social media context definitively represents a key factor: video streams are gathered in a crowdsourcing paradigm, so they can be processed to capture and represent the mood of the crowd. For instance, exploiting this process, information about what is really salient from the point of the view of the audience can be retrieved; once that the most salient subject is identified, then it is possible to further investigate the reasons that has generated a so great focus on it. This foundings might be used as a powerful cue to make specific improvements to change the degree of appeal of the scene.

In literature, several solutions have been proposed in the field of multi-device video inference crowd-saliency driven [11–14]. However, these works expect strict prior conditions to solve the problems related with the use of different recording devices, indeed they use the same model of device for each experimentation or manually set the number of scenes of interest, introducing bias. Some works [11–13] exploit a high time consuming 3D scene reconstruction technique based upon "structure from motion" to find position and direction of each acquiring device. In ref. [14] the framework MoViMash tries to replicate the behavior of a movie director implementing a framework that learn from labeled sets of video frames how to and when switch scenes, nevertheless this is a technique hardly adaptable in a real-time context, since learning phase should be tuned to every different recorded scenes.

In ref. [1] the RECfusion framework is presented for the first time: video streams by multi-device are analyzed with the aim of finding the most popular video stream, representing the strongest "mood" of the crowd. In the present work, we augment RECfusion framework with the functionality of tracking all the identified scenes in video streams, not only the most popular one at each time, introducing scenes story log and allowing the selection of scenes of interest between the available ones.

The main aims of the proposed RECfusion framework extension are: firstly, analysis of video streams from multi-source multi-device context, secondly, identification of the scenes of interest through clustering of video sequences, and finally, time tracking of the computed scenes clusters. These tasks are synchronization between video stream uploaded by multiple users, how to perform clustering of video scenes and cluster recognition for the time tracking purpose. Indeed, the clustering phase runs every time slot, while clusters of different time slots have to be properly matched since scenes (and clusters) could change within each video stream.

The rest of the paper is structured in the following way: first of all an overview of RECfusion framework is given, then the proposed framework extension is discussed. Secondly, some experimental results are shown to asses the soundness of the proposed cluster tracking method. Finally, we propose some future works and conclusions.

RECfusion Overview

In the RECfusion framework, developed by Telecom Italia JOL Wave in collaboration with the University of Catania [1, 2],

DOI: 10.2352/ISSN.2470-1173.2016.7MOBMU-292

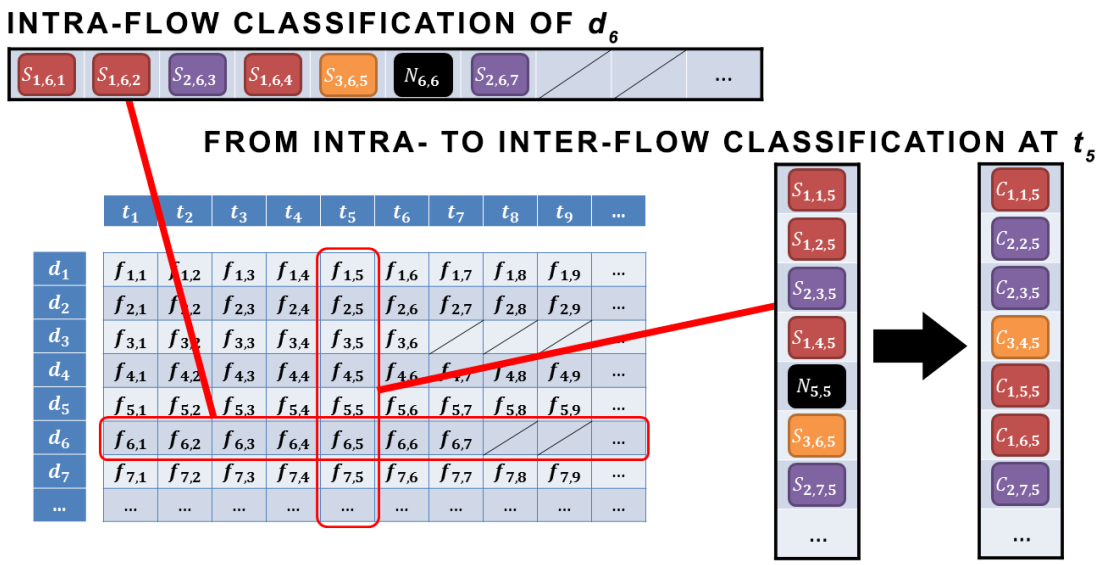

Figure 1. *RECfusion intra- and inter-flow classifications. Video frames are represented in a timetable. On the upper-side, the intra-flow classification of d_6 is shown; in the array the frames are labeled according to the definition of intra-flow classified frame, so, for instance, the third label refers to scene 2 of device 6 at time 3. On the right-side, starting from an example of intra-flow classification, the inter-flow classification at t_5 is shown; in the array the frames are labeled according to the definition of inter-flow classified frame, so, for instance, the third element of the array refers to a frame in cluster 2, acquired by device 3 at time 5.*

video streams multi-source multi-devices are gathered and synchronized by a server. Frames are processed in every time-slot with algorithms that normalize color data, in order to increase the chance of recognize the same scene between the very different recording devices [15–18].

Two kinds of scene classification can be defined: intra-flow and inter-flow ones. Intra-flow classification routine takes video frames of a single video stream as input, and returns the same set of frames labeled with respect to the classified scenes within the single video stream. Then, every time slot, inter-flow classification routine takes, from all the video streams, video frames labeled by intra-flow classification as input, and returns the same set of frames clustered with respect to the classified scenes within all the video streams. In this way, RECfusion can distinguish the scenes in a video stream through the intra-flow classification, and can exploit the inter-flow classification to find which devices are acquiring the same scene, since they are clustered in the same cluster. Video frames f can be represented in a timetable in the following way (Figure 1): video streams from different devices d are inserted in the timetable row-wise, while time-slots t are counted column-wise. When a video stream ends, consecutive video-frames of the row are marked with slashed cells. Each intra-flow classified frame S is labeled with a scene ID, a device ID and a time-stamp. When a frame is classified as noise it is marked with a N label, and no scene ID is assigned. Similarly to intra-flow classified frame S, each inter-flow classified frame C is labeled with a cluster ID, a device ID and a time-stamp.

A new time slot is defined each time the "configuration of scenes" changes, that is when a scene acquired by a device changes or noise occurs. However a new time slot is also defined if a configuration of scenes persists for more than n frames. Since in Ref. [1] this number of frames is set to $n = 61$, assuming a frame rate of 30 *fps*, it is guaranteed that at most every 2 seconds a new time slot is defined, and a new device is elected as the most representative between the set of acquiring devices. The most representative device is defined as the closest to the centroid of the cluster with the highest number of devices, properly called for this reason "the most popular cluster". Indeed, this cluster chosen on its visual content popularity could be used as a powerful cue to understand the "mood" of the crowd.

RECfusion Proposed Extension

RECfusion, as described in the previous Section, selects the most representative video device, keeping the focus on what is most popular in that moment, just like a real director could do. However, one could also be interested on what happens in the less popular scenes, so it could be useful to track all the clusters upon time and eventually have the possibility to inspect a specific one of them. Indeed, we define this proposed extension to RECfusion framework as *cluster tracking*, since once a new scene cluster is defined in a time slot, then it can also be recognized in another successive time slot too.

To exploit inter-flow classification, clusters are labeled with a cluster ID, however it is important to notice that clusters with same cluster IDs but different timestamps could not be the same semantic cluster (could not represent the same scene). After all, also scenes labeled by intra-flow classification with same scene IDs but different device IDs could not represent the same acquired scene. So, inter-flow classification identifies intra-flow classified frames that represent the same scene in a time-slot, while clus-

DOI: 10.2352/ISSN.2470-1173.2016.7MOBMU-292

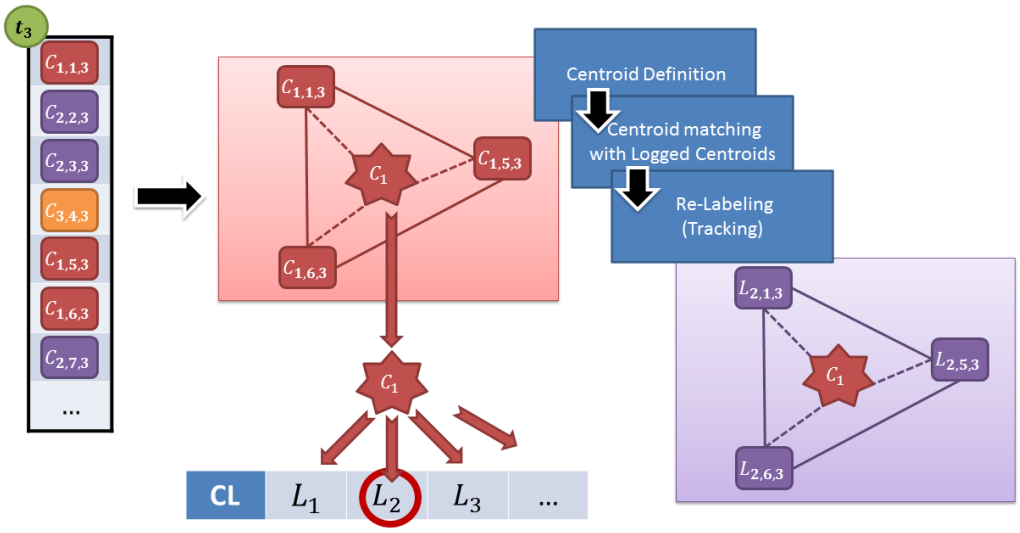

Figure 2. *Cluster tracking: logged-clusters matching. Starting from an example of intra-flow classification at t_3, we consider cluster C_1 and represent it in red. Then a centroid (the red star) for cluster C_1 is defined. The computed centroid is compared with logged-centroids stored in cluster log (marked with CL). In this case, a match occurs with logged-cluster L_2 (represented in purple).*

ter tracking identifies inter-flow classified clusters that represent a scene in the whole acquisition session.

Cluster tracking procedure has an initialization phase in which we define a cluster log, that is a list in which centroids of tracked clusters are stored. For this reason, we refer to stored centroids (and then clusters) as *logged-centroids*. For each time-slot, clusters computed by inter-flow classification are compared with the logged ones and if a match occurs then the cluster ID label is changed in the matched logged-cluster ID label (Figure 2). Each cluster-tracked frame is labeled with logged-cluster ID, a device ID and a time-stamp. Obviously, centroids are not re-labeled, since we are changing the labels of the cluster, not its frames; centroids will be used in the next phase of logged-centroid update. Exploiting cluster tracking it is possible to found that all devices in a cluster are acquiring a scene identified in the whole video by a specific logged-cluster. Finally, logged-centroids are updated and/or if the computed cluster did not match any already existing logged-cluster then new ones are defined.

Cluster Tracking Method

More in detail, for each time slot, when a new cluster is identified through Intra-flow classification we try to match this cluster with the logged ones. In order to compare clusters, we compare centroids, that are represented by equalised histograms [15]. As for Intra-flow and Inter-flow classification [1], also for cluster tracking we use the definition of distance between equalised histograms employed by [19]:

$$d(h_{LC},h_C) = \frac{\Sigma(h_{LC}-h_C)^2}{\Sigma(h_{LC})^2} \qquad (1)$$

where h_{LC} and h_C are the weighted histograms related to logged-centroid and centroid in evaluation. Accordingly to this definition, distance could range between 0 (minimum) and 2 (maximum), so we consider two centroids to correctly match if they have a distance lesser then a threshold T_{LC}. Now, we can distinguish between logged-clusters matching and logged-clusters update.

Logged-Clusters Matching

An example of logged-clusters matching procedure is shown in Figure 2. As said, we try to match centroids of each time slot with logged-centroids. If at least one logged-centroid positively match, then it will be used in the tracking procedure: all frames labeled with the same cluster ID at this time slot are re-labeled with the matched logged-cluster ID. If several logged-centroids positively match, then the closest one will be considered the matching one. Notice that matched logged-centroids cannot be matched more times by the other centroids of the same time-slot: in this way multiple matches of the same logged-centroids are simply avoided. However, we randomize the order in which centroids of a time-slot are compared with the logged-centroids, so the clusters with lower cluster-IDs (that are the firsts to be identified) have no significantly advantages with respect to other ones. On the other hand, if none logged-centroid matches, then a new one is created, initialized with the histogram values of centroid in evaluation. Finally, matched logged-centroid have to be updated.

Logged-Clusters Update

We defined a cluster centroid as the average histogram of its frame histograms. When a new centroid is created, we use an equally weighted mean, so every frame histograms equally participate to its creation. If a logged-centroid matched a centroid during the matching phase, then it has to be updated. For each logged-centroids, we store in the cluster log also a counter for the

DOI: 10.2352/ISSN.2470-1173.2016.7MOBMU-292

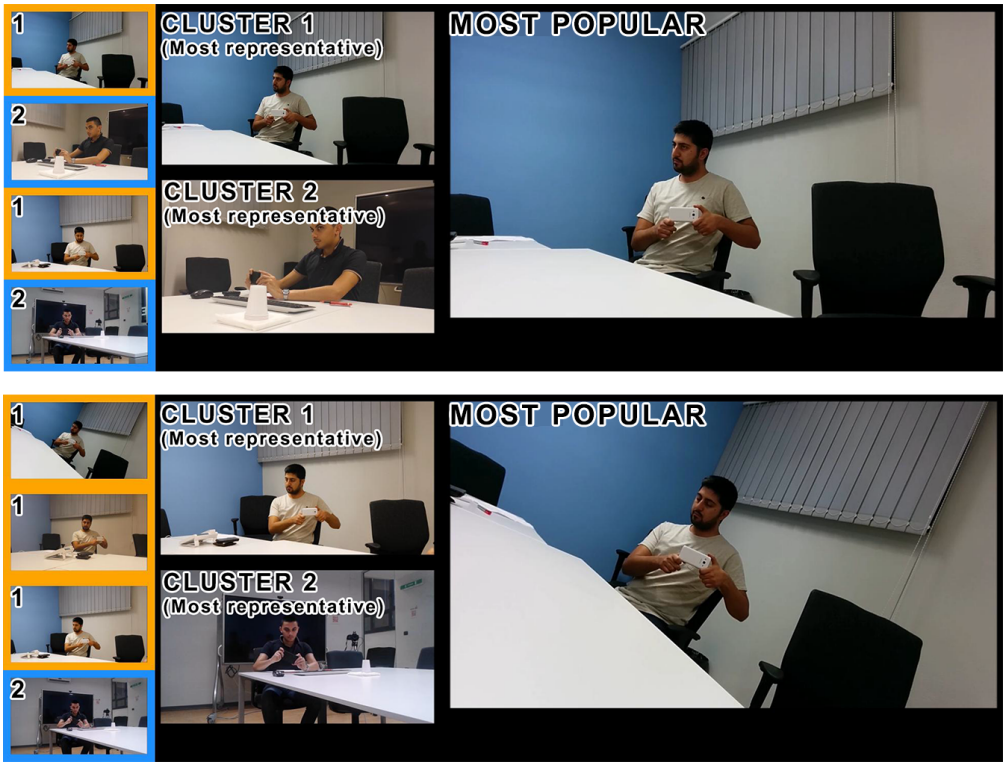

Figure 3. *Example of Cluster Tracking on two different time slots of video sequence "Meeting" (from RECfusion dataset [2]).*

quantity of histograms that had contributed to it. Through this counter we can update logged-centroid using a properly weighted mean between logged-centroid and centroid in evaluation, where the weights are the counter itself for the former and the number of histograms (frames) for the latter. Counter of histograms in the cluster log could be clearly much higher than the number of histograms in the re-labeled cluster: actually it continuously raise in time, making centroids of re-labeled clusters even less influencing in the updating. In other words, logged-centroids become more stable with increasing number of updates.

Experimental Results

In our experiments we have planned several experimentations to process multi-source multi-device videos from different scenarios and used the proposed framework to infer the main semantic clusters contained in the video streams. In particular, we have used RECfusion Dataset (2015) [1,2]. We have manually labeled this dataset, in order to define a set of ground truth clusters to estimate the quality and soundness of the proposed approach. The new RECfusion Dataset with labeled cluster consists of 3 scenarios with 3 clusters for each one of them. Moreover, the dataset has approximatively an overall of 7000 frames subdivided in 150 time slots.

In Figure 3 is shown a comparison of cluster tracking results on two different time slots of a video. On the left side, video streams are highlighted in yellow or blue with respect to the tracked cluster. Then, the most representative (the nearest to cluster centroid) video stream for each cluster and for the most popular cluster (as defined in Ref. [1]) is shown. Notice how the video stream chosen by [1] could differ from the one suggested by this paper, since in this extension we have introduced the logged-clusters update phase. In Figure 4 another comparison of cluster tracking results is shown: in this case three clusters (yellow, blue and red) have been tracked.

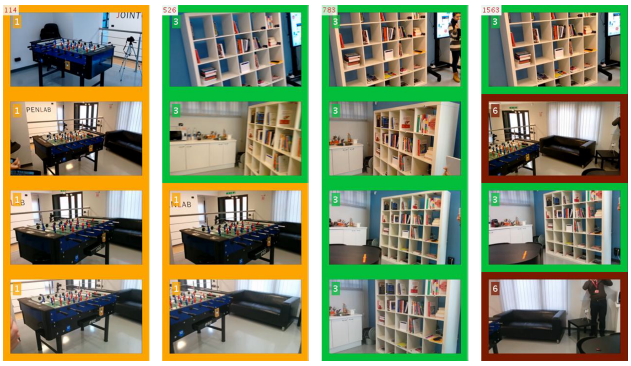

Figure 4. *Example of Cluster Tracking on four different time slots of video sequence "Foosball Room" (from RECfusion dataset [2]).*

As explained in Cluster Tracking Method description, the comparison phase between computed centroids (clusters) and logged ones implies a definition of distance between centroids and, consequently, of a threshold T_{LC}. We supposed that almost near (similar) centroids, hence clusters, have to be considered as the same semantic cluster. Although $T_{LC} = 1$ in ref. [1], in our experimentations we exploited our new manually labeled dataset

DOI: 10.2352/ISSN.2470-1173.2016.7MOBMU-292

to investigate the existence of a better threshold value for cluster tracking purpose. We tested several thresholds values and then we compared True Positive Rate (TPR, also called Recall), True Negative Rate (TNR, also called Specificity) and Accuracy of the Cluster Tracking Method for each tested threshold. Experimental results are shown in Figures 5, 6 and 7. Experimentally, we derived the best threshold value for T_{LC} equal to 0.15, in which we gained the highest mean TPR value (86%) and among the highest mean TNR (98%) and mean Accuracy (99%) values.

We have uploaded ground truth labels, output videos and validation results on [3].

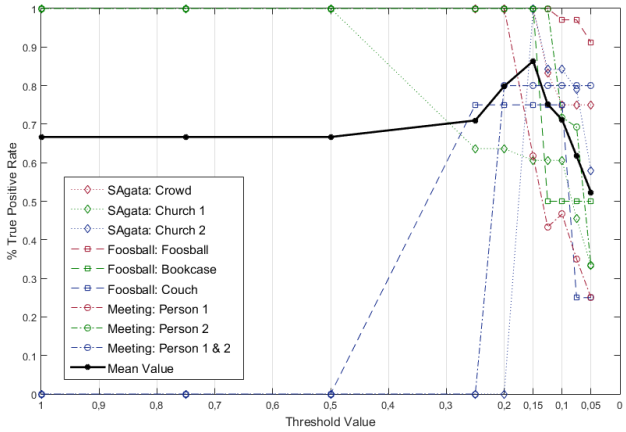

Figure 5. *RECfusion Cluster Tracking Recall (TPR).*

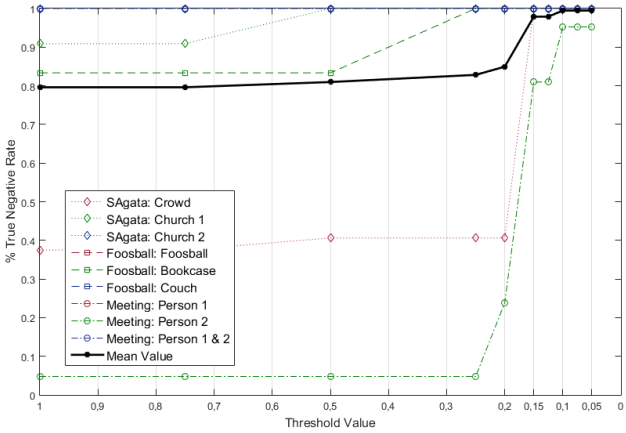

Figure 6. *RECfusion Cluster Tracking Specificity (TNR).*

Graphical User Interface

We have developed a Graphical User Interface for the proposed RECfusion extension, allowing the user to select scenes of interest between the available ones. A screenshot of the GUI is shown in Figure 8. Typical video player commands (like Start, Pause, Stop, ...) have been inserted. Active clusters are shown on the left, together with the RECfusion [1] output on the bottom; user can select one of the active clusters from the "Virtual Director" panel and dynamically view it on the right side of the GUI switching from a cluster to another. The selected video stream is

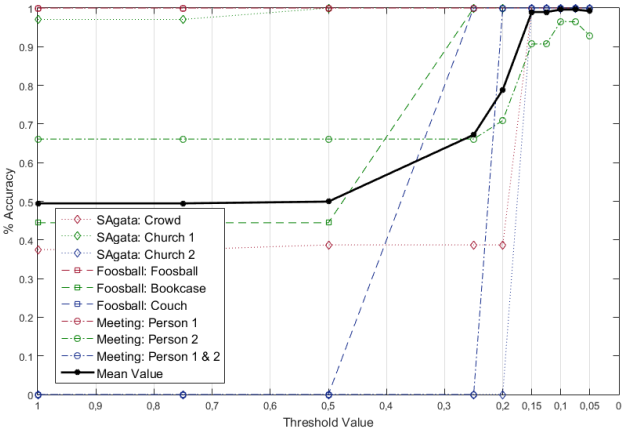

Figure 7. *RECfusion Cluster Tracking Accuracy.*

redirect to the output stream, so it could be eventually recorded as it has been mounted by the user. In the social media context, this functionality could enhance user-experience during the navigation of the video streams gathered by the community. Anyone might virtually become a director: information about the number of devices in each cluster gives a cue about the popularity of that cluster, so in every moment the user is able to choose the most popular cluster from the point of view of the community or the most interesting cluster from his own point of view.

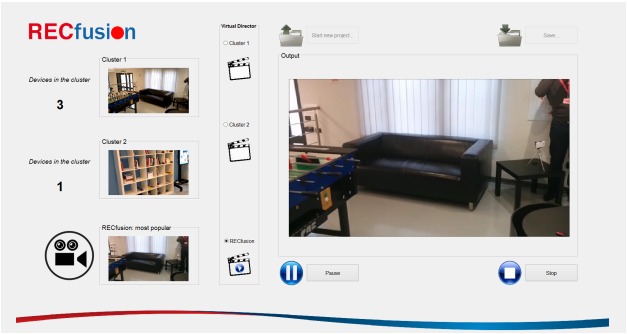

Figure 8. *A screenshot of the RECfusion Graphical User Interface showing the Cluster Tracking framework.*

Conclusion and Future Works

In this work we have presented an extension for already existing RECfusion framework [1,2]. A novel automatic video cluster tracking algorithm, allowing identification of different scenes in the gathered video streams and selection for each of them of the best recording device, has been proposed. The Cluster Tracking Method has been described, and the definition of Cluster Log as a list of representative centroids has been given. We stressed the attention that a proper selection of a threshold T_{LC} in the cluster matching process allows a better tracking performance.

As future works, we are planning to add some other functionalities, like the ones concerning on Assistive Technology or Security issues. For instance, RECfusion might be used exploiting wearable cameras to answer question like "How much time

DOI: 10.2352/ISSN.2470-1173.2016.7MOBMU-292

have I spent in a specific room?", or exploiting surveillance cameras it might give an advice on when the scenario is subject to some changes. Also the possibility to proper enhance input video stream will be considered [20–22]. Further useful studies will be definitively conducted in these fields.

Acknowledgments

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

References

[1] A. Ortis, G.M. Farinella, V. D'Amico, G. Torrisi, L. Addesso, and S. Battiato, RECfusion: Automatic Video Curation Driven by Visual Content Popularity, Proc. ACM Multimedia, pg. 1179–1182 (2015).

[2] RECfusion website: http://www.recfusionproject.altervista.org

[3] RECfusion - Cluster Tracking website: http://recfusionproject.altervista.org/clustertracking.htm

[4] A. Kokaram, R. Morris, W. Fitzgerald, and P. Rayner, Detection of missing data in image sequences, J. IEEE Transactions on Image Processing 4, 11 (1995).

[5] A. Kokaram, R. Morris, W. Fitzgerald, and P. Rayner, Interpolation of missing data in image sequences, J. IEEE Transactions on Image Processing 4, 11 (1995).

[6] F. Stanco, D. Allegra, and F.L.M. Milotta, Tracking Error in Digitized Analog Video. Automatic Detection and Correction, J. Multimedia Tools and Applications (MTAP), (2014).

[7] F. Stanco, D. Allegra, and F.L.M. Milotta, Detection and Correction of Mistracking in Digitalized Analog Video, Proc. ICIAP, pg. 218–227 (2013).

[8] D. Dakopoulos, and N.G. Bourbakis, Wearable obstacle avoidance electronic travel aids for blind: A survey, J. IEEE Transaction on Systems, Man, and Cybernetics-Part C: Applications and Reviews, 40, 1 (2010).

[9] J.R. Terven, J. Salas, and B. Raducanu, New opportunities for computer vision-based assistive technology systems for the visually impaired, J. Computer, 47, 4 (2014).

[10] F.L.M. Milotta, D. Allegra, F. Stanco, and G.M. Farinella, An Electronic Travel Aid to Assist Blind and Visually Impaired People to Avoid Obstacles, Proc. CAIP, pg. 604–615 (2015).

[11] I. Arev, H.S. Park, Y. Sheikh, J. Hodgins, and A. Shamir, Automatic editing of footage from multiple social cameras, J. ACM Transactions on Graphics (TOG), 33, 4 (2014).

[12] H.S. Park, E. Jain, and Y. Sheikh, 3d social saliency from head-mounted cameras, Proc. Advances in Neural Information Processing Systems (NIPS), pg. 422–430 (2012).

[13] Y. Hoshen, G. Ben-Artzi, and S. Peleg, Wisdom of the Crowd in Egocentric Video Curation, Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pg. 587–593 (2014).

[14] M.K. Saini, R. Gadde, S. Yan, and W.T. Ooi, MoViMash: online mobile video mashup, Proc. ACM international conference on Multimedia, pg. 139–148 (2012).

[15] G. Finlayson, S. Hordley, G. Schaefer, G.Y. Tian, Illuminant and device invariant colour using histogram equalisation, J. Pattern Recognition, 38, 2 (2005).

[16] G.M. Farinella, D. Ravì, V. Tomaselli, M. Guarnera, and S. Battiato, Representing Scenes for Real–Time Context Classification on Mobile Devices, J. Pattern Recognition, 48, 4 (2015).

[17] G.M. Farinella, and S. Battiato, Scene classification in compressed and constrained domain, J. Computer Vision, 5, 5 (2011).

[18] F. Naccari, S. Battiato, A. Bruna, A. Capra, and A. Castorina, Natural scenes classification for color enhancement, J. IEEE Transactions on Consumer Electronics, 51, 1 (2005).

[19] J. Domke, Y. Aloimonos, Deformation and viewpoint invariant color histograms, Proc. British Machine Vision Conference (BMVC), (2006).

[20] A. Bosco, S. Battiato, A. Bruna and R. Rizzo, Noise Reduction for CFA Image Sensors Exploiting HVS Behaviour, J. Sensors, 9, 3 (2009).

[21] G. Puglisi, and S. Battiato, A robust image alignment algorithm for video stabilization purposes, J. IEEE Transactions on Circuits and Systems for Video Technology, 21, 10 (2011).

[22] S. Battiato, A.R. Bruna, and G. Puglisi, A Robust Block Based Image / Video Registration Approach for Mobile Imaging Devices, J. IEEE Transactions on Multimedia, 12, 7 (2010).

Author Biography

Filippo L.M. Milotta *received his MS in Computer Science (summa cum laude) in 2014 from University of Catania. He is currently a Ph.D. student with grant by Telecom Italia in "Multi device video analysis and summarization for high bandwidth connected environments".*

Sebastiano Battiato *is Associate Professor (INF/01) with the Department of Mathematics and Computer Science, University of Catania, where he is teacher of Computer Vision. His research interests include ISP Algorithm for Embedded Devices, Adaptive Techniques for Image Enhancement and Coding, Image/Video Forensics. He is a senior member of the IEEE. More info are available on* www.dmi.unict.it/∼battiato

Filippo Stanco *is Associate Professor (INF/01) with the Department of Mathematics and Computer Science, University of Catania, where he is teacher of Multimedia. His research interests include digital restoration, zooming, super-resolution, artifacts removal, interpolation, texture and GIS. More info are available on* www.dmi.unict.it/fstanco

Valeria D'Amico *is the head of the Telecom Italia Joint Open Lab. Her interests include Smart Cities, Corporate Entrepreneurship e Social Innovation. She has a Degree in Electronic Engineering at the University of Catania, and she has a Master in Business Administration achieved at the SDA Bocconi di Milano.*

Giovanni Torrisi *is a Telecom Italian researcher at the Joint Open Lab WAVE of Catania. His research interests include Wearable Devices, Mobile Design & Development, Data Visualization. He has a Degree in Computer Science (summa cum laude) at the University of Catania.*

Luca Addesso *is a Telecom Italian researcher at the Joint Open Lab WAVE of Catania. His research interests include Wearable Devices & Gesture Recognition, Mobile Multimedia and Fast Prototyping. He has a Degree in Electronic Engineering at the University of Florence.*