

# The Social Picture: Advanced Image Analysis Applications

F. L. M. Milotta<sup>1,2</sup> and M. Bellocchi<sup>1,2</sup> and S. Battiato<sup>1</sup>

<sup>1</sup>University of Catania, Department of Mathematics and Computer Science,  
Viale A. Doria 6, 95125 Catania, Italy

<sup>2</sup>Telecom Italia, JOL WAVE,  
Viale A. Doria 6, 95125 Catania, Italy

---

## Abstract

*In The Social Picture (TSP) an huge amount of crowdsourced social images can be collected and explored. We distinguish three main kind of events: public, private and cultural heritage related ones. The framework embeds a number of advanced Computer Vision algorithms, able to capture the visual content of images and organize them in a semantic way. In this paper we employ VisualSFM (VSFM) to add new features in TSP through the computation of a 3D sparse reconstruction of a collection within TSP. VisualSFM creates a N-View Match (NVM) file as output. Starting from this NVM file, which characterizes the 3D sparse reconstruction, we are able to build two important relationships: the one between cameras and points and the one between cameras themselves. Using these relationships, we implemented two advanced Image Analysis applications. In the first one, we consider the cameras as nodes in a fully connected graph in which the edges weights are equal to the number of matches between cameras. The spanning tree of this graph is used to explore images in a meaningful way, obtaining a scene summarization. In the second application, we define three kinds of density maps with relation to image features: density map, weighted-density map and social-weighted-density map. Results of a test conducted on a collection from TSP is shown.*

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Display Algorithms—Viewing Algorithms. I.3.8 [Computer Graphics]: Applications—

---

## 1. Introduction

Social networks have become increasingly useful to understand people opinion and trends. Particularly, social media have changed the communication paradigm of people sharing multimedia data: users express emotions and share experiences in social networks. In social events (e.g., parties, concerts, and sport matches) users are gradually changing in the so called prosumers, as they do not just use but also produce and share multimedia data related to what has captured their interest with mobile devices. The redundancy in these data together with annexed metadata (e.g., geolocation, tags, and mood-tag) can be exploited to infer social information about the attitude of the audience. For instance, systems such as MoVi-Mash [SGYO12], ViComp [BC16] and RECFusion [BFM\*17] are able to generate a video which describes the crowd interest starting from a set of videos by considering scene content popularity. Indeed, the popularity of a visual content is an important cue to understand the mood of crowd attending to an event or estimate how much parts of a cultural heritage are perceived as interesting. Large scale visual data from social media and other multimedia information gathered by multiple sources (e.g., mobile devices) can be processed with Machine Learning and Computer Vision algorithms in order to infer knowledge about social contexts [WL15] or organize images by visual content.

In our previous work, we presented a framework called The Social Picture (TSP) [BFM\*16], in which images are gathered from social networks or uploaded directly in the repository by users through a mobile app and a website. The framework is capable of collect, analyze and organize huge flows of visual data, and to allow users the navigation of image collections generated by the community. In TSP three categories of image collections are distinguished: social events, private events and cultural heritage landmarks. The collections are processed with several tools which include automatic clustering of images, intensity heatmaps and automatic image captioning. These tools allow TSP to provide users a number of representative image prototypes related to each stored collection, exploitable for different purposes (e.g., selection of the most meaningful pictures of a painting during a showcase in a museum). Automatic clustering is implemented using a CNN representation [ZLX\*14] and employing an AlexNet architecture [KSH12]. For each image, the  $fc7$  features are extracted and the t-SNE algorithm [MH08] is employed to compute a 2D embedding representation characterizing the pairwise distances between visual features. The intensity heatmap is another tool implemented in the framework. It consists of a map of values related to the number of collected pictures containing visual areas similar to the ones of a specific landmark building or area of interest. Users can interact with the heatmap selecting points on the map and retrieving

images that contributed to generate intensity values in that specific point. Finally, the automatic image captioning [KFF15] is a tool used to create and suggest descriptions of images, that come useful for text based queries performed by users.

In this paper we employed VisualSfM (VSFM) [Wu13] to compute visual matching between images within a cultural heritage collection of TSP and to obtain a 3D sparse reconstruction of landmarks. Using VSFM and its 3D reconstruction, we define two new advanced Image Analysis applications added to TSP. In the first one, we consider the cameras as nodes in a fully connected graph in which the edges weights are equal to the number of matches between cameras. The spanning tree of this graph is used to explore images in a meaningful way, obtaining a scene summarization. In the second application, we define three kinds of density maps with relation to image features: density map, weighted-density map and social-weighted-density map. Results of a test conducted on a collection from TSP is shown.

The paper is organized as follows. Overview of VSFM and its integration in our framework is described in Section 2, together with the definition of the Model (the data structure) employed for the process of advanced Image Analysis applications implementation. Then, we present exploration of the spanning tree representation and density maps in Sections 3 and 4, respectively. Conclusion and final discussion are drawn in Section 5.

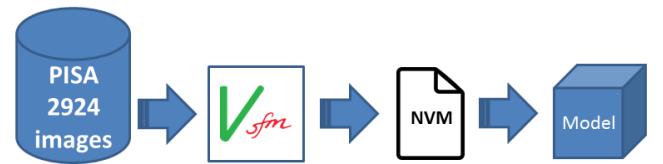
## 2. VSFM integration and Model definition

VisualSfM (VSFM) [Wu13] is a powerful tool of 3D reconstruction from a set of images, exploiting Structure from Motion (SfM), publicly available online [WoV]. VSFM maintains high accuracy by regularly re-triangulating the feature matches that initially fail to triangulate. VSFM performs a linear-time incremental SfM method [ZW15]. Hence it fits our scenario in which we want to estimate the 3D reconstruction of images within a collection, where new images could be added in any moment by users.

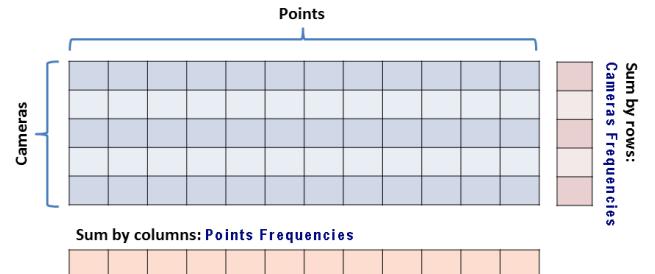
In this paper we used 2924 photos from the cultural heritage collection called *Pisa*, that was already been used as study-case in [BFM\*16] for Heatmap computation based on image retrieval and image matching.

Once images are loaded into VSFM, it extracts the visual features (SIFT [Low01] and GIST [OT01]) from them. Then, VSFM matches visual features and computes 3D sparse reconstruction. VSFM builds more than one single 3D-reconstructed model, as is possible that different scenes exist within a single dataset of photos or VSFM is not able to associate a set of coherent photos to the same model (due to too much different points of view). We chose the best reconstructed model, which is the one with the highest number of cameras. VSFM saves this result in a N-View Match (NVM) file. In order to read the NVM file we designed a proper parser. The obtained data structure has been enriched with other meta-data in our framework (e.g., GPS tag and focal length). It represents our *Model* for the process of advanced Image Analysis applications implementation (Figure 1).

The NVM file has an own template [WoV]: for each model, all cameras (images) and 3D points in the reconstruction are listed.



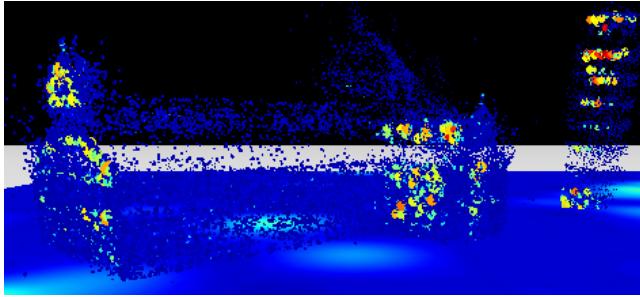
**Figure 1:** Initial workflow: images of a collection are processed with VSFM and a Model is obtained through parsing of the NVM file (3D reconstruction) and meta-data aggregation from our framework.



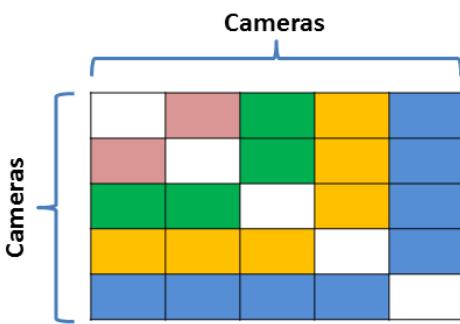
**Figure 2:** Camera – Point – Correspondences (CPC) matrix and row-wise and column-wise sums. CPC is of boolean type.

For each point there is a full list of all the cameras that “view” that point. In other words, for each point the information about which camera uses that point as visual feature for matching is stored. On the other hand, given a camera, the relationship about what are the features used by that camera is implicit. In order to make this latter relationship explicit, we define a  $C \times P$  matrix, where  $C$  is the number of cameras and  $P$  is the number of points. This matrix is named *Camera-Point-Correspondences (CPC)*. The CPC matrix is of boolean type: it has a value 1 in position  $(c, p)$  if and only if the camera  $c$  used the point  $p$  as visual feature for matching. The row-wise sum of CPC gives as result the total number of points that each camera view, while the column-wise sum gives as result how many times each point has been viewed by cameras (Figure 2). Both of these sums can be normalized w.r.t. the maximum row-wise and column-wise. The normalization is used to obtain the “frequencies of view some point” for cameras and the “frequencies of been viewed by a camera” for points. These values can be used in place of the *color vertex* of 3D points, so they can be viewed in a 3D-points frequency representation (Figure 3).

We also defined a matrix that represents relationships between cameras used in the 3D sparse reconstruction. This second matrix is named *Camera-Camera-Matches (CCM)*. For each row  $c$  in CPC we compute the sum of the logical AND between  $c$  and the other cameras  $\bar{c}$ . The so obtained values are stored in CCM, in the proper cell in position  $(c, \bar{c})$ . In this way, the matrix CCM is symmetric w.r.t. the principal diagonal, which is null since is meaningless to consider self-matches (Figure 4). The values in CCM are non-negative. For values equal to 0 we can assume that corresponding cameras do not match at all.



**Figure 3:** A detail of 3D-points frequency representation on The Social Picture web platform: the points with the highest feature-frequency are represented with bigger shape and red colors, while on the contrary points with the lowest feature-frequency are represented with smaller shape and blue colors.



**Figure 4:** Camera-Camera-Matches (CCM) matrix. This symmetric matrix is obtained from the CPC boolean matrix as sum of the logical AND between each camera (represented as a row in CPC) and the other ones.

The computation of the CPC and CCM matrices is known as scene summarization of the dataset [SSS07].

### 3. Scene Summarization

Scene summarization is stated as the issue of select a set of canonical images that represents the visual content of a scene. In [KSX14, SSS07] a graph-based approach for scene summarization is presented, where the graph is built using matching visual features between images. We follow a similar method, employing the Model defined in Section 2, obtained from the VSFM output enriched with meta-data from TSP.

Values in CCM matrix can be seen as the edge-weights of an undirected graph. Indeed, cameras can be considered as nodes in a fully connected undirected graph in which the edge-weights are equal to the number of matches between cameras. The spanning tree of this graph can be used to explore images in a meaningful way, in a sort of tour of the selected collection. We employed the known minimum-spanning-tree (MST) construction algorithm from Kruskal [Kru56] as it is particularly performing approach in terms of complexity:  $O(V \cdot E)$  in the worst case, where  $V$  are the nodes and  $E$  are the edges of the graph.

Values stored in the CCM matrix represent how many cameras (images) are similar to each other. For this reason, we considered the values in CCM matrix as cost. However, since the Kruskal algorithm sorts the edges in increasing order of cost, we modified the original values of CCM as follows. Firstly, the reciprocal of each value in CCM is computed, in order to convert maximum values in minimum, and viceversa. Then, as it is not possible to define the reciprocal for values equal to 0, we set their cost to 1 (the highest possible value because CCM has non-negative values). In this way, all the new values are normalized in the range (0; 1), accordingly to the non-negative costs requirement of the algorithm of Kruskal.

For each node (camera) in the MST a score is computed. We define this score as the sum of the edge-weights adjacent to each node. It can be computed for each camera as the row-wise sum of CCM (or equally column-wise because CCM is symmetric). Then, we choose the node with the highest score as the source node of the graph traversal, implemented with a Depth First Search (DFS). This choice is made because very similar images are in the same path of the MST. When a node has more than one adjacent node (e.g., the node is root of a subtree in the MST), then it can be considered as the centroid of images contained in its subtree [KSX14].

An example of DFS graph traversal on MST of the selected collection has been publicized online on [http://youtu.be/PJgGBbD\\_rrc](http://youtu.be/PJgGBbD_rrc). Two consecutive images of the traversal are shown in Fig. 5. This kind of image summarization video is hardly evaluable with an objective metric. Instead, it is usually evaluated with a subjective consensus [KSX14, SSS07]. We are still in the process of investigate a better evaluation method. More in details, we state that scores of nodes may be exploited to discard the most unrelated images during the traversal. However, to assess a valid threshold for score values a wider experimentations should be conducted, with an higher number of collections. We remand this investigation to future works.

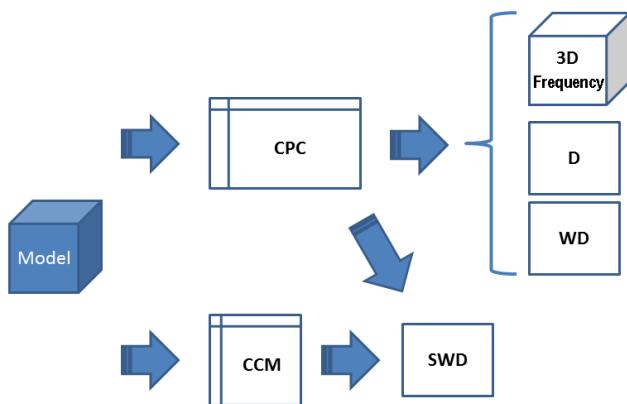
### 4. Feature Density Maps

Exploiting CPC matrix we can select all the points used as features by a given camera. Through the *Model*, which is parsed from NVM file, we also know the position of features in the image. Given the positions of visual features, we can define several kinds of maps (Figure 6):

- **Density Map (D-map):** characterizes the spatial density of the inlier visual features of an image; the inlier visual features are the one used for matching and 3D sparse reconstruction. In other words, D-map highlights locations of the image in which visual features are more dense.
- **Weighted-Density Map (WD-map):** characterizes the spatial density of the inlier visual features of an image, weighted w.r.t. their feature-frequency (as computed from CPC matrix). In other words, WD-map characterizes the density and importance of visual features within an image.
- **Social-Weighted-Density Map (SWD-map):** similar to the WD-map, but it also take into account the inlier visual features from images matching (directly or indirectly, through exhaustive matching - See Section 4.3.1) with the reference one. In other words, SWD-map characterizes the density and “social”



**Figure 5:** Two consecutive frames of Depth-First-Search graph traversal of the Minimum-Spanning-Tree of Pisa Collection.



**Figure 6:** Creation of Density Maps from the Model.

importance of visual features within an image, since the feature-frequency values are obtained taking into account also the images acquired by several points of view, which might have *WD-maps* potentially different.

- **Cumulative Map (*C-map*):** similar to the *SWD-map*, but the *C-map* does not characterize density of robust visual features. The *C-map* highlights parts of images frequently viewed in similar images in the dataset, computing the perspective transformations between them. So, the *C-maps* gives a smoother idea of what is socially salient from the point of view of the crowd. The intensity heatmap described in [BFM\*16] can be considered a *C-map*.

Some example of density maps is shown in Table 1. A deeper description of each map is given in the following subsections.

#### 4.1. Density Map (*D-map*)

The visual feature density is obtained quantizing the 2D space of the image and counting how many features are contained in each quantized interval. Density values obtained are normalized w.r.t. the maximum obtained. This map is named *Density Map (D-map)*. Note that the visual features contained in the NVM file (and so in the *Model*) are all “inlier”, this means that they are the one used for matching and 3D sparse reconstruction. They are a subset of

the whole possible visual features, already filtered by VSFM. The majority of the inlier visual features can be usually found near *corner points* particularly robust to scale and rotation variations.

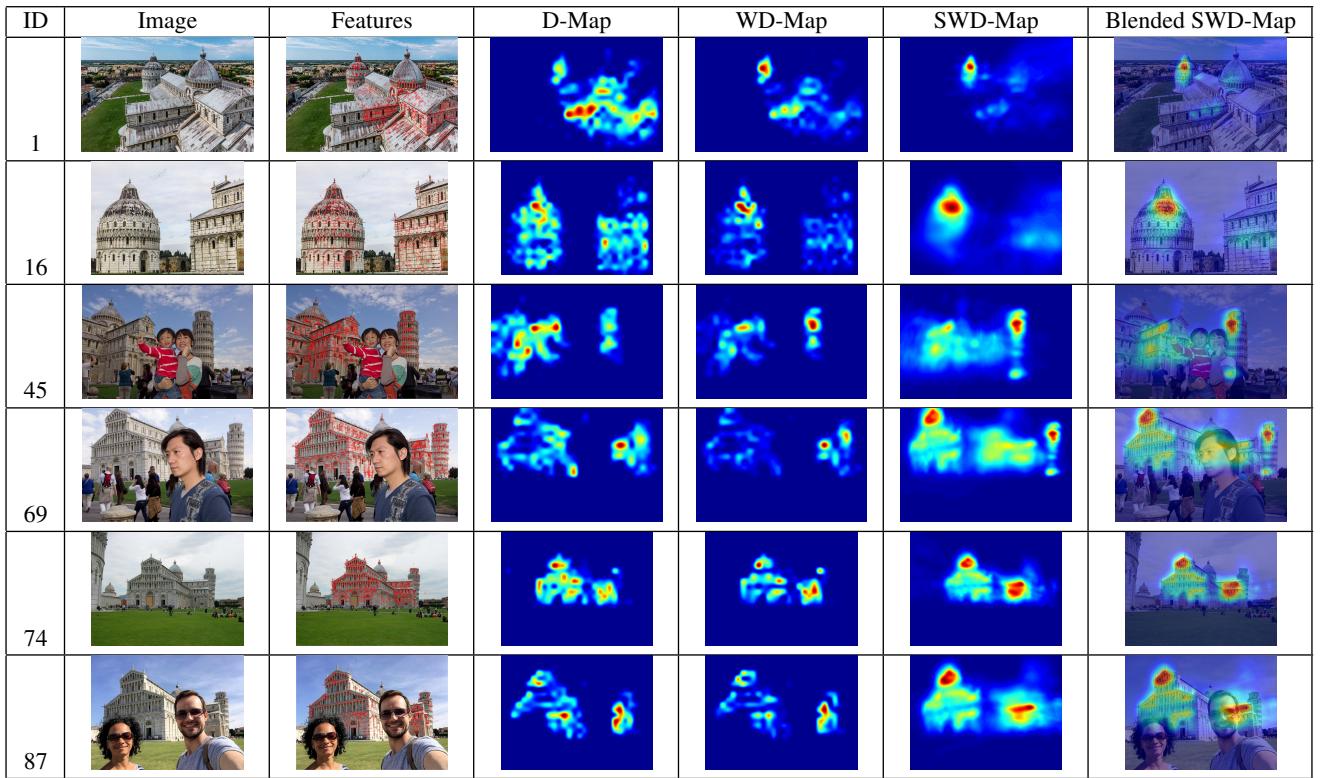
#### 4.2. Weighted-Density Map (*WD-map*)

The *D-maps* can be further refined taking into account a weight for each visual feature. The weight is equal to the feature-frequency, as computed through the column-wise sum from *CPC* matrix. In this way, each feature contributes differently to the computation of the density map during the quantization of the 2D space of the image. Density values obtained are normalized w.r.t. the maximum obtained in the whole map. The so obtained map is named *Weighted-Density Map (WD-map)*. The *D-maps* characterize where the majority of visual features can be found, while the *WD-maps* characterize where the most important and robust visual features can be found. Indeed, it is possible that a low number of very salient features generates a value in a *WD-map* higher than a bunch of low salient features. Given an image, the position of visual features in the *WD-map* is the same than the one in the *W-map*, since we just changed the weight of the features. What we want to highlight in the *WD-maps* is that in this kind of maps we are stressing the importance of the features from the point of view of the whole dataset, not just of a single image.

#### 4.3. Social-Weighted-Density Map (*SWD-map*)

It is easy to find what images match with an input one exploiting *CCM* matrix. We select the row in *CCM* related to the input image. Then, we look for images with a number of matching features higher than 0. We can either set a threshold  $t_m$  on the number of matching features in order to filter the image matching, obtaining the most reliable ones that reduce the risk of erroneous transform estimations. In the experiment described in this paper, we set  $t_m = 30$ . For all the images that have more than  $t_m$  features in common, we estimate the perspective transformation between them. A *T* transform matrix is obtained for each transform estimation. In *SWD-map* computation, we do not directly transform the images, but their *WD-maps*, without the normalization w.r.t. the maximum described in Section 4.2. All the non-normalized *WD-maps*, transformed with their own *T*, are summed to the non-normalized *WD-map* of the reference image. This final result is normalized w.r.t.

**Table 1:** Examples of Density Maps from Pisa Collection. From left to right, ID in the collection (ID), original image (Image), visual features marked on the original image (Features), Density-Map (D-Map), Weighted-Density Map (WD-Map), Social-Weighted-Density Map (SWD-Map) and a blended image between the original one and its SWD-Map are shown.



the maximum obtained by the sum of all the weighted feature densities. This map is named *Social-Weighted-Density Map (SWD-map)*, since it is generated by a “social knowledge” that is able to decide what features of an image are meaningful.

The rationale behind the computation of *SWD-maps* is that images similar to a reference one might contain information that is not present in the reference image. For instance, similar images might contain objects that in the reference image is hidden due to occlusion or a bad point of view. Through *SWD-maps* some parts of the reference image might be judged as meaningful by a pool of similar images and each one of them independently define what is meaningful for itself. A pro of *SWD-maps* is the capability to highlight salient-features region even if there is some occlusion in the image (Images 45, 69 and 87 in Table 1), while a con is the issue related to erroneous transformations, that might generate wrong density estimation in the final *SWD-map*.

#### 4.3.1. Exhaustive Image Matching

The *SWD-map* computation performs image matching between the images in a collection from TSP matching with an input image, accordingly to *CCM* matrix. This kind of image matching can be seen as the first level of an exhaustive image matching. In order to become “exhaustive”, we want to match also the images on the second, third, ...,  $n^{\text{th}}$ -level of matching, where in each level  $L$  we have

all the images matching with images at level  $L - 1$ . The exhaustive matching is completed when no more images can be matched (when an empty level is reached). Is important that each image is matched at most once to avoid loop in the process. Exhaustive image matching is referred in the literature as Query Expansion or Re-Query [CPS\*07]. However, in this paper we adopted a different approach to implement the exhaustive matching.

To realize exhaustive image matching we define a matching queue  $Q$  in which reference images of level  $L$  are enqueued. Hence, for instance at level 1,  $Q$  contains just the first reference image. We build a matrix named Matching Values ( $MV$ ), selecting the rows of *CCM* corresponding to the reference images in  $Q$ . Once  $MV$  is created images in  $Q$  are dequeued one by one. Then, we remove from  $MV$  the columns corresponding to images already considered as reference one (from level 1 to  $L$ ) and the columns with maximum equals to 0, since they cannot be matched with any reference image. For each remaining column  $C_c$ , we find the row  $C_r$  in which there is the maximum of the column and we compute the transformation matrix  $T(c, r)_L$  characterizing the perspective transformation between cameras  $C_c$  and  $C_r$  at level  $L$ . Transformations  $T$  can be concatenated from level  $L$  to 1, transforming all the images within the exhaustive matching process w.r.t. the initial reference image. Finally, cameras corresponding to remaining columns, if any, are enqueued in  $Q$  for level  $L + 1$ . If  $Q_{L+1}$  is empty, then the exhaustive matching will be over.

#### 4.4. Cumulative Map (*C-map*)

The *Cumulative Maps* (*C-maps*) are similar to the *SWD-map*: given an image, we use the visual features computed by VSFM to estimate a perspective transform  $T$  between the reference image and all its matching images. However, differently from the *SWD-maps*, in the *C-maps* we transform the matching images using a mask, properly a matrix with only values 1. All the masks, transformed with their own  $T$ , are summed to the mask of the reference image. This final result is normalized w.r.t. the maximum summed value obtained. Differently from the *SWD-maps*, the *C-maps* do not characterize density of robust visual features, but they highlight parts of the reference images frequently viewed in similar images in the dataset. The intensity heatmap described in [BFM\*16] can be considered a C-map. Visual features are often found near *corner points*, but salient objects are not necessarily made of corner points. Moreover, users are not interested in visual features when they acquire a photo. So, the *C-maps* gives a smoother idea of what is socially salient from the point of view of the crowd.

### 5. Conclusion

In this paper we presented an extension of our framework The Social Picture (TSP). VisualSfM (VSFM) [Wu13] has been employed to compute visual matching between images within a cultural heritage collection of TSP and to obtain a 3D sparse reconstruction of landmarks. Using VSFM and its 3D reconstruction, we defined new features added to TSP, such as the 3D-points frequency, and presented two advanced Image Analysis applications: scene summarization and density-maps.

Through the scene summarization we were able to create a video with a set of canonical images representing the visual content of a selected collection. This kind of summarizing-video is hardly evaluable with an objective metric. Instead, it is usually evaluated with a subjective consensus [KSX14, SSS07]. We defined a score for each image in the scene and stated how it may be exploited to discard the most unrelated images during the traversal. We remanded the assessment of a valid threshold for score values to a wider future experimentation.

We shown how density-maps can be used together with the SfM technique to highlight parts of the image with robust visual features. Several types of density-maps have been defined with different aims. Particularly, *SWD-maps* represent a good tool to stress the presence of visual features even when a strong occlusion is present in the image.

### References

- [BC16] BANO S., CAVALLARO A.: ViComp: composition of user-generated videos. *Multimedia Tools and Applications* 75 (2016), 7187–7210. 1
- [BFM\*16] BATTIATO S., FARINELLA G. M., MILOTTA F. L., ORTIS A., ADDESO L., CASELLA A., D’AMICO V., TORRISI G.: The social picture. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval* (2016), ACM, pp. 397–400. 1, 2, 4, 6
- [BFM\*17] BATTIATO S., FARINELLA G., MILOTTA F., ORTIS A., STANCO F., D’AMICO V., ADDESO L., TORRISI G.: Organizing videos streams for clustering and estimation of popular scene. In *19th International Conference on Image Analysis and Processing (ICIAP 2017)* (September 2017). 1
- [CPS\*07] CHUM O., PHILBIN J., SIVIC J., ISARD M., ZISSERMAN A.: Total recall: Automatic query expansion with a generative feature model for object retrieval. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on* (2007), IEEE, pp. 1–8. 5
- [KFF15] KARPATHY A., FEI-FEI L.: Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 3128–3137. 2
- [Kru56] KRUSKAL J. B.: On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society* 7, 1 (1956), 48–50. 3
- [KSH12] KRIZHEVSKY A., SUTSKEVER I., HINTON G. E.: Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (2012), pp. 1097–1105. 1
- [KSX14] KIM G., SIGAL L., XING E. P.: Joint summarization of large-scale collections of web images and videos for storyline reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2014), pp. 4225–4232. 3, 6
- [Low01] LOWE D.: Object recognition from local scale-invariant features. *Proceedings of the International Conference on Computer Vision* (2001). 2
- [MH08] MAATEN L. V. D., HINTON G.: Visualizing data using t-sne. *Journal of Machine Learning Research* 9, Nov (2008), 2579–2605. 1
- [OT01] OLIVA A., TORRALBA A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision* (2001). 2
- [SGY012] SAINI M. K., GADDE R., YAN S., OOI W. T.: Movimash: online mobile video mashup. In *ACM International Conference on Multimedia* (2012), pp. 139–148. 1
- [SSS07] SIMON I., SNAVELY N., SEITZ S. M.: Scene summarization for online image collections. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on* (2007), IEEE, pp. 1–8. 3, 6
- [WL15] WEYAND T., LEIBE B.: Visual landmark recognition from internet photo collections: A large-scale evaluation. *Computer Vision and Image Understanding* 135 (2015), 1–15. 1
- [WoV] WEBSITE OF VISUALSFM: //ccwu.me/vsfm/. 2
- [Wu13] WU C.: Towards linear-time incremental structure from motion. In *3DTV-Conference, 2013 International Conference on* (2013), IEEE, pp. 127–134. 2, 6
- [ZLX\*14] ZHOU B., LAPEDRIZA A., XIAO J., TORRALBA A., OLIVA A.: Learning deep features for scene recognition using places database. In *Advances in neural information processing systems* (2014), pp. 487–495. 1
- [ZW15] ZHENG E., WU C.: Structure from motion using structure-less resection. In *Proceedings of the IEEE International Conference on Computer Vision* (2015), pp. 2075–2083. 2