Computers & Graphics 63 (2017) 28–36

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Technical Section

Social media based 3D visual popularity☆

Abdullah Bulbul[a,*], Rozenn Dahyot[b]

[a] Department of Computer Engineering, Ankara Yildirim Beyazit University, 06010 Ankara, Turkey
[b] School of Computer Science and Statistics, Trinity College Dublin, Dublin 2, Ireland

ARTICLE INFO

Article history:
Received 1 June 2016
Revised 18 January 2017
Accepted 20 January 2017
Available online 2 February 2017

Keywords:
Social media
Popularity
Visual attention
3D cities

ABSTRACT

This paper proposes to use a geotagged virtual world for the visualization of people's visual interest and their sentiment as captured from their social network activities. Using mobile devices, people widely share their experiences and the things they find interesting through social networks. We experimentally show that accumulating information over a period of time from multiple social network users allows to efficiently map and visualize popular landmarks as found in cities such as Rome in Italy and Dublin in Ireland. The proposed approach is also sensitive to temporal and spatial events that attract visual attention. We visualize the calculated popularity on 3D virtual cities using game engine technologies.

© 2017 Elsevier Ltd. All rights reserved.

1. Introduction

In the last decade, social networks have become the prominent means for people for sharing their experience. Thanks to their increasing success, the amount of user generated geolocated time-stamped multimedia data that is shared online has increased dramatically. People's interests, sentiments and visual attention are now well reflected into social media and this information is up-to-date, correlates with spatiotemporal events and reflects user's relationships to their immediate environment. In this paper, we define the concept of visual popularity extracted from social media which aims at highlighting this linkage between people with their surroundings. We propose to use a mixture of data harvested from the web for automatically creating up to date 3D environments reflecting popularity and sentiments associated with locations (cf. Fig. 1). In particular, by analyzing pictures posted on social media, we propose to find out *visually popular* structures (e.g., buildings, monuments) in the environment that attract the most visual attention, and that deserves pictures to be taken and shared. Secondly, we propose to automatically visualize these popular landmarks directly in a 3D environment by controlling illumination in game engines, and alternatively by altering colors of the meshes to enhance popularity and sentiment visualization (Sections 3, 4 and 5). We conclude by discussing applications (e.g., games and virtual visits) of using social media information along with game technologies in Section 6.

2. Related work

Social media are loosely defined as websites and applications that enable users to create and share content, and can be used for interacting with other users. Second Life[1] is an example of a social media platform associated with 3D visual rendering, providing real-time avatars for online interaction in the virtual environment created by users. In a similar fashion, this paper aims at automatically creating a virtual environment as 3D representation of real cities, in which information extracted from social media can be used to inform users navigating in the virtual space. Of particular interest is the usage of posted images and the inference of the locations they capture. We review first some related work on inferring camera location from photographs and on 3D city reconstruction.

Inferring locations of pictures. Given an image database of geolocated city street scenes, Zhang and Kosecka proposed to compute the GPS location of a novel query image using SIFT feature matching [1]. The camera location and orientation of the query image is estimated also by robust triangulation. Jacobs et al. [2] considered widely distributed camera networks and aimed at geolocating the cameras using synchronized natural events (e.g., weather) by comparing camera images with geo-registered satellite weather images.

☆ This article was recommended for publication by L.P. Santos.

* Corresponding author.
E-mail address: abulbul@ybu.edu.tr (A. Bulbul).
URL: http://ybu.edu.tr/abulbul/ (A. Bulbul)

[1] Second Life, online virtual world, http://secondlife.com , (15/01/2017).

http://dx.doi.org/10.1016/j.cag.2017.01.005
0097-8493/© 2017 Elsevier Ltd. All rights reserved.

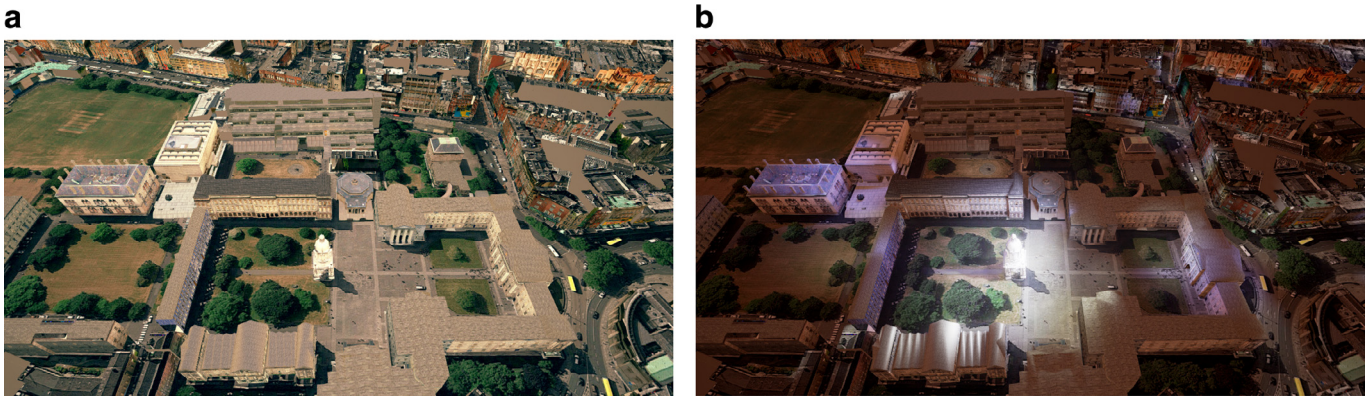

Fig. 1. Popularity based illumination using virtual lighting. A view over Trinity College Dublin (a) is illuminated (b) according to the popularity extracted from geolocated images posted on social media. The Trinity College Campanile is the most popular visual landmark building in that area.

Hays and Efros proposed to estimate the location of a query image by comparing its visual content with a large dataset of geolocated images covering the world [3]. Their system performed significantly better than random chance, with accuracies in the order of several kilometers. Zamir and Shah proposed to infer the GPS location of an image by matching it with a large dataset of geolocated images from Google street view [4,5]. The closest geolocated image in the database provides an estimate of the GPS location of the query image. More recently, Castaldo et al. [6] proposed to infer location of a semantically segmented image (input query) in a geographic information system.

Crandall et al. [7] proposed inferring locations of large collections of images shared in Flickr by utilizing SIFT feature matching and similarity of textual information accompanying the photos. In addition to organizing the images according to visual and textual similarity, their method is also used to reveal the most popular landmarks. Simon et al. [8] used SIFT features to find out a subset of images from many to generate a meaningful summary of a scene. Li et al. [9] proposed a method for pose estimation by geo-registering 3D feature points at a worldwide scale. Middelberg et al. [10] proposed a similar system for real-time use in mobile devices by an initial SLAM performed in the mobile device followed by global registration on the main server.

3D reconstruction of cities. Traditionally creating virtual 3D cities have been used in various fields such as computer games, urban planning, architectural visualization, crowd and traffic simulations. For instance, O'Sullivan et al. have created a Virtual Dublin, an interactive model of the city of Dublin (Ireland) developing their own Ogre[2] based engine for first person viewing and for real time navigation, in an environment populated with buildings designed by artists [11,12]. Such approach for 3D city modeling is labor intensive and gets very quickly out of date when aiming for a realistic virtual model. As an alternative, 3D reconstruction techniques from multiple view images have become popular. In a pioneering work [13–15], Snavely et al. proposed to use online image collections to infer the viewpoint of each photograph and a sparse 3D model (point cloud) of the scene. However, sparse point cloud representation with many vertices to represent a whole city is not suitable and compact enough for a game engine for providing realtime interaction to users.

Xiao et al. [16] proposed an automatic approach to generate street-side 3D photo-realistic models from images captured at ground level. Their overall approach is based on the availability of pre-computed semi-dense 3D point clouds and camera positions (based on structure from motion techniques from images with potentially additional GPS/INS information available). Priors are then used to process point clouds to recover building meshes with texture map. A reported computation time is 23 h (2 h for SfM, 19 h for segmentation, and 2 h for partition and modeling) for reconstructing 202 building blocks in Pittsburg area from 10,498 images, on a small cluster composed by 15 normal desktop PCs (in 2009). Likewise using SfM based approach, Torii et al. [17] reported more than 30 h for reconstruction on the same dataset. Anguelov et al. [18] presented the work done at Google for capturing images, GPS data, and laser range data at street level. The 3D reconstruction is performed from laser data to populate Google Earth.

Matzen and Snavely proposed to reconstruct a 3D spatio-temporal urban scene from a large online image repository where scene appearance changes overtime [19]. Such reconstruction is then used for automatically time stamping new images.

3. Overview

Social media data considered in this paper is harvested from Twitter[3] and Instagram[4]. We also use information harvested from the OpenStreetMap[5], such as building footprints with geolocated landmarks to help creating 3D cities automatically. OpenStreetMap has the advantage of providing open source data that is constantly updated by online contributing users that often have knowledge of their real local environment. Additional information is automatically extracted from Google Street view. Combining social media and 2D maps have found applications for rendering tweet activities and sentiment maps [20], and this paper aims at rendering extracted information from social media in 3D virtual environments that is suitable for user navigation and interactions. We harvest social media data using hashtags and/or location based queries, and sentiment scores for the text of the posts are computed and stored in a database[20]. This information is then visualized in a 3D environment. Fig. 2 presents an overview of our system.

Public data shared within Instagram and Twitter are gathered through their public API's. These data include visual imagery and accompanying information such as keywords, time stamps, and geo-coordinates. While the photos shared through social media reflect the popularity of the visible scenery, the majority of these images, for instance photos of personal items or food, is not useful for determining visual popularity of the surroundings. Another important feature of photos shared through social media is that they are

[2] OGRE3D, 3D rendering engine, http://www.ogre3d.org , (17/01/2017).

[3] Twitter, social networking service, https://twitter.com , (17/01/2017).

[4] Instagram, photo-sharing site, https://www.instagram.com , (17/01/2017).

[5] OpenStreetMap, free wiki World map, https://www.openstreetmap.org , (17/01/2017).

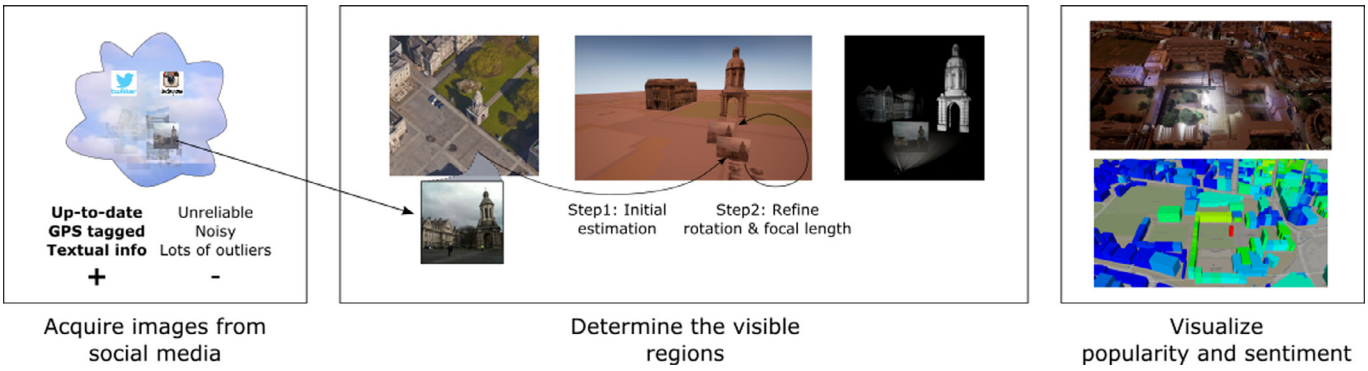

Fig. 2. Overview of our system for visualizing popularity and sentiment extracted from social media in a 3D environment.

possibly modified, noisy, or include occluding objects in large portions such as faces in selfies, that make them unreliable for matching with other images taken in the same location using Computer Vision algorithms. Structure from motion (SfM) algorithms for 3D reconstruction fail to produce plausible 3D models when the input data contains too many outlier images and partially occluded images. Social media imagery is therefore not reliable as a source of information for these algorithms. Instead, we make use of more reliable and publicly available localized image databases captured from street level such as Google Street View. Google has a public API allowing users to reach street level imagery with a maximum of 640× 640 resolution. These are distributed almost homogeneously through the streets, and each of them is accompanied with a geo-coordinate with an error of up to a few meters. Google's satellite images and OpenStreetMap's data providing the building structures are both used for creating cities in 3D.

Although the images from social media are not reliable for 3D reconstruction, they can still be used to get an idea of the visible area with the presence of reference images. Here, our purpose is determining the most visible regions among images shared through social media. While this problem resembles the camera localization problem, in our case the important aspect is determining the visible area rather than accurately positioning a camera in 3D space. More specifically, our contributions in this paper are:

- We consider that a GPS location is already associated to the social media post and image. We propose harvesting Google street view images on the fly only in the same area to refine the GPS location as well as recovering the camera parameters of the image in the post (Section 4).
- We propose to reconstruct a 3D model of cities using information from Open Street maps and Google street view. While Google maps now provides 3D models for buildings, these are not available through their API. We propose here a cheap and efficient alternative for creating a 3D virtual model of a city that can be uploaded to a Game engine for providing first person navigation experience to users. Using Unreal 4[6] , we visualize the reconstructed 3D city providing one player navigation experience for multiple platforms.
- Finally, we propose to use this virtual 3D environment for visualizing information extracted from social media data. In particular, we show how controlling lights in the virtual environment allows to highlight popular photographic landmarks. As such this defines a 3D popularity map similar to a 3D saliency map because it captures what caught people's eyes.

[6] https://www.unrealengine.com.

4. Visual popularity computation

We note $\mathcal{S}$ the set of images harvested from social media with their location information. We note $\mathcal{R}$, the set of reference images queried from Google Street View with their camera parameters and geo-coordinates. Location information in $\mathcal{R}$ is then accurate within a few meters, which is sufficient for our purpose. A pinhole camera model (with translation, rotation, and focal length) is associated with each image in the reference set [15,21]. For the images in $\mathcal{S}$, in addition to the camera parameters to estimate (cf. Sections 4.1 and 4.2), we also define a confidence value for computing visual popularity (cf. Section 4.3).

The approach proposed in this paper has been evaluated with 3 datasets associated with different cities: Dublin (Ireland), Rome (Italy) and Pittsburgh (USA) (top row of Fig. 8). Table 1 presents the duration (in days), areas (km^2) and corresponding size of the set $\mathcal{S}$ in these scenarios.

4.1. Initialization of camera parameters in $\mathcal{S}$

Camera parameters associated with an image in $\mathcal{S}$ are initialized with the parameters of the most similar image in $\mathcal{R}$. We address here how reference set $\mathcal{R}$ is defined for computational efficiency, and how it is used to infer the camera parameters of images in $\mathcal{S}$.

Determining the search domain $\mathcal{R}$. Comparing each image in $\mathcal{S}$ with a set $\mathcal{R}$ corresponding to all possible Google street view images is not computationally efficient. Instead the geo-coordinates associated with each image in $\mathcal{S}$ determines the center of the specific search domain $\mathcal{R}$. Note that, because the geo-coordinates in $\mathcal{S}$ may or may not correspond to the location of the device when the photo is taken but instead the location of the device when the photo is actually shared on social media, location information in $\mathcal{S}$ is less accurate than in $\mathcal{R}$ (cf. Fig. 3). Consequently a trade-off between the computation time (which changes proportional to the square radius of search domain) and the chance of the best reference image to be in the search domain $\mathcal{R}$, needs to be found (Fig. 4(b)). Fig. 4(a) shows a scatter plot of the correctly assigned reference image locations relative to the corresponding social media images, i.e., for each query image from $\mathcal{S}$, there is a dot at the relative 2D location of the corresponding street level image. Increasing search radius from 100 to 200 m requires 4 times more comparisons while increasing the chance of including the best match by only 23%.

Initial guess of camera parameters for images in $\mathcal{S}$. To determine the most similar image in $\mathcal{R}$ to the query image in $\mathcal{S}$, we employ SIFT feature matching [22]. The straightforward approach would be selecting the image with the highest number of feature matches in

Table 1
Visible region estimation for images in $\mathcal{S}$.

City	Area (km^2)	Activity duration	$\lvert\mathcal{S}\rvert$	$\lvert\mathcal{S}_{inlier}\rvert$	Proc. time (min)
Rome	0.698	20 days	1284	234 (18%)	113
Pittsburgh	4.826	30 days	5669	158 (3%)	457
Dublin	0.865	21 days	2568	171 (7%)	328

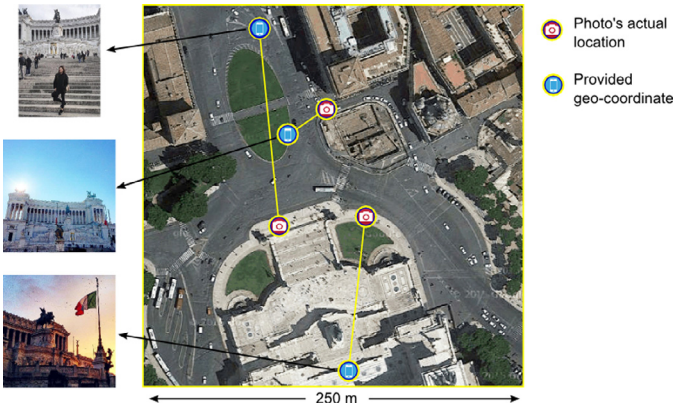

Fig. 3. Examples of photos posted on social media with their accompanying geo-coordinates and their actual camera locations (Rome).

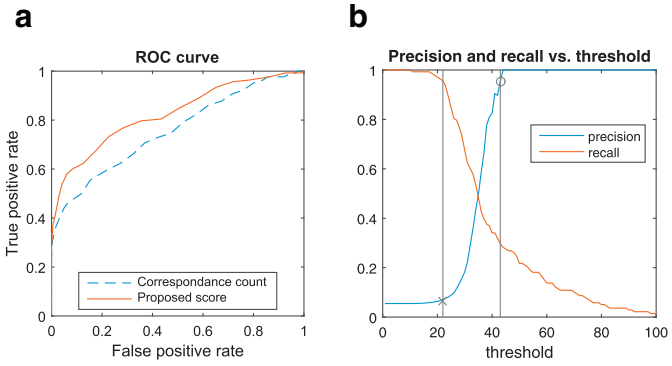

Fig. 5. (a) Classification performance of the proposed score vs. number of correspondences. The closer curve to top left corner is the better. (b) Precision and recall according to score threshold (*lt* threshold for × and *ht* threshold for ○).

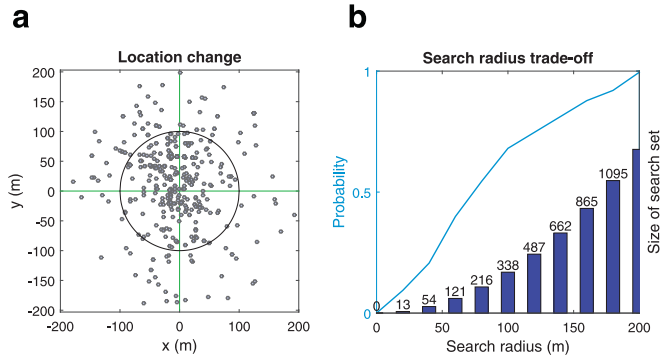

Fig. 4. (a) Locations of the best reference images relative to the query images from $\mathcal{S}$ are shown. The circle depicts 100 m of search radius. (b) bars show the number of reference images in the search domain according to its radius. The line plot shows the probability of having included the best reference image in the search set. (Data is accumulated from the three cities.)

our search domain. However, we found that to increase the accuracy without sacrificing performance, we can use a simple heuristic to reward coherence of horizontal positions of successive features matches. This heuristic is based on the assumption that the images are taken with a roughly upright orientation and the features over corresponding objects should have the same horizontal ordering among different pictures.

Assume that $\mathcal{M} = \{m^{(i)} = (s_x^{(i)}, s_y^{(i)}, r_x^{(i)}, r_y^{(i)})\}_{i=1,\dots,n}$ is a set of n correspondences between image $s \in \mathcal{S}$ and image $r \in \mathcal{R}$. First, we sort the correspondences in ascending order according to the horizontal coordinate s_x in image s of the correspondences. Then we classify correspondences using the horizontal coordinate r_x in image r according to following criterion:

$$\begin{cases} m^{(i)} \in \mathcal{M}_{favorable} & \text{if } r_x^{(i)} > r_x^{(i-1)} \\ m^{(i)} \in \mathcal{M}_{unfavorable} & \text{otherwise} \end{cases} \quad (1)$$

The score for image pair (s, r) is calculated as:

$$\text{score}(s, r) = 2\ |\mathcal{M}_{favorable}| - |\mathcal{M}_{unfavorable}| \quad (2)$$

We have evaluated the accuracy of our heuristic over 2500 photos from social media and the corresponding reference images used to initialize the cameras. Fig. 5(a) shows the advantage of using the proposed score over using the number of correspondences with a receiver operating characteristic (ROC) curve. Here, a true-positive means that there is an overlap between the image from social media and the selected reference image. The same set of photos is also used to determine the acceptance and rejection criteria of the images from social media. Fig. 5(b) shows the precision and recall values according to threshold values. Selecting a high threshold, e.g., the point denoted with ○ in the plot provides a high precision; however, only one third of the correct matches would be used. Therefore, we use a low value, shown as × in the plot as the rejection threshold and set the confidence value of the cameras as:

$$\text{confidence}(s) = \begin{cases} 1 & \text{if score}(s, r_{best}) > ht \\ 0 & \text{if score}(s, r_{best}) < lt \\ \dfrac{\text{score}(s, r_{best}) - lt}{ht - lt} & \text{otherwise} \end{cases}, \quad (3)$$

where *lt* and *ht* stands for the low and high thresholds shown in Fig. 5(b). The geo-coordinates and camera parameters associated with the best reference image are transferred to the social media image s as well as the confidence value associated with this match.

4.2. Refinement of camera parameters in $\mathcal{S}$

The camera parameters of s can be further refined to understand the physical region of interest captured in the image s, and for a better alignment with the visible area in the corresponding image r. In a standard motion estimation problem, where we have an input camera s and a set of reference cameras from R, the purpose of refinement is finding rotation, translation, and focal length values to minimize the total re-projection error of matching between s and R. This is a simpler form of structure from motion problem as all cameras other than s are static. This form of optimization works for precisely calibrated reference cameras, e.g., if reference cameras consists of SfM outputs. However, when Google Street-view images are used, the optimization usually do not converge to a plausible solution. In that case, one option is to re-

calibrate the cameras in R before optimizing s, which is a time consuming process. Besides, even in that case, low quality input images has a high probability of being erroneously translated by a large amount, which introduces the risk of having occluding models in between the region of interest and the camera, i.e.; the camera may incorrectly be placed behind a wall, especially around narrow streets, making it unable to show the desired location.

Therefore, we keep the location of the camera fixed while performing the optimization only on the rotation and focal length of the camera in image space by employing Homography as follows:

$$(\widehat{\Delta f, \Delta \mathrm{R}}) = \arg \min_{(\Delta f, \Delta \mathrm{R})} \sum_{m \in \mathcal{M}} {E_m}^2, \tag{4}$$

$$\text{with } E_m = p_s - \Delta f \Delta \mathrm{R} p_r, \tag{5}$$

where E_m is the image based re-projection error for the pair m, p_s and p_r are 2D positions of the matching features from input image s and corresponding reference r. Δf and $\Delta \mathrm{R}$ are the additional rotation and focal length that would redirect the camera to the desired location while keeping its location fixed. This approach performs well if the features correspondences belong to a roughly planar surface, or there is not much depth difference throughout the scene. We have experienced with both full camera refinement in 3D and our camera refinement with fixed location. Although our camera refinement step is not sensitive to depth differences within the visible area, it resulted in more reliable outputs. However, in presence of precisely calibrated reference cameras and high quality input images the first method would be a more accurate option. Fig. 6 shows sample results of the found visible regions with our technique to refine camera parameters. We have experimented with input images gathered from social media around several cities. The publicly available images are pooled from Instagram and Twitter for a duration between 20 and 30 days. Table 1 shows the number of photos processed for each city along with the processing time when the radius is 100 meters. In this table $|S_{inlier}|$ stands for the most confident cameras. The outliers among the input images can be very high ($>$ 80%) depending on the city, e.g., more images shared from Rome belong to visual points of interest compared to the other two cities used in this paper.

4.3. Visual popularity for 3D models

Having a set of calibrated cameras in $\mathcal{S}$, the popularity map over 3D models is computed by taking the confidences of the cameras into account as follow:

$$P(e) = \frac{\sum_{s \in V} \text{confidence}(s)}{\text{size}(e)}, \tag{6}$$

where e is an element of the set on which we calculate popularity such as vertices or faces on 3D mesh models and V is the subset of $\mathcal{S}$ in which e is visible. In case the elements have varying areas, a division by their size, i.e., area of a face, gets rid of the size effect which would otherwise favor larger elements. For instance, when the elements are vertices, the size of the elements is constant.

Fig. 7 shows the images that contribute to the most popular regions of the experimented cities and Fig. 8 shows examples of calculated popularity values along with the corresponding regions from Google 3D Earth View. As seen in these figures, the most popular areas correspond to the well-known touristic marks. For instance, from the experimented region of Rome, Pantheon (Fig. 7(a)) and Trevi Fountain (Fig. 7(c)) are found as the most popular architectures. Similarly, the Trinity College Campanile (Fig. 7(b)) is the most popular point of interest within Dublin's experimented area. In Pittsburgh, however, there is a more homogeneous distribution of popularity, and as seen from Fig. 7(d) that is because the shared

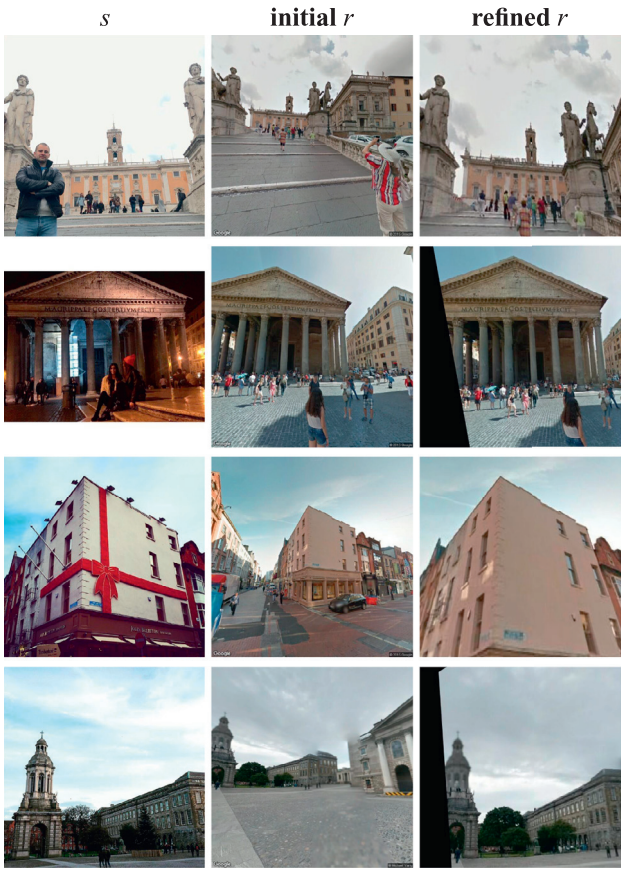

Fig. 6. Examples: input images s (left), corresponding most similar reference images r (middle) that are used to initialize camera parameters, and right, transformed reference images to show the visible area with our refined camera parameters.

Fig. 7. Images from social media in $\mathcal{S}$ that contribute to the popularity of specific locations in the experimented cities: (a) Pantheon (Rome), (b) Trinity College Campanile (Dublin), (c) Trevi Fountain (Rome) and (d) Pittsburgh.

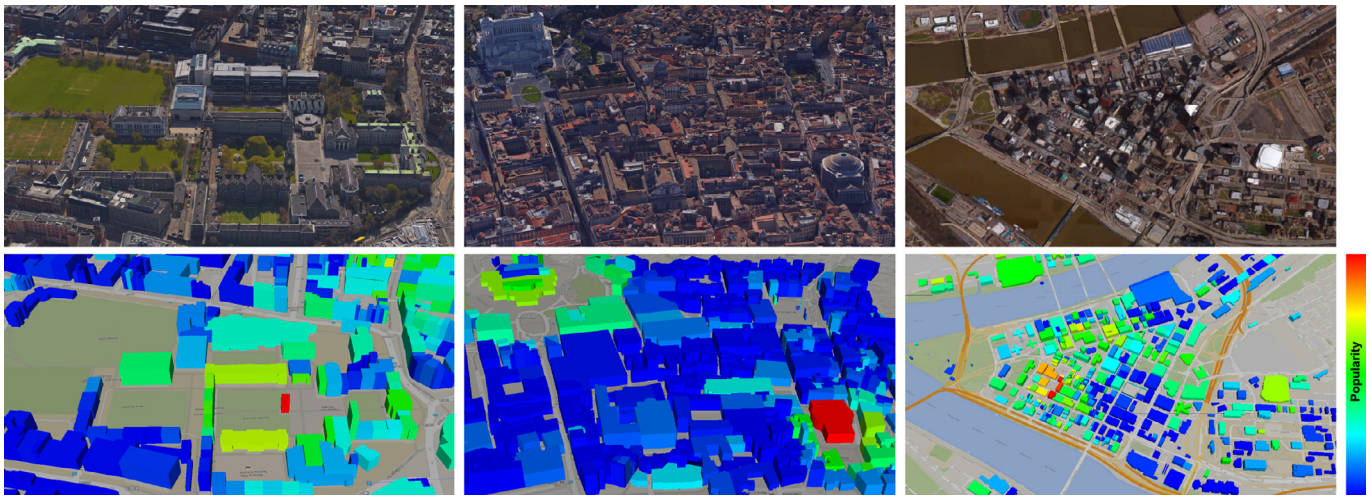

Fig. 8. Top: 3D views from Google 3D; Bottom: Our calculated per-building popularity for the corresponding areas. Popularity values are normalized within each area (the color red indicating the most popularity values, to the blue color to the lowest ones). Cities from left to right: Dublin, Rome, and Pittsburgh. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

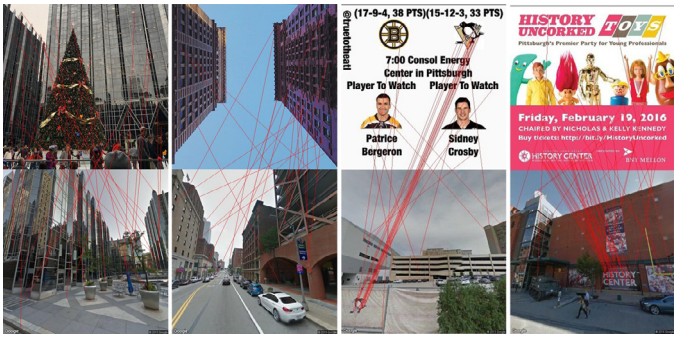

Fig. 9. Failure examples, Top: input images, Bottom: matches from $\mathcal{R}$. These images exemplifies cases caused by repeating structures (leftmost image), upright orientation assumption (second image from the left), logos and texts (the two images on the right).

Fig. 10. Left: Example of buildings generated using our approach. Right: street level image from Google street view (top), with our corresponding view in our 3D environment (bottom).

photos are the silhouettes of the city, rather than a specific location.

Limitations. Although the popular regions are successfully determined in most cases, our method suffers from the inherent limitations of SIFT feature matching. For example, the lack of features or having reflective surfaces (e.g., glass windows) in large proportions decrease detectability of a region. Also, having multiple buildings with the same appearance confuses the method as well as company logos which can be found at various locations. Another factor is the up to dateness of the reference images, e.g., new buildings and façade alterations missing in $\mathcal{R}$ may cause a lack of correspondences between the query in $\mathcal{S}$ and the reference sets. Fig. 9 shows several failure cases.

5. Visual popularity rendering

3D visualization requires a 3D model of the area of interest. 3D models of cities created by artists may exist such as the city center of Dublin built within the Metropolis project [12], but these models are not common and are quickly outdated due to the fast changing urban landscape. We propose here a simple and computationally efficient alternative for creating 3D models of cities with buildings as elementary elements. Our system is using information available online and our reconstruction can easily be kept up-to-date when this information shared online is also kept up-to-date. Our models are then used within Unreal engine to render visual popularity of the urban landscape.

5.1. Fast city generation

Our city generation method is dependent on the availability of localized street-level images (Google Street View images are used) and the contour lines showing the areas occupied by the buildings (OpenStreetMap data harvested on the fly is used for building contours). Public access to this information is possible for many areas of the world and it is getting even more common and widespread.

Building generation is based on back-projecting the street level images onto building blocks. Buildings are generated by utilizing their contours in OpenStreetMap, where each building is represented by a 2D polygon and each corner of a polygon is associated with a geo-coordinate. These geo-coordinates are first transformed to our model-space coordinates so that we have 2D footprints of each building where one unit length corresponds to 1 m. Then each edge of the footprint is converted to a simple rectangular wall by assigning a default height value. After having these initial 3D models of buildings, each side of them is textured by projecting corresponding street level images onto them. Finally, a color thresholding based sky detection method similar to the one used in [23] is employed to determine the final heights of the buildings. Fig. 10 shows a few examples generated with our approach for which artifacts occasionally occur. Our approach is limited for representing complex geometries of buildings and its success also depends on the coherence between geolocated street level imagery

Table 2
City generation.

City	Area (km^2)	Num. of buildings	Num. of ref locs	Generation time (m:s)
Rome	0.698	543	1323	5:33
Pittsburgh	4.826	517	5930	14:03
Dublin	0.865	810	1342	6:12

and the geolocated footprints of buildings which here have been collected from two providers of data, Google and OpenStreetMap. Moreover, Google street view images flatten occluding objects such as trees that are then mapped on our buildings. Image processing algorithms can be used to improve 3D building reconstruction, including the modeling and texture generation steps by employing methods based on vanishing lines or by exploiting multiple views [24,25]. We avoid any complex algorithms in our city generation step due to the large number of buildings to be generated efficiently as well as the diversity of our scenarios that include any type of buildings (e.g., historical) from various time periods. Table 2 shows our city generation times for several cities, using not yet optimized research code. Our technique performs very well considering computation times reported in the review section (Section 2), and we are able to reconstruct large urban areas in a few minutes on a standard PC.

5.2. Visualization with social lights

Each estimated camera view associated with social media (set $\mathcal{S}$) contributes to the popularity of the visible area. Therefore, if we represent each of these cameras with a spot light directed to the same area, the 3D environment would be lit according to their popularity. However, using camera locations as is do not result in a natural illumination because the cameras are located very close to ground level and directed horizontally or even tilted upwards in many cases. Better natural lighting can be achieved when the lights are positioned higher looking down from slightly left on the scene captured in the social media image [26]. Therefore, we elevate the spot lights and tilt them downwards to lit the area of interest. To accomplish that, we first determine the average depth of the visible buildings from a camera's viewpoint; then the corresponding light's height is calculated so that when the light is directed towards the calculated distance it has a 45 degrees of downwards tilt value. We avoid translating the lights to the left because of the potential occlusion problems in the scene. The intensities of the lights are adjusted according to the confidence values associated with the cameras in $\mathcal{S}$. Fig. 11 shows the three cities illuminated with their social media oriented lights.

5.3. Sentiment based coloring

As each light originate from a social media post, it is possible to utilize the shared textual content too. We assign a sentiment score to each post by querying the deep learning based sentiment analysis tool within CoreNLP library [20,27]. Then, the sentiment score is reflected to the color of the light by interpolation between yellow (positive sentiment), white (neutral sentiment) and blue (negative sentiment). Fig. 12 illustrates our sentiment rendering approach. Currently, the sentiment classifier is used in its original state, which could be improved by training specifically on social media data.

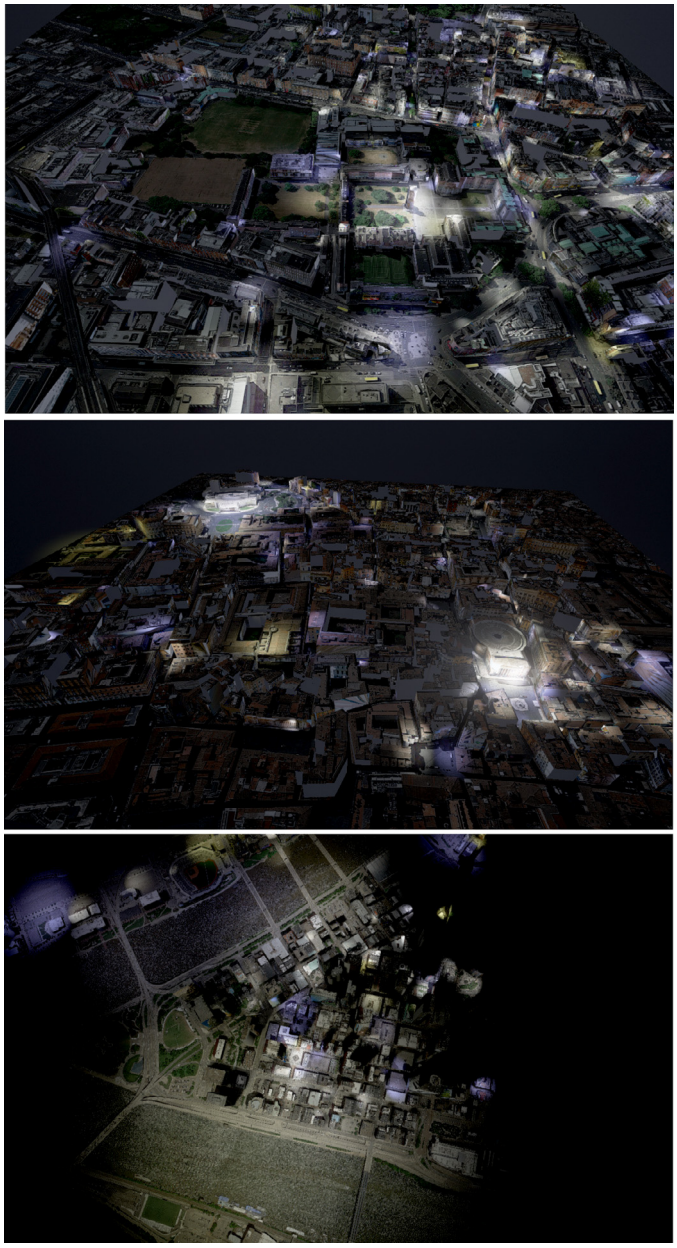

Fig. 11. Illumination with social lights. From top to bottom: Dublin, Rome and Pittsburgh.

6. Discussion and conclusion

We have presented a platform for automatically creating up-to-date 3D rendering of cities from data available online. Thanks to game engine technologies such as Unreal, immersive first person experience is provided to navigate in real-time through the reconstructed world. Our approach is therefore very suitable for quickly providing virtual visit experiences to online users for instance. In addition, we have shown how such environment can be used to overlay information inferred from online social networks [35], and in particular providing visual popularity rendering for visualizing what people photograph and what they feel about it. The proposed system is sensitive to current trends in visual popularity; in addition to well-known touristic spots, it is possible to discover visual

Fig. 12. Sentiment based coloring. Top: photos from social media assigned with different sentiment scores. Bottom: corresponding lights in isolation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 13. Sample images from social media indicating a visual point of interest. We've noticed this mural painting as it is significantly illuminated by social lights.

points of interest those reflect a temporal event or a salient modification within the environment that attracts people's attention. A mural painting (Fig. 13) that we have noticed during this study by following unexpected social lights in Dublin can be shown as an example of that.

Synthesis of 3D virtual world. When no 3D model (e.g., Metropolis for Dublin) is available for an area, our current approach for creating a 3D virtual world is based on efficiently combining building footprints with images available from online repositories for creating more up-to-date neighborhoods. More advanced image processing algorithms in combination with GIS database could be used not only for extracting buildings in cities but also to identify and efficiently render any individual fixed element composing the landscape anywhere (e.g., roads, trees, fields, lakes, mountains). Our approach for visualizing 3D visual popularity could ultimately be validated in the countryside as long as geolocated pictures posted online share content with online repositories.

Generation of tourist maps and guides. The work presented in this paper could be further extended to efficiently highlight landmarks on 2D or 3D maps for users such as tourists visiting an area [28]. While information about the nature and functionality of buildings can be harvested from existing digital map services online (e.g., Microsoft Live (www.live.com), Google Maps (maps.google.com) and Yahoo Maps (maps.yahoo.com), OpenStreetMap), the analysis of social media data relevant to a particular place and time can be helpful for highlighting events and exhibitions and inform on people sentiment. Our platform could be extended further for efficiently visualizing any GIS databases made accessible online.

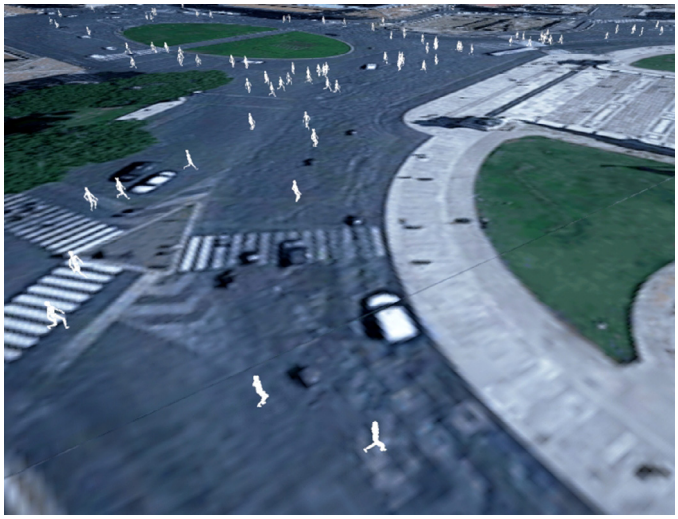

Fig. 14. Automatically placing agents according to social media activity.

Saliency. Knowledge of important regions in a scene always plays a major role in many computer graphics algorithms and applications such as optimization of the rendering process [29], model simplification [30], and guiding designers and artists when building and refining 3D scenes. Consequently, there have been a lot of efforts for automatically finding the visually important or salient regions in 3D environments [31,32]. These models mostly utilize the distribution of geometric and appearance related features to determine the distinct and salient features which is limited to revealing only the bottom up direction of visual attention without having user specific factors such as prior experiences and viewer intentions [33] which constitutes the top-down direction of visual attention and not less important than the scene specific bottom-up part. Social media based visual popularity can be helpful for including the top-down factors in determining the visually important regions of a city.

Towards a populated virtual environment. Other directions of our research investigate how social networks, online information, game engines and virtual reality technologies can be used together for creating new interactive, immersive and more human experience online. Indeed geolocated social network users can be used to place avatars [34], to animate them and to populate the virtual cities with realistic displacement patterns, for providing human interactions in the context of a virtual visit or a gaming experience (e.g., Fig. 14).

Acknowledgments

This work has been supported by EU FP7-PEOPLE-2013-IAPP GRAISearch grant (612334).

References

[1] Zhang W, Kosecka J. Image based localization in urban environments. In: Proceedings of the third international symposium on 3D data processing, visualization, and transmission (3DPVT'06). Washington, DC, USA: IEEE Computer Society; 2006. p. 33–40. doi:10.1109/3DPVT.2006.80. 0-7695-2825-2
[2] Jacobs N, Satkin S, Roman N, Speyer R, Pless R. Geolocating static cameras. In: Proceedings of the international conference on computer vision; 2007. p. 1–6.
[3] Hays J, Efros A, et al. Im2gps: estimating geographic information from a single image. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 2008.. IEEE; 2008. p. 1–8.
[4] Zamir AR, Shah M. Accurate image localization based on Google maps street view. In: Proceedings of the european conference on computer vision; 2010. p. 255–68.
[5] Zamir AR, Shah M. Image geo-localization based on multiple nearest neighbor feature matching using generalized graphs. IEEE Trans Pattern Anal. Mach. Intell. 2014;36(8).

[6] Castaldo F, Zamir A, Angst R, Palmieri F, Savarese S. Semantic cross-view matching. In: Proceedings of the international conference on computer vision; 2015. p. 9–17.
[7] Crandall DJ, Backstrom L, Huttenlocher D, Kleinberg J. Mapping the world's photos. In: Proceedings of the 18th international conference on World Wide Web. New York, NY, USA: ACM; 2009. p. 761–70. doi:10.1145/1526709.1526812. 978-1-60558-487-4
[8] Simon I, Snavely N, Seitz SM. Scene summarization for online image collections. In: Proceedings of the IEEE 11th international conference on computer vision; 2007. p. 1–8. doi:10.1109/ICCV.2007.4408863.
[9] Li Y, Snavely N, Huttenlocher D, Fua P. Worldwide pose estimation using 3d point clouds. In: Proceedings of the computer vision–ECCV. Springer; 2012. p. 15–29.
[10] Middelberg S, Sattler T, Untzelmann O, Kobbelt L. Scalable 6-dof localization on mobile devices. In: Proceedings of the computer vision–ECCV. Springer; 2014. p. 268–83.
[11] Hamill J, O'Sullivan C. Virtual Dublin: a framework for real-time urban simulation. J WSCG 2003;11(1):221–5.
[12] O'Sullivan C, Ennis C. Metropolis: multisensory simulation of a populated city. In: Proceedings of the Third International Conference on Games and Virtual Worlds for Serious Applications; 2011. 978-0-7695-4419-9/11
[13] Snavely N, Seitz SM, Szeliski R. Photo tourism: exploring photo collections in 3d. In: Proceedings of the conference SIGGRAPH. New York, NY, USA: ACM Press; 2006. p. 835–46. 1-59593-364-6
[14] Snavely N, Seitz S, Szeliski R. Modeling the world from internet photo collections. Int J Comput Vis 2008;80(2):189–210. doi:10.1007/s11263-007-0107-3.
[15] Agarwal S, Snavely N, Simon I, Seitz SM, Szeliski R. Building rome in a day. In: Proceedings of the international conference on computer vision. Kyoto, Japan; 2009. p. 72–9.
[16] Xiao J, Fang T, Zhao P, Lhuillier M, Quan L. Image-based street-side city modeling. ACM Trans Graph 2009;28(5) 114:1–114:12. doi:10.1145/1618452.1618460.
[17] Torii A, Havlena M, Pajdla T. From Google street view to 3d city models. In: Proceedings of the 12th IEEE International Conference on computer vision workshops; 2009. p. 2188–95. doi:10.1109/ICCVW.2009.5457551.
[18] Anguelov D, Dulong C, Filip D, Frueh C, Lafon S, Lyon R, et al. Google street view: Capturing the world at street level. Computer 2010;43(6):32–8. doi:10.1109/MC.2010.170.
[19] Matzen K, Snavely N. Scene chronology. In: Proceedings of the European conference on computer vision; 2014. p. 615–30.
[20] Dahyot R, Brady C, Bourges C, Bulbul A. Information visualisation for social media analytics. In: Proceedings of the international workshop on computational intelligence for multimedia understanding. Prague, Czech Republic; 2015.
[21] Wu C. Towards linear-time incremental structure from motion. In: Proceedings of the 2013 international conference on 3D vision.. Washington, DC, USA: IEEE Computer Society; 2013. p. 127–34. doi:10.1109/3DV.2013.25. 978-0-7695-5067-1
[22] Wu C. SIFTGPU: a GPU implementation of scale invariant feature transform. http://cs.unc.edu/~ccwu/siftgpu; 2007 (accessed 09.02.17).
[23] Hernández J, Marcotegui B. Morphological segmentation of building façade images. In: Proceedings of the 16th IEEE international conference on image processing. IEEE; 2009. p. 4029–32.
[24] Sinha SN, Steedly D, Szeliski R, Agrawala M, Pollefeys M. Interactive 3d architectural modeling from unordered photo collections. ACM Trans Graph 2008;27(5) 159:1–159:10. doi:10.1145/1409060.1409112.
[25] Ceylan D, Mitra NJ, Li H, Weise T, Pauly M. Factored facade acquisition using symmetric line arrangements. Comput Graph Forum (EUROGRAPHICS) 2012;31(2pt3):671–80.
[26] Sun J, Perona P. Where is the sun? Nature Neurosci 1998;1(3):183–4.
[27] Socher R, Perelygin A, Wu JY, Chuang J, Manning CD, Ng AY, et al. Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the conference on empirical methods in natural language processing, vol. 1631. Citeseer; 2013. p. 1642–1642.
[28] Grabler F, Agrawala M, Sumner RW, Pauly M. Automatic generation of tourist maps. ACM Trans Graph 2008;27(3) 100:1–100:11. doi:10.1145/1360612.1360699.
[29] Chalmers A, Debattista K, dos Santos LP. Selective rendering : computing only what you see. In: Proceedings of the 4th international conference on computer graphics and interactive techniques in Australasia and Southeast Asia - GRAPHITE. ACM; 2006. p. 9–18. http://wrap.warwick.ac.uk/48109/
[30] Lee CH, Varshney A, Jacobs DW. Mesh saliency. In: Proceedings of the ACM SIGGRAPH Papers.. New York, NY, USA: ACM; 2005. p. 659–66. doi:10.1145/1186822.1073244.
[31] Shilane P, Funkhouser T. Distinctive regions of 3d surfaces. ACM Trans Graph 2007;26(2). doi:10.1145/1243980.1243981.
[32] Song R, Liu Y, Martin RR, Rosin PL. Mesh saliency via spectral processing. ACM Trans Graph 2014;33(1) 6:1–6:17. doi:10.1145/2530691.
[33] Itti L, Koch C. Computational modelling of visual attention. Nature Rev Neurosci 2001;2(3):194–203.
[34] Bulbul A, Dahyot R. Populating virtual cities using social media. Comput Anim Virtual Worlds 2016. doi:10.1002/cav.1742. Cav.1742
[35] Du R, Varshney A. Social street view: blending immersive street views with geo-tagged social media. Proceedings of the 21st International Conference on Web3D Technology 2016:77–85.