Received: 30 June 2016 | Revised: 18 September 2016 | Accepted: 26 September 2016
DOI: 10.1002/cav.1742

WILEY

RESEARCH ARTICLE

Populating virtual cities using social media

Abdullah Bulbul[1] | **Rozenn Dahyot**[2]

[1]Yıldırım Beyazıt University, Ankara, Turkey
[2]Trinity College, Dublin, Ireland

Correspondence
Abdullah Bulbul, Yıldırım Beyazıt Üniversitesi Batı Kampüsü, Ayvalı 150. Sokak, Keçiören 06010 ANKARA.
Email: abulbul@ybu.edu.tr

Funding information
EU FP7-PEOPLE-2013-IAPP GRAI Search, Grant/Award Number: 612334

Abstract

We propose to automatically populate geo-located virtual cities by harvesting and analyzing online contents shared on social networks and websites. We show how pose and motion paths of agents can be realistically rendered using information gathered from social media. 3D cities are automatically generated using open-source information available online. To provide our final rendering of both static and dynamic urban scenes, we use Unreal game engine.

KEYWORDS

computer animation, crowd simulations, social media, virtual worlds

1 | INTRODUCTION

Generating pedestrians to populate synthetic scenes with a plausible distribution is a research area of interest in Computer Animation for the purpose of creating visually realistic virtual world. Several past studies investigated the interaction of agents, their motion paths, and the realism of the resulting animation.[1–3] Perceptual studies indicated that humans are substantially capable of differentiating real agent behaviors from synthetically generated ones; as a consequence, there have been attempts to form rules for generating perceptually plausible pedestrians.[2,4]

In this work, we propose to use online data from social networks, which are common platforms for sharing personal experiences, for populating cities with pedestrians. Indeed, by 2016, the number of active users in popular social networks exceeded 300 million users,[5] and this number is continuously increasing. This shared data provide useful information and clues about how and when people use the space in cities, their locations and orientations in that space, what they think about it, and their sentiments.[6] This content shared online can provide important information for efficiently populating 3D geo-located virtual cities by providing an accurate distribution of crowds that is both spatially heterogeneous and varying over time. This can be useful for the simulations of populated virtual cities, for generating populated game environments, and creating more lively virtual visits for tourists.

We propose to create static agents, positioned and oriented according to shared visual content on social media platforms, and dynamic agents that follow motion paths produced by exploiting GPS coordinates of the shared content (Section 3). Finally, we propose to visualize our populated virtual cities in a game engine in order to provide efficient rendering.

2 | RELATED WORK

We first present some past studies related to our work on creation of virtual cities and in virtual pedestrians.

2.1 | Virtual city generation

In a well-known study, Parish and Müller[7] produce virtual cities using a procedural approach based on L-system.[8] Their method takes population and water–land maps as inputs and generates a virtual city including streets and simplistic buildings. A context-aware shape grammar for procedural building generation was introduced by improving L-system.[9,10] Thomas and Donikian[11] proposed a city generation method that is dedicated to facilitating plausible behavioral animations for virtual agents.

More recently, several works have proposed image-based techniques for generating cities using 3D shape inference from multiple views.[12,13] However, the resulting dense 3D representation of the environment can be computationally intensive to manipulate in a game engine; in particular, when a large area of a city is animated. Our approach for generating cities uses information from OpenStreetMap as explained below.

 Copyright © 2016 John Wiley & Sons, Ltd.

```
<way id="92839167">
 <nd ref="1076670463"/>
 <nd ref="1076670445"/>
 <nd ref="1076670468"/>
 <nd ref="1076670404"/>
 <nd ref="2169156781"/>
 <nd ref="1076670463"/>
 <tag k="addr:city" v="Genève"/>
 <tag k="addr:country" v="CH"/>
 <tag k="addr:housenumber" v="3"/>
 <tag k="addr:postcode" v="1204"/>
 <tag k="addr:street" v="Rue Frank-Martin"/>
 <tag k="building" v="yes"/>
</way>

<node id="1076670463" lat="46.2027301" lon="6.1455962"/>
<node id="1076670445" lat="46.2028261" lon="6.1456942"/>
<node id="1076670468" lat="46.2027431" lon="6.1458660"/>
<node id="1076670404" lat="46.2026494" lon="6.1457675"/>
<node id="2169156781" lat="46.2026874" lon="6.1456888">
 <tag k="addr:city" v="Genève"/>
 <tag k="addr:country" v="CH"/>
 <tag k="addr:housenumber" v="3"/>
 <tag k="addr:postcode" v="1204"/>
 <tag k="addr:street" v="Rue Frank-Martin"/>
</node>
```

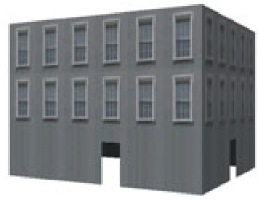



FIGURE 1 Left: definition of a building in OpenStreetMap. Each building has a contour formed by a polygon (< way >) with geo-coordinates (< node >) and may include optional extra information. Top right: corresponding area on map; bottom right: generated building

In OpenStreetMap, each area such as a building or garden is associated with a polygonal footprint for which geo-coordinates are indicated at each corner. When generating a city, we first transform all geo-coordinates to our model space such that the center of the generated area is transformed to the origin. Then for each building, a 3D mesh model is generated by converting the building contours to walls. The optional information in OpenStreetMap, "building-height" or "num-levels," determines the heights of the buildings if any of them is present. Otherwise, a default value is assigned. The textures of the buildings are assigned by repeating a unit element along the walls of the buildings. Figure 1 shows an example building definition from OpenStreetMap in xml format.

Similarly, roads are generated based on the information present in OpenStreetMap and, their usage by pedestrians is restricted to have a more natural motion path. It is possible to utilize more information such as trees, traffic signs, and bus stops to generate more detailed cities; however, for the current study, we are mainly interested in building and road structures.

While our approach is limited for representing complex architectures, the general city structure can be generated very quickly. Our simple mesh models provide real-time rendering capabilities for game engines when considering large-scale cities. Figure 2 shows examples of generated cities. Another advantage is that the generated cities can be kept up-to-date when the utilized information shared online is also kept up-to-date (on OpenStreetMap).

2.2 | Generating and controlling pedestrians

Terzopoulos proposed an artificial life framework emulating the rich complexity of pedestrians in urban environments integrating motor, perceptual, behavioral, and cognitive components for modeling pedestrians as autonomous agents.[1] The resulting platform is used as a simulator for large-scale distributed visual sensor network with applications to video surveillance. Several classes of behaviors are predefined for pedestrians in a virtual train station (e.g., commuters and tourists).[14]

Ennis et al. studied the perception of pedestrian formations[2] in virtual open urban spaces in static scenes. To generate realistic situations, images capturing the environment are manually labeled to mark people's position and orientation in the space. Ren et al.[15] detect and tract pedestrians in videos for the purpose of inserting virtual characters in the video (mixed reality). Dynamic path planning is used for animating virtual pedestrians avoiding collision with real ones.

Lerner et al[16] present a crowd simulation technique based on examples created from tracked video segments of real pedestrian crowds. In a following study, they used real crowd videos to evaluate the naturalness of individual agent behaviors.[17] In a similar study, Lee et al[18] utilize aerial videos to obtain examples of motion trajectories, which are later used in similar cases. Both of these methods require manually tracking pedestrians in the input videos to form a set of possible pedestrian behaviors. More recently, such video-driven methods are improved in terms of the capability of tracking pedestrians in videos[19] and inferring plausible motion paths.[20,21]

3 | POPULATING VIRTUAL WORLD

3.1 | System overview

Our aim is to populate the 3D environments that have a correspondence in real world, such as the ones shown in Figure 2, according to data gathered from social media. Figure 3 shows

FIGURE 2 Sample cities generated using our approach explained in Section 2.1

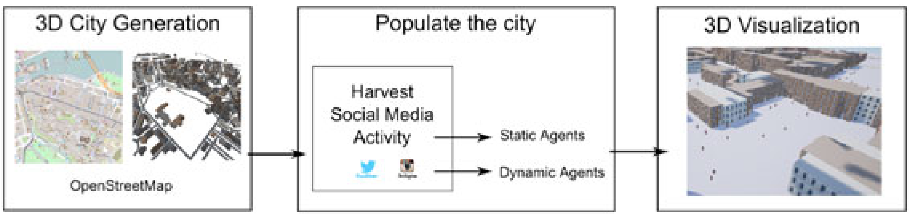


FIGURE 3 Overview

an overview of our study. Similarly to Dahyot et al.,[6] we harvest publicly shared localized content from two popular social networks, Twitter and Instagram, using their public application programming interface.

Each shared post has accompanying geo-coordinates and time stamp, as well as textual and visual content. The GPS coordinates correspond to where the post has been sent on the social network, which is different from where the image associated with the post has been taken. The GPS location of the post is accurate within about 8 m, which can be utilized for our purposes.[22] For the experiments in this paper, we have accumulated social media activity for 3 weeks around Dublin city center, which includes 2,568 posts each of which is accompanied with a photo.

We follow two different strategies to generate static and dynamic agents. For the static case, it is crucial to correctly identify the orientations of individual agents for a perceptually plausible representation.[2] Therefore, we utilize the shared visual content to determine the visible scene and the location of the photographer.

Note that the geo-location accompanying a harvested post belongs to the location of the mobile device when sharing the content rather than the location where the photo is taken. Thus, it is not reliable to assume that the photo is taken at the specified location. However, we can reliably assume that the user has physically appeared at that location to post the content in social media, which is utilized for generating the paths of dynamic agents.

3.2 | Static agents

Our main idea is to generate virtual agents whose position and orientation are determined according to those of real social media users when they take their shared photos. We assume that people have similar visual interest; thus, if a region is photographed by a user, it is assumed that this spot is visually interesting, and we can expect other people in the space to share the same focus of attention and look at the same spot. The camera parameters that are used to take the shared photos need to be found first to infer the position and the orientation of the photographer. To achieve that, similarly to

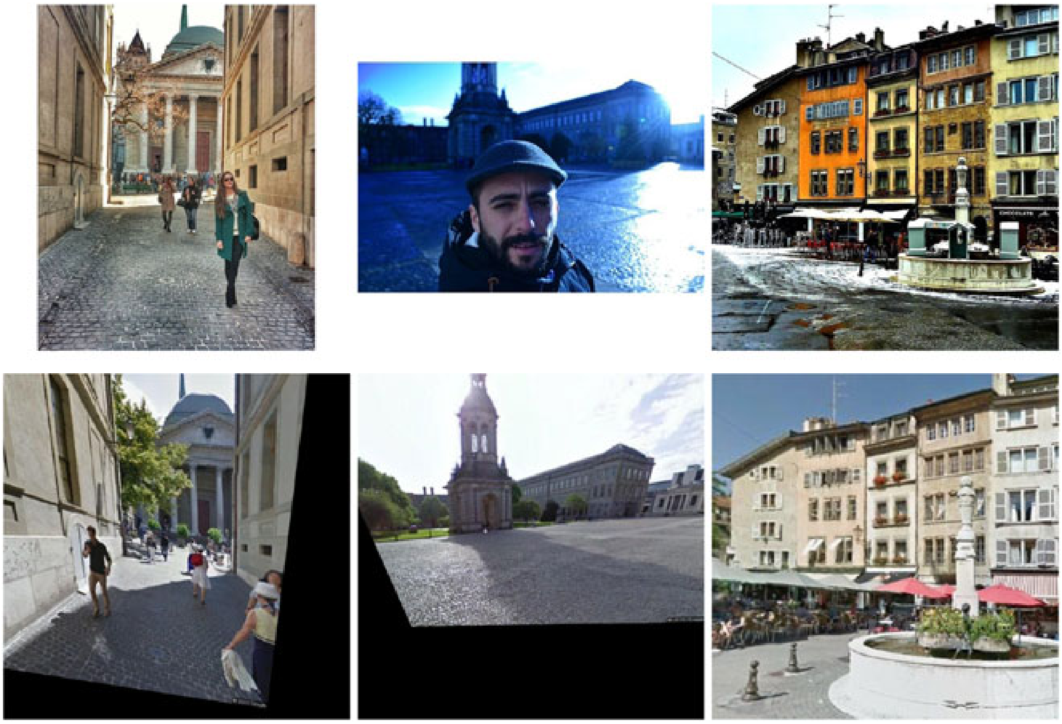


FIGURE 4 Sample results from extracting photo locations. Top row shows the input images and bottom row shows the found correspondences from the reference set. The black areas are due to transformations to match the visible areas of the images

Zhang et al.,[23] we search the shared visible region in a set of reference images for which the location and orientation information is already known. The reference set is formed by Google Street View images, which are homogeneously distributed along streets and publicly available with a maximum resolution of 640×640.

3.2.1 | Extracting locations

Image localization is based on scale invariant feature transform (SIFT) feature matching between the photos obtained from social media and a set of reference images queried from Google Street View. It is assumed that the images in the reference set have accurate geo-locations, that is, with an error not exceeding a few meters. Each shared post is also accompanied with a GPS-based geo-location corresponding to the location of the mobile device at the time of posting, rather than the location where the image was taken. Therefore, to find where the image was actually taken, we search available reference images within a radius r of the posted content's location. A larger value of r increases the probability of having the correct correspondence in the reference set while requiring more comparisons as the size of the reference set is proportional to r^2.

It is worth noting that the majority of the images do not have a correct correspondence in the reference set, as most of the shared photos are taken indoors or their content are not relevant to surrounding visual environment, for example, personal items, food, and so on. Besides, the photos may be taken at very far locations, which makes it impractical to include them in the set of reference images. Consequently, we target the photos of the environment that are shared within a sufficiently close location to where they are actually taken and set r as 100 m.

Camera parameters correspond to location, orientation, and focal length. Parameters for an input image are initialized with the parameters of the most similar image in the reference set according to SIFT feature matching.[24] Then, to avoid incorrectly matched image pairs, a Fundamental matrix is searched between corresponding feature points of the input and reference images with random sample consensus (RANSAC) method. If a Fundamental matrix cannot be constructed, the image pairs are eliminated.

The camera parameters of remaining input images are further refined to match with the captured physical regions. We achieve this by employing homography, which provides a perspective transformation matrix between the input and reference images. Using the acquired transformation matrix, we refine the orientation and focal length of each input image for a better alignment with the visible area in the corresponding reference image. Figure 4 shows several results from detected locations. Out of the 2,568 photos in our input data set, 171 locations are extracted for static agents. The main reason for the low recall rate is the high percentage of irrelevant photos, that is, more than 65% of the posted pictures are unusable as these capture food or personal items for instance.

3.2.2 | Generating groups

Group behaviors of agents have also been well-studied.[25,26] It is more realistic to have groups of agents rather than having only agents on their own. To generate groups of people, we again utilize the shared visual content, and we infer the number of members in a group according to the number of detected frontal faces[27] in the photo, assuming that the faces directed towards the camera are related to the photographer sharing the picture.

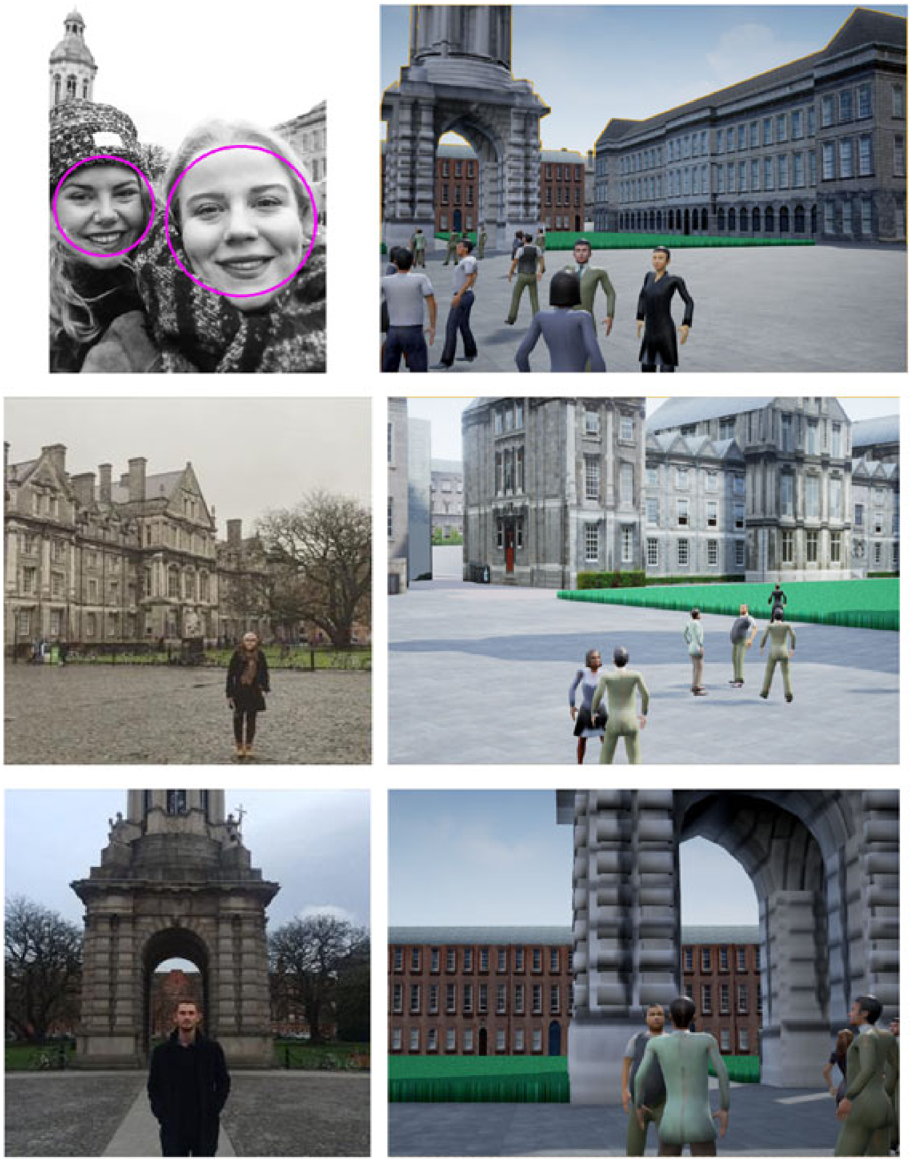


FIGURE 5 Left: input images from social media in the front square of Trinity College Dublin. Right: corresponding groups of agents (building models from Metropolis project,[28] for a better evaluation). For instance, in the top row, the two detected faces (depicted with the circles) lead to generating two agents and the photographer in front of the campanile

FIGURE 6 Generated groups formations. Left: the agents in the group form a conversation circle. Right: the agents are placed forming a line for taking a photo

After determining the number of agents in a group, for a plausible group behavior, it is important to place and orient each of the members in a meaningful way according to the context and other agents. For that purpose, we either make the group members form a conversation circle or line them up in front of the point of interest while one of them takes

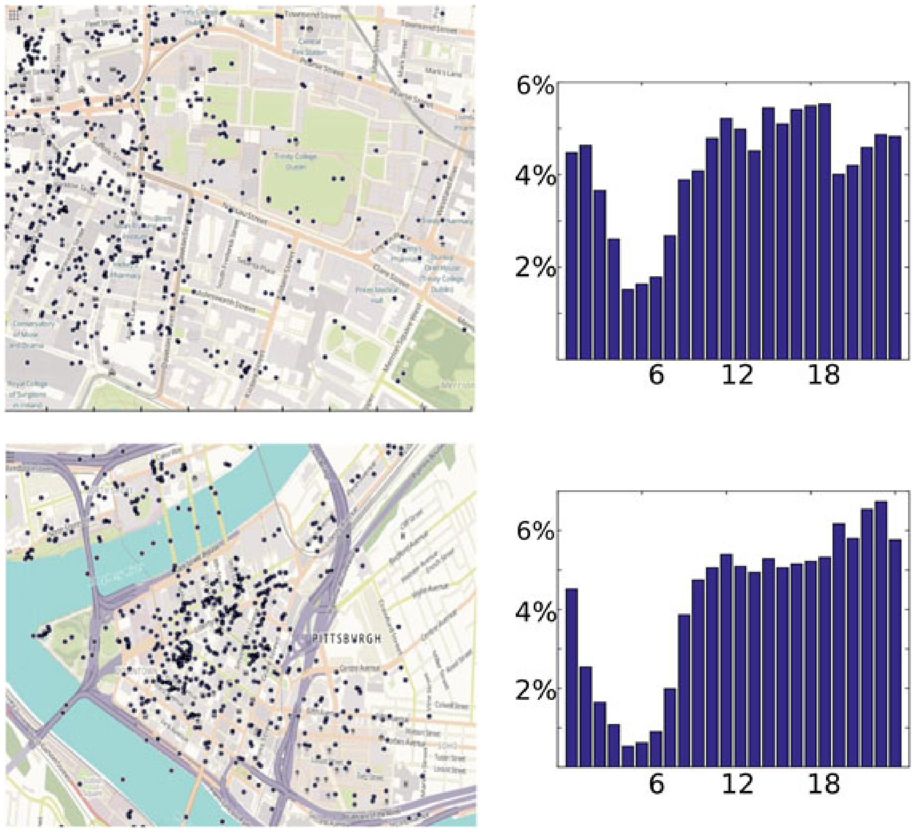


FIGURE 7 Left: spatial distribution of social activity. Right: temporal distribution of activity, each bin corresponds to an hour of day. (Top row: Dublin, bottom row: Pittsburgh)

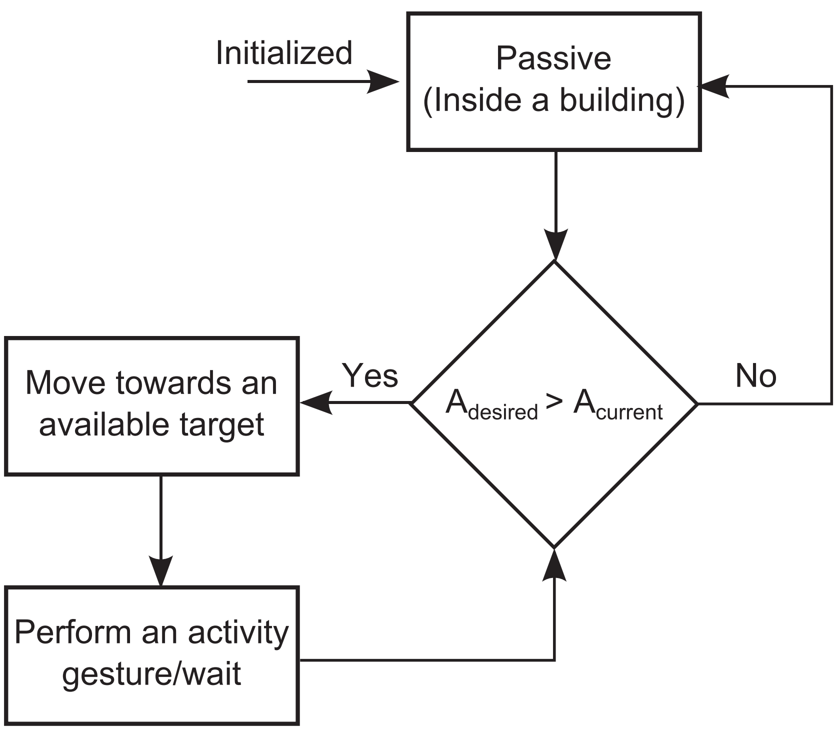


FIGURE 8 A generated agent's states of motion

their picture, mimicking the scenario in the shared photo (see Figures 5 and 6).

3.3 | Dynamic agents

When determining motion paths, predefined motion paths lead to uniform agent behaviors and using totally random paths lead to non-natural walking trajectories. We propose a method that distributes agents according to real-users' social media activity. For instance, a street with a higher number of social media activity should appear crowded with more pedestrians in our virtual environment.

Although social media activity does not directly provide the walking trajectories of individual users, that is, a very few proportion of users share multiple posts at different locations during a day, the overall activity accumulated from the entire community can still inform us about the spatial and temporal distribution of pedestrians in a city. Figure 7 shows the distribution of social media activity around several cities.

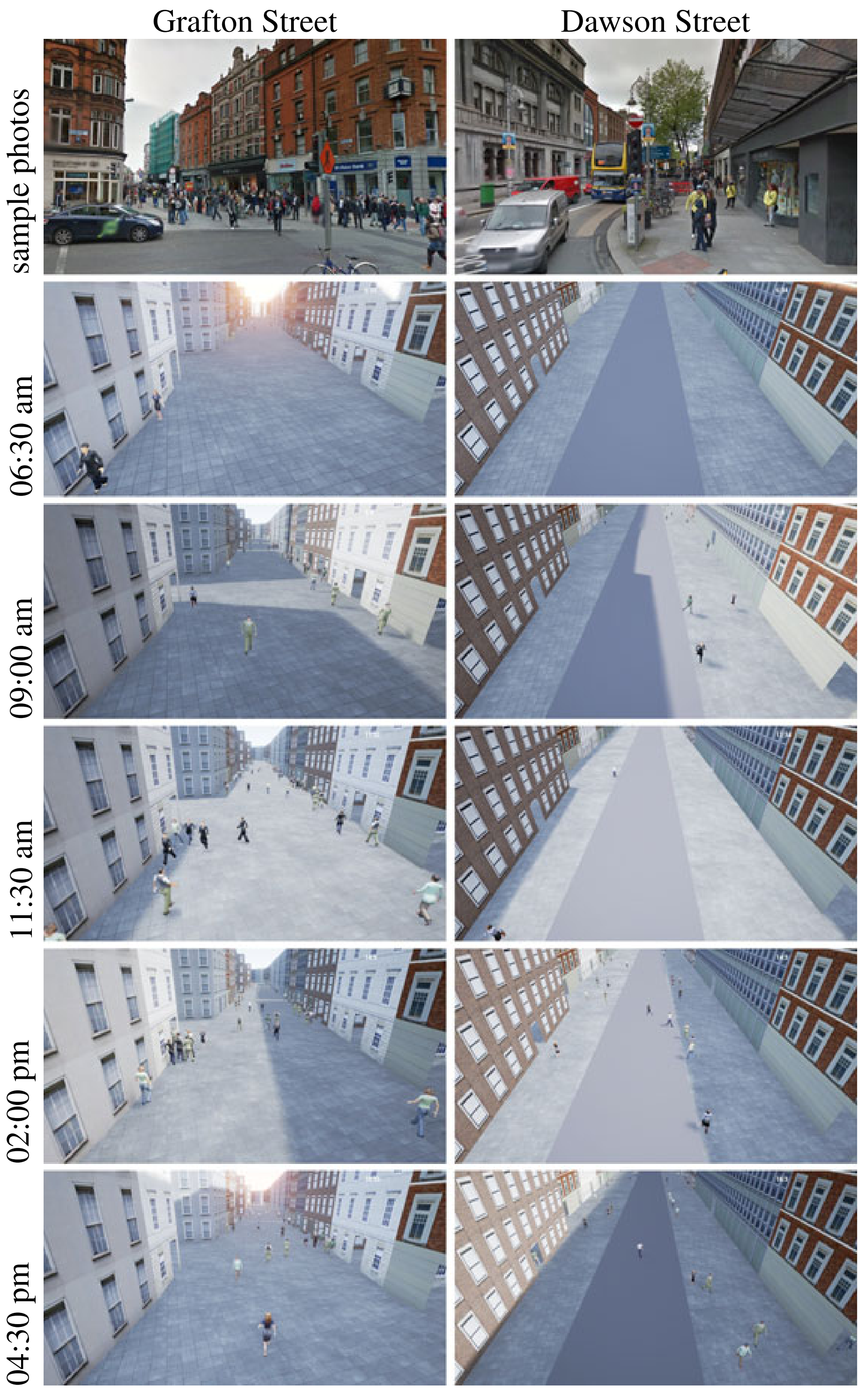


FIGURE 9 Simultaneous views of different areas at the same city (with $M = 1{,}500$)

The geo-locations of the shared posts are directly utilized in dynamic agent generation part without analyzing the contents of the shared media as we are mainly interested in where the user has physically appeared. After accumulating social media activity for a sufficiently long time (about 3 weeks in our experiments), we get a set of posts each having a location and time information. While this information is used to control the density of pedestrians, our system takes the maximum number of active pedestrians M as an input from the user providing a control on the overall density. At each time of the day, the number of dynamic agents are determined by M and the temporal distribution of social media activity.

We initialize M passive agents inside randomly selected buildings. By passive pedestrians we mean the ones standing inside a building, therefore, they are not visible in the streets. At the most active time of the day, all M pedestrians become active, and at other times, the number of active agents are adjusted proportionally to the volume of social media activity at that time.

To achieve that, our system initially calculates the largest number of posts shared within a 1-hr window of the whole day (noted $N_{\max}$). Then it continuously keeps track of the number of posts shared within the following hour, denoted by $N_{current}$. Then the desired number of active agents becomes

$$A_{desired} = M \times \frac{N_{current}}{N_{\max}}. \tag{1}$$

Each active agent moves to one of the currently available targets, which is formed by the locations of the posts shared within the following hour, that is, contributing to $N_{current}$. It is worth noting that the available targets are formed by the activity of the whole community rather than consecutive posts from a single individual. When the agent reaches the target, he stands for a while and performs a gesture, for example, looks at his phone, takes a picture, and so on; then the agent may become passive, if the number of currently active agents $A_{current}$ is greater than $A_{desired}$ or stay active otherwise. This procedure is shown in Figure 8. Such a strategy provides spatially and temporally different densities of agents reflecting a real-world distribution. Figure 9 shows an example of dynamic agent management process. As seen in this figure, at different times of the day, agents are differently distributed in the streets. Note that the proposed method defines motion targets rather than precisely providing motion trajectories for each agent. It is possible to employ other agent behavior-related studies such as[3] that determines how an agent travels between two points and how he interacts with other agents.

4 | DISCUSSION AND CONCLUSION

We have proposed a processing pipeline that automatically generate virtual cities using up-to-date online geographic information system (OpenStreetMap) and populate these worlds using information extracted from real people posting on social networks (Twitter, Instagram). Using additional geo-located image repositories (Google Street View), orientation of pedestrians taking and sharing pictures is recovered to inform the game engine of both location and orientation of agents in the virtual world. The static and dynamic rendering has been performed using the Unreal engine.

Generating plausible pedestrians, placed and oriented in a meaningful way or moving in the city to reflect real distribution of crowds, can find useful applications in games, for instance, where using social media data to generate agents can provide timely information about the distribution of people in a city. Information extracted from social networks can ultimately have an impact for mixed, virtual, and augmented realities.

Our strategy directs agents to places more often when they are more demanded by real people, for instance, entertainment-related areas may be more crowded at later hours or at weekend. It can be argued that the proposed method for locating static agents is biased towards reflecting tourists' activities rather than gathering overall behavior of all people in the city. While this is true to some extent, the shared photos still provide meaningful places for pedestrians to stand for a while such as visual points of interest and places where temporary events happen.

It is worth noting that, in this study, we do not use a one-to-one mapping between social media activity and virtual pedestrians as there are far less activity in social media compared to real pedestrians. Therefore, as explained in Section 3.3, we accumulate the activity for a period of time, for example, 3 weeks in our experiments, to get sufficient data of pedestrians providing a smoother distribution. This strategy also helps compensating the discarded photos that are shared outside of the search radius (Section 3.2).

Our system is currently having a few limitations that will be overcome in our future work. This includes improving our building geometry and textures that are currently not realistic and implementing collision avoidance for the agents. Future work will also investigate how social media and online websites can provide rich, dynamic, and realistic content for creating virtual visits that are both entertaining and interactive. For instance, it may be possible to acquire semantic information related to the different areas of the city by analyzing the textual content of the shared posts and utilizing extra data available in OpenStreetMap such as the types of the buildings, for example, residence, retail, school, and so on, or opening hours of retail shops. Such semantic information can be used for refining the behaviors and the motion paths of the agents.

Investigating the possibilities of utilizing social media and other publicly available sources for simulating vehicle traffic in virtual worlds on top of pedestrians is another interesting future research direction. Recently, Chao et al.[29] have analyzed this mixed traffic situation and proposed a method for modeling vehicle–pedestrian interaction behaviors.

Evaluation of the presented study is a challenging one as there is not a ground truth or state-of-the-art methods that are directly comparable. We have visually presented the results of static and dynamic pedestrian generation methods (Figures 5 and 9). One possible means of further verification is subjective evaluation of the results by people who are familiar with the simulated areas. Comprehensive evaluation of the presented methods also remains as an important direction of future work.

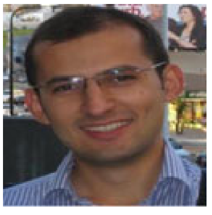
ACKNOWLEDGMENTS

This work has been supported by EU FP7-PEOPLE-2013-IAPP GRAI Search grant (612334). 3D humanoid models are generated using Autodesk Character Generator.

REFERENCES

1. Terzopoulos D. A reality emulator featuring autonomous virtual pedestrians and its application to distributed visual surveillance. IEEE Virtual Reality 2008 Conf. (VR 08); Reno, NV; 2008. p.1–4.
2. Ennis C, Peters C, O'Sullivan C. Perceptual effects of scene context and viewpoint for virtual pedestrian crowds. ACM Trans Appl Percept. 2011;8(2):10:1–10:22.
3. Pettré J, Ciechomski PdH, Maïm J, Yersin B, Laumond J-P, Thalmann D. Real-time navigating crowds: Scalable simulation and rendering. Computer Animation and Virtual Worlds. 2006;17(3-4):445–455.
4. Peters C, Ennis C, McDonnell R, O' Sullivan C. Crowds in context: Evaluating the perceptual plausibility of pedestrian orientations. Proceedings of Eurographics Short Papers; Crete, Greece 2008. p.227–230.
5. Leading global social networks. Available from: http://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/ [Accessed on 04/02/2016].
6. Dahyot R, Brady C, Bourges C, Bulbul A. Information visualisation for social media analytics. International Workshop on Computational Intelligence for Multimedia Understanding; Prague, Czech Republic; 2015.
7. Parish YIH, Müller P. Procedural modeling of cities. Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '01.ACM: New York, NY, USA; 2001. p.301–308.
8. Prusinkiewicz P, Lindenmayer A. The algorithmic beauty of plants. New York, NY, USA: Springer-Verlag New York, Inc.; 1990.
9. Müller P, Wonka P, Haegler S, Ulmer A, Van Gool L. Procedural modeling of buildings. ACM Trans Graph. 2006;25(3):614–623.
10. Schwarz M, Müller P. Advanced procedural modeling of architecture. ACM Trans Graph. 2015;34(4):107:1–107:12.
11. Thomas G, Donikian S. Modelling virtual cities dedicated to behavioural animation. Comput Graph Forum. 2000;19(3):71–80.
12. Snavely N, Seitz SM, Szeliski R. Modeling the world from internet photo collections. Int J Comput Vis. 2008;80(2):189–210.
13. Xiao J, Fang T, Zhao P, Lhuillier M, Quan L. Image-based street-side city modeling. ACM Trans Graph. December 2009;28(5):114:1–114:12.
14. Shao W, Terzopoulos D. Autonomous pedestrians. Graph Models. 2007;69(5-6):246–274.
15. Ren Z, Gai W, Zhong F, Pettré J, Peng Q. Inserting virtual pedestrians into pedestrian groups video with behavior consistency. Visual Comput. 2013;29(9):927–936.
16. Lerner A, Chrysanthou Y, Lischinski D. Crowds by example. Comput Graph Forum. 2007;26(3):655–664.
17. Lerner A, Chrysanthou Y, Shamir A, Cohen-Or D. Data driven evaluation of crowds. Proceedings of the 2nd International Workshop on Motion in Games, MIG '09.Springer-Verlag: Berlin, Heidelberg; 2009. p.75–83.
18. Lee KH, Choi MGl, Hong Q, Lee J. Group behavior from video: A data-driven approach to crowd simulation. Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '07.Eurographics Association: Aire-la-Ville, Switzerland, Switzerland; 2007. p.109–118.
19. Zhang K, Zhang L, Yang M-H. Real-time compressive tracking. Proceedings of the 12th European Conference on Computer Vision - Volume Part III, ECCV'12.Springer-Verlag: Berlin, Heidelberg; 2012. p.864–877.
20. Bera A, Kim S, Manocha D. Efficient trajectory extraction and parameter learning for data-driven crowd simulation. Proceedings of the 41st Graphics Interface Conference, GI '15.Canadian Information Processing Society:Toronto, Ont., Canada, Canada; 2015. p.65–72.
21. Kapadia M, Chiang I-k, Thomas T, Badler NI, Kider JT, Jr.. Efficient motion retrieval in large motion databases. Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, I3D '13.ACM:New York, NY, USA; 2013. p.19–28.
22. Zandbergen PA. Accuracy of iphone locations: A comparison of assisted gps, wifi and cellular positioning. Trans GIS. 2009;13:5–25.
23. Zhang W, Kosecka J. Image based localization in urban environments. Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06), 3DPVT '06.IEEE Computer Society:Washington, DC, USA; 2006. p.33–40.
24. Wu C. SiftGPU: A GPU implementation of scale invariant feature transform (SIFT). Available from: http://cs.unc.edu/~ccwu/siftgpu; 2007. Accessed January 17, 2016.
25. Ennis C, O'Sullivan C. Perceptually plausible formations for virtual conversers. Comput Animat Virtual Worlds. 2012;23(3-4):321–329.
26. Karamouzas I, Overmars M. Simulating and evaluating the local behavior of small pedestrian groups. IEEE Transactions on Visualization and Computer Graphics. 2012;18(3):394–406.
27. Viola P, Jones M. Rapid object detection using a boosted cascade of simple features. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001. CVPR '01, vol. 1; Kauai, HI, USA 2001. p.I–511–I–518.
28. O' Sullivan C, Ennis C. Metropolis: Multisensory simulation of a populated city. Third International Conference on Games and Virtual Worlds for Serious Applications; Athens, Greece 2011.
29. Chao Q, Deng Z, Jin X. Vehicle-pedestrian interaction for mixed traffic simulation. Comput Animat Virtual Worlds. 2015;26(3-4):405–412.

AUTHORS BIOGRAPHIES



Abdullah Bulbul is currently an assistant professor in the Computer Engineering Department of Yildirim Beyazit University in Turkey. His research interests include Computer Graphics, 3D vision, perception, and 3D reconstruction. He received his BS and PhD degrees in Computer Engineering from Ihsan Dogramaci Bilkent University in 2007 and 2012, respectively. During his PhD studies, he has also worked for 3DPhone project (EU 7th FP). Between 2012 and 2014, he worked as a postdoctoral researcher in Vision Science group of University of California at Berkeley, which is followed by another postdoctoral study at Trinity College Dublin between 2012 and 2016 where he has worked for GRAISearch project (EU 7th FP).

Rozenn Dahyot is an associate professor in Statistics in Trinity College Dublin (Ireland), with research interests in mathematical statistics, digital signal processing, and visual data analytics. She received her PhD degree from Université Louis Pasteur in Strasbourg (France) in 2001, working on robust detection and recognition of objects in road scene image databases. From 2002 to 2005, she worked as a research fellow in Trinity College Dublin and in Cambridge university (UK), and since 2005, she has been lecturing in mathematics and statistics in the School of Computer Science and Statistics in Trinity College Dublin. She is the coordinator of the project GRAISearch FP7-PEOPLE-2013-IAPP (612334) on the 'Use of Graphics Rendering and Artificial Intelligence for Improved Mobile Search Capabilities' (2014–2018).

How to cite this article: Bulbul A, Dahyot R. Populating virtual cities using social media. *Comput Anim Virtual Worlds*. 2017;28:e1742. https://doi.org/10.1002/cav.1742