Building and Environment 168 (2020) 106475

Contents lists available at ScienceDirect

Building and Environment

journal homepage: http://www.elsevier.com/locate/buildenv

Technical note

GSV2SVF-an interactive GIS tool for sky, tree and building view factor estimation from street view photographs

Jianming Liang [a,b], Jianhua Gong [a,b], Jinming Zhang [a,c,*], Yi Li [a,b], Dong Wu [a,d], Guoyong Zhang [a,d]

[a] State Key Laboratory of Remote Sensing Science, Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing, 100101, China
[b] Zhejiang-CAS Application Center for Geoinformatics, Jiashan, Zhejiang, 314100, China
[c] Institute of Geospatial Information, Information Engineering University, Zhengzhou, 450052, China
[d] University of Chinese Academy of Sciences, Beijing, 100049, China

ARTICLE INFO

Keywords:
Sky view factor
Tree view factor
Building view factor
Google street view
Urban environment

ABSTRACT

Sky View Factor (SVF) is a commonly used indicator of urban geometry. The availability of street-level SVF measurements has been fairly limited due to the high costs of field survey. The Google Street View (GSV) serves a massive storage of panorama data that can be utilized to obtain SVF measurements. Yet, automatic extraction of SVFs from panoramas is a complicated process that involves multiple sophisticated computation technologies including machine learning, big image data processing, SVF estimation and geographic information systems (GIS), which constitute major hurdles for the end users. In this light, we developed an easy-to-use GIS-integrated tool (GSV2SVF) to streamline the workflow of extracting SVFs from GSV images and therefore making this vast treasure trove of information conveniently available to everyone at a mouse click. As by-products in addition to the SVF, the results obtained from each GSV panorama are accompanied with the tree view factor (TVF) and the building view factor (BVF), which together can provide a more holistic characterization of the outdoor built environment. GSV2SVF is freely available with source code at https://github.com/jian9695/GSV2SVF. A video is available at https://github.com/jian9695/GSV2SVF/blob/master/Video.mp4 and https://youtu.be/k00wCnuzuvE.

1. Introduction

The sky view factor (SVF), tree view factor (TVF) and building view factor (BVF) respectively represents the proportion of the hemisphere that is accounted for by the sky, trees and buildings when viewing from a ground location. SVF is a dimensionless measure varies between 0 and 1 used to characterize the sky openness on natural terrains or built-up surfaces [1]. SVF, TVF and BVF are three important visual indices of the urban outdoor environment [2].

Applications of SVF have been well-documented in the literature. From a physical point of view, SVF determines the amount of solar radiation lost due to sky obstruction. From a pedestrians' or residents' point of view, SVF closely relates to accessibility to light and thermal comfort. Previous research suggested significant relationships between SVF and various urban environmental indicators. The relationships between SVF and urban temperature, although found to be a strong one in many cases, varies with location, time and season [3]. A case study of the relationship between SVF and thermal comfort in a hot dry climate provides evidence that the compact urban form is more thermally comfortable than dispersed form [4]. An inverse relationship between noise and SVF was found which can be explained by fact that buildings can effectively block traffic noise [5]. SVF was found to have a significant influence on micro-scale wind environment [6]. Global positioning system (GPS) reception was shown to be able to explain over 69% of the variation of SVF in urban areas [7].

Traditionally, a number of SVF estimation methods and tools have been developed to derive SVF from various sources, including hemispheric photographs [8], digital elevation models (DEMs) [9], Lidar-based digital surface models (DSMs) [10], 3D city models [11] and GPS signal reception [7].

Recently, Google Street View (GSV) [12] has been increasingly known as a massive untapped source of SVF information.

* Corresponding author. State Key Laboratory of Remote Sensing Science, Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing, 100101, China.
E-mail address: zhangjm@radi.ac.cn (J. Zhang).

https://doi.org/10.1016/j.buildenv.2019.106475
Received 27 August 2019; Received in revised form 7 October 2019; Accepted 12 October 2019
Available online 13 October 2019
0360-1323/

Table 1
Main features of GSV2SVF.

Feature	Operations	Feedback
Map navigation	1. Interactive navigation by mouse movement 2. Zoom to coordinates 3. Switch base maps	Refreshed map content
GSV panorama viewing	Right click to update GSV view	Updated GSV
Sample collection	1. Add a sample by coordinates 2. Add samples by left click	Queued GSV samples
Sample densification	Specify a distance at which interpolation will take place.	Densified samples
Import samples	Import sample locations from an Esri Shapefile for batch processing	Queued GSV samples
View factor computation	Hit the Compute button to start processing.	SVF, TVF and BVF results in structured formats
View factor mapping	Draw SVF, TVF or BVF values in colored markers in the map.	Map of SVF, TVF or BVF distribution
Export	Define a rectangular region and export the selected results to a designated folder.	Results in text and Shapefile.

Carrasco-Hernandez et al. [13] was among the earliest to explore the feasibility of estimating SVF from GSV. The approach by Carrasco-Hernandez et al. [13], however, does not scale to large-scale SVF acquisition tasks due to its reliance on manual sky delineation. Liang et al. [14] automated sky delineation using a deep-learning framework which classifies GSV into 12 classes including sky, vegetation and building and effectively overcame the challenge of obtaining SVF data from GSV on large scales. To date, a plethora of deep learning and GSV-based SVF studies have been reported with focuses on utilization of GSV to map the urban environment and help answer socio-environmental questions. Notably, Gong et al. [2] developed a more comprehensive framework which extended the GSV-derived sky view factor information to include TVF and BVF. Zeng et al. [15] proposed a non-deep learning approach to achieve automatic sky delineation and reported a high computational performance and good accuracy. However, unlike the deep learning frameworks of Liang et al. [14] and Gong et al. [2], the non-deep learning approach [15] was designed only to detect sky areas.

GSV provide a close-up look at many aspects of the urban space including buildings, roads, pedestrians, traffic, greenery and socioeconomic activities, and this information have been effectively used in various types of spatial analysis. Among these analyses, the visible green space and visual enclosure for street, which can be measured by TVF and SVF, were found to have significant influence on street walkability [16] and mental health [17].

In spite of the plethora of studies utilizing deep learning to extract information from GSV, little has been done to ease the accessibility of SVF information from GSV for the average users, who do not normally have the level of technical expertise needed to implement an operational solution. For example, the GSV-based SVF estimation solution developed by Liang et al. [14], although proved capable of providing reliable SVF estimates at low costs and can potentially be extended to account for TVF an BVF, requires the integration of multiple challenging computation technologies including deep learning, image processing and geographic information systems (GIS). Furthermore, in addition to these challenges, to transform such a prototype solution into an operational tool, many other technical issues need to be resolved, including software robustness, usability and ease of use, and all of which requires extensive design, development and test efforts.

In this light, we take the initiative to develop an integrated tool based on Liang et al. [14]'s prototype solution to share with the average users, aiming firstly to unleash the potential of GSV for providing low-cost SVF data and ease the accessibility of SVF information and secondary to propagate the hard-earned knowledge and technical experience.

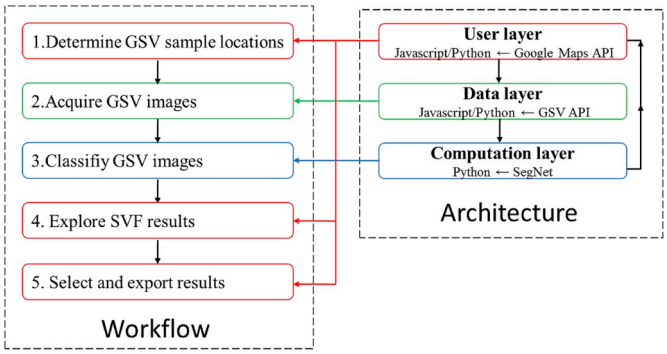

Fig. 1. Software architecture and workflow.

2. Methods

The automatic SVF estimation method [14] is a four-step workflow. Firstly, a set of image tiles are downloaded from a street view service such as GSV [12]. Street view images can be requested from these services via their respective APIs. Secondly, the requested image tiles are stitched together into a full-view panoramic image in a cylindrical projection. Thirdly, the panoramic image is fed to a deep learning image classifier for pixel-wise classification. The image classifier used, SegNet [18], is a fully convolutional neural network developed specifically to achieve pixel-wise sematic segmentation. It classifies a street scene into 12 classes including sky, tree and building. When used for SVF estimation, the classified panoramic image, in a cylindrical projection, is delineated into sky and non-sky areas and transformed to an azimuthal projection. The SVF is calculated according to the following algorithm [14]:

$$SVF = \sum_{i=0}^{n} \omega \times sky(i) \Big/ \sum_{i=0}^{n} \omega \tag{1}$$

where is n the total number of pixels, ω is a weight associated with each pixel, and *sky(i)* is a binary function determined by whether the sky is visible at a pixel [67]:

$$f(i) = \begin{cases} 1, & if\ alpha = 0 (pixel\ is\ sky) \\ 0, & if\ alpha > 0 (pixel\ is\ not\ sky) \end{cases} \tag{2}$$

In estimating TVF and BVF, the binary function *sky(i)* in equation (1) is substituted to make the following forms:

$$TVF = \sum_{i=0}^{n} \omega \times tree(i) \Big/ \sum_{i=0}^{n} \omega \tag{3}$$

$$BVF = \sum_{i=0}^{n} \omega \times building(i) \Big/ \sum_{i=0}^{n} \omega \tag{4}$$

where *tree(i)* and *building(i)* respectively determines whether a pixel is tree and building:

The complex SVF extraction processes are integrated into an easy-to-use interactive software framework. Two main use cases were being considered in designing the software: (1) the primary use case is interactively exploring GSV images in Google Maps and extracting, mapping and exporting SVFs, TVFs and BVFs; (2) the second use case is batch-processing a large number of GSV images with a user-supplied coordinate list to map an entire street, street network, neighborhood or even an entire city. Considering these needs, we developed a streamlined software architecture consisting of three logical layers, the user layer, the data layer and the computation layer (Fig. 1).

The user layer is a Graphics User Interface (GUI) designed to interact with Google Maps and receive user input (Fig. 2). It was programmed using a combination of JavaScript and Python. The GUI window is a Python application that hosts a set of user action buttons and two central

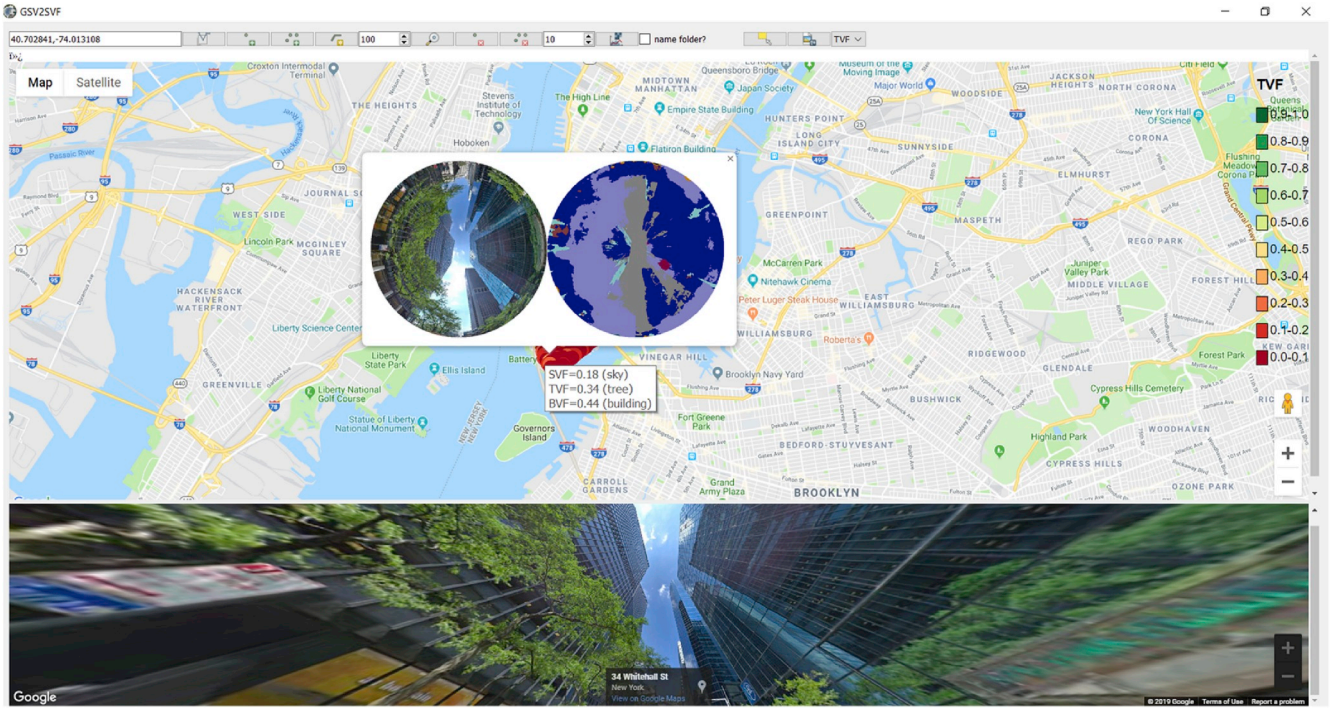

Fig. 2. The GSV2SVF user interface.

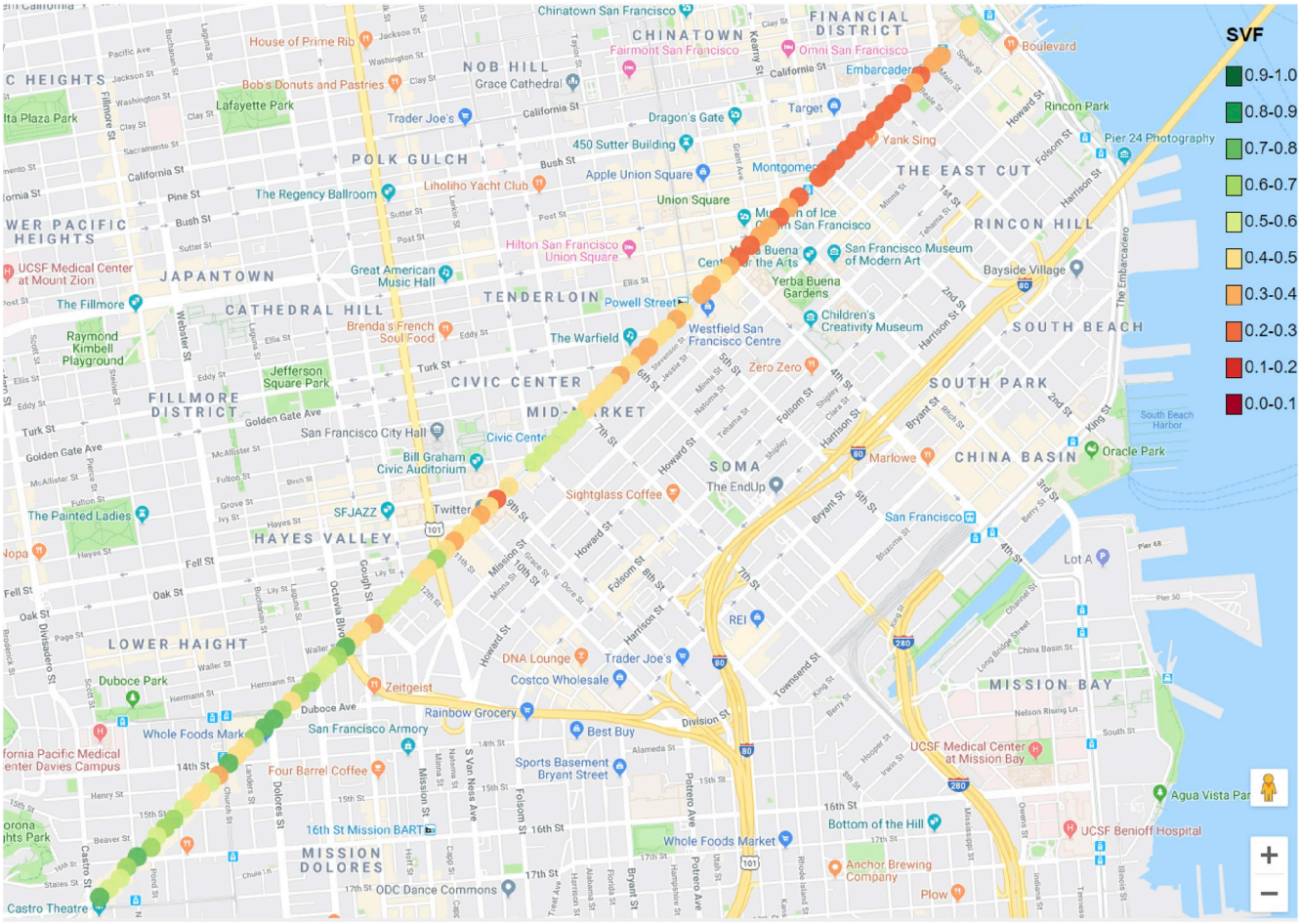

Fig. 3. Mapping of SVF distribution along Market Street, downtown San Francisco in GSV2SVF.

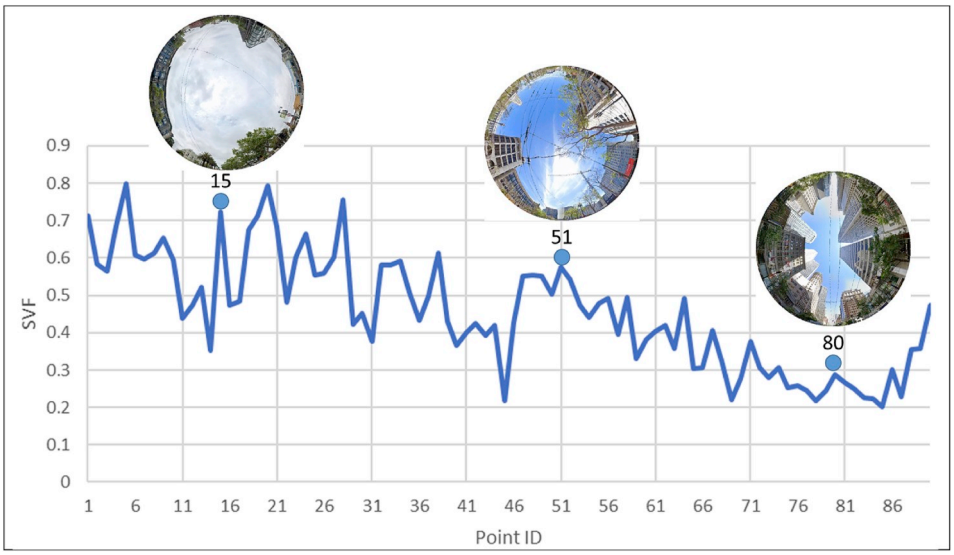

Fig. 4. Statistical analysis of SVF distribution for Market Street, downtown San Francisco.

views respectively displaying Google Maps and GSV. The interactive Google Maps and GSV views are embedded in the Python window using the Google JavaScript APIs. The user layer listens to user actions and handles user requests for view factor estimation. The requests are forwarded to the data layer for querying and downloading GSV images, which are reconstructed into full panoramas and then fed to the computation layer for classification and view factor estimation. The results are returned from the computation layer to the user layer and visualized in Google Maps, and they can also be exported in ESRI Shapefile (.shp) and text (.csv) formats for external analysis.

The data layer is also constructed using a combination of JavaScript and Python code. On the JavaScript end, the GSV API method "getPanorama()" is called to query the GSV database for availability of data within a radius of the requested coordinates, and a StreetViewPanoramaData object that contains the meta-data of a GSV panorama, including the actual coordinates and an identifier, is returned (https://developers.google.com/maps/documentation/javascript/streetview). The actual coordinates at which a panorama is located can differ from the coordinates requested, as panoramic photographs are collected at varying intervals rather than continually. The image tiles associated with a panorama are downloaded from GSV by HTTP request with the identifier.

Within the computation layer, panoramic images are classified using Caffe SegNet and transformed into fisheye images for SVF, TVF and BVF estimation. The Caffe code, written in C++ with a Python wrapper, was pulled from GitHub and compiled locally. Panoramic images acquired from GSV are classified using Caffe SegNet into 12 categories, including sky, tree and building. The results returned to the user layer for a requested location consists of a panoramic image, a fisheye image, a classified fisheye image and a text file that stores the coordinates, identifier and SVF/TVF/BVF values.

Finally, GSV2SVF maintains a caching system that stores the results from all previous sessions. The cached results are loaded into Google Maps when GSV2SVF starts and can be exported by spatial selection.

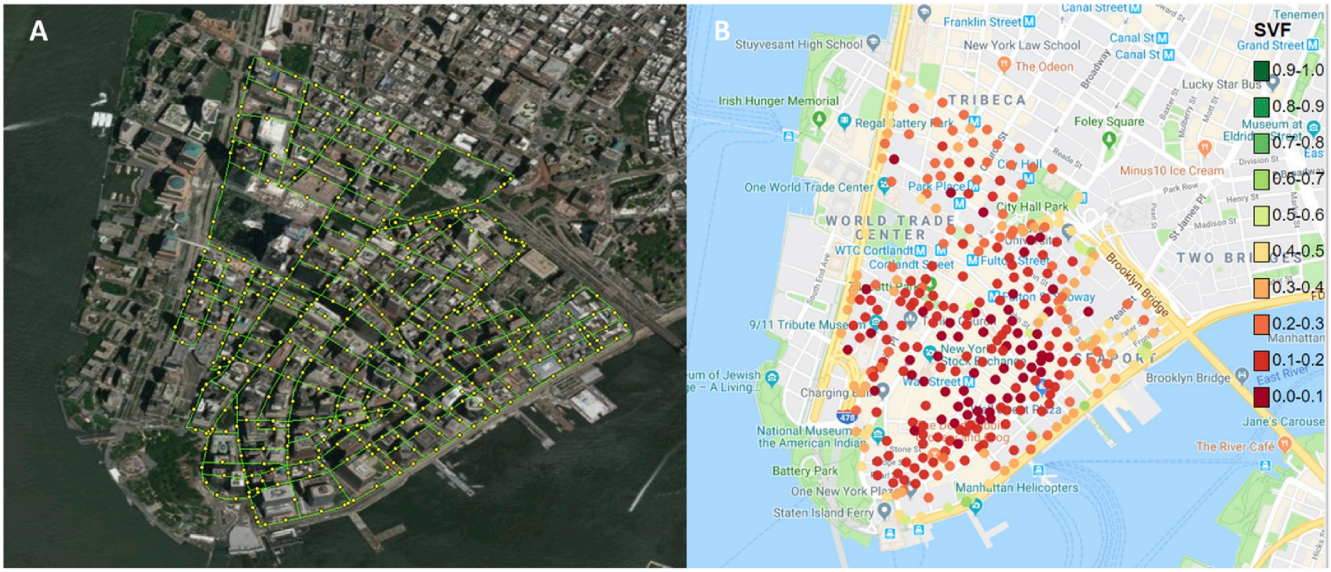

Fig. 5. Mapping of SVF distribution in Manhattan's Financial District: (a) road network data used to provide sample locations; (2) spatial distribution of SVF rendered in red to green (0–1). (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

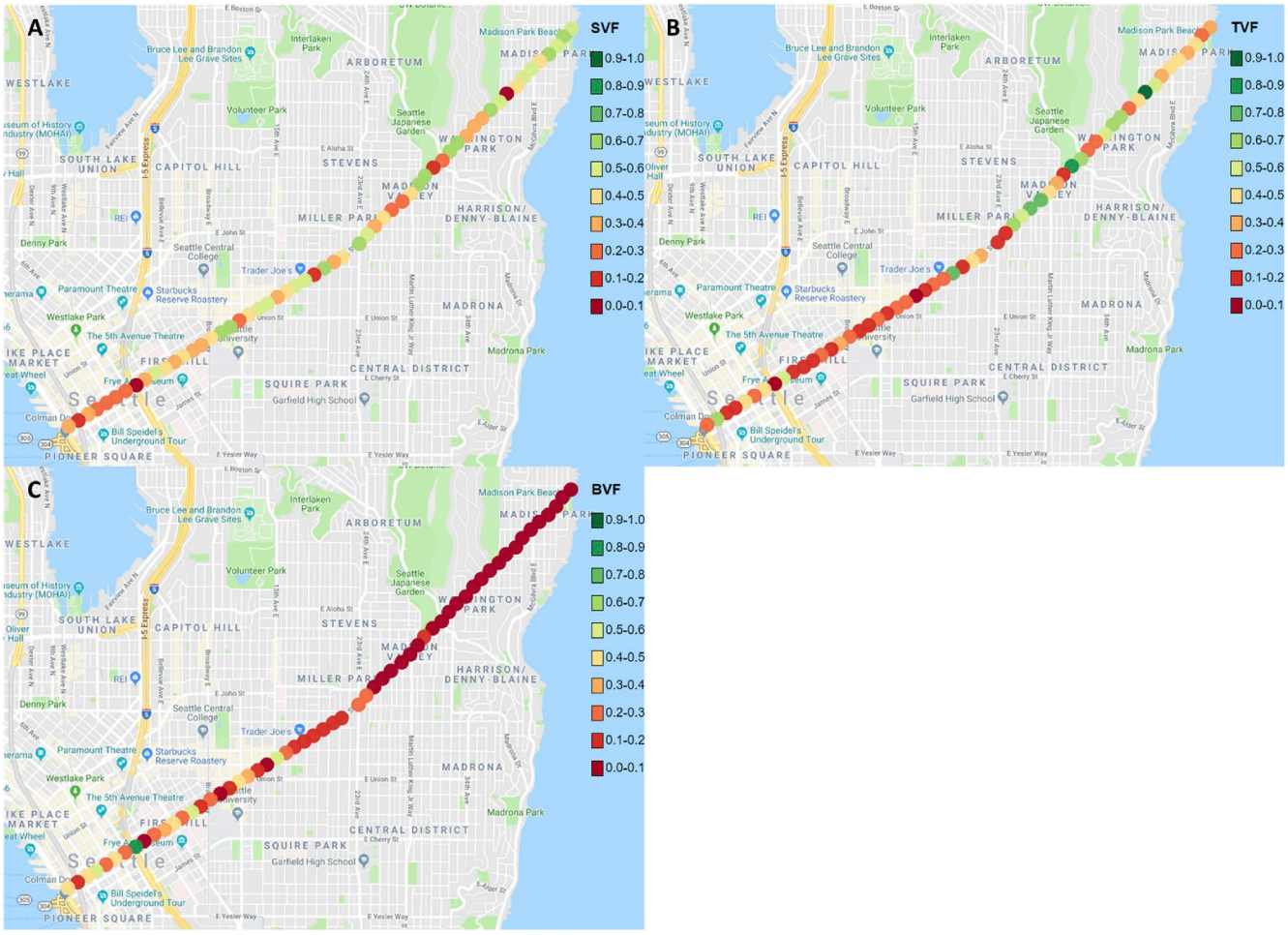

Fig. 6. Mapping of SVF, TVF and BVF along Madison Street, downtown Seattle: (a) SVF; (b) TVF; (c) BVF.

3. Results and discussion

3.1. Software setup

GSV2SVF was designed, developed and delivered as a freely available software with which the user can obtain SVF, TVF and BVF estimates around the world at locations where GSV data are available (see Table 1) and, in the meanwhile, develop a spatial understanding of the built environment being researched by exploring Google Maps and GSV (see Table 1).

When setting up the software, the user will need to provide a Google API key (https://developers.google.com/maps/documentation/javascript/get-api-key) and understand the policies regarding the usage limits. The program can run with or without CUDA. When run in pure CPU mode, it can take up to 30 s to finish processing one GSV panorama task. By comparison, when run on CUDA, the image classification process may achieve an acceleration of up to 10x and can finish one panorama task within a few seconds. A minimum of 10 GB system memory is needed to run in CPU mode, while a minimum of 4 GB video memory is needed to run in CUDA mode. We describe three use scenarios below to help users understand how GSV2SVF can be utilized to effectively assist in analysis of the urban environment.

3.2. Use scenarios

The first scenario is a characterization of the canyon geometry of Market Street, a major thoroughfare transecting downtown San Francisco (Fig. 3). Sample points from Metro Castro Station to the northeastern end near the water front were interactively collected in GSV2SVF. Sample densification was applied so that a new sample point would be interpolated every 50 m. A total 90 GSV samples were obtained over a distance of about 5 km (Fig. 3).

The results were exported for statistical analysis in Excel (Fig. 4). Overall, the SVF on market street ranges from 0.20 to 0.80 with an average of 0.46. From point 70 to 90 which run through the high-rise, high-density core area of downtown San Francisco, the SVF averages only 0.28. By contrast, the section in the low-rise neighborhoods from point 1 to 28 is dominated by high SVF values averaging 0.61. The fisheye views at point 15, 51 and 80 well characterize the large differences in SVF between the low-rise section and the downtown core section, and the transition between them.

In the second scenario, we mapped the SVF distribution in the Manhattan Financial District, which has a high concentration of high-rise buildings (Fig. 5). A road network map in ESRI Shapefile format was acquired from NYS GIS Clearinghouse to provide sample locations at which SVF would be estimated. Fig. 5 shows the vertices at the road center lines at which GSV sample would be queried for. The shapefile was imported into GSV2SVF for batched SVF acquisition and a total 363 SVF estimates were generated and rendered in Google Maps. As expected, the visualization shows that a majority of the SVF values fall in the low range (0.0–0.4, from light to dark red) due to the clustering of high-rise building. Statistical analysis shows an average SVF of only 0.23, which is even lower than that in the San Francisco downtown core.

In the third scenario, we captured the proportions of sky, tree and building coverage along Madison Street, downtown Seattle. Madison Street starts at the Seattle waterfront and extends 6 km northeast

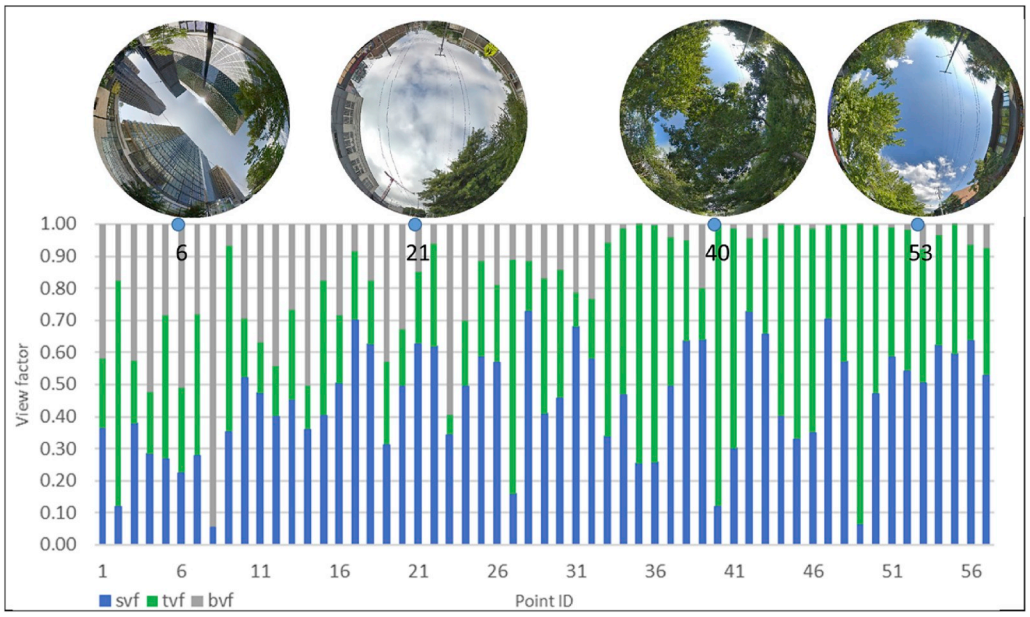

Fig. 7. Statistical analysis of the SVF, TVF and BVF proportions for Madison Street, downtown Seattle.

	A	B	C	D	E	F	G	H
1	id	panoid	date	lat	lon	svf	tvf	bvf
2	0	l-O0SLa5bpEJQ-SD9Q0mVg	2019-06	47.60401	-122.338	0.348859	0.206512	0.402912
3	1	awZ__XZJaMfZMUaIezeVNA	2019-06	47.60447	-122.337	0.116319	0.679132	0.170039
4	2	cUPVnqLy5SZsDfAkkDhv0A	2019-06	47.60499	-122.336	0.367819	0.188784	0.415209
5	3	gtOuJK57RL8Wh3I4DElksw	2019-05	47.60544	-122.334	0.273484	0.18581	0.504894
6	4	820Ol8hDkewSsxX32l-VtQ	2019-05	47.60595	-122.333	0.251233	0.416854	0.265573
7	5	KFnu_tGPMsgEbh2JqD0KOA	2017-07	47.60642	-122.332	0.218323	0.254282	0.495594
8	6	8o-o2tTwL2pbbVGc6ARnMg	2019-05	47.6069	-122.331	0.27605	0.432812	0.276565
9	7	U-nhQGFJK8HBpsAlIhvLLQ	2019-05	47.60738	-122.33	0.051289	0.000389	0.849236
10	8	ANZ4YOEg2y-AxGBY4fbVlw	2018-08	47.60787	-122.329	0.336246	0.549345	0.065019
11	9	hTtE_aA_UuoxUIqoKhLr7Q	2018-08	47.60838	-122.327	0.500914	0.174646	0.282887
12	10	Yy-bCicN3fnoWpextMf6xQ	2018-08	47.60877	-122.326	0.459938	0.154006	0.358015

Fig. 8. Table schema of the GSV2SVF export.

through Downtown Seattle, ending at Lake Washington. GSV samples were collected with a distance interval of 100 m along the street, yielding a total 57 samples. SVF, TVF and BVF estimates were computed and mapped in GSV2SVF (Fig. 6).

The results were then exported in CSV format for further analysis in Excel (Fig. 7). The stacked bar graph (Fig. 7) exemplifies how GSV2SVF can be used to decompose the outdoor built environment into SVF, TVF and BVF. There are large variations in the proportions of sky, tree and building coverage from southwest to northeast Madison Street. The southwest half are typical of high-rise, high-density urban cores, featuring high building coverage and low tree coverage well captured by the fisheye view at point ID 6 (Fig. 7). By contract, the northeast half is characteristic of low-rise residential areas with higher tree coverage and lower building coverage. Unlike in the example of downtown San Francisco (Fig. 4), however, there is not a sharp increase in sky openness from urban core toward the outlying residential areas. The lower sky openness in the residential section of Madison Street can be partially accounted for by the high tree coverage, as evidenced by the fisheye views at point 40 and 53 (Fig. 7).

3.3. Discussion

As shown in the above use scenarios, the results from GSV2SVF not only can be interactively viewed in Google Maps, but also can be exported for external analysis in two formats, Esri Shapefile (.shp) and comma-separated text (.csv) (see Table 1). The Esri Shapefile can be imported in GIS software such as ArcGIS, QGIS and Google Earth for advanced spatial analysis and mapping, for example spatial clustering and spatial regression. The comma-separated text file can be directly opened in Microsoft Excel to make graphs and charts which provide a quick overview of the statistics. To do more advanced statistical analysis, such as multivariate regression with independent socioeconomic variables, one can also load the comma-separated text files in R or Python environment to use with the routines. Both the Esri Shapefile and comma-separated text export formats have the same table schema with eight columns (Fig. 8): (1) “id” is the sequential order; (2) “panoid” is the GSV ID returned from the Google Maps API query; (3) “data” is the date of acquisition of the GSV panorama; (4) “lat” is the WGS84 latitude of the location; (5) “lon” is the WGS84 longitude of the location; (6) “svf” is the sky view factor; (7) “tvf” is the tree view factor; (8) “bvf” is the building view factor.

To some extent, GSV can be used to supplement or even substitute for field hemispherical photography, especially in cases where only point-based SVF estimates are needed. An important use case of SVF has been to assist in research aimed at understanding the relationship between urban geometry and temperature, which varies not only in space, but also with time diurnally and seasonally [3]. With the extensive spatial coverage of GSV, one can easily gather sufficient SVF samples

either within cities or across cities. SVF data obtained from GSV may also be used to improve understanding of the effects of urban geometry on urban environmental indicators such as outdoor thermal comfort, solar radiation, daylighting availability, traffic noise and wind flow, which have implications for urban policy making.

4. Conclusions

We have introduced GSV2SVF, a new tool for easy acquisition of SVF, TVF and BVF information from the massive GSV panorama database. The tool has been demonstrated, through three use scenarios, to be capable of helping users achieve the following objectives:

(1) Effectively collect and visualize SVF, TVF and BVF information in a GIS environment;
(2) Batch processing large numbers of GSV panorama samples to serve purposes such as spatial analysis;
(3) Export results in structured formats for analysis in statistical environments or custom visualization.

In summary, the easy-to-use interface ensures that users without expertise of GIS, machine learning and SVF estimation can intuitively and quickly obtain what they need and can instead focus their energy on analysis. Further work will focus on improving usability and user experience in response to user feedback. Since the tool has only been used in limited test cases, it remains uncertain whether the reported accuracy and applicability will hold under different conditions, and we are hoping that more field measurements and user case reports will help improve the robustness.

Acknowledgments

This research was supported by the National Natural Science Foundation of China Grant 41701469 and Grant 41871323, and the CAS Zhejiang Institute of Advanced Technology Special Fund [grant number ZK-CX-2018-04].

References

[1] T.R. Oke, Canyon geometry and the nocturnal urban heat island: comparison of scale model and field observations, Int. J. Climatol. 1 (1981) 237–254, https://doi.org/10.1002/joc.3370010304.
[2] F.Y. Gong, Z.C. Zeng, F. Zhang, X. Li, E. Ng, L.K. Norford, Mapping sky, tree, and building view factors of street canyons in a high-density urban environment, Build. Environ. 134 (2018) 155–167, https://doi.org/10.1016/j.buildenv.2018.02.042.
[3] J. Unger, Intra-urban relationship between surface geometry and urban heat island: review and new approach, Clim. Res. 27 (2004) 253–264, https://doi.org/10.3354/cr027253.
[4] E. Johansson, Influence of urban geometry on outdoor thermal comfort in a hot dry climate: a study in Fez, Morocco, Build. Environ. 41 (2006) 1326–1338, https://doi.org/10.1016/j.buildenv.2005.05.022.
[5] L.T. Silva, F. Fonseca, D. Rodrigues, A. Campos, Assessing the influence of urban geometry on noise propagation by using the sky view factor, J. Environ. Plan. Manag. 61 (2018) 535–552, https://doi.org/10.1080/09640568.2017.1319804.
[6] F. Yang, F. Qian, S.S.Y. Lau, Urban form and density as indicators for summertime outdoor ventilation potential: a case study on high-rise housing in Shanghai, Build. Environ. 70 (2013) 122–137, https://doi.org/10.1016/j.buildenv.2013.08.019.
[7] L. Chapman, J.E. Thornes, Real-time sky-view factor calculation and approximation, J. Atmos. Ocean. Technol. 21 (5) (2004) 730–741, https://doi.org/10.1175/1520-0426(2004)021<0730:RSFCAA>2.0.CO;2.
[8] C.S.B. Grimmond, S.K. Potter, H.N. Zutter, C. Souch, Rapid methods to estimate sky-view factors applied to urban areas, Int. J. Climatol. 21 (2001) 903–913, https://doi.org/10.1002/joc.659.
[9] P.P. Kastendeuch, A method to estimate sky view factors from digital elevation models, Int. J. Climatol. 33 (2013) 1574–1578, https://doi.org/10.1002/joc.3523.
[10] C. Kidd, L. Chapman, Derivation of sky-view factors from lidar data, Int. J. Remote Sens. 33 (2012) 3640–3652, https://doi.org/10.1080/01431161.2011.635163.
[11] J. Liang, J. Gong, J. Sun, J. Liu, A customizable framework for computing sky view factor from large-scale 3D city models, Energy Build. 149 (2017) 38–44, https://doi.org/10.1016/j.enbuild.2017.05.024.
[12] GSV, Google street view. https://developers.google.com/maps/documentation/streetview/, 2017. (Accessed 27 April 2018).
[13] R. Carrasco-Hernandez, A.R. Smedley, A.R. Webb, Using urban canyon geometries obtained from Google Street View for atmospheric studies: potential applications in the calculation of street level total shortwave irradiances, Energy Build. 86 (2015) 340–348, https://doi.org/10.1016/j.enbuild.2014.10.001.
[14] J. Liang, J. Gong, J. Sun, J. Zhou, W. Li, Y. Li, J. Liu, S. Shen, Automatic sky view factor estimation from street view photographs - a big data approach, Remote Sens. 9 (2017), https://doi.org/10.3390/rs9050411.
[15] L. Zeng, J. Lu, W. Li, Y. Li, A fast approach for large-scale Sky View Factor estimation using street view images, Build. Environ. 135 (2018) 74–84, https://doi.org/10.1016/j.buildenv.2018.03.009.
[16] L. Yin, Z. Wang, Measuring visual enclosure for street walkability: using machine learning algorithms and Google Street View imagery, Appl. Geogr. 76 (2016) 147–153, https://doi.org/10.1016/j.apgeog.2016.09.024.
[17] A. Larkin, P. Hystad, Evaluating street view exposure measures of visible green space for health research, J. Expo. Sci. Environ. Epidemiol. (2018) 1–10, https://doi.org/10.1038/s41370-018-0017-1.
[18] V. Badrinarayanan, A. Kendall, R. Cipolla, Segnet: a deep convolutional encoder-decoder architecture for image segmentation, arXiv preprint arXiv:1511.00561, https://arxiv.org/abs/1511.00561, 2015.