Real-Time 3D Information Visualization on Mobile Devices: Efficient Occlusion Detection for Geospatial Applications

Agata Migalska[a]
Department of Control Systems and Mechatronics, Wrocław University of Science and Technology, Wrocław, Poland

Keywords: Augmented Reality, Geospatial Data Visualization, Occlusion Detection.

Abstract: The rapid advancement of geospatial applications on mobile devices has revolutionized outdoor activities and professions by providing invaluable tools for navigating, exploring, and highlighting geographic features. These applications often face notable challenges, including limited connectivity as well as battery and storage constraints. The objective of this paper is to provide users with a seamless user experience and augmented reality interactions in diverse outdoor scenarios within geospatial applications on mobile devices. Towards this goal, we introduce visibility map of a point that contains distances from the point to the closest obstacles at multiple angular viewpoints. We propose that every point of interest is associated with a such a visibility map, pre-calculated on a digital world model, in order to render annotated data in real-time without relying on time-consuming depth estimation. We specifically address the aspect of displaying 3D virtual information that is registered at different elevations, justified by the importance of elevation information in domains such as earth science, culture science or construction and by a lack of methods that would allow that. Through our study, we demonstrate the real-time rendering capability and accuracy of our proposed method. We present the findings, discuss potential limitations, and suggest future research directions.

1 INTRODUCTION

Geospatial applications on mobile devices have become indispensable tools for outdoor activities and professions, offering navigation assistance, exploration capabilities, and the ability to highlight important geographic features. The integration of Augmented Reality (AR) has further revolutionized the user experience by enriching the perceived world with additional information and expanding the user's understanding of their surroundings (Bowman et al., 2003; Billinghurst et al., 2015). Location-aware AR applications enable the attachment of location-specific virtual content to real-world scenes and allow users to access and manage this information (Höllerer et al., 1999).

In many existing geospatial applications a point of interest is described by geographical latitude and longitude and assumed to be located at the ground level. As an example, let us consider AR tourism applications that suggests walking routes passing by local attractions and provide additional content to the user once they arrive at a certain location (Pica et al., 2018; González-Delgado et al., 2020). An attraction is considered to be at the altitude of Earth at given latitude and longitude. Indeed, in many use cases, such as navigation and routing, an exact object's placement above the ground is of little concern.

What remains unrepresented, however, are the following:

- Points in three-dimensional space that do not correspond to ground-level entities, such as a bird's nest or a wall socket.
- Paths that vary in altitude, such as climbing routes or the course of water and sewage systems within a wall.
- Areas situated above the terrain, such as geological layers within a towering rock wall or hazardous portions of a building's elevation.

Clearly, certain geospatial entities exist above the ground level and necessitate that full and precise elevation information is captured, persisted and presented visually.

In addition to outdoor sports enthusiasts, such as rock climbers, numerous professions can benefit from precise three-dimensional information rendering in geospatial applications. Examples include highlight-

[a] https://orcid.org/0000-0002-0653-026X

Migalska, A.
Real-Time 3D Information Visualization on Mobile Devices: Efficient Occlusion Detection for Geospatial Applications.
DOI: 10.5220/0012398300003660
In *Proceedings of the 19th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2024) - Volume 1: GRAPP, HUCAPP and IVAPP*, pages 277-284
ISBN: 978-989-758-679-8; ISSN: 2184-4321

ing noteworthy parts of buildings and monuments for further studies and for touristic purposes, marking boundaries discovered while working with digital rock outcrop models (Tavani et al., 2019), identifying loose rocks to ensure seamless functioning of a rockfall monitoring system (Cayla and Martin, 2018), or annotating the course of water and electricity systems within walls.

Geospatial applications operating in outdoor environments often contend with challenges. Firstly, they face limited internet connectivity, especially in rural and mountain regions (Fedorov et al., 2016). Furthermore, they may be restricted on battery usage, given the limited availability of power sources in outdoor settings (Stylianidis et al., 2020). Finally, their operation may be opposed by limited storage capacity of a mobile device (Shi et al., 2012).

It is therefore recommended that all calculations and processing are performed offline on a mobile device, to ensure seamless operation regardless of the network signal strength. Furthermore, these applications must be designed with low-power consumption in mind to preserve battery life. Additionally, considering the constrained storage capacity of mobile devices, it is crucial to minimize the volume of stored data. Finally, for immaculate user experience the real-time performance is necessary.

Our objective is thus to determine visibility of virtual objects in a perceived real-world scene in order to overlay these objects on top of a camera preview, as showcased in Figure 1. The primary challenge towards this goal is that the information about visibility and potential occlusions in the scene is not readily available.

The authors of the comprehensive study of research and development in the field of AR (Billinghurst et al., 2015) point towards depth estimation techniques to accurately compose virtual and real world images. However, existing depth-estimation solutions lack real-time performance. For instance, in the Mobile AI 2022 Monocular Depth Estimation challenge (Ignatov et al., 2023), the fastest run time achieved by the AHA HIT Team was 37ms (27 FPS). In applications that involve additional tasks such as calculating node visibility or scene rendering, depth estimation becomes time-consuming and cannot offer seamless user experience. Moreover, depth maps generated by existing algorithms often have low resolutions (e.g., 640x480px), presenting an additional challenge.

We propose to leverage information on angular visibility of annotated points in order to efficiently render a visible subset of them on the user's mobile device's screen. Each point is thus accompanied by a vector of distances from that point to the closest obstacles at specific viewing angles. At runtime, this visibility information is combined with the camera's location and attitude to render only the content that is visible in the camera preview.

We make a two-fold contribution.

- We propose a novel method for visibility calculation in complex scenes that strikes a balance between accuracy and resource efficiency, allowing for real-time performance and seamless user experience.
- We provide a data representation framework for annotating three-dimensional points, polylines, and polygons, facilitating the incorporation of rich spatial information into applications while minimizing storage requirements.

2 RELATED WORK

The problems our research addresses have been observed and scrutinized by other authors as well. (Shi et al., 2012) propose that the calculations for rendering are executed on a remote server in order to ensure real-time interactive 3D mobile graphics. However, this method relies on a stable connection to a remote server and is not well suited for outdoor scenarios. (Jain et al., 2015) recommend that geometric representations of environment are utilized to reduce the search space to device's neighborhood. Our work follows this principle in that we propose that geospatial data are split into regional chunks.

One prominent application of outdoor AR that utilizes information about local world geometry is mountain peak detection, where the goal is to overlay mountain landscapes with peak information. (Karpischek et al., 2009) propose to use so-called view-sheds of mountain peaks; a view-shed is a union of geospatial areas from which a given peak is visible. To determine which peaks are visible from a camera location, view-sheds for all mountains are consulted. (Fedorov et al., 2016) propose that the virtual panorama visible form the viewpoint of a user is estimated from an online Digital Elevation Model (DEM) and from its correspondence to the actual image framed by the camera. Once the correspondence is found, meta-data from DEM (e.g. peak name, altitude, distance) are projected onto the view. DEM data supports only a single z-value for each a x,y location which renders the approach unsuitable for elevation-variable data.

(Westhead et al., 2013) recognize the potential of AR techniques in transforming complex three-dimensional information into a user-friendly format

Figure 1: Real-time 3D Information Visualization on Mobile Devices: two polylines rendered as a camera preview overlay with their full visibility indicated with solid lines and partial visibility with a dashed line.

and introduce the British Geological Survey BGS Map Viewer, which provides a 3D view but is limited to presenting 2D information on a 3D surface.

A different approach to information display is presented in tourism AR applications. The work of (González-Delgado et al., 2020) focuses on the development of georoutes to enhance tourism in natural filming locations in Spain. Users scan QR codes along their route to access relevant information, including textual, graphical, and interactive content. Once again, the elevation information is culled.

Elevation information is not discarded in (Stylianidis et al., 2020), the research that presents LARA, a solution for visualizing 3D underground infrastructure models in real time. The models are retrieved from a 3D GIS-Geodatabase residing on a mobile device. Their work, alike ours, is concerned with 3D real-time rendering of 3D models. However, considering that the main goal for LARA is to provide an "X-ray" viewing of the underground infrastructure, occlusion detection is of no relevance.

3 PROBLEM STATEMENT

Assumptions Regarding the World. The following assumptions are made regarding the world: (1) A mesh model representing the world is created in advance. (2) A domain expert annotates features of interest in the world model in the form of points, polylines and polygons. (3) The world is considered to be static and undergoes minimal changes over time.

Objective. For each annotated point, the objective is to determine whether it is visible to a camera, given the position and orientation of the camera and the geometry of the surrounding world, in order to be properly overlaid on the mobile device's camera preview.

Requirements for the Mobile Application. The mobile application has the following requirements arising from concerns raised in relation to mobile AR for outdoor use: (1) It operates without requiring an internet connection during runtime. (2) The data needed for the algorithm execution is of small volume. (3) The algorithm implemented in the application is resource-efficient to minimize battery usage. (4) The algorithm performs in real-time.

4 PROPOSED METHOD

Figure 2: System diagram.

The proposed solution consists of three modules: (1) a geospatial data pre-processing module that is responsible for the creation of visibility maps, (2) a data serving module that serves spatially-cropped files with annotations as well as their corresponding visibility maps, and (3) a Mobile AR application that performs visibility testing and overlays the camera's

preview with rendered visible annotated points. The proposed system is presented in Figure 2.

4.1 Geospatial Data Pre-Processing Module

We propose to calculate visibility of each annotated point from the world model, that is a 3D mesh model that can be constructed from LIDAR scans, UAV photographs or hand-held camera photographs (Tavani et al., 2019). A domain expert, for instance a geologist or an archaeologist, introduces virtual information to the world model by annotating points, polylines, and polygons in a 3D coordinate system. In order to determine the visibility of an annotated point, we introduce the concept of a point visibility map and propose associating each annotated point with such a map. Point visibility map is a vector of Euclidean distances from a point to the closest world object in each of selected angular directions. To construct a visibility map, we employ a spherical coordinate system and assume a unit sphere. Each point on the sphere can be represented by two parameters: a polar angle θ ($0 \leqslant \theta \leqslant \pi$) that measures the angle from the positive z-axis towards the xy-plane, and an azimuthal angle ϕ ($0 \leqslant \phi \leqslant 2\pi$) that represents the angle from the positive x-axis towards the positive y-axis. We propose to employ equal-angle sphere sampling scheme for its computational advantages but, in principle, any deterministic sampling scheme can be utilized. In the equal-angle sampling scheme, the ranges of both polar and azimuthal angles are divided into N intervals of equal width. We are then able to approximate the sphere with $N \times N$ quadrilateral faces in 3D. An illustration of a sphere approximation in spherical coordinates for $N = 8$ is presented in Figure 3.

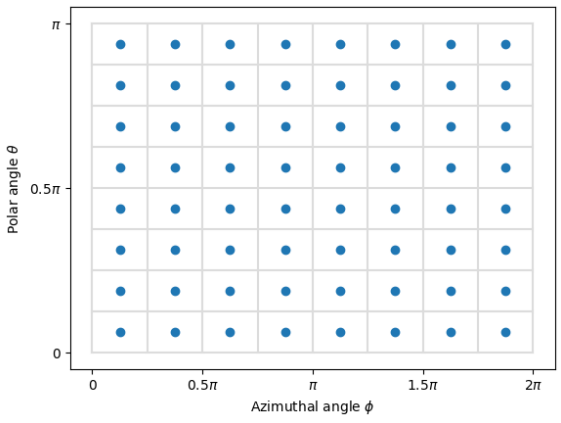

Figure 3: Unit sphere approximation with 8×8 rectangular cells of equally spaced azimuthal and polar angles. Blue dots, in the middle of each cell, represent their centers.

A ray is emitted from the center of a sphere, i.e. an annotated point, in the direction of the center of a cell and continues until it encounters the nearest obstacle in the world model. The distance traveled by the ray is recorded in the visibility map under the index corresponding to the ray's azimuthal and polar angles. The procedure is repeated for all considered directions and yields a visibility map of a point in the world model.

4.2 Data Serving Module

The remote server module is responsible for serving region-specific files containing annotated geometries with their 3D world coordinates as well as visibility maps for each point in annotated geometries.

In outdoor scenarios, the geometry of a scene can be very complex; consider, for instance, rock outcrops or a Romanesque stone bas relief on the portal of a cathedral. Annotated geometries, such as points, polylines, and polygons, are relatively small in size compared to such complex geometry of a scene. Following the approach of existing map applications designed for offline use (e.g. Mapy.cz (http://mapy.cz) or MapsWithMe (http://maps.me)), it is assumed that data for a particular region is downloaded and stored on a mobile device in advance, allowing for offline processing. The data transferred to a mobile device includes only annotated geometries and calculated visibility maps for each point.

4.3 Mobile AR Application

The following are the steps to determine the visibility of a given point on a mobile device. It is assumed that the camera is calibrated in an off-line process and that the camera's intrinsic matrix K is known.

1. Establish the camera's location in the coordinate system of the world model.
2. Prune the search space based on the camera's location and perform a frustum culling test (outside of the proposed method).
3. Obtain the camera's extrinsic matrix M, given by

$$M = \begin{bmatrix} R_{(3\times3)} & T_{(3\times1)} \\ 0_{(1\times3)} & 1_{(1\times1)} \end{bmatrix}_{(4\times4)}, \qquad (1)$$

   where R is the camera rotation matrix and T is the translation column vector.
4. For each point within the user's frustum, perform the visibility test described in Section 4.3.1.
5. If the point is visible, calculate the coordinates (u, v) of the point in the Pixel Coordinate System

using the equation:

$$z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = KM \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}, \quad (2)$$

where z_c represents the z-coordinate of the camera relative to the world origin, and draw the point.

6. For each pair of points in a polyline, draw the edge if both points are visible. If only one of the points is visible, indicate it with appropriate marking.

The procedure can be performed for all points simultaneously using matrix operations.

4.3.1 Point's Visibility Test

We first introduce a general approach applicable to any sphere sampling scheme. We then refine the approach to equal-angle sampling scheme.

Let P be a point for which we want to determine visibility, and let Q be a camera's position. Firstly, compute the vector $\vec{R}$ from point P to camera Q and its norm $d = \|\vec{R}\|$. Then, find the index of the vector that is closest to $\vec{R}$ with respect to polar and azimuthal angles. For an arbitrary sampling scheme, the closest vector is the one that maximizes cosine similarity between $\vec{R}$ and itself,

$$S_C(\vec{D}, \vec{R}) = \frac{\vec{D} \cdot \vec{R}}{\|\vec{D}\|\|\vec{R}\|}. \quad (3)$$

Since $\|\vec{D}\| = 1$ for all vectors in the scheme, it is enough to find the vector that maximizes the dot product between itself and $\vec{R}$. Once the closest vector is found and its corresponding visibility retrieved from the visibility map, the point is announced visible if the interpolated value is no less than the distance d. Otherwise, it is not visible to the camera. The steps are outlined in Algorithm 1.

Data: Point P in 3D coordinates
Camera position Q in 3D coordinates
Visibility map $V \in \mathbb{R}_{(N\times N,1)}$
Visibility vectors $D \in \mathbb{R}^3_{(N\times N,3)}$
Result: true if Q visible from P, false otherwise

$\vec{R} \leftarrow Q - P$;
$d \leftarrow \|\vec{R}\|$;
$j \leftarrow \arg\max_{i \in 1,\ldots,|\vec{D}|} \vec{D[i]} \cdot \vec{R}$;
return $V[j] \geqslant d$;

Algorithm 1: Point Visibility Test.

Provided that equal-angle sphere sampling scheme is selected, polar and azimuthal angles are spaced by $\theta_\Delta = \frac{2\pi}{N}$ and $\phi_\Delta = \frac{\pi}{N}$, respectively. That allows for constant-time calculations of the vector closest to $\vec{R}$. Step-by-step calculations of the proposed Point's Visibility Test for equal-angle sphere sampling scheme are showcased in Algorithm 2.

Data: Point P in 3D coordinates
Camera position Q in 3D coordinates
Visibility map $V \in \mathbb{R}_{(N\times N,1)}$
$\phi_\Delta = \pi/N$ - spacing of Φ
$\theta_\Delta = 2\pi/N$ - spacing of Θ
Result: true if Q visible from P, false otherwise

$\vec{R} \leftarrow Q - P$;
$d \leftarrow \|\vec{R}\|$;
$\phi_R \leftarrow \text{atan2}\left(\vec{R}_y, \vec{R}_x\right)$;
$\theta_R \leftarrow \arccos\left(\vec{R}_z/d\right)$;
$i_\phi \leftarrow \lfloor \phi_R/\phi_\Delta \rfloor$;
$j_\theta \leftarrow \lfloor \theta_R/\theta_\Delta \rfloor$;
return $V[i_\phi \cdot N + j_\theta] \geqslant d$;

Algorithm 2: Point Visibility Test in equal-angle sphere sampling scheme.

Regarding the time complexity of the method, it is important to highlight that the algorithm is executed only once per image, considering all eligible points as a single $M \times 3$ matrix. The code requires a total of three matrix multiplications (two to obtain the points' coordinates in the pixel coordinate system and one to calculate the norm), along with one matrix addition, two invocations of inverse trigonometric functions, and the complexity of the selected interpolation method. Additionally, the algorithm benefits from being executable in parallel. Due to its low time complexity, the method ensures real-time performance.

5 RESULTS

5.1 Implementation

The Geospatial Data Pre-processing Module was implemented in Python and is available in GitHub[1]. This module takes in a 3D mesh model of the world, 3D coordinates of annotated paths and the number of directions N. For each annotated point, the module calculates its $N \times N$ visibility map by calculating the z-buffer from the point in the considered direction. In our implementation we employed a ray-triangle intersection algorithm (Snyder and Barr, 1987; Shirley

[1] https://github.com/OutdoorAR-POC/GeoDataPreProcessing

et al., 2021) that utilizes barycentric coordinates to represent triangles in a world mesh model.

The Mobile AR Application was implemented in Kotlin programming language, the default programming language for applications dedicated for Android devices. A Proof-of-Concept implementation can be found in GitHub [2]. All input files, including visibility maps and geospatial annotations as well as images and camera extrinsic information, were provided as resources to the application.

5.2 Case Study

The case study was concerned with a rock outcrop, representative of complexity observed in such scenarios, whose 3D model was obtained from photographs using the Structure from Motion (SfM) technique (Schonberger and Frahm, 2016; Carrivick et al., 2016) in Meshroom software (Griwodz et al., 2021). The resulting mesh was further cleaned and decimated with Meshlab (Cignoni et al., 2008). To test the visibility of different geometries around the rock outcrop, four multi-point polylines and polygons were designed, as illustrated in Figure 4.

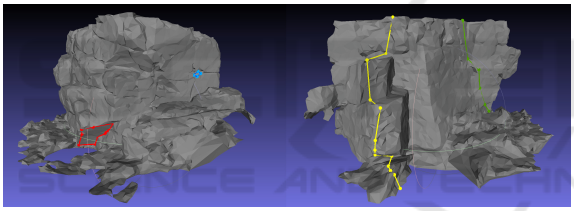

Figure 4: 3D world model with annotated geometries. Left: red and blue polygons. Right: yellow and green polylines.

In order to test the algorithm we utilized the same set of images as for the 3D model creation enriched with data output by the SfM process - camera intrinsic matrix and camera extrinsic matrices for each image. The algorithm was tested on 77 views of the rock formation, each captured from a different viewpoint. Example results for $N = 8$ are presented in Figure 5, showcasing correct outcomes of the method from various viewpoints.

Figure 5: Example results for $N = 8$ from various viewpoints.

5.2.1 Effect of Visibility Map Size

In our study, we experimented with different number of equidistant angles (N), including 2, 4, 8, 16, and 32 angles. The resulting visibility maps had sizes $|V|$ of 4, 16, 64, 256, and 1024, respectively. We present accuracy and F1-score for varying sizes $|V|$ in Table 1. From the results, we find that $N = 8$ provides accurate visibility measurements with the accuracy score of 0.9423 and F1-Score of 0.8793. Increasing N to 16 does not significantly improve the accuracy (0.9649), yet yield a significant boost on the F1-Score that attains 0.9272. Increasing N to 32 slightly improves the scores but the increase in improvement is offset by a fourfold increase in the required disk space.

When evaluated visually in images, the method demonstrated some limitations when additional ob-

[2]https://github.com/OutdoorAR-POC/MobileApp

Table 1: Performance scores of the algorithm depending on the number of viewpoints N.

N	$\|V\|$	Accuracy	F1-Score
2	4	0.7609	0.4401
4	16	0.8793	0.7384
8	64	0.9423	0.8793
16	256	0.9649	0.9272
32	1024	0.9710	0.9402

jects, not included in the digital world model, were present in images.

5.2.2 Effect of Sphere Sampling Scheme

Equal-angle sampling, although computationally attractive, gives points that are not uniformly distributed on a sphere. In the following experiment we utilized the Fibonacci lattice (González, 2010), in which points are almost evenly spaced on the sphere surface, and we constructed visibility maps of the same size as in the previous experiment. From the results, presented in Table 2, it can be concluded that for $|V| \geq 64$ accuracy and F1 scores are comparable to those obtained for the equal-angle sampling scheme. For smaller sizes of $|V|$, the results are significantly better with the Fibonacci lattice, however, they are still not sufficiently high.

Table 2: Performance scores of the algorithm for Fibonacci lattice sphere sampling.

$\|V\|$	Accuracy	F1-Score
4	0.8335	0.6599
16	0.8980	0.7747
64	0.9423	0.8785
256	0.9618	0.9218
1024	0.9733	0.9446

5.2.3 "Small Mesh" Experiment

We conducted a space complexity comparison between a visibility map and a simplified mesh. Intuitively, complex scene geometries with a limited number of annotated geometries favor the use of a visibility map. On the other hand, a simple scene geometry accompanied by a large number of annotated geometries would likely benefit from a simplified mesh. The key question then is: how does the size of a visibility map, i.e. a vector of $|V|$ double precision numbers, compare to that of a mesh file? The mesh file contains a list of 3D vertices, with single precision coordinates, and a list of faces, represented as triples of integer indices of vertices. We assume $8B$ for double precision, $4B$ for single precision, and $4B$ for unsigned integer. Let $|W|$ and $|F|$ be the numbers of vertices and faces in the small mesh, respectively. The space requirement of the visibility map is then $8|V|$, whereas that of the mesh accounts for $12(|F|+|W|)$. That implies that if $|V| = 64$ ($N = 8$), then the total number of faces and vertices in a mesh, $|F|+|W|$, would have to be approximately 43 for the space requirements to equal.

The original mesh was decimated to approximately 1024, 2048 and 4096 faces with Quadratic Edge Collapse Decimation in Meshlab (Cignoni et al., 2008). The obtained results are presented in Table 3. It can be observed that, for the scene and annotations considered in our case study, accuracy and F1 scores comparable to visibility maps of size $|V| = 64$ are achieved no sooner than when $|F|+|W| = 2048+4095$.

Table 3: Performance scores of direct ray tracing on simplified meshes.

$\|W\|$	$\|F\|$	Accuracy	F1-Score
513	1023	0.8843	0.7335
1025	2047	0.9118	0.7979
2048	4095	0.9396	0.8674

6 DISCUSSION AND FUTURE WORK

The proposed method offers several noteworthy advantages. It accurately renders annotated geometries and it adheres to all four stated requirements, including low time and space complexities. The method, however, does have some limitations that need to be considered. Firstly, it relies on a well-mapped world where all objects are accurately scanned and included in the digital model. If there exist discrepancies between the real world and a model then annotated points may be incorrectly classified as visible or not visible. Secondly, the method requires that the mesh of the world is created in order to calculate visibility maps. While in multiple professional scenarios such models are nevertheless created for analytic purposes, this requirement could pose a computational challenge.

One possible direction of future research is to combine the proposed method with coarse depth maps. This integration could help account for vegetation and objects that are not registered in the world's model. Additionally, alternative interpolation methods, such as bi-linear interpolation, could be explored in search for more accurate results compared to the nearest neighbor interpolation used in our study.

7 CONCLUSION

In this paper, we address the challenge of real-time 3D information visualization on mobile devices under constraints of limited connectivity, battery capacity, storage limitations, and real-time performance. We propose associating each annotated point with a visibility map that captures visibility from that point in multiple angular directions. By leveraging the camera's position and orientation and identifying the nearest viewing angle within each point's visibility map, we efficiently determine the point's visibility in the scene. Obtained experimental results demonstrate that the proposed method enables real-time rendering of annotated data and provides an accurate alternative to techniques relying on scene depth estimation.

REFERENCES

Billinghurst, M., Clark, A., and Lee, G. (2015). A survey of augmented reality. *Foundations and Trends® in Human–Computer Interaction*, 8(2-3):73–272.

Bowman, D. A., North, C., Chen, J., Polys, N. F., Pyla, P. S., and Yilmaz, U. (2003). Information-rich virtual environments: theory, tools, and research agenda. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 81–90.

Carrivick, J. L., Smith, M. W., and Quincey, D. J. (2016). *Structure from Motion in the Geosciences*. John Wiley & Sons.

Cayla, N. and Martin, S. (2018). Chapter 16 - digital geovisualisation technologies applied to geoheritage management. In Reynard, E. and Brilha, J., editors, *Geoheritage*, pages 289–303. Elsevier.

Cignoni, P., Callieri, M., Corsini, M., Dellepiane, M., Ganovelli, F., and Ranzuglia, G. (2008). MeshLab: an Open-Source Mesh Processing Tool. In Scarano, V., Chiara, R. D., and Erra, U., editors, *Eurographics Italian Chapter Conference*. The Eurographics Association.

Fedorov, R., Frajberg, D., and Fraternali, P. (2016). A framework for outdoor mobile augmented reality and its application to mountain peak detection. In *Augmented Reality, Virtual Reality, and Computer Graphics: Third International Conference, AVR 2016, Lecce, Italy, June 15-18, 2016. Proceedings, Part I 3*, pages 281–301. Springer.

González, Á. (2010). Measurement of areas on a sphere using fibonacci and latitude–longitude lattices. *Mathematical Geosciences*, 42:49–64.

González-Delgado, J. Á., Martínez-Graña, A., Holgado, M., Gonzalo, J. C., and Legoinha, P. (2020). Augmented reality as a tool for promoting the tourist value of the geological heritage around natural filming locations: A case study in "Sad Hill"(The Good, the Bad and the Ugly Movie, Burgos, Spain). *Geoheritage*, 12:1–11.

Griwodz, C., Gasparini, S., Calvet, L., Gurdjos, P., Castan, F., Maujean, B., De Lillo, G., and Lanthony, Y. (2021). Alicevision Meshroom: An open-source 3D reconstruction pipeline. In *Proc. 12th ACM Multimed. Syst. Conf. - MMSys '21*. ACM Press.

Höllerer, T., Feiner, S., Terauchi, T., Rashid, G., and Hallaway, D. (1999). Exploring MARS: developing indoor and outdoor user interfaces to a mobile augmented reality system. *Computers & Graphics*, 23(6):779–785.

Ignatov, A., Malivenko, G., Timofte, R., Treszczotko, L., Chang, X., Ksiazek, P., Lopuszynski, M., Pioro, M., Rudnicki, R., Smyl, M., et al. (2023). Efficient single-image depth estimation on mobile devices, mobile AI & AIM 2022 challenge: report. In *Computer Vision–ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part III*, pages 71–91. Springer.

Jain, P., Manweiler, J., and Roy Choudhury, R. (2015). Overlay: Practical mobile augmented reality. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, pages 331–344.

Karpischek, S., Marforio, C., Godenzi, M., Heuel, S., and Michahelles, F. (2009). Swisspeaks–mobile augmented reality to identify mountains. In *Workshop at the International Symposium on Mixed and Augmented Reality 2009 (ISMAR 2009)*.

Pica, A., Reynard, E., Grangier, L., Kaiser, C., Ghiraldi, L., Perotti, L., and Del Monte, M. (2018). Geoguides, urban geotourism offer powered by mobile application technology. *Geoheritage*, 10:311–326.

Schonberger, J. L. and Frahm, J.-M. (2016). Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113.

Shi, S., Nahrstedt, K., and Campbell, R. (2012). A real-time remote rendering system for interactive mobile graphics. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 8(3s):1–20.

Shirley, P., Ashikhmin, M., and Marschner, S. (2021). *Fundamentals of computer graphics*. AK Peters/CRC Press, 5th ed. edition.

Snyder, J. M. and Barr, A. H. (1987). Ray tracing complex models containing surface tessellations. *ACM SIGGRAPH Computer Graphics*, 21(4):119–128.

Stylianidis, E., Valari, E., Pagani, A., Carrillo, I., Kounoudes, A., Michail, K., and Smagas, K. (2020). Augmented reality geovisualisation for underground utilities. *PFG*, 88:173–185.

Tavani, S., Corradetti, A., Granado, P., Snidero, M., Seers, T. D., and Mazzoli, S. (2019). Smartphone: An alternative to ground control points for orienting virtual outcrop models and assessing their quality. *Geosphere*, 15(6):2043–2052.

Westhead, R., Smith, M., Shelley, W., Pedley, R., Ford, J., and Napier, B. (2013). Mobile spatial mapping and augmented reality applications for environmental geoscience. *Journal of Internet Technology and Secured Transactions*, 2(1-4):185–190.