

AI-Accelerated Viewshed Computation for High-Resolution Elevation Models

Cédric Schwencke, Dominik Stütz and Dimitri Bulatov^a
Fraunhofer IOSB Ettlingen, Gutleuthausstrasse 1, 76275 Ettlingen, Germany

Keywords: Viewshed Computation, Visibility Analysis, Neural Networks, Digital Terrain Models, Digital Surface Models, Supervised Learning, GIS, Environmental Monitoring.

Abstract: Viewshed computation, essential for visibility analysis in GIS applications, involves determining visible areas from a given point using the digital terrain model (DTM) and digital surface model (DSM). The traditional methods, though accurate, can be computationally intensive, especially with increasing search distances and high-resolution elevation DSMs. This paper introduces a novel approach leveraging neural networks to estimate the farthest visible point (FVP). At this point the viewshed computation could be aborted, which significantly reducing computation time without compromising accuracy. The proposed method employs a fully connected neural network trained on varied terrain profiles, achieving over 99% accuracy in visibility predictions while reducing the required amount of computations by more than 90%. This approach demonstrates substantial performance gains, making it suitable for applications requiring fast visibility analysis.

1 MOTIVATION

The computation of viewshed in outdoor point clouds, which are often sampled into more manageable DSMs, has numerous applications. For instance, according to (Pan et al., 2020), viewshed analysis can distinguish visible and invisible areas from a DSM in mountainous areas and determine the suitability of regions for forest tourism practices (Lee et al., 2019; Hognogi et al., 2022). Viewshed computation is also applicable in aquatic and marine systems (Arnon et al., 2023; Qiang et al., 2019). Additionally, it can predict the velocity expansion of sound waves or potentially dangerous gases, aiding in decision-making processes.

Despite the efficiency of viewshed analysis for a single viewpoint, the processing time becomes significant when calculating multiple viewpoints, known as total-viewshed computation. Moreover, the computation time increases dramatically with a growing maximum view distance or higher resolution. For many applications, a large or nearly infinite view distance is necessary.

In this paper, we focus on improving a total-viewshed computation algorithm based on line of sight (LoS). LoSs are used for simplification. To speed up the viewshed computation on each LoS, we introduce a novel approach. We estimate the FVP on

a LoS to abort the visibility computation earlier. An example of a LoS in profile view is shown in Figure 1.

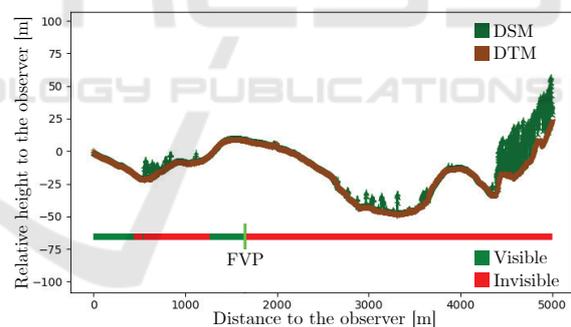


Figure 1: This graph illustrates the visibility of a target (1.9m high), placed on the DTM, from the observer's perspective along the LoS. The brown line represents the DTM, while the green areas indicate the DSM. The visibility is depicted through a straight line at the bottom: red indicates invisible segments, and green indicates visible segments. The vertical bright green line marks the FVP that remains visible from the observer's location.

Defining a hard cutoff criterion, such as roughness, is challenging due to the variety of topographies. Although the computation of a single LoS requires few operations, it is challenging to undercut this for the cutoff criterion. Humans can intuitively and quickly assess the necessary extent of viewshed computations when examining terrain profiles, inspiring the use of a neural network to estimate the FVP.

^a <https://orcid.org/0000-0002-0560-2591>

Parallelizing the computation using multiple LoSs makes fully connected networks promising for performance enhancement. The output of the neural network may either exceed or fall short of the true value of the FVP. In the former case, unnecessary computation resources are used, but the result remains correct. In the latter case, computation resources are saved at the risk of errors.

As illustrated in Figure 2, for 90% of the LoS, the furthest visible point is within a distance of 1000m towards the observer. Therefore, it is often unnecessary to calculate the entire LoS for a correct viewshed result.

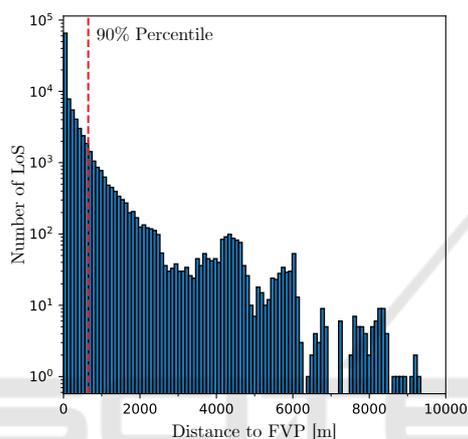


Figure 2: Histogram over distances to the FVP on a LoS, visible by an observer. The data samples were taken across all datasets described in Section 4.

In Section 2, we provide a comprehensive overview of related work to accelerate viewshed computation. Our proposed method is introduced in Section 3. We describe the datasets used, present the results, and draw the main conclusions in Sections 4, 5, and 6, respectively.

2 RELATED WORK

This section provides an overview of notable methods for total Viewshed computation and explores innovative strategies for enhancing computational efficiency.

Within at least a quarter-of-century, the field of Viewshed computation has been explored intensively regarding increased efficiency. To begin with, (Stewart, 1998) proposed a fast algorithm for computing the horizon at all sample points in a digital elevation map, demonstrating improved accuracy and speed, particularly for terrains with over 100000 sample points, making it suitable for visibility culling and rendering. Building on this, (Tabik et al., 2013) in-

roduced a novel algorithm that computes the total-viewshed for all points in a terrain simultaneously, significantly outperforming traditional GIS viewshed tools by combining visible heights and low-cost computations. (Tabik et al., 2015)’s work emphasized efficiency through effective data structures and high data reuse across multiple viewpoints. Similarly, (Qarah, 2020) developed a parallel GPU algorithm utilizing a radial-sweep approach, achieving remarkable speed improvements over traditional CPU-based methods, while (Pan et al., 2020) recommended the Matryoshka doll algorithm to enhance computation efficiency in viewshed analysis. Based on (Tabik et al., 2015), (Sanchez-Fernandez et al., 2021) further contributed to the field with their skewed digital elevation model (sDEM), which improved memory access locality in rotational sweep algorithms, demonstrating substantial speedups compared to conventional GIS software. The application of visibility-based path planning heuristics by (Sanchez-Fernandez et al., 2022) showcased the potential of total-viewshed computation for maximizing visual coverage in UAV monitoring tasks. (Heyns and Van Vuuren, 2013) focused on facility placement optimization based on viewshed visibility percentages, proposing methods to estimate viewshed efficiently with minimal accuracy loss. (Wang et al., 2023) introduced an innovative algorithm that computes viewshed using reference planes, resulting in significant reductions in computing time compared to traditional LoS based methods. Recently, (Arnon et al., 2023) presented ViewShedR, an open-source tool designed for cumulative, subtractive, and elevated LoS analyses, facilitating accessibility for scholars in environmental and ecological studies. (Parent and Lei-Parent, 2023) developed a 2D viewshed approach that utilizes land cover to estimate visibility extents, proving effective for large study areas where trees and buildings serve as primary obstacles. Lastly, (Zhang et al., 2021) introduced a Spark-based parallel computing approach for the XDraw algorithm, significantly enhancing efficiency and scalability in Viewshed analysis for high-resolution raster DSM.

Collectively, these contributions highlight the ongoing evolution and innovation in viewshed computation. We continue this by pursuing a novel approach that significantly reduces the initial computation effort by intelligently estimating the FVP for the first time.

3 METHODOLOGY

This section outlines the methodology used to speed up the viewshed computation using neural networks. We begin with an overview of the conventional method, followed by the introduction of an AI-based approach. Detailed descriptions of the network architecture, training data, and the loss function are provided subsequently.

3.1 Preliminaries

A viewshed computation method described by (Häufel et al., 2023) use LoS. Key terms of this method are described in Figure 3. This method calculates the viewshed of a target (t) from an observer (o) as:

$$z_o := \text{DTM}(o) + h_o \quad (1)$$

$$z_t := \text{DTM}(t) + h_t \quad (2)$$

where h_o is the observer and h_t the target height. To calculate the viewshed of a target on the LoS, the maximum slope variable S^* is initialized to $-\infty$. On the LoS, the distance from the target towards the observer $\rho = [1, \dots, D]$ is defined, where ρ_i is the distance to the target and D the maximum visibility distance. The current LoS slope S is computed for every target position as:

$$S := \max \left(\frac{z_t - z_o}{\rho_t}, \frac{\text{DSM}(t) - z_o}{\rho_t} \right) \quad (3)$$

The boolean visibility on the LoS v_ρ and the maximum slope S^* are updated according to:

$$\text{If } S > S^* : v_\rho := 1 \text{ and } S^* := S \text{ else } v_\rho := 0. \quad (4)$$

3.2 Viewshed-AI Architecture

Human intuition allows a quick estimation of the FVP in terrain profiles, which is in most cases much smaller than D . Inspired by this, we propose using a neural network to estimate the FVP and relinquish the viewshed computation on the LoS much further afar, whereby ‘‘much’’ means a safety margin to be imposed is recommendable, as we will see later. In either way, doing so would massively reduce the computation expense while maintaining accuracy, especially when parallelized over multiple LoSs.

The procedure is schematically explained in Figure 4 with hypothetical numerical in- and outputs. We employ a fully connected neural network, consisting of an input layer, hidden layers with ReLU activation functions, and dropout layers. The input layer gets the concatenated profiles of the normalized digital surface model (nDSM) and the DTM. Additionally, the

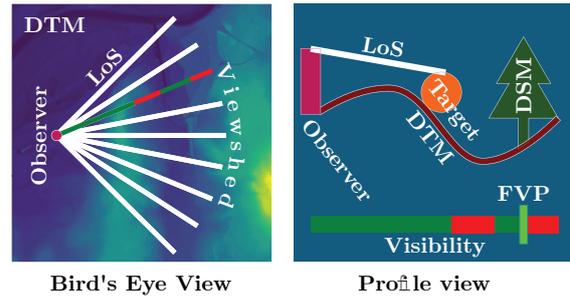


Figure 3: Key terms related to viewshed computation using LoS. It depicts the observer’s location, the target being evaluated, and the LoS that indicate visibility. The DSM and DTM are essential for determining visibility. The viewshed represents the area visible from the observer’s point. The figure includes both a bird’s eye view, showcasing the terrain and visibility relationships from above, and a profile view, illustrating elevation changes and sight lines. Green and Red colors along the LoS marks areas where the orange target is (in-) visible from the observer.

target height value is appended. The network’s final layer outputs the predicted FVP.

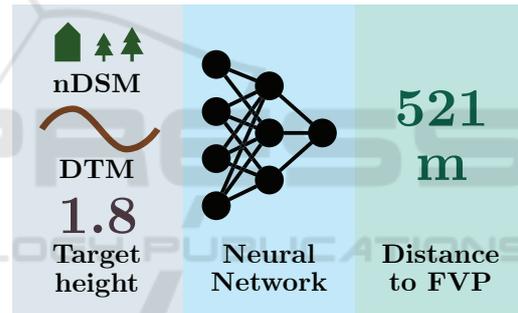


Figure 4: The process for estimating the FVP. The input data in the gray box consists of a nDSM, a DTM adjusted for observer height, and the target height. A neural network processes this information to determine the FVP. The numeric values shown are illustrative examples.

The number of neurons in hidden layers and dropout rates can be adjusted to the problem. This network reliably learns relationships within a vector and has a very short evaluation time. The architecture is chosen to balance evaluation time and accuracy, initially focusing on feed-forward networks.

3.3 Training Data

Training data is created by selecting random strips of length l from our DTMs and the corresponding DSMs. Each strip includes an observer and target height chosen in range 0.5 m to 4.0 m.

The network’s input layer receives a vector containing the DTM, nDSM, and target height. The input

vector is structured as.

$$[\text{DTM} \mid \text{nDSM} \mid \text{Target Height}] .$$

with an input dimension of $D = 2l + 1$. Hereby, the nDSM denotes normalized DSM, which is the elevation over the ground, or the difference between DSM and DTM. Furthermore, the DTM and nDSM are subtracted towards the absolute observer height to make their values relative towards him.

The training dataset contains the input vector, visibility at each point, and the calculated FVP, see Figure 3. The FVP is generated by determining the slope between z_o and z_s and comparing it with the maximum slope from previous points.

3.4 Loss Function

The loss function influences the network's training by weighting errors.

Underestimating ρ^* should be penalized much more than overestimating this variable. The latter would result in the necessity of making more computations during viewshed computation, which is a smaller threat than the former, that is, overlooking potentially visible points, because it may signify a particular value or alarm. Our function should be somewhat skewed towards positive differences between actual and predicted values of ρ^* .

Another way to encourage rather farther visibility is to impose scale invariance and proportional error measurement. To this end, we based our function on the Huber loss, which is quadratic for small and linear for large deviations between actual and predicted values. Our modified Huber loss $L : \mathbb{N} \times \mathbb{N} \times \{0, 1\}^l \rightarrow \mathbb{R}_{\geq 0}$ with user-specified parameters β , γ , and δ is defined by Equation (5).

$$L(x, \rho^*, \mathbf{v}) = \begin{cases} \delta(L_\rho - \frac{1}{2}\delta) & \text{if } \delta < L_\rho, \\ \frac{1}{2}L_\rho^2 & \text{if } 0 \leq L_\rho \leq \delta, \\ \frac{1}{2}\beta L_\rho^2 + \gamma L_\mathbf{v} & \text{if } -\delta \leq -L_\rho < 0, \\ \delta\beta(-L_\rho - \frac{1}{2}\delta) + \gamma L_\mathbf{v} & \text{otherwise,} \end{cases} \quad (5)$$

where $L_\rho = x - \rho^*$ and $L_\mathbf{v} = \sum_{i=x^++1}^{\rho^*} v_i$.

In this loss function, x is the FVP generated by the network, ρ^* is the actual FVP, \mathbf{v} indicates visibility at each point (1 is visible, 0 is invisible), and $x^+ = \max(x, 0)$. The vector \mathbf{v} is hereafter called truth vector. The parameter β weights the loss for $x - \rho^* < 0$,

δ controls the interval for quadratic deviations, and γ adds a penalty for unconsidered visibility.

The choice of β , δ , and γ influences the balance between computation time and error minimization. An optimal set of parameters minimizes early FVP errors while maximizing computation efficiency. We will turn to this issue during the ablation study. However, in short, δ is the trade-off between linear and quadratic error, typical for the Huber function, β is the skewing parameter for negative $x - \rho^*$, and γ allows taking into account the number of points for which visibility was incorrectly predicted.

4 DATASETS

The proposed method can be trained and tested for any dataset for which DSM and DTM are available. Of course, the more diverse datasets exist, the more generalized the output model is. We considered three datasets from our repository, which cover a range from very flat to hilly terrain. They originate from three European regions: Karsava in Eastern Latvia, Altmühltal is in Bavaria in Southern Germany, and Wetter is in the province of Hesse in Central Germany. Latvian dataset is quite flat and has quite many forest regions. Bavaria is a bit hilly, and Hesse even more so. The dataset from Bavaria, also referred to as Greding, was used by other scientific publications, such as (Kuester et al., 2021). Key parameters for describing the terrain of the three datasets can be seen in Figure 5. The resulting dataset contains an equal amount of every region, which can ensure a sufficient diversity in the data.

The considered height profiles start at arbitrary points in the dataset and extend in arbitrary directions, each measuring a length of 10000m. Through sampling in different directions, the variety can be further increased. Some 10% of them were used for training and the rest validating the algorithm.

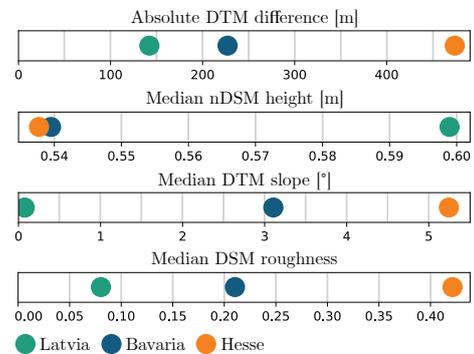


Figure 5: Parameters of considered datasets.

5 RESULTS

To assess the performance of the network, we straightforwardly measure the absolute error between the predicted and actual FVP. Besides, we want to track the relative performance accuracy ε of the truth vector \mathbf{v} , using the same notation as in (5):

$$\varepsilon = 1 - \frac{1}{l} \sum_{i=x^++1}^{p^*} v_i, \quad (6)$$

with l describing the length of the LoS. The reason to take it into consideration lies in the later application. The higher number of visible points after the generated stop leads to a higher error. If the p^* leads to an early stop with only a few visible points much further along the LoS, the saved time could be more beneficial.

With these considerations in mind, we will discuss network designs, qualitative and quantitative findings, as well as a comparison with the conventional Viewshed Algorithm, optionally accelerated with a max-slope criterion, in the remainder of this section.

5.1 Ablation Study

In this section, we will consider a few network designs and provide experiments with different network parameters. We chose four networks with different amounts of layers and neurons because we aim for a fast inference method, whilst maintaining a high accuracy.

Table 1: Number of neurons in reviewed network architectures with parameters $(\beta, \gamma, \delta) = (2, 100, 500)$.

Layer-Number \rightarrow				
Network-ID \downarrow	1	2	3	4
1	2048	1024	512	—
2	1024	512	256	—
3	2048	1024	512	256
4	1024	512	256	128

As can be seen in Figure 6a, all four networks produce similar results; networks with three layers overshoot more than those with four. Otherwise, having 2048 neurons in the first hidden layer results in a lower deviation of absolute errors. This behavior is desired; nevertheless, a higher number of neurons and layers leads to a longer evaluation time. A suitable network has to be chosen according to the specifics of the application. While experimenting with different parameters, some tendencies could be derived. Increasing the value of β results in a higher deviation of the errors, while shifting the mean upwards. This

behavior should be expected, as the network tends to overestimate the FVP value due to the composition of the loss function. Adjusting the parameters γ or δ does not change the outcome as much as expected. A correlation between those parameters needs to be further investigated. Overall, it is not possible to come to a final conclusion because the mutual dependencies of the parameters have proven to be relatively complex.

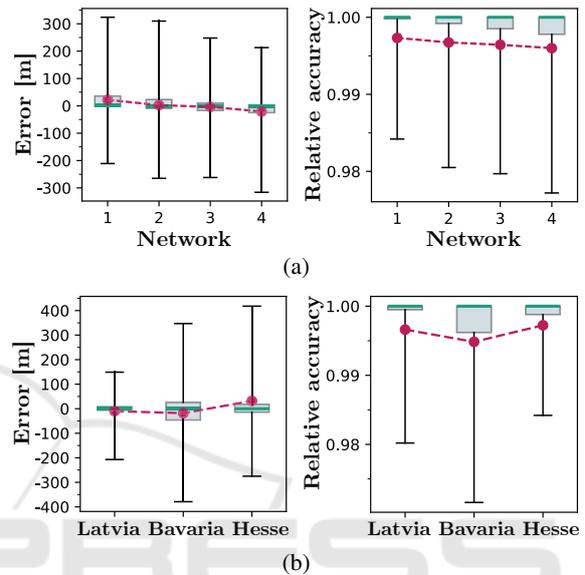


Figure 6: Box plots of errors and accuracy ε for the networks mentioned in Table 1 (a) and for the different regions (b). The black lines indicate the quantile deviation between the 5% and the 95% quantile. The gray box depicts the interquartile range, and the green line shows the median. The red dots represent the empirical mean.

5.2 Qualitative Evaluation

The results discussed from now on were generated with network-ID 3 from Table 1. This network represents a suitable choice according to the hyperparameter optimization framework Optuna (Akiba et al., 2019). Afterward, the network design has been slightly modified based on empirical observations. In Figure 7, we can observe six examples (at the top) in which the network predicts the FVP with an extremely high accuracy, while three examples on the bottom exhibit different kinds of deviations. In the successful examples of the FVP retrieval, the second row shows a few challenging situations. It can be observed that despite some FVPs being far away or not on top of a hill, the network predicts the FVP with a high accuracy. The least pleasant for our application errors of too early FVPs happen occasionally close to the forests, although there are some vertical slopes further away. The most FVP errors of our applica-

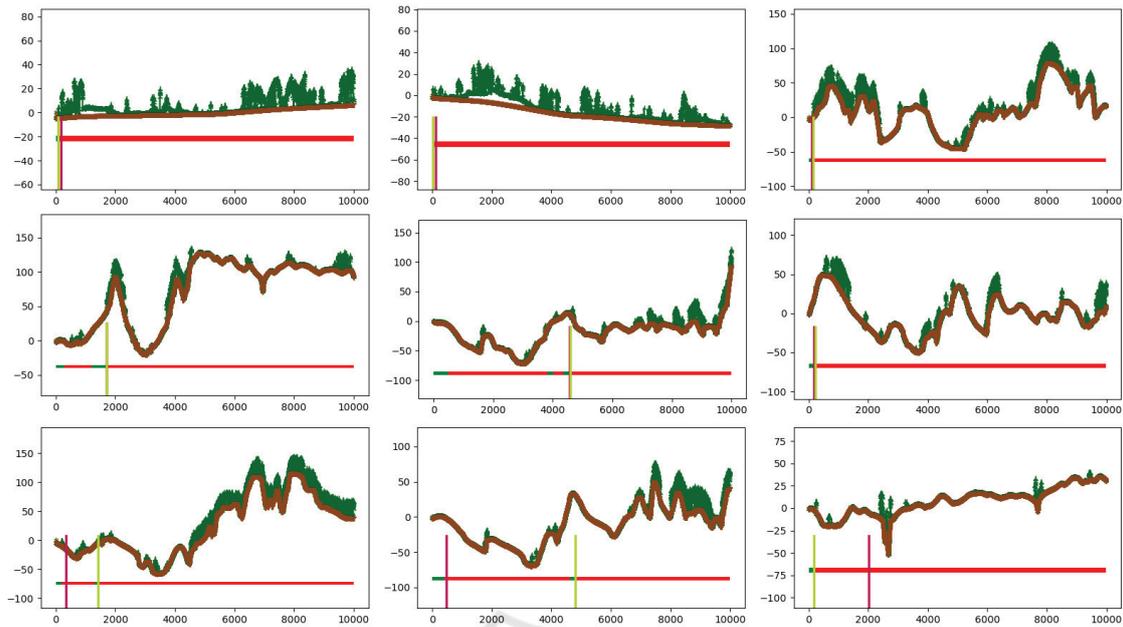


Figure 7: Examples of six correctly predicted FVPs (accuracy ϵ over 99%, top two rows) and two incorrectly predicted FVPs (bottom rows, accuracy ϵ below 90%). Vertical green and purple bars indicate the actual and predicted points. The elevation profiles in the DSM and DTM are shown in green and brown. Observed and non-observed points are represented by green and red fragments, respectively, in the horizontal line below.

tion occur occasionally near forests, although there are some vertical slopes further away. Fortunately, in the majority of these cases, only a few actually visible points are missed.

5.3 Quantitative Assessment

In this section, we discuss the accuracy produced by the aforementioned network. From Figure 2 it becomes clear that in the elevation profiles along the 10000m, some 90% of the deviations constitute less than 250m. The accuracy of the truth vector exceeds 97% for 95% of the test data. Our network is able to estimate the approximate position of the FVP. It is also not critical to achieve a slightly too early FVP because a safety margin of around 250m, barely noticeable in the application of the method in Section 3.1, can be added. The accuracy of our network is $\pm 250m$, which might seem modest but represents only 2% of the total visibility range of 10000m. By adding this 2% as a safety margin, we achieve no errors in 90% of the cases. Unfortunately, the errors are unevenly distributed in three different regions, as becomes evident from Figure 6b. In Latvia, the results are much more accurate than in Bavaria. This indicates that the network produces better outputs for flatter regions. Performance of the method in very mountainous regions is yet to be tested.

5.4 Comparison with Max-Slope Criterion in Terms of Time and Accuracy

Table 2 represents the percentage of needed computations compared to an implementation with no abort condition, along with the achieved accuracy. For comparison, two other abort criteria are displayed. Those abort the computation if the regarded view angle between the observer height and a point on the DSM exceeds a certain threshold. The network is displayed with different safety margins.

Table 2: Percentage of computations needed using different FVP criteria, compared to no FVP criterion for Latvia (L), Bavaria (B), and Hesse (H). We denote in the left-most column network 3 from Table 1 by N3.

FVP-crit. ↓ Region →	Computations [%]			mean Accuracy [%]		
	L	B	H	L	B	H
1°	7.82	19.53	11.48	99.99	99.82	99.80
15°	38.65	53.98	46.68	99.99	99.99	99.99
N3	1.02	4.00	5.30	99.69	99.51	99.75
N3[+250]	3.42	6.40	7.7	99.90	99.82	99.91
N3[+500]	5.92	8.90	10.20	99.96	99.91	99.95

Using the network-generated FVP for aborting reduces the amount of computations quite significantly. Implementing a simple criterion like an angle-based

abort condition is a much simpler strategy and can also reduce the number of computations. Although the angle-based criteria with a 15° condition lead to a near-perfect accuracy, it may fail depending on the terrain regarded. We can see that Karsava in Latvia has so many trees and so many quite short values for FVP, that the simple strategy to abort as soon as 1° slope is achieved is very difficult to beat in terms of accuracy. A high deviation of absolute errors, seen in Figure 6b, leads to a significantly higher percentage of needed computations. But also for other regions, the 1° max slope criterion can only be outperformed only by means of a safety margin, and at a cost of necessary computations. However, in Table 2, we displayed the mean values, which are susceptible for the outliers. Taking into account the 5% quantile, the 1° criterion achieves 0.99 for Latvia, but 0.98 for Bavaria and Hesse. The network with a safety margin of 250 achieves 1 for Latvia and Hesse and 0.99 for Bavaria. This is why the apparently good performance of the simple max slope criterion is misleading.

Overall, the computing time is hardware- and implementation-dependent. With an Intel® Xeon® Gold 6154 CPU, the time needed to infer the FVP for one LoS is approximately 0.03ms, when the input-data contains 10000 LoSs.

6 CONCLUSION

In this paper, we presented a novel approach to accelerate the viewshed computation. For each LoS, the FVP is estimated, after which the viewshed computation is aborted. Before the FVP the viewshed computation is exactly as before. Ideally, this results in no loss of accuracy compared to traditional methods. Since the distance towards the FVP is significantly smaller than the desired maximum visibility distance, in most cases substantial speed gains can be achieved.

We achieve an accuracy of $\pm 250\text{m}$. Our test data shows that we could save more than 90% of the computations, whilst maintaining a high accuracy when dealing with large viewshed distances or high resolution datasets. This efficiency cannot be achieved by traditional LoS algorithms and even with the simple 1° criterion, we achieved much less points to be tested with comparable accuracy.

As this is our initial study with this approach, we are confident that further improvements in accuracy and speed are achievable. There are many possibilities within network architecture, making it unlikely that we have found the optimal solution in the first attempt.

In future studies, we aim to enhance performance

through skip connections or parallel network architectures. Additionally, we plan to investigate the aptitude of convolutional neural networks. Deeper networks are not currently planned due to performance considerations.

ACKNOWLEDGEMENTS

Freely available AI-based text generation techniques helped summarize Section 2. The sources were selected by the authors who also carefully double-checked the resulting text. We also thank the reviewers for their insightful comments.

REFERENCES

- Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.
- Arnon, E., Uzan, A., Handel, M., Cain, S., Toledo, S., and Spiegel, O. (2023). Viewshedr: a new open-source tool for cumulative, subtractive and elevated line-of-sight analysis. *Royal Society Open Science*, 10(6):221333.
- Häufel, G., Bulatov, D., Emter, T., Frese, C., Kottler, B., Peterleit, J., Schmidt, A., Solbrig, P., Stütz, D., and Wernerus, P. (2023). Ghost: Getting invisibly from position A to position B. In *Emerging Imaging and Sensing Technologies for Security and Defence VIII*, volume 12740, pages 32–43. SPIE.
- Heyns, A. and Van Vuuren, J. (2013). Terrain visibility-dependent facility location through fast dynamic step-distance viewshed estimation within a raster environment. In *Proceedings of the Conference of the Operations Research Society of South Africa*, pages 112–121. Operations Research Society of South Africa.
- Hognogi, G.-G., Pop, A.-M., Mălăescu, S., and Nistor, M.-M. (2022). Increasing territorial planning activities through viewshed analysis. *Geocarto International*, 37(2):627–637.
- Kuester, J., Gross, W., and Middelman, W. (2021). 1D-convolutional autoencoder based hyperspectral data compression. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 43:15–21.
- Lee, K. Y., Seo, J. I., Kim, K.-N., Lee, Y., Kweon, H., and Kim, J. (2019). Application of viewshed and spatial aesthetic analyses to forest practices for mountain scenery improvement in the Republic of Korea. *Sustainability*, 11(9):2687.
- Pan, Z., Tang, J., Tjahjadi, T., Wu, Z., and Xiao, X. (2020). A novel rapid method for viewshed computation on dem through max-pooling and min-

- expected height. *ISPRS International Journal of Geo-Information*, 9(11):633.
- Parent, J. R. and Lei-Parent, Q. (2023). Rapid viewshed analyses: A case study with visibilities limited by trees and buildings. *Applied Geography*, 154:102942.
- Qarah, F. F. (2020). *Efficient viewshed computation algorithms on GPUs and CPUs*. PhD Thesis at the University of South Florida.
- Qiang, Y., Shen, S., and Chen, Q. (2019). Visibility analysis of oceanic blue space using digital elevation models. *Landscape and Urban Planning*, 181:92–102.
- Sanchez-Fernandez, A. J., Romero, L. F., Bandera, G., and Tabik, S. (2021). A data relocation approach for terrain surface analysis on multi-gpu systems: a case study on the total viewshed problem. *International Journal of Geographical Information Science*, 35(8):1500–1520.
- Sanchez-Fernandez, A. J., Romero, L. F., Bandera, G., and Tabik, S. (2022). Vpp: Visibility-based path planning heuristic for monitoring large regions of complex terrain using a uav onboard camera. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15:944–955.
- Stewart, A. J. (1998). Fast horizon computation at all points of a terrain with visibility and shading applications. *IEEE Transactions on Visualization and Computer Graphics*, 4(1):82–93.
- Tabik, S., Cervilla, A. R., Zapata, E., and Romero, L. F. (2015). Efficient data structure and highly scalable algorithm for total-viewshed computation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(1):304–310.
- Tabik, S., Zapata, E., and Romero, L. (2013). Simultaneous computation of total viewshed on large high resolution grids. *International Journal of Geographical Information Science*, 27(4):804–814.
- Wang, Z., Xiong, L., Guo, Z., Zhang, W., and Tang, G. (2023). A view-tree method to compute viewsheds from digital elevation models. *International Journal of Geographical Information Science*, 37(1):68–87.
- Zhang, J., Zhao, S., and Ye, Z. (2021). Spark-Enabled XDraw viewshed analysis. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:2017–2029.