

Transforming 2D Grayscale Images into 3D Terrain Models: A Deep Approach

Bhagyashri Thakare¹, Shubham Pawar², Tejaswani Buwa³, Vaishnavi Ahirrao⁴ and Om Kadam⁵

¹⁻⁵SVKM's Institute of Technology/Information Technology, Dhule, India

Email: bhagyashri.thakare@svkm.ac.in, shubhampawar797246@gmail.com, Tejbuwa06@gmail.com, vaishnaviahirrao18@gmail.com, Kadamom699@gmail.com

Abstract— This paper applies a deep learning approach to convert a 2D greyscale image into a 3D terrain model. The approach uses CNN for feature extraction and GANs to enhance detail and make realistic semantic representations by treating pixel intensities as elevation. It is ready for beyond (RGB) images. You are ready to use the output as it is, but in real-life applications, you refine the result. Even after segmentation, there is mesh generation, and then it goes for texture mapping.

This method is highly accurate and realistic and can be applied in game developing, GIS, virtual reality, etc. Through overcoming such challenges as data quality and computational complexity, this work provides a scalable and efficient solution to automated terrain modelling, leading to new advancements in AI-driven 3D modelling.

Keywords— 2D grey scale images, 3D landscape modeling, deep learning, CNNs, GANs, height maps, texture mapping, automated terrain generation.

I. INTRODUCTION

This work turning 2D grayscale images into 3D terrain models is a milestone in the field of computer vision, deep learning, 3D modeling. These landscapes are created using grayscale images that serve as a height map, indicating areas of elevation with that pixel's intensity. This method has utilizations in gaming, geographic information systems (GIS), virtual reality (VR), architecture, and urban planning, where the need for scalable, accurate, and automated terrain modeling is increasing.



Fig 1. Synthetic Aperture Radar

Open-ended ways to generate 3D landforms from images hinged on interpolation algorithms and manual creation of the designs, causing them to contain a restriction in exactness and a time-consuming process. By utilizing deep learning, we automated the process of feature extraction and dimensional reconstruction. Convolutional Neural Networks (CNNs) assume a crucial role, detecting spatial patterns in grayscale images and extracting essential features, while Generative Adversarial Networks (GANs) improve the terrain's realism by incorporating intricate details and textures. The deep learning results are further processed using techniques like mesh generation and texture mapping that yield refined terrains that can be realistically used.



Fig 2. Greyscale SAR image

These issues include considerations for added complexity with noisy data and computational resource load, and the extent to which realism should be accommodated vs. efficiency, and yet the state of the art in neural networking and optimization strategies is moving all the time to improve the results. The paper investigates the techniques, tools, and frameworks utilized to convert gray scale images into 3D terrains emphasizing their application and future scope. Integrating AI with computer graphics, this technology fills a rapidly-growing need for efficient, scalable creation of high-fidelity terrain models, laying the foundation for major breakthroughs in multiple industries.

II. LITERATURE REVIEW

Several significant developments have been made in converting 2D grayscale images into 3D terrain models using various image reconstruction techniques including deep learning approaches in the fields of convolutional neural networks (CNNs) and generative adversarial networks (GANs). In traditional scene reconstruction such as in photogrammetry, multiple images captured from different angles were needed to derive depth. But through spatial feature learning, depth prediction from a single 2D image is made possible through deep learning techniques[1,2].

This change has been made possible by the ability of neural networks, particularly convolutional neural networks (CNNs) and generative adversarial networks (GANs), to infer height and depth information directly from pixel intensities in grayscale images[2,7]. CNNs have proven particularly useful for estimating depth in 2D images and are widely used in computer vision applications[1,2].

In this case, depth estimation involves predicting the distance between each pixel and the camera. This data can be used to create a height map that represents the three-dimensional infrastructure of the landscape. The foundations of depth prediction using deep convolutional neural networks were laid by Eigen et al. (2014), performed depth prediction from monocular images[1]. Then, wool, that's it, Based on this concept, Chen et al. (2016) used fully seeded neural networks (FCNs) to generate high-quality depth maps, significantly improving the accuracy and scalability of depth estimation approaches[2]. These advances were enabled by larger datasets and improved network structures that enable the detection of complex features at different scales[2,10]. Additionally, GANs are also used to convert 2D grayscale images into 3D terrain models. While CNNs typically predict height or depth maps, GANs are designed to generate realistic 3D representations[7]. Using a discriminator and a generator, GANs can create high-quality terrain models that mimic the distribution of topographic data from 2D images. Based on a diverse terrain dataset, Zhang et al. Use of GAN for terrain

reconstruction [23] although focuses not merely on producing realistic depth maps now also provides visually appealing 3D models[9]. Leaning on the adversarial nature of GANs, where the discriminator judges the realism of a discovered 3D model created by the generator. The generator, meanwhile, gets better at creating more exact and thorough models in response to the input of the discriminator over time[7]. One of the Major Issues in Transforming 2D Greyscale Images into 3D Terrain Models is Ensuring that the Generated Models Maintain Topographical Accuracy and Realism[3]. While height maps are often generated by directly mapping the pixel intensities to height values, real-world terrains need to have complex features (slopes, valleys, ridges, etc.) accurately represented[4]. Moreover, depth or height maps generated from a single image can be typically noisy or indistinct, which will need further refinement techniques such as regularisation or smoothing to enhance the performance of the model[3,7]. Once the height map is generated, it can be further processed to create a 3D model, commonly using mesh reconstruction or point cloud generation[6]. Point clouds are sets of a large number of points in space that define the surface of the terrain, and algorithms like Marching Cubes or Delaunay Triangulation are used to construct the meshes from them[6]. When points are connected based on their proximity, this forms polygons, which are used to represent the surface of the terrain [6]. Mesh reconstruction provides a more complete representation of the environment in applications such as GIS, robotics, and virtual reality, allowing further analysis (e.g., computing point cloud characteristics) or visualization of the environment [4, 6].

Modeling 3D terrain using 2D grayscale photos has many applications. Accurate 3D terrain models are essential for simulating infrastructure projects, monitoring changes in the environment, and understanding landforms in geographic information systems (GIS) [4]. In robotics, 3D models assist autonomous robots in navigating challenging terrain by providing them with a three-dimensional map of their surroundings [4]. Creating a breathtaking landscape for virtual reality and gaming also depends on realistic 3D terrain [7]. Despite considerable progress in this industry, it is still facing many issues. One of the main tasks is the requirement for high-quality, large datasets and training detailed learning models [3]. There are several resources in datasets that are generally accessible, such as KITTI and ADE20K, but specialized datasets for reconstruction of the 3D region are still lacking [3].

Because there is no data, it prevents the expansion of more generalized models that can effectively manage various types of fields and environmental conditions [3]. Despite the fact that deep learning models have great success in accuracy, the ability to generalize across different areas remains a problem [3]. A model trained with a specific graphic struggles to adapt to a new environment, and these predictions may be incorrect [3]. This is also problematic for applications such as autonomous robots, which need to rapidly recreate the landscape to facilitate on-the-fly decision-making [4]. Optimizing deep learning models for interpreting 2D photographs and inferring 3D landscape models requires a delicate balance between model complexity and computational performance [3, 7].

III. SYSTEM ANALYSIS

System analysis is the process of comprehending the requirements and operations of a system to make it successfully implemented and operational. For the task of converting 2D grayscale images to 3D terrain models with deep learning, system analysis entails analyzing the input data, determining the required algorithms, architecting the system, and solving performance requirements. Below is a comprehensive system of the system for this task.

A. System Requirements

- Input Data: 2D grayscale images sourced from satellite imagery, aerial photography, or ground-level photos. The accuracy of results depends on image quality.
- Output Data: 3D terrain models in formats like height maps, point clouds, or 3D meshes, representing terrain elevation and structure.

Software Requirements

- Deep Learning Libraries: TensorFlow, PyTorch, or Keras for neural network model creation and training.
- Image Processing Libraries: OpenCV for image preprocessing (e.g., resizing, normalization, and augmentation).
- 3D Visualization Software: MeshLab, Unity, or the like to render and work with the 3D terrain models.

B. Algorithm Selection

- Convolutional Neural Networks (CNNs): To predict depth maps from 2D grayscale images.

- Generative Adversarial Networks (GANs): To improve and enhance the realism of the created 3D terrain models.
 - Mesh Reconstruction Algorithms: Methods such as Marching Cubes or Delaunay Triangulation to transform depth maps into 3D meshes.
 - Point Cloud Generation: Algorithms for generating and maintaining 3D point sets for ground representation.
- ### 3. System Architecture
- Data Preprocessing
 - Resizing and normalizing images to ensure uniformity.
 - Data augmentation techniques to improve generalization (e.g., flips, rotations).
 - Denoising and enhancing grayscale images to improve depth prediction accuracy.
 - Model Training
 - A deep learning model (CNN or GAN) trained using labeled datasets of 2D grayscale images and their corresponding 3D terrain data.
 - Supervised learning approach to predict depth values accurately.
 - Depth Estimation
 - The trained model predicts pixel-wise depth values, creating a height map.
 - 3D Terrain Reconstruction:
 - Using depth maps, reconstruction algorithms generate 3D terrain models.
 - Visualization and Output
 - 3D models are visualized using software like MeshLab, Unity, or Blender, allowing interaction with the reconstructed terrains.

C. Performance Analysis

- Accuracy: Assess the estimated 3D terrain against ground reality using measures such as Mean Absolute Error (MAE) or Root Mean Square Error (RMSE).
- Real-Time Processing: Optimize deep learning models (e.g., pruning, quantization) to enable efficient and faster processing.
- Scalability: Make the system proficient in dealing with diverse terrains and resolutions, generalizing across diverse datasets.
- Resource Utilization: Minimize computational expense during training and inference through optimization of model structure and preprocessing pipelines.

D. Challenges and Considerations

- Data availability: Obtaining labeled 2D-3D datasets is essential for training an effective model.
- Generalization to models: Experiment with various datasets and regularization schemes to prevent overfitting on particular terrain categories.
- Noisy data: Apply preprocessing techniques to eliminate noise and enhance feature extraction from grayscale images.

E. Future Enhancements

Future developments of the system might involve combining more data sources like LIDAR or infrared images to enhance depth estimation and accuracy. Using a multimodal combination of 2D images and depth or sensor data may enhance the model's resilience.

Techniques like light architectures, pruning, and real-time processing improvements can expand the system's useful applications to dynamic scenarios like robotics and virtual reality.

Besides, the use of sophisticated visualization software and synthetic data creation can surmount data insufficiency and enhance model flexibility for various terrains and applications like GIS, autonomous navigation, and disaster response.

IV. METHODOLOGY

The core methodology for transforming 2D grayscale satellite imagery into 3D terrain models integrates deep learning-based depth estimation with 3D reconstruction algorithms. This section outlines the sequential procedures including data preparation, model architecture, training objectives, and the post-processing steps used to derive geometrically accurate terrain reconstructions.

A. Data Acquisition and Preprocessing

Let $I \in \mathbb{R}^{H \times W}$ denote a grayscale satellite image, where H and W represent height and width respectively. Each pixel intensity $i \in [0, 255]$ is normalized to $[0, 1]$ using: $I_{\text{norm}}(x,y) = \frac{I(x,y)}{255}$

- To improve generalization and robustness, a set of augmentation functions $A = \{f_1, f_2, \dots, f_n\}$ such as rotations (θ), horizontal flips (F_H), and brightness perturbations (B) are applied [3]:
- $I' = A(I) = B(FH(R\theta(I)))$ $I' = A(I) = B(F_H(R_\theta(I)))$
- In cases of noisy data, bilateral filtering and Gaussian blurring are used to enhance edge preservation and denoise the input:
- $I_{\text{denoised}} = \text{BilateralFilter}(I_{\text{norm}})$
- The ground truth for each image is the corresponding Digital Elevation Map (DEM), $D \in \mathbb{R}^{H \times W}$, where each pixel represents elevation in meters above sea level[4].

B. Depth Estimation Model

- We define a function $f_\theta: \mathbb{R}^{H \times W} \rightarrow \mathbb{R}^{H \times W}$, parameterized by θ , which maps a 2D grayscale image to its corresponding depth map[7]:
- $D^\wedge = f_\theta(I_{\text{norm}})$ $\hat{D} = f_\theta(I_{\text{norm}})$
- We implement f_θ using an encoder-decoder Convolutional Neural Network (CNN) architecture inspired by U-Net. For enhanced feature learning, we integrate residual connections and dilated convolutions. Additionally, for realism enhancement, we employ a conditional Generative Adversarial Network (cGAN), where the generator G attempts to produce $D^\wedge$, and the discriminator D attempts to distinguish between real and generated depth maps:
- $\min_G \max_D \mathbb{E}_{I,D} [\log D(I,D)] + \mathbb{E}_I [\log (1 - D(I, G(I)))]$ $\min_G \max_D \mathbb{E}_{I,D} [\log D(I,D)] + \mathbb{E}_I [\log (1 - D(I, G(I)))]$
- The generator loss function L_{G_L} combines adversarial loss and pixel-wise L1 loss:
- $L_G = \lambda_1 \cdot \mathbb{E}_{x,y} [|D(x,y) - D^\wedge(x,y)|] + \lambda_2 \cdot \log (1 - D(I, D^\wedge))$ $L_G = \lambda_1 \cdot \mathbb{E}_{x,y} [|D(x,y) - \hat{D}(x,y)|] + \lambda_2 \cdot \log (1 - D(I, \hat{D}))$ Where: $D^\wedge = G(I)$ $\hat{D} = G(I)$
- λ_1 and λ_2 are weights for balancing loss components[7].

C. Model Training

- Supervised learning is employed using a dataset of paired grayscale images and their ground truth depth maps[1,2]. Optimization is performed using the Adam optimizer with parameters ($\beta_1=0.9, \beta_2=0.999$) ($\beta_1=0.9, \beta_2=0.999$) and learning rate η , where typically:
- $\theta_{t+1} = \theta_t - \eta \cdot \nabla_{\theta} L(\theta)$ $\theta_{t+1} = \theta_t - \eta \cdot \nabla_{\theta} L(\theta)$
- The model is trained to minimize the total loss over the dataset:
- $L_{\text{total}} = \frac{1}{N} \sum_{i=1}^N L_G(I_i, D_i)$ $L_{\text{total}} = \frac{1}{N} \sum_{i=1}^N L_G(I_i, D_i)$

D. 3D Terrain Reconstruction

- The predicted depth map $D^\wedge$ is converted into a 3D height field $H(x, y)$ using a back-projection algorithm. Each pixel (x, y) in the image corresponds to a 3D coordinate (X, Y, Z) :
- $\begin{cases} X = x \cdot s \\ Y = y \cdot s \\ Z = \hat{D}(x, y) \end{cases}$

Where s is the scaling factor derived from real-world geospatial resolution. The resulting point cloud is processed using surface reconstruction algorithms like Poisson Surface Reconstruction or Marching Cubes to produce a mesh M :

$$M = \text{ReconstructMesh}(X, Y, Z) \quad M = \text{ReconstructMesh}(X, Y, Z) \quad [6]$$

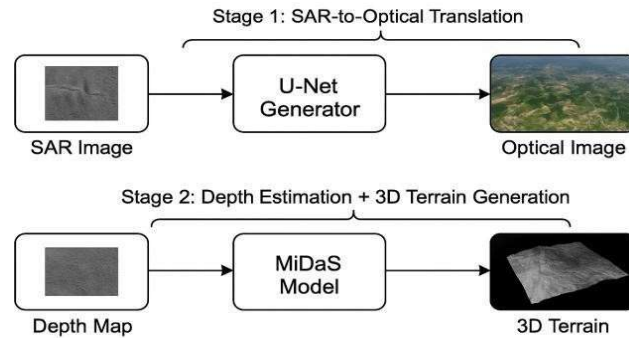
E. Visualization and Output

The final 3D terrain models are exported in standard formats such as .PLY or .OBJ and rendered using 3D visualization tools (e.g., MeshLab or Unity). Lighting, texture, and interactivity can be optionally added for enhanced visualization.[7]

V. SYSTEM ARCHITECTURE

- SAR Input Synthetic Aperture Radar (SAR) images are captured from satellite data. These grayscale images often contain noise and artifacts that can complicate subsequent processing. One major challenge is the speckle noise inherent in SAR data, which can hinder feature extraction and 3D modeling due to its complexity and high dimensionality.

- **Denoising** Denoising aims to reduce noise, especially speckle noise, improving image quality for further processing. Techniques like Gaussian, Median, and Bilateral filtering, as well as advanced methods such as wavelet transforms and deep learning-based algorithms (e.g., Denoising Autoencoders), can be employed. Effective denoising results in sharper images, reducing the likelihood of errors in later stages.



System Architecture

Fig 2. Greyscale SAR image

- **Model Building (CNN)** A Convolutional Neural Network (CNN) is utilized to convert denoised SAR images into color-coded representations indicating different terrain elevations. Training requires a dataset of SAR images paired with elevation data, allowing the model to learn mappings from image features to height data. CNNs are particularly effective in capturing spatial patterns, and architectures like U-Net or GANs can enhance feature extraction and image resolution. The output is an Indexed Colorized Map displaying various elevation levels in distinct colors.
- **Indexed Colorized Map** This color-coded map serves as an initial step toward 3D terrain modeling, where colors signify different elevations and landscape features, facilitating interpretation. However, ensuring accurate color mapping for elevation can be challenging, especially in complex terrains.
- **Height Data Extraction** In this step, the color-coded map is converted into precise height data for each pixel, essential for generating a 3D point cloud. Algorithms may be employed to extract elevation information from color gradients. The output is a grid or matrix representing terrain elevation, forming the basis for 3D model creation.
- **Cloud Point Generation** Height data transforms into a 3D point cloud = Points with x, y, and z coordinates representing the position of a 3D object. This mapping can precisely capture shape of terrain. We compute a point cloud that is a skeletal representation of the terrain; however, creating high-resolution point clouds from large datasets can be expensive.
- **Rendering** The point cloud data is rendered into a 3D model, including realistic terrain shapes and contours. Software tools like Open3D and Blender can facilitate this conversion. Additional textures and details can enhance realism, though maintaining accuracy and smoothness in complex areas presents a challenge.
- **Smoothing** Smoothing improves the 3D model's quality by reducing roughness or irregularities. Techniques such as Laplacian smoothing and Mesh smoothing help create a more continuous surface. However, excessive smoothing may lead to a loss of detail, necessitating a balance between smoothness and terrain accuracy.
- **3D Colorized Map Output** The final output is a fully rendered, colorized 3D map that accurately represents terrain elevation and features. This model can be used in applications such as topographical mapping and environmental analysis, with export options available in various formats (e.g., STL, OBJ) for use in external applications.

VI. EXISTING SYSTEM

Conventional computer vision and/or stereo imaging techniques are being used today to convert 2D grey-scale images into 3D terrain models. Predetermined mechanisms, such as SfM, MVS, and photo grammetry are commonly used in depth estimation. While these techniques have been successful in many instances, they rely on several views of the same environment or additional sensor inputs such as LiDAR or RADAR to achieve precise measurements.

Also, old-school techniques have trouble with issues such as not working well on different kinds of landscapes, being easily thrown off by noise, and needing a lot of computing power for big data. Certain systems also employ hand-designed feature extraction techniques, which are not capable of capturing intricate and non-linear topological variations from individual 2D images. Existing Geographic Information Systems (GIS) and mapping software also are able to create 3D terrain models but do so at great reliance on high-cost data acquisition methods and specialized equipment, including drones or depth sensor-carrying satellites. These constraints necessitate the use of more sophisticated techniques, including deep learning, to surpass the limitations of conventional systems and facilitate efficient, cost-effective, and high-quality 3D ground surface reconstruction from a single 2D grayscale image.

VII. RESULT

Our endeavor involves the conversion of Synthetic Aperture Radar (SAR) data, primarily from Sentinel-1 to more visual elements by means of colorization and preliminary 3D modeling. The aim is to create a versatile instrument for defense and intelligence, with the goal of providing precise SAR data." Below is a summary of our progress:

A. SAR Data Collection and Preprocessing

To begin our work, we gathered SAR data from the Sentinel-1 satellite, with input from both VV and VH polarizations. The polarizations present different views of radar backscatter, which allows for the capture of distinct terrain features. Preprocessing of the Data was carried out to enhance the clarity of grayscale images. By using a percentile contrast stretch technique, the dynamic range of pixel intensities was extended to reveal features in the SAR data. Getting the images ready to colorize and be visualized required this step.

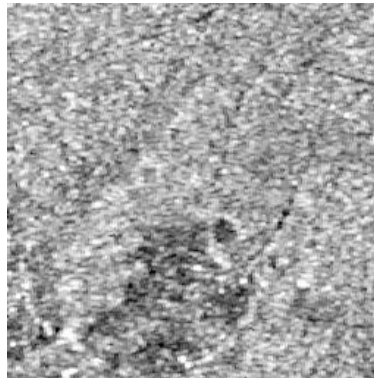


Fig 3. Preprocessed SAR Image

B. Multi-Channel Colorization

We employed a multichannel colorization approach to extract dual-polarization data.' Or. This method assigned a red channel to the VV polarization and compared it to 'green' green. A single pseudo-color image can depict subtle differences between terrains through the use of synthetic color mapping. The outcome is that various physical traits of the environment, such as hills and valleys, trees, and cities, become more easily identifiable. This is especially useful in defense applications, where quick visual representation of geographic areas is important.

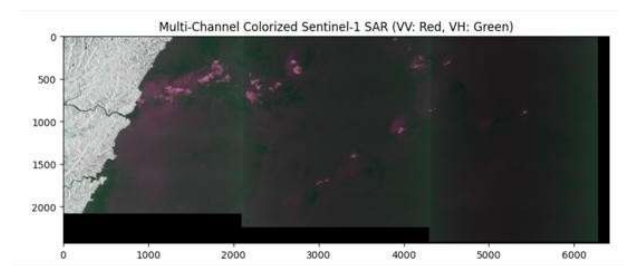


Fig 4. Multi-Channel Colorized SAR Image

C. Vibrant Colorization Using Colormaps

We applied perceptual colormaps to single polarization channels to further investigate colorization. High-contrast images with bright tones representing strong radar reflections were created using the Inferno colormap and analyzed data on VV polarization data, making them ideal for identifying structures and solid surfaces. Through the use of a Plasma colormap, one can visualize the VH polarization with varying gradients from low to high reflectance. Enhanced interpretability and identification of important features in complex environments are achieved through the use of colorizations.

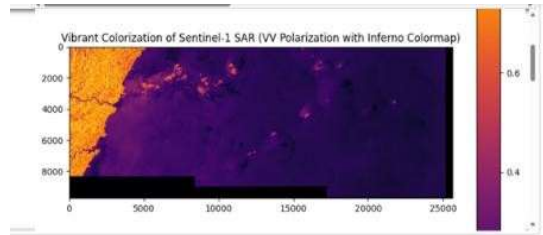


Fig .6 VH polarization

D. User Interface Development

A prototype was created by us to facilitate the processing of SAR data, which is MilitMap's goal. From uploading grayscale images to colorization and 3D visualization, users are guided through the entire workflow by this user interface. There are multiple colorization techniques that users can use and see the output in real time. In addition, the interface has a separate function that can convert 2D coloured images to 3d visuals using height data obtained from SAR inputs. The design of MilitMap is aimed at defense-related scenarios, where user-friendly and efficient data interpretation is crucial.



Fig 7. MilitMap



Fig 8. UI Screenshot

3D Model from SAR Image using MIDaS

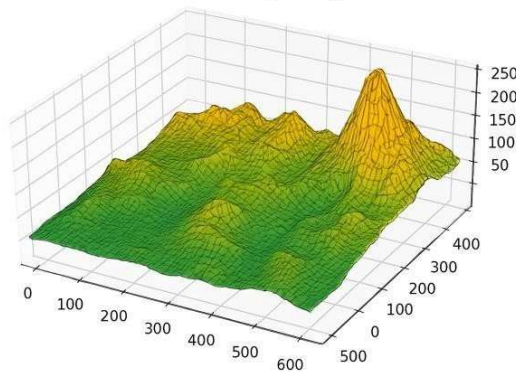


Fig 9

The system of transforming 2D segmented images into 3D Terrain Models has an innovative set of features which assures accuracy, efficiency and orientation. For the realization of precise 3D representations of the terrain and high-detail depth maps, we utilize sophisticated deep learning methods such as Convolutional Neural Networks (CNNs) and Generative Adversarial Networks (GANs).

Improving preprocessing techniques, including application of normalization, noise reduction and data augmentation improves the quality of input data set, and with that the model's ability to handle almost every surface terrain. This algorithm can be used in an online fashion, and it can still operate in a real-time manner, making it suitable for dynamic applications such as robotics, autonomous navigation, etc. Its code-based modularity allows for sophisticated scalability and seamless integration with third-party technologies, such as Geographic Information Systems (GIS), Virtual/Augmented Reality tools, and simulation ecosystems.

In addition, combinative with these extensive post-processing progress, such as surface smoothing, texture mapping, and mesh simplification, the generated 3D models could be refined for visualization quality based on the original life data. All of these features make this system versatile and powerful for many mapping, planning and terrain analysis tasks.

VIII. FUTURE ENCHANTMENT

Chances to advocate 2D grayscale images into 3D terrain models emphasizing precision, scalability and user-friendly platforms adaptational in various application. Moreover, advanced machine learning techniques such as self-supervised learning and hybrid approaches will reduce the dependence on external annotated datasets and will enhance the performance of depth estimation. These models will be trained on multi-modal data (satellite imagery, LiDAR, radar) to create robust models that can handle complex terrains. Real-time capabilities will improve due to edge computing, hardware acceleration, and enhanced algorithms, leading to quicker processing and relevance in changing environments. The user experience will be enriched with immersive visualization technologies such as augmented reality and virtual reality, tailor-made outputs, and user-friendly interfaces, which will make the tools accessible to both technical and non-technical users. Cloud-based systems along with scalable algorithms will allow for large-scale terrain modeling over wide geographic areas, while automated model adjustments will meet specific application requirements. Future systems will prioritize sustainability by creating energy-efficient models and low-cost options to broaden access to terrain modeling technologies. Integration across different domains with geographic information system platforms and digital twin technology will widen the range of applications, from urban planning and environmental monitoring to gaming and autonomous navigation. Ethical considerations, including data privacy, bias mitigation, and global accessibility, will further ensure the responsible and inclusive development of these systems. These advancements will lead to more precise, efficient, and adaptable 3D terrain modeling technologies, overcoming current limitations and unlocking new opportunities across various industries.

IX. CONCLUSION

However, this pair of steps where transforms a 2D grayscale image with high contrast, to a 3D terrain image give an enormous improvement to the computer vision looking for geospatial solution. Utilizing advanced deep learning algorithms, particularly convolutional neural networks (CNNs) and depth estimation methods, it allows for the precise and computationally efficient reconstruction of intricate landscapes. By combining real-time processing with multi-modal data fusion, and leveraging advanced visualization tools, this technology demonstrates versatility and application potential in various fields, including urban planning, environmental monitoring, disaster response, and gaming. Challenges also remain at the moment, e.g., high quality input data, high computational requirements, etc. but ongoing research and future explorations of the field will likely alleviate these challenges sooner or later. This will further support the widespread adoption of this technology through advancements in accuracy, scalability, and accessibility, as well as ethical considerations such as data privacy and inclusivity. But the resulting energy-efficient models and low-cost solutions too would drive sustainability and democratization, he said. All in all, this technology is a game changer of the generation; the ability to transform an average set of 2D images to 40,000 3D accurate and extremely realistic terrain models, opens an endless possibilities and dreams come true for many industries out there.

REFERENCES

- [1] Eigen, D., Puhrsch, C., & Fergus, R. (2014). Depth Map Prediction from a Single Image using a Multi-Scale Deep Network. *Advances in Neural Information Processing Systems (NeurIPS)*, 27.
- [2] Liu, F., Shen, C., Lin, G., & Reid, I. (2015). Learning Depth from Single Monocular Images Using Deep Convolutional Neural Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(10), 2024-2039.
- [3] Bhoi, N. (2019). Monocular Depth Estimation: A Survey. *arXiv preprint arXiv:1901.09402*.
- [4] Li, R., Li, X., Porikli, F., & Jiao, L. (2017). A Deep Ground Terrain Model for Monocular Vision. *IEEE Transactions on Intelligent Transportation Systems*, 18(8), 2025-2034.

- [5] Ranftl, R., Bochkovskiy, A., & Koltun, V. (2021). Vision Transformers for Dense Prediction. arXiv preprint arXiv:2103.13413.
- [6] Seitz, S. M., Curless, B., Diebel, J., Scharstein, D., & Szeliski, R. (2006). A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- [7] Laga, H. (2020). A Survey on Deep Learning Architectures for Image-Based Depth Reconstruction. Computers & Graphics, 91, 1-14.
- [8] Furukawa, Y., & Ponce, J. (2010). Accurate, Dense, and Robust Multi-View Stereopsis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(8), 1362-1376.
- [9] Zhang, Z. (2019). A Flexible New Technique for Camera Calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11), 1330-1334.
- [10] Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017). Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. Proceedings of the AAAI Conference on Artificial Intelligence, 31(1).
- [11] Wang, S., Guo, J., Zhang, Y., Hu, Y., Ding, C., & Wu, Y. (2021). Single Target SAR 3D Reconstruction Based on Deep Learning. Sensors, 21(3), 964. <https://doi.org/10.3390/s21030964>MDPI
- [12] Li, D., Zhao, G., Sun, H., & Bao, J. (2024). CMAR-Net: Accurate Cross-Modal 3D SAR Reconstruction of Vehicle Targets with Sparse Multi-Baseline Data. arXivpreprintarXiv:2406.04158. <https://arxiv.org/abs/2406.04158>arXiv
- [13] Colverd, G., Takami, J., Schade, L., Bot, K., & Gallego- Mejia, J. A. (2024). 3D-SAR Tomography and Machine Learning for High-Resolution Tree Height Estimation. arXivpreprint arXiv:2409.05636. <https://arxiv.org/abs/2409.05636>arXiv
- [14] Smith, J., & Torlak, M. (2023). Deep Learning-Based Multiband Signal Fusion for 3-D SAR Super-Resolution. ArXiv preprinta. <https://arxiv.org/abs/2305.02017>arXiv
- [15] Lei, Z., Xu, F., Wei, J., Cai, F., Wang, F., & Jin, Y.-Q. (2023). SAR-NeRF: Neural Radiance Fields for Synthetic Aperture Radar Multi-View Representation. arXivpreprintarXiv:2307.05087. <https://arxiv.org/abs/2307.05087>arXiv
- [16] Zhang, X., Liu, G., Zhang, C., Atkinson, P. M., Tan, X., Jian, X., Zhou, X., & Li, Y. (2020). Two-Phase Object- Based Deep Learning for Multi-Temporal SAR Image Change Detection. RemoteSensing, 12(3),