Efficient Volume Estimation for Dynamic Environments using Deep Learning on the Edge

Chandan Kumar  
Department of Computer Science  
Iowa State University  
Ames, IA, 50011 USA  
chandan@iastate.edu

Yamini Mathur  
Department of Computer Science  
Iowa State University  
Ames, IA, 50011 USA  
yamini@iastate.edu

Ali Jannesari  
Department of Computer Science  
Iowa State University  
Ames, IA, 50011 USA  
jannesar@iastate.edu

Abstract—The utility of edge devices has increased in volume estimation of uneven terrains. Existing techniques utilize several geo-tagged images of the landscape, captured in-flight by an edge device mounted over a UAV, to generate 3D models and perform volume estimation through manual boundary marking. These methods, although accurate, require significant time, human effort and are heavily dependent on GPS. We present an efficient deep learning framework that detects the object of interest and automatically determines the volume (independent of GPS) of the detected object on-the-fly. Our method employs a stereo camera for depth sensing of the object and overlays a unit mesh grid over the object’s boundary to perform volume estimation. We explore the accuracy vs computational complexity trade-off on variations of our technique. Experiments indicate that our method reduces the time for volume estimation by several orders of magnitude in contrast to existing methods and is independent of GPS as well. Also, to the best of our knowledge, this is the first method that can perform volume analysis in a dynamic environment.

Index Terms—Deep Learning, Volume Calculation, Edge AI.

I. INTRODUCTION

There has been a dramatic increase in the utilization and advancement of embedded AI in recent years. As with other emerging technologies, the ever-increasing compactness and computing capabilities of embedded AI devices will inevitably open up a multitude of possibilities. There have been rapid developments that have taken place in surveying with the help of Unmanned Aerial Vehicles (UAVs) to understand and map landscapes correctly and accurately. The approach to calculate and determine the volume of objects using UAVs has proved to significantly reduce the time taken to complete these tasks. The technical advancements have led UAVs to embed several types of equipment sensors like GPS, cameras, LiDAR, Ultrasonic, etc. The data collected from this sensory equipment, coupled with techniques like Photogrammetry [10] [4] and Orthomosaic [8] [4], has had a significant impact on surveying and 3D model generation. These 3D model generations have led to several results, including the development of a variety of software that is constructed with the aim to reduce time and effort, and improvement in the accuracy of the 3D modeling methods. Data collected for these surveying purposes has been used to further research in numerous fields for instance, architecture and civil engineering for applications such as 3D building reconstruction [25] and crack detection [24].

However, in spite of all these advancements, the current methods include the process of sending a UAV over an entire area to capture several geo-tagged images, which is then brought back to a base station or, sent through the cloud as input to the software which subsequently builds into a 3D model and finally draws boundaries over the area of interest manually. This existing procedure is cumbersome and requires several hours of work. Consequently, it leads to these methods deeming sub-optimal in a dynamic environment where determining the volume is constrained by time [10]. Examples of these dynamic environments are mining regions where the ore is mined continuously, and by the time current methods are applied to determine the volume of mined material or waste, the volume would have already changed, making it not possible to conduct effective momentary decision making. Furthermore, in the case of natural disasters like floods and storms, there are large pits left in the ground, or large sections of roads are blown away. In cases of such natural calamities, the subsequent search and rescue mission cannot wait for hours after the disaster to start the search and rescue operations. Consequently, in the event of such devastating natural disasters, it is imperative to determine the volume of pits and holes that can potentially be dangerous and can hamper rescue operations. By employing these current techniques for such tasks that require surveying and mapping in rapidly changing environments, by the time the mapping is completed, the landscape would have already changed, and possibly developed new pits and the already existing ones may have grown further. The delay in mapping directly hampers the ability to make the right decisions in already troubled times. Consequently, if surveying and determining the volume happens in near real-time, it can greatly benefit the decision making, operation, and also determine the additional requirements for the dynamic environments.

The subsequent paper is organized as follows. Section II discusses the related works in this area as well as compares and contrasts with our work. It also gives a glimpse of the recent advancements in the techniques that have been developed. Section III gives an insight to our contribution. Section IV explains our approach to the problem and how it is being solved by us. This section is subdivided into four subsections, each of which provide a detailed working of different pieces
of our approach. While calculating the volume of the object, we make the following two assumptions:

- The surface of the detected object is planar (non-rugged).
- The lateral surface is not hollow.

In Section V, the experimental results are reported, tabulated, and discussed. Lastly, Section VI provides the conclusion of our work.

II. RELATED WORKS:

Unmanned Aerial Systems have now become common practice in collecting visual information like images and videos in fields of application, such as mapping [8], monitoring [8], structural monitoring, cultural heritage documentation, agriculture, and other surveying applications. Volume calculation [8] remains a very critical task in a dynamic environment. Although different techniques for volumetric capture from range data have been an active research topic for a long time. Still, research on outdoor volumetric capture using UAVs is limited. It gained in popularity through the availability of affordable depth cameras and parallel computing hardware.

Several attempts have been made to calculate the volume of stockpiles in mining areas [31], determine crop yield in agriculture, prepare a recovery process in case of natural disasters [8], etc. Most of these existing works utilize surveying methods [3] and photogrammetry techniques [10], [4], [6], [5], [9] for the calculations. These methods require taking multiple geotagged images from a drone and uploading them to computer software that can create an orthomosaic [4], [8] and/or Digital Surface Model(DSM) [8] of the entire landscape. Then, a human operator defines a boundary in this 3D environment, based on the points defined by GPS [4], for the object of which the volume has to be calculated.

However, these methods have proven to be inefficient when the task needs to be accomplished in a short time or, in a risk area [10]. Furthermore, it is noteworthy that these methods are heavily reliant on GPS data.

Depth cameras have been very popular in a variety of research involving UAVs [11]. They have been very frequently used for indoor mapping applications [12], [19], [17], testing SLAM techniques for UAVs equipped with laser scanners [18], depth cameras [20], [21], [22], [23]. They have been used in reconstructing 3D point cloud data to make accurate measurements from a set of real images of an object to make measurements of an object’s form [9], computer-aided drawing(CAD) to simulate frames of objects in a simulated environment. In an outdoor environment, architectural reconstruction is another major application of 3D reconstruction [11]. However, these methods are time consuming and require a lot of human effort.

Other works represent 3D shapes as a probability distribution of binary variables on a 3D voxel grid, using a convolutional deep belief network [14] and usage of both voxel grids and meshes to represent 3D shapes [13]. These approaches make use of voxels, meshes for 3D generation Truncated Signed Distance Function (TSDF) and Heat Kernel Signatures (HKS) for pose estimation applications. However, our approach implements directly on a 3D object and we make use of only Euclidean Distance.

Furthermore, notable work has also been conducted to calculate volume using Deep Learning in the food industry [28] [29] and in the medical industry using medical data [30] for estimating volume of the heart. However, they use it mainly for boundary marking [28] or employ a pre-built 3D model of items under consideration [29].While these methods are interesting and have tried to eliminate the process of manual boundary marking, they still undergo the problem of preprocessing preventing the process from becoming applicable in all cases as well as relevant in dynamic environments.

III. CONTRIBUTION

The contributions of our work are four-fold.

- An algorithm to calculate volume on the fly (Section : IV-D): We present a novel algorithm to calculate the volume of an object for dynamic environments using Deep Learning [Section: IV-D]. This work aims at reducing the time and effort invested in the traditional methods to estimate volume by a large margin. (Fig. 1, Fig. 2)
- Removing dependence on GPS (Section : IV-C): Our algorithm makes it possible to calculate the volume of objects without requiring GPS. To the best of our knowledge, no other technique has been successfully able to achieve volume calculation without requiring GPS. The lack of dependence on the GPS makes the task of volume calculation work accessible in outdoor as well as the indoor environments.
- Experimental evaluation (Section : V): We compare our work against the ground truth. We also compare our work to the currently existing methods. We will provide all our experiments and results.
- Artifacts for reproducibility: We make the artifacts for reproducibility publicly available so as to support further development over our work.

IV. OUR APPROACH

Volumetric Analysis for dynamic environments is a complex problem which involves Object Detection, Instance Segmentation, boundary markings, depth sensing and volume calculation. We tackle each of these problems and provide a novel approach to calculating volume of objects in dynamically changing environments.
Fig. 1. State-of-the-art method (9 Steps) for volume estimation based on photogrammetry and 3D reconstruction.

Fig. 2. Our proposed Approach (5 Steps) for efficient, fast and accurate volume estimation in dynamic environments.

Fig. 3. Workflow of our Volume approach.

changing environments. First, we bring the camera to the area of the object of interest. For this we mount a stereo camera to an edge device on-board a UAV. While UAVs are one of the platforms that can be utilised, the method is also tested on a test-bed proving it to be usable in robotic arms as well. Using the stereo depth sensing camera (ZED Mini) we detect the object and draw the boundary (both Bounding Box and Mask). All the Cartesian coordinates over the boundary of the object and all integer coordinates within the boundary are then calculated. Finally, the camera calculates the depth of each coordinate within the boundary and determines the volume of the object (See Figure 2).

A. Hardware Platform

We performed our experiment on Nvidia Jetson TX2, which comes with Nvidia Pascal architecture and has 256 NVIDIA CUDA cores, a dual-core Denver 2 64-bit CPU and a quad-core ARM 57 complex with 8GB of 128-bit LPDDR4 onboard memory. It runs Ubuntu Linux4Tegra (L4T). This GPU is mounted on Auvidia J-120 carrier board to provide compactness of design and keeping the weight down. We also use ZED Mini Stereo camera for Depth Detection. We mount the entire hardware (Nvidia TX2 mounted on Auvidia carrier board and ZED-Mini camera) on the custom drone built in our lab.

The frame of the drone is Tarot 650 Sport Quadcopter (Fig. 4).

In the next setup, we create a test-bed with 8020 Aluminium frame with camera fitted at a fixed height (See Figure). The ability of camera movement along the horizontal plane is manual but was not needed for the experiment. This setup allows us to create the real environment inside the lab.
this setup, we perform the experiment on Nvidia AGX Xavier instead of Nvidia Jetson TX2. This development platform comes with 512-cores Volta GPU with tensor cores, 8-Core ARM v8.2 64-Bit CPU, 8 MB L2 + 4 MB L3 and 32 GB 256-Bit LPDDR4x — 137 GB/s onboard memory. The ZED camera is plugged directly to Nvidia AGX Xavier and does not require any carrier board. Nvidia AGX Xavier also runs Ubuntu Linux4Tegra(L4T), similar to Nvidia Jetson TX2, thus keeping the environment same. While the neural network model on these edge devices are run for inference, the model is trained on Dell Precision 7820 Tower with Intel Xeon(R) Silver 4114 CPU @ 2.2GHz processor with 20 cores, running Ubuntu 20.04.1 LTS with 94GB RAM and Nvidia Titan XP GP102 graphics.

B. Object Detection and Segmentation

We train our model on Mask-RCNN [7]. Mask-RCNN [7] is a two stage detector that has state of the art accuracy outperforming every other single stage detector currently on COCO Minival Benchmark. Mask-RCNN [7] is built on the top of Faster-RCNN [1]. While it is known that two-stage detectors are slower [33], but are more accurate than single stage detectors [27]. The choice of using a two-stage detector over single stage detector, despite the fact that they are slower, is accuracy.

We know that the IMU drift present in the stereo camera would have some impact over the depth values resulting in some distortions in the volume calculation. ZED camera has a mean error of 13.87 % [26] in depth measurements and can reach up to 23.29 % [26]. However, despite of having a high margin of error, the decision of using ZED camera for our experiment depends on several factors, namely faster recording of depth images [26], better memory management [26], capturing details in low light conditions [26] and longer range along with it’s portability. The dataset we have used for training is a collection of high resolution images taken from a DJI drone over an open pit mine. These images are annotated to create a COCO-style dataset.

We trained our model with a Mask-RCNN model, pretrained on COCO that uses the Mobilenet(v2) backbone. Transfer learning is our next step on the pretrained model on the custom COCO-style dataset we have created with custom labels to detect the heaps, which is our object of interest.

C. Removing dependence on GPS

Since we are trying to implement our work in mining and other remote areas, we wanted to ensure that the our work is relevant and usable in environments that will lack GPS signal. Independence from GPS makes our work suitable for all kinds of environments. In order to achieve this independence, and still be able to map the real-world distances, we use a different frame of reference. We use a frame of reference called the World Frame, which essentially marks the position from which we start the tracking through the camera as (0,0) in the Cartesian coordinates. For our case, we just need the Cartesian distances to find the boundary points and all the points within the boundary points. This hinders our ability to determine the real-world position of the detected object, but does not interfere in detecting the object correctly and determining its volume. With this method we only remove the dependence of GPS for volume estimation task, however, other tasks, such as pose estimation etc. might still be dependent on GPS.

D. Volume Estimation

We construct a 3D Bounding Box(Fig. 6) across the detected object(O) as well as a mask over the detected object. We divide the bounding box into a mesh of small cuboidal cells [4] each having a length(L)= 1 mm(1 × 10⁻³ m) , width(W)= 1 mm(1 × 10⁻³ m) and height(H)= height of the camera from the ground (fixed before each flight). The volume of a cuboid is given by,
Algorithm 1: Our algorithm

Input: Live Stream from camera
Output: Detected object, volume

1: while Camera Stream=True do
2:    run trained model
3:    detect heap \(\triangleright\)Keeps detecting heap till camera is on
4:    if accuracy \(\geq\)90\% then
5:      generate masks[] \(\triangleright\) Generates masks if object is detected with 90\% + accuracy
6:      counter=0
7:    else
8:      detect heap
9:    end if
10:   for All rows in masks[] do
11:      while counter \(!=\)number of rows do
12:        if Scale=1 then \(\triangleright\)Taking into account scaling effect when the image represents large volumes
13:          Convert to World Frame
14:          if Material_type == Key of AoR then
15:            Apply AoR(R)
16:            \(\triangleright\)AoR(R) represents a user defined pre-loaded dictionary containing angle of repose of known materials
17:          else
18:            continue
19:        end if
20:      else
21:        continue
22:      end if
23:    end if
24:    end for
25:    calculate volume from newArray[] \(\triangleright\)calculates volume
26: end while

\[ V=L \times W \times H. \]

Hence, the total volume of the bounding box,

\[ V_{BoundingBox} = \sum_{i=1}^{n} L_i \times W_i \times H_{Cam} \quad (1) \]

where,

- \(L_i\): Length of the Cuboidal Cell
- \(W_i\): Width of the Cuboidal Cell
- \(H_{Cam}\): Height of the Camera

It is evident that the top surface of the object would not be plane and can carry some curvature which may increase or decrease the height of the object. However, we have divided the surface into a mesh of 1\(\times\)1\(\times\)1 mm in length and width. Height is the height of the camera.

Thus, the height of the column can be stated as height of the column with plane surface \(\pm\) height of the curved surface. From equation 2, we have,

\[ H_{Col} = H_{Col}(plane) \pm \Delta h \quad (3) \]

We fly the drone mounted with ZED Camera vertically above O (Fig. 7) and try to capture depth data of each point of the mesh. Hence,

\[ H_{Cam} = \text{Detected depth}(D) + \text{Height of the object}(O) \]

\(\Rightarrow\) Height of the object(Obj), \(H_{Obj} = H_{Cam} - \text{Detected depth}(D)\)

(Figure 7)

Where, \(H_{Cam}\) : Height of Camera

\(D\) = Detected Depth or Camera to Object height

Therefore,

\[ V_O = \sum_{i=1}^{n} L_i \times W_i \times H_O \quad (4) \]

When the height of the camera becomes greater than the detected depth, it means it has arisen due to an error in depth calculation by ZED camera as this is not possible in the case of plane surface taken into consideration in this experiment. Also, when the height of the camera becomes equal to the detected depth, then it is observable that there is a void space which does not participate in total volume of the object i.e. at this particular coordinate the object is absent. Hence for increasing the accuracy, we equate the depth at those points to 0 where,

\[ H_{Cam} \geq D \quad (5) \]
to the height of the camera. Since we know, by virtue of manually fixing the height of the camera before the experiment, such points do not either exist or do not lie over the object hence should not be considered for volume estimation. However, the approximation and volume calculation formula (Equation 1) would only be valid when the coordinates are close to each other. When the adjacent coordinates are far from each other there would be a case of empty volume spaces. This case might arise as we increase the size of the object keeping the same image size. In such a case, the distance between two adjacent pixels points \( x \) and \( y \) might not be as close to two adjacent coordinates \( X \) and \( Y \) in the real world object. In such a scenario, the concept of using the volume as a continuation summation would not work, as the equation,

\[
\lim_{w \to 0} \Delta W \neq 1 \tag{6}
\]

\[
\lim_{l \to 0} \Delta L \neq 1 \tag{7}
\]

Hence,

\[
\lim_{l \to 0} \lim_{w \to 0} \Delta A \neq 1 \tag{8}
\]

where: \( A \) is area

becomes not equal to 1. Thus, in order, to take volume as a summation of adjacent points we need to ensure that we take into account the the scaling effect of the object. A scale for a photograph is defined as the ratio of the distance on the photo to the corresponding distance on the ground. It can be expressed in following way:

\[
\text{Scale} = \frac{f}{H} = \frac{\text{Sensor Pixel Size}}{\text{Ground Pixel Size}} \tag{9}
\]

Where:

\( f \): Focal length of Camera

\( H \): Height above the ground level

Using the scale, we create additional real world coordinates in between two adjacent pixel coordinates. The number of real world coordinates created depends on the distance between two adjacent points after the scaling effect. The algorithm creates additional points to ensure that the distance between two adjacent real world coordinates is not greater than 1mm. However, these additional points do not have any volume as they do not exist in pixel coordinates and hence the depth cannot be estimated by the camera. To resolve this issue, we use the following two methods:

1) **Utilising Angle of Repose(R):** Since we are looking at granular materials in this experiment, we take into account the angle of repose [34] of the material to predict the height on next point. The neural network is supplied the angle of repose values for different material which the neural network uses as per the detection. In the absence of the value of angle of repose, the neural network switches to the method 2 i.e. mimicking angle of repose.

2) **Mimicking Angle of Repose(\( R_m \)):** In the event of unknown material type, our network tries to mimic the angle of repose by calculating the angle \( \theta \) between the two adjacent real world point coordinates \( A \) and \( B \). The calculation of Mimicked angle of Repose(\( R_m \)) is as under:

\[
P = |H_A - H_B| \tag{10}
\]

\[
B = SF \times DE \tag{11}
\]

\[
\tan \theta = \frac{P}{B} \tag{12}
\]

Where:

\( H_A \): Height at coordinate A

\( H_B \): Height at coordinate B

\( SF \): Scaling Factor

\( DE \): Distance Equivalent

We calculate the height of any pixel point A on the image and the height of another adjacent pixel point B on the image to get ‘P’ (Equation 10) and from the formula of Scale (Equation 9), we can get ‘B’. Using Equation 10 and 11, we can calculate the angle \( \theta \) between them.

Using the above two methods we can calculate the height of additional adjacent points as we already know the height of points which exist in both pixel and real world coordinate systems.

V. EXPERIMENTAL RESULTS AND DISCUSSIONS

We demonstrate a novel approach to solve the problem of volume estimation that is applicable for dynamically changing environments, which does not require human intervention, and manual labor and can work in both GPS enabled as well as GPS-denied environments. The purpose of this work is to employ it on to edge devices so as to take the machine

\[\text{N.R. : Not Reported}\]
We have limited our algorithm to calculate volume of objects which are detected with more than 90% accuracy. This is done to ensure that the algorithm does not pick up random objects with low confidence and calculates their volume. We have tabulated the result of our model in the Table II. The results are based on the calculations from our test bed, where we could compare and contrast with ground truth. For our experiments, we have used mason sand to create the heap. To validate our approach against a verified ground truth, we measured the volume of sand using a measuring flask and created the heap by pouring the sand to form a heap of definite volume acting as our ground truth.

After contrasting the computation time to other methods in Table III, which take several hours to estimate volume of arbitrary object using existing methods, our method does performs this calculation and estimation in just a few seconds. We could achieve this by removing the pre-processing overhead present in most of the methods. We also report the minimum error percentage of each method as well as the Mean error percentage. Our method performs better in both the parameters. While it is also noteworthy that only Volume R-CNN can be used for multiple and general objects as well as the only method that can be used in dynamic environments. In table IV, we compare our method with the ground truth. We take multiple readings with varied parameters to ensure that the method applies to different volumes. Keeping in mind the IMU drift and error in depth measurement of the ZED camera, our method performs very close to the ground truth with percentage error as low as 1.24%. It is noticeable that the maximum percentage error is 9.812%. Also, as the volume of the ground truth increases the time taken by our model increases as the model has to process more data points. However, the increase in computation time is not linear and is still under 30 seconds.

While performing the experiments, we noticed that the value of volume changes in each run of the code. This seems to be due to the error in depth detection of ZED camera. We also observed that the lighting also plays a role in object detection as the number of data points increased in bright lighting conditions and decreased when the lighting was low. Also, since this method depends on depth detection for volume estimation, the method would not work in the event of occlusion. However, the method would try to find the volume of object if it is partially occluded based on the mask generation by the object detector.

Our method is scalable to a distributed architecture as well. Multiple edge devices can be connected locally to work coherently or can also be connected over a network. The edge device can also be used to send data over the cloud to perform the calculations.

### VI. CONCLUSION

This work is an attempt to perform volume calculation for dynamic environments. There have been several attempts to estimating the entire geographical landscape. However, the existing methods are tedious, time-consuming and not relevant in dynamic environments and condition based monitoring systems such as an active mining location, or a disaster relief operation. Our method on the other hand, proposes a different approach to solving the problem which is applicable for dynamic environments as well. Our new algorithm uses deep learning and depth detection to calculate the volume of target object in a dynamic environment. The method can be applied to any object that has a planar base and does not
require any data pre-processing. Also, it undoubtedly brings down the time required to estimate volume from several hours to a few seconds as shown by our results, along with reducing human effort to almost zero making it an automated process and ensures independence from GPS for the task; enabling our methodology of calculating volume to be used in any environment. However, we also take into account that the volume measurement does not remain precise, owing to the hardware limitations such as IMU drift. Also the error rate seems to go up with the increase in volume. Hence, this method cannot be utilised to make precise measurement. However, in case of many condition monitoring systems in mines, farms, landfills and other dynamic areas it can be utilised well to do a volume estimation.

In future, we plan to work on making this approach better by increasing the accuracy and reducing the error rate while also trying to make it light weight. We also plan to extend it to even non-planar surface bases and objects with lateral hollowness as well as reduce the time taken for volume estimation.

REFERENCES


