Towards Quantifying the Amount of Uncollected Garbage through Image Analysis

Susheel Suresh*
susheel.suresh@gmail.com

Tarun Sharma*
tarunsharma.pes@gmail.com

Prashanth T.K.*
prashanth9c007@gmail.com

Subramaniam V
subramaniamkv@pes.edu

Dinkar Sitaram
dinkar.sitaram@gmail.com

Nirupama M
nirupamap@pes.edu

Department of Computer Science
P.E.S Institute of Technology
Bengaluru, India 560085

ABSTRACT

Civic authorities in many Indian cities have a tough time in garbage collection and as a result there is a pile up of garbage in the cities. In order to manage the situation, it is first required to be able to quantify the issue. In this paper, we address the problem of quantification of garbage in a dump using a two step approach. In the first step, we build a mobile application that allows citizens to capture images of garbage and upload them to a server. In the second step, back-end performs analysis on these images to estimate the amount of garbage using computer vision techniques. Our approach to volume estimation uses multiple images of the same dump (provided by the mobile application) from different perspectives, segments the dump from the background, reconstructs a three dimensional view of the dump and then estimates its volume. Using our novel pipeline, our experiments indicate that with 8 different perspectives, we are able to achieve an accuracy of about 85 % for estimating the volume.

CCS Concepts

•**Computing methodologies** → *Object recognition; Reconstruction;*

Keywords

Automatic Segmentation and 3D Reconstruction; Segmentation using Deep Learning, Structure from Motion, Volume Estimation of Garbage Dumps

1. INTRODUCTION

One of the most common problems faced by metropolitan and emerging cities, is the problem of garbage collection and disposal. In the past century, as the world's population has grown and become more urban and affluent, waste production has risen tenfold. By 2025 it will double again [15]. Even in the city Bangalore (India), the local municipal committee Bruhat Bengaluru Mahanagara Palike (BBMP) estimates that 3,500 tonnes of garbage is produced by the city everyday [1]. Despite considerable efforts, nearly 20 per cent of the waste still remains to be picked or is picked irregularly, giving the city a despicable look. Many media reports are presented on these issues and it is a very important topic not only for the civic authorities, but also for the citizens. As a result uncollected garbage leads to endemics and other problems for residents.

To perform image analysis and obtain meaningful insights, having a dataset of garbage images is necessary. Since there was no existing dataset of this kind, we have developed an android application for the purpose of crowd sourcing the task of data collection and asked users to take 8 GPS tagged images of a dump from different views going in a clockwise or anti clockwise manner around the accessible part of the garbage dump [1].

We propose a zero cost, citizen powered volumetric estimation of garbage using our novel pipeline. This would help the municipality devise an efficient collection route and also get an estimation of garbage distribution throughout the city. The pipeline has three main stages - Segmentation, 3D Reconstruction and Volume Estimation.

We use a state of the art convolutional neural network, AlexNet for segmentation. We also compare the results of two other approaches, sliding window edge thresholding, and sliding window classification using feedforward neural networks.

The segmented images are then used to generate a 3D model of the scene using concepts of Structure from Motion and Multi-view Stereo.

The whole system has been trained to estimate the volumes of complex structures extracted from their noisy environments. A high accuracy is not a strong prerequisite for this project because the volume of garbage heap will always

*These three authors contributed equally to the work.

[1] We plan to release the dataset once substantial data is collected in the near future

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ICVGIP, December 18-22, 2016, Guwahati, India
© 2016 ACM. ISBN 978-1-4503-4753-2/16/12…$15.00
DOI: http://dx.doi.org/10.1145/3009977.3010061

possess a slightly significant error due to the waste items being spread out on the pile. But an estimation of the surface area and volume of the pile will facilitate in indexing different dumps by their approximate volumes and help in formulation of optimised garbage collection routines, hence aiding the civic authorities .

Details of the segmentation methods are described in section 4, 3D reconstruction in section 5, and volume estimation in section 6. In section 7, we evaluate the results of our pipeline on measured volume.

2. RELATED WORK

Our work is related to several fields in computer vision:

- **Object recognition** i.e. Segmentation of garbage dumps from a scene.

  Traditional approaches to segmentation, include image clustering [25, 26]. Image regions are clustered based on pixel intensities using an algorithm like K-means. This methodology would not work in our case because the region of interest (garbage dump), has varying pixel intensities in random distributions (very high spatial frequency). Also the number of clusters could not be fixed because the background would keep changing for every dump. Machine learning is very popular for pattern recognition and neural networks are favourable since they learn relevant features on their own. Deep convolutional networks are being used for various tasks today [22, 29]. Long et. al. [23] have designed a fully connected convolutional neural network for per pixel semantic segmentation of objects in a scene. They used a training set of 8498 images from the PASCAL VOC dataset [11]. This approach of semantic segmentation would require a lot of images to train. We did not try this approach because of two reasons, first we did not have enough training data, and second, we found that our 3D reconstruction algorithm works better when we provide a little context around the garbage versus only the exact garbage boundary as would be in this case. The extra context provides extra feature points which are used for reconstruction. The bounding box method that we chose, is inspired from the work of Szegedy et. al. [33].

- **3D reconstruction** using techniques like structure from motion and multi-view stereo.

  Conventional approaches to 3D reconstruction are 3D scanning, and the use of depth/range cameras, but they are very costly and cumbersome to use in our case. On the contrary there are image based approaches like space carving [20], and structure from motion (SFM) [14] inspired from multi-view geometry. Space carving requires prior camera calibrated input images but as in our case, images are crowd sourced in real world setting and effective camera calibration is not achievable, thus space carving is not suitable. Thus SFM is much more promising for the problem at hand. Incremental SFM techniques [5, 32] have come a long way over the last few years and have been successfully used for the reconstruction of increasingly large photo collections, like building rome in a day [2]. With the advent of near linear time incremental SFM techniques [36], fast image matching and efficient bundle adjustment strategies, state of the art 3D reconstruction is possible.

- **Surface reconstruction** for delivering watertight surfaces which are used to find volume.

  Our pipeline shares similarities with a lot of fields in image-based modeling and metric estimation of surfaces. The first activity regarding live 3D reconstruction on mobile devices appeared in Wendel et al.[35] on a distributed framework with a micro air instrument variant. A shape-from-silhouette pipeline running in real time on a mobile phone was presented by Prisacariu et al. [28]. Despite the impressive performance, the method fails to cover the facet of generating manifold shapes, concavity capturing structures or 3D blanketing. 3D reconstruction of dense noisy samples was done before by Tanskanen et al. [34] where scaling was done by sensor tracking. Our project revolves around crude estimations of garbage heaps and as a result, an empirical scaling factor easily fits into the puzzle.

3. PROPOSED PIPELINE AND SYSTEM OVERVIEW

Here we introduce our novel pipeline for automatic segmentation and 3D reconstruction.

Almost all of the current methods, take a semi- automatic approach for finding the ROI for 3D modelling. Target cutout is done manually using Photoshop or Grabcut like tools to interactively segment the foreground object in these images. KinectFusion [16] also talks about real time segmentation and 3D reconstruction, but segmentation of interested objects in a scene is done by a user through direct interaction. When a user moves an object of interest, changes are detected in real-time, allowing the repositioned object to be cleanly segmented from the background model. This approach cannot be taken in case of static images and thus requires a robust segmentation algorithm.

In our case study the segmentation algorithm automatically finds the region of interest i.e. the garbage from a scene, as discussed earlier. Only while training the network, we need labelled data. Our method can be applied for all purposes where reconstructing is required only for specific objects in a scene.

Fig. 1 is an illustrative representation of our pipeline.

4. SEGMENTATION

Before the 3D reconstruction can be performed on the image, preprocessing, like segmenting out the garbage from the rest of the image is required. We have used a machine learning approach for this task. We use a bounding box regression approach using Alex Krizhevsky's AlexNet [19]. The results of two other approaches have also been compared to, segmentation using sliding windows and edge thresholding, and segmentation using sliding windows and fully connected neural networks. The results of segmentation were compared on a test set of 200 images and we found that the AlexNet performs the best. The results were compared using the intersection over union method, which is the standard metric to compare bounding box results in competitions like ILSVRC [30].

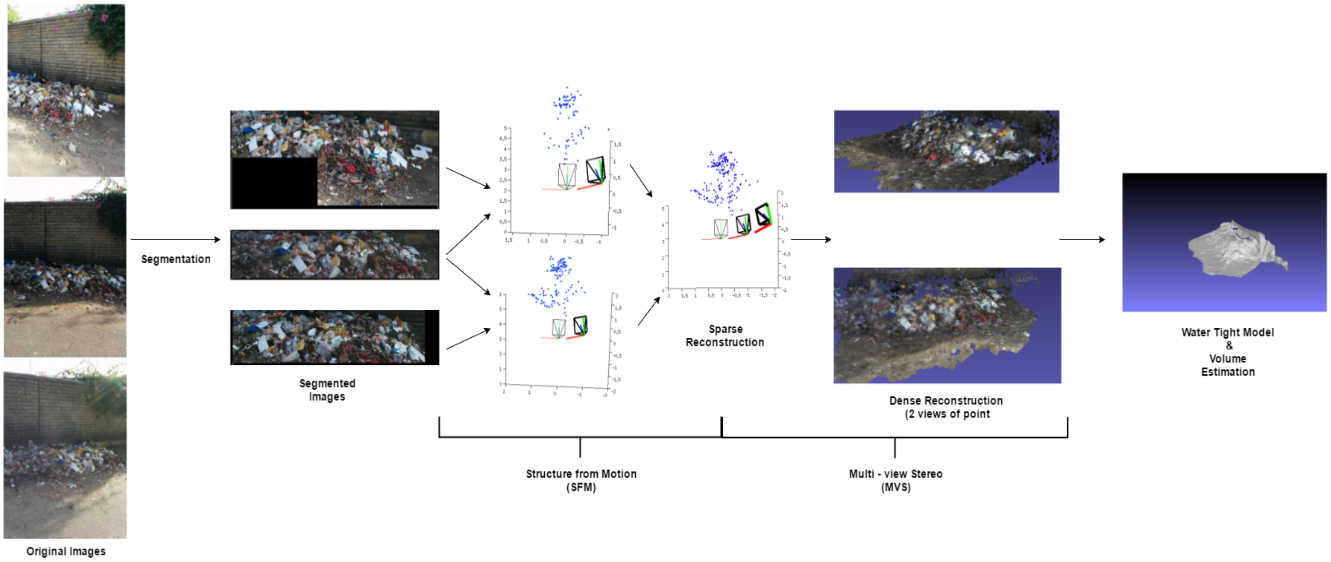
Segmentation
Segmented Images
Sparse Reconstruction
Dense Reconstruction (2 views of point
Water Tight Model & Volume Estimation
Structure from Motion (SFM)
Multi - view Stereo (MVS)
Original Images

Figure 1: Complete proposed pipeline

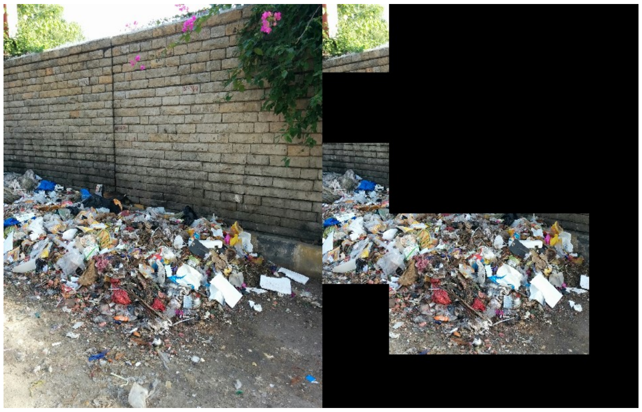

Figure 2: Results of segmentation using sliding window and edge thresholding approach. Left image is original image. Right is image after segmentation.

For the segmentation of the garbage from the rest of the background, the results of three different methods have been discussed in the following sections.

4.1 Method 1: Sliding window and edge thresholding

In this method, a 100 x 100 sliding window is moved over the entire image. In each grid, the number of edges and corners are calculated using Canny edge detector [6]. We used an open source image processing toolkit called OpenCV [4] for this task. Grids with number of edges greater than a certain threshold are retained while the other grids are discarded. The idea is that grids containing garbage, will have a lot of clutter, and hence have a higher spatial frequency. We hope to segment out the high spatial frequency region from the low spatial frequency region (background) by looking at the number of edges and corners. We found edges to be a better metric to compare instead of corners. The threshold was determined experimentally and was set to 25 edges. This method works quite well in certain scenarios as is shown in Fig. 2. Gaussian blur was used to smooth the image before counting edges. This method works well when the garbage is against a background such as a wall, but this method fails to perfectly segment out the garbage when there are complex objects in the background such as vehicles, trees, dogs, humans etc. The grids with these complex objects also exceed the edge threshold and hence give an improper segmentation of the garbage. It is because of these types of failures that we had decided to move to a machine learning approach.

4.2 Method 2 : Sliding window and fully connected neural network

In order to segment out only the garbage from a scene, we used a fully connected feedforward neural network. The neural network will learn the best features representing garbage instead of handcrafting features (like edge counts) manually. The same 100 x 100 grid is moved around the image but this time the grid is passed through a 3 layer neural network with two outputs, garbage or non garbage. Depending on the output of the network, the grid is retained or discarded. This is done for all the grids. For training the network, grids from images where garbage is present (garbage class) were manually extracted . All other grids were non garbage class. Data was augmented by flipping the grids horizontally, vertically and horizontally then vertically. We were able to extract a total of 1,900 grids containing garbage from a dataset of 450 images of garbage that we collected. After applying the transformations, the dataset consisted of 7,600 garbage grids and 18,000 non garbage grids from which 7,600 non garbage grids were randomly sampled. The neural network used consisted of 3 layers with an architecture of 1024, 512, 2. The grids were resized to 32 x 32 before being fed into the network. Mini batch stochastic gradient descent, with a batch of 64 and a learning rate of 0.1 was used. The activation function used in the first two layers was sigmoid and softmax in the last layer. Because the network was only 3 layers, we trained the network on an Intel 3rd gen core i7 CPU. The network was trained for 30 epochs. This took roughly one hour of training time. This method gave us satisfactory results as is shown in Fig. 3. We found that because of the sliding window approach used, the black boxes (no garbage) created by our segmentation was affecting the performance of the 3D reconstruction. It was due to this reason that we decided to move to a bounding box approach.

4.3 Method 3 : Bounding box segmentation using CNN

In this method, we train a state of the art deep convolutional neural network, AlexNet [30] on the task of predicting a bounding box around the garbage using regression. We used an open source framework called Caffe [17] for this task. Bounding boxes for around 500 images of garbage were manually drawn and the upper left x-coordinate, y-coordinate, width and height of the box were recorded. 300 of these were used for training and 150 for testing. We have used a modified version of AlexNet with 4 outputs and trained the network using Euclidean loss for regression. The loss function represented by E is shown below.

$$E = \frac{1}{2N}\sum_{i=1}^{N} ||x_i^1 - x_i^2||_2^2$$

Mini batch stochastic gradient descent with a batch size of 5 was used. Since the task is regression rather than classification, we use the ReLU activation function [9]. The input images were resized to 224 x 224 before being fed into the network. The network was fine tuned on weights from the ImageNet dataset [10] (which has 14 million images) as this is shown to give better performance in [27, 31, 21]. The network was trained for 180,000 iterations on an NVIDIA GeForce GTX Titan X GPU which has a memory of 12 GigaBytes. This took around 14 hours. The training loss graph against number of iterations is shown in Fig. 4. A learning rate of 0.001 and a weight decay of 0.1 was used. For validation, we use the intersection over union method to evaluate performance. This is the standard metric for bounding box localisation tasks on challenges such as ILSVRC [30]. This method checks if the ratio of intersected area to union area is above a certain threshold (0.5). After 180,000 iterations of training, we were able to get a mean intersection over union score of 0.82 on the validation set. We cannot use this metric for the first two approaches as they do not output bounding boxes. An example of the result on a testing images is shown in Fig. 5.

5. 3D RECONSTRUCTION

Given a short baseline image sequence I with n frames taken by a freely moving camera, parameters can be estimated reliably by the SFM techniques.

The set of camera parameters for frame t in an image sequence is denoted as $C = \{K, R, T\}$, where K is the intrinsic matrix, R is the rotation matrix, and T is the translation vector.

We use a typical incremental SFM system, where two view reconstructions are first estimated upon successful feature matching between two images, 3D models are then reconstructed by initializing from good two-view reconstructions, then repeatedly adding matched images, triangulating feature matches, and bundle-adjusting the structure and motion.

Our system employs the SFM method of Wu et al. [36]. We further improve the performance of it by:

- Sorting the images as an ordered sequence according to incremental image taken, and thereby reduce the computational cost from $O(n^2)$ to $O(n)$ in feature matching procedure. This can be done because images are taken in a sequence by a user and they are GPS tagged.

5.1 Implementation

- **Camera information gathering**: As focal-length is especially useful for camera recovery. We extract the focal length from the EXIF tags of a digital photo and to convert it to pixel units using the following formula.

$$F(px) = \frac{I(px) * F(mm)}{CCD(mm)}$$

  Where I is the image width (px), CCD is the sensor size (mm) in mobile phones and F is focal length (mm).

- **Feature Matching**: We first process the image sets using Wu et al.'s GPU implementation of SIFT [24].

- **Sparse Reconstruction**: Next we used a multi-core bundle adjustment algorithm [37] to generate a sparse 3D reconstruction using structure from motion, along with camera calibration parameters for each image. These parameters are a focal length f, 3 x 3 camera rotation matrix R, and 3-vector camera translation t.

- **Dense Reconstruction**: For the final leg we apply Furukawa and Ponce's PMVS algorithm [12, 13] to generate a dense point reconstruction, The PMVS algorithm is considered state-of-the-art in the area of dense reconstruction, and performs very well on even highly unstructured image sets containing variations in lighting, image exposure, lens type, etc.

The reconstruction is performed up to an arbitrary scale, so the distances in the resulting object space do not correspond to the true distances in real-life space, section 6 which is also part of our pipeline is devoted to find the exact metric measurements and then compute volume.

Many sets of segmented garbage images , each with 8 images in each set and some toy example images had to be reconstructed to fine tune the scaling factor. Table 1 gives the time taken for each task. All images have a size resized to 500 x 500 pixels after segmentation.

The complete 3D Reconstruction was performed on a system with 4th Gen Intel Core i7 chip (4 cores clocked @ 2.4GHz), 8 GB DDR3L SDRAM @ 1600 MHz and NVIDIA's GeForce GT750M GPU of 2GB memory.

6. SURFACE RECONSTRUCTION

The process of obtaining reliable manifold surfaces out of point sets by triangulation has never proven to provide results which have conformed to the 3D modeling on a metric scale. Many parallel experiments were conducted in a setting which resembled real-world conditions to test the generation of robust 3d models: A miniature clay idol of the Hindu deity Krishna, and a plastic box with rounded corners that holds a bar of soap, a globe, a cubical cardboard box etc. The process of surface reconstruction had to be executed sans a

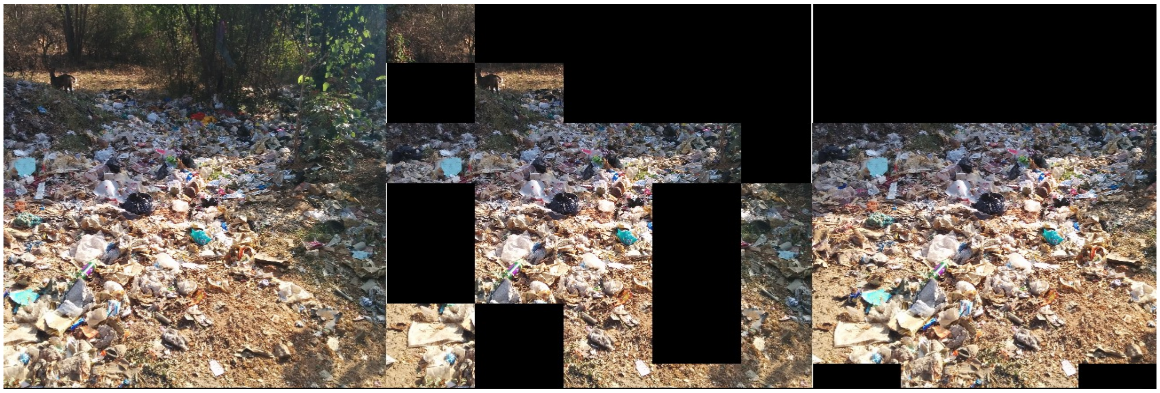

Figure 3: Comparison of results of segmentation using sliding windows with edge thresholding vs fully connected neural network. Left is original image. Center is segmentation result using edge thresholding. Right is segmentation result using fully connected neural network.

Name of Experiment	Time: Feature Matching (s)	Time: Sparse + Dense Reconstruction (s)	Total Time (s)
Garbage Dump 1	1.67	9.86	11.53
Garbage Dump 2	1.83	10.20	12.03
Garbage Dump 3	2.11	12.45	14.56
Krishna Idol (Toy Example)	2.47	15.36	17.83

Table 1: Total time taken for 3D reconstruction

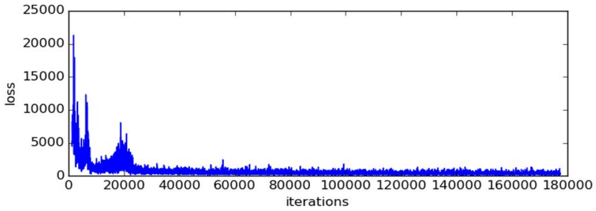

Figure 4: Training loss vs number of iterations for training of bounding box segmentation using AlexNet.

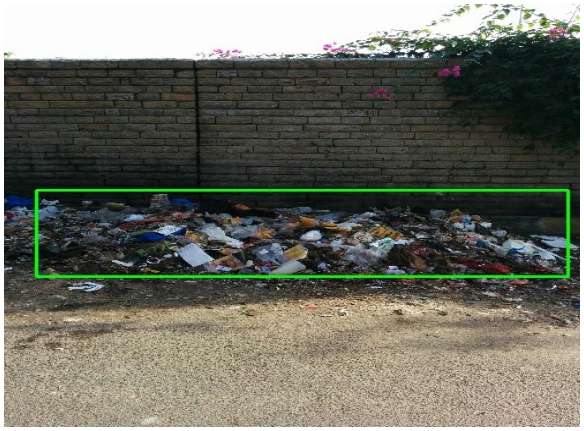

Figure 5: Results of segmentation using bounding box regression on AlexNet on validation image.

360 degree capture of the scene, nor a reliable camera used for studio purposes.

For all the trials, a variety of mobile phone cameras had to be used in order to replicate the conditions of the target application which involves citizens using phones to capture the heap. The user also had to walk around the objects of interest, roughly estimating positions for suitable photographs. 8 images were taken for each trial, where an angle of approximately 180 degrees of the anterior scene objects were spanned. After the point clouds were generated, we had to create a manifold mesh around the cloud to obtain a watertight object whose volume could then be measured.

6.1 Poisson Disk Sampling

The resulting point clouds from the reconstruction experimentation resulted in being rarified with a significant amount of noise . Surface reconstruction shows quicker performances for smaller samples, thus leading to a filtering for subsamples at a ratio of roughly $1 : 6^{th}$ the number of points present in the point cloud, to ease computation. In effect, the samples are randomly placed with the restriction that no two samples are closer together than a certain distance. The Poisson Disk Sampling method was seen to be the best fit for our point cloud as it has a history of providing evenly spaced subsamples for non-robust point clouds.

A prior computation of normals for each point was a requisite for meshing in the primitive method. Since this is an experiment that does not require tailored meshes of extreme precision, the extrapolated normals could be calculated without exploiting triangle connectivity, with a high number of nearest neighbours of 500. The viewing position had to be taken as -2 to develop flipped normals aligned outward from the scene, which were one of the very few orientations that resulted in a manifold mesh around the noisy cloud.

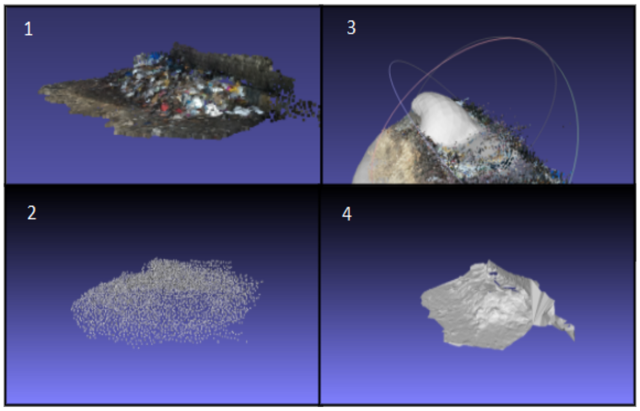

Figure 6: Results of surface reconstruction. 1 is the point cloud from 3D reconstruction, 2 is the poisson disk samples, 3 and 4 are poisson and ball pivot surface reconstructions respectively.

The Poisson subsampling described in [8] was performed with the help of the open source Meshlab tool [7].

6.2 Poisson Surface Reconstruction

Since the scene was poorly captured a result of the real-world imitation , the resulting point cloud poisson disk samples had a high number of holes, non-manifold faces and edges and considerably large cavities in the structure as seen in Fig.6, which disallows the generation of a watertight mesh unlike a point cloud generated in laboratory conditions.

Poisson Surface Reconstruction [18] is a technique which reconstructs the implicit function f whose value is zero at the points p_i and whose gradient at the points p_i equals the normal vectors $n_i$for a subsample.

The choice of Octree Depth was taken to be as an arbitrarily low number mainly because a very high accuracy in the measurement of volume of heaps is not a glaring objective, hardly an obtaining of volume in the required units would suffice, and also because a higher depth would result in a very long duration of mesh generation.

6.3 Ball Pivot Point Surface Reconstruction

This was an advancement to the previous method as it uses a much lighter algorithm called Ball-Pivot Point [3] with subsequent interpolations. The principle is very simple: Three points form a triangle if a ball of a user-specified radius p touches them without containing any other point. Starting with a seed triangle, the ball pivots around an edge (i.e., it revolves around the edge while keeping in contact with the edge's endpoints) until it touches another point, forming another triangle. The process continues until all reachable edges have been tried, and then starts from another seed triangle, until all points have been considered. The process can then be repeated with a ball of larger radius to handle uneven sampling densities.

After the summing the time taken by the tedious normal computation and Poisson Surface reconstruction as T_{nc+ps} and comparing it with the direct BP as T_{bp} , Table 2 clearly shows that BP is more optimized in performance. This is because the algorithm runs in linear time, linear in the number of sample points and linear storage in comparison to the log-linear system of the Poisson system .

The surface reconstruction stage was carried out on a 6^{th} Generation Intel Core i5 Processor, an 8GB, DDR3L RAM @ 1600 MHz and AMD Radeon R5 M335 4GB GPU chip.

7. VALIDATION AND RESULTS

The volume of each garbage dump was measured while the image dataset was being created. When we were creating the image dataset, we manually measured the approximate enclosure boundary volume (length, breadth and height) of each garbage dump. There is significant room for error in the experiments since one would never be able to truly determine the volume of trash at a dump until the waste material was compressed and measured. For example, our model cannot estimate the volume of trash concealed by the dump below the surface level. The target is to estimate the volume of the imaginary enclosure the garbage dump would lie in as it would at least serve as an index for further big data analytics as there is no big data and image analysis done regarding exposed garbage dumps. Although the enclosure volume may seem larger than the actual waste volume, residual waste material which would be segmented out in the first stage would slightly compensate the gap. The scaling factor of 0.229 was obtained empirically after thorough experimental testing on more than 30 garbage dumps of various shapes and sizes , each located in different locations and environments (See Table 3 for sample results).

7.1 Comparative Study Results

Here, we give results of our comprehensive study which backs our assumption to choose certain methods over others in the different stages of our pipeline.

We chose a garbage dump in Turahalli (a small commune in Bangalore) to find the percent error, between the actual and computed volume using different methods from the same stage. Three different methods were considered for study in the segmentation/detection stage namely, sliding window with edge thresholding, with neural networks and bounding box using CNN and two methods for water tight surface reconstruction were examined viz. poisson and ball-pivot. The actual volume of the dump was **2.6721** m^3.

A high error of 24.30% was achieved when edge thresholding and poisson reconstruction were used. It dropped to 22.13% when the same was accompanied with ball-pivot instead of poisson reconstruction. We attained lower error percentage values of 19.32 and 16.19 when we used sliding window with neural networks in our first stage. The former was followed up with poisson and later with ball-pivot. Shifting gears to the bounding box approach with CNN features in the initial stage, we arrived at the lowest error percentages of 14.61 and 12.43, with poisson and ball pivot respectively.

However this system is not universal enough to estimate the volume of any object, as the pipeline is complex structured object specific.

8. CONCLUSIONS

We have presented a methodical pipeline for estimating the volume of complex random structures like garbage dumps and heaps using segmentation and dense stereo-based 3D reconstruction in metric units. In order to address the major challenges posed by the underlying hardware limitations and to meet the robustness and context specific requirements of

No. of Vertices in Point Cloud	Normal Computation Time: T_{nc} (ms)	Poisson Surface Reconstruction Time: T_{ps} (ms)	T_{nc+ps} (ms)	Ball Point Pivoting Reconstruction Time: T_{bp} (ms)
Garbage Dump (466)	125	379	504	99
Toy Doll (2262)	78	545	623	111
Globe (167)	70	315	385	66
Table Clock (2444)	88	588	676	106
Cardboard Box (893)	79	405	484	94

Table 2: Relationship between computation time and number of subsamples with regard to surface reconstruction techniques

Location with Coordinates of the Dump	Measured actual volume (m^3)	Computed volume from pipeline (m^3)	Percent Error (%)	Total Execution Time (s) Segmentation+ Reconstruction+Volume estimation
Domlur (12.958688, 77.637766)	1.198	0.999	16.61	2.2 + 243 + 1.5 = 246.7
BTM Layout (12.917430, 77.602048)	3.442	3.029	11.99	2.4 + 257 + 1.7 = 261.1
Marathahalli (12.962009, 77.694047)	0.855	0.982	14.85	2.1 + 248 + 1.4 = 251.5
Nayandanahalli (12.941958, 77.524358)	1.402	1.119	20.18	2.7 + 261 + 1.7 = 265.4

Table 3: Results of Garbage Volume Estimation

the application, we integrated multiple novel solutions. We put forth a machine learning algorithm to extract an object, in this case a pile of garbage from the surrounding environment. Using modules that leverage the GPU the system for the highly powerful Structure-from-Motion process, we have accelerated the scene reconstruction. Additionally, we have explored and compared the different methods used to reconstruct and mesh a garbage heap in terms of efficiency because the task of surface reconstruction has to be decided based on the pattern followed by the point clouds generated. The pipeline was tested in both indoor and external settings and mainly garbage heaps in Bangalore city. Moreover, we have provided a method for generation of a dataset of garbage dumps tagged with volume values, hence opening doors to more big data analytics, image analysis and primarily the cleansing of the city.

9. ACKNOWLEDGMENTS

This work was funded by the Cloud Computing and Big Data Lab at P.E.S University. We also acknowledge the enthusiastic participation of engineering students from the university who assisted in taking photographs of many garbage dumps in the city and contributing to the dataset.

10. REFERENCES

[1] BBMP Statistics: Solid Waste Management Cell. http://bbmp.gov.in/documents/10180/512162/City+Statistics+New+Microsoft+Office+Word+Document.pdf/148f685d-58cd-402c-9c5c-bccb344eda2d. Accessed: 15-07-2016.

[2] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building rome in a day. In *2009 IEEE 12th international conference on computer vision*, pages 72–79. IEEE, 2009.

[3] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE transactions on visualization and computer graphics*, 5(4):349–359, 1999.

[4] G. Bradski. Opencv. *Dr. Dobb's Journal of Software Tools*, 2000.

[5] M. Brown and D. G. Lowe. Unsupervised 3d object recognition and reconstruction in unordered datasets. In *Fifth International Conference on 3-D Digital Imaging and Modeling (3DIM'05)*, pages 56–63. IEEE, 2005.

[6] J. Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

[7] P. Cignoni, M. Corsini, and G. Ranzuglia. Meshlab: an open-source 3d mesh processing system. *Ercim news*, 73(45-46):6, 2008.

[8] M. Corsini, P. Cignoni, and R. Scopigno. Efficient and flexible sampling with blue noise properties of triangular meshes. *IEEE Transactions on Visualization and Computer Graphics*, 18(6):914–924, 2012.

[9] G. E. Dahl, T. N. Sainath, and G. E. Hinton. Improving deep neural networks for lvcsr using rectified linear units and dropout. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8609–8613. IEEE, 2013.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[11] M. Everingham and J. Winn. The pascal visual object classes challenge 2011 (voc2011) development kit. *Pattern Analysis, Statistical Modelling and Computational Learning, Tech. Rep*, 2011.

[12] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Towards internet-scale multi-view stereo. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1434–1441. IEEE, 2010.

[13] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on*

pattern analysis and machine intelligence, 32(8):1362–1376, 2010.

[14] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[15] D. Hoornweg and P. Bhada-Tata. What a waste: a global review of solid waste management. *Urban development series knowledge papers*, 15:1–98, 2012.

[16] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568. ACM, 2011.

[17] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

[18] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, 2006.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[20] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):199–218, 2000.

[21] B. Li, C. Shen, Y. Dai, A. van den Hengel, and M. He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1119–1127, 2015.

[22] F. Liu, C. Shen, and G. Lin. Deep convolutional neural fields for depth estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5162–5170, 2015.

[23] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[24] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[25] D. Naik and P. Shah. A review on image segmentation clustering algorithms. *Int J Comput Sci Inform Technol*, 5(3):3289–93, 2014.

[26] A. Oliver, X. Munoz, J. Batlle, L. Pacheco, and J. Freixenet. Improving clustering algorithms for image segmentation using contour and region information. In *2006 IEEE International Conference on Automation, Quality and Testing, Robotics*, volume 2, pages 315–320. IEEE, 2006.

[27] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1717–1724, 2014.

[28] V. A. Prisacariu, O. Kähler, D. W. Murray, and I. D. Reid. Simultaneous 3d tracking and reconstruction on a mobile phone. In *Mixed and Augmented Reality (ISMAR), 2013 IEEE International Symposium on*, pages 89–98. IEEE, 2013.

[29] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[30] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[31] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, 35(5):1285–1298, 2016.

[32] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM transactions on graphics (TOG)*, volume 25, pages 835–846. ACM, 2006.

[33] C. Szegedy, A. Toshev, and D. Erhan. Deep neural networks for object detection. In *Advances in Neural Information Processing Systems*, pages 2553–2561, 2013.

[34] P. Tanskanen, K. Kolev, L. Meier, F. Camposeco, O. Saurer, and M. Pollefeys. Live metric 3d reconstruction on mobile phones. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 65–72, 2013.

[35] A. Wendel, M. Maurer, G. Graber, T. Pock, and H. Bischof. Dense reconstruction on-the-fly. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1450–1457. IEEE, 2012.

[36] C. Wu. Towards linear-time incremental structure from motion. In *2013 International Conference on 3D Vision-3DV 2013*, pages 127–134. IEEE, 2013.

[37] C. Wu, S. Agarwal, B. Curless, and S. M. Seitz. Multicore bundle adjustment. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3057–3064. IEEE, 2011.