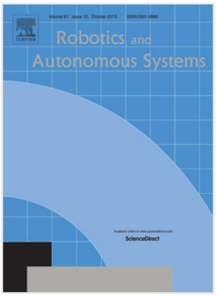


3DCOMET: 3D compression methods test dataset

Javier Navarrete, Vicente Morell, Miguel Cazorla, Diego Viejo, José García-Rodríguez, Sergio Orts-Escolano

PII: S0921-8890(15)00223-7
DOI: http://dx.doi.org/10.1016/j.robot.2015.09.028
Reference: ROBOT 2556

To appear in: *Robotics and Autonomous Systems*

Please cite this article as: J. Navarrete, V. Morell, M. Cazorla, D. Viejo, J. García-Rodríguez, S. Orts-Escolano, 3DCOMET: 3D compression methods test dataset, *Robotics and Autonomous Systems* (2015), http://dx.doi.org/10.1016/j.robot.2015.09.028

HIGHLIGHTS

- **Dataset useful for testing 3D compression methods.**
- **Contains more than 100 3D pointclouds with different texture and structure**
- **Several tools for showing comparatives between methods.**
- **Baseline information from state-of-the-arts compression methods.**

3DCOMET: 3D Compression Methods Test Dataset

Javier Navarrete, Vicente Morell, Miguel Cazorla*, Diego Viejo, José García-Rodríguez, Sergio Orts-Escolano

Institute for Computer Science Research, University of Alicante., P.O. Box 99. 03080, Alicante, Spain.

Abstract

The use of 3D data in mobile robotics applications provides valuable information about the robot's environment. However usually the huge amount of 3D information is difficult to manage due to the fact that the robot storage system and computing capabilities are insufficient. Therefore, a data compression method is necessary to store and process this information while preserving as much information as possible.

A few methods have been proposed to compress 3D information. Nevertheless, there does not exist a consistent public benchmark for comparing the results (compression level, distance reconstructed error, etc.) obtained with different methods. In this paper, we propose a dataset composed of a set of 3D point clouds with different structure and texture variability to evaluate the results obtained from 3D data compression methods. We also provide useful tools for comparing compression methods, using as a baseline the results obtained by existing relevant compression methods.

Keywords: 3D data, Data compression, Dataset

1. Introduction

In the recent years, the number of applications using 3D data processing has considerably increased due to the appearance of cheap 3D sensors such as Microsoft Kinect. This 3D information, usually provided as a set of 3D points obtained by 3D cameras is useful in many applications, such as medicine,

*Corresponding author
Email address: miguel.cazorla@ua.es (Miguel Cazorla)

entertainment industry, robotics, and many others. However, it is difficult to both manage the huge amount of data obtained by these cameras for common 3D application and just store it for further processing. In the present paper we describe a dataset that allows 3D point compression, evaluation and benchmarcking.

There exists a bunch of datasets for 3D applications. To evaluate current challenging computer vision problems we can find specific datasets for object detection and recognition, simultaneous localization and mapping (SLAM), human behavior recognition, gesture recognition, etc.

In [1], a dataset for human grasping behavior evaluation in unstructured environments is presented. Wide-angle head-mounted camera videos are recorded by two housekeepers and two machinists during their regular work activities, and the grasp types, objects, and tasks are analyzed. The full dataset represents a wide range of grasping action behaviors spanning much of the typical human hand usage.

For robotic algorithms, a huge amount of training data containing both people and environment is required. The dataset presented in [2] is specifically designed for training and testing algorithms for people detection in indoor environments. [3] provides a large dataset containing RGB-D and ground-truth data aiming to establish a novel benchmark for the evaluation of visual odometry and visual Simultaneous Localization And Mapping (SLAM) systems.

Several datasets are focused on object recognition. [4] includes over 50 classes of color and depth image pairs, gathered in real domestic and office environments. The 3D Table Top Object Dataset [5] has three categories (mouse, mug and stapler) and provides 200 test images with cluttered backgrounds. [6] is a large labeled dataset containing hundreds of objects and more than fifty categories. The main objective of this work was to introduce some techniques for RGB-D based object recognition demonstrating that the combination of color and depth information, leads to a substantial improvement of recognition performance.

A general purpose dataset for 3D computer vision tasks is presented in [7]. It could be used to benchmark multiple problems, such as 3D mesh reconstruction (with and without RGB-D), object instance recognition, and object categorization. The authors had invited dataset users to contribute adding new objects with the aim of increasing the dataset size continuously, providing new test scenes and results through the authors' website. Moreover, they present a method to calibrate a multi-camera system and many other modules.

However, to the best of our knowledge, there does not exist a dataset that allows 3D point cloud compression evaluation and benchmarking. In this paper, we propose a new dataset specifically designed for testing 3D compression methods. It is composed of a comprehensive set of both real and synthetic 3D images. The dataset also includes several methods for the automation of the testing task. It also allows to perform the test under different combinations of texture and structure, which are the aspects that most affect compression methods. Furthermore, we include some statistic tools for the analysis of the obtained results. (compression ratio, mean error, and many others).

All the data is available online under the Creative Commons Attribution license (CC-BY 3.0) at

http://www.rovit.ua.es/dataset/3dcomet/

The rest of the paper is organized as follows: Section 2 contains a brief review of common 3D compression methods. Section 3 explains the motivation of the dataset. In Section 4 we describe the dataset. Then, Section 4 introduces the provided evaluation and comparison tools for 3D compression methods. Finally, Section 6 draws the conclusions and directions for future work.

2. 3D Data Compression

3D data compression increases the bandwidth available for data transfer and allows to reduce physical storage. Both features are desirable in robotics applications, specially for real time applications.

There are several classifications of compression methods based on loss of information, codification or point clustering. In a classification based on loss of information, there are two types of methods: lossy and lossless. Once a point cloud has been compressed and decompressed, lossless methods return the same original point cloud without any error in point color or coordinates. On the other hand, lossy methods usually return point clouds with some errors.

Some lossy methods are based on space organization, using different structures to address data compression i.e. a hexagonal grid [8], or an *octree* [9]. After that, an encoding process is applied to these structures in order to minimize redundancy. Changes between different frames are detected in a real time sequence, and this allows to encode the information that has been changed, so that it is not necessary to transmit or store all information. Spatial division is parametrized so that the level of precision-compression can vary and in this way the execution time can be controlled.

Lossless methods preserve all the information of the point cloud. In [10], an octree is also used as an element to represent space, and encoding is used to reduce information size. Using the same technique, [11] shows how it is possible to encode point clouds with more than 1 billion of points. Moreover, with an efficient memory implementation, it is also possible to use the compressed dataset as input for other high level algorithms such as RANSAC.

Codification based compression methods take advantage of the information redundancy which is commonly found in large streams of data. As an example of this kind of techniques, general compression algorithms such as the well known [12] or [13] can be outlined. General data compression is a well established field of research and we can find highly optimized implementations. Nevertheless, it is difficult to use a general compression method under specific time requirements. Moreover, it is necessary to decompress the data before using it, which represents another disadvantage for this kind of methods. Furthermore, datasets in which information redundancy is minimal, can be hardly compressed by lossless algorithms.

Assuming a certain loss of information, we can find in the literature some techniques based on point clustering using geometric considerations. [14] uses the eigenvalues from the whole point cloud to extract curvature information, detecting repetitions and removing duplicities. [15] makes a plane extraction, concave hull and Delaunay triangulation to replace coplanar points by planar patches. [16] combines normal saliency with the point cloud structure in an octree.

This kind of algorithms are highly dependent on point cloud structure to get great compression ratios. The more structure the point cloud has, the better compression ratio is obtained. In contrast, for point clouds in which data are mainly unstructured, the compression ratio obtained is low. Furthermore, the presence of noise in the data can affect both the compression rate and accuracy of these methods. Nevertheless, as those methods perform structure extraction and point clustering, the compressed data obtained can be used for other processing methods like data registration. It is also possible to increase the compression ratio by applying an encoding method to the already compressed data set. Following this approach, in [17] a curve model prediction is applied to compress input data. After that, a codification method is used for reducing the transmission bit-rate.

3. Motivation

Nowadays, there is an increasing interest in methods that reduce the size of a 3D data set. These methods can be used to improve the transfer rate of the datasets or to reduce the disk space required to store them. Therefore, it is interesting to know how existing 3D compression methods perform under different conditions, as it is important to have a mechanism to compare those methods to the ones that may appear in the near future. Nevertheless, to the best of our knowledge, there does not exists yet a standard dataset that could be used to establish the strengths and weaknesses of each approach. Untill now, each of the proposed method uses its own dataset to test its throughput and to make comparisons between methods. In order to make this task easier, we present in this paper a comprehensive data set for 3D compression methods. It also includes all the required tools to test and compare those methods. Finally, four baseline methods are provided and evaluated showing the obtained results.

4. Dataset Description

The dataset includes real 3D images, obtained from a RGB-D sensor like the Kinect [18] camera or the Primesense carmine 1.09 short range sensor, along with synthetic images generated from 3D virtual models. Real images allow to test and compare 3D compression methods in normal working conditions whereas synthetic images give us a ground truth that allows to obtain quantitative results for each tested method.

We have used a quantitative and qualitative approach to classify images in the dataset. The qualitative approach organizes 3D images following the two main criteria that affect the performance of 3D data compression methods: the level of texture present in the surfaces of the scene and its structure level.

In this way we differentiate three categories according to the texture level: 3D images that present plain or low texture, those with a medium texture detail and, finally, scenes that are highly textured. Regarding the scene structure we differentiate: a first level of point clouds highly structured in which most of the objects are formed by simple geometric forms like planes;

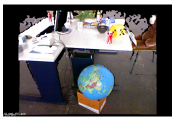
Texture Structure	High	Medium	Low
High			
Medium			
Low			

Table 1: Dataset examples of real data.

point clouds with a medium level of structure in which the unstructured objects, like trees, bushes, etc., present in the scenes are in the same proportion than the structured ones; finally, a category of images mainly formed by unstructured objects.

The qualitative approach is supported by quantitative values calculated from the point clouds. We propose to use two measures, one for structure and one for texture. For the structure, we have included the curvature metric. For a given a 3D point, the eigenvalues of the covariance matrix of a set of 3D point neighbors are calculated. This covariance matrix is the same as the one used for normal calculation. The curvature for this point is obtained using the formula:

$$c = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2} \tag{1}$$

where λ_i are the eigenvalues of the covariance matrix and λ_0 is the smallest one. The curvature for a point cloud is the mean value for all the 3D points and is returned when the normals of a 3D point cloud are calculated with the PCL library. Note that a low

Table 2: Curvature data calculations of structure and texture categories

Structure	Real	Synthetic	Texture	Real	Synthetic
Low	0,060	0,073	Low	6,78	3,05
Medium	0,043	0,023	Medium	7,23	6,33
High	0,017	0,018	High	7,33	6,83

curvature value indicates high structure, having in mind that we define structure as the presence of planes.

For texture analysis, we have included the entropy metric. The entropy of an image is a statistical measure of randomness that can be used to characterize the texture of the input image. Entropy is defined as:

$$F_{entropy}(I) = -p * \log(p) \tag{2}$$

where p **contains the histogram counts**

Table 2 shows the mean values of the different categories for the quantitative measures. This demonstrates the reliability of the qualitative classification.

The dataset contains a total of 102 point clouds in PCD (Point Cloud Data) format. For each type of data (real or synthetic) we then subdivide it into three new levels or categories of structure appearance which represent high, medium or low structure presence. Then for each structure category we subdivide it into three levels or categories of texture appearance. Each category consists of five different point clouds in order to add some variability to the dataset. Finally, we add a *special* group in order to include a more challenging set of point clouds. These special point clouds are divided into real and synthetic with six examples of each category.

Tables 1 and 3 show examples of each category of structure/texture of the real and synthetic point clouds respectively. Table 4 shows two examples of the special category of both real and synthetic.

We used different devices and tools to obtain all the point clouds. Eight of the provided point clouds came from the TUM-RGBD-Dataset [3]. We acquired and generated the rest of the data using different technologies and devices. Basically, we acquired the dataset using a Primesense Carmine 1.09 and a Microsoft Kinect (v1). For the synthetic point clouds, we used our own application which works with different models and can be applied to different

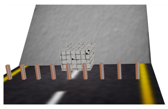
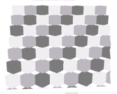
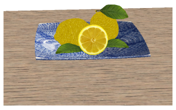
Texture Structure	High	Medium	Low
High			
Medium			
Low			

Table 3: Dataset examples of synthetic data.

textures. Using a Raycasting method, we simulated a depth-sensor avoiding acquisition errors (outliers, noise) and obtaining ground truth data for later validation. **The capture of synthetic 3D information is made using Java3D. We use different loaders to introduce objects in a scene that are positioned to create scenes. Primitives as cubes, spheres an planes are also used. In our case we use different scenes to create artificially the different degrees of structure in the dataset. For each created scene we use PickSegment objects to simulate beams according to a pinhole camera model, and with PickIntersection and PickResult we get the intersection point coordinates and its rob color. We fixed some restrictions in geometry of the virtual camera to approximate Kinect parameters: horizontal opening 57, vertical 43, resolution of 640x480 pixels and maximum distance. Java3D features allow textures and colors to be applied to objects. Thus we can compose scene variations and evaluate the effect of textures in compression methods.**

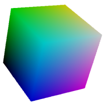
Real	Synthetic
	

Table 4: Examples of point clouds of the special category.

5. Tools of the Dataset

We have included two tools in our dataset in order to facilitate the analysis and evaluation of different 3D compression methods. The first tool is the statistics generator. This is a an application that analyses the results of a given method, developed in C++ using the PointCloud library[19]. The structure for each test is organized in a directory with three subdirectories: original, compressed and decompressed data. These three subdirectories must keep the original structure. The original data are the images in PCD binary (uncompressed) format. Then, from the compressed data, each file is compared with the original one, obtaining the compression ratio (see Equation 3), which is the ratio between the compressed file size and the original one. From the uncompressed subdirectory and for each file, the Root Mean Square (RMS) error for distance and color are calculated. These errors are obtained by searching for each point in the uncompressed file the closest point in the original file. The result is stored in a text file where each line contains the name of the file, the compression ratio, the mean metric distance and the mean color distance. This is due to lower color variance and therefore the color distance between the original and the compressed color is lower.

$$\text{Compression rate} = \frac{\text{Compressed size}}{\text{Original size}} * 100 \tag{3}$$

The second tool is a Gnuplot script which shows plots from the files generated with the previous tool. It takes one or more files containing compression rates, distance and color errors for each file in the dataset. The script has several input parameters. The first one is a string with the input files. Up to 6 files can be given to the script. If no other parameter is selected, the obtained output is similar to the one showed in Figure 1, three plots with the mean and standard deviation of all the files in the dataset, grouped by real, synthetic and special data.

Figure 1 shows the results obtained using two specific 3D compression methods available in the PointCloud library and two general compression methods provided along with most of the compression utilities like 7Zip. We have also added another geometric compression method. Regarding the methods implemented in the PointCloud library, we have used the octree based compression [9] using the 24 bits version (we have not found differences in the 8 and 32 bits versions) and the PCD binary compression (Marc Lehmann's LZF algorithm, an implementation can be found in the LibLZF library [1]) methods. The first one is a lossy method and the second is a lossless one. We have also included another compression method implemented with the PCL library [15], which is based on point grouping. This method has some parameters that need to be tuned to obtain the best results. We show two executions of this method changing the k parameter, used for applying the Kmeans method for segmentation: Morell2014k1 is for k=1 and Morell2014k5 for k=5. Moreover, we have used two file compression methods (both lossless): the gzip or LZ77 method (variation of [12]) and the bzip2 or Burrows method [13].

The results in Figure 1 are organized in four categories (the qualitative classification is explained above). The Complete category contains all the dataset: Synthetic (all the point clouds generated synthetically); Real (the point clouds taken with the Kinect camera) and, finally, the Special category (composed of several point clouds with special configurations different from the previous categories). An example of special category is one point cloud with a map reconstruction of 360 degrees. Inside each category and for each value (compression rate, distance and color error) the mean value and the standard deviation of all point clouds in that category is presented.

[1] http://software.schmorp.de/pkg/liblzf

In this first result, the Morell2014k1 is the one which provides the best compression rates but the texture error is too high. This is due to the fact that with k=1 the method is not able to reconstruct the texture of the point cloud as all the colors of a plane are replaced by one single color. With k=5 the method is not able to compress as much as with k=1, but the texture error is lower. However, the Octree method provides a good compression rate (below the lossless methods) and the distance and texture errors are insignificant.

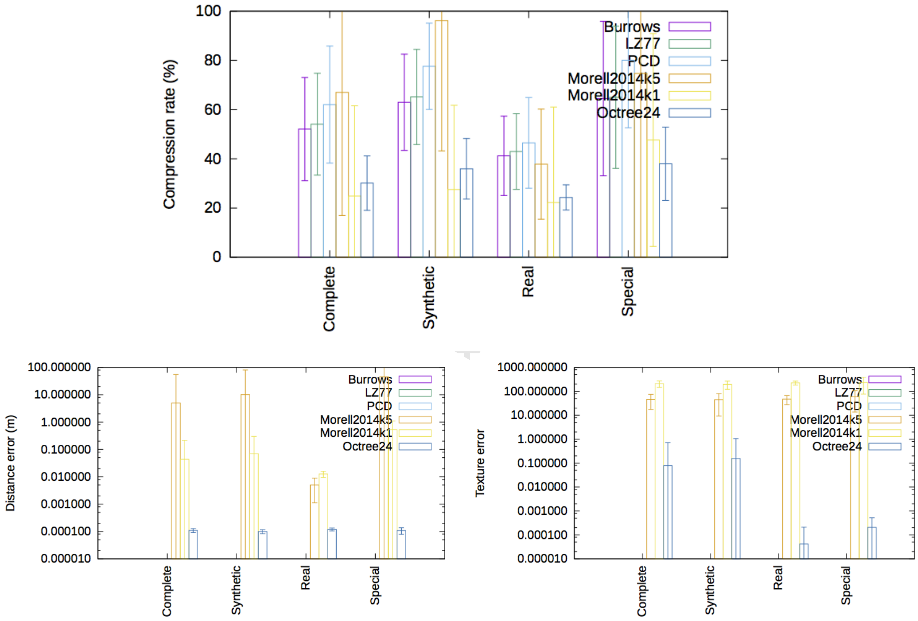


Figure 1: Comparison of the tested 3D compression methods with the whole dataset.

A second input parameter can be selected in order to show a desired information: for example, results from the complete dataset with respect to different levels of structure, or results using only the synthetic data with respect to different levels of texture (subset). This last option is shown in Figure 2. Different combinations are allowed for evaluating the complete, real and synthetic data, texture and structure levels as different subsets. In this case, a similar procedure used for plots is shown in Figure 1. For example, all the point clouds with high texture are included in the category Texture High and the mean and standard deviation is presented. In this case, the behavior of the Morell2014 and Octree method is similar to the previous one. It has to be outlined that with a low texture, the Octree method is able to get a better compression rate and the texture error is lower than for higher textures.

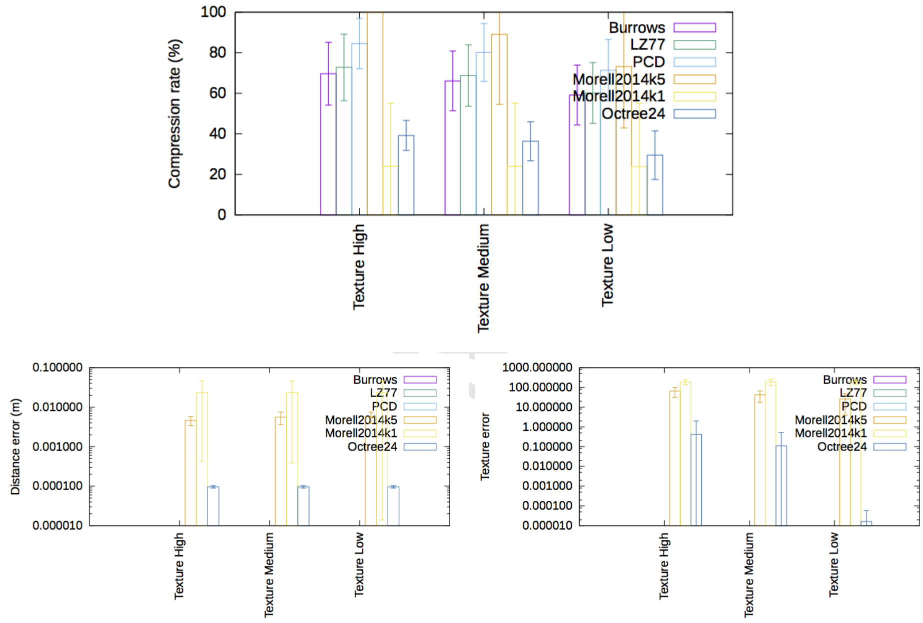


Figure 2: Comparison of the tested 3D compression methods using synthetic data and showing different levels of texture.

As shown in Figures 1 and 2, binary methods (PCD, LZ77 and Burrows) provide a reasonable compression rate (over 60%). The comparison of only these 3 methods shows that, although PCD (the compression method provided by the PCL library) gives worse compression rates than the other two, it is worth using it as it is included in the PCL and does not require the use of an external method (from operating system). Obviously, as it is a lossless method it does not have neither distance nor color error in the reconstructed point clouds. With real images, these methods are able to get a better compression rate. This is due to the fact that synthetic images contain much less 3D points (a real point cloud could have more than 300.000 3D points and a synthetic one only a few thousands) and thus the file size for synthetic images is smaller and the compression rate is lower.

Geometric methods (Octree and Morell2014) are able to get a better compression rate as they make use of geometric information. Although Morell2014K1 gets better compression rate than the Octree method, it gives a bigger distance and color error. The compression rate of both methods is less affected by the real and synthetic point clouds. The analysis of the Octree method shows that it works better with real point clouds since this kind of point clouds are organized. In an organized point cloud each 3D point has a high probability of being close to a neighbor (in the range image). Thus, the points to compress are close in the 3D space and the Octree method takes advantage of this feature.

Figure 2 shows an example of a different way to compare compression methods. In this case, we show the results with respect to texture level. Binary methods are not affected by the texture level. Again the geometric methods provide better compression rates although the Octree method is able to get a better compression rate when the texture is low.

We then present the results obtained from the quantitative measures. Figure 3, 4 and 5 show the results for curvature and entropy metrics. For each figure, the results (compression rate, distance and color error) with respect to curvature and entropy are shown. Our tool can provide separated plots for real and synthetic point clouds. We found that this feature is very useful as we have detected that the compression methods behave differently when working with real or synthetic point sets. While with real point clouds binary methods are able to get an acceptable and very similar (40%) compression rate, geometric methods provide significantly better compression rates. As explained before, binary methods do not compress well the synthetic point clouds due to the low file size. With respect to geometric methods, it seems that a high curvature level provide a better compression rate in real im-

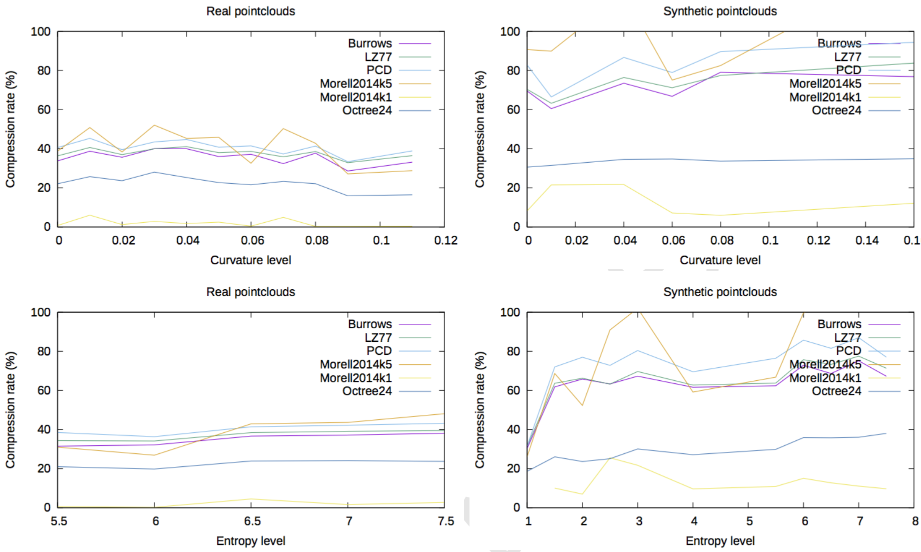


Figure 3: Showing the compression rate of the tested 3D compression methods with respect to curvature and entropy.

ages, but the differences are not significative. The opposite occurs with entropy level. The first conclusion is that all the compression methods work better with real point clouds (i.e. organized ones).

6. Conclusions

3D data compression is a challenging and relevant topic specially for mobile robotics where large amounts of 3D data are processed. Several methods have been proposed to deal with this problem. In this paper, we present a dataset which can serve as a benchmark for comparison and evaluation of different 3D compression methods. The dataset has been proposed to capture the variability of different parameters: real and synthetic data, different structure and texture levels. Together with the data, we have developed some useful tools for execution and evaluation of existing 3D compression methods in order to obtain an easy way to perform comparison among methods.

As an example, we have included the analysis of the comparison of five existing methods. The Octree lossy method obtained a good compression rate

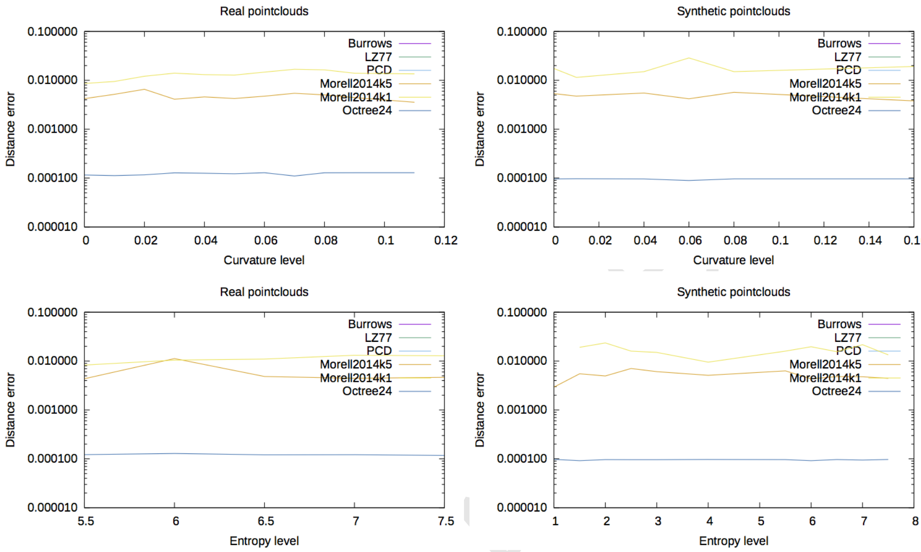


Figure 4: Showing the distance error of the tested 3D compression methods with respect to curvature and entropy.

with some decompression error in space and color distance. The Morell2014 method is a bit difficult to tune up for a generic point cloud. It has too many parameters which resulted to be critical for obtaining good results for images with any kind of structure and texture. Regarding the tested lossless methods, the best results in the entire dataset and the different categories were obtained with the Burrows o bzip2 method. With this example we show the validity and and usefulness of the proposed dataset for comparing compression methods for 3D data.

As future work we will extend the dataset to include more variability and identify special data that may be challenging for existing methods.

7. Acknowledgments

This work was partially supported by grant DPI2013-40534-R of the Ministerio of Economia y Competitividad of the Spanish Government, supported with Feder funds, and Valencian's Government project GV/2014/097.

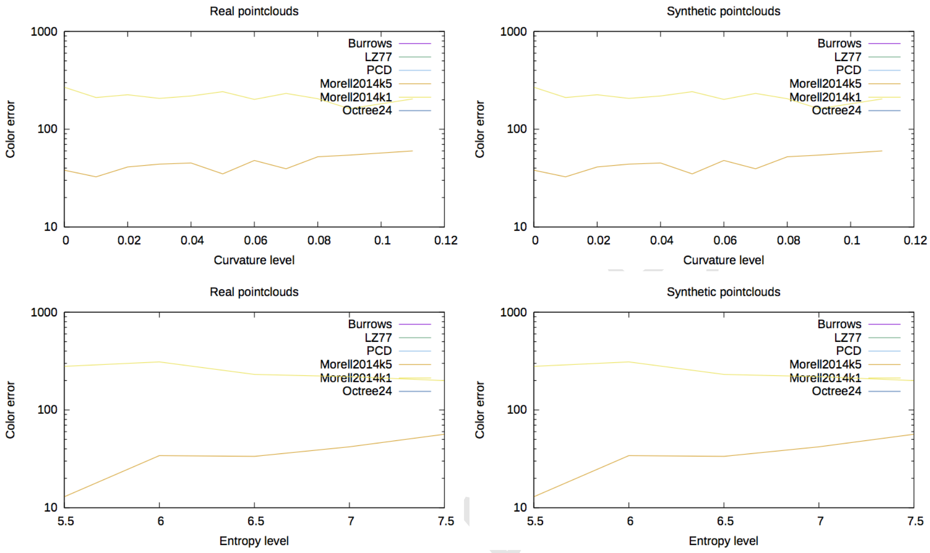


Figure 5: Showing the color error of the tested 3D compression methods with respect to curvature and entropy.

References

[1] Ian M Bullock, Thomas Feix, and Aaron M Dollar. The yale human grasping dataset: Grasp, object, and task data in household and machine shop environments. *The International Journal of Robotics Research*, page 0278364914555720, 2014.

[2] Caroline Pantofaru. User observation & dataset collection for robot training. In *Proceedings of the 6th international conference on Human-robot interaction*, pages 217–218. ACM, 2011.

[3] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[4] Allison Janoch, Sergey Karayev, Yangqing Jia, Jonathan T Barron, Mario Fritz, Kate Saenko, and Trevor Darrell. A category-level 3d object dataset: Putting the kinect to work. In *Consumer Depth Cameras for Computer Vision*, pages 141–165. Springer, 2013.

[5] Min Sun, Gary Bradski, Bing-Xin Xu, and Silvio Savarese. Depth-encoded hough voting for joint object detection and shape recovery. In *Computer Vision–ECCV 2010*, pages 658–671. Springer, 2010.

[6] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1817–1824. IEEE, 2011.

[7] Arjun Singh, James Sha, Karthik S Narayan, Tudor Achim, and Pieter Abbeel. Bigbird: A large-scale 3d database of object instances. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 509–516. IEEE, 2014.

[8] Yuxue Fan, Yan Huang, and Jingliang Peng. Point cloud compression based on hierarchical point clustering. In *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2013 Asia-Pacific*, pages 1–7, Oct 2013.

[9] J. Kammerl, N. Blodow, R.B. Rusu, S. Gedikli, M. Beetz, and E. Steinbach. Real-time compression of point cloud streams. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 778–785, May 2012.

[10] Ruwen Schnabel and Reinhard Klein. Octree-based point-cloud compression. In *Symposium on Point-Based Graphics*, 2006.

[11] Jan Elseberg, Dorit Borrmann, and Andreas Nuchter. One billion points in the cloud - an octree for efficient processing of 3d laser scans. *ISPRS Journal of Photogrammetry and Remote Sensing*, 76(0):76 – 88, 2013. Terrestrial 3D modelling.

[12] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *Information Theory, IEEE Transactions on*, 23(3):337–343, May 1977.

[13] M. Burrows and D. J. Wheeler. A block-sorting lossless data compression algorithm. Technical report, Digital Equipment Corporation, 1994.

[14] Sam Friedman and Ioannis Stamos. Online detection of repeated structures in point clouds of urban scenes for compression and registration. *International Journal of Computer Vision*, 102(1-3):112–128, 2013.

[15] Vicente Morell, Sergio Orts, Miguel Cazorla, and Jose Garcia-Rodriguez. Geometric 3d point cloud compression. *Pattern Recognition Letters*, 50(0):55 – 62, 2014.

[16] Haifeng Yu, Rui Wang, Junli Chen, Liang Liu, and Wanggen Wan. Saliency computation and simplification of point cloud data. In *Computer Science and Network Technology (ICCSNT), 2012 2nd International Conference on*, pages 1350–1353, Dec 2012.

[17] I. Daribo, R. Furukawa, R. Sagawa, and H. Kawasaki. Adaptive arithmetic coding for point cloud compression. In *3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), 2012*, pages 1–4, Oct 2012.

[18] Microsoft Corp. Redmond wa. kinect for xbox 360., 2012.

[19] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

***Biography of each author**

Javier Navarrete
Javier Navarrete obtained his BSc in Computer Science in 2002 and his Master degree in 2012 both from the University of Alicante. Since 2008, he is a Technician and a researcher in the University Institute for Computing Research (IUCR) at the University of Alicante. His research interests are focused on 3D vision and robotics .

Vicente Morell
Vicente Morell received a BSc , MSc and PhD in Computer Science from the University of Alicante (Spain) in 2008, 2010 and 2014 respectively. He is currently a post-doc student and researcher in the Artificial Intelligence Department at the University of Alicante. His research interests are focused on computer vision, robotics and neural networks.

Miguel Cazorla Miguel Cazorla received a BS degree in Computer Science from the University of Alicante (Spain) in 1995 and a PhD in Computer Science from the same University in 2000. He is currently Associate Professor in the Dept Computer Science and Artificial Intelligence at the University of Alicante. His research interests are focused on computer vision and mobile robotics (mainly using vision to implement robotics tasks).He has published more than 100 papers in JCR journals and international conferences.

Diego Viejo
Diego Viejo obtained his BSc and MSc in Computer Science in 2002 and his PhD in 2008 both from the University of Alicante. Since 2004, he is a lecturer and a researcher in the Department of Computer Science and Artificial Intelligence (DCCIA) at the University of Alicante. His research interests are focused on 3D vision applied to mobile robotics.

Jose Garcia-Rodriguez
Jose Garcia received his Ph.D. degree, with specialization in Computer Vision and Neural Networks, from the University of Alicante (Spain). He is currently Associate Professor at the Department of Computer Technology of the University of Alicante. His research areas of interest include: computer vision, computational intelligence, machine learning, pattern recognition, robotics, man-machine interfaces, ambient intelligence, computational chemistry, and parallel and multicore architectures. He has authored +90 publications in journals and top conferences and revised papers for several journals like Journal of Machine Learning Research, Computational intelligence, Neurocomputing, Neural Networks, Applied Softcomputing, Image Vision and Computing, Journal of Computer Mathematics, IET on Image Processing, SPIE Optical Engineering and many others, chairing sessions in the last five editions of IJCNN and IWANN, organizing special issues in journals like Neural Processing Letters, and participating in program committees of several conferences including IJCNN, ICRA, ICANN, IWANN, KES, ICDP and many others.

Sergio Orts-Escolano

Sergio Orts-Escolano received his BSc and MSc in Computer Science from the University of Alicante (Spain) in 2008 and 2010 respectively. He received the PhD degree from University of Alicante in 2014. He is currently a Postdoctoral researcher in the Computer Technology Department at the University of Alicante. His research interests include 3D computer vision, real-time, GPU computing and neural networks.

***Photo of each author**

Javier Navarrete

Vicente Morell

Miguel Cazorla

Diego Viejo

Jose Garcia-Rodriguez

Sergio Orts-Escolano