The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Non-Cubic Occupied Voxel Lists for Robot Maps

Julian Ryde and Michael Brünig

Abstract— **An alternative to the conventional quantization for occupied voxel lists in both 2D and 3D is presented. The performance metrics of the hexagonal lattice in 2D and the face centred and body centred cubic lattices in 3D are investigated and compared to their square and cubic counterparts. It is found that quantization to alternative lattices yields some improvements. Ultimately, the D3 or face centred cubic lattice is highlighted for its lower quantization error, lower rotation variability and higher order rotational symmetry. It has three times less occupied voxel count pose variability than a standard cubic occupied voxel list. These improvements have implications for SLAM and path planning.**

I. Introduction

Navigating in an unknown or even dynamically changing environment remains a difficult task for any robot. The first step is usually to sense the environment in either 2D or 3D and then to build a suitable representation that can be used for navigation purposes, i.e. path planning, obstacle avoidance or object recognition. Depending on the input sensors utilized, the gathered information about the environment can be very rich and needs to be processed to reduce the computational load for interpreting the data for the afore mentioned purposes. Different approaches have been chosen to achieve this goal. One of the simplest and most robust ones is the creation of occupied voxel lists, [1]. The incoming data, i.e. the 2D or 3D points, are quantized in cells of equal size. Cells that contain a number of data points above a predefined threshold are marked as occupied. The threshold is often chosen to be as low as 1. Subsequently, tasks like path planning only need to ensure that a path does not interfere with occupied cells.

While this method is well known and implemented in many robots, little attention has so far been paid to the shapes of the cells in the grid. A straightforward approach is to use a linear quantization along the axes that form either the robot's or a global coordinate frame. This results in the well known square cells in 2D and cubes in 3D, respectively. The advantage is the simplicity of the quantizer which requires no more than a computationally inexpensive rounding operation. There are however significant disadvantages to this method and surprisingly simple alternatives exist that can avoid these shortfalls without sacrificing computational simplicity. In order to describe the disadvantage a simple thought experiment can be made in 2D. A line of points with the distance $\sqrt{2}a$ between the points, when sampled into a 2D grid of square cells of size a^2, will result in a solid diagonal line of occupied cells if sampled from 45°. The same setting would lead to a broken line of occupied and unoccupied cells when the sampling grid is in alignment (90°) with the line of points. The orientation of the coordinate frame used to build the occupied voxel list has a significant impact on the result.

Authors are with the Autonomous Systems Laboratory, CSIRO ICT Centre, Brisbane, Australia julian.ryde@csiro.au

In Section V we show examples of this rotational dependency and how it affects the overall system. The perfect cell size for rotational invariance in 2D would be a circle but non overlapping circles cannot cover a plane without gaps so they are not suitable to build occupied voxel lists. The perfect solution in 2D is a hexagonal structure that provides a minimum amount of rotational dependence, it can cover a 2D plane without gaps and fast algorithms exist for quantizing the incoming data. The theory behind these approaches has been thoroughly investigated for vector quantization and extensive literature is available, [2], [3], [4], [5]. Section II outlines the theoretical background borrowed from that domain and establishes the relation to the occupancy map problem. Very little evidence in the literature exists of the use of hexagonal occupied voxel lists in 2D and to our knowledge this is the first work that describes and evaluates the extension to 3D. For a comparison between the proposed and existing methods, the computational complexity and comparison metrics are discussed in Section II-B and Section V outlines the results.

Another difficulty of square or cubic cell shape is the definition of nearest neighbours to an occupied cell as the surrounding cells have different distances, i.e. a and $\sqrt{2}a$ for the cell size a^2 in 2D. With hexagonal structures in 2D, this problem does not exist, all neighbours are located at the same distance and all share one edge with the cell. This remains the case when appropriate structures are used in 3D; all neighbours then share one face.

The theoretical background especially that of lattice theory is explained in Section II. Various related work is provided in Section III some from robotics but much of it from vector quantization research. The quantizing algorithms and performance metrics used are explained in Section IV. The results and overall analysis of the introduced methods are presented in Sections V and VI respectively.

II. Theoretical Background

The conversion of continuous data into discrete data is quantization. This is an important first step in the processing of incoming sensor data for robotics. The sensor data can be quantized onto a map consisting of a square grid in 2D or cubic lattice structure in 3D. Algorithms for performing these quantizations are described in Section IV-A. The remainder of this section presents some of the background and terminology necessary for understanding the method and analysis.

TABLE I
SUMMARY OF THE LATTICES CHARACTERISED IN THIS WORK

Strukturbericht	Description	Voronoi/Primitive Cell
Z2	Square	Square
A2	Hexagonal	Hexagon
Z3	Cubic	Cube
D3 $\equiv$ A3	FCC	Rhombic Dodecahedron
D3* $\equiv$ A3*	BCC	Truncated Octahedron

A. Lattice Theory

A lattice may be defined as a subset of points in a space that are generated by discrete translations only. The primitive cell of a lattice is the volume or area associated with an individual lattice point. In this way these primitive cells must be space filling. The ubiquity of lattices is somewhat disguised by the synonyms for lattices and primitive cells. Lattices can be referred to as meshes, grids, trellises or Bravais lattices in crystallography and similarly the primitive cell as the voronoi cell or Wigner-Sitz cell in solid state physics. These are different to tessellations or their 3D counterpart honeycombs which can incorporate rotations between the composite cells whereas a lattice is restricted to translations.

Table I contains the lattices that are described and applied to mobile robot mapping in this research. The *Strukturbericht* designation is a concise taxonomy for classifying the structure of lattices that was originally used in crystallography. Table I presents the slightly more intuitive descriptions of face centered cubic (FCC) and body centered cubic (BCC) lattices. The polytope of the primitive or voronoi cell is also listed.

A BCC lattice may be considered as a cubic lattice with additional lattice points at the centre of each cube body. Similarly an FCC lattice is a cubic lattice with extra lattice points at the centre of each cube face.

B. Performance Metrics

There are a number of performance metrics that are considered for quantitative analysis of the performance of the lattice quantization.

1) Voxel count: This is the simplest metric and is essentially the size of the set of lattice points that the input points are quantized to. It gives an idea of the compression and an approximate representation for a difference between two points sets. If the voxel count is different then the underlying lattice sets must be different, however a similar voxel count is only a necessary not sufficient condition for lattice set similarity.

2) Distortion/quantization error: The parameters for evaluating the performance of a vector quantizer are application dependent however the distortion or quantization error is generally used to indicate the error introduced by the quantization process. The distortion is the mean Euclidean separation between points and their corresponding quantized lattice points.

3) Variation/variability: Clearly low distortion error is a good aspect however another metric is how much variation the lattice quantization exhibits under pose and coordinate transformations. This variability for data with mean, μ, and standard deviation, σ, is assessed by the coefficient of variation σ/μ. Alternatively the inter-quartile variability factor, the ratio of the interquartile range to the median could be used.

C. Lattice Properties

Various lattice properties have been proved in the vector quantization literature. The two that are that are most relevant are the packing density and the quantization performance. The packing density can be ascertained by placing identical spheres centered at each lattice point that are just touching their neighbouring spheres. The ratio of the volume of these spheres to that of the entire space spanned by the lattice is the packing density. We conjecture that the packing density is related to the isotropy and the quantization performance is clearly good for minimising the errors introduced by quantization. It is known, [6], that the A2 lattice is optimal in both its density of packing and quality of quantization. However in 3D it is more complicated. It has been shown the body centered cubic lattice D3* is the optimal 3D lattice quantizer for uniformly distributed data, [2]. It has been proven that A3 and D3 (FCC) are the densest lattice packings. It has been further conjectured but not proven that this is the optimal packing, lattice or otherwise. The work discussing theoretical optimality for various lattices or packings assumes a random uniform distribution of points which does not necessarily hold for the distribution of occupied voxels in robot maps.

III. Related Work

There are not many examples [7], [8] in the robotics literature describing non-square occupancy grids. Occupancy grids and even occupied voxel lists are rare in 3D and this work is the first, to our knowledge, describing non-cubic occupied voxel lists. Related data structures often used for 3D mapping can be based on octrees as described by [9], [10] and used in [11]. Other disciplines have been quicker to embrace the advances in vector quantization with [12] examining distribution of galaxies in the universe in a manner similar to occupancy grids. In [4] it is pointed out that the conventional spherical packing is a face-centred cubic (FCC) lattice. The FCC lattice points are generated by selecting the points of a cubic lattice whose coordinates sum to an even number. This gives insight into the algorithms used for these lattice quantizers. For enlightenment into lattices that would be useful for robotics it is also worth noting the background literature in spherical packing. For a lattice it is known that the FCC is the optimal packing solution however what is not known is whether there are any non-lattice packings that are superior. This work is confined to lattice representations of space because the algorithms for quantization are generally faster. This densest lattice, D3, is of interest however densest does not mean the best quantizer which according to [4] is the A3* lattice.

A. Vector Quantization

For much of the lattice quantization algorithms we turn to the vector quantization literature which for many years has been discussing these very concepts. Many aspects of all vector quantizers their performance and algorithms are discussed in [3]. Lattice quantizers are described as the appropriate choice for uniform quantization. The performance of the hexagonal lattice versus that of the square lattice is thoroughly compared in [5], [13] for uniform input data.

The algorithms for quantizing the maps are slightly modified versions of those found in [14], [15].

IV. Method

Scan data is acquired from laser range sensors both 2D and 3D. In order to represent this data it is stored in occupied voxel lists [16], [1] which are similar to occupancy grids but maintain a list of only those cells that are likely to be occupied.

Repeated elements are accumulated to generate the occupied voxel list with each entry a tuple $[a, b, c, o]$. Where the integers a, b, and c describe the lattice point and the occupancy, o, is the number of points associated with that lattice point.

The scans are combined to produce a map and the map quantized to different lattices Z2, A2, Z3, D3 and D3*. The pose or coordinate frame invariance of these lattice descriptions is then assessed by inspecting how the distortion and voxel counts vary with rotation.

A. Quantizing Algorithms

The notation used to describe these algorithms is that the coordinates of a single unquantized point are represented by the tuple X, the lattice indices resulting from quantization are the integer tuple A. Rounding down or the floor operation is represented with $\lfloor x \rfloor$. The lattice spacing is ϵ.

1) Zn: Quantization on the Zn (square, cubic) lattice is most straightforward and is done naturally in many applications, however it is included here for completeness.

$$A = \lfloor X/\epsilon + 0.5 \rfloor \quad (1)$$

This is equivalent to conventional rounding.

2) A2: Quantization onto an A2 (hexagonal) lattice is performed by recognising that the dual lattice of the hexagonal lattice is an equilateral triangle tessellation which itself can be expressed as two rectangular lattices offset by half a cell from one another (Fig. 1). The algorithm consists of conventional quantization on the two rectangular lattices and then determining the closer of the two lattice points by a standard Euclidean distance comparison.

The x and y components of the lattice spacing are ϵ_x and ϵ_y respectively.

$$\epsilon_x = 3t, \epsilon_y = \sqrt{3}t \quad (2)$$

where t is the side length of the hexagonal cell. The lattice indices of the point (x, y) for the lattice A_0 and the lattice A_1 offset from A_0 by $r_1 = (\epsilon_x/2, \epsilon_y/2)$ are then given by

$$A_0 = \lfloor x/\epsilon_x + 0.5 \rfloor \epsilon_x, \lfloor y/\epsilon_y + 0.5 \rfloor \epsilon_x \quad (3)$$

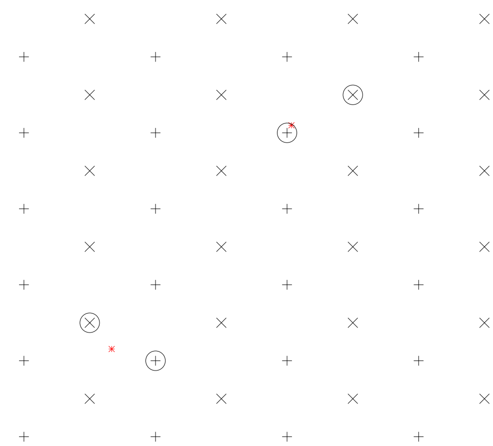

Fig. 1. The hexagonal quantization algorithm for two example points. The triangular tessellation dual of the hexagonal lattice is represented as two rectangular lattices with identical cell dimensions $(3t, \sqrt{3}t)$ with the edge length of a hexagonal cell being t. One lattice is offset by half a cell in both axes. The points are quantized to the rectangular lattices separately and of the resulting two lattice points the closer corresponds to the hexagonal cell containing the original data point.

$$A_1 = (\lfloor x/\epsilon_x \rfloor + 0.5)\epsilon_x, (\lfloor y/\epsilon_y \rfloor + 0.5)\epsilon_y \quad (4)$$

3) Dn: The algorithm for Dn is very similar to that presented in [14]. First $f(x)$ is defined as

$$f(x) = \lfloor x + 0.5 \rfloor \quad (5)$$

This is different to the definition of [14] but is done in this manner for ease of implementation. Erroneous quantization will occur for values of x with a decimal of exactly 0.5, however this will be very unlikely for real noisy data. First $\delta = x - f(x)$ and $w = |\delta|$. The function $g(x)$ is the same as $f(x)$ save that the component of x for which w is largest is rounded the incorrect direction, namely further from zero. Between $f(x)$ and $g(x)$ the one with an even sum of components corresponds to the nearest Dn lattice point.

4) D3:* Since the D3* lattice maybe be represented as the union of two lattices, [14], A_0 and A_1 offset by $r_1 = (0.5, 0.5, 0.5)$ from A_0, then $A_0 = f(X)$ and $A_1 = r_1 + f(X - r_1)$. Finally select whichever of A_0 or A_1 is closer to X via a comparison of the Euclidean distances $||X - A_0||$ and $||X - A_1||$.

B. Unit Cell Area/Volume

In order to make a fair comparison between the different lattice quantizations tested the unit cell areas or volumes need to be ascertained. These volumes and areas are theoretically evaluated and then verified with a sample based approach.

In 2D the area of the unit cell is simple to calculate geometrically. For A2 the area of the hexagonal unit cell is $3\sqrt{3}/2t^2$ where t is the side length. The sample based approach operates by uniformly generating a sample of N points across a cubic volume, V, that is guaranteed to encompass the unit cell associated with a particular lattice point. The points are then all quantized according to the appropriate lattice quantizer and the number of points, n,

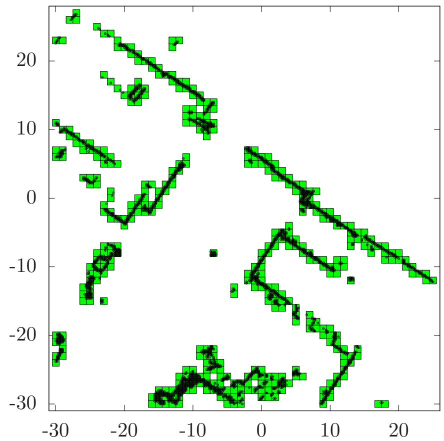

Fig. 2. Conventional square ($1\mathrm{m}^2$ area) quantization of a 2D map.

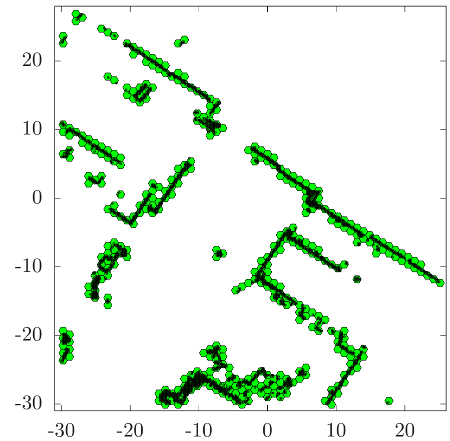

Fig. 3. Quantization of the same 2D map into hexagons with area of $1\mathrm{m}^2$.

associated with the particular lattice point retained. Thus an estimate of the volume of the unit cell is $v = nV/N$ which becomes more accurate as $N \rightarrow \infty$.

V. Results

Both synthetic data and that from two experimental platforms is used to test the effectiveness of the proposed lattice quantization approaches. Synthetic data is used to highlight situations in which better lattice quantization become important and to check the correctness of the algorithms. One platform [17] is operating outdoors in an aluminium smelting plant and is currently human controlled. This platform records 2D laser scan data which is combined into maps using a multi-resolution [16] mapping algorithm an example of which is shown in Fig. 2.

The second platform is an indoor robot equipped with a 3D laser range scanner [18] which acquires 3D scans in a stop-and-go manner. This is due to the relatively slow scan speed (2Hz) and inevitable scan distortion that would result from significant motion. Although the performance of the lattice quantizations on real data is the primary interest of this work it is insightful to inspect the results of synthetic data which helps highlight aspects and behaviours that might otherwise be masked in real data.

Random uniformly distributed data is tested for the different quantizers in order to check the algorithms involved. As is expected the resulting distortion of the quantized lattice points is insensitive to coordinate transforms.

For the 2D quantization results of the data displayed in Fig. 2 the square quantization had a distortion of 0.382 ± 0.0046 and a voxel count of 390 ± 13.7. The corresponding hexagonal quantization of the same data in Fig. 3 had a distortion of 0.378 ± 0.0040 and a voxel count of 376 ± 7.02. The variation of hexagonal and square voxel count with rotation of the original scan data is plotted in Fig. 4.

Fig. 5 allows visual comparison of the cubic, FCC and BCC quantizers for a single scan. The ceiling of the scan is removed to aid visual clarity. The voxel volumes are all one litre.

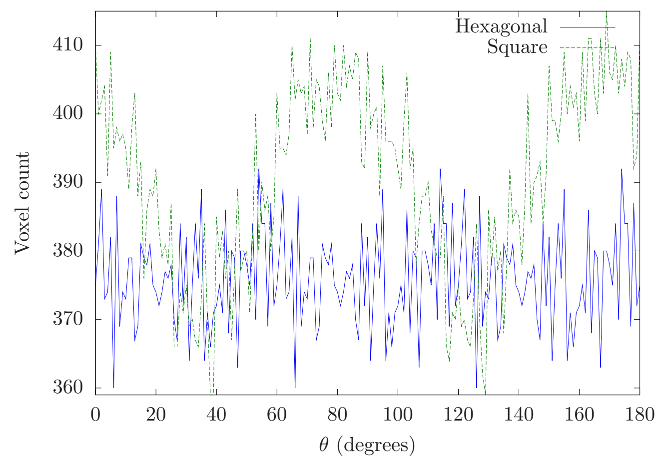

Fig. 4. Voxel count as a function of map rotation. Ideally this metric should be independent of map rotations. Note the strong 90°periodicity due to orthogonal structure of the environment.

VI. Analysis

The relative time for the lattice quantizations is given in Table II. This gives an idea of the computational effort required for each of the lattice quantization methods. The results were from timings of the quantizations to a resolution of 0.1m under Octave for the 3D scan containing 80,000 points which were represented as 7,500 lattice points. The two main disadvantages of alternative lattice quantizations are conceptual complexity and larger processing time. Usually the time taken to quantize the incoming sensor data is very small compared to the subsequent operations that need to be

TABLE II

Summary of results from indoor 3D scan

Lattice	Distortion σ/μ	Voxel σ/μ	Relative Time
Cube	0.0066	1.7%	1
D3	0.0065	0.7%	5
D3*	0.0033	0.6%	10

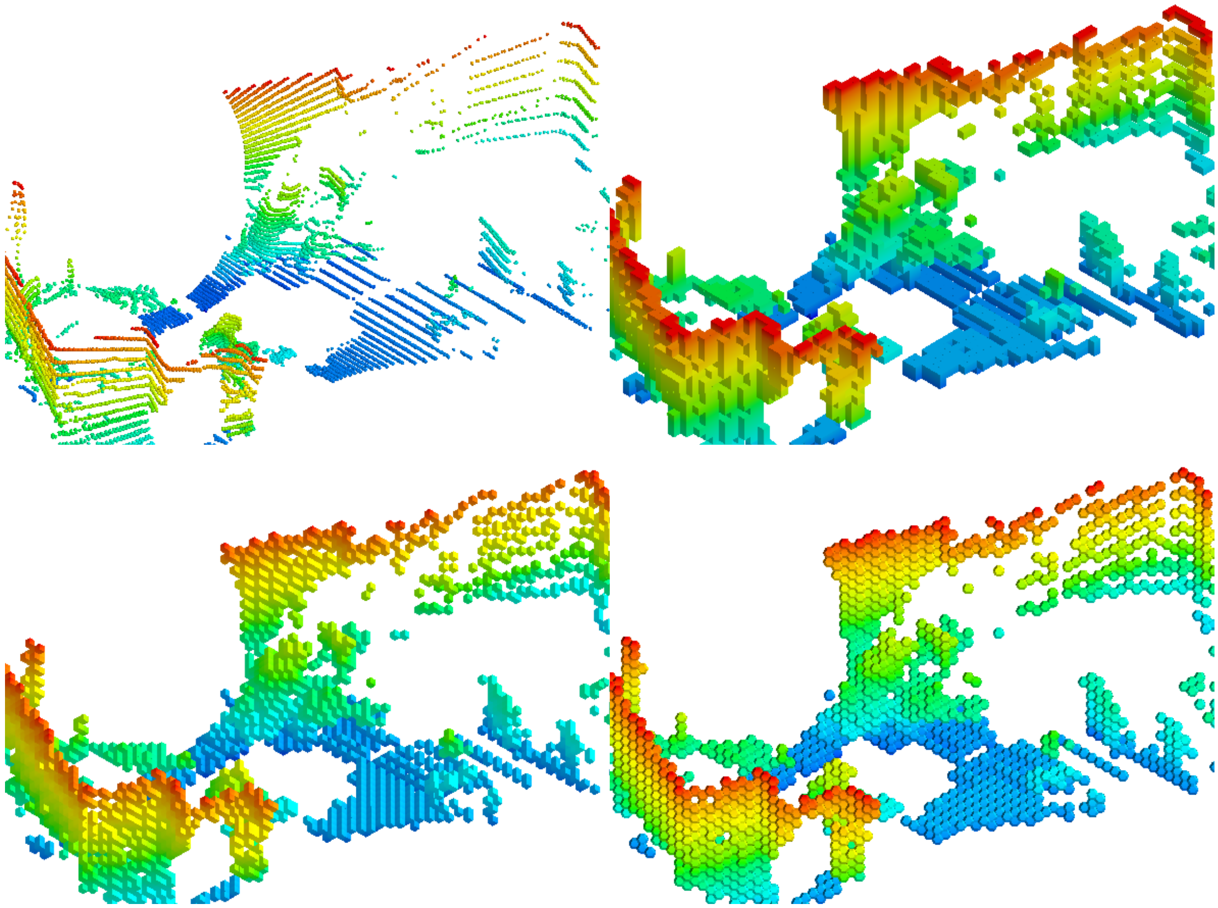

Fig. 5. A single scan quantised to cubic, FCC and BCC occupied lists all with voxel volumes of one litre. Voxels are coloured by height.

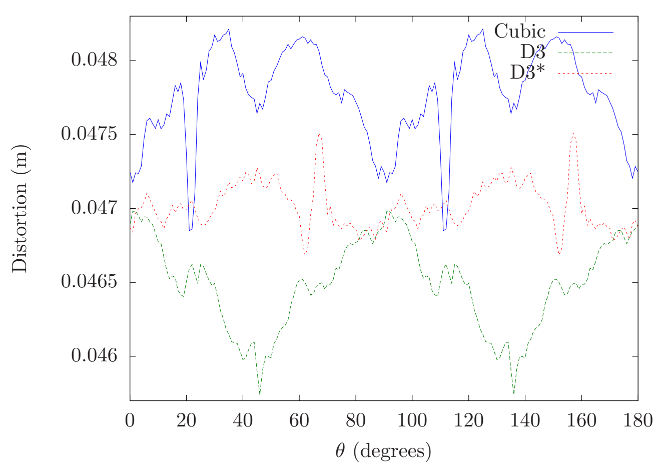

Fig. 6. Distortion as a function of map rotation about the z-axis for the 3D scan (Fig. 5) of an indoor environment.

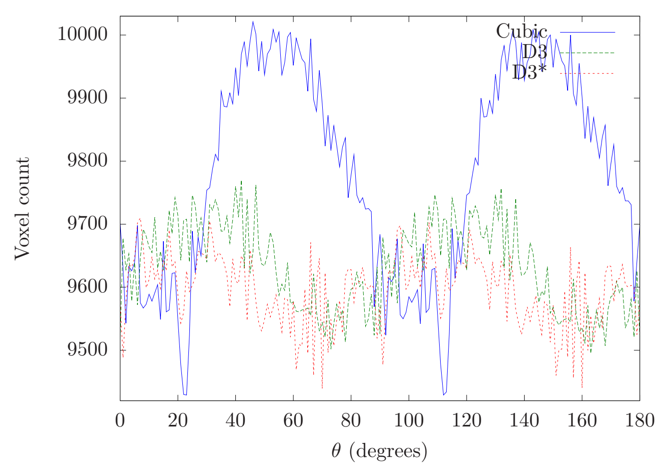

Fig. 7. Voxel count variation with rotation for Zn, D3 and D3* for the 3D scan data depicted in Fig. 5.

performed on the lattice data. For example path planning and localisation; the localisation process requires many tests of the scan lattice against a map lattice or previous scan lattice, [16], [18]. Compared to the alternative of feature extraction, lattice quantization is much faster. Some might propose the standard cubic quantization at better resolutions as a simpler alternative to D3 and D3* lattice quantization. Unfortunately the improvement in resolution comes at a large memory cost. But more importantly does not allow the occupied voxel list to correctly and innately represent the error in the underlying data. Strictly speaking the resolution of the occupied voxel list should not be limited by computational concerns, but should be similar in size to the error in the complete system.

The spikes in Fig. 6 occur with the alignment of walls in the room with the lattice planes. Fig. 6 demonstrates the more efficient quantization capabilities of the D3 and D3* lattices which have both lower overall distortion and reduced distortion variation. These two lattices again perform better when considering the voxel count and its variation in Fig. 7.

A. Advantages of A2, D3 and D3*

1) Neighbour queries: Neighbour queries are important for two main applications. In path planning it is necessary to rapidly check adjacent cells for obstacles. In a square or cubic lattice path planning is made difficult because each cell does not have a clearly defined set of nearest neighbours. However in a hexagonal lattice each cell has 6 equidistant nearest neighbours. Standard path planning algorithms will naturally result in smoother, shorter and consequently more efficient routes. This naturally extends to 3D with the D3 and D3* lattices. For localisation and mapping it is also important to be able to quickly ascertain the similarity of individual scans to a global map or previous scan as in scan matching. This can be done by accessing some measure of the proximity of scan points to occupied regions of the map. This proximity can be estimated by checking the coincident lattice points. Those lattices which are more isotropic return a more consistent proximity estimate regardless of pose or coordinate frame.

2) Representation efficiency: The additional dimension for 3D maps drastically increases the amount of data required to represent 3D environments and so any improvement in representation efficiency is welcome. Essentially being able to represent the original data most accurately with minimal information is the very essence of vector quantization. By applying a D3 or D3* lattice quantization algorithm the same map may be represented more compactly as is clear from all the distortion graphs. These graphs show a marked improvement in distortion or quantization noise for both D3 and D3* over the cubic system for the same volume of unit cells.

3) Rendering/appearance: Although the focus of this paper is on a quantitative performance comparison the qualitatively visually superior appearance of an alternative lattice quantization is apparent from Fig. 3. For the hexagonal quantizer the walls appear smoother and are generally more appropriately represented with the structure of the environment clearer.

VII. Conclusion

The conventional approach to quantization for robotics involves occupancy grids or occupied voxel lists on a cubic lattice because this quantization process is convenient, in that it is both easy to understand and compute. Some work has touched on this idea in 2D with a hexagonal representation however there appears to be no prior work using a non-cubic occupied voxel list representation in the field of robotics. The cubic representation is suboptimal and this research both highlights and explains this and further proposes a solution that is both theoretically optimal and experimentally better. We propose a replacement to the occupancy grid, termed the non-cubic occupied voxel list. Performance of these non-cubic occupied voxel lists is tested on both 2D and 3D data from indoor and outdoor environments. This solution involves quantizing in the A2 lattice for 2D data and for either the D3 or D3* lattice for 3D data. From Fig. 6 the distortion is lower with D3 however the rotational variability of the distortion is lower with D3*. The voxel count variability with rotation is similar for both D3 and D3*. Although it would appear that D3 and D3* are similar in performance the higher order of rotational symmetry of D3 would make it a better choice for robot maps. Further work is likely to entail implementation of common robotics algorithms on D3 and D3* lattices such as SLAM and path planning.

References

[1] J. Ryde and H. Hu, "Mobile robot 3D perception and mapping with multi-resolution occupancy lists," in *Proc. of IEEE Int. Conf. on Mechatronics and Automation (ICMA 2007)*, Harbin, Heilongjiang, China, Aug. 2007.

[2] E. S. Barnes and N. J. A. Sloane, "The optimal lattice quantizer in three dimensions," 1983.

[3] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Kluwer Academic Publishing Group, 2001.

[4] J. H. Conway, N. J. A. Sloane, and E. Bannai, *Sphere Packing, Lattices and Groups*. Springer, 1999.

[5] S. D. Servetto, V. A. Vaishampayan, and N. J. A. Sloane, "Multiple description lattice vector quantization," *Proc. 1999 Data Compression Conference*, pp. 13–22, 1999.

[6] L. Fukshansky, "Revisiting the hexagonal lattice: On optimal lattice circle packing," 1991.

[7] H. J. Quijano and L. Garrido, "Improving cooperative robot exploration using an hexagonal world representation," *Proc. of the Electronics, Robotics and Automotive Mechanics Conference*, pp. 450–455, 2007.

[8] E. S. H. Hou and D. Zheng, "Mobile robot path planning based on hierarchical hexagonal decomposition and artificial potential fields," *Journal of Robotic Systems*, vol. 11, pp. 605–614, 2007.

[9] C. Jackins and S. Tanimoto, "Oct-trees and their use in representing three-dimensional objects," *Computer Graphics Image Process*, vol. 14, no. 3, pp. 249–270, 1980.

[10] D. Meagher, "Geometric modelling using octree encoding," *Computer Graphics Image Process*, vol. 19, no. 2, pp. 129–147, 1982.

[11] J. Poppinga, M. Pfingsthorn, S. Schwertfeger, K. Pathak, and A. Birk, "Optimized octtree datastructure and access methods for 3d mapping," in *IEEE Safety, Security, and Rescue Robotics (SSRR)*, 2007.

[12] T. Kiang, "Rhombic cell analysis - a new way of probing the large-scale structure of the universe." *Chinese Journal of Astronomy and Astrophysics*, vol. 3, pp. 95–105, 2003.

[13] R. Zamir, "On lattice quantization noise," *IEEE Transactions On Information Theory*, vol. 42, pp. 1152–1159, 1996.

[14] J. Conway and N. Sloane, "Fast quantizing and decoding algorithms for lattice quantizers and codes," *IEEE Transactions on Information Theory*, vol. 28, pp. 227–232, 1982.

[15] ——, "A fast encoding method for lattice codes and quantizers," *IEEE Transactions on Information Theory*, vol. 29, pp. 820–824, 1983.

[16] J. Ryde and H. Hu, "Mutual localization and 3D mapping by cooperative mobile robots," in *Proc. of Int. Conf. on Intelligent Autonomous Systems*, The University of Tokyo, Tokyo, Japan, Mar. 2006.

[17] J. Roberts, A. Tews, C. Pradalier, and K. User, "Autonomous hot metal carrier – navigation and manipulation with a 20 tonne industrial vehicle," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, Roma, Italy, 2007.

[18] J. Ryde and H. Hu, "3D laser range scanner with hemispherical field of view for robot navigation," in *Proc. of IEEE/ASME Advanced Intelligent Mechatronics Int. Conf. on Mechatronics and Automation (AIM 2008)*, Xi'an, China, Jul. 2008.