Texture-Based Segmentation of 3D Probabilistic Occupancy Maps for Robot Navigation

Bassel Abou Merhy, Pierre Payeur, Emil M. Petriu
School of Information Technology and Engineering
University of Ottawa, Ottawa, ON, Canada
bassel@site.uottawa.ca, ppayeur@site.uottawa.ca, petriu@site.uottawa.ca

Abstract

In previous work, an unsupervised texture-based segmentation algorithm was introduced to process 2D probabilistic occupancy maps used to navigate mobile robots. Given that numerous robotic applications are carried out in 3D environments, this paper presents the extension of the approach to cover the case of 3D probabilistic occupancy maps. The proposed unsupervised segmentation technique relies on the analysis of texture defined by a double distribution of the "local binary pattern" (LBP) and the "contrast" (C) operators in order to subdivide the space into regions characterized by a uniform occupancy state. The algorithm is able to identify in separate segments the different objects present in an environment through the analysis of their proximity. In order to keep the method tractable in 3D applications, a redefinition of the texture unit and an adaptation of the subdivision process are presented that provide very satisfactory results while significantly reducing the size of the texture distribution histograms that need to be computed and iteratively compared to achieve the segmentation in uniform regions.

Keywords: 3D segmentation, probabilistic maps, local binary pattern, contrast, texture.

1 Introduction

Probabilistic occupancy grids offer a powerful mode of representation of occupancy maps computed from the fusion of data collected from multiple points of view. But the use of range sensors characterized by a limited angular resolution plays a major role in introducing uncertainty on the topological distribution of objects in the environment. This leads to occupancy maps that do not exhibit sharp transitions on objects' boundaries as would be desirable for the extraction of the occupancy information in autonomous robotic applications where mobile robots or manipulator arms need to detect safe navigation areas, or objects on which they need to perform an action. Taking into account all of these factors, this work aims at extending a segmentation algorithm that was previously developed in the context of 2D occupancy maps [1], to generalize it up to a scheme capable of subdividing a probabilistic grid of a 3D environment into regions characterized by uniform and deterministic occupancy states.

The characteristics of the probabilistic representation of occupancy grids and the application considered in autonomous robotics impose some constraints on the design of an appropriate segmentation algorithm. Unlike supervised schemes that assume a preliminary knowledge of the processed model, an unsupervised segmentation is necessary here as autonomous robots typically operate in a priori unknown environments.

Considering that the workspace is progressively scanned by a laser range finder or a sonar only along specific directions, the resulting map of explored space reveals the content of the workspace only along a series of rays corresponding to the beams emitted by the active range sensor. Because of this property, a region-based segmentation approach is favoured over a boundary-based one which tends to confuse the rays with the edges of the objects. Approaches based on classical methods such as split and merge [2], pyramid node linking [3], as well as quadtrees [4] were the first to provide unsupervised region-based segmentation. Recent unsupervised segmentation methods explore, either multi-resolution filtering, using Gabor filters [5] or the wavelets [6], or statistics with hidden Markov fields [7].

Given the specific characteristics of probabilistic occupancy maps used in robot guidance, an approach that combines contrast and texture properties to identify regions of uniform occupancy state reveals to be a more appropriate strategy for differentiating between segments exhibiting different occupancy properties. Ojala et al. [8, 9, 10] proposed such a segmentation technique based on "Local Binary Pattern" and "Contrast" (*LBP/C*) operators to subdivide images with sharp patterns in the context of texture classification. But unlike the images that they considered, the transitions between free and occupied spaces in probabilistic maps do not define clear boundaries and spread out according to the uncertainty level introduced by the sensor and data fusion. Refinement to the original *LBP/C* segmentation mechanism has been initially proposed in [1] to handle smooth transitions in complex images while achieving accurate contours definition corresponding to segments with deterministic occupancy states found in the context of robot guidance: free, unknown and occupied regions.

The quality of the results achieved in the 2D case and applied to mobile robots navigation motivated the extension of the 2D segmentation scheme to the 3D case in order to address the problem of manipulator robots collision-free path planning. The development of the extended scheme took into account the minimisation of the impact of the models' size increase on the execution time. This issue is handled by a clever redefinition of the texture unit which is adapted to the 3D representation without degrading the overall performance of the segmentation.

2 Texture Representation

Ojala and Pietikainen [8] defined the principle of texture analysis using the distribution of the "Local Binary Pattern" (*LBP*). This operator describes the local texture characterizing the entourage of each pixel. Although it constitutes an important source of information about the local texture characteristics, the *LBP* alone cannot represent the contrast between the values of neighbour pixels. Thereafter, Ojala et al. extended their scheme by adding a "Contrast" operator, *C*, for each local texture unit. Thus, the texture in a given area of the model is to be characterized by a double distribution of *LBP* and *C* measures. The proposed segmentation algorithm exploits this multi-variable distribution to achieve classification of uniform regions.

2.1 Bidimensional Texture Representation

In the original scheme [8], the *LBP* and *C* values are calculated for every pixels of a given region in the original image, except for those located on the external boundaries. For each processed pixel, a block of size [3 x 3] pixels immediately surrounding the pixel of interest is considered (figure 1a). A discretisation process, which leads to a binary representation, is applied to this texture unit. In this process, the central pixel is used as a threshold, and all neighbouring pixels with a value higher or equal to the threshold are set to one, the others are set to zero (figure 1b). Next, the binary values are multiplied by binomial weights (figure 1c) and the results (figure 1d) are added, excluding the value of the central pixel, in order to obtain the *LBP* value of the texture unit.

The *LBP* parameter is combined with a simple measure of contrast that is equal to the difference between the average of the pixels' original values having a unity binary representation (i.e. 1 after application of the threshold) and the average of the pixels' original values having a null binary representation.

The resulting *LBP/C* distribution is encoded in a bidimensional histogram of size [256 x *b*]. The first dimension size comes from the fact that a central pixel has 8 neighbour cells for a texture unit of size [3x3] pixels. Each of them being represented by a binary variable (figure 1b), a total of 2^8 possible *LBP* values can be computed. The second dimension size, *b*, corresponds to the number of levels of discretisation of contrast, *C*. This decimal value belongs to the interval [-254; 255] for an image encoded on 8 bits. For this reason, a process of discretisation is necessary to determine the correspondence between a given value and its position in the *LBP/C* distribution histogram. The selection of a proper number of bins, *b*, for the *C* values remains a compromise between precision and performance.

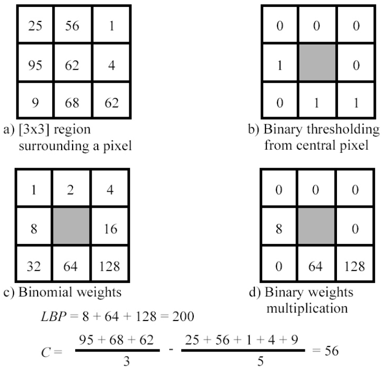

25 56 1
95 62 4
9 68 62
a) [3x3] region surrounding a pixel
0 0 0
1 0
0 1 1
b) Binary thresholding from central pixel
1 2 4
8 16
32 64 128
c) Binomial weights
0 0 0
8 0
0 64 128
d) Binary weights multiplication
LBP = 8 + 64 + 128 = 200
C = (95 + 68 + 62)/3 - (25 + 56 + 1 + 4 + 9)/5 = 56

Figure 1: *LBP* and *C* values estimation for a 2D texture unit.

2.2 Adapted Tridimensional Texture Representation

A segmentation scheme based on the analysis of a double distribution of the *LBP* and *C* operators directly depends on the definition of the texture unit. As introduced previously, the dimensions of the *LBP/C* histogram are defined by the range of the *LBP* possible values and by the number of levels of discretisation, *b*, selected to represent the contrast. The 2D texture unit of size [3x3] implies eight neighbour cells for each central pixel, which leads to 2^8 = 256 possible values for the *LBP*. Under this framework, if a supplementary dimension was introduced, the size of the distribution histogram would double for each additional neighbouring cell. As a result, fully considering a cubic texture unit of size [3x3x3] voxels involves 26 neighbours for each central voxel, as shown in figure 2.

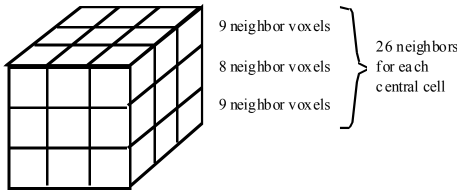


Figure 2: 3D texture unit of size [3x3x3] with its 26 neighbours.

The proposed redefinition of the texture unit keeps the histogram dimensionality low by taking into account only the neighbour voxels that share a face with the central one. Edge neighbours and vertex neighbours are eliminated. The resulting texture unit is symmetrical relative to the three Cartesian planes (x, y), (x, z) and (y, z) respectively, as illustrated in figure 3. The symmetry implies that the proposed segmentation algorithm does not discriminate between any of the three dimensions of the space during the analysis of the textures which characterize the environment occupancy. The adapted texture unit results in distribution histograms of size [2^6 x *b*]. In spite of the reduction of the histogram size by a factor four compared to the 2D case, the results of the 3D segmentation algorithm remain coherent and stable, as will be shown in the following sections.

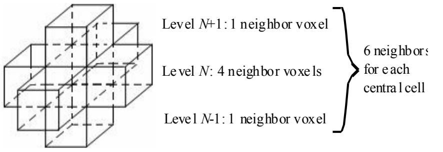

Level N+1: 1 neighbor voxel
Level N: 4 neighbor voxels
Level N-1: 1 neighbor voxel
6 neighbors for each central cell

Figure 3: 3D representation of the adapted texture unit.

In a similar way to the 2D case, for a given 3D region, the *LBP* and *C* values are computed for each voxel, except for those located on the external edges of the map. For each considered voxel, the six neighbours with which the latter shares a face are considered. Figure 4 presents a flattened representation of a tridimensional texture unit. The central voxel as well as its right, left, bottom and top neighbours are located at level *N*, while the neighbour voxels at the lower and upper levels are located respectively at levels *N-1* and *N+1*. The texture unit (figure 4a) first undergoes a discretisation process which leads to a binary representation. The occupancy probability of the central voxel is used as a threshold and all neighbours with a higher or equal value are set to one, the others are set to zero (figure 4b). The binary values obtained are multiplied by binomial weights (figure 4c) and the results (figure 4d) are added, excluding the central voxel value, in order to obtain the *LBP* value of the tridimensional texture unit of interest (here, *LBP* = 2+8+16+32 = 58).

The calculation of contrast follows the model introduced by Ojala and Pietikainen [8]. Thus, *C* corresponds to the difference between the average of the neighbouring voxels' intensity having a unity binary representation (after thresholding) and the average of the neighbouring voxels' intensity having a null binary representation. For examples, the contrast value of the texture unit presented in figure 4 is computed as follows: *C* = [(8+7+7+6)/4]-[(2+1)/2] = 5.5. For the 3D case, the resulting *LBP/C* distribution is represented in a two-dimensional histogram of size [64x*b*]. Based on [8] and on our previous 2D results presented in [1], a *b* value of 8 or 16 leads to similar results. However, according to the tests we carried out, an attempt to further decrease the number of discretisation levels, *b*, to 4 reduces considerably the sensitivity of the segmentation algorithm to the differences in contrast between occupied and free regions. Consequently, a value of 8 discretisation levels on the contrast was selected, resulting in a *LBP/C* histogram of size [64x8] in the 3D case.

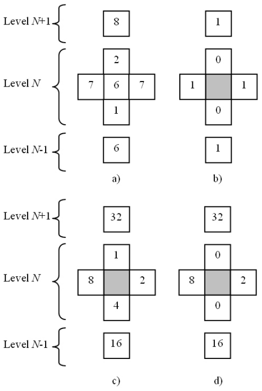


Figure 4: Calculation of the 3D texture unit's *LBP/C* characteristics.

2.3 Texture Comparison

The logarithmic likelihood ratio, the G-statistic [11], is used to compare histograms of *LBP/C* distributions, in both 2D and 3D cases. It provides robust means to classify uniform map segments. The value of *G* indicates the level of similarity between two frequency distributions. It estimates the likelihood that two compared regions have similar texture and contrast distributions. This measurement of similarity is calculated as follows:

$$\begin{aligned} G = {} & 2\left[\sum_{s,m}\sum_{i=1}^{n} f_i \times \ln(f_i)\right] - 2\left[\sum_{s,m}\sum_{i=1}^{n} f_i \times \ln\left(\sum_{i=1}^{n} f_i\right)\right] \\ & -2\left[\sum_{i=1}^{n}\left(\sum_{s,m} f_i\right)\times \ln\left(\sum_{s,m} f_i\right)\right] \\ & +2\left[\left(\sum_{s,m}\sum_{i=1}^{n} f_i\right)\times \ln\left(\sum_{s,m}\sum_{i=1}^{n} f_i\right)\right] \end{aligned} \qquad (1)$$

where f_i corresponds to the number of texture units characterized by a pair of (*LBP*; *C*) values in bin *i*. *s* and *m* represent the two distributions to compare, and *n* equals [64x*b*] which is the number of bins in each of the analyzed histograms.

3 3D Segmentation Algorithm

The structure of the tridimensional segmentation algorithm is inspired by a split and merge approach and is divided in three successive phases. The hierarchical division subdivides iteratively the probabilistic model in cubic blocks of uniform texture. Next, the segments creation phase merges adjacent regions with a similar occupancy state. Finally, the refinement phase improves the contours' localisation between neighbouring segments. The adjustments that were made to adapt the initial 2D segmentation technique to take into account the third dimension are detailed in this section.

3.1 Hierarchical Division

The objective of the 3D hierarchical division consists of subdividing the occupancy map in cubic blocks of variable sizes and relatively uniform texture. This phase first subdivides the probabilistic model in regions of size [S_{max}xS_{max}xS_{max}] voxels, S_{max} being equal to 64. A modified version of the uniformity test proposed by Ojala and Pietikainen [8] has been developed in order to determine if a "parent" subdivision contains heterogeneous textures and must be subdivided in eight sub-blocks of equal size or not. After having identified the eight subdivisions for each block, the *LBP/C* distribution histogram is computed in each subdivision. The eight resulting histograms are used to calculate the 28 logarithmic likelihood ratios, eq. (1), between each of the 28 possible pairs of subdivisions.

The largest and the smallest G-statistic values, denoted respectively by G_{max} and G_{min}, are identified. The parent block is considered non-uniform and thus subdivided if the ratio between G_{max} and G_{min} is higher than a threshold value, *X*. A proper value has been estimated empirically as *X*=1.2, as it provides a good discrimination between regions of different textures.

$$R = \frac{G_{\max}}{G_{\min}} > X \qquad (2)$$

If a block is recursively subdivided, its eight sub-regions undergo the same test. The iterative process is applied on the subdivided blocks until the subdivision size reaches [S_{min}xS_{min}xS_{min}], S_{min} being equal to 8 voxels. In a similar way to what was observed in the 2D case [1], a value of S_{min} equal to 8 rather than 16 provides more stable segmentation results as it leads to three recursive subdivision levels.

3.2 Segments Creation

The goal of the 3D segments creation phase consists of merging adjacent subdivisions that are characterized by a similar occupancy state until a convergence criterion is met. The segments creation starts by identifying for each of the subdivisions obtained at the end of the first phase, its neighbours from the top, the right and the lower level. This process guarantees that, at the end of the subdivisions scan, all possible pairs of adjacent regions will have been taken into consideration, and merged if they share similar texture characteristics.

The neighbouring blocks characterized by a mean occupancy probability, *MOP*, in the same interval are merged in a single region. This parameter is defined as the average cells' occupancy probability in a given region, R_i, and is calculated as follows:

$$MOP_i = \frac{1}{M \cdot N \cdot O} \sum_{l=0}^{O-1} \sum_{k=0}^{N-1} \sum_{j=0}^{M-1} OP(r_{j,k,l}) \Big| r_{j,k,l} \in R_i \qquad (3)$$

where *M*, *N* and *O* are the dimensions of the region R_i, and $OP(r_{j,k,l})$ is the confidence level on the occupancy state of a voxel *r* with coordinates (*j*, *k*, *l*).

The motivations behind the choice of the *MOP* value as a criterion to evaluate the similarity between adjacent regions are related to the nature of the probabilistic occupancy map. After normalizing the cells' occupancy probability values over the range [0;1], three possible occupancy states are considered: *i*) if a region is totally unknown, it is characterized by an occupancy probability in the interval [0.498; 0.502] and is tagged as *unknown*; *ii*) if the occupancy probability belongs to the interval [0; 0.498[, this region of space is mostly free and is tagged as *free*; *iii*) finally, an occupancy probability in the interval]0.502; 1] shows a mostly occupied region of space that is tagged as *occupied*. These deterministic states, $S(R_i)$ = {*free*, *unknown*, *occupied*}, can indicate whether or not a region is safe for robot navigation.

3.3 Refinement

The completion of the first two phases leads to an approximate segmentation given that the boundaries between the segments do not correspond perfectly to the edges between adjacent regions that have a different occupancy state [1]. Consequently, the last phase is dedicated to perfect the contours localisation of the segments by reallocating the voxels on the boundaries between two or several adjacent regions. The refinement process can be divided in two parts: the first one deals with the boundaries between the *free – unknown* segments, while the second handles boundaries between the *occupied – free* and *free – occupied* segments.

The first refinement step is based on the fact that the range of occupancy probability values leading to an unknown segment classification is relatively narrow, being contained in the interval [0.498; 0.502]. Even if a segment overlaps between an unknown region and a free one by a limited number of voxels, it will still be considered as free by the segments creation phase. Therefore, the space whose occupancy is free always juts out into the unknown one, which is unsafe for robot navigation and collision avoidance. Thus, the first refinement step consists of expanding the unknown segments to the detriment of their free neighbours. At the implementation level, this process consists of scanning the tridimensional model along the six possible directions: right – left, left – right, bottom – top, top – bottom, below – above and above – below. Along the direction of each scan, the free cells, located on the boundary between a free and an unknown segments, with occupancy probability equal to 0.5 are reassigned to the unknown segment. This process continues until a cell with a different value is met. The six directions scanning procedure ensures the coverage of all possible boundary shapes.

The second refinement step consists, in a similar way to the first one, of scanning twice the probabilistic map in the six possible directions. During each of these scans, boundaries are considered respectively between the *free – occupied* and *occupied – free* segments. Along the direction of a scan, when a *free – occupied* (*occupied – free*) boundary is reached, the three following voxels in the adjacent occupied (free) segment are examined.

The choice to consider several voxels beyond the boundary between two segments of known occupancy state comes from the existence of unknown regions between the adjacent probing directions of an active range sensor. Any misclassification in these unknown areas can create discontinuities in the segments, and thus distort the object detection process. According to our investigation, the choice of three voxels provides the best safety/performance ratio.

Depending on the value of the three considered cells, four reclassification cases are identified: *i*) if the three cells have a normalized occupancy probability of 0.5, no reclassification is applied given that these pixels might fall into the unknown region between adjacent sensor measures; *ii*) if at least one of the three voxels has a normalized occupancy probability strictly higher (lower) than 0.5, and the other voxels have an occupancy probability equal to 0.5, no reclassification is applied; *iii*) if at least one of the voxels has an occupancy probability strictly lower (higher) than 0.5, and the other voxels have an occupancy probability equal to 0.5, then the three voxels are reassigned to the free (occupied) segment; and *iv*) if at least one of the three voxels has an occupancy probability strictly lower (higher) than 0.5, and at least another one is strictly higher than 0.5, then the voxels with value lower (higher) or equal to 0.5 are reassigned to the free (occupied) segment.

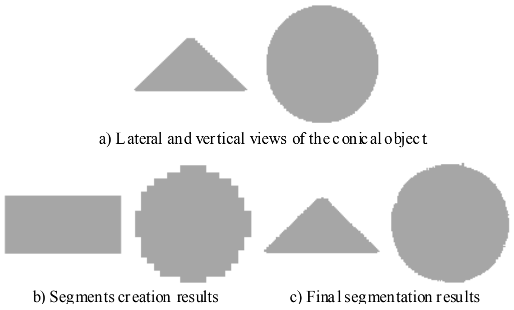
4 Experimental Results

In this section, segmentation results on three 3D probabilistic models each of size [320x320x128] are presented. These tridimensional probabilistic maps of an environment are created using a stack of planar images of the same size separated by a constant gap to create volume. These images are obtained using a laser range finder simulator for planar surface mapping that was developed in previous work. The width, the height and the depth of these environments correspond respectively to the dimensions of the images ([320x320]) and the number (128) of images that were piled up.

The occupied space shape as well as the number and position of the range sensor's points of view differ from a model to another. Gray areas in figures 5a, 6a and 7a represent the occupied regions by the objects in the models. These objects have respectively a conical, cubical and cylindrical shape.

The range sensor scans, used in the probabilistic models construction, are collected assuming a Gaussian error with standard deviation of 4 cm on the range measurements. The step angle between two adjacent sensor's rays which defines the angular resolution is fixed to 0.5 degree over all the measurements. The parameters used in this implementation of the segmentation technique are the same as the ones described in the preceding sections and are not modified when dealing with objects of different shape. In the hierarchical division phase, the size of the first level of subdivided blocks, S_{max} is [64x64x64] voxels, and three subdivision levels are conducted, leading to a S_{min} of [8x8x8] voxels.

Figures 5b, 6b and 7b present the segmented maps after the segments creation phase. The segments obtained at the end of the second step approximate well the shape of the regions present in the probabilistic map of the environment for all cases. Segmentation results obtained after the refinement phase are shown in figures 5c, 6c and 7c respectively for the three probabilistic maps. Important improvement on contours definition is achieved. The rendering of these 3D objects uses a slicing approach to allow easy navigation inside the map for accurate evaluation of the segmentation performance. The white regions observed inside the cubical and cylindrical objects in figures 6 and 7 respectively, correspond to the interior volume of the objects that was occluded from the sensor and therefore was never scanned. Beyond the external shape of the 3D objects, unknown regions are also correctly identified by the proposed segmentation scheme.

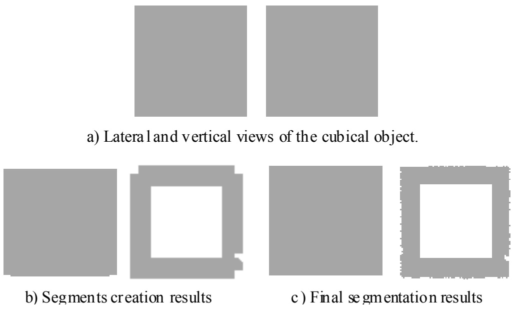


a) Lateral and vertical views of the conical object

b) Segments creation results

c) Final segmentation results

Figure 5: Segmentation results of a 3D environment containing a conical object.

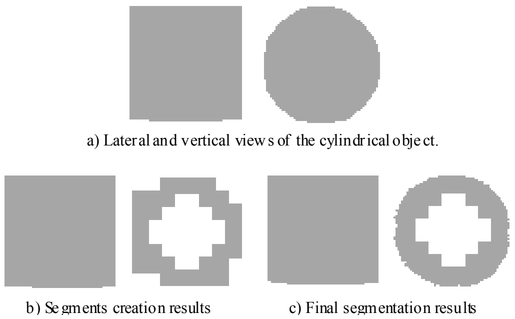


a) Lateral and vertical views of the cubical object.

b) Segments creation results

c) Final segmentation results

Figure 6: Segmentation results of a 3D environment containing a cubical object.



a) Lateral and vertical views of the cylindrical object.

b) Segments creation results

c) Final segmentation results

Figure 7: Segmentation results of a 3D environment containing a cylindrical object.

From a qualitative point of view, these experimental results demonstrate that the segmentation technique previously developed for bidimensional maps [1] can readily be extended to process tridimensional probabilistic maps of an environment. The proposed extension opens the door to a multitude of applications involving interaction between autonomous manipulator robots and their environment in exploration or manufacturing tasks.

From a quantitative point of view, the proposed tridimensional segmentation algorithm remains computationally efficient for a static model of the environment when global path planning is considered. This was made possible by the redefinition of the texture unit for 3D voxels that compresses the intermediate histogram representation by a factor of 1048576 [= (2^{26} x 8) / (2^{6} x 8)]. On average, the segmentation of a probabilistic model of size [320x320x128] voxels, as presented here, requires less than 11 minutes, which represents about 50 μsec per voxel. The segmentation time is minor compared to what is required to generate tridimensional probabilistic maps (data acquisition, registration, and range data fusion). Therefore it offers a realistic solution considering the accuracy of the segmented maps that can be achieved.

5 Conclusion

This paper proposes and evaluates an extended version of a texture-based segmentation scheme to operate on 3D probabilistic maps as obtained from limited resolution active range sensors. The redefinition of an optimal "Local Binary Pattern" and "Contrast" texture metrics successfully limits the increase in processing workload associated with the additional dimension introduced in the model which is known to drive numerous algorithms into a computational explosion.

The proposed texture unit definition is designed to preserve the robustness of the *LBP/C* texture mapping, thus leading to accurate segmentation results of tridimensional objects. Such segmented 3D maps can readily be used by manipulator arms classical path planning approaches to navigate without collision in cluttered environments, or to perform tasks on specific objects. It can also operate as a first processing stage in pattern recognition and object classification in 3D space.

6 References

[1] B. Abou Merhy, P. Payeur and E.M. Petriu, "Application of segmented 2D probabilistic occupancy maps for mobile robot path planning", *Proceedings International Instrumentation and Measurement Technology Conference (IMTC2006)*, Sorrento, pp 2342-2347 (2006).

[2] X. Wu, "Adaptive Split-and-merge segmentation based on piecewise least-square approximation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(8), pp 808-815 (1993).

[3] F. Arman and J.A. Pearce, "Unsupervised classification of cell images using pyramid node linking", *IEEE Transactions on Biomedical Engineering*, 37(6), pp 647-650 (1990).

[4] M. Spann and R. Wilson, "A quad-tree approach to image segmentation which combines statistical and spatial information", *Pattern Recognition*, vol. 18, pp 257-269 (1985).

[5] D.P. Mital, "Texture segmentation using Gabor filters", *Proceedings International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies,* Brighton, UK, vol. 1, pp 109-112 (2000).

[6] M. Unser, "Texture classification and segmentation using wavelet frames", *IEEE Transactions on Image Processing*, vol. 4, pp 1549-1560 (1995).

[7] H. Choi and R.G. Baraniuk, "Multiscale image segmentation using wavelet-domain hidden Markov models", *IEEE Transactions on Image Processing*, vol. 10, pp 1309-1321 (2001).

[8] T. Ojala and M. Pietikainen, "Unsupervised texture segmentation using feature distributions", *Pattern Recognition*, vol. 32, pp 477-486 (1998).

[9] T. Mäenpää, T. Ojala, M. Pietikäinen and M. Soriano, "Robust texture classification by subsets of local binary patterns", *Proceedings International Conference on Pattern Recognition*, vol. 3, pp 935–938 (2000).

[10] T. Ojala, M. Pietikäinen and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp 971-987 (2002).

[11] R.R. Sokal and F.J. Rohlf, *Biometry: The Principles and Practice of Statistics in Biological Research*, 2nd Edition, W. H. Freeman and Company, San Francisco (1981).