Point Cloud Data Filtering and Downsampling using Growing Neural Gas

Sergio Orts-Escolano Vicente Morell José García-Rodríguez Miguel Cazorla

***Abstract*— 3D sensors provide valuable information for mobile robotic tasks like scene classification or object recognition, but these sensors often produce noisy data that makes impossible applying classical keypoint detection and feature extraction techniques. Therefore, noise removal and downsampling have become essential steps in 3D data processing. In this work, we propose the use of a 3D filtering and downsampling technique based on a Growing Neural Gas (GNG) network. GNG method is able to deal with outliers presents in the input data. These features allows to represent 3D spaces, obtaining an induced Delaunay Triangulation of the input space. Experiments show how GNG method yields better input space adaptation to noisy data than other filtering and downsampling methods like Voxel Grid. It is also demonstrated how the state-of-the-art keypoint detectors improve their performance using filtered data with GNG network. Descriptors extracted on improved keypoints perform better matching in robotics applications as 3D scene registration.**

I. Introduction

Historically, humans have the ability to recognize an environment they visited before based on the 3D model they unconsciously build in their heads based on the different perspectives of the scene. This 3D model is built with some extra information so that humans can extract relevant features [1] that will help in future experiences to recognize the environment and even possible present objects. This learning method has been transferred to mobile robotics field over the years. So, most current approaches in scene understanding and visual recognition are based on the same principle: keypoint detection and feature extraction on the perceived environment. Over the years most efforts in this area have been made towards feature extraction and keypoint detection on information obtained by traditional image sensors [2], [3], existing a gap in feature-based approaches that use 3D sensors as input devices. However, in recent years, the number of research papers concerned with 3D data processing has increased considerably due to the emergence of cheap 3D sensors capable of providing a real time data stream and therefore enabling feature-based computation of three dimensional environment properties like curvature, getting closer to human learning processes.

The Kinect device[1], the time-of-flight camera SR4000[2] or the LMS-200 Sick laser[3] mounted on a sweeping unit are examples of these devices. Besides, providing 3D information, some of these devices like the Kinect sensor can also provide color information of the observed scene. However, using 3D information in order to perform visual recognition and scene understanding is not an easy task. The data provided by these devices is often noisy and therefore classical approaches extended from 2D to 3D space do not work correctly. The same occurs to 3D methods applied historically on synthetic and noise-free data. Applying these methods to partial views that contains noisy data and outliers produces bad keypoint detection and hence computed features does not contain effective descriptions. Consequently, in order to perform an effective keypoint detection is needed to remove as much noise as possible keeping descriptive features as corners or edges.

Classical filtering techniques like median or mean have been widely used to filter noisy point clouds [4], [5] obtained from 3D sensors like the ones previously mentioned. The median filter is one of the simplest and wide-spread filters that has been applied. It is efficient and simple to implement but can remove noise only if the noisy pixels occupy less than one half of the neighbourhood area. Moreover, it removes noise but at the expense of smoothing corners and edges of the input data.

Another filtering technique frequently used in point cloud noise removal is the Voxel Grid method. The Voxel Grid filtering technique is based on the input space sampling using a grid of 3D voxels to reduce the number of points. This technique has been traditionally used in the area of computer graphics to subdivide the input space and reduce the number of points [6], [7]. The Voxel Grid method presents some drawbacks: sensitivity to noisy input spaces. Moreover, since all the points present will be approximated (i.e., downsampled) with their centroid it does not represent the underlying surface accurately.

More complex noise removal techniques from image processing [8] have been used also on point clouds: the Bilateral filtering technique, [9] applied on depth maps obtained from 3D sensors, allows to remove noise considering corners and edges using Gaussian functions and range kernels. However, Bilateral filtering is not able to deal with outliers in the input point cloud.

Based on the Growing Neural Gas network [10] several authors proposed related approaches for surface reconstruction

Sergio Orts-Escolano and Jose Garcia-Rodriguez are with the Department of Computer Technology of the University of Alicante (email: {sorts, jgarcia}@dtic.ua.es).

Vicente Morell and Miguel Cazorla are with the Department of Science of the Computation and Artificial Intelligence Department of the University of Alicante (email: {vmorell, miguel}@dccia.ua.es).

[1]Kinect for XBox 360: http://www.xbox.com/kinect Microsoft

[2]Time-of-Flight camera SR4000 http://www.mesa-imaging.ch/prodview4k.php

[3]LMS-200 Sick laser: http://robots.mobilerobots.com/wiki/SICK_LMS-200_Laser_Rangefinder

978-1-4673-6129-3/13/$31.00 ©2013 IEEE

applications [11], [12]. However, most of these contributions do not consider noisy data obtained from RGB-D cameras using noise-free CAD models.

In this paper, we propose the use of a 3D filtering and downsampling technique based on the GNG network. By means of a competitive learning, it makes an adaptation of the reference vectors of the neurons as well as the interconnection network among them, obtaining a mapping that tries to preserve the topology of an input space. Besides, GNG method is able to deal with outliers in the input data. These features allow to represent 3D spaces, obtaining an induced Delaunay Triangulation of the input space very useful to easily obtain features like corners, edges and so on. Filtered point cloud produced by the GNG method is used as input of many state-of-the-art 3D keypoint detectors in order to demonstrate how the filtered and downsampled point cloud improves keypoint detection and hence feature extraction and matching in 3D registration methods. Proposed method is compared with one state-of-the-art filtering techniques, the Voxel Grid method. Results presented in Section IV show how the proposed method overperforms Voxel Grid method in input space adaptation and noise removal. In Figure 1, it is shown a general system overview of the steps involved in the 3D scene registration problem.

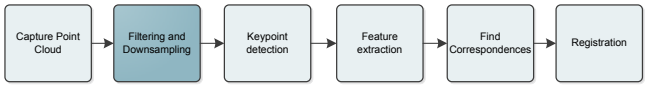

Fig. 1. General system overview

In this work we focus on the processing of 3D information provided by the Kinect sensor. Experimental results show that the random error of depth measurement increases with increasing distance to the sensor, and ranges from a few millimeters up to about 4 cm at the maximum range of the sensor. More information about the accuracy and precision of the Kinect device can be found in [13].

The rest of the paper is organized as follows: first, a section describing briefly the GNG algorithm is presented. In section III the state-of-the-art 3D keypoint detectors are described. In section IV we present some experiments and discuss results obtained using our novel approach. Finally, in section V we give our conclusions and directions for future work.

II. GNG Algorithm

With Growing Neural Gas (GNG) [10] method a growth process takes place from minimal network size and new units are inserted successively using a particular type of vector quantization. To determine where to insert new units, local error measures are gathered during the adaptation process and each new unit is inserted near the unit which has the highest accumulated error. At each adaptation step a connection between the winner and the second-nearest unit is created as dictated by the competitive Hebbian learning algorithm. This is continued until an ending condition is fulfilled, as for example evaluation of the optimal network topology or fixed number of neurons. The network is specified as:

- A set N of nodes (neurons). Each neuron $c \in N$ has its associated reference vector $w_c \in R^d$. The reference vectors can be regarded as positions in the input space of their corresponding neurons.
- A set of edges (connections) between pairs of neurons. These connections are not weighted and its purpose is to define the topological structure. An edge aging scheme is used to remove connections that are invalid due to the motion of the neuron during the adaptation process.

The GNG learning algorithm to map the network to the input manifold is as follows:

1) Start with two neurons a and b at random positions w_a and w_b in R^d.
2) Generate at random an input pattern ξ according to the data distribution $P(\xi)$ of each input pattern.
3) Find the nearest neuron (winner neuron) s_1 and the second nearest s_2.
4) Increase the age of all the edges emanating from s_1.
5) Add the squared distance between the input signal and the winner neuron to a counter error of s_1 such as:

$$\triangle error(s_1) = \|w_{s_1} - \xi\|^2 \tag{1}$$

6) Move the winner neuron s_1 and its topological neighbors (neurons connected to s_1) towards ξ by a learning step ϵ_w and ϵ_n, respectively, of the total distance:

$$\triangle w_{s_1} = \epsilon_w(\xi - w_{s_1}) \tag{2}$$

$$\triangle w_{s_n} = \epsilon_n(\xi - w_{s_n}) \tag{3}$$

For all direct neighbors n of s_1.

7) If s_1 and s_2 are connected by an edge, set the age of this edge to 0. If it does not exist, create it.
8) Remove the edges larger than a_{max} . If this results in isolated neurons (without emanating edges), remove them as well.
9) Every certain number λ of input patterns generated, insert a new neuron as follows:
   - Determine the neuron q with the maximum accumulated error.
   - Insert a new neuron r between q and its further neighbor f:

$$w_r = 0.5(w_q + w_f) \tag{4}$$

   - Insert new edges connecting the neuron r with neurons q and f, removing the old edge between q and f.
10) Decrease the error variables of neurons q and f multiplying them with a consistent α. Initialize the error variable of r with the new value of the error variable of q and f.
11) Decrease all error variables by multiplying them with a constant γ.

12) If the stopping criterion is not yet achieved (in our case the stopping criterion is the number of neurons), go to step 2.

This method offers further benefits over simple noise removal and downsampling algorithms: due to the incremental adaptation of the GNG, input space denoising and filtering is performed in such a way that only concise properties of the point cloud are reflected in the output representation. Moreover, the GNG method has been modified regard original version, considering also original point cloud colour information. Once GNG network has been adapted to the input space and it has finished learning step, each neuron of the net takes colour information from nearest neighbours in the original input space. Colour information of each neuron is calculated as the average of weighted values of the K-nearest neighbours, obtaining a interpolated value of the surrounding point. Color values are weighted using Euclidean distance from input pattern to neuron reference vector. In addition, this search is considerably accelerated using a Kd-tree structure. Colour downsampling is performed to apply keypoint detectors and feature extractors that deal with colour information.

III. Keypoint detectors/descriptors

In this section, we present the state-of-the-art 3D keypoint detectors used to test and measure the improvement achieved using GNG method to filter and downsample the input data. In addition, we describe main 3D descriptors and feature a correspondence matching method that we used in our experiments.

A. Keypoint detectors

First keypoint detector used is the widely known SIFT (Scale Invariant Feature Transform) [14] method. It performs a local pixel appearance analysis at different scales. SIFT features are designed to be invariant to image scale and rotation. SIFT detector has been traditionally used in 2D image but it has been extended to 3D space. 3D implementation of SIFT differs from original in the use of depth and curvature as the intensity value. SIFT detector uses neighbourhood at each point within a fixed-radius assigning its intensity as the Gaussian weighted sum of the neighbours' intensity values in order to archive the 4-dimensional difference of Gaussians (DoG) scale space. Then it detects local maxima in these 4D DoG scale space when the point value is greater than all its neighbours.

Other keypoint detectors used are based on a classical Harris 2D keypoint detector. In [15] a refined Harris detector is presented in order to detect keypoints invariable to affine transformations. 3D implementations of these Harris detectors [16] use surface normals of 3D points instead of 2D gradient images. Harris detector and its variants, extended from 2D keypoint detectors, have been tested using the proposed method in Section IV. All Harris variants have in common covariance matrix computation, but each variant makes a different evaluation of the trace and the determinant of the covariance matrix. Noble's variant corners detection algorithm [17] evaluates the ratio between the determinant and the trace of the covariance matrix. Tomasi's variant [18] performs eigenvalue decomposition over the covariance matrix using the smallest eigenvalue as keypoint score. Lowe's variant performs in a similar way than Noble's one but evaluating the ratio between the determinant and the squared trace of the covariance matrix. These small changes in the evaluation of the covariance matrix generate different keypoint detection as we will see in Section IV.

B. Descriptors

Once keypoints have been detected, it is necessary to extract a descriptor over these points. In the last few years some descriptors that take advantage of 3D information have been presented. In [19] a pure 3D descriptor called Point Feature Histograms (PFH) is presented. The goal of the PFH formulation is to encode the points k-neighborhood geometrical properties by generalizing the mean curvature around the point using a multi-dimensional histogram of values. This highly dimensional hyperspace provides an informative signature for the feature representation, is invariant to the 6D pose of the underlying surface, and copes very well with different sampling densities or noise levels present in the neighbourhood. A PFH representation is based on the relationships between the points in the k-neighborhood and their estimated surface normals. Briefly, it attempts to capture as best as possible the sampled surface variations by considering all the interactions between the directions of the estimated normals.

A simplification of the descriptor described above is presented as Fast Point Feature Histograms (FPFH) [20] and is based on a histogram of the differences of angles between the normals of the neighbour points. This method is a fast refinement of the Point Feature Histogram (PFH) that computes its own normal directions and it represents all the pair point normal diferences instead of the subset of these pairs which includes the keypoint. It reduces the computational complexity of the PFH algorithm from $O(nk^2)$ to $O(nk)$.

In addition, we used another descriptor called Signature of Histograms of OrienTations (SHOT) [21], which is based on obtaining a repeatable local reference frame using the eigenvalue decomposition around an input point. Given this reference frame, a spherical grid centered on the point divides the neighbourhood so that in each grid bin a weighted histogram of normals is obtained. The descriptor concatenates all such histograms into the final signature. There is also a color version (CSHOT) proposed in [22] that adds color information.

C. Feature matching

Correspondences between features or feature matching methods are commonly based on the euclidean distances between feature descriptors. Moreover, a rejection step is necessary in order to remove false positives from the previous step. One of the most used method to check the transformation between pairs of matched correspondences is based on the RANSAC (RANdom SAmple Consensus) algorithm

[23]. It is an iterative method that estimates the parameters of a mathematical model from a set of observed data which contains outliers. In our case, we used this method to search a 3D transformation (our model) which best explain the data (matches between 3D features). At each iteration of the algorithm, a subset of data elements (matches) is randomly selected. These elements are considered as inliers and a model (3D transformation) is fitted to those elements. The rest of the data is then tested against the fitted model and included as inliers if its error is below a given threshold. If the estimated model is reasonably good (its error is low enough and it has enough matches), it is considered as a good solution. This process is repeated a number of iterations and the best solution is returned.

IV. Experimentation

We performed different experiments on indoor scenes to evaluate the effectiveness and robustness of the proposed method. First, a normal estimation method is computed in order to show how simple features like estimated normals are considerably affected by noisy data. Secondly, accurate input space adaptation capabilities of the GNG method are showed calculating the Mean Square Error (MSE) of filtered point clouds regard their ground truth. Due to the impossibility of obtaining ground truth data from the Kinect Device, the experiment is performed using synthetic CAD models and data obtained from a simulated Kinect sensor. Finally, the proposed method is applied to 3D scene registration to show how keypoint detection methods are improved obtaining more accurate transformations. 3D scene registration is performed on a dataset comprised of 90 overlapped partial views of a room. Partial views are rotated 4 degrees in order to cover 360 degrees of the scene. Partial views were captured using the Kinect device mounted in a robotic arm with the aim of knowing the ground truth transformation. Experiments implementation, 3D data management (data structures) and their visualization have been developed using the PCL[4] library.

A. Improving normal estimation

Normal estimation methods based on the analysis of the eigenvectors and eigenvalues of a covariance matrix created from the nearest neighbours are very sensitive to noisy data. Therefore, in the first experiment, we computed normals on raw and filtered point clouds in order to demonstrate how a simple 3D feature like normal or curvature estimation can be affected by the presence of noise and how the proposed method improves normal estimation producing more stable normals.

In Figure 2 it is visually explained the effect caused by normal estimation on noisy data. Normal estimation is computed on the original and filtered point cloud using the same radius search: $r_s = 0.1$ (meters).

In Figure 3 can be observed how more stable normals are estimated using filtered point cloud produced by the GNG method. $20,000$ neurons and $1,000$ patterns are used as configuration parameters for the GNG method in the filtered point cloud showed in Figure 3 (Right).

[4]The Point Cloud Library (or PCL) is a large scale, open project [24] for 2D/3D image and point cloud processing.

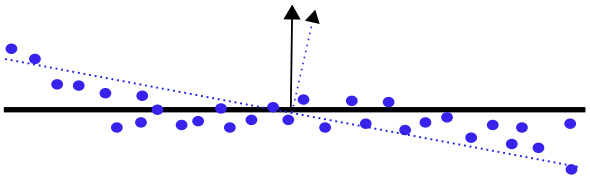

Fig. 2. Noise causes error in the estimated normal

B. Filtering quality: input space adaptation

In this experiment we demonstrated how the GNG method yields better input space adaptation to noisy data than other filtering and downsampling methods like Voxel Grid. In order to perform the experiment, ground truth data is required to calculate the MSE error regarding original data. As the Kinect device does not provide ground truth information, synthetic CAD models and data obtained from a simulated Kinect sensor are used as ground truth. For simulating a Kinect sensor and obtaining ground truth data, Blensor software [25] has been used. It allows us to generate synthetic scenes and to obtain partial views of the generated scene as if a Kinect device was used. The main advantage of this software is that it provides ground truth. In Figure 4 can be seen a synthetic scene generated using Blensor. On the left side it is shown the ground truth scene without noise and in the right side it is shown the partial view of the scene as if a Kinect device had captured it. Added noise is simulated using a Gaussian distribution with different deviation factors.

To perform this experiment, we calculated the MSE of the filtered point cloud with Voxel Grid and with the GNG method respect to the ground truth. The MSE is used to measure the filtered point cloud error relative to the ground truth and therefore it is a quantitative measure of the accuracy of the filtered point cloud. MSE is expressed in meters. Voxel Grid method presents some drawbacks as it does not allow specifying a fixed number of points, as the number of points is given by the voxel size used for building the grid. We forced the convergence to an approximate number in our experiments, making it comparison fairer. By contrast, the GNG neural network allows us to specify the exact number of end points that represent the input space. The experiment demonstrates the accuracy of the representation generated by the GNG network structure compared with Voxel Grid method.

Table I shows the adaptation MSE for different models and scenes using the GNG and the Voxel Grid method. On the top of the Table I, adaptation MSE is calculated for partial views obtained from a simulated Kinect sensor (meters) while in the bottom is calculated for synthetic CAD models (millimetres). Different levels of added noise σ are applied to the ground truth data. Results presented in Table I shows how the GNG method provides a lower mean error and therefore better adaptation to the original

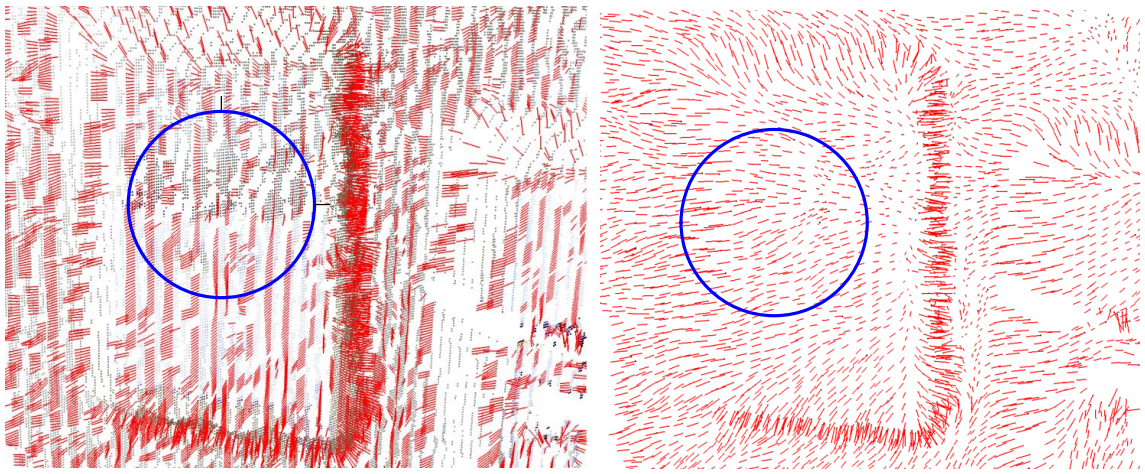

Fig. 3. Normal estimation comparison. Left: Normal estimation on raw point cloud. Right: Normal estimation on filtered point cloud produced by the GNG method

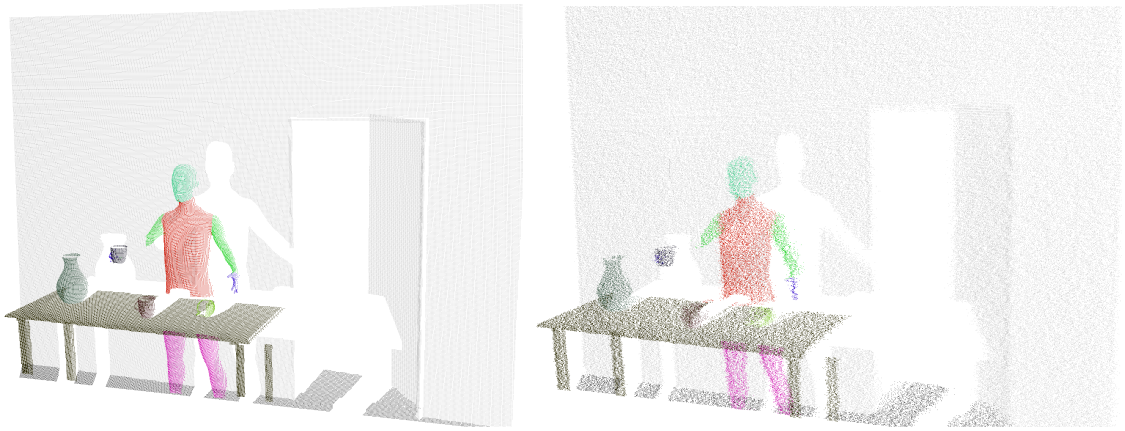

Fig. 4. Synthetic scene. Left: Ground truth without noise. Right: simulated Kinect view with added noise: $\sigma = 0.5$

TABLE I

INPUT SPACE ADAPTATION MSE FOR DIFFERENT MODELS. VOXEL GRID VERSUS GNG.

simulated Kinect	VG 5000	VG 10000	GNG 5000 250λ	GNG 10000 500λ
scene 1 $\sigma = 0.15$	0.0067	0.0064	**0.0017**	0.0022
scene 1 $\sigma = 0.25$	0.0197	0.0181	**0.0053**	0.0065
scene 1 $\sigma = 0.40$	0.0475	0.0430	**0.0156**	0.0185
scene 2 $\sigma = 0.15$	0.0053	0.0051	**0.0013**	0.0017
scene 2 $\sigma = 0.25$	0.0148	0.0135	**0.0041**	0.0051
scene 2 $\sigma = 0.40$	0.0372	0.0336	**0.0122**	0.0143

CAD model	VG 5000	VG 10000	GNG 5000 250λ	GNG 10000 500λ
model 1 $\sigma = 0.15$	0.0643	0.0641	0.0684	**0.0559**
model 1 $\sigma = 0.25$	0.0981	0.0994	0.0768	**0.0642**
model 1 $\sigma = 0.40$	0.2037	0.2276	**0.0903**	0.0924
model 2 $\sigma = 0.15$	0.1540	0.1504	0.1756	**0.1209**
model 2 $\sigma = 0.25$	0.3055	0.3227	0.1938	**0.1430**
model 2 $\sigma = 0.40$	0.8259	0.8895	0.2346	**0.2122**

input space, maintaining a better quality of representation in areas with a high degree of curvature and removing the noise generated by the sensor. The Voxel Grid method filters all the points present approximating (i.e., downsampling) them with their centroid, it does not represent the underlying surface accurately causing a worse adaptation. Experiments were performed with a fixed number of points, and in the case of GNG it was tested with different number of input signals λ generated by iteration, and different number of neurons, obtaining better results with a higher number of adjustments with the sacrifice of higher computation times. In Figure 5 can be visually observed adaptation MSE presented in table I for model 1. Voxel Grid output representations does not accurately fit input space, smoothing information in the edges and corners. From Table I we can observe that for used CAD models $10,000$ neurons are needed to represent accurately the input space, since that kind of models have more points. However for synthetic scenes are only needed $5,000$ neurons to represent the input space due to resolution of the input

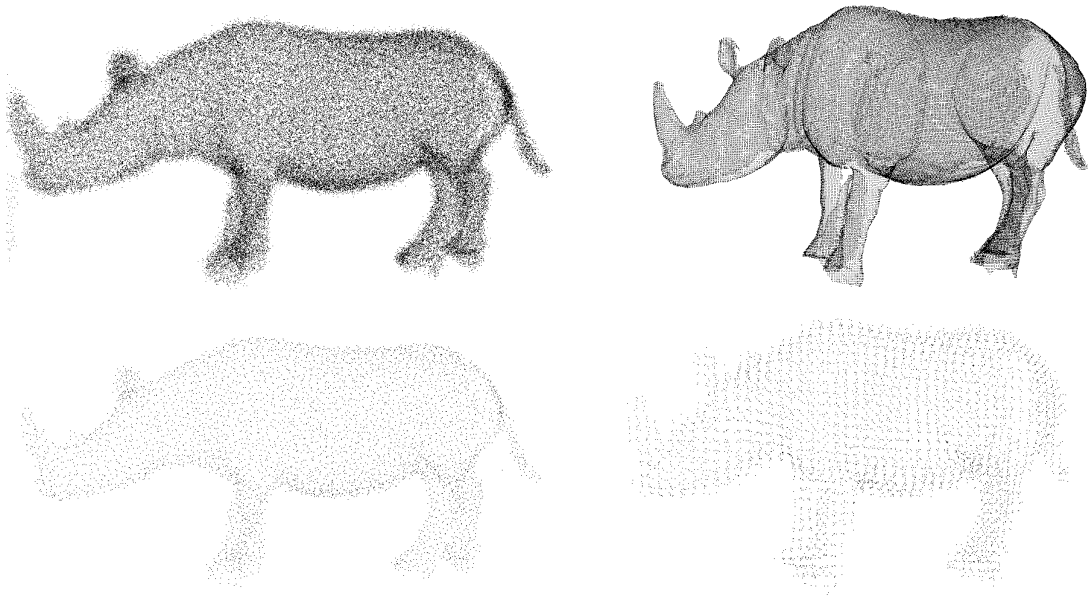

Fig. 5. Filtering quality using 10,000 points. GNG vs Voxel Grid comparison. Top Left: Noisy model $\sigma = 0.4$. Top Right: Original CAD model. Bottom left: filtered model using GNG method. Bottom right: filtered model using Voxel Grid.

space is lower.

C. Improving 3D keypoint detectors performance

In the last experiment, we used some keypoint detectors introduced in section III in order to test noise reduction capabilities of the GNG method for 3D scene registration problem. Root Mean Square (RMS) deviation measure [26] is used to measure obtained transformation error. It calculates the average difference between two affine transformations: the ground truth and the estimated one. Furthermore, we used a different number of neurons and patterns to obtain a downsampled and filtered representation of the input space. Different configurations have been tested, ranging from $5,000$ to $20,000$ neurons and 250 to $2,000$ patterns per epoch. In addition, the same configurations were tested with the Voxel Grid method to establish a comparison with the proposed method. In Figure 6 it is visually shown correspondences matching calculated over filtered point clouds using the proposed method. Red and blue points represent keypoints detected on filtered point clouds using the GNG method. Green lines are correct matchings between computed descriptors of two different views of the same scene.

Experiments were performed using different search radius for keypoint detection and feature extraction methods. Search radius influences directly on the size of the extracted features, making methods more robust against occlusions. A balance between large and small values must be found depending on the size of the present objects in the scene and the size of the features we want to extract. For the used dataset, the best estimated transformations are found using keypoint detector search radius 0.1 and 0.05 and feature extractor search radius 0.2.

In Table II it is shown how using GNG output representation as input cloud for the registration step, lower RMS errors are obtained in some detector-descriptor combinations. SIFT detector combined with the FPFH descriptor obtains lower RMS errors compared with Voxel Grid filtered point clouds and original point clouds. Lowe detector obtained lower RMS error for all descriptors. Tomasi and Noble detectors produced better results for the CSHOT descriptor and finally, Harris 3D is also improved with FPFH and CSHOT descriptors. However, Voxel Grid filtered point clouds also obtains good transformations in a few detector-descriptor combinations as SIFT-CSHOT or Tomasi-FPFH since some keypoint detectors and descriptors have implicit capabilities to deal with noisy data. SIFT or Tomasi are examples of keypoint detectors that deal well with noisy or incomplete data. There are some blank values in Table II caused by some configurations that did not find any keypoint and therefore registration process was not able to continue with feature extraction step.

One remarkable example showed in Table II that demonstrates capabilities of the proposed method is the combination of GNG filtered point clouds with the Noble detector and CSHOT feature descriptor. This combination obtains the most accurate transformation compared with other methods. For the same combination, the proposed method obtains less than 2 centimetres error whereas original point cloud

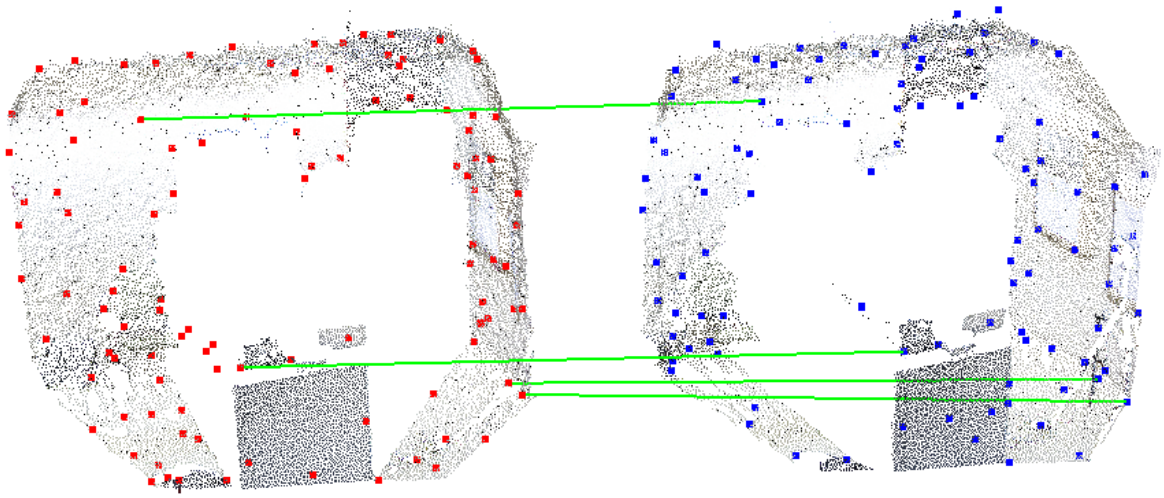

Fig. 6. Registration example done with the Lowe keypoint detector and the FPFH descriptor using a GNG representation with 20000 neurons.

TABLE II

RMS DEVIATION ERROR IN METERS OBTAINED USING DIFFERENT DETECTOR-DESCRIPTOR COMBINATIONS. COMBINATIONS ARE COMPUTED ON THE ORIGINAL POINT CLOUD (RAW), AND DIFFERENT FILTERED POINT CLOUDS USING VOXEL GRID AND THE PROPOSED METHOD.

keypoint_r = 0.15; descriptor_r = 0.2																				
Keypoint detector	**SIFT**				**HARRIS 3D**				**Tomasi**				**Noble**				**Lowe**			
Descriptor	**FPFH**	**CSHOT**	**FPH**	**FPHRGB**	**FPFH**	**CSHOT**	**FPH**	**FPHRGB**	**FPFH**	**CSHOT**	**FPH**	**FPHRGB**	**FPFH**	**CSHOT**	**FPH**	**FPHRGB**	**FPFH**	**CSHOT**	**FPH**	**FPHRGB**
Raw point cloud	0.109	0.047	0.055	0.034	2.201	0.106	0.154	0.059	0.909	0.122	0.125	0.058	2.201	0.106	0.154	0.048	2.201	0.106	0.154	0.048
GNG 20000n 1000p	0.065	0.043	0.088	0.053	**0.058**	0.182	**0.060**	**0.049**	0.810	0.073	**0.088**	0.075	0.060	0.182	0.098	0.108	0.060	0.182	0.098	0.108
GNG 10000n 500p	**0.047**	0.036	0.069	0.042	0.148	**0.035**	0.108	0.062	0.529	0.065	0.305	0.295	0.148	**0.045**	0.108	**0.036**	0.148	0.076	0.108	**0.036**
GNG 5000n 250p	0.160	0.117	0.535	0.035	0.263	0.299	0.125	0.101	**0.081**	**0.054**	0.102	0.189	0.251	0.221	0.125	0.088	0.251	0.457	0.125	0.088
VG 20000	0.086	0.034	0.153	0.040	0.063	0.046	0.065	0.052	0.300	-	0.176	0.081	**0.044**	0.054	**0.088**	0.038	**0.044**	**0.047**	**0.088**	0.038
VG 10000	0.055	**0.022**	0.086	**0.025**	0.213	0.623	0.248	0.053	0.240	0.056	0.105	0.066	0.055	0.881	0.383	0.053	0.055	0.404	0.383	0.053
VG 5000	0.108	0.043	**0.052**	0.030	1.203	0.068	0.084	0.106	0.946	0.151	0.327	**0.049**	0.137	0.092	0.171	0.054	0.137	0.092	0.171	0.054

keypoint_r = 0.1; descriptor_r = 0.2																				
Keypoint detector	**SIFT**				**HARRIS 3D**				**Tomasi**				**Noble**				**Lowe**			
Descriptor	**FPFH**	**CSHOT**	**FPH**	**FPHRGB**	**FPFH**	**CSHOT**	**FPH**	**FPHRGB**	**FPFH**	**CSHOT**	**FPH**	**FPHRGB**	**FPFH**	**CSHOT**	**FPH**	**FPHRGB**	**FPFH**	**CSHOT**	**FPH**	**FPHRGB**
Raw point cloud	0.157	0.036	0.055	0.034	0.307	0.049	0.176	0.045	0.360	0.076	0.245	0.045	0.307	0.066	0.176	0.045	0.307	0.066	0.176	0.045
GNG 20000n 1000p	0.184	0.031	0.088	0.053	**0.116**	0.069	0.053	0.051	2.282	0.042	0.753	0.052	**0.109**	0.065	0.149	0.066	**0.109**	**0.065**	0.149	0.066
GNG 10000n 500p	0.155	0.068	0.069	0.042	0.153	**0.047**	0.136	0.064	0.178	**0.035**	1.727	0.065	0.153	0.079	0.096	**0.034**	0.153	0.079	0.096	**0.034**
GNG 5000n 250p	**0.043**	0.096	0.535	0.035	0.155	0.057	0.058	0.113	0.181	0.073	0.120	0.057	0.344	0.093	0.133	0.081	0.344	0.093	0.133	0.081
VG 20000	0.086	0.029	0.153	0.040	0.771	0.054	**0.032**	**0.046**	0.110	0.054	**0.026**	**0.046**	0.771	-	**0.062**	0.047	0.771	-	0.062	0.047
VG 10000	0.055	**0.020**	0.086	**0.025**	0.338	0.148	0.071	0.058	0.281	0.077	0.238	0.071	0.338	0.117	0.063	0.058	0.338	0.095	**0.050**	0.058
VG 5000	0.108	0.043	**0.052**	0.030	0.355	0.050	0.881	0.045	**0.107**	0.039	0.146	0.063	0.162	**0.060**	0.115	0.087	0.162	0.080	0.115	0.087

keypoint_r = 0.05; descriptor_r = 0.2																				
Keypoint detector	**SIFT**				**HARRIS 3D**				**Tomasi**				**Noble**				**Lowe**			
Descriptor	**FPFH**	**CSHOT**	**FPH**	**FPHRGB**	**FPFH**	**CSHOT**	**FPH**	**FPHRGB**	**FPFH**	**CSHOT**	**FPH**	**FPHRGB**	**FPFH**	**CSHOT**	**FPH**	**FPHRGB**	**FPFH**	**CSHOT**	**FPH**	**FPHRGB**
Raw point cloud	0.157	0.036	0.055	0.034	0.033	0.036	0.117	0.043	0.148	0.053	0.126	0.050	0.106	0.099	0.127	0.066	0.106	0.099	0.127	0.066
GNG 20000n 1000p	0.184	0.031	0.088	0.053	0.117	0.041	0.079	0.056	0.070	0.067	0.088	0.071	0.223	0.052	0.094	0.049	0.223	0.052	0.094	0.049
GNG 10000n 500p	0.155	0.068	0.069	0.042	0.077	0.063	**0.056**	0.070	0.103	**0.026**	0.051	0.064	**0.032**	**0.020**	0.102	0.055	**0.032**	**0.020**	0.071	0.049
GNG 5000n 250p	**0.043**	0.096	0.535	0.035	0.155	0.057	0.486	0.079	0.111	0.045	0.231	0.094	0.060	0.033	**0.037**	0.081	0.060	0.033	**0.037**	0.081
VG 20000	0.086	0.080	0.153	0.040	0.082	0.027	0.151	0.043	**0.070**	0.053	0.136	0.046	0.088	0.063	0.071	0.063	0.141	0.063	0.071	0.084
VG 10000	0.055	**0.020**	0.086	**0.025**	**0.056**	0.062	0.140	**0.037**	0.045	0.044	**0.032**	**0.043**	0.106	0.062	0.101	**0.042**	0.106	0.056	0.101	**0.042**
VG 5000	0.108	0.043	**0.052**	0.030	0.064	**0.022**	0.077	0.056	0.090	0.040	0.071	0.052	0.167	0.025	0.097	0.054	0.167	0.034	0.097	0.054

keypoint_r = 0.02; descriptor_r = 0.2																				
Keypoint detector	**SIFT**				**HARRIS 3D**				**Tomasi**				**Noble**				**Lowe**			
Descriptor	**FPFH**	**CSHOT**	**FPH**	**FPHRGB**	**FPFH**	**CSHOT**	**FPH**	**FPHRGB**	**FPFH**	**CSHOT**	**FPH**	**FPHRGB**	**FPFH**	**CSHOT**	**FPH**	**FPHRGB**	**FPFH**	**CSHOT**	**FPH**	**FPHRGB**
Raw point cloud	0.157	0.036	0.055	0.038	0.103	0.039	0.073	0.026	0.064	0.026	0.079	0.042	0.078	0.046	0.083	**0.025**	0.078	0.046	0.083	0.167
GNG 20000n 1000p	0.184	0.031	0.088	0.053	0.066	0.036	0.175	0.049	0.101	0.033	**0.049**	**0.018**	0.071	0.030	0.081	0.060	0.071	0.030	0.081	0.060
GNG 10000n 500p	0.155	0.068	0.069	0.042	0.084	0.050	0.061	0.036	**0.018**	0.030	0.084	0.044	0.064	**0.017**	0.086	0.063	0.064	0.017	0.086	0.063
GNG 5000n 250p	**0.043**	0.096	0.535	0.035	0.099	0.046	**0.034**	0.058	0.049	0.038	0.064	0.065	0.101	0.020	**0.034**	0.058	0.101	**0.020**	**0.034**	0.058
VG 20000	0.086	0.052	0.153	0.040	**0.036**	**0.032**	0.195	0.044	0.060	0.042	0.065	0.050	**0.056**	0.026	0.134	0.046	**0.056**	0.026	0.134	0.046
VG 10000	0.055	**0.022**	0.086	**0.025**	0.084	0.034	0.081	0.034	0.074	**0.026**	0.072	0.027	0.058	0.039	0.074	0.036	0.058	0.040	0.074	**0.036**
VG 5000	0.108	0.044	**0.052**	0.030	0.123	0.044	0.073	**0.027**	0.123	0.040	0.073	0.027	0.123	0.031	0.088	0.036	0.123	0.039	0.088	0.036

produces almost 6 centimetres error in the best case.

V. CONCLUSIONS AND FUTURE WORK

In this paper we have presented a filtering and down-sampling method which is able to deal with noisy 3D data captured using low cost sensors like the Kinect Device. The proposed method calculates a GNG network over the raw point cloud, providing a 3D structure which has less information than the original 3D data, but keeping the 3D topology. We demonstrated how the proposed method obtains better adaptation to the input space than other filtering methods like Voxel Grid, obtaining lower adaptation MSE

on simulated scenes and CAD models. Moreover, it is shown how state-of-the-art keypoint detection algorithms perform better on filtered point clouds using the proposed method. Improved keypoint detectors are tested in a 3D scene registration process, obtaining lower transformation RMS errors in most detector-descriptor combinations. The most accurate transformations between different scenes are obtained using the proposed method.

Future work includes the integration of the proposed filtering method in a indoor mobile robot localization application.

REFERENCES

[1] A. M. Treisman and G. Gelade, "A feature-integration theory of attention," *Cognitive Psychology*, vol. 12, no. 1, pp. 97–136, Jan. 1980. [Online]. Available: http://dx.doi.org/10.1016/0010-0285(80)90005-5

[2] M. Szummer and R. Picard, "Indoor-outdoor image classification," in *Content-Based Access of Image and Video Database, 1998. Proceedings., 1998 IEEE International Workshop*, jan 1998, pp. 42 –51.

[3] H. Tamimi, H. Andreasson, A. Treptow, T. Duckett, and A. Zell, "Localization of mobile with omnidirectional vision using particle filter and iterative sift," *Robotics and Autonomous Systems*, vol. 54, no. 9, pp. 758 – 765, 2006.

[4] A. Nuchter, H. Surmann, K. Lingemann, J. Hertzberg, and S. Thrun, "6d slam with an application in autonomous mine mapping," in *In Proceedings of the IEEE International Conference on Robotics and Automation*, 2004, pp. 1998–2003.

[5] L. Kobbelt and M. Botsch, "A survey of point-based techniques in computer graphics," *Computers & Graphics*, vol. 28, pp. 801–814, 2004.

[6] C. Connolly, "Cumulative generation of octree models from range data," in *Robotics and Automation. Proceedings. 1984 IEEE International Conference on*, vol. 1, mar 1984, pp. 25 – 32.

[7] R. Martin, I. Stroud, and A. Marshall, "Data reduction for reverse engineering," *RECCAD, Deliverable Document 1 COPERNICUS project, No 1068*, p. 111, 1997.

[8] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proceedings of the Sixth International Conference on Computer Vision*, ser. ICCV '98. Washington, DC, USA: IEEE Computer Society, 1998, pp. 839–846.

[9] J. Wasza, S. Bauer, and J. Hornegger, "Real-time preprocessing for dense 3-d range imaging on the gpu: Defect interpolation, bilateral temporal averaging and guided filtering," in *ICCV Workshops*, 2011, pp. 1221–1227.

[10] B. Fritzke, *A Growing Neural Gas Network Learns Topologies*. MIT Press, 1995, vol. 7, pp. 625–632.

[11] Y. Holdstein and A. Fischer, "Three-dimensional surface reconstruction using meshing growing neural gas (mgng)," *Vis. Comput.*, vol. 24, no. 4, pp. 295–302, Mar. 2008. [Online]. Available: http://dx.doi.org/10.1007/s00371-007-0202-z

[12] D. Viejo, J. Garcia, M. Cazorla, D. Gil, and M. Johnsson, "Using gng to improve 3d feature extraction-application to 6dof egomotion." *Neural Netw*, 2012.

[13] K. Khoshelham and S. O. Elberink, "Accuracy and resolution of kinect depth data for indoor mapping applications," *Sensors*, vol. 12, no. 2, pp. 1437–1454, 2012. [Online]. Available: http://www.mdpi.com/1424-8220/12/2/1437

[14] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[15] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *Computer Vision ECCV 2002*, ser. Lecture Notes in Computer Science, A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, Eds. Springer Berlin Heidelberg, 2002, vol. 2350, pp. 128–142.

[16] I. Sipiran and B. Bustos, "Harris 3d: a robust extension of the harris operator for interest point detection on 3d meshes," *Vis. Comput.*, vol. 27, no. 11, pp. 963–976, Nov. 2011. [Online]. Available: http://dx.doi.org/10.1007/s00371-011-0610-y

[17] J. Noble, "Finding corners," *Image and Vision Computing*, vol. 6, pp. 121–128, 1988.

[18] J. Shi and C. Tomasi, "Good features to track," in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*. IEEE, 1994, pp. 593–600.

[19] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz, "Aligning Point Cloud Views using Persistent Feature Histograms," in *Proceedings of the 21st IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Nice, France, September 22-26*, 2008.

[20] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, may 2009, pp. 3212 –3217.

[21] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *Proceedings of the 11th European conference on computer vision conference on Computer vision: Part III*, ser. ECCV'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 356–369. [Online]. Available: http://dl.acm.org/citation.cfm?id=1927006.1927035

[22] F. Tombari and Salti, "A combined texture-shape descriptor for enhanced 3d feature matching," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*, sept. 2011, pp. 809 –812.

[23] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[24] R. B. Rusu and S. 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.

[25] M. Gschwandtner, R. Kwitt, A. Uhl, and W. Pree, "BlenSor: Blender Sensor Simulation Toolbox Advances in Visual Computing," ser. Lecture Notes in Computer Science, G. Bebis, R. Boyle, B. Parvin, D. Koracin, S. Wang, K. Kyungnam, B. Benes, K. Moreland, C. Borst, S. DiVerdi, C. Yi-Jen, and J. Ming, Eds. Berlin, Heidelberg: Springer Berlin / Heidelberg, 2011, vol. 6939, ch. 20, pp. 199–208. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-24031-7_20

[26] M. Jenkinson, "Measuring transformation error by rms derivation," Oxford Centre for Functional Magnetic Resonance Imaging of the Brain (FMRIB), Departament of Clinical Neurology, University of Oxford, John Radcliffe Hospital, Headley Way, Headington, Oxford, UK, Tech. Rep., 2003.