Local-to-Global Signature Descriptor for 3D Object Recognition

Isma Hadji[1], and G. N. DeSouza[2],

Vision-Guided and Intelligent Robotics Lab (ViGIR)
Electrical and Computer Engineering Dept.
University of Missouri
Columbia, MO, USA, 65211
[1]ih9p5@mail.missouri.edu, [2]DeSouzaG@missouri.edu

Abstract. In this paper, we present a novel 3D descriptor that bridges the gap between global and local approaches. While local descriptors proved to be a more attractive choice for object recognition within cluttered scenes, they remain less discriminating exactly due to the limited scope of the local neighborhood. On the other hand, global descriptors can better capture relationships between distant points, but are generally affected by occlusions and clutter. So, we propose the Local-to-Global Signature (LGS) descriptor, which relies on surface point classification together with signature-based features to overcome the drawbacks of both local and global approaches. As our tests demonstrate, the proposed LGS can capture more robustly the exact structure of the objects while remaining robust to clutter and occlusion and avoiding sensitive, low-level features, such as point normals. The tests performed on four different datasets demonstrate the robustness of the proposed LGS descriptor when compared to three of the SOTA descriptors today: SHOT, Spin Images and FPFH. In general, LGS outperformed all three descriptors and for some datasets with a 50-70% increase in Recall.

1 Introduction

Object recognition is arguably the most important topic in the field of computer vision because of the different applications that this task can serve. Such applications include, scene understanding, robot navigation, tracking, assistive technology and many others. This importance has also grown in the past decade as research has steered towards the use of 3D instead of 2D information for the task of object recognition. But as in the years of 2D object recognition, designing good feature vectors, or descriptors, is still the most critical step involved in 3D object recognition. Indeed, the descriptors have the greatest effect on the overall recognition result, as argued in [1].

The techniques adopted for object description can be divided into two main categories; global or local. At first, global descriptors seem a more natural choice, since they seek to encode the entire set of keypoints into a single feature vector describing the object. This makes global features far more discriminating given

that the entire geometry of the object is taken into account. However, being able to observe as much as possible of this same geometry becomes a condition for the success of global descriptors. On the other hand, local descriptors rely only on the local neighborhood of each keypoint, making the recognition more robust in scenes with clutter and occlusion [1], but for the price of increased sensitivity to changes in those neighborhoods (e.g. instrument noise). Another advantage of local descriptors is in the ability to perform point to point correspondences, which makes pose estimation using, for example RANSAC, much easier.

In this research we propose a new descriptor built at the up-to-now unseemly intersection between these two paradigms. The Local-to-Global Signature – or LGS descriptor – is local in the sense that each feature vector describes a single keypoint, rather than the entire object, but at the same time global as it looks beyond the local neighborhood (support regions) to describe the properties of keypoints with respect to the entire object. Unlike traditional global descriptors, the LGS overcomes issues related to occlusion and clutter by constructing signature vectors instead of histogram-based vectors. The advantages of using signatures to reduce the effects of occlusion are further detailed in Section 3, but again, that is not the only contribution of the proposed descriptor. In summary, the LGS was built around the following five main ideas: 1) Relying on surface point classification to capture the entire geometry of the object (global property); 2) Describing keypoints (local property), but using both local and global support regions grouped by the same surface class; 3) Using signatures (global property) to avoid loss of information and mitigate the effects of occlusion; 4) Using distributions of L2-distances to encode the relationship between keypoint and support regions to increase robustness to noise and eliminate the use of sensitive features such as surface normals; and 5) Using confidence on the relationship above to improve the matching during object recognition.

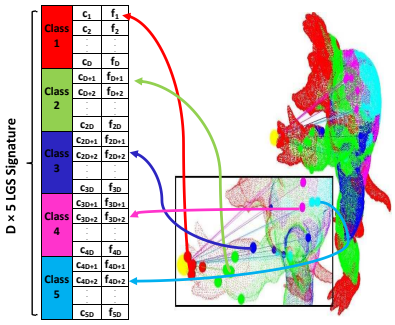


Fig. 1. Steps of the construction of the LGS descriptor: yellow dots represent keypoints to be described; points are color coded according to the surface class to which they belong, from sharp regions in red to smooth surfaces in light blue; the signature consists of the distances f between the keypoint and its local and global support regions, plus the corresponding confidences in their assignments.

Figure 1 illustrates the overall construction of the LGS descriptor. These ideas will be further detailed in the remaining of the paper which is organized as follows: Section 2 briefly surveys the most prominent 3D descriptors used in the field. In section 3, we introduce the LGS descriptor and highlight the different steps involved in constructing it. Section 4 describes in details the experiments, the datasets used and presents all the results obtained. Finally, we conclude with a discussion on the advantages of the proposed approach, its limitations as well as possible directions for future work.

2 Background and Related Work

As briefly mentioned in Section 1, 3D descriptors belong to two main categories; global and local. Local descriptors focus on the local neighborhood of keypoints, while Global descriptors represent the complete object by either using all or a subsample of the detected keypoints. The techniques used to encode the relationships between keypoints can also be divided in two sub-categories: signature-based and histogram-based. A detailed comparison between these two sub-categories can be found in [2].

Among the first successful local 3D descriptors employing histograms we find the Point Feature Histogram (PFH) [3]. PFH describes the angular variations between each two surface normals in the k-neighborhood of a region of interest. Because this descriptor models variations between normals at all pairs of points, it is very computationally intensive. For that reason, Fast PFH (FPFH) was introduced in [4] to speed up the process, but in detriment to its power to discriminate objects. An alternative was then introduced in [5], where the authors proposed a 3D extension of the 2D Shape Context descriptor [6]. This descriptor relies on the position of keypoints in a local neighborhood (support region) with respect to a virtual spherical grid. Each spherical grid accumulates a weighted sum of the points falling in that grid. In order to achieve repeatability, the north pole of the spherical grid is first aligned with the direction of the surface normal at the point being described. This alignment is sensitive to the choice of the reference frame – a major issue for any descriptor based on surface normals. So, a major breakthrough in terms of 3D descriptors was introduced by the SHOT descriptor ([2]), where the authors addressed this problem with the reference frame repeatability by attaching a unique reference frame to each point. Once the reference frame is determined, the local support of each point is discretized in a way similar to that in the 3D Shape Context descriptor. Another contribution of SHOT was that the histogram of each grid is concatenated to form the final signature. This technique for defining local repeatable reference frames was later used to improve the 3D Shape Context in [7] by taking the initial representation proposed for 3D Shape Context and disambiguate it using the unique reference frame introduced by the SHOT descriptor. Although quite different in terms of their implementations, the afore mentioned techniques are very similar in concept. Their reliance on local neighborhoods and low-level fea-

tures, such as surface normals, compromise their ability to discriminate objects and make them quite sensitive to viewing perspectives and noise.

As far as signature-based descriptors are concerned, Point Signatures (PS) [8] is possibly the best known descriptor. It relies on point positions also within a local neighborhood, but it encodes these positions in the local sphere in terms of the angle between the surface normal at the point and the signed distance of the points to the plane separating the sphere into two halves. While this method captures exactly the structure of the local neighborhood, it remains very sensitive to small changes in the normal estimation and the reference frame. Another signature-based descriptor is proposed in [9], where signatures are built from the depth values of the local surface after its normal vector has been aligned with the Z axis. Also, in order to reduce the dimensionality of the signature, the method resorts to a PCA subspace of the feature vector. Although signatures usually lead to a better discriminating power, the reliance on local neighborhoods causes low repeatability in keypoint matching since many keypoints in one object can have similar local structures.

Finally, when it comes to global descriptors, the View-point Feature Histogram (VFH) [10], which is an extension to the FPFH, is one of the most widely used. Instead of describing relationships between points in local neighborhoods, VFH does so for every point in the cloud with respect to their centroid. In addition to this, VFH introduces view point variance – an important addition when it comes to estimating object pose. Another attempt of generalizing FPFH was provided in Global FPFH [11]. In this case, FPFH descriptors are used locally for every point in the object and a conditional random field is trained in order to classify the collected local descriptors into a set of primitives. Next, the relationships between keypoints are encoded by counting the number and type of transitions between primitives while traversing keypoints in an octree. Later, in Global Radius-based Surface Descriptor, GRSD ([12]), a modification to GFPFH was proposed by eliminating the first classifier and adding a geometric solution to the shape primitives. A similar approach called Global Structure Histogram, or GSH, was also proposed in [13] with the goal of capturing the global structure of the object. In order to do so, the authors followed the same procedure proposed in [11], but using a clustering algorithms to learn the points classes and by encoding the relationships between keypoints in all clusters using geodesic distances.

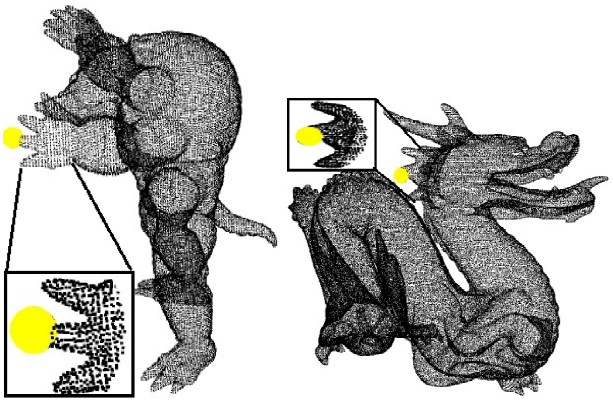
In this work, we maintain that adopting the strengths of both local and global descriptors can lead to highly discriminating features. These features can be successfully used in the task of object recognition and pose estimation within scenes, even if they suffer from clutter and occlusion.

3 Proposed Method

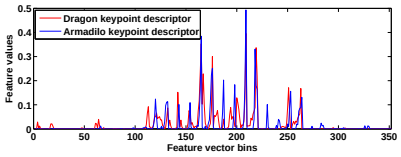
3.1 Motivation

The proposed LGS descriptor can be regarded as a bridge between local and global paradigms. Recently, global 3D descriptors introduced interesting ideas

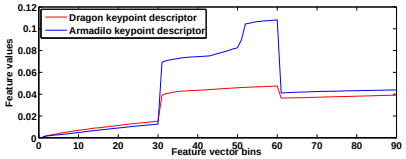
involving the classification of object surfaces based on their shapes. These ideas should ultimately increase the discriminating power of the descriptor. However, it is clear that a great amount of information is lost in the process of encoding the relationships between all keypoints of an object into a single feature vector. In addition to that, global descriptors are very sensitive to clutter and they require a good segmentation algorithm as a pre-processing step ([1]). Also, even if a perfect segmentation could be obtained, occlusions can have a major effect on the performance of global descriptors. On the other hand, a common problem that may affect all local descriptors is the inability to discriminate similar support regions not only within the same object but also across different objects. To illustrate this idea, we extracted both the SHOT and the LGS descriptors from two different keypoints coming from two completely different objects. We set the radius size of the support region used to estimate the SHOT descriptors to 25 times the mesh resolution – i.e. almost twice the recommended size of 15 times the mesh resolution [2]. As Figure 2 clearly shows, the SHOT descriptors for these two keypoints are almost identical even though they come from very different objects, while the LGS makes clear distinction between the same two keypoints.



(a) Locally similar keypoints from two different objects



(b) SHOT feature vector



(c) LGS feature vector

Fig. 2. Contrast between the SHOT and the proposed LGS descriptor: (a) two keypoints (shown in yellow) falling on similar local neighborhoods, but coming from two completely different objects. Notice the similarity between the two, red and blue, SHOT descriptors in (b) as opposed to the clear contrast between the LGS descriptors in (c), also in red and blue.

For these reasons, the proposed descriptor starts from a local approach, where each keypoint is represented with a unique feature vector – i.e descriptor. Then, it turns global by looking at the entire object while capturing the structure of the object with respect to that keypoint.

3.2 LGS descriptor

As we just mentioned, the main advantage of the LGS descriptor resides in the idea of looking beyond the local support. In fact, it looks at different regions representing different properties of the object. This is accomplished by: (i) assigning classes to all the points in the cloud (global property); (ii) describing keypoints (local property), but using both local and global support regions; (iii) using signatures to describe the relationships between keypoints and selected points from all the assigned classes; (iv) using L2 norm to robustly encode such relationships; and (v) using confidence on the relationship above during matching.

Point Classification

To classify the points we use the radius-based surface classification proposed in the RSD descriptor ([12]). Our technique starts form the assumption that every two points fall on a sphere. Therefore, within each neighborhood the radii of all virtual spheres are estimated using points locations. Then, the maximum and minimum sphere radii present in the local neighborhood are derived. For more details we refer the reader to [12].

In the original proposal of RSD, the authors used both the minimum and maximum radii to classify points as belonging to one of the geometric primitive shapes. In our implementation however, we chose to classify points from very sharp to very smooth. Our decision for this classification approach is motivated by one main argument: to be able to find a pre-defined number of classes independently of the object considered. This may not be feasible when using shape primitives where some specific shapes may not appear in all objects. In addition, since this classification scheme is independent of primitive shapes, it allows the algorithm to vary the number of classes by simply varying the ranges of sharpness and smoothness. For example, we cannot expect to find a toroid shape in all objects, but we can reasonably assign different levels of sharpness or smoothness to any surface based on its curvature.

In our implementation, we rely only on the minimum radius from the RSD method described above, where a very small radius is an indication of a very sharp surface, and a large radius represents a smooth one. After deriving the minimal radii associated with each point on the surface of the object, the algorithm can split the radii values into N different ranges, representing the N classes of the object's surfaces. If for example $N = 3$, any point with radius $r < \alpha * mesh\,resolution$ would be assigned to class 1; a point with radius $\alpha * mesh\,resolutionr < r < (\alpha + 5) * mesh\,resolution$ is assigned to class 2; and all other points are assigned to class 3. In our experiments, the value of α was empirically set to 6.

Since LGS uses continuous ranges of radii values for surface point classification, fuzzy regions may emerge. These fuzzy regions contain points with values of radii that are close to the end of one range and the beginning of the next one. Therefore, these points could belong to any of the two consecutive ranges. It is easy to understand that these regions are unstable and points can move from one class to another as noise is added or removed. Also, noise can cause spikes to appear on otherwise smooth surfaces. Therefore, to cope with these potential instability in surface classification, we propose the assignment of both a class and a membership to the class for each point in the cloud. Basically, after the initial crisp classes are assigned to each point, the algorithm searches over each point and calculates a coefficient of confidence that the specific point belongs to the assigned class. These confidences c approximate the probability that a current point p belongs to class n given the number of points from that class in its local support region; that is:

$$c = \frac{\#\, of\ points\ in\ class\ n\ in\ the\ neighborhood\ of\ p}{Total\ number\ of\ points\ in\ the\ neighborhood\ of\ p} \tag{1}$$

The rational behind these confidences is the assumption that in any small neighborhood points are more likely to belong to the same surface class. Also, a low confidence indicates that the point belongs to a fuzzy or noisy region, where multiple classes may be assigned. This determines when the algorithm is to give high or low weights to the points used in the next steps of the algorithm.

Signature Construction

Once all points are assigned to one of the N classes and their confidences are calculated, the LGS descriptor is constructed in a signature-based fashion. Figure 1 should help the reader in understanding the following steps. First, for every keypoint, the algorithm finds its corresponding k-nearest neighbors within each class. Next, the k-neighbors in each class are sorted based on their distance from the keypoint and divided into D clusters. Then, the median L2-distances from the keypoint to the points falling in each cluster form a D dimensional feature vector. These distances are the actual features f_l of the LGS descriptor. Finally, feature vectors representing the neighborhoods of the keypoint in each one of the N classes are concatenated to form the final signature whose length is equal to $D \times N$. Again, Figure 1 illustrates the construction of a simplified signature in the LGS descriptor. It is important to mention again that the main motivation for using a signature-based descriptor is its potential robustness to occlusions. In fact, if parts of the object happen to be occluded in the scene, only the entries corresponding to those parts of the object will be altered in the LGS descriptor, while the rest is unchanged. On the other hand, a histogram-based descriptor would be completely affected if the support region of a keypoint is partially occluded.

Descriptor Matching

In parallel to constructing the LGS descriptor, the algorithm builds a second feature vector filled with the confidences c_l corresponding to each feature point

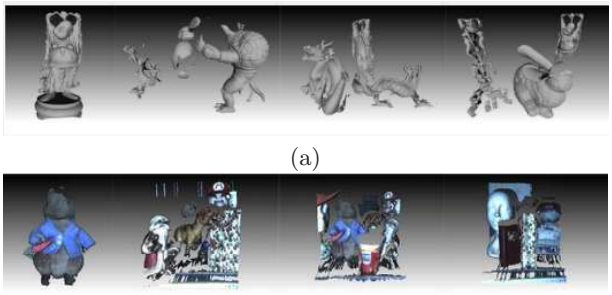
f_l. Once again, this idea is illustrated in Figure 1 for a very simplified case. These confidences are used as weights during the matching stage, when the LGS computes the distance d_{ij} between a pair of signatures (i, j). In other words, the distance between each entry of the pair of signatures is multiplied by the corresponding minimum confidence. This allows LGS to reduce the effect of unstable points located on fuzzy regions as discussed in the beginning of this section. Mathematically, the weighted distance used to compare LGS descriptors is given by:

$$d_{ij} = sqrt(\sum_{l=1}^{(D*N)} min(c_{li}, c_{lj}) * (f_{li} - f_{lj})^2) \tag{2}$$

4 Experimental Results and Discussion

In order to evaluate the discriminating power of the proposed LGS signature, we devised testing scenarios that highlight the robustness of LGS for the case of a model-scene matching framework.

In that sense, we present two main experiments to validate our work, using two well-recognized datasets from the literature [14, 2]. The first dataset referred in [14] as Retrieval dataset consists of synthetic data with added noise at three levels equal to 10%, 30% and 50% of the mesh resolution. The second dataset, and the most challenging one, contains real data acquired using stereo cameras. It presents the biggest challenge for any descriptor due to: the level of occlusion and clutter; the presence of smoother and more similar objects, and the larger reconstruction errors between scenes and models. Figure 3 presents examples of models and scenes from these two datasets.



(a)

(b)

Fig. 3. Sample models and scenes. (a) Stanford synthetic dataset. (b) UniBo Vision Lab Stereo dataset. [14]

4.1 Experiment 1 - Parameter Selection

As explained in Section 3, the proposed LGS descriptor may be affected by the choice of two parameters: the number of classes N used to construct the signature and the number k of neighbors used in each class. In this section, we present an experiment highlighting the effect of each one of these parameters on the performance of the LGS descriptor. This also provides the reader with an intuition regarding how they can be set.

In general, a descriptor T is considered robust if for any given keypoint in the scene, its exact correspondence can be found in the model. The steps followed for establishing correspondences are as follows:

1. For each model-scene pair (m, s), find the set of keypoints k_m and k_s and describe each keypoint using the T descriptor.
2. Compare the descriptor for each keypoints in the scene against all the descriptors in the model and take as a correspondence the descriptor from the model with the shortest distance – for the LGS, that is given by eq (2).
3. Find the true correspondences using the provided ground-truth transformation between model and scene.
4. Compare the correspondences from step 2 and step 3 and count total number of true/false matches.

In order to abstract the performance of the descriptor from the effect of non-repeatable keypoints, we followed the same framework proposed in [2] for keypoint detection. More specifically, we down-sampled each model so that only 5% of the original cloud was kept. These will be the model keypoints. Then, using the ground-truth transformation, we found the corresponding keypoints in the scene. These steps guaranteed 100% repeatable keypoints.

We opted for down-sampling the keypoints to ensure that these are uniformly distributed over the object and that the algorithm is not biased by the type or location of the keypoints.

Number of Classes As previously mentioned the first step towards building the LGS signature is to classify points based on the surface type to which they belong. In particular, we classify surfaces from very sharp to very smooth. In this experiment, we highlight the effect of the number of classes by varying them from 1 to 5 classes, and allowing accordingly from 1 to 5 shades of sharpness/smoothness. Figure 4 summarizes the results obtained for variable number of classes when the number of neighbors is fixed at 300 nearest neighbors in each class. It is worth noting that for a number of classes equal to 1, the LGS descriptor approaches a local signature-based descriptor where all nearest neighbors are close to the described keypoint.

As we can see from Figure 4, the number of classes plays an important role in the performance of the LGS. This becomes particularly clear for the Retrieval dataset, where changing from two to three classes improves the percentage of good correspondences by as much as 15% (at noise level equal to 0.5). This

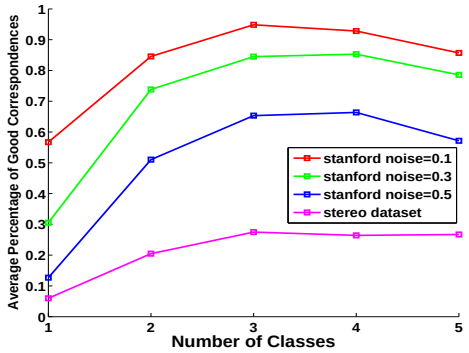


Fig. 4. Average percentage of good correspondences versus number of classes.

dataset contains objects with highly different surfaces and shapes, therefore a small number of classes is not discriminating enough. As one would expect, a small number of classes does not capture enough of the variations in the structure of these objects. On the other hand, adding too many classes also leads to reduced discriminating power. We attribute this to the drop in the classification accuracy that affects the regions used in building the signature. In fact, increasing the number of classes involves using smaller ranges to assign points to different surface types. Therefore, forcing the presence of more fuzzy and unstable regions whose points can be assigned to any one of the neighboring ranges.

In order to support this claim, we performed an evaluation of the classification accuracy versus the number of classes. For each point in the model, we let the algorithm find its corresponding point in the scene and we checked whether they had been assigned to the same class (see Figure 5). As we claimed, by adding more classes we observed a drop in the classification accuracy, which ultimately caused a drop in the number of good correspondences.

K-Neighborhoods The LGS descriptor consists of concatenating features from the k-closest neighbors in each class. So, the number of neighbors does play a role in the stability of the signature. In fact, a very small neighborhood implies relying more heavily on points that fall close to fuzzy regions – i.e. junction regions between two classes. As mentioned earlier in Section 3, these are non-stable regions since points on those regions can switch classes very easily in the presence of sensor noise. On the other hand, neighborhoods too large can cause the algorithm to look beyond stable regions, likely falling on the next fuzzy region. In addition to that, very large neighborhoods may imply using much bigger parts of the object which can therefore cause the LGS signature to become more affected by occlusions.

Once again we validate these claims by evaluating the effect of different neighborhood sizes on the discrimination of the LGS descriptor. Given the results obtained in the previous experiment, we fixed the number of classes to 3 for this

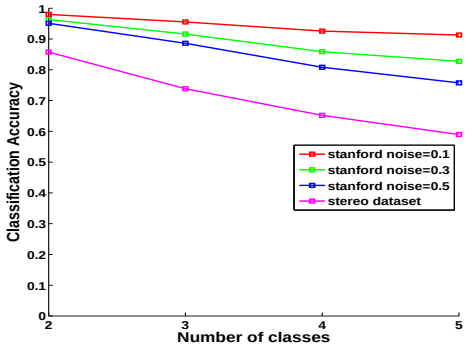


Fig. 5. Classification accuracy versus number of classes

test. We varied the neighborhood sizes from 100- to 1000-nearest neighbors per class. It should be noted here that given the definition of the LGS signatures, varying the neighborhoods sizes implies varying the length of the signatures. In particular, as previously mentioned, in the algorithm for the LGS, each neighborhood is split into smaller clusters and the median distance of each small cluster is used. For example, in our implementation, we used clusters with 10 points each, therefore for a number of classes $N = 3$ and a neighborhood size $k = 100$, we have the number of clusters in each class $D = 100/10$, leading to a signature with dimension $D * N = 30$.

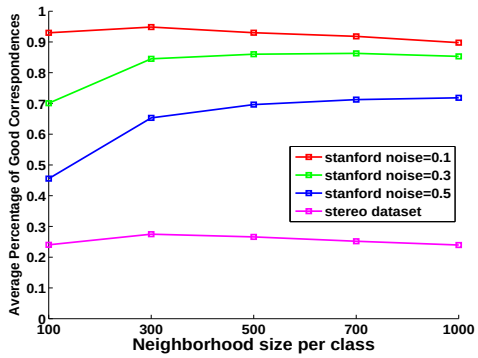


Fig. 6. Average percentage of good correspondences versus neighborhood size per class. In this test we used 3 classes.

Figure 6 summarizes the results obtained from varying the neighborhoods sizes. These results prove two main points: (i) we can see that the percentage of good correspondences increases in the beginning as we use larger neighborhoods.

Again here, this is more evident with the Retrieval dataset where increasing the neighborhood size from 100 to 300 per class improves the percentage of correspondences by about 20% in some cases (at noise level equal to 0.5). This confirms our initial statement regarding the importance of looking beyond the local neighborhood when constructing local descriptors; (ii) interestingly enough, we can see that the LGS descriptor is not too sensitive to the use of larger neighborhoods. We attribute this to the advantages brought by using a signature-based feature vectors, as well as the use of class confidences as weights during the matching stage. This could be seen as making the LGS less sensitive to occlusions.

4.2 Experiment 2 - Matching capability

In this experiment, we validate our proposal in terms of matching capability, using the *Recall* versus $1 - Precision$ metric that captures both true and false positives as argued in [2, 15, 16]. In order to find potential correspondences, for every feature in the scene, the algorithm computes the first and second nearest neighbors in the model. Then, a match is established between the scene feature vector and its nearest neighbor in the model if the ratio between the first two nearest neighbors is bellow a certain threshold, as suggested in [2, 17]. This threshold is the value that is varied from $0\,to\,1$ in order to produce the *Recall* versus $1 - Precision$ curves in Figure 7. A correct match is counted as a true positive, and as a false positive otherwise. The total number of correspondences is known from the ground truth and therefore *Recall* and $1 - Precision$ are calculated as follows:

$$Recall = \frac{True\,Positives}{Total\,number\,of\,correspondences} \tag{3}$$

$$1 - Precision = \frac{False\,positives}{False\,positives + True\,Positives} \tag{4}$$

We include in this experiment a quantitative comparison against three SOTA descriptors: i) FPFH ii) Spin Images and iii) SHOT. We chose FPFH and Spin Images as representatives of histogram-based local 3D descriptor and also because they are arguably the most widely used descriptors in the field. Also, we selected SHOT for comparison with a descriptor based on a signature of histograms, and again because it represents one of the state-of-the-art local 3D descriptors, achieving the best results on the datasets used here.

For the LGS descriptor, we picked the best parameters learnt from the previous experiment. In particular, we use number of classes $N = 3$ with the number of neighbors per class k set to 300. Given that we split each neighborhood to clusters of 10 points, this lead to a 90-dimensional signature. For the other descriptors used in this comparison, the main parameter to be set is the radius size of the local support region of the keypoints. In this case, we used the same size recommended in [2] – i.e. 15 times the mesh resolution.

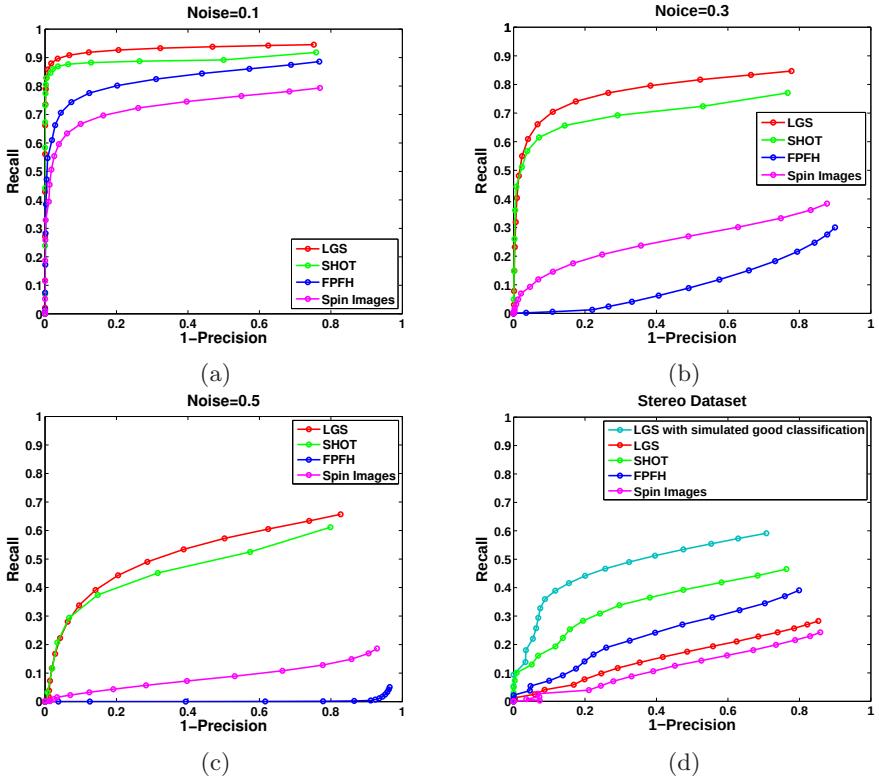


Fig. 7. Quantitative comparison between descriptors using the Recall vs 1-Precision metric for (a) Retrieval Dataset with noise=0.1, (b) noise=0.3, (c) noise =0.5, and (d) The Stereo dataset.

As the results demonstrate, in most cases the LGS proved to be more discriminating than the different local descriptors tested in this paper. This should support the claim regarding the importance of looking beyond the local neighborhood for keypoint description. In addition, the use of signature proved to hold a higher discriminating power since LGS and SHOT always outperformed the histogram-based approaches. Also, given that the LGS descriptor does not directly rely on sensitive features, such as point normals, it was less affected by noise than SHOT as Figures 7 (a) through (c) demonstrate – we discuss case (d), next.

One limitation of using the proposed approach is highlighted in the last experiment using the Stereo Dataset. In this case, the results for the LGS where only better than the Spin Images. In fact, as previously mentioned in Section 4.1, for the Stereo dataset, both models and scenes are reconstructed from different stereo pairs, which causes differences in the shapes of the objects found in the scene and the model. This in turn leads to a less accurate surface classification (Figure 5), which ultimately affects the LGS signature. In fact, many of the

points in the model are assigned different classes from the ones in the scene. As a consequence, the support regions found by the algorithm in each class of the model are also different from the ones found in the scene.

In order to further illustrate the effect of this severe mis-classification observed on this last dataset, we simulated 100% good classification of the points. Specifically, we first classified points in the models into 3 classes as described in Section 3. Then, using the ground-truth transformation, we assigned the same class to the corresponding points in the scenes. Finally, using this simulated classification results, we constructed the LGS descriptor as usual. As can be seen from Figure 7(d), provided a good classification, the LGS signature can still achieve higher discrimination than any of the tested descriptors. While this can be seen as a limitation of our method and indeed requires future improvements, it also further proves the benefits of looking beyond the local neighborhood and encoding more of the structure of objects in describing each keypoint. In addition, the four experiments together prove that relying on a signature instead of a histogram allows LGS to capture more details, increasing therefore the discriminating capability of the descriptor.

5 Conclusion

In this paper we proposed a novel signature-based 3D keypoint descriptor that bridges the gap between local and global descriptors. That the LGS addresses some of the difficulties inherent to each of these paradigms became clear when we compared the proposed LGS to the SOTA descriptors. In addition, we highlighted the benefits of using signatures and their role in avoiding great loss of information – as opposed to what happens with histogram-based approaches. Finally, we showed that the relative positions of the keypoints with respect to local and global support regions hold enough discriminating power while they replace low-level features such as point normals, which are very sensitive to noise. Our results clearly demonstrate the relevance of the proposed LGS descriptor while highlighting some of its limitations. Future directions for this research include addressing the problems caused by data acquisition and sensor noise when it comes to mis-classification of points in the support regions. In particular, we believe that a better surface classification scheme can lead to much more robust signatures as demonstrated in Figure 7 (d). An automatic selection of the number of classes N, and the testing of LGS descriptor for the entire object-recognition pipeline against the descriptors used here, and potentially others (e.g. GSH, GFPFH, and GRSD) should be also part of our future work.

References

1. Aldoma, A., Marton, Z.C., Tombari, F., Wohlkinger, W., Potthast, C., Zeisl, B., Rusu, R.B., Gedikli, S., Vincze, M.: Tutorial: Point cloud library: Three-dimensional object recognition and 6 dof pose estimation. IEEE Robot. Automat. Mag. **19** (2012) 80–91

2. Tombari, F., Salti, S., Di Stefano, L.: Unique signatures of histograms for local surface description. In: Proceedings of the 11th European Conference on Computer Vision Conference on Computer Vision: Part III. ECCV'10, Berlin, Heidelberg, Springer-Verlag (2010) 356–369
3. Rusu, R.B., Marton, Z.C., Blodow, N., Beetz, M.: Persistent point feature histograms for 3d point clouds. In: Proceedings of the 10th International Conference on Intelligent Autonomous Systems (IAS-10), Baden-Baden, Germany. (2008)
4. Rusu, R., Blodow, N., Beetz, M.: Fast point feature histograms (fpfh) for 3d registration. In: Robotics and Automation, 2009. ICRA '09. IEEE International Conference on. (2009) 3212–3217
5. Frome, A., Huber, D., Kolluri, R., BÃŒlow, T., Malik, J.: Recognizing objects in range data using regional point descriptors. In: EUROPEAN CONFERENCE ON COMPUTER VISION. (2004) 224–237
6. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. IEEE Trans. Pattern Anal. Mach. Intell. **24** (2002) 509–522
7. Tombari, F., Salti, S., Di Stefano, L.: Unique shape context for 3d data description. In: Proceedings of the ACM Workshop on 3D Object Retrieval, New York, NY, USA, ACM (2010) 57–62
8. Chua, C.S., Jarvis, R.: Point signatures: A new representation for 3d object recognition. Int. J. Comput. Vision **25** (1997) 63–85
9. Mian, A., Bennamoun, M., Owens, R.: On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes. International Journal of Computer Vision **89** (2010) 348–361
10. Rusu, R., Bradski, G., Thibaux, R., Hsu, J.: Fast 3d recognition and pose using the viewpoint feature histogram. In: Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on. (2010) 2155–2162
11. Rusu, R.B., Holzbach, A., Bradski, G., Beetz, M.: Detecting and segmenting objects for mobile manipulation. In: Proceedings of IEEE Workshop on Search in 3D and Video (S3DV), held in conjunction with the 12th IEEE International Conference on Computer Vision (ICCV), Kyoto, Japan (2009)
12. Marton, Z.C., Pangercic, D., Blodow, N., Beetz, M.: Combined 2D-3D Categorization and Classification for Multimodal Perception Systems. The International Journal of Robotics Research **30** (2011) 1378–1402
13. Madry, M., Ek, C., Detry, R., Hang, K., Kragic, D.: Improving generalization for 3d object categorization with global structure histograms. In: Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on. (2012) 1379–1386
14. Tombari, F., Salti, S., Di Stefano, L.: Performance evaluation of 3d keypoint detectors. International Journal of Computer Vision **102** (2013) 198–220
15. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Trans. Pattern Anal. Mach. Intell. **27** (2005) 1615–1630
16. Ke, Y., Sukthankar, R.: Pca-sift: A more distinctive representation for local image descriptors. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR'04, IEEE Computer Society (2004) 506–513
17. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60** (2004) 91–110