2012 Second Joint 3DIM/3DPVT Conference: 3D Imaging, Modeling, Processing, Visualization & Transmission

# On the Affinity between 3D Detectors and Descriptors

Samuele Salti, Alioscia Petrelli, Federico Tombari, Luigi Di Stefano
*Computer Vision Lab, DEIS, University of Bologna, Italy*
{ *samuele.salti, alioscia.petrelli, federico.tombari, luigi.distefano*} *@unibo.it*

***Abstract*—The literature on local invariant 3D features is growing, also fostered by the advent of cheap off-the-shelf 3D sensors. Although several recent proposals in the field include both a detector and a descriptor, some of the most successful and used descriptors do not define a companion detector. Moreover, as vouched by the related field of image features, detectors and descriptors defined within the same proposal do not necessarily yield the highest performance when used together. Hence, in this work we investigate on the effectiveness of the many possible combinations between state-of-the-art 3D detectors and descriptors, so as to identify optimal pairs as well as highlight well-matched detectors for those descriptors lacking a companion feature detection algorithm.**

***Keywords*-3D detector; 3D descriptor; performance evaluation**

## I. Introduction

Local invariant image features are key to many successful computer vision applications, such as automatic registration, object detection with clutter and occlusions, image categorization, object category detection. The computation of local invariant features relies on two stages, referred to as *detection* and *description*. Detection deals with the extraction of repeatable keypoints from images. Description projects the neighborhood of a keypoint into a proper feature space. Typical local features are covariant with rotation and scale, and robust to small affine or perspective distortions. Some of the most successful proposals define both stages, *e.g.* DoG and SIFT [1], FastHessian and SURF [2]. However, the problems of detection and description can be solved orthogonally, so that several proposals [3]–[5] as well as prominent evaluation work [6], [7] address only one of the two stages. As a result of such orthogonality, researchers have been deploying hybrid combinations, such as the recent ORB features [8] which rely on modified versions of independently introduced detection (FAST [4]) and description (BRIEF [5]) algorithms. Finally, as vouched by studies on image features [9], [10], the optimal combinations of detectors and descriptors are not necessarily those proposed together by the authors, so that it is relevant to evaluate their possible combinations to single out the best performing ones.

The related field of local features for 3D data has not reached yet a comparable level of maturity, although applications based on 3D local features are emerging: retrieval [11], object detection [12]–[14], shape registration [15], [16], shape categorization [17], [18]. Moreover, research efforts in this field are significantly fostered by the availability of affordable 3D sensors, above all the Microsoft Kinect. Recent papers on 3D features define both a detector and a descriptor [13], [15], [19]–[21]. Yet, some widely adopted proposals focus on the description stage only [16], [22], [23] and either describe all [22] or a random subset of points [16], [23].

Grounded on similar motivations as in [9], [10], in this paper we investigate on the effectiveness of the possible combinations between existing 3D feature detection and description algorithms. We carry out this study with the aim of identifying the best 3D detector/descriptor pairs in three diverse and important application scenarios, namely shape registration, object recognition and shape retrieval. Our contribution is three-fold: we single out effective 3D detector/descriptor pairs for each scenario by exhaustively exploring the Cartesian product between two sets comprising state-of-the-art proposals for each of the two stages; we highlight detectors suitable to be deployed together with widespread proposals addressing the description stage only; we assess the performance level currently achieved by state-of-the-art 3D features and highlight the related open issues.

To the best of our knowledge, this paper is the first attempt to single out the best combinations between 3D detectors and descriptors. In fact, the proposed evaluations of 3D detectors [24]–[26] and 3D descriptors [26], [27] focus on the two stages separately, likewise the well known evaluation work concerning local invariant image features [6], [7].

## II. 3D Detectors

This section briefly reviews the state-of-the-art methods for 3D keypoint detection considered in our investigation. 3D detectors can be divided into two categories, namely *fixed-scale* and *adaptive-scale* detectors. The key step common to both categories is the selection of keypoints as local extrema of a *saliency* measure.

*Fixed-scale detectors* find distinctive keypoints at a specific, constant scale which is provided as a parameter to the algorithm. *Local Surface Patches* (LSP), introduced in [12], defines the saliency of a vertex according to its Shape Index, which in turn is based on the maximum and minimum curvatures at the vertex. Extrema are considered keypoints if their Shape Index is significantly greater or smaller than the mean Shape Index within the given support.

*Intrinsic Shape Signatures* (ISS) were introduced in [20]. ISS saliency measure is based on the EigenValue Decompo-

978-0-7695-4873-9/12 $26.00 © 2012 IEEE
DOI 10.1109/3DIMPVT.2012.10

sition (EVD) of the scatter matrix of the points belonging to the support of a vertex. Only vertexes whose ratio between two successive eigenvalues is below a threshold are considered. Among these vertexes, the saliency is given by the magnitude of the smallest eigenvalue, so as to consider as keypoints only those vertexes exhibiting a large variation along every principal direction.

Likewise in [24], the 3D detector presented in [13] is referred to here as *KeyPoint Quality* (KPQ). Analogously to ISS, saliency is based on the scatter matrix. Pruning of non-distinctive vertexes, though, is achieved by thresholding the ratio between the maximum lengths along the first two principal axes, which is computed after alignment of the support to the canonical reference frame given by principal directions. As for saliency, it is determined by means of an empirical combination of curvatures within the support of the vertex. To limit the sensitivity of the estimation to noise and sampling density, curvatures are computed over a smoothed and re-sampled surface fitted to the aligned data by means of a surface fitting algorithm [28].

The common structure of *adaptive-scale detectors* starts building a scale-space defined on the surface, thus extending to the case of 3D data the well-known concept defined for images. Successively, a characteristic scale is associated to each vertex, which is selected as the maximum of the saliency along the scale dimension.

The proposal in [29], hereinafter recalled as Laplace-Beltrami Scale-Space (LBSS) as done in [24], builds the scale-space by computing an invariant defined as the exponential dumping of an operator, which can be interpreted as the displacement of a point along its normal by a quantity proportional to the mean curvature. Hence, for simple shapes such as perfect spheres or planes, the saliency measure employed by LBSS is proportional to the mean curvature.

MeshDoG [21] constructs the scale-space by applying different normalized Gaussian derivatives through the DoG operator, which is a well-known approximations of the normalized Laplacian [1]. The operator is not computed directly on the geometry of the mesh, but either on the mean curvature, the Gaussian curvature or the photometric appearance of a vertex.

In addition to the fixed-scale detector, in [13] an adaptive-scale method is also proposed, which will be referred to as KPQ-AS. The scale-space is built by increasing the size of the support over which the pruning term used by KPQ, *i.e.* the ratio between the maximum lengths along the first two principal axes, is computed. Then, automatic scale selection for each keypoint is carried out.

## III. 3D Descriptors

This section briefly reviews the 3D descriptors included in our investigation. We use the same taxonomy of 3D descriptors as in [16]. It divides 3D descriptors into two main categories, namely *Signatures* and *Histograms*.

As far as Signatures are concerned, in our investigation we include Point Signatures [30] and the descriptor proposed in the same paper as the KPQ detector [13]. In *Point Signatures* (PS), the signature is given by the signed height of the 3D curve obtained by intersecting a sphere centered at the keypoint with the surface. In [13] (KPQ), the signature is given by the third coordinate of each vertex of the support expressed in the local RF, after the same surface fitting and resamplig [28] step as in the detection phase has been performed.

As for Histogram-based methods, *Spin Images* [22], computes 2D histograms of points falling within a cylindrical volume by means of a plane that ”spins” around the normal. *3D Shape Context* [23] modifies the basic idea of Spin Images by accumulating 3D histograms of points within a sphere centered at the feature point. The canonical reference frame defined by 3D Shape Context is not unique, as one of its axis is chosen randomly. This results in the need to create multiple description for each keypoint during the matching stage, which affects both the effectiveness as well as the memory and computation costs of the matching process. To overcome these limitations, in [31] Unique Shape Context (USC) is proposed, which deploys the same description approach as in [23] but relies on a unique and repeatable canonical reference frame.

Two recent descriptors try to leverage on the benefits of both Signatures and Histograms. Both descriptors encode spatially localized histograms. Although not explicitly grounded on the adopted taxonomy, the MeshHoG descriptor [21] can be interpreted as a hybrid descriptor which uses histograms of curvatures. The SHOT descriptor [16], instead, was proposed in the paper that introduced the taxonomy and relies on histograms of normal orientations.

## IV. Evaluation Methodology

This section presents the datasets as well as the methodology and metrics used in each of the three application scenarios addressed by our evaluation.

### A. Datasets

For registration experiments we used *Armadillo*, *Dragon* and *Bunny* from the Stanford 3D Scanner Repository[1], *Amphora*, *Buste*, *Dancing Children*, *Fish*, *Glock* and *Neptune* from the Aim@Shape dataset[2] and sets of views of four objects acquired by means of a Kinect sensor (*Frog*, *Squirrel*, *Duck* and *Mario*). The ground truth for the first two sets is available together with the data, while we estimated the ground truth for the Kinect views by manually producing a coarse registration and then automatically refining it by means of *Scanalyze*[3]. Fig. 1 depicts sample views from the datasets.

[1] http://graphics.stanford.edu/data/3Dscanrep

[2] http://www.aimatshape.net/resources

[3] http://graphics.stanford.edu/software/scanalyze/

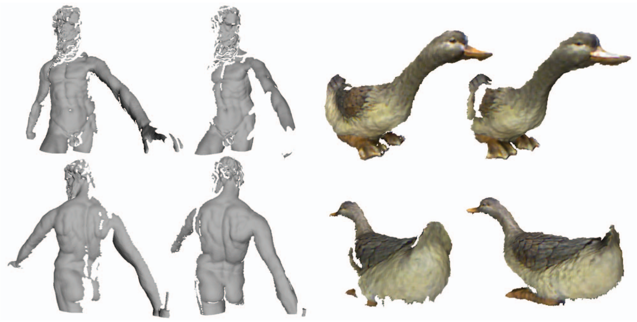

Figure 1. Registration datasets: sample views from *Neptune* and *Duck*.

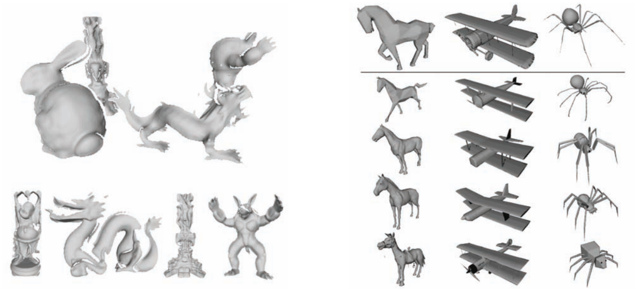

Figure 2. Left: one scene (top) and some model views (bottom) from *Virtual Stanford*. Right: three sample queries from *PSB* (top row) and some correct results from the ground truth in the corresponding columns.

We run object recognition experiments on a synthetic dataset, *Virtual Stanford*, that was created to reproduce typical data acquired with modern 3D sensors, such as the Microsoft Kinect, but with ground truth information known by construction. To this purpose, we implemented a Kinect simulator which first generates depth-maps of 640x480 pixels from a specific vantage points by ray casting, then adds Gaussian noise and quantizes z-coordinates, with both the noise variance and the quantization step increasing with distance [32]. Finally, we apply bilateral filtering to the depth maps to smooth out noise and quantization artifacts. The original models from the Stanford 3D Scanner Repository were placed randomly in groups of 3, 4 or 5 to create 50 different scenes. For each scene, 6 different views were acquired by the Kinect simulator. A set of 20 2.5D views for each model constitutes the model library, *i.e.* we focus on 2.5D versus 2.5D object recognition. Fig. 2 reports one exemplar scene and views from four models.

As for shape retrieval, we rely on the *Princeton Shape Benchmark dataset* [33]. It contains 1814 models, split into a train set and a test set. We used the `coarse1` classification level, wherein categories are directly related to shape, such as ”bed”, ”humans”, ”seat”, ”gun”, etc... (see Fig. 2).

### B. Application-oriented Approach

When evaluating detectors, descriptors as well as their combinations, two approaches can be envisaged: focusing on the performance of the feature algorithm regardless of the performance of the addressed application or, alternatively, on the performance of the overall application, indirectly assessing through it the actual performance of the detector and/or descriptor.

The first approach is perhaps more general, as it is unaffected by the choice of the additional stages that need to be plugged in together with detection and description in order to build up the complete pipeline. As pointed out in [34], the correct way to compare descriptors in descriptor matching experiments relies on Precision-Recall curves, as the total number of negatives is not well-defined in such experiments. Yet, Precision–Recall curves do not allow to summarize performance into a uniquely defined figure of merit[4], as it is the case *e.g.* of the Area Under the Curve (AUC) for ROC curves. However, such a kind of compact performance indicator is definitely required to manage and interpret the outcome of extensive experiments like those presented in this paper.

An important issue when comparing detectors/descriptors is the absolute number of keypoints and, hence, matches. In practice, a saturation effect occurs when using features in real applications: above a certain number of matches the performance of the application is insensitive to their actual number, whereas below another threshold the application simply fails, regardless of the relative number of good matches out of the total number of features. Precision–Recall or ROC curves do not capture this important aspect, as they normalize true positives and false positives with respect to absolute values of positives and negatives. To overcome this limitation, one might think to plot just true positives vs. false positives, but it is again hard to define a meaningful indicator to summarize performance, due to such curves spanning different regions depending on the number of keypoints extracted by the detector.

On the other hand, to focus on the performance of the overall application offers a clear and direct snapshot of the performance of a detector/descriptor pair in a specific context. Moreover, such an approach allows well defined and meaningful performance indexes to be derived straightforwardly. However, as already pointed out, the application-oriented approach requires the definition of a complete algorithm chain, which must be suited to all the considered detector/descriptor pairs (*e.g.* it must be neutral with respect to descriptors defining or not a local reference frame). To deal with this issue, as well as to emphasize the impact of detector/descriptor pairs on performance, we opted for three very simple, baseline application pipelines addressing registration, object recognition and shape retrieval. These are described below together with the associated performance

[4] For example, the definition of the average precision used in the Pascal VOC Challenge has been changed during the editions and in all versions it requires to modify the curve in order to make precision monotonically decreasing

indexes.

*1) Registration:* To register two 3D views of a model, indicated here as $v_1$ and $v_2$, we first establish correspondences by matching descriptors with their nearest neighbor. Given correspondences, we apply RANSAC together with a classical absolute orientation algorithm and then refine the alignment by Generalized ICP [35]. The inlier tolerance for RANSAC and the maximum matching threshold for GICP were tuned on a randomly selected subset of views. The resulting values were $8 * mr$ for RANSAC and $10 * mr$ for GICP, where $mr$ stands for *mesh resolution*, *i.e.* the average edge length of the meshes. Given the estimated rotation and translation aligning $v_1$ to $v_2$, denoted as $H_{12}$ in homogeneous coordinates, the Root Mean Squared Error (RMSE) for the pair is computed as follows:

$$RMSE_{v_1,v_2} = \sqrt{\frac{1}{N_{v_1}} \sum_{i=1}^{N_{v_1}} (\mathbf{H}_{12}\tilde{\mathbf{p}}_i - \mathbf{H}_{12}^{GT}\tilde{\mathbf{p}}_i)^2} \quad (1)$$

where $N_{v_1}$ is the number of vertexes in $v_1$, $\tilde{p}_i$ are the 3D homogeneous coordinates of the $i$-th vertex of $v_1$ and $\mathbf{H}_{12}^{GT}$ is the ground truth transformation between $v_1$ and $v_2$.

The registration is considered successful if the RMSE between the views is below a threshold ($\epsilon = 5 * mr$). Given all the pairs of a model, its registration rate is defined as

$$r_{reg} = \frac{\#\text{registered pairs}}{\#\text{registrable pairs}} , \quad (2)$$

where a pair is considered registrable if, when aligned by using ground truth transformation, the area overlap between the views is larger than 10% of their area. Finally, the registration rate over the whole registration dataset is given by the mean registration rate over all the models.

*2) Object recognition:* To carry out object recognition in scenes with clutter and occlusions, we complement feature extraction and description with a baseline *correspondence grouping* (CG) stage [12], aimed at grouping correspondences into geometrically consistent subsets, each one holding consensus for the presence of a specific model instance in the current scene under the assumption of rigid model-to-scene transformations.

Specifically, given a scene $s$ and one view $v$ of the model $m$, CG subdivides correspondences into subsets where, for every two pairs of matching keypoints $\{k_l^s, k_n^v\}$, $\{k_p^s, k_q^v\}$, the following spatial relationship is satisfied:

$$\left| ||k_n^v - k_q^v||_2 - ||k_l^s - k_p^s||_2 \right| < \varepsilon \quad (3)$$

with $\varepsilon$ being a parameter of this method, intuitively representing the consensus set bandwidth. Starting from strongest correspondences, the subdivision into subsets proceeds iteratively, until no more correspondences can be added or subsets can be merged. If a subset cardinality is above a consensus size threshold the model is considered present in the scene. Given the correspondences, its pose can also be estimated.

Hence, given a scene $s$ and the set of views $\{m_{i,j}\}_{j=1}^N$ of the model $m_i$, we first establish correspondences associated with each view by matching descriptors according to the ratio of distances criterion [1], so as to limit the influence of clutter. Given these correspondences, we select as best view that yielding more matches,

$$m_{i,best} = \arg\max_{m_{i,j}} \#\text{matches}(m_{i,j}, s) \ . \quad (4)$$

We then run CG between the best view $m_{i,best}$ and the scene $s$ in order to decide upon model presence/absence. Given the established presence/absence for each model and the ground truth, we can define true positives and false positives as well as positives and negatives. By varying the consensus size threshold for CG, we can plot the corresponding ROC curve and estimate the AUC, which is used as performance index in these experiments. The ratio threshold for descriptor matching and the consensus set bandwidth for CG were tuned on a random subset of the dataset. The resulting values were $0.85$ for the ratio threshold and $10 * mr$ for the consensus set bandwidth.

*3) Shape retrieval:* For object retrieval, we first normalize each model and query to align it with its principal directions and fit it in the unit cube centered in the origin, as the PSB dataset presents a large size variability. We also normalize the point density of the dataset by resampling each model to the same number of vertexes (4000), as done for example in [36]. Given a query $q$ and a set of models $\{m_i\}_{i=1}^N$, we establish a set of correspondences by matching each descriptor of $q$ with its 1-NN in each model $m_i$. Let $d_k(q, m_i)$ be the distance of the $k$-th descriptor of $q$ from its nearest neighbor in $m_i$. We rank models in the retrieval list according to the mean distance of their descriptors from $q$, *i.e.*

$$m_i \preceq m_j \Leftrightarrow \frac{1}{M} \sum_{k=1}^{M} d_k(q, m_i) \leq \frac{1}{M} \sum_{k=1}^{M} d_k(q, m_j) \ , \quad (5)$$

where M is the number of descriptors extracted from the query. Given the ranked list of models, we use the second tier [33] as performance index of retrieval experiments. This index is computed as the ratio between the number of correct retrieval result in the first $2C$ positions of the list and $C$, where $C$ is the number of models belonging to the query category in the testing set.

### C. Parameters

All parameters of detectors and descriptors have been fixed for the experiments on all datasets. Default parameters proposed by the authors in the original publications have been used. For MeshDoG we use the mean curvature as quality measure, for we found that it yields better results than the Gaussian curvature.

Following the methodology proposed in [24], adaptive-scale detectors and fixed-scale detectors have been run on approximately the same set of scales. Detectors were tuned to inspect the set $\Sigma = \{8mr, 14mr, 20mr, 26mr, 32mr, 38mr\}$: this allows detectors to look for discriminative and repeatable structures ranging from point-wise scales to local and object sub-part scales. As scale-invariant detectors define different spacing between adjacent scales, they could only be tuned to approximately inspect the same scales. Moreover, the first and last scale are used only to assess the presence of a local extremum in the immediately subsequent or antecedent scale by KPQ-SI and LBSS. Therefore for KPQ-SI, which uses uniform spacing for the scale space, *i.e.* $\sigma_k = \sigma_{k-1} + \Delta\sigma$, we set $\sigma_0 = 2mr$, $\Delta\sigma = 6mr$ and the number of scales $N_s = 8$; for LBSS, which uses exponential spacing, *i.e.* $\sigma_k = \sigma_{k-1} * \Delta\sigma$, we set $\sigma_0 = 3.75mr$, $\Delta\sigma = 1.6$ and $N_s = 7$; for MeshDoG, which uses octaves and scales like SIFT, *i.e.*

$$\sigma_k = \sqrt{(k \mod O_s + 1) 2^{k/O_s} + O_s \sum_{i=0}^{k/O_s} 2^i} \; \sigma_0 \qquad (6)$$

we set $\sigma_0 = 6mr$, the number of octaves per scales $O_s = 6$ and $N_s = 3$.

To report the mean result for a combination of a descriptor and a fixed-scale detector we consider in each experiment the radius that yields the best performance.

## V. Experimental results

We provide results through tables showing for each detector/descriptor pair the mean performance index on the datasets.

### A. Registration

As for registration, overall the performance are quite low, *i.e.* the best method can align on average 30% of the registrable view pairs. When interpreting this results, however, it is important to remind that we defined as registrable those pairs whose overlap is at least 10% of their area. This threshold was chosen to highlight whether some combinations were particularly suited to register difficult view pairs ( *i.e.* exhibiting a small overlap), though, in general, global registration of all views can be accomplished even without registering such difficult cases, provided a sufficient number of views is available.

The best performance is provided by the pair ISS/PS (Table I). ISS turns out the best performer among detectors, as nearly all descriptors yield their -substantially- highest registration rate when used in conjunction with such detector. The second best detector is KPQ-AS, despite scatter matrix-based detectors (*e.g.* KPQ, KPQ-AS and ISS alike) being not invariant nor robust to the local modifications of the mesh due to self occlusions and vantage point variations. Nevertheless, such detectors outperform those relying on point-wise curvature estimations, *e.g.* MeshDoG and LSP. We ascribe this counterintuitive result to the inherent inferior robustness to noise of the latter. LBSS consistently provides the lowest registration rates: this can be ascribed to the small amount of keypoints highlighted by its saliency [24], which is not sufficient to correctly align most of the views.

Except when ISS is used, given the same detector there are not such clear differences between descriptors. As a result, all the descriptors are well matched in this experiment with ISS. Only MeshHoG performs better when used in conjunction with its own detector, MeshDoG.

### B. Object Recognition

Results on the Virtual Stanford dataset are reported in table II.

The best combination is ISS/USC, closely followed by ISS/SHOT. ISS turns out again the best performer among detectors: all descriptors yield the highest AUC when used in conjunction with such detector. The detector is robust to the distortions of the detailed shapes of the Stanford models created by the Kinect simulator, its saliency still capturing more repeatable and distinctive structures than other detectors. We were unable to test KPQ-AS as it requires more than an hour to process one of the 300 scenes in the dataset.

The best descriptors are SHOT, USC, PS, and KPQ. It is interesting to note that there is a wide gap in performance between USC and 3DSC when using ISS and KPQ, *i.e.* when the detector is well-matched to USC because it grounds the saliency on the scatter matrix, which is the same entity used by USC to compute its unique local reference frame. Likewise SHOT, which deploys the same local RF as USC, shows more affinity with ISS than with other detectors. Finally, the deployment of descriptors with the detector originally introduced in the same proposal does not result in best performance, *e.g.* ISS/KPQ is better than KPQ/KPQ and ISS/MeshHoG is better than MeshDoG/MeshHoG.

### C. Retrieval

Results on the Princeton Shape Benchmark dataset are reported in table III. The best combinations are ISS/SHOT, KPQ-AS/SHOT, LSP/SHOT and MeshDoG/SHOT. The retrieval experiment requires robustness to intra-class variations rather than noise and vantage point variations, the latter pair of nuisances being absent in the data. In fact, point-wise curvatures detectors (LSP and MeshDoG) turn out as effective as or even better than those based on the scatter matrix. With these working conditions, point-wise curvatures allow to highlight small, distinctive details which are likely to co-occur within the class. On the other hand, KPQ performs poorly when compared with the other experiments: the use of a larger support than point-wise curvatures combined with surface smoothing and resampling

Table I

Mean registration rates on Kinect, Stanford, and Aim@Shape datasets.

| Desc.<br>Det. | 3DSC | KPQ | MeshHoG | PS | SHOT | SI | USC |
|---|---|---|---|---|---|---|---|
| ISS | 0.3 | 0.3 | 0.19 | 0.32 | 0.28 | 0.29 | 0.23 |
| KPQ | 0.12 | 0.18 | 0.13 | 0.25 | 0.15 | 0.13 | 0.1 |
| LSP | 0.12 | 0.17 | 0.09 | 0.17 | 0.15 | 0.11 | 0.08 |
| KPQ-AS | 0.24 | 0.21 | 0.2 | 0.18 | 0.19 | 0.26 | 0.19 |
| LBSS | 0 | 0.01 | 0.04 | 0.03 | 0.02 | 0.01 | 0.01 |
| MeshDoG | 0.15 | 0.17 | 0.23 | 0.18 | 0.16 | 0.14 | 0.1 |

Table II

Mean AUCs on Virtual Stanford dataset.

| Desc.<br>Det. | 3DSC | KPQ | MeshHoG | PS | SHOT | SI | USC |
|---|---|---|---|---|---|---|---|
| ISS | 0.54 | 0.59 | 0.55 | 0.64 | 0.74 | 0.61 | 0.76 |
| KPQ | 0.51 | 0.55 | 0.55 | 0.55 | 0.54 | 0.56 | 0.53 |
| LSP | 0.51 | 0.58 | 0.54 | 0.58 | 0.6 | 0.54 | 0.56 |
| KPQ-AS | - | - | - | - | - | - | - |
| LBSS | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 |
| MeshDoG | 0.48 | 0.47 | 0.52 | 0.47 | 0.47 | 0.46 | 0.49 |

in absence of noise hinders retrieving the same structures in presence of local modifications due to intra-class variations.

The best descriptor is by far SHOT, which obtains the best second tier with every detector. An interesting comparison can be performed between SHOT and USC, as they are describing the same keypoints and deploy the same local RF. The notable difference in performance is to be ascribed to two factors: USC, being a pure histogram of points, has to use a fine tessellation of the local neighborhood to achieve distinctiveness, which turns out inherently less robust to intra-class variations than the coarse subdivision used by SHOT; SHOT accumulates first order differential entities in its histograms, which tolerate higher intra-class variations compared to the raw data used by USC. KPQ is a signature of smoothed and resampled points: a signature of raw data is unlikely to be effective in presence of intra-class variations; the surface smoothing and resampling is indeed detrimental as the data do not present noise. It was not possible to use 3DSC in this experiment, as it is the largest descriptor (1980 entries) and the set of descriptors extracted from the training set and used at test time to rank the database models does not fit into the memory limit of 32-bits systems.

## VI. Running time

Table IV shows the mean running time required by the detection, description and matching stages in the registration of a view pair. The reported times are aggregated by averaging the registration times across all the view pairs of a model and then over all the models of the registration dataset.

The fastest pair is LSP/SI, closely followed by LSP/MeshHoG, LSP/PS and ISS/SI. The detector plays the most important role in determining the overall efficiency of a pair. When deploying KPQ-AS and LBSS it takes on average about 20 minutes to register a pair of views of a model, regardless of the descriptor. Even with KPQ the overall running time is dominated by the detection time, but the registration time for a pair drops to about 1 minute and a half. Only with faster detectors we can appreciate the difference between the evaluated pairs. The majority of combinations obeys the rule that using faster detector or descriptor results in shorter running time, without exhibiting particular affinity between considered methods.

## VII. Final Considerations

Overall, the best pairs identified by the proposed evaluation turn out ISS/PS and ISS/SHOT. The former is the best for registration and the third-best for object recognition; the latter is close to the best pair for registration, is the second-best for object recognition and the best for retrieval. Both pairs also exhibit short running time. Other effective combinations are ISS/USC, which is the best for object recognition, and ISS/3DSC and ISS/KPQ, which are the second-best for registrations. Some methods turned out too demanding though: 3DSC memory requirements do not allow its deployment with large databases, such as those

Table III
MEAN SECOND TIERS ON PRINCETON SHAPE BENCHMARK DATASET.

| Desc.<br>Det. | 3DSC | KPQ | MeshHoG | PS | SHOT | SI | USC |
|---|---|---|---|---|---|---|---|
| ISS | - | 0.11 | 0.17 | 0.22 | 0.31 | 0.22 | 0.22 |
| KPQ | - | 0.09 | 0.17 | 0.18 | 0.24 | 0.2 | 0.15 |
| LSP | - | 0.12 | 0.19 | 0.19 | 0.29 | 0.24 | 0.18 |
| KPQ-AS | - | 0.06 | 0.2 | 0.21 | 0.3 | 0.25 | 0.25 |
| LBSS | - | 0.08 | 0.07 | 0.1 | 0.13 | 0.1 | 0.12 |
| MeshDoG | - | 0.06 | 0.2 | 0.17 | 0.27 | 0.22 | 0.15 |

Table IV
MEAN RUNNING TIME IN SECONDS ON KINECT, STANFORD, AND AIM@SHAPE DATASETS.

| Desc.<br>Det. | 3DSC | KPQ | MeshHoG | PS | SHOT | SI | USC |
|---|---|---|---|---|---|---|---|
| ISS | 4.25 | 7 | 3.83 | 2.31 | 3.47 | 1.79 | 5.08 |
| KPQ | 85.3 | 91.14 | 84.34 | 84.22 | 84.22 | 83.67 | 89.29 |
| LSP | 3.56 | 7.61 | 1.7 | 1.72 | 3.14 | 1.19 | 3.38 |
| KPQ-AS | 1186.91 | 1233.37 | 1181.69 | 1176.85 | 1176.52 | 1176.24 | 1185.91 |
| LBSS | 1123.49 | 1123.64 | 1126.53 | 1123.44 | 1123.48 | 1123.48 | 1123.49 |
| MeshDoG | 6.87 | 32.05 | 7.68 | 3.89 | 2.94 | 2.82 | 5.67 |

typically used in shape retrieval scenarios; KPQ-AS computation time does not scale well with the number of vertices in the mesh, and therefore it cannot be used on detailed scenes such as those of our object recognition experiment.

As the above summary clearly indicates, the detector referred to as ISS is the most effective. Therefore, it turns out a reasonable choice to try in place of random sampling for those descriptors lacking a companion detection stage. Alternatives are KPQ-AS for registration, which is well matched with both SI and 3DSC, and KPQ-AS and MeshDoG for retrieval, which provide better performance than ISS with both SI and MeshHoG. Overall, current state-of-the-art 3D features seem effective when dealing with non-Kinect data, *i.e.* the CAD models of the PSB dataset and the laser scanner views used for registration. On the other hand, Kinect data, as those used in our object recognition scenario, set forth significant challenges for the evaluated feature algorithms. This is, indeed, one major open issue to be addressed by future research on local 3D features.

As for future work, we plan to extend our evaluation, by including recent relevant detectors such as, in particular, the HKS [19] detector and the proposals in [37] and in [38]. As for descriptors, we plan to include the local extension of the well-known Spherical Harmonics global descriptor [39] and HKS-SI [40].

## REFERENCES

[1] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *IJCV*, vol. 60, pp. 91–110, 2004.

[2] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, “Surf: Speeded up robust features,” *CVIU*, vol. 110, no. 3, pp. 346–359, 2008.

[3] J. Matas, O. Chum, U. Martin, and T. Pajdla, “Robust wide baseline stereo from maximally stable extremal regions,” in *BMVC*, vol. 1, London, 2002, pp. 384–393.

[4] E. Rosten, R. Porter, and T. Drummond, “Faster and better: A machine learning approach to corner detection,” *PAMI*, vol. 32, no. 1, pp. 105–119, Jan. 2010.

[5] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “BRIEF: Binary Robust Independent Elementary Features,” in *ECCV*, Sept. 2010.

[6] K. Mikolajczyk and C. Schmid, “A performance evaluation of local descriptors,” *PAMI*, vol. 27, no. 10, pp. 1615–1630, 2005.

[7] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, “A Comparison of Affine Region Detectors,” *IJCV*, vol. 65, no. 1-2, pp. 43–72, 2005.

[8] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, “ORB: An efficient alternative to SIFT or SURF,” in *ICCV*, 2011, pp. 2564–2571.

[9] P. Moreels and P. Perona, “Evaluation of features detectors and descriptors based on 3D objects,” in *ICCV*, vol. 1, oct. 2005, pp. 800–807.

[10] A. Dahl, H. Aan ands, and K. Pedersen, “Finding the best feature detector-descriptor combination,” in *3DIMPVT*, may 2011, pp. 318 –325.

[11] A. M. Bronstein, M. M. Bronstein, L. J. Guibas, and M. Ovsjanikov, “Shape Google: geometric words and expressions for invariant shape retrieval,” *ACM Trans. Graph.*, vol. 30, pp. 1:1–1:20, feb 2011. [Online]. Available: http://doi.acm.org/10.1145/1899404.1899405

[12] H. Chen and B. Bhanu, “3D free-form object recognition in range images using local surface patches,” *Pattern Recognition Letters*, vol. 28, no. 10, pp. 1252–1262, 2007.

[13] A. S. Mian, M. Bennamoun, and R. A. Owens, “On the repeatability and quality of keypoints for local feature-based 3D object retrieval from cluttered scenes,” *IJCV*, vol. 89, no. 2-3, pp. 348–361, 2010.

[14] F. Tombari and L. Di Stefano, “Hough voting for 3D object recognition under occlusion and clutter,” *IPSJ Trans, on CVA*, vol. 4, pp. 20–29, 2012.

[15] J. Novatnack and K. Nishino, “Scale-dependent/invariant local 3D shape descriptors for fully automatic registration of multiple sets of range images,” in *ECCV*, 2008, pp. 440–453.

[16] F. Tombari, S. Salti, and L. Di Stefano, “Unique signatures of histograms for local surface description,” in *ECCV*, 2010, pp. 356–369.

[17] J. Knopp, M. Prasad, G. Willems, R. Timofte, and L. Van Gool, “Hough transform and 3D SURF for robust three dimensional classification,” in *ECCV*, 2010.

[18] S. Salti, F. Tombari, and L. Di Stefano, “On the use of implicit shape models for recognition of object categories in 3D data,” in *ACCV*, 2010, pp. 653–666.

[19] J. Sun, M. Ovsjanikov, and L. Guibas, “A concise and provably informative multi-scale signature based on heat diffusion,” in *Proc. Symp. Geom. Proc.*, 2009, pp. 1383–1392.

[20] Y. Zhong, “Intrinsic shape signatures: A shape descriptor for 3D object recognition,” in *ICCV-WS:3DRR*, 2009.

[21] A. Zaharescu, E. Boyer, and K. Varanasi, “Surface feature detection and description with applications to mesh matching,” *CVPR*, 2009.

[22] A. Johnson and M. Hebert, “Using spin images for efficient object recognition in cluttered 3D scenes,” *PAMI*, vol. 21, no. 5, pp. 433–449, 1999.

[23] A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik, “Recognizing objects in range data using regional point descriptors,” in *ECCV*, vol. 3, 2004, pp. 224–237.

[24] F. Tombari, S. Salti, and L. Di Stefano, “Performance evaluation of 3D keypoint detectors,” *IJCV*, pp. 1–23, 2012.

[25] T.-H. Yu, O. Woodford, and R. Cipolla, “An evaluation of volumetric interest points,” in *3DIMPVT*, may 2011, pp. 282 –289.

[26] A. M. Bronstein, M. M. Bronstein, and et. al., “Shrec 2010: robust feature detection and description benchmark,” in *3DOR*, 2010.

[27] P. Heider, A. Pierre-Pierre, R. Li, and C. Grimm, “Local shape descriptors, a survey and evaluation,” in *3DOR*, 2011, pp. 49–56.

[28] J. D’Errico, “Surface fitting using gridfit,” MATLAB Central File Exchange, July 2010.

[29] R. Unnikrishnan and M. Hebert, “Multi-scale interest regions from unorganized point clouds,” in *CVPR-WS: S3D*, 2008.

[30] C. S. Chua and R. Jarvis, “Point signatures: A new representation for 3D object recognition,” *IJCV*, vol. 25, no. 1, pp. 63–85, 1997.

[31] F. Tombari, S. Salti, and L. Di Stefano, “Unique shape context for 3D data description,” in *3DOR*. New York, NY, USA: ACM, 2010, pp. 57–62.

[32] J. Smisek, M. Jancosek, and T. Pajdla, “3D with kinect,” in *ICCV-WS*, nov. 2011, pp. 1154 –1160.

[33] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser, “The princeton shape benchmark,” in *Shape Modeling International*, Jun. 2004.

[34] Y. Ke and R. Sukthankar, “PCA-SIFT: a more distinctive representation for local image descriptors,” in *CVPR*. IEEE Computer Society, 2004, pp. 506–513.

[35] A. Segal, D. Haehnel, and S. Thrun, “Generalized-ICP,” in *Proceedings of Robotics: Science and Systems*, Seattle, USA, June 2009.

[36] Y. Liu, H. Zha, and H. Qin, “Shape Topics: A compact representation and new algorithms for 3D partial shape retrieval,” in *CVPR*, 2006.

[37] H. Fadaifard and G. Wolberg, “Multiscale 3D feature extraction and matching,” in *3DIMPVT*, may 2011, pp. 228 –235.

[38] I. Sipiran and B. Bustos, “Harris 3D: a robust extension of the Harris operator for interest,” *Int. J. Vis. Comput.*, vol. 27, pp. 963–976, 2011.

[39] P. Shilane and T. Funkhouser, “Selecting distinctive 3D shape descriptors for similarity retrieval,” in *Proc. Shape Modeling International*, 2006.

[40] M. Bronstein and I. Kokkinos, “Scale-invariant heat kernel signatures for non-rigid shape recognition,” in *Proc. Int. Conf. Computer Vision and Pattern Recognition (CVPR)*, 2010.