SCurV: A 3D descriptor for object classification

Antonio J Rodríguez-Sánchez, Sandor Szedmak and Justus Piater

Abstract— **3D Object recognition is one of the big problems in Computer Vision which has a direct impact in Robotics. There have been great advances in the last decade thanks to point cloud descriptors. These descriptors do very well at recognizing object instances in a wide variety of situations. Of great interest is also to know how descriptors perform in object classification tasks. With that idea in mind, we introduce a descriptor designed for the representation of object classes. Our descriptor, named SCurV, exploits 3D shape information and is inspired by recent findings from neurophysiology. We compute and incorporate surface curvatures and distributions of local surface point projections that represent flatness, concavity and convexity in a 3D object-centered and view-dependent descriptor. These different sources of information are combined in a novel and simple, yet effective, way of combining different features to improve classification results which can be extended to the combination of any type of descriptor. Our experimental setup compares SCurV with other recent descriptors on a large classification task. Using a large and heterogeneous database of 3D objects, we perform our experiments both on a classical, flat classification task and within a novel framework for hierarchical classification. On both tasks, the SCurV descriptor outperformed all other 3D descriptors tested.**

I. INTRODUCTION

Over the last decade, we have seen the appearance of many types of descriptors. Some are global, others exploit local information, some are for 2D applications, others can be applied in 3D. The bibliography is as varied as their applications. In the 3D case, even though there is a large number of studies that compare descriptors in different situations, it is still unclear how these descriptors compare with one another in a large classification task since most of them report results on small instance recognition data sets. Thus, it would be of interest to test the different methods on a classification task using the same common large database that includes both a large number of objects and a wide variety of different classes. This is of interest in Computer Vision and Robotics in order to evaluate the best strategies that are to be considered for the future. Nonetheless, in order to build robots and systems, we want them to behave in ways similar to humans, thus not only recognizing a small set of object instances, but also being able to classify large sets of object classes as humans do.

The human visual system is indeed the inspiration of our own descriptor, SCurV. Biological inspired computations can be used for Computer Vision tasks as shown in previous works [1], [2], [3]. We developed SCurV to extract values similar to the ones computed by certain neurons in the brain. In 3D, neurophysiological studies have found neurons that are selective to surface curvature, orientation and the position of surface fragments [4]. Configuration of 3D surface fragments is encoded by those neurons in an object-centered frame, but the spatial frame is only partially defined by the object since neural representations are different depending on viewpoint. The descriptor we propose (SCurV) also combines those two relatively contradictory facts about neural encoding through a novel way of combining feature vectors by using the tensor product. The features we compute contain a global representation based on curvatures and a local representation based on viewpoint distributions. We then compare this descriptor to ten recent descriptors on the same classification task to evaluate if our descriptor can outperform current 3D descriptors in the aforementioned task.

Considering the large number of 3D descriptors introduced over recent years, it is of also of great interest for the Robotics and Computer Vision community to provide a framework where we tan take complementary descriptors and improve recognition performance by simply combining them, thus exploiting the strengths of each one. We also show that the way we combine our global-centered object representation and the local-pointwise object distribution may be extended to any feature combination whose information is complementary in order to improve classification results. Our way of combining descriptors is a kernel version of the tensor product that avoids some disadvantages of the classical concatenation of features.

Most current classification tasks involve multiclass or binary classification. For such tasks, a dog, a cat and a boat are different to the same degree; they belong to different classes in a flat sense, where all classes are at the same level. For humans, they are different, but a dog is more different from a boat than a dog is from a cat (both are quadruped animals). Humans are able to classify objects in a hierarchical fashion (e.g. bees and ants are insects, dogs and horses are quadruped animals, etc.), which seems to be the result of geometric similarity [5]. Finally, we also provide a framework where we evaluate SCurV and other state-of-the-art descriptors at a hierarchical classification of object classes.

II. RELATED WORK

Descriptors operate in what is currently known as point clouds, which typically correspond to points from range images. Most of the applications of these descriptors are in Robotics, involving tasks such as object search or manipulation. The literature on 3D descriptors is very extensive, so we

Intelligent and Interactive Systems, Institute of Computer Science, University of Innsbruck, Technikerstrasse 21a, Innsbruck, Austria Antonio.Rodriguez-Sanchez@uibk.ac.at Sandor.Szedmak,Justus.Piater@uibk.ac.at

will try to summarize here the most recent methods. Some of these descriptors are extensions of 2D shape counterparts. These include 3D versions of Belongie's shape context [6] such as 3D shape context [7], unique shape context [8] and intrinsic shape context [9] or an extension of Johnson and Hebert's spin images [10] to intrinsic spin images [11]. Another set of descriptors obtain different values (such as angles or normals) that are grouped into histograms. These include point feature histograms (PFH) [12], viewpoint feature histograms (VFH) [13] and other variants [14], [15], [16], as well as the ensemble of shape functions [17]. Additional approaches include the use of eigenvalue decomposition for a repeatable reference frame in the signature of histograms of orientations (SHOT) [18], voting schemes (point pair features [19]), spherical harmonics [20] and the spherical blurred shape model [21]. Some of the aforementioned descriptors (those available from the PCL) are further summarized next in section II-B.

A. Comparing Descriptors

Previous work has been done on comparing 3D descriptors. Recent work [22], [23] has tested their performance in outdoor scene scenarios. Alexandre [24] performed a descriptor comparison based on keypoint extraction. Some other works [25], [21] have tested descriptors using the Washington dataset [26], the former [25] present a thorough study regarding robustness to different degrees of noise, while the latter [21] adds sign language and facial expression performance. It is also worth to mention the work of Tombari et al. [27] on 3D detectors perfomance under noise, clutter, occlusion and viewpoint. The work of Akgul and collaborators [28] showed that density-based shape descriptors outperform histogram-based descriptors. That study, as well as other recent work (e.g. [29]) have dealt with content-based image retrieval, thus not evaluating performance in a classification task, but using similarity measures between object instances.

We are interested here in classification performance. The aforementioned work in the image retrieval literature provided us with the dataset we were looking for, containing objects as different as a ship and a spider, objects found in nature (animals, trees, etc.), man-made objects (hammers, lamps, guitars, ...), and with a wide variety of shapes and forms, including textured objects and objects with inscriptions. That dataset is the Princeton Shape Benchmark (PSB) [30]. The object models in this dataset were synthetically generated (alas, there is no comparable database of real 3D objects). Being extensively used in image retrieval, it is also highly suited to 3D object classification. A particularly nice feature of the PSB is the organization of its classes into a hierarchy, thus allowing us to perform not only the classical flat classification, but also report results on a hierarchical classification.

B. 3D Descriptors

For our comparison we selected ten state-of-the-art descriptors based on feature histograms, histograms of orientations, spin images, shape context and shape distributions, all available from the Point Cloud Library (PCL)[1] [31].

a) ***The Ensemble of Shape Functions (ESF) [17]:*** ESF follows the D2 shape distribution approach of [32], which is the distribution of the distances between pairs of random points on the shape. Wohlkinger and Vincze propose three types of distribution (approximated as histograms): A histogram of distances between points whose junction lines lie on the object's surface (*on* distances), another histogram for lines lying outside the surface going through points that are on the surface (*off* distances), and a third histogram of distances which is a *mix* of both.

b) ***Point Feature Histograms (PFH, FPFH, VFH, CVFH, OUR-CVFH) [12], [14], [13], [15], [16]:*** Rusu presented the Point Feature Histograms (PFH) in 2008 and later updates in 2009 (Fast Point Feature Histograms, FPFH), 2010 (Viewpoint Feature Histograms, VFH), 2011 (Clustered VFH) and 2012 (OUR-CVFH). PFH is based on computing the difference between the normals of two points in a neighborhood and the line joining both points, which define a Darboux frame (*u, v, w*). FPFH [14] is a later version of the descriptor that, instead of using all possible point pairs inside the neighborhood, sets one point against the points in the neighborhood. VFH [13] is another update of FPFH where a viewpoint component is added that contains information regarding the histogram of the angles between a viewpoint direction and each normal. CVFH [15] and OUR-CVFH [16] are extensions of VFH, where the former adds an angular and an L1 distributions, while the latter includes a Reference Frame component in addition to the angular distribution.

c) ***Signature of Histograms of Orientations (SHOT) [8]:*** SHOT first defines an invariant local reference and then computes the difference between two points normals in point clouds. SHOT consists of two steps: the computation of a histogram and a signature. The histogram is the number of points that fall into bins with respect to a function (the cosine) of the angle between the difference of the normals. The signature corresponds to a spherical grid that combines radial, azimuth and elevation axes. This grid is divided into 32 bins filled with the histogram counts, each of which is multiplied by a weight proportional related to the distance to the central bin.

d) ***Shape Context descriptors:*** The shape context [6] is a global descriptor that captures the distribution of points with respect to specific points in the shape. We will report results on 3D Shape Context [7] and Unique Shape context [8]. The 3D Shape Context (3DSC) corresponds to the histogram of the relative coordinates of all the shape points with respect to the current point. Matching is performed computing three values: 1) the *shape term* 2) the *appearance term* and 3) the *position term*. These three terms are combined by a weighted sum. The Unique Shape Context (USC) [8] avoids the need for a descriptor computation over different rotations as used in 3DSC. USC computes a unique unambiguous reference frame that is repeatable over both the normal axis and the

[1] http://docs.pointclouds.org/trunk/group__features.html

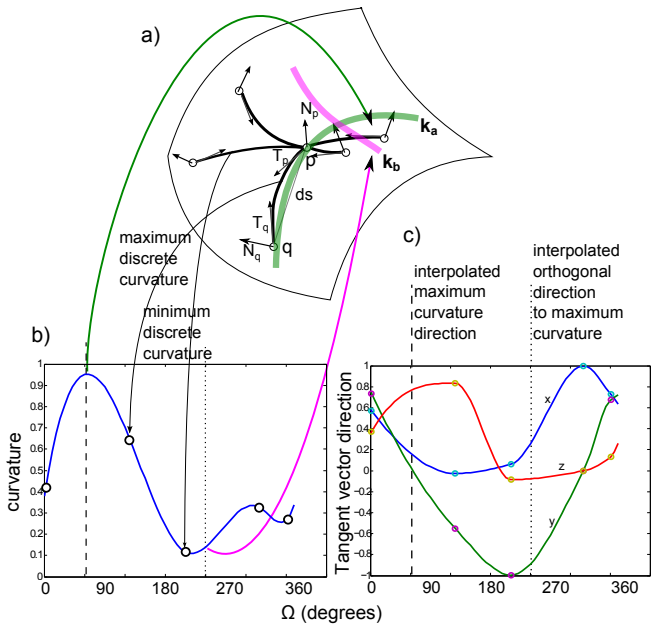

Fig. 1. How principal curvatures in the surface are approximated. a) First, curvatures (bold black curves) are computed over the surfaces using pairs of points (blank circles), one of them being the center point p. N and T correspond to the normal and tangent vectors, arrows give their direction. b) Curvatures are interpolated using a spline, providing the maximum curvature k_a. c) Tangent directions (x, y and z vector values) corresponding to the two orthogonal surface curvatures are interpolated in the same way. The interpolated tangent directions provide the curvature value corresponding to k_b in (b) through its polar angle Ω value (see text).

tangent plane.

*e) **Spin Images (SI)** [10]:* One can think of spin images as finding oneself in one point of a mesh and recording the radial and elevation coordinates of the points that fall in the neighborhood. These distance point values are grouped together into a histogram. A shape is then a set of those histograms.

III. SURFACE CURVATURE AND VIEWPOINT LOCAL DISTRIBUTION DESCRIPTOR (SCURV)

We describe here in detail our SCurV descriptor. This descriptor is the result of computing the following quantities:

1) A global object-centered representation based on surface curvature.
2) A local viewpoint-centered representation providing degrees of convexity, concavity and flatness.
3) The final descriptor, which is the result of computing the tensor product between the global object surface components and the local viewpoint distributions.

Both parts of the descriptor (as well as those summarized in section II-B) work with point clouds. Since the PSB uses mesh files to represent 3D objects, 3D points are uniformly sampled from the triangulated surface as later explained in detail in section IV-B.

A. Global object-centered representation

For every point, a surface neighborhood is defined that includes a number of points as obtained from a point cloud selected by their proximity. For our experiments we tested different neighborhood sizes (6, 8, 16, 32, 64 and 128 neighbors). Since results were independent of the neighborhood size and larger sizes meant more computation time, we used the smallest neighborhood size tested (six neighbors; less would be insufficient for our approach). Points falling inside the surface neighborhood will provide a curvature grid along the surface. We are interested in computing a close approximation to the surface neighborhood principal curvatures. We will do this by obtaining the maximum curvature on the surface neighborhood and its orthogonal. First, curvatures between the point of interest (p, at the center of the neighborhood), and every other point (q_i) falling within the neighborhood are computed as (Fig. 1a)

$$k_i = \left\| \frac{dT_{p,q_i}}{ds_{p,q_i}} \right\|, \tag{1}$$

where dT is the difference in tangent between the curve joining points p and q_i, and ds is the Euclidean distance between the points. Tangent vectors are easily obtained by noticing that the normal vector associated to every point is orthogonal to its tangent plane (assuming that the point cloud is sampled from a smooth manifold). If we project the line joining p and q_i into those tangent planes, we obtain the tangent vectors corresponding to p and q_i. Thus, we can obtain dT and the curvatures between p and q_i (black curves in fig. 1a and blank circles in fig. 1b). Each tangent vector also has an associated direction (x, y and z vector values at the colored circles in fig. 1c).

1) Curvatures on the surface: At this point we have a number K ($K = 5$ in our case) of curvature values. The surface curvature will be obtained from the principal curvatures, i.e. the maximum curvature obtained at a given point, and its orthogonal curvature. These two curvatures will define the surface neighborhood as shown in figure 1a. But since we are obtaining curvatures from isolated points in the surface, the point-wise maximum and minimum curvatures on the surface neighborhood may be quite different to the real principal curvatures in the surface due to quantization (we have access to a set of points in the surface). In order to solve this problem and obtain better surface curvature approximations, we fit all the curvatures k_i with a spline interpolation (Fig. 1b) as well as also interpolating their associated tangent directions (Fig. 1c).

Splines are piecewise polynomial functions connected smoothly by joining points commonly known as knots with a polynomial. Cubic splines are splines of order 4 (four coefficients are required to specify a cubic polynomial). In our case, the point-wise k_i values obtained previously would be the knots in a 1-D spline (Fig. 1b). A spline is defined in each of the g knot intervals as [33]

$$s(x) = \sum_{j=0}^{m} c_{j,i}(x-k_i)^j,\ i = 0,\ldots,g, \tag{2}$$

for a spline of order $m+1$ ($m = 3$ in our case) with coefficients $c_{j,i}$. These coefficients are obtained by solving a tridiagonal linear system [34].

Descriptor	SCurV	ESF	VFH	CVFH	OUR-CVFH	PFH	FPFH	SHOT	USC	3DSC	SI
Global/Local	Both	Global	Global	Both	Both	Local	Local	Local	Local	Local	Local
Size	Global: 231, Local: 1000	640	308	308	308	125	33	352	1980	1980	153

TABLE I

DESCRIPTORS USED IN THE EXPERIMENTAL SECTIONS. SEE THE PCL [31] DOCUMENTATION FOR DETAILS.

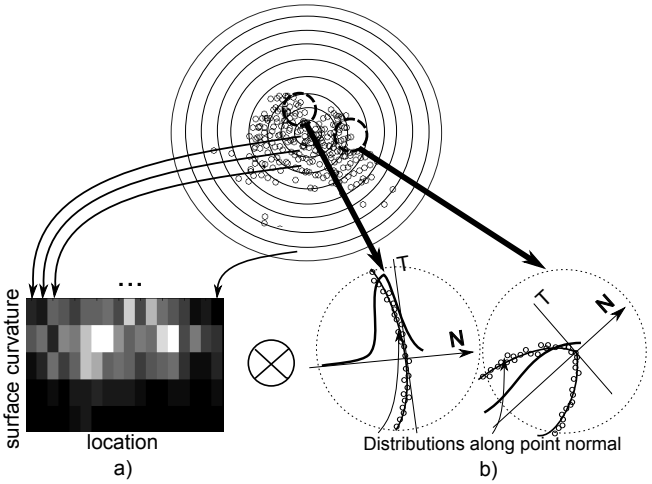

Fig. 2. Summary of the descriptor. SCurV is the tensor product between a global shape representation (a) and a local viewpoint distribution component (b). Top: Point cloud. a) Surface curvature-location representation. Plotted are the averaged curvatures after spline interpolation of the two orthogonal surface curvatures (Section III-A). For display purposes not all spherical bins are shown. b) The local distribution corresponds to the projection of the neighboring points onto the normal of the center point (Section III-B). The left inset shows the point distribution corresponding to a flat surface; the distribution on the right corresponds to a sharp/pointy surface.

Every q_i point falling in the neighborhood of the central point p is projected into its associated tangent plane (defined through its normal vector). They are then attached to a polar coordinate angle (Ω) with respect to the center point p on this projected plane. The knots of the spline intepolation will correspond to their point cloud quantized curvature k_i values (y-axis in Fig. 1b) with respect to those polar coordinate angles (Ω values, x-axis in Fig. 1b). The first knot will be also the last in order to close the spline. This same process is used to interpolate their tangent vector coordinates with respect to the central point p as shown in fig.1c. Back to the computation of principal curvatures, the largest value of the spline computed this way will correspond to the maximum curvature on the surface manifold, k_a (Fig.1b). The curvature whose tangent is orthogonal to k_a's tangent along the spline will be the other curvature of interest (k_b) to describe the surface neighborhood curvature. Note that it will not always correspond to the minimum curvature in the spline due to interpolation errors, although it will be close to it. In order to compute this latter curvature, we must obtain the vector that is perpendicular to the tangent of curvature k_a inside the surface neighborhood. The Ω values in figures 1b and 1c x-axis relate the curvatures with their interpolated tangent directions in the neighborhood surface. Thus, we must obtain the tangent direction orthogonal to the tangent corresponding to k_a (Fig. 1c). Its polar angle value Ω will provide us with the curvature k_b and is found by computing the cosine of the angle between the tangent direction of k_a and the rest of interpolated tangent directions. The place where it is closest to 0 provides the polar angle value (Fig. 1c) for obtaining the corresponing k_b in figure 1b.

There exist two commonly-used surface curvature measures, the Gaussian and mean curvatures. Since using both is quite redundant, we use the mean curvature (the arithmetic mean of the principal curvatures) $k_{\text{mean}} = \frac{1}{2}(k_a + k_b)$ in our SCurV descriptor since it produces better results.

2) Object radius computation: Part of the information contained by the descriptor consists of a 2D histogram (Fig. 2a) where one dimension (the y-axis in fig. 2a) is the surface curvature computed before and grouped into bins in a range [0,1] (steps of 0.1 in our experiments). The other dimension consists of the radial distance of the curvature value with respect to the centroid of the object grouped in bins inside the sphere that covers the object (x-axis in fig. 2a). The number of spherical radial bins used is 21, and the number of curvature bins is 11. This 2D histogram is transformed into a vector ϕ^{gl} for its combination with the viewpoint representation explained next.

B. Local viewpoint-centered representation

For each point, the t nearest points (t must be sufficiently large to cover large neighborhoods, e.g. 100) are selected and projected onto the line going through that point with the direction of the corresponding normal vector (Fig. 2b). After sorting the projections we have a data sequence that approximates the inverse function of the one-dimensional distribution of the projections. For example, if a point lies on a flat surface, that distribution concentrates in one point; if the point is on a convex (concave) surface, then the bulk of the projected distribution falls into the negative (positive) part of its normal. In this way the projected distribution can express the properties of the local shape.

The sorted vectors of projections are clustered by k-means clustering to create "bag-of-words" type feature vectors (the number of clusters is 1000 in our experiments). For each object, we then group the points into a histogram by counting those falling into the same cluster. The histograms are normalized to have unit L_1 norm to form a proper probability density function, producing a vector ϕ^{lc}.

C. SCurV as a result of the combination of global object-centered and local viewpoint-centered shape representations

We use the tensor product to combine the object-centered and the viewpoint-centered representations computed before. We prefer the tensor product over concatenation because when feature vectors are concatenated, then the family of the distributions and the type of kernels could be lost, e.g. if the features follow a Gaussian distribution, the concatenated

feature would generally not follow that distribution. On the other hand, the marginal distribution is preserved by the tensor product. The reason for this is that if the feature representations have independent distributions, then the marginal distributions of the tensor product reproduce the distributions of the original factors.

The feature vectors of the 3D objects generated by the two different procedures explained before are combined by the tensor product, which implicitly yields all possible combinations of the components of those vectors. Here we apply the tensor product as an operation which can join two independent vector spaces into one. One can show that the tensor product of the features leads to a kernel which is the point-wise product of the kernels built up on the distinct features separately; thus the computation of the joint feature has a complexity of $O(n^2)$, where n is the number of sample items considered. It is based on the identity, valid in any Hilbert space,

$$\langle \otimes_{r=1}^{n_r} \phi^r(x_i), \otimes_{r=1}^{n_r} \phi^r(x_j) \rangle = \prod_{r=1}^{n_r} \langle \phi^r(x_i), \phi^r(x_j) \rangle , \quad (3)$$

where $\langle , \rangle$ denotes the inner product, and $\otimes_{r=1}^{n_r}$ expresses the tensor product of n_r different vectors (n_r=2 in our case). This identity allows us to combine features in a computationally simple way: Kernels can be independently computed and combined from all possible configurations. For example if we consider our two feature representations, ϕ^{gl} and ϕ^{lc}, then the joint kernel $\mathbf{K}$ can be computed by the point-wise product of the kernels $\mathbf{K}^{\phi^{gl}}$ and $\mathbf{K}^{\phi^{lc}}$, since

$$\begin{aligned} \mathbf{K}_{ij} &= \langle \phi^{gl}(x_i) \otimes \phi^{lc}(x_i), \phi^{gl}(x_j) \otimes \phi^{lc}(x_j) \rangle \\ &= \langle \phi^{gl}(x_i), \phi^{gl}(x_j) \rangle \langle \phi^{lc}(x_i), \phi^{lc}(x_j) \rangle \quad = \mathbf{K}_{ij}^{\phi^{gl}} \mathbf{K}_{ij}^{\phi^{lc}} . \end{aligned} \quad (4)$$

This equation shows that even though the starting point is the tensor product, it is implemented as the point-wise product of the kernels, avoiding the need to work with very large dimensions as well as being computationally more efficient. For more details on the tensor product, the theoretical background and proofs of the statements included here, please consult [35].

IV. EXPERIMENTAL SETUP

A. The Princeton Shape Benchmark 3D object model database

The Princeton Shape Benchmark (PSB) [30] consists of 1814 synthetic objects or models grouped into 90 training and 92 test classes. We will use this dataset not in an image retrieval task as most of other works do, but on two different classifcation tasks. Since many classes in the test do not appear in the training (and viceversa), for our classification purposes we grouped all training and test classes, giving a total of 161 different classes. Even though it is extensively used in the shape-based retrieval literature, it has attracted little attention from the 3D descriptor or 3D object recognition literature due to the heterogeneity in terms of classes and intraclass representations of 3D objects (see fig. 3a for some examples). The main reasons why this is a challenging database was already reported by Jayanti et al. [36], and maybe that is why it has been neglected in the 3D object classification literature. Nevertheless, we found that 1) this dataset contains a very large number of heterogeneous classes and objects, 2) objects are classified hierarchically and 3) it is the closest to how humans hierarchically classify the world (eg. fig. 3b). This three aspects are of interest for 1) test our SCurV descriptor on a large and heterogeneous classification task, 2) evaluate it against other state-of-the-art descriptors on that task, and 3) perform a novel hierarchical classification of object classes. For a comparison of the PSB with other databases and additional advantages of using this dataset, please consult Shilane and co-workers [30].

B. Preprocessing

The objects are represented by point clouds uniformly subsampled from the original PSB mesh files. To produce realistic point clouds out of the mesh files describing the objects, a random sampling is applied which selects points from the triangulated surfaces. In our experiments, the number of sample items for feature computation is set to 5000. The reason for using smaller samples to generate some features is the reduction of otherwise too expensive computation. Uniform sampling reduces the high variance caused by the different triangulation methods applied in the generation of the PSB. The distribution of those points is uniform with respect to the entire surface of the object, thus the probability of selecting a point from a triangle is proportional to the area of that triangle.

C. Maximum Margin Regression classifier

We will make use of the same classifier for all the descriptors. We will apply a maximum margin based regression (MMR) technique [37], which is an extension of the well known Support Vector Machine (SVM), but which provides a better scheme for flat and hierarchical multiclass classification as well as some other interesting properties, which are summarized next. In maximum margin regression, the aforementioned descriptors will serve as input, and the predicted outputs represent the classes in the flat and hierarchical classification tasks. The outputs, the classes, are also represented via feature vectors to allow the application of a regression approach (MMR) which can reduce the computational complexity of the flat and hierarchical multiclass learning to the complexity of a single binary classification.

MMR relies on the fact that the normal vector of the separating hyperplane can be interpreted as a linear operator mapping the feature vectors of input items into the space of the feature vectors of the outputs. In the case of the SVM, the space of the outputs is a simple one-dimensional space only containing the vectors (-1) and $(+1)$ corresponding to the two classes. Multiclass (including hierarchical) classification is a simple application of the MMR framework. In this scenario, a feature vector of class c corresponding to an instance i is represented as an indicator vector of dimension equal to the number of classes, i.e., the vector component c is equal to 1 and all others are 0. Other multiclass representations which

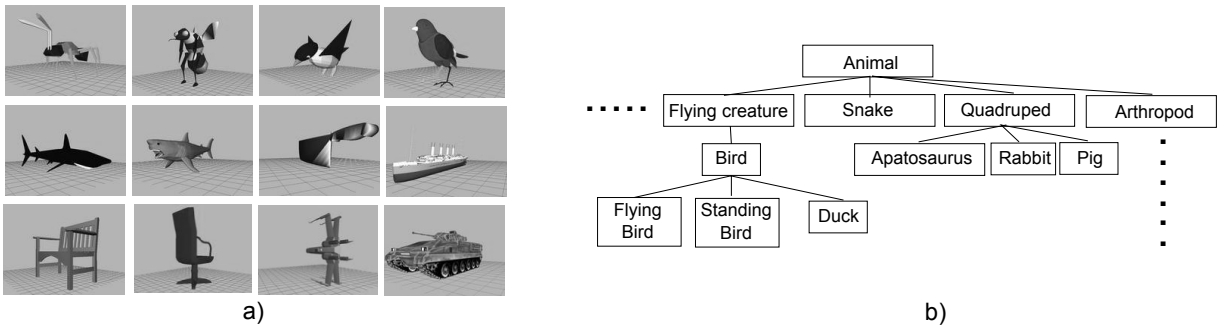

Fig. 3. a) 3D object examples from the Princeton Shape Benchmark, including insects (ant, bee), birds, sharks, a knife, a boat, chairs, a fighter jet and a tank. b) Examples of classes organized into hierarchies in the Princeton Shape Benchmark

can incorporate available prior knowledge (e.g. imbalanced class proportions) can be performed via appropriately chosen output features. Further details regarding MMR can be found in [37].

D. Evaluation: Hierarchical and flat multiclass classifications

A hierarchy of classes is represented by a rooted directed acyclic graph (DAG). The root corresponds to a superclass containing all subclasses. The next layer consists of those subclasses which are not contained by any other classes. Every new layer consists of those classes which are contained by any class one layer closer to the root. The directed edges start from the root and point to the contained classes from the containers.

To build the kernel on top of this type of representation the following steps are applied: (1) The nodes of the graph are indexed by topological order, i.e., if there is a directed edge from node A to node B then $i_A < i_B$ holds for their labels. (2) The feature vector $\phi(N)$ of a node N is an element of $\{0,1\}^n$, where n equals the number of nodes. The components of $\phi(N)$ correspond to the nodes and are indexed by their topological order. (3) A component of the feature vector $\phi(N)$ is 1 if the corresponding node is on the shortest path from the root to N, otherwise it is set to 0. This will correspond to the indicator of the path from root to node.

For the evaluation of the hierarchical classification we followed the work of [38]. Thus, the best evaluation measures for this type of classification are precision, recall and their combination into the F1 score. They are given by combinations of the true positives T_p, false positives F_p and false negatives F_n,

$$P = \frac{T_p}{T_p + F_p},\ R = \frac{T_p}{T_p + F_n},\ F1 = \frac{2PR}{P+R} \tag{5}$$

where P is the precision and R is the recall. A perfect descriptor would have a recall value of 1 for any precision. The $F1$ measure summarizes both as their harmonic mean.

In the hierarchical classification task, nodes that appear both in the true and in the corresponding predicted paths leading from the root node to the object class in the graph are considered as the true positives. False positives nodes are those that lie on the predicted paths but not on the corresponding true paths. False negatives are those nodes that can be found in the true paths but not in the corresponding predictions.

In the flat classfication task, the classes can be modeled by a two-level tree graph where one level contains only the root node, and all object classes nodes occur in the second level. Since the path consists of second-level nodes connected only to the root, the numerical values of F_p and F_n are equal (as are then P and R). Thus, for the flat classification we evaluate the *accuracy*, defined as the number of instances in the diagonal of the confusion matrix divided by the total number of instances.

In both the hierarchical and flat classification cases the root node is excluded when the precision, recall, F1 and accuracy scores are computed.

V. EXPERIMENTAL RESULTS

In the computation of the prediction results, a 5-fold cross-validation procedure was applied to all descriptors. In the learning procedure, first linear base kernels were computed from the corresponding features. On top of those kernels, a Gaussian non-linear kernel transformation was applied to increase the flexibility and prediction capability of the underlying features. The parameter corresponding to each Gaussian kernel was found by cross validation restricted to the training data, which we divided into validation test and validation training partitions. Then, the learner was trained only on the validation training items. The values of the parameters for MMR were chosen such as to maximize the $F1$ and *accuracy* scores on the validation test for the hierarchical and flat multiclass classifications, respectively.

A. Flat classification

By flat multiclass classification (or just, flat) we mean the classical classification where all objects are classified as being equally different. Table II shows that the SCurV descriptor outperforms all other descriptors at a classical classification task with the 161 different classes and 1814 different objects from the PSB. We also report the case when we use the simple concatenation of the global and

	Flat	Hierarchical		
Descriptor	accuracy	Precision	Recall	F1
SCurV	**0.39**	**0.56**	**0.54**	**0.55**
concat(ϕ^{gl},ϕ^{lc})	**0.35**	**0.45**	**0.45**	**0.45**
ESF	0.34	0.47	0.45	0.46
CVFH	0.20	0.32	0.31	0.31
OUR-CFH	0.18	0.27	0.25	0.26
VFH	0.14	0.27	0.26	0.26
SHOT	0.14	0.24	0.23	0.23
USC	0.12	0.19	0.17	0.18
3DSC	0.11	0.19	0.17	0.18
SI	0.10	0.18	0.16	0.17
PFH	0.08	0.16	0.14	0.15
FPFH	0.08	0.16	0.14	0.15

TABLE II

CLASSIFICATION RESULTS FOR SCURV AND TEN OTHER DESCRIPTORS.

	Flat	Hierarchical		
Descriptor	accuracy	Precision	Recall	F1
SCurV	**0.36**	**0.54**	**0.52**	**0.53**
ESF	0.34	0.47	0.44	0.46
CVFH	0.20	0.31	0.29	0.30
OUR-CFH	0.14	0.25	0.23	0.24
VFH	0.12	0.22	0.21	0.22

TABLE III

CLASSIFICATION RESULTS UNDER NOISE FOR SCURV AND FOUR OTHER DESCRIPTORS.

local representations (instead of the tensor product). ESF provides very good results as well on this task. The other 3D descriptors accuracy scores proved to be worse. We can also see that VFH, CVFH and OUR-CVFH perform much better than either PFH and FPFH alone. As commented in section II, they extend PFH with the concatenation of new elements, showing the importance of combining different sources of information in the descriptor.

B. Hierarchical classification

The results produced by SCurV and the other eight descriptors on a hierarchical multiclass classification of the Princeton Shape Benchmark data are also found in table II. The precision and recall were computed node-wise for all paths representing the classes of the test items. In this way not only the *leaf* classes were considered, but also the higher level superclasses were taken into account. Our descriptor outperforms all competition, being followed closely by ESF. CVFH, OUR-CVH, SHOT and VFH and the shape context approaches show similar performance.

We would like to stress here that even though these results are better than for the flat classification, they cannot directly be compared to those. Results in these experiments are boosted for all the descriptors by the simple fact that where a correct classification in the previous experiment only considered the leaf classes, now the classes above them are also considered.

C. Robustness to noise

Curvature-based computations are prone to be influenced by noise. All the objects from the PSB were corrupted by adding Gaussian noise ($variance = 0.1$) to every object point, and we ran the same classification processes. For comparison we used the four best-performing competing descriptors (ESF, CVFH, OUR-CVFH and VFH). Table II shows the robustness of ScurV as well as those other four descriptors. ESF and CVFH results were barely influenced by noise, while SCurV, OUR-CVFH and VFH provided classification scores below the noise-free results. Even though SCurV is affected by noise, the spline interpolation mitigates its effect. Thus, it is less prone to noise than the other two histogram-based feature descriptors.

D. Discussion

a) ***Descriptor performance comparison****:* Our descriptor clearly outperforms all other descriptors at the task of object classification. The reason for this is the fact that our approach focuses on a descriptor that combines a global surface representation through curvature computation and a local viewpoint distribution. We performed our experiments using different types of descriptors (table I). In this work we are interested in their classification capabilities. This task (as well as the dataset) may suit global descriptors better than local descriptors. Nevertheless, the study performed here is of great interest for two reasons: First, the task involved was not instance recognition but class recogntion. Thus we tested how well different descriptors withstand intraclass variation. Secondly, this was done over a very large dataset containing many different object classes, thus also evaluating the class-discriminative power of the descriptors tested. Our evaluation holds for object classification only. Our results and conclusions do not necessarily carry over to other tasks.

We tested SCurV and four other descriptors on the same classifcation tasks with added noise. Noise had a higher effect on SCurV than on ESF or CVFH, although it still maintained the highest classification scores. It would be interesting to test how much noise can SCurV withstand compared to its competitors. Another limitation of our approach is computation time. Every object required approximately 40 seconds on an Intel CPU Q9550. ESF and the Feature Histogram approaches were faster; SHOT run on similar timing and all others were computationally even more intensive. The part most computationally intensive in SCurV was the k-means clustering (section III-B). In fact, the object-centered component (section III-A) took just slightly less than a second per object. For future work we plan to replace k-means by another, more efficient clustering method. Finally, our descriptor still has to prove its validity under occlusion, clutter, viewpoint and non-rigid shapes.

b) ***Advantages of combining features with the tensor product****:* As mentioned in section III-C, the tensor product preserves the family of the distributions of the factors as well as the type of the kernels, i.e. the pointwise product of Gaussian kernels is also Gaussian. Similarly, the pointwise product of polynomial kernels also remains polynomial.

The advantage of the tensor product is shown in the results, compare in table II, SCurV with concat(ϕ^{gl},ϕ^{lc}). We performed some further tests to evaluate the tensor product

on feature combinations. The best case occurred when we combined SCurV with ESF; the classification performance increased to an *accuracy* = 0.43 in the flat classification and an *F1 score* = 0.58 in the hierarchical classification. Compare these values to those of SCurV and ESF in table II, where they were considered separately.

*c) **We performed a novel hierarchical classification evaluation of descriptors**:* It would be of interest for future work to evaluate if the proposed scheme for hierarchical classification can have an effect on improving the overall classification results at the lowest level. Thus, to study if using hierarchical information can correctly classify leaf nodes that were misclasified when using flat classificaiton, i.e. could hierarchical information be used for a better overall multiclass classification performance?

VI. CONCLUSIONS

We have presented a 3D descriptor, named SCurV, which is designed for object class classification tasks. SCurV combines a global-surface and a viewpoint-dependent representations in a novel way using the tensor product. Our descriptor outperformed other state-of-the-art descriptors at two classification tasks with and without noise, where we performed a flat and a novel hierarchical multiclass classification tasks.

ACKNOWLEDGMENT

The research leading to these results has received funding from the EU seventh Framework Programme FP7/2007-2013 under grant agreement no. 270273, Xperience.

REFERENCES

[1] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, "Object recognition with cortex-like mechanism," *IEEE TPAMI*, vol. 29, no. 3, pp. 411–426, 2007.

[2] A. Rodríguez-Sánchez and J. Tsotsos, "The importance of intermediate representations for the modeling of 2D shape detection: Endstopping and curvature tuned computations," *CVPR*, pp. 4321–4326, 2011.

[3] G. Azzopardi, A. Rodríguez-Sánchez, J. Piater, and N. Petkov, "A push-pull CORF model of a simple cell with antiphase inhibition improves SNR and contour detection," *PLOS ONE*, vol. 9, no. 7, p. e98424, 2014.

[4] Y. Yamane, E. Carlson, K. Bowman, Z. Wang, and C. Connor, "A neural code for three-dimensional object shape in macaque inferotemporal cortex." *Nature Neuroscience*, vol. 11, no. 11, pp. 1352–1360, 2008.

[5] C. Baldassi, A. Alemi-Neissi, M. Pagan, J. J. DiCarlo, R. Zecchina, and D. Zoccolan, "Shape similarity, better than semantic membership, accounts for the structure of visual object representations in a population of monkey inferotemporal neurons," *PLOS Computational Biology*, vol. 9, no. 8, p. e1003167, 2013.

[6] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE TPAMI*, vol. 24, no. 24, pp. 509–522, 2002.

[7] M. Körtgen, M. Novotni, and R. Klein, "3D shape matching with 3D shape contexts," in *The 7th Central European Seminar on Computer Graphics*, 2003.

[8] F. Tombari, S. Salti, and L. Di Stefano, "Unique shape context for 3D data description," in *Proc. of the ACM workshop on 3D object retrieval*. ACM, 2010, pp. 57–62.

[9] I. Kokkinos, M. Bronstein, R. Litman, and A. Bronstein, "Intrinsic shape context descriptors for deformable shapes," in *CVPR*, 2012, pp. 159 –166.

[10] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE TPAMI*, vol. 21, no. 5, pp. 433–449, 1999.

[11] X. Wang, Y. Liu, and H. Zha, "Intrinsic spin images: A subspace decomposition approach to understanding 3D deformable shapes," in *Proc. 3DPVT*, vol. 10, 2010, pp. 17–20.

[12] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz, "Towards 3D point cloud based object maps for household environments," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 927–941, 2008.

[13] R. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3D recognition and pose using the viewpoint feature histogram," in *IROS*, 2010, pp. 2155 –2162.

[14] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *ICRA*, 2009, pp. 3212 –3217.

[15] A. Aldoma, M. Vincze, N. Blodow, D. Gossow, S. Gedikli, R. B. Rusu, and G. Bradski, "CAD-model recognition and 6DOF pose estimation using 3d cues," in *ICCV Workshops*, 2011, pp. 585–592.

[16] A. Aldoma, F. Tombari, R. B. Rusu, and M. Vincze, "OUR-CVFH–oriented, unique and repeatable clustered viewpoint feature histogram for object recognition and 6dof pose estimation." ÖAGM-DAGM, 2012.

[17] W. Wohlkinger and M. Vincze, "Ensemble of shape functions for 3D object classification," in *ROBIO*, 2011, pp. 2987 –2992.

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

[19] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3D object recognition," in *CVPR*, 2010, pp. 998–1005.

[20] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Rotation invariant spherical harmonic representation of 3D shape descriptors," in *SIGGRAPH symposium on Geometry processing*, 2003, pp. 156–164.

[21] O. Lopes, M. Reyes, S. Escalera, and J. Gonzalez, "Spherical blurred shape model for 3-d object and pose recognition: Quantitative analysis and hci applications in smart environments," 2014.

[22] M. I. Restrepo, B. A. Mayer, and J. L. Mundy, "Object recognition in probabilistic 3-d volumetric scenes." in *ICPRAM (2)*, 2012, pp. 180–190.

[23] H. Kim and A. Hilton, "Evaluation of 3D feature descriptors for multimodal data registration," in *3D Vision*, 2013, pp. 119–126.

[24] L. A. Alexandre, "3D descriptors for object and category recognition: a comparative evaluation," in *IROS*, vol. 1, no. 2, 2012.

[25] Y. Salih, A. S. Malik, N. Walter, D. Sidibé, N. Saad, and F. Meriaudeau, "Noise robustness analysis of point cloud descriptors," in *Advanced Concepts for Intelligent Vision Systems*. Springer, 2013, pp. 68–79.

[26] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view rgb-d object dataset," in *ICRA*, 2011, pp. 1817–1824.

[27] F. Tombari, S. Salti, and L. Di Stefano, "Performance evaluation of 3D keypoint detectors," *International Journal of Computer Vision*, vol. 102, no. 1-3, pp. 198–220, 2013.

[28] C. B. Akgul, B. Sankur, Y. Yemez, and F. Schmitt, "3D model retrieval using probability density-based shape descriptors," *IEEE TPAMI*, vol. 31, no. 6, pp. 1117–1133, 2009.

[29] E. Boyer, A. M. Bronstein, M. M. Bronstein, B. Bustos, T. Darom, R. Horaud, I. Hotz, Y. Keller, J. Keustermans, A. Kovnatsky, *et al.*, "SHREC 2011: robust feature detection and description benchmark," *arXiv preprint arXiv:1102.4258*, 2011.

[30] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser, "The Princeton Shape Benchmark," in *Shape Modeling Applications*, 2004, pp. 167–178.

[31] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *ICRA*, 2011, pp. 1–4.

[32] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Shape distributions," *ACM T on Graphics*, vol. 21, no. 4, pp. 807–832, 2002.

[33] M. Bennamoun and G. Mamic, *Object Recognition, fundamentals and case studies*. Springer, 2002.

[34] C. De Boor, "A practical guide to splines," 1978.

[35] J. Lee, *Introduction to Smooth Manifolds*, ser. Graduate Texts in Mathematics. Springer, 2003, vol. 218.

[36] S. Jayanti, Y. Kalyanaraman, N. Iyer, and K. Ramani, "Developing an engineering shape benchmark for CAD models," *Computer-Aided Design*, vol. 38, no. 9, pp. 939–953, 2006.

[37] K. Astikainen, L. Holm, E. Pitkänen, S. Szedmak, and J. Rousu, "Towards structured output prediction of enzyme function," in *BMC Proceedings, 2 (Suppl 4):S2*, 2008.

[38] N. Cesa-Bianchi, C. Gentile, A. Tironi, and L. Zaniboni, "Incremental algorithms for hierarchical classification," *NIPS*, 2004.