Histograms of Oriented Gradients for 3D Object Retrieval

Maximilian Scherer
Interactive Graphics Systems Group
TU Darmstadt, Germany
maximilian.scherer@gris.tu-darmstadt.de

Michael Walter
Interactive Graphics Systems Group
TU Darmstadt, Germany
michael.walter@gris.tu-darmstadt.de

Tobias Schreck
Interactive Graphics Systems Group
TU Darmstadt, Germany
tobias.schreck@gris.tu-darmstadt.de

ABSTRACT

3D object retrieval has received much research attention during the last years. To automatically determine the similarity between 3D objects, the global descriptor approach is very popular, and many competing methods for extracting global descriptors have been proposed to date. However, no single descriptor has yet shown to outperform all other descriptors on all retrieval benchmarks or benchmark classes. Instead, combinations of different descriptors usually yield improved performance over any single method. Therefore, enhancing the set of candidate descriptors is an important prerequisite for implementing effective 3D object retrieval systems.

Inspired by promising recent results from image processing, in this paper we adapt the Histogram of Oriented Gradients (HOG) 2D image descriptor to the 3D domain. We introduce a concept for transferring the HOG descriptor extraction algorithm from 2D to 3D. We provide an implementation framework for extracting 3D HOG features from 3D mesh models, and present a systematic experimental evaluation of the retrieval effectiveness of this novel 3D descriptor. The results show that our 3D HOG implementation provides competitive retrieval performance, and is able to boost the performance of one of the best existing 3D object descriptors when used in a combined descriptor.

Keywords: 3d model, retrieval, hog, gradient, histogram

1. INTRODUCTION

The increasing number of available 3D models is accompanied by the need for ever more sophisticated methods to retrieve those models. There are several ways to tackle this task[16]. One prominent approach, which we also adhere to, is the computation of high-dimensional *feature vectors* to describe the *global shape* of 3D models. The (dis-)similarity between two models can then be expressed by the distance between two feature vectors. So under these conditions, the remaining task at hand is the extraction of feature vectors, that are able to capture and effectively discriminate the shape of 3D models. Note that for the remainder of this paper, we will use the terms feature vector and descriptor interchangeably, as feature vectors are the only descriptors we consider.

In the past, several methods that have proven to be effective in image analysis were successfully adapted to the task of 3D object retrieval and classification[5, 18]. Intuitively, this can be explained by the fact that the human perceptual system judges *similarity* of 3-dimensional shapes to a large extent based on 2-dimensional projections. This strongly motivates our approach of adapting the successful "**H**istograms of **O**riented **G**radients"[6] feature computation from 2D images to 3D models.

The contribution of this paper is twofold. First, we extend the HOG descriptor to 3D mesh models, evaluate its performance, and make the implementation publicly available. Second, we provide a framework for the extraction of different descriptor instances based on the gradient computation method employed. By choosing a method to compute gradients from meshes, our implementation may be easily adapted to compute corresponding feature vectors (see Section 3 for details).

The remainder of this paper is structured as follows. After discussing related work in Section 2, we discuss the HOG descriptor in Section 3. Specifically, the definition of HOG in the 2D image domain is reviewed, as well as our approach to apply it in the 3D domain is discussed. The main challenge of defining suitable gradients on 3D meshes is considered in detail. Section 4 demonstrates the effectiveness of our descriptor by evaluating it on several well-known 3D retrieval benchmarks. A detailed discussion of the obtained results is provided in Section 5. Section 6 concludes and outlines options for future work in this area.

2. RELATED WORK

There is a wealth of available approaches for 3D model retrieval [16], which can be classified into several different categories. In terms of the addressed retrieval problem, methods can be distinguished by supporting global or partial model similarity. Global methods determine the overall similarity of the shape of whole models, while partial methods analyze for local similarities. Note that in both cases, as similarity is a rather fuzzy and domain-dependent concept, there are no absolute solutions to neither the global nor the local 3D similarity problem. Consequently, all methods proposed so far in 3D object retrieval are of heuristic nature, and their usefulness for solving a specific retrieval task needs to be evaluated experimentally.

In our work, we consider the global 3D similarity problem. Our approach compares the global shape of 3D models by relying on histograms of 3D *gradients*. Gradient is not used in a strict mathematical sense throughout this paper, but rather to describe a notion of *intensity* at a given 3D coordinate in a certain direction (see Section 3). Therefore, our approach relates to histogram-based and gradient-based shape description methods.

A histogram-based approach is proposed by Zaharia et al in [21], called *Shape Spectrum Descriptor*. There, the histogram contains values based on the curvature of several points on the model surface. We will briefly come back to such an approach when describing future work (Section 6). In [1], the authors decompose the object space into cells and compute a global *3D Shape Histogram*, where the frequency of 3D points that fall into a given cell form one bin in the shape histogram. A recent approach that relies on surface normals called *Multishell Extended Gaussian Images* was proposed by Wang et al in [4]. In contrast to our approach, these normals are not encoded in histograms. Rather, they are mapped onto the unit sphere and this spherical function is then expanded into spherical harmonics. The resulting coefficients form the basis for this descriptor.

In [8], Glomb extends the Harris-Operator[9] to 3D meshes. He faces similar challenges as we do, since corner detection using the Harris-Operator relies on the aggregated value of gradients in a specific region. Accordingly, he proposes several methods to define gradients on meshes and also evaluates them. We will come back to his results later (see Section 6).

Our approach utilizes Euclidean distance fields for gradient-computation, which is a well-studied field with many differing applications. For further details please refer to [7, 11].

Due to the heuristic nature of the definition of 3D shape descriptors, experimental evaluation of their retrieval performance is important. Several 3D retrieval benchmarks have been established to date. The benchmarks each consist of a set of 3D mesh models along with a classification of models into similarity classes (ground truth). Generic benchmarks such as the Princeton Shape Benchmark [14] or the Konstanz 3D Shape Benchmark [3] contain generic models such as humans, cars, animals, furniture etc. More specialized benchmarks containing e.g., mechanical engineering models [10] or furniture models [20] exist. These benchmarks can be used to compute metrics such as Precision and Recall to compare the retrieval performance of individual 3D descriptors.

3. 3DHOG DESCRIPTOR

In this section, we first recall the 2D HOG algorithm, and then introduce the proposed concept to adapt it to the 3D domain. Alternatives for calculation of 3D gradients from a distance field representation of 3D mesh models are discussed. An implementation of the 3DHOG descriptor is described and made available.

3.1 The HOG Descriptor for Images

In the 2D image domain, successful methods like Harris-Corner detection [9] and the well-known Scale-Invariant-Feature-Transform (SIFT) algorithm [12] rely on aggregated gradients. The HOG descriptor organizes gradients into histograms. As the first step, the gradient image is computed by convolving the input image with an appropriate filter mask. Then, a grid of histograms is constructed, where each histogram organizes the respective gradients into bins according to their orientation. Each gradient votes into the corresponding bin using its length. To preserve locality, such a histogram is computed for each cell in an evenly spaced grid. Accordingly, each cell contains the same number of gradients (depending on the cell size), and gets assigned exactly one histogram. The cells themselves are then organized in rectangular blocks, which might even overlap. The histogram values of all cells within one block are concatenated to form a vector. The vector of each block is then normalized and subsequently, the concatenation of all those block-vectors yields the final feature vector. Note that the order of concatenation is arbitrary but fixed, hence an entry of the resulting feature vector always contains information about the same image region (the *block* it originates from). Further details and an evaluation of the 2D HOG descriptor can be found in [6].

3.2 Extending HOG to 3D Meshes

Extending the HOG approach to 3D mesh models mainly consists of two steps. First, we need to extract *gradients* from our mesh. Second, we need to organize these 3-dimensional gradients into bins using appropriate histograms computed over uniformly spaced grid-cells. The second step is straightforward, as we simply extend the grid and histogram dimension each by one. Then, we can convert each gradient into spherical coordinates (eq. 1) and bin it according to its orientation (*zenith* $\theta \in [0, \pi)$ and *azimuth* $\phi \in [0, 2\pi)$).

$$\begin{pmatrix} \theta \\ \phi \\ r \end{pmatrix} = \begin{pmatrix} \arccos \frac{z}{\sqrt{x^2+y^2+z^2}} \\ \operatorname{atan2}(x,y) \\ \sqrt{x^2+y^2+z^2} \end{pmatrix} \tag{1}$$

The first step, however, does not generalize as easily.

To compute the image-gradient, several approaches are considered in [6]. According to their evaluation, the convolution of the image with a 1D $[-1, 0, +1]$ filter mask is most suitable. This approximates the partial first-order derivative according to:

$$\begin{aligned} \nabla f(x,y) &= \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right)^T && (2) \\ &\approx \begin{pmatrix} f(x+1,y) - f(x-1,y) \\ f(x,y+1) - f(x,y-1) \end{pmatrix} && (3) \end{aligned}$$

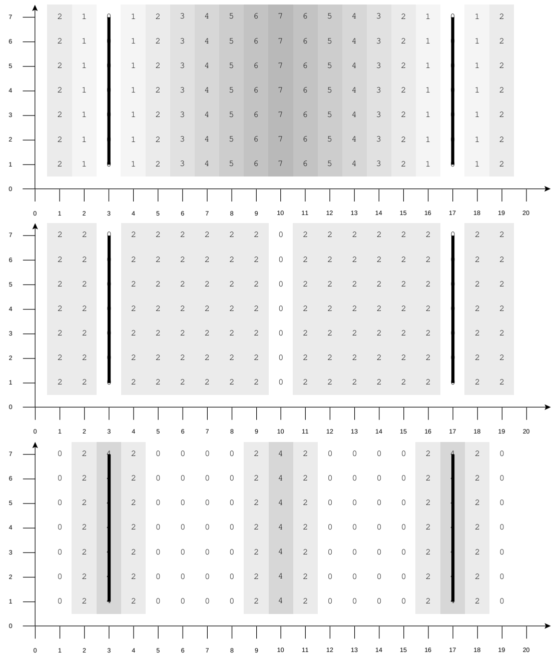
To extend this to 3D, we need a pre-processing step to compute a notion of *neighborhood* and *intensity* from a polygonal mesh. We do this by computing a 3-dimensional *Euclidean distance field* as a real valued function defined on a discrete, regular 3D grid spanning the entire volume the mesh occupies. We will denote grid-cells as *voxels*. Each voxel contains the distance from its center to the surface of the mesh. To formalize this statement, let us define a distance function $f : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \mapsto \mathbb{R}$, which maps each voxel, defined by its indices $x, y, z \in \mathbb{N}$, to its corresponding value:

$$f(x,y,z) = \min_{\mathbf{x} \in \Sigma} \|\mathbf{x} - center(x,y,z)\|_2, \tag{4}$$

where Σ is the set of all points on the surface of the mesh and $center : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \mapsto \mathbf{p}$ is a function that returns the center coordinate of a given voxel.

Once the distance field is computed, the gradient computation can be carried out analogously to the gradient computation on images by convolving the distance field with the $[-1, 0, +1]$ filter mask in three dimensions.

Note that the distance field strongly depends on position and size of the object. This demands normalization of the mesh prior to distance field computation. We rely on established methods for mesh normalization. Specifically, we provide translation invariance by moving the center of mass of the mesh to the origin; we provide scale invariance by scaling the mesh into the unit cube. Finally, we normalize for rotation using a weighted PCA analysis. For details on these normalization steps, see [18].



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
7	2	1	0	1	2	3	4	5	6	7	6	5	4	3	2	1	0	1	2
6	2	1		1	2	3	4	5	6	7	6	5	4	3	2	1		1	2
5	2	1		1	2	3	4	5	6	7	6	5	4	3	2	1		1	2
4	2	1		1	2	3	4	5	6	7	6	5	4	3	2	1		1	2
3	2	1		1	2	3	4	5	6	7	6	5	4	3	2	1		1	2
2	2	1		1	2	3	4	5	6	7	6	5	4	3	2	1		1	2
1	2	1		1	2	3	4	5	6	7	6	5	4	3	2	1		1	2

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
7	2	2	0	2	2	2	2	2	2	0	2	2	2	2	2	2	0	2	2
6	2	2		2	2	2	2	2	2	0	2	2	2	2	2	2		2	2
5	2	2		2	2	2	2	2	2	0	2	2	2	2	2	2		2	2
4	2	2		2	2	2	2	2	2	0	2	2	2	2	2	2		2	2
3	2	2		2	2	2	2	2	2	0	2	2	2	2	2	2		2	2
2	2	2		2	2	2	2	2	2	0	2	2	2	2	2	2		2	2
1	2	2		2	2	2	2	2	2	0	2	2	2	2	2	2		2	2

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
7	0	2	4	2	0	0	0	0	2	4	2	0	0	0	0	2	4	2	0
6	0	2		2	0	0	0	0	2	4	2	0	0	0	0	2		2	0
5	0	2		2	0	0	0	0	2	4	2	0	0	0	0	2		2	0
4	0	2		2	0	0	0	0	2	4	2	0	0	0	0	2		2	0
3	0	2		2	0	0	0	0	2	4	2	0	0	0	0	2		2	0
2	0	2		2	0	0	0	0	2	4	2	0	0	0	0	2		2	0
1	0	2		2	0	0	0	0	2	4	2	0	0	0	0	2		2	0

Figure 1: 2D illustration of a distance field and derived gradient fields for a simple structure consisting of vertical bars. The top image shows the distance field. The gradient field using the $[-1, 0, +1]$ mask is shown in the middle image, while the bottom image shows the gradient field using the $[1, 0, -2, 0, 1]$ mask.

3.3 Alternative Gradient Definition

While distance fields are very convenient for adopting HOG to the 3D domain, we found the first-order gradients approximated by the simple edge detector described above not to be the best option for our purposes. To demonstrate this, we visualize a simple, 2-dimensional distance field (see figure 1, top). The mesh in this case only consists of two walls. The respective gradient field computed using the $[-1, 0, +1]$ convolution mask is shown in the middle. The numbers and grey-scales represent the magnitude of the gradients. Note that the gradients are mainly distributed among the empty space and disappear at the local extrema of the distance field, in particular at the local minima, which represent the walls in the distance field. Intuitively, this does not provide a proper analogy to gradients in images. A pixel contains information about the reflected light at a certain point in the world. Accordingly, the gradient reveals abrupt changes in intensity and an image of walls for example, would certainly exhibit strong gradients near the edges of the walls. To elaborate this aspect, we consider a different definition of *gradient*. Usually, the first order derivatives in each dimension (see equation 2) constitute the gradient. To compute gradients from the 3-dimensional distance field, we propose to use the second order derivative in each dimension. This modified gradient can be approximated by applying the simple edge detector twice, which results in the same gradients, which a convolution with a $[1, 0, -2, 0, 1]$

mask would have yielded (see equation 5). At the bottom, figure 1 shows the resulting gradient field when computed with the $[1,0,-2,0,1]$ convolution mask. Such a gradient field provides a proper analogy to image-gradients. We further elaborate on this in Section 5.

$$\nabla f(x,y,z) = \left(\frac{\partial f}{\partial x^2}, \frac{\partial f}{\partial y^2}, \frac{\partial f}{\partial z^2}\right)^T \tag{5}$$

$$\approx \begin{pmatrix} f(x-2,y,z)+f(x+2,y,z)-2\cdot f(x,y,z) \\ f(x,y-2,z)+f(x,y+2,z)-2\cdot f(x,y,z) \\ f(x,y,z-2)+f(x,y,z+2)-2\cdot f(x,y,z) \end{pmatrix} \tag{6}$$

3.4 3DHOG Extraction Algorithm

Summing up the previous subsections, we present the algorithm we implemented to extract the 3DHOG feature vector from triangular meshes. The actual Java source code of our implementation is also freely available for download[1]. For brevity's sake the details of interpolation are omited here, for further information please refer to the actual source code. Please also refer to figure 2, which summarizes the main processing steps of our 3D HOG extraction pipeline.

```
choose Size of Voxel, Cell, Block
FOREACH Model $m$ DO
    normalize Trans., Scale, Rot. of $m$
    FOREACH Voxel $v_{i,j,k}$ IN DistanceField $\mathbf{D}$ DO
        computeShortestDistance( $v_{i,j,k}, m$ )
    GradientField $\mathbf{G}$ = conv3( $\mathbf{D}, [1,0,-2,0,1]$ )
    FOREACH Gradient $\vec{g}$ IN $\mathbf{G}$ DO
        transformToSphericalCoordinates( $\vec{g}$ )
        interpolateNeighborhoodOf( $\vec{g}$ )
        insert $\vec{g}$ into its CellHistogram $h_c(\theta,\phi)$
    FOREACH CellHistogram $h_c$ DO
        with $h_c$ as $\vec{c}$
        append $\vec{c}$ to its BlockVector $\vec{b}$
    FOREACH BlockVector $\vec{b}$ DO
        append normalize( $\vec{b}$ ) to $\vec{f}_{3dhog}$
    save $\vec{f}_{3dhog}$ for model $m$
```

4. EVALUATION

We conducted experiments on several established benchmarks to evaluate the retrieval performance of the 3DHOG descriptor. We are interested in assessing the relative performance of the two alternative gradient definitions. We also compare the performance of 3DHOG with established existing 3D descriptors. Finally, we assess the retrieval performance achievable when combining 3DHOG with complementary descriptors.

4.1 Experimental setup

To evaluate the retrieval precision of 3D descriptors, several established benchmark data sets exist. For our experiments, we used the Princeton Shape Benchmark Test Partition (PSB) [14], the 2009 SHREC Generic Shape Retrieval Contest dataset (SHREC) [13], and the Konstanz 3D shape database (KN-DB) [3]. Each of these datasets includes generic 3D mesh models along with classification information grouping the models into similar classes. 3DHOG descriptors were calculated for all models, and retrieval experiments were performed by producing nearest neighbor rankings for all query objects in each benchmark data set, using the L1 norm on the respective feature vectors. *Precision versus Recall* diagrams were obtained by averaging over all queries in a given data set. In addition, several well-known single value retrieval precision measures (*nearest neighbor*, *first tier*, *second tier*, *e-measure*, *discounted cumulative gain*) [14] were calculated from the rankings. We use these measures to compare the retrieval precision of 3DHOG.

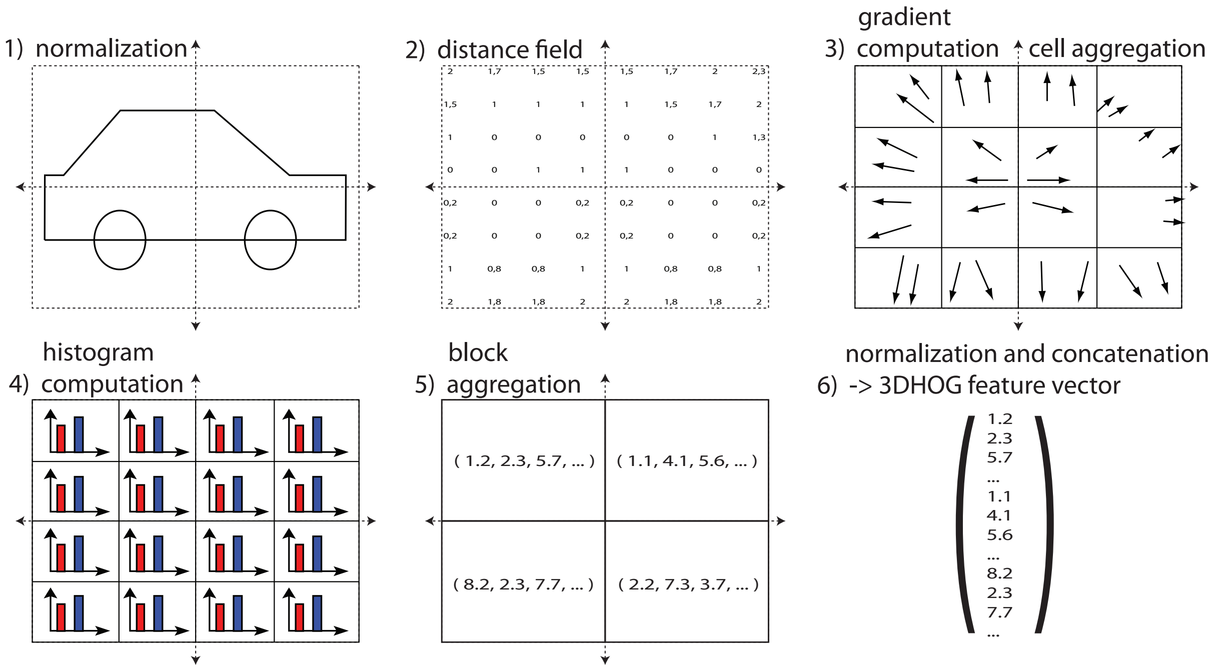
We extracted 3DHOG descriptors for all benchmark data sets using our implementation described in Section 3.4. The 3DHOG parameterization we used and a discussion thereof is given in Section 5.

We use the following alternative 3D descriptors to evaluate 3DHOG against: A 438-dimensional Depth-Buffer descriptor (*DBD438*), a 300-dimensional Silhouette-based descriptor (*SIL300*), and a 136-dimensional descriptor based on Radial Extent function (*RSH136*). Descriptor details can be found in [18]. We also consider a 472-dimensional descriptor which statically combines the aforementioned three descriptors (*DSR472*) [19]. Note that descriptors based on Depth Buffers to date are among the most effective 3D descriptors [5, 3], and that the hybrid DSR descriptor has been shown to substantially improve over the Depth Buffer descriptor [19].

4.2 Experimental Results

To confirm our intuition about the alternative 3DHOG gradient definitions from Section 3, we first extracted Precision-Recall-Curves for both variants of our descriptor. Comparison of the results obtained by first order gradients and second order gradients indicated significantly better results using second order gradients. Subsequently in all following experiments, second order gradients were used. Figure 5 shows Precision-Recall-Curves of the 3DHOG descriptor compared to the DBD, SIL, RSH, and DSR descriptors. As expected, the combined descriptor DSR consistently outperforms the other descriptors on all benchmarks. Our 3DHOG descriptor performs quite well compared to the single descriptors. While it yields similar precision results on the KN-DB benchmark as the RSH descriptor does, on the PSB benchmark the RSH de-

[1] http://www.gris.informatik.tu-darmstadt.de/projects/vsa/3dhog/3dhog.zip



2	1,7	1,5	1,5	1,5	1,7	2	2,3
1,5	1	1	1	1	1,5	1,7	2
1	0	0	0	0	0	1	1,3
0	0	1	1	1	0	0	0
0,2	0	0	0,2	0,2	0	0	0,2
0,2	0	0	0,2	0,2	0	0	0,2
1	0,8	0,8	1	1	0,8	0,8	1
2	1,8	1,8	2	2	1,8	1,8	2

Figure 2: The major processing steps of the 3DHOG feature vector extraction pipeline

Benchmark	Descriptor	NN	FT	ST	E	DCG
KN-DB	DSR472	0.70	0.42	0.54	0.29	0.69
	DBD438	0.61	0.34	0.43	0.23	0.60
	SIL300	0.58	0.31	0.41	0.22	0.59
	RSH136	0.56	0.28	0.37	0.20	0.55
	3DHOG	0.59	0.29	0.36	0.20	0.55
PSB-Test	DSR472	0.65	0.40	0.51	0.29	0.66
	DBD438	0.60	0.33	0.42	0.24	0.59
	SIL300	0.54	0.30	0.40	0.23	0.58
	RSH136	0.51	0.25	0.34	0.20	0.53
	3DHOG	0.58	0.27	0.35	0.21	0.55
SHREC	DSR472	0.82	0.50	0.64	0.43	0.78
	DBD438	0.72	0.39	0.50	0.34	0.70
	SIL300	0.71	0.39	0.53	0.36	0.70
	RSH136	0.68	0.35	0.47	0.32	0.67
	3DHOG	0.75	0.41	0.52	0.35	0.71

Table 1: Scalar retrieval precision measures for the descriptors on the benchmarks.

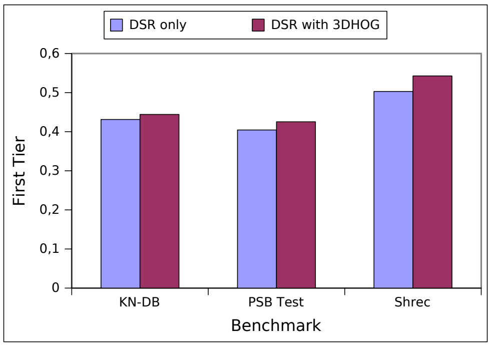


Figure 3: Precision on first tier of DSR (standalone) and in combination with 3DHOG. Better retrieval performance indicates that 3DHOG captures complementary, descriptive 3D features.

scriptor is outperformed by our descriptor. On the SHREC benchmark, 3DHOG outperforms all the individual descriptors (RSH, SIL, DBD). Table 1 gives the scalar precision metrics for the descriptors on the benchmarks, supporting this observation.

We also compared the retrieval precision of 3DHOG to its strongest competitor descriptor in our setting (DSR472), considering not per-benchmark average but per-class average metrics. Figure 4 shows the results for the SHREC benchmark. The results on the other benchmarks are similar. The chart shows that there exist classes of objects in the benchmark which the 3DHOG descriptor manages to retrieve more effectively than the DSR. This indicates that for certain query classes, 3DHOG is among the best 3D descriptors in our setting.

Finally, experiments were carried out combining the DSR and the 3DHOG descriptor in a hybrid descriptor. For combination, a static distance-based aggregation was used after distance normalization based on the variance as described in [17]. Figure 3 compares the precision on the first tier metric of the DSR472 descriptor alone, and in combination with the 3DHOG descriptor on all three benchmarks. Note that the retrieval precision is consistently improved on all three benchmarks by including 3DHOG. Precision-Recall-Curves were extracted as well and support this point (Figure 6). Since the DSR descriptor is already a combination of three single descriptors, these results suggest that the 3DHOG descriptor is able to capture meaningful, complementary 3D features, not captured by the other three descriptors. Clearly, 3DHOG is a valuable contribution to the set of candidate 3D descriptors, as by combination of feature vectors, we are able to improve the precision of 3D retrieval systems. Note that improvements in retrieval precision usually cannot be obtained only by upscaling system hardware, but foremost require methodological improvements. This work contributes toward this end.

5. DISCUSSION

The retrieval results presented in the previous section demonstrate the applicability of the 3DHOG. The main point for discussion, as well as future work in this context, is the definition and computation of *gradients*.

We experimentally found that using second order derivatives improves over the use of first order derivatives. We attribute this to the fact that for effective 3D description, not only information about local extrema near the mesh surface, but also information about extrema inside the mesh are useful. The second order derivate reveals precisely such extrema by high values and sets monotonical regions to zero. This could also explain why this computation of gradients yielded superior results to using, for example, surface normals as gradients, which was also evaluated but not presented. Those only assign high values near the mesh surface, but always zero in and around the mesh (as they are not defined there).

Apart from the gradient definition, the choice of parameters has a significant impact on the performance of the 3DHOG method. The first parameter to set is the number of voxels in the distance field by choosing the edge-length $r_{x,y,z}$ of each voxel. During this discretization process we loose less information if we choose a small edge-length – naturally this induces a higher computational time. Our implementation uses a straight-forward algorithm to compute the distance field. Optimization is necessary to compute high-resolution distance fields efficiently.

The next parameter is the cell-size $c_{x,y,z}$, which determines how many gradients – computed from the distance field – are aggregated in one histogram. Thereby we can choose the *degree of locality* the descriptor exhibits.

The histograms themselves are defined by the number of bins they provide for the gradient's zenithal and azimuthal orientation $(\theta, \phi)^T$(see eq. 1). This introduces the usual trade-off between accuracy and stability.

The last major parameters we have to choose are the size $b_{x,y,z}$ of the blocks as well as their overlap $o_{x,y,z}$. These determine how many neighboring cells are normalized with each other.

For our evaluation good first parameter-space guesses based on the results by Dalal and Triggs [6] were used and some further exploration was carried out. The parameters yielding the best descriptor we found and its dimensionality can be found in table 2. Surprisingly, the block overlap did not have the positive influence it had in the experiments of Dalal and Triggs [6], so it was set to 0.

Parameter	Value
$r_{x,y,z}$	$\frac{2}{52}$
$c_{x,y,z}$	12 vxl
bins(θ)	9
bins(ϕ)	9
$b_{x,y,z}$	2 cells
$o_{x,y,z}$	0 cells
dimensionality	5184

Table 2: Parameter settings used to extract 3DHOG descriptors

We are aware of two efficiency implications of 3DHOG: High dimensionality of the feature vector, and the expensive computation of the distance field in our simple implementation. Dimensionality could be tackled by subsequently applying a dimensionality reduction technique. E.g., in [18] PCA was indicated to improve retrieval efficiency without significantly reducing retrieval effectiveness. Accordingly, appropriate application of PCA can even improve effectiveness due to noise filtering effects. Efficient computation of distance fields is already well-studied and subject to current research [11, 15]. consequently, accelerating 3DHOG feature vector extraction is possible by using more advanced algorithms.

6. CONLUSION & FUTURE WORK

In this paper we derived and implemented a new 3D descriptor (3DHOG) based on the HOG feature vector, which is well-known in 2D image processing. The main challenge of defining meaningful gradients was looked at in detail. One concept to implement 3DHOG was presented in general – an actual open-source implementation is made publicly available. The performance of the descriptor was evaluated against established retrieval benchmarks and results indicated competitive retrieval performance. The descriptor captures 3D object information complementary to those of existing edescriptors, and in combination with them manages to boost retrieval performance.

The concept of using HOG for 3D model retrieval seems promising, and we like to further explore the parameter space and alternative gradient definitions. Preliminary experiments by defining gradients as the local curvature of the mesh's surface were conducted. First results of this approach could not match the original implementation. This can be explained by the fact, that curvature is only defined on the mesh surface. Accordingly combination with second-order derivate distance fields could be the key to further enhancement: Using curvature to describe the mesh near the surface, and using the distance derivatives to describe the mesh's interior.

As mentioned in related work (Section 2), particular similarities of the challenges faced can be drawn with the work by Glomb[8], who extends the Harris-operator to 3D meshes. The methods proposed by Glomb include *gaussian functions*, *quadric surfaces*, *Haussdorff distance* and *fitted surface distances*. All of these methods differ drastically from our approach, as they require the definition of a set of neighbors for each vertex. His findings suggest that fitting quadric surfaces is more suitable than the other extensions. Accordingly, we plan to adapt this approach for use with 3DHOG. Recalling the algorithm for feature computation as described in Section 3.4, we only have to substitute gradient computation on the distance field by the actual derivative of the fitted quadric at the specified position. The retrieval performance of the 3DHOG using gradients defined this way will also be evaluated.

Additional drawbacks of *exact* Eucleadian distance transforms occur, because the distance function is not differentiable at points that are equidistant from the mesh surface [2]. This may lead to undesirable outliers when convolving the distance field to approximate second order derivatives. Accordingly, we will examine the use of *smooth distance transforms* [2], which sacrifice a certain amount of accuracy for differentiability at each point.

ACKNOWLEDGEMENTS

We thank Bernt Schiele and the Multimodal Interactive Systems group for inspiring discussions on the HOG method and options for porting it to 3D. Furthermore, we thank colleagues in the Interactive-Graphics Systems Group and in Fraunhofer IGD A2. We thank Christian Wojek for providing a HOG implementation for images; Thomas Kalbe for supplying a render-engine for distance fields; and Matthias Bein for providing source code for curvature computation.

This work was partially supported by the German Research Foundation DFG within the LIS Leistungszentrum PROBADO under grant INST 9055/1-1.

REFERENCES

[1] Mihael Ankerst, Gabi Kastenmüller, Hans-Peter Kriegel, and Thomas Seidl. 3d shape histograms for similarity search and classification. In *Spatial Database*, page 207–226. Springer, 1999.

[2] Arpan Biswas and Vadim Shapiro. Approximate distance fields with non-vanishing gradients. *Graph. Models*, 66(3):133–159, 2004.

[3] B. Bustos, D. Keim, D. Saupe, T. Schreck, and D. Vranic. An experimental effectiveness comparison of methods for 3D similarity search. *International Journal on Digital Libraries, Special Issue on Multimedia Contents and Management*, 6(1):39–54, 2006.

[4] H.-S. Wong D. Wang, J. Zhang and Y. Li. 3d model retrieval based on multishell extended gaussian image. In *VISUAL*, page 426–437, 2007.

[5] Y. T. Shen D. Y. Chen, X. P. Tian and M. Ouhyoung. On visual similarity based 3d model retrieval. In *Proc. of Eurographics Workshop*, volume 22, pages 223–232, 2003.

[6] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR (1)*, pages 886–893, 2005.

[7] R. Fabbri, L. Da F. Costa, J. C. Torelli, and O. M. Bruno. 2d euclidean distance transform algorithms: A comparative survey. *ACM Comput. Surv.*, 40(1):1–44, 2008.

[8] Przemysław Głomb. Detection of interest points on 3d data: Extending the harris operator. In *Computer Recognition Systems 3*, volume 57 of *Advances in Soft Computing*, pages 103–111. Springer Berlin / Heidelberg, 2009.

[9] C. Harris and M. Stephens. A combined corner and edge detection. In *Proceedings of The Fourth Alvey Vision Conference*, pages 147–151, 1988.

[10] S. Jayanti, Y. Kalyanaraman, N. Iyer, and K. Ramani. Developing an engineering shape benchmark for cad models. *Computer-Aided Design*, 38(9):939–953, 2006.

[11] Mark W. Jones, J. Andreas B?rentzen, and Milos Sramek. 3d distance fields: A survey of techniques and applications. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):581–599, 2006.

[12] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[13] NIST. Shape Retrieval Contest (SHREC), New Generic Shape Benchmark Dataset. http://www.itl.nist.gov/iad/vug/sharp/benchmark/shrecGeneric/, 2009.

[14] Philip Shilane, Patrick Min, Michael Kazhdan, and Thomas Funkhouser. The princeton shape benchmark. *Shape Modeling and Applications, International Conference on*, 0:167–178, 2004.

[15] Avneesh Sud, Miguel A. Otaduy, and Dinesh Manocha. Difi: Fast 3d distance field computation using graphics hardware. *Comput. Graph. Forum*, 23(3):557–566, 2004.

[16] Johan W. H. Tangelder and Remco C. Veltkamp. A survey of content based 3d shape retrieval methods. *Multimedia Tools Appl.*, 39(3):441–471, 2008.

[17] Tobias Schreck. *Effective Retrieval and Visual Analysis in Multimedia Databases*. PhD thesis, Universität Konstanz, Universitätsstr. 10, 78457 Konstanz, 2007.

[18] D.V. Vranić. *3D Model Retrieval*. PhD thesis, University of Leipzig, German, 2004.

[19] Dejan V. Vranic. Desire: a composite 3d-shape descriptor. In *ICME*, pages 962–965, 2005.

[20] R. Wessel, I. Blümel, and R. Klein. A 3d shape benchmark for retrieval and automatic classification of architectural data. In *Eurographics 2009 Workshop on 3D Object Retrieval*, pages 53–56, 2009.

[21] T. Zaharia and F. J. Preteux. 3d-shape-based retrieval within the mpeg-7 framework. In *Society of Photo-Optical Instrumentation Engineers Conference Series*, page 133–145, 2001.

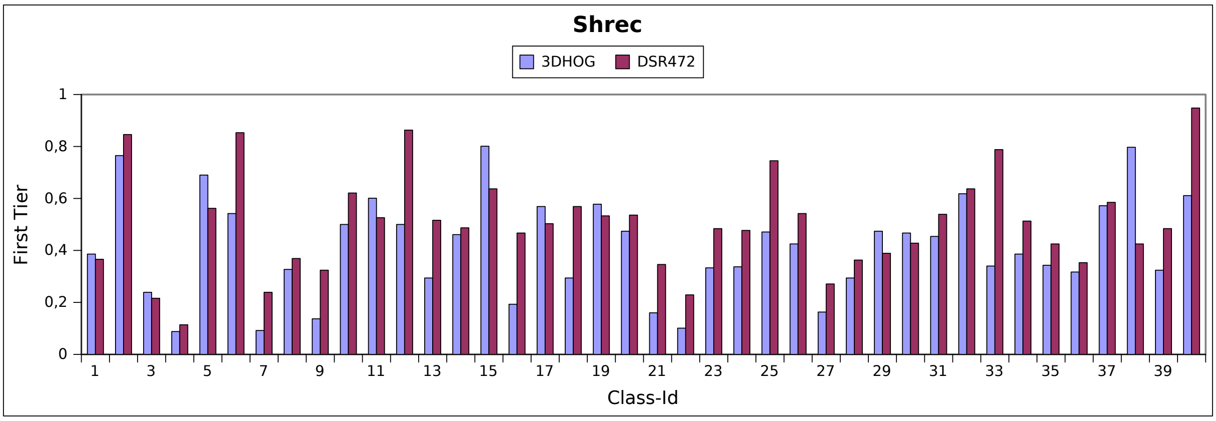


Figure 4: Precision on first tier of individual classes of the SHREC benchmark. In several query classes (e.g., 11, 15, 38), 3DHOG outperforms one of the best hybrid 3D descriptors (DSR).

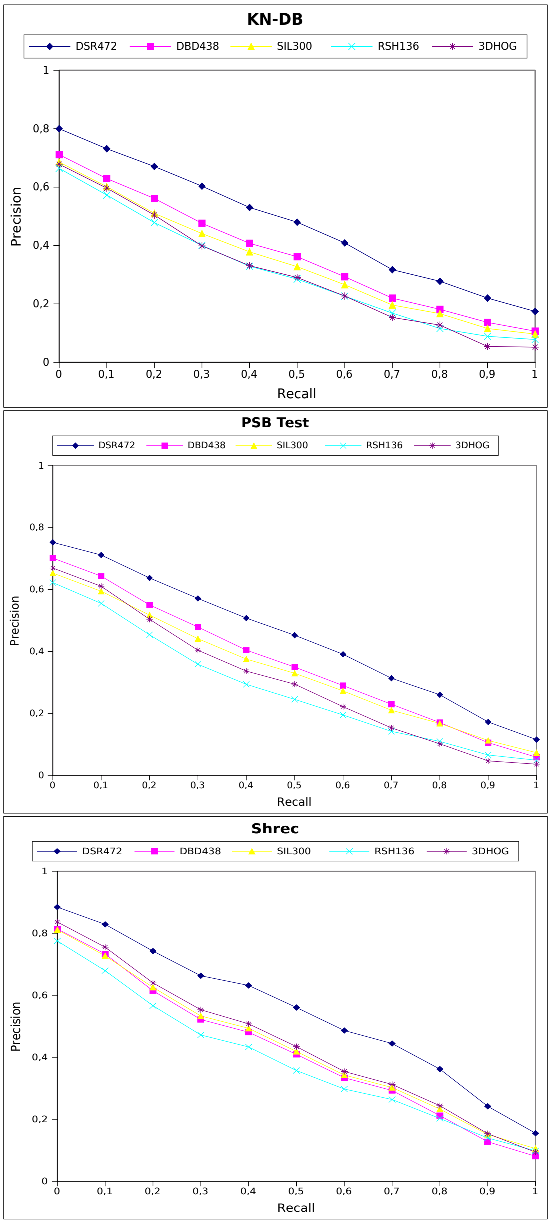


Figure 5: Precision-Recall-Curves on the benchmarks.

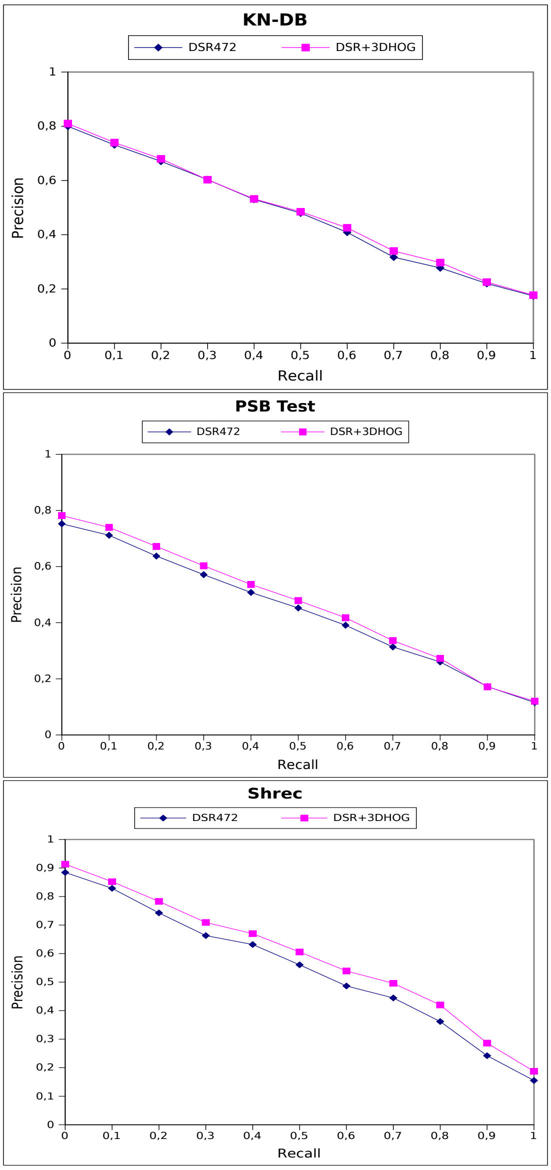


Figure 6: Precision-Recall-Curves of DSR (standalone) and in combination with 3DHOG on the benchmarks.