Robust 3-D Object Skeletonisation for the Similarity Measure

Christian Feinen[1], David Barnowsky[2], Dietrich Paulus[2] and Marcin Grzegorzek[1]
[1]*Research Group for Pattern Recognition, University of Siegen, Hoelderlinstr. 3, 57076 Siegen, Germany*
[2]*Research Group for Active Vision, University of Koblenz-Landau, Universitaetsstr. 1, 56070 Koblenz, Germany*

Keywords: 3-D Skeleton Extraction, 3-D Curve Skeletons, 3-D Object Retrieval, 3-D Acquisition and Processing, Human Perception and Cognition.

Abstract: In this paper we introduce our approach for similarity measure of 3-D objects based on an existing curve skeletonization technique. This skeletonization algorithm for 3-D objects delivers skeletons thicker than 1 voxel. This makes an efficient distance or similarity measure impossible. To overcome this drawback, we use a significantly extended skeletonization algorithm (by Reniers) and a modified Dijkstra approach. In addition to that, we propose features that are directly extracted from the resulting skeletal structures. To evaluate our system, we created a ground truth of 3-D objects and their similarities estimated by humans. The automatic similarity results achieved by our system were evaluated against this ground truth in terms of precision and recall in an object retrieval setup.

1 INTRODUCTION

Automatic similarity measure of objects is important for many object recognition systems. One of the most important similarity features considered by humans is shape (Pizlo, 2008). Powerful and promising shape descriptors are skeletons. While skeletonization approaches in 2-D provide satisfying results (Chang, 2007; Bai and Latecki, 2008), their extensions to 3-D are lacking of robustness and accuracy. Additionally, 3-D skeletons underlie several constraints: (i) Only cylindrical geometry delivers curve skeletons; cubes result in so-called surface skeletons. (ii) The skeleton extraction process includes complex mathematical instruments. (iii) Skeletonization methods require a high computational effort. Additionally, a plenty of 3-D approaches for skeletonization deliver skeletal structures that are wider than one voxel, e.g., (Reniers, 2008). However, skeletons are capable to reduce the dimensionality of an object while preserving important shape characteristics in terms of geometry and topology.

Although, a plethora of research was done in recent decades, the major amount was dedicated to obtain skeletons and only a little in using them in object recognition systems. In this paper we try to close the gap between extraction and use of skeletons as a similarity measure. *The significant worth of our work is the constitution of a complete processing pipeline.* Therefore, we employ a skeletonization algorithm proposed by Reniers (Reniers, 2008). Given the fact that this algorithm does not generate perfect skeletons, we were forced to extend the Dijkstra algorithm to estimate both, the shortest paths between skeleton endpoints as well as the location of junction points within the skeleton. We are aware of the presence of already existing skeleton extraction methods that guarantee one voxel wide skeletons, however, Reniers' method produces skeletons with a stable topology analogical to the observed object. This property has the highest priority for our proposed method, since we decided to incorporate only topological information. These features are directly obtained from the skeletal structure. From a practical point of view, this decision is reasonable because skeletons are particularly capable to encode the topological structure of an object in a highly efficient way. To evaluate our system, a ground truth of 3-D objects is used whose similarities are estimated by humans.

Our paper is structured as follows. We start by discussing relevant existing skeletonization algorithms for 2-D and 3-D objects (Section 2). Afterwards, Section 3 introduces Reniers' skeletonization method as well as the improvements made to it within our own approach. Subsequently, Section 4 describes the proposed similarity measure. In Section 5, we quantitatively evaluate our system based on the ground truth qualified by humans. Finally, Section 6 concludes our work and grants a deeper look into upcoming future tasks.

Feinen C., Barnowsky D., Paulus D. and Grzegorzek M. (2013).
Robust 3-D Object Skeletonisation for the Similarity Measure.
In *Proceedings of the 2nd International Conference on Pattern Recognition Applications and Methods*, pages 167-175
DOI: 10.5220/0004255601670175

2 RELATED WORK

In the late 1960 Harry Blum introduced an initial approach for describing objects by skeletons (Blum, 1967) and the well known *medial axis (transform)* has been proposed. In subsequent years further methods have been developed in order to extract skeletons primarily in 2-D. Many of them are mapped into 3-D, but there are still problems. One of the most popular methods are *Thinning* algorithms. This is one of the most frequently used techniques to generate curve skeletons of 3-D objects. The idea is to remove iteratively the object's surface (3-D) (or boundary (2-D)) (Palágyi and Kuba, 1998; Palágyi and Kuba, 1999; Ma and Sonka, 1996). *Voronoi* algorithms are also a popular representative in this domain. The resulting skeletons are a subset of voronoi edges (Ogniewicz and Ilg, 1992). Main drawback associated to this type of algorithms is the high computational complexity. However, the basic idea was successfully mapped into 3-D space (Ogniewicz and Kübler, 1995). 3-D Skeletons can also be extracted by the so-called *distance transform* methods. We also use this transform in our system. During the execution of such a method a distance map is generated that stores the distances of each location to the closest point on the boundary or surface (Fabbri et al., 2008; Borgefors, 1996; Reniers, 2008). Consequently, all points on the boundary have a zero value. Analogical to the *grass fire flow* assumption, this approach implies that skeletal axes are located at singularities within the distance map. "Singularities are the points where the field is non-differentiable. When the distance field is seen as a height map, the singularities can be seen as the mountain ridges and peaks" (Reniers, 2008). Other techniques employ *deformable models* (Sharf et al., 2007) or *level sets* (Hassouna and Farag, 2007) to approximate the surface of an object in order to estimate curve skeletons. Input of these methods are point clouds or surface meshes. The output is a thin and connected 3-D curve skeleton that constitutes a desired structure in many skeleton-based algorithms. A further approach which is working on the surface mesh uses a so-called *laplacian-based contraction* to contract the mesh iteratively until it converges against its centerline (Cao et al., 2010). A good introduction to the most popular methods is given by the authors K. Siddiqi and S. Pilzer in (Siddiqi and Pizer, 2008).

There is plethora of different approaches related to object recognition, but most of them are working in 2-D. Only less are able to perform directly in 3-D by using skeletons for a similarity measure. Moreover, talking about 3-D skeleton-based retrieval methods generally includes both, *view-based* and *model-based* approaches. Since the proposed method belongs to the latter, only such algorithms are discussed in the following. In order to retrieve some information about view-based approaches, the work presented in (Macrini et al., 2002) should be mentioned here. The authors propose a method where the 3-D model is represented by a collection of 2-D views. These views are taken for object recognition using shock graphs. In (Bai and Latecki, 2008) a 2-D *Path Similarity Skeleton Graph Matching* approach is presented which is mapped into 3-D by the authors of (Schäfer, 2011). Further techniques based on 3-D skeletal representations are introduced in (Cornea et al., 2005) and (Sundar et al., 2003). As in our work, the authors of (Cornea et al., 2005) are using the distance transform to generate skeletons. However, in contrast to our approach, they are using the Earth Mover's Distance (EMD) to compute a dissimilarity value. Moreover, such an approach is not suitable for our work due to the fact that our skeletons are wider than one voxel.

Apart form this, other competing methods in this area are based on so-called *surface skeletons*. In (Hayashi et al., 2011), e.g., a 3-D shape similarity measurement technique is proposed which uses surface skeletons of voxelized 3-D shapes. This method is similar to our approach, but the use of surface skeletons can be ambiguous in case of objects consisting of simple geometry parts and this, in turn, affects the accuracy and robustness of the system. In (Zhang et al., 2005) a further method to perform 3-D object retrieval by using surface skeletons is presented. In this work, the authors place the major focus on the representation of symmetries of 3-D objects, especially in context of articulated 3-D models.

3 SKELETONIZATION

Reniers used a distance transform to skeletonize 3-D objects in order to perform segmentation purposes (Reniers, 2008). These skeletons are wider than one voxel which, in turn, does not affect his segmentation approach. In our case, this exacerbates an efficient computation of distinctive skeleton properties. One-voxel wide skeletons provide attractive computational advantages to estimate feature points, e.g., skeletal junction points. However, the objective of our approach is to be capable to process these non-perfect, n voxel wide skeletons (with $n \in \mathbb{N}$ and $n > 1$) in order to guarantee stable and accurate topological information. Of course, this includes a thinning procedure somehow, the detection of end- and junction points, the estimation of so-called *nodal areas*, the calcula-

tions of geodesics as well as the computation of feature values. All these points are subject in the Sections 3.2 - 3.4, whereas Section 3.1 provides a brief introduction to Reniers' work.

3.1 Skeleton Extraction According to Reniers' Distance Transform

Reniers' skeleton extraction method is based on a distance transform. It consists of three steps, namely (i) the computation of the so-called *Tolerance-based Feature Transform (TFT)*, (ii) the computation of the geodesic paths, and (iii) the actual computation of the skeleton. For each point **p** inside the 3-D object, the TFT calculates feature points lying on the object's surface. The feature points are computed based on an adjustable tolerance basis as shown in Figure 1. The TFT was developed in order to handle problems

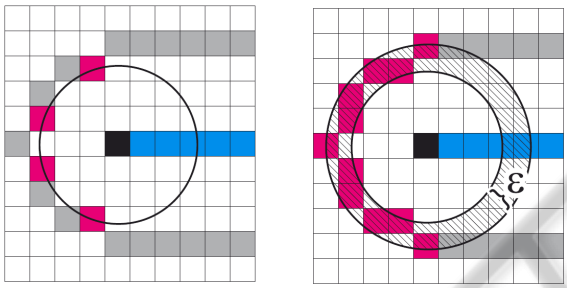


Figure 1: Working Principle of TFT. The figure shows the advantage of using an adjustable tolerance basis (ε) in context of a discretized space. From the top-view of an object it is clearly noticeable that much more feature points are detectable by using the TFT.

which occur when the input data has a discretized structure (volumetric data). As shown in Fig. 2(a), it is not possible to find at least two feature points *on the contour* whose lengths are identical to an arbitrary interior point (black square) inside the rectangle. This phenomenon is caused by the fact that the number of voxels is even. In addition to this, it is obvious that this phenomenon inhibits the estimation of at least two geodesics between such contour feature points. Fig. 2(b) shows one of the two possible geodesics on the contour, please note that the geodesic path to the opposite side would increase the number of voxels by two. These problems are avoidable by adding a new parameter to adjust a tolerance threshold. Thus, we are able to find at least two geodesics as shown in Fig. 2(c). Moreover, the detection of at least two geodesic paths grants us the possibility to detect so-called *Jordan curves* and this, in turn, enables us to locate the desired skeleton points. The actual estimation of geodesics between all feature points is performed by using a *Dijkstra* algorithm. Unfortunately, this calculation is the most time consuming step within Reniers' algorithm. The computation of the tolerance-based feature transform is not trivial, please refer (Reniers and Telea, 2006) to obtain detailed information about it. According to the formal definition, a point **p** belongs to the skeleton set, if condition (1) is satisfied. Here, Γ indicates the set of all geodesics between the feature points. More precisely, a point **p** is part of the skeletal structure, if it holds a Jordan curve.

$$\mathbf{p} \in S \Leftrightarrow |\Gamma| \geq 2 \tag{1}$$

The concrete estimation of Jordan curves are a bit more complicated as described here. Please refer (Reniers, 2008) for a detail description. The result skeleton has several drawbacks related to our purpose: (i) The skeleton is wider than one-voxel. (ii) Junction points are not exactly defined due to the use of the TFT and (iii) the distances between junction and endpoints are not uniquely calculable. The Listing below illustrates the explanations above based on some pseudo code lines in order to grants a deeper understanding of this technique.

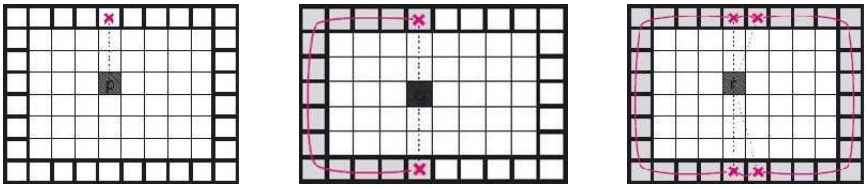
```
1:  compute feature transform F on Ω
2:  for each object voxel p ∈ Ω do
3:      F̄ ← ⋃_{x,y,z∈{0,1}} F(p_x + x, p_y + y, p_z + z)
4:      Γ ← ⋃_{a≠b∈F̄} shortestpath(a, b)
5:      if Γ contains a Jordan curve then
6:          S ← S ∪ {p⃗}
7:      else
8:          surface skeleton or non-skeleton voxel
9:      end if
10: end for
```

Line 1-3 compute the TFT for each point **p** inside the object Ω. Therefore, **line 1** processes an ordinary feature transform and **line 3** stores the set of all feature points $\overline{F}$ obtained by the TFT according to the tolerance values (here: $x,y,z \in \{0,1\}$).

Line 4 computes the set of geodesics Γ. The geodesic paths are calculated between all features points which are stored in $\overline{F}$.

Line 5-9 investigate the set of geodesics Γ generated above in order to detect possible Jordan curves. These curves can be generated by at least two geodesic paths. If this is the case, the currently observed point **p** is moved to the set of skeleton points (Eq. 2, where S indicates the set of skeleton points).

$$S \leftarrow S \cup \mathbf{p} \tag{2}$$



(a) (b) (c)

Figure 2: Problems occurring by using a discretized space. 2(a) shows the problematic to find the center line of the rectangle due to the even number of interior points. Connected to this problem, 2(b) figures out that the estimation of geodesics is inhibited as well. 2(c) illustrates the special handling of this problem by using the TFT.

3.2 Skeleton Feature Set

With respect to the problems stated in the previous section, some modifications of the skeletal structure are necessary in order to derive an adequate feature set. These modifications are subject of Sec. 3.3 and 3.4. However, in order to modify the skeletons generated by the algorithm stated above, considerations about an adequate similarity measure have to be made initially. This means, the feature set has to be highly compatible regarding both, discrimination power and consistency to structure changes. Consequently, this vector strongly correlates to the modifications of the skeleton structure. Thus, we added the following three constraints to our deliberations. First, we only incorporate topological information which are directly obtainable from the skeletal structure without considering geometrical information (which could be derived from the 3-D object as well). Actually, this aim makes it very challenging to discriminate our objects. However, it perfectly figures out the strength of using skeletons to describe articulated objects. It also demonstrates the robustness of our modifications made to the skeletons. Second, the features have to be invariant to rotation, translation and scaling which is an obvious and significant advantage for comparing objects in 3-D. Third, the computation of our feature set should be fast in order to speedup the proposed method. That makes it practical for many applications.

This section introduces the proposed feature vector. As already mentioned, the feature set is topologically oriented and includes five values (Eq. 3).

$$\mathbf{f}_v = (m_1, m_2, m_3, m_4, m_5)^T \tag{3}$$

Feature 1 indicates the number (m_1) of end points $\mathbf{e}_0, \mathbf{e}_1, ..., \mathbf{e}_n$. This descriptor reflects the complexity of the object.

Feature 2 is the number (m_2) of junction points $\mathbf{n}_0, \mathbf{n}_1, ..., \mathbf{n}_n$. This descriptor extends the complexity value (Feature 1).

Feature 3 describes the mean number of outgoing paths of a junction point. This can be regarded as a distribution factor between all end- and junction points. The higher the value the more endpoints connect to the same junction point on average. The function $c(\mathbf{n}_i, \mathbf{e}_j)$ is a binary function. It returns the value one, if there is a connection between the endpoint $\mathbf{e}_j$ and the junction point $\mathbf{n}$.

$$m_3 = \frac{\sum_{i=1}^{m_2} \sum_{j=1}^{m_1} c(\mathbf{n}_i, \mathbf{e}_j)}{m_2} \tag{4}$$

Feature 4 determines the mean distance of all skeletal paths (between endpoints). This value encodes the lengths of all object segments to one value.

$$m_4 = \frac{\sum_{i=1}^{m_1} \sum_{j=i+1}^{m_1} d(\mathbf{e}_i, \mathbf{e}_j)}{m_1/2 * (m_1 - 1)} \tag{5}$$

Feature 5 represents the variance of all shortest path lengths. This value indicates the regularity of lengths inside the different object parts.

$$m_5 = \frac{\sum_{i=1}^{m_1} \sum_{j=i+1}^{m_1} (d(\mathbf{e}_i, \mathbf{e}_j) - m_4)^2}{m_1/2 * (m_1 - 1)} \tag{6}$$

3.3 Preprocessing

In order to obtain a feature vector as introduced in Sec. 3.2 minimal demands on end- and junction points have to be satisfied. These requirements concern both, outlier removal and consistency of the skeleton structure. This became obvious during our work, as we observed some instability caused by outliers as well as by problems regarding the skeleton structure (stated in Sec. 3.1). Hence, one major point is the modification of Reniers' skeletons with respect to the skeletal structure. This issue is debated in Sec. 3.4. In this section we briefly discuss the outlier removal which constitutes only a small part within our

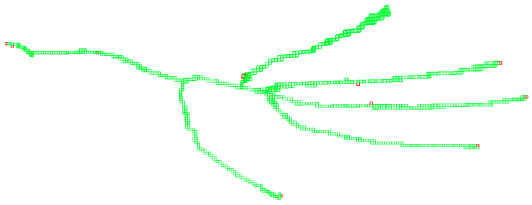


Figure 3: The figure shows a skeleton after the preprocessing step. The preprocessing is based on morphological operations. In this case a $3 \times 3 \times 3$ cube is used to perform a dilatation followed by an erosion.

whole processing pipeline. Therefore, we employ well-known state-of-the-art methods form the image processing domain, e.g., thinning and morphological operators. Since the skeleton extraction method is time consuming, we tried to employ computationally attractive techniques to speedup the whole processing pipeline. tasks. In order to evaluate each method, we directly visualized their results with *OpenGL*. This was not much effort due to the fact our the dataset is easily manageable. Thereby, we observed that the best results are achieved with morphological operators even we are running the risk that they do not compulsorily preserve the skeleton's topology structure. In our case a dilatation followed by an erosion is performed. For both operations, we use a structure element (cube) of size $3 \times 3 \times 3$. The topological changes caused by this combination of structure and structure element are not able to influence our method negatively. They also outperformed thinning algorithms. In this way 60% to 70% of all wrong voxels could be eliminated on average. The remaining 30% are uncritical since our algorithm is able to handle such an amount of outliers. Figure 3 shows a skeletons after the preprocessing step.

3.4 Modified Version of the Dijkstra Algorithm

This section describes our final approach. As stated above, there are still outlier voxels. They do not affect the proposed method negatively since our algorithm will remove them as well during its execution. In addition to this, it detects the skeleton junction points and computes the values of our feature vector. Therefore, we propose the *Dijkstra-Skeleton algorithm (DSA)* as a modified version of the well-known Dijkstra algorithm. Since the DSA consists of multiple steps, we will provide an overview about its operating principle below. Summarized, we benefit from two significant advantages by using this new version. First, we are able to compute our feature vector. Second, as already stated, we get rid of the remaining wrong endpoints.

1. The skeleton structure is mapped to a continuous graph. This means all voxels are transformed into an adjacency matrix. This matrix stores the distances between these nodes.
2. Expansion of skeleton nodes to *skeleton nodal areas (SNA)*. SNAs are sets of points adjacent to skeleton junction points which are thicker than one voxel inside the skeleton. They are necessary to prevent miscalculations of shortest paths. This can happen, if the original skeleton junction point is wider than one voxel. In this case the junction point runs the risk of being skipped and this, in turn, would lead to a wrong result. This is highly crucial because the first SNA has to be assigned to each shortest path that is touched by it. In order to guarantee this assignment all SNAs have to be detected. Therefore, we utilize a $3 \times 3 \times 3$ structure element.
3. Computation of all shortest paths and storage of the first SNA that is passed during the computation.
4. Validity check of Dijkstra paths based on two exclusion criteria.
   - An endpoint is located between two SNAs.
   - Multiple endpoints are connected to the same SNA and the length of the currently observed endpoint is not the maximum.

   Fig. 4 demonstrates both cases. According to our exclusion criteria, endpoint $\mathbf{e}_2$ and $\mathbf{e}_3$ is removed from the graph.
5. Computation, normalization and storage of all remaining values of our feature vector introduced in Sec. 3.2.

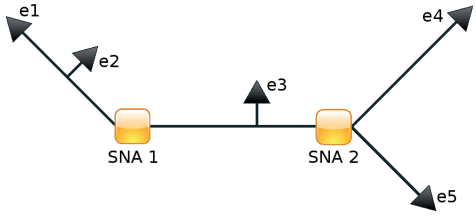


Figure 4: Conceptual representation of a skeleton graph. The black arrows $e_1, e_2, \ldots, e_5$ indicate the skeleton endpoints and the yellow rectangles *SNA*1 and *SNA*2 are *skeleton nodal areas*. By executing the modified Dijkstra algorithm the endpoints e_2 and e_3 are removed.

4 SIMILARITY MEASURE

This section briefly describes the selected similarity measure that is used to compare two 3-D objects. In

this work, we employ the well-known *cosine angle* as defined in (7). Since we are able to arrange our features in a vector, this similarity measure, or more precisely the distance, is highly suitable for our purposes. A further benefit is its normalization power regarding the vector lengths. In order compare two objects based on their skeletons, we compute the distance between their feature vectors.

$$sim_{\cos}(\mathbf{f}_n, \mathbf{f}_m) = \frac{\langle \mathbf{f}_n, \mathbf{f}_m \rangle}{\|\mathbf{f}_n\| * \|\mathbf{f}_m\|} \tag{7}$$

In addition to this, we rate the quality of a query with well-known indicators from the field of information retrieval, namely *recall* (completeness) and *precision* (accurateness). An small example is shown in Table 2. In order to be capable to compute these values, a ground truth (GT) has been generated.

4.1 Ground Truth

The GT consists of totally 14 objects. We know that this amount of objects is much to less in order to perform a comprehensive evaluation. Although there are some standardized databases consisting of more objects, they are not useful for our project in this first iteration. The reason for this is the fact that we want to investigate relations to the human perception. Therefore, we used a group of volunteers (consisting of 15 students from different research disciplines) and asked them to rate the the similarity of each object pair within our database. Even for a small database like ours, the similarity of 91 object pairs has to be rated by a single person. If we would split up this work, we will run the risk to jeopardize the GT's consistency. This, in turn, would be dramatically, because the GT a crucial factor to determine the number of true positives for every query. That means, the GT provides the basis to compute the so-called optimal hitting set. Unfortunately, the number of possible (true) hits is not constant in our case caused by the way of determining similarity groups. Summarized we were faced with several problems: First, the objects belonging to each class are not spread uniformly. Second, test persons were responsible to cluster our objects into "similarity groups" according to the human understanding of similarity. On the one hand, this data provides a high degree of interesting information which is discussed in Sec. 5. On the other hand, we had to ensure that we do not overload our volunteers with to many rating tasks. Third, the optimal hitting set was estimated by a heuristic as explained as follows:

1. For each object O_h all remaining GT objects are arranged in a descending order within a list according to their corresponding GT similarity values ($s(O_h)$).
2. Afterwards, we compute all differences of similarity values as shown in Eq. 8 where d_{O_v} indicates the similarity value of an object according to its list position (v).

$$\Delta(d_{O_v}, d_{O_{v+1}}) \tag{8}$$

3. Finally, we detect the position of the fourth largest delta value (Δ) and select all objects above this row as part of the optimal hitting set (H_G).

An example of the procedure is shown in Table 1. The decision to use the fourth largest difference as a threshold is based on empirical observations. Besides this, the actual rating of similarities, which has been performed by our volunteers, was unrestricted. Every test person was free to choose continuously different perspectives on the objects in order to rate their similarity.

Table 1: Example execution of the proposed heuristic to select whose objects that build an optimal hitting set H_G for an arbitrary query object O_h.

list	$s(O_h)$	$\Delta(d_{O_v}, d_{O_{v+1}})$	Δ-Position	$O_v \in H_G$
O_i	0.9	-	-	$\surd$
O_j	0.65	0.25	1	$\surd$
O_k	0.64	0.01	7	$\surd$
O_l	0.45	0.2	3	$\surd$
O_m	0.3	0.15	4	x
O_n	0.06	0.24	2	x
O_o	0.02	0.04	5	x
O_p	0.0	0.02	6	x

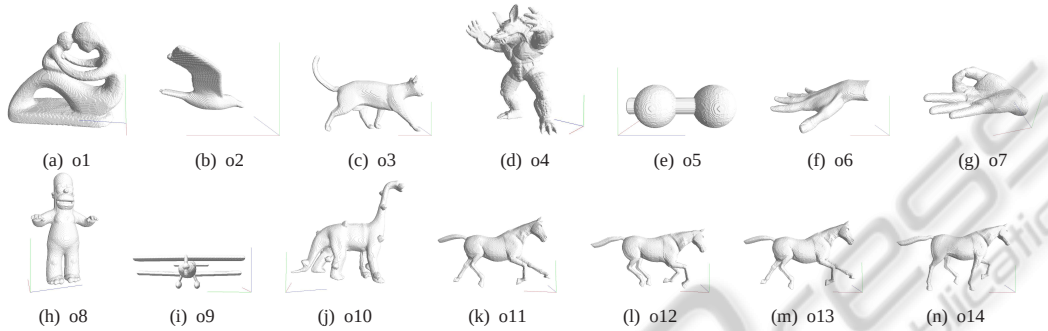
5 EVALUATION

This section presents the results of the evaluation based on the proposed method. The objects of our data set are shown in Fig. 5. Unfortunately, our database is limited to 14 objects. Nevertheless, these objects are quite challenging and our results are quite promising.

As already mentioned in Sec. 4.1, the GT similarity information was given by human volunteers. Admittedly, one of the human characteristics is the skill to decide similarity on fuzzy degrees and thus the human perception is not binary. However, our tests discovered two major things. (i) Expressing this vagueness in numbers constitutes a challenging job for humans and this (ii) makes it even harder to arrange these values subsequently in a consistent and uniform distributed way. Latter is also affected by the circumstance that each person scales this fuzziness

Table 2: Some easy example calculations in terms of *Recall* and *Precision*. All content elements are fictitious.

query object	optimal #hits (GT)	actual #hits (SM)	Recall	Precision
O_a	O_x, O_y, O_z	O_x, O_y, O_z	1	1
O_b	O_i, O_j, O_k, O_l	O_c, O_d	0	0
O_c	O_i, O_j, O_k, O_l	O_i, O_j	0.5	1
O_d	O_i, O_j	O_i, O_j, O_k, O_l	1	0.5
O_e	O_i, O_j, O_k, O_l	O_i, O_j, O_m, O_n	0.5	0.5



(a) o1 (b) o2 (c) o3 (d) o4 (e) o5 (f) o6 (g) o7

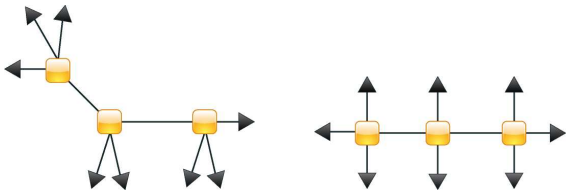
(h) o8 (i) o9 (j) o10 (k) o11 (l) o12 (m) o13 (n) o14

Figure 5: Overview about our object database consisting of 14 objects. As shown, most of the objects are articulating. Articulated objects are highly suitable to work with skeletons.

totally different. In order to handle this gap of perception and capability to express vagueness in values, we limited this information to two categories: *similar* and *not similar*. Consequently, we lost the possibility to employ a rational differentiation between these two quantities. As a result of this, we were forced to compare between two different scales:

1. an ordinal scaled range obtained by humans and
2. a continuous and metric interval-scaled range derived from our similarity measure.

Working with different scales is hard and excludes a direct comparison of both values. Thus, the whole evaluation has to be expressed in terms of *similar* or *not similar*. Additionally, we have to consider a very interesting and crucial phenomena which we call "unconscious background-knowledge". This means, humans refer to unconscious relationships between objects during the similarity rating. For example, it is not surprising that the object group consisting of Fig. 5(o11) to Fig. 5(o14) obtains a high similarity value. But in case of the objects of Fig.5(o5), 5(o6) and 5(o7) it is a bit surprising. However, with the knowledge in mind it is replicable. Unconscious knowledge guides persons to form factual connections. Considering this information, we evaluated our method. As already stated, we evaluated the results in terms of *recall (R)* and *precision (P)*. The evaluation results are shown in Table 4. Altogether, most of the results are quite promising and it can been assumed that our systems rates similarity of 3-D objects according to a certain degree of human perception. Nevertheless, it has to be remarked that the object group of horses leads to several outliers and that its result is not as accurate as expected. Although all horses are found by our algorithm, there are multiple wrong classified objects. The problem is associated with the decisions we made in respect of the proposed feature vector. Since we decided to consider only the topological information of the objects, the feature vector is not able to discriminate different objects with similar topology in an adequate way (Fig. 6). A further aspect in context of our feature vector relates to *feature 3*. This feature does not provide strong differences within the range of its values and consequently there is no additional information.



(a) Horse (b) Three-man kayak

Figure 6: Two synthetically generated skeletons. The first one could be the skeleton of a horse, the second could be a three-man kayak. As it is shown, the skeletons are apparently different, but in the case that only their topological structure is considered, the skeletons are not differentiable any more.

Table 3: Overview about the human similarity values between each combination of objects. To simplify the interpretation we encoded object pairs with a highly rated similarity red and object pairs with low similarity blue.

	o_1	o_2	o_3	o_4	o_5	o_6	o_7	o_8	o_9	o_{10}	o_{11}	o_{12}	o_{13}	o_{14}
o_1	-													
o_2	0,10	-												
o_3	0,12	0,20	-											
o_4	0,20	0,07	0,27	-										
o_5	0,12	0,20	0,13	0,05	-									
o_6	0,13	0,12	0,15	0,13	0,17	-								
o_7	0,13	0,11	0,12	0,10	0,17	0,90	-							
o_8	0,19	0,07	0,40	0,47	0,10	0,20	0,13	-						
o_9	0,08	0,46	0,07	0,07	0,09	0,07	0,10	0,06	-					
o_{10}	0,22	0,07	0,38	0,30	0,07	0,10	0,17	0,17	0,07	-				
o_{11}	0,13	0,10	0,61	0,23	0,07	0,07	0,10	0,19	0,07	0,60	-			
o_{12}	0,13	0,10	0,60	0,23	0,10	0,07	0,10	0,19	0,07	0,59	0,95	-		
o_{13}	0,12	0,10	0,63	0,23	0,10	0,07	0,10	0,19	0,07	0,59	0,95	0,92	-	
o_{14}	0,13	0,10	0,60	0,23	0,10	0,07	0,10	0,19	0,07	0,61	0,96	0,96	0,95	-

Table 4: On the left you can see the query object which corresponds to Fig. 5. In the second and third column both set of objects are arranged, the objects which are classified as similar by our test persons and the objects retrieved by our similarity measure. The last to columns show the corresponding Recall and Precision value.

query	optimal hitting set (GT)	actual hitting set (similarity measure)	Recall	Precision
O_1	O_2, O_3, O_4, O_5, O_6, O_8, O_{10}, O_{11}, O_{12}, O_{13}, O_{14}	O_3, O_4, O_8, O_9, O_{10}, O_{11}, O_{12}, O_{13}, O_{14}	0.73	0.89
O_2	O_3, O_5, O_6, O_7, O_9	O_1, O_3, O_4, O_5, O_6, O_7, O_9, O_{10}, O_{11}, O_{12}, O_{13}, O_{14}	1.00	0.42
O_3	O_{11}, O_{13}	O_7, O_9, O_{11}	0.50	0.33
O_4	O_1, O_3, O_8, O_{10}, O_{11}	O_1, O_8, O_9, O_{10}, O_{11}, O_{12}, O_{13}, O_{14}	0.80	0.50
O_5	O_6, O_7	O_1, O_2, O_3, O_4, O_7, O_8, O_9, O_{10}, O_{11}, O_{12}, O_{13}, O_{14}	0.50	0.08
O_6	O_1, O_2, O_3, O_5, O_7, O_8, O_{10}	O_3, O_7, O_{11}	0.29	0.67
O_7	O_1, O_2, O_5, O_6, O_8, O_9, O_{10}	O_3, O_6	0.14	0.50
O_8	O_1, O_3, O_4, O_6, O_{10}, O_{11}	O_1, O_4, O_9, O_{10}, O_{11}, O_{12}, O_{13}, O_{14}	0.67	0.50
O_9	O_2, O_5, O_7, O_8, O_{11}	O_1, O_4, O_8, O_{10}, O_{11}, O_{12}, O_{13}, O_{14}	0.40	0.25
O_{10}	O_{11}	O_1, O_4, O_8, O_9, O_{11}, O_{12}, O_{13}, O_{14}	1.00	0.13
O_{11}	O_{13}	O_1, O_3, O_4, O_9, O_{11}, O_{12}, O_{13}, O_{14}	1.00	0.13
O_{12}	O_{14}	O_1, O_4, O_8, O_9, O_{10}, O_{11}, O_{13}, O_{14}	1.00	0.13
O_{13}	O_{11}	O_1, O_4, O_8, O_9, O_{10}, O_{11}, O_{12}, O_{14}	1.00	0.13
O_{14}	O_{12}	O_1, O_3, O_4, O_8, O_9, O_10, O_{11}, O_{12}, O_{13}	1.00	0.11

6 CONCLUSIONS

In this paper we introduce an innovative approach for 3-D object retrieval based on 3-D curve skeletons. Therefore, we employ an already existing skeleton extraction technique with several drawbacks in respect to our project. To overcome the most significant problems of the resulting 3-D skeletons (e.g., a curve thickness of more than one voxel), we propose a modified version of the *Dijkstra* algorithm. Additionally, we present a new feature set. This feature set only considers the topological structure of the skeleton which makes it quite challenging to discriminate objects. We decide to ignore the geometrical information in order to prove the robustness of skeletons. The feature vector, in turn, is combined with a suitable similarity measure, the cosine angle, that enables us to evaluate our method. In addition to this, we generate a ground truth database consisting of 3-D objects. This database is clustered to "similarity groups" by volunteers. Given these information, we perform a challenging evaluation showing quite promising results that justify our research. The evaluation values are expressed in well-known terms from the field of information retrieval. In the future, we will compare our algorithm to other state-of-the-art techniques. In this context we are going to use a new medical 3-D object datasets as well as more comprehensive and standardized databases as the *McGill 3-D Shape Benchmark*, *Princeton Shape Benchmark* and

objects from the *AIM@Shape* database. As a result of this, our method is directly comparable to other methods. Moreover, since the medical 3-D object database is already evaluated based on skeletons by using their geometrical features, we are able to compare these results against whose of our method. Finally, we are going to combine both, the topological and geometrical information. Related to this, we will extend our reference set (GT) as well and we are going to investigate other possibilities of 3-D object representation. Further research plans consider also other skeletonization algorithms, features and feature sets as well as other input data structures (e.g. point clouds, meshes). The latter point is quite interesting considering the steadily increasing amount of, e.g., Kinect devices and the number of research based on such a device. Besides this, we plan to improve all of our tests in terms of invariance power and noise sensitivity.

ACKNOWLEDGEMENTS

This work was funded by the German Research Foundation (DFG) as part of the Research Training Group GRK 1564 “Imaging New Modalities”.

REFERENCES

Bai, X. and Latecki, L. (2008). Path similarity skeleton graph matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(7):1282–1292.

Blum, H. (1967). A transformation for extracting new descriptors of shape. In Wathen-Dunn, W., editor, *Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press, Cambridge.

Borgefors, G. (1996). On digital distance transforms in three dimensions. *Computer Vision and Image Understanding*, 64(3):368 – 376.

Cao, J., Tagliasacchi, A., Olson, M., Zhang, H., and Su, Z. (2010). Point cloud skeletons via laplacian based contraction. In *Shape Modeling International Conference (SMI), 2010*, pages 187–197.

Chang, S. (2007). Extracting skeletons from distance map. *International Journal of Computer Science and Network Security*, 7(7):213–219.

Cornea, N., Demirci, M., Silver, D., Shokoufandeh, Dickinson, S., and Kantor, P. (2005). 3d object retrieval using many-to-many matching of curve skeletons. In *Shape Modeling and Applications, 2005 International Conference*, pages 366–371.

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

Hassouna, M. and Farag, A. (2007). On the extraction of curve skeletons using gradient vector flow. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8.

Hayashi, T., Raynal, B., Nozick, V., and Saito, H. (2011). Skeleton features distribution for 3d object retrieval. In *proc. of the 12th IAPR Machine Vision and Applications (MVA2011)*, pages 377–380, Nara, Japan.

Ma, C. M. and Sonka, M. (1996). A fully parallel 3d thinning algorithm and its applications. *Comput. Vis. Image Underst.*, 64(3):420–433.

Macrini, D., Shokoufandeh, A., Dickinson, S., Siddiqi, K., and Zucker, S. (2002). View-based 3-d object recognition using shock graphs. In *Proceedings of the 16 th International Conference on Pattern Recognition (ICPR’02) Volume 3 - Volume 3*, ICPR ’02, pages 24–28, Washington, DC, USA. IEEE Computer Society.

Ogniewicz, R. and Ilg, M. (1992). Voronoi skeletons: theory and applications. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR ’92., 1992 IEEE Computer Society Conference on*, pages 63–69.

Ogniewicz, R. L. and Kübler, O. (1995). Hierarchic voronoi skeletons. *Pattern Recognition, Vol. 28*, pages 343–359.

Palágyi, K. and Kuba, A. (1998). A 3d 6-subiteration thinning algorithm for extracting medial lines. *Pattern Recognition Letters 19*, pages 613–627.

Palágyi, K. and Kuba, A. (1999). Directional 3d thinning using 8 subiterations. In *Proceedings of the 8th International Conference on Discrete Geometry for Computer Imagery*, DCGI ’99, pages 325–336, London, UK, UK. Springer-Verlag.

Pizlo, Z. (2008). *3D Shape: Its Unique Place in Visual Perception*. The MIT Press, Cambridge - Massachusetts, London - England.

Reniers, D. (2008). *Skeletonization and Segmentation of Binary Voxel Shapes*. PhD thesis, Technische Universiteit Eindhoven.

Reniers, D. and Telea, A. (2006). Quantitative comparison of tolerance-based feature transforms. *First International Conference on Computer Vision Theory and Applications (VISAPP)*, pages 107–114.

Schäfer, S. (2011). Path similarity skeleton graph matching for 3d objects. Master’s thesis, Universität Koblenz-Landau.

Sharf, A., Lewiner, T., Shamir, A., and Kobbelt, L. (2007). On-the-fly Curve-skeleton Computation for 3D Shapes. *Computer Graphics Forum*, 26(3):323–328.

Siddiqi, K. and Pizer, S. (2008). *Medial Representations: Mathematics, Algorithms and Applications*. Springer Publishing Company, Incorporated, 1st edition.

Sundar, H., Silver, D., Gagvani, N., and Dickinson, S. (2003). Skeleton based shape matching and retrieval. In *Shape Modeling International, 2003*, pages 130–139.

Zhang, J., Siddiqi, K., Macrini, D., Shokoufandeh, A., and Dickinson, S. (2005). Retrieving articulated 3-d models using medial surfaces and their graph spectra. In *Proceedings of the 5th international conference on Energy Minimization Methods in Computer Vision and Pattern Recognition*, EMMCVPR’05, pages 285–300, Berlin, Heidelberg. Springer-Verlag.