SkelNetOn 2019 Dataset and Challenge on Deep Learning for Geometric Shape Understanding

http://ubee.enseeiht.fr/skelneton/

İlke Demir[1], Camilla Hahn[2], Kathryn Leonard[3], Geraldine Morin[4], Dana Rahbani[5],
Athina Panotopoulou[6], Amelie Fondevilla[7], Elena Balashova[8], Bastien Durix[4], Adam Kortylewski[5,9]

[1]DeepScale, [2]Bergische Universität Wuppertal, [3]Occidental College, [4]University of Toulouse,
[5]University of Basel, [6]Dartmouth College, [7]Université Grenoble Alpes, [8]Princeton University,
[9] Johns Hopkins University

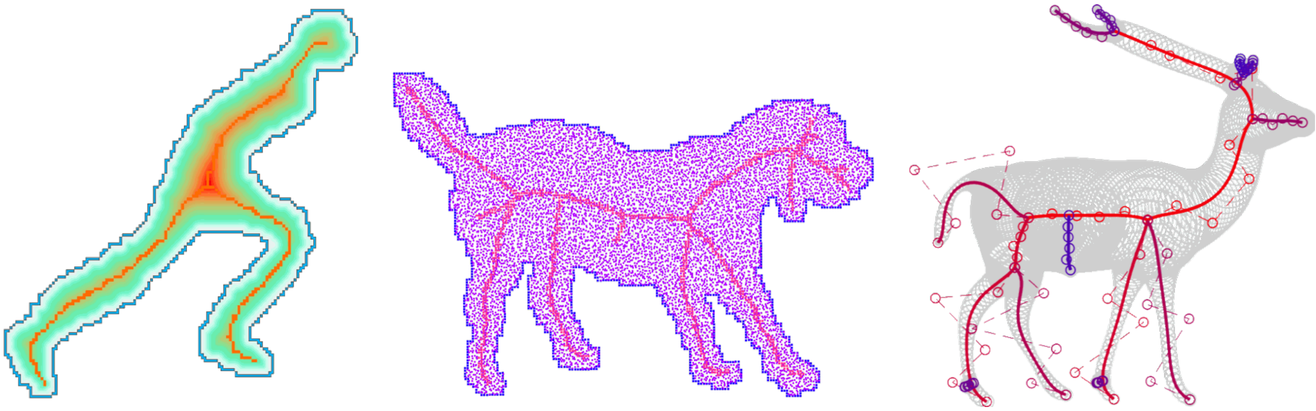


Figure 1: **SkelNetOn Challenges:** Example shapes and corresponding skeletons are demonstrated for the three challenge tracks in pixel (left), point (middle), and parametric domain (right).

Abstract

We present SkelNetOn 2019 Challenge and Deep Learning for Geometric Shape Understanding workshop to utilize existing and develop novel deep learning architectures for shape understanding. We observed that unlike traditional segmentation and detection tasks, geometry understanding is still a new area for investigation using deep learning techniques. SkelNetOn aims to bring together researchers from different domains to foster learning methods on global shape understanding tasks. We aim to improve and evaluate the state-of-the-art shape understanding approaches, and to serve as reference benchmarks for future research. Similar to other challenges in computer vision domain [6, 22], SkelNetOn tracks propose three datasets and corresponding evaluation methodologies; all coherently bundled in three competitions with a dedicated workshop co-located with CVPR 2019 conference. In this paper, we describe and analyze characteristics of each dataset, define the evaluation criteria of the public competitions, and provide baselines for each task.

1. Introduction

Public datasets and benchmarks play an important role in computer vision especially as the rise of deep learning approaches accelerate the relative scalability and reliability of solutions for different tasks. Starting with ImageNet [27], COCO [22], and DAVIS [6] for detection, captioning, and segmentation tasks respectively, competitions bring fairness and reproducibility to the evaluation process of computer vision approaches and improve their capabilities by introducing valuable datasets. Such datasets, and the corresponding challenges, also increase the visibility, availability, and feasibility of machine learning models, which brings up even more scalable, diverse, and accurate algorithms to be evaluated on public benchmarks.

Computer vision approaches have shown tremendous progress toward understanding shapes from various data formats, especially since entering the deep learning era. Although detection, recognition, and segmentation approaches achieve highly accurate results, there has been relatively less attention and research dedicated to extracting topological and geometric information from shapes. However, geometric representations (skeletons, medial axis, boundary curves, etc.) of a shape provide a compact and intuitive representation of the shape for modeling, synthesis, compression, and analysis. Generating or extracting such representations is significantly different from segmentation and recognition tasks, as they condense both local and global information about the shape, and they also often combine topological and geometrical recognition.

We observe that the main challenges for such shape abstraction tasks are (i) the inherent dimensionality reduction from the shape to the skeleton, (ii) the domain change as the true skeletal representation would be best expressed in a continuous domain, and (iii) the trade off between the noise and representative power for the skeleton to prohibit overbranching but still not lose the shape. Although the lower dimensional representation is a clear advantage for shape manipulation, it raises the challenge of characterizing features and representation, especially for deep learning. Computational methods for skeleton extraction are abundant, but are typically not robust to noise on the boundary of the shape (see [33], for example). Small changes in the boundary result in large changes to the skeleton structure, with long branches describing insignificant bumps on the shape. Even for clean extraction methods such as Durix's robust skeleton [12] used for our dataset, changing the resolution of an image changes the value of the threshold for keeping only the desirable branches. Training a neural network to learn to extract a clean skeleton directly, without relying on a threshold, would be a significant contribution to skeleton extraction. In addition, recent deep learning algorithms have shown great results in tasks requiring dimensionality reduction and such approaches could be easily applied to the shape abstraction task we describe in this paper.

Our observation is that deep learning approaches are useful for proposing generalizable and robust solutions for such ill-posed problems. Our motivation arises from the fact that such deep learning approaches need comprehensive datasets, similar to 3D shape understanding benchmarks based on ShapeNet [9], SUNCG [32], and SUMO [34] datasets, with corresponding challenges. The tasks and expected results from such networks should also be well-formulated in order to evaluate and compare them properly. We chose skeleton extraction as the main task, to be investigated in pixel, point, and parametric domains in increasing complexity.

In order to solve the proposed problem with deep learning and direct more attention to geometric shape understanding tasks, we introduce SkelNetOn Challenge. We aim to bring together researchers from computer vision, computer graphics, and mathematics to advance the state of the art in topological and geometric shape analysis using deep learning. The datasets created and released for this competition will serve as reference benchmarks for future research in deep learning for shape understanding. Furthermore, different input and output data representations can become valuable testbeds for the design of robust computer vision and computational geometry algorithms, as well as understanding deep learning models built on representations in 3D and beyond. The three SkelNetOns are defined below:

- **Pixel SkelNetOn:** As the most common data format for segmentation or classification models, our first domain poses the challenge of extracting the skeleton pixels from a given shape in an image.

- **Point SkelNetOn:** The second challenge track investigates the problem in the point domain, where the shapes will be represented by point clouds as well as the skeletons.

- **Parametric SkelNetOn:** The last domain aims to push the boundaries to find parametric representation of the skeleton of a shape, given its image. The participants are expected to output the skeleton of the shape represented as parametric curves, together with a radius function.

In the next section, we introduce how the skeletal models are generated. We then inspect characteristics of our datasets and the annotation process (Section 3), detailed description of the tasks and formulations of evaluation metrics (Section 4), and state-of-the-art methods as well as our baselines (Section 5). The results of the competitions will be presented in the Deep Learning for Geometric Shape Understanding Workshop during the 2019 International Conference on Computer Vision and Pattern Recognition (CVPR) in Long Beach, CA on June 17th, 2019. As of March 15th, 2019, more than 160 participants have registered in SkelNetOn competitions and there are 25 valid submissions in the leaderboard over the three tracks.

2. Skeletal Models and Generation

The Blum medial axis (BMA) is the original skeletal model, consisting of a collection of points equidistant from the boundary (the skeleton) and their corresponding distances (radii) [3]. The BMA produces skeleton points both inside and outside the shape boundary. Since each set, interior and exterior, reconstructs the boundary exactly, we select the interior skeleton to avoid redundancy. For the interior BMA, skeleton points are centers of circles maximally inscribed within the shape and the radii are their associated circles radii. For a discrete sampling of the boundary, Voronoi approaches to estimating the medial axis are well-known, and proven to converge to the true BMA as the boundary sampling becomes dense [25]. In the following, we refer to the skeleton points and radius of the shape together as the interior BMA.

The skeleton offers an intuitive and low dimensional representation of the shape that has been exploited for shape recognition, shape matching, and shape animation. However, this representation also suffers from poor robustness: small perturbations of the boundary may cause long branches to appear that model only a small boundary change. Such are uninformative about the shape. These perturbations also depend on the domain of the shape representation, since the noise on the boundary may be the product of coarse resolution, non-uniform sampling, and approximate parameterization. Many approaches have been proposed to remove these uninformative branches from an existing skeleton [1, 10, 13], whereas some more recent methods offer a skeletonization algorithm that directly computes a clean skeleton [12, 19].

We base our ground truth generation on this second approach. First, we apply one-pixel dilation and erosion operations on the original image to close some negligible holes and remove isolated pixels that might change the topology of the skeleton. We manually adjust the shapes if the closing operation changes the shape topology. Then, the skeleton is computed with 2-, 4-, and 6-pixel thresholds. In other words, the Hausdorff distance from the shape represented by the skeleton to the shape used for the skeletonization is at most 2, 4, or 6 pixels [12]. We then manually select the skeleton that is visually the most coherent for the shape from among the three approximations to produce a skeleton which has the correct number of branches. Finally, we manually remove some isolated skeleton points or branches if spurious branches still remain.

We compiled several shape repositories [5, 4, 20, 29] for our dataset with $1,725$ shapes in 90 categories. We used the aforementioned skeleton extraction pipeline for obtaining the initial skeletons, and created shapes and skeletons in other domains using the shape boundary, skeleton points, and skeleton edges.

3. Datasets

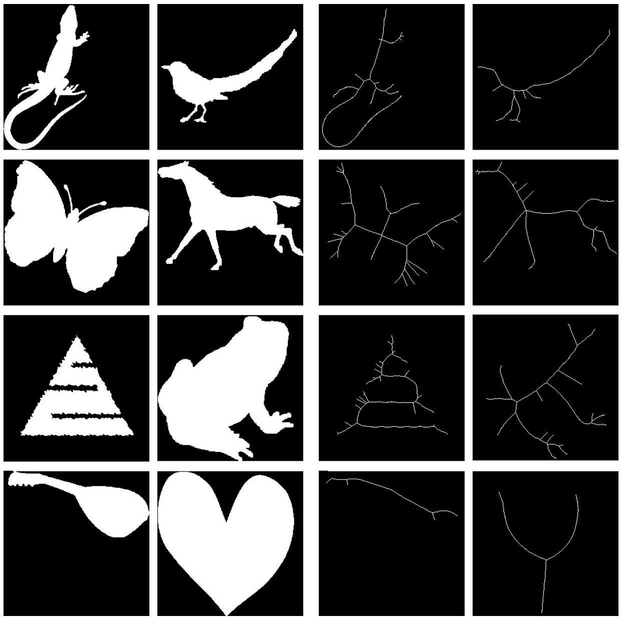
We converted the shapes and their corresponding ground truth skeletal models into three representation domains: pixels, points, and Bézier curves. In this section, we will discuss these datasets derived from the initial skeletonization.

3.1. Shapes and Skeletons in Pixels

The image dataset consists of $1,725$ black and white images given in portable network graphics format with size 256×256 pixels, split into $1,219$ training images, 242 validation images, and 266 test images.

We provide samples from every class in both the test and validation sets. There are two types of images: the shape images which represent the shapes in our dataset (Figure 2a), and the skeleton images which represent the skeletons corresponding to the shape images (Figure 2b). In the shape images, the background is annotated with black pixels and the shape with a closed polygon filled with white pixels. In the skeleton images, the background is annotated with black pixels and the skeleton is annotated with white pixels. The shapes have no holes; some of them are natural, while others are man-made. If one overlaps a skeleton image with its corresponding shape image, the skeleton will lie in the "middle" of the shape (i.e., it would be an approximation of the shape's skeleton).

For generating shape images, inside of the shape boundaries mentioned in Section 2 are rendered in white, whereas outside is rendered in black. The renders are then cropped, padded, and downsampled to 256×256. No noise was added or removed, therefore all expected noise is due to pixelation or resizing effects. For generating skeleton images, the skeleton points as well as all pixels linearly falling between two consecutive skeleton points are rendered in white, on a black background. By definition of the skeletons from the original generation process, we assume adjacency within 8-neighborhood in the pixel domain, and provide connected skeletons.



(a) Shapes (b) Skeletons

Figure 2: **Pixel Dataset.** A subset of the shape and skeleton image pairs is demonstrated from our pixel dataset.

3.2. Shapes and Skeletons in Points

Similar to the image dataset, the point dataset consists of $1,725$ shape point clouds and corresponding ground truth skeleton point clouds, given in the basic point cloud export format `.pts`. Sample shape point clouds and their corresponding skeleton point clouds are shown in Figure 3.

The dataset is again split into $1,218$ training point clouds, 241 validation point clouds, and 266 test point clouds. We derive the point dataset by extracting the boundaries of the shapes (mentioned in Section 2) as two-dimensional point clouds. We fill the closed region within this boundary by points that implicitly lie on a grid with granularity $h = 1$. After experimenting with over/under sampling the point cloud with different values of h, we end up with this balancing value because the generated point clouds were representative enough to not lose details, and still computationally reasonable to process. Even though the average discretization parameter is given as $h = 1$ in the provided dataset, we shared scripts[1] in the competition starting kit to coarsen or populate the provided point clouds so participants are able to experiment with different granularities of the point cloud representation. To prevent the comfort of regularity which is observed in the pixel domain, we add some uniformly distributed noise and avoid any structural dependency in the later computed results. The noise is scaled by the sampling density, and we also provide scripts to apply noise with other probability distributions, such as Gaussian noise.

Ground truth for the skeletal models are given as a second point cloud which only contains the points representing the skeleton. To compute the skeletal point clouds, we computed the proximity of each point in the shape point cloud to the skeleton points and skeleton edges from the original dataset. Shape points closer than a threshold (depending on h) to any original skeleton points or edges in Euclidean space are accepted for the skeleton point cloud. This generation process allows one-to-one matching of each point in the skeleton point cloud to a point in the shape point cloud, thus the ground truth can be converted to labels for the shape point cloud if the task in hand would be assumed as a segmentation task.

3.3. Parametric Skeletons

Finally, the parametric skeleton dataset consists of $1,725$ shape images and corresponding ground truth parametric skeletons, exported in tab separated `.csv` format. The dataset is again split into $1,218$ training shapes, 241 validation shapes, and 266 test shapes. The shape images are created as discussed in Section 3.1, and the parametric skeletons are modeled as degree five Bézier curves. Each curve corresponds to a branch of the skeleton, where the first two coordinates describe the $\{x, y\}$ location in the image, and the third coordinate is the radius r associated with the point. Curves are represented as degree five Bézier curves, and the output is a vector containing the 3D (x, y, r) coordinates of the control points of each branch.

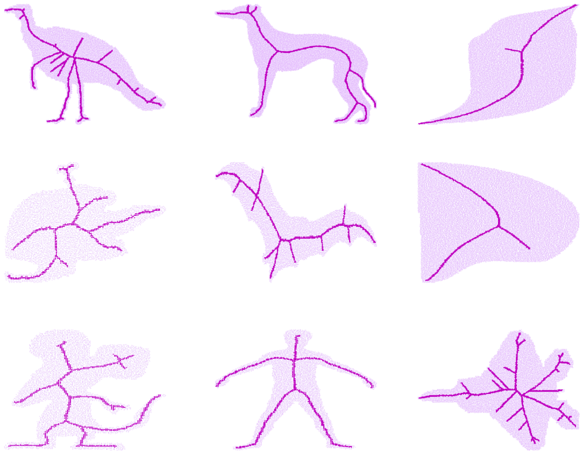


Figure 3: **Point Dataset.** A subset of our shape and skeleton point cloud pairs is demonstrated.

$$v = [x_0^0, y_0^0, r_0^0, x_1^0, y_1^0, r_1^0, ..x_5^0, y_5^0, r_5^0, x_0^1, y_0^1, r_0^1..], \quad (1)$$

where $b_i^j = (x_i^j, y_i^j, r_i^j)$ is the i-th control point of the j-th branch in the medial axis.

From the simply connected shapes of the dataset mentioned in Section 2, we first extract a clean medial axis representation. For a simply connected shape, the skeleton is a tree, whose joints and endpoints are connected by curves, which we call proto-branches. Unfortunately, the structure of the tree is not stable. Because skeletons are unstable in the presence of noise, a new branch due to a small perturbation of the boundary could appear and break a proto-branch into two branches. Moreover the tree structure gives a dependency and partial order between the branches, not a total order. To obtain a canonical parametric representation of the skeleton, we first merge branches that have been broken by a less important branch. We then order the reduced set of branches according to branch importance. For both steps, we use a salience measure, the Weighted Extended Distance Function (WEDF) function on the skeleton [20], to determine the relative importance of branches. The WEDF function has been shown to measure relative importance of shape parts in a way that matches human perception [8].

Merging branches. First, we identify pairs of adjacent branches that should be joined to represent a single curve: a branch is split into two parts at a junction induced by a child branch of lower importance if the WEDF function is continuous across the junction. When two child branches joining the parent branch are of equal importance, then the junction is considered an end point of the parent curve. Figure 4 shows the resulting curves.

[1] https://github.com/ilkedemir/SkelNetOn

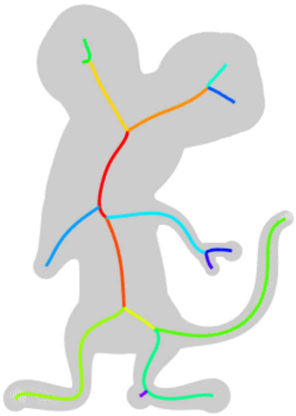


Figure 4: **Skeletal Branches.** The red curve passes continuously through the arm and nose curves. However, legs (bottom) and ears (top) split the main red curve into two parts of equal importance, becoming the end-points.

Computation of the Bézier approximation. Each individual curve resulting from the merging process is approximated by a Bézier curve of degree five, whose control points have three parameters (x, y, r), where (x, y) are coordinates, and r is the radius. The end points of the curve are interpolated, and the remaining points are determined by a least square approximation.

Branch ordering We then order the branches by importance to have a canonical representation. We estimate a branch importance by the maximal WEDF value of a point in the branch. The branches can then be ordered, and their successive list of control points is the desired output.

4. Tasks and Evaluations

4.1. Pixel Skeleton Classification

Generating skeleton images from the given shape images can be posed as a pixel-wise binary classification task, or an image generation task. This makes it possible to evaluate performance by comparing a generated skeleton image, pixel by pixel, to its ground truth skeleton image. Such a comparison automatically accounts for common errors seen in skeleton extraction algorithms such as lack of connectivity, double-pixel width, and branch complexity.

However, using a simple $L1$ loss measurement would provide a biased evaluation of image similarity. One can see this by looking at any of the skeleton images: the black background pixels far outnumber the white skeleton ones, giving the former much more significance in the final value of the computed $L1$ loss. To minimize the effects of class imbalance, the evaluation is performed using the $F1$ score, which takes into account the number of skeleton and background pixels in the ground truth and generated images. This is consistent with metrics used in the literature and will enable further comparisons with future work [30]. The number of skeleton pixels (positive) or background pixels (negative) is first counted in both the generated and ground truth skeleton images. The $F1$ score is then calculated from the harmonic average of the precision and recall values as follows:

$$F1 = \frac{2 \times precision \times recall}{precision + recall}, \tag{2}$$

using

$$\begin{aligned} Precision &= \frac{TP}{TP + FP} \\ Recall &= \frac{TP}{TP + FN}, \end{aligned} \tag{3}$$

where TP, FN, and FP stand for the number of pixels for true positives, false negatives, and false positives respectively.

4.2. Point Skeleton Extraction

Given a 2D point set representing a shape, the goal of the point skeleton extraction task is to output a set of point coordinates corresponding to the given shape's skeleton. This can be approached as a binary point classification task or a point generation task, both of which end up producing a skeleton point cloud that approximate the shape skeleton. The output set of skeletal points need not be part of the original input point set. The evaluation metric for this task needs to be invariant to the number and ordering of the points. The metric should also be flexible for different point sampling distributions representing the same skeleton. Therefore, the results are evaluated using the symmetric Chamfer distance function, defined by:

$$\begin{aligned} Ch(A,B) = \frac{1}{|A|}\sum_{a\in A}\min_{b\in B}||a-b||_2 + \\ \frac{1}{|B|}\sum_{b\in B}\min_{a\in A}||a-b||_2, \end{aligned} \tag{4}$$

where A and B represent the skeleton point sets to be compared. We use a fast vectorized `Numpy` implementation of this function in order to compute Chamfer distances quickly in our evaluation script.

4.3. Parametric Skeleton Generation

The parametric skeleton extraction task aims to recover the medial axis as a set of 3D parametric curves from an input image of a shape. Following the shape and parametric skeleton notations introduced in Section 3.3, different metrics can be proposed to evaluate such a representation. In particular, we can measure either the distance of the output medial axis to the ground-truth skeleton, or its distance to the shape described in the input image. Although the second method looks better adapted to the task, it does not take into account several properties of the medial axis. It would be difficult, for example, to penalize disconnected branches or redundant parts of the medial axis.

We evaluate our results by distance to the ground truth medial axes in our database, since the proposed skeletal representation in the dataset already guarantees the properties introduced above and in Section 3.3, and are ordered in a deterministic order. We use the mean squared distance between the control points on the original and predicted branches as:

$$MSD(b,\tilde{b}) = \frac{1}{6}\sum_{i=0}^{5}\left((x_i-\tilde{x}_i)^2+(y_i-\tilde{y}_i)^2+(r_i-\tilde{r}_i)^2\right), \tag{5}$$

where $b = (x_i, y_i, r_i)_{i=\{0..5\}}$ is a branch in the ground truth data, and $\tilde{b} = (\tilde{x}_i, \tilde{y}_i, \tilde{r}_i)_{i=\{0..5\}}$ is the corresponding branch in the output data.

The evaluation metric needs to take into account models with an incorrect number of branches, since this number is different for each shape. We penalize each missing (or extra) branch in the output with a measure on the length of the branch in the ground truth (or in the output). We use a measure called missing branch error (MBE) for each missing or extra branch b:

$$MBE(b) = \frac{1}{5}\sum_{i=0}^{4}(x_{i+1}-x_i)^2+(y_{i+1}-y_i)^2+\frac{1}{6}\sum_{i=0}^{5}r_i^2. \tag{6}$$

Finally, the evaluation function between an output vector $\tilde{V}$ and its associated ground truth V is defined as:

$$D(V,\tilde{V}) = \frac{1}{N_b}\left(\sum_{j=0}^{n_b-1} MSD(b^j,\tilde{b}^j) + \sum_{j=n_b}^{N_b-1} MBE(\hat{b}^j)\right) \tag{7}$$

where N_b, n_b are respectively the number minimal and maximal of branches in the ground truth and in the output, and $\hat{b}$ are branches in the vector containing the maximal number of branches.

5. State-of-the-art and Baselines

Skeleton extraction has been an interesting topic of research for different domains. We briefly introduce these approaches in addition to our own deep learning based baselines. The participants are expected to surpass these baselines in order to be eligible for the prizes.

5.1. Pixel Skeleton Results

Early work on skeleton extraction was based on using segmented images as input [2]. A comprehensive overview of previous approaches using pre-segmented inputs is presented in [28]. With the advancement of neural network technology, the most recent approaches perform skeletonization from natural images with fully convolutional neural networks [30, 31]. In this challenge, we consider the former type of skeletonization task using pre-segmented input images.

We formulate the task of skeletonization as image translation from pre-segmented images to skeletons with a conditional adversarial neural network. For our baseline result, we use a vanilla pix2pix model as proposed in [16]. We apply a distance transform to preprocess the binary input shape as illustrated in Figure 5. We found that this preprocessing of the input data enhances the neural network learning significantly. The model is trained with stochastic gradient descent using the $L1$-image loss for 200 epochs. We measure performance in terms of $F1$ score achieving a test accuracy of **0.6244** on the proposed validation set.

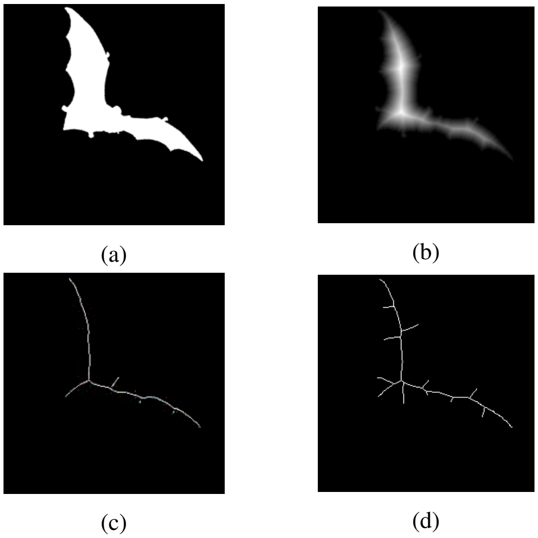


Figure 5: **Pixel SkelNetOn Baseline.** (a) Original shape image. (b) Distance transformed shape image. (c) Baseline prediction result. (d) Ground truth skeleton.

5.2. Point Skeleton Results

Skeleton and medial axis extraction from point clouds have been extensively explored in 2D and 3D domains by using several techniques such as Laplacian contraction [7], mass transport [17], medial scaffold [21], locally optimal projection [23], maximal tangent ball [24], and local L_1 medians [15]. Although these approaches extract approximate skeletons, the dependency on sampling density, non-uniform and incomplete shapes, and the inherent noise in the point clouds are still open topics that deep learning approaches can handle implicitly. One such recent approach (P2PNet [36]) builds a Siamese architecture to build spatial transformer networks for learning how points can be moved from a shape surface to the medial axis and vice versa.

Based on the definition of this task in Section 4, it can be formulated as (i) a generation task to create a point cloud by learning the distribution from the original shapes and skeleton points, (ii) a segmentation task to label the points of a shape as skeletal and non-skeletal points, and (ii) a spatial task to learn the pointwise translations to transform the given shape to its skeleton. We chose the second approach to utilize state-of-the-art point networks. First, we obtained ground truth skeletons as labels for the original point cloud (skeletal label as 1, and non-skeletal labeled as 2). Then we experimented with PointNet++ [26] architecture's part-segmentation module to classify each pixel into two classes, defined by the labels. As the dimensionality of the classes are inherently different, we observed that the skeleton class collapses. Similarly, if we over-sample the skeleton, non-skeleton class collapses. To overcome this problem, we experimented with dynamic weighting of samples, so that the skeletal points are trusted more than non-skeletal points. The weights are adjusted at every 10^{th} epoch as well as the learning rate, to keep the balance of the classes. Our approach achieved an accuracy of 58.93%, measured by the mean IoU over all classes. Even though the mIoU is a representative metric for our task, we would like to evaluate our results better by calculating the Chamfer distance of skeletons in shape space in future.

5.3. Parametric Medial Axis Results

Parametric representations of the medial axis have been proposed before, in particular representations with intuitive control parameters. For example, Yushkevich. et al, [37] use a cubic B-spline model with control points and radii; similarly Lam et al. [18] use piecewise Bézier curves.

We train a Resnet-50 [14] modified for regression. Inputs are segmented binary images, and outputs are six control points in $\mathbb{R}^3$ that give (x, y, r) values for the degree five Bézier curves used to model medial axis branches.

The parametric data is the most challenging of the three ground truths presented here, for two reasons. First, the number of entries in the ground truth varies with the number of branches in the skeleton. Second, the Bézier control points do not encode information about the connectivity of the skeleton graph. To overcome the first challenge, we set all shapes to have 5 branches. For those shapes with fewer than 5 branches, we select the medial point with the maximum WEDF value (a center of the shape) as the points in any non-existent branches, and set the corresponding radii to zero. We do not address the second issue for our baseline results. We use a simple mean squared error measure on the output coordinates as our loss function. While more branches are desirable to obtain quality models, there is a trade-off between the number of meaningless entries in the ground truth for a simple shape (with a low number of branches), and providing for an adequate number of branches in the ground truth of a complex shape (with many branches).

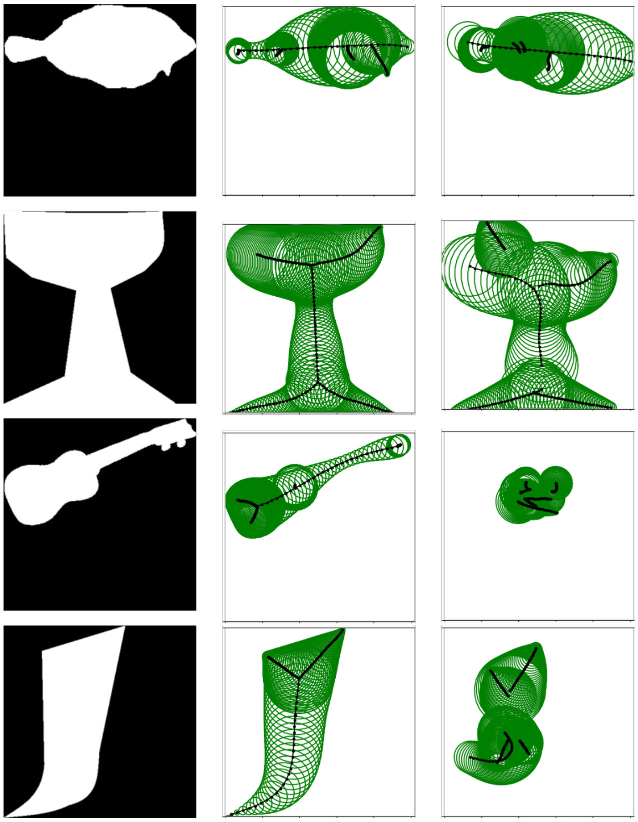


Figure 6: **Parametric SkelNetOn Baseline.** Input image (left), ground truth parametric medial axis (middle), prediction from our model (right).

We obtain a validation loss of 1405 pixels, with a corresponding training loss of 121 pixels, using early stopping. For comparison, loss at start of training is in the order of $50,000$ pixels.

As shown in Figure 6, branch connectivity is only occasionally produced by our trained model. The model does well with location, orientation, and scale of shape, but often misses long and narrow appendages such as the guitar neck (third row).

6. Conclusions

We presented SkelNetOn Challenge at Deep Learning for Geometric Shape Understanding, in conjunction with CVPR 2019. Our challenge provides datasets, competitions, and a workshop structure around skeleton extraction in three domains as Pixel SkelNetOn, Point SkelNetOn, and Parametric SkelNetOn. We introduced our dataset analysis, formulated our evaluation metrics following our competition tasks, and shared our preliminary results as baselines following the previous work in each domain.

We believe that SkelNetOn has the potential to become a fundamental benchmark for the intersection of deep learning and geometry understanding. We foresee that the challenge and the workshop will enable more collaborative research of different disciplines on the crossroads of shapes. Ultimately, we envision that such deep learning approaches can be used to extract expressive parametric and hierarchical representations that can be utilized for generative models [35] and for proceduralization [11].

Acknowledgments

Without the contributions of the rest of our technical and program committees, this workshop would not happen, so thank you everyone, in particular Veronika Schulze for her contributions throughout the project, Daniel Aliaga as the papers chair, and Bedrich Benes and Bernhard Egger for the support. The authors thank AWM's Women in Shape (WiSH) Research Network, funded by the NSF AWM Advances! grant. We also thank the University of Trier Algorithmic Optimization group, who provided funding for the workshop where this challenge was devised. We would also like to thank NVIDIA for being the prize sponsor of the competitions. Lastly, we would like to acknowledge the support and motivation of the CVPR chairs and crew.

References

[1] Attali, D. and Montanvert, A. Computing and Simplifying 2D and 3D Continuous Skeletons. 67(3), 1997.

[2] X. Bai, L. J. Latecki, and W.-Y. Liu. Skeleton pruning by contour partitioning with discrete curve evolution. *IEEE transactions on pattern analysis and machine intelligence*, 29(3):449–462, 2007.

[3] Blum, H. Biological shape and visual science. *Journal of theoretical Biology*, 38(2):205–287, 1973.

[4] A. Bronstein, M. Bronstein, and R. Kimmel. *Numerical Geometry of Non-Rigid Shapes*. Springer Publishing Company, Incorporated, 1 edition, 2008.

[5] E. M. Bronstein, M. M. Bronstein, A. M. Bruckstein, and R. Kimmel. Analysis of two-dimensional non-rigid shapes. *IJCV*, 2007.

[6] S. Caelles, A. Montes, K.-K. Maninis, Y. Chen, L. Van Gool, F. Perazzi, and J. Pont-Tuset. The 2018 DAVIS Challenge on Video Object Segmentation. *ArXiv e-prints*, Mar. 2018.

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

[8] A. Carlier, K. Leonard, S. Hahmann, G. Morin, and M. Collins. The 2d shape structure dataset: A user annotated open access database. *Computers & Graphics*, 58:23–30, 2016.

[9] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

[10] Chazal, F. and Lieutier, A. The λ-medial Axis. *Graphical Models*, 67(4), 2005.

[11] I. Demir and D. G. Aliaga. Guided proceduralization: Optimizing geometry processing and grammar extraction for architectural models. *Computers & Graphics*, 74:257 – 267, 2018.

[12] Durix, B. and Chambon, S. and Leonard, K. and Mari, J.-L. and Morin, G. The propagated skeleton: a robust detail-preserving approach. In *DGCI*, 2019.

[13] Giesen, J. and Miklos, B. and Pauly, M. and Wormser, C. The Scale Axis Transform. In *Symp. on Comp. Geometry*, 2009.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778, 2016.

[15] H. Huang, S. Wu, D. Cohen-Or, M. Gong, H. Zhang, G. Li, and B. Chen. L1-medial skeleton of point cloud. *ACM Trans. Graph.*, 32(4):65:1–65:8, July 2013.

[16] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[17] A. C. Jalba, A. Sobiecki, and A. C. Telea. An unified multiscale framework for planar, surface, and curve skeletonization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(1):30–45, Jan 2016.

[18] J. H. Lam and Y. Yam. A skeletonization technique based on delaunay triangulation and piecewise bezier interpolation. In *2007 6th International Conference on Information, Communications & Signal Processing*, pages 1–5. IEEE, 2007.

[19] Leborgne, A. and Mille, J. and Tougne, L. Extracting Noise-Resistant Skeleton on Digital Shapes for Graph Matching. In *International Symp. on Visual Computing*, 2014.

[20] K. Leonard, G. Morin, S. Hahmann, and A. Carlier. A 2d shape structure for decomposition and part similarity. In *ICPR Proceedings*, 2016.

[21] F. F. Leymarie and B. B. Kimia. The medial scaffold of 3d unorganized point clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2):313–330, Feb 2007.

[22] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*. European Conference on Computer Vision, September 2014.

[23] Y. Lipman, D. Cohen-Or, D. Levin, and H. Tal-Ezer. Parameterization-free projection for geometry reconstruction. *ACM Trans. Graph.*, 26(3), July 2007.

[24] J. Ma, S. W. Bae, and S. Choi. 3d medial axis point approximation using nearest neighbors and the normal field. *The Visual Computer*, 28(1):7–19, Jan 2012.

[25] R. L. Ogniewicz. Skeleton-space: a multiscale shape description combining region and boundary information. In *CVPR'94 Proceedings*, 1994.

[26] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, pages 5099–5108, 2017.

[27] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[28] P. K. Saha, G. Borgefors, and G. S. di Baja. A survey on skeletonization algorithms and their applications. *Pattern Recognition Letters*, 76:3–12, 2016.

[29] T. B. Sebastian, P. N. Klein, and B. B. Kimia. Recognition of shapes by editing their shock graphs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(5):550–571, May 2004.

[30] W. Shen, K. Zhao, Y. Jiang, Y. Wang, X. Bai, and A. Yuille. Deepskeleton: Learning multi-task scale-associated deep side outputs for object skeleton extraction in natural images. *IEEE Transactions on Image Processing*, 26(11):5298–5311, 2017.

[31] W. Shen, K. Zhao, Y. Jiang, Y. Wang, Z. Zhang, and X. Bai. Object skeleton extraction in natural images by fusing scale-associated deep side outputs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 222–230, 2016.

[32] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. *Proceedings of 29th IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[33] Tagliasacchi, A. and Delame, T. and Spagnuolo, M. and Amenta, N. and Telea, A. 3D Skeletons: A State-of-the-Art Report. 35(2):573–597, 2016.

[34] L. Tchapmi, D. Huber, F. Dellaert, I. Demir, S. Song, and R. Luo. The 2019 sumo challenge workshop 360 indoor scene understanding and modeling. `http://sumochallenge.org/`.

[35] J. Wu, C. Zhang, T. Xue, W. T. Freeman, and J. B. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems*, pages 82–90, 2016.

[36] K. Yin, H. Huang, D. Cohen-Or, and H. Zhang. P2p-net: Bidirectional point displacement net for shape transform. *ACM Trans. Graph.*, 37(4):152:1–152:13, July 2018.

[37] P. Yushkevich, P. T. Fletcher, S. Joshi, A. Thall, and S. M. Pizer. Continuous medial representations for geometric object modeling in 2d and 3d. *Image and Vision Computing*, 21(1):17–27, 2003.