Distance and Edge Transform for Skeleton Extraction

Xiaojun Tang*†, Rui Zheng†, Yinghao Wang†,‡
† BOE Technology Group Co.,Ltd, Beijing, China
‡ Faculty of Engineering and IT , The University of Melbourne, Victoria, Australia
{tangxiaojun, zhengr}@boe.com.cn, yinghao.wang@student.unimelb.edu.au

Abstract

The shape skeleton or medial axis, is a concise shape description, defined by the centers of the maximally inscribed circles. Skeletonization algorithms support many applications, including optical character recognition, object recognition, pose estimation, shape matching, biomedical image analysis, etc. Usually, classical algorithms tend to produce redundant skeleton branches at edge noise regions and require a branch pruning post process. Recently many CNN based algorithms achieved significant performance improvements compared with classical algorithms. Most deep learning algorithms directly used the shape of image as input data and its complex for end-to-end learning algorithms to fit the transformation from shape to skeleton. In this work, we proposed to use Smooth Distance Estimation (SDE) and Edge transformation to preprocess the input shape. Combined with a modified U-Net model and multi-model ensemble, the proposed method achieved 0.8129 F1 score in the Pixel SkelNetOn validation set, 1.5752 symmetric chamfer distance in the Point SkelNetOn validation set and 6407.4 squared distance score in the Parametric SkelNetOn validation set.

1. Introduction

The shape skeleton, or medial axis, is a concise shape description, defined by the centers of the maximally inscribed circles. The set of centers and radii completely describes the shape [1]. Skeletonization algorithms support many applications, including optical character recognition [13], object recognition [18], pose estimation [17], shape matching [16], biomedical image analysis [10, 8, 15], etc.

There are three categories of classical skeleton algorithms: morphological thinning based on iterative boundary removal, geometric methods based on Voronoi diagram and distance transform-based methods [11]. Usually, classical algorithms are difficult to distinguish between edge noises and real shapes, and tend to produce redundant skeleton branches at edge noise regions.

With the population of CNNs, deep learning-based algorithms achieved great success in image classification, object detection, image segmentation and image transformation, etc.

In recent years, CNN algorithms have been used for skeleton extraction. In [11], the authors introduced a U-Net based approach for direct skeleton extraction and got a high performance on the pixel SkelNetOn 2019 test set. [9] designed a unique version of HED architecture along with the U-Net structure and got significant performance improvements. [5] built a fully convolutional network named Feature Hourglass Network (FHN) for skeleton detection and achieved better performance compared with the baseline on both the Pixel SkelNetOn and Point SkelNetOn datasets.

In [3], the authors provided a baseline algorithm which used a vanilla pix2pix model with distance transformation preprocessing. They found that the preprocessing of the input data enhanced the neural network learning significantly. In this work, we continue to study new preprocessing methods and propose to use Smooth Distance Estimation (SDE) and Edge transformation to preprocess the input shape. Combined with a modified U-Net model and multi-model ensemble, the proposed method got excellent performances in the pixel SkelNetOn, point SkelNetOn and parametric SkelNetOn validation set.

2. Methods

2.1. U-Net based architecture

We formulated the skeletonization as a pixel classification problem and designed a pixel skeleton classification network based on the U-Net architecture [12]. Thanks to the down sampling and up sampling blocks, the U-Net architecture has a large receptive field and is suitable for skeleton extractions. Like the baseline [3] which applied a distance transform to preprocess the binary input shape, we used the Smooth Distance Estimation [4] to get a more accurate SDE image. In addition, we applied an edge transform to get an edge image which was concatenated with the SDE image and used as the U-Net input.

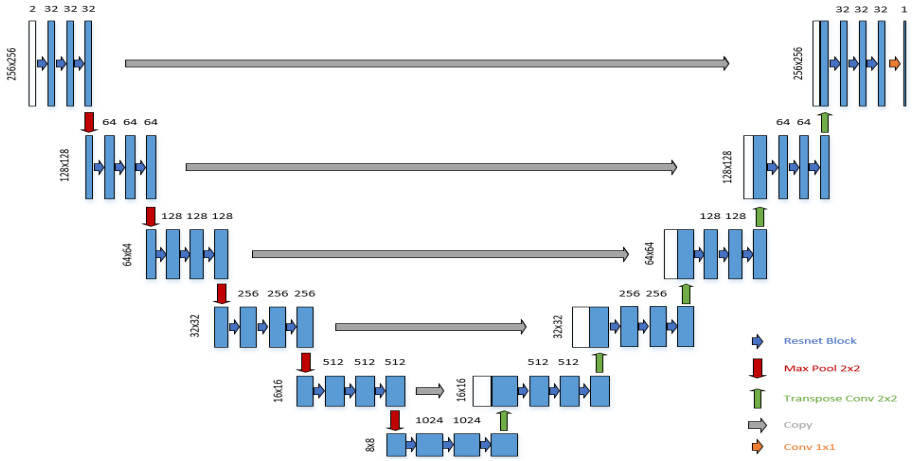

Figure 1. U-Net based architecture: Each down sampling block and up sampling had a sequence of three Resnet blocks. At the end of each down-sampling block a max pooling layer was applied. Each up-sampling block contained a transposed convolution layer, concatenated with a corresponding output of down sampling block. At the end of the last up sampling block, a 1x1 convolution was applied to generate skeleton predictions.

In order to improve the feature express ability of the U-Net model, a sequence of three Resnet blocks was applied in each down sampling block and up sampling block. At the end of each down sampling block a max pooling layer was applied. Each up-sampling block contained a transposed convolution layer, concatenated with a corresponding output of the down sampling block. At the end of the last up sampling block, a 1x1 convolution was applied to generate skeleton predictions.

2.2. Data preprocess

Similar to [3], we applied a one-pixel dilation and erosion operation on the original shape image to close some negligible holes and remove isolated pixels that might change the topology of the skeleton.

2.3. Distance transformation

In [3], the authors proposed to apply distance transformations for input shape preprocess. They found that the preprocess enhanced the neural network learning significantly. While the distance transformation was sensitive to edge noises, [4] proposed the Smooth Distance Estimation (SDE) for robust distance transformations. In this paper, we used the SDE as the U-Net model input, which might guide the model training to focus on the SDE ridge and help the model locating skeleton points more precisely. The SDE values are displayed in Figure 2, which shows that the skeleton points are always located at the SDE ridges.

The traditional distance transformation calculates the min distance from point P to the shape boundary as the following formula shows. Where B is the shape boundary points set.

$$d_{\mathcal{B}}(P) = \min_{B \in \mathcal{B}} \|P - B\|$$

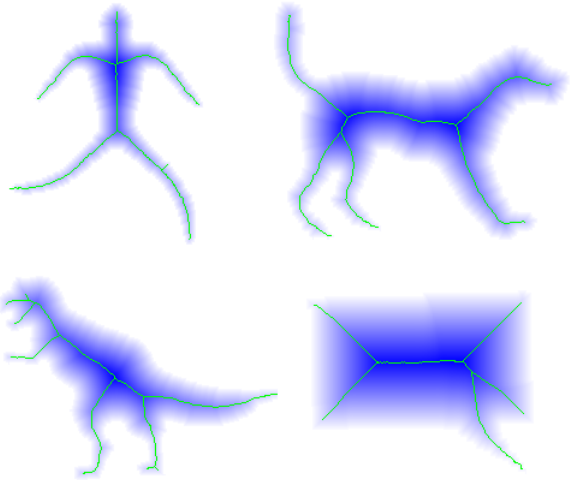

Figure 2. The SDE values are displayed in blue and the skeleton curves are displayed in green. The skeleton points are always located at the SDE ridges. It may guide the U-Net model to focus on the SDE ridges. (Best viewed in color)

The following formulas describe how to calculate the SDE. Where $p_{\mathcal{B},\sigma}(P, B)$ is a weight assigned to the point B of the shape boundary. $P_\sigma(t)$ is a semi-normal distribution with parameter σ. The parameter σ is chosen as 0.7.

$$SDE_{\mathcal{B},\sigma}(P) = \frac{\sum_{B \in \mathcal{B}} \left(p_{\mathcal{B},\sigma}(P, B) \|P - B\|\right)}{\sum_{B \in \mathcal{B}} \left(p_{\mathcal{B},\sigma}(P, B)\right)}$$

$$p_{\mathcal{B},\sigma}(P, B) = P_\sigma\left(\|P - B\| - d_{\mathcal{B}}(P)\right)$$

$$P_\sigma(t) = \frac{2}{\sigma\sqrt{2\pi}} \int_t^{+\infty} e^{-\frac{t^2}{2\sigma^2}} dt$$

2.4. Edge transformation

It was observed that whether a skeleton branch existed was depended on the edge of the shape. The skeleton extraction model should pay more attention to edge regions. We proposed to use an edge transformation and concatenate the edge image with the SDE image.

The shape edges are displayed in Figure 3. The skeleton branches always point to the edge bumps. It may guide the model training to focus on the edge bumps.

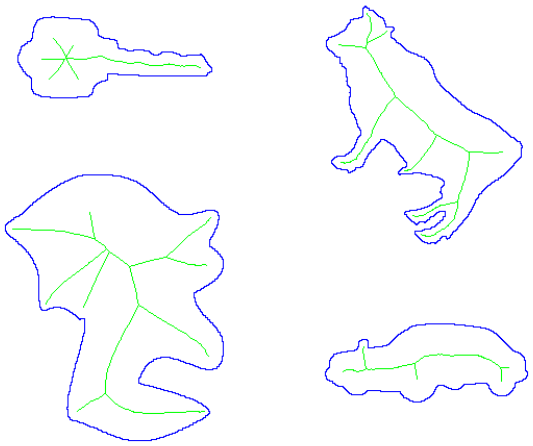

Figure 3. The shape edges are displayed in blue and the skeleton curves are displayed in green. The skeleton branches always point to the edge bumps. It may guide the model training to focus on the edge bumps.

2.5. Multi-model ensemble

In general, multi-model ensembles can improve the accuracy of neural network models. In [11], eight models trained on different subsets of training data were chosen for the model ensemble, and a simple average of predictions increased F1-score by 4%. In this work, we chose five models trained on different subsets of training data and the F1-score is increased by 0.58% on the validation set.

2.6. Pixel skeleton classification

In the Pixel SkelNetOn challenge [3], input shapes were provided and participants were asked to generate the corresponding skeletons. All shapes were represented as binary shape images (the top line of Figure 4), and skeletons were also represented as binary images (the bottom line of Figure 4). In the skeleton images, the background was annotated with black pixels and the skeleton was annotated with white pixels. In order to generate skeleton images, the U-Net model prediction results were passed through a sigmoid layer to get skeleton probability values in the range between 0 and 1. The final skeleton image was generated by binarization of the probability using an optimal threshold.

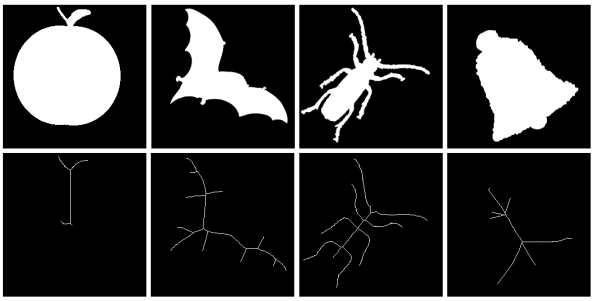

Figure 4. Pixel SkelNetOn challenge: All shapes were represented as binary shape images (top line), and skeletons were also represented as binary images (bottom line).

2.7. Point skeleton Extraction

In the Point SkelNetOn challenge [3], both the shapes and the skeletons were represented by point clouds. Similar to [5], we drew the point cloud of each shape in a white background according to the corresponding coordinates and generated the binary shape images. Then skeleton images were generated by passing the shape images through the U-Net model and the final skeleton point clouds were converted from the skeleton images.

2.8. Parametric skeleton generation

In the Parametric SkelNetOn challenge [3], all shapes were represented as shape images. The parametric skeleton was modeled as degree five Bezier curves. Each curve corresponded to a branch of skeleton, where the first two coordinates described the {x, y} location in the image of an inscribed circle center, and the third coordinate was the radius r associated with the point.

In this paper, we generated parametric skeletons by skeleton branch extraction, Bezier curve fitting and branch ordering.

Skeleton branch extraction: At first, we generated skeleton images using the trained U-Net model as Section 2.6. Since the predicted skeleton may contain broken lines, we used a shortest distance connection to connect all broken lines to form a connected skeleton. Then proto-branches were generated by splitting the connected skeleton at all junctions. Finally, pairs of adjacent branches were joined to represent a single curve if the Weighted Extended Distance Function (WEDF) was continuous across the junction [3].

Bezier curve fitting: For degree five Bezier curves, given six control points $\{P_0,P_1,...,P_5\}$, a Bezier curve was calculated as

$$B(t) = \sum_{j=0}^{5} \binom{5}{j} P_j \left(1 - \frac{t}{N-1}\right)^{5-j} \frac{t}{N-1}^{j},$$

$$t \in [0, N-1]$$

Where N denotes the point number of the corresponding branch. P_j=[x_j,y_j,r_j] denotes the j-th control point.

The first control point (P_0) and the last control point (P_5) were directly set as the first point and the last point of the skeleton branch respectively. Other control points {P_1,P_2,P_3,P_4} were calculated by minimizing the following fitting loss.

$$L = \sum_{t=0}^{N-1} \|B(t) - L(t)\|_2$$

Where $L(t)$=[x_t,y_t,r_t] denotes the t-th point of the skeleton branch.

Branch ordering: All branches were ordered by their importances to have canonical representations, and the branch importance was estimated by the maximal WEDF value of the points in the branch [3]. The skeleton branches were sorted by their part areas calculated as the following formula [7].

$$S = \frac{1}{2} \sum_{t=0}^{N-2} (r_t + r_{t+1}) * \|[x_t - x_{t+1},\ y_t - y_{t+1}]\|_2$$

3. Experimental results

The proposed U-Net model was trained on the Pixel SkelNetOn training set, and evaluated on the Pixel SkelNetOn, the Point SkelNetOn and the Parametric SkelNetOn validation sets.

Pixel SkelNetOn consists of 1,725 black and white images given in portable network graphics format with size 256 × 256 pixels, split into 1,218 training images, 241 validation images, and 266 test images.

Point SkelNetOn consists of 1,725 shape point clouds and corresponding ground truth skeleton point clouds, given in the basic point cloud export format .pts, split into 1,218 training point clouds, 241 validation point clouds, and 266 test point clouds.

Parametric SkelNetOn 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.

3.1. Implementation

To train the model we randomly split the Pixel SkelNetOn training set on train and cross-validation in proportion 80% : 20%. The cross-validation set was used for reducing learning rate and saving the best model checkpoints. Adam optimizer was used and F1-score was used to evaluate the model performance.

Similar to [11], the weighted focal loss was used in model training. We found that wpos = 50 and wneg = 1.0 along with = 2 gave the best results. The data augmentation was implemented by rotations on 90 degrees and horizontal flips of input images and targets. The model training learning rate was initialized with 0.001 and reduced after 50 epochs to 10% if validation loss did not improve. The batch size was set to 4. The total epochs were set to 1000.

3.2. Results

In the original U-Net architecture, each down sampling block and up sampling block used two convolution as basic process unit. These basic process units could be replaced by other operations, such as Resnet blocks and Densenet blocks. We compared the performances of different types of basic process units. All experiments used the SDE and Edge as input and the results are shown in Table 1. A sequence of three Resnet blocks achieves better F1-score compared with convolutions and ResDense blocks.

Table 1. The result of comparison between different types of basic process unit on the Pixel SkelNetOn validation set.

Basic process unit	F1-score
Convolution	0.7787
ResDense Block[19]	0.7963
Resnet Block	0.8071

We compared the performances of different types of input datas. The experiment results are shown in Table 2. All experiments use Resnet Block as the basic process unit. Using the SDE and Edge as the model input achieves the best F1-score compared with the shape and hard distance. By using a multi-model ensemble, the F1 score is further improved to 0.8129.

Samples of the pixel skeleton, point skeleton and parametric skeleton results are displayed in Figure 5, 6, 7 respectively. Our U-Net based model achieved 0.8129 F1 score in the Pixel SkelNetOn validation set, 1.5752 symmetric chamfer distance [3] in the Point SkelNetOn validation set and 6407.4 squared distance score [3] in the Parametric SkelNetOn validation set.

Table 2. Comparison of different input datas on the Pixel SkelNetOn validation set.

Input data	F1-score
Shape	0.7593
Hard distance [3]	0.7787
Hard distance + Edge	0.7886
SDE + Edge	0.8071
SDE + Edge (Multi-model ensemble)	0.8129

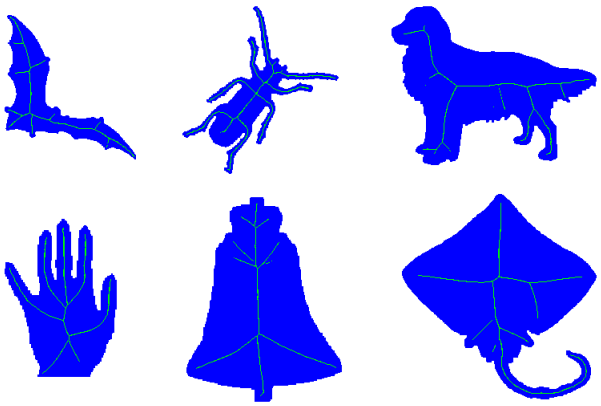

Figure 5. Samples of the pixel skeleton classification results. We achieved 0.8129 F1 score in the Pixel SkelNetOn validation set. (Best viewed in color)

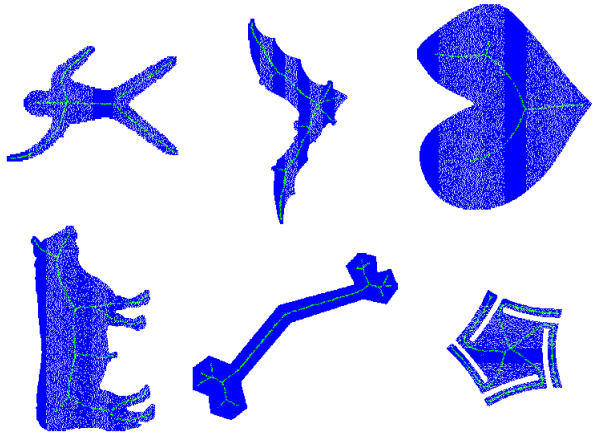

Figure 6. Samples of the point skeleton extraction results. We achieved 1.5752 symmetric chamfer distance in the Point SkelNetOn validation set. (Best viewed in color)

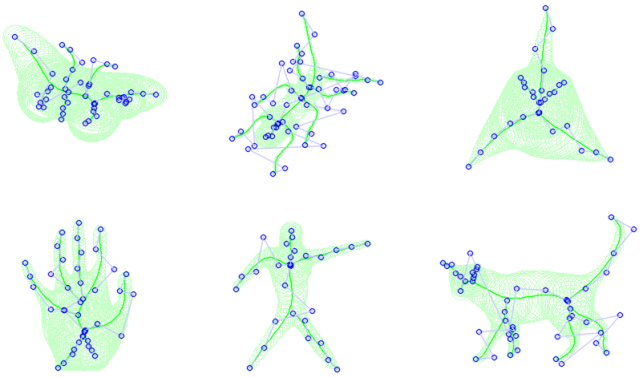

Figure 7. Samples of the parametric skeleton generation results. The skeleton curves are displayed in green and the Bezier control points are displayed in blue. We achieved 6407.4 squared distance score in the Parametric SkelNetOn validation set. (Best viewed in color)

4. Conclusion

In this paper, we proposed to apply the Smooth Distance Estimation (SDE) and Edge transformation to preprocess the input shape, and designed a modified U-Net architecture for skeleton extraction. Experiment results showed that the proposed method got excellent performances in the pixel SkelNetOn, point SkelNetOn and parametric SkelNetOn validation set.

References

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

[2] Alexander M Bronstein, Michael M Bronstein, Alfred M Bruckstein, and Ron Kimmel. Analysis of two-dimensional non-rigid shapes. *International Journal of Computer Vision*, 78(1):67–88, 2008.

[3] Ilke Demir, Camilla Hahn, Kathryn Leonard, Geraldine Morin, Dana Rahbani, Athina Panotopoulou, Amelie Fondevilla, Elena Balashova, Bastien Durix, and Adam Kortylewski. Skelneton 2019: Dataset and challenge on deep learning for geometric shape understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[4] Bastien Durix, Sylvie Chambon, Kathryn Leonard, Jean-Luc Mari, and Géraldine Morin. The propagated skeleton: A robust detail-preserving approach. In *International Conference on Discrete Geometry for Computer Imagery*, pages 343–354. Springer, 2019.

[5] Nan Jiang, Yifei Zhang, Dezhao Luo, Chang Liu, Yu Zhou, and Zhenjun Han. Feature hourglass network for skeleton detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[6] Kathryn Leonard, Geraldine Morin, Stefanie Hahmann, and Axel Carlier. A 2d shape structure for decomposition and part similarity. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 3216–3221. IEEE, 2016.

[7] Chang Liu, Dezhao Luo, Yifei Zhang, Wei Ke, Fang Wan, and Qixiang Ye. Parametric skeleton generation via gaussian mixture models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[8] KM Meiburger, SY Nam, E Chung, LJ Suggs, SY Emelianov, and F Molinari. Skeletonization algorithm-based blood vessel quantification using in vivo 3d photoacoustic imaging. *Physics in Medicine & Biology*, 61(22):7994, 2016.

[9] Sabari Nathan and Priya Kansal. Skeletonnet: Shape pixel to skeleton pixel. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[10] Lenuta Pana, Simona Moldovanu, Nilanjan Dey, Amira S Ashour, and Luminita Moraru. Brain tissue evaluation based on skeleton shape and similarity analysis between hemispheres. *Computation*, 8(2):31, 2020.

[11] Oleg Panichev and Alona Voloshyna. U-net based convolutional neural network for skeleton extraction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[13] Punam K Saha, Gunilla Borgefors, and Gabriella Sanniti di Baja. A survey on skeletonization algorithms and their applications. *Pattern recognition letters*, 76:3–12, 2016.

[14] Thomas B Sebastian, Philip N Klein, and Benjamin B Kimia. Recognition of shapes by editing their shock graphs. *IEEE Transactions on pattern analysis and machine intelligence*, 26(5):550–571, 2004.

[15] Dirk Selle, Bernhard Preim, Andrea Schenk, and H-O Peitgen. Analysis of vasculature for liver surgical planning. *IEEE transactions on medical imaging*, 21(11):1344–1357, 2002.

[16] Kaleem Siddiqi, Ali Shokoufandeh, Sven J Dickinson, and Steven W Zucker. Shock graphs and shape matching. *International Journal of Computer Vision*, 35(1):13–32, 1999.

[17] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 4724–4732, 2016.

[18] Cong Yang, Oliver Tiebe, Kimiaki Shirahama, and Marcin Grzegorzek. Object matching with hierarchical skeletons. *Pattern Recognition*, 55:183–197, 2016.

[19] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2472–2481, 2018.