International Journal of Computer Vision (2024) 132:1219–1241
https://doi.org/10.1007/s11263-023-01926-3

Skeleton Ground Truth Extraction: Methodology, Annotation Tool and Benchmarks

Cong Yang[1] · Bipin Indurkhya[2] · John See[3] · Bo Gao[4] · Yan Ke[4] · Zeyd Boukhers[5] · Zhenyu Yang[6] · Marcin Grzegorzek[7]

Received: 23 June 2022 / Accepted: 8 October 2023 / Published online: 1 November 2023

Abstract

Skeleton Ground Truth (GT) is critical to the success of supervised skeleton extraction methods, especially with the popularity of deep learning techniques. Furthermore, we see skeleton GTs used not only for training skeleton detectors with Convolutional Neural Networks (CNN), but also for evaluating skeleton-related pruning and matching algorithms. However, most existing shape and image datasets suffer from the lack of skeleton GT and inconsistency of GT standards. As a result, it is difficult to evaluate and reproduce CNN-based skeleton detectors and algorithms on a fair basis. In this paper, we present a heuristic strategy for object skeleton GT extraction in binary shapes and natural images. Our strategy is built on an extended theory of diagnosticity hypothesis, which enables encoding human-in-the-loop GT extraction based on clues from the target's context, simplicity, and completeness. Using this strategy, we developed a tool, SkeView, to generate skeleton GT of 17 existing shape and image datasets. The GTs are then structurally evaluated with representative methods to build viable baselines for fair comparisons. Experiments demonstrate that GTs generated by our strategy yield promising quality with respect to standard consistency, and also provide a balance between simplicity and completeness.

1 Introduction

Skeleton Ground Truth (GT) is critical to the success of supervised skeleton extraction in binary shapes (Panichev et al., 2019) and natural images (Wang et al., 2019) (hereafter referred to as "shape" and "image", respectively, see Fig. 1a, b). A number of modern skeleton detectors, i.e. AdaLSN (Liu et al., 2021) and SkeletonNetV2 (Nathan and Kansal, 2021), are based on Convolutional Neural Networks (CNN), which are trained using skeleton GTs from image and shape datasets, respectively. Moreover, skeleton GT is important to facilitate skeleton-related algorithms such as pruning (Bai et al., 2007), matching (Bai and Latecki, 2008), and classification (Bai et al., 2009). In addition to skeletonization with morphological and geometrical operations (Giesen et al., 2009; Ge and Fitzpatrick, 1996; Jalba et al., 2015; Liu et al., 2011; Telea and Wijk, 2002; Zhang and Suen, 1984), skeleton GT extraction should also meet the eye-level view assumption (Firestone and Scholl, 2014) of skeleton simplicity and completeness in different domains. For clarity in terminology, commonly used skeleton components (Bai and Latecki, 2008; Bai et al., 2007; Cornea et al., 2007) and expressions (Shen et al., 2013) are defined (see Fig. 1c, d):

- Endpoint: a skeleton point with only one adjacent point.
- Junction point: a skeleton point with three or more adjacent points.
- Connection point: a skeleton point that is neither an endpoint nor a junction point.
- Skeleton branch: a sequence of connection points within two directly connected skeleton points.
- Skeleton simplicity: higher skeleton simplicity means simpler skeleton structure, e.g. minimal number of branches.

Communicated by Oliver Zendel.

✉ Cong Yang
cong.yang@suda.edu.cn

1 Soochow University, Suzhou, China
2 Jagiellonian University, Cracow, Poland
3 Heriot-Watt University (Malaysia), Putrajaya, Malaysia
4 Clobotics, Shanghai, China
5 Fraunhofer FIT, Sankt Augustin, Germany
6 Southeast University, Nanjing, China
7 University of Lübeck, Lübeck, Germany

- Skeleton completeness: higher completeness means a finer-grained representation of object features, e.g. small branches correlated to shape boundary perturbations.

As presented in Fig. 2, the requirement of complexity is different in real-world applications (Saha et al., 2016). For instance, in the scenario of farmland ridge detection for agricultural robot navigation, the ridge skeletons are relatively simple and close to curves). Differently, plant root skeletons are primarily complex, thereby preserving root hair and other details. To properly encode such requirement, skeleton GT extraction is normally addressed by a human-in-the-loop fashion (Ilke et al., 2019). Particularly, an optimal skeleton GT requires a trade-off between its simplicity and completeness. Thus, following the convention in (Bai et al., 2007; Firestone and Scholl, 2014; Lowet et al., 2018; Shen et al., 2013; Yang et al., 2016), skeleton GT is a satisfaction of the branch simplicity between domain requirements and human perception. An intuitive explanation of such trade-off is that a skeleton GT should satisfy the requirement of simplicity in various domains, while including a proper number of desirable branches (aka. completeness) to preserve object geometrical features. Otherwise, for instance, a skeleton with over-detailed branches could lead to a higher cost on computation and an occurrence of over-fitting problems on matching (Bai and Latecki, 2008). However, one crucial limitation in existing skeleton GTs lies in the *lack of clarity* and *inconsistency of standards*.

Lack of clarity: Skeleton GTs of most existing shape datasets are unclear. As presented in Table 1, only two (SkelNetOn (Ilke et al., 2019) and WH-SYMMAX (Shen et al., 2016b)) of thirteen actively used shape datasets have publicly available skeleton GTs, though SkelNetOn is only accessible to the registered participates of the SkelNetOn Challenge (Ilke et al., 2019). For image datasets, skeleton GTs are semi-automatically extracted by object segmentation and skeletonization approaches (Durix et al., 2019; Shen et al., 2011). However, it is unclear whether humans have a similar and stable perception on simplicity and completeness, especially under different contexts from object foreground, background and shape. Context usually refers to the source of contextual associations to be exploited by the visual system (Oliva and Torralba, 2007). A natural way of representing the context of an object is in terms of its relationship to other objects. In our case, object context is defined as an object's foreground, background, and shape, which are primarily associated with the object skeleton. Theoretically, shape is part of the information in the foreground, while we can easily extract shape by binarizing and filling an object's foreground. Here, we denote shape as an independent context since ten datasets contain only shapes without foreground (see Table 1). In short, there are two uncertainties: (1) it is unclear whether humans have a similar and stable perception of skeleton simplicity and completeness, and (2) it is unclear whether such a perception could be influenced by object foreground and background. Such uncertainties were not structurally studied in existing literature. They can have a tremendous impact on training CNN-based skeleton detectors, making it difficult to compare different skeleton-related algorithms.

Inconsistency of standards: We observe glaring inconsistencies among various existing GTs: (1) GT skeletons among existing shape datasets are not always the same. For example, in Fig. 3a, the main skeleton branches are shortened whereas some spurious skeleton branches remain in the mouth, neck and hind leg regions. In contrast, in another dataset shown in Fig. 3b, only the main branches (not shortened ones) are preserved. (2) GT skeletons from existing image datasets are not consistent. We can clearly see that the GT skeletons in Fig. 3d are in discrete segments, rather than a single connected medial-axis as in Fig. 3c. Skeleton GT in Fig. 3e is not accurate. (3) GT skeletons of the shape and the image datasets are not always consistent (Fig. 3b, c). Although the main skeleton branches are preserved in both horses, skeleton branches in (c) are shortened. Typically, the shortening of branches may cause blurring between the branches of significant visual parts and branches resulting from noise (Bai et al., 2007). To sum up, the standards on GT structure (simplicity, completeness, connectivities to branch and boundary) are not consistent. As a result, evaluating skeleton-related pruning, matching and classification approaches with inconsistent GT is an ill-posed problem.

In this paper, we introduce an annotation tool, SkeView, for skeleton GT extraction in image and shape datasets. To do so, we first report an empirical study of human perception on skeleton structure based on the theory of diagnosticity hypothesis (Tversky, 1977). Diagnosticity hypothesis aims to capture the effect of context on target similarity from the perspective of human perception. In our case, exploring human perception on skeleton structure by varying the object context (foreground, background, and shape), time, and participants. Based on these studies, we introduce a general strategy for extracting skeleton GT in image and shape datasets. Our proposed strategy is able to encode human-in-the-loop GT extraction based on clues from the target context, simplicity and completeness. Using this strategy, SkeView is designed and developed to generate skeleton GTs for existing datasets including those in Table 1. Our generated GTs have consistent standards, and properly represent the object geometrical and topological features. These aspects provide a reliable benchmark for assessment. Thus, we can systematically evaluate representative methods using our GTs on skeleton detectors and skeleton-based algorithms, and generate viable baselines for the community.

It should be emphasized that introducing a new skeletonization method is not the focus of this paper, though

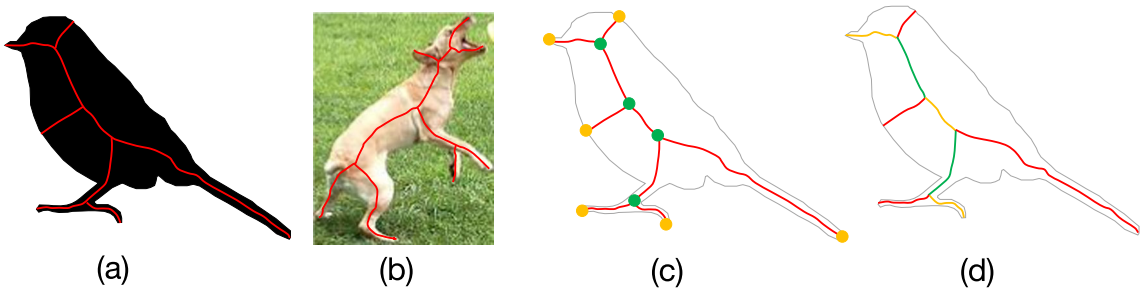

Fig. 1 Definition of skeletons and components in a higher level: **a** in binary shape, **b** in natural image, **c** endpoints (orange) and junction points (green), **d** skeleton branches with different colours (Color figure online)

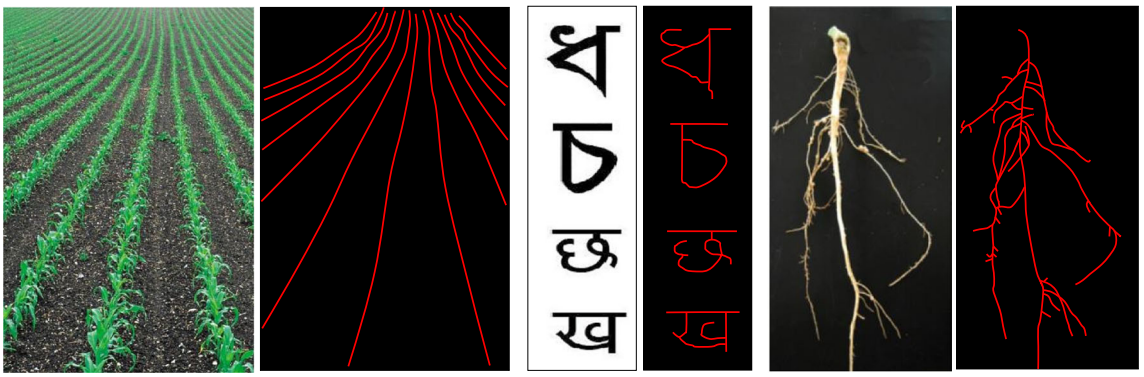

Fig. 2 Object skeletons (from simple to complex) in various applications: **a** farmland ridge detection for agricultural robot navigation (Li and Qu, 2018; Shokouh et al., 2021), **b** character recognition (Bag et al., 2011; Zhang et al., 2015), and **c** plant analysis (Bucksch, 2014; Sharma et al., 2021)

Table 1 Comparison of skeleton GT in actively used shape and image datasets

Dataset	Type	Size	GT	Dataset	Type	Size	GT
Animal2000 (Bai et al., 2009)	Shape	2000	×	ArticulatedShapes (Ling and Jacobs, 2007)	Shape	40	×
SkelNetOn (Ilke et al., 2019)	Shape	1725	√	Kimia99 (Sebastian et al., 2004)	Shape	99	×
Kimia216 (Sebastian et al., 2004)	Shape	216	×	MPEG7 (Latecki et al., 2000)	Shape	1400	×
MPEG400 (Yang et al., 2014)	Shape	400	×	SwedishLeaves (Söderkvist, 2001)	Shape	1125	×
Tari56 (Asian and Tari, 2005)	Shape	56	×	Tetrapod120 (Yang et al., 2016)	Shape	120	×
SK506 (Shen et al., 2016a)	Image	506	√	SK1491 (Shen et al., 2017)	Image	1491	√
SYMMAX300 (Tsogkas and Kokkinos, 2012)	Image	300	√	SymPASCAL (Ke et al., 2017)	Image	1435	√
EM200 (Yang et al., 2014)	S&I	200	×	SmithsonianLeaves (Ling and Jacobs, 2007)	S&I	343	×
WH-SYMMAX (Shen et al., 2016b)	S&I	328	√	**Our**	**S&I**	All	√

S&I: Shape and Image. √ (Yes) and × (No) denote whether skeleton GT of the full dataset is public available. The size column detail the number of images in each dataset

SkeView can be extended for this purpose. This is because our proposed strategy is applied semi-automatically, and therefore is not suitable for real-time (or quasi real-time) skeleton extraction in various applications. Moreover, desirable properties of skeletons have been well-defined (in 2D at least) via Blum Transform (Blum, 1967), discontinuities of the Distance Transform (Ge and Fitzpatrick, 1996), and many other equivalent definitions from Ogniewicz and Ilg (1992), Telea and Wijk (2002), Latecki et al. (2000), Bai et al. (2007) and Cornea et al. (2007), etc. Therefore, in this paper, we underscore the suitability of SkeView for training and testing data extraction of skeleton GTs, especially in this era of deep learning. Moreover, skeletons and GTs can be defined in general on a higher level, while it is not possible to find a general definition on a lower level, particularly towards various applications. This is because different applications may have different requirements on skeleton properties (e.g., 2D, 3D, and simplicity). Thus, we also underscore the generaliza-

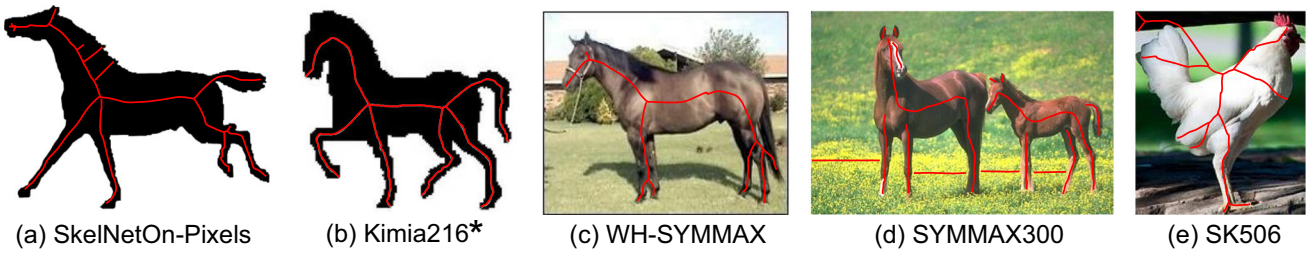

Fig. 3 Skeleton GTs in shape (SkeNetOn (Ilke et al., 2019) and Kimia216 (Sebastian et al., 2004)) and image (WH-SYMMAX (Shen et al., 2016b), SYMMAX300 (Tsogkas and Kokkinos, 2012) and SK506 (Shen et al., 2016a)) datasets. ⋆: since there is no public available GT for Kimia216, we take the optimal pruning result of a horse shape presented in Bai et al. (2007) for comparison

tion of our GTs (see Sect. 4.2) to specific vision tasks in the original datasets, such as skeleton detector training, skeleton matching and shape retrieval, etc.

Succinctly, the main contribution is that we introduce a general strategy to extract skeleton GTs in shape and image datasets. Our strategy meaningfully considers human perception on skeleton simplicity and completeness to adopt various requirements for real-world applications. We present a tool, SkeView, which utilises the proposed methodology to generate skeleton GTs in image and shape datasets. This contributes towards facilitating practical applications and proper benchmarking in future. We also generate skeleton GTs for 17 actively used datasets in Table 1 to build new baselines on a consistent and standardized manner. Our comprehensive evaluation demonstrates the efficacy of SkeView, highlighting the need for a new perspective for CNN-based skeleton detectors to become practically relevant and feasible.

2 Related Works

We present here a brief overview of several existing methods that were proposed for extracting skeleton GTs. For a more thorough treatment on skeletonization methods, compilations by Saha et al. (2016), Tagliasacchi et al. (2016) and Liu et al. (2011) offer sufficiently good reviews.

2.1 GT in Shape Datasets

Figure 4 presents existing approaches that could be applied for skeleton GT extraction in shape datasets. As mentioned in Sect. 1, these methods are normally applied semi-automatically to meet human perception on complexity. Otherwise, the extracted GTs are too simple or contain redundant small branches. For instance, Bai et al. (2007) requires a stop parameter k to control the simplicity of skeleton structures. If k is fixed without manual calibration, redundant small branches are not removed completely in simple shapes, e.g. the GT in Fig. 4a with a fixed $k = 30$. In contrast, the GT in Fig. 4b is extracted based on an optimized k in SkeView. We can clearly see that it is more perception friendly in terms of balancing the skeleton simplicity and completeness.

In contrast to semi-automatic approaches, purely manual GT extraction is conducted with more user interaction, typically using a variety of tools. As shown in Fig. 4c, Firestone and Scholl (2014) developed an application for a touch-sensitive tablet computer to display single closed geometric shapes, thereby collecting touch data from the participants. Each participate could tap on the displayed shapes anywhere they wished. The collection of their tapped locations provide a global representation of the crowd-sourced perception of major skeletons (aka. GTs). Instead of generating skeletons from scratch, Yang et al. (2016) generated a set of GT candidates with different complexity, and then applied a voting scheme based on questionnaires. Each participant was provided with three candidates in a questionnaire, and was asked to select the most promising one, or to draw a new one. Though both these manual approaches can capture crowd-sourced perceptions on skeleton complexity in a proper manner, they are not efficient enough for datasets with a massive number of shapes. Unlike the purely manual approaches, our proposed strategy is more efficient as it generates GT via SkeView semi-automatically and in parallel.

2.2 GT in Image Datasets

In practice, GTs in image datasets are extracted semi-automatically via two steps: segmentation and skeletonisation. The segmentation step is mostly applied manually. For instance, in the SYMMAX300 (Tsogkas and Kokkinos, 2012) dataset, each image was accompanied by 5-7 human segmentations. Thus, multiple binary objects can be obtained for the followed skeletonisation and integration. Although purely manual segmentation can properly ensure the integrity of objects while reducing boundary noises, it is not efficient enough to be applied in practice, particularly preparing massive skeleton GTs for training scenarios. In terms of the skeletonisation step, some existing shape skeleton extrac-

Fig. 4 Skeleton GT extraction in shape dataset: **a** automatically with a fixed pruning power (Bai et al., 2007). **b** semi-automatically with a manual optimized pruning power in SkeView. **c** Purely manual via shape tapping (Firestone and Scholl, 2014)

tion approaches (Bai et al., 2007; Shen et al., 2011; Telea and Wijk, 2002) are applied semi-automatically on the shape of segmented objects. As shown in Fig. 3, these skeleton extraction approaches have different preferences on skeleton geometry and topology. Moreover, it is not clear whether humans have a similar and stable perception of skeleton complexity under different contexts. As a result, skeleton GTs in the existing image datasets are not very consistent (see Fig. 3c, d, e).

In contrast, our proposed method is better in terms of efficiency and consistency. Specifically, our strategy is more general and standardized, as it is built on a structural study of human perception on skeleton GT. Besides, SkeView has an easy-to-use user interface, and a set of convenient functions to improve the efficiency of GT extraction in both shapes and images.

3 Methodology

Here, we first present a study of human perception of skeleton structure based on the theory of diagnosticity hypothesis (Tversky, 1977). Based on these observations, we introduce a strategy for skeleton GT extraction in the shape and the image datasets.

3.1 Diagnosticity Hypothesis

The diagnosticity hypothesis is a classic framework to explore the relation between similarity and context (or grouping) in the domain of cognitive science (Skov and Sherman, 1986). Specifically, the diagnosticity hypothesis implies that the change in context, induced by the substitution of an odd element, will change the similarities in a predictable manner. An example is shown in Fig. 5: consider two sets of four countries, which differ in only one of their elements (p and q). The four countries of each set were presented to participants, who were instructed to select the country most similar to Austria (a). Note that this experiment was done in the 1970s, so one has to remember the political map of Europe at that

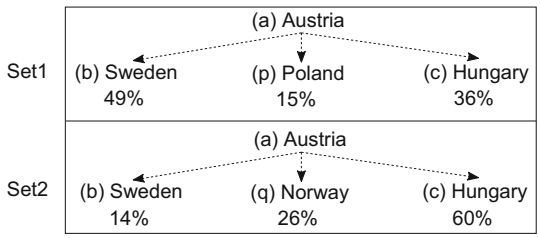

Fig. 5 An example of diagnosticity hypothesis (Tversky, 1977). The percentage of participants who selected each country (as most similar to Austria) is presented below the name

time. The final statistical results are shown in percentages. It is interesting to observe that the selection results in Set 1 and Set 2 are different (Austria (a) is grouped with Sweden (b) in Set 1, and with Hungary (c) in Set 2) by changing only one element (p to q), though both (p) and (q) are not the final results. The diagnosticity hypothesis example in Fig. 5 demonstrates that human perception of selection (a country most similar to Austria) could be influenced by a change of context (from Poland to Norway). In our case, human perception of selection (a branch to prune) could be affected by the shift in object contexts, such as shape, foreground, and background.

Accordingly, our study was conducted by evaluating the robustness of human perception on skeletons spatially and temporally. In other words, (1) perception of an object skeleton in the context of object shape, segmented foreground and full image, (2) perception of an object skeleton in different time slots, and (3) perception of an object skeleton by different volunteers. Thus, our study is an extension of diagnosticity hypothesis: verifying whether a skeleton GT could be robust for different people, at different times, and in different contexts. Due to the limitations of face-to-face surveying during the global pandemic (Fanelli and Piazza, 2020), we developed a phone application (APP) to collect perceptions from different participants, as presented in Fig. 6. Our APP contained four major components: a counter showing processed/remaining images (top right), a setting panel for the boundary, colour and transparency (top left), a selection area

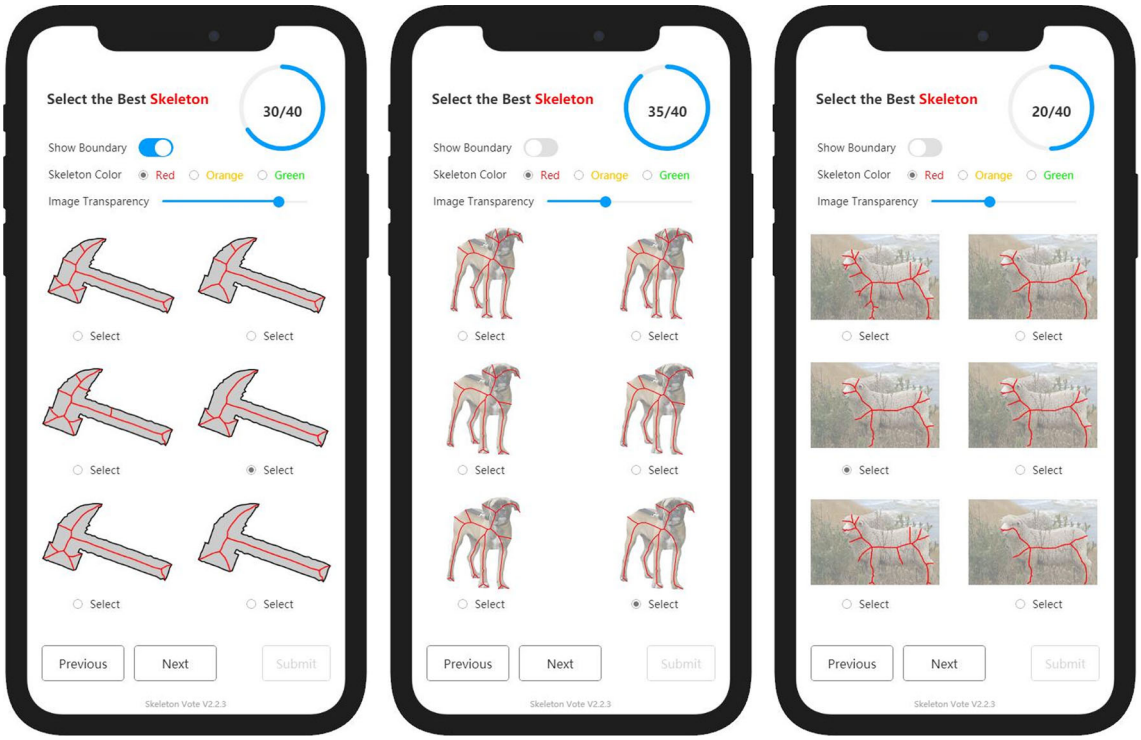

Fig. 6 Interfaces of our APP for skeleton selection (best viewed in color)

for the skeletons (middle), and buttons for page navigation and submission (bottom). In total, 90 volunteers (45 females, 45 males) participated in the study (January to March, 2021), most of whom were students and teachers from Northeast Normal University (NENU), China.

We randomly selected 30 images from the existing datasets in Table 1, and applied manual segmentation and semi-automatic skeletonization with the method introduced in Bai et al. (2007). For some images with complex backgrounds, we intentionally generate two segmented samples, a promising one and a noisy one, for comparison. We generated six skeleton candidates for each shape with different levels of complexity, resulting in a total of 40×6 skeletons. To reduce the influence of context from different formats, we organized our volunteers into three groups (30 in each group) and presented object shapes, foregrounds and full images to each group independently. To facilitate the study (see Table 2), we repeated the survey every two weeks so that the effect of context memorisation could be reduced. For each trial, the skeleton format was changed in each group so that the three formats could be fully surveyed from all groups. We also conducted an additional survey seven weeks later, using the format of the first survey, to measure the stability of results with respect to time passing.

For quantitative analysis, the number of endpoints (more branches implies more endpoints) is used in our study. It should be noted that skeleton simplicity (see Eq. 5 in Sect. 4.2.4) can also be used for the quantitative analysis. Particularly, it has higher discriminative power than the number of endpoints. Here, we employed the number of endpoints in Table 2 for two reasons: (1) The differences between manually voted skeletons from Shape, Object, and Image are distinct, e.g., 378, 326, and 314, respectively. Thus, endpoint statistics are already enough to tell the difference at the coarse-grained level. (2) It is easier to count and visually recheck, particularly in our user study scenario using the questionnaire in APP. Based on the statistics shown in Table 2, we found that the number of endpoints in shapes, foregrounds and full images ("shape", "object", "image") are within [373, 380], [322, 330] and [314, 320], respectively. In other words, each group has a rather consistent perception on skeleton structure, with differences of only about 2%. However, as shown in Fig. 7, individual perception are varied, ranging from 365 to 385 for shapes, 323 to 338 for objects and 310 to 332 for images. For instance, ID 27 prefers concise skeletons while the perception of IDs 11 and 28 are erratic. We believe the idea of group integration (Tsogkas and Kokkinos, 2012; Yang et al., 2016) produces a more consistent performance than the individual scheme in (Ke et al., 2017; Shen et al., 2016b, a, 2017). As the endpoint numbers on January 21 and March 04 were almost the same, we can assume that the human perception of skeleton structure is stable over time. Considering the mean values of shape (377) vs. object (326), we find that the foreground context has a con-

Table 2 Statistics of total endpoint numbers from the most voted skeletons

Date	Group1			Group2			Group3		
	Shape	Object	Image	Shape	Object	Image	Shape	Object	Image
Jan 21	378	–	–	–	330	–	–	–	315
Feb 04	–	326	–	–	–	320	380	–	–
Feb 18	–	–	314	373	–	–	–	322	–
Mar 04	378	–	–	–	330	–	–	–	315

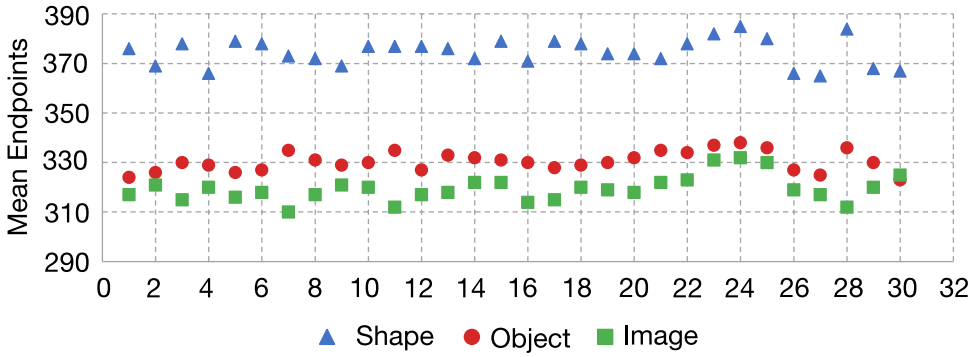

Fig. 7 Comparison of each participant in Group2. The participant IDs are shown on the horizontal axis (1–30) (best viewed in color)

siderable influence on human perception, with about 13.5% reduction from shape to object formats. However, the difference between object (326) and image (316) is less obvious, with only about 3.1% reduction.

To better understand these results, the most voted skeletons of a sheep image are presented in Fig. 8a, together with its two segmentations (noisy (b) and good (d)) and their corresponding shapes. We intentionally eliminated the fore- and background of the object in (c) and (e) to reduce their context influence. The only difference is that Shape 1 contains noises in the top-left region (head and neck). We find that skeletons in (a), (b) and (d) are almost the same. This is understandable as illusions from the background and the boundary noise can be easily filtered by human inspection. However, as presented in (c) and (e), most volunteers tended to use more skeleton branches to fill their perceptual gaps on shapes (where there is less context information). In cognitive science, the perceptual gap (Teichmann et al., 2021) refers to cognitive biases from information gaps, such as occlusion (internal and external) and misunderstanding, etc. In our case, a perceptual gap occurs since it is difficult to identify the original object (a sheep or something else) from the noisy shape in (c). As a result, volunteers tend to use more skeleton branches to fill their perceptual gaps in this shape. For instance, it is difficult to identify the original object of Shape 1 in Fig. 8c, particularly at the head and neck regions. As a result, the skeleton in Shape (c) is erroneously more extensive than the ones in (b), (d) and (e). Overall, our observations can be summarized as follows:

- O1: Perception is robust to the time and volunteer groups.
- O2: Perception is robust to segmented objects and images.
- O3: Perception of shapes is not robust and is easily influenced by deformations from noises and occlusion.
- O4: People tend to use more skeleton branches when there exist perceptual gaps on shapes, and vice versa.

These four observations are used to design the strategy and Graphical User Interface (GUI) of the annotation tool for extracting the skeleton GT in the image and the shape datasets.

3.2 Strategy

Given an image $\mathbf{I}$, let $\mathbf{M}$ and $\widehat{\mathbf{M}}$ denote a segmented object and its shape, respectively. Let the final GT skeleton be $\mathbf{S}$. In brief, our GT extraction strategy is composed by two steps: preprocessing and pruning. The preprocessing step includes target object segmentation (for image datasets) and initial GT extraction in a coarse level. Then, a heuristic pruning process is conducted semi-automatically based on the above observations (O1-O4) and the human perception on simplicity and completeness.

Such coarse-to-fine strategy can inherently improve the efficiency of GT extraction, as most time-consuming operations are automatically applied in the first step. Specifically, based on the segmented $\widehat{\mathbf{M}}$ (Fig. 9b, c) with He et al. (2017), the skeletonization approach Shen et al. (2013) is employed for extracting the initial skeleton. This process effectively reduces the workload of the manual pruning that follows, as most of the redundant branches are removed in the initial skeleton. To bring more flexibility, we intentionally preserve

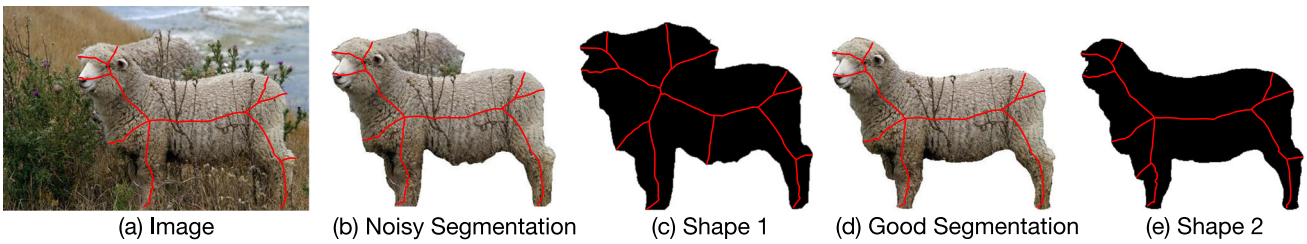

Fig. 8 Most selected skeletons of a sheep in the full image, two segmentations, and the correlated shapes, using our APP (best viewed in color)

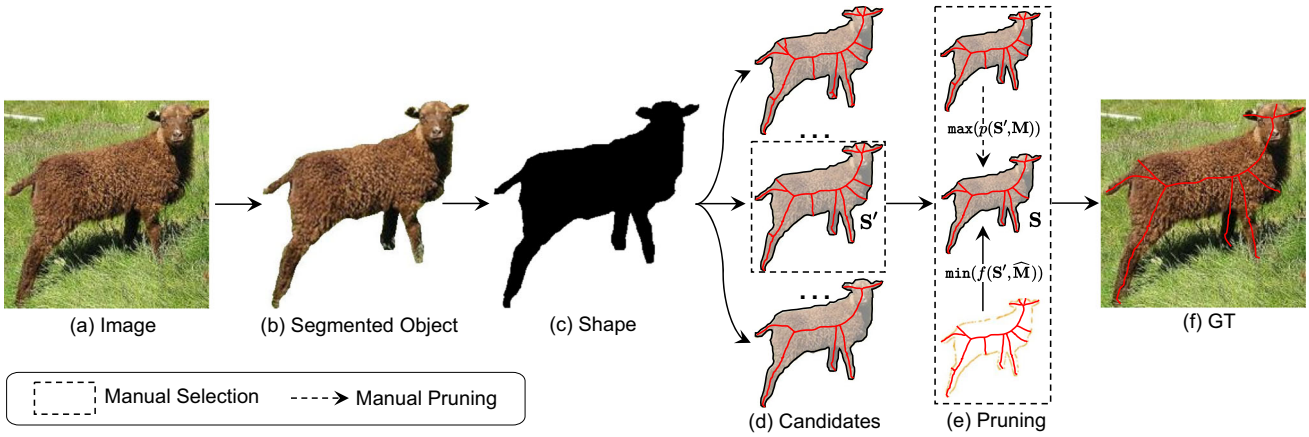

Fig. 9 Pipeline of our proposed skeleton GT generation strategy in the image scenario. $p(\cdot)$ and $f(\cdot)$ denote the satisfaction of human perception and shape reconstruction error, respectively (best viewed in color)

more branches than the optimal ones from the automatic approach to generate a set of candidates with different levels of complexity (Fig. 9d). $\mathbf{S}'$ denotes the selected initial skeleton from the candidates.

The second step of skeleton pruning is a heuristic and a semi-automatic process to identify the skeleton GT: that is, maximizing the (human) perceptual simplicity while keeping the skeleton as much complete as possible. For simplicity, Fig. 9d, e depicts how the skeletons appear during the selection and the pruning processes. This is motivated by our observations in O2 and O3. For completeness, inspired by Shen et al. (2013), we introduce a shape reconstruction error to represent the skeleton completeness: that is, keeping the reconstruction error of $\mathbf{S}$ to $\widehat{\mathbf{M}}$ as small as possible (Fig. 9e). Then, the skeleton GT $\mathbf{S}$ is extracted by:

$$\mathbf{S}_{image} = \max(p(\min(f(\mathbf{S}', \widehat{\mathbf{M}})), \mathbf{M})) \quad . \tag{1}$$

where $p(\cdot)$ and $f(\cdot)$, respectively, represent the (human) perceptual satisfaction and the shape reconstruction error. Thus, Eq. 1 is a semi-automatic annotation method as the variable $f(\cdot)$ is computed automatically and $p(\cdot)$ is determined by a human during manual pruning (i.e. manual selection of branch candidates for pruning). That is, calculating $f(\cdot)$ to inspire a human on trading-off the skeleton simplicity (domain requirement) and completeness ($f(\cdot)$ value). As a result, $p(\cdot)$ and $f(\cdot)$, respectively, are inherently maximized and minimized.

The rational behind Eq. 1 is that, as O4 suggests, people intend to use fewer branches (simple skeleton) on $\mathbf{I}$ and $\mathbf{M}$. This applies the diagnosticity hypothesis, whereby factors from other contexts (i.e. the reconstruction error) could potentially influence human perception. In practice, $p(\cdot)$ is maximized by dynamically selecting and pruning branches based on the eye-level view assumption of skeleton simplicity, and hints from the reconstruction error $f(\mathbf{S}', \widehat{\mathbf{M}})$:

$$f(\mathbf{S}', \widehat{\mathbf{M}}) = \frac{|\Lambda(\widehat{\mathbf{M}}) - \Lambda(R(\mathbf{S}'))|}{\Lambda(\widehat{\mathbf{M}})} \quad . \tag{2}$$

where $\Lambda(\cdot)$ denotes the area in terms of pixels, $R(\mathbf{S}')$ is the shape reconstructed from $\mathbf{S}'$:

$$R(\mathbf{S}') = \bigcup_{s \in \mathbf{S}'} B(s, r(s)) \quad . \tag{3}$$

where $r(s)$ is the radius of the maximal disc $B(s, r(s))$ centered at a point $s \in \mathbf{S}'$. In practice, $r(s)$ is approximated with the values of the distance transform at s. Motivated by the observation in O1, we suggest to conduct observations according to Eq. 1 by at least three participants, and heuristically take $\mathbf{S}_{image}$ to be the one with the maximum votes (when 2 skeletons are the same) or median reconstruction

error (when 3 skeletons are different). To promote the efficiency of the human-in-the-loop approach, we introduce a new tool, SkeView, in Sect. 4 with various functions for segmentation, initialization, pruning, and integration.

For the shape scenario, as presented in Fig. 10, our strategy is similar to the workflow from Fig. 9c–f. As there is no **M** displayed below the skeletons, only shape contour and the skeleton are fused in the illustration in Fig. 10c. Thus, shape skeleton GT is generated by:

$$\mathbf{S}_{shape} = \texttt{max}(p(\texttt{min}(f(\mathbf{S}', \widehat{\mathbf{M}})), \widehat{\mathbf{M}})) \quad . \tag{4}$$

where $p(\cdot)$ and $f(\cdot)$ are same to Eq. 1. An intuitive example is presented in Fig. 11. We can clearly observe the changes in simplicity (SS) and reconstruction error (RE) during the pruning process. With the hints from RE and SS, most volunteers tend to select the third one (marked by the rectangle) since it strikes the best balance, being structurally complete and relatively simple. As shown in Figs. 9f and10d, skeleton GT generated by our strategies are perception-friendly, while at the same time properly balancing the skeleton simplicity and the shape reconstruction error.

4 Annotation Tool and Ground Truth

In this section, we first introduce the design of an annotation tool, SkeView, based on our proposed strategy. Then, using SkeView, we generate GTs for the 17 existing datasets shown in Table 1.

4.1 SkeView

To facilitate the strategies in Eq. 1 and 4, we developed a tool, SkeView, for extracting skeleton GTs in shape and image datasets. The user interface contains five major panels (see Fig. 12):

(a) Source. SkeView supports five source data types including shape, image, object (segmented foreground) and skeleton (only for pruning-related operations).

(b) Operations. This includes image segmentation (only available for the "Image" format), initial skeleton generation/selection, and dynamic skeleton branch pruning. For instance, there are two modes to address segmentation: manual and semi-automatic. If the manual mode is selected, users can dynamically plot a polygon to crop the region-of-interest. Otherwise, a Mask RCNN model (He et al., 2017), pretrained with COCO dataset (Lin et al., 2014), is loaded to extract the initial segmentation masks. Then, the selected mask is transformed into a polygon by uniformly inserting interactive plots along the mask boundary. This way, each interactive plot can be manually moved to optimize the shape of the mask. For images with multiple objects (i.e. SYMMAX300 (Tsogkas and Kokkinos, 2012) and SymPASCAL (Ke et al., 2017)), users can flexibly add and remove targets via buttons.

For initial skeleton extraction, the automatic mode Shen et al. (2013) is selected by default. According to the proposed strategy in Sect. 3.2, we intentionally added slightly more branches in the initial skeleton to provide more flexibility in the following pruning step. SkeView also allows users to generate initial skeleton semi-automatically using the discrete curve evolution (DCE) method Bai et al. (2007) by varying the stop parameter k. Either way, users can coarsely add (or remove) skeleton branches by simply clicking on the "+" (or "-") buttons until the generated skeleton is satisfactory. SkeView preserves all branches in each step of the skeleton evolution from complex to simple in Shen et al. (2013) and Bai et al. (2007). This operation is functionally similar to the skeleton selection process in Fig. 9d. Finally, as presented in Fig. 12e, users can finely prune redundant branches by selecting a target branch (marked in yellow) and clicking the "Prune" button (or the "Delete" key).

(c) Exports. Each export format is a structure with multiple elements: "Skeleton" (skeleton binary matrix, list of endpoints and junction points), "Object" (segmented foreground, shape and boundary matrices) and "Thumb" (pure skeleton and preview images, as shown in (e)). SkeView also preserves the pruning parameters and the correlated skeletons for future domain mapping and learning.

(d) Reconstruction error. In this panel, current and historic reconstruction errors (Eq. 2) of each target are displayed during skeleton initialization and pruning. To facilitate comparison between the current and the previously pruned skeleton, the current reconstruction error is presented in bold font at the top right corner, and also plotted dynamically (as blue points) on the graph. Moreover, users can easily click a point to load the previous pruning result for visualization and reconsideration.

(e) Preview and branch selection. Users can preview images, segmented objects and initial skeletons in this panel. Similar to the APP in Fig. 6, the background transparency, skeleton colour and boundary visibility can be adjusted here. During the fine-grained pruning process, users can select multiple branches by clicking on the target while pressing the "Shift" key.

Tsogkas (2016) have introduced a tool with a user interface for annotating skeletons by manually drawing poly-lines. Besides being less efficient due to its purely manual operation, it also cannot ensure the symmetry of poly-lines according the 2D object contour. SkeView is advantageous in both these aspects. As SkeView is developed for individual users, we also provide a tool for skeleton integration and selection from a group of users (Fig. 12 (bottom left)). As presented in Fig. 13, skeletons from multiple users are presented together for final determination of the acceptable annotation.

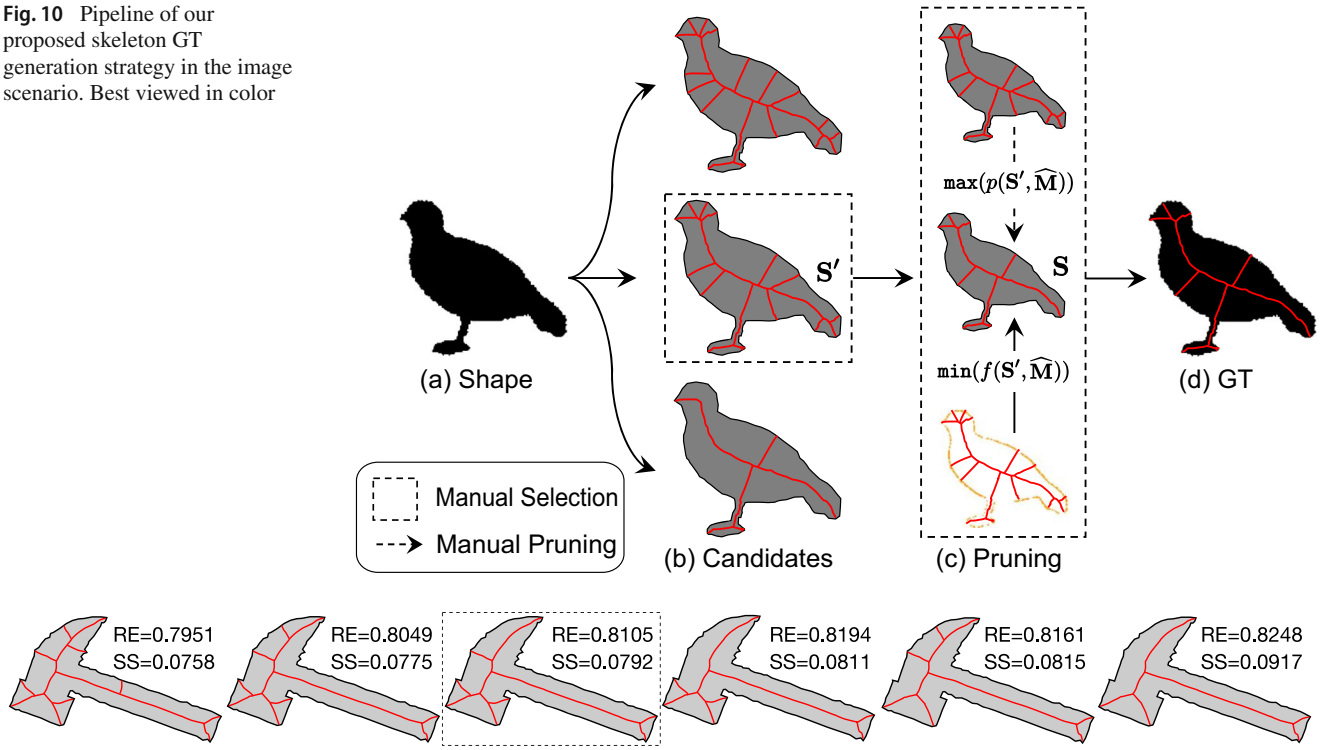

Fig. 10 Pipeline of our proposed skeleton GT generation strategy in the image scenario. Best viewed in color

Fig. 11 The changes in simplicity (SS) and reconstruction error (RE) during the pruning

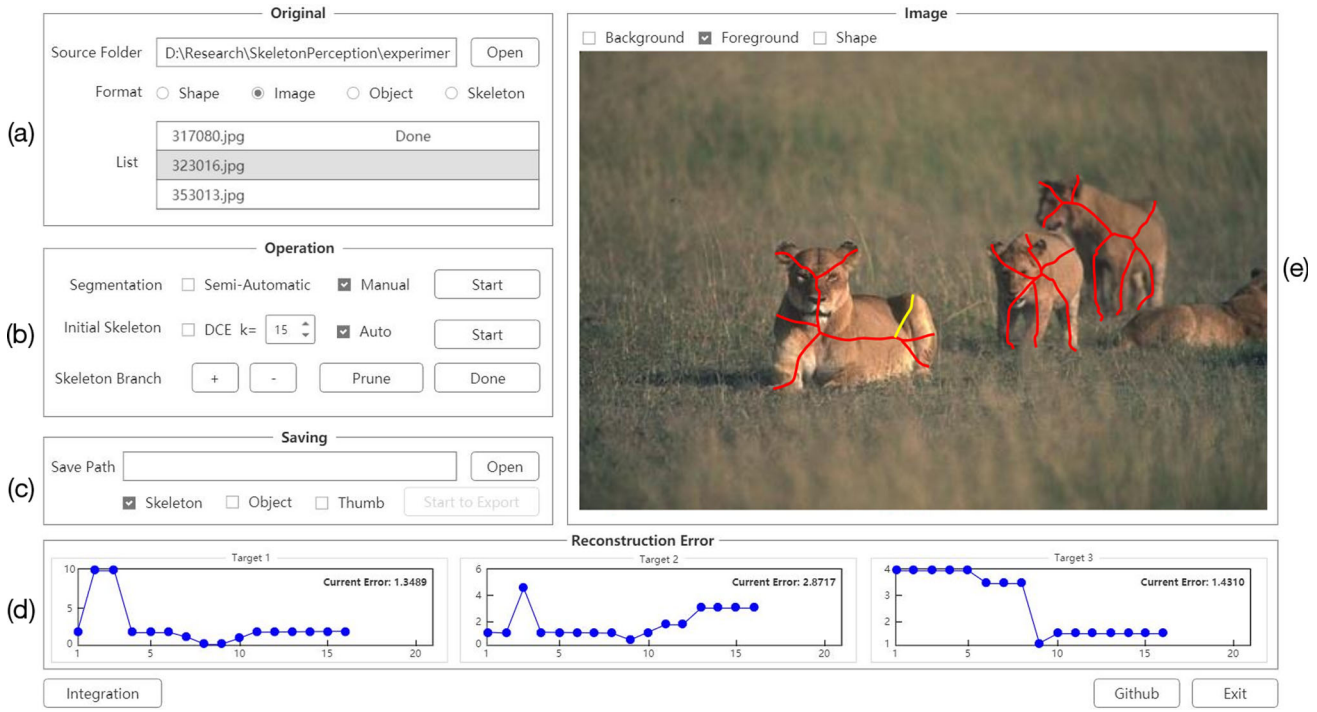

Fig. 12 User interface of SkeView. **a** Data and format selection. **b** Operations including segmentation, initial skeleton generation and pruning. **c** Result format and exporting. **d** Reconstruction error and log. **e** Preview of image, object, shape and their skeletons

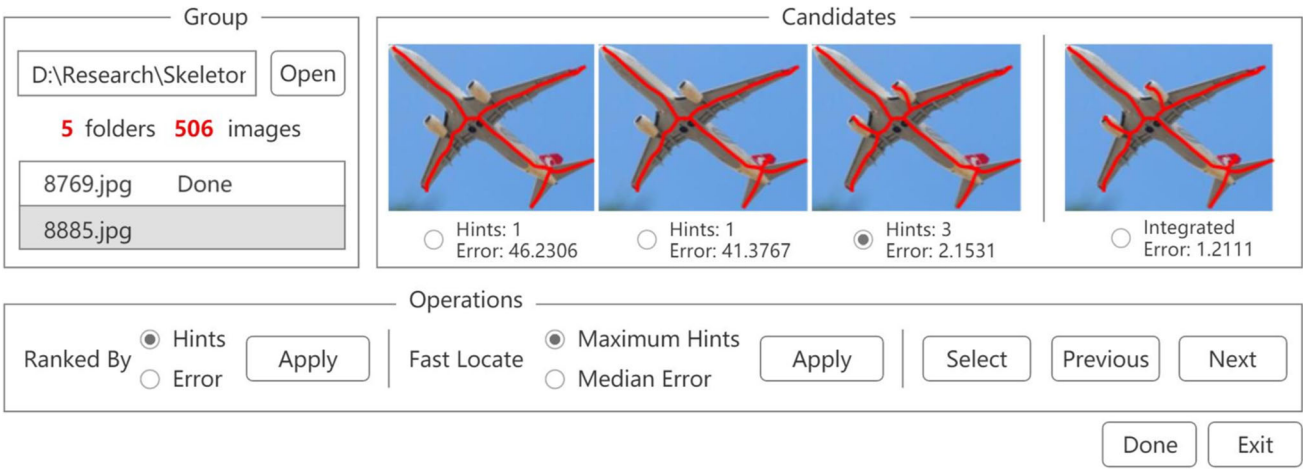

Fig. 13 User interface of the integration function

The tool can automatically count the duplicated skeletons ("Hints") and calculate reconstruction error ("Error"). By default, the final skeleton is automatically selected according to the maximum "Hints" and median "Error". For groups with fewer than three volunteers, SkeView integrates branches from the candidates to extract a new skeleton candidate. To evaluate the efficiency, we compare SkeView with the method in Shen et al. (2016a) on SK506 dataset. Our statistics show that the time cost per image is reduced from 86.4 to 27.2 s. This suggests that SkeView is suitable for medium-scale datasets which are mostly those listed in Table 1.

For large-scale datasets (e.g. more than 10,000 shapes), an efficient way is to generate initial skeletons using the SkeView semi-automatic method with big pruning power (e.g. $k = 50$). The pruning process is conducted via online labelling tools (such as LabelMe (Russell et al., 2008)) by drawing bounding boxes on the endpoints that are intended to preserve. The pruned skeleton is generated by mapping skeleton paths between the preserved endpoints to a zero matrix. Compared with the branch-based pruning in Fig. 12c, the box-based pruning only offers a slight consideration of the context arising from dense endpoints. However, it is more efficient for the purpose of group collaboration with its range of rich online annotation tools (Dasiopoulou et al., 2011). SkeView is developed with Matlab R2015b with GUIDE for user interface. The toolbox SkeView is compiled into executable applications in both Windows and Linux. We will be making SkeView (including source codes) publicly available to the developers and the research community.

4.2 Ground Truth

To ensure the quality of annotation, each GT was generated by four participants (two males and two females) from NENU. To meet different requirements in image and shape datasets, image GTs included segmented foregrounds, binary shapes, skeletons, lists of endpoints and junction points. Shape GTs included skeletons, and the list of endpoints and junction points. All skeleton branches in our GTs are one pixel wide, and are connected to shape boundaries: this meets the quality requirements of most skeleton extraction and matching algorithms. Users can intentionally dilate and dilute a GT skeleton point depending on algorithms (Atienza et al., 2019; Wang et al., 2019). In practice, there are two strategies to ensure the application requirements on GT properties:

- Annotation documents: written by domain experts, detail the annotation and quality requirements, including annotation examples and corner cases.
- Annotation training: annotators (volunteers in our case) study the annotation documents, followed by a trial-checking process using some samples.

Built on that, annotators not only follow their perception of skeleton simplicity and reconstruction error, but also consider the requirements of different domains. Besides, such strategies can ensure the quality and consistency of GTs. In our case, the extracted shape skeletons on the existing ten datasets are general enough for CNN-based skeleton detector training and skeleton matching. This is because these datasets were typically collected for the shape retrieval scenario. In terms of the four image datasets, the datasets are used for general object detection and analysis. Thus, our GTs not only respect

their original setting and domain requirements, but also have better quality, clarity, and consistency.

4.2.1 Image Datasets

Figures 14 and 15 present comparisons between the original GTs and our GTs generated by SkeView among four image datasets:

SK506 Shen et al. (2016a) (also known as SK-SMALL) was selected from the MS COCO Lin et al. (2014) dataset, with 506 natural images (300 for training and 206 for testing) and 16 object classes including humans, animals and artifacts. For each image, there is only one target for the skeleton GT generation. Due to inaccurate segmentation and unstable individual perception, the quality of the original GT is not promising. For instance, as shown in Fig. 14 (top), we observe the following issues: shortened branches of the elephant, the asymmetric branches in the airplane, an overly-simplified skeleton for the bird, and noisy branches in the hydrant. In contrast, GTs generated by SkeView are better in terms of various qualitative properties: consistency, perception friendliness, and the representation of object geometrical features.

SK1491 Shen et al. (2017) (also known as SK-LARGE) is an extension of the SK506 by selecting more images from the MS COCO dataset. It includes 1,491 images (746 for training and 745 for testing). Similar to SK506, there is one target for each image and the GT skeletons are annotated in the same way.

SYMMAX300 Tsogkas and Kokkinos (2012) is adapted from the Berkeley Segmentation Dataset (BSDS300) (Martin et al., 2001) with 300 images (200 for training and 100 for testing). There are multiple targets in most images. This dataset is used for local reflection symmetry detection, which is a low-level image feature, without paying attention to the concept of 'object'. While most branches are disconnected and the original GTs do not encode information about the connectivity of skeleton branches. Hence, it is ill-suited to evaluate object skeleton extraction methods as a large number of symmetries occur in non-object parts (see the bear, rhinoceros and lion images in Fig. 15 (top)). For this, we regenerated GTs only on target objects, as it was more meaningful to use object symmetry (foreground) instead of whole-image symmetry. As suggested in Tsogkas (2016), we ignore images without specific target objects.

SymPASCAL (Ke et al., 2017) was selected from the PASCAL-VOC dataset (Everingham et al., 2010), with 1,435 images (648 for training and 787 for testing). Most images contain multiple targets, partial visibility and complex backgrounds. However, there are still noisy symmetries from the background, incomplete skeleton graph and shortened skeleton branches. In contrast, GTs from SkeView focus only on the foregrounds, maintaining the same quality as with the other three image datasets. In Fig. 15, we clearly observe that our GTs in SYMMAX300 and SymPASCAL have the same quality as SK506, and skeleton branches for each object are well-connected. Such features can ensure a reliable evaluation on both skeleton extraction and matching algorithms (Bai and Latecki, 2008).

It should be noted that our annotation mainly captures the 2D contours, and partly loses the 3D symmetry awareness for some objects in images. However, our labelling is superior to the original GTs of the four image datasets, especially considering the consistency standards, branch connectivity and distinguished graphs. As a result, our GTs are more applicable for training and testing CNN-based skeleton detectors, as well as benchmarking skeleton-related pruning, matching and classification algorithms. In the future, we plan to update SkeView for 3D object and symmetry annotation (Tagliasacchi et al., 2016) based on our strategy.

4.2.2 Image and Shape Datasets

There are three datasets with both images and corresponding foreground shapes (Fig. 16). For this, we extracted initial skeletons using shapes and applied pruning using images in SkeView.

EM200 Yang et al. (2014) contains 200 microscopic images (10 classes) of environmental microorganisms (EM). There are two types of segmented foregrounds provided by the original dataset: those generated manually or semi-automatically with the methods introduced in Li et al. (2013). This dataset is challenging on colourless, transparent and spindly regions (flagellum). To ensure the quality of GTs, we employed the manual approach for initial skeleton generation. Then an efficient pruning in SkeView can best protect skeleton branches in those spindly regions for fine-grained EM matching and classification.

SmithsonianLeaves Ling and Jacobs (2007) contains 343 leaves (187 for training and 156 for testing) from 93 different species of plants. Each leaf was photographed on a plain background. K-means clustering was employed to estimate the foreground based on colour, followed by morphological operations to fill in small holes. Thus, this dataset is relatively less challenging with respect to occlusion and complex backgrounds, but has richer geometrical characteristics. Our GTs can be used by botanists to compute leaf similarity in the digital archives of the specimen types.

WH-SYMMAX Shen et al. (2016b) contains 328 cropped images (228 for training and 100 for testing) from the Weizmann Horse dataset (Borenstein and Ullman, 2002). Each image contains one manually segmented target. The original skeleton annotations are not only inconsistent concerning completeness across different horse shapes but also contain shortened branches. On the other hand, our GTs yield better quality with respect to consistency and completeness.

Fig. 14 Comparison of the original (yellow) GTs with the ones generated with our SkeView (red) in SK506 (Shen et al., 2016a) and SK1491 (Shen et al., 2017) datasets (Color figure online)

Fig. 15 Comparison of the original (yellow) GTs with the ones generated with our SkeView (red) in SYMMAX300 (Tsogkas and Kokkinos, 2012) (top two rows) and SymPASCAL (Ke et al., 2017) (bottom two rows) (Color figure online)

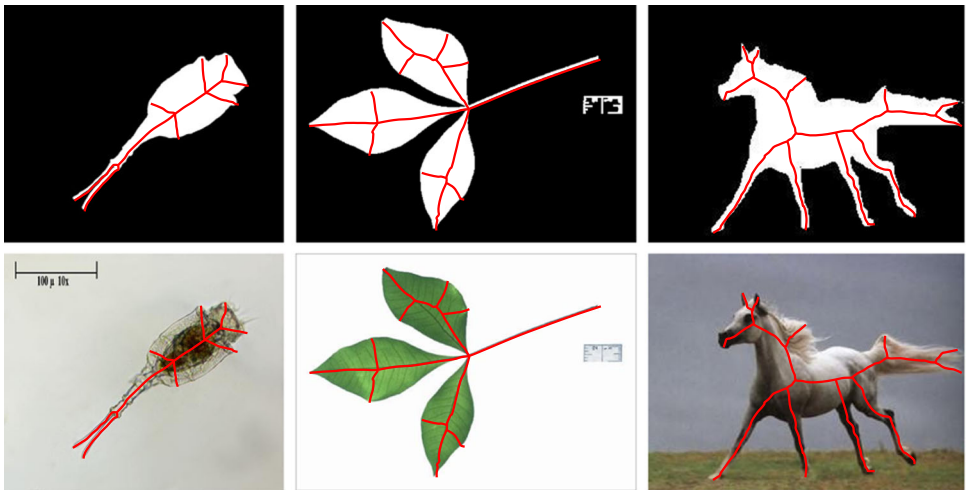

Fig. 16 From left to right, skeleton GT of EM200 (Yang et al., 2014), Smithsonian Leaves (Ling and Jacobs, 2007) and WH-SYMMAX (Shen et al., 2016b) generated by SkeView (Color figure online)

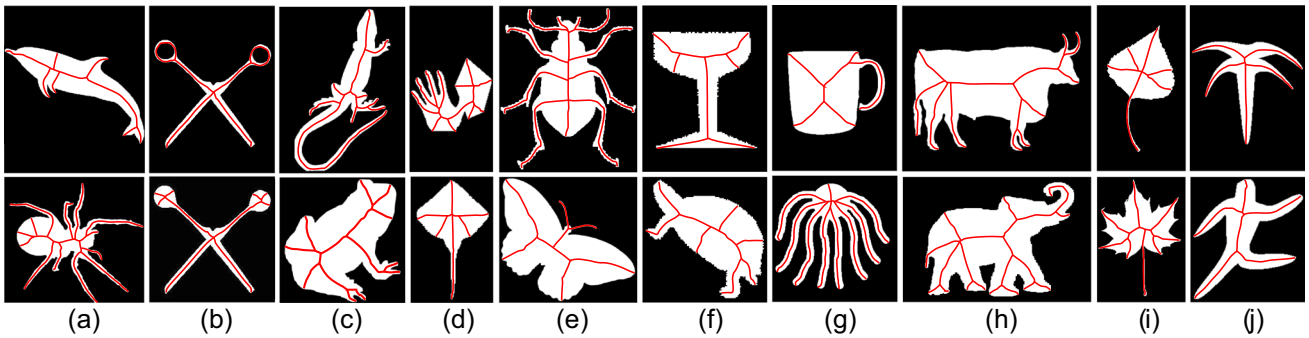

Fig. 17 Skeleton GTs of **a** Animal2000 (Bai et al., 2009), **b** ArticulatedShapes (Ling and Jacobs, 2007), **c** SkelNetOn (Ilke et al., 2019), **d** Kimia99 (Sebastian et al., 2004), **e** MPEG7 (Latecki et al., 2000), **f** Kimia216 (Sebastian et al., 2004), **g** MPEG400 (Yang et al., 2014), **h** Tetrapod120 (Yang et al., 2016), **i** SwedishLeaves (Söderkvist, 2001), **j** Tari56 (Asian and Tari, 2005) datasets generated by SkeView

4.2.3 Shape Datasets

Figure 17 presents samples of ten shape datasets and their GTs generated by SkeView:

Animal2000 Bai et al. (2009) contains 2,000 shapes (20 categories, 100 shapes each) ranging from poultry and domestic pets to insects and wild animals. Each class is characterised by large intra-class shape variations. Due to occlusion, some parts of certain objects (e.g. legs) are missing. There are also holes and boundary noises in some shapes due to incorrectly segmented foregrounds and backgrounds. This dataset is actively used in shape matching, classification and skeleton-based shape retrieval. As the shape category can be easily identified by human perception, critical parts of an object (e.g. legs, head, tentacles) are all preserved by the skeleton branches of our GTs.

ArticulatedShapes Ling and Jacobs (2007) contains 40 images from eight different objects. This challenging dataset consists of various tools including scissors with holes. To preserve the original topology, our GTs at such regions are closed branches (Fig. 17b (top)). Most existing matching algorithms (Bai and Latecki, 2008) cannot properly deal with skeleton graph structures with cycles, however we could provide skeleton GTs after filling the holes (Fig. 17b (bottom)).

SkelNetOn Ilke et al. (2019) contains 1,725 shapes (1,218 for training, 241 for validation and 266 for testing) represented as pixels. All shapes are of high quality with the holes and isolated pixels having removed by morphological operations (dilation and erosion) and manual adjustments. However, skeleton branches in this dataset are shortened and suffer from imbalance in simplicity, i.e. the original GTs in some shapes are extremely simple while others are overly complex. As such, it is difficult to conduct a fair comparison on skeleton-related algorithms such as extraction and matching. Moreover, this dataset is available only to registered participants in the SkelNetOn Challenge (Ilke et al., 2019). Our GTs are only for the purpose of skeleton quality analysis as shown in Table 3.

Kimia99 Sebastian et al. (2004) contains 99 shapes (9 categories, 11 shapes each) assembled from a variety of sources such as tools and hands, etc. Challenges in each category come from occlusion, and articulation of missing parts. To

Table 3 Mean reconstruction error (RE) and skeleton simplicity (SS) of our GTs. Skeletons from the automatic approaches DCE (Bai et al., 2007) (fixed $k = 10$), AutoSke (Shen et al., 2013) and Grafting (Yang et al., 2020) are also detailed for reference

	SK1491		EM200		Kimia216		MPEG7		Animal2000		SkelNetOn	
	RE	SS	RE	SS	RE	SS	RE	SS	RE	SS	RE	SS
DCE	0.90	**0.09**	**0.90**	0.07	**0.81**	0.09	**0.92**	0.08	0.86	**0.09**	0.87	0.07
AutoSke	0.89	0.08	0.91	**0.15**	0.82	**0.11**	**0.92**	**0.11**	0.86	**0.09**	0.85	**0.09**
Grafting	0.89	0.07	**0.90**	0.13	0.82	**0.11**	**0.92**	**0.11**	0.86	0.08	0.85	**0.09**
GTs	**0.88**	0.05	**0.90**	0.07	**0.81**	0.08	**0.92**	0.08	**0.85**	0.07	**0.82**	0.07

In each dataset, the lowest RE (towards complete skeleton structure) and the highest SS (towards simple skeleton structure) values are in boldface.

avoid topology violation of shapes, branches of extrinsic regions (e.g. Figure 17d (top)) are preserved in GTs.

MPEG7 Latecki et al. (2000) contains 1,400 (70 categories, 20 shapes each) shapes defined by their outer closed contours. It poses challenges with respect to deformation (e.g. change of view points and non-rigid object motion) and noises (e.g. quantisation and segmentation noise). This dataset is actively used for benchmarking shape representation, matching and retrieval algorithms (Yang et al., 2016, 2020). Similar to Kimia99, our GTs respect the topology of original shapes and properly preserve the challenges posed in each category.

Kimia216 Sebastian et al. (2004) contains 216 shapes (18 categories, 12 shapes each) selected from the MPEG7 dataset. It is actively used in skeleton extraction, pruning, matching and shape retrieval scenarios. Our GTs in this dataset form a subset of MPEG7.

MPEG400 Yang et al. (2014) contains 400 shapes selected from the MPEG7 dataset (20 categories, 20 shapes each). Instead of directly using the original shapes, boundary noises of these shapes were manually removed for ablation study. Thus, our GTs are slightly different from the corresponding ones in the MPEG7 dataset.

Tetrapod120 Yang et al. (2016) contains 120 tetrapod animal shapes from six classes. As shapes of some species are visually similar, this dataset is normally employed to evaluate shape matching and fine-grained classification algorithms. An advantage of SkeView is that branches of major regions are preserved. However, our GTs are not recommended for evaluating fine-grained classification algorithms as some animal species can only be distinguished via branches in small regions (e.g. floppy vs. pointy ears).

SwedishLeaves Söderkvist (2001) contains 1125 leaf shapes from 15 different Swedish tree species, with 75 leaves per species (25 for training, 50 for testing). This dataset is challenging as some species are quite similar. Past works (Ling and Jacobs, 2007; Söderkvist, 2001) have shown that it is not possible to distinguish them based on shape features alone. We do not intend to perform the same task using our GT skeletons. Instead, our GTs can be used for a wider scope of tasks – evaluating general skeleton extraction, pruning and matching algorithms.

Tari56 Asian and Tari (2005) contains 56 shapes (14 categories, 4 shapes each) for evaluating matching performance under visual transformations. Shapes of the same category show variations in orientation, scale, articulation and small boundary details. Motivated by this, our GT skeletons are useful for evaluating various skeleton-based shape matching algorithms. This is because our GTs contain branches with respect to the major and contextual shape regions. Moreover, our skeleton GTs are inherently robust to orientation and scale.

4.2.4 Properties

We discuss two measured properties of skeleton GTs: the mean Reconstruction Error (RE) and Skeleton Simplicity (SS). RE is already calculated by Eq. 2. Here, SS is calculated by:

$$s(\mathbf{S}) = \texttt{exp}(-\texttt{log}(\Gamma(S) + 1)) \quad . \tag{5}$$

where $\Gamma(S)$ denotes the normalized curve length of skeleton S. Since the GT skeletons are one pixel wide, $\Gamma(S)$ can be calculated simply from the number of skeleton points, normalized by the average path length of the skeleton. A constant value of 1 is added to ensure that the value from log function is positive. Equation 5 is motivated by the intuitive understanding that shorter skeletons have simpler structures. Here, SS is used for quantitative analysis since the differences between GTs are primarily at the fine-grained level. We note that another quantitative way to measure the simplicity of $\mathbf{S}$ is to use the number of junction and endpoints. However, experiments in Yang et al. (2020) show that endpoint statistics (both mean and standard deviation values) from different methods are similar to each other. In contrast, $\Gamma(S)$ is more distinguishable as it is sensitive to slight changes in the skeleton structure. In other words, SS has higher discriminative power than the number of endpoints, particularly at the fine-grained level.

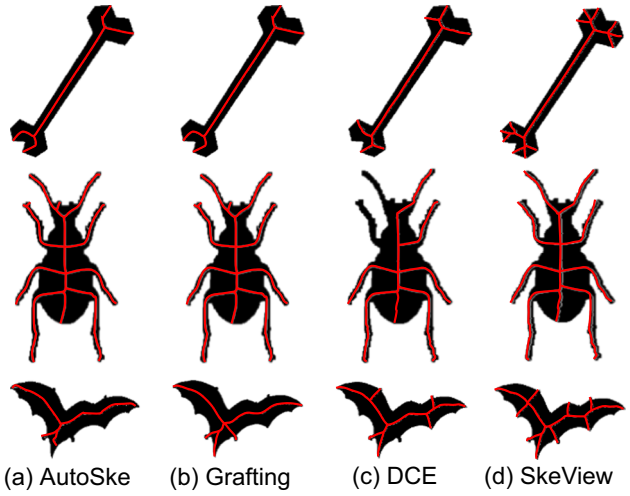

Fig. 18 Sample GTs generated by **a** AutoSke (Shen et al., 2013), **b** Grafting (Yang et al., 2020), **c** DCE (Bai et al., 2007) and **d** SkeView in MPEG7 dataset

Table 3 presents the RE and SS of our GTs. For comparison, skeletons generated by three automatic approaches DCE (Bai et al., 2007) (with the fixed stop parameter $k = 10$ recommended in Yang et al. (2016)), AutoSke (Shen et al., 2013) and Grafting (Yang et al., 2020) are also presented. We first report their statistical distribution of RE and SS values. Taking the Kimia216 dataset as an example, the statistics of 216 shapes are twofold: statistics within the same method and between different methods:

- With our proposed SkeView, the values are close to each other. Notably, the statistical distribution of RE is between 0.80−0.82, and SS is between 0.08−0.09. In other words, our GTs strike a stable balance, being structurally complete and relatively simple.
- With different approaches (DCE, AutoSke, Grafting), the distributions are more varied. RE and SS of DCE: 0.73−0.97, 0.05−0.09; RE and SS of AutoSke: 0.67−0.94, 0.06−0.14; RE and SS of Grafting: 0.73−0.91, 0.06−0.14. Though the mean values of RE and SS are close to SkeView, skeleton structures are unstable. In other words, some skeletons are either too simple, or too complex. This phenomenon is inherently similar to our observations in Sect. 3.1, such as O1 (Perception is robust to the time and volunteer groups) and O2 (Perception is robust to segmented objects and images).

We also observe that our GTs have the lowest RE, while the structures are more complex (smaller SS means more complex structure). For instance, RE in the MPEG7 datasets are the same (0.92), while SS of AutoSke (Shen et al., 2013) and Grafting (Yang et al., 2020) are the smallest (0.11). However, as shown in Fig. 18a, b, their skeleton completeness are not perceptually promising. Though skeletons generated by the DCE method (Bai et al., 2007) are the simplest in both SK1491 and Animal2000 datasets, their RE are relatively high (the first row in Table 3) while their skeleton structures are not visually promising (Fig. 18c). Overall, our GT skeletons strike the best balance, being perceptually friendly, structurally complete (in most cases), and relatively simple (the median SS is only 0.03 lower than AutoSke).

5 Benchmarks

In this section, we present a benchmark evaluation of skeleton detectors (mostly CNN-based methods) and skeleton-based matching methods using our GTs. For fairness, all settings follow their original papers unless stated otherwise.

5.1 Skeleton Detectors in Shapes

To quantitatively evaluate the performance of different skeleton detectors, we employed the average error pixel (AEP) proposed in Krinidis and Chatzis (2009) as the error measure. Specifically, it measures the error $e(\widehat{\mathbf{S}}, \mathbf{S})$ between a detected skeleton $\widehat{\mathbf{S}}$ against a GT $\mathbf{S}$ using the mean square error of their skeleton points:

$$e(\widehat{\mathbf{S}}, \mathbf{S}) = \frac{1}{N}\sum_{i=1}^{N}\sqrt{(\widehat{\mathbf{S}}_x(i) - \mathbf{S}_x(i))^2 + (\widehat{\mathbf{S}}_y(i) - \mathbf{S}_y(i))^2} \quad . \tag{6}$$

where $(\widehat{\mathbf{S}}_x(i), \widehat{\mathbf{S}}_y(i))$ are the coordinates of a skeleton point in $\widehat{\mathbf{S}}$, N is their total number of points, and $(\mathbf{S}_x(i)), \mathbf{S}_y(i))$ is the closest point in $\mathbf{S}$ to the point $(\widehat{\mathbf{S}}_x(i), \widehat{\mathbf{S}}_y(i))$. Table 4 details the evaluation results of five representative methods on eight shape datasets. The Physics method (Krinidis and Chatzis, 2009) generates skeleton points iteratively starting from a boundary point set based on a physics-based deformable model. Though it can be used to obtain stable skeletons with a fixed parameter setting, the results are not symmetric to the boundary and are sensitive to noises. The BPR (Shen et al., 2011) method of pruning skeletons is based on the context (modelled by the bending potential ratio) of the boundary segment that corresponds to the branch. The U-Net (Panichev et al., 2019) is a typical CNN-based method, which employs a modified U-Net architecture for direct skeleton regression. We can clearly see that most of the skeletons generated by AutoSke (Shen et al., 2013) have the lowest AEP and thus are closest to GTs. Though the DCE (Bai et al., 2007) method achieves the best result on Animal2000, it is still close to the result generated by AutoSke (only around 0.04 lower). Among all the methods, the CNN-based U-Net has the lowest performance, with around 0.31 and 2.62 higher AEP than the

Table 4 Average error pixel (AEP) of shape skeletons from different methods and datasets

	Kimia216	Ani2000	SL1	SL2	Tari56	MPEG7	AS	EM
DCE (Bai et al., 2007)	1.04	**0.97**	8.05	5.84	0.91	3.90	0.61	6.18
AutoSke (Shen et al., 2013)	**0.80**	1.01	**3.67**	**3.19**	**0.51**	**3.03**	**0.39**	**4.10**
Physics (Krinidis and Chatzis, 2009)	1.29	1.18	10.15	7.09	1.09	4.73	0.67	7.47
BPR (Shen et al., 2011)	0.88	1.14	4.13	3.66	0.56	3.33	0.44	4.55
U-Net (Panichev et al., 2019)	1.41	1.32	11.15	7.87	1.32	5.65	0.81	8.32

Ani2000: Animal2000. SL1: SmithsonianLeaves. SL2: SwedishLeaves. AS: ArticulatedShapes. EM: EM200. The smallest AEP in each dataset are shown in boldface

AutoSke method in Animal2000 and MPEG7, respectively. This is because skeletons generated by CNN-based methods normally yields low-quality branches (Zhang et al., 2022). To verify it, we visualize skeletons from existing five CNN-based methods trained using our GTs (see Fig. 19). We can clearly observe the noisy, disjointed, and incomplete skeleton branches.

5.2 Skeleton Detectors in Images

Skeletons in image are usually represented by binary maps after applying non-maximal suppression (NMS) and thresholding. The binary maps between the generated and the GT skeletons are matched pixel-wise to calculate the precision and recall values of the skeleton points. In practice, some small localization errors are allowed. Here, we used the F1 score (i.e. $2\times(\text{precision}\times\text{recall})/(\text{precision}+\text{recall})$) to evaluate the performance of skeleton detectors on image datasets. Particularly, five CNN-based methods (HED (Xie and Tu, 2015), SRN (Ke et al., 2017), Hi-Fi (Zhao et al., 2018), DeepFlux (Wang et al., 2019) and Ada-LSN (Liu et al., 2021)) are trained and tested using our GTs. To address a fair evaluation, we used their original settings on the backbone, image input size, and loss function. The initial network weights were initialized by Xavier. We optimized the loss functions using AdamW (Loshchilov and Hutter, 2018) with a mini-batch size of 1, an initial learning rate of 2e−4, and a weight decay of 1e−4. Regarding data augmentation, we applied random transformations to the image, including rotation, flipping, resizing and color jittering. For each method, we generated a precision-recall curve by varying the threshold value. The optimal threshold was selected as the one that produces the highest F1 score along the curve.

Their best results are presented in Table 5. We can see that Ada-LSN (Liu et al., 2021) achieves the highest score on almost all datasets. Figure 20 presents some sample results. It can be observed that the predictions are generally convergent towards our GTs, though their performances are clearly different. Particularly, most methods are not feasible to generate high-quality skeleton graphs. For instance, skeletons from HED (Holistically-Nested Edge Detection), SRN (Side-output Residual Network), and Hi-Fi (Hierarchical Feature integration) contains lots of noise and limited smoothness. Deep-Flux and AdaLSN (Adaptive Linear Span Network) output clearer and slimmer results, while there remain some false positive points, disjointed segments, and incomplete branches. The main reason is that most CNN-based methods output noisy, disjointed, and incomplete skeleton branches in heat maps (also called skeleton maps). Some networks cannot guarantee the topological and geometrical features in the representation. In practice, skeleton heat maps from existing CNN-based methods usually require heavy and semi-automatic processes to extract slim skeletons (one pixel wide). Nevertheless, the geometrical and topological features of the processed skeletons are still not ensured. We can also observe that F1 scores on the EM200 are the lowest among all the evaluated datasets. A major reason for this occurrence is that the training data is very limited (only 10 images), resulting in under-fitted models.

It is still possible to extract one-pixel-wide skeleton graphs. As presented in Fig. 20 (rightmost), we introduced a novel framework, BlumNet, for object skeleton extraction from shapes and images (Zhang et al., 2022). Unlike skeleton heat map regression with existing CNN-based methods, BlumNet decomposes a skeleton graph into structured components and simplifies the skeleton extraction problem into graph component detection and assembling tasks. Consequently, the quality of extracted skeletons is dramatically improved since BlumNet directly outputs slim, low-noise, and identified skeleton graphs. It should be noted that BlumNet was trained using our GTs from SkeView.

Figure 21 visually compares the detected skeletons on two sample images from SK1491 and SmithsonianLeaves. Specifically, skeletons from DeepFlux (Wang et al., 2019) and Ada-LSN (Liu et al., 2021) are slimmer and these two methods yield better performances in terms of continuity and completeness. Skeletons produced by HED are not well-integrated and are prone to noise. Though SRN and Hi-Fi yield clearer skeletons, they are not smooth and contain many false positive points. It is also interesting to find that most methods yield a better F1 score using the training and testing data from our GTs. For instance, the Fl score of Deep-

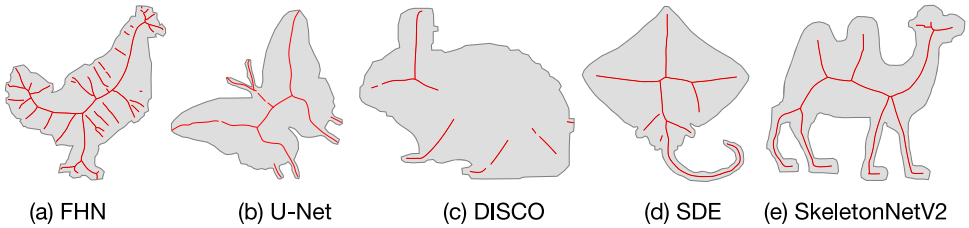

Fig. 19 Shape skeleton extraction using CNN-based methods: FHN (Jiang et al., 2019), U-Net (Panichev et al., 2019), DISCO (Song et al., 2021), SDE (Tang et al., 2021), and SkeletonNetV2 (Nathan and Kansal, 2021)

Table 5 F1 scores of skeleton detectors in images

	SK506	SK1491	SYMM	SymP	EM200	SL	WHS
HED (Xie and Tu, 2015)	0.552	0.494	0.431	0.370	0.298	0.580	0.741
SRN (Ke et al., 2017)	0.652	0.677	0.447	0.443	0.303	0.593	0.780
Hi-Fi (Zhao et al., 2018)	0.693	0.727	0.460	0.458	0.311	0.620	0.822
DeepFlux (Wang et al., 2019)	0.715	0.752	0.494	**0.520**	0.315	0.625	0.849
Ada-LSN (Liu et al., 2021)	**0.748**	**0.798**	**0.497**	0.504	**0.319**	**0.672**	**0.883**

SYMM: SYMMAX300. SymP: SymPASCAL. SL: SmithsonianLeaves. WHS: WH-SYMMAX

Fig. 20 Image skeleton detection results from HED (Xie and Tu, 2015), SRN (Ke et al., 2017), Hi-Fi (Zhao et al., 2018), DeepFlux (Wang et al., 2019), Ada-LSN (Liu et al., 2021), and BlumNet (Zhang et al., 2022) methods, trained using our GTs

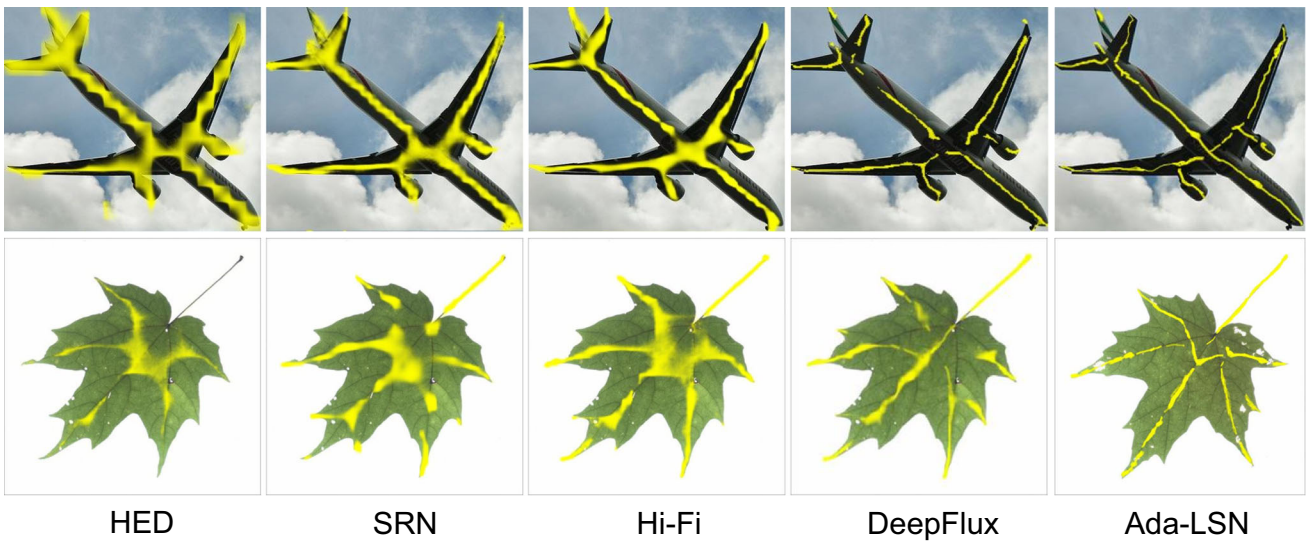

Fig. 21 Visual comparison of HED (Xie and Tu, 2015), SRN (Ke et al., 2017), Hi-Fi (Zhao et al., 2018), DeepFlux (Wang et al., 2019), and Ada-LSN (Liu et al., 2021) skeletons, trained using our GTs

Table 6 Skeleton detection performance (F1 scores) comparison of 6 methods trained and tested using GTs from Original and SkeView on SK1491

	HED (Xie and Tu, 2015)	SRN (Ke et al., 2017)	Hi-Fi (Zhao et al., 2018)	DeepFlux (Wang et al., 2019)	Ada-LSN (Liu et al., 2021)
GT (original)	**0.497**	**0.678**	0.724	0.732	0.786
GT (SkeView)	0.494	0.677	**0.727**	**0.752**	**0.798**

Flux (Wang et al., 2019) method improved from 0.732 to 0.752 on the SK1491 dataset (Table 6). This is because, comparing to the original GTs in the SK1491, our GTs are more consistent and possess better completeness in representing objects' geometrical features (see Fig. 14). As a result, these CNN-based models converge faster and generalize easier.

5.3 Skeleton Matching

In practice, the similarity of a shape pair can be calculated by matching their skeleton graphs. For instance, the Path Similarity (PS) method (Bai and Latecki, 2008) aims to match skeleton endpoints using the similarity between their corresponding skeleton paths (Fig. 22 (left)). The final shape similarity is calculated by summing up the similarity of corresponded endpoints. Based on the idea of shape context (Belongie et al., 2002), the skeleton context (SC) method (Kamani et al., 2016) employs log-polar histograms to describe sample skeleton points along the paths for matching (Fig. 22 (right)). The final shape similarity is computed by adding the distances between the matched skeleton points. We employ these methods to build baselines using our GTs on the Kimia216, MPEG400, Tetrapod120 and SwedishLeaves datasets (as they are actively used in this scenario). Specifically, we use each shape as a query and retrieve ten most similar shapes from the whole dataset according to their similarities. As shown in Table 7, the final value in each position (columns) is the total number of occurrences that matches query class at that position based on all shapes within a dataset. For example, the third position of the PS method on Kimia216 dataset shows that from 216 retrieved results in this position, 197 shapes have the same class as their query. We can see that the PS (Bai and Latecki, 2008) method achieves the best result among all the datasets, across all positions. This is because the PS method employs geodesic paths for matching endpoints, which makes it robust to scaling, rotation, occlusion, and also to same-class-objects of different topological structures.

For the MPEG7 and Animal2000 datasets, the Bulls-eye Scores (BES) (Latecki et al., 2000) are normally computed for quantitative evaluation. BES is calculated as a ratio between the correctly matched shapes to the total number of possible matches. For instance, as there are 1400 and 2000 queries in MPEG7 (20 in each class) and Animal2000 (100 in each class) datasets, the total number of possible matches are 1400×20 and 2000×100, respectively. Accordingly, we employ GTs for skeleton-based shape retrieval. In addition to the PS and SC algorithms, the High-order (HO) matching method proposed in Yang et al. (2020) is also used in the evaluation. The HO method fuses similarities between the skeleton graphs with their geometrical relations characterized by multiple skeleton endpoints. Motivated by Yang et al. (2020), Bai (2012) and Kontschieder et al. (2010), experiments on both datasets are clustered into two groups: (1) pairwise matching similar to the experiments in Table 7, and (2) context-based matching by increasing the discrimination between different classes within the shape manifold. For this, the Mutual kNN Graph (MG) (Kontschieder et al., 2010) and Co-Transduction (CT) (Bai, 2012) methods are employed (Table 8).

For the pairwise experiments, we can clearly see that the HO method yields the best performance in both datasets. For the context experiments, we find that the BES improves after applying the MG and CT methods on the matching results from PS and HO. However, we find that both methods are ineffective on SC, with a decline in BES. The main reason is that similarity values between skeletons as calculated by the SC method are close to each other, and this results in poor shape retrieval performance: 13.67% and 8.88% on MPEG7 and Animal2000, respectively. Thus, the similarity values within skeletons of the same class are easily mixed with other classes.

6 Discussion and Conclusion

We present a brief overview of the challenges posed by the GT baselines and possible directions for future research.

6.1 Analysis of Challenges

Skeleton Extraction: For most shape skeleton extraction approaches, we find that they cannot properly handle shapes with long and narrow (or lathy) regions. For instance, needle-like axopodia of actinophryid (EM200), petiole of leaves (SwedishLeaves), and antenna of insects (MPEG7). One possible solution is to generate their skeletons regionally, followed by integration and post-pruning steps. It is also

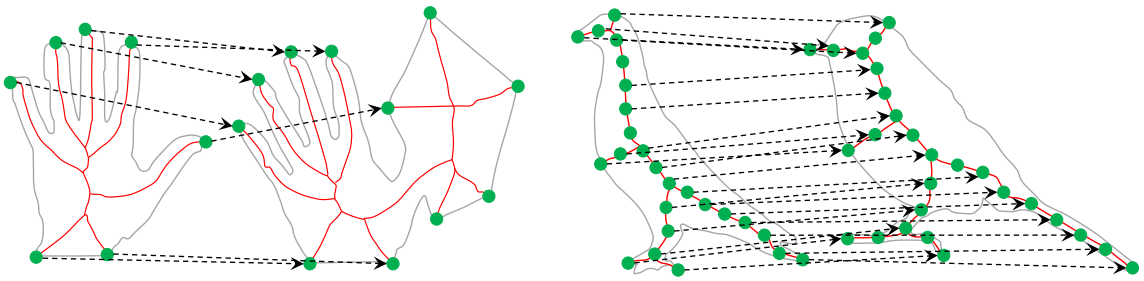

Fig. 22 Skeleton-based shape matching algorithms in our evaluation. Left: Path Similarity (PS) (Bai and Latecki, 2008). Right: Skeleton Context (SC) (Kamani et al., 2016)

Table 7 Comparison of two skeleton-based shape retrieval methods using our GTs

Kimia216	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
PS (Bai and Latecki, 2008)	216	206	197	185	173	169	154	150	140	119
SC (Kamani et al., 2016)	196	91	80	82	77	71	70	72	61	57
MPEG400	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
PS (Bai and Latecki, 2008)	400	384	380	367	363	351	339	338	327	318
SC (Kamani et al., 2016)	382	131	148	164	162	150	144	142	117	123
Tetrapod120	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
PS (Bai and Latecki, 2008)	120	110	90	86	73	73	72	55	55	51
SC (Kamani et al., 2016)	113	48	33	24	34	32	18	31	23	21
SwedishLeaves	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
PS (Bai and Latecki, 2008)	1125	974	914	881	868	833	813	790	785	767
SC (Kamani et al., 2016)	1057	220	207	203	198	190	204	198	169	184

PS: path similarity. SC: skeleton context

Table 8 Bulls-eye scores (BES, %) of three skeleton matching methods (PS: Path Similarity (Bai and Latecki, 2008), SC: Skeleton Context (Kamani et al., 2016) and HO: High-order Matching (Yang et al., 2020)) based on MPEG7 (M7) and Animal2000 (A2) GTs

	PS	PS+MG	PS+CT	SC	SC+MG	SC+CT	HO	HO+MG	HO+CT
M7	62.96	75.46	80.98	13.67	9.40	13.20	78.74	83.22	**87.28**
A2	24.26	29.52	34.27	8.88	6.54	8.32	34.14	37.95	**40.19**

MG (Mutual *k*NN Graph (Kontschieder et al., 2010)) and CT (Co-Transduction (Bai, 2012)) are their context-based extensions. The best scores are in boldface

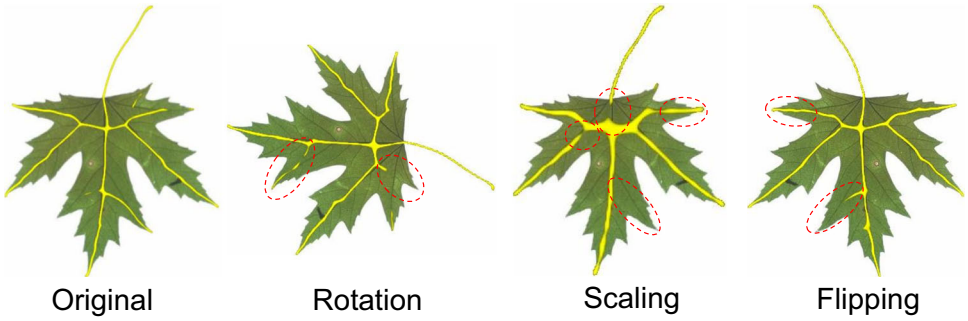

Fig. 23 Evaluation of Ada-LSN (Liu et al., 2021) on rotation, scaling and flipping

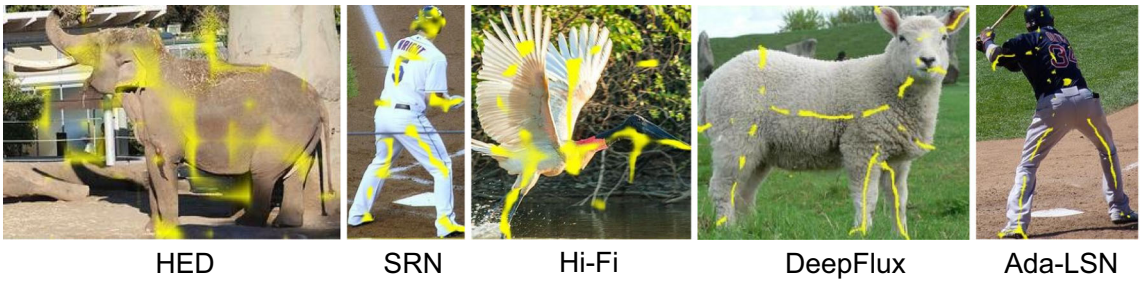

Fig. 24 Poor detection samples from HED (Xie and Tu, 2015), SRN (Ke et al., 2017), Hi-Fi (Zhao et al., 2018), DeepFlux (Wang et al., 2019) and Ada-LSN (Liu et al., 2021) methods

interesting to note that most approaches are not guaranteed to preserve the topology of shapes containing holes. The simple way to resolve this issue is to fill the holes during the pre-processing step. Connected shapes with such kinds of complexity frequently occurs in the real-world but are rarely studied, e.g. class No. 10 in the SwedishLeaves dataset. Inspired by the theorems in Bai et al. (2007), we suggest to incorporate boundary curves from both shapes and holes for skeleton extraction. For image skeleton extraction, the influence of the quality of training data is obvious. As a direct result of the consistent quality of our GTs, the F1 scores in Table 5 are generally higher than their original reported results (Ke et al., 2017; Liu et al., 2021; Xie and Tu, 2015; Wang et al., 2019; Zhao et al., 2018). However, it is desirable for the community to introduce higher quality and larger scale datasets. Our GTs also capture richer dynamics that cannot be learned from existing datasets. For instance, we find that all CNN-based methods in Table 5 are sensitive to image rotation, scaling and flipping, which are fundamental requirements towards robust skeleton extraction in the real-world. However, there remains some inadequately addressed issues. As shown in Fig. 23, even for Ada-LSN (Liu et al., 2021) trained with data augmentation (rotation, scaling and flipping) on the SmithsonianLeaves dataset, we still find that some major skeletons branches are shortened, disconnected, and erased. For this, junction points, endpoints and skeleton graph could be encoded to restrict skeleton regression during the training period.

Skeleton Matching: We further evaluate the pairwise matching algorithms in Table 8 using the ArticulatedShapes dataset. This is because its GTs contain closed branches (Fig. 17b (top)) with holes in tools such as scissors. We found that both the PS (Bai and Latecki, 2008) and HO (Yang et al., 2020) algorithms cannot properly deal with skeleton graphs containing cycles. Though the SC (Kamani et al., 2016) algorithm can be applied to such skeletons, it yields a poor performance with only 13.67 and 8.88 BES in MPEG7 and Animal2000, respectively. Therefore, we propose to improve the existing matching algorithms to support skeletons with closed branches. In Fig. 24, we see that most skeletons predicted in images are discontinuous with different widths and false positive points. In such cases, it is difficult to apply the existing algorithms for matching, classifying, and retrieval. In particular, these algorithms have been designed for one-pixel wide skeletons. To facilitate using image skeletons in practice, we propose to explore post-processing algorithms to bridge the gap between the image skeletons and the existing matching algorithms. Thus, a significant amount of research in future is necessary before image skeletons can become practically robust in many real-world objects.

6.2 Conclusion

We introduced a heuristic strategy for skeleton GT extraction in shape and image datasets. Our strategy is substantiated on both theoretical grounding and empirical investigation of human perception of skeleton complexity. To facilitate this, we developed a tool, SkeView, for skeleton GT extraction and used it on 17 existing image and shape datasets. We also systematically evaluated the existing skeleton extraction and matching algorithms to generate valid baselines using our GTs. Experiments demonstrate that our GT is consistent and can properly balance the trade-off between skeleton simplicity and completeness. We expect that the release of SkeView and the GTs to the community will benefit future research, particularly to address practical real-world challenges in CNN-based skeleton detectors and matching algorithms.

Acknowledgements Research activities leading to this work have been supported by the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (Grant Number: 22KJB520008) and the Research Fund of Clobotics (Grant Number: KB1801ZW201609-03). We would like to thank Zixuan Chen from Darmstadt University of Technology (Germany) for his help in assembling the first version of SkeView.

Data Availability The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

References

Asian, C., & Tari, S. (2005). An axis-based representation for recognition. In *IEEE International Conference on Computer Vision* (vol. 2, pp. 1339–1346).

Atienza, R., et al. (2019). Pyramid u-network for skeleton extraction from shape points. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 1–4).

Bag, S., Bhowmick, P., & Harit, G. (2011). Recognition of Bengali handwritten characters using skeletal convexity and dynamic programming. In *International Conference on Emerging Applications of Information Technology* (pp. 265–268).

Bai, X., Liu, W., & Tu, Z. (2009). Integrating contour and skeleton for shape classification. In *IEEE International Conference on Computer Vision Workshops* (pp. 360–367).

Bai, X., et al. (2012). Co-transduction for shape retrieval. *IEEE Transactions on Image Processing, 21*(5), 2747–2757.

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

Bai, X., Latecki, L. J., & Liu, W. (2007). Skeleton pruning by contour partitioning with discrete curve evolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 29*(3), 449–462.

Belongie, S., Malik, J., & Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*(4), 509–522.

Blum, H. (1967). A transformation for extracting new descriptors of shape. In *Models for Perception of Speech and Visual Forms* (pp. 362–380).

Borenstein, E., & Ullman, S. (2002). Class-specific, top-down segmentation. In *European Conference on Computer Vision* (pp. 109–122).

Bucksch, A. (2014). A practical introduction to skeletons for the plant sciences. *Applications in Plant Sciences, 2*(8), 1400005.

Cornea, N. D., Silver, D., & Min, P. (2007). Curve-skeleton properties, applications and algorithms. *IEEE Transactions on Visualization and Computer Graphics, 13*(3), 530–548.

Dasiopoulou, S., Giannakidou, E., Litos, G., Malasioti, P., & Kompatsiaris, Y. (2011). A survey of semantic image and video annotation tools. In *Knowledge-driven Multimedia Information Extraction and Ontology Evolution* (pp. 196–239).

Durix, B., Chambon, S., Leonard, K., Mari, J.-L., & Morin, G. (2019). The propagated skeleton: A robust detail-preserving approach. In *International Conference on Discrete Geometry for Computer Imagery* (pp. 343–354).

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision, 88*(2), 303–338.

Fanelli, D., & Piazza, F. (2020). Analysis and forecast of Covid-19 spreading in China, Italy and France. *Chaos, Solitons & Fractals, 134*, 109761.

Firestone, C., & Scholl, B. J. (2014). Please tap the shape, anywhere you like: Shape skeletons in human vision revealed by an exceedingly simple measure. *Psychological Science, 25*(2), 377–386.

Ge, Y., & Fitzpatrick, J. M. (1996). On the generation of skeletons from discrete Euclidean distance maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 18*(11), 1055–1066.

Giesen, J., Miklos, B., Pauly, M., & Wormser, C. (2009). The scale axis transform. In *Proceedings of the 25th Annual Symposium on Computational Geometry* (pp. 106–115).

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. In *IEEE International Conference on Computer Vision* (pp. 2961–2969).

Ilke, D., et al. (2019). Skelneton 2019: Dataset and challenge on deep learning for geometric shape understanding. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 1–9).

Jalba, A. C., Sobiecki, A., & Telea, A. C. (2015). An unified multiscale framework for planar, surface, and curve skeletonization. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 38*(1), 30–45.

Jiang, N., et al. (2019). Feature hourglass network for skeleton detection. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 1–5).

Kamani, M. M., Farhat, F., Wistar, S., & Wang, J. Z. (2016). Shape matching using skeleton context for automated bow echo detection. In *IEEE International Conference on Big Data* (pp. 901–908).

Ke, W., Chen, J., Jiao, J., Zhao, G., & Ye, Q. (2017). SRN: Side-output residual network for object symmetry detection in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1068–1076).

Kontschieder, P., et al. (2010). Beyond pairwise shape similarity analysis. In *Asian Conference on Computer Vision* (pp. 655–666).

Krinidis, S., & Chatzis, V. (2009). A skeleton family generator via physics-based deformable models. *IEEE Transactions on Image Processing, 18*(1), 1–11.

Latecki, L.J., Lakamper, R., & Eckhardt, T. (2000). Shape descriptors for non-rigid shapes with a single closed contour. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 424–429).

Li, Y., & Qu, H. (2018). LSD and skeleton extraction combined with farmland ridge detection. In *International Conference on Intelligent and Interactive Systems and Applications* (pp. 446–453).

Li, C., Shirahama, K., Czajkowska, J., Grzegorzek, M., Ma, F., & Zhou, B. (2013). A multi-stage approach for automatic classification of environmental microorganisms. In *International Conference on Image Processing, Computer Vision, and Pattern Recognition* (p. 1).

Lin, T.-Y., et al. (2014). Microsoft coco: Common objects in context. In *European Conference on Computer Vision* (pp. 740–755).

Ling, H., & Jacobs, D. W. (2007). Shape classification using the inner-distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 29*(2), 286–299.

Liu, L., Chambers, E. W., Letscher, D., & Ju, T. (2011). Extended grassfire transform on medial axes of 2D shapes. *Computer-Aided Design, 43*(11), 1496–1505.

Liu, C., Tian, Y., Chen, Z., Jiao, J., & Ye, Q. (2021). Adaptive linear span network for object skeleton detection. *IEEE Transactions on Image Processing, 30*, 5096–5108.

Loshchilov, I., & Hutter, F. (2018). Decoupled weight decay regularization. In *International Conference on Learning Representations* (pp. 1–19).

Lowet, A. S., Firestone, C., & Scholl, B. J. (2018). Seeing structure: Shape skeletons modulate perceived similarity. *Attention, Perception, & Psychophysics, 80*(5), 1278–1289.

Martin, D., Fowlkes, C., Tal, D., & Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In

IEEE International Conference on Computer Vision (Vol. 2, pp. 416–423).

Nathan, S., & Kansal, P. (2021). Skeletonnetv2: A dense channel attention blocks for skeleton extraction. In *IEEE International Conference on Computer Vision Workshops* (pp. 2142–2149).

Ogniewicz, R., & Ilg, M. (1992). Voronoi skeletons: Theory and applications. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 63–69).

Oliva, A., & Torralba, A. (2007). The role of context in object recognition. *Trends in Cognitive Sciences, 11*(12), 520–527.

Panichev, O., et al. (2019). U-net based convolutional neural network for skeleton extraction. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 1–4).

Russell, B. C., Torralba, A., Murphy, K. P., & Freeman, W. T. (2008). Labelme: A database and web-based tool for image annotation. *International Journal of Computer Vision, 77*(1–3), 157–173.

Saha, P. K., Borgefors, G., & di Baja, G. S. (2016). A survey on skeletonization algorithms and their applications. *Pattern Recognition Letters, 76*, 3–12.

Sebastian, T. B., Klein, P. N., & Kimia, B. B. (2004). Recognition of shapes by editing their shock graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 26*(5), 550–571.

Sharma, V., Jääskö, K., Yiannacou, K., Koivikko, A., Lampinen, V., & Sariola, V. (2021). Performance comparison of fast, transparent and biotic heaters based on leaf skeletons. *Advanced Engineering Materials*, 1–11.

Shen, W., Zhao, K., Jiang, Y., Wang, Y., Zhang, Z., & Bai, X. (2016). Object skeleton extraction in natural images by fusing scale-associated deep side outputs. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 222–230).

Shen, W., Bai, X., Hu, R., Wang, H., & Latecki, L. J. (2011). Skeleton growing and pruning with bending potential ratio. *Pattern Recognition, 44*(2), 196–209.

Shen, W., Bai, X., Hu, Z., & Zhang, Z. (2016). Multiple instance subspace learning via partial random projection tree for local reflection symmetry in natural images. *Pattern Recognition, 52*, 306–316.

Shen, W., Bai, X., Yang, X., & Latecki, L. J. (2013). Skeleton pruning as trade-off between skeleton simplicity and reconstruction error. *Science China Information Sciences, 56*(4), 1–14.

Shen, W., Zhao, K., Jiang, Y., Wang, Y., Bai, X., & Yuille, A. (2017). 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.

Shokouh, G.-S., Magnier, B., Xu, B., & Montesinos, P. (2021). Ridge detection by image filtering techniques: A review and an objective analysis. *Pattern Recognition and Image Analysis, 31*(3), 551–570.

Skov, R. B., & Sherman, S. J. (1986). Information-gathering processes: Diagnosticity, hypothesis-confirmatory strategies, and perceived hypothesis confirmation. *Journal of Experimental Social Psychology, 22*(2), 93–121.

Söderkvist, O. (2001). Computer vision classification of leaves from Swedish trees. In Master Thesis, Linköping University (pp. 1–74).

Song, S., Bae, H., & Park, J. (2021). Disco-u-net based autoencoder architecture with dual input streams for skeleton image drawing. In *IEEE International Conference on Computer Vision Workshops* (pp. 2128–2135).

Tagliasacchi, A., Delame, T., Spagnuolo, M., Amenta, N., & Telea, A. (2016). 3D skeletons: A state-of-the-art report. In *Computer Graphics Forum* (Vol. 35, pp. 573–597).

Tang, X., Zheng, R., & Wang, Y. (2021). Distance and edge transform for skeleton extraction. In *IEEE International Conference on Computer Vision Workshops* (pp. 2136–2141).

Teichmann, L., Edwards, G., & Baker, C. I. (2021). Resolving visual motion through perceptual gaps. *Trends in Cognitive Sciences, 25*(11), 978–991.

Telea, A., & Wijk, J. J. v. (2002). An augmented fast marching method for computing skeletons and centerlines. In *Proceedings of VisSym* (pp. 251–258).

Tsogkas, S. (2016). Mid-level representations for modeling objects. PhD thesis, Université Paris Saclay (COmUE).

Tsogkas, S., & Kokkinos, I. (2012). Learning-based symmetry detection in natural images. In *European Conference on Computer Vision* (pp. 41–54).

Tversky, A. (1977). Features of similarity. *Psychological Review, 84*(4), 327–352.

Wang, Y., Xu, Y., Tsogkas, S., Bai, X., Dickinson, S., & Siddiqi, K. (2019). Deepflux for skeletons in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5287–5296).

Xie, S., & Tu, Z. (2015). Holistically-nested edge detection. In *IEEE International Conference on Computer Vision* (pp. 1395–1403).

Yang, C., Indurkhya, B., See, J., & Grzegorzek, M. (2020). Towards automatic skeleton extraction with skeleton grafting. *IEEE Transactions on Visualization and Computer Graphics*, 1–1.

Yang, C., Li, C., Tiebe, O., Shirahama, K., & Grzegorzek, M. (2014). Shape-based classification of environmental microorganisms. In *International Conference on Pattern Recognition* (pp. 3374–3379).

Yang, C., Tiebe, O., Grzegorzek, M., & Indurkhya, B. (2016). Investigations on skeleton completeness for skeleton-based shape matching. In *Signal Processing: Algorithms, Architectures, Arrangements, and Applications* (pp. 113–118).

Yang, C., Tiebe, O., Pietsch, P., Feinen, C., Kelter, U., & Grzegorzek, M. (2014). Shape-based object retrieval by contour segment matching. In *IEEE International Conference on Image Processing* (pp. 2202–2206).

Yang, C., Tiebe, O., Shirahama, K., & Grzegorzek, M. (2016). Object matching with hierarchical skeletons. *Pattern Recognition, 55*, 183–197.

Zhang, Y., Sang, L., Grzegorzek, M., See, J., & Yang, C. (2022). Blumnet: Graph component detection for object skeleton extraction. In *ACM International Conference on Multimedia* (pp. 5527–5536).

Zhang, Z., Shen, W., Yao, C., & Bai, X. (2015). Symmetry-based text line detection in natural scenes. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2558–2567).

Zhang, T. Y., & Suen, C. Y. (1984). A fast parallel algorithm for thinning digital patterns. *Communications of the ACM, 27*(3), 236–239.

Zhao, K., Shen, W., Gao, S., Li, D., & Cheng, M.-M. (2018). Hi-fi: hierarchical feature integration for skeleton detection. In *International Joint Conference on Artificial Intelligence* (pp. 1191–1197).