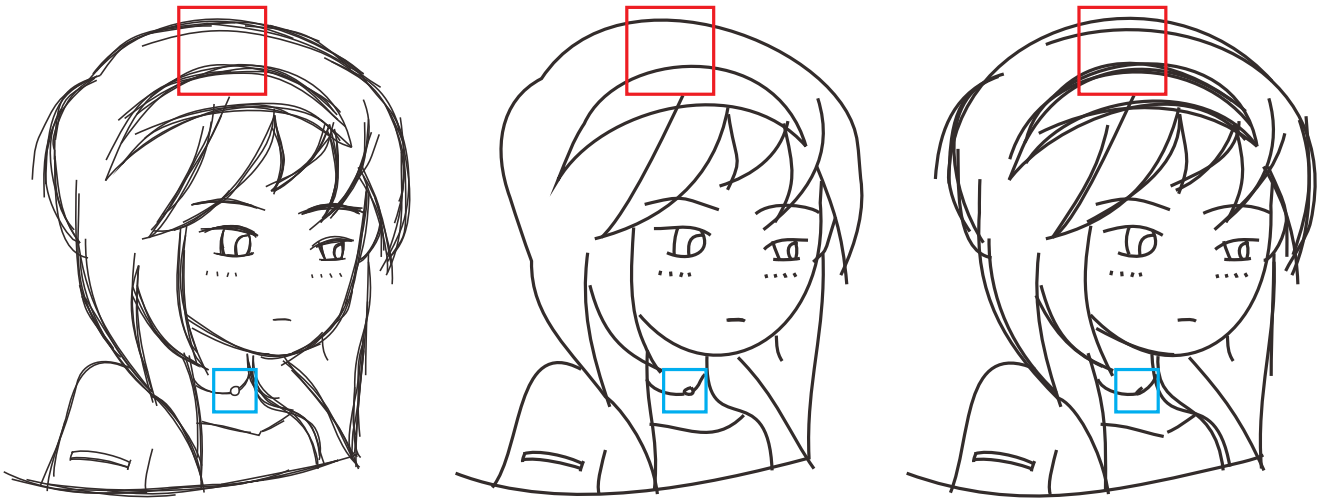
Closure-aware Sketch Simplification

Xueting Liu[1,2] Tien-Tsin Wong[1,2] Pheng-Ann Heng[1]
[1]The Chinese University of Hong Kong*
[2]Nanshan District Key Laboratory, Shenzhen Research Institute, The Chinese University of Hong Kong



(a) Input sketch *(b) Our simplified sketch* *(c) [Barla et al. 2005] (ε=7)*

Figure 1: *"Girl". (a) An input sketch where the large-scale hairs are depicted by coarse strokes and the small-scale necklace is depicted by fine strokes. (b) In aware of closure, our method groups the coarse strokes and preserves the fine details in the same sketch. The input sketch contains* 240 *input strokes and* 1173 *initial regions.* 50 *stroke groups and* 19 *perceptual regions are obtained after simplification. The whole simplification process takes* 3.8 *minutes. (c) The existing methods that rely on absolute inter-stroke properties either over-group the details or leave the coarse strokes ungrouped.*

Abstract

In this paper, we propose a novel approach to simplify sketch drawings. The core problem is how to group sketchy strokes meaningfully, and this depends on how humans understand the sketches. The existing methods mainly rely on thresholding low-level geometric properties among the strokes, such as proximity, continuity and parallelism. However, it is not uncommon to have strokes with equal geometric properties but different semantics. The lack of semantic analysis will lead to the inability in differentiating the above semantically different scenarios. In this paper, we point out that, due to the gestalt phenomenon of *closure*, the grouping of strokes is actually highly influenced by the interpretation of regions. On the other hand, the interpretation of regions is also influenced by the interpretation of strokes since regions are formed and depicted by strokes. This is actually a chicken-or-the-egg dilemma and we solve it by an iterative cyclic refinement approach. Once the formed stroke groups are stabilized, we can simplify the sketchy strokes by replacing each stroke group with a smooth curve. We evaluate our method on a wide range of different sketch styles and semantically meaningful simplification results can be obtained in all test cases.

CR Categories: J.5 [Computer Application]: Arts and Humanities—Fine Arts;

Keywords: sketch simplification, gestalt analysis, law of closure

1 Introduction

Sketching is usually the very first stage for presenting and communicating ideas in various domains, e.g. artwork drawing, product design, storyboard creation, etc. It allows artists to focus on the overall design instead of drawing the fine details. But the trade-off is the untidiness, since it is natural for artists to use several short strokes to depict a long curve (e.g. Figure 2(a)). Therefore, once the sketch designs are finalized, artists further need to transform the sketchy strokes into simplified clean drawings. In aid of the drawing process, some pen-based digital design tools have become accessible (e.g. Adobe Illustrator, CorelDRAW). With these tools, artists can directly draw on computers via tablets and digital pens. But it is still natural for artists to sketch first and then transform sketches into simplified clean drawings. To simplify a sketch drawing, the artist usually repetitively traces the sketchy strokes and replaces them with the clean ones. As one may imagine, this manual tracing process is quite tedious and time-consuming.

In this paper, we aim at converting a rough sketch drawing into a simplified clean drawing automatically. However, even if the input sketch is composed of digital strokes, this task remains challenging. This is mainly because higher-level semantic understanding is required in understanding the semantics of the rough strokes. To simplify a sketch drawing, the key is to understand which

*e-mail:{xtliu,ttwong,pheng}@cse.cuhk.edu.hk

ACM Reference Format
Liu, X., Wong, T., Heng, P. 2015. Closure-aware Sketch Simplification.
ACM Trans. Graph. 34, 6, Article 168 (November 2015), 10 pages. DOI = 10.1145/2816795.2818067
http://doi.acm.org/10.1145/2816795.2818067.

SIGGRAPH Asia '15 Technical Paper, November 02 – 05, 2015, Kobe, Japan.

DOI: http://doi.acm.org/10.1145/2816795.2818067

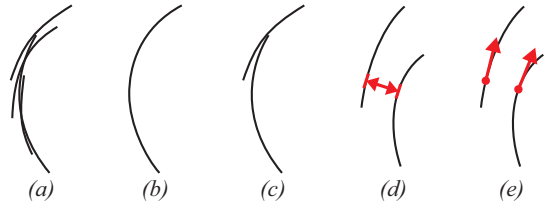


Figure 2: *(a) Input strokes. (b) A perceptual stroke that can represent the rough strokes in (a). (c) The simplified result of (a) via stroke reduction. (d) Inter-stroke proximity. (e) Inter-stroke continuity.*

input strokes semantically refer to the same *perceptual stroke*, and therefore can be grouped together. For example, in Figure 2(a), humans can easily observe that all the four strokes refer to the same perceptual stroke as shown in Figure 2(b), but this is not easy for the computer. To evaluate whether two input strokes refer to the same perceptual stroke, the existing methods only rely on low-level geometric properties among the strokes, such as inter-stroke distances (Figure 2(d)) and angular differences (Figure 2(e)). But in reality, the artists tend to use coarser strokes to depict large shapes (red square in Figure 1(a)) while finer strokes to depict fine details (blue square in Figure 1(a)). In other words, it is possible to have strokes that are geometrically the same but semantically different, as illustrated in Figure 3. Due to the lack of semantic analysis, the existing methods are unable to distinguish the semantic differences among the pairs of red strokes in Figures 3(a)-(d) since they are all geometrically the same. In comparison, we propose a sketch simplification method that can correctly distinguish and group the four pairs of red strokes by analyzing higher-level semantics (Figures 3(e)-(h)).

The grouping of strokes can actually be explained by the gestalt phenomena of human visual perception [Wertheimer 1923], which are mostly studied qualitatively in psychology. Theoretically speaking, the existing computational methods that group strokes based on inter-stroke distances and angular differences can be regarded as modeling two forms of gestalts, the law of proximity and the law of continuity, respectively. However, in this paper, we point out that these two laws alone are not sufficient to understand the semantics needed in stroke grouping. An important gestalt law is missed – *the law of closure*. This law suggests that humans tend to perceptually group elements (strokes in our case) together if they form a closed shape. Hence, the notion of regions formed by strokes plays an important role in the semantic analysis of sketchy strokes. For example, in Figure 3(b), the two red strokes depict the same part of the rectangle region, so they refer to the same perceptual stroke. In contrast, the two red strokes in Figure 3(c) depict two different rectangle regions, and the two red strokes in Figure 3(d) depict different parts of the slim region, so both stroke pairs do not refer to a single perceptual stroke. Based on this observation, if we could identify the regions first, we could better interpret the semantics of the strokes. Here, by saying stroke interpretation, we mean to interpret which strokes refer to the same *perceptual stroke* and therefore belong to the same stroke group.

However, extracting regions from a sketch drawing is an open problem due to the sketchiness of the strokes. Without knowing the semantics of the strokes, it would be quite difficult to correctly identify the regions. So this is actually a dilemma of chicken-or-the-egg, as regions are formed by strokes, and strokes are in turn interpreted by regions. To resolve this dilemma, we propose to resolve region identification and stroke grouping by an iterative cyclic approach. Given the input strokes, we first extract an initial set of the potential regions. With the initial regions, we can group the input strokes based on not only proximity and continuity, but also closure. The grouping of the strokes further influences the identification of the regions, and the refined regions will influence the grouping of the strokes in the next iteration. This iterative cyclic refinement process continues until there is no more change in the formed stroke groups. Comparing to existing methods that only rely on lower-level inter-stroke properties (Figure 1(c)), our method accomplishes higher-level semantic analysis by accounting for closure (Figure 1(b)). Hence, our method can distinguish the semantic differences in Figures 3(a)-(d). We demonstrate the effectiveness of our method on various types of sketches. Our contributions can be summarized as followed.

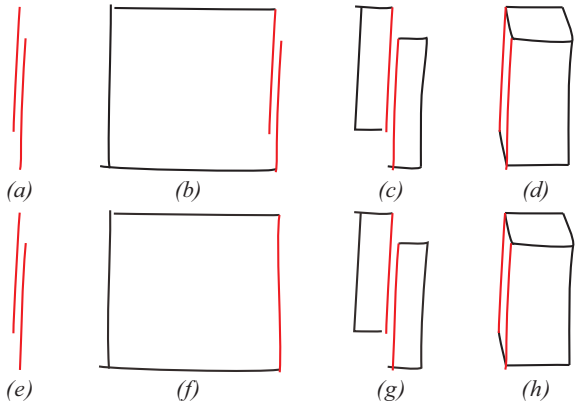


Figure 3: *(a)-(d) The four pairs of red strokes are geometrically the same but semantically different. (e)-(h) The simplified results generated by our method.*

- We make the first attempt in incorporating the law of closure into the semantic analysis of the sketches.
- We propose an iterative cyclic refinement approach to resolve the mutual influence of regions and strokes.

2 Related Work

The key of sketch simplification is to replace the sketchy strokes with a set of clean strokes and at the same time preserve visual contents. The existing methods related to sketch simplification can be roughly classified into three categories: progressive drawing, stroke reduction and stroke grouping.

Progressive Drawing During sketching, an artist usually incrementally modifies a sketch by drawing new strokes on top of the existing ones. Several interactive drawing systems are proposed so that artists may modify digital sketches naturally [Bae et al. 2008; Grimm and Joshi 2012]. In particular, Baudel [1994] proposed an interactive stroke editing system to mimic the natural modification process. To edit the existing digital strokes, instead of adjusting the control points, users can naturally draw on top of the existing strokes, and the system will update the existing strokes to match the user input. Kara et al. [2006] further proposed to modify 3D strokes by dragging the existing 3D stroke points towards the input 2D strokes. However, the progressive drawing systems still require the artists to focus on drawing the details instead of sketching the overall design. Besides, these methods generally rely on stroke ordering information, while our method does not.

Stroke Reduction To simplify a sketch drawing, attempts have also been made in removing the strokes one by one until the drawing is simple enough. In general, the strokes are first sorted according to their significance values, then the stroke with the least significance is removed. When measuring the significance value of a stroke in the 2D space, position, local density, and line length are usually analyzed [Preim and Strothotte 1995]. When the strokes are rendered from 3D inputs, depth and silhouette information are also studied in addition to the 2D properties [Deussen and Strothotte

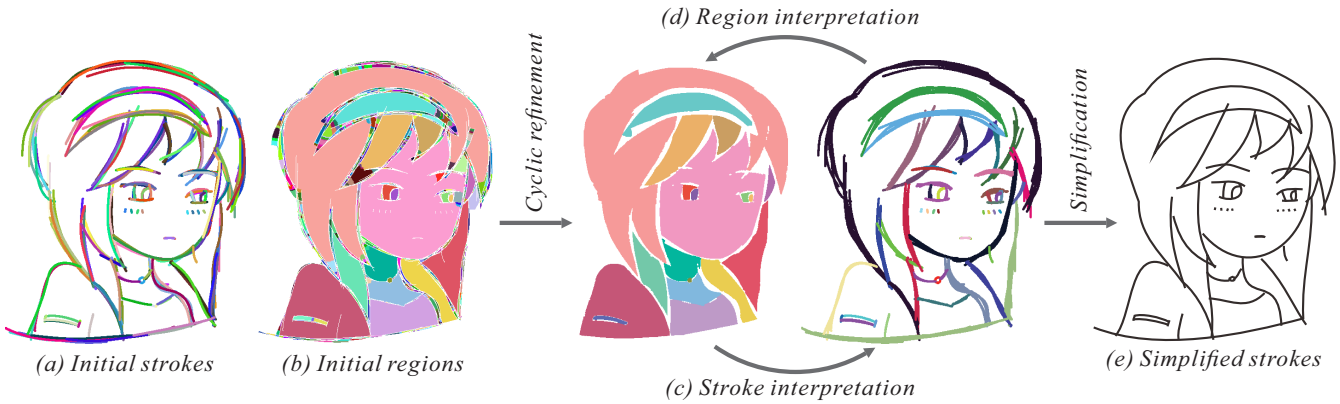


Figure 4: *System overview.*

2000; Wilson and Ma 2004; Grabli et al. 2004]. The stroke reduction methods are quite useful in the LOD representation of 2D and 3D line drawings. However, these methods only simplify sketch drawings by removing existing strokes in the original sketch, and no new stroke is created. But in reality, the artists may draw dozens of short strokes to depict a long curve. In other words, there could be no single stroke that can represent the whole curve, and the resultant drawing may remain sketchy (e.g. Figure 2(c)).

Stroke Grouping The technique that is mostly related to our application is stroke grouping. Several automatic/semi-automatic methods have been proposed to group strokes in a bottom-up manner based on inter-stroke proximity, continuity, and parallelism [Pavlidis and Wyk 1985; Rosin 1994; Lindlbauer et al. 2013]. In particular, Barla et al. [2005a; 2005b] proposed to group strokes based on the overlapping zone, i.e. the proportions where the two strokes are at a distance less than a fixed threshold ε. Fu et al. [2011] followed Barla's method in stroke grouping for animating the line drawings. Shesh and Chen [2008] also studied the extent of overlapping as a criterion for stroke grouping. Orbay and Kara [2011] further introduced a learning-based stroke grouping system based on similar inter-stroke properties. Noris et al. [2012] developed an interactive stroke editing system which groups the strokes based on both inter-stroke properties and user inputs. Recently, Chien et al. [2014] proposed to group strokes by analyzing the pairing of stroke ends, which is similar to the extent of overlapping. Different from the above bottom-up methods, Pusch et al. [2007] proposed to simplify sketch drawings in a top-down manner by box subdivision. However, all the above methods analyze inter-stroke properties only in an *absolute* sense without considering closure. That is, two strokes are grouped together if their distance, angular difference, and other geometric properties are within certain *absolute* thresholds. Relying on these absolute inter-stroke properties, the existing methods are unable to distinguish the stroke pairs that are geometrically the same but semantically different. In contrast, by accounting for the gestalt phenomenon of closure, our method measures the inter-stroke properties in a relative sense (relative to the regions they formed). With this formulation, even if two pairs of strokes are geometrically the same, our method can still distinguish their semantic difference according to the formed regions.

The gestalt phenomenon of closure is also studied in stroke selection [Saund and Moran 1994] and architectural drawing abstraction [Nan et al. 2011]. However, these methods model closure independently and do not consider how closure influences proximity and continuity. Therefore, these methods still cannot distinguish the stroke pairs that are geometrically the same but semantically different. Chen et al. [2013] further proposed to simplify raster sketch images via an image processing approach.

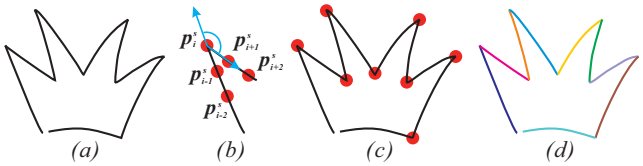


Figure 5: *(a) A discontinuous stroke. (b) A sharp change in the drawing direction. (c) The breaking points. (d) The smooth segments.*

3 Overview

Our system flow is presented in Figure 4. The input of our system is a set of digital strokes drawn by the user via a tablet and a stylus. For raster input, one may consider vectorizing the raster sketch image as a preprocessing step using existing line vectorization techniques [Noris et al. 2013]. Each stroke s is represented by a sequence of densely sampled stroke points $\{\boldsymbol{p}_1^s, \boldsymbol{p}_2^s, \dots\}$. Here, $\boldsymbol{p}_i^s$ is the 2D position of the i-th stroke point on s. Since complex strokes may complicate the subsequent stroke grouping (Figure 5(a)), we first break each stroke into multiple stroke segments where each stroke segment is a smooth curve (Figure 5(d)). To determine the breaking points, we measure if there is a sharp change in the drawing direction at each stroke point (Figure 5(b)). By breaking each stroke at all breaking points (Figure 5(c)), we obtain a set of smooth stroke segments as the initial strokes (Figure 5(d)&Figure 4(a)). In the rest of this paper, all the strokes we discussed are assumed smooth.

With the initial strokes, we can already group the strokes based on inter-stroke proximity and continuity in an absolute sense. However, to measure the influence of closure, we need to first identify the regions formed by strokes. But region detection in hand-drawn sketches is an open problem. Since the input strokes are sketchy, the commonly used flood-fill method may cause the leaking problem. To extract regions from rough sketches, leak-proof region extraction methods may be adopted [Qu et al. 2006; Zhang et al. 2009; Sýkora et al. 2009; Noris et al. 2012]. In particular, we adopt the trapped-ball method [Zhang et al. 2009] to extract all potential regions (the maximal trapped-ball radius is set to 8 pixels). We regard the extracted regions as the initial regions.

Note that the initial regions may not be consistent with the *perceptual regions* that humans perceive. Since the initial region map is over-segmented, multiple initial regions may together represent one perceptual region. Furthermore, some initial regions are formed due to the sketchiness of the strokes but do not even perceptually exist. We need to further interpret which initial regions belong to the same perceptual region, and which initial regions do not perceptually exist. This interpretation of the initial regions is called *region interpretation*. To correctly interpret the regions, we need to first interpret the strokes as regions are

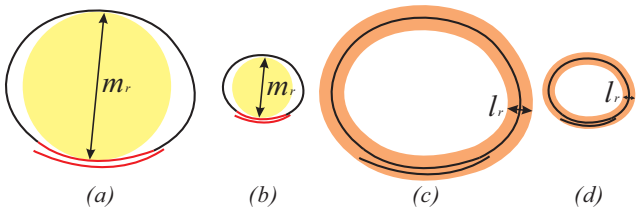


Figure 6: *(a)&(b) The relative distances of the two pairs of red strokes are similar with respect to the nearby regions. (c)&(d) The width of the boundary band of a region is positively related to the scale of this region.*

formed by strokes. But correctly interpreting the strokes is our original purpose. To resolve this chicken-or-the-egg dilemma, we propose an iterative approach which resolves stroke interpretation and region interpretation cyclically. During each iteration, we first fix the region interpretation and interpret the strokes with respect to the regions by accounting for closure (Figure 4(c) & Section 4). Then we fix the stroke interpretation and identify the perceptual regions based on the current grouping of strokes (Figure 4(d) & Section 5). This mutual refinement process continues until there is no more change in the formed stroke groups (Section 6). Finally, the strokes that represent the same perceptual stroke are replaced by a single smooth Bézier curve (Figures 4(e) & Section 4).

4 Stroke Interpretation

In this section, we explain how to interpret the strokes with respect to a set of perceptual regions by accounting for closure. In the rest of this section, all regions we discussed are actually the perceptual regions obtained in the previous iteration. In order to tell whether two initial strokes are the same perceptual stroke, we also analyze inter-stroke properties, but with respect to the regions. Based on these properties, we group the strokes into a set of stroke groups where strokes that semantically refer to the same perceptual stroke should be grouped together.

4.1 Closure-aware Inter-stroke Properties

Like the existing methods, we also analyze inter-stroke proximity, continuity, and parallelism in order to judge whether two strokes are the same perceptual stroke. But unlike the existing works that measure these properties in an absolute sense, we analyze them in a relative sense with respect to the regions.

Closure-aware Proximity Proximity refers to the closeness among the strokes. If two strokes are quite close to each other, it is more likely that they are in the same stroke group. The existing methods measure inter-stroke proximity in an absolute sense. That is, two strokes s and t are considered near if their geometric distance

$$D(s,t) = \min_{i,j} \|\boldsymbol{p}_i^s - \boldsymbol{p}_j^t\| \tag{1}$$

is smaller than a fixed threshold. But in reality, artists tend to use coarser strokes to depict large shapes and finer strokes to depict fine details. So a fixed threshold either over-groups the fine strokes or leaves the coarse strokes ungrouped. In comparison, we measure inter-stroke proximity in a relative sense with respect to the nearby regions.

The intuition of our relative proximity is that, if the two strokes are the boundaries of a larger region, their relative distance with respect to this region should be smaller (Figure 6(a)). On the contrary, if the two strokes are the boundaries of a smaller region, their relative distance should be larger (Figure 6(b)). For example, in Figures 6(a)&(b), though the absolute distance between the two red strokes in (a) is larger than that in (b), the relative distances of the two pairs are similar since the strokes in (a) are the boundaries of a larger region. Therefore, for every two strokes s and t that border

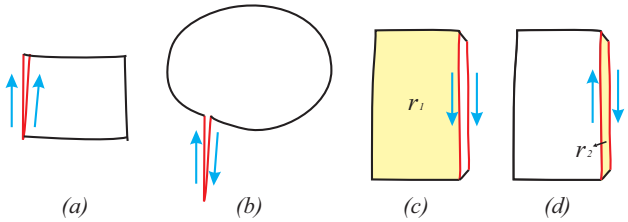


Figure 7: *(a)&(b) The two pairs of red strokes are geometrically the same, but the pair in (a) is relatively continuous while the pair in (b) is not. (c)&(d) With respect to different regions, the relative continuity between two strokes may be different.*

a region r, we can formulate the relative distance between s and t with respect to r as

$$\tilde{D}_r(s,t) = \frac{D(s,t)}{l_r} \tag{2}$$

where l_r is the scale of r which suggests the potential coarseness of the boundary strokes that surround region r. We observe that l_r is positively related to m_r which is the radius of the largest disc that can be placed inside r (Figures 6(a)&(b)), and we formulate it as

$$l_r = \alpha m_r^\beta \tag{3}$$

where α and β are scaling factors and set to 5 and 0.4 empirically in our experiments. In Figures 6(c)&(d), we visualize the scales of the regions as the widths of the boundary bands surrounding the regions. Intuitively, the boundary strokes of a region should be mostly inside this boundary band.

To calculate the relative distance between two strokes with respect to the nearby regions, we need to first identify which regions are nearby. We define that, if a stroke s is inside or partially inside the boundary band of region r, then r is a nearby region of s and s is a potential boundary stroke of r. With respect to all the nearby regions, we formulate the relative distance between two strokes s and t as

$$\tilde{D}(s,t) = \min_r \tilde{D}_r(s,t) \quad \forall r \text{ nearby } s,t \tag{4}$$

Through this formulation, we link inter-stroke proximity to the interpretation of regions.

Closure-aware Continuity Continuity measures whether two strokes have similar directions. If so, it is very likely that they refer to the same perceptual stroke. Note that the direction of a stroke here does not mean the drawing direction by the user. It is a kind of intepretation with respect to the current region of interest (explained shortly). Typically, two strokes s and t are considered continuous if their angular difference in either direction

$$A(s,t) = 1 - \left|\boldsymbol{\theta}(\boldsymbol{p}_{i^*}^s) \cdot \boldsymbol{\theta}(\boldsymbol{p}_{j^*}^t)\right| \tag{5}$$

is smaller than a fixed threshold. Here,

$$\{i^*, j^*\} = \arg\min_{i,j} \|\boldsymbol{p}_i^s - \boldsymbol{p}_j^t\| \tag{6}$$

are the indices of the nearest stroke points on s and t respectively.

$$\theta(\boldsymbol{p}_i^s) = \frac{\boldsymbol{p}_{i+1}^s - \boldsymbol{p}_i^s}{\|\boldsymbol{p}_{i+1}^s - \boldsymbol{p}_i^s\|} \tag{7}$$

is the unit tangent vector at stroke point $\boldsymbol{p}_i^s$. However, this absolute angular difference is insufficient to distinguish the two pairs of red strokes in Figures 7(a)&(b) which have the same angular difference but different continuities. To distinguish the semantic difference in continuity between these two pairs, we propose to analyze inter-stroke continuity in a relative sense with respect to the nearby regions. For example, in Figure 7(a), with respect to the rectangular region, the two strokes are also relatively continuous. But in Figure 7(b), with respect to the bubble region, there is a sharp

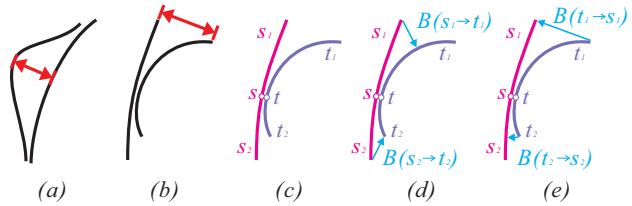


Figure 8: *(a)&(b) The two pairs of strokes are both near and continuous, but both pair do not refer to a single perceptual stroke. (c) Each stroke is broken into two segments (d)&(e) The deviations between the paired stroke segments.*

change in the direction between the two strokes, and therefore they are not relatively continuous.

In fact, the tangent vector at a stroke point may point either forward or backward along the drawing direction. With respect to a nearby region, the tangent direction may point either along the clockwise direction or along the anticlockwise direction. To calculate the relative angular difference between two strokes with respect to a region, the tangent directions of the two strokes used for comparison should be along the same direction of the region (either clockwise or anticlockwise). For example, the relative tangent directions of the red strokes are visualized in Figures 7(a)&(b) where we adopt the clockwise direction without loss of generality. Now we can formally write the relative angular difference between two strokes s and t with respect to region r as

$$\tilde{A}_r(s,t) = 1 - \tilde{\boldsymbol{\theta}}_{\boldsymbol{r}}(\boldsymbol{p}_{\boldsymbol{i}*}^{\boldsymbol{s}}) \cdot \tilde{\boldsymbol{\theta}}_{\boldsymbol{r}}(\boldsymbol{p}_{\boldsymbol{j}*}^{\boldsymbol{t}}) \tag{8}$$

where $\tilde{\boldsymbol{\theta}}_{\boldsymbol{r}}(\boldsymbol{p}_{\boldsymbol{i}*}^{\boldsymbol{s}})$ is the unit tangent vector at stroke point $\boldsymbol{p}_{\boldsymbol{i}*}^{\boldsymbol{s}}$ along the clockwise direction of r. With respect to all the nearby regions, the relative angular difference between two strokes s and t is set to

$$\tilde{A}(s,t) = \max_r \tilde{A}_r(s,t) \quad \forall r \text{ nearby } s,t \tag{9}$$

Intuitively, if two strokes are not continuous with respect to any one of the nearby regions, then they are not relatively continuous.

Closure-aware Parallelism Parallelism measures how well two strokes are aligned. If two strokes refer to the same perceptual stroke, they must not deviate too much from each other at the corresponding stroke points. For example, Figures 8(a)&(b) show two pairs of strokes that are both near and continuous, but neither of them refer to a single perceptual stroke due to the large deviations.

In practice, to measure the largest deviation between two strokes s and t, we first break each stroke into two stroke segments at the nearest stroke points (Figure 8(c)). Assume that s is broken into s_1 and s_2, and t is broken into t_1 and t_2. The obtained stroke segments could be paired up according to the relative continuity between s and t. Without loss of generality, we assume s_1 corresponds to t_1 and s_2 corresponds to t_2. Now we can observe that, if s and t refer to the same perceptual stroke, the largest deviation between the corresponding stroke segments (s_1 and t_1, s_2 and t_2) must be small. To measure the largest deviation between two stroke segments s_k and t_k ($k = 1, 2$), we could simply check whether all the stroke points on one stroke segment (s_k or t_k) are close to the other segment (t_k or s_k). We formally write the maximal distance from the stroke points on s_k to t_k (Figure 8(d)) as

$$B(s_k \to t_k) = \max_i \min_j \|\boldsymbol{p}_{\boldsymbol{i}}^{\boldsymbol{s_k}} - \boldsymbol{p}_{\boldsymbol{j}}^{\boldsymbol{t_k}}\| \tag{10}$$

Intuitively, if $B(s_k \to t_k)$ is small, it suggests that all the stroke points on s_k are close to t_k, and therefore s_k can be represented by t_k. The maximal distance from t_k to s_k can be defined similarly (Figure 8(e)). Then the largest deviation between two stroke segments s_k and t_k can be calculated as

$$B(s_k, t_k) = \min\{B(s_k \to t_k), B(t_k \to s_k)\} \tag{11}$$

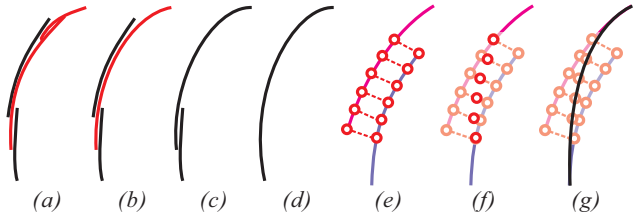


Figure 9: *(a)-(d) The stroke grouping process. (e)-(g) The proxy stroke is computed by smoothly interpolating the corresponding stroke points of the two strokes.*

The largest deviation between two strokes s and t is formulated as the larger deviation between the corresponding segment:

$$B(s,t) = \max_{k=\{1,2\}} B(s_k, t_k) \tag{12}$$

Note that the measurement of deviation is based on the measurement of distance, so it also relies on the scales of the regions they depict. That is, we actually measure the largest deviation between two strokes s and t with respect to the nearby regions as

$$\tilde{B}(s,t) = \min_r \frac{B(s,t)}{l_r} \quad \forall r \text{ nearby } s,t \tag{13}$$

Recall that l_r is the scale of region r. In this way, we link inter-stroke parallelism to the regions as well.

4.2 Stroke Grouping

With the defined closure-aware proximity, continuity and parallelism, we then group the strokes in a bottom-up greedy way. To do so, we first form a stroke pair for every two strokes in the sketch, and sort all the stroke pairs in an ascending order based on proximity, continuity, and parallelism. Here, proximity is first compared, and then continuity and parallelism. Then we check if the two strokes in the first pair is relatively near, continuous, and parallel, i.e.

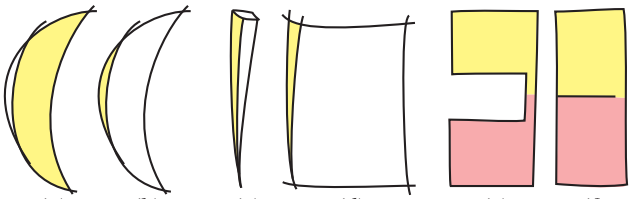
$$\tilde{D}(s,t) < T_D \ \wedge \ \tilde{A}(s,t) < T_A \ \wedge \ \tilde{B}(s,t) < T_B \tag{14}$$

where T_D, T_A, and T_B are thresholds, and set to 1, 0.2, 2 empirically. Note that the constraint on proximity is tighter than the constraint on parallelism. The tight constraint on proximity ensures that strokes referring to the same perceptual stroke must be quite close at some point. The loose constraint on parallelism allows certain degree of sketchiness of a stroke group. If the first stroke pair satisfies this criterion, we mark the two strokes as the same perceptual stroke and replace them with a *proxy stroke* that can represent both strokes (Figures 9(a)&(b)). If not, we simply skip to the next pair. This process is repeated until no stroke pair can be grouped together (Figures 9(a)-(d)). Note that when measuring inter-stroke properties between the two strokes in a stroke pair, we actually make use of the proxy strokes if the strokes are already replaced by proxy strokes. Figures 9(e)-(g) visualizes how a proxy stroke is formed by smoothly interpolating the corresponding stroke points between two strokes with a Bézier spline.

Figure 4(c) visualizes our computed stroke groups by color-coding strokes in the same group with the same color. By incorporating the law of closure, we successfully group coarse strokes and at the same time preserve fine strokes. We also want to point out that with different region interpretations, the relative inter-stroke properties are different, and the formed stroke groups are also different. This is why our computed stroke groups can be iteratively refined during cyclic refinement.

5 Region Interpretation

In this section, we present how we interpret regions based on a stroke interpretation. To interpret regions, we intend to merge



(a) (b) (c) (d) (e) (f)

Figure 10: *(a)&(b) The yellow region in (a) is a perceptual region surrounded by complete boundary strokes, while the yellow region in (b) is a non-perceptual region surrounded by partial strokes. (c)&(d) The yellow region in (c) is a perceptual region intentionally formed by the artist, while the yellow region in (d) is a non-perceptual region accidentally formed when creating the large square shape. (e)&(f) Both pairs of regions are more completed if merged together. But the two regions in (e) form one perceptual region, while the two regions in (f) do not.*

regions that belong to the same perceptual region and distinguish perceptual regions from non-perceptual ones. To do so, we first analyze the characteristics of perceptual regions. Then we show how to form perceptual regions from the initial regions.

5.1 Perceptual Region vs. Non-perceptual Region

To distinguish perceptual regions from non-perceptual regions, we propose to analyze from two aspects: region completeness and region independence.

Region Completeness Region completeness measures whether a region is a perceptual region with respect to the surrounding strokes. If a region is a perceptual region, it should be intentionally drawn by the artist. Therefore, it is very likely that the strokes that form this region are drawn to fully depict this region. That is, the *whole* stroke should be attached to the region's boundary (e.g. Figure 10(a)). In contrast, if the region is not a perceptual region, the boundary strokes that form this region are actually drawn to depict other regions. So it is very likely that only parts of the boundary strokes are attached to this unintended region (e.g. Figure 10(b)). Through analyzing whether a region is surrounded by complete boundary strokes, we could measure whether a region is likely to be a perceptual region or a non-perceptual region. We call this concept as *region completeness*.

When measuring the completeness of the boundary strokes in depicting a region, we actually measure the completeness of the *stroke groups* since multiple short strokes may form one long perceptual stroke. In particular, we compute the completeness of a boundary stroke group g in depicting the region r as

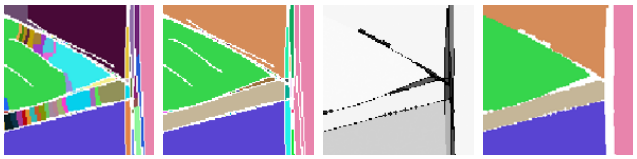
$$V(g,r) = \frac{\sum_{s\in g}\sum_{\boldsymbol{p}\in s}\left(G_{l_r} * O_r\right)_{\boldsymbol{p}}}{\sum_{s\in g} N_s} \tag{15}$$

where N_s is the number of stroke points on stroke s, O_r is a binary image where the boundary pixels of r are set to 1, G_{l_r} is a Gaussian filter with standard deviation l_r, and $*$ is the convolution operator. Intuitively, if $V(g,r)$ is large, it means that stroke group g is quite attached to the boundary of region r, and there is a high probability that the strokes in g are intentionally drawn to depict r.

The completeness of a region is high if the whole shape is formed by stroke groups that are intentionally drawn to depict this region. Therefore, for each boundary pixel $\boldsymbol{b}$ of region r, we select the maximal completeness of the surrounding stroke groups as the completeness value of this boundary point, i.e.

$$C(\boldsymbol{b}) = \max_{g}\left(G_{l_r} * I_g\right)_{\boldsymbol{b}} V(g,r) \tag{16}$$

where I_g is a binary image where the pixels of the strokes in stroke group g are set to 1. Now we can define the completeness of the region r as the average completeness value of all its boundary points as



(a) (b) (c) (d)

Figure 11: *(a) Initial regions. (b) Merged regions. (c) Perceptual likelihoods. (d) Perceptual regions.*

$$C(r) = \frac{1}{|\Gamma_r|}\sum_{b\in\Gamma_r} C(\boldsymbol{b}) \tag{17}$$

Here, Γ_r is the set of boundary points of r. Intuitively, the completeness of a region tells the completeness degrees of the surrounding stroke groups in depicting this region. With this formulation, the region in Figure 10(a) has a much higher completeness value than the region in Figure 10(b) according to the completeness degrees of the surrounding boundary strokes. Note that the completeness of a region will change if the surrounding stroke groups change, and the changes of the completeness values will directly influence the formed perceptual regions.

Region Independence Region completeness alone may be insufficient to distinguish the two yellow regions in Figures 10(c)&(d) since both regions have high completeness values. However, while the region in (c) is a perceptual region intentionally formed by the artist, the region in (d) is a non-perceptual region unintentionally formed when creating the large rectangular shape. From these two cases, we observe that whether a region is a perceptual region highly depends on its neighboring regions. Therefore, we propose to analyze whether a region is independently formed or is a side-effect of the other regions. We call this feature as *region independence*. In particular, we measure the independence degree of region r by computing the average distance from the pixels inside r to the nearby regions $\Omega(r)$ as

$$P(r) = 1 - \frac{1}{|N_r|}\max_{r'\in\Omega(r)}\sum_{\boldsymbol{p}\in r}\left(G_{l_{r'}} * I_{r'}\right)_{\boldsymbol{p}} \tag{18}$$

where N_r is the number of pixels inside region r. Note that the distance between r and r' is also related with the scale $l_{r'}$ of r'. Intuitively, if all the pixels inside a region are close to any one of the nearby regions, the independence value of this region should be low, and it is very likely that this region is a non-perceptual region.

5.2 Perceptual Region Formation

Based on the formulated region completeness and region independence, we now may form perceptual regions from the initial regions through two steps – merging and classification.

Region Merging Since the initial regions are over-segmented, we first merge the regions to avoid misclassification in the next step. We merge two spatially neighboring regions r and r' if they do not have a shared boundary (stroke group) and the merged region has a higher completeness value, i.e.

$$\nexists g \quad \text{s.t.} \quad \left(V(g,r) > T_L\right) \ \wedge \ \left(V(g,r') > T_L\right) \tag{19}$$

and

$$2C(r\cup r') - C(r) - C(r') > T_M \tag{20}$$

Here T_L and T_M are empirically set to 0.9 and 0.05 respectively in our experiments. The reason we adopt the shared boundary criterion is to differentiate the two cases in Figures 10(e)&(f) where the merged region has a higher completeness value in both cases. Then we merge the regions in a bottom-up greedy way. To do so, we first form a region pair for any two neighboring regions, and sort the region pairs based on the increments of completeness values in merging in a descending order. Then we sequentially check if each region pair satisfies the above two criteria. If so, the two regions are merged together into one region. The completeness of this merged

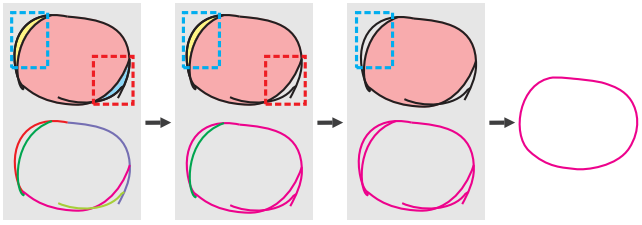


Figure 12: *The iterative cyclic refinement process.*

region is re-calculated in order to check if it can be further merged with the other regions. This merging process repeats until no more merging can take place. Figure 11(b) shows the merged regions computed from the initial regions in Figure 11(a).

Region Classification After region merging, we further classify the merged regions into perceptual regions and non-perceptual regions. A perceptual region should be both completed and independent, i.e.

$$C(r)P(r)^{\gamma} > T_C \tag{21}$$

Here, T_C is set to 0.5. γ is a weighting factor and set to 0.2 in our experiments. The regions that fail this criterion are regarded as non-perceptual regions and discarded. Figure 11(c) visualizes the probabilities of the regions being perceptual regions. Figure 11(d) shows the classified perceptual regions based on Figure 11(c).

6 Cyclic Refinement

As we have discussed above, region interpretation and stroke interpretation are mutually influenced by each other. So any method that resolves the interpretations sequentially may fail in correctly interpreting the semantics. Instead, we propose a novel iterative refinement system which interprets strokes and regions cyclically. During each iteration, we first fix stroke interpretation and form perceptual regions. Then we fix region interpretation and form stroke groups. The iterative refinement process stops when there is no more change on the formed stroke groups.

Figure 12 illustrates our iterative cyclic refinement process. There are five initial strokes and three initial regions in this sketch (Figure 12(a)). Then we start our cyclic refinement process by first interpreting the regions. In the first iteration (Figure 12(b)), each initial stroke is regarded as a single stroke group, and we interpret the regions based on this stroke interpretation. We can observe that the blue region in the dashed red box is identified as a non-perceptual region and discarded due to its low completeness and low independence. But the yellow region in the dashed blue box has relatively high completeness and is preserved. Then, based on this region interpretation, we group the five initial strokes into two stroke groups. In the second iteration (Figure 12(c)), with the refined stroke groups, the completeness of the yellow region becomes rather low, and therefore is identified as a non-perceptual region. With the refined region interpretation, all the initial strokes are grouped together into one stroke group in this iteration. The refinement process terminates in the third iteration since no more change can take place (Figure 12(d)). In our experiments, this refinement process usually takes $3 \sim 5$ rounds to converge.

As the system progressively forms the stroke groups, it also constructs smooth proxy strokes that can represent the stroke groups. In other words, the proxy strokes after convergence are actually the simplified strokes (Figure 4(e)).

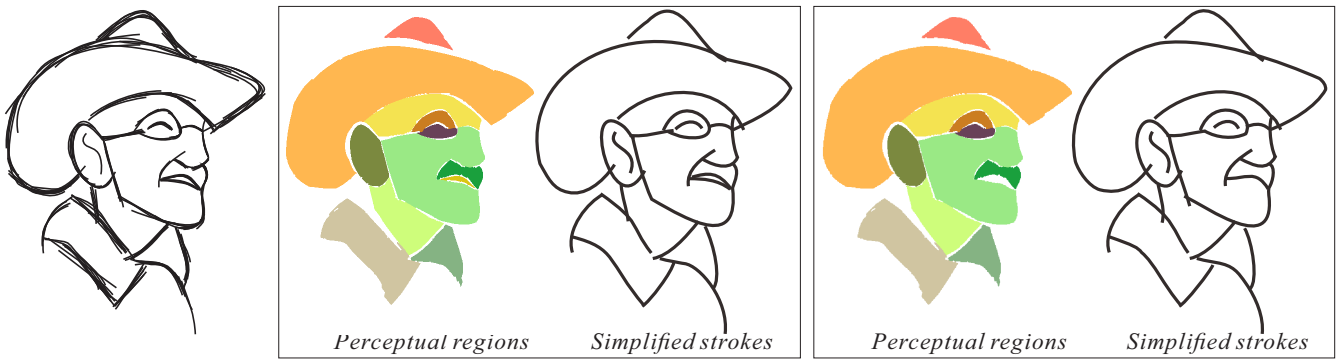
7 Results and Discussions

To validate the effectiveness of our method, we apply our method on sketches with various styles, including two illustrative sketches (Figure 3&16), a Japanese-style manga sketch (Figure 1), a portrait sketch (Figure 13), an architectural sketch (Figure 14), an object sketch (Figure 15), and two western-style cartoon sketches (Figures 17&18). We also compare our closure-aware sketch simplification method with the existing methods which are all based on absolute properties (Figures 1, 14, 15, 16, and 17). With the iterative cyclic refinement system, our method can simplify strokes and extract perceptual regions simultaneously. The data statistics of each figure is reported in the corresponding caption. More results and comparisons can be found in the supplementary materials.

Figures 3(e)-(h) show our simplified results of the sketches presented in Figures 3(a)-(d). Here, the four pairs of red strokes in (a)-(d) are geometrically the same but semantically different. The strokes in (a) are not grouped together since there is no region nearby and the absolute proximity between the two strokes is low. In contrast, the two red strokes in (b) are grouped together due to the nearby large square region. In (c), the two regions are both in small-scale, hence the two strokes are relatively further apart and not grouped together. In (d), the two strokes are relatively discontinuous with respect to the slim region in between them, and hence are not grouped together.

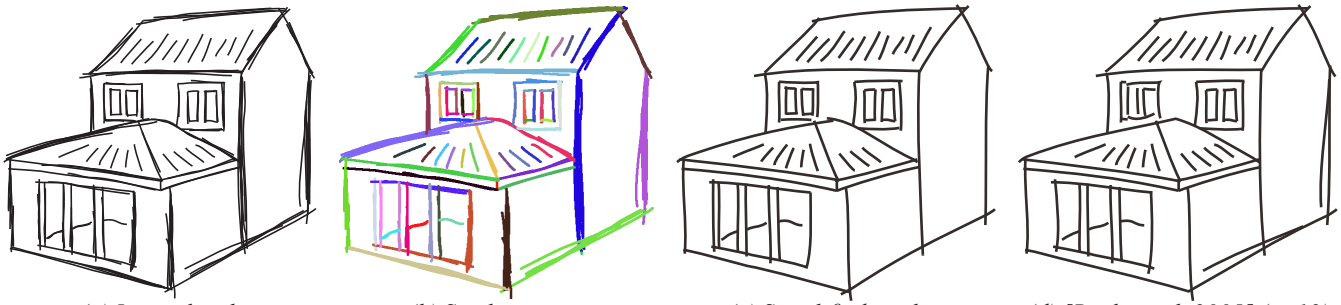
Figure 1 shows a sketch of a girl where the large-scale hairs are depicted by coarse strokes and the small-scale necklace is depicted by fine strokes. Comparing to [Barla et al. 2005a] that relies on absolute inter-stroke properties, our method successfully groups the large-scale coarse strokes and at the same time preserves the fine details. We further present three challenging examples in Figures 13-15 where coarse-scale rough strokes and small-scale fine strokes exist in the same sketch. Our method achieves meaningful simplified results in all these cases. While the parameters in our system are generally fixed, changing the parameters (T_D, T_A, T_B, and T_C) may change stroke interpretation and region interpretation as well. In Figure 13(c), increasing T_C results in a tighter constraint in perceptual region classification and marks the mouth region as a non-perceptual region, so the strokes around the mouth region are grouped together. In Figure 15(d), decreasing T_D and T_B results in tighter constraints in stroke grouping and hence the sketch is simplified less aggressively. We also compare our results with the results generated by [Barla et al. 2005a] and [Shesh and Chen 2008] in Figures 1, 14, and 15. When facing these challenging sketches, the existing methods generally either leave coarse-scale strokes ungrouped or over-group fine-scale strokes. This is because that the existing methods only analyze inter-stroke properties in an absolute sense without considering closure. In contrast, our method achieves higher degree of simplicity while preserving details by accounting for closure. Figure 16 shows an illustrative example where the input strokes are quite rough, but our method can still successfully identify the perceptual regions and group the strokes. Though the method proposed by [Noris et al. 2012] could achieve similar stroke grouping result, their method requires manual labeling before stroke grouping. In comparison, our method interprets the semantics of the strokes fully automatically. We further present a comparison in Figure 17. While our goal is to analyze how likely a sketchy stroke belongs to a perceptual stroke with closure in mind, Noris et al.'s method [2012] serves a different purpose that tries to group the strokes belonging to the same higher-level logical part (not necessarily a region) of the drawing. Since a region or even a logical part could be depicted by multiple perceptual strokes, our grouping is less aggressive than that of Noris et al.'s method. For further evaluation, we invited a user to manually group these sketchy strokes according to our goal. The comparison is shown in Figure 17(d).

We also demonstrate a potential application of our interpretations in Figure 18. With the computed region interpretation, we can recover



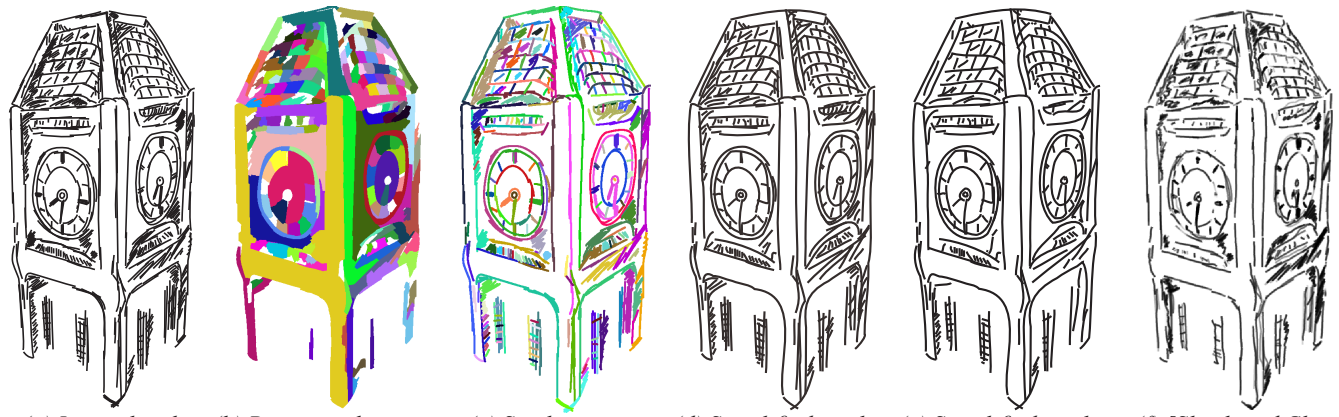
(a) Input sketch (b) Simplification result 1 (T_C =0.5) (c) Simplification result 2 (T_C =0.6)

Figure 13: *"Grandpa". The input sketch contains* 194 *input strokes and* 1499 *initial regions.* 58 *stroke groups and* 21 *perceptual regions are obtained after simplification. The whole simplification process takes* 1.7 *minutes. Here, (b) is generated using the default parameters.*



(a) Input sketch (b) Stroke groups (c) Simplified strokes (d) [Barla et al. 2005] (ε=12)

Figure 14: *"House". The input sketch contains* 187 *input strokes and* 827 *initial regions.* 71 *stroke groups and* 23 *perceptual regions are obtained after simplification. The whole simplification process takes* 2.3 *minutes.*



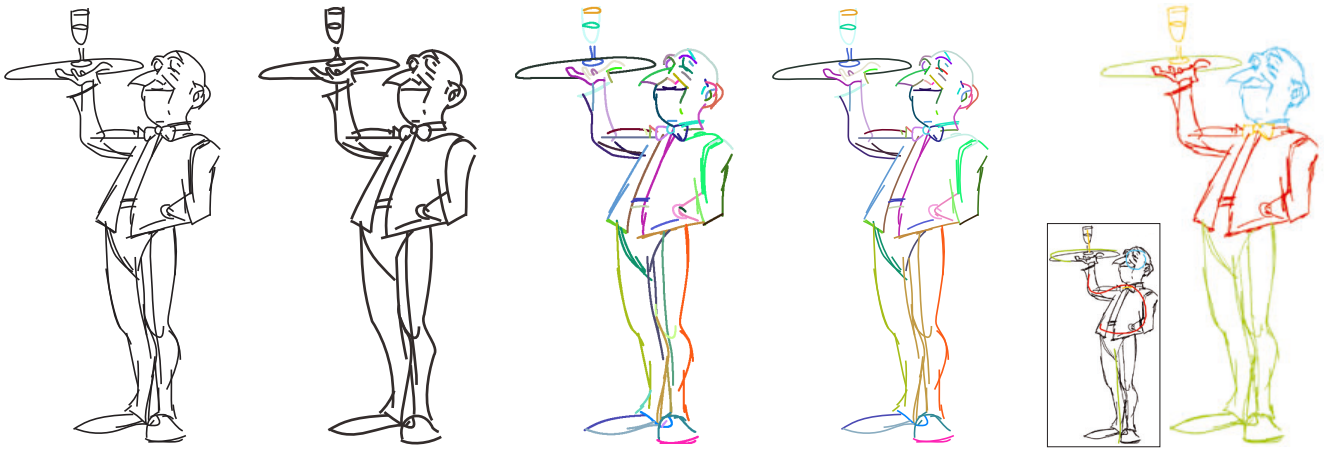
(a) Input sketch (b) Perceptual regions (c) Stroke groups (d) Simplified strokes (T_D=0.4, T_B=0.6) (e) Simplified strokes (T_D=1.0, T_B=2.0) (f) [Shesh and Chen 2008]

Figure 15: *"Clock". (a)-(e) The input sketch is re-traced from [Shesh and Chen 2008] and contains* 820 *input strokes and* 425 *initial regions. Our simplification process takes* 9.9 *minutes and obtains* 246 *stroke groups and* 198 *perceptual regions. Here, (e) is generated using the default parameters. (f) Result from [Shesh and Chen 2008] produced from their original input.*

a 2.5D layer map for the input sketch by analyzing occlusion relationships among the regions [Dimiccoli and Salembier 2009b; Dimiccoli and Salembier 2009a; Liu et al. 2013]. Our simplified result is presented in (b), and the identified perceptual regions are extended in (c). By analyzing the T-junctions, we can construct a region ordering graph that represents the occlusion relationships among the regions. Finally, by topologically sorting the ordering graph, we obtain a layer map (a rough depth map) in (d) that can be further used for 3D geometry recovery, or stereoscopic rendering.

Timing statistics All our experiments are conducted on PC with 3.1GHz CPU, 32 GB system memory. The total computational time for each sketch is reported in the corresponding caption. We observe that the computational time highly depends on the number of the input strokes and the number of the initial regions. Currently, the whole system is implemented with Matlab. The code is not optimized and no GPU is used.

Limitations Our assumption that important strokes like silhouettes and suggestive contours constitute the boundaries of closed regions may be violated. That is, there might be strokes that do no depict any region. Under this circumstance, we can only analyze inter-stroke properties in an absolute sense. This may lead to similar over-grouping or under-grouping problems as resulted in the existing closure-unaware methods. Moreover, when a small-scale perceptual region is depicted by very coarse

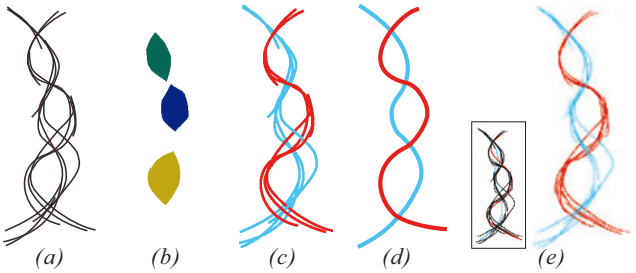


(a) Input sketch *(b) Simplified strokes* *(c) Stroke groups* *(d) Manual labeling* *(e) [Noris et al. 2011]*

Figure 17: *"Waiter". (a)-(c) The input sketch is re-traced from [Noris et al. 2011] and contains* 126 *input strokes and* 93 *initial regions. Our simplification process takes* 0.9 *minutes and obtains* 85 *stroke groups and* 36 *perceptual regions. (d) Manually labeled stroke groups. (e) Result from [Noris et al. 2011] produced from their original input. The small image on the left shows their manual label. ©Walt Disney Animation Studios.*

(a) Input *(b) Simplified strokes* *(c) Extended perceptual regions* *(d) Layer map*

Figure 18: *"Duck". The input sketch contains* 167 *input strokes and* 1271 *initial regions.* 31 *stroke groups and* 14 *perceptual regions are obtained after simplification. The whole simplification process takes* 2.0 *minutes. In the layer map, darker pixels are nearer to the view point while brighter pixels are further away.*



(a) *(b)* *(c)* *(d)* *(e)*

Figure 16: *"Spiral". (a) Input sketch re-traced from [Noris et al. 2011]. (b) Perceptual regions. (c) Stroke groups. (d) Simplified strokes. (e) Result from [Noris et al. 2011], and the small image on the left shows the manual label. The input sketch contains* 12 *input strokes and* 44 *initial regions. Our simplification process takes* 0.5 *minutes and obtains* 2 *stroke groups and* 3 *perceptual regions. ©Walt Disney Animation Studios.*

strokes, we may misclassify this region as a non-perceptual region and over-group the surrounding strokes. Besides, our current implementation does not incorporate user control. Nevertheless, it is feasible to incorporate such control by regarding the user input as hard constraints during the iterations. Finally, we do not explicitly handle closed curves in the current implementation. We may further study whether the curves are closed based on our region interpretation.

8 Conclusion

In this paper, we present a novel closure-aware sketch simplification method that converts rough sketches into clean line drawings. Our key contribution is the introduction of closure in the semantic analysis of sketches. While the grouping of strokes is influenced by the perceptual regions, regions are in turn formed and influenced by the perceptual strokes. So we further propose a novel iterative cyclic refinement system to resolve this chicken-or-the-egg problem. Comparing to the existing methods that only rely on lower-level inter-stroke properties in an absolute sense, our method accomplishes stroke grouping in a higher-level semantic sensible fashion.

While our current method does not explicitly handle decorative strokes and closed strokes, we shall study how to unify decorative strokes and closed strokes into our framework. Further study is needed for identifying such user intention. Also, the determined region interpretation may be more meaningful for other applications such as segmentation and colorization.

Acknowledgements

This project is supported by Hong Kong RGC General Research Fund (Project No. CUHK14200915 and CUHK417411), NSFC (Project No. 61272293), and Shenzhen Nanshan Innovative Institution Establishment Fund (Project No. KC2013ZDZJ0007A).

References

BAE, S., BALAKRISHNAN, R., AND SINGH, K. 2008. Ilovesketch: as-natural-as-possible sketching system for creating 3d curve models. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology, Monterey, CA, USA, October 19-22, 2008*, 151–160.

BARLA, P., THOLLOT, J., AND SILLION, F. X. 2005. Geometric clustering for line drawing simplification. In *Proceedings of the Eurographics Symposium on Rendering Techniques*, 183–192.

BARLA, P., THOLLOT, J., AND SILLION, F. X. 2005. Geometric clustering for line drawing simplification. In *ACM SIGGRAPH 2005 Sketches*, ACM, 96.

BAUDEL, T. 1994. A mark-based interaction paradigm for free-hand drawing. In *ACM Symposium on User Interface Software and Technology*, 185–192.

CHEN, J., GUENNEBAUD, G., BARLA, P., AND GRANIER, X. 2013. Non-oriented MLS gradient fields. *Comput. Graph. Forum 32*, 8, 98–109.

CHIEN, Y., LIN, W.-C., HUANG, T.-S., AND CHUANG, J.-H. 2014. Line drawing simplification by stroke translation and combination. In *The 5th International Conference on Graphic and Image Processing*, 90690X–90690X.

DEUSSEN, O., AND STROTHOTTE, T. 2000. Computer-generated pen-and-ink illustration of trees. In *SIGGRAPH*, 13–18.

DIMICCOLI, M., AND SALEMBIER, P. 2009. Exploiting t-junctions for depth segregation in single images. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2009, 19-24 April 2009, Taipei, Taiwan*, 1229–1232.

DIMICCOLI, M., AND SALEMBIER, P. 2009. Hierarchical region-based representation for segmentation and filtering with depth in single images. In *Proceedings of the International Conference on Image Processing, ICIP 2009, 7-10 November 2009, Cairo, Egypt*, 3533–3536.

FU, H., ZHOU, S., LIU, L., AND MITRA, N. J. 2011. Animated construction of line drawings. *ACM Trans. Graph. 30*, 6, 133.

GRABLI, S., DURAND, F., AND SILLION, F. X. 2004. Density measure for line-drawing simplification. In *The 12th Pacific Conference on Computer Graphics and Applications (PG 2004)*, 309–318.

GRIMM, C., AND JOSHI, P. 2012. Just drawit: A 3d sketching system. In *Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling*, SBIM '12, 121–130.

KARA, L. B., D'ERAMO, C. M., AND SHIMADA, K. 2006. Pen-based styling design of 3d geometry using concept sketches and template models. In *Proceedings of the 10th ACM Symposium on Solid and Physical Modeling*, 149–160.

LINDLBAUER, D., HALLER, M., HANCOCK, M. S., SCOTT, S. D., AND STUERZLINGER, W. 2013. Perceptual grouping: selection assistance for digital sketching. In *The ACM International Conference on Interactive Tabletops and Surfaces, ITS '13, St Andrews, United Kingdom - October 06 - 09, 2013*, 51–60.

LIU, X., MAO, X., YANG, X., ZHANG, L., AND WONG, T.-T. 2013. Stereoscopizing cel animations. *ACM Trans. Graph. 32*, 6, 223.

LIU, J., FU, H., AND TAI, C. 2014. Dynamic sketching: simulating the process of observational drawing. In *Proceedings of the Workshop on Computational Aesthetics, CAe '14, Vancouver, British Columbia, Canada, August 8-10, 2014*, 15–22.

NAN, L., SHARF, A., XIE, K., WONG, T.-T., DEUSSEN, O., COHEN-OR, D., AND CHEN, B. 2011. Conjoining gestalt rules for abstraction of architectural drawings. *ACM Trans. Graph. 30*, 6, 185.

NORIS, G., SÝKORA, D., SHAMIR, A., COROS, S., WHITED, B., SIMMONS, M., HORNUNG, A., GROSS, M. H., AND SUMNER, R. W. 2012. Smart scribbles for sketch segmentation. *Comput. Graph. Forum 31*, 8, 2516–2527.

NORIS, G., HORNUNG, A., SUMNER, R. W., SIMMONS, M., AND GROSS, M. H. 2013. Topology-driven vectorization of clean line drawings. *ACM Trans. Graph. 32*, 1, 4.

ORBAY, G., AND KARA, L. B. 2011. Beautification of design sketches using trainable stroke clustering and curve fitting. *IEEE Trans. Vis. Comput. Graph. 17*, 5, 694–708.

PAVLIDIS, T., AND WYK, C. J. V. 1985. An automatic beautifier for drawings and illustrations. In *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 1985)*, 225–234.

PREIM, B., AND STROTHOTTE, T. 1995. Tuning rendered line-drawings. In *Proceedings of the 3rd Central European Conference on Computer Graphics*, 227–237.

PUSCH, R., SAMAVATI, F. F., NASRI, A. H., AND WYVILL, B. 2007. Improving the sketch-based interface. *The Visual Computer 23*, 9-11, 955–962.

QU, Y., WONG, T.-T., AND HENG, P.-A. 2006. Manga colorization. *ACM Trans. Graph. 25*, 3, 1214–1220.

ROSIN, P. L. 1994. Grouping curved lines. In *Proceedings of the British Machine Vision Conference (BMVC 1994)*, 1–10.

SAUND, E., AND MORAN, T. P. 1994. A perceptually-supported sketch editor. In *ACM Symposium on User Interface Software and Technology*, 175–184.

SHESH, A., AND CHEN, B. 2008. Efficient and dynamic simplification of line drawings. *Comput. Graph. Forum 27*, 2, 537–545.

SÝKORA, D., DINGLIANA, J., AND COLLINS, S. 2009. Lazybrush: Flexible painting tool for hand-drawn cartoons. *Comput. Graph. Forum 28*, 2, 599–608.

WERTHEIMER, M. 1923. Untersuchungen zur lehre von der gestalt. ii. *Psychological Research 4*, 1, 301–350.

WILSON, B., AND MA, K.-L. 2004. Rendering complexity in computer-generated pen-and-ink illustrations. In *Proceedings of the 3rd International Symposium on Non-Photorealistic Animation and Rendering*, 129–137.

ZHANG, S.-H., CHEN, T., ZHANG, Y.-F., HU, S.-M., AND MARTIN, R. R. 2009. Vectorizing cartoon animations. *IEEE Trans. Vis. Comput. Graph. 15*, 4, 618–629.