EUROGRAPHICS 2023 / K. Myszkowski and M. Nießner
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 42 (2023), Number 2

Img2Logo: Generating Golden Ratio Logos from Images

Kai-Wen Hsiao[1], Yong-Liang Yang[2], Yung-Chih Chiu[1], Min-Chun Hu[1], Chih-Yuan Yao[3], Hung-Kuo Chu[1]

[1]National Tsing Hua University
[2]University of Bath
[3]National Taiwan University of Science and Technology

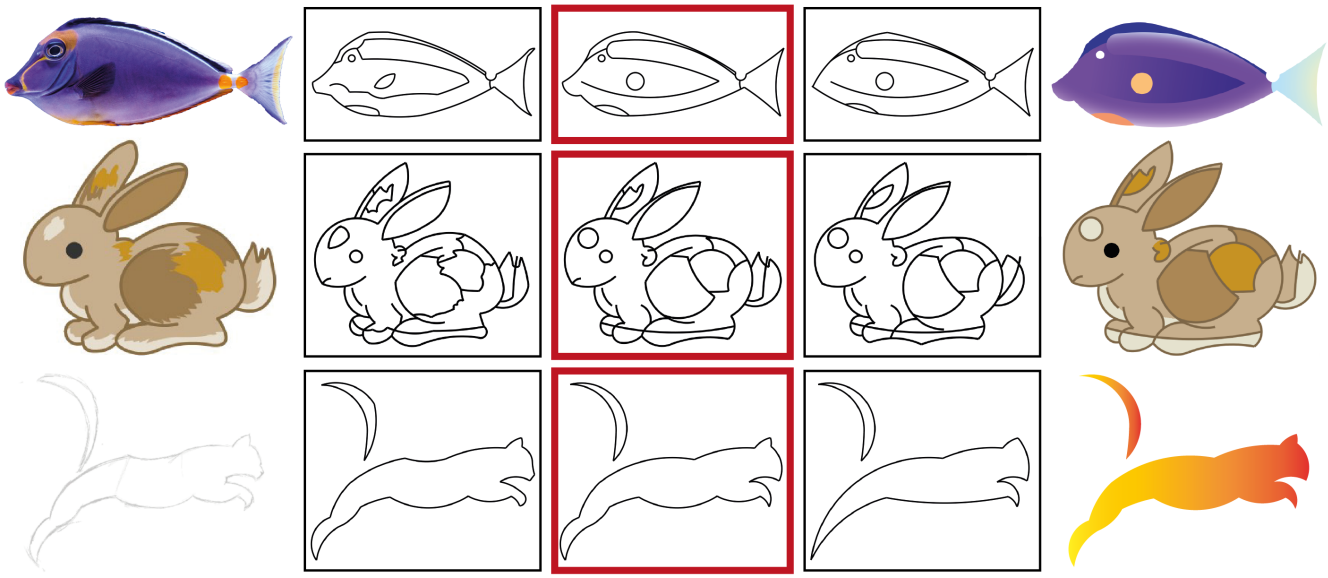


Figure 1: Our framework can automatically convert input images of various forms (from top to bottom: real photo, clip art, and line drawing) into logo abstractions consisting of circular arcs with golden ratios. Our progressive optimization approach can efficiently generate sequential results with increasing abstraction level (middle). The user can easily edit our output (highlighted in red box) by filling colors to create a high-quality logo (right).

Abstract

Logos are one of the most important graphic design forms that use an abstracted shape to clearly represent the spirit of a community. Among various styles of abstraction, a particular golden-ratio design is frequently employed by designers to create a concise and regular logo. In this context, designers utilize a set of circular arcs with golden ratios (i.e., all arcs are taken from circles whose radii form a geometric series based on the golden ratio) as the design elements to manually approximate a target shape. This error-prone process requires a large amount of time and effort, posing a significant challenge for design space exploration. In this work, we present a novel computational framework that can automatically generate golden ratio logo abstractions from an input image. Our framework is based on a set of carefully identified design principles and a constrained optimization formulation respecting these principles. We also propose a progressive approach that can efficiently solve the optimization problem, resulting in a sequence of abstractions that approximate the input at decreasing levels of detail. We evaluate our work by testing on images with different formats including real photos, clip arts, and line drawings. We also extensively validate the key components and compare our results with manual results by designers to demonstrate the effectiveness of our framework. Moreover, our framework can largely benefit design space exploration via easy specification of design parameters such as abstraction levels, golden circle sizes, etc.

CCS Concepts

• ***Computing methodologies*** → *Computer Graphics;*

© 2023 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

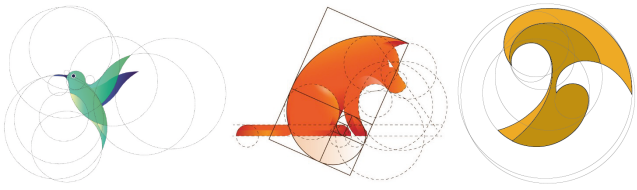
1. Introduction

Logos are regarded as an effective medium to convey the designer's spirit, idea, and intention via a concise 2D pattern. Such an intriguing nature lays the popularity of logo design in building recognition. A well-designed logo is more than creating a perfect visual brand mark of a company, group, character, *etc*, it also helps people to understand the purpose and value it represents. Logos can be generally categorized into three types, including text-only logo, pictorial logo, and combination logo [Eva21]. We found the *pictorial* logos particularly interesting and challenging. It requires the designer to compose an abstracted shape of an object only using simple geometric primitives (e.g., points, lines, arcs, *etc*).

According to several online courses regarding pictorial logo design process [Sal19; Bok18; Mar22], a custom grid system that contains specifically arranged squares with inscribed circles (as shown in Fig. 2b and Fig. 3a) is commonly used by the designers. Compared with other grid systems (e.g., uniform grid) formed by families of straight lines, it not only provides modulated arcs at different scales for a better approximation of curved shapes, but also correlates to the famous Fibonacci sequence and golden ratio [Mei18] in terms of the grid size and circle radii. In this paper, we call the logos generated based on modulated circular arcs with golden ratios in-between as the *golden ratio* logos. Fig. 2 shows some golden ratio logos created by designers.

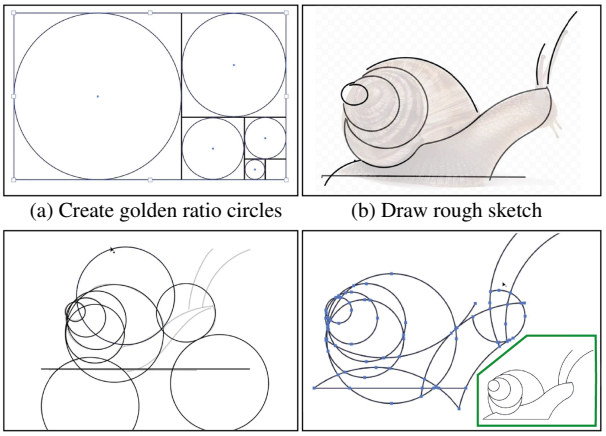
However, as shown in Fig. 3, existing modern graphic design tools (e.g., Adobe Illustrator) only assist designers by computing a set of golden ratio circles. The designer needs to manually place circles and trace arcs in a trial-and-error process to depict the target shape, which is rather skill-demanding and time-consuming. Although some computational methods exist for a similar flat design by converting images into pictograms (*i.e.,* icons) [LALR16; SATV18; LBK17; KBYU20] or cliparts [FLB17], the final results are fundamentally different from golden ratio logos due to no constraint on using circles with specific sizes. Therefore, there is still a lack of an effective tool for the present design context.

In this work, we present a novel computational design framework to automatically abstract input image objects into logo-like shapes that strictly adhere to the golden ratio design constraints (see Fig. 1). The keys to our framework are i) a set of design principles identified from existing designs created by designers in practice; ii) a novel formulation that implements the design principles as a constrained optimization problem; and iii) a tailor-made algorithm that solves the optimization problem efficiently. The result is an appealing abstracted form where all the prominent feature lines (contours, semantic boundaries, etc.) of the input object are represented by circular arcs extracted from a set of golden ratio circles. These circular arcs all together well approximate the characteristics of the input object. Moreover, our framework is flexible and supports intuitive design exploration. Specifically, we devise a progressive optimization strategy to generate results with increasing abstraction levels, offering multiple initial design choices for the designer. The designer can also alter the design constraints such as specifying the set of golden ratio circles (e.g., size, number), locally guiding the abstraction level using scribbles, *etc* (see supplemental video). We demonstrate the efficacy of our framework on a wide variety of input images, including real photos, clip arts, and line drawings. We further evaluate our framework by validating its key components, comparing the results with those created by designers, exhibiting the design exploration possibilities, and conducting a user study.



(a) (b) (c)

Figure 2: **Examples of golden ratio logos.** Designers utilize a bounded set of golden ratio circles to compose the final logo designs. In practice, designers usually use Fibonacci sequence grid (see (b) and Fig. 3a) to approximate a set of golden ratio radii because Fibonacci sequence converges to golden ratio (between neighboring numbers) as the list grows. (From left to right: ©Sean Xue, ©DAINOGO, ©Lindsay Marsh [Mar22].)



(a) Create golden ratio circles (b) Draw rough sketch

(c) Fit sketch with circles (d) Arc extraction

Figure 3: **Golden ratio logo design workflow.** The designers start with a set of golden ratio circles (a) to approximate a rough sketch of the target shape (b). They have to manually place circles to fit the sketch (c), and trace out the circular arcs to generate the final design in the green box (d) (©Design Quickies).

In summary, our work makes the following contributions:

- We present the first computational design framework for automatic generation of logo abstractions with golden ratio circular arcs from raster images.
- We identity a set of design principles and formulate the whole design problem as a constrained optimization guided by the principles.
- We devise a progressive optimization approach, which is efficient and capable of producing results with increasing abstraction levels to facilitate the exploration of design space.

© 2023 Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

2. Related Work

2.1. Image Abstraction

Our work is related to a broad topic called image abstraction, which aims at automatically converting an input image into an abstracted form for computational design purposes. During the abstraction process, information loss is inevitably introduced. Hence the key challenge is how to preserve the content of the original image subject to the abstraction constraints. In practice, researchers have explored different types of constraints regarding different image properties. For instance, image color can be restricted to quantized color space [Hec82; OB91], low regional variations [DS02; WOG06], limited tones [GRG04; RL10], etc. Image resolution can also be reduced to generate pixel-art-like abstraction [GDA*12; HWH*18; KSP13]. Image content can be represented by simple shape primitives such as points [MARI17], edges [KD09], curves [BLP10; MS11], parts [MDS09], characters [ZCR*16], collages [HZZ11], etc. Image style can be enforced to match a painting style [JYF*20]. To our best knowledge, the only work in the same context (golden ratio circular arc abstraction) as ours is [IMF22]. However, it assumes that the golden ratio circles (with both center and radius) are provided by the designer beforehand. Its contribution is limited to determining a subset of regions intersected from the input circles to approximate the reference shape. In contrast, our work can automatically generate golden ratio circle layout that well approximates the raw image content without any input from the designer, which is a much harder computational design problem to solve.

2.2. Icon-like Image Generation

How to computationally generate an icon-like image has been explored in the field. Traditional approaches are based on carefully designed algorithmic procedures to iconify a target concept/object. Liu *et al.* [LALR16] present a 2D data-driven image iconifiction framework by combining pieces from existing pictogram examples. Favreau *et al.* [FLB17] convert photos to clip arts by stacking multiple gradients using opaque and semi-transparent layers. Lin *et al.* [LYC18] generate icon-like binary images from 3D shapes via a joint 2D/3D optimization according to a set of flat design principles. Zhao *et al.* [ZKH*20] propose a system that can automatically generate compound icons based on textual queries from an annotated icon dataset. With the recent advances of deep learning, learning-based approaches exploit the representation power of neural networks for icon-like image generation. Sage *et al.* [SATV18] build a dataset called LLD with 600k+ logos, and train generative adversarial networks (GANs) conditioned on clustered labels for logo synthesis and manipulation. Karamatsu *et al.* [KBYU20] utilize CycleGAN [ZPIE17] and UNIT [LBK17] as the backbone networks to translate real-world images to icon-like images. Our work falls into the procedural category which does not require any data collection and model learning. The output circular arcs are guaranteed to satisfy radius constraints while respecting the input shape, thus can largely benefit the application of golden ratio icon design.

2.3. Line Drawing Vectorization

Another related topic (but with a different goal) is line drawing vectorization, which studies how to convert rasterized strokes into vectorized curves with compact representations, such that there is no visual artifacts when up-scaling the image (smooth shading can also be created based on a vector-based curve primitive [OBW*08]). Prior works can also be classified into non-learning-based and learning-based approaches. The former generally perform stroke decomposition to the input line drawings followed by curve approximation [NHS*13; FLB16; LRS18; BS19; SBBB20], while the latter relies on a pre-trained model to infer curve segmentation or/and representation [SII18a; SII18b; GZH*19; EVA*20; DYH*21; PNCB21]. Our work also requires boundary curve decomposition and approximation, and outputs a vectorized shape. The main difference is that our work is heavily constrained. The curve representation is restricted to circles, and the radii of the circles must have golden ratios in-between. Besides, less number of circles is desirable to meet the practical design requirement.

3. Design principles

By studying existing design guidelines and practices [Sal19; Bok18], we identify a number of principles to guide our design of optimization objectives and constraints as follows.

Golden ratio. As the requirement of the golden ratio design, the abstraction result is enforced to comprise circular arcs extracted from a set of golden ratio circles. This means that the solution space is highly constrained as the underlying circle radii cannot be arbitrary but only discrete, and the ratio between any two circles must be one or multiples of the golden ratio ϕ.

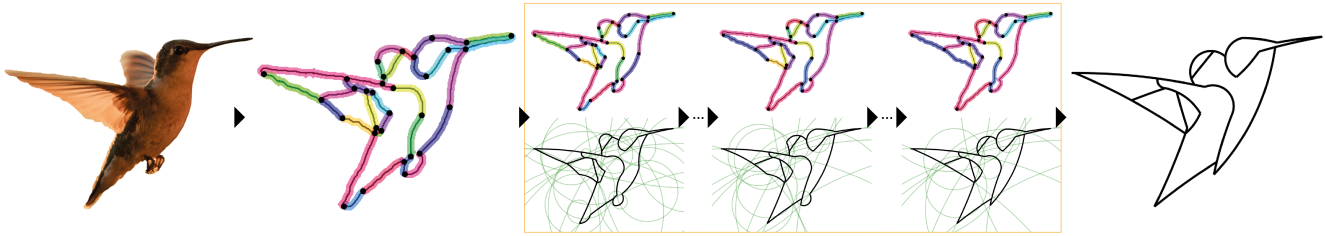
Simplicity. For abstraction and design purposes, the result is expected to contain as few number of golden ratio circles as possible. This can not only provide a concise presentation, but also favor the designers to manually adjust and fine-tune the result.

Similarity. Although the circle radius and number are constrained, the abstracted form should well approximate the target shape, allowing a correct comprehension of the shape content and its prominent feature lines such as contours, semantic boundaries, *etc*.

Connectivity. As an approximation of feature line segments (we call them *line segments* for short hereafter), the extracted circular arcs are required to preserve the connectivity of their corresponding line segments. This topology preservation is crucial to avoid visual artifacts such as isolated arcs, erroneous shape components, *etc*.

Note that some principles conflict with each other. For example, to optimize the similarity, a trivial solution is to increase the number of circular arcs while reducing their lengths, and only perform arc fitting at the vicinity of each pixel. But this is obviously against the simplicity. Furthermore, the problem has a complicated combinatorial nature due to decomposing/merging line segments and approximation using circles with discrete radii. This makes search-based method infeasible, *i.e.,* the exhaustive search is computationally prohibitive while the stochastic search is inefficient and non-intuitive. To tackle this problem, we propose a progressive optimization procedure by incrementally merging line segments and fitting circular arcs. The main idea resembles the strategy employed by the mesh simplification [GH97] and shape approximation [LZZ*21]. It brings several advantages including making the

© 2023 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.



(a) Input (b) Preprocessing (Sec. 4) (c) Progressive optimization (Sec. 5) (d) Abstraction result

Figure 4: **Framework overview**. Given an input image (a), we first extract prominent feature lines (b) in a preprocessing step, then perform progressive optimization to fitting circular arcs onto line segments while merging them (c), resulting in an abstracted shape (d) that respects golden ratio design principles. Here we use different colors to depict individual line segments in (b). For clarity we only sample three abstractions during the progressive optimization in (c).

optimization problem tractable and providing sequential approximations with continuous abstraction levels that largely benefit the design exploration.

4. Overview

Given an input 2D object image, our framework can automatically abstract the shape of object by representing its prominent features (contours, semantic boundaries, etc.) using circular arcs with golden ratios, such that the resultant abstracted form is composed of a set of 'golden' circular arcs that adheres the proposed design principles. Fig. 4 illustrates an overview of our framework. In the preprocessing stage, to facilitate the subsequent circular arc fitting, we first extract from the input raster image (Fig. 4a) a set of line segments to represent the shape of object (Fig. 4b). In the main stage, we abstract line segments using golden ratio circular arcs in a bottom-up manner. Specifically, the system incrementally fits golden ratio circular arcs to line segments and merges neighboring line segments while minimizing the geometric approximation error and preserving the connectivity (i.e., the junction points) between line segments. During the optimization, the system continuously outputs a sequence of golden ratio representations with lessened circular arcs and increased abstraction level (Fig. 4c). The whole process ends when the approximation error exceeds a pre-defined threshold, yielding the final abstraction result (Fig. 4d).

Preprocessing. The goal in this stage is to convert a 2D image into a set of representative line segments that can well depict the shape of object in terms of contours and semantic boundaries. We leverage several existing image processing techniques to handle images of different contexts. For the real photo, we first perform a flow-based image abstraction [KD09] followed by an image thinning to extract salient feature lines at pixel-level. In the context of clip art, a simple flood-fill method followed by tracing contours of separated regions can do the task well. For the artistic line drawing in the real world scenario, advanced sketch simplification method such as [LWH15] can be utilized to obtain its clean and concise counterpart. Then we over-segment the extracted feature lines using Ramer–Douglas–Peucker algorithm [DP73] and obtain a set of line segments for the later optimization. Note that each line segment is represented as a sequence of pixels.

5. Algorithm

In this section, we present the algorithmic details of our framework. First we introduce the overall problem formulation (Sec. 5.1). Then we present our progressive approach based on iterative optimizations (Sec. 5.2), followed by elaborating the optimization procedure within each iteration (Sec. 5.3). Finally, we include further details on algorithm acceleration (Sec. 5.4).

5.1. Problem Formulation

Our algorithm is essentially formulated on the line segments obtained from the preprocessing step and denoted as $\mathcal{L} = \{\mathcal{S}, \mathcal{J}\}$, where $\mathcal{S}$ comprises individual line segments and $\mathcal{J}$ represents the connectivity (*i.e.,* the junction points) between line segments. The goal is to abstract $\mathcal{L}$ using a set of circular arcs $\mathcal{A}$ while respecting the design principles in Sec. 3. We formulate the abstraction process as a constrained optimization problem:

$$\min_{\mathcal{A}} |\mathcal{A}| \quad \text{s.t. } G^r(\mathcal{A}),\ E^s(\mathcal{A},\mathcal{L}) < T,\ G^c(\mathcal{A},\mathcal{L}), \tag{1}$$

that minimizes the objective $|\mathcal{A}|$, representing the number of abstracted circular arcs (the *simplicity* principle), subject to the *golden ratio* (G^r), *similarity* (E^s), and *connectivity* (G^c) constraints. Fig. 5 shows an example illustration of using 3 circular arcs to abstract 8 line segments.

Golden ratio constraint. $G^r(\mathcal{A})$ ensures that any pair of circular arcs in the set $\mathcal{A}$ must follow the golden ratio principle. The constraint is defined as:

$$r_i / r_j = \phi^z \quad \forall a_i, a_j \in \mathcal{A},\ r_i \geq r_j,\ z \in \mathbb{Z}^*, \tag{2}$$

where a and r represent respectively the circular arc and its radius, and the ratio between two radii must be a power (z is a non-negative integer) of *golden ratio* $\phi = (1+\sqrt{5})/2$.

Similarity constraint. $E^s(\mathcal{A},\mathcal{L})$ addresses the similarity between the abstracted arc set $\mathcal{A}$ and the input shape $\mathcal{L}$ as:

$$E^s(\mathcal{A},\mathcal{L}) = \frac{1}{|\mathcal{A}|}\sum_i E^f(a_i,\mathcal{S}_i) + \frac{1}{|\mathcal{J}'|}\sum_k \|\mathbf{d}_k\| \tag{3}$$

where the first term is the average Fréchet distance [EM94] over circular arc $a_i \in \mathcal{A}$ and its associated line segments $\mathcal{S}_i$ (*i.e.,* fitting

© 2023 Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

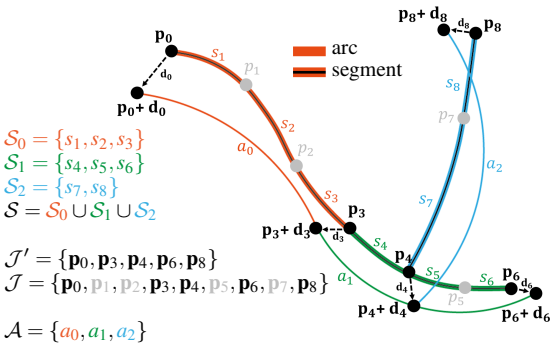


Figure 5: **Example illustration of our formulation.** A line segment set $\mathcal{L} = \{\mathcal{S}, \mathcal{J}\}$ is abstracted using a circular arc set $\mathcal{A}$ with golden ratios.

$\forall s \in \mathcal{S}_i$ with a_i). $\mathcal{S}_i$ is a subset of $\mathcal{S}$ and $\mathcal{S} = \bigcup \mathcal{S}_i$. The second term measures how far the abstracted circular arcs deviate from the original line segments at the junctions $\mathcal{J}'$ using the junction displacement vector $\mathbf{d}_k$ (see the definition of $\mathcal{J}'$ in the paragraph below). During optimization, we constrain $E^s(\mathcal{A}, \mathcal{L})$ within a user-specified tolerance T.

Connectivity constraint. $G^c(\mathcal{A}, \mathcal{L})$ is designed for preserving the connectivity between line segments in the final abstraction. As shown in Fig. 5, as a circular arc a_i may associate with multiple line segments $s \in \mathcal{S}_i$, we denote a junction set $\mathcal{J}'$, which contains junction points connecting between $\mathcal{S}_i$'s (thus is a subset of $\mathcal{J}$). We say that $\mathcal{A}$ is topologically consistent with $\mathcal{L}$ if for each junction point $\mathbf{p}_k \in \mathcal{J}'$ where $\mathcal{S}_i$ connects to, the associated arc a_i must connect to a unique junction point $\mathbf{p}_k + \mathbf{d}_k$ with $\mathbf{d}_k$ indicating a displacement vector. Thus, we formulate this constraint as:

$$G^c(\mathcal{A}, \mathcal{L}) \equiv \Phi(a_i, \mathbf{p}_k + \mathbf{d}_k) = 0 \quad \forall \mathcal{S}_i \in \Omega(\mathbf{p}_k), \forall \mathbf{p}_k \in \mathcal{J}'. \quad (4)$$

where Φ is the implicit representation of arc a_i and $\Omega(\mathbf{p}_k)$ contains all segments that connect to $\mathbf{p}_k$. Fig. 6 illustrates the effectiveness of the connectivity constraint.

Finding the globally optimal arc set $\mathcal{A}^*$ in Eqn. 1 is extremely difficult. First, the problem has combinatorial complexity due to the enormous feasible combinations of the input line segments ($\mathcal{S}_i$'s) for abstracting circular arcs (a_i's). Second, the problem is highly constrained with different types of constraints. In the next section, we propose a progressive optimization approach that iteratively merges line segments and fits golden ratio circles.

5.2. Progressive Approach

To make the optimization in Eqn. 1 tractable and efficient, we explore the design space with increasing abstraction levels by iteratively merging neighboring line segments. Which pair of line segments to merge first depends on how well the similarity and connectivity constraints are satisfied.

Note that this strategy resembles the manual design practices of designers (see supplemental video), who favor long circular arcs for abstracting continuous line segments.

More precisely, given the input line segments $\mathcal{L} = \{\mathcal{S}, \mathcal{J}\}$, at the beginning of our algorithm, we first fit the initial segment set $\mathcal{S}^0 = \mathcal{S}$ with the arc set $\mathcal{A}^0$ (we elaborate this step in Sec. 5.3). Then we iteratively simplify the line segments and optimize the arc set by merging line segments one pair at a time. To this end, we denote a set that contains all mergeable line segment pairs as $\mathcal{M} = \{(s_i, s_j) | s_i, s_j \in \mathcal{S}^0\}$. A *mergeable line segment pair* is defined as the two line segments that share at least one end point. To determine the priority of the line segments to merge, for each pair in $\mathcal{M}$ we calculate a merge error (i.e., similarity loss caused by fitting less arcs due to the merge) compared with $E^s(\mathcal{A}^0, \mathcal{L}^0)$, where $\mathcal{L}^0 = \mathcal{L}$.

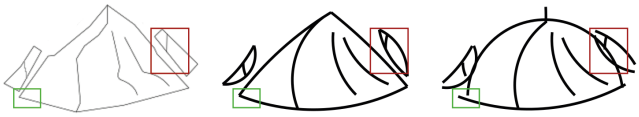


Figure 6: **Connectivity preservation.** We use connectivity constraint to ensure *topology consistency* between the original line segments (a) and the abstracted circular arcs (b). Otherwise the result (c) can either have disconnected arcs (green box), or redundant intersections (red box) that are not expected as no such intersection in the original line segments.

However, in practice, this local operation may occasionally cause *redundant intersection* where two circular arcs intersect at a point not belonging to the junction set (see the red boxes in Fig. 6). Hence, we introduce a penalty term in the merge error to prohibit the occurrence of redundant intersection.

Overall, our algorithm iteratively performs the following steps for each iteration t with t starting from 0:

1. Perform golden ratio arc fitting to $\mathcal{L}^t$, resulting in arc set $\mathcal{A}^t$ (Sec. 5.3).
2. Evaluate the similarity score $E^s(\mathcal{A}^t, \mathcal{L}^t)$ between $\mathcal{A}^t$ and $\mathcal{L}^t$. If it is larger than the given tolerance T, stop and output $\mathcal{A}^{t-1}$ as the result.
3. Select a mergeable line segment pair (s_i, s_j) with the smallest merge error among all candidates in $\mathcal{M}$.
4. Merge pair (s_m, s_n) into $s_{m,n}$ and generate new line segment representation $\mathcal{L}^{t+1}$ with reduced number of segments.
5. Update the merge error for segment pairs in $\mathcal{L}^{t+1}$ that contain $s_{m,n}$.
6. $t = t + 1$ and go back to (1).

5.3. Fitting Arcs with Golden Ratio

In each iteration of the progressive optimization, given a line segment representation $\mathcal{L}$ with the corresponding line segment set $\mathcal{S}$ and junction set $\mathcal{J}$ (here we omit the iteration notation t for simplicity), our goal is to find an optimal circular arc set $\mathcal{A}$ that best fits $\mathcal{S}$ (E^s) while satisfying connectivity constraint (G^c) and golden ratio constraint (G^r). The optimization problem is defined as:

$$\mathcal{A} \equiv \min_{\mathcal{A}} E^s(\mathcal{A}, \mathcal{L}) \quad \text{s.t. } G^c(\mathcal{A}, \mathcal{L}),\ G^r(\mathcal{A}). \quad (5)$$

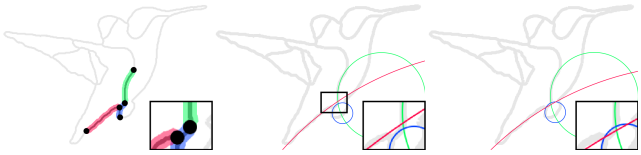
The above equation is also not easy to solve given it is highly constrained due to the discrete radii and limited angle of individual circular arcs, and the connectivity preservation between arcs.

© 2023 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.



(a) Line segments (b) Golden ratio circle fitting (c) Arc extraction

Figure 7: **Golden ratio arc fitting.** For each iteration and its line segments (a), we first perform constrained circle fitting respecting golden ratio, similarity, and connectivity principles (b), then extract circular arcs to generate the (intermediate) result in the current iteration (c).



(a) Line segments (b) Fitted circles (c) Connectivity optimized

Figure 8: **Golden ratio circle fitting.** For each iteration, we first fit circles (b) with golden ratios to individual line segments (a), then optimize the circle locations (c) under connectivity constraints of the corresponding line segments (in the same color). For clarity we only show a few circles. Note that (c) satisfies connectivity constraint as in (a), while (b) does not (blue circle and red circle are disconnected).

Our strategy is to first transfer the problem to a *constrained least square circle fitting problem* [LZZ06; Gri17], which can be efficiently solved using a general numerical solver. Then we extract arcs from the circles, and validate the similarity (difference from the input) and connectivity (with redundant intersection or not). The process is shown in Fig. 7.

Golden ratio circle fitting. We transfer Eqn. 5 to a constrained least squares circle fitting form by replacing arc with circle and rewrite the objective function as a squared error between circles and their corresponding line segments as:

$$\mathcal{C} \equiv \min_{\mathcal{C}} E^l(\mathcal{C},\mathcal{L}) \quad \text{s.t. } G^c(\mathcal{C},\mathcal{L}),\ G^r(\mathcal{C}). \tag{6}$$

The objective function E^l is defined as the sum of squared error E^{sqr} between the fitted circle c_i and its associated line segments $\mathcal{S}_i$, along with the junction displacement error:

$$E^l(\mathcal{C},\mathcal{L}) = \frac{1}{|\mathcal{S}|}\sum_i E^{sqr}(c_i,\mathcal{S}_i) + \frac{1}{|\mathcal{J}|}\sum_k \|\mathbf{d}_k\|^2. \tag{7}$$

The *connectivity constraint* G^c now forces the circles to intersect at the point $(\mathbf{p}_k+\mathbf{d}_k)$ if the corresponding line segments meet at $\mathbf{p}_k$. We formulate the constraint as:

$$G^c(\mathcal{C},\mathcal{L}) \equiv \Phi(c_i,\mathbf{p}_k+\mathbf{d}_k) = 0 \quad \forall s_i \in \Omega(\mathbf{p}_k), \forall \mathbf{p}_k \in \mathcal{J}. \tag{8}$$

We can further rewrite $\Phi(c_i,\mathbf{p}_k+\mathbf{d}_k)$ as:

$$\Phi(c_i,\mathbf{p}_k+\mathbf{d}_k) = \|\mathbf{o}_i-(\mathbf{p}_k+\mathbf{d}_k)\| - r_i, \tag{9}$$

where $\mathbf{o}_i$ and r_i denote the center point and radius of circle c_i, respectively.

To satisfy the *golden ratio constraint*, we represent the radius r_i of the circle c_i as:

$$r_i = \phi^{z_i+\sigma}, \tag{10}$$

where $z_i \in \mathbb{Z}^*$ is a non-negative integer determined by fitting a golden ratio circle to s_i. $\sigma \in [0,1)$ is called the *bias* that determines the set of discrete golden ratio circle radii (see Sec. 5.4 for how to select σ). Note that the *golden ratio constraint* is naturally satisfied based on Eqn. 10 as for any pair of radius (r_i,r_j), their ratio is:

$$\frac{r_i}{r_j} = \frac{\phi^{z_i+\sigma}}{\phi^{z_j+\sigma}} = \phi^{z_i-z_j}, \quad \forall c_i,c_j \in \mathcal{C}, r_i \geq r_j, z \in \mathbb{Z}^*, \tag{11}$$

where $z_i - z_j$ is guaranteed to be a non-negative integer.

Based on the above golden ratio radius representation, we devise a two-stage method to fit the circle set. In the first stage, we fit one circle with radius r_i for each segment s_i independently without considering the golden ratio constraint and the connectivity constraint. Then we query the golden ratio radius $\bar{r}_i$ by:

$$\bar{r}_i = \phi^{\lceil \log r_i/\log\phi - \sigma \rceil + \sigma}. \tag{12}$$

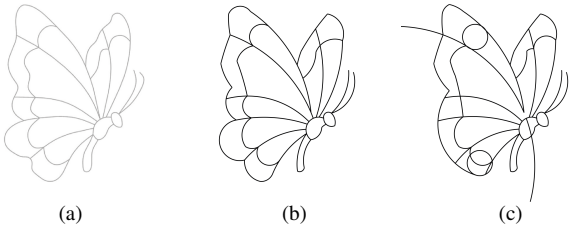
Once we have the fitted circles with golden ratio radii, we can remove the golden ratio constraint from Eqn. 6 by fixing these golden ratio radii. Finally, we obtain a general least squares circle fitting problem subject to the equality (connectivity) constraint as follows:

$$\begin{aligned} &\mathcal{C}_o \equiv \min_{\mathcal{C}_o} E^l(\mathcal{C}_o,\mathcal{L}) \\ &\text{s.t. } \|\mathbf{o}_i-(\mathbf{p}_k+\mathbf{d}_k)\| - \bar{r}_i = 0,\ \forall s_i \in \Omega(\mathbf{p}_k), \forall \mathbf{p}_k \in \mathcal{J}. \end{aligned} \tag{13}$$

Here $\mathcal{C}_o$ denotes the circles with fixed radii but unknown locations. The above equation attempts to optimize the circle location (*i.e.,* $\mathbf{o}_i$) and the junction offset (*i.e.,* $\mathbf{d}_k$) such that the similarity error is minimized and the connectivity constraint is fulfilled. An example of this two-stage method is shown in Fig. 8. Note that there is no hard constraint on either circle center or junction displacement. Thus we can always find circles that intersect with each other as expected. The worst case happens when the value of the objective function becomes very high (i.e., the circles after optimization/relocation cannot fit the line drawing well, thus harm the similarity), but it is still a feasible solution under golden ratio and connectivity constraints.

Arc extraction. After we find the optimal golden ratio circle set $\mathcal{C}$ for a given line segment set $\mathcal{L}$, the next step is to extract arc a_i from circle $c_i \in \mathcal{C}$ by determining two end points on the circle. To meet the connectivity constraint, the extracted arc a_i should contain the corresponding junctions $\{\mathbf{p}_k+\mathbf{d}_k\}$ in the same way as its associated line segment s_i connects to $\mathbf{p}_k$. Therefore, for each fitted circle c_i and its associated junction points $\{\mathbf{p}_k+\mathbf{d}_k\}$, we first enumerate the combination of two junctions to form a set of candidate arcs and then remove those arcs violating the connectivity constraint (*i.e.,* not passing through all the junctions). Among the valid candidates, we select the one with minimum Fréchet distance from the associated line segment s_i as the resultant arc. We find Fréchet distance a better curve distance measurement than conventional sum-

© 2023 Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.



(a) (b) (c)

Figure 9: **Circular arc extraction.** We extract circular arcs by measuring differences from line segments (a). Using Fréchet distance (b) leads to better results than using sum-of-squares distance (circle fitting error) (c).

of-squares distance (circle fitting error) as the latter cannot penalize the redundant part of the arc (see Fig. 9).

5.4. Additional Details

Line segments simplification. The time complexity of the progressive optimization algorithm highly depends on the size of the initial line segment set $\mathcal{S}_0$. A large one generally leads to more iterations and merge error evaluations. To improve the efficiency, we simplify the initial line segments as follows. If a line segment $s_{m,n}$ (merged from (s_m, s_n)) can be well approximated by an arc (average fitting error smaller than 1.0), then s_m and s_n do not need to be abstracted separately then merged afterwards. This operation significantly improves the efficiency (4 times faster at least), and also produces informative and visually pleasing results (see Sec. 6.2).

Selection of bias σ. We select the bias $\sigma \in [0,1)$ in Eqn. 10 based on the simplified line segments as extracted above. We leverage the longest line segment s to estimate bias σ as:

$$\sigma = \frac{\log r}{\log \phi} - \lfloor \frac{\log r}{\log \phi} \rfloor, \tag{14}$$

where the radius r is derived by general least squares circle fitting to s. The reason to use the longest segment is that it is more robust to aliasing/noisy pixels than short segments for circle fitting. Note that the bias can also be adjusted as a free parameter according to the user preferences in practice.

Localized circle fitting. Although the golden ratio circle fitting in Eqn. 6 can be solved numerically, the computational complexity is still high because the solution space involves a large number of variables and equality constraints. In fact, during the progressive procedure, the line segment representation $\mathcal{L}^{t+1}$ in iteration $t+1$ is only a local update of $\mathcal{L}^t$ by merging a pair of neighboring line segments. This inspires us to perform a localized circle fitting instead of the global optimization for acceleration. More specifically, we only fit a new circle (center+radius) to the merged line segment, while keeping the parameters of all other circles from the previous iteration. However, due to the change of connectivity caused by the merge, we also need to update the centers of the new circle and its neighboring circles to locally preserve the connectivity at the affected junctions. In practice, such localized circle fitting can largely reduce the number of variables and constraints, and significantly improve efficiency (see Sec. 6.4). In rare cases, due to fixing those junctions not affected by the merge (as boundary conditions) to localize the optimization, there may be no feasible solution. In such cases, we solve the circle fitting globally instead.

Iteration	0	23	28
Result			
$\lvert\mathcal{A}\rvert$	51	28	23

Table 1: **Simplicity validation.** Results with increasing simplicity from the progressive optimization. Less circular circles/arcs are generated as the iterations progress.

6. Evaluation

6.1. Visual Results

We evaluate our framework on a wide variety of input images with varying formats and complexity, including real photos, clip arts, and line drawings. The images are collected from the internet. Some of the line drawings are adopted from the benchmark dataset in [YVG20]. Fig. 1, Fig. 4, and Fig. 10 shows various results automatically generated by our framework. The results demonstrate that our work can effectively generate logo-like abstraction from different types of images under golden ratio constraint. By taking into account different design principles, the results well approximate the input object images while preserving the similarity and connectivity of their line segments representation. More results can be found in the supplementary.

6.2. Validation of Key Components

In this subsection, we validate the effectiveness of the key components in our framework.

Simplicity objective. Our progressive approach naturally generates a sequence of results with increasing simplicity. Table 1 shows a sample of results and their statistics during the optimization. As we progressively merge line segments and fit circles, fewer circular arcs are generated and the abstraction level is increased.

Similarity constraint. By adjusting the prescribed tolerance T in Eqn. 1, we can easily adjust the level of similarity compared with the line segments. Table 2 shows the results and their statistics generated from the same input line segments with varying T. It can be seen that lower T leads to higher similarity but lower simplicity, which meets our expectation.

Connectivity constraint. Fig. 11 demonstrates the effectiveness of adding the topology constraint in Eqn. 9 while fitting circles and minimizing the displacement of junctions in Eqn. 7. Compared with the initial line segments (Fig. 11a), the resultant circular arcs

© 2023 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.



Figure 10: **Visual results**. Our framework can generate visually plausible golden ratio logo abstractions automatically from images in different formats, including real photos (row 1 and 2), clip arts (row 3 and 4), and line drawings (row 5). The smaller inset shows the raw input image.

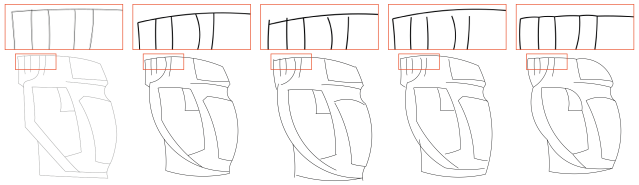
meet at perturbed junctions in the same way with only small displacements (Fig. 11b), which is much better than the result with no connectivity constraints (Fig. 11c) where arcs do not intersect correctly.

Also, a trivial approach that enforces circular arcs to pass

© 2023 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

T	4.0	**5.0**	9.0
Result			
$\|\mathcal{A}\|$	31	27	20

Table 2: **Similarity validation.** Results with less similarity can be controlled by increasing the parameter T of the similarity constraint in Eqn. 1. The original bird can be found in Fig. 10 top row (where $T = 5.0$).



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

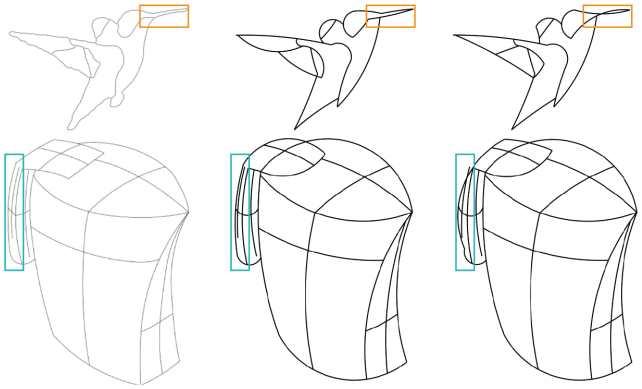
Figure 11: **Connectivity validation.** (a) Initial line segments; (b) Result with connectivity constraint; (c) Without connectivity constraint; (d) With only constraints on start and end junctions while merging line segments; (e) Same as (d) but not no merge of line segments.

through the start and end junctions of the corresponding line segments still lead to missing intersections at the junctions with valence greater than 2 (Fig. 11d).

Further, a naive way of not merging segments can easily ensure that each circular arc passes through the start and end junctions of its associated line segment (Fig. 11e), but this prevents abstraction of multiple continuous line segments and yields 'waved' artifacts (see the zoomed-in inset).

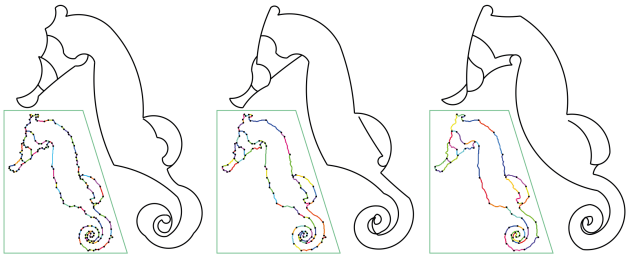
Redundant intersection avoidance. Although the topology constraint can preserve the connectivity at the junction points locally, there is no guarantee that circular arcs do not have other redundant intersections globally (see the red box in Fig. 6c). In practice, we check and penalize redundant intersection at each iteration to avoid merging invalid pairs. Fig. 12 compares results with and without considering redundant intersection. Wrong connectivity can be easily caused by redundant intersections in the zoomed-in region.

Robustness to initial segmentation. We evaluate how sensitive our algorithm is to the different initial line segmentation by altering the tolerances of the Ramer–Douglas–Peucker algorithm in the preprocessing. Fig. 13 shows a visual comparison between different initial line segments and subsequent abstraction results. It can be seen from the insets that the higher the error tolerance value, the less the number of line segments, and thus leading to the elimination of high frequency details. We also find that our algorithm



(a) (b) (c)

Figure 12: **Redundant intersection avoidance.** The input line segments (a) and the result with (b) and without (c) avoiding redundant intersections.



(a) 1.5 (b) 3.0 (c) 6.0

Figure 13: **Robustness to initial line segments.** Results generated from different segmentation tolerances (segments in different colors in the insets) of the Ramer–Douglas–Peucker algorithm.

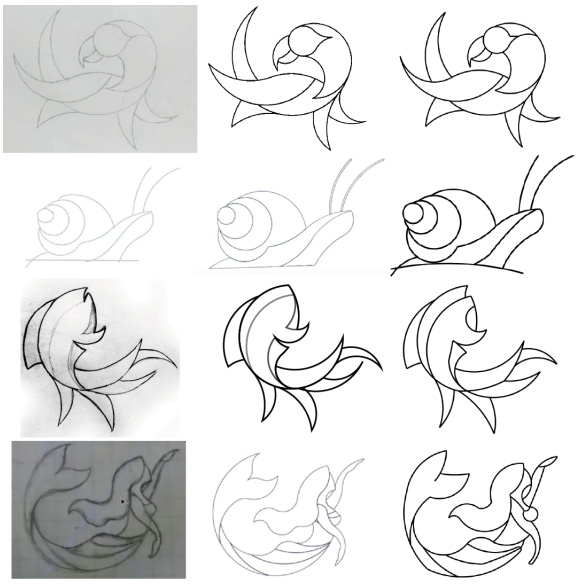
produces similar overall shapes after abstraction, which shows the ability of our framework to deal with different initial segmentation.

6.3. Comparison with Artworks

As there is no previous work on gold ratio logo abstraction from images, we compare our results with designs manually created by designers from their online tutorials (see Fig. 14). We can see that our automated results conform with manual designs well, and can be generated efficiently ($\sim$ 1 minute) without any user intervention. In contrast, manual designs require more much time due to repeatedly aligning and cutting circles using graphic design software by hand (see the user study in Sec. 8 and also the supplementary video).

Note that the designers sometimes prefer more abstracted results with concise/aesthetic considerations while omitting some detailed features (e.g., the fish mouth in Fig. 14), while our results are constrained on similarity and connectivity. But the users can easily refine the automated results by design exploration (see Fig. 17), which is much more convenient than starting from scratch.

© 2023 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.



(a) Sketch (b) Manual design (c) Ours

Figure 14: **Comparison with artistic designs.** Given an input sketch (a), our framework can generate golden ratio logo abstraction (c) comparable to manual design by the designer (b). (From top to bottom: ©Mahk Graphics [Gra20], ©Design Quickies [Qui20], ©Mohamed Achraf [Ach17], ©Aiten Arts [Art19].)

Localized circle fitting	Line segments simplification	Time (sec)
No	No	$\geq$ 2hrs
Yes	No	257.47
No	Yes	338.87
Yes	Yes	**65.37**

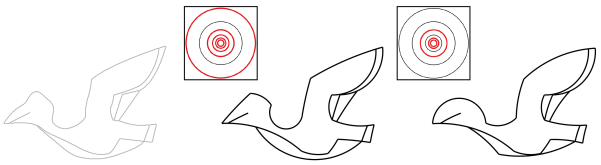
Table 3: **Timing performance.** Comparison of computational performances using the fish example in Fig. 1. Localized circle fitting and line segments simplification makes significant improvements.

6.4. Efficiency Analysis

To evaluate the efficiency of our framework, we carefully test its performance under different settings using the fish example shown in Fig. 1. Our computational platform is a desktop PC with Intel i9 CPU of 3.60GHz. Our framework is implemented using Python. The timings can be found in Table 3, which shows that the line segments simplification and localized circle fitting strategies (Sec. 5.4) largely improve the computational performance of our progressive optimization. The former effectively reduces the number of line segments (from 77 to 36) to initialize the optimization, while the latter constrains the optimization to a local region (typically has 4-8 segments) affected by merging line segments.

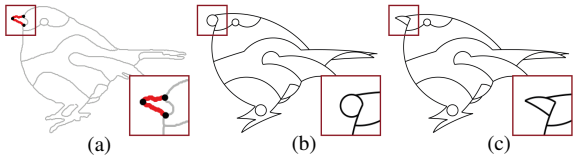
7. Design Exploration

Although our framework generates results automatically (see Fig. 10), it also allows the user to efficiently explore the design space by tuning key parameters or directly editing the automated results to generate personalized golden ratio logos (see supplemental video).



(a) Line segments (b) Automated setting (c) User-specified setting

Figure 15: **Design exploration using different golden ratio settings.** Given an input line segments (a), our framework can generate different logo abstraction results respecting the different settings (automated (b) vs. user-specific (c)) of golden ratio circles (highlighted in red in the inset). Note that circles are scaled down here for compactness.



(a) (b) (c)

Figure 16: **Design exploration by local abstraction level specification.** The user can specify salient feature (bird beak) in a local region (a) to avoid feature loss due to merging of line segments (b), such that salient feature can be preserved in the final result (c).

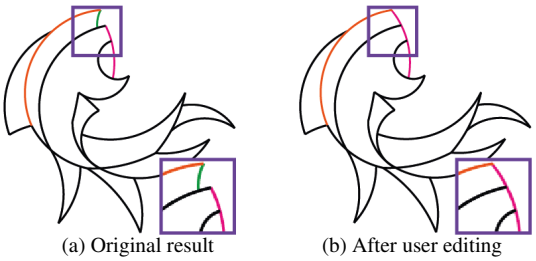
User-guided exploration. In Sec. 6.2, we already show how the key components and their parameters affect the automated results. The framework also enables other design exploration possibilities guided by the user. For instance, the user can constrain the golden ratio circles involved in the arc fitting in Sec. 5.3 by specifying a small subset of radii. Fig. 15 shows some exploration results using different radii settings. Our framework also allows the user to adjust the abstraction level by enabling and disabling the merge of line segments during the optimization. This helps to preserve salient features that cannot be well captured by the similarity constraint. Fig. 16 shows an example of manually constraining the abstraction level to better preserve the beak of a bird.

User edits. Based on the automated results, the user can easily perform post-editing according to her/his preferences to generate personalized logos. The user can directly delete/connect/perturb arcs based on their underlying circles to create variants from the automated result, while still satisfying the golden ratio constraint. An example is shown in Fig. 17. The user can also edit colors of the regions bounded by circular arcs to generate colorized logos. Different color effects can be achieved based on simple color palette, gradients, *etc*, as shown in Fig. 18.

8. User Study

In this section, we evaluate the effectiveness of our framework in helping graphic designers in practice by conducting a user study. The goal is to compare our framework based design with conventional manual design in terms of efficiency and user experience. To this end, we recruited four designers who have experience in

© 2023 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.



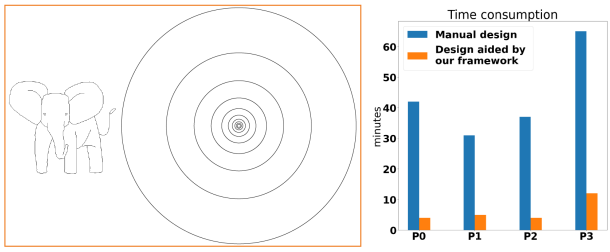
(a) Original result (b) After user editing

Figure 17: **User editing of circular arcs.** The user can easily edit (e.g., delete, add, connect) the arcs/circles in the automated result (a) to generate a variant design (b).



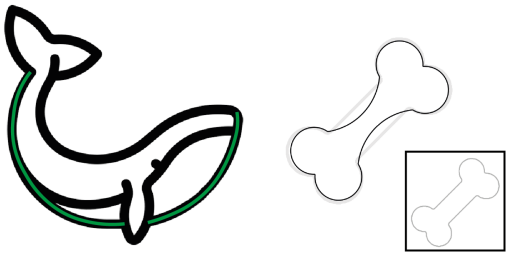
(a) (b)

Figure 18: **User editing of colors.** The user can easily generate personalized logos by editing the colors in the regions bounded by circular arcs. Here we show examples generated from real photo (a), and clip art (b).



(a) The target shape and golden ratio circles (b) Efficiency plot

Figure 19: **User study.** (a) The target shape and the set of golden ratio circles used in both manual and semi-automatic design stages. (b) The time cost of two stages across four participants P0-P3.



(a) (b)

Figure 20: **Special line segment cases yet to consider.** (a) Disconnected line segments cannot be abstracted using one circular arc. (b) Straight line segments are not suitable to fit circular arcs.

graphic design for at least three years. To make a fair comparison, the participants were asked to create a gold ratio logo design from the same target shape using a set of predefined golden ratio circles in both manual and framework-aided design process (see Fig. 19a). The user study included two stages. Firstly, the participants used their favorite graphic design tools (all participants chose Adobe Illustrator) to finish the task. Secondly, the participants created the design with the aid of our framework. The function provided by our framework includes 1) select an abstraction level from the progressive optimization, and 2) enable/disable the merge of line segments (as described in Sec. 7) during the progressive optimization. Note that the order of two stages accords with the typical practice of transferring from conventional design to computer-aided design. The work time was restricted to 90 minutes in both stages and we gave a tutorial of our framework beforehand. After the study, we also collected subjective feedback from the participants.

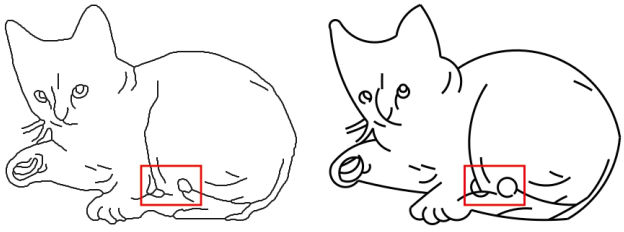
Fig. 19b shows the time consumption (in minutes) of two stages across individual participants. The mean and standard deviation are 43.75 and 12.87 for manual design, and 6.25 and 3.34 for the design aided by our framework. It is not surprising that the efficiency can be significantly improved when using our framework. In general, all participants agreed that our framework can greatly improve the efficiency as there was no need to manually select/align circles and trace arcs to fit the target shape. Besides, the results from our framework were generally smoother and more consistent, which was hard to achieve in the manual designs due to the error accumulation during the trial-and-error process (all the results can be found in the supplementary). Moreover, one participant expressed that our progressive results can also provide different abstraction inspirations. On the other hand, one common suggestion is to integrate more flexible editing tools (e.g., move circles around) as those in Adobe Illustrator. This can be easily done by directly adding features to the current framework, or exporting our results as vector graphics to the general design tools for post-processing.

9. Discussion

In this work, we present the first computational framework that can automatically convert an input image into a logo abstraction formed by circular arcs with golden ratios. We first identify a set of principles that ensure the characteristics and quality of the golden ratio design. Then we formulate the problem as a constrained optimization according to the identified principles. We also propose a progressive approach that can efficiently solve the optimization problem by iteratively simplifying the feature line structure of the input image and fitting golden ratio circular arcs. We extensively evaluate our framework and validate its key components using various images of different formats. Via the user study, designers appreciate the functionality and efficiency of our framework, and look forward to having more editing freedom in the future. Besides, our framework can be easily extended to fit arbitrary circles and splines by relaxing/replacing the golden ratio constraint, thus having the potential to also benefit image vectorization. Furthermore, our progressive approach can inspire other image/shape abstraction works to explore results at different levels of detail.

Limitations. Although the proposed iterative optimization is advantageous to efficiently generate progressive abstraction results, one limitation is that disconnected line segments cannot be ab-

© 2023 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.



(a) Line segments (b) Abstraction result

Figure 21: **Complex line segment structure.** The complex structure of line segments may cause local feature losses when overly merging line segments while fitting golden ratio circular arcs.

stracted using one circular arc as done by designers in some special occasions (see Fig. 20a). This is because we extract/merge semantic line segments directly in the line segment representation $\mathcal{L}$ based on line continuity (only merge neighboring segments) and similarity (between circular arc and merged segments). The semantic parts of the input object, which are helpful to extract more meaningful and even disconnected line segments, are not considered so far. Semantic segmentation of images in different formats (e.g., photo, clip arts, sketches) is challenging by itself and beyond the scope of this work. Also, the format/quality of the input image can affect the logo abstraction results. Unlike clip arts whose contents are simpler, real photos and line drawings have more complex structure due to the large variation of colors, strokes, shadings, *etc*, such inputs may cause feature losses with our method. Therefore, to ensure the abstraction quality, we filter out short/noisy line segments in the preprocessing. But for complex line segment structures, overly merging neighboring line segments while fitting golden ratio circular arcs can still lose features (see Fig. 21).

Future work. In the future, we plan to make our framework more general to include other constituent line elements such as straight line segments in addition to only golden ratio circular arcs. This can help to avoid shape distortions when abstracting long and straight line segments using circular arcs, in particular with limited radii (see Fig. 20b). Also, we would like to further improve the regularity of the results by considering more global shape properties such as shape symmetry, shape semantics, *etc*. Besides, for non-clean input from real photos, relaxing (unreliable) connectivity constraint or allowing manual topology change would be useful to expand the exploration space. Last but not least, we will investigate how to automatically fill colors to the abstracted golden ratio logo according to the full color information of the input image. Some existing image vectorization techniques can be exploited.

Acknowledgments

We are grateful to the anonymous reviewers for their comments and suggestions. This work is supported in part by the National Science and Technology Council (110-2221-E-007-060-MY3 and 110-2221-E-007-061-MY3), the RCUK grant CAMERA (EP/M023281/1, EP/T022523/1), and a gift from Adobe.

References

[Ach17] ACHRAF, MOHAMED. *How to design a logo with golden Ratio #2 | Adobe Illustrator Tutorial*. https://www.youtube.com/watch?v=NXKYUpWfCjA&t=470s. 2017 10.

[Art19] ARTS, AITEN. *How to Design a Logo with Golden Ratio | Mermaid*. https://www.youtube.com/watch?v=TLxZ8XWPBSo&t=1238s. 2019 10.

[BLP10] BARAN, ILYA, LEHTINEN, JAAKKO, and POPOVIĆ, JOVAN. "Sketching Clothoid Splines Using Shortest Paths". *Computer Graphics Forum* 29.2 (2010), 655–664 3.

[Bok18] BOKHUA, GEORGE. *Mastering Logo Design: Gridding with the Golden Ratio*. https://www.skillshare.com/classes/Mastering-Logo-Design-Gridding-with-the-Golden-Ratio/774191846. 2018 2, 3.

[BS19] BESSMELTSEV, MIKHAIL and SOLOMON, JUSTIN. "Vectorization of line drawings via polyvector fields". *ACM Transactions on Graphics* 38.1 (2019), 1–12 3.

[DP73] DOUGLAS, DAVID H and PEUCKER, THOMAS K. "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature". *Cartographica: the international journal for geographic information and geovisualization* 10.2 (1973), 112–122 4.

[DS02] DECARLO, DOUG and SANTELLA, ANTHONY. "Stylization and Abstraction of Photographs". *ACM Transactions on Graphics* 21.3 (2002), 769–776 3.

[DYH*21] DAS, AYAN, YANG, YONGXIN, HOSPEDALES, TIMOTHY M, et al. "Cloud2curve: Generation and vectorization of parametric sketches". *IEEE CVPR*. 2021, 7088–7097 3.

[EM94] EITER, THOMAS and MANNILA, HEIKKI. *Computing Discrete Fréchet Distance*. Tech. rep. Christian Doppler Laboratory, 1994 4.

[EVA*20] EGIAZARIAN, VAGE, VOYNOV, OLEG, ARTEMOV, ALEXEY, et al. "Deep vectorization of technical drawings". *ECCV*. 2020, 582–598 3.

[Eva21] EVAMY, M. *LOGO*. 2021. ISBN: 9784802512138. URL: https://books.google.com.tw/books?id=KWN-zgEACAAJ 2.

[FLB16] FAVREAU, JEAN-DOMINIQUE, LAFARGE, FLORENT, and BOUSSEAU, ADRIEN. "Fidelity vs. simplicity: a global approach to line drawing vectorization". *ACM Transactions on Graphics* 35.4 (2016), 1–10 3.

[FLB17] FAVREAU, JEAN-DOMINIQUE, LAFARGE, FLORENT, and BOUSSEAU, ADRIEN. "Photo2clipart: Image Abstraction and Vectorization Using Layered Linear Gradients". *ACM Transactions on Graphics* 36.6 (2017) 2, 3.

[GDA*12] GERSTNER, TIMOTHY, DECARLO, DOUG, ALEXA, MARC, et al. "Pixelated Image Abstraction". *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering*. NPAR '12. 2012, 29–36 3.

[GH97] GARLAND, MICHAEL and HECKBERT, PAUL S. "Surface Simplification Using Quadric Error Metrics". *ACM SIGGRAPH*. 1997, 209–216 3.

[Gra20] GRAPHICS, MAHK. *Golden Circle || Golden Ratio || Logo Design Concept*. https://www.youtube.com/watch?v=KaEJK7z2H_s. 2020 10.

[GRG04] GOOCH, BRUCE, REINHARD, ERIK, and GOOCH, AMY. "Human Facial Illustrations: Creation and Psychophysical Evaluation". *ACM Transactions on Graphics* 23.1 (2004), 27–44 3.

[Gri17] GRIBOV, A. "Approximate Fitting a Circular Arc When Two Points Are Known". *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. Vol. 2. 2017, 15–16 6.

[GZH*19] GUO, YI, ZHANG, ZHUMING, HAN, CHU, et al. "Deep line drawing vectorization via line subdivision and topology reconstruction". *Computer Graphics Forum*. Vol. 38. 7. 2019, 81–90 3.

© 2023 Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

[Hec82] HECKBERT, PAUL. "Color Image Quantization for Frame Buffer Display". *ACM SIGGRAPH*. 1982, 297–307 3.

[HWH*18] HAN, CHU, WEN, QIANG, HE, SHENGFENG, et al. "Deep Unsupervised Pixelization". *ACM Transactions on Graphics* 37.6 (Nov. 2018), 243:1–243:11 3.

[HZZ11] HUANG, HUA, ZHANG, LEI, and ZHANG, HONG-CHAO. "Arcimboldo-like Collage Using Internet Images". *ACM SIGGRAPH Asia*. 2011 3.

[IMF22] IBRAHIM, KHALED HOSSAMELDIN SADEK, MOHAMED, ALI KHATER, and FAROUK, AHMED. "Genetic Algorithm Golden Ratio Design Model for Auto Arts". *Handbook of Nature-Inspired Optimization Algorithms: The State of the Art: Volume I: Solving Single Objective Bound-Constrained Real-Parameter Numerical Optimization Problems*. Ed. by MOHAMED, ALI, OLIVA, DIEGO, and SUGANTHAN, PONNUTHURAI NAGARATNAM. Cham: Springer International Publishing, 2022, 229–242. ISBN: 978-3-031-07512-4. DOI: 10.1007/978-3-031-07512-4_8. URL: https://doi.org/10.1007/978-3-031-07512-4_8 3.

[JYF*20] JING, YONGCHENG, YANG, YEZHOU, FENG, ZUNLEI, et al. "Neural Style Transfer: A Review". *IEEE Transactions on Visualization and Computer Graphics* 26.11 (2020), 3365–3385 3.

[KBYU20] KARAMATSU, TAKURO, BENITEZ-GARCIA, GIBRAN, YANAI, KEIJI, and UCHIDA, SEIICHI. "Iconify: Converting Photographs into Icons". *Proceedings of ACM MMArt*. 2020, 7–12 2, 3.

[KD09] KYPRIANIDIS, JAN ERIC and DÖLLNER, JÜRGEN. "Real-Time Image Abstraction by Directed Filtering". *ShaderX7 - Advanced Rendering Techniques*. Ed. by ENGEL, W. 2009, 285–302 3, 4.

[KSP13] KOPF, JOHANNES, SHAMIR, ARIEL, and PEERS, PIETER. "Content-Adaptive Image Downscaling". *ACM Transactions on Graphics* 32.6 (2013), to appear 3.

[LALR16] LIU, YIMING, AGARWALA, ASEEM, LU, JINGWAN, and RUSINKIEWICZ, SZYMON. "Data-Driven Iconification". *International Symposium on Non-Photorealistic Animation and Rendering (NPAR)*. 2016, 113–124 2, 3.

[LBK17] LIU, MING-YU, BREUEL, THOMAS, and KAUTZ, JAN. "Unsupervised Image-to-Image Translation Networks". *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. 2017, 700–708 2, 3.

[LRS18] LIU, CHENXI, ROSALES, ENRIQUE, and SHEFFER, ALLA. "StrokeAggregator: Consolidating Raw Sketches into Artist-Intended Curve Drawings". *ACM Transactions on Graphics* 37.4 (2018) 3.

[LWH15] LIU, XUETING, WONG, TIEN-TSIN, and HENG, PHENG-ANN. "Closure-aware Sketch Simplification". *ACM Transactions on Graphics* 34.6 (Nov. 2015), 168:1–168:10 4.

[LYC18] LIN, YOU-EN, YANG, YONG-LIANG, and CHU, HUNG-KUO. "Scale-Aware Black-and-White Abstraction of 3D Shapes". *ACM Transactions on Graphics* 37.4 (2018) 3.

[LZZ*21] LIU, ZHONG-YUAN, ZHANG, ZHAN, ZHANG, DI, et al. "Modeling and Fabrication with Specified Discrete Equivalence Classes". *ACM Transactions on Graphics* 40.4 (2021) 3.

[LZZ06] LIU, KE, ZHOU, FU-QIANG, and ZHANG, GUANG-JUN. "Radius constraint least-square circle fitting method and error analysis". *Journal of Optoelectronics Laser* 17.5 (2006), 604 6.

[Mar22] MARSH, LINDSAY. *Logo Design Mastery In Adobe Illustrator*. https://www.udemy.com/course/logo-design-mastery-in-adobe-illustrator/. 2022 2.

[MARI17] MARTN, DOMINGO, ARROYO, GERMN, RODRGUEZ, ALEJANDRO, and ISENBERG, TOBIAS. "A Survey of Digital Stippling". *Comput. Graph.* 67.C (2017), 24–44 3.

[MDS09] MI, XIAOFENG, DECARLO, DOUG, and STONE, MATTHEW. "Abstraction of 2D Shapes in Terms of Parts". *Proceedings of the 7th International Symposium on Non-Photorealistic Animation and Rendering*. NPAR '09. 2009, 15–24 3.

[Mei18] MEISNER, GARY B. *The Golden Ratio: The Divine Beauty of Mathematics*. Race Point Publishing, 2018 2.

[MS11] MCCRAE, JAMES and SINGH, KARAN. "Neatening Sketched Strokes Using Piecewise French Curves". *Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling*. SBIM '11. 2011, 141–148 3.

[NHS*13] NORIS, GIOACCHINO, HORNUNG, ALEXANDER, SUMNER, ROBERT W, et al. "Topology-driven vectorization of clean line drawings". *ACM Transactions on Graphics* 32.1 (2013), 1–11 3.

[OB91] ORCHARD, M.T. and BOUMAN, C.A. "Color quantization of images". *IEEE Transactions on Signal Processing* 39.12 (1991), 2677–2690 3.

[OBW*08] ORZAN, ALEXANDRINA, BOUSSEAU, ADRIEN, WINNEMÖLLER, HOLGER, et al. "Diffusion Curves: A Vector Representation for Smooth-Shaded Images". *ACM Trans. Graph.* 27.3 (Aug. 2008), 1–8. ISSN: 0730-0301. DOI: 10.1145/1360612.1360691. URL: https://doi.org/10.1145/1360612.1360691 3.

[PNCB21] PUHACHOV, IVAN, NEVEU, WILLIAM, CHIEN, EDWARD, and BESSMELTSEV, MIKHAIL. "Keypoint-Driven Line Drawing Vectorization via PolyVector Flow". *ACM Transactions on Graphics* 40.6 (2021) 3.

[Qui20] QUICKIES, DESIGN. *Golden Ratio Snail Logo From Sketch | Follow Along Tutorial | Adobe Illustrator*. https://www.youtube.com/watch?v=Bo1UBNXtV0M. 2020 10.

[RL10] ROSIN, PAUL L. and LAI, YU-KUN. "Towards Artistic Minimal Rendering". *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering*. NPAR '10. 2010, 119–127 3.

[Sal19] SALEEM, MOEGAMAT. *Golden Ratio Logo Design Master Class*. https://www.udemy.com/course/golden-ratio-logo-design-master-class/. 2019 2, 3.

[SATV18] SAGE, ALEXANDER, AGUSTSSON, EIRIKUR, TIMOFTE, RADU, and VAN GOOL, LUC. "Logo synthesis and manipulation with clustered generative adversarial networks". *IEEE CVPR*. 2018, 5879–5888 2, 3.

[SBBB20] STANKO, TIBOR, BESSMELTSEV, MIKHAIL, BOMMES, DAVID, and BOUSSEAU, ADRIEN. "Integer-Grid Sketch Simplification and Vectorization". *Computer Graphics Forum* 39.5 (2020), 149–161 3.

[SII18a] SIMO-SERRA, EDGAR, IIZUKA, SATOSHI, and ISHIKAWA, HIROSHI. "Mastering Sketching: Adversarial Augmentation for Structured Prediction". *ACM Transactions on Graphics* 37.1 (2018) 3.

[SII18b] SIMO-SERRA, EDGAR, IIZUKA, SATOSHI, and ISHIKAWA, HIROSHI. "Real-Time Data-Driven Interactive Rough Sketch Inking". *ACM Transactions on Graphics* 37.4 (2018) 3.

[WOG06] WINNEMÖLLER, HOLGER, OLSEN, SVEN C., and GOOCH, BRUCE. "Real-Time Video Abstraction". *ACM Transactions on Graphics* 25.3 (2006), 1221–1226 3.

[YVG20] YAN, CHUAN, VANDERHAEGHE, DAVID, and GINGOLD, YOTAM. "A Benchmark for Rough Sketch Cleanup". *ACM Transactions on Graphics* 39.6 (2020) 7.

[ZCR*16] ZOU, CHANGQING, CAO, JUNJIE, RANAWEERA, WARUNIKA, et al. "Legible Compact Calligrams". *ACM Transactions on Graphics* 35.4 (2016) 3.

[ZKH*20] ZHAO, NANXUAN, KIM, NAM WOOK, HERMAN, LAURA MARIAH, et al. "ICONATE: Automatic Compound Icon Generation and Ideation". CHI '20. 2020, 1–13 3.

[ZPIE17] ZHU, JUN-YAN, PARK, TAESUNG, ISOLA, PHILLIP, and EFROS, ALEXEI A. "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks". *IEEE ICCV*. 2017, 2242–2251 3.

© 2023 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.