Pyramid of Arclength Descriptor for Generating Collage of Shapes

Kin Chung Kwan[1,2]   Lok Tsun Sinn[1,2]   Chu Han[1,2]   Tien-Tsin Wong[1,2]   Chi-Wing Fu[1]

[1]The Chinese University of Hong Kong*

[2]Shenzhen Research Institute, The Chinese University of Hong Kong

Figure 1: *"Icons in PAD": a collage of shapes generated with our scale- and rotation-invariant shape descriptor, namely pyramid of arclength descriptor (PAD); this novel descriptor improves the shape-matching efficiency, thus facilitating the generation of complex results.*

Abstract

This paper tackles a challenging 2D collage generation problem, focusing on shapes: we aim to fill a given region by packing irregular and reasonably-sized shapes with minimized gaps and overlaps. To achieve this nontrivial problem, we first have to analyze the boundary of individual shapes and then couple the shapes with partially-matched boundary to reduce gaps and overlaps in the collages. Second, the search space in identifying a good coupling of shapes is highly enormous, since arranging a shape in a collage involves a position, an orientation, and a scale factor. Yet, this matching step needs to be performed for every single shape when we pack it into a collage. Existing shape descriptors are simply infeasible for computation in a reasonable amount of time. To overcome this, we present a brand new, scale- and rotation-invariant 2D shape descriptor, namely *pyramid of arclength descriptor* (PAD). Its formulation is locally supported, scalable, and yet simple to construct and compute. These properties make PAD efficient for performing the partial-shape matching. Hence, we can prune away most search space with simple calculation, and efficiently identify candidate shapes. We evaluate our method using a large variety of shapes with different types and contours. Convincing collage results in terms of visual quality and time performance are obtained.

Keywords: Image collage, 2D shapes, Tiling, Shape descriptor, Scale-invariant, Partial-shape matching

Concepts: •**Computing methodologies** → **Non-photorealistic rendering; Shape analysis;**

1 Introduction

Collage is a form of visual art created by pasting elements like photos, clippings, and ribbons onto a canvas region. It exists in several forms, e.g., mosaics, which are made up of colored glass and stone, and photomontages, which cut and compose photographs. In this paper, we focus on generating collages composed of shapes, without much consideration of the color. Our goal is to *pack 2D shapes to fill a canvas with minimized overlaps and gaps in-between the shapes* (e.g., Fig. 2). This form of collage is extensively used in commercial advertisement, 2D design, and many other illustrations.

During the design, artists pack each shape one-by-one in a trial-and-error manner, by trying various scale, position, and orientation

Figure 2: *Examples: collages designed by artists, courtesy of Postmaster General of Hongkong Post (left) and lalan (right).*

of the shape. Obviously, the search space is enormous. Currently, such collages are *manually created*; we are not aware of any automatic algorithm that can achieve the task within a tractable period of time. To ease the difficulty in matching/coupling shapes and enhance the visual pleasantness, artists usually include padding in-between shapes and insert tiny shapes to avoid oversized empty space, e.g., the small stars and sector shapes in right sub-figure of Fig. 2. In this paper, we present an automatic method to efficiently address this challenging shape packing problem. Note that our packing problem is not the same as the standard packing problem in geometry, which ignores object changes and scaling.
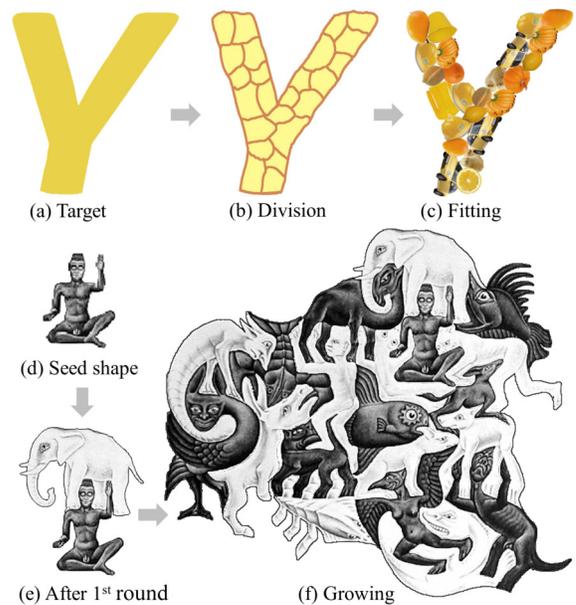
Existing collage-related methods such as [Hausner 2001; Kim and Pellacini 2002; Huang et al. 2011] take a top-down approach in generating collages. They divide the canvas into cells and fit shapes into each cell region, see Fig. 3(a)-(c). Hence, the shapes are implicitly assumed to be more-or-less convex with presumed scale to fit the cells. As a result, their ability in handling more general shapes and achieving good coupling of shapes is questionable.

On the other hand, we may take a bottom-up approach to progressively pack shapes similar to the way artists do. However, this requires a highly efficient partial-shape matching process, since every time we pack a shape into a collage, we have to continuously translate, scale and rotate the shape to evaluate the shape coupling, see Fig. 12(c)&(d). This involves an exceedingly large search space: a large number of candidates (shapes), and for each shape, there can be unlimited number of instances (each with different scale, location, and/or orientation). While existing shape descriptors such as curvature scale space [Mokhtarian and Mackworth 1992; Mokhtarian et al. 1996], shape context [Belongie et al. 2002], and triangle-area representation [Alajlan et al. 2007] work well for measuring the overall similarity between (whole) shapes, they cannot be used directly for partial-shape matching. Even we extend them by dividing a shape into multiple instances of curve segments for matching, there are exceedingly large choices of segments on each shape, thus leading to a huge search space. Hence, the approach is simply infeasible for efficient computation (see Section 2 for details).
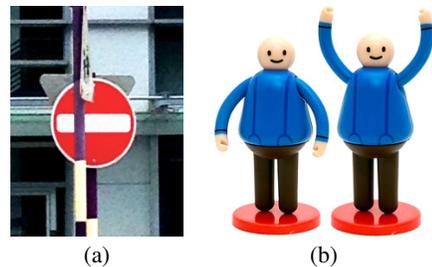
In this work, we take a bottom-up approach to generate collages *without assumption nor restriction on shapes* and *within a tractable amount of time*. Given a collection of shapes, our method starts with a seed shape and iteratively fills the given region piece-by-piece, see Fig. 3(d)-(f). In each iteration, we find the best docking location, orientation, and scale of a shape by maximizing the arclength (which is a partially-matched boundary) shared between the shape and the current collage. Given N and M as the number of sample points along the boundary of a shape and the current collage, respectively. If we employ an existing shape descriptor to match instances of curve segments, we need $N^2 M^2$ matching tests. Our key to tackle the enormous search space within tractable time is a novel shape descriptor, namely *pyramid of arclength descriptor* (in short, PAD). Since PAD describes the local shape around a point (*locally supported*) in a *scale- and rotation-invariant* fashion, it can effectively trim down the search space for partial-shape matching, allowing us to perform only NM tests[1] to identify the best docking position of a shape onto the collage (see Section 2 for details). Moreover, it has a simple vector form, which is just a sequence of normalized arclength values in a pyramidal scale (Section 3), so it is easy to construct and fast to compute.

Unlike existing top-down methods, which produce collages with more-or-less convex shapes, our method can achieve better shape coupling, even for arbitrarily irregular shapes, see Fig. 1. By augmenting the objective function in the shape matching process, we

---

[1]If we use a more sophisticated algorithm such as kNN to accelerate, we can reduce the number of tests to $N^2 \log M$ and $N \log M$, respectively (see Section 2 for details).

Figure 3: *Top row (a)-(c): a typical top-down approach. Bottom row (d)-(f): a bottom-up approach (our result); All M.C. Escher works ©2016 The M.C. Escher Company - the Netherlands. All rights reserved. Used by permission. www.mcescher.com.*

Figure 4: *(a) Partial occlusion due to view blocking (e.g., a partially-blocked sign) and (b) Partial change of shape.*

can generate collages with various design goals, e.g., avoiding over-shrinking and oversizing, upside-down pieces, etc. To demonstrate the applicability of our method, we generate a large number of collage results with different types of shapes and contours, and conduct a user study to compare the visual quality of our results against existing methods. Moreover, we conduct experiments to evaluate the timing statistics of our method, demonstrating its high efficiency in generating collages of different shapes.

PAD can also be applied to many other pattern recognition applications. Since PAD makes no assumption on the closeness of the shapes, it can be used to match both closed shapes and open curves. Its partial-shape matching capability is highly useful in many complicated real-world scenarios, when there exists *partial occlusion*, see Fig. 4(a), and *partial change of shape*, see Fig. 4(b). The contributions in this work are summarized as follow:

- We propose a simple, scalable, locally supported, scale-, and rotation-invariant pyramid of arclength descriptor (PAD) that facilitates efficient partial-shape matching. It can be applied to pattern recognition applications that involve partial-shape matching and tackle partial occlusion problem.

- We can efficiently generate collages of shapes with "shape coupling" among arbitrarily irregular shapes of controllable scales, by drastically reducing the search space.

2 Related Work

Collages There are several related computational methods in generating 2D collages. Photo collage [Rother et al. 2006; Goferman et al. 2010] considers a slightly different problem with the goal of packing a region with intersecting photos, where the photo boundary can be soft. Puzzle solving methods [Yao and Shao 2003; Goldberg et al. 2004], on the other hand, do not need to consider the scale of pieces since the sizes of puzzle pieces are fixed.
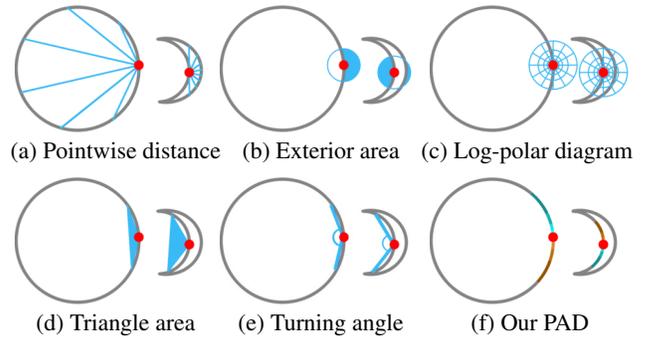
Kaplan and Salesin [2000; 2004] presents a special type of tiling that resembles certain artworks of M.C. Escher. Their method modifies a given 2D shape via constrained optimization, so that the modified shape can tile a plane. Dalal et al. [2006] used an FFT-based correlation to generate image collage, but their work only supports more-or-less convex shapes. Existing methods for tiling [Hausner 2001; Kim and Pellacini 2002; Xu and Kaplan 2007; Orchard and Kaplan 2008; Hu et al. 2016; Zou et al. 2016] mostly take a top-down approach, which subdivides the given canvas region into cells of similar sizes and then fills each cell with a 2D object by maximizing the overall shape similarity. Such 2D object could be a clip art or a photo segment. However, since the subdivision process does not consider the shape of the given 2D objects, the tessellated cells are more-or-less circular in shape and the resulting tiling is less interesting with mostly simple and short-length object contacts. To generate more irregular cells, Huang et al. [2011] introduced another top-down approach that subdivides the canvas region according to its color. However, their method ignores the shape of the given objects, so a good match may not be found for some of the cells. Reinert et al. [2013] presented a framework to perturb 2D objects in the canvas by equalizing the gaps in-between objects. Since their method is not rotation-invariant nor scale-invariant, they cannot pack objects in the collage results with good coupling. Besides 2D puzzles, Gal et al. [2007] and Huang et al. [2014] explored bottom-up approaches to generate 3D collage, but their problem setting does not require scale invariance.

Compared to previous works, we take a bottom-up approach to generate 2D collages, focusing on packing objects according to shapes. Thanks to the proposed PAD, which enables us to efficiently match shapes along a partial-shape boundary with scale- and rotation-invariances. We thus can efficiently match and pack shapes into a collage and produce more intriguing results. Without the PAD, this bottom-up approach would be computationally intractable. This is also why most previous works do not take a bottom-up approach since scale invariance needs to be considered during the partial-shape matching process.
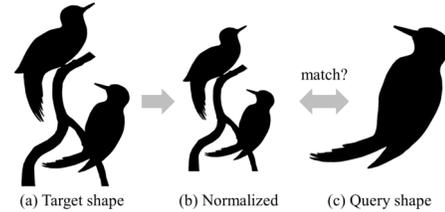
Shape Descriptor Two-dimensional shape descriptors can be roughly classified into two categories: *global shape descriptors*, which describe the overall shape characteristic, and *local shape descriptors*, which describe local shape regions by local features.

Global shape descriptors Typical methods include Hu moments [Hu 1962], Fourier descriptors [Granlund 1972; Persoon and Fu 1977], Zernike moments [Khotanzad and Hong 1990], image moments [Belkasim et al. 1991; Sheng and Shen 1994], wavelet descriptors [Chuang and Kuo 1996], generic Fourier descriptor [Zhang and Lu 2002], and Radon-transform descriptor [Tabbone et al. 2006]. They compactly describe the characteristics over the whole shape, so we can efficiently measure overall shape similarity. However, as they pay more attention to the whole shape, local shape characteristics are usually lost, so they are ineffective for partial-shape matching, which focuses on local segments.

To extend global descriptors to support partial-shape matching with scale invariance, we may divide a shape into multiple instances of curve segments, describe each instance using a global descriptor, and then perform partial-shape matching by measuring the shape

(a) Pointwise distance (b) Exterior area (c) Log-polar diagram

(d) Triangle area (e) Turning angle (f) Our PAD

Figure 5: *By appropriately scaling either the new moon or the circle shape, we should be able to perfectly match local regions of the shapes at corresponding red dots. However, none of these local features (a)-(e), except PAD in (f), can produce the same descriptor (visualized as blue elements) for the local regions around red dots.*
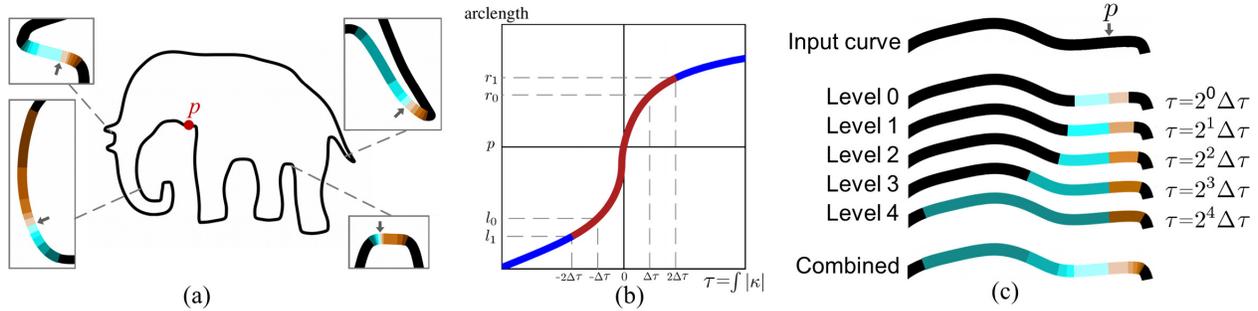

(a) Target shape (b) Normalized (c) Query shape

Figure 6: *Partial-shape matching aims to find (c) in a local region of (a); existing local descriptors approach this by normalizing the shapes to certain scale (b&c) for matching, but this may not work.*

similarity between instances of different shapes. To apply this extension to collage generation, we consider N and M as the number of sample points along the boundary of a shape and the current collage, respectively. Since each curve segment instance has two endpoints, we thus have N^2 and M^2 choices of segment instances on the given shape and on the current collage, respectively. Therefore, to find the best docking location to put the shape onto the collage with scale invariance, a brute-force approach would need $N^2 M^2$ matching tests, which is inefficient and intractable.

Local shape descriptors Typical methods include turn function of polygons [Arkin et al. 1991], curvature scale space [Mokhtarian and Mackworth 1992; Mokhtarian et al. 1996], shape context [Belongie et al. 2002], shape signatures [Lee et al. 2006], triangle-area representation (TAR) [Alajlan et al. 2007], the SKS algorithm [Krish and Snyder 2008], IS-Match [Donoser et al. 2010], and integral invariants [Hong and Soatto 2015], which describe a local region in a shape by using various local features (Fig. 5). Please refer to [Veltkamp and Hagedoorn 2001; Yang et al. 2008; Van Kaick et al. 2011] for comprehensive surveys on these descriptors.

Existing local shape descriptors do not explicitly consider scale. Fig. 5 illustrates this issue with an example: if we can appropriately scale the new moon shape (or equivalently the circle shape), the inner side of the moon at the red dot should match perfectly the outer side of the circle at the other red dot. In other words, if a local descriptor is scale invariant rather than scale dependent, it should produce the same descriptor for the local regions around the two red dots: one on circle and one on moon. However, none of them (except PAD) possess scale invariance (Fig. 5). Note particularly the coverage difference for each local feature on the two shapes.

To extend local descriptors to support partial-shape matching with scale invariance, one approach is to normalize the two shapes to certain scale for matching, e.g., shape context [Belongie et al. 2002] uses the mean of point-wise distances. However, such normalization may not always be able to find an appropriate scale, so we may

Figure 7: *Our PAD. (a) Consider point p (red dot) on a curve. (b) We plot the "scale-invariant" integral of absolute curvature (τ) against the arclength measured from each side of p along the curve. (c) Each arclength (left: cyan & right: brown) corresponds to certain fixed amount of integrals (levels) of $\Delta\tau$; they together form a local PAD at p. See blown-up views in (a) for more examples of PAD at different positions.*

not be able to correctly match partial shapes using local descriptors, see Fig. 6 for an example. Another approach is to produce multiple instances of local descriptors in different scales at each local sample point on the shapes. However, we will need a large K and produce KN instances on the docking shape and KM instances on the collage contour. As a result, we will need K^2NM matching tests, which is still inefficient for collage generation.

In sharp contrast, PAD explicitly considers scale in its formulation when describing the local region around a point. Hence, it can effectively trim down the search space for scale-invariant partial-shape matching, where NM tests are sufficient for identifying the best docking location of a shape with scale invariance.

One may argue to use k-nearest neighbor (kNN) searching method to accelerate the search. One way is to build a kd-tree on the M^2 instances of a query shape for the case of global shape descriptors (or KM instances for local shape descriptors). This can reduce the number of matching tests from N^2M^2 to $N^2\log M$ for global shape descriptors (or from K^2NM to $KN\log(KM)$ for local shape descriptors). In our case, if we apply the same kNN method to PAD, we can further reduce the time complexity from NM to $N\log M$, so our method is still more efficient for partial-shape matching with scale invariance.

Besides 2D descriptors, there are 3D shape descriptors such as shape distribution [Osada et al. 2002] and heat kernel signature [Sun et al. 2009]. However, adaptation is required to apply them to 2D. More comprehensive survey of 3D descriptors can be found in [Tangelder and Veltkamp 2008].

3 Pyramid of Arclength Descriptor

The core of our collage generation approach is an efficient partial-shape matching method. To achieve that, we first describe a novel pyramid of arclength descriptor (PAD), which describes a local portion of the shape around each boundary point in a scale-invariant manner. Before presenting its formulation, we first define a scale-invariant domain.
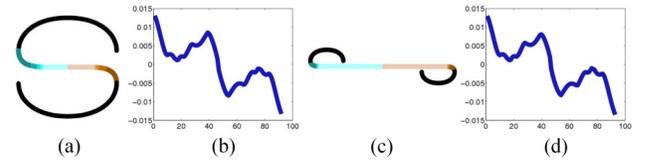
3.1 Scale-Invariant Domain

The scale-invariant domain we utilized is based on the *integral of absolute curvature*. Given a curve, the integral of absolute curvature, τ, over a curve segment between points s and t is defined as

$$\tau(s:t) = \int_s^t |\kappa(x)| \, \mathrm{d}x \, ,$$

where $\kappa(x)$ is the curvature at point x on the curve.

It can be easily proved that τ is scale-invariant. As curvature is inversely (linearly) proportional to the scale factor while arclength is linearly proportional to the scale factor. Their product cancels out the influence of scale. Hence, when the curve is scaled by a factor

Figure 8: *Plotting curvature against integral of absolute curvature (Cui et al. [2009]) may produce very similar signatures (b)&(d) even for different shapes (a)&(c). In sharp contrast, our PAD can produce substantially different descriptions (see the color coding in (a)&(c), like Fig. 7(c)), enabling us to differentiate the two shapes.*

of α, τ remains unchanged:

$$
\begin{aligned}
\tau(\alpha s : \alpha t) &= \int_{\alpha s}^{\alpha t} |\bar{\kappa}(\bar{x})| \, \mathrm{d}\bar{x} = \int_s^t |\bar{\kappa}(\alpha x)| \, \mathrm{d}\alpha x \\
&= \int_s^t |\kappa(x)| \, \mathrm{d}x = \tau(s:t) \, ,
\end{aligned}
$$

where $\bar{x}$ and $\bar{\kappa}$ are point and curvature, respectively, on the scaled curve. Note that the curvature of point x on the scaled curve $\bar{\kappa}(\alpha x)$ is inversely proportional to the scale factor, i.e., $\bar{\kappa}(\alpha x) = \frac{1}{\alpha}\kappa(x)$.
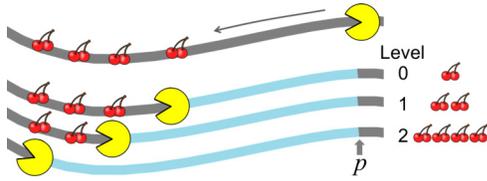
The integral of absolute curvature is well known in measuring the tightness of a surface. Hamann and Chen [1994] selected feature points in this domain, while Cui et al. [2009] designed a curve signature based on the integral of absolute curvature and performed normalized cross correlation (NCC) for scale-invariant curve matching. However, matching curves in this way may not work in general because different curves can have very similar signatures, see Fig. 8 for an example: the NCC value for matching the two curves shown in Fig. 8(a)&(c) are as high as 0.9989, where 1.0 means two curves are identical. Hence, having similar signatures may not imply a good match.

3.2 PAD Vector

Although describing a curve using the integral of absolute curvature can take away the influence of scale, it cannot precisely characterize a shape. In other words, we cannot solely rely on the information left in this single domain for shape matching. In this paper, we propose to utilize the shape information in two domains together to precisely characterize a shape. The second domain we utilized is the arclength domain, which is originally scale-dependent.

Consider a point p on the curve shown in Fig. 7(a). Our goal is to define a local scale-invariant descriptor to quantify the local shape centered at p. To do so, we construct a pyramid of arclength intervals centered at p (Fig. 7(c)), such that each interval corresponds to a fixed integral value of absolute curvature ($\Delta\tau$), cumulated from p. Our PAD is defined by using this set of arclength values.

To construct our PAD, we need to consider both left and right sides of p. To simplify the discussion, we denote l_i and r_i as the arclength

Figure 9: *Cumulating the integral of absolute curvature along the curve is analogous to Pacman collecting cherries on its way.*

Figure 10: *Collage generation with inward docking (red butterfly) or outward docking (black butterfly) onto the gray canvas.*

Figure 11: *This work uses arbitrary shapes of varying sizes.*

Figure 12: *(a) Signed distance field of S. (b) Docking S' onto S. (c) Manual docking S' in a single scale. (d) S' in multiple scales. (e) Our PAD considers scaling, translation, and rotation simultaneously, see some of the best $K\%$ matching candidates.*

values measured along the curve from the left and right sides of p, respectively, and both l_i and r_i correspond to the same amount of integral of absolute curvature, which is $2^i \Delta\tau$.

Clearly, one arclength level, say l_0 and r_0, may not uniquely describe a shape since different shapes may still have the same integral of absolute curvature and the same arclength value. Such a chance of "coincidence" can be significantly reduced by considering a pyramid of arclength values that correspond to different levels of absolute curvature integral, i.e., $2^i \Delta\tau$, etc. Hence, we form a 1D vector with n levels of "left" and "right" arclength values:

$$\begin{bmatrix} l_{n-1} & l_{n-2} & \cdots & l_1 & l_0 & r_0 & r_1 & \cdots & r_{n-2} & r_{n-1} \end{bmatrix}^T .$$

In other words, PAD's shape description power, or the ability to distinguish shapes, is *scalable*, i.e., controllable by n. In all our experiments, we use $n = 5$ and $\Delta\tau = 0.2$, which is sufficient for matching shapes in the collage results presented in the paper.

However, the above formulation is still scale-dependent. Hence, we further convert it to be scale-invariant by dividing each element in the vector by the corresponding last-level arclength value. After that, we can also remove the last-level elements since they are always one after the division. Thus, we define the PAD at p as

$$m(p) = s \begin{bmatrix} \hat{l}_{n-2} & \cdots & \hat{l}_1 & \hat{l}_0 & \hat{r}_0 & \hat{r}_1 & \cdots \hat{r}_{n-2} \end{bmatrix}^T .$$

where $\hat{l}_i = \frac{l_i}{2^i l_{n-1}}$ and $\hat{r}_i = \frac{r_i}{2^i r_{n-1}}$ and $s \in \{1, -1\}$ indicates the local shape convexity (+1) or concavity (-1) around p. Note that the weights 2^{-i} help normalize the influence of different arclength levels since the larger the level, the longer the arclength value is. With this normalization, we can bound the values to $[0, 1]$ and avoid the dominance of large magnitude PADs in our subsequent distance metric computation. Furthermore, unlike the *global* normalization approach in existing descriptors (see Section 2), our normalization is *local*, since we divide by the length of the longest *local* interval, so we can achieve local support with scale invariance.
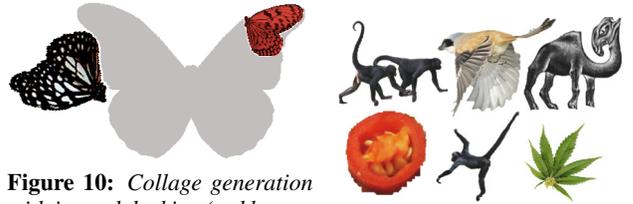
Fig. 9 illustrates another way to understand the PAD formulation. Imagine we distribute cherries along a curve at locations separated by equal amount of cumulative absolute curvature ($\Delta\tau$). Consider Pacman walks along the curve from p and collects cherries on its way. The length of the Pacman path to collect fixed numbers (2^i) of cherries corresponds to the arclength values in PAD vector.

The blown-up images in Fig. 7(a) shows example PADs at various locations of the shape. In general, curly segments lead to shorter arclengths, and smaller PAD values. It can also be easily seen that the above PAD vector is rotation-, translation-, and scale-invariant, and it is highly compact with only eight values for n=5.

Given $m(p)$ and $m(q)$ as PADs at points p and q, respectively, we define the PAD distance between the local shapes at p and q as:
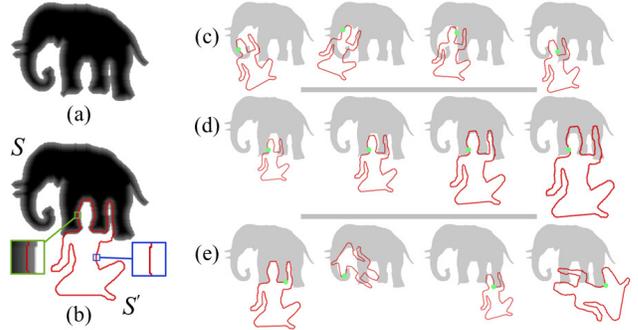
$$D(p, q) = \max_i \{ \, | \, m_i(p) - m_i(q) \, | \, \} , \qquad (1)$$

where $m_i(p)$ is the i-th element (normalized arclength) in $m(p)$. A small $D(p, q)$ indicates similar local shapes surrounding p and q, even these local shapes are of different scales.

So far, the description above refers to the scenario of inward docking, see Fig. 10, where we dock the red butterfly onto the interior of the gray canvas. In fact, we can also perform outward docking, e.g., see the black butterfly in Fig. 10. To do so, we just need to reverse the order of elements in $m(p)$ (or $m(q)$), equivalently flipping the left and right sides around p (or q), and flipping the sign for convexity and concavity (s in PAD) before applying Eq. 1.

3.3 Partial-Shape Matching using PAD

Manual partial-shape matching can be performed by iteratively docking the two given shapes at different positions (Fig. 12(c)) with the goal of maximizing the arclength shared between the shapes. However, this soon becomes intractable when we further consider multiple scales in the docking (Fig. 12(d)), as the search space expands with an extra continuous dimension.

PAD supports efficient partial-shape matching with scale invariance. To do so, we start with an arbitrary point p_i on contour of shape S and find the best matching point p'_j on shape S' with the smallest $D(p_i, p'_j)$. A small D indicates a good match between the local shape around p_i and the local shape around p'_j, and such a match is the best not just *over all positions* around S', but also *over all scales* of S'. In particular, PAD can drastically reduce the search space to find good matching locations, using only NM matching tests for N and M sample points on S and S', respectively.

Although PAD can highly efficiently find good matching locations in terms of local shapes, not all these locations are good if we consider longer arclength distances from the matched local regions, see Fig. 12(e). This is because PAD is a local descriptor, so it considers only local shapes for matching. Hence, we need further evaluation on locations reported from PAD. In detail, we retain as candidates the best $K\%$ matching pairs (among the NM pairs) with PAD distance $D <$ threshold β. In all our experiments, K=1 and β=0.4.

In the further evaluation, we explicitly transform and dock the two shapes at each candidate matching position and measure the arclength along the matched local portion. To do so, we first trans-
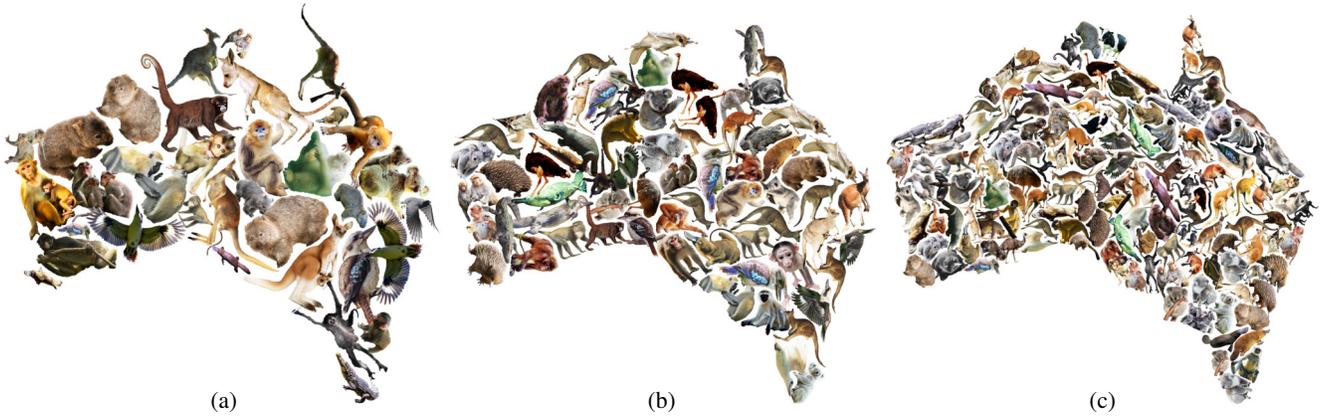
Figure 13: *"Australian Animals." Our collage results generated using the same set of input clip arts but different scale preferences.*

| Score: 0.525 | Score: 0.288 | Score: 0.247 | Score: 0.163 |

Figure 14: *Partial-matching similarity varies from a perfect match to a less matchable, and then a poorly-matching case (left to right).*

Figure 15: *Our bottom-up collage generation framework.*

form S' to the space of S by aligning the end-points of the first-level (finest) arclength intervals in the matched PADs. In detail, this alignment is done by i) identifying the left and right first-level end-points (see Fig. 7(a)&(c)) around the associated matched location on each shape (S and S'), and ii) rotating and scaling S', such that its two endpoints coincide with the corresponding endpoints of S. Second, we construct a signed distance field for shape S with zeros on the shape boundary (Fig. 12(a)). Depending on whether we dock S' onto S from outside or inside, the signed distance field can be synthesized with negative or positive interior values, respectively. Next, we rasterize the transformed S' on S's distance field, and measure the arclength of the matched portion along S''s boundary. The minimal distance from a boundary point on S' to the boundary of S can be easily looked up by the distance field (Fig. 12(b)). A boundary point on S' is classified as matched if the lookup distance is below a user defined threshold δ. We set δ as 4 pixels in our experiments. The total arclength of the matched portion is simply the count of matched boundary points. We repeat this dock-and-evaluate process for each candidate position, and determine the one with the longest matched arclength as the best docking position.

Thanks to the scale invariance of PAD, we need to perform docking and evaluation only *once* per distinct docking position (Fig. 12(e)). Hence, the evaluation remains tractable as we can efficiently find the top $K\%$ candidates. This is evidenced by the timing statistics shown in Section 5. Without PAD, such docking process has to be performed at all positions (Fig. 12(c)) in all scales (Fig. 12(d)), which is impractical even with modern GPU-equipped machines.

To demonstrate the description power of PAD and the further evaluation, we show the matching scores of docking the red duck shape onto four other shapes (Fig. 14). In each case, we plot the best matching position. From the scores (measured as the arclength along the shared boundary between a shape pair), one can see that our method can reasonably describe the change of partial-matching similarity as the matching situation changes from a perfect match to a less matchable, and further to a poorly-matching case.
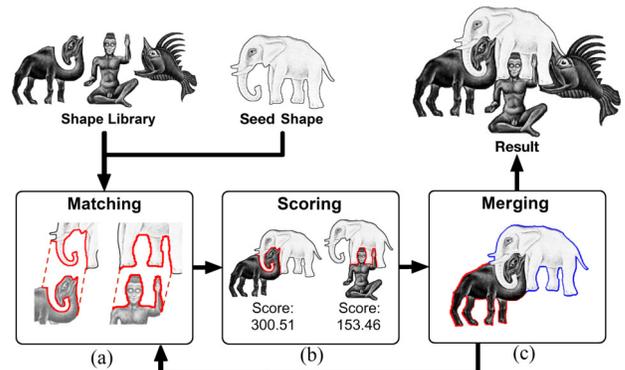
4 Collage Generation

Given a library of wide variety of shapes in arbitrary scale, we want to fill a given canvas with the shapes while minimizing the gap and overlap area among the shapes. Fig. 15 shows our collage generation framework, which is an iterative process. In each round, we start from a target shape, which is either a seed shape given by the user or a merged shape that represents the current collage (Fig. 15(c)). Then, we obtain the best $K\%$ candidates (Fig. 15(a)) for matching with the target shape by partial-shape matching. Note that these candidates may include the same shape docked on the target shape with different locations and scales. After that, we evaluate an objective function (see below) for each of them (Fig. 15(b)), and merge the best choice with the target shape (Fig. 15(c)) to form the new target shape for the next round.

In the matching process, we emphasize that maximizing the arclength shared between shapes (see red segments in Fig. 15(a)&(b)) is not the only criterion for aesthetic purpose. The amount of overlap between shapes, the size of gaps between shapes, the shape orientation, the selected shape scale, and the overall color composition of the collage all contribute to the quality of the results. Therefore, the second step in the framework is to score the matching candidates based on an application-driven objective function, e.g.,

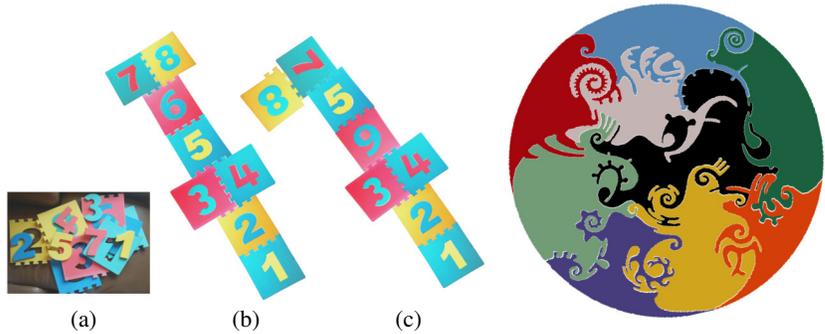$$Z = \frac{\omega_1 L}{\sqrt{\omega_2 \sum O_i^2 + \omega_3 \sum G_i^2}} \, ,$$

where L is the shared arclength between the shapes, O_i is the size of the ith overlapping region (since there can be multiple overlapping regions), G_i is the size of the ith gap region, and ω_i's are weights. The sum of squares formulation for O_i and G_i helps to penalize

Figure 16: *"Flock of Birds." A seamless collage that features infinite horizontal scrolling; orientation constraint is used in objective function.*
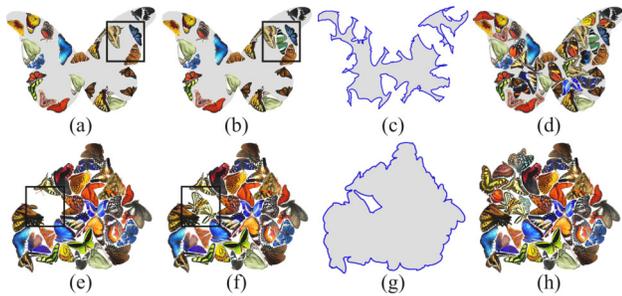

Figure 17: *"Doodle."*

Figure 18: *Puzzle solving. (a) The puzzle pieces. (b) Ground-truth. (c) Our result.*

Figure 19: *"A circle puzzle." A result with closely-packed pieces.* © *John S. Stokes III.*

Figure 20: *Multiple-boundary support. (a) To fill the gray region. (b) A new shape (inside the box) may divide the canvas into two parts. (c) Resulting boundary curves (blue). (d) Both parts filled with shapes. (e)-(h) show a similar scenario when filling a region.*

large and continuous overlapping and gap regions. Since different applications have different objectives, the above Z function is an exemplary objective function, which can be further extended or modified to meet different design goals.

In practice, we can also add constraints to avoid certain matching candidates. For instance, we may avoid packing a large shape with a tiny one if the candidate's size exceeds a preferred scale range. Fig. 13 shows three versions of "Australian Animals," each generated using a different size range. Another common constraint is orientation. Some objects may not be easily recognizable when they are upside-down, so we may avoid a candidate if its orientation deviates too much, e.g., we generate Fig. 16 with this constraint.

The last step in each iteration is shape merging, where we combine the best candidate shape with the current target to form the new

target shape (Fig. 15(c)). Due to the nice local-support property of PAD, we do not need to recalculate the entire PAD for the merged shape. We can simply "cut and paste" corresponding runs of PADs from the target and candidate shapes (see the blue and red runs in Fig. 15(c)) for the new target. The only computation needed is to recompute the PADs around the joints between the two runs.

Collage in a Canvas Shape Besides generating collages without boundary, we may prepare a canvas shape to bound a collage, e.g., the gray butterfly shape shown in Fig. 10. To do so, we take the canvas boundary as the seed shape and perform inward rather than outward docking. Moreover, since the collage generation process may break the canvas region into disjoint partitions, e.g., see the gray regions in Fig. 20(c), we use multiple boundary curves to represent the interior regions for filling shapes. Note also that similar scenario may happen when we grow a region (Fig. 20(g)).

Discussion Since PAD is a curvature-based descriptor, one may wonder if our collage generation method could be too strict. However, it is worth to note that our matching process is not solely determined by a single curvature profile, since PAD encodes arclengths in multiple scales based on accumulative curvature. Hence, it is not sensitive to small shape changes, see the examples in Fig. 14.

5 Results and Discussion

We create multiple collage results by using clip arts collected from the Internet, e.g., Fig. 1, 13, 16, 17 and 21. Altogether, we have collected more than 1,200 distinct clip arts. Fig. 13 presents three collage versions created with 130 Australian animal photos using a simplified Australia outline as the canvas boundary. We synthesize these results solely by adjusting the scale-range constraint in the

(a)   (b)

Figure 21: *"Butterflies in Butterfly." (a) An initial collage result. (b) After deformation.*
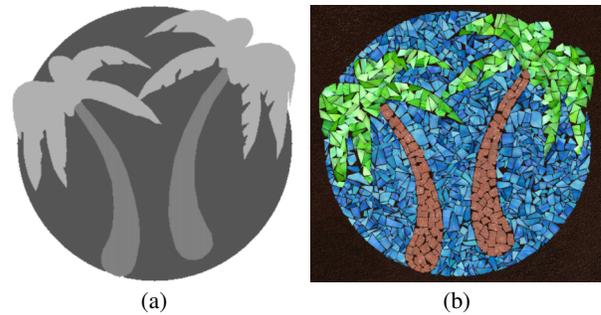
objective function, and use the same set of clip arts to produce all these results without any modification on the input clip arts.

Different collage results can be obtained by using a different seed shape or different weights in the objective function. We may also create a seamless seed shape that is cyclic. Fig. 16 shows one such example with matched clip arts on left and right borders. In addition, to improve the visual quality and to present individual clip art more clearly, we may introduce a padding around each clip art shape in the generated collages. Readers can refer to the supplementary material for more collage results, as well as some of the clip-art sets employed to generate the results.
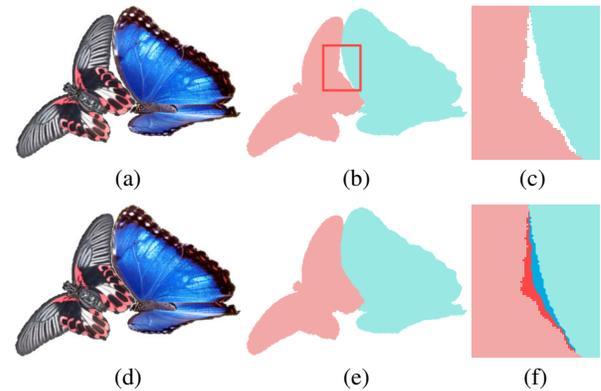
Creative Design   Fig. 1 and 17 demonstrate the potential of our method for design purpose. Existing artworks similar to Fig. 1 are generally created by manually packing the clip arts piece by piece in a labor intensive manner. By using our method, we can efficiently fill a target region with a lot of clip arts fully automatically, e.g., we fill a region in the shape of letters PAD (see Fig. 1) by using a library of around 100 distinct icons. Recall that our method allows a shape to possess multiple separate boundaries. If the users want to control the placement of certain pieces, they can simply place the pieces onto the target region; our method can take their boundary as part of the seed shape. Fig. 17 shows one such example, where individual characters in SIGGRAPH and the canvas together form a nontrivial seed shape for the collage. Users can also control the scale range and shape orientation by modifying the objective function.

Puzzle Solving   As a side product, our method can also be used to solve puzzles, although we have to emphasize that puzzle solving is not our major strength. It is because our iterative method may not find the ground truth (if exists), since it may be trapped by a local optimum. Fig. 18 and 19 show two puzzle solving examples. Since the puzzle pieces in Fig. 19 are unique in shape, our method can obtain the ground truth. This result also shows that our method can identify strongly-coupled shapes if the input library has any. For the case of Fig. 18, due to the non-uniqueness of the shapes, our method (Fig. 18(c)) cannot generate the target result (Fig. 18(b)), but it can fill all the numbers with their corresponding floor mats since the interior boundaries of the mats are unique.

Object-based Texture Synthesis   Another interesting application of our method is object-based texture synthesis, e.g., mosaic texture. Given an image of an object composed of discrete elements (e.g., stone wall and mosaics), we can extract some of the discrete elements from the image by segmentation, and then gather them as
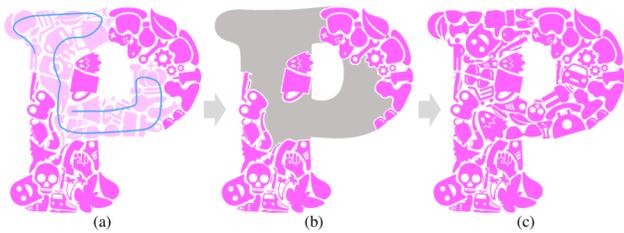

(a)   (b)

Figure 22: *Image mosaic. (a) Seed shapes (each in a different grayscale). (b) Our synthesized mosaic.*

(a)   (b)   (c)

(d)   (e)   (f)
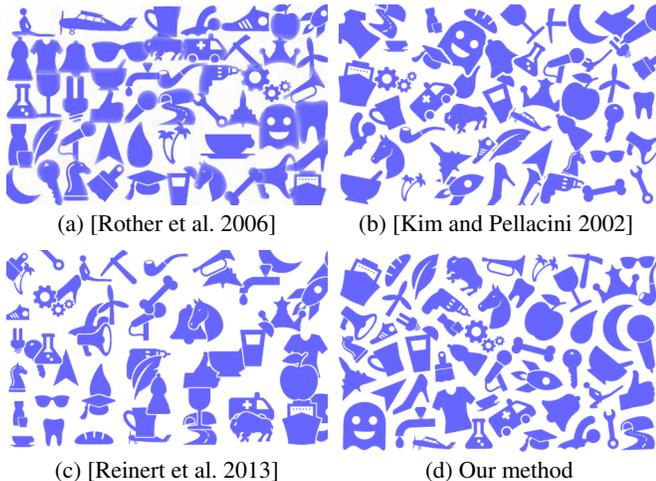
Figure 23: *Deformation. (a) Initial collage result. (b) Color-code the shapes. (c) Blown-up view on the gap. (d) Deformed shapes. (e) Corresponding color coding. (f) Voronoi-based deformation.*

a library of clip arts. Fig. 22 shows an example. We extract the mosaic tiles from some real mosaic photographs. These tiles together form an input clip-art library, with which we fill each grayscale region in Fig. 22(a) by matching both color and shape. Readers are recommended to zoom into Fig. 22(b) for a better visualization.

Deformation   Since we generate collages with arbitrary clip arts, we may not always avoid gaps and overlaps. Concerning this, we include an *optional* deformation step to enhance the collage generation results. To do so, we create a Voronoi diagram in the gap and overlap regions, and project the original shape boundary onto

Figure 24: *User-controlled local refinement.*

Method	Preference	Stylish	Coupling
[Rother et al. 2006]	2.17	3.25	3.58
[Kim and Pellacini 2002]	3.83	3.58	3.00
[Reinert et al. 2013]	3.00	2.42	2.64
Our method	5.17	5.17	4.67

Table 1: *User study results. Average scores in range [1, 6]: a higher score indicates better rating given by the participants.*

(a) [Rother et al. 2006]   (b) [Kim and Pellacini 2002]

(c) [Reinert et al. 2013]   (d) Our method

Figure 25: *Collage results generated by four different methods (including ours (d)) using the same set of input shapes.*

the boundary of the associated Voronoi cell. By this projection, we can deform the shapes by standard image morphing. Fig. 23(a)-(c) show the gap in-between two butterfly shapes, whereas Fig. 23(d)-(e) show how the gap is filled. In addition, Fig. 23(f) shows the Voronoi cells in darkened colors. This optional deformation step works well mainly for organic shapes, e.g., butterfly, see how it deforms the collage result in Fig. 21(a) and produces Fig. 21(b).

User-controlled Local Refinement   When designing collages, artists usually need to refine a result iteratively to improve its aesthetics. We offer an *optional* tool for local refinement of collages, see Fig. 24. User may first scribble to mark a group of pieces for removal (Fig. 24(a)), and our tool can automatically remove the associated shapes (Fig. 24(b)) and refill the empty region (Fig. 24(c)). Usually, the refilled result is not the same as before, since the tiling order is likely to be different from the original one, and we also introduce randomness into the refilling process to create more variety.

Comparison with Existing Collage Methods   We compare our method with three existing top-down methods: AutoCollage [Rother et al. 2006], jigsaw image mosaic (JIM) [Kim and Pellacini 2002], and packing layout [Reinert et al. 2013]. Since these methods fundamentally have a different design goals, degrees of freedom and characteristics, it may not be ideal to compare them with our work; however, it is worth to note that these methods are already the most closely-related works in terms of collage generation. Fig. 25 shows collages generated from them as well as from ours, using the same set of input shapes with similar setting. In addition, we limit each shape to appear exactly once, reduce the orientation changes, and do not allow irregular deformation and manual intervention when generating these collages. Readers are recommended to zoom into Fig. 25 for a better visualization. Note that

to ensure the same set of shapes to be shown in results of all four methods for comparison, we first use our method to generate a collage and identify the shapes inside (Fig. 25(d)). Then we feed these shapes as inputs to other methods to generate the other collages.

To quantify the visual quality of the results, we conduct a user study with 13 participants. In the study, we show the results to each participant in a random order to avoid bias. For each result, the participant has to grade it with a score from 1 (the worst) to 6 (the best) in terms of their overall preference, shape coupling, and stylishness. Table 1 shows the statistics. Among the methods, most participants agree that our result exhibits the strongest coupling, and is the most stylish one. In addition, our result is also the most preferred.

AutoCollage (Fig. 25(a)) can avoid overlap regions of interest, but it fails to suppress gaps and cannot effectively couple the shapes. Both JIM (Fig. 25(b)) and packing layout (Fig. 25(c)) implicitly assume the shapes are more-or-less convex, so they may fail to couple the shapes as well. For instance, JIM divides the canvas into cells and selects a shape to fit into each cell; since we limit each shape to appear once for comparison, this further limits the choice of shapes in the collage generation process. Hence, some highly concave shapes could have to be put into some convex cells, thereby leading to poorer collage results. Similarly, packing layout (Fig. 25(c)) also has a large degree of shape overlap in result. One possible explanation is due to the unsuccessful initial Voronoi-based distribution of shapes, and the subsequent trapping by the local optimum during the adjustment. Since there is no user intervention, serious overlap is observed as it fails to escape from the local optimum. In sharp contrast, we can employ our objective function to penalize overlap and gap, large deviation in orientation, as well as large deviation in scale. Together with the maximization of shape coupling via PAD, we can obtain the best result among the methods (Fig. 25(d)) in terms of various metric on visual quality.

Comparison with Existing Descriptors   To demonstrate the effectiveness of PAD in collage generation, we compare it with several state-of-the-art shape descriptors. In the first experiment, we evaluate the descriptor's ability in performing partial-shape matching, while in the second experiment, we compare the time performance of applying different descriptors in generating collages.

In the first experiment, we employ a large set of 1,400 shapes. For each shape, we produce ten blocking instances by linearly clipping the shape from left to right (Fig. 26 (left)). Then, we try to match the clipped instances with the original complete shape. Through this experiment, we can observe at what level of blocking, the descriptor fails to recognize the shape. Four state-of-the-art descriptors are compared: shape context [Belongie et al. 2002], triangle area representation [Alajlan et al. 2007], curvature scale space [Mokhtarian et al. 1996], and integral invariants [Manay et al. 2004]. Fig. 27 plots the success rate against the blocking ratio, showing that PAD performs the best. Even up to 80% blocking, we still achieve a success rate of 50%. In contrast, the success rates of all other descriptors drop sharply when the blocking reaches ≈20%. Their inferior performance is mainly due to their inability of partial-shape matching.

Cases	No. of candidates	No. of PADs per shape	No. of iterations	No. of docking choices	% Pruned	Total time	Pruning of docking choices	Distance field evaluation	Objective function evaluation	Merging
Fig. 1	120	4,960	194	7.5×9^{10}	99.96	186.3 min	16%	3%	80%	1%
Fig. 13(a)	128	6,804	334	2.1×10^{10}	99.95	22.8 min	49%	12%	36%	3%
Fig. 13(b)	128	6,804	69	2.6×10^{10}	99.94	51.9 min	57%	15%	26%	3%
Fig. 13(c)	128	6,804	143	2.0×10^{10}	99.93	122.9 min	64%	12%	22%	2%
Fig. 21	83	4,201	102	1.1×10^{10}	99.83	405.3 min	17%	7%	27%	49%
Fig. 16	86	3,631	90	3.0×10^{9}	99.90	7.0 min	42%	11%	28%	19%
Fig. 17	112	4,136	92	6.0×10^{9}	99.04	25.7 min	15%	51%	25%	9%
Fig. 18	18	12,340	18	4.0×10^{15}	99.09	3.2 min	10%	47%	1%	42%
Fig. 19	9	50,247	9	1.1×10^{9}	99.95	0.2 min	23%	7%	69%	1%
Fig. 22	734	1,225	1,421	3.8×10^{10}	99.99	718.3 min	11%	1%	59%	29%

Table 2: *Timing statistics of our collage results.*
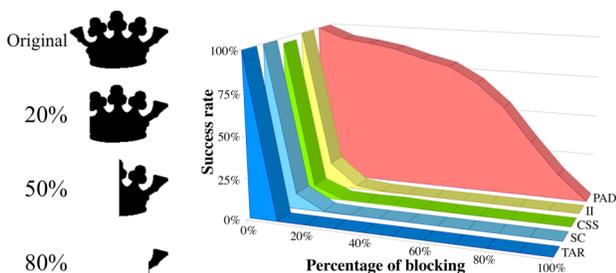

Figure 26: *Example blocking instances.*

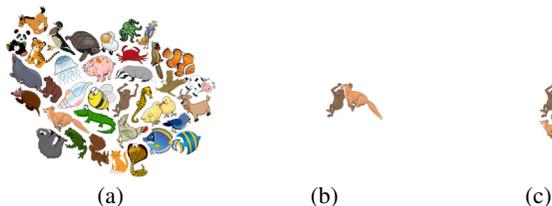**Figure 27:** *Plot of success rate against blocking ratio (see Fig. 26).*

Figure 28: *Collage generation with (a) our PAD, (b) extended shape context, and (c) extended Hu moments.*

One may argue that we may extend an existing descriptor to support partial-shape matching, e.g., by naïvely dividing the shape into multiple (theoretically infinite) curve segments and performing shape matching with these instances, see Section 2. In the second experiment, we evaluate this naïve extension by comparing PAD with two descriptors, shape context [Belongie et al. 2002] and Hu moments [Gal et al. 2007] (used in scale-dependent 3D collage). Note that, due to the nature of descriptors, not all descriptors used in the first experiment can be extended for use in the second experiment. In this experiment, we replace PAD matching with shape context and Hu moments (both extended naïvely for partial-shape matching), but retain the collage generation framework. For fairness, all descriptors (including ours) utilize only CPU in the computation, and we employ the same set of 35 distinct shapes, where the monkey shape is the seed and each shape appears only once. All three methods are allowed to run for 12 hours. Our method completes the task (Fig. 28(a)) after 750 seconds, while the other two can only complete one and three iterations (Fig. 28(b)&(c)) after 12 hours. Although their results look comparable to ours (thanks to the same collage generation framework), they can only pack a few shapes within the time limit. This evidences the efficiency of PAD.

Implementation and Performance   To prepare an input clip art for collage generation, we fit its boundary with B-splines and compute local curvature along the B-splines. We then sample PADs densely and evenly along the boundary. This is an offline data preparation step performed only once per input clip-art shape.
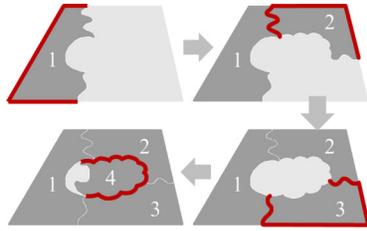
Even though we have a dense sampling of PADs, our method can effectively prune away most candidate docking positions by measuring their PAD distances. No further evaluation with the distance field is needed for these candidates. Moreover, both the docking and distance field evaluation procedures are perfectly parallelizable, so we perform them on the GPU by using CUDA. This enables us to efficiently compute the partial-shape matching. For instance, our method consumes only 64 milliseconds on average to evaluate two shapes with 25,000,000 docking positions. This amount of docking candidate positions already excludes the scale dimension as PAD is scale-invariant. In addition, the signed distance field design can benefit from the hardwired texture look-up functionality available on the GPU, including value access and interpolation.

Table 2 details the timing statistics in generating the collage results shown in the paper. It tabulates the number of candidates in an associated clip-art library, the average number of local PADs per clip art, the number of iterations (i.e., the number of pieces in the collage result), the number of candidate docking choices, the percentage of docking choices pruned before further evaluation, and the total running time. The total running time is further divided into four parts: pruning, distance field evaluation, scoring, and merging. All the experiments are performed on a machine with dual Intel Xeon E5-2670 CPUs with 8 NVIDIA Tesla K20m GPUs. Obviously, the time for distance field evaluation is highly affected by K. Thanks to the description power of PAD, we can prune away more than 99% of docking choices before the more time-consuming distance field evaluation. Hence, we can work with the originally intractable collage problem in a tractable manner.

Throughout all our experiments, the number of PAD levels (n) is 5 and the integral value of absolute curvature $\Delta\tau$ is 0.2. Note that these parameters are scale-invariant since they roughly associate with how human interprets curves. For the sampling rate of PADs along boundary, we sample roughly one PAD for every 0.2 pixel units along the clip-art boundary.

Limitations   One limitation of PAD is that it relies on curvature. This means that individual straight lines in open curves cannot be handled by our method since straight line has zero curvature value. However, matching straight lines can simply be done by matching endpoints of the lines without requiring PAD-based matching. The second limitation concerns with the iterative process undertaken by the collage generation framework. Since it is greedy by nature, we may not be able to obtain the global optimum re-

sult. Moreover, when we iteratively pack shapes into a collage (see inset figure for an example), we may eventually form hole(s) in the collage. It is more likely to happen in later steps of the packing. Depending on the richness of shape variety in the clip-art library, we may or may not be able to find a good shape that well-matches all neighboring shapes. For instance, the best-matched shape "4" in the example can only well-match the red boundary while leaving a gap in-between shapes "1" and "4." Therefore, we cannot guarantee perfect shape coupling over the entire collage.

On the other hand, some shapes may intersect others in the results, particularly if we want to more closely pack the shapes with small gaps. It is because unless we have a very rich set of input shapes, when the packing process tries to fit a hole (see inset figure above), it may have to sacrifice the overlap criterion for the gap criterion (see O_i and G_i in the objective function in Section 4). However, such overlap is usually small and not obvious, and the user may apply local refinement and deformation to try to avoid it.

Another limitation is that our current PAD design may not be noise-proof. Noise may sometimes be regarded as features in the scale-invariant domain. Lastly, our current implementation does not consider color harmony and style consistency among the input clip arts. Unless incorporated into the objective function, color information is not considered in the collage generation. Hence, the color composition in a collage result may not be harmonious. To resolve this issue, we suggest to perform the color harmonization [Cohen-Or et al. 2006]. Similarly, style inconsistency may also worsen the quality of the collage results. We recommend to adopt the style-similarity measurement proposed by Garces et al. [2014] to avoid putting together clip arts of inconsistent style. However, this may significantly increase the computational overhead.

6 Conclusion

This paper presents an efficient bottom-up solution to generate collages with arbitrary and irregular shapes (not only more-or-less convex) in a scale-invariant fashion. To achieve this, we need a fast partial-shape matching process to quickly identify docking shapes with the best matching position, scale, and orientation. However, existing approaches involve exhaustive tests over an exceedingly large number of docking combinations in all scales, so they are infeasible to support the collage generation. Our key to overcome this issue is a novel shape descriptor, pyramid of arclength descriptor (PAD), which quantifies local shapes with scale invariance. By PAD, we can efficiently perform partial-shape matching while considering scales simultaneously. Therefore, we can drastically reduce the search space, making bottom-up collage generation feasible. Moreover, PAD's local-support property allows us not just to uniformly describe both closed shape and open curve, but also to perform partial-shape matching even in the presence of partial occlusion. Furthermore, we demonstrated the effectiveness of PAD through various collage examples and generated results with over 1,200 clip arts. Lastly, PAD may also contribute to many other shape-recognition applications such as road-sign recognition.

Acknowledgments

References

ALAJLAN, N., EL RUBE, I., KAMEL, M. S., AND FREEMAN, G. 2007. Shape retrieval using triangle-area representation and dynamic space warping. *Pattern Recognition 40*, 7, 1911–1920.

ARKIN, E. M., CHEW, L. P., HUTTENLOCHER, D. P., KEDEM, K., AND MITCHELL, J. S. 1991. An efficiently computable metric for comparing polygonal shapes. *IEEE Trans. Pat. Ana. & Mach. Int. 13*, 3, 209–216.

BELKASIM, S. O., SHRIDHAR, M., AND AHMADI, M. 1991. Pattern recognition with moment invariants: A comparative study and new results. *Pattern Recognition 24*, 12, 1117–1138.

BELONGIE, S., MALIK, J., AND PUZICHA, J. 2002. Shape matching and object recognition using shape contexts. *IEEE Trans. Pat. Ana. & Mach. Int. 24*, 4, 509–522.

CHUANG, G.-H., AND KUO, C.-C. 1996. Wavelet descriptor of planar curves: Theory and applications. *IEEE Trans. on Image Processing 5*, 1, 56–70.

COHEN-OR, D., SORKINE, O., GAL, R., LEYVAND, T., AND XU, Y.-Q. 2006. Color harmonization. *ACM Trans. on Graph (SIGGRAPH) 25*, 3, 624–630.

CUI, M., FEMIANI, J., HU, J., WONKA, P., AND RAZDAN, A. 2009. Curve matching for open 2D curves. *Pattern Recognition Letter 30*, 1 (Jan.), 1–10.

DALAL, K., KLEIN, A. W., LIU, Y., AND SMITH, K. 2006. A spectral approach to NPR packing. In *Int. Symp. on Non-Photorealistic Animation and Rendering*, ACM, 71–78.

DONOSER, M., RIEMENSCHNEIDER, H., AND BISCHOF, H. 2010. Efficient partial shape matching of outer contours. In *Asian Conf. on Comp. Vis.* Springer, 281–292.

GAL, R., SORKINE, O., POPA, T., SHEFFER, A., AND COHEN-OR, D. 2007. 3D collage: expressive non-realistic modeling. In *Int. Symp. on Non-Photorealistic Animation and Rendering*, ACM, 7–14.

GARCES, E., AGARWALA, A., GUTIERREZ, D., AND HERTZMANN, A. 2014. A similarity measure for illustration style. *ACM Trans. on Graph (SIGGRAPH) 33*, 4, 93:1–93:9.

GOFERMAN, S., TAL, A., AND ZELNIK-MANOR, L. 2010. Puzzle-like collage. *Computer Graphics Forum 29*, 2, 459–468.

GOLDBERG, D., MALON, C., AND BERN, M. 2004. A global approach to automatic solution of jigsaw puzzles. *Computational Geometry: Theory and Applications 28*, 2-3 (June), 165–174.

GRANLUND, G. H. 1972. Fourier preprocessing for hand print character recognition. *IEEE Trans. on Comp. 100*, 2, 195–201.

HAMANN, B., AND CHEN, J.-L. 1994. Data point selection for piecewise linear curve approximation. *Computer Aided Geometric Design 11*, 3, 289–301.

HAUSNER, A. 2001. Simulating decorative mosaics. In *Proc. of SIGGRAPH*, 573–580.

HONG, B.-W., AND SOATTO, S. 2015. Shape matching using multiscale integral invariants. *IEEE Trans. Pat. Ana. & Mach.*

Int., 1, 151–160.

HU, W., CHEN, Z., PAN, H., YU, Y., GRINSPUN, E., AND WANG, W. 2016. Surface mosaic synthesis with irregular tiles. *IEEE Trans. Vis. & Comp. Graphics 22*, 3 (Mar.), 1302–1313.

HU, M.-K. 1962. Visual pattern recognition by moment invariants. *IRE Trans. on Info. Theory 8*, 2, 179–187.

HUANG, H., ZHANG, L., AND ZHANG, H.-C. 2011. Arcimboldo-like collage using internet images. *ACM Trans. on Graph (SIGGRAPH Asia) 30*, 6 (Dec.), 155:1–155:8.

HUANG, Z., WANG, J., FU, H., AND LAU, R. W. 2014. Structured mechanical collage. *IEEE Trans. Vis. & Comp. Graphics 20*, 7, 1076–1082.

KAPLAN, C. S., AND SALESIN, D. H. 2000. Escherization. In *Proc. of SIGGRAPH*, 499–510.

KAPLAN, C. S., AND SALESIN, D. H. 2004. Dihedral escherization. In *Proc. of Graphics Interface*, Canadian Human-Computer Communications Society, 255–262.

KHOTANZAD, A., AND HONG, Y. H. 1990. Invariant image recognition by Zernike moments. *IEEE Trans. Pat. Ana. & Mach. Int. 12*, 5, 489–497.

KIM, J., AND PELLACINI, F. 2002. Jigsaw image mosaics. In *SIGGRAPH 2002*, 657–664.

KRISH, K., AND SNYDER, W. 2008. A new accumulator-based approach to shape recognition. In *Advances in Visual Computing*, vol. 5359. 157–169.

LEE, S.-M., ABBOTT, A. L., CLARK, N. A., AND ARAMAN, P. A. 2006. A shape representation for planar curves by shape signature harmonic embedding. In *Proc. IEEE Conf. on Comp. Vis. and Pat. Rec.*, vol. 2, 1940–1947.

MANAY, S., HONG, B., YEZZI, A., AND SOATTO, S. 2004. Integral invariant signatures. In *Proc. Euro. Conf. on Comp. Vis.*, Springer, 87–99.

MOKHTARIAN, F., AND MACKWORTH, A. K. 1992. A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Trans. Pat. Ana. & Mach. Int. 14*, 8, 789–805.

MOKHTARIAN, F., ABBASI, S., AND KITTLER, J. 1996. Efficient and robust retrieval by shape content through curvature scale space. In *Int. Workshop on Image Databases and Multimedia Search*, 35–42.

ORCHARD, J., AND KAPLAN, C. S. 2008. Cut-out image mosaics. In *Int. Symp. on Non-Photorealistic Animation and Rendering*, 79–87.

OSADA, R., FUNKHOUSER, T., CHAZELLE, B., AND DOBKIN, D. 2002. Shape distributions. *ACM Trans. on Graph 21*, 4, 807–832.

PERSOON, E., AND FU, K.-S. 1977. Shape discrimination using Fourier descriptors. *IEEE Trans. on Systems, Man and Cybernetics 7*, 3, 170–179.

REINERT, B., RITSCHEL, T., AND SEIDEL, H.-P. 2013. Interactive by-example design of artistic packing layouts. *ACM Trans. on Graph (SIGGRAPH Asia) 32*, 6 (Nov.), 218:1–218:7.

ROTHER, C., BORDEAUX, L., HAMADI, Y., AND BLAKE, A. 2006. Autocollage. In *ACM Trans. on Graph (SIGGRAPH)*, 847–852.

SHENG, Y., AND SHEN, L. 1994. Orthogonal Fourier-Mellin moments for invariant pattern recognition. *The Journal of the Optical Society of America-A 11*, 6, 1748–1757.

SUN, J., OVSJANIKOV, M., AND GUIBAS, L. 2009. A concise and provably informative multi-scale signature based on heat diffusion. In *Computer Graphics Forum*, vol. 28, Wiley Online Library, 1383–1392.

TABBONE, S., WENDLING, L., AND SALMON, J.-P. 2006. A new shape descriptor defined on the Radon transform. *Comp. Vis. & Image Understanding 102*, 1, 42–51.

TANGELDER, J. W., AND VELTKAMP, R. C. 2008. A survey of content based 3D shape retrieval methods. *Multimedia tools and applications 39*, 3, 441–471.

VAN KAICK, O., ZHANG, H., HAMARNEH, G., AND COHEN-OR, D. 2011. A survey on shape correspondence. In *Computer Graphics Forum*, vol. 30, Wiley Online Library, 1681–1707.

VELTKAMP, R. C., AND HAGEDOORN, M. 2001. State of the art in shape matching. In *Principles of visual information retrieval*. Springer, 87–119.

XU, J., AND KAPLAN, C. S. 2007. Calligraphic packing. In *Proc. of Graphics Interface*, ACM, 43–50.

YANG, M., KPALMA, K., AND RONSIN, J. 2008. A survey of shape feature extraction techniques. *Pattern Recognition*, 43–90.

YAO, F.-H., AND SHAO, G.-F. 2003. A shape and image merging technique to solve jigsaw puzzles. *Pattern Recognition Letters 24*, 12, 1819–1835.

ZHANG, D., AND LU, G. 2002. Shape-based image retrieval using generic Fourier descriptor. *Signal Processing: Image Communication 17*, 10, 825–848.

ZOU, C., CAO, J., RANAWEERA, W., ALHASHIM, I., TAN, P., SHEFFER, A., AND ZHANG, H. 2016. Legible compact calligrams. *ACM Trans. on Graph (SIGGRAPH) 35*, 4 (July), 122:1–122:12.