Structured Shape-Patterns from a Sketch: A Multi-Scale Approach Supplementary Material

In this supplementary material, we first provide some computational details of the analysis. Sect. 2 describes the overlap control in the synthesis steps of our method. Sect. 3 provides details of the user study that we performed to validate our results.

1 Fine-to-Coarse Analysis: Computational Details

1.1 From Strokes to Shapes

We compute the centroid of a stroke as follows. We first determine whether a stroke can be considered as opened or closed by computing the distance between its first point and last point. If it is considered opened then we define its centroid as the mid-point between its first point and last point; if it is closed, the centroid is the barycenter of the stroke coordinates. We use the Euclidean distance between two centroids as our clustering function.

For the distance between bounding boxes, we define for each stroke its oriented bounding box (OBB), after computing the stroke main directions using Principal Component Analysis. The distance between two bounding boxes is defined as follows. Each bounding box is defined by four vertices (corners) and two main directions. We project all eight vertices along the four main directions of the two OBBs and check the minimum and maximum values for each quartet of vertices. For each direction, if the minimum/maximum of one box overlaps the minimum/maximum of the other, then the two bounding boxes are considered overlapping and their distance is 0. Otherwise, we first compute the projection of each vertex of one box along each segment of the other box and the opposite. Finally if needed, we compute the distance between the vertices.

1.2 Clustering algorithms & parameters

Here are more details on our clustering algorithms and their parameters. To cluster bounded strokes into bounded shapes, we rely on the bounding box distance presented above and we compute the pairwise distance between the input strokes. For this clustering, we use a classical clustering algorithm and we group the bounding boxes that are intersecting (distance threshold of 0). To cluster central points that are the closest, we are also using a classical clustering algorithm but with the Euclidean distance and a threshold that we empirically determined at 0.35. For the bounded segments, we first determine the first set of clusters by grouping them by orientation, using an implementation of the Mean Shift algorithm with a bandwidth parameter of 0.2 and the Euclidean distance on the distance between their main direction. Then, we refine these clusters with another Mean Shift algorithm with a distance between the segment's centers and bandwidth of 6.

For the fibers, we define two lines belonging to the same anisotropy by having them either intersect and having a difference of orientation lower than 0.05 or not intersect and having an orientation difference lower than 0.2. Finally, to refine these clusters, we rely on our perceptual distance. We determined the bandwidth by computing the minimal value of the data on each axis and we take as our bandwidth the minimal value between this value and 0.1.

The table below presents a sum up of our clustering, algorithm, distance, and parameters, respectively.

Elements	Algo.	Dist.	Param.
Bounded box	Classic	bounded box	0
Point pos.	Classic	Euclidean	0.35
Segments or.	Mean Shift	Euclidean	0.2
Segments pos.	Mean Shift	Euclidean	6
Fibers or.	Classic	Euclidean	0.2
Fibers pos.	Mean Shift	perc. dist	calculated

Note that we provide a debug menu to let the user change these parameters if necessary.

1.3 Displacement Areas for Bounded Shapes

The displacement area of each bounded texture (represented with dashed lines in Fig. 11 c) in the paper) is computed as the ranges of allowed values for perturbing the position of the texture's center along the support median and its orthogonal direction. These ranges are computed as follows. For each support segment, we first compute the bounding box of the associated texture, using its extreme values along the support median of the segment and the orthogonal direction. For each support median, and based on the box position, we first sort the segments along the orthogonal axis to the median, to determine the allowed displacement for each, using a sweeping algorithm. We then update the bounding box of the shapes with these new displacement areas, before sorting these boxes along the support median primary axis and applying the same algorithm to determine the displacement along the orthogonal axis.

2 Coarse-to-Fine Synthesis: Overlap control through curved structures

The curvature computation for the ribbon borders is detailed as follows.

In case of overlap, each ribbon will collide twice with its lead ribbon, on both sides of IS. For symmetry reasons, we then bend both sides of the ribbon, even if one of the intersecting regions is out of the output domain. We keep as intersection points the ones that are closest to the fiber median midpoint (I_1 and I_2 in red in Fig. 12 of the paper). Among them, one belongs to the ribbon upper border and the other one to the lower border. This gives the direction in which the ribbon should be bent. We project the intersection points on the other side of the ribbon (P_1 and P_2 in blue in Fig. 12), as this is where the curvature radius is the lowest. We then generate both border curves and shift the ribbon back onto the fiber median. To preserve continuity between the original line border of the ribbon and its curved version, we consider the midpoints (M_1 and M_2 in green in Fig. 12) between a projected point and the other intersection point as inflection points. For each of these inflection points (M), the key idea is to find the circle C that passes through M and remains inside the lead ribbon, as illustrated in Fig. 12.

By symmetry, we will explain it for one border curve, the other will be handled similarly.

The idea is to shift P along the ribbon's secondary axis (Lm in Fig. 12), and towards the interior of the lead ribbon. Let P' be the translated point and α the value of this displacement. We have $\alpha \in]0; \alpha_{limit}[$, α_{limit} representing the displacement to the point P'_{limit}

that is the intersection between Lm and the line passing by M and following the lead ribbon principal axis (Lt in Fig. 12). Taking an α value closer to 0 will result in a higher curvature while taking it closer to α_{limit} will result in a lower curvature but in a direction closer to the fiber median.

In the following, we use $\overrightarrow{n_u}$ to represent the normalized vector of vector $\vec{u}$ and (x_A, y_A) for the coordinates of point A. To find the radius and center of C, we have the following properties :

1. $M \in Lt$, $M \in C$, and Lt is tangent to C on M,
2. $P \in Lt$, $P' \in C$, and $P' = P + \alpha * \overrightarrow{n_{Lm}}$

Let $a*x+b*y+c=0$ be Lt's line equation with $a,b,c \in \mathbb{R}$ and $a^2+b^2=1$. From the first property, we have :

$$\overrightarrow{OM} = (x_M - x_O, y_M - y_O) \tag{1}$$

$$\left\|\overrightarrow{OM}\right\| = R_C \qquad \overrightarrow{n_{OM}} = \pm(a,b) \tag{2}$$

The sign of $\overrightarrow{n_{OM}}$ depends on the curvature turn side, or more precisely if $\overrightarrow{n_{OM}}$ is in the same direction as the normal of the fiber median direction. From these equations, we obtain:

$$O = M + R_C * \overrightarrow{n_{OM}} \iff \begin{cases} x_O = x_M \pm R_c * a \\ y_O = y_M \pm R_c * b \end{cases} \tag{3}$$

From the definition of P', we have :

$$P' = P + \alpha * \overrightarrow{n_{OM}} \iff \begin{cases} x_{P'} = x_P \pm \alpha * a \\ y_{P'} = y_P \pm \alpha * b \end{cases} \tag{4}$$

with $\alpha \in \mathbb{R}^{+*}$ and :

$$(x_{P'} - x_O)^2 + (y_{P'} - y_O)^2 = R_C^2 \tag{5}$$

By replacing $(x_{P'}, y_{P'})$ by Eq. 4, (x_O, y_O) by Eq. 3, we obtain :

$$((x_P \pm \alpha * a) - (x_M \pm a * R_C))^2 + ((y_P \pm \alpha * b) - (y_M \pm b * R_C))^2 = R_C^2 \tag{6}$$

By developing the equation and using the following properties:

- $a^2 + b^2 = 1$
- $x_P^2 + y_P^2 + x_M^2 + y_M^2 - 2*x_P*x_M - 2*y_P*x_M = \left\|\overrightarrow{MP}\right\|^2$
- $P \in Lt$ and $M \in Lt$ thus, $a(x_M - x_P) + b(y_M - y_P) = 0$.

we can reduce Eq. 5 into : $\alpha^2 - 2*\alpha*R_C + \left\|\overrightarrow{MP}\right\|^2 = 0$. Moreover, by replacing R_C by $(\tau * w_t)$ we have :

$$\alpha^2 - 2*\alpha*\tau*w_t + \left\|\overrightarrow{MP}\right\|^2 = 0 \tag{7}$$

Through this equation, we have a model to bend our fiber median through the use of two parameters (τ and α) and the ribbon data. We also note that Eq. 7 has a solution if $\tau^2 * w^2 \geq \left\|\overrightarrow{MP}\right\|^2$.

Indeed, from τ, w and $\left\|\overrightarrow{MP}\right\| \in \mathbb{R}^+$, we can define $\tau_{min} = \frac{\left\|\overrightarrow{MP}\right\|}{w}$ and therefore $\alpha_{\tau=\tau_{min}} = \left\|\overrightarrow{MP}\right\|$. A value of $\alpha_{\tau=\tau_{min}}$ leads to a circular arc of an angle close to $\pi/2$ that can result in a point P' out of the lead ribbon. In practice, we take α in the middle of the interval defined previously and deduce τ from Eq. 7 with the relation :

$$\tau = \frac{\alpha^2 + \left\|\overrightarrow{MP}\right\|^2}{2*w*\alpha} \tag{8}$$

Therefore, with circle C known, we can compute the intersections of the limit line (in dashed orange in Fig. 13) with this circle. This corresponds to the point (M') on which we need to stop the curve, which initially started from M. In other words, we bend the corresponding ribbon border until reaching M'. If the latter is out of the domain, we cut the part out of the domain, otherwise, we successively copy and paste this curve and a reversed curve (obtained by a mirror rotation) until reaching the output domain borders. We apply this bending process to all the texture borders.

3 User Study

This section completes Sec. 6.3 with screenshots of our online study as well as some results from users during drawing sessions.

3.1 Drawing Session

During the drawing session, a user was asked to complete a provided sketch in an extended domain, delimited by a black square. The ratio between the input sketch and the output domain was 1 : 2. For this interactive session, we ordered the tasks in increasing order of complexity. Using a toolbox composed of a pencil and an eraser, the user had no limit of time to achieve the tasks. Fig. 1-4 present examples provided for the tasks and some results from the users. In addition, most of the examples were created with a mouse by non-artists.

For the first example, isotropic distributions were presented. The task was to preserve the bounded shapes during their replication. The idea was mainly to familiarize the user with the interface and the main goal. Fig. 1 presents on the left, the provided input, and on the right, four results from users.

For the second task, we first introduced the notion of unbounded shapes without giving any indication to the user. We expected users to extend and replicate the unbounded shapes to the extended domain while avoiding overlaps both inside a group and with another group. As depicted in Fig. 2, users globally respected the anisotropic distributions, while some decided to create loops (see the bottom right result).

We completed the previous example's set of tasks by adding a singularity with two vertical lines. We supposed that users would extend all the curves but only replicate the horizontal ones. From all the results and as presented in the sample in Fig. 3, most of the outputs consist in just extending the bounded strokes to the output domain. This task has not been considered for our validation.

Finally, we ended this drawing session with an input composed of bounded shapes aligned in two directions. The objective was to check whether users would preserve the anisotropic distributions. From the results in Fig. 4, users perceived different patterns regarding the anisotropic distributions of these shapes. Considering the limited space available, users usually favored either one shape or one direction over the other.

3.2 Selection Session

After letting users complete a provided input, they were asked to choose one or several outputs (depending on the task) from a provided 2D input. The goal was to pick the ones they thought to be closest to the example. We believed that it was important for them to see at least two alternative options, no matter what they had drawn before. The value appearing under each choice shows the proportion of users who picks this choice. For the last task, as users could pick all 3D immersion choices, we affected either 1.0 if the choice was unique, $\frac{1}{2}$ if two choices were picked, and $\frac{1}{3}$ if all the solutions were considered correct.

For the first two examples illustrated in Fig. 5, the user was asked to pick one result between two alternative methods, or none of them. We focused these examples on unbounded shapes and on our two less-common design guidelines, which are the non-repetition of singularity, and no overlaps of unbounded shapes in the output if they are not present in the input. The first input showed two groups of strokes that are getting closer to the end of the input space. The choice was between no overlap avoidance, a solution to overlap avoidance or none of these outputs. Depending on the user choice at this step, we adapted the outputs for the second example to only focus on the replication of singular elements. The values underneath the repetitiveness task represent the proportion of users who picked no repetition on singular elements for the case of with and without collision.

For this second set of comparisons, we let the user pick both outputs if they were considered perceptually equal.

3.3 Results

To validate H1 (the fact that all alignments and groupings are intentional), we asked users to draw the extended textures they imagined around a series of four inputs, two for bounded shapes (each drawn using several strokes) and two for unbounded fiber-like shapes. Among users, 100% preserved the stroke clustering into shapes in their sketch, and all of them but one (97%) preserved the grouping of fiber-like shapes. Moreover, 76% respected the anisotropy directions of bounded shapes in their drawings. Moreover, the alternative results of the second part of the study, which did not preserve stroke clustering were never selected as correct.

H2 (the hypothesis about the explicitness of repetitiveness) was also validated by most users. We checked whether users duplicated the repetitive patterns in the input exemplar when manually extending the distribution to the extended output. This was always the case for the bounded strokes, while 63% (see Fig. 3 b) bottom) elongated but did not replicate the unbounded shapes (probably indicating that the notion of texture was well understood for beginners). In the second part of the user study, we provided an example with a non-repeated elongated fiber, among repeated ones with another orientation. 56.7% preferred the output where the isolated fiber was not duplicated, against 40% for repetition, and 3.33% neutral.

H3 (avoiding overlaps, when not present in the input) was validated by checking the users' drawings, with a global result of 73% or overlap-free drawings. We also noted that 91.4% maintained groupings and avoided overlaps between unbounded strokes when they shared the same anisotropy direction, and the space between groups seemed constant, whereas only 76.6% achieved it when the input was irregular. During the second stage of the user study, we also asked users to choose between the output from our method with or without the curvature step (with undesired overlaps in the second case). 54.3% chose the overlap-free output, 31.4% chose straighter lines with more overlaps, and 14% were neutral.

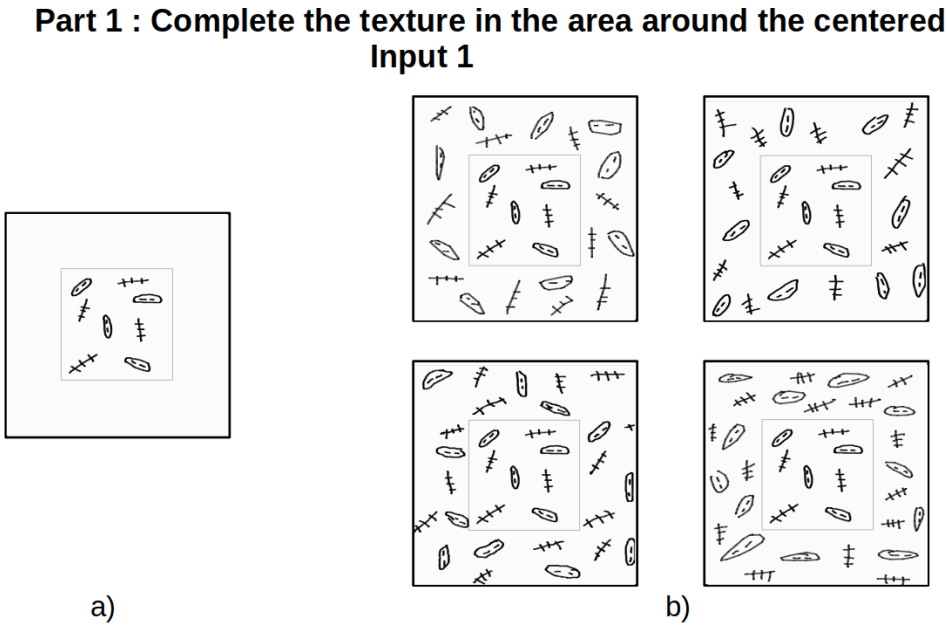

Figure 1: Replication of bounded shapes: a) Provided input with empty space around it; b) Set of four outputs taken from drawings by users.

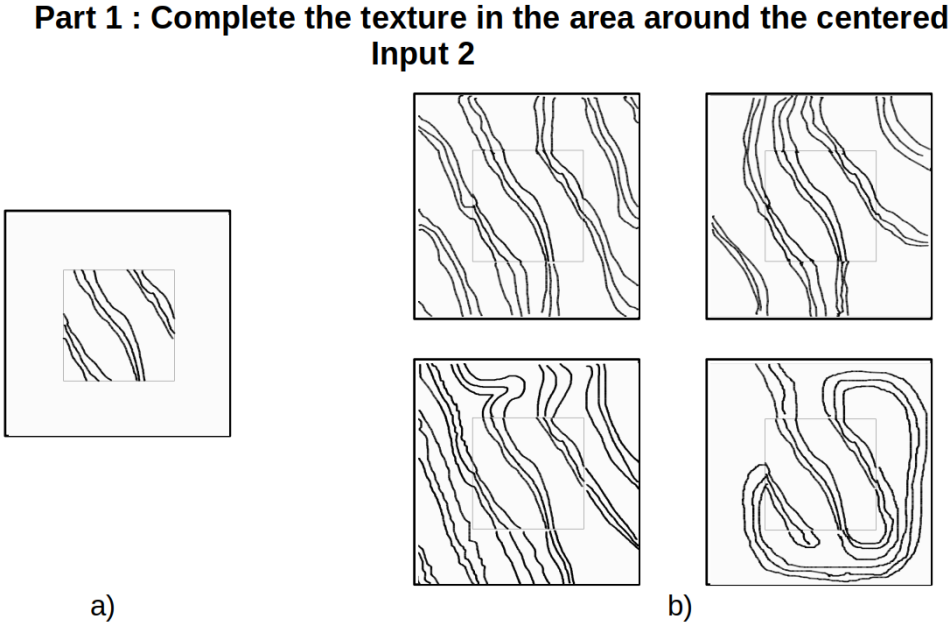

Figure 2: Unbounded shapes extension and replication without collision: a) Provided input with empty space around it; b) Set of four outputs taken from drawings by users.

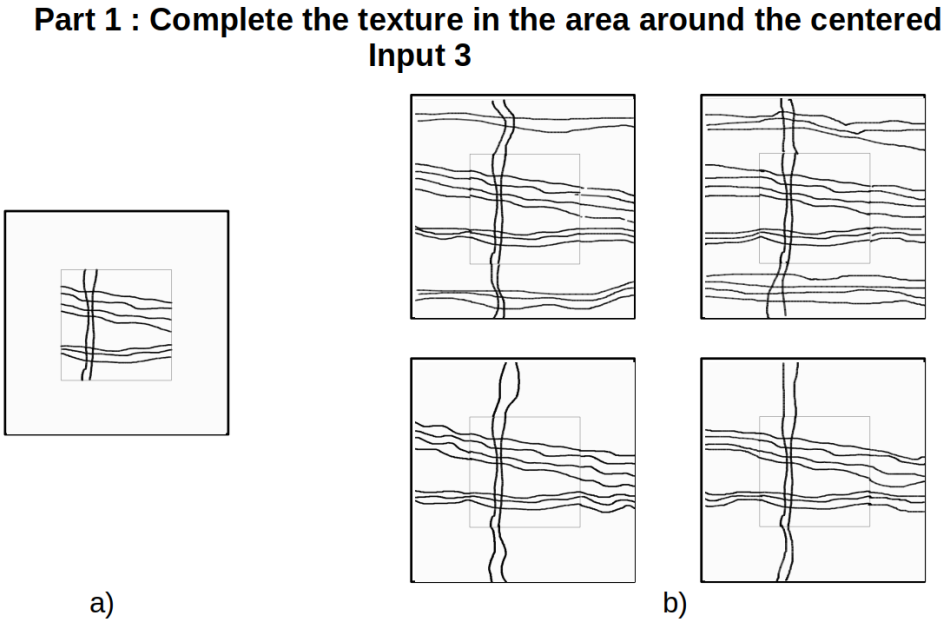

Figure 3: No replication of singularity and no overlap for unbounded shapes: a) Provided input with empty space around it; b) Set of four outputs taken from drawings by users.

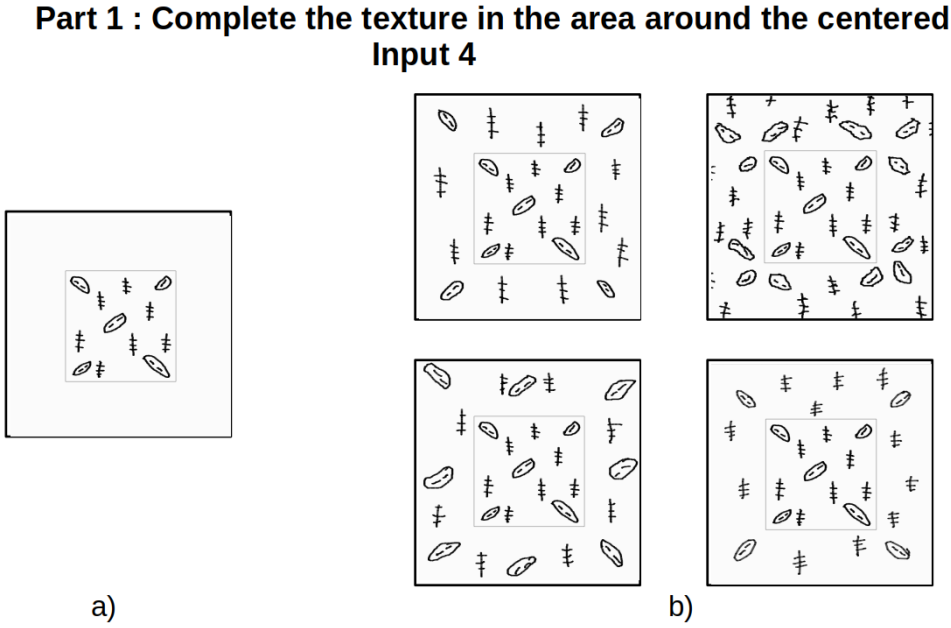

Figure 4: Bounded shapes alignments: a) Provided input with empty space around it; b) Set of four outputs taken from drawings by users

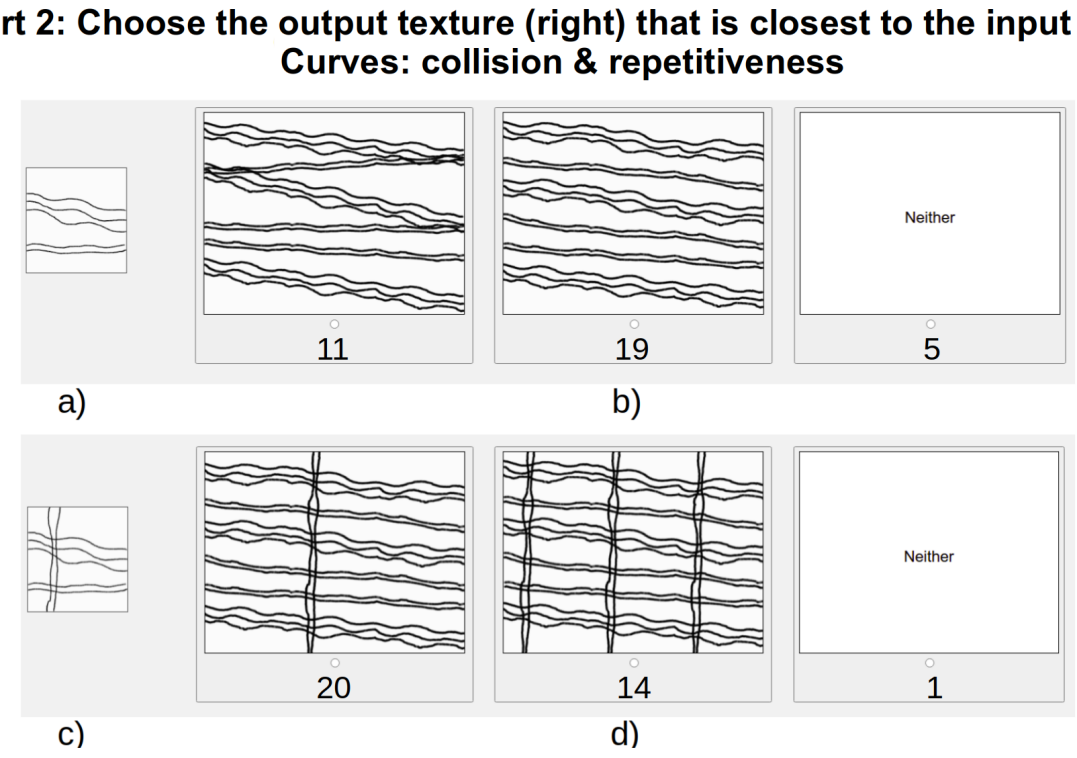

Figure 5: Comparison between overlaps or not: a) input, b) choice. Comparison between repetitiveness or not: c) input d) choice.

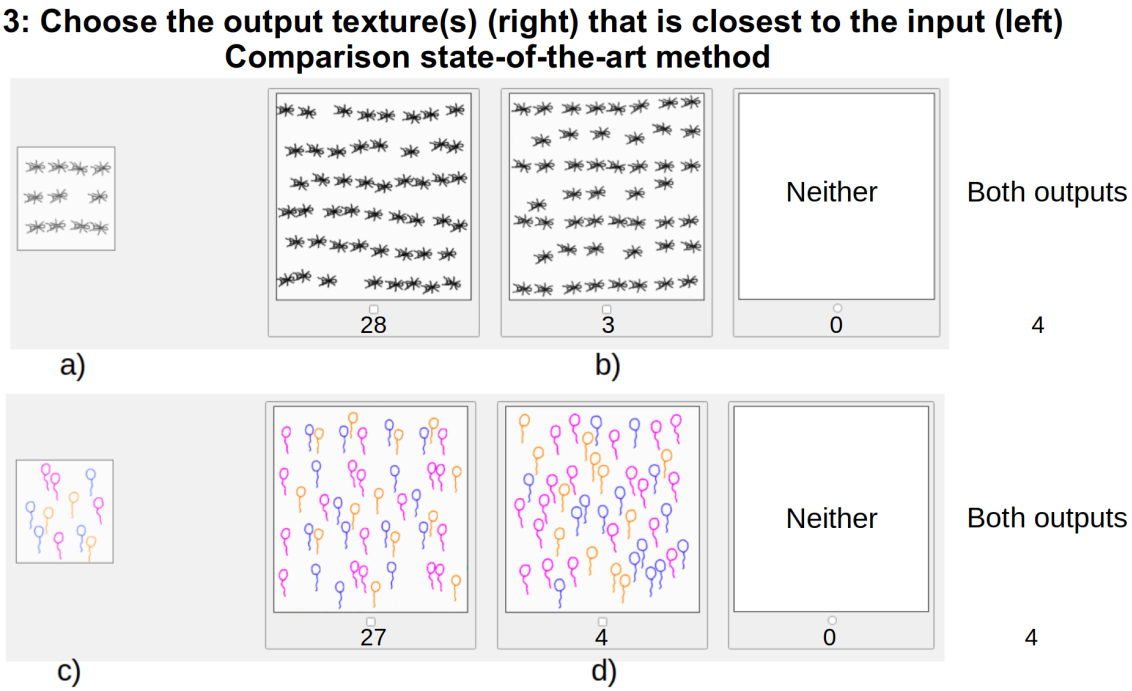

Figure 6: Comparison with the best state-of-the-art method Landes et al.[8].