Construction and Fabrication of Reversible Shape Transforms Supplementary Material

SHUHUA LI, Dalian University of Technology and Simon Fraser University
ALI MAHDAVI-AMIRI, Simon Fraser University
RUIZHEN HU*, Shenzhen University
HAN LIU, Simon Fraser University and Carleton University
CHANGQING ZOU, University of Maryland, College Park
OLIVER VAN KAICK, Carleton University
XIUPING LIU, Dalian University of Technology
HUI HUANG*, Shenzhen University
HAO ZHANG, Simon Fraser University

CCS Concepts: • **Computing methodologies → Shape analysis**;

ACM Reference Format:
Shuhua Li, Ali Mahdavi-Amiri, Ruizhen Hu, Han Liu, Changqing Zou, Oliver van Kaick, Xiuping Liu, Hui Huang, and Hao Zhang. 2018. Construction and Fabrication of Reversible Shape Transforms Supplementary Material. *ACM Trans. Graph.* 37, 6, Article 190 (November 2018), 13 pages. https://doi.org/10.1145/3272127.3275061

In this supplementary material, we describe more details and offer more results for our algorithms.

1 LENGTH OF CONGRUENT SEGMENTS

To compute point-level congruency score S_c (Section 4.1), we simplify shape P and obtain $\hat{P}$ to avoid progressively growing two equal-length segments on many points on P. We then classify points on $\hat{P}$ into seven classes for computing the length of two congruent segments noted as $L(C_l^{\hat{p}})$ and $L(C_r^{\hat{p}})$ (Figure 1 (Bottom)). Points in the first four classes most likely have congruent segments while the points in the other three classes usually do not. For these three classes, we still use curve similarities to assign a (more likely) low congruency score.

(a) If two adjacent feature points $\hat{p}_{l1}$ and $\hat{p}_{r1}$ of $\hat{p}$ are convex points, then $L(C_l^{\hat{p}}) = L(C_r^{\hat{p}}) = min(d(\hat{p}, \hat{p}_{l1}), d(\hat{p}, \hat{p}_{r1}))$;

(b) For concave feature point $\hat{p}_{l1}$ and convex feature point $\hat{p}_{r1}$,
   (1) if $d(\hat{p}, \hat{p}_{l1}) \approx d(\hat{p}, \hat{p}_{r1})$ and $\angle \hat{p}_{l2}\hat{p}_{l1}\hat{p} \approx \angle \hat{p}_{r2}\hat{p}_{r1}\hat{p}$, then $L(C_l^{\hat{p}}) = L(C_r^{\hat{p}}) = min(d(\hat{p}, \hat{p}_{l2}), d(\hat{p}, \hat{p}_{r2}))$.
   (2) if $d(\hat{p}, \hat{p}_{l1}) \approx d(\hat{p}, \hat{p}_{r1})$ and $\angle \hat{p}_{l2}\hat{p}_{l1}\hat{p} \gg \angle \hat{p}_{r2}\hat{p}_{r1}\hat{p}$, then $L(C_l^{\hat{p}}) = L(C_r^{\hat{p}}) = min(d(\hat{p}, \hat{p}_{l1}), d(\hat{p}, \hat{p}_{r1}))$.
   (3) if $d(\hat{p}, \hat{p}_{l1}) \gg d(\hat{p}, \hat{p}_{r1})$, then $L(C_l^{\hat{p}}) = L(C_r^{\hat{p}}) = d(\hat{p}, \hat{p}_{r1})$.
   (4) if $d(\hat{p}, \hat{p}_{l1}) \approx d(\hat{p}, \hat{p}_{r1})$ and $\angle \hat{p}_{l2}\hat{p}_{l1}\hat{p} \ll \angle \hat{p}_{r2}\hat{p}_{r1}\hat{p}$, then $L(C_l^{\hat{p}}) = L(C_r^{\hat{p}}) = min(d(\hat{p}, \hat{p}_{l2}), d(\hat{p}, \hat{p}_{r2}))$.
   (5) if $d(\hat{p}, \hat{p}_{l1}) \ll d(\hat{p}, \hat{p}_{r1})$, then $L(C_l^{\hat{p}}) = L(C_r^{\hat{p}}) = min(d(\hat{p}, \hat{p}_{l2}), d(\hat{p}, \hat{p}_{r1}))$.

(c) If two adjacent points $\hat{p}_{l1}$ and $\hat{p}_{r1}$ of $\hat{p}$ are concave points, then $L(C_l^{\hat{p}}) = L(C_r^{\hat{p}}) = min(d(\hat{p}, \hat{p}_{l2}), d(\hat{p}, \hat{p}_{r2}))$;

(d) For convex feature point $\hat{p}_{l1}$ and concave feature point $\hat{p}_{r1}$, there are also five sub-cases, which are similar to those in $(b)((d1) \rightarrow (b1), (d2) \rightarrow (b4), (d3) \rightarrow (b5), (d4) \rightarrow (b2)$, and $(d5) \rightarrow (b3))$.

Here, thresholds for length and angle approximation are $\frac{L^c}{20}$ and $\frac{\pi}{6}$.

2 CONJUGATE TRUNKS ADJUSTMENT

As discussed in Section 4.3, trunks T and T' attaining the highest CRS score for shapes P and Q are not necessarily conjugate. Therefore, they need to be adjusted to obtain conjugate (T_P, T_Q). Figure 2 shows the adjustments for two shape pairs. To do so, T_P and T_Q (polygons in solid lines) attain average edge lengths of T and T' (polygons in dashed lines). With n edges for T_P, a global optimization method aims to make each angle of T_P close to its corresponding angle in T and minimize the total angle difference. Here we adopt a local method as T is very close to T_P. We randomly make $n-3$ consecutive angles of T_P to be the same as those in T. Fixing $n-3$ angles and the edge lengths, the remaining three angles can be precisely calculated since there is no degree of freedom. As the edges and angles of T_P are slightly modified, the vertices of T_P are not necessarily on P (contours in dashed lines). To remove this artifact, we scale and translate exterior pieces of P into T_P and obtain a slightly different shape $\tilde{P}$ (contours in solid lines). The process for T_Q and Q is also the same. Note that T and T' are usually very

*Joint corresponding authors: ruizhen.hu@gmail.com, hhzhiyan@gmail.com

Authors' addresses: Shuhua Li, School of Mathematical Sciences, Dalian University of Technology, School of Computing Science, Simon Fraser University, sue142857@gmail.com; Ali Mahdavi-Amiri, Simon Fraser University; Ruizhen Hu, Shenzhen University; Han Liu, Simon Fraser University, Carleton University; Changqing Zou, University of Maryland, College Park; Oliver van Kaick, Carleton University; Xiuping Liu, Dalian University of Technology; Hui Huang, Shenzhen University; Hao Zhang, Simon Fraser University.

0730-0301/2018/11-ART190 $15.00
https://doi.org/10.1145/3272127.3275061

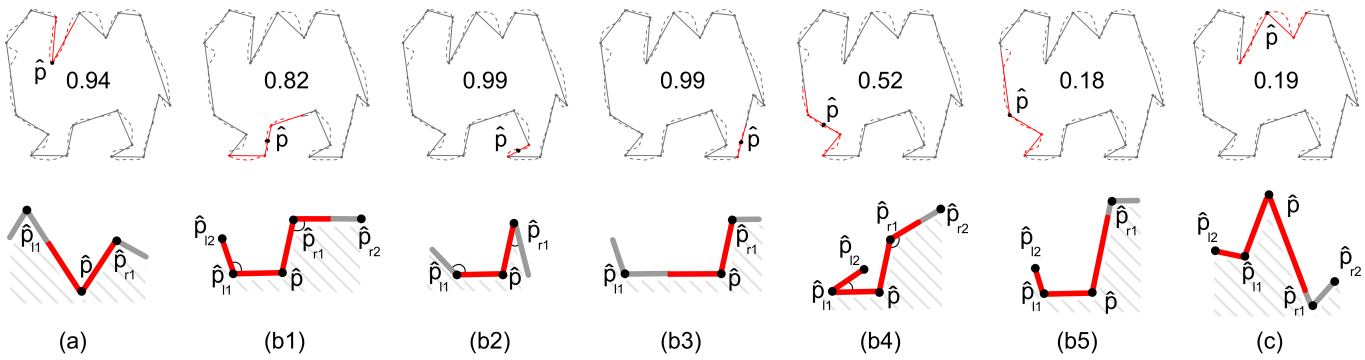

Fig. 1. (Top) We compute $\{C_l^{\hat{p}}, C_r^{\hat{p}}\}$ (red solid segments) for $\hat{p}$ (black point) on simplified shape $\hat{P}$ (solid line) and their mapping on P (dashed line) is $\{C_l^{p}, C_r^{p}\}$ (red dashed segments). The curve similarity of $\{C_l^{p}, C_r^{p}\}$ is our congruency score (number). (Bottom) Seven classes of points on $\hat{P}$ for computing $\{C_l^{\hat{p}}, C_r^{\hat{p}}\}$.

close to T_P and T_Q and therefore the deformation of shapes P and Q resulting from this process is minor.

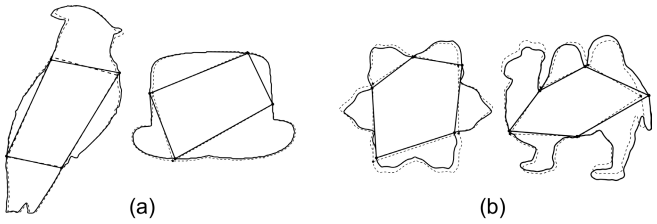

Fig. 2. Conjugate trunks adjustment for bird and hat (a), flower and camel (b). Input contours P and Q, and their best candidate trunk pair T and T' are in dashed lines; Conjugate trunks T_P and T_Q after adjustments and slightly deformed contours are in solid lines.

3 THE EFFICIENT COMPUTATION OF THE AREA REVERSIBILITY SCORE

In Section 4.2, we define the cross-reversibility score (CRS) for any trunk pair (T, T'). The area reversibility term of the score (Equation 5) requires efficiently computing the area of gaps and overlaps among shape pieces inside trunks, and the area of regions outside trunks. Here, we only discuss the computation of these areas for T as the process for T' is similar.

Given an edge correspondence between T and T', we adjust them to be exactly conjugate (refer to the previous section 'Conjugate Trunks Adjustment'). Trunk T with n vertices divides the boundary of shape P into n curves, which are denoted as $\{C_1, C_2, ..., C_n\}$. Then, n curves are rotated and translated into T' according to the edge correspondences. The transformed curves are supposed to fit in T' with minor gaps, overlaps and regions outside T'. Here, we also use $\{C_1, C_2, ..., C_n\}$ for the pieces enclosed by these transformed curves. We convert these pieces into binary masks and compute those areas by Boolean operation.

Polygons can be easily converted to a binary region-of-interest mask, whose pixel value attains one for polygons inside and zero otherwise. We covert $\{C_1, C_2, ..., C_n\}$ and T' into masks $\{M_{C_1}, M_{C_2}, ..., M_{C_n}\}$ and $M_{T'}$, respectively. The masks for gaps, overlaps and regions outside T' are generated by Boolean operation as follow:

$$M_{union} = M_{C_1} \bigcup M_{C_2} \bigcup ... \bigcup M_{C_n},$$
$$M_{out} = M_{union} \bigcap (\sim M_{T'}),$$
$$M_{gap} = (\sim M_{union}) \bigcap M_{T'},$$
$$M_{unionInside} = M_{union} \bigcap M_{T'},$$

The areas of objects in the binary image M_{out} and M_{gap} are the areas of regions outside T' and gaps, respectively. The area of overlaps is defined as $\sum_{i=1}^{i=n} area(M_{C_i} \bigcap M_{T'}) - area(M_{unionInside})$. In practice, we enlarge $\{C_1, C_2, ..., C_n\}$ and T' by the same scale to ensure the resolution of masks, as shape P has a unit area. Area errors are reduced by the same scale later. All masks are ensured to be in the same size.

4 GAP ELIMINATION

When conjugate trunks are created and curves $\{C_1, \ldots, C_n\}$ are obtained by transferring curves along the boundary of P into T_Q, some gaps and overlaps may exist. In Section 4.3, we discussed how to remove overlaps. Here, gap elimination is described in details. We use Laplacian editing method that deforms two curves into opposite directions to reduce their gaps. However, the handle anchor a_i is outside C_j and a_j is outside C_i (see Figure 8e in manuscript). In each iteration, we use the farthest handle anchor pair among all curves instead of two curves. If the deformation causes new overlaps, we bisect the step size until no overlaps will be generated. We iterate until the distance between two farthest handle anchors is less than 0.01. We set points of C_i, which are close (less than 0.01 distance) to other curves, as static anchors when deforming C_i. We further remove minor gaps by automatic curve merging. We define the distances between a point p and a curve C, and between two curves C_1 and C_2 as the following:

$$d(p, C) = \min_{c \in C} d(p, c), d(C_1, C_2) = \min_{c_1 \in C_1} d(c_1, C_2).$$

A junction is a meeting point of at least three dissection curves. We compute candidate junctions on each curve C_i to divide the curve into segments for merging (blue points in Figure 8f in manuscript).

Fig. 3. Triangular mesh and mesh texture of input shape (left) and output shape (right).

For any two curves C_j and C_l, the point on C_i, which minimizes $d(p, C_j) + d(p, C_l)$ and makes the distance smaller than 0.05, is a candidate junction of C_i and denoted as p^i_{jl}. Two candidate junctions p^i_{jl} and $p^i_{j'l'}$ with distance smaller than 0.05 are merged to form $p^i_{jlj'l'}$. We merge a segment from one curve to the closest segment from other curves. Since two closest segments are very similar, we use the one with smaller index as the merged segment. Eventually, tiny gaps are removed and segments are smoothed slightly.

Curve self-intersection is prohibitive in both overlap elimination and gap elimination. But, minor folded curve segments may still occur, which will be removed in the final curve smoothing. The boundary deformation always terminates without any overlaps since no more overlaps are introduced during gap elimination. In most cases, gaps were reduced below the distance threshold for automatic curve merging and then completely eliminated. In a few cases, the Laplacian editing method terminated with gaps larger than the threshold and we had to manually divide the gap. The failure comes from the lack of sampling points on curves. We will introduce adaptive sampling into our algorithm. Note that in all results which are claimed to be generated automatically, the gaps are completely eliminated automatically.

5 AUTOMATIC TEXTURE TRANSFER

When the input shapes already have textures, it is desired to transfer the available texture to their deformed RIOT pair. To do so, we triangulate input shape P and obtain texture coordinates for vertices (Figure 3 (Left)). Then, we replace the boundary vertices with those from deformed shape $\tilde{P}$ and smooth interior vertices via Laplacian to obtain deformed mesh and its associated texture (Figure 3 (Right)). Having the texture of the deformed mesh, we can take the texture of each piece by triangulating pieces and sampling the texture.

6 COMPARISONS WITH MANUAL DESIGNS

Here, we provide some additional comparisons with manual results created by Prof. Jin Akiyama. As apparent in Figure 4, our method automatically produces almost the same RIOT solutions (right) in comparison with manual design (left).

7 STATISTICS AND TIMING FOR THE GALLERY

For a more concrete picture of statistics and timing, we have provided Table 1 for the 12 reversible shape pairs in the gallery (see Figure 13 in the manuscript). The average time to compute reversibility scores (RS) and a quick cross-reversibility score (QCRS) for each pair are 0.15 seconds and 2.8 seconds, respectively. The average time for candidate trunks (CT) and the (slow) cross-reversibility score (CRS) for each pair of shapes are respectively 20.4 and 13.0 seconds. The average time for the boundary deformation (BD) which turns

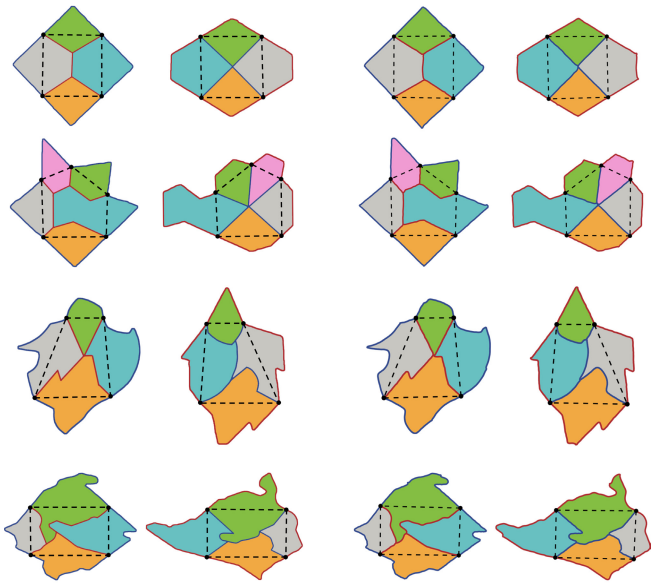

Fig. 4. Additional comparisons between manual designs (left) and our results (right) on the same input.

out to be the most time consuming step is 127.3 seconds for two shapes. The number of sample points along a shape boundary (N) and sparse sample points for diverse polygons (M) are 202 and 35 in average. In the table, we use the larger N and M of two shapes in a pair. These statistics and timings are similar to those of the large shape collection in the manuscript. More complex boundaries have higher run-time for RS since they have more necks need to be computed while the simpler shapes like the face of Charlie Chaplin has the lowest RS run-time. For QCRS, CT and CRS, a larger set of convex points after simplification will lead to larger polygon spaces and higher run-time (e.g., the squirrel). The yellow fish and hare (paired with tortoise) are the opposite cases. Lower run-time for CRS in the flower and camel pair is due to the small number of candidate trunks. Lower cross-reversibility score means that more boundary deformations needed (e.g., the Monkey King and the banana); example for the opposite case is the ambulance and the bell. Longer boundaries with more features lead to more contour points (N) and larger point space (M) for candidate vertices due to adaptive sampling (Monkey King); the ambulance and the bell is the opposite case.

8 ADDITIONAL FABRICATED MODELS

We have fabricated some of the pairs presented in the paper. Figure 5 illustrates maple leaf to beaver and Figure 6 shows a RIOT between butterfly and fish.

9 USER STUDY

We conducted a user study to assess human capabilities to decide whether an approximate RIOT exists for a shape pair. In this study, participants learn five RIOT examples provided by Jin Akiyama (three simple ones from Figure 4 and two more complex ones from

Table 1. Timing and statistics on the gallery results.

	RS(s)	QCRS(s)	CT(s)	CRS(s)	BD(s)	N	M
	0.11	2.7	25.1	10.3	**67.6**	**166**	28
	0.15	**0.8**	15.7	17.8	103.6	186	35
	**0.09**	1.4	18.6	14.2	72.0	168	**26**
	0.17	6.6	14.9	**2.6**	99.5	225	43
	0.14	1.1	16.1	22.7	126.5	208	33
	**0.20**	3.9	20.7	10.3	114.5	218	37
	0.14	1.6	17.0	7.6	126.7	172	33
	0.16	1.8	20.7	2.9	101.1	222	35
	0.18	2.4	25.8	9.2	**228.7**	**236**	**47**
	0.15	**8.5**	**35.6**	**41.3**	209.6	216	38
	0.15	1.6	20.9	9.2	175.2	222	36
	0.13	**0.8**	**13.5**	8.2	102.8	186	32
Avg	0.15	2.8	20.4	13.0	127.3	202	35
Max	0.20	8.5	35.6	41.3	228.7	236	47
Min	0.09	0.8	13.5	2.6	67.6	166	26

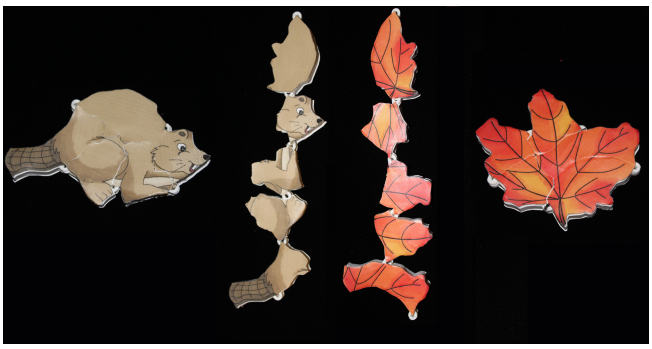

Fig. 5. Fabricated model of the RIOT between maple leaf and beaver.

Figure 14 in the manuscript) and one approximate RIOT example from the overview (Figure 3 in the manuscript). Then, they are asked the yes or no question: 'Does an approximate RIOT exist for each of the shape pairs?'. Among 16 pairs presented, the answer to eight of them was positive. These eight images were selected from the gallery (Figure 13 in the manuscript) and had cross-reversibility score (CRS) larger than 0.7. The other eight (negative ones) attained CRS lower than 0.3. All the shape pairs and visual examples can be found at this link.

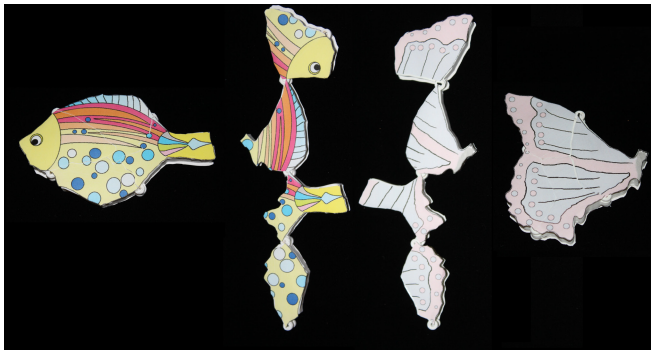

Fig. 6. Fabricated model of the RIOT between butterfly and fish.

10 TOP 100 SHAPE PAIRS BASED ON QCRS

As discussed in Section 6, we use the quick cross-reversibility score (QCRS) to identify potential reversible shape pairs as inputs for RIOT construction. While the slower cross-reversibility score (CRS) in RIOT construction will provide a more accurate reversibility assessment, to evaluate QCRS against CRS, we show reversible transforms computed fully automatically by our algorithm in Figure 7 to Figure 11 for the top 100 shape pairs following the ranking of their QCRS. Note that their ranking numbers are marked in the top left corner and their QCRS and CRS are placed side by side in their subfigure. As shown in the figures, the CRS is typically large when the QCRS of a shape pair is large and our algorithm typically produces a reasonable solution for such a shape pair.

11 ADDITIONAL RESULTS

Here, we provide more results illustrating how our method can successfully generate RIOT pairs for complicated shapes. All RIOT results in figure 12 and figure 13 are generated fully automatically except the 'rooster and crown' pair and the 'crown and bird' pair, which need minor user interactions for the foot of the cock and the head of the bird. The RIOT results in the first three rows of figure 12 are from the large shape collection. Those in the other three rows and in figure 13 are from shapes that have been collected from the internet.

In order to remove the effect of manually designed textures on the overall look and quality of the final results, we also provide all results without textures in figure 14 and figure 15. These results include the teaser result in figure 1 and gallery results in figure 14 in the paper, and additional results in figure 12 and figure 13 in this supplementary.

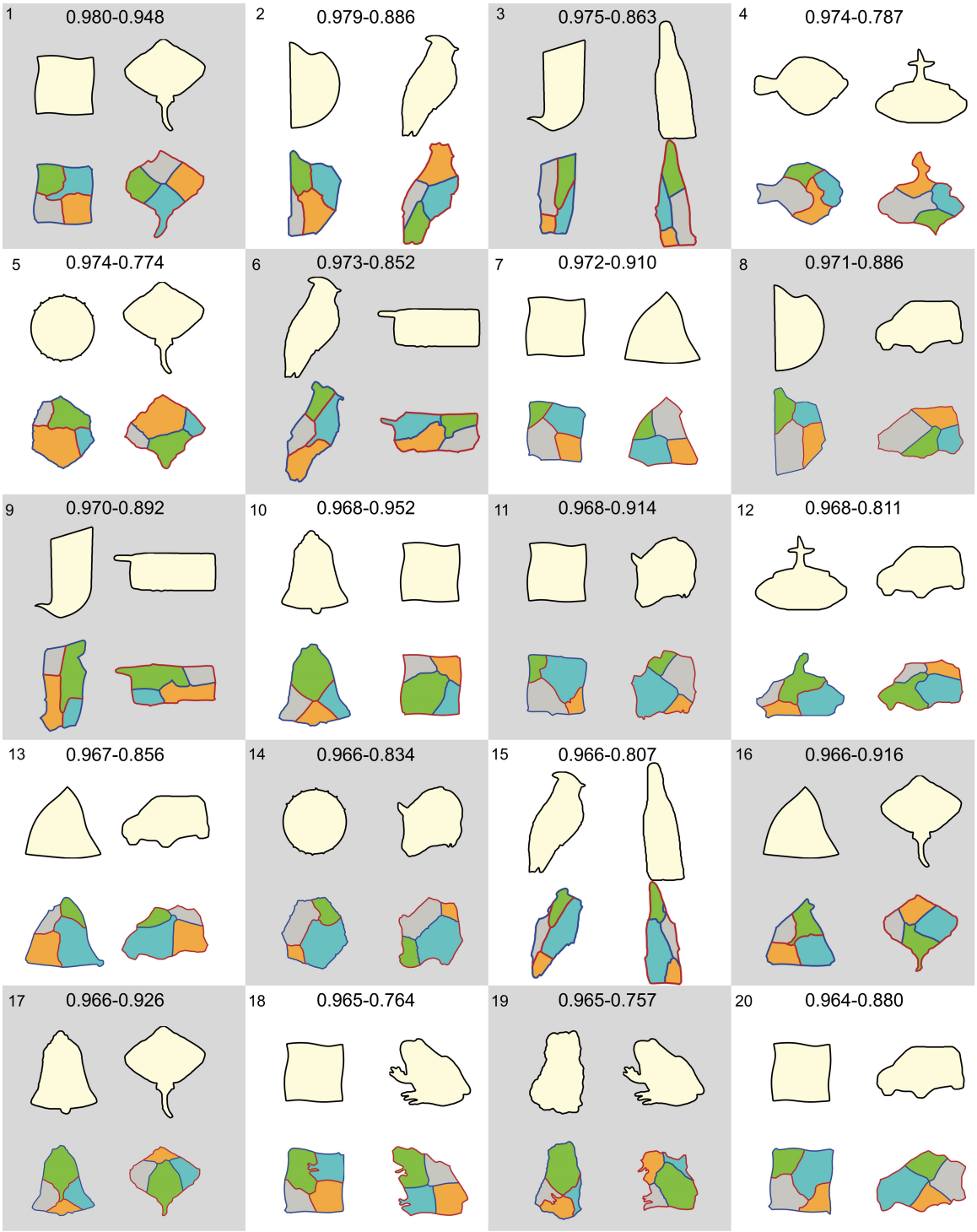

Fig. 7. Reversible transforms (bottom in each sub-figure) computed fully automatically by our algorithm for the first to the 20th shape pairs (top in each sub-figure) with respect to the ranking of their QCRS. Note that their rankings are shown in the top left corner and their QCRS and CRS are placed side by side in their sub-figure.

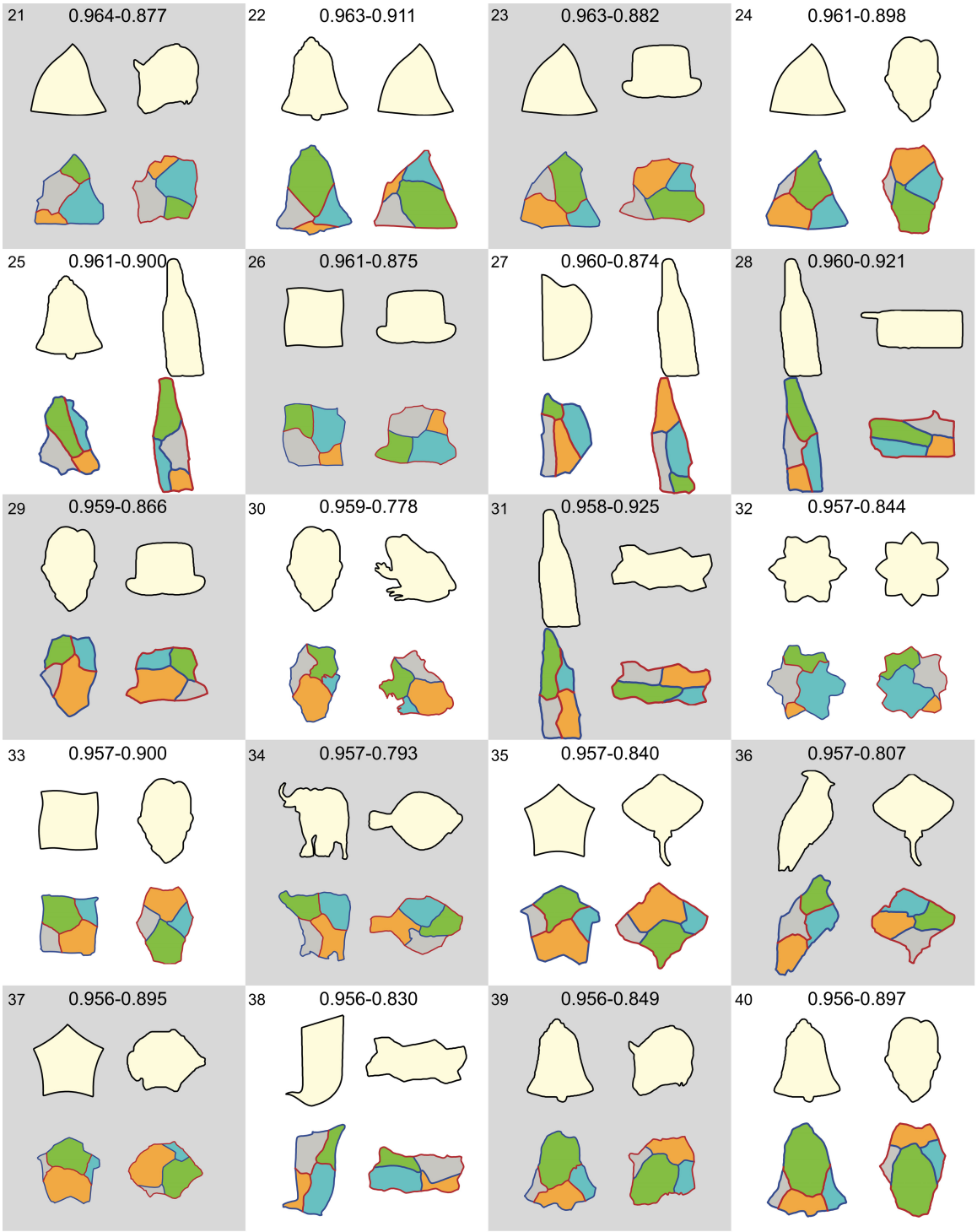

Fig. 8. Reversible transforms (bottom in each sub-figure) computed fully automatically by our algorithm for the 21st to the 40th shape pairs with respect to the ranking of their QCRS.

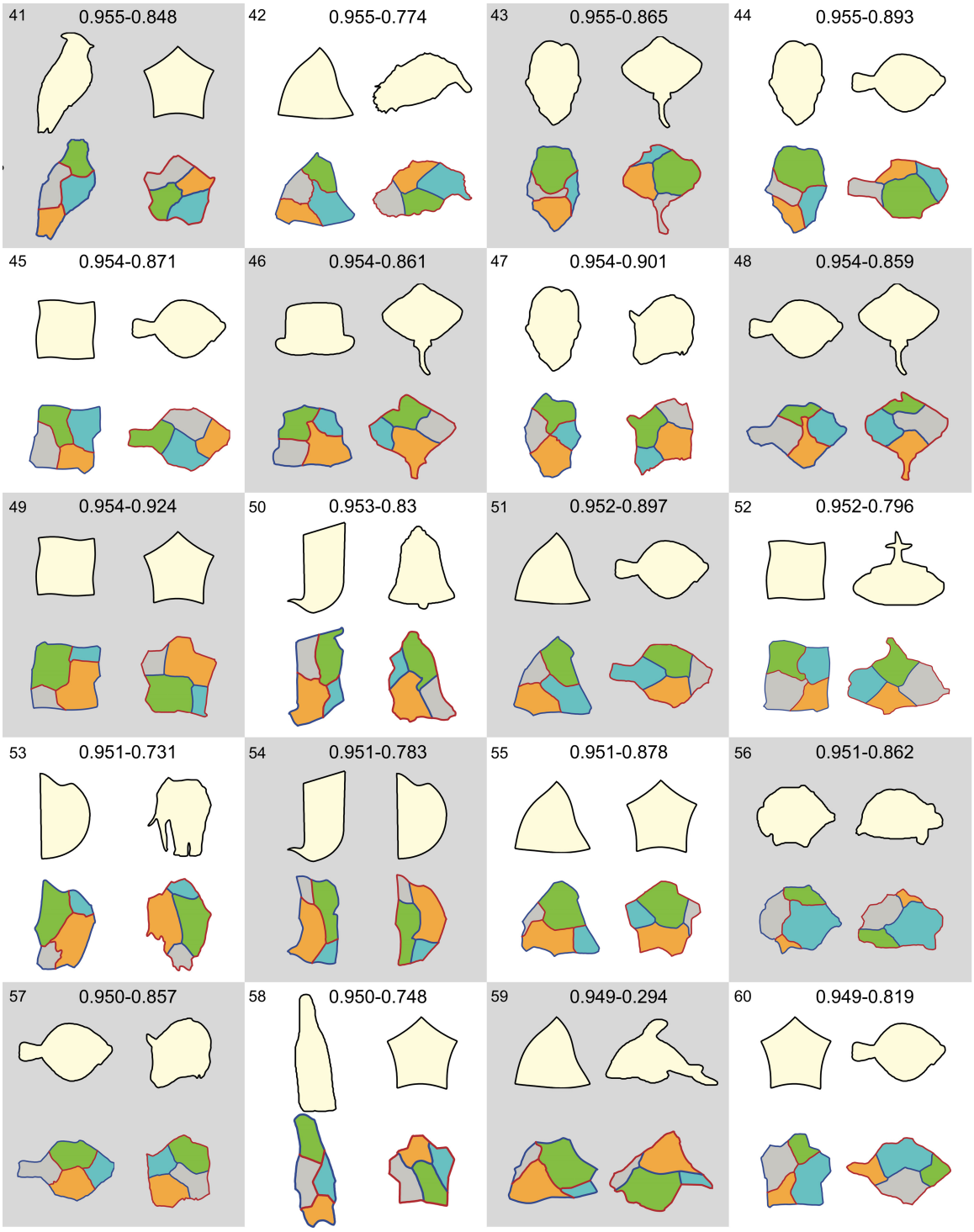

Fig. 9. Reversible transforms (bottom in each sub-figure) computed fully automatically by our algorithm for the 41st to the 60th shape pairs with respect to the ranking of their QCRS.

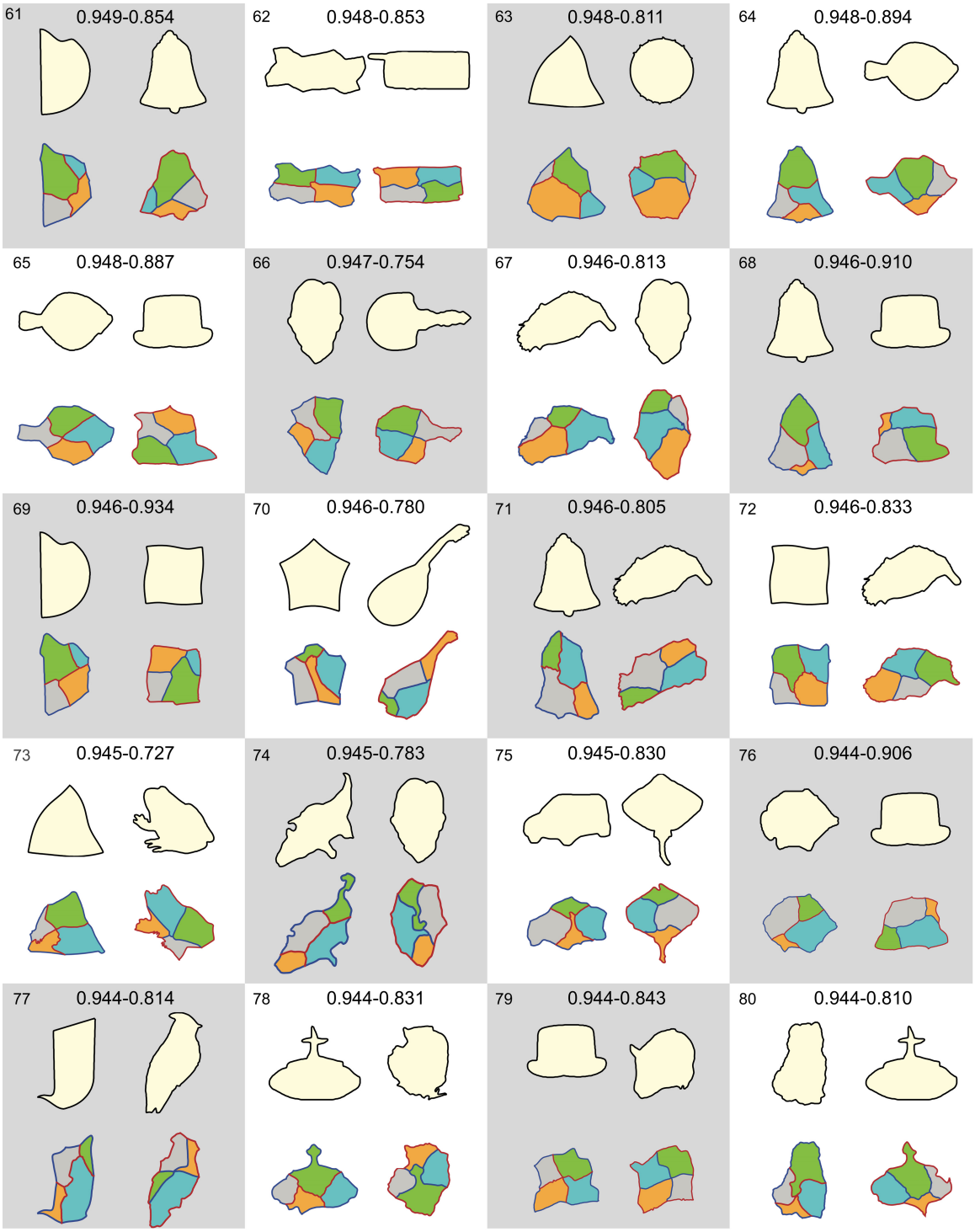

Fig. 10. Reversible transforms (bottom in each sub-figure) computed fully automatically by our algorithm for the 61st to the 80th shape pairs with respect to the ranking of their QCRS.

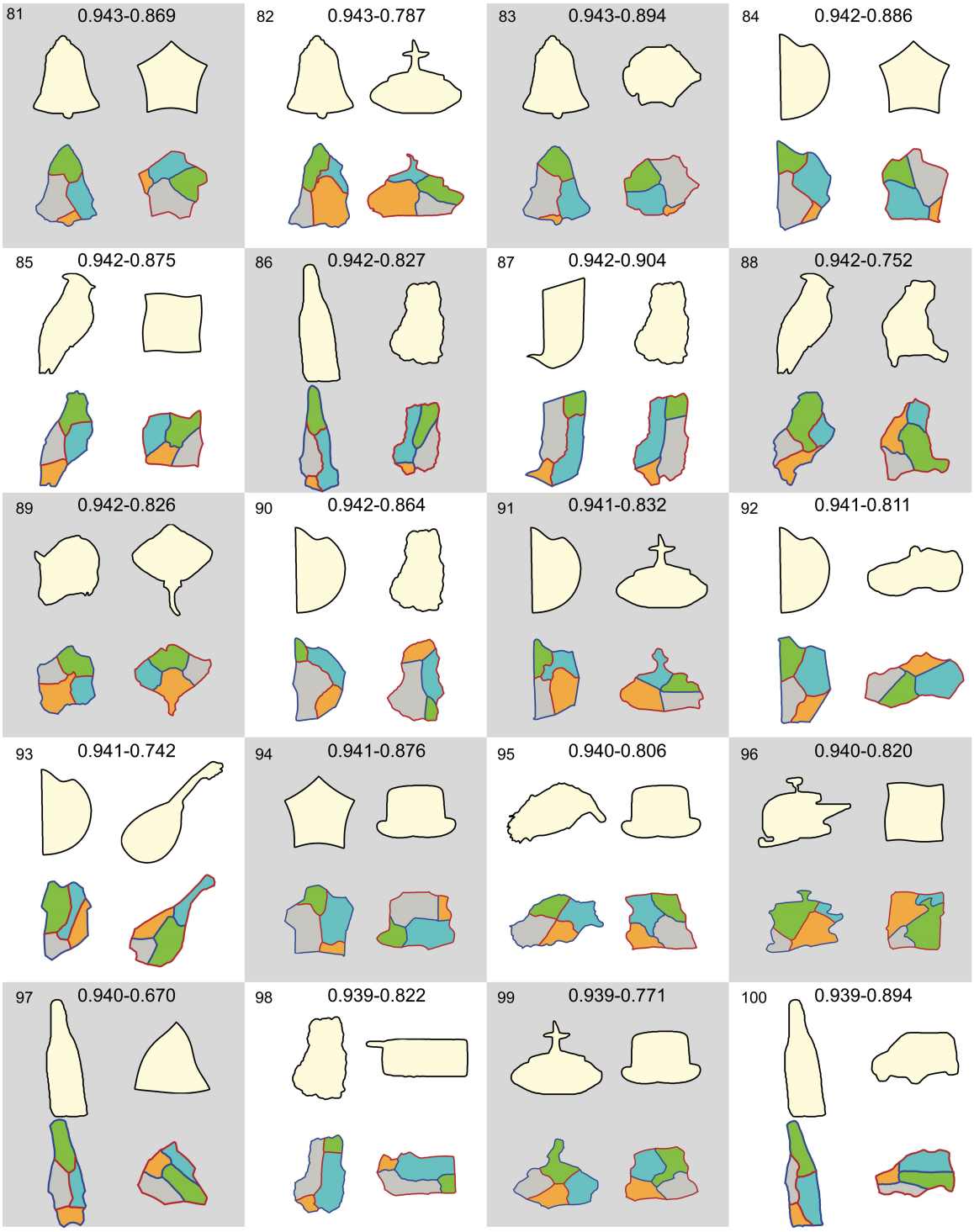

Fig. 11. Reversible transforms (bottom in each sub-figure) computed fully automatically by our algorithm for the 81st to the 100th shape pairs with respect to the ranking of their QCRS.

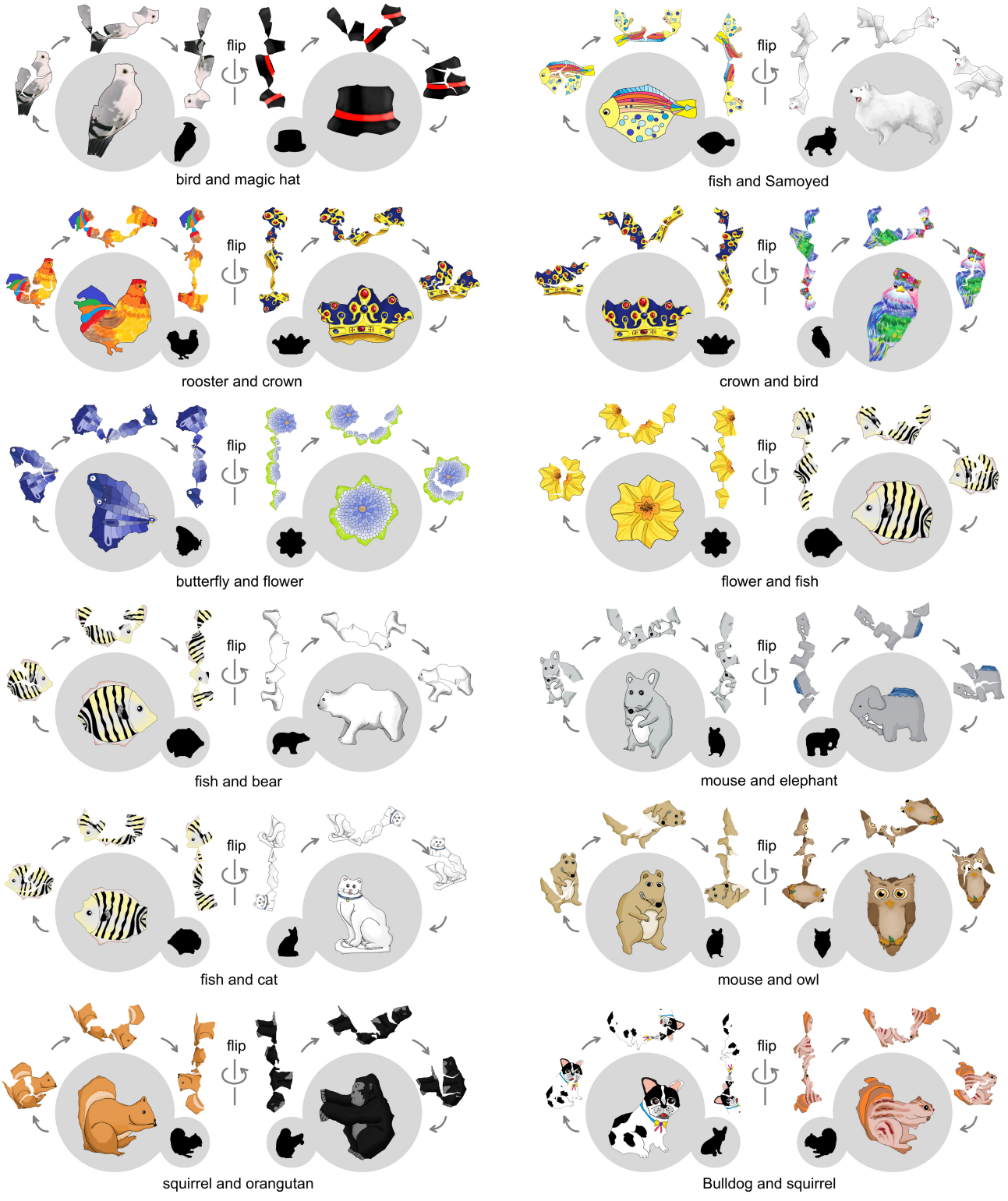

Fig. 12. Additional reversible shape transforms computed fully automatically by our algorithm (except the 'rooster and crown' pair and the 'crown and bird' pair). For each pair, we show the input shapes in silhouette images and the resulting (possibly deformed) shapes which induce a RIOT in texture. Hinged dissections are shown in a circular sequence.

Fig. 13. Additional reversible shape transforms computed fully automatically by our algorithm.

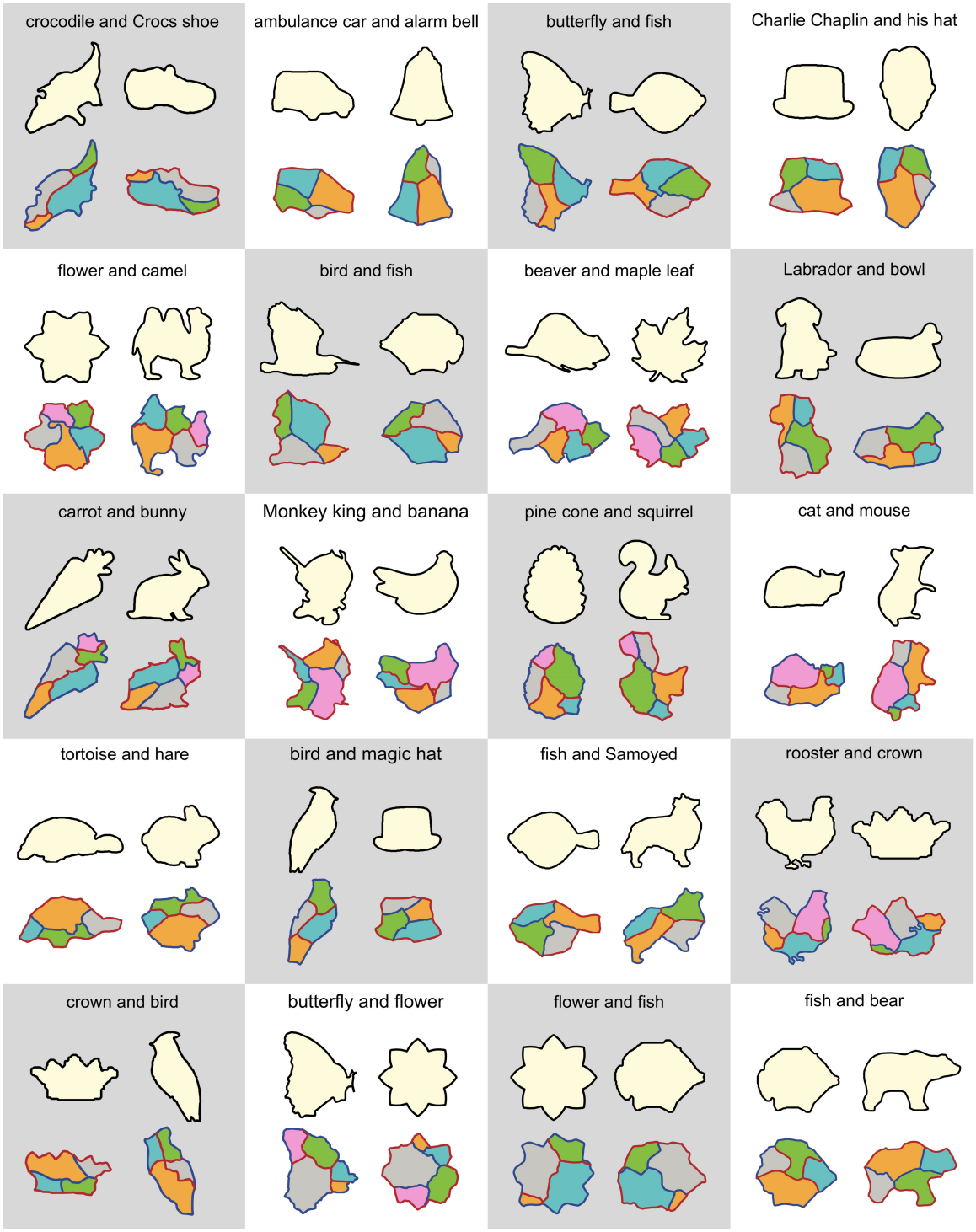

Fig. 14. Reversible shape transforms without manually designed textures (bottom in each sub-figure), which correspond to those with textures in figure 1, 14 in the paper and figure 12 in this supplementary. For each pair, we also show the input shapes in the top in each sub-figure.

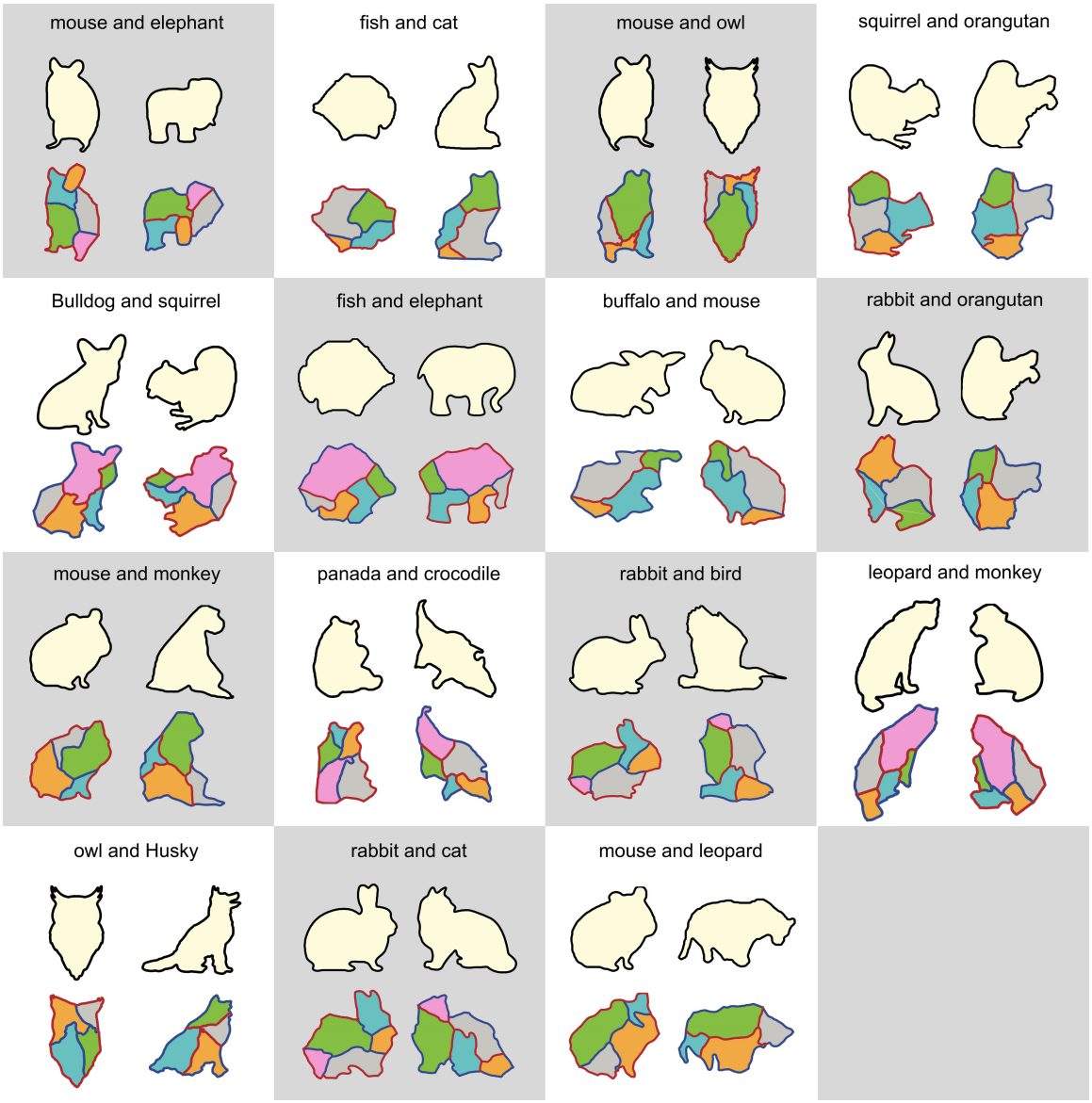

Fig. 15. Reversible shape transforms without manually designed textures (bottom in each sub-figure), which correspond to those with textures in figure 12 and figure 13 in this supplementary. For each pair, we also show the input shapes in the top in each sub-figure.