Supplementary Materials
Continuous and Orientation-preserving Correspondences via Functional Maps

JING REN, KAUST
ADRIEN POULENARD, École Polytechnique
PETER WONKA, KAUST
MAKS OVSJANIKOV, École Polytechnique

ACM Reference Format:
Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. 2018. Supplementary Materials Continuous and Orientation-preserving Correspondences via Functional Maps. *ACM Trans. Graph.* 37, 6, Article 248 (November 2018), 3 pages. https://doi.org/10.1145/3272127.3275040

Complete comparison. Tables 2–5 show the complete results of different methods with different initialization (WKS or SEG) of four datasets: 200 pairs of FAUST isometric shapes, 400 pairs of FAUST non-isometric shapes, 568 pairs of TOSCA isometric shapes, and 190 pairs of TOSCA non-isometric shapes. Fig. 2 shows an example of the maps computed from different methods and visualized via texture transfer.

Coverage. As discussed in Sec.4.2.4 Coverage, we would like to improve the coverage of the point-wise maps (and improve the bijectivity at the same time) using the following heuristic: for each uncovered vertex, we find its neighbor with the largest pre-image size, then we pick a vertex in the pre-image and re-map it to the uncovered vertex. For this selection, we consider the map in the opposite direction to improve the bijectivity at the same time.

Assume we have two point-wise maps $T_{12} : S_1 \to S_2$ and $T_{21} : S_2 \to S_1$ and we want to use the above heuristic and T_{21} to improve the coverage of T_{12} (or vice versa). There are two main issues of the this heuristic: (1) It's hard to parallelize: once we update the map T_{12} on a single vertex, the search space for the uncovered vertices on S_2 is changed and needs to be updated. Therefore, it will be time-consuming to process every uncovered vertex in a sequence, given that the coverage of the initial maps from WKS descriptors is around 20%. (2) For each uncovered vertex p, we find it's neighbor y with largest pre-image size. If $x := T_{21}(p)$ is in the pre-image set of y w.r.t. the map T_{12}, we can set $T_{12}(x) = p$. This can improve the coverage and bijectivity at the same time. However, the initial point-wise maps T_{12}, T_{21} are computed independently with poor bijectivity, and for the non-isometric pairs, the accuracy of the initial maps are also limited. Therefore, it's unlikely to have $T_{21}(p)$ lying in the pre-image set of the vertex y and we fail to find a vertex on S_1 to be mapped to the uncovered p.

Therefore, in our implementation, we process all the uncovered vertices at the same time for multiple iterations, and we do not update the search space for the vertices within each iteration (even though they should be updated). Also we relaxed the condition of having $T_{21}(p)$ to be in the pre-image set of y. Specifically, we combined a couple of steps with different complexity: (assuming we are improving the coverage of T_{12})

(1) for a vertex v on S_2 with $|T_{12}^{-1}(v)| := n > 1$, map $n-1$ vertices in its pre-image set $\{x \in S_1 \mid T_{12}(x) = v\}$ to the neighbors of v where the uncovered neighbors of v have a higher priority to get assigned if the number of neighbors of v is larger than $n - 1$. For a point-wise map reconstructed from a functional map, it's likely to have a region mapped to a single vertex, this step will spread the correspondences to its neighbors and improve the coverage.
(2) use T_{21} to improve T_{12} directly: for each of the uncovered vertex v on S_2, let $y = T_{12}\big(T_{21}(v)\big) \in S_2$, we check if $|\,T_{12}^{-1}(y)\,| > 1$, i.e., if the vertex $T_{21}(v)$ on S_1 is mapped to a vertex on S_2 via T_{12} whose pre-image under T_{12} has multiple elements. If so, we can improve T_{12} by mapping $T_{21}(v)$ to v.
(3) for an uncovered vertex v on S_2, find the nearest neighbor $w \in S_1$ of $T_{21}(v)$ such that the size of the pre-image of $T_{12}(w)$ under T_{12} is larger than one. Then we can improve the coverage of T_{12} by mapping w to v.
(4) for an uncovered vertex v on S_2 with neighbors $\mathcal{N}$, define the search space as $\mathcal{W} = \bigcup_{w \in \mathcal{N}} \left\{T_{12}^{-1}(w) \,\middle|\, |\,T_{12}^{-1}(w)\,| > 1\right\}$, i.e., for those neighbors of v with pre-image size larger than 1, define the union of their pre-image as the search space. Find the nearest neighbor $x \in \mathcal{W}$ to $T_{21}(v)$, then we can improve T_{12} by mapping x to v.

(1) and (2) can efficiently improve the coverage to some extent and get stuck quickly. Then (3) and (4) with a larger and relatively more complicated search space can further improve the coverage. Therefore, in our implementation, these steps are executed in a sequence for multiple iterations to improve the coverage of T_{12} and T_{21}.

Runtime. We report the runtime of eight randomly selected pairs from the four datasets (see Table 1).

Authors' addresses: Jing Ren, KAUST, jing.ren@kaust.edu.sa; Adrien Poulenard, École Polytechnique, adrien.poulenard@inria.fr; Peter Wonka, KAUST, pwonka@gmail.com; Maks Ovsjanikov, École Polytechnique, maks@lix.polytechnique.

0730-0301/2018/11-ART248
https://doi.org/10.1145/3272127.3275040

Pairs	BIM	PMF(heat)	SEG + ICP	WKS + directOp + BCICP	SEG + directOp + BCICP
FAUST tr_reg_023 and tr_reg_028	162.56	134.68	37.30	247.53	392.66
FAUST tr_reg_080 and tr_reg_084	172.47	120.99	35.14	227.80	282.92
FAUST tr_reg_025 and tr_reg_075	148.25	118.29	37.73	341.13	352.08
FAUST tr_reg_070 and tr_reg_090	152.70	119.21	34.54	303.02	462.53
TOSCA cat0 and cat10	1149.6	142.81	36.09	119.16	144.79
TOSCA centaur0 and centaur1	5822.3	169.27	16.18	137.29	74.62
TOSCA gorilla0 and david0	166.79	104.30	30.53	231.71	181.62
TOSCA gorilla0 and victoria11	188.15	147.73	15.73	302.29	263.99

Table 1. Runtime. We select two random pairs from the four datasets and report the running time (in sec) of different methods.

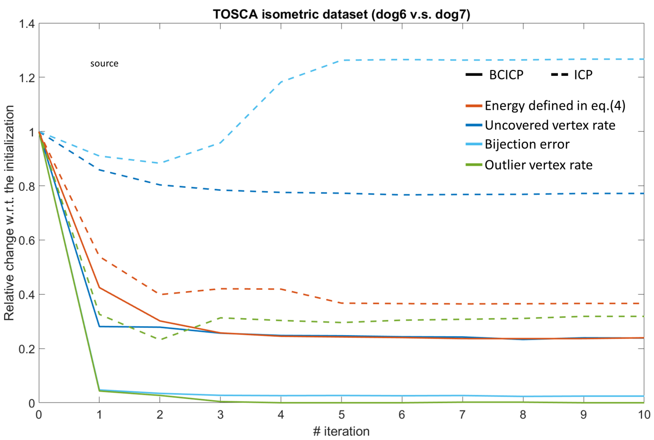

Fig. 1. Convergence of four different measurements over 10 iterations of BCICP (the solid lines) compared to the regular ICP (the dashed lines).

Convergence. Fig. 1 shows another example of the convergence rate of BCICP (solid lines) compared to ICP (dashed lines) w.r.t. different measurements, including the energy defined in Eq. (**??**) (orange), the fraction of uncovered vertices (dark blue), the bijectivity error (light blue), and the ratio of vertices that are categorized as outliers (green).

BCICP to improve PMF and BIM results. An advantage of our unifying framework is that it enables us to easier mix and match different techniques. BCICP not only works with functional maps, but also with PMF and BIM. In Fig. 3 left and middle we show how BCICP can consistently improve BIM and PMF on the FAUST dataset. In Fig. 3 right we demonstrate this improvement on the SHREC dataset. One interesting aspect of these tests is that we use different initializations of PMF.

Methods	Ave geodesic error($\times 10^{-3}$)		
	per vertex	per map	direct
BIM	43.69	44.43	93.17
PMF (heat kernel)	57.89	61.66	62.06
WKS Initialization	48.53	176.65	189.79
WKS + ICP	42.52	120.65	150.84
WKS + PMF(gauss kernel)	79.28	130.22	162.52
WKS + directOp	45.33	146.84	148.60
WKS + symmOp	49.75	217.57	304.57
WKS + directOp + BCICP	25.94	39.28	44.16
WKS + symmOp + BCICP	36.65	133.92	330.38
SEG Initialization	38.64	41.15	41.15
SEG + ICP	27.52	30.09	30.09
SEG + PMF (gauss kernel)	76.95	79.69	79.69
SEG + BCICP	24.93	27.22	27.22
SEG + directOp	36.09	38.41	38.41
SEG + directOp + BCICP	**22.63**	**24.83**	**24.83**

Table 2. Complete comparison: 200 FAUST Isometric pairs.

Methods	Ave geodesic error($\times 10^{-3}$)		
	per vertex	per map	direct
BIM	45.49	46.30	79.38
PMF (heat kernel)	55.92	59.95	60.54
WKS Initialization	68.17	219.37	241.54
WKS + ICP	65.59	166.37	210.11
WKS + PMF(gauss kernel)	70.71	130.51	213.47
WKS + directOp	61.98	181.81	188.96
WKS + symmOp	69.74	222.22	316.72
WKS + directOp + BCICP	29.33	46.90	51.31
WKS + symmOp + BCICP	43.34	131.86	284.00
SEG Initialization	42.86	57.45	57.45
SEG + ICP	30.94	45.68	45.68
SEG + PMF (gauss kernel)	67.02	80.04	80.04
SEG + BCICP	25.21	39.77	39.77
SEG + directOp	40.44	54.69	54.69
SEG + directOp + BCICP	**23.50**	**37.29**	**37.29**

Table 3. Complete comparison: 400 FAUST non-Isometric pairs

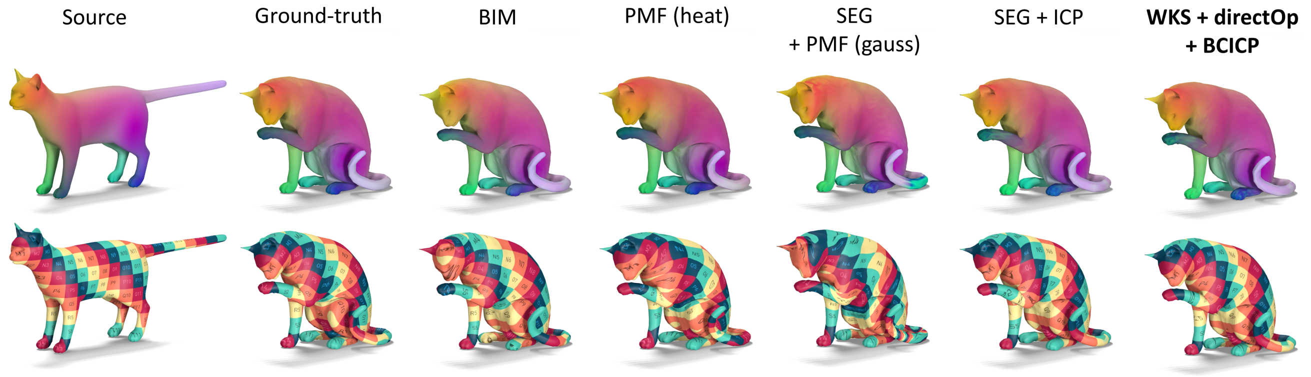

Fig. 2. Texture transfer: We compare our algorithm to different competing methods. We use the vertex-to-vertex maps (visualized in the first row) to transfer the texture to the target shape (visualized in the second row). The texture-transfer provides a good visualization for the local distortion of the resulting maps.

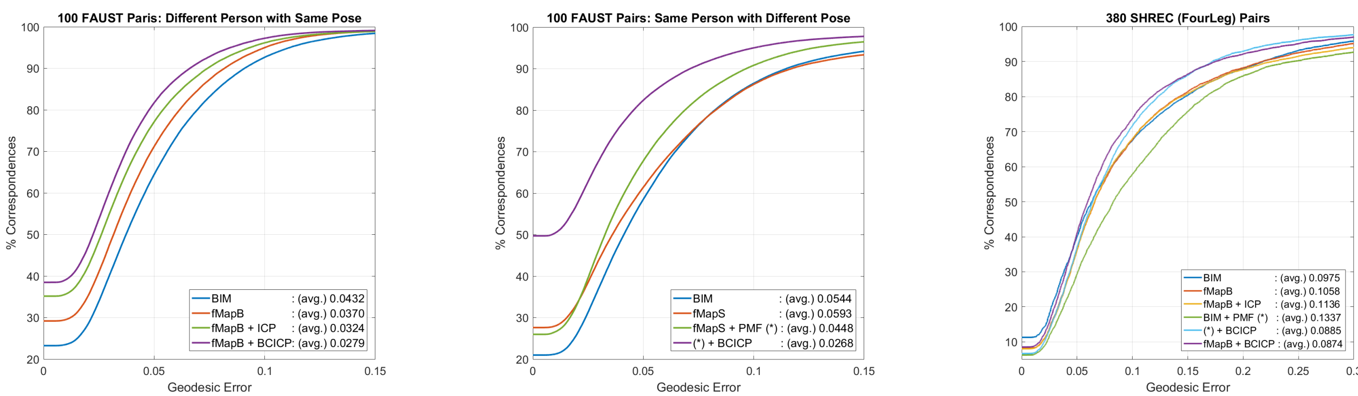

Fig. 3. (left) BCICP improves BIM on the FAUST dataset. (middle) BCICP improves PMF on the FAUST dataset. PMF is initialized using functional maps. (right) BCICP improves BIM and PMF on the SHREC dataset. PMF is initialized with BIM.

Methods	Ave geodesic error($\times 10^{-3}$)		
	per vertex	per map	direct
BIM	41.20	45.14	70.76
PMF (heat kernel)	49.03	71.44	71.82
WKS Initialization	48.94	160.36	200.80
WKS + ICP	44.90	128.45	194.01
WKS + PMF(gauss kernel)	70.02	124.69	212.87
WKS + directOp	44.54	74.56	75.94
WKS + symmOp	45.03	78.74	329.03
WKS + directOp + BCICP	**28.25**	**38.56**	**43.34**
WKS + symmOp + BCICP	30.12	41.76	344.67
SEG Initialization	52.99	71.40	83.06
SEG + ICP	42.93	61.36	73.30
SEG + PMF (gauss kernel)	70.88	83.85	101.01
SEG + BCICP	36.55	54.29	65.86
SEG + directOp	42.56	60.55	69.59
SEG + directOp + BCICP	34.80	52.13	61.36

Table 4. Complete comparison: 568 TOSCA Isometric pairs.

Methods	Ave geodesic error($\times 10^{-3}$)		
	per vertex	per map	direct
BIM	256.11	265.83	358.99
PMF (heat kernel)	190.21	249.14	287.29
WKS Initialization	329.57	439.67	454.57
WKS + ICP	314.82	406.67	449.75
WKS + PMF(gauss kernel)	169.18	227.67	362.10
WKS + directOp	230.79	340.80	342.76
WKS + symmOp	230.39	341.05	343.61
WKS + directOp + BCICP	**90.03**	**112.68**	**120.96**
WKS + symmOp + BCICP	112.02	163.51	234.64
SEG Initialization	140.02	174.56	210.08
SEG + ICP	145.92	178.14	214.05
SEG + PMF (gauss kernel)	168.30	192.46	228.28
SEG + BCICP	120.45	150.71	187.38
SEG + directOp	137.73	170.20	205.44
SEG + directOp + BCICP	114.94	144.45	180.39

Table 5. Complete comparison: 190 TOSCA non-Isometric pairs