

CCuantuMM: Cycle-Consistent Quantum-Hybrid Matching of Multiple Shapes

—Supplementary Material—

Harshil Bhatia^{1,2} Edith Tretschk² Zorah Löhner³ Marcel Seelbach Benkner³
 Michael Moeller³ Christian Theobalt² Vladislav Golyanik²

¹Indian Institute of Technology, Jodhpur ²MPI for Informatics, SIC ³Universität Siegen

Overview

This supplementary material contains additional results. We present them in the order they are mentioned in the main paper. First, Sec. **A** provides an overview of our notation. Sec. **B** formally states the N -shape algorithm. Sec. **C** contains large-scale figures showing the 100 matched FAUST shapes. Sec. **D** describes the elimination of higher-order terms in detail. Sec. **E** states the weight matrix \tilde{W} of the final QUBO. Sec. **F** shows that the total energy almost never decreases in practice. Sec. **G** compares our anchor scheme to using random triplets. Sec. **H** discusses the time complexity of our proposed method. Sec. **I** provides more implementation details on initialisation and deciding trivial cycles. Sec. **J** discusses the minor embeddings on real quantum hardware and contains further QPU experiments. Sec. **K** presents more results, including more ablation experiments. Finally, Sec. **L** compares existing quantum computer vision works for alignment tasks.

In addition, the supplementary material contains a video showing the evolution of the matchings of a ten-shape FAUST instance over the course of the optimisation. We visualise the matchings by fixing the colouring of the top-left shape and transferring this colouring to all other shapes according to the estimated correspondences.

A. Notations

Tab. 1 summarises the notation we use.

B. N -Shape Algorithm

We provide the formal N -shape algorithm as Alg. 1.

C. Matching 100 Shapes

Fig. 1 shows qualitative results of all 100 matched FAUST shapes.

D. Higher-Order Terms

In this section, we give the expansions for the third term of (7), namely $E_{\mathcal{X}\mathcal{Z}}(P_{\mathcal{X}\mathcal{Z}}(\alpha, \beta))$, as it contains higher-

Notation	Meaning
$\mathcal{I}, \mathcal{J}, \mathcal{X}, \mathcal{Y}, \mathcal{Z}$	shapes (represented as meshes)
$P_{\mathcal{I}\mathcal{J}}$	permutation matrix from shape \mathcal{I} to \mathcal{J}
x, y, u, v	vertices of a shape
\mathcal{P}	set of permutations
\mathcal{S}	set of shapes
W	energy matrix
N	number of shapes
I	identity matrix
$E_{\mathcal{I}\mathcal{J}}(P, Q)$	$\text{vec}(P)^\top W_{\mathcal{I}\mathcal{J}} \text{vec}(Q)$
$E_{\mathcal{I}\mathcal{J}}(P)$	$E_{\mathcal{I}\mathcal{J}}(P, P)$
$F_{\mathcal{I}\mathcal{J}}(A, B)$	$E_{\mathcal{I}\mathcal{J}}(A, B) + E_{\mathcal{I}\mathcal{J}}(B, A)$
α, β	decision variables
k	size of decision variables α, β
m	number of worst vertices
$I_{\mathcal{X}\mathcal{Y}}(x)$	relative inconsistency of vertex x for $P_{\mathcal{X}\mathcal{Y}}$
$V_{\mathcal{X}}$	set of worst vertices for shape \mathcal{X}
$\mathcal{C}_{\mathcal{X}}$	set of cycles for shape \mathcal{X}
A	anchor shape
C_i	$(e_i - I)P_{\mathcal{X}\mathcal{Y}}$; single-cycle update of $P_{\mathcal{X}\mathcal{Y}}$
\tilde{C}_j	$(e_j - I)P_{\mathcal{Y}\mathcal{Z}}$; single-cycle update of $P_{\mathcal{Y}\mathcal{Z}}$
K_{ij}	$C_i \tilde{C}_j$

Table 1. Notation used in this work.

order terms. Fig. 12 shows the equations. The terms in red are cubic or bi-quadratic and we hence assume them to be 0, except for the summands that simplify to quadratic terms. For example, $\alpha_i \beta_j \alpha_i \beta_j E_{\mathcal{X}\mathcal{Z}}(C_i \tilde{C}_j, C_i \tilde{C}_j)$ simplifies to the linear summand $\alpha_i \beta_j E_{\mathcal{X}\mathcal{Z}}(C_i \tilde{C}_j, C_i \tilde{C}_j)$. This then yields (8) from the main paper.

E. Final QUBO

Fig. 2 shows the weight matrix $\tilde{W} \in \mathbb{R}^{2k \times 2k}$ that is used for the final QUBO.

F. Evolution During Optimisation

Fig. 3 depicts how the total energy (11) and PCK evolve during optimisation. Importantly, the energy almost never increases in practice. Furthermore, our algorithm converges

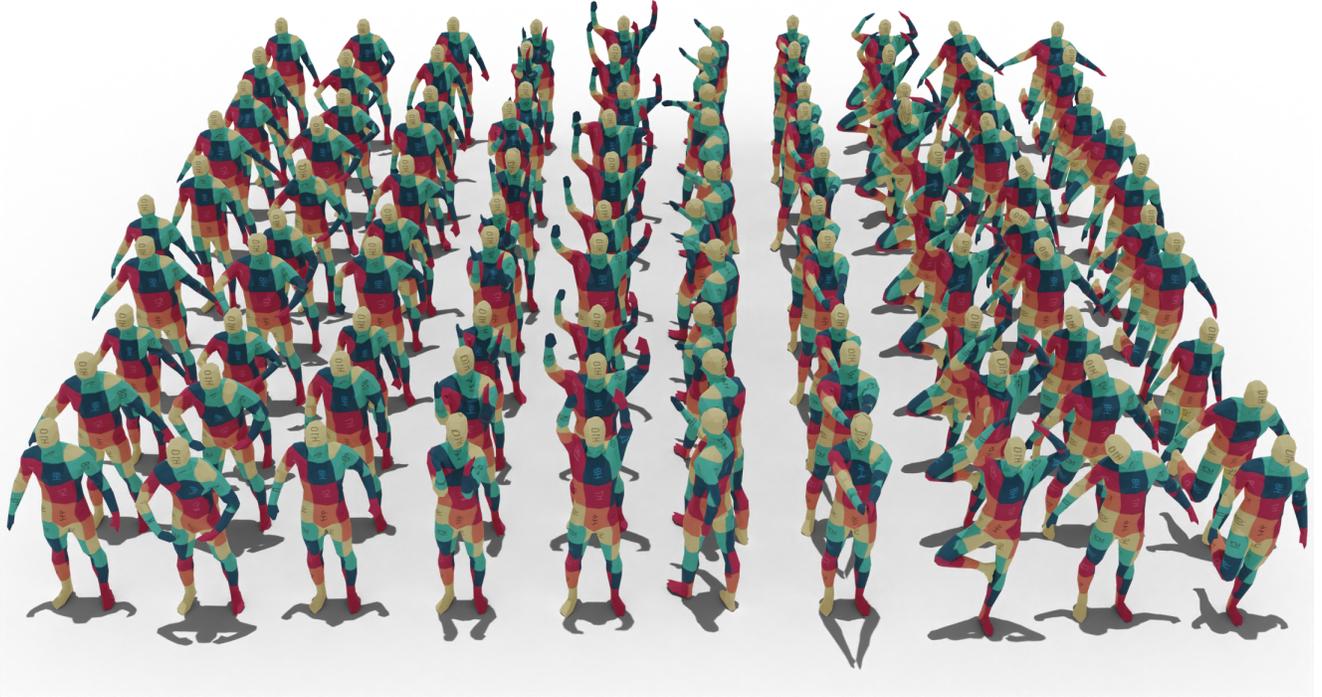


Figure 1. **Extended version of Fig. 1 in the main paper.** We visualise the matchings between all 100 FAUST shapes via texture transfer. All correspondences are cycle-consistent by design.

$$\tilde{W}_{ij} = \begin{cases} E_{YZ}(C_i, C_i) + E_{XY}(P_{XY}, C_i) + E_{XY}(C_i, P_{XY}) + E_{XZ}(P_{XY}P_{YZ}, C_iP_{YZ}) + E_{XZ}(C_iP_{YZ}, P_{XY}P_{YZ}) + E_{XZ}(C_iP_{YZ}, C_iP_{YZ}) & \text{if } i = j \leq k \\ E_{XY}(\tilde{C}_j, \tilde{C}_j) + E_{YZ}(P_{YZ}, \tilde{C}_j) + E_{YZ}(\tilde{C}_j, P_{YZ}) + E_{XZ}(P_{XY}P_{YZ}, P_{XY}\tilde{C}_j) + E_{XZ}(P_{XY}\tilde{C}_j, P_{XY}P_{YZ}) + E_{XZ}(P_{XY}\tilde{C}_j, P_{XY}\tilde{C}_j) & \text{if } i = j > k \\ E_{YZ}(C_i, C_j) + E_{XZ}(C_iP_{YZ}, C_jP_{YZ}) & \text{if } i \neq j, i \leq k, j \leq k \\ E_{XY}(\tilde{C}_i, \tilde{C}_j) + E_{XZ}(P_{XY}\tilde{C}_i, P_{XY}\tilde{C}_j) & \text{if } i \neq j, i > k, j > k \\ E_{XZ}(P_{XY}P_{YZ}, C_i\tilde{C}_j) + E_{XZ}(P_{XY}\tilde{C}_j, C_iP_{YZ}) + E_{XZ}(P_{XY}\tilde{C}_j, C_i\tilde{C}_j) + E_{XZ}(C_iP_{YZ}, P_{XY}\tilde{C}_j) & \\ + E_{XZ}(C_iP_{YZ}, C_i\tilde{C}_j) + E_{XZ}(C_i\tilde{C}_j, P_{XY}P_{YZ}) + E_{XZ}(C_i\tilde{C}_j, P_{XY}\tilde{C}_j) & \\ + E_{XZ}(C_i\tilde{C}_j, C_iP_{YZ}) + E_{XZ}(C_i\tilde{C}_j, C_i\tilde{C}_j) & \text{if } i \neq j, i \leq k, j > k \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Figure 2. The final QUBO weight matrix \tilde{W} . For indices $i > k$, we define $\tilde{C}_i = \tilde{C}_{i-k}$.

close to the ground truth. A sudden and significant improvement occurs as soon as the schedule switches from geodesics to Gaussians. We note that although the objective uses Gaussians in the second half of the schedule, both the total energy plotted here and the PCK are based on the (unfiltered) geodesic distances.

G. Fixed Anchor vs. Random Triplets

Fig. 4 compares our anchor-based scheme with using random triplets. We see a small improvement with our scheme.

H. Time Complexity

Our method mainly consists of constructing and solving the QUBO matrix \tilde{W} , which is based on W . However, standard meshes contain thousands of vertices, which makes naïvely calculating the full $W \in \mathbb{R}^{n^2 \times n^2}$ not feasible due to the memory restrictions. Fortunately, we do not need to compute the full W but only a small set of its entries. This is due to the extreme sparsity of the $c_i - I$ matrix (only four non-zero elements) since we only consider 2-cycles. Using k 2-cycles leads to a worst-case time complexity of $\mathcal{O}(nk^2)$ for the three-shape Alg. 1 from the main paper (*i.e.*, one sub-sub-iteration). For the sub-sub-iterations of a sub-iteration, \tilde{W} is constant and we thus need to compute \tilde{W}

Algorithm 1 CCuantuMM (Matching N Shapes; Sec. 4.2)

Input: \mathcal{S}, T
Output: \mathcal{P}

- 1: initialise \mathcal{P}^{init} from HKS descriptors
 - 2: determine anchor $A \in \mathcal{S}$
 - 3: $\mathcal{P}^0 \leftarrow \{P_{\mathcal{I}A}^{init}\}_{\mathcal{I} \in \mathcal{S}, \mathcal{I} \neq A}$
 - 4: randomly pick $\mathcal{X}^{-1} \in \mathcal{S}$ ▷ technicality for \mathcal{Z}^0
 - 5: **for** $i = 0$ to $2T(|\mathcal{S}| - 1)$ **do** ▷ iterations
 - 6: **if** $i \bmod (|\mathcal{S}| - 1) == 0$ **then**
 - 7: $\mathcal{S}' \leftarrow \mathcal{S} \setminus \{A\}$ ▷ for stratified sampling
 - 8: **end if**
 - 9: randomly pick $\mathcal{X}^i \in \mathcal{S}'$
 - 10: $\mathcal{S}' \leftarrow \mathcal{S}' \setminus \{\mathcal{X}^i\}$
 - 11: $\mathcal{Y}^i \leftarrow A$
 - 12: $\mathcal{Z}^i \leftarrow \mathcal{X}^{i-1}$
 - 13: get $P_{\mathcal{X}^i A}^i, P_{\mathcal{Z}^i A}^i$ from \mathcal{P}^i
 - 14: $P_{\mathcal{X}^i \mathcal{Y}^i}^i \leftarrow P_{\mathcal{X}^i A}^i$
 - 15: $P_{\mathcal{Y}^i \mathcal{Z}^i}^i \leftarrow (P_{\mathcal{Z}^i A}^i)^{-1}$
 - 16: $P_{\mathcal{X}^i \mathcal{Z}^i}^i \leftarrow P_{\mathcal{X}^i A}^i (P_{\mathcal{Z}^i A}^i)^{-1}$
 - 17: mode \leftarrow geodesic **if** $i < T(|\mathcal{S}| - 1)$ **else** Gaussian
 - 18: $\{P_{\mathcal{X}^i \mathcal{Y}^i}^{i+1}, P_{\mathcal{Y}^i \mathcal{Z}^i}^{i+1}, P_{\mathcal{X}^i \mathcal{Z}^i}^{i+1}\} \leftarrow$ run Alg. 1 with mode
 - 19: on $(\{P_{\mathcal{X}^i \mathcal{Y}^i}^i, P_{\mathcal{Y}^i \mathcal{Z}^i}^i, P_{\mathcal{X}^i \mathcal{Z}^i}^i\}, \{\mathcal{X}^i, \mathcal{Y}^i, \mathcal{Z}^i\})$
 - 20: $P_{\mathcal{X}^i A}^{i+1} \leftarrow P_{\mathcal{X}^i \mathcal{Y}^i}^{i+1}$
 - 21: $P_{\mathcal{Z}^i A}^{i+1} \leftarrow (P_{\mathcal{Y}^i \mathcal{Z}^i}^{i+1})^{-1}$
 - 22: $\mathcal{P}^{i+1} \leftarrow (\mathcal{P}^i \setminus \{P_{\mathcal{X}^i A}^i, P_{\mathcal{Z}^i A}^i\}) \cup \{P_{\mathcal{X}^i A}^{i+1}, P_{\mathcal{Z}^i A}^{i+1}\}$
 - 23: **end for**
 - 24: **return** $\mathcal{P} = \mathcal{P}^{2T(|\mathcal{S}|-1)}$
-

only once for each sub-iteration. In addition, each iteration has $k-1$ sub-iterations, resulting in a time complexity of $\mathcal{O}(nk^3)$ for each iteration. Furthermore, our implementation for computing \tilde{W} is significantly faster in practice than the sub-sampling technique proposed in Q-Match.

I. Implementation Details

I.1. Initialisation

The initial set of permutations \mathcal{P}^{init} is computed using a descriptor-based similarity $DS_{\mathcal{I}\mathcal{J}} \in \mathbb{R}^{n \times n}$ between all n vertices of shape pairs $\mathcal{I}, \mathcal{J} \in \mathcal{S}$. Specifically, $DS_{\mathcal{I}\mathcal{J}}(u, v)$ contains the similarity (inner product) of the normalised heat-kernel-signatures (HKS) [4] descriptors (which we extend by an additional dimension indicating whether a vertex lies on the left or right side of a shape) of vertex u of \mathcal{I} and vertex v of \mathcal{J} . The left-right descriptors reduce left-right flips in the solution since the shape classes we consider are globally symmetric, which is not captured by the local HKS descriptors. This is standard practice in the shape-matching literature [9]. The solution of a linear assignment problem on $DS_{\mathcal{I}\mathcal{J}}$ is then the initial $P_{\mathcal{I}\mathcal{J}}^{init}$.

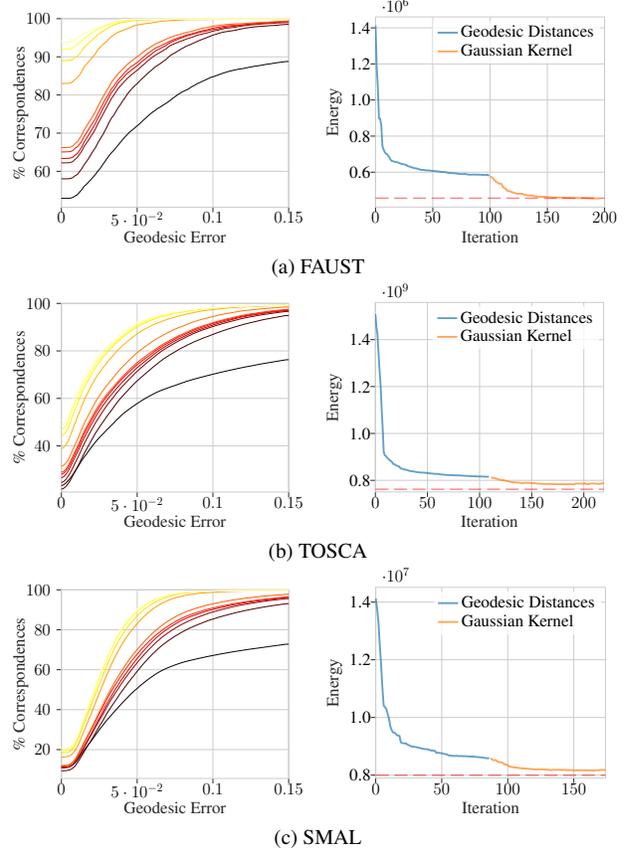


Figure 3. Evolution during the optimisation (left) of the PCK, depicted with a colour bar gradient, and (right) of the total energy. The horizontal red-dashed line is the energy of the ground-truth solution. The results are on (a) a class of FAUST containing ten shapes, (b) the cat class of TOSCA containing eleven shapes, and (c) the cat class of SMAL containing nine shapes.

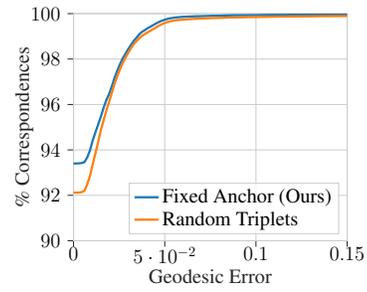


Figure 4. PCK when using a fixed anchor (as our method does) and when using random triplets. We average across all classes of FAUST.

I.2. Pre-Computing Trivial Cycles

The matrix \tilde{W} consists of couplings (quadratic terms) and linear terms. Numerical experiments show that there often exist cycles with linear terms that dominate the cor-

responding coupling terms. This happens when a cycle is largely uncorrelated to the rest of the cycles in the current set of cycles. In this case, the decision for such a cycle can be made trivially.

We now derive an inequality for how large the linear term has to be such that the couplings can be neglected in the optimisation problem. Consider the QUBO problem:

$$\min_{\alpha \in \{0,1\}^k} \alpha^T W \alpha + b^T \alpha, \quad (2)$$

where α are the decision variables, $b \in \mathbb{R}^k$ is a vector, and W is a symmetric matrix with zeros on the diagonal representing the couplings. We can look at the terms that depend on α_q for fixed q separately:

$$\begin{aligned} \alpha^T W \alpha &= \alpha_q \sum_{i \neq q} W_{q,i} \alpha_i + \alpha_q \sum_{i \neq q} \alpha_i W_{i,q} \\ &+ \sum_{j \neq q} \left(\sum_{i \neq q} \alpha_j W_{j,i} \alpha_i + \alpha_j b_j \right) + \alpha_q b_q. \end{aligned}$$

As W is symmetric, $W_{i,q} = W_{q,i}$ holds and we can write

$$\begin{aligned} \alpha^T W \alpha &= \alpha_q \left(b_q + 2 \sum_{i \neq q} W_{q,i} \alpha_i \right) \\ &+ \sum_{j \neq q} \left(\sum_{i \neq q} \alpha_j W_{j,i} \alpha_i + \alpha_j b_j \right). \end{aligned}$$

It follows that if:

$$|b_q| \geq \sum_{i \neq q} 2|W_{q,i}|, \quad (3)$$

then we can make the decision based on the sign of b_q : If b_q is positive we do not choose the cycle as doing so would increase the energy. This reduces the number of physical qubits required for the embedding.

J. Minor Embeddings and Other QPU Experiments

J.1. QPU Processing and Annealing Time

Optimising a single QUBO uses $\sim 40ms$ of total QPU processing time for 200 anneals. This is also called the *QPU access time* [7]. However, there are also several overheads that occur when solving QUBOs by accessing a D-Wave annealer via the cloud. For example, a latency when connecting to the D-Wave annealer and a post processing time. The QPU access time also includes a programming time.

These overheads can be orders of magnitude greater than the time taken by the actual annealing, which is very short since we use the default annealing schedule and the default annealing time of 20 μs .

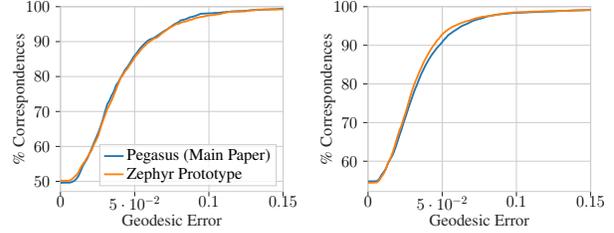


Figure 5. PCK curves for (left) a three-shape and (right) a ten-shape inter-class FAUST instance on both QPU architectures.

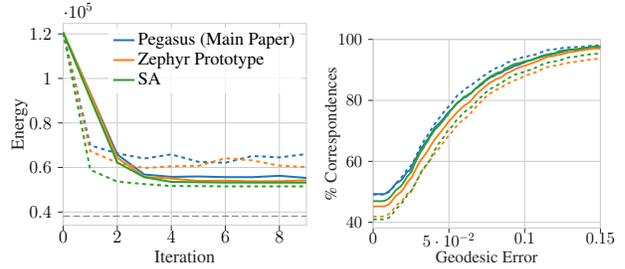


Figure 6. Quantitative results when using (solid) 20 and (dashed) 40 worst vertices on a three-shape FAUST instance. We show (left) the energy evolution during optimisation and (right) the final PCK curves.

J.2. Minor Embeddings

As explained in the main paper, not all physical qubits on a real quantum processing unit (QPU) can be connected (coupled) with each other. Thus, a minor embedding of the logical-qubit graph (defined by non-zero entries of the QUBO matrix) into the physical-qubit graph (defined by the hardware) is required. This can lead to a chain of multiple physical qubits representing a single logical qubit.

Our logical input graph is a clique. Due to the limited connectivity of current hardware, a clique cannot be directly embedded onto the physical annealer. We thus require a minor embedding, which is commonly computed using Cai *et al.*'s method [5]. Fig. 13 visualises an example minor embedding. We note that, since our input graphs are cliques, a generalised embedding can be pre-computed and reused, not impacting the time complexity.

J.3. Minor Embeddings in Practice

In this section, we investigate the empirical impact of minor embeddings on the solutions. The minor embeddings depend on the qubit topology, *i.e.* the physical qubit connectivity pattern. Here, we show results on the D-Wave Advantage 4.1 with its Pegasus topology (used in the main paper) and also first results on a D-Wave Advantage2 prototype of the next-generation Zephyr topology, which has a higher connectivity than Pegasus.

Fig. 5 shows PCK curves on both topologies when using

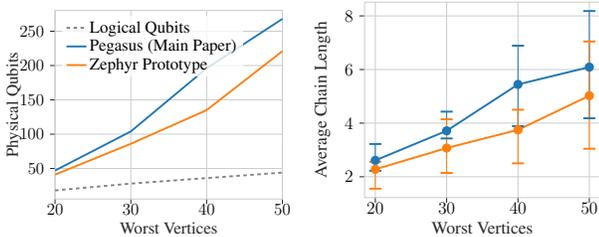


Figure 7. Structural changes of the minor embeddings when using more worst vertices. We show (left) the number of physical qubits and (right) the average chain length, for both QPU topologies. The number of *logical* qubits equals the number of worst vertices.

$m=2k=20$ worst vertices. Both architectures obtain similar results, although they are very slightly worse than SA. However, as discussed in the main paper, we find that the performance of CCuantuMM degrades significantly when using more than 20 worst vertices with QA. Specifically, Fig. 6 shows that the quality of the matchings worsens when using 40 worst vertices on both topologies, with Zephyr obtaining slightly better results. Still, in both cases, the quality is worse than the matching quality obtained by SA. Only SA shows the desired behaviour of improving when more worst vertices are used.

The cause for these results lies with the structure of the minor embeddings and not the plain number of physical qubits. Fig. 7 shows how the structural properties of the minor embeddings evolve as the number of worst vertices increases. For 20 worst vertices, they are similar. However, for 40 worst vertices, Zephyr uses fewer physical qubits and smaller chains, which explains the very slight performance advantage in Fig. 6. Physical qubits in a chain representing a single logical qubit are less likely to all anneal to the same value the longer the chain is. Longer chains become unstable and hence inconsistent, leading to inferior solution quality.

K. Further Results and Ablations

K.1. Additional Qualitative Results

Fig. 14, Fig. 15, and Fig. 16 provide additional qualitative examples of matchings on TOSCA and SMAL calculated with our method and the competitors.

K.2. Variation Within a Dataset

Within a dataset, some instances have more difficult deformations and are inherently harder to match than easier instances, independent of the method employed. We investigate the extent of this variation by taking a closer look at TOSCA. We observe a significant variation of PCK curves across different classes in Fig. 8 and of AUC in Tab. 2.

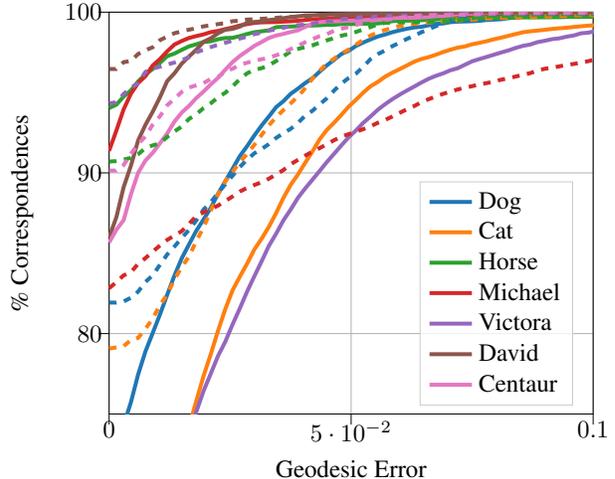


Figure 8. PCK for seven different classes of TOSCA. We plot our method with solid lines and IsoMuSh [9] with dashed lines.

	Dog	Cat	Horse	Michael	Victoria	David	Centaur
Ours	0.957	0.917	0.990	0.992	0.912	0.989	0.983
IsoMush	0.959	0.959	0.981	0.939	0.991	0.996	0.984

Table 2. AUC across seven different classes of TOSCA.

K.3. Influence of Descriptors

We ablate the need for left-right indicators when using HKS descriptors for initialisation. Fig. 9 contains results on the cat class of TOSCA. Without left-right indicators, we observe flips in the matchings on TOSCA and partial flips on FAUST for inter-class instances. This is expected since both our method and IsoMuSh exploit intrinsic properties of the shapes, which are invariant to such symmetric flips.

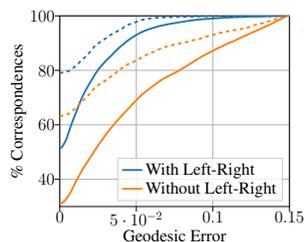


Figure 9. PCK with and without left-right descriptors. We plot our method with solid lines and IsoMuSh [9] with dashed lines.

K.4. Noise Perturbation

We investigate the robustness of our approach to noise. To that end, we analyse the effect of adding synthetic perturbations to the shapes. Specifically, for each FAUST mesh, we add a Gaussian-distributed offset along the vertex normal to each vertex position. This ensures the meshed structure of the shape does not change. Fig. 10 visualises the

amount of noise we experiment with. Fig. 11 shows that our method is significantly more robust to noise than IsoMuSh.

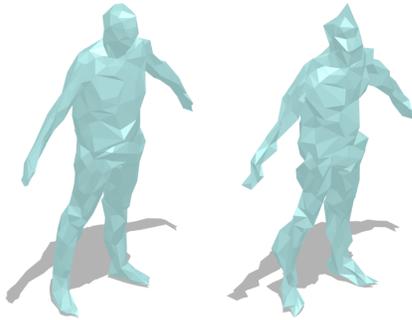


Figure 10. Example of perturbing the geometry on FAUST, with the noise variance set to 0.01 (left) and 0.02 (right).

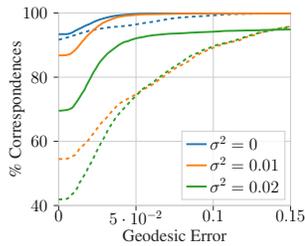


Figure 11. PCK for different amounts of perturbation (Gaussian noise with variance σ^2). We plot our method with solid lines and IsoMuSh [9] with dashed lines. We report results on a class of FAUST.

L. Related Quantum Computer Vision Works

Several quantum methods tackle alignment tasks, as Tab. 3 shows. However, only Q-Match is relevant for comparisons. Several works only consider two point clouds and cannot handle the multi-matching setting of our work, do not consider fully non-rigid transformations, or only operate on point clouds, not meshes. QGM [13] only considers two graphs with at most four vertices. Q-Sync [2] similarly only works on at most five vertices, far fewer than what our method can handle.

References

- [1] Federica Arrigoni, Willi Menapace, Marcel Seelbach Benkner, Elisa Ricci, and Vladislav Golyanik. Quantum motion segmentation. In *Eur. Conf. Comput. Vis.*, 2022. 7
- [2] Tolga Birdal, Vladislav Golyanik, Christian Theobalt, and Leonidas Guibas. Quantum permutation synchronization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021. 6, 7
- [3] Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. *Numerical geometry of non-rigid shapes*. Springer Science & Business Media, 2008. 10, 11
- [4] Michael M Bronstein and Iasonas Kokkinos. Scale-invariant heat kernel signatures for non-rigid shape recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2010. 3
- [5] Jun Cai, William G Macready, and Aidan Roy. A practical heuristic for finding graph minors. *arXiv preprint arXiv:1406.2741*, 2014. 4
- [6] Rigetti Computing. <https://www.rigetti.com/>, 2022. 7
- [7] D-Wave. Operation and timing. https://docs.dwavesys.com/docs/latest/c_qpu_timing.html, 2021. 4
- [8] D-Wave Systems, Inc. Leap, 2022. 9
- [9] Maolin Gao, Zorah Lahner, Johan Thunberg, Daniel Cremers, and Florian Bernard. Isometric multi-shape matching. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021. 3, 5, 6
- [10] Vladislav Golyanik and Christian Theobalt. A quantum computational approach to correspondence problems on point sets. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 7
- [11] Natacha Kuete Meli, Florian Mannel, and Jan Lellmann. An iterative quantum approach for transformation estimation from point sets. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022. 7
- [12] Mohammadreza Noormandipour and Hanchen Wang. Matching point sets with quantum circuit learning. In *International Conference on Acoustics, Speech and Signal Processing*, 2022. 7
- [13] Marcel Seelbach Benkner, Vladislav Golyanik, Christian Theobalt, and Michael Moeller. Adiabatic quantum graph matching with permutation matrix constraints. In *Int. Conf. 3D Vis. (3DV)*, 2020. 6, 7
- [14] Marcel Seelbach Benkner, Zorah Löhner, Vladislav Golyanik, Christof Wunderlich, Christian Theobalt, and Michael Moeller. Q-match: Iterative shape matching via quantum annealing. In *Int. Conf. Comput. Vis.*, 2021. 7

Method	Problem	Transformation	Input Type	# Inputs	# Points	# Qubits (per sweep)	QPU	Iterative
QA, CVPR 2020 [10]	TE, PSR	A/R	point clouds	2	$\leq 5k^*$	≤ 140	2000Q	
IQT, CVPR 2022 [11]	TE	R	point clouds	2	$\leq 1.5k^*$	≤ 10	2000Q	✓
qKC, ICASSP 2022 [12]	PSR	R	point clouds	2	$\leq 2k^*$	4-6	Rigetti [6]	✓
QGM, 3DV 2020 [13]	GM	F	graphs	2	≤ 4	≤ 50	2000Q	
QSync, CVPR 2021 [2]	PS, GM	F	perm. matrices	≤ 5	≤ 5	$\leq 1.5k$	Adv1.1	
QuMoSeg, ECCV 2022 [1]	MS	F	segm. matrices	≤ 9	≤ 200	≤ 250	Adv{1.1;4.1}	
Q-Match, ICCV 2021 [14]	MA	R/NR	meshes	2	≤ 500	≤ 200	Adv4.1	✓
CCuantuMM (Ours)	MA	R/NR	meshes	≤ 100	$\leq 1k$	≤ 50	Adv4.1, Adv2 prototype	✓

Table 3. Overview of related quantum methods for alignment tasks on point sets, graphs and meshes. “**”: according to the experiments reported in the paper; the methods can also process larger point clouds. Key: “TE”: transformation estimation; “PSR”: point set alignment; “GM”: graph matching; “PS”: permutation synchronisation; “MS”: motion segmentation; “MA”: mesh alignment; “A”: affine transformation; “R”: rigid transformation; “NR”: non-rigid deformations; “F”: the method operates on features extracted in a pre-processing step (and can support both rigid and non-rigid transformations). Note that only Q-Match [14] can be extended and applied to our data.

$$\begin{aligned}
E_{XZ}(P_{XY}(\alpha, \beta)) &= E_{XZ} \left((P_{XY} + \sum_{i=1}^k \alpha_i C_i) \cdot (P_{YZ} + \sum_{j=1}^k \beta_j \tilde{C}_j), (P_{XY} + \sum_{q=1}^k \alpha_q C_q) \cdot (P_{YZ} + \sum_{l=1}^k \beta_l \tilde{C}_l) \right) \\
&= E_{XZ} \left(P_{XY} P_{YZ}, (P_{XY} + \sum_{q=1}^k \alpha_q C_q) \cdot (P_{YZ} + \sum_{l=1}^k \beta_l \tilde{C}_l) \right) + E_{XZ} \left(\sum_{j=1}^k \beta_j P_{XY} \tilde{C}_j, (P_{XY} + \sum_{q=1}^k \alpha_q C_q) \cdot (P_{YZ} + \sum_{l=1}^k \beta_l \tilde{C}_l) \right) \\
&\quad + E_{XZ} \left(\sum_{i=1}^k \alpha_i C_i P_{YZ}, (P_{XY} + \sum_{q=1}^k \alpha_q C_q) \cdot (P_{YZ} + \sum_{l=1}^k \beta_l \tilde{C}_l) \right) + E_{XZ} \left(\sum_{i=1}^k \sum_{j=1}^k \alpha_i \beta_j C_i \tilde{C}_j, (P_{XY} + \sum_{q=1}^k \alpha_q C_q) \cdot (P_{YZ} + \sum_{l=1}^k \beta_l \tilde{C}_l) \right) \\
&= E_{XZ}(P_{XY} P_{YZ}, P_{XY} P_{YZ}) + \sum_{l=1}^k \beta_l E_{XZ}(P_{XY} P_{YZ}, P_{XY} \tilde{C}_l) + \sum_{q=1}^k \alpha_q E_{XZ}(P_{XY} P_{YZ}, C_q P_{YZ}) \\
&\quad + \sum_{q=1}^k \sum_{l=1}^k \alpha_q \beta_l E_{XZ}(P_{XY} P_{YZ}, C_q \tilde{C}_l) + \sum_{j=1}^k \beta_j E_{XZ}(P_{XY} \tilde{C}_j, P_{XY} P_{YZ}) + \sum_{j=1}^k \sum_{l=1}^k \beta_j \beta_l E_{XZ}(P_{XY} \tilde{C}_j, P_{XY} \tilde{C}_l) \\
&\quad + \sum_{j=1}^k \sum_{q=1}^k \beta_j \alpha_q E_{XZ}(P_{XY} \tilde{C}_j, C_q P_{YZ}) + \sum_{j=1}^k \sum_{q=1}^k \sum_{l=1}^k \beta_j \alpha_q \beta_l E_{XZ}(P_{XY} \tilde{C}_j, C_q \tilde{C}_l) + \sum_{i=1}^k \sum_{q=1}^k \alpha_i E_{XZ}(C_i P_{YZ}, P_{XY} P_{YZ}) \\
&\quad + \sum_{i=1}^k \sum_{l=1}^k \alpha_i \beta_l E_{XZ}(C_i P_{YZ}, P_{XY} \tilde{C}_l) + \sum_{i=1}^k \sum_{q=1}^k \alpha_i \alpha_q E_{XZ}(C_i P_{YZ}, C_q P_{YZ}) + \sum_{i=1}^k \sum_{q=1}^k \sum_{l=1}^k \alpha_i \alpha_q \beta_l E_{XZ}(C_i P_{YZ}, C_q \tilde{C}_l) \\
&\quad + \sum_{i=1}^k \sum_{j=1}^k \alpha_i \beta_j E_{XZ}(C_i \tilde{C}_j, P_{XY} P_{YZ}) + \sum_{i=1}^k \sum_{j=1}^k \sum_{l=1}^k \alpha_i \beta_j \beta_l E_{XZ}(C_i \tilde{C}_j, P_{XY} \tilde{C}_l) + \sum_{i=1}^k \sum_{j=1}^k \sum_{q=1}^k \alpha_i \beta_j \alpha_q E_{XZ}(C_i \tilde{C}_j, C_q P_{YZ}) \\
&\quad + \sum_{i=1}^k \sum_{j=1}^k \sum_{q=1}^k \sum_{l=1}^k \alpha_i \beta_j \alpha_q \beta_l E_{XZ}(C_i \tilde{C}_j, C_q \tilde{C}_l)
\end{aligned} \tag{4}$$

Figure 12. We expand the third term of (7) from the main paper, which yields higher-order terms (highlighted in red). When the summands constituting these terms are truly cubic or bi-quadratic, we assume them to be 0, which results in the QUBO (8).

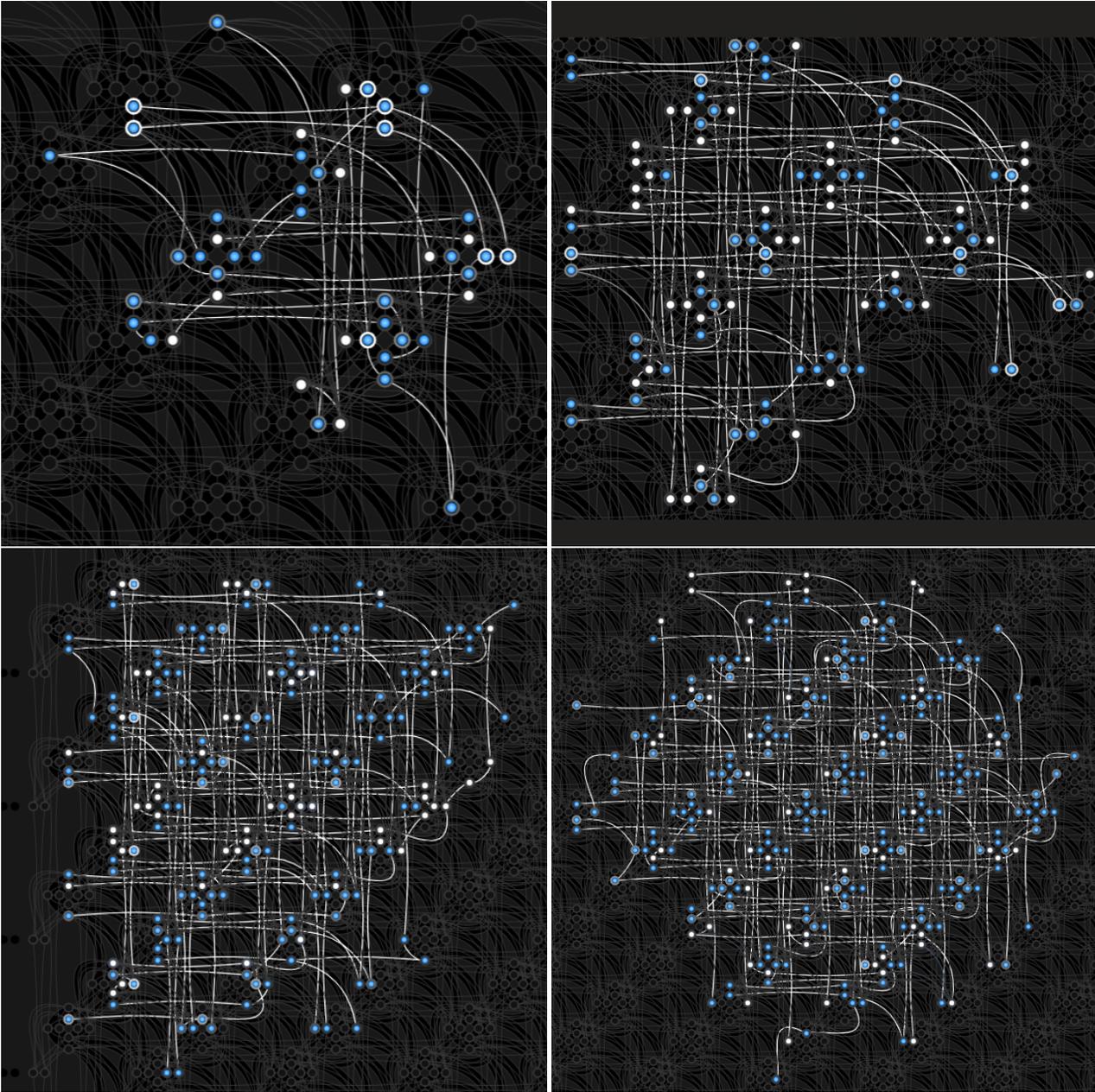


Figure 13. Visualisation of an example minor embedding on Pegasus (which we use in the main paper). The visualisation is obtained via D-Wave Leap 2's problem inspector [8] for (upper left) 20, (upper right) 30, (lower left) 40, and (lower right) 50 worst vertices. Each node depicts a physical qubit and the edges depict the chains of the minor embeddings.

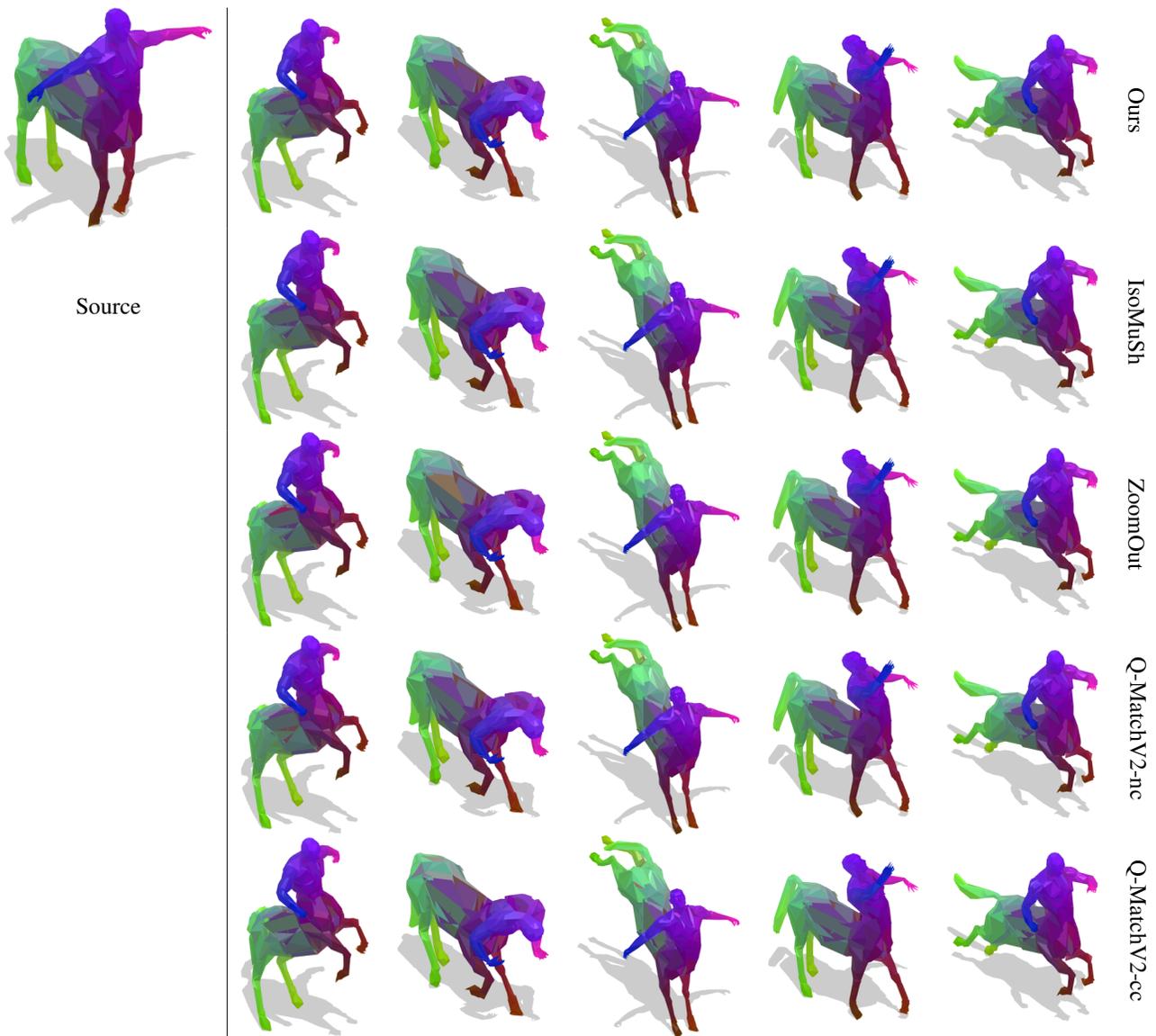


Figure 14. Qualitative results on the TOSCA [3] centaur class. We colour a source shape and transfer this colouring to target shapes via the matches estimated by our method and competitors.

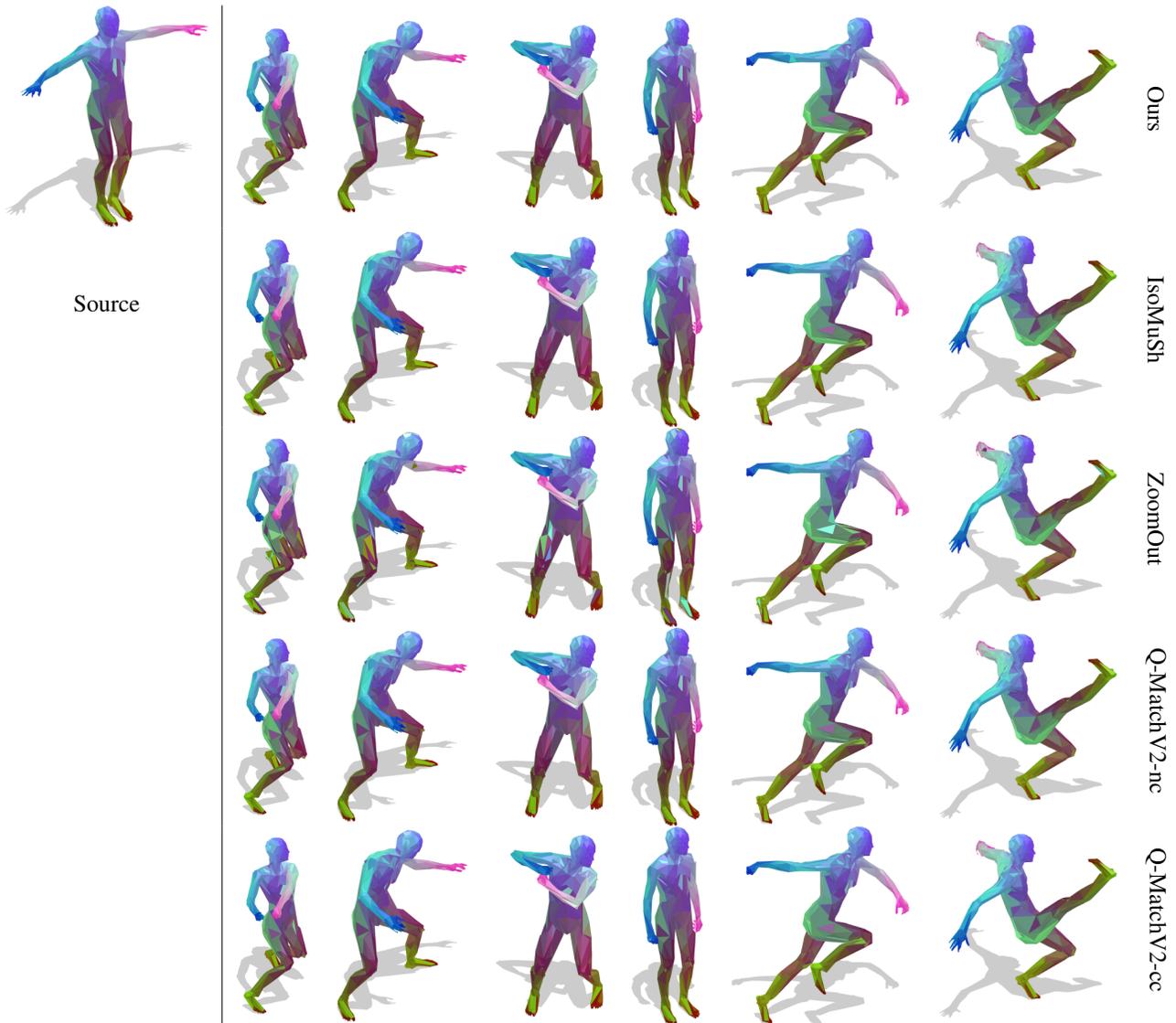


Figure 15. Qualitative results on the TOSCA [3] David class. We colour a source shape and transfer this colouring to target shapes via the matches estimated by our method and competitors.



Figure 16. Qualitative results on a subset of the SMAL dog class. We colour a source shape and transfer this colouring to target shapes via the matches estimated by our method and competitors.