Q-Match: Iterative Shape Matching via Quantum Annealing

CG-Lunch Talk, 29.04.2021

Vladislav Golyanik

max planck institut informatik
4D and Quantum Computer Vision

Overview

1) Introduction

Motivation and Contributions

2) Preliminaries

* Adiabatic quantum computing
* Related work (quantum CV)
* Alpha-Expansion
* Cyclic permutations

3) Q-Match

* Overview
* Cyclic alpha-expansion
* Algorithm
* Choosing cycles

4) Experimental Results

* Quantitative results
* Problem sizes and runtime
* Success probabilities
* Minor embeddings

Introduction

Introduction

Generic shape matching can be formulated as QAP:

$$\min_{X \in \mathbb{P}_n} \; E(X) := \mathbf{x}^{\mathrm{T}} W \mathbf{x}$$

$$\mathbf{x} \;=\; \mathrm{vec}(X)$$

$$\mathbb{P} \;\subset\; \{0,1\}^{n \times n} \text{ (permutation matrix)}$$

$$W \in \mathbb{R}^{n^2 \times n^2}$$

Introduction

Generic shape matching can be formulated as QAP:

$$\min_{X \in \mathbb{P}_n} \; E(X) := \mathbf{x}^{\mathrm{T}} W \mathbf{x}$$

$$\mathbf{x} \;=\; \mathrm{vec}(X)$$

$$\mathbb{P} \;\subset\; \{0,1\}^{n \times n} \text{ (permutation matrix)}$$

$$W \in \mathbb{R}^{n^2 \times n^2}$$

* The solution space is **exponential in *n***
* ***NP*-hard problem**; finding global optima for large inputs is unfeasible
* Allows quadratic costs for matching point pairs, regards point neighbourhoods
* Existing methods either **do not guarantee globally-optimal solutions** or have prohibitive runtime complexity

Introduction

Generic shape matching can be formulated as QAP:

$$\min_{X \in \mathbb{P}_n} \; E(X) := \mathbf{x}^{\mathrm{T}} W \mathbf{x}$$

$$\mathbf{x} \;=\; \mathrm{vec}(X)$$

$$\mathbb{P} \;\subset\; \{0,1\}^{n \times n} \text{ (permutation matrix)}$$

$$W \in \mathbb{R}^{n^2 \times n^2}$$

* The solution space is **exponential in *n***
* ***NP*-hard problem**; finding global optima for large inputs is unfeasible
* Allows quadratic costs for matching point pairs, regards point neighbourhoods
* Existing methods either **do not guarantee globally-optimal solutions** or have prohibitive runtime complexity

Our approach:

* Use a QPU to solve QAP without relaxations, while providing theoretical global optimality guarantees. It can be advantageous compared to, *e.g.*, simulated annealing.

Contributions

* **Cyclic alpha-expansions** suitable for modern AQC
* **Q-Match:** heterogeneous iterative method for QAP (including shape matching)

Contributions

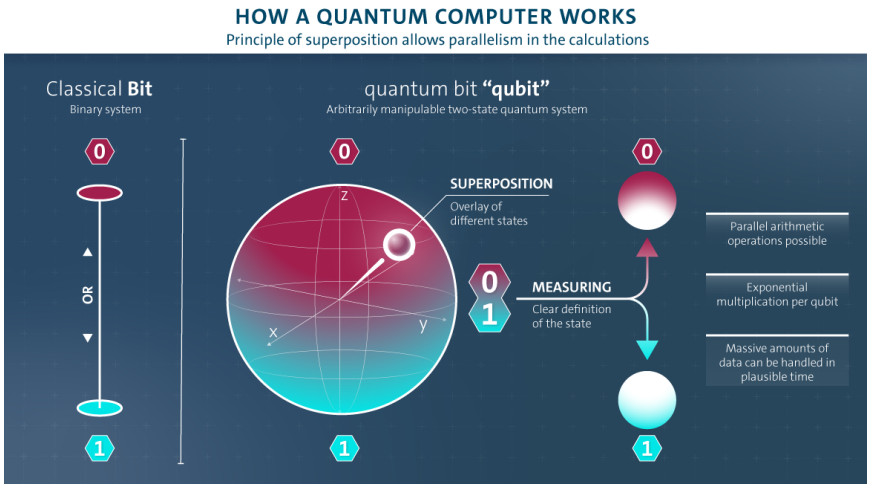
* **Cyclic alpha-expansions** suitable for modern AQC
* **Q-Match:** heterogeneous iterative method for QAP (including shape matching)

Experimental results:
* Q-Match can solve real-world problems (n = 500)
* It outperforms quantum SotA and a classical method (functional maps)
* Q-Match is evaluated on D-Wave Advantage system 1.1 (5436 qubits)
* The experiments are reported for ~25 minutes of QPU time

SotA:
* n = 7 (permutation synchronisation), n = 4 (graph matching)

Preliminaries

Adiabatic Quantum Computing



One qubit:

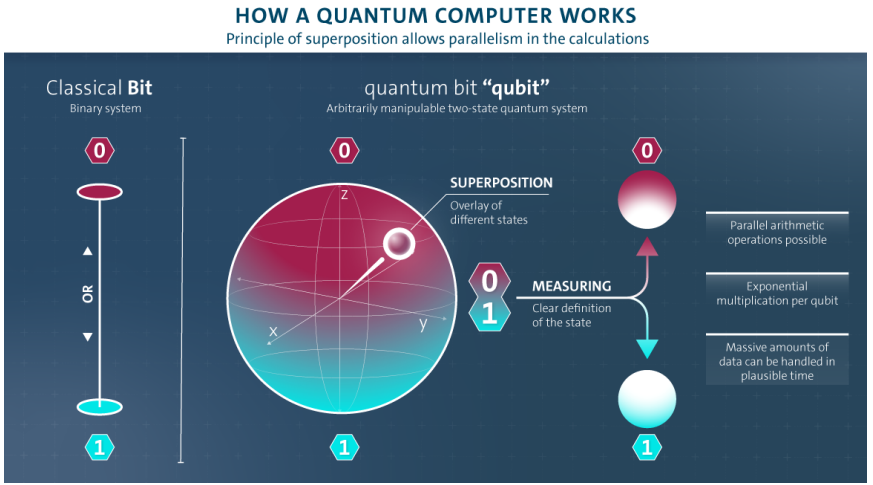
$$|\phi\rangle = \alpha|0\rangle + \beta|1\rangle \qquad \alpha, \beta \in \mathbb{C} \qquad |\alpha|^2 + |\beta|^2 = 1$$

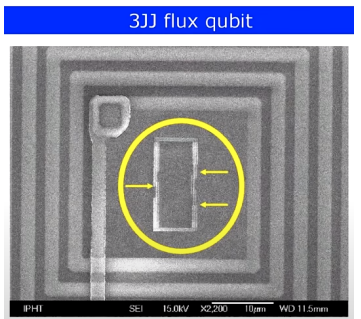
Two qubits:

$$|\psi\rangle \in \mathbb{H} \otimes \mathbb{H}$$

Images: https://www.volkswagenag.com/en/news/stories/2019/11/where-is-the-electron-and-how-many-of-them.html
Image of Advantage sys. 1.1 (qubit and QPU); D-Wave Systems

Adiabatic Quantum Computing





$_{41}$**Nb**

Superconducting flux qubit

One qubit:

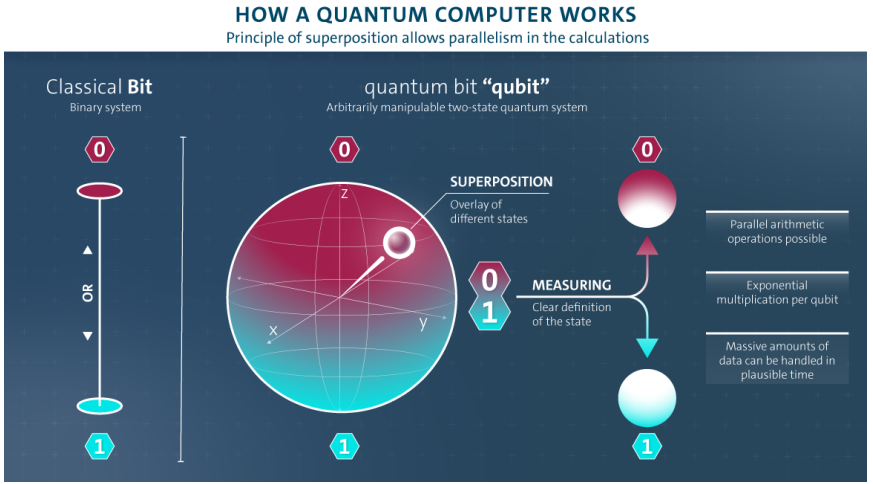
$$|\phi\rangle = \alpha|0\rangle + \beta|1\rangle \qquad \alpha, \beta \in \mathbb{C} \qquad |\alpha|^2 + |\beta|^2 = 1$$

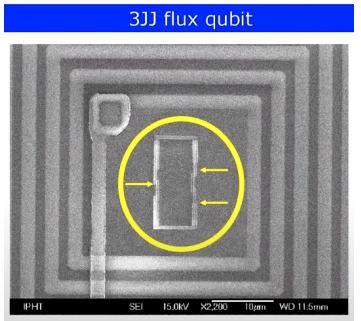
Two qubits:

$$|\psi\rangle \in \mathbb{H} \otimes \mathbb{H}$$

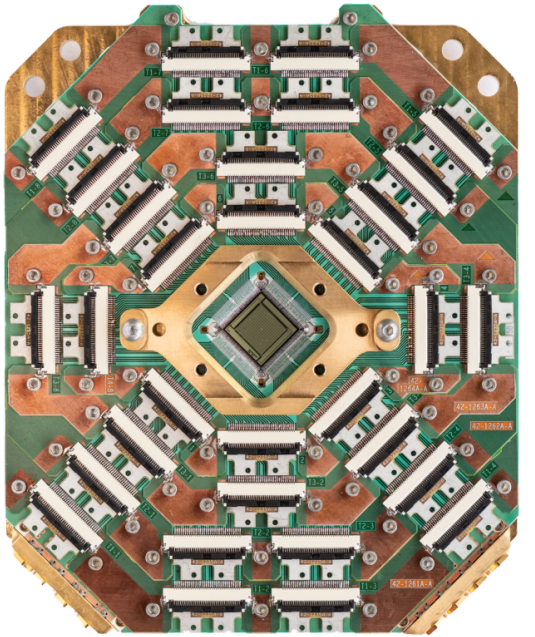
Images: https://www.volkswagenag.com/en/news/stories/2019/11/where-is-the-electron-and-how-many-of-them.html
Image of Advantage sys. 1.1 (qubit and QPU); D-Wave Systems

Adiabatic Quantum Computing





41 **Nb**



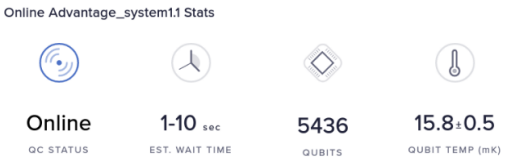
Superconducting flux qubit

One qubit:

$$|\phi\rangle = \alpha|0\rangle + \beta|1\rangle \qquad \alpha, \beta \in \mathbb{C} \qquad |\alpha|^2 + |\beta|^2 = 1$$

Two qubits:

$$|\psi\rangle \in \mathbb{H} \otimes \mathbb{H}$$



Images: https://www.volkswagenag.com/en/news/stories/2019/11/where-is-the-electron-and-how-many-of-them.html
Image of Advantage sys. 1.1 (qubit and QPU); D-Wave Systems

Adiabatic Quantum Computing

$$H(t) = \left(1 - \frac{t}{\tau}\right) H_I + \frac{t}{\tau} H_P$$

Farhi *et al.* A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem. Science, 2001.

Image: Birdal and Golyanik *et al.* Quantum Permutation Synchronization. CVPR, 2021.

Adiabatic Quantum Computing

$$H(t) = \left(1 - \frac{t}{\tau}\right) H_I + \frac{t}{\tau} H_P$$

Transition between initial and problem Hamiltonians

Initial state:

$$|\psi(t=0)\rangle = \bigotimes_{i=1}^{n} \frac{1}{\sqrt{2}} \left(|0\rangle + |1\rangle\right)$$

Solvable optimisation problem:

$$\min_{s \in \{-1,1\}^n} s^\top J s + b^\top s$$

Farhi *et al.* A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem. Science, 2001.

Image: Birdal and Golyanik *et al.* Quantum Permutation Synchronization. CVPR, 2021.

Adiabatic Quantum Computing

$$H(t) = \left(1 - \frac{t}{\tau}\right) H_I + \frac{t}{\tau} H_P$$

Transition between initial and problem Hamiltonians

Initial state:

$$|\psi(t=0)\rangle = \bigotimes_{i=1}^{n} \frac{1}{\sqrt{2}} \left(|0\rangle + |1\rangle\right)$$

Solvable optimisation problem:

$$\min_{s \in \{-1,1\}^n} s^\top J s + b^\top s$$

Every AQC algorithm includes six steps:

1) **QUBO preparation**
2) Minor embedding
3) **Quantum annealing (sampling)**
4) Unembedding
5) Bitstring selection
6) Solution interpretation

Farhi *et al.* A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem. Science, 2001.

Image: Birdal and Golyanik *et al.* Quantum Permutation Synchronization. CVPR, 2021.

Adiabatic Quantum Computing

$$H(t) = \left(1 - \frac{t}{\tau}\right) H_I + \frac{t}{\tau} H_P$$

Transition between initial and problem Hamiltonians

Initial state:

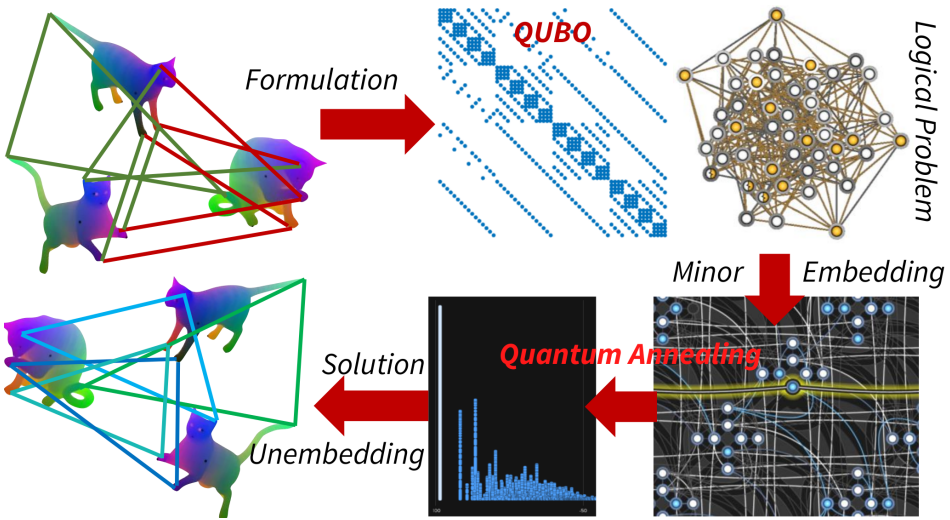
$$|\psi(t=0)\rangle = \bigotimes_{i=1}^{n} \frac{1}{\sqrt{2}} \left(|0\rangle + |1\rangle\right)$$

Solvable optimisation problem:

$$\min_{s \in \{-1,1\}^n} s^\top J s + b^\top s$$

Every AQC algorithm includes six steps:

1) **QUBO preparation**
2) Minor embedding
3) **Quantum annealing (sampling)**
4) Unembedding
5) Bitstring selection
6) Solution interpretation



Farhi *et al.* A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem. Science, 2001.
Image: Birdal and Golyanik *et al.* Quantum Permutation Synchronization. CVPR, 2021.

Quantum Computer Vision Methods

Algorithm	Problem	Processor	# qubits	total QPU time
QUBO Suppression [70]	non-maximum suppression in human tracking	2X	1000	n/a
QA [45, 46]	transformation estimation and point set alignment	2000Q	2048	60 sec.*
QGM [89]	graph matching (two graphs, up to four points)	2000Q	2048	$2 - 2.5$ min.
QuantumSync (ours)	permutation synchronization (multiple views, multiple points)	Adv. 1.1	5436	> 15 min.

Table 4. Overview of several recent quantum computer vision methods published at computer vision conferences and our *QuantumSync*. Note that the right-most column reports the overall experimental QPU runtime in the evaluation of the methods. "*": QA has been recently tested on D-Wave 2000Q; the results are reported in the supplementary document [46].

[70] Li and Ghosh. Quantum-soft QUBO Suppression for Accurate Object Detection. ECCV, 2020.
Table: Birdal and Golyanik *et al.* Quantum Permutation Synchronization. CVPR, 2021.

Quantum Computer Vision Methods

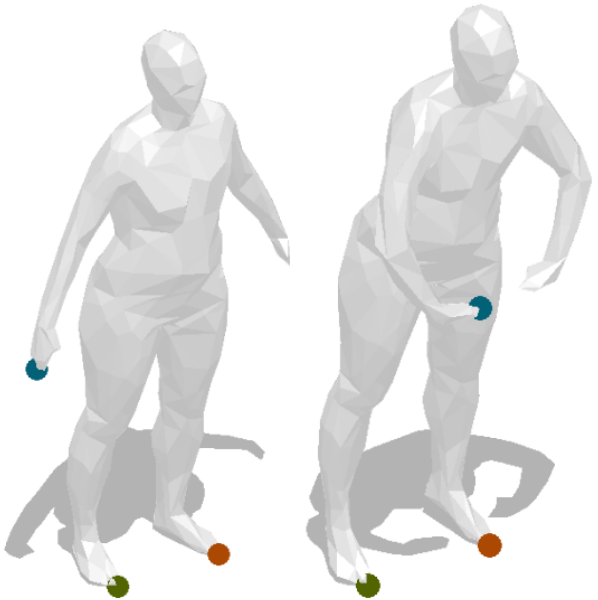
Algorithm	Problem	Processor	# qubits	total QPU time
QUBO Suppression [70]	non-maximum suppression in human tracking	2X	1000	n/a
QA [45, 46]	transformation estimation and point set alignment	2000Q	2048	60 sec.*
QGM [89]	graph matching (two graphs, up to four points)	2000Q	2048	$2 - 2.5$ min.
QuantumSync (ours)	permutation synchronization (multiple views, multiple points)	Adv. 1.1	5436	> 15 min.

Table 4. Overview of several recent quantum computer vision methods published at computer vision conferences and our *QuantumSync*. Note that the right-most column reports the overall experimental QPU runtime in the evaluation of the methods. "*": QA has been recently tested on D-Wave 2000Q; the results are reported in the supplementary document [46].

Q-Match	QAP, Shape Matching	Adv. 1.1	5436	~25 min.

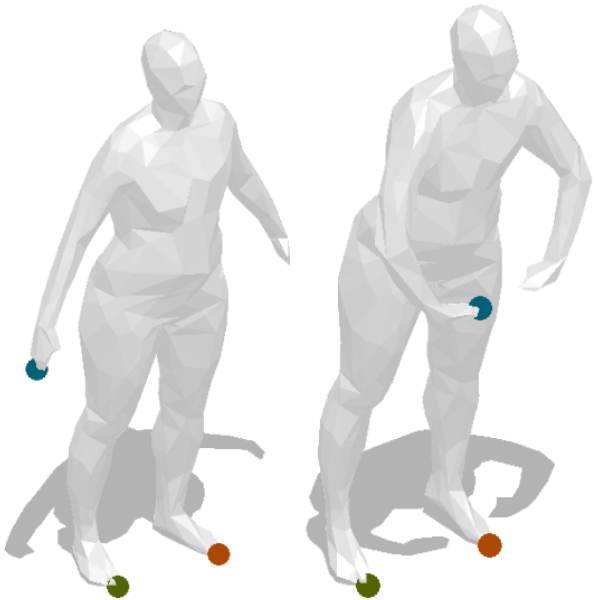
[70] Li and Ghosh. Quantum-soft QUBO Suppression for Accurate Object Detection. ECCV, 2020.
Table: Birdal and Golyanik *et al.* Quantum Permutation Synchronization. CVPR, 2021.

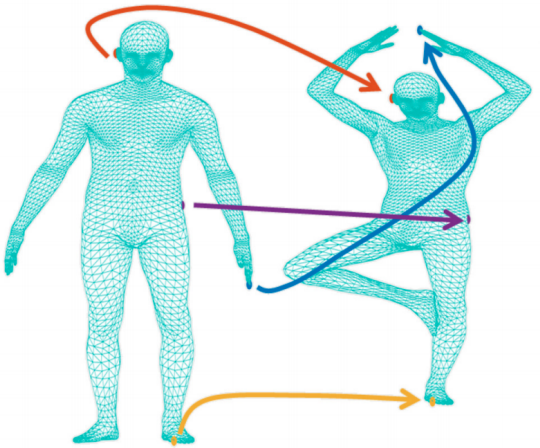
Quantum Graph Matching

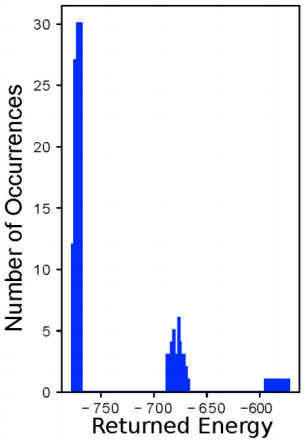


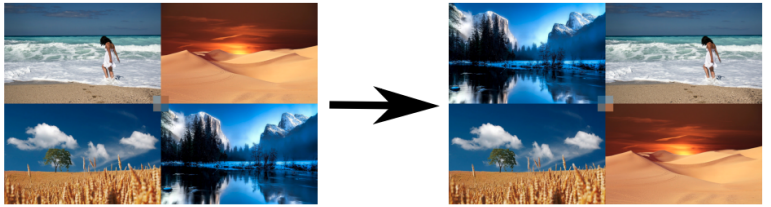
[1] Seelbach Benkner *et al.* Adiabatic Quantum Graph Matching with Permutation Matrix Constraints. 3DV, 2020.

Quantum Graph Matching









[1] Seelbach Benkner *et al.* Adiabatic Quantum Graph Matching with Permutation Matrix Constraints. 3DV, 2020.

Permutation-ness as Linear Equality Constraints

Proposition 2. *The constrained minimization:*

$$\arg\min_{\mathbf{x}\in\mathcal{B}} \mathbf{x}^\top \mathbf{Q}'\mathbf{x} \quad s.t. \quad \mathbf{A}\mathbf{x} = \mathbf{b}$$

can be turned into an (unconstrained) QUBO

$$\arg\min_{\mathbf{x}\in\mathcal{B}} \mathbf{x}^\top \mathbf{Q}\mathbf{x} + \mathbf{s}^\top \mathbf{x},$$

where $\mathbf{Q} = \mathbf{Q}' + \lambda \mathbf{A}^\top \mathbf{A}$ *and* $\mathbf{s} = -2\lambda \mathbf{A}^\top \mathbf{b}$.

$$\mathbf{A}_i = \begin{bmatrix} \mathbf{I} \otimes \mathbf{1}^\top \\ \mathbf{1}^\top \otimes \mathbf{I} \end{bmatrix}$$

[1] Seelbach Benkner *et al.* Adiabatic Quantum Graph Matching with Permutation Matrix Constraints. 3DV, 2020.
[2] Birdal and Golyanik *et al.* Quantum Permutation Synchronization. CVPR, 2021.

Permutation-ness as Linear Equality Constraints

Proposition 2. *The constrained minimization:*

$$\arg\min_{\mathbf{x}\in\mathcal{B}} \mathbf{x}^\top \mathbf{Q}'\mathbf{x} \quad s.t. \quad \mathbf{A}\mathbf{x} = \mathbf{b}$$

can be turned into an (unconstrained) QUBO

$$\arg\min_{\mathbf{x}\in\mathcal{B}} \mathbf{x}^\top \mathbf{Q}\mathbf{x} + \mathbf{s}^\top \mathbf{x},$$

where $\mathbf{Q} = \mathbf{Q}' + \lambda \mathbf{A}^\top \mathbf{A}$ *and* $\mathbf{s} = -2\lambda \mathbf{A}^\top \mathbf{b}$.

$$\mathbf{A}_i = \begin{bmatrix} \mathbf{I} \otimes \mathbf{1}^\top \\ \mathbf{1}^\top \otimes \mathbf{I} \end{bmatrix}$$

$$\begin{aligned} &\arg\min_{\{\mathbf{x}\in\{0,1\}^{n^2} \mid \quad A\mathbf{x}=\mathbf{b}\}} \mathbf{x}^\mathrm{T} W \mathbf{x} + \mathbf{c}^T \mathbf{x} \\ = &\arg\min_{\mathbf{x}\in\{0,1\}^{n^2}} \mathbf{x}^\mathrm{T} W \mathbf{x} + \mathbf{c}^T \mathbf{x} + \lambda ||A\mathbf{x} - \mathbf{b}||^2 \end{aligned}$$

Example:

$$A = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}.$$

[1] Seelbach Benkner *et al.* Adiabatic Quantum Graph Matching with Permutation Matrix Constraints. 3DV, 2020.
[2] Birdal and Golyanik *et al.* Quantum Permutation Synchronization. CVPR, 2021.

Alpha Expansion

+ an efficient algorithm to minimise $E(f) = E_{smooth}(f) + E_{data}(f).$

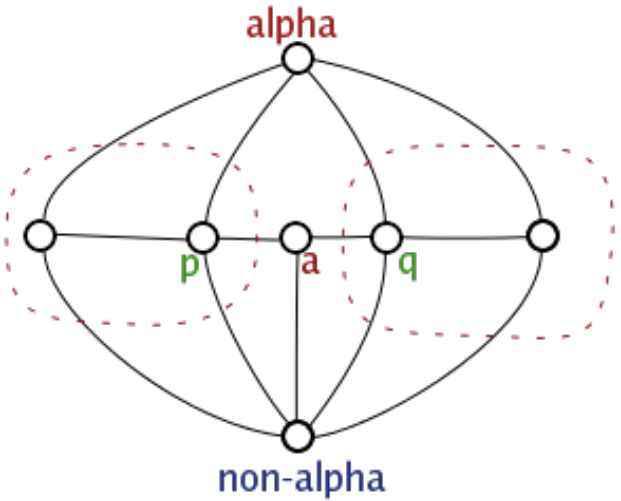
- involves pairs of pixels only
- discontinuity preserving

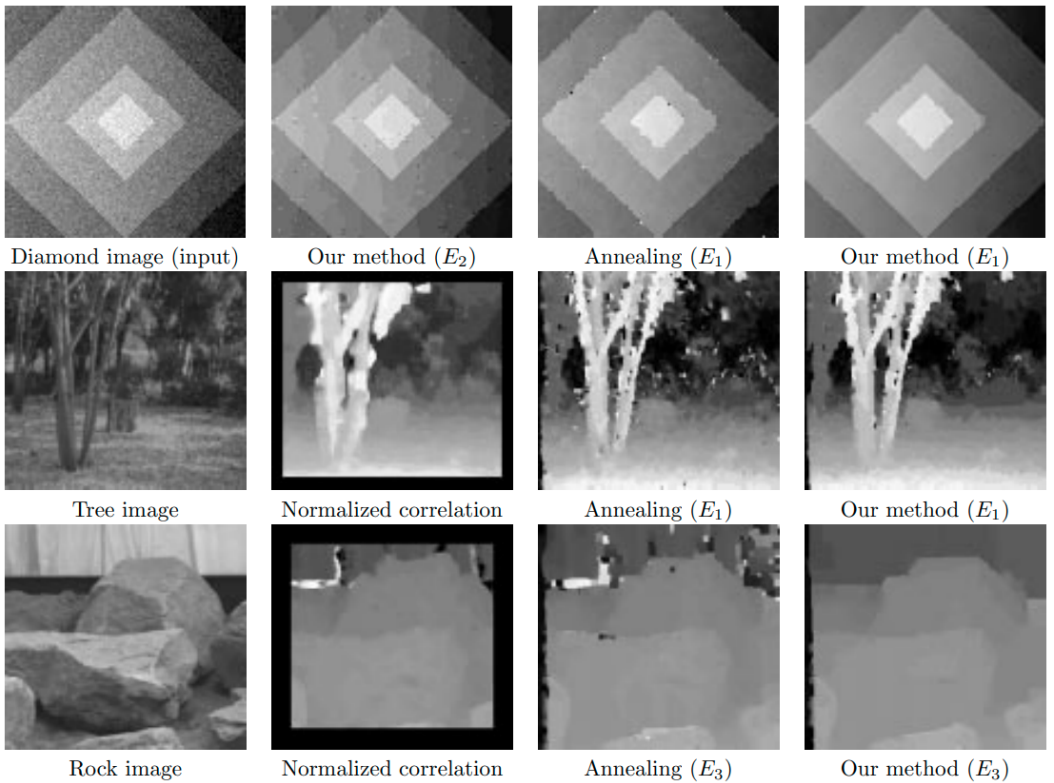
$$E_{data}(f) = \sum_{p \in \mathcal{P}} D_p(f_p)$$

+ goal: find labelling which is piecewise smooth and consistent with the data
+ for label alpha, the algorithm assigns an arbitrary number of pixels to alpha
+ generates a labelling so that there are no expansion moves decreasing the energy
+ the solution is within a known factor of the global minimum

Boykov *et al.* Fast Approximate Energy Minimization via Graph Cuts. TPAMI, 2001.

Alpha Expansion





Boykov *et al.* Fast Approximate Energy Minimization via Graph Cuts. TPAMI, 2001.

Cyclic Permutations

Permutations:

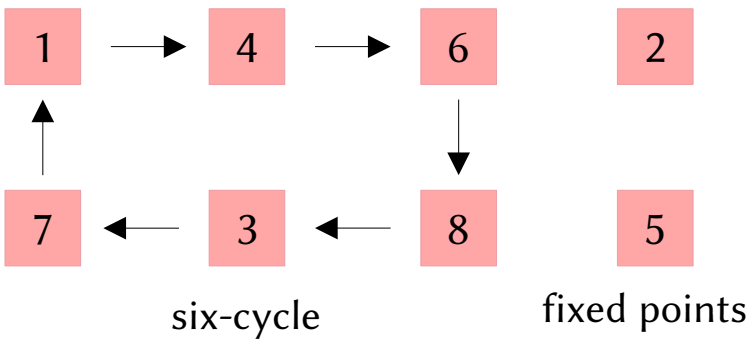
$$\mathbb{P}_n = \{X \in \{0,1\}^{n \times n} \mid \sum_i X_{ij} = 1,\ \sum_j X_{ij} = 1\ \forall i, j\}.$$

Cyclic Permutations

Permutations:

$$\mathbb{P}_n = \{X \in \{0,1\}^{n \times n} \mid \sum_i X_{ij} = 1, \ \sum_j X_{ij} = 1 \ \forall i, j\}.$$

k-cycles:

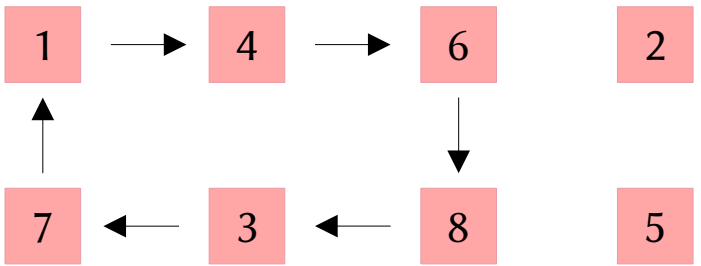


Cyclic Permutations

Permutations:

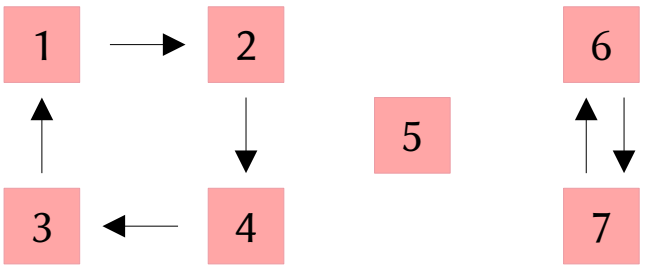
$$\mathbb{P}_n = \{X \in \{0,1\}^{n \times n} \mid \sum_i X_{ij} = 1,\ \sum_j X_{ij} = 1\ \forall i, j\}.$$

k-cycles:



six-cycle fixed points

Disjoint permutations commute:



four-cycle fixed point two-cycle

Any X can be written as $X = \prod_{i=0}^{N} c_i$, i.e., a product of 2-cycles (or, generally, disjoint *k-cycles*).

Q-Match

The Proposed Method :: Q-Match

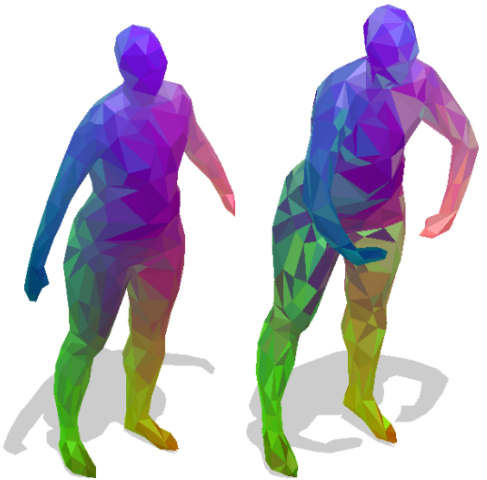
Given: 3D shapes M and N,

both discretised with n vertices.

$$W_{i\cdot n+k,j\cdot n+l} = |d^g_M(i,j) - d^g_N(k,l)|$$

+ Geometric meaning of $d^g(a,b)$ influences the structure of QAP

Find: optimal P



M N

Q-Match :: The Main Idea

Want to solve but cannot:

$$\min_{X \in \mathbb{P}_n} E(X) := \mathbf{x}^{\mathrm{T}} W \mathbf{x} \qquad W_{i \cdot n + k, j \cdot n + l} = |d_M^g(i,j) - d_N^g(k,l)|$$

Q-Match :: The Main Idea

Want to solve but cannot:

$$\min_{X \in \mathbb{P}_n} E(X) := \mathbf{x}^{\mathrm{T}} W \mathbf{x} \qquad W_{i \cdot n + k, j \cdot n + l} = |d_M^g(i,j) - d_N^g(k,l)|$$

Instead solve

$$\arg\min_{\{P \in \mathbb{P}_n \mid \exists \alpha \in \{0,1\}^m :\, P = \left(\prod_i c_i^{\alpha_i}\right) P_0\}} E(P)$$

(cyclic alpha-expansion)

Q-Match :: The Main Idea

Want to solve but cannot:

$$\min_{X \in \mathbb{P}_n} E(X) := \mathbf{x}^{\mathrm{T}} W \mathbf{x} \qquad W_{i \cdot n + k, j \cdot n + l} = |d_M^g(i,j) - d_N^g(k,l)|$$

Instead solve

$$\underset{\{P \in \mathbb{P}_n \mid \exists \alpha \in \{0,1\}^m :\, P = \left(\prod_i c_i^{\alpha_i}\right) P_0\}}{\arg\min} E(P)$$

(cyclic alpha-expansion)

... leading to

$$\min_{\alpha \in \{0,1\}^m} \alpha^\top \tilde{W} \alpha \qquad \tilde{W}_{ij} = \begin{cases} E(C_i, C_j) & \text{if } i \neq j, \\ E(C_i, C_i) + E(C_i, P_0) + E(P_0, C_j) & \text{otherwise.} \end{cases}$$

$$E(Q,R) = \mathrm{vec}(Q)^T W \mathrm{vec}(R)$$

$$P(\alpha) = P_0 + \sum_{i=1}^{m} \alpha_i \frac{(c_i - I) P_0}{C_i}$$

Q-Match :: The Main Idea

Want to solve but cannot:

$$\min_{X \in \mathbb{P}_n} E(X) := \mathbf{x}^{\mathrm{T}} W \mathbf{x} \qquad W_{i \cdot n + k, j \cdot n + l} = |d_M^g(i,j) - d_N^g(k,l)|$$

Instead solve

$$\arg\min_{\{P \in \mathbb{P}_n \mid \exists \alpha \in \{0,1\}^m :\, P = \left(\prod_i c_i^{\alpha_i}\right) P_0\}} E(P)$$

(cyclic alpha-expansion)

... leading to

not submodular

$$\min_{\alpha \in \{0,1\}^m} \alpha^\top \tilde{W} \alpha \qquad \tilde{W}_{ij} = \begin{cases} E(C_i, C_j) & \text{if } i \neq j, \\ E(C_i, C_i) + E(C_i, P_0) + E(P_0, C_j) & \text{otherwise.} \end{cases}$$

$$E(Q,R) = \mathrm{vec}(Q)^T W \mathrm{vec}(R)$$

$$P(\alpha) = P_0 + \sum_{i=1}^{m} \alpha_i \frac{(c_i - I) P_0}{C_i}$$

Cyclic Alpha-Expansion

Assume $C = \{c_1, ..., c_m\}$ is a set of disjoint cycles.

Cyclic Alpha-Expansion

Assume $C = \{c_1, \ldots, c_m\}$ is a set of disjoint cycles.

Consider

$$\operatorname*{arg\,min}_{\{P \in \mathbb{P}_n \mid \exists \alpha \in \{0,1\}^m : P = \left(\prod_i c_i^{\alpha_i}\right) P_0\}} E(P)$$

binary vector parametrising P

initial permutation

Cyclic Alpha-Expansion

Assume $C = \{c_1, \ldots, c_m\}$ is a set of disjoint cycles.

Consider

$$\operatorname*{arg\,min}_{\{P \in \mathbb{P}_n \mid \exists \alpha \in \{0,1\}^m :\ P = \left(\prod_i c_i^{\alpha_i}\right) P_0\}} E(P)$$

binary vector parametrising P

initial permutation

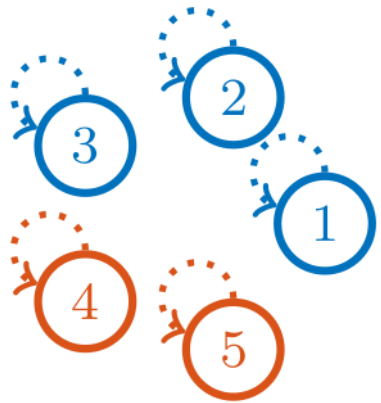
+ This optimisation problem decides for each cycle whether it is applied or not.
+ Its complexity depends on the number of cycles and not *n*.

Q-Match :: The Main Idea

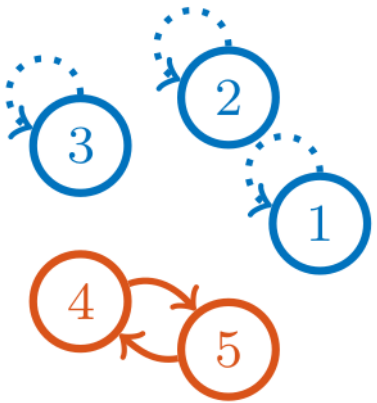
Given two cycles C_1, C_2 we parametrize all combinations with two binary variables α_1, α_2

$$\begin{pmatrix} 1-\alpha_1 & 0 & \alpha_1 & 0 & 0 \\ \alpha_1 & 1-\alpha_1 & 0 & 0 & 0 \\ 0 & \alpha_1 & 1-\alpha_1 & 0 & 0 \\ 0 & 0 & 0 & 1-\alpha_2 & \alpha_2 \\ 0 & 0 & 0 & \alpha_2 & 1-\alpha_2 \end{pmatrix}$$

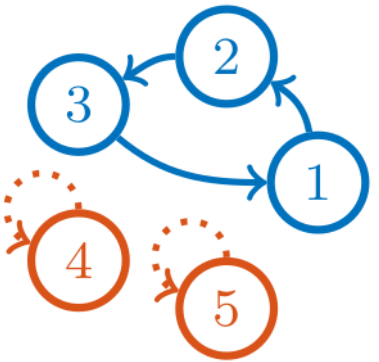
Possible permutations for all choices of α_1, α_2:



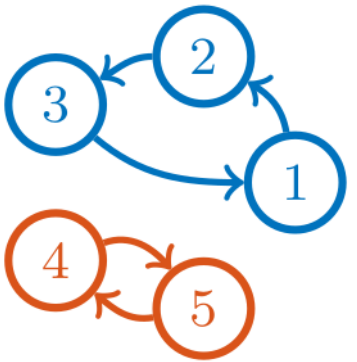
$\alpha_1 = 0, \alpha_2 = 0$



$\alpha_1 = 0, \alpha_2 = 1$



$\alpha_1 = 1, \alpha_2 = 0$



$\alpha_1 = 1, \alpha_2 = 1$

Q-Match :: The Main Idea

$$\min_{X \in \mathbb{P}_n} \; E(X) := \mathbf{x}^{\mathrm{T}} W \mathbf{x}$$

W

Initial QAP formulation; cannot be solved on QPU

initial matrix of costs; large; cannot be precomputed and stored; its entries are computed on demand in each iteration

Q-Match :: The Main Idea

$$\min_{X \in \mathbb{P}_n} \; E(X) := \mathbf{x}^{\mathsf{T}} W \mathbf{x}$$

Initial QAP formulation; cannot be solved on QPU

W

initial matrix of costs; large; cannot be precomputed and stored; its entries are computed on demand in each iteration

$$\min_{\alpha \in \{0,1\}^m} \; \alpha^{\top} \tilde{W} \alpha$$

QUBO formulation based on cyclic alpha-expansion; can be solved on QPU

$\tilde{W}$

matrix of QUBO costs; requires known W

Q-Match :: The Main Idea

$$\min_{X \in \mathbb{P}_n} E(X) := \mathbf{x}^{\mathsf{T}} W \mathbf{x}$$

Initial QAP formulation; cannot be solved on QPU

W

initial matrix of costs; large; cannot be precomputed and stored; its entries are computed on demand in each iteration

$$\min_{\alpha \in \{0,1\}^m} \alpha^{\top} \tilde{W} \alpha$$

QUBO formulation based on cyclic alpha-expansion; can be solved on QPU

$\tilde{W}$

matrix of QUBO costs; requires known W

W_s

a $k^2 \times k^2$ reduction of W based on *k* worst matches

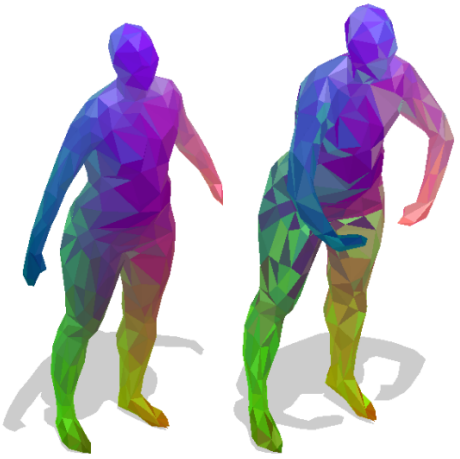
Q-Match

Initialise P_0 via descriptor-based similarity

repeat until converged

obtain I_M and I_N and choose from them a set of *k* random and disjoint 2-cycles

construct a submatrix of worst matches W_s



M N

Q-Match

Initialise P_0 via descriptor-based similarity

repeat until converged

obtain I_M and I_N and choose from them a set of *k* random and disjoint 2-cycles

construct a submatrix of worst matches W_s

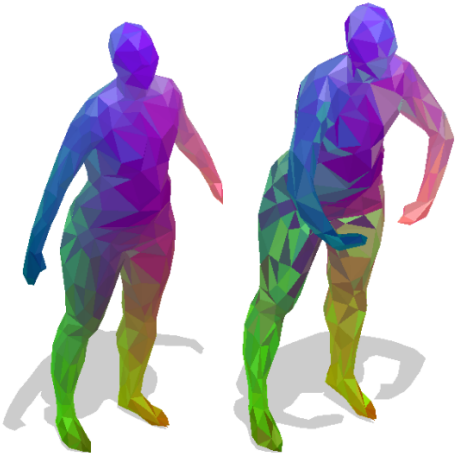
repeat until every 2-cycle occurred

choose a random set of 2-cycles

calculate $\tilde{W}_s$ and solve $\min_{\alpha \in \{0,1\}^m} \alpha^\top \tilde{W} \alpha$ on QPU

$P_i = \left(\prod_j c_j^{\alpha_j} \right) P_{i-1}$

apply the obtained permutation to worst matches



M N

Q-Match

Initialise P_0 via descriptor-based similarity

repeat until converged

obtain I_M and I_N and choose from them a set of *k* random and disjoint 2-cycles

construct a submatrix of worst matches W_s

repeat until every 2-cycle occurred

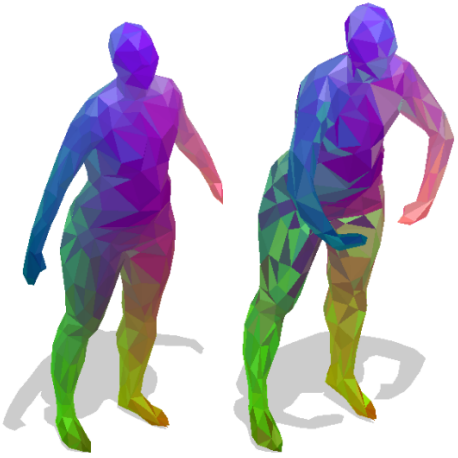
choose a random set of 2-cycles

calculate $\tilde{W}$ and solve $\min_{\alpha \in \{0,1\}^m} \alpha^\top \tilde{W} \alpha$ on QPU

$P_i = \left(\prod_j c_j^{\alpha_j} \right) P_{i-1}$

apply the obtained permutation to worst matches

NP-hard; decides to apply c_i or not



M N

Q-Match :: Choosing Cycles

* Observation: entries of QAP are highly correlated (isometric shape matching)
  ⟶ target explicitly points with high energy scores
  ⟶ the test is based on detection of point mapping inconsistencies

We choose 2-cycles:

Lemma 4.1. *Every permutation P can be written as $P = QR$, where Q and R are products of disjoint 2-cycles.*

Q-Match :: Choosing Cycles

* Observation: entries of QAP are highly correlated (isometric shape matching)
  ⟶ target explicitly points with high energy scores
  ⟶ the test is based on detection of point mapping inconsistencies

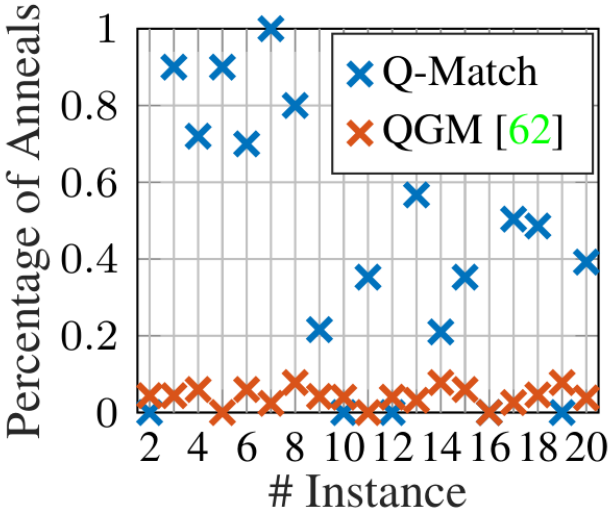
We choose 2-cycles:

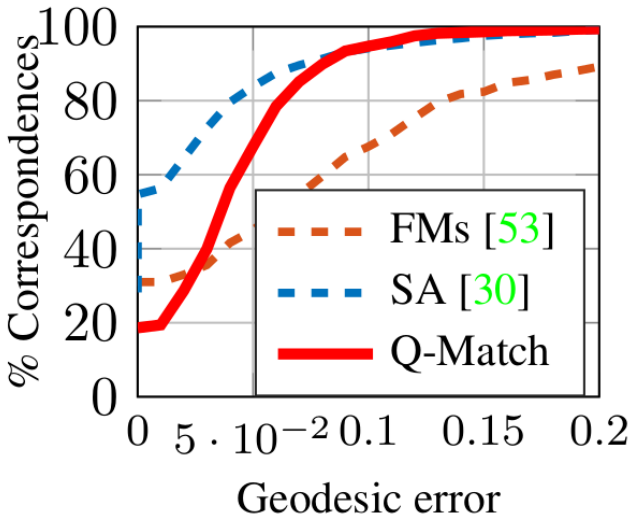
Lemma 4.1. *Every permutation P can be written as $P = QR$, where Q and R are products of disjoint 2-cycles.*

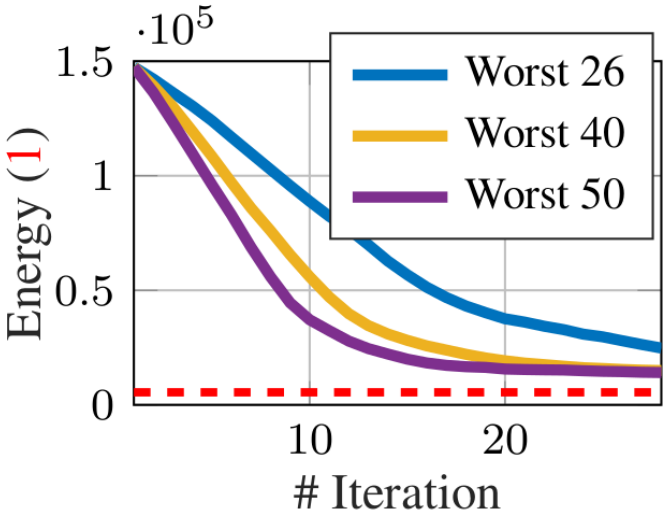
see pre-print for the proof

Experimental Results

Experimental Results





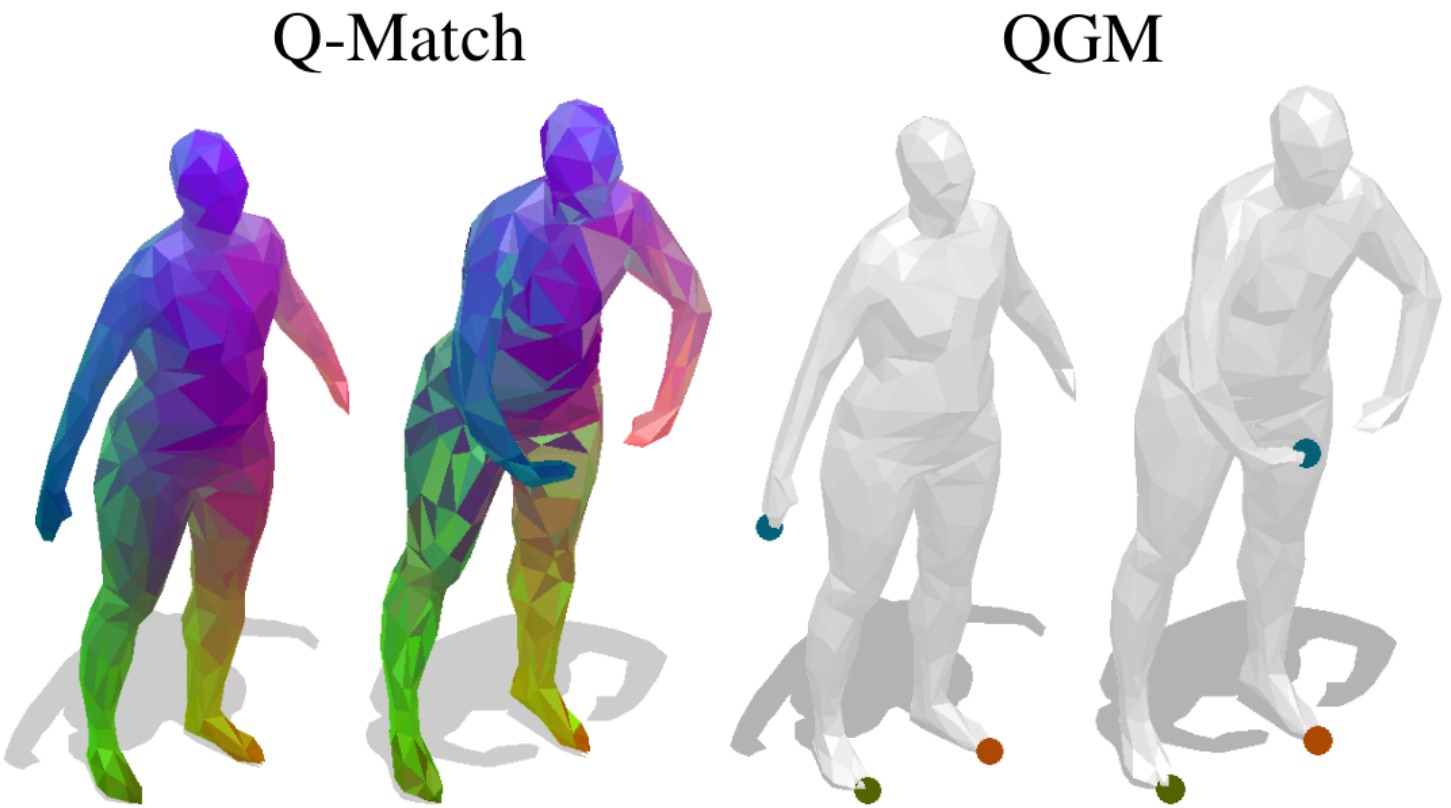


Success rates on 20 random problems.

Cumulative error (left) and convergence (right) on FAUST.

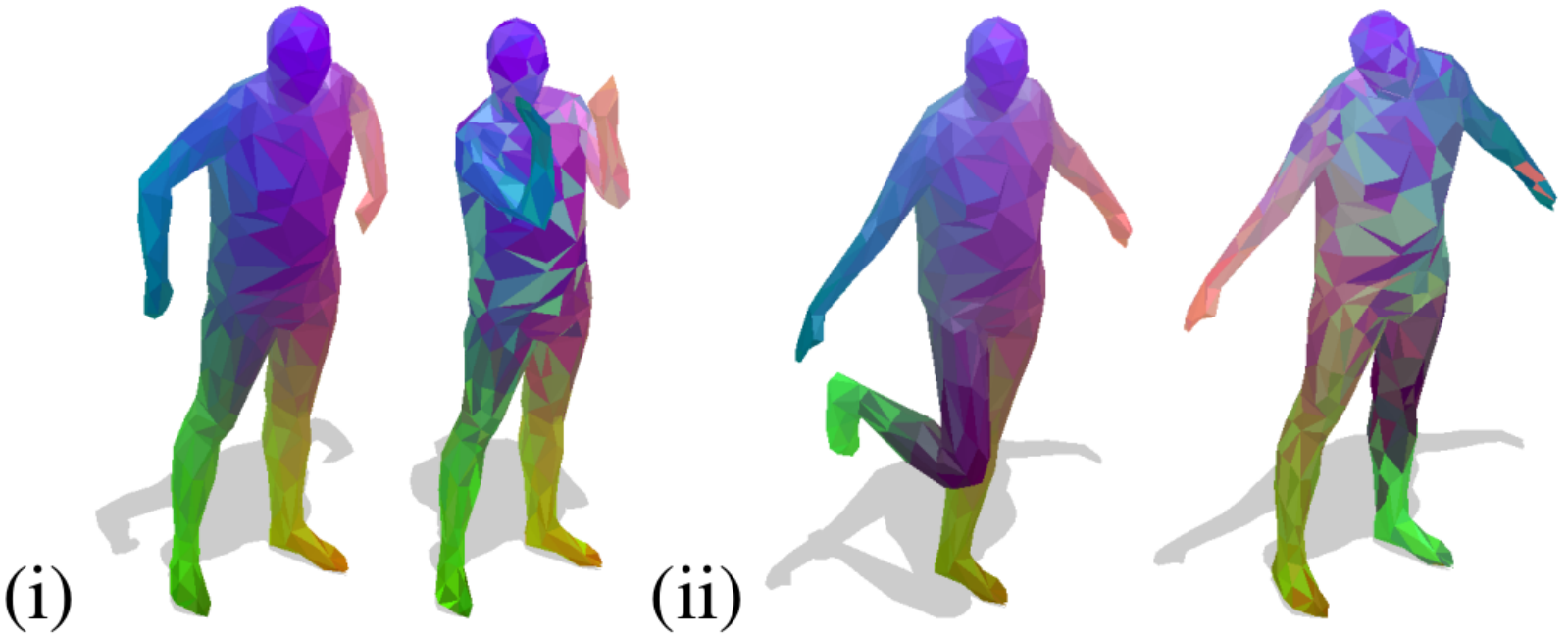
[30] Holzschuh *et al.* Simulated annealing for 3d shape correspondence. 3DV, 2020.
[53] Ovsjanikov *et al.* Functional maps: a flexible representation of maps between shapes. SIGGRAPH, 2012.

Experimental Results



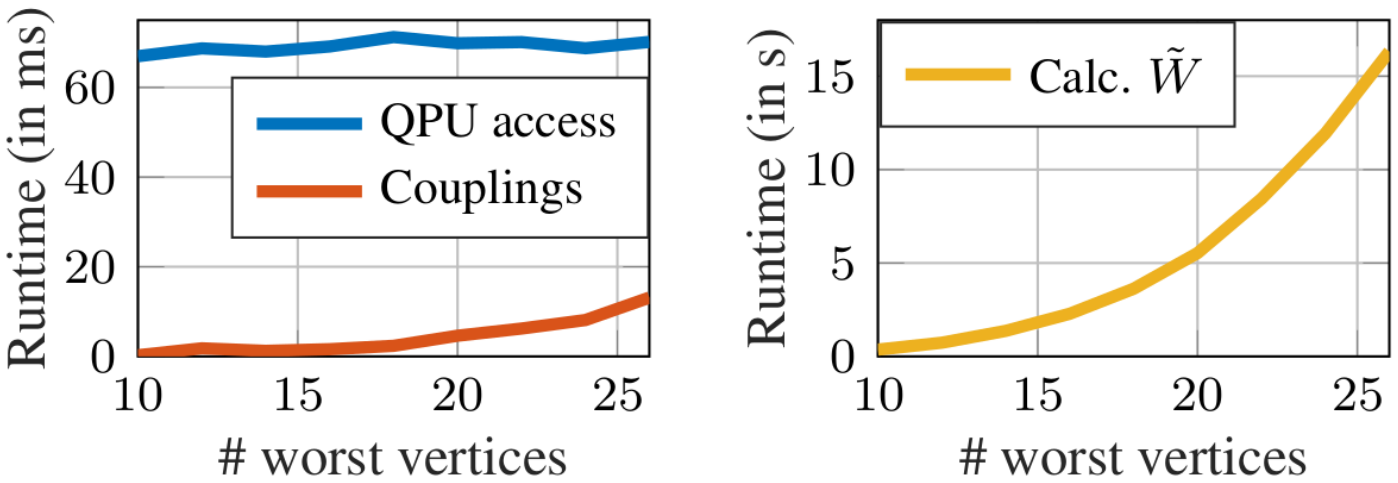
QGM: Seelbach Benkner *et al.* Adiabatic Quantum Graph Matching with Permutation Matrix Constraints. 3DV, 2020.

Experimental Results



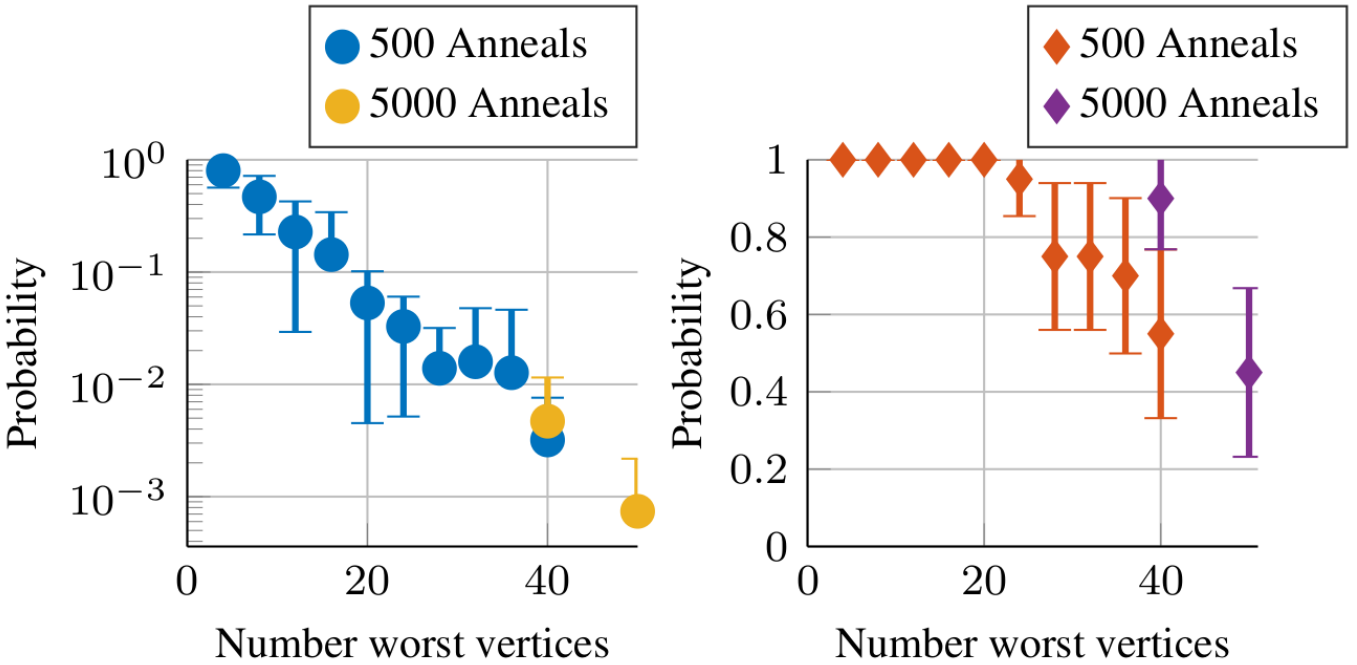
Example correspondences from the FAUST registrations.

Experimental Results



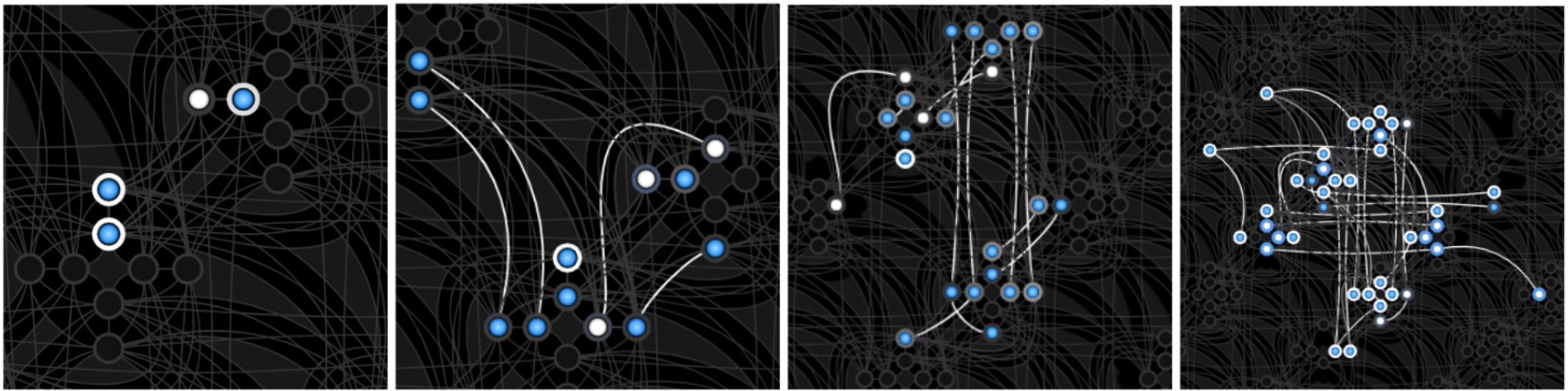
Influence of the problem size on the runtime.

Experimental Results



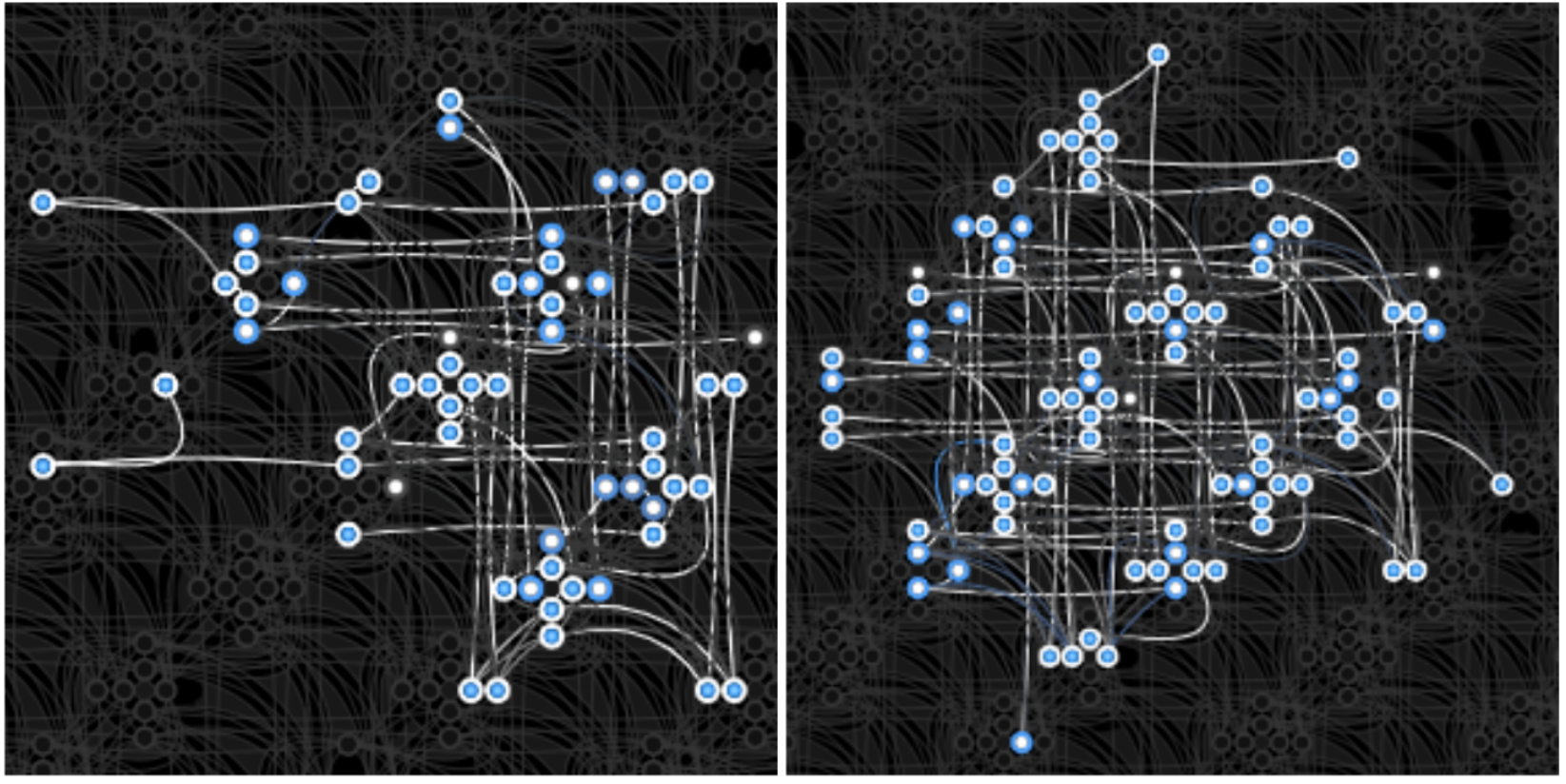
Success probability (left) and the fraction of executions where the best solution is the optimum (right).

Experimental Results



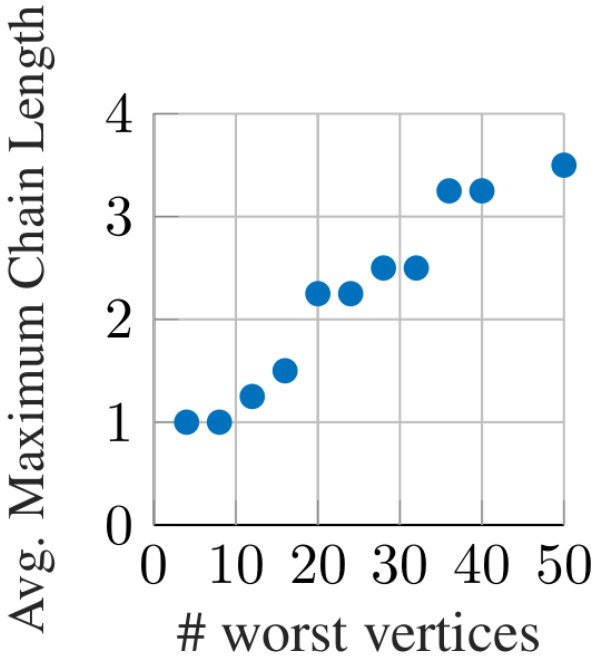
Minor embeddings for the variants with 8, 16, 24, 32 worst vertices, respectively.

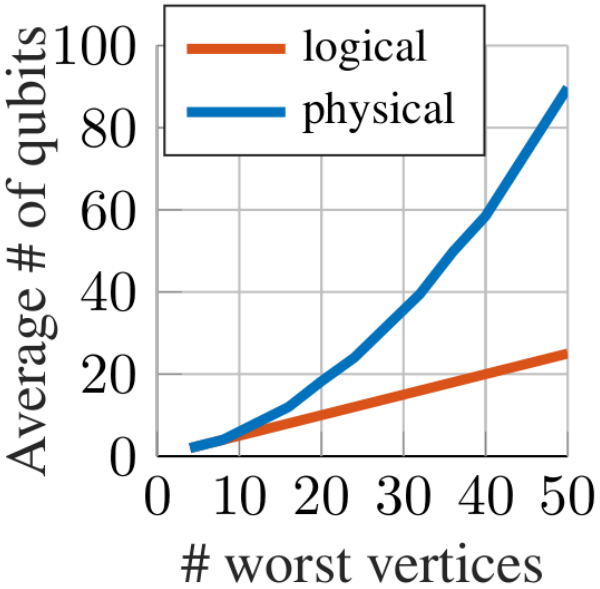
Experimental Results



Minor embeddings for the variants with 40 and 50 worst vertices, respectively.

Experimental Results





Conclusions and Next Steps

* Q-Match can solve QAP of sizes encountered in practical applications
  ⟶ claimed for the first time for a quantum method (CV/CG literature)
* No explicit permutation constraints and the iterative approach allow us to outperform previous quantum SotA
  ⟶ both *in terms of problem sizes and success probabilities*
* Q-Match outperforms a classical method

Conclusions and Next Steps

* Q-Match can solve QAP of sizes encountered in practical applications
  ⟶ claimed for the first time for a quantum method (CV/CG literature)
* No explicit permutation constraints and the iterative approach allow us to outperform previous quantum SotA
  ⟶ both *in terms of problem sizes and success probabilities*
* Q-Match outperforms a classical method

Next steps:
⟶ Apply the same principles to multi-shape matching

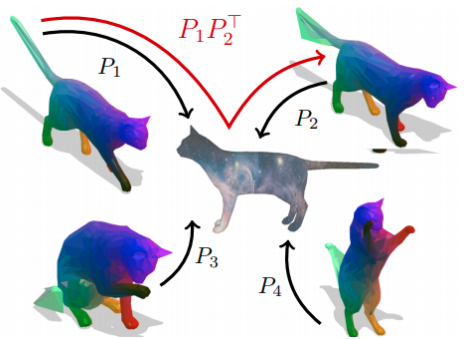


Image: Gao *et al.* Isometric Multi-Shape Matching. CVPR, 2021.

Co-Authors

Marcel

Zorah

Christof

Christian

Michael

UNIVERSITÄT SIEGEN

max planck institut informatik

Thank You!

Questions?