An Iterative Quantum Approach for Transformation Estimation from Point Sets

Natacha Kuete Meli Florian Mannel Jan Lellmann
Institute of Mathematics and Image Computing, University of Luebeck
https://www.mic.uni-luebeck.de

Abstract

We propose an iterative method for estimating rigid transformations from point sets using adiabatic quantum computation. Compared to existing quantum approaches, our method relies on an adaptive scheme to solve the problem to high precision, and does not suffer from inconsistent rotation matrices. Experimentally, our method performs robustly on several 2D and 3D datasets even with high outlier ratio.

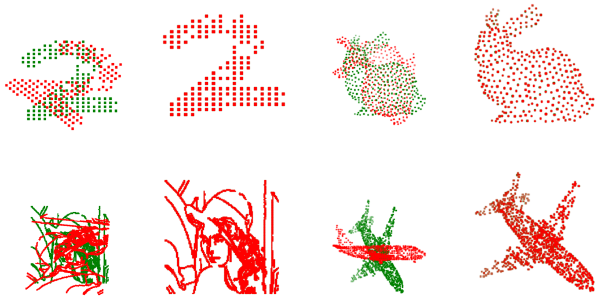
1. Introduction and Related Work

The advent of quantum computing (QC) offers a completely new computing paradigm taking advantage of principles of quantum mechanisms, with the promise of exponential acceleration of selected problems, as well as a considerable reduction in resources such as data storage space [12, 25, 31, 36]. Intuitively, leveraging the fact that quantum systems can assume mixed states, essentially existing in several pure states simultaneously, allows for computations to act on all these states simultaneously. This effect, known as quantum parallelism, sets quantum computers apart from their classical analogues, which can only perform sequential calculations [28].

Adiabatic quantum computation (AQC), which is a subfield of QC, has emerged as a promising approach for approximately solving notoriously difficult combinatorial problems, say NP-hard on a classical computers [21, 22]. One of the problem classes generally addressed by AQC optimization algorithms are the so-called quadratic unconstrained binary optimization (QUBO) problems, which have the form

$$\min_{q\in\{0,1\}^n} q^\top W q + c^\top q, \tag{1}$$

where W is a *coupling matrix* and c is a vector of *biases*. The encoding of practical optimization problems in the form of a QUBO problem is generally the first and the most crucial step for developing AQC-based algorithms. This step is followed by the embedding of the problem into the quantum hardware following techniques such as [8, 9].



(a) 2D (b) 3D

Figure 1. Results of our iterative quantum approach for transformation estimation (IQT) from **(a)** 2D point sets (MNIST [26] and Lena [20]) and **(b)** 3D point sets (Stanford bunny [30] and completion3D [35]). Green points represent reference points and red points represent template points. The initial alignment is shown on the left and the result of the registration on the right.

The problem is subsequently solved on the quantum hardware, the solution bit-string q is un-embedded, and the solution of the original problem is decoded. QUBO-based algorithms have already been developed for some computer vision problems such as matching problems [2, 3, 5], that naturally deal with binary variables. In this work, we will focus on a related problem, the one of aligning point sets.

Determining the best rigid transformation that aligns two coordinate systems based on corresponding landmarks is known as *transformation estimation (TE)* [11, 15, 16]. It finds practical application in computer vision fields such as robotics and image processing. For instance, in image registration, one of the preparatory key steps consists in roughly aligning the images to be registered based on corresponding landmarks [23]. Certain properties are desirable from a method for solving the TE problem. They include, among others, the ability of the method to accurately model the transformation that aligns the point sets, the ability to handle high dimensional and size increasing data, and the robustness against real data challenges such as noise and outliers [11].

Whereas computer vision abounds with powerful classical algorithms to solve the TE problem, *e.g.*, ICP [4, 37]

and CPD [24], there are still very few methods available for quantum computers. Golyanik and Theobalt [14] pioneered a quantum approach *(QA)* for solving the 2D transformation estimation problem based on approximating rotation matrices by linear combinations of basis matrices, where the linear coefficients are binary variables computed by AQC. Among the limitations of QA are that its accuracy is fixed, that it cannot handle noise, and that it is difficult to extend to 3D due to the increasing number of degrees of freedom and, consequently, of required qubits. Moreover, the matrix that it generates is usually not orthogonal and the algorithm does not ensure that it is close to an orthogonal matrix, either.

In this work, we build on [14] to propose a new method that can solve the transformation estimation problem on point sets in 2D and 3D to any desired accuracy even in the presence of noise, and whose matrix converges to an orthogonal matrix as the accuracy increases.

The **main contributions** of this work are as follows:

- Based on a K-bit binary representation, we propose an approximation scheme for the special orthogonal group in order to estimate rotation matrices using adiabatic quantum computing (Section 3.1).
- We use the approximation scheme to derive, first, a QUBO formulation of the transformation estimation problem (Section 3.2) and, second, an algorithm that adapts the approximation in such a way that its error vanishes in practice (Section 3.3).
- We experimentally validate the algorithm by registering 2D and 3D data sets using the D-Wave quantum annealer (Section 4). Some of the results are shown in Fig. 1.

2. Mathematical Preliminaries

2.1. Transformation Estimation from Point Sets

Let $N, D \in \mathbb{N}$. Given a *reference set* $\mathcal{X} = \{x_i\}_{i=1}^N$ and a *template set* $\mathcal{Y} = \{y_i\}_{i=1}^N$ of D-dimensional points, the purpose of transformation estimation (TE) is to determine a rigid transformation $f^* : \mathbb{R}^D \to \mathbb{R}^D$ that solves

$$\min_{f \in SE(D)} d(\mathcal{X}, f(\mathcal{Y})), \tag{RP}$$

where d is an appropriate similarity measure and $SE(D)$ is the D-dimensional special Euclidean group. The latter is formed by combinations of rotations and translations; its degrees of freedom are usually calculated as $P = D(D+1)/2$. Any transformation $f \in SE(D)$ can be represented in the form $f(y) = Ry + t$, where $R \in \mathbb{R}^{D,D}$ is a rotation matrix and $t \in \mathbb{R}^D$ is a translation vector. Choosing for d the least-squares distance yields

$$d(\mathcal{X}, f(\mathcal{Y})) = \sum_{i=1}^{N} \|x_i - f(y_i)\|_2^2 = \sum_{i=1}^{N} \|x_i - Ry_i - t\|_2^2 . \tag{2}$$

It is well-known [11, 16] that by introducing mass-point centering $\tilde{\mathcal{X}} = \mathcal{X} - \bar{\mathcal{X}}$ and $\tilde{\mathcal{Y}} = \mathcal{Y} - \bar{\mathcal{Y}}$, with $\bar{\mathcal{X}}$ and $\bar{\mathcal{Y}}$ denoting the centroids of the original point sets, the rigid registration problem reduces to finding the best rotation matrix R^*, i.e., the solution of

$$\min_{R \in SO(D)} \sum_{i=1}^{N} \|\tilde{x}_i - R\tilde{y}_i\|_2^2 . \tag{RPR}$$

This reduces the transformation space of the problem to the D-dimensional special orthogonal group $SO(D)$ with exactly $P = D(D-1)/2$ degrees of freedom. The optimal translation t^* is then deducted from R^* by $t^* = \bar{\mathcal{X}} - R^*\bar{\mathcal{Y}}$.

2.2. Adiabatic Quantum Computation

Adiabatic quantum computation (AQC) is an optimization paradigm which relies on the adiabatic theorem [1, 17] saying that the lowest-energy state, called *ground state,* of a mechanical system is the solution of an optimization problem [18]. Theoretically, the evolution of a quantum system of $n \in \mathbb{N}$ particles at a time $t \in [0, T]$, typically re-scaled to $s = t/T \in [0, 1]$, can be described by a *Hamiltonian* $H(s)$ on a Hilbert space $\mathcal{H} = \mathbb{C}^{\otimes 2^n}$, with the *state* of the system given by a unit vector $|\psi(s)\rangle \in \mathcal{H}$.

Qubits. A qubit (quantum-bit) $|\psi\rangle$ is the fundamental unit of quantum information [25, 31]. It is the state of a one-particle quantum system expressed as a superposition of a ground state $|0\rangle$ and an excited state $|1\rangle$ according to $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, with $|\alpha|^2 + |\beta|^2 = 1$, $\alpha, \beta \in \mathbb{C}$. The states $|0\rangle = (1, 0)^\top$ and $|1\rangle = (0, 1)^\top$ are the basis states of the system called *eigenstates* or *stationary states.* The reason for this is that the qubit $|\psi\rangle$ irreversibly collapses to $|0\rangle$ or $|1\rangle$ with probability $|\alpha|^2$ or $|\beta|^2$ when it is measured. We denote by $|+\rangle$ the uniform superposition state, i.e., $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$. This definition of the state vector extends to an n-particle quantum system by considering the tensor product $|\psi\rangle = \otimes_{i=1}^n |\psi_i\rangle$ of the n single-qubit-states. The eigenstates of the system are then given by $\{|0\rangle, |1\rangle\}^{\otimes n}$.

Transverse Field Ising Hamiltonian. Adiabatic quantum algorithms for optimization usually use so-called stochastic Hamiltonians [1]. In the computational basis $\{|0\rangle, |1\rangle\}^{\otimes n}$, a Hamiltonian H is stochastic if it fulfils the condition

$$\langle q|H|q'\rangle \leq 0 \quad \forall q, q' \in \{0, 1\}^n \text{ with } q \neq q'. \tag{3}$$

The main idea behind AQC with stochastic Hamiltonians consists in initializing the system with a Hamiltonian H_0 proportional to the transverse field, i.e.,

$$H_0 = -\mathcal{A}\left(\sum_{k=1}^{n}\sigma_k^x\right), \tag{4}$$

where σ_k^x denotes the Pauli matrix [25] in the x direction acting on the kth particle of the system and $\mathcal{A}$ is a positive constant. The ground state of H_0 is the uniform superposition state $|\psi(0)\rangle = |+\rangle^{\otimes n}$. Then, another Hamiltonian H_1 based on the Ising energy function is prepared with coupling strengths W_{kl} between particles k and l, and external biases h_k over the particles k as

$$H_1 = -\mathcal{B}\left(\sum_{k,l=1}^{n} W_{kl}\sigma_k^z \otimes \sigma_l^z + \sum_{k=1}^{n} c_k\sigma_k^z\right), \tag{5}$$

where again σ_k^z denotes the Pauli matrix in the z direction acting on the kth particle and $\mathcal{B}$ a positive constant. If the parameters W_{kl} and c_k are chosen as quadratic and linear parameters of a QUBO problem as in Eq. (1), then the ground state $|\psi(1)\rangle$ of H_1 encodes the solution of that problem. As the time s evolves from 0 to 1, the initial Hamiltonian H_0 is transformed into H_1, describing a time-dependent stochastic system Hamiltonian

$$H(s) = (1-s)H_0 + sH_1. \tag{6}$$

Quantum System Evolution. The evolution of the system generated by $H(s)$ over the time $s \in [0,1]$ is governed by the Schrödinger equation

$$i\hbar\frac{\partial\,|\psi(s)\rangle}{T\partial s} = H(s)\,|\psi(s)\rangle\,, \tag{7}$$

whose solution defines a time-dependent unitary transformation $U(s)$ of the initial state vector $|\psi(0)\rangle$ [1]. If s satisfies the condition of adiabacity, that is, if s varies sufficiently slowly, then the adiabatic theorem states that $U(s)$ will map, with high probability, the ground state of H_0 into the ground state of H_1, which is the solution of the optimization problem [17].

D-Wave Quantum Annealing. D-Wave quantum computers [34] are designed to solve QUBO optimization problems of the form (1) using transverse field Ising Hamiltonians in the form of Eq. (6). The quantum processor creates a network of logical qubits according to the problem size, which is then embedded in the quantum hardware. The network starts at $s = 0$ in a global superposition over all possible eigenstates. In a process called quantum annealing, as $s \to 1$, the provided couplings and biases are changed into magnetic fields that deform the state landscape, emphasizing the state that is most likely to be the solution of the optimization problem. D-Wave quantum annealers are made available through the Leap quantum cloud service [32], and the D-Wave quantum algorithms can be implemented in Python using the Ocean software [33].

3. Iterative Quantum Transformation Estimation (IQT)

Like many computer vision problems, (RPR) is intrinsically not a binary optimization problem and as such not directly solvable on current quantum annealers. We show that linearizing the rotation matrix and introducing binary coefficients allows to approximate the least-squares registration problem by a QUBO problem. Since the 2D and 3D cases are very similar, we present our method from a general point of view and only switch to a case-by-case discussion when necessary.

Unlike [14], our approach contains the number of qubits K as a free parameter that can be chosen by the user. Among others, this is critical to obtain an algorithmic scheme that can be carried out on current quantum hardware. Despite the fixed number of qubits, our algorithm successively increases the approximation precision of the rotation parameters, thereby enabling its computation to arbitrary precision.

Another issue is that optimizing over the rotation group $SO(D)$ calls for an orthogonality constraint on the rotation matrix R. One way to avoid this constraint, as already introduced in [10, 19], is to leverage the one-to-one correspondence between the Lie group $SO(D)$ and its algebra $\mathfrak{so}(D)$. This allows to optimize (RPR) over the *linear* space $\mathfrak{so}(D)$, which is the set of skew-symmetric matrices M of dimension D. Specifically, we will use the exponential map $R = \exp(M)$ to associate to $M \in \mathfrak{so}(D)$ the rotation matrix $R \in SO(D)$ [13].

Fixed-Point Representation. In order to represent the solutions with increasing accuracy, we use K bits to approximate a point x in an interval $[a,b]$ according to

$$x = a + \frac{b-a}{s}\sum_{k=0}^{K-1} q_k 2^k, \tag{8}$$

with binary variables $q_k \in \{0,1\}$ and scaling factor $s = 2^K - 1$. Apparently, x belongs to the grid that results from the discretization of the interval $[a,b]$ using s bins. We refer the reader to [27, 29] for alternative approaches for fixed- and floating-point representations on quantum annealers.

3.1. Approximation of Rotation Matrices

Rotation Matrix in 2D. In 2D, the only possible skew symmetric matrix $M \in \mathfrak{so}(2)$ is the one-parameter matrix

$$M(\theta) = \theta S = \theta \cdot \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \tag{9}$$

where $\theta \in \mathbb{R}$ represents the rotation angle. The associated rotation matrix R is

$$R = e^{M(\theta)} = (\cos\theta) I + (\sin\theta) S, \tag{10}$$

and any 2D rotation matrix can be written in this way [13].

In [14] the authors proposed to treat $\cos\theta$ and $\sin\theta$ as independent variables (say ω_1, ω_2) and to then optimize with respect to both. This has the disadvantage that the resulting matrix $R = \omega_1 I + \omega_2 S$ may be not a rotation matrix, as this requires $\omega_1^2 + \omega_2^2 = 1$. To obtain a QUBO problem that avoids this issue, we regard R as a nonlinear function of θ and determine the optimal θ in a multi-step process. Given a current angle θ_c, we linearize R around θ_c:

$$\begin{aligned} R(\theta) = g(\theta) I + h(\theta) S &\approx \Big[g(\theta_c) + g'(\theta_c)(\theta - \theta_c) \Big] I \\ &+ \Big[h(\theta_c) + h'(\theta_c)(\theta - \theta_c) \Big] S, \end{aligned} \tag{11}$$

where $g(\theta) = \cos\theta$ and $h(\theta) = \sin\theta$.

To obtain a QUBO problem, we quantize θ according to Eq. (8): For a fixed search window size $\Delta > 0$, we express $\theta \in [\theta_c - \Delta, \theta_c + \Delta]$ by $\theta = \theta_c - \Delta + \frac{2\Delta}{s} \sum_{k=0}^{K-1} q_k 2^k$. Inserting this representation into Eq. (11) and using $\hat{v} := \frac{2\Delta}{s} \sum_{k=0}^{K-1} q_k 2^k$, we obtain the approximation

$$R(\hat{v}) \approx R_c + \Big(g'(\theta_c) I + h'(\theta_c) S \Big) \hat{v}, \tag{12}$$

where $R_c := [g(\theta_c) - g'(\theta_c)\Delta] I + [h(\theta_c) - h'(\theta_c)\Delta] S$ is a term that depends on θ_c and Δ, but is independent of the unknown $\hat{v}$ and hence of the optimization variables $q_k \in \{0,1\}$, $k = 0, \dots, K-1$.

For any of the mass-centered points $\tilde{y}_i$, $i = 1, \dots, N$, as in (RPR), we thus find

$$R\tilde{y}_i \approx R_c \tilde{y}_i + R_i \hat{v}, \tag{13}$$

where

$$R_i := \Big(g'(\theta_c) I + h'(\theta_c) S \Big) \tilde{y}_i \tag{14}$$

is independent of $\hat{v}$ and hence of the unknowns q_k. The approximation (13) is at the heart of the QUBO problem that we solve.

Rotation Matrix in 3D. In the 3D case, the complete rotation parameters can be recorded in a vector $v = (v_1, v_2, v_3)^\top \in \mathbb{R}^3$ encoding the rotation angle $\theta = \|v\|_2$ and the rotation axis $x = v / \|v\|_2 = (x_1, x_2, x_3)^\top$. The skew-symmetric matrix $M \in \mathfrak{so}(3)$ has the form

$$M(v) = \theta \cdot \begin{pmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{pmatrix}, \tag{15}$$

and the exponential map associates to $M(v)$ the rotation matrix

$$R = e^{M(v)} = I + \frac{\sin\theta}{\theta} M(v) + \frac{1 - \cos\theta}{\theta^2} M^2(v), \tag{16}$$

where $\theta = \theta(v) = \|v\|_2$, see [13]. For convenience, we introduce the functions $g(v) = (\sin\|v\|_2) / \|v\|_2$ and $h(v) = (1 - \cos\|v\|_2) / \|v\|_2^2$ for $v \neq 0$ and $g(0) = 1$, $h(0) = 1/2$, $g'(0) = h'(0) = 0$.

Since $M(v)$ is linear in v, we have $M'(v_c)(w) = M(w)$ for any $v_c, w \in \mathbb{R}^3$. Hence, linearization of R around the current vector v_c yields

$$\begin{aligned} R(v) &= I + g(v) M(v) + h(v) M^2(v) \\ &\approx I + \Big[g(v_c) + g'(v_c)(v - v_c) \Big] M_c + g(v_c) M(v - v_c) \\ &\quad + \Big[h(v_c) + h'(v_c)(v - v_c) \Big] M_c^2 \\ &\quad + h(v_c) \Big[M(v - v_c) M_c + M_c M(v - v_c) \Big], \end{aligned} \tag{17}$$

where we used $M_c := M(v_c)$. Assuming that the components v_j, $j = 1, 2, 3$ are in the search window $[(v_c)_j - \Delta, (v_c)_j + \Delta]$ for some $\Delta > 0$, we make use of the discretization $v_j = (v_c)_j - \Delta + \frac{2\Delta}{s} \sum_{k=0}^{K-1} q_k^{(j)} 2^k$ from Eq. (8). Introducing $q_k := (q_k^{(1)}, q_k^{(2)}, q_k^{(3)})^\top$ and $\hat{v} := \frac{2\Delta}{s} \sum_{k=0}^{K-1} q_k 2^k$ we can, in slight abuse of notation, rewrite this as $v = v_c - \Delta + \hat{v}$. Thus, we arrive at the approximation

$$\begin{aligned} R(\hat{v}) &\approx R_c + \Big[g(v_c) + h(v_c) M_c \Big] M(\hat{v}) \\ &\quad + h(v_c) M(\hat{v}) M_c + M_c g'(v_c) \hat{v} + M_c^2 h'(v_c) \hat{v}, \end{aligned} \tag{18}$$

where again the term R_c depends on v_c and Δ, but not on the optimization variables $q_k \in \{0,1\}^3$, $k = 0, \dots, K-1$. It is straightforward to check that $M(\hat{v}) w = -M(w) \hat{v}$ for any $w \in \mathbb{R}^3$, so we deduce

$$R\tilde{y}_i \approx R_c \tilde{y}_i + R_i \hat{v}, \tag{19}$$

where the matrices R_i are given for $i = 1, \dots, N$ by

$$\begin{aligned} R_i &= M_c \tilde{y}_i g'(v_c) + M_c^2 \tilde{y}_i h'(v_c) \\ &\quad - h(v_c) M(M_c \tilde{y}_i) - \Big[g(v_c) + h(v_c) M_c \Big] M(\tilde{y}_i), \end{aligned} \tag{20}$$

and are independent of $\hat{v}$. As final step, we introduce

$$q = \begin{pmatrix} q_0 \\ q_1 \\ \vdots \\ q_{K-1} \end{pmatrix} \in \{0,1\}^{KP} \qquad \text{and} \qquad U = \frac{2\Delta}{s} \begin{pmatrix} D_0 & D_1 & \dots & D_{K-1} \end{pmatrix} \in \mathbb{R}^{P,PK}, \tag{21}$$

where for $k = 0, \dots, K-1$ the diagonal matrix $D_k \in \mathbb{R}^{P,P}$ has the entries 2^k. This notation allow us to rewrite $\hat{v} = Uq$, so $R_i\hat{v} = R_iUq$, both in 2D and in 3D.

Note that in both cases, the number K of qubits used for quantizing $\hat{v}$ can be chosen by the user, for instance based on the available hardware. In particular, as quantum computers become more powerful, larger values for K than those currently feasible can be used, which is likely to decrease the required wall-clock time of our method.

3.2. QUBO Formulation

By design, we have $R\tilde{y}_i \approx R_c\tilde{y}_i + R_i\hat{v}$ by Eq. (13) and Eq. (19). Crucially, $\hat{v}$ and thus q appear linearly in this approximation, so $\|\tilde{x}_i - R\tilde{y}_i\|_2^2$ is quadratic in q. By ignoring the q-independent terms, we find that (RPR) can be approximated in terms of $\hat{v}$ by

$$\min_{q\in\{0,1\}^{KP}} \sum_{i=1}^{N} \left(\|R_i\hat{v}\|_2^2 + 2\left\langle R_c\tilde{y}_i - \tilde{x}_i, R_i\hat{v} \right\rangle \right). \tag{22}$$

As derived in Eq. (21), we have $\hat{v} = Uq$. Therefore, we propose the following QUBO to approximate (RPR):

$$\min_{q\in\{0,1\}^{KP}} q^\top Wq + c^\top q, \tag{QUBO}$$

where, with the notation $\hat{y}_i = R_c\tilde{y}_i - \tilde{x}_i$,

$$W = U^\top \left(\sum_{i=1}^{N} R_i^\top R_i \right) U \quad \text{and} \quad c = 2U^\top \sum_{i=1}^{N} R_i^\top \hat{y}_i. \tag{23}$$

To solve (QUBO) on the quantum annealer, we only need to pass the coupling parameters $W \in \mathbb{R}^{KP,KP}$ and the biases $c \in \mathbb{R}^{KP}$. Note that the dimensions of W and c do not depend on the number of points N. At the end of the annealing procedure, the measured bit-string q is used to compute the rotation parameter according to

$$\theta = \theta_c - \Delta + Uq, \quad \text{resp.,} \quad v = v_c - \Delta + Uq. \tag{24}$$

3.3. Quantum Transformation Estimation

Our iterative quantum transformation estimation strategy (IQT), described in Algorithm 1, consists in solving (QUBO) for different Taylor approximation points θ_c (2D case) or v_c (3D case). We start by making an initial guess for the rotation parameter such as $\theta_c = 0 \in \mathbb{R}$ in 2D, respectively, $v_c = 0 \in \mathbb{R}^3$ in 3D. For the search interval, we use an initial radius of $\Delta = \pi$. After assembling the matrices W and c, (QUBO) is solved. The solution q provides the next iterate θ or v via Eq. (24).

An important part of our algorithm is that we treat Δ as an adaptive parameter; we aim to decrease Δ whenever possible, which results in an increasingly better approximation of the solution and the associated rotation matrix. We decrease Δ when the error between the current and the previous rotation parameter becomes smaller than some threshold function of the binning size $2\Delta/s$, which we view as an indicator that the best solution within the current K-bits interval discretization of $[v_c - \Delta, v_c + \Delta]$ has been reached.

Algorithm 1: Iterative quantum approach for the transformation estimation problem (IQT) from point sets.

Data: $\tilde{\mathcal{X}}, \tilde{\mathcal{Y}}$: Point sets to register
maxit : maximal number of iterations
θ_c or v_c : initial rotation parameter
Δ : radius of the search interval
κ : threshold for decrease of Δ
K : number of qubits
$j \leftarrow 0$, Initialise R_c, and let $\tau = \frac{\kappa\Delta}{2^K - 1}$.
while $j <$ *maxit* **do**
  Construct R_i, U and $\hat{y}_i$ for $i = 1, \dots, N$.
  Compute W and c according to Eq. (23).
  Solve (QUBO) and measure solution q.
  Compute θ, resp., v according to Eq. (24).
  if $|\theta - \theta_c| < \tau$, *resp.,* $\|v - v_c\|_2 < \tau$ **then**
    $\Delta \leftarrow \Delta/2$.
    $\tau \leftarrow \tau/2$.
  end
  $\theta_c \leftarrow \theta$, resp., $v_c \leftarrow v$.
  $R_c \leftarrow R_c(\theta_c, \Delta)$, resp., $R_c \leftarrow R_c(v_c, \Delta)$.
  $j \leftarrow j + 1$.
end
return *Current parameter* θ_c*, resp.,* v_c*, and associated* R *(via Eq.* (11)*, resp., Eq.* (17)*).*

3.4. Classical Transformation Estimation

In Section 4 we compare the IQT algorithm with its classical relaxed version named iterative classical transformation estimation (ICT) that results from replacing the QUBO problem by a classical unconstrained quadratic programming (QP) problem. Specifically, with $R\tilde{y}_i = R_c\tilde{y}_i + R_iv$ from Eq. (13), resp., Eq. (19), and continuous variables $v \in \mathbb{R}^P$, we construct the QP problem

$$\min_{v\in\mathbb{R}^P} v^\top Wv + c^\top v, \tag{25}$$

		QPU/Topology	2000Q			Advantage 1.1		
		No. of Qubits K	3	5	10	3	5	10
2D	Synthetic	$\lvert\theta-\theta^*\rvert$	$2.42e-5$	$1.06e-10$	$\mathbf{1.66e-14}$	$2.28e-4$	$1.13e-8$	$\mathbf{2.58e-12}$
		$\lVert R-R^*\rVert_F$	$3.42e-5$	$2.26e-10$	$\mathbf{2.24e-14}$	$4.00e-4$	$1.06e-8$	$\mathbf{3.65e-12}$
		Avg. chain break	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$
	MNIST	$\lvert\theta-\theta^*\rvert$	$1.71e-4$	$1.35e-8$	$\mathbf{2.55e-15}$	$4.70e-5$	$1.00e-8$	$\mathbf{2.65e-9}$
		$\lVert R-R^*\rVert_F$	$1.28e-4$	$1.91e-8$	$\mathbf{4.71e-15}$	$6.65e-5$	$1.41e-8$	$\mathbf{3.74e-9}$
		Avg. chain break	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$
	Lena	$\lvert\theta-\theta^*\rvert$	$2.53e-4$	$7.94e-9$	$\mathbf{5.32e-15}$	$2.08e-4$	$1.44e-9$	$\mathbf{6.33e-12}$
		$\lVert R-R^*\rVert_F$	$3.58e-4$	$1.12e-8$	$\mathbf{7.25e-15}$	$2.95e-4$	$2.03e-9$	$\mathbf{8.95e-12}$
		Avg. chain break	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$
3D	Synthetic	$\lVert v-v^*\rVert_2$	$4.20e-4$	$\mathbf{4.00e-6}$	$8.76e-1$	$9.70e-5$	$\mathbf{6.71e-7}$	$0.80e-2$
		$\lVert R-R^*\rVert_F$	$3.80e-4$	$\mathbf{1.63e-6}$	$6.39e-1$	$1.73e-4$	$\mathbf{1.45e-6}$	$1.96e-1$
		Avg. chain break	$\mathbf{0}$	$\mathbf{0}$	0.01	$\mathbf{0}$	$\mathbf{0}$	0.002
	Bunny	$\lVert v-v^*\rVert_2$	$1.37e-5$	$\mathbf{3.61e-7}$	$1.25e-1$	$3.11e-4$	$\mathbf{9.52e-8}$	$1.45e-1$
		$\lVert R-R^*\rVert_F$	$2.30e-4$	$\mathbf{1.51e-6}$	$1.82e-1$	$3.70e-4$	$\mathbf{1.20e-7}$	$3.12e-1$
		Avg. chain break	$\mathbf{0}$	$\mathbf{0}$	0.01	$\mathbf{0}$	$\mathbf{0}$	0.001
	Completion	$\lVert v-v^*\rVert_2$	$9.58e-5$	$\mathbf{8.59e-8}$	$5.54e-1$	$7.75e-5$	$\mathbf{6.89e-5}$	$5.24e-1$
		$\lVert R-R^*\rVert_F$	$1.28e-4$	$\mathbf{1.12e-7}$	$7.15e-1$	$1.22e-4$	$\mathbf{9.53e-5}$	$4.85e-1$
		Avg. chain break	$\mathbf{0}$	$\mathbf{0}$	0.01	$\mathbf{0}$	0.004	0.03

Table 2. Distance of the reconstructed rotation parameters θ resp. v and matrices R of IQT to the ground truth (θ^* resp. v^*, R^*) for different numbers of qubits K on two D-Wave graph topologies (2000Q and Advantage 1.1). The results are reported after 15 optimization steps of the same problem in both the 2D and the 3D case. In addition, the chain breaks fraction averaged over the 15 iterations are provided. Smallest errors are observed using $K = 10$ qubits in the 2D case and $K = 5$ in 3D. For 3D problems, larger topologies lead to chain breakage, which degrades the optimization process.

with $W = \sum_{i=1}^{N} R_i^\top R_i$ and $c = 2\sum_{i=1}^{N} R_i^\top \hat{y}_i$. The solution of this QP problem is equivalent to that of the system of linear equations $2Wv = -c$.

4. Experimental Results

In this section, we experimentally evaluate accuracy, robustness and run times of a hybrid quantum-classical implementation of the IQT algorithm. The coupling parameters W and biases c are prepared in Python 3.9.2 on an Intel i7-7700K CPU machine with 16 GB RAM and the QUBO is subsequently solved on a D-Wave quantum annealer with the default annealing time of $20\mu s$ and 100 reads per optimization iteration. The best solution at each iteration is chosen to be the lowest energy eigenstate of the system.

The data sets used in the experiments are listed in Tab. 1. They include synthetic point sets in addition to the point sets shown in Fig. 1, namely the MNIST [26] and Lena edges [20] point sets in 2D, and the Stanford bunny [30] and the completion3D [35] point sets in 3D. The number of points varies from 150 for the synthetic point sets to 4845 in Lena.

Ablation Study on K. Design choices of our method include the number of qubits K used for the numerical representation of the unknown variables and the D-Wave graph topology. We evaluate the precision of the reconstructed rotation parameter and matrix for $K \in \{3, 5, 10\}$ on the 2D and 3D point sets for 15 iterations. Recall that in total KP qubits are necessary to encode the QUBO problem, where $P = 1$ in 2D and $P = 3$ in 3D. We run the experiment on the D-Wave 2000Q QPU which uses the Chimera topology [7] and on the D-Wave Advantage 1.1 QPU which uses the Pegasus topology [6]. We compare the computed rotation parameters and matrices with the respective ground truths. The results are displayed in Tab. 2. We observe that

	Data set name	Number of points
2D	Synthetic points	150 (variable)
	MNIST [26]	150
	Lena edges [20]	4845
3D	Synthetic points	150 (variable)
	Stanford bunny [30]	500
	completion3D [35]	2048

Table 1. Data sets used in the experiments with the number of points contained in the reference set.

a greater number of qubits does not necessarily increase the precision. As an example, the error for $K = 10$ qubits in the 3D case is much larger than the errors for $K = 3$ and $K = 5$. This discrepancy is related to the average chain break fraction, implying that more qubits translate into a noisier system, thereby increasing the amount of miscalculations. In the subsequent experiments, we use $K = 10$ in 2D and $K = 5$ in 3D, and execute the algorithm on the D-Wave 2000Q QPU, which seems to be the best configuration for the task at hand.

Sensitivity To Outliers. In order to verify the robustness of our method against outliers, we randomly add noise to a fraction of the template points; said fraction is called the *outlier ratio*. To measure the accuracy of the proposed method, we adopt two metrics proposed by [14]:

$$e_R := \left\| I - R^\top R \right\|_F \quad \textit{(consistency error)}$$

$$\text{and} \quad e_A := \frac{\|\tilde{\mathcal{X}} - R\tilde{\mathcal{Y}}\|_F}{\|\tilde{\mathcal{X}}\|_F} \quad \textit{(alignment error)},$$

where $\|\cdot\|_F$ is the Frobenius norm. While the consistency error measures the orthonormality of the reconstructed rotation matrix, the alignment error expresses how accurately the template points are mapped to the reference points. We remark that if R^* is the optimal solution of (RPR) obtained for v^* and if $v_c \to v^*$ for v_c generated by Algorithm 1, then the approximation of the rotation matrix given by the right-hand side in Eq. (12), respectively, Eq. (18) converges to R^* with error order $O(\|\theta_c - \theta^*\|_2^2)$, resp., $O(\|v_c - v^*\|_2^2)$.

In Fig. 2 we display the registration errors of IQT and its classical counterpart ICT as a function of the outlier ratio. We plot the errors for the 5th, 10th, and 15th iterations of IQT on 2D and 3D synthetic point sets. The errors in 2D are compared with those of the QA approach from [14] with the complete data sets as interaction points, i.e., $K = N$ in the sense of QA. As expected, both IQT and ICT converge to an orthogonal matrix while reducing the alignment error, with the consistency error being virtually independent of the outlier ratio. In contrast, the benchmark method QA [14] is more susceptible to producing non-orthogonal matrices as the outlier ratio increases.

Next, we highlight the resemblance between the convergence and the robustness of IQT and ICT, thus justifying the usefulness of the K-bits binary discretization, as well as the ability of the quantum computer to approximate real numbers by such a discretization.

Fig. 3 presents registration results of IQT for the completion3D data set [35]. We illustratively display the results for point pairs without noise and with 50% outlier ratio. We observe that IQT finds a good transformation even in the presence of strong noise, with accuracy comparable to classical computation (ICT) while generating consistent (i.e., almost orthogonal) rotation matrices, none of which holds for the existing QA method.

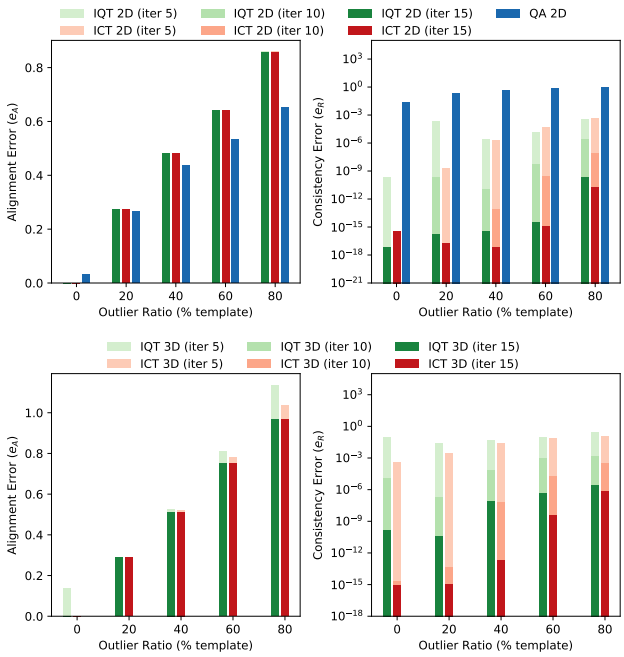
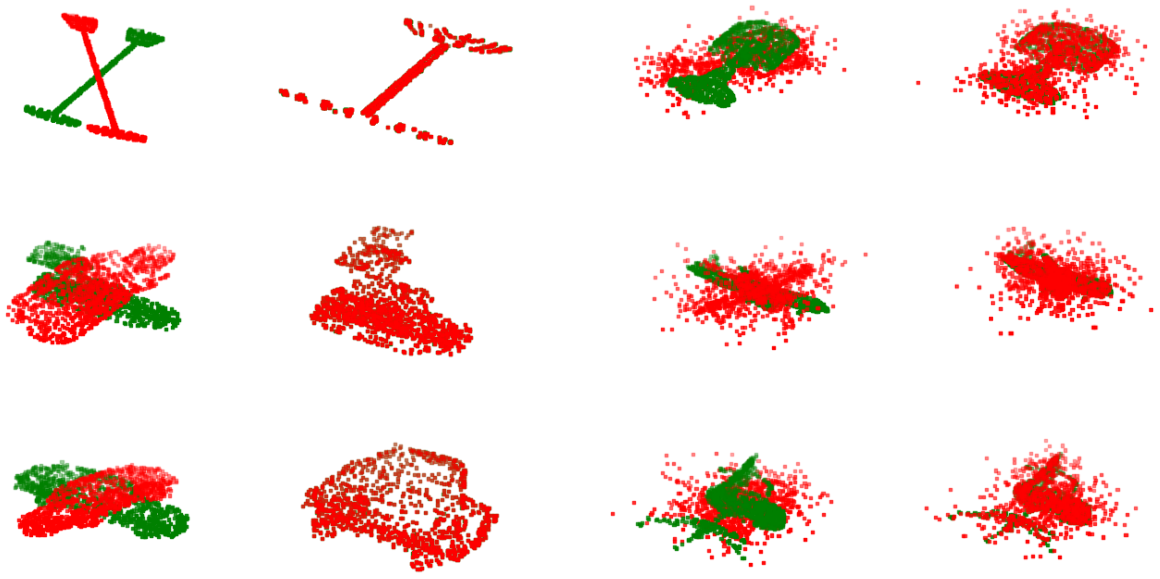


Figure 2. Robustness of the IQT method (**ours, green**) and its classical version ICT (**ours, red**) against outliers. Shown are the alignment error e_A (**left**) and the consistency error e_R (**right**, note logarithmic scale) as function of the outlier ratio for the 5th, 10th, 15th iteration on the synthetic 2D (**top row**) and 3D (**bottom row**) point sets. The proposed IQT method consistently achieves the same accuracy as the classical ICT method while avoiding consistency errors introduced by the QA method (**blue**) from [14].

Computational Costs and Timing. The most computationally expensive operation of our method is, as in [14], the preparation of the couplings and biases at each iteration. Preparing the matrices W and c requires $O(N\xi^2K^2P^4)$ and $O(N\xi\xi'KP^2)$ operations, where ξ and ξ' represent the cost for computing R_i and $R_c\tilde{y}_i - \tilde{x}_i$. Tab. 3 compares the total run time in milliseconds of the QPU access time of IQT and the CPU access time for ICT for solving the QP for various parameter configurations, averaged over 10 problem instances and for one iteration. For both IQT and ICT, we provide the time for the matrix preparations as well. We observe that both the QPU and CPU access times remain relatively constant with respect to the different combinations. On the contrary, the matrix preparation time grows with the number of points N and exceeds the QPU and CPU access time for moderate N.

Limitations. IQT should be seen as a first step towards efficient transformation estimation and ultimately point cloud



(a) Perfect data.

(b) Data corrupted with outliers.

Figure 3. Performance of IQT in registering **(a)** perfect point sets pairs and **(b)** reference point sets to template point sets with an outlier ratio of 50%. The point sets used are from the completion3D data set [35]. Green points represent reference points and red points represent template points. The initial alignment is shown on the left and the result of the registration on the right. Even with strong noise, the IQT algorithm provides a robust transformation estimation.

	No. of Qubits K	IQT: Matrix preparation / QPU access time (ms)			ICT: Matrix preparation / CPU access time (ms)		
N		150	1500	20000	150	1500	20000
2D	$K = 10$	2.50/34.67	23.20/34.67	311.25/34.69	6.99/0.10	55.56/0.08	334.7/0.06
	$K = 20$	2.51/35.32	23.00/34.79	305.19/35.26			
	$K = 40$	2.65/35.42	23.14/35.39	306.12/35.38			
3D	$K = 5$	3.43/35.04	32.07/34.92	425.56/34.96	6.58/0.01	34.84/0.01	421.77/0.01
	$K = 10$	3.50/35.23	32.16/35.23	420.18/35.24			
	$K = 15$	3.43/35.39	31.68/35.44	419.02/35.40			

Table 3. Runtime comparison for solving the quadratic problem (QUBO) on synthetic data using our quantum formulation (IQT, **left**) and its classical counterpart (ICT, **right**). All values are averaged over 10 instances for each problem size N. Experimentally, IQT runtime is independent of the number of qubits K. Matrix preparation dominates the QPU/CPU access time for solving the system.

registration on quantum hardware. Currently, the run time is dominated by preparation of the matrices both in the quantum as in the classical case. Limited availability of quantum hardware currently introduces additional time penalties for transfer to quantum computing service providers. Ultimately, it would be beneficial to have an intrinsical (circuit-model) quantum formulation that does not rely on AQC, which we leave for future work.

5. Conclusion

We investigated an adiabatic quantum formulation for transformation estimation from point sets and proposed an iterative quantum approach to optimally align point sets. Our method uses a flexible K-bits discretization for approximating the rotation matrices with refinement for improved accuracy. The resulting QUBO problem can be deployed and run on the D-Wave quantum annealer. Compared to the reference method QA [14], our approach robustly generates accurate rotation matrices which do not suffer from non-orthogonality even in the presence of strong noise.

References

[1] Tameem Albash and Daniel A. Lidar. Adiabatic quantum computation. *Reviews of Modern Physics*, 90(1), Jan 2018. 2, 3

[2] Marcel Seelbach Benkner, Vladislav Golyanik, Christian Theobalt, and Michael Moeller. Adiabatic quantum graph matching with permutation matrix constraints. *CoRR*, abs/2107.04032, 2021. 1

[3] Marcel S. Benkner, Zorah Lähner, Vladislav Golyanik, Christof Wunderlich, Christian Theobalt, and Michael Möller. Q-match: Iterative shape matching via quantum annealing. *CoRR*, abs/2105.02878, 2021. 1

[4] P.J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *PAMI*, 14(2):239–256, 1992. 1

[5] Tolga Birdal, Vladislav Golyanik, Christian Theobalt, and Leonidas J. Guibas. Quantum permutation synchronization. *CoRR*, abs/2101.07755, 2021. 1

[6] Kelly Boothby, Paul Bunyk, Jack Raymond, and Aidan Roy. Next-generation topology of d-wave quantum processors, 2020. 6

[7] Paul I. Bunyk, Emile M. Hoskinson, Mark W. Johnson, Elena Tolkacheva, Fabio Altomare, Andrew J. Berkley, Richard Harris, Jeremy P. Hilton, Trevor Lanting, Anthony J. Przybysz, and Jed Whittaker. Architectural considerations in the design of a superconducting quantum annealing processor. *IEEE TAS*, 24(4):1–10, 2014. 6

[8] Vicky Choi. Minor-embedding in adiabatic quantum computation: I. The parameter setting problem. *Quantum Information Processing*, 7:193–209, Oct 2008. 1

[9] Vicky Choi. Minor-embedding in adiabatic quantum computation: II. Minor-universal graph design. *Quantum Information Processing*, 10:343–353, Jun 2011. 1

[10] Shaoyi Du, Nanning Zheng, Shihui Ying, and Jianyi Liu. Affine iterative closest point algorithm for point set registration. *Pattern Recognition Letters*, 31(9):791–799, 2010. 3

[11] D.W. Eggert, A. Lorusso, and R.B. Fisher. Estimating 3-d rigid body transformations: a comparison of four major algorithms. *Machine Vision and Applications*, 9, March 1997. 1, 2

[12] Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21:467–488, 1982. 1

[13] Jean H. Gallier and Jocelyn Quaintance. Lie groups, lie algebra, exponential map. In *Differential Geometry and Lie Groups*, pages 537–570. Springer, 2019. 3, 4

[14] Vladislav Golyanik and Christian Theobalt. A quantum computational approach to correspondence problems on point sets. *CVPR*, abs/1912.12296, 2019. 2, 3, 4, 7, 8

[15] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2nd edition, 2003. 1

[16] Berthold K. P. Horn, Hugh M. Hilden, and Shahriar Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *J. Opt. Soc. Am. A*, 5(7):1127–1135, Jul 1988. 1, 2

[17] Sabine Jansen, Mary-Beth Ruskai, and Ruedi Seiler. Bounds for the adiabatic approximation with applications to quantum computation. *Journal of Mathematical Physics*, 48(10):102111, Oct 2007. 2, 3

[18] Tadashi Kadowaki and Hidetoshi Nishimori. Quantum annealing in the transverse Ising model. *Phys. Rev. E*, 58:5355–5363, Nov 1998. 2

[19] Pei Yean Lee. Geometric optimization for computer vision. *Ph.D. thesis. Australian National University*, Apr 2005. 3

[20] Bo Li, Aleksandar Jevtic, Ulrik Söderström, Shafiq Ur Réhman, and Haibo Li. Fast edge detection by center of mass. *ICISIP*, pages 103–110, 2013. 1, 6

[21] Andrew Lucas. Ising formulations of many NP problems. *Frontiers in Physics*, 2, 2014. 1

[22] Catherine C. McGeoch. Adiabatic quantum computation and quantum annealing: Theory and practice. *Synthesis Lectures on Quantum Computing*, 5(2):1–93, 2014. 1

[23] Jan Modersitzki. FAIR: Flexible Algorithms for Image Registration. *Proc SIAM*, 2009. 1

[24] Andriy Myronenko and Xubo Song. Point set registration: Coherent point drift. *PAMI*, 32(12):2262–2275, 2010. 2

[25] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 10th edition, 2010. 1, 2, 3

[26] Point Cloud Mnist 2D. https://www.kaggle.com/cristiangarcia/pointcloudmnist2d. 1, 6

[27] Giovani G. Pollachini, Juan P. L. C. Salazar, Caio B. D. Góes, Thiago O. Maciel, and Eduardo I. Duzzioni. Hybrid classical-quantum approach to solve the heat equation using quantum annealers. *Phys. Rev. A*, 104:032426, Sep 2021. 3

[28] Eleanor Rieffel and Wolfgang Polak. An introduction to quantum computing for non-physicists. *ACM Computing Surveys*, 32, 09 1998. 1

[29] Michael L. Rogers and Robert L. Singleton. Floating-point calculations on a quantum annealer: Division and matrix inversion. *Frontiers in Physics*, 8:265, 2020. 3

[30] The stanford 3d scanning repository. http://graphics.stanford.edu/data/3Dscanrep/. 1, 6

[31] Robert S. Sutor. *Dancing with Qubits*. Packt Publishing, 2019. 1, 2

[32] D-Wave Systems. D-Wave Leap. https://www.dwavesys.com/take-leap, Oct 2021. 3

[33] D-Wave Systems. D-Wave Ocean Software Documentation. https://docs.ocean.dwavesys.com/, Oct 2021. 3

[34] D-Wave Systems. QPU Solver Datasheet. https://docs.dwavesys.com/docs/latest/doc_qpu.html, Oct 2021. 3

[35] Lyne P. Tchapmi, Vineet Kosaraju, Hamid Rezatofighi, Ian Reid, and Silvio Savarese. Topnet: Structural point cloud decoder. In *CVPR proceedings*, pages 383–392, June 2019. 1, 6, 7, 8

[36] Fei Yan, Shan Zhao, Salvador E. Venegas-Andraca, and Kaoru Hirota. Implementing bilinear interpolation with quantum images. *Digit. Signal Proc.*, 117:103149, 2021. 1

[37] Zhengyou Zhang. Iterative point matching for registration of free-form curves and surfaces. *IJCV*, 13:119–152, 1994. 1