A Hybrid Quantum-Classical Algorithm for Robust Fitting

Anh-Dzung Doan 1 Michele Sasdelli 1 David Suter 2 Tat-Jun Chin 1
1 School of Computer Science, The University of Adelaide, South Australia
2 Centre for AI&ML, School of Science, Edith Cowan University, Western Australia

Abstract

Fitting geometric models onto outlier contaminated data is provably intractable. Many computer vision systems rely on random sampling heuristics to solve robust fitting, which do not provide optimality guarantees and error bounds. It is therefore critical to develop novel approaches that can bridge the gap between exact solutions that are costly, and fast heuristics that offer no quality assurances. In this paper, we propose a hybrid quantum-classical algorithm for robust fitting. Our core contribution is a novel robust fitting formulation that solves a sequence of integer programs and terminates with a global solution or an error bound. The combinatorial subproblems are amenable to a quantum annealer, which helps to tighten the bound efficiently.

1. Introduction

Imperfections in sensing and processing in computer vision inevitably generate data that contain outliers. Therefore, it is necessary for vision pipelines to be robust against outliers in order to mitigate their harmful effects.

In 3D vision, where a major goal is to recover the scene structure and camera motion, a basic task is to fit a geometric model onto noisy and outlier prone measurements. This is often achieved through the consensus maximisation framework [18]: given $N$ data points $\mathcal{D} = \{p_i\}_{i=1}^N$ and a target geometric model parametrised by $x \in \mathbb{R}^d$, let $P_N$ be the power set of index set $\{1, \ldots, N\}$. We aim to solve

$$\begin{align*}
\max_{\mathcal{I} \in P_N, x \in \mathbb{R}^d} & \quad |\mathcal{I}| \\
\text{s.t.} & \quad r_i(x) \leq \epsilon \quad \forall i \in \mathcal{I},
\end{align*}$$

(1)

where $r_i(x)$ is the residual of point $p_i$ w.r.t. $x$, and $\epsilon$ is a given inlier threshold. The form of $r_i(x)$ depends on the specific geometric model (more details in Sec. 3). A candidate solution $(\mathcal{I}, x)$ consists of a consensus set $\mathcal{I}$ and its “witness” (an estimate) $x$, where the points in $\mathcal{I}$ are the inliers of $x$. Problem (1) seeks the maximum consensus set $\mathcal{I}^*$, whose witness $x^*$ is a robust estimate of the model.

Many computer vision systems employ random sampling heuristics, i.e., RANSAC [32] and its variants (e.g., [5, 6, 20, 57, 68, 69]), for consensus maximisation. The basic idea is to repeatedly fit the model on randomly sampled minimal subsets of $\mathcal{D}$, and return the $\hat{x}$ with the largest consensus set $\mathcal{I}_k$. Such heuristics can only approximate (1) and generally do not provide optimality guarantees or error characterisation, e.g., a tight bound on the discrepancy $|\mathcal{I}_k^*| - |\mathcal{I}^*|$. Moreover, $\hat{x}$ is subject to randomness, and post-processing or reruns are often executed to vet the result.

Unfortunately, consensus maximisation is provably intractable [4, 16], hence there is little hope in finding efficient algorithms that can solve it exactly. While there has been active research into globally optimal algorithms [13, 17, 44, 54, 55], such techniques are realistic only for small input instances (small $d$, $N$ and/or number of outliers [17]).

Bridging the gap between exact algorithms that are costly and randomised heuristics that offer no quality assurances is an important research direction in robust fitting with practical ramifications. Towards this aim, deterministic approximate algorithms [14, 41, 42, 59, 73] eschew exhaustive search (e.g., branch-and-bound) and randomisation, and instead adopt deterministic subroutines such as convex optimisation, proximal splitting, etc. These methods avoid the vagaries of random sampling, and some can even guarantee convergence [41, 42, 59]. However, none of them provide error bounds. Indeed, complexity results [4, 16] also preclude efficient approximate solutions with error bounds.

Partly buoyed by the dominance of deep learning in computer vision, learning-based solutions to robust geometric fitting have been developed [10, 60, 70]. Such techniques leverage statistics in large datasets to learn a mapping from

\[ \text{Source code: https://github.com/dadung/HQC-robust-fitting} \]

\[ \text{This also depends on using a correct } \epsilon. \text{ The large volume of works that apply consensus maximisation suggest setting } \epsilon \text{ is usually not a concern.} \]
Our contributions We propose a new approach that leverages quantum computing for consensus maximisation. Our core contribution is a consensus maximisation algorithm that iteratively solves a sequence of integer programs and terminates with either $x^*$ or a suboptimal solution $\hat{x}$ with a known error bound $|I^*| - |\hat{I}| \leq \rho$. The integer programs are amenable to a quantum annealer [64, Chap. 8], which is utilised to tighten the bound efficiently. Since our method employs convex subroutines and random sampling, it is a hybrid quantum-classical algorithm [11, 36, 40, 56].

We will present results using an actual quantum computer, the D-Wave Advantage [23], as well as simulation. While our technique does not yet outperform state-of-the-art algorithms, in part due to the limitations of current quantum technology, our work represents a concrete application of quantum computing in computer vision. We hope to inspire future efforts on this topic in the community.

3. Reformulating consensus maximisation

In this section, we describe our novel reformulation for consensus maximisation and relevant theoretical results, before presenting the usage of quantum annealing in Sec. 4 and the overall algorithm in Sec. 5.

3.1. Preliminaries

Following [19], we consider residuals $r_i(x)$ that are quasiconvex, which encapsulates many geometric models of interest in computer vision [38]. Formally, if the set

$$\{ x \in \mathbb{R}^d \mid r_i(x) \leq \alpha \}$$

is convex for all $\alpha \geq 0$, then $r_i(x)$ is quasiconvex. Note that assuming quasiconvex residuals does not reduce the computational hardness of consensus maximisation [16].

For $C \in P_N$, define the minimax problem

$$g(C) = \min_{x \in \mathbb{R}^d} \max_{i \in C} r_i(x).$$

For quasiconvex $r_i(x)$, (3) is a quasiconvex program [3, 30], which is polynomial-time solvable. Note that $g(C) \leq \epsilon$ implies that $C$ is a consensus set, since all the points in $C$ are within error $\epsilon$ to the extremiser of (3).

Define the “feasibility test”

$$f(C) = \begin{cases} 0 & \text{if } g(C) \leq \epsilon; \\ 1 & \text{otherwise.} \end{cases}$$

Any $C$ such that $f(C) = 0$ implies that $C$ is a consensus set. Problem (1) can thus be restated as

$$\max_{I \in P_N} |I|, \quad \text{s.t. } f(I) = 0,$$

with the witness $x$ for any feasible $I$ obtainable through computing $g(I)$ to evaluate $f(I)$.

Given a consensus set $I$ with witness $x$, the points in the complement $O = \{1, \ldots, N\} \setminus I$ are the outliers to $x$. The “dual” of problem (5) is therefore

$$\min_{O \in P_N} |O|, \quad \text{s.t. } f(\{1, \ldots, N\} \setminus O) = 0,$$

i.e., find the model with the least number of outliers.
**Definition 1** (True inliers and true outliers). Let $\mathcal{I}^*$ be the maximum consensus set and $\mathcal{O}^* = \{1, \ldots, N\} \setminus \mathcal{I}^*$. We call $\mathcal{I}^*$ the “true inliers” and $\mathcal{O}^*$ the “true outliers”.

**Property 1** (Monotonicity). For (3) with quasiconvex residuals, given subsets $\mathcal{P}, \mathcal{Q}, \mathcal{R} \in \mathcal{P}_N$ with $\mathcal{P} \subseteq \mathcal{Q} \subseteq \mathcal{R}$, we have $g(\mathcal{P}) \leq g(\mathcal{Q}) \leq g(\mathcal{R})$. By extension, we also have that $f(\mathcal{P}) \leq f(\mathcal{Q}) \leq f(\mathcal{R})$. See [3, 30] for more details.

Intuitively, adding points to a feasible subset can only make it infeasible; the converse cannot be true. This leads to the following crucial concept.

**Definition 2** (Basis). A basis $\mathcal{B} \subset \{1, \ldots, N\}$ is a subset such that $g(\mathcal{B}') < g(\mathcal{B})$ for every $\mathcal{B}' \subset \mathcal{B}$.

Intuitively, removing any point from a basis $\mathcal{B}$ will cause the minimax value of the subset to shrink.

**Property 2** (Combinatorial dimension). The combinatorial dimension $\delta$ of minimax problem (3) is the upper bound on the size of bases [3, 30]. For quasiconvex $r_i(x_i), \delta = 2d + 1$.

**Claim 1.** If basis $\mathcal{B}$ is infeasible, i.e., $f(\mathcal{B}) = 1$, then $|\mathcal{B} \cap \mathcal{O}^*| \geq 1$, i.e., an infeasible basis $\mathcal{B}$ contains at least one true outlier.

**Proof.** See Sec. A in supplementary material.

### 3.2. Hypergraph vertex cover

Define the binary $N$-vector

$$ z = [z_1, \ldots, z_N] \in \{0, 1\}^N, \quad (7) $$

where the set of indices corresponding to nonzero $z_i$’s are

$$ \mathcal{C}_z = \{i \in \{1, \ldots, N\} | z_i = 1\}. \quad (8) $$

The outlier minimisation problem (6) can be reexpressed as

$$ \min_{z \in \{0, 1\}^N} \|z\|_1, \quad \text{s.t. } f(\{1, \ldots, N\} \setminus \mathcal{C}_z) = 0, \quad (9) $$

where $z_i = 1$ implies the i-th point is removed as an outlier.

Let $\{\mathcal{B}_{(k)}\}_{k=1}^K$ be $K$ binary $N$-vectors that correspond to all infeasible bases of the problem, i.e., for each $k,$

$$ f(\mathcal{C}_{\mathcal{B}_{(k)}}) = 1, \quad \|\mathcal{B}_{(k)}\|_1 \leq \delta, \quad (10) $$

where the latter appeals to the combinatorial dimension (Property 2). Also, the number of infeasible bases $K = O(N^\delta).$ Define the hypergraph $H$ with vertex set $\mathcal{V}$ and hyperedge set $E$ respectively as

$$ H = (\mathcal{V}, E), \quad \mathcal{V} = \{1, \ldots, N\}, \quad E = \{\mathcal{C}_{\mathcal{B}_{(k)}}\}_{k=1}^K. \quad (11) $$

Recall that hypergraphs are a generalisation of graphs, where a hyperedge can be incident with more than two vertices [2]. In our hypergraph (11), each hyperedge connects vertices that form an infeasible basis; see Fig. 1.

**Claim 2.** A subset $\mathcal{I} \subseteq \mathcal{V}$ is a consensus set iff it is an independent set of hypergraph $H$.

**Proof.** See Sec. B in supplementary material. \hfill □

Claim 2 proves that finding the maximum consensus set $\mathcal{I}^*$ is equivalent to finding the maximum independent set of $H$. Since the complement of an independent set is a vertex cover, it justifies to minimise the vertex cover

$$ \min_{z \in \{0, 1\}^N} \|z\|_1 \quad \text{s.t. } \mathcal{B}^T_z \geq 1, \quad \forall k = 1, \ldots, K, \quad (VC) $$

which is an 0-1 integer linear program (ILP). Setting $z_i = 1$ implies removing the i-th vertex, and the constraints ensure that all hyperedges are “covered”, i.e., at least one vertex in each hyperedge is removed (cf. Claim 1).

The hypergraph formalism has been applied previously in geometric fitting [2, 46, 47, 58]. However, the target problem in [2, 46, 47, 58] was higher order clustering (e.g., via hypergraph cuts), which is very distinct from our aims.

Formulation (VC) is impractical for two reasons:

- Hypergraph vertex cover is intractable;
- The number of hyperedges in $H$ is exponential.

However, the form (VC) is amenable to a quantum annealer, as we will show in Sec. 4. To deal with the number of hyperedges, we propose a hybrid quantum-classical algorithm in Sec. 5 that incrementally generates hyperedges.

### 4. Quantum solution

We first provide a basic introduction to quantum annealing, before describing our quantum treatment of (VC).

#### 4.1. Quantum annealing

A quantum annealer solves optimisation problems through energy minimisation of a physical system. A Hamiltonian defines the energy profile of a quantum system, which is composed of a number of interacting qubits. The system’s state is initialised at the lowest energy of the initial Hamiltonian and annealed such that the its final state gives the desired solution. At the end of the annealing, the Hamiltonian can be obtained from the following model

$$ \sum_{n} Q_{nn} q_n + \sum_{n < m} Q_{nm} q_n q_m = \mathbf{q}^T \mathbf{Q} \mathbf{q}. \quad (12) $$

The measurement collapses the $N$-qubit quantum state into $\mathbf{q} = [q_1, q_2, \ldots, q_N]$, where $q_n \in \{0, 1\}, \mathbf{Q} \in \mathbb{R}^{N \times N}$. The elements of $\mathbf{Q}$ define the couplings between qubits and their biases; see [64, Chap. 8] for more details.

A quantum annealer solves a problem of the form

$$ \min_{\mathbf{q} \in \{0, 1\}^N} \mathbf{q}^T \mathbf{Q} \mathbf{q}. \quad (13) $$
which is the quadratic unconstrained binary optimisation (QUBO). QUBO is intractable on a classical machine, but a quantum annealer, by virtue of the physical processes described above, may solve the problem efficiently. It allows $N$-qubits to evolve through superposed and entangled states (quantum tunnelling), and $q$ is obtained from the final measurement; see Sec. 4.3 on practical limitations.

4.2. Hypergraph vertex cover as QUBO

To simplify description of the main algorithm in Sec. 5, we first generalise (VC). Let $A$ be a subset of the hyperedges $E$ of the hypergraph $H$:

$$A = \{C_{a(v)}^M \}_{m=1}^M \subseteq E = \{C_{b(v)}^K \}_{k=1}^K. \quad (14)$$

Define the 0-1 ILP

$$I(A) = \min_{x \in \{0,1\}^N} \|z\|_1, \quad \text{s.t.} \quad A^T z \geq 1_M, \quad (15)$$

where $A \in \{0,1\}^{N \times M}$ is obtained by horizontally stacking the binary $N$-vectors $\{a_{(m)}\}_{m=1}^M$ corresponding to the hyperedges in $A$, and $1_M$ is the vector of $M$ ones.

We can recover (VC) from (15) by setting $A = E$. Moreover, since $A \subseteq E$, it is clear that $I(A) \leq I(E)$.

To formulate (15) as a QUBO, we first convert the inequalities into equalities. Define $\delta' = \delta - 1$. For each constraint $a_{(m)}^T z \geq 1$ in (15), we incorporate $\delta'$ binary slack variables $t_{(m)} = [t_{m,1}, \ldots, t_{m,\delta'}]^T$ into the constraint

$$a_{(m)}^T z - t_{(m)}^T 1_{\delta'} = 1; \quad (16)$$

recall that each $a_{(m)}$ has exactly $\delta$ elements with value 1. All $M$ equality constraints can be expressed in matrix form

$$H_A [v^T 1]^T = 0, \quad (17)$$

where $v = [z^T t_{(1)}^T, \ldots, t_{(m)}^T, \ldots, t_{(M)}^T]^T \in \{0,1\}^{N+\delta'M}$, $H_A = [A^T -S -1_M] \in \{0,1\}^{(N+M+\delta'M) \times (N+M+\delta'M)}$ and $S = I_M \otimes 1_{\delta'}$, the $M \times M$ identity matrix $I_M$, and Kronecker product $\otimes$. Also, the objective $\|z\|_1$ can be expressed in the quadratic form

$$\begin{pmatrix} v^T & 1 \end{pmatrix} [v^T, 1; J] [v^T, 1]^T, \quad (18)$$

with $J = \begin{pmatrix} I_N & 0_{N \times \delta'} \\ 0_{\delta' \times N} & 0_{\delta' \times M+1} \end{pmatrix}$, where $0$ is a zero matrix with the size specified in its subscript, and $I_N$ is the $N \times N$ identity matrix.

With penalty parameter $\lambda > 0$, we lift the constraints (17) into the objective to yield the QUBO

$$Q_{\lambda}(A) = \min_{v \in \{0,1\}^{N+\delta'M}} \left[ v^T I \right] [v^T, 1; J + \lambda H_A^T H_A] [v^T 1]^T. \quad (19)$$

Further algebraic manipulation is required to remove the constant 1 from (19) before exactly matching (13); see Sec. C in the supp. material for details. In the following, we will discuss solving (19) using a quantum annealer.

4.3. Practical considerations and limitations

We frame the discussion here in the context of SOTA quantum annealer—the D-Wave Advantage [23].

**Challenges** Problem (19) is an application of the quadratic penalty method [53, Chap. 17]. While fundamental results exist that allow $Q_{\lambda}(A)$ to equal $I(A)$, they invariably require $\lambda$ to approach a large value. However, the precision of D-Wave Advantage is limited to 4-5 bits [22,28], which precludes the usage of large penalty parameters.

Second, although there are $>5000$ qubits in D-Wave Advantage, the topology of quantum processing unit (QPU) rules out a fully connected model, i.e., the $Q$ matrix allowable is not dense [25,49]. Given an arbitrary $Q$, a minor embedding step [9,12,62] is required to map the QUBO onto the QPU topology. The embedding consumes extra physical qubits reducing the number of physical qubits available.

As alluded in Sec. 4.1, the annealing process “gradually” transitions (NB: by human scale the transition is rapid) the
quantum system from the initial Hamiltonian to the final Hamiltonian. Current quantum annealers are not able to completely isolate external noise from the process, which affects the quality of the solution.

To obtain a useful solution, during the annealing process, the system must have a non-negligible probability of staying in the lowest energy state. If the system jumps to a higher energy state, it will fail in solving the QUBO (19) optimally. The spectral gap is the minimum gap between the lowest and the second lowest (higher) energy states, which affects the probability of staying in the lowest energy state; see [24, 31] for details. We will investigate the spectral gap issue in Sec. D of the supplementary material.

Why quantum annealing? The above issues limit the scale of problems and quality of solutions attainable with current quantum annealers. However, quantum technology is advancing steadily, and the vision community should be prepared for potential breakthroughs, as like-minded colleagues are also advocating [7, 8, 19, 34, 63, 65]. Moreover, our main algorithm combines quantum and classical computation to leverage the strengths of both paradigms.

5. Main algorithm

Alg. 1 presents our overall algorithm. At its core, our algorithm aims to solve (VC), i.e., find the minimum outlier set and the sampled hyperedges $A$. Other main characteristics of the algorithm are:

• At each iteration, the QUBO (19) based on the current hyperedges $A$ is solved using quantum annealing.

• The penalty $\lambda$ for (19) decays following a schedule defined by hyperparameters $\lambda, \gamma$ and $M$ (Step 6).

• Hyperedges are sampled from a candidate vertex set $V'$, which is updated based on the current results (Sec. 5.2).

The algorithm terminates with the best estimate $z_{best}$ of the minimum outlier set and the sampled hyperedges $A$.

In the following, we show how the outputs of Alg. 1 can be used to derive an error bound for consensus maximisation, and the rationale of our hyperedge sampling technique.

5.1. Error bound

Consider the relaxation of (15)

$$LP(A) = \min_{z \in [0,1]^N} \|z\|_1, \ \text{s.t.} \ A^Tz \geq 1_M$$

which is a linear program. We must have that

$$LP(A) \leq I(A) \leq I(E).$$

Due to the factors in Sec. 4.3, the solution $z$ by the quantum annealer on (19) can be suboptimal. Given the best solution $z_{best}$, if the $I_{best} = V \setminus C_{z_{best}}$ is a consensus set, by Claim 2, $z_{best}$ is a vertex cover of (VC). We must have that

$$LP(A) \leq I(E) \leq \|z_{best}\|_1.$$  

Using the fact that $|I^*| = N - I(E)$, we thus have

$$|I| - |I_{best}| \leq \|z_{best}\|_1 - LP(A).$$

If the RHS is 0, then $I_{best}$ is the globally optimal solution.

5.2. Heuristic for sampling hyperedges

Recall that a hyperedge is an infeasible basis. A simple way to generate hyperedges is to randomly sample $\delta$-subsets from $V$ until we find an infeasible subset, which will not be efficient. To improve efficiency, our sampling technique maintains a candidate vertex set $V' \subseteq V$ where $f(V') = 1$, and takes the active set $S$ of $V' \setminus \{S\}$, where

$$g(S) = g(V'),$$

as a hyperedge. Intuitively, the active set of $V'$ is a basis with equal value with $V'$. To generate diverse hyperedges, two strategies are employed to maintain $V'$:

• Take $V' = V \setminus C_z$ if it is not a consensus set (Step 16);

• If a new consensus set $I = V \setminus C_z$ is found (Step 10), set $V'$ as the union of $C_z$ with a random subset of $I$ (Step 14).

5.3. Hyperparameter selection

The penalty value and decay schedule play important roles in Alg. 1 to quickly find a consensus set and tighten the error bound, see Sec. E in supplementary material for details. The precise values for the parameters used and/or investigated in our experiments will be provided in Sec. 6.
6. Experiments

6.1. Synthetic data

We first examine the performance of D-Wave Advantage (version 1.1) [23] on our robust fitting formulation via synthetic data. We generated 2D points \( D = \{ (a_i, b_i) \}_{i=1}^N \) for 1D linear regression \( (x \in \mathbb{R}^2) \) with residual \( r_i(x) = |a_i x - b_i| \), where \( 0 \leq a_i, b_i \leq 1 \). For a randomly chosen ground truth \( x \), a proportion of the points are corrupted with Gaussian noise of \( \sigma_{in} = 0.1 \) to form inliers, with the rest by Gaussian noise of \( \sigma_{out} = 1.5 \) to simulate outliers.

Due to the cost of accessing the QPU, our results in this subsection were not derived from many data repetitions. However, each QPU input instance was invoked with 10,000 anneals, which typically consumed \( \approx 1.3 \) seconds. Also, see Sec. F in the supp. material for details on embedding (19) (e.g., number of qubits, runtime) onto the QPU.

Comparison between CPU and QPU. We first compare CPU and QPU performance on our QUBO (19) (i.e., independent of Alg. 1), with \( A \) containing all hyperedges \( E \). The CPU solver used was Gurobi [35], which solves (19) exactly via exhaustive search, hence practical only for small instances. Fig. 2 plots the number of outliers \( ||z||_1 \) (lower is better) optimised by the solvers as a function of

- Penalty \( \lambda \in [0.1, 100] \), with \( N = 50 \), outlier ratio = 0.2.
- Outlier ratio \( \in [0.1, 0.6] \), with \( N = 20 \), \( \lambda = 1.0 \).
- \( N \in [10, 100] \), with \( \lambda = 1.0 \), outlier ratio = 0.2.

As expected (see Sec. 4.3), the gap in quality between the CPU solution and the “ground truth” provided by Gurobi increased with the examined parameters, indicating that the QPU is more reliable on “easier” instances of (19).

Main algorithm. Fig. 3 illustrates running Alg. 1 on synthetic 1D linear regression instances with \( N = 20, 50 \), and 100 points, each with outlier ratio 0.2. The QUBO subroutine (19) in the main algorithm was solved using the QPU with \( \lambda = 1.0 \) (no \( \lambda \) decay was done). The values \( ||z||_1 \) and \( LP(A) \) were plotted as a function of the size of \( A \), i.e., number of hyperedges. The results mainly illustrate the feasibility of solving robust fitting using quantum annealing.

Comparing to simulated annealing. In the context of Alg. 1, we compared quantum annealing (QA) and simulated annealing (SA) [37] (on CPU with 10,000 anneals) in solving the QUBO subroutine (Line 8 of Alg. 1). A synthetic 1D linear regression instance with \( N = 20 \) and outlier ratio 0.2 was generated. The penalty \( \lambda \) was set to 0.5 (no \( \lambda \) decay was done). Fig. 4 shows the runtime of QA and SA across the iterations of Alg. 1 (for QA, the cost of embedding (19) onto the QPU was excluded; again, see Sec. F in the supp. material for details), and the Hamming distance between the z’s found by the methods in each iteration.

The results illustrate that the runtime of SA (on CPU) grew steadily as the number of sampled hyperedges \( A \) increased, whereas the runtime of QA remained largely constant across the iterations, which suggests that the underlying physical processes of QA were not affected significantly by problem size (as long as the problem “fits” on the QPU). Also, Fig. 4 shows the solutions obtained by QA and SA are largely the same; this supports using SA in place of QA to examine the efficacy of Alg. 1 on larger sized real data.

6.2. Real data

We tested our method on real data for fundamental matrix estimation and multi-view triangulation. We used SA (on CPU) in place of QA to allow Alg. 1 to handle bigger problems. Two variants of Alg. 1 were executed:

![Figure 2. Comparisons between CPU and QPU on QUBO (19).](image)

![Figure 3. Number of outliers \( ||z||_1 \) optimised by QPU and lower bound \( LP(A) \), plotted across the iterations of Alg. 1.](image)
Figure 4. Comparing quantum annealing (on D-Wave Advantage) and simulated annealing (on classical computer).

- Alg. 1-E, where the algorithm was terminated as soon as a consensus set was found (Line 10).
- Alg. 1-F, where the algorithm was run until the maximum iterations $M$ (300 for fund. matrix, 200 for triangulation).

We compared our method to i) random sampling methods: RANSAC (RS) [32], LO-RANSAC (LRS) [20], and Fixing LO-RANSAC (FLRS) [43], ii) deterministic algorithms: Exact penalty (EP) [41], and Iterative biconvex optimization (IBCO) [14], and iii) quantum robust fitting (QRF) [19]. Each method was run 100 times and average results were reported. All experiments were conducted on a system with 2.6 GHz processor and 16 GB of RAM.

### 6.2.1 Fundamental matrix estimation.

We evaluated our method on linearised fundamental matrix fitting [61, Chapter 4], where $x \in \mathbb{R}^6$. We used inlier threshold $\epsilon = 0.03$ for the algebraic residual (convex in x, hence also quasiconvex), and penalty parameters $\lambda = 1.0$, $\gamma = 0.5$, $M = 50$, and $\lambda = 0.01$.

We used three image pairs from VGG [72] (Castle, Valbonne, and Zoom) and three image pairs from sequence 00 of KITTI odometry [33] \(^3\) (frame indices 104-108, 198-201, and 738-742). In each pair, SIFT features [48] were extracted and matched using VLFeat [1]; Lowe’s second nearest neighbour test was also applied to prune matches.

Fig. 5 shows the intermediate outputs of Alg. 1-F on Castle, KITTI 198-201 and KITTI 738-742, particularly the lower bound of the solution. See Sec. G in the supplementary material for the plots for the other points. Interestingly, the results show that it was more difficult to find a tight lower bound here (especially Nikolai point 134 and Linkoping point 1). This could be due to numerical inaccuracies in solving the minimax problem (3) for quasi-convex residuals [3, 30], which affected the efficacy of hyperedge sampling. Table 2 shows the quantitative results; a similar conclusions as that of Table 1 can be drawn. In particular, note that only our method was able to provide error bounds; in the case of Tower point 132, the global solution was provably found by the algorithm (gap is zero).

### 6.2.2 Multi-view triangulation

Points 134 & 534 from Nikolai, points 1 & 14 from Linkoping, and points 3 & 132 from Tower [29] were used. In this task, 3D coordinates of those 3D points ($x \in \mathbb{R}^3$) were estimated using reprojection error (which is quasiconvex [38]) under outliers. The inlier threshold and penalty were respectively $\epsilon = 1$ pixels and $\lambda = 5$. The decay parameters were $\gamma = 0.5$, $M = 50$, and $\lambda = 0.03$.

Fig. 7 shows the intermediate outputs of Alg. 1-F on Nikolai point 134, Linkoping point 1 and Tower point 132, particularly the lower bound of the solution. See Sec. G in the supplementary material for the plots for the other points.

Table 1 compares our method with the others. Overall, the quality of our method was comparable to the others, with Alg. 1-F providing higher quality and tighter bound than Alg. 1-E. Note that only our method returned error bounds (Sec. 5.1), which allowed to deduce that Alg. 1-F found consensus sets that were close to the optimum. As expected, the fastest methods were the random sampling approaches. Our method was much slower than the others, mainly due to the usage of SA. However, our experiments in Sec. 6.1 shows that QA can improve the speed of SA up to a factor of 10 without affecting solution quality (see Fig. 4). Hence, we expect Alg. 1 to be more competitive as quantum annealer capacity improves. Fig. 6 qualitatively illustrates the results of Alg. 1-E.

\(^3\)CC BY-NC-SA 3.0 License [21].

### 7. Weaknesses and conclusions

There are two main shortcomings: First, Alg. 1 was validated on an actual quantum computer only for small scale synthetic data (for reasons covered in Sec. 4.3). To fully realise the potential of the algorithm, testing with real data on a quantum computer is needed. Second, our results reveal that the hyperedge sampling procedure is also crucial to Alg. 1. Developing a more effective way of sampling hyperedges is an interesting research direction.
Table 1. Fundamental matrix estimation results. Alg. 1 employed simulated annealing (quantum annealing could be faster by 10 times). Only Alg. 1 amongst all methods compared here returned error bounds.

Table 2. Multi-view triangulation results. Alg. 1 employed simulated annealing (quantum annealing could be faster by 10 times). Only Alg. 1 amongst all methods compared here returned error bounds.

Conclusions Our work illustrates the potential of quantum annealing for robust fitting. It outperforms (in simulation) the only other quantum approach in robust fitting [19] as well as offers an error bound to mitigate the weakness of current QPU. We hope that our work helps trigger further development on applying quantum computers in robust fitting and computer vision applications.

Acknowledgement

This work was supported by Australian Research Council ARC DP200101675, and D. Suter acknowledges funding under Australian Research Council grant DP200103448.
References


