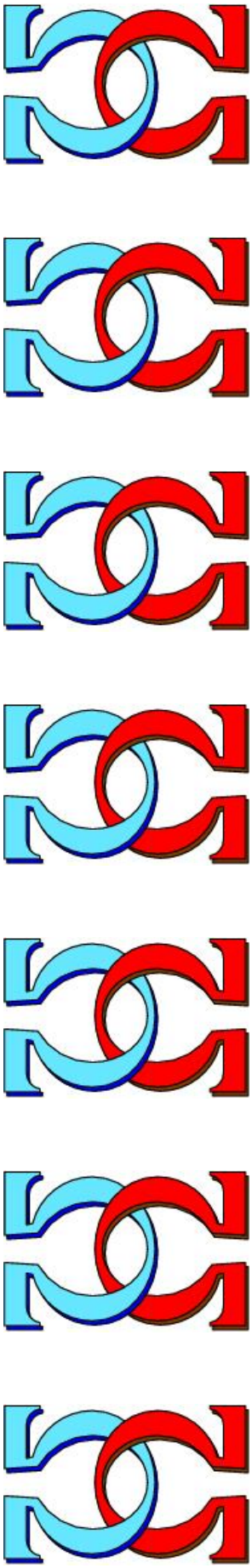

CDMTCS
Research
Report
Series

An Improved Quantum Solution for the Stereo Matching Problem

S. Heidari, M. Rogers, P. Delmas

University of Auckland, NZ

CDMTCS-559
November 2021

Centre for Discrete Mathematics and
Theoretical Computer Science

An Improved Quantum Solution for the Stereo Matching Problem

1st Shahrokh Heidari
Intelligent Vision Systems Lab
University of Auckland
Auckland, New Zealand
ORCID: 0000-0002-1727-939X

2nd Mitchell Rogers
Intelligent Vision Systems Lab
University of Auckland
Auckland, New Zealand

3rd Patrice Delmas
Intelligent Vision Systems Lab
University of Auckland
Auckland, New Zealand

***Abstract*—As the most computationally intensive part of a stereo vision system, stereo matching has been the focus of intense research activities for the last four decades. We present the first attempts to compare a quantum stereo matching solution with state-of-the-art approaches on the Middlebury stereo datasets. We first looked at quantum annealing computation as a way to interact with the D-Wave quantum computer and then improved the quantum solution to the stereo matching problem found in the literature. Using a line-by-line approach, we traded accuracy off for the qubits availability in the QPU (Quantum Processor Unit). Our findings show that it is possible to obtain results from real-sized images despite the scarcity of physical qubits in the quantum hardware. While the current quantum solution solves a class P stereo matching problem, its real advantage over classical stereo matching algorithms will arise with NP-hard problems which will be the focus of our future research.**

***Index Terms*—Quantum annealing, D-Wave, Stereo matching**

I. Introduction

Researchers have been looking for efficient methods to solve image processing and computer vision problems for some time, including quantum approaches. A quantum orthogonal image classifier [1] was proposed in 1997, and a definition of a quantum image [2] in 2003. The central promise of quantum techniques comes from their potential to reduce the algorithmic time complexity of some solutions significantly [3]. For this reason, significant effort is being made to solve image processing problems using quantum algorithms. Stereo vision has been a focus of interest for decades due to its potential to solve 3D mapping problems in real-world applications such as automotive [4] and robot vision [5]. A stereo vision system mimics human vision to estimate a 3D model of a scene using two or more digital cameras. It has many steps, from camera calibration and rectification to the final output used for distance measurement or 3D reconstruction [6]. The most computationally intensive part of the system is known as stereo matching. Many different stereo matching methods have been studied, such as block matching, dynamic programming, belief propagation, graph-cut, and learning-based techniques [7], [8]. As one of the most accurate stereo matching techniques, graph-cut-based methods [9], [10] use max-flow and min-cut methods on a particular structured graph to obtain the disparity map. Later research has used machine learning methods based on random forests and decision trees [11]. In addition, further research has recently used deep learning for the stereo matching problem [12]. Despite dramatic improvements in parallel processing hardware capability, deep-learning-based methods have great accuracy but are still computationally expensive. To date, only one paper [13] used quantum techniques for the stereo matching problem, namely the theoretical layout of a quantum annealing algorithm using a graph-cut approach. Here, we improve the method proposed [13] in terms of qubits requirements, thus allowing an entire run on the D-wave quantum computer and the first results on accepted stereo matching evaluation datasets [7]. The paper is organised as follows: Section II describes quantum annealing computation, an improved version of the quantum algorithm [13], and its complexity. In Section III, we describe the experimental results. We conclude with some comments and open problems in Section IV.

II. An improved quantum solution

A. Quantum Annealing

The most noticeable difference between classical and quantum computations is the concept of state. A state in a classical algorithm is shown by a register of n bits, each having value 0 or 1, while in a quantum algorithm, each state is represented by a register of n qubits having particular properties (such as *superposition* and *entanglement*). *Superposition* means that, unlike classical bits, a qubit can be 0 and 1 simultaneously, and *entanglement* refers to a situation where qubits after interaction are not independent anymore [14]. There are two main models of quantum computing: the quantum gate model [15] and the quantum annealing model [16]. The former is based on operating quantum circuits on a register of n qubits. The *superposition* property of qubits makes the register hold all 2^n states simultaneously, and the *entanglement* property makes it possible to apply a quantum circuit on all 2^n states in a constant time. This is called *quantum parallelism*, which promises considerable speedups over classical computations because a classical computation model such as the Turing machine would have to update 2^n registers to obtain the same results [14]. The advantages of the quantum gate model are

978-1-6654-0645-1/21/$31.00

that it is theoretically well studied and has many physical implementations. The main disadvantage is the fragility of qubits, a significant obstacle for manufacturing, controlling, error-correcting, protecting from noise, and, of course, cost. The quantum annealing model, a relatively new model of quantum computing, was introduced in [16] as an alternative equivalent [17] to the quantum gate model, where qubits are particles in a quantum system that evolves based on special forces acting on them. These forces are either external (from other sources) or internal (from interactions among qubits), and at any time, they are characterised by a time-varying Hamiltonian[1] (see [14]). What quantum annealing tries to do is to find a state of the quantum system which has the lowest energy. Therefore, quantum annealing algorithms are developed to solve optimisation problems. These problems are often represented mathematically as Quadratic Unconstrained Binary Optimisation (QUBO). A QUBO problem is a mathematical problem consisting of the minimisation of a quadratic objective function $q(\mathbf{x}) = \mathbf{x}^T Q \mathbf{x}$, where $\mathbf{x}$ is an n-vector of binary variables and Q is an $n \times n$ matrix. The matrix Q can be chosen to be upper-diagonal, and therefore $q(\mathbf{x})$ can be rewritten as: $q(\mathbf{x}) = \sum_i Q_{i,i} x_i + \sum_{i<j} Q_{i,j} x_i x_j$, where the diagonal terms $Q_{i,i}$ are the linear coefficients and the off-diagonal terms $Q_{i,j}, i < j$ are the quadratic coefficients [14]. Linear coefficients are the external forces, and quadratic coefficients are the internal forces in the Hamiltonian description. Take the Xor of two binary variables $\mathbf{x} = (x_1, x_2) \in \{0,1\}^2$ as a simple problem. $Xor(x_1, x_2) = 1$ if the values of x_1 and x_2 are not identical. If we define $q(\mathbf{x}) = -x_1 - x_2 + 2x_1x_2$, then we have

$$\{x_1 = 0, x_2 = 0\}, \{x_1 = 1, x_2 = 1\} \rightarrow q(\mathbf{x}) = 0, \quad (1)$$

$$\{x_1 = 0, x_2 = 1\}, \{x_1 = 1, x_2 = 0\} \rightarrow q(\mathbf{x}) = -1. \quad (2)$$

The objective function $q(\mathbf{x})$ is the QUBO model of the problem Xor over two binary variables. Considering (2), the lowest energies belong to those states of x_1 and x_2 which have different values. Therefore, minimising $q(\mathbf{x})$ gives optimal solutions of the Xor problem. The binary variables in a QUBO model are called logical variables. To minimise a QUBO model using a quantum annealer, depending on the complexity of the QUBO model, one or more physical qubits in a quantum computer are allocated to each logical variable in the QUBO formula. D-Wave Systems made the first commercially available quantum annealer in 2011. The D-Wave quantum computer solves QUBO problems using its Quantum Processor Unit (QPU), a network consisting of tiny metal loops called physical qubits connected via couplers. The newly released D-Wave machine Advantage works with $5,000$ qubits, $35,000$ couplers, and a 15-way qubit connectivity graph called Pegasus[2]. In a QUBO model, the linear coefficients define the external forces to physical qubits, and the quadratic ones define the internal forces related to the interactions between physical qubits (couplers). In the QPU, these forces are applied to qubits (the tiny metal loops) using magnetic fields. Given a QUBO problem to the QPU, quantum annealing begins to reach a state of the model with the lowest energy. This state is the optimal solution to the problem. Before using the D-Wave quantum computer for stereo matching, we first provide the preliminaries in the next section.

[1] A mathematical description of a system to show its total energy.

[2] https://www.dwavesys.com/solutions-and-products/systems/

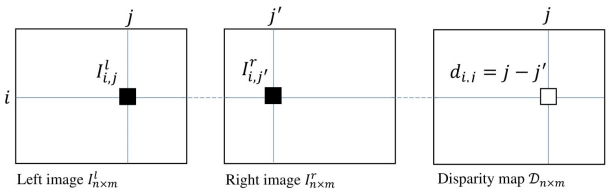

Fig. 1. A simple illustration to describe the concept of disparity.

B. Notation

Set I^l and I^r as the left and right stereo images defined by $n \times m$ matrices of positive 8-bits integers between 0 and 255. $I^l_{i,j} = I^l(i,j)$ shows the left image intensity at position (i,j), where $0 \leq I^l_{i,j} \leq 255$. Suppose $\mathcal{D}$ is an $n \times m$ disparity matrix, where each element of $\mathcal{D}$ named $d_{i,j}$ denotes the allocated disparity value to the pixel at position (i,j), where $1 \leq i \leq n$ and $1 \leq j \leq m$. As Fig. 1, we define the disparity of a pixel at position (i,j) as $d_{i,j}$, which locates the corresponding pixel of $I^l_{i,j}$ in the right image as $I^l_{i,j} = I^r_{i,(j-d_{i,j})}$ [7]. Furthermore, given a set of disparity values $L = \{d_{min}, \ldots, d_{max}\}$, $d_{i,j} \in L$; where d_{min} and d_{max} are the lowest and highest possible values for the disparities, respectively. The energy of allocating $\mathcal{D}$ to I^l and I^r is computed by $E(\mathcal{D})$ (3) [7].

$$E(\mathcal{D}) = E_d(\mathcal{D}) + \lambda E_s(\mathcal{D}), \quad (3)$$

$$E_d(\mathcal{D}) = \sum_{i=1}^{n} \sum_{j=1}^{m} \left(I^l_{i,j} - I^r_{i,(j-d_{i,j})} \right)^2, \quad (4)$$

$$E_s(\mathcal{D}) = \sum_{i=1}^{n-1} \sum_{j=1}^{m-1} \rho\left(d_{i,j} - d_{i+1,j}\right) + \rho\left(d_{i,j} - d_{i,j+1}\right), \quad (5)$$

where E_d computes the cost of allocating different disparity values to the pixels, E_s denotes the defined smoothness assumption, which penalises the energy function when disparity values of neighbouring pixels are different, and λ is a weighting factor for setting the relative importance of the smoothness term. Finally, ρ is a function to adjust the smoothness assumption [7]. The primary goal of minimising (3) is to find the optimal disparity values of all pixels with the minimum energy. In the simplest case, when ρ is a function that returns the absolute value of its input, a configuration of $\mathcal{D}$ which minimises (3) can be computed by solving the minimum s-t cut problem in a specially structured graph with two terminals s and t [10, p. 42]. This type of graphs was first introduced in [18] for an N-camera stereo matching problem, and then modified in [10] for a pair of stereo images. The following explains the graph structure described in [13], and the minimum s-t cut definition.

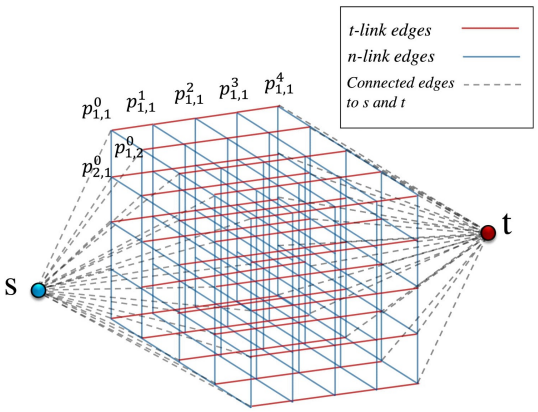

Fig. 2. The structured graph for a pair of stereo images with size 5×5 when $L = \{0, \ldots, 3\}$ and $k = 4$.

Set $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{C})$ as an undirected weighted graph, where $\mathcal{V}$ and $\mathcal{E}$ are sets of vertices and edges, respectively, and $\mathcal{C}$ is a function which allocates a positive weight to each edge. Given a set of possible disparities $L = \{d_{min}, \ldots, d_{max}\}$, Algorithm 1 shows how to structure the graph. We first set k as the number of disparities, and then a chain of $k+1$ vertices is defined. Edges of the chain are called *t-link*. Next, vertices of the chain are connected to the chains related to the pixels at positions $(i+1, j)$ and $(i, j+1)$ correspondingly. These edges are called *n-link*. Furthermore, two special vertices named s and t are defined, which are connected to $p^{d_{min}}_{i,j}$ and $p^{d_{max}+1}_{i,j}$, respectively [13]. Fig. 2 illustrates the graph structure for a pair of stereo images with size 5×5 when $L = \{0, \ldots, 3\}$ and $k = 4$.

Algorithm 1 Graph structure $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{C})$

$k \leftarrow (d_{max} - d_{min}) + 1$ ▷ number of disparities
for each pixel at position (i, j) **do**
  $\mathcal{V} \leftarrow \{p^{d_{min}}_{i,j}, \ldots, p^{d_{max}+1}_{i,j}\}$ ▷ $k+1$ vertices
  $\mathcal{E} \leftarrow \{p^{d}_{i,j}, p^{d+1}_{i,j}\}, d_{min} \leq d \leq d_{max}$ ▷ *t-link* edges
for each $p^d_{i,j} \in \mathcal{V}, d_{min} \leq d \leq d_{max} + 1$ **do**
  if $p^d_{i+1,j} \in \mathcal{V}$ **then**
    $\mathcal{E} \leftarrow \{p^d_{i,j}, p^d_{i+1,j}\}$ ▷ *n-link* edges
  if $p^d_{i,j+1} \in \mathcal{V}$ **then**
    $\mathcal{E} \leftarrow \{p^d_{i,j}, p^d_{i,j+1}\}$ ▷ *n-link* edges
$\mathcal{V} \leftarrow s, t$
for $p^{d_{min}}_{i,j} \in \mathcal{V}$ **do**
  $\mathcal{E} \leftarrow \{s, p^{d_{min}}_{i,j}\}$
for $p^{d_{max}+1}_{i,j} \in \mathcal{V}$ **do**
  $\mathcal{E} \leftarrow \{p^{d_{max}+1}_{i,j}, t\}$

The number of vertices $|\mathcal{V}|$, and the number of edges $|\mathcal{E}|$ in $\mathcal{G}$ can be given as (6) and (7). [13].

$$|\mathcal{V}| = (k+1)nm + 2, \tag{6}$$

$$|\mathcal{E}| = nm(k+2) + ((m-1)n + (n-1)m)(k+1). \tag{7}$$

Given $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{C})$ with two terminals s and t, an *s-t* cut partitions $\mathcal{V}$ into two subsets S and T, where $s \in S$ and $t \in T$. The cost of an *s-t* cut is the sum of weights of all edges that start at a vertex $u \in S$ and end at a vertex $v \in T$. The minimum *s-t* cut is the smallest possible *s-t* cut cost [19]. The following shows the weight initialisation in $\mathcal{G}$. Set a *t-link* edge in $\mathcal{G}$ as $t^d_{p_{i,j}} = \{p^d_{i,j}, p^{d+1}_{i,j}\}$, where $d_{min} \leq d \leq d_{max}$. The function $\mathcal{C}$ allocates a positive weight to $t^d_{p_{i,j}}$ as (8) [13].

$$\mathcal{C}(t^d_{p_{i,j}}) = \left(I^l_{i,j} - I^r_{i,(j-d)}\right)^2 + c, \tag{8}$$

where, c is a constant, and $\left(I^l_{i,j} - I^r_{i,(j-d)}\right)^2$ computes the cost of choosing d as the allocated disparity to the pixel at position (i, j). To initialise *n-link* edges, $\mathcal{C}$ allocates λ to them; where λ is a weighting factor for setting the relative importance of the smoothness term which is often set to a constant in our case. The constant c is set as $c > 4k\lambda$ [10, p. 42] to make sure that an *s-t* cut severs just one *t-link* for each pixel. Finally, edges $\{s, p^{d_{min}}_{i,j}\}$ and $\{p^{d_{max}+1}_{i,j}, t\}$ are initialised by a big number because they should not be selected for the minimum *s-t* cut [13]. Different classical algorithms can be used, such as Ford-Fulkerson [20] and Dinic [21], to find the minimum *s-t* cut in the structured graph. The minimum *s-t* cut $\mathcal{E}_{cut} \subset \mathcal{E}$ is a subset of edges which includes *t-link* and *n-link* edges. The selected *t-link* edges ($t^d_{p_{i,j}} = \{p^d_{i,j}, p^{d+1}_{i,j}\}$, $d_{min} \leq d \leq d_{max}$) define directly the disparity assigned to the pixel at position (i, j) [13]. We set $\mathcal{D}$ as the disparity map defined by an $n \times m$ matrix, with $d_{i,j}$ as its element. If $t^d_{p_{i,j}} \in \mathcal{E}_{cut}$, then $d_{i,j}$ is set to d [13]. Therefore, $\mathcal{D}$ will be the final output, showing the allocated disparity values. In the next section, we present an improved quantum solution to find the minimum *s-t* cut in the structured graph, and therefore to obtain the disparity map.

C. The improved quantum algorithm

The method [13] suggests a QUBO model for the graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{C})$ to solve the stereo matching problem. Recently, Krauss et al. [19] introduced a QUBO model that solves the minimum *s-t* cut problem with significantly fewer logical variables than [13]. Here, the QUBO model [19] is used to improve the quantum solution [13] in terms of QUBO requirements. Given $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{C})$, we set a binary variable x_v for each vertex $v \in \mathcal{V}$, which indicates $v \in S$ if $x_v = 1$, and $v \in T$ if $x_v = 0$. The variables allocated to vertices s and t are x_s and x_t, respectively. Therefore, we have a vector of binary variables as $\mathbf{x} \in \{0, 1\}^{|\mathcal{V}|}$, where $|\mathcal{V}|$ is the number of vertices (6) in $\mathcal{G}$. The objective function to find the minimum *s-t* cut can be given as (9) [19, p. 5].

$$\mathcal{H}(\mathbf{x}) = \mathcal{H}_1(\mathbf{x}) + \mathcal{H}_2(\mathbf{x}), \tag{9}$$

$$\mathcal{H}_1(\mathbf{x}) = \alpha_{s,t}(-x_s + x_s x_t), \tag{10}$$

$$\mathcal{H}_2(\mathbf{x}) = \sum_{\{u,v\} \in \mathcal{E}} \mathcal{C}(\{u, v\})(x_u - x_u x_v), \tag{11}$$

where $\mathcal{C}(\{u, v\})$ returns the weight of $\{u, v\} \in \mathcal{E}$, and $\alpha_{s,t}$ is a scaling factor such that $\alpha_{st} > \sum_{\{u,v\} \in \mathcal{E}} \mathcal{C}(\{u, v\})$ [19].

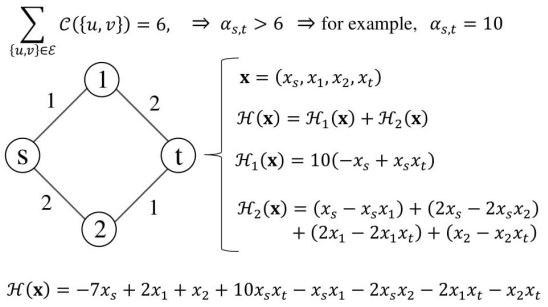

Fig. 3. The QUBO model [19] for a simple graph.

Considering (10), $\mathcal{H}_1$ is to ensure that we have always $s \in S$ and $t \in T$ because $\alpha_{s,t}(-x_s + x_s x_t)$ has the lowest energy when $x_s = 1$ and $x_t = 0$ (which means $s \in S$ and $t \in T$). According to the *s-t* cut definition, a given edge $\{u, v\} \in \mathcal{E}$ is in the *s-t* cut if it starts at $u \in S$ and ends at $v \in T$. Therefore, $\mathcal{H}_2$ (11) indicates that for each $\{u, v\} \in \mathcal{E}$, $x_u - x_u x_v$ is zero except when $x_u = 1$ and $x_v = 0$ (which means $v \in S$ and $u \in T$). Because $x_u - x_u x_v$ is scaled by $\mathcal{C}(\{u, v\})$, the penalty of choosing $\{u, v\}$ in the *s-t* cut is exactly its weight. Therefore, minimising $\mathcal{H}(\mathbf{x})$ gives an optimal solution which partitions $\mathcal{V}$ into two subsets S and T with the lowest cost such that $s \in S$ and $t \in T$. Fig. 3 illustrates a simple graph with four vertices. Minimising $\mathcal{H}(\mathbf{x})$ for this graph gives a unique solution as $x_s = 1, x_1 = 0, x_2 = 1$ and $x_t = 0$. Therefore, edges $\{s, 1\}$ and $\{2, t\}$ are in the minimum *s-t* cut. Regarding the stereo matching problem, we first structure the discussed graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{C})$ from a pair of stereo images. Next, the objective function (9) is defined as a QUBO model to find the minimum *s-t* cut in $\mathcal{G}$. In fact, $\mathcal{H}$ is given to the D-Wave QPU for minimisation. The optimal solution gives us $|\mathcal{V}|$ binary variables which partitions $\mathcal{V}$ into two subsets S and T. An edge $\{u, v\} \in \mathcal{E}$ is in the minimum *s-t* cut if $x_u = 1$ and $x_v = 0$. The *t-link* edges in the minimum cut are used to define the allocated disparities as discussed in the previous section.

D. The complexity of the quantum algorithm

As the number of qubits in today's quantum computers are very limited, even a simpler QUBO formula for our problem would be ideal. Given (6) and (7), the method [13] needs $2|\mathcal{E}|+|\mathcal{V}|$ QUBO variables while our method works with only $|\mathcal{V}|$ QUBO variables to solve the stereo matching problem. For instance, if we have a 40×40 pair of stereo images with 8 disparity levels, the number of needed logical variables in [13] would be $102,562$, while that for our method would be $14,402$ logical variables. Considering the complexity of the methods, both are based on solving the minimum *s-t* cut problem, a class **P** problem, which can be efficiently solved in polynomial time [22] using classical algorithms (such as Ford-Fulkerson [20] and Dinic [21]). In order to evaluate the performance of a quantum annealing algorithm, we need to assess the time complexity of filling the QUBO matrix and the annealing process. Generally, the annealing time of a QUBO problem scales as $\mathcal{O}(e^{\sigma N^{\beta}})$, where N is the number of logical variables in the QUBO formula, σ and β are factors dependent to the problem and machine [19]. While we acknowledge that the time complexity of the annealing is a subject of ongoing research [19], we will assume that it is $\mathcal{O}(1)$ through this article. Therefore, we only need to compute the time complexity of QUBO preparations for the quantum stereo matching methods. The method [13] scales as $\mathcal{O}(|\mathcal{E}|)$ to fill the QUBO matrix. We used the QUBO model [19] to improve [13], which needs $3|\mathcal{E}| + 2$ operations [19]. Hence, the QUBO preparations of both methods scale as $\mathcal{O}(|\mathcal{E}|)$. However, to find the minimum *s-t* cut in $\mathcal{G}$, classical algorithms Ford-Fulkerson [20], and Dinic [21] have complexity of $\mathcal{O}(|\mathcal{E}|f)$ and $\mathcal{O}(|\mathcal{V}|^2|\mathcal{E}|)$, respectively; where f is the maximum flow value through the graph.

III. Experimental Results

The newly released D-Wave Advantage accommodates 5000 physical qubits. Although the method [13] has been improved, there are still technical limitations to working with real-world image sizes due to the scarcity of available physical qubits in the QPU. We, therefore, took two approaches to tackle the problem. Firstly, we decided to independently structure a graph for each line of pixels followed by independent minimisation on the D-Wave QPU. This means in structuring graph $\mathcal{G}_r$, where $1 \leq r \leq n$, for each line of pixels, we only connected each chain of vertices $p_{r,j}^{d_{min}}, \ldots, p_{r,j}^{d_{max}+1}$ to the corresponding chain of adjacent pixel at position $(r, j+1)$, where $1 \leq j \leq m$. Fig. 4 illustrates this graph structure for the first line of pixels in a pair of stereo images with size 5×8 when $L = \{0, \ldots, 3\}$, and $k = 4$. The second approach was to leverage D-Wave QPU using cloud-based Leap's Hybrid Solvers[3], by which both classical and quantum resources are used to minimise the energy function optimally. When using hybrid solvers, a big-sized QUBO problem is decomposed into small sub-problems for the minimisation. They also identify which part of the problem is better to be directly solved by the QPU. Finally, we used *D-Wave Ocean SDK toolkit* to submit our problem to the cloud-based Leap's Hybrid Solver. Furthermore, we selected four pairs of stereo images from the Middlebury 2001-datasets, namely *Tsukuba* (288×384), *Bull* (381×433), *Venus* (380×434), and *Sawtooth* (383×434) having disparity levels $\{5, \ldots, 14\}, \{3, \ldots, 20\}, \{2, \ldots, 20\}$, and $\{4, \ldots, 18\}$, respectively. We could not use the newest Middlebury image datasets because of their large sizes and high disparity levels. Fig. 5 illustrates the left images in the first row, truth maps in the second row, and the corresponding D-Wave disparity maps (named DW) in the third row. The streaky lines in our results are because of using only one line of pixels to make the graph at each step. We, therefore, post-processed the results using a vertical median image filter followed by bilateral filtering to smooth images without blurring the edges. The improved DW maps can be seen in the forth

[3] https://docs.dwavesys.com/docs/latest/doc_leap_hybrid.html

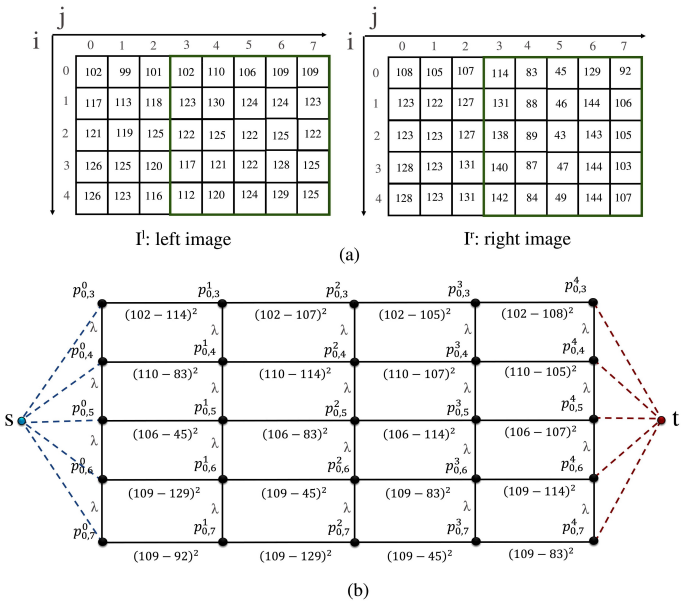

Fig. 4. The line-by-line graph structure for a simple image. a) The left and right images, and b) the graph structure for the first line. Here, $n = 5$, $m = 8$, $k = 4$, and $\lambda = k = 4$. We ignored the first three columns in the stereo images as they are occluded in the right image.

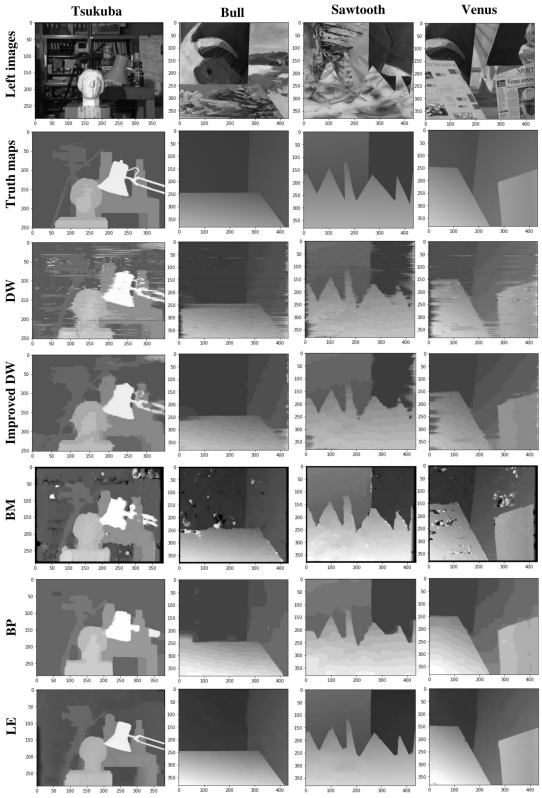

Fig. 5. Obtained disparity maps for the algorithms including the improved quantum solution, Block-matching, Belief propagation, and Continuous 3D-label stereo matching method which uses local expansion moves [12].

TABLE I
COMPUTED *RMS* AND *B* FOR THE OBTAINED DISPARITY MAPS.

datasets	DW		BM		BP		LE	
	rms	*b*	*rms*	*b*	*rms*	*b*	*rms*	*b*
Tsukuba	1.8	12.8	1.74	13	1.66	9	1.01	2.9
Bull	1.3	5.4	2.76	23	1.71	8	0.25	0.3
Sawtooth	1.9	9.9	3.34	22	1.96	10	0.81	2.8
Venus	1.4	9.8	3.27	26	2.40	6	0.62	2.31

TABLE II
PER-LINE RUNNING TIME FOR THE DISCUSSED METHODS.

datasets	*n*	*m*	*k*	DW	BM	BP	LE
Tsukuba	288	384	10	0.15 s	0.03 s	0.16 s	0.14 s
Bull	381	433	18	0.21 s	0.05 s	0.31 s	0.15 s
Sawtooth	380	434	15	0.17 s	0.05 s	0.26 s	0.13 s
Venus	383	434	19	0.21 s	0.05 s	0.34 s	0.16 s
Tree	240	295	18	0.18 s	0.03 s	0.21 s	0.11 s
Lib	175	297	18	0.17 s	0.04 s	0.21 s	0.11 s
Castle	282	398	39	0.49 s	0.08 s	0.52 s	0.15 s

row in Fig. 5. We have also provided results of some classical methods on the selected image datasets: Block matching (BM), Belief propagation (BP), and Local-expansion based method (LE) proposed in [12]. We also used two metrics defined in [7] for the evaluation, namely root-mean-squared (12), and percentage of bad matching pixels (13).

$$rms = \sqrt{\frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}(d_{i,j} - t_{i,j})^2}, \tag{12}$$

$$b = \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}(|d_{i,j} - t_{i,j}| > \delta_d) \times 100, \tag{13}$$

where $d_{i,j}$ is the element of the disparity matrix $\mathcal{D}$ obtained from the given algorithm, and $t_{i,j}$ is the corresponding element in the truth disparity matrix. Furthermore, nm is the total number of pixels, and δ_d is the disparity error tolerance. We set $\delta_d = 1.0$ as defined in [7]. Table I shows the computed *rms* and b for the discussed methods. We traded accuracy off for the qubits availability in the QPU using a line-by-line approach to get results from bigger images. Despite these practical limitations, Table I shows that our proposed method is significantly better than the BM method, and it is very close to the BP results. However, the LE method's results are much better than ours. For further analysis, we also chose three natural images named *Tree* (240×295), *Lib* (175×297), and *Castle* (282×398) having disparity levels $\{11, 26\}$, $\{6, 23\}$, and $\{1, 38\}$, respectively. Figure 6 shows the results for the discussed methods. Furthermore, to have some comparisons in terms of running time, we have provided Table II, which shows the per-line running time for the discussed methods. We used hybrid solvers for the minimisation, which are accessible through the D-Wave cloud-based services. These solvers use both classical and quantum resources. Therefore, Table II shows the QPU-access time of our method as we did not access the D-Wave classical resources.

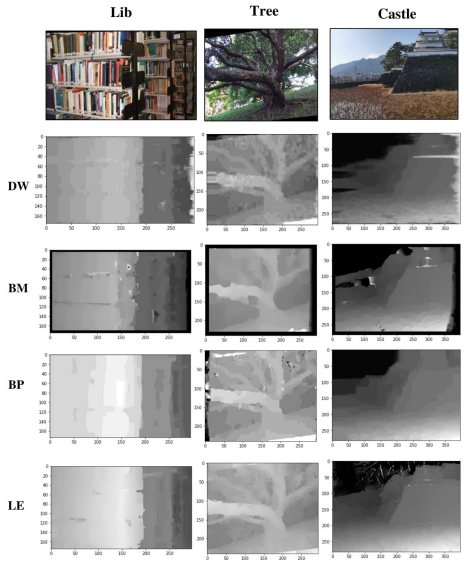

Fig. 6. Disparity maps for the selected natural images.

IV. Conclusion

This paper provided the first attempt at a practical quantum solution to stereo matching by leveraging the inherent epipolar geometry constraints by reducing the previously introduced quantum solution 2D search space to a line-by-line method. Our solution improved the number of QUBO variables required, thus allowing to run stereo pairs with up to 39 disparity levels and 383×434 pixels size. We further provided the first comparison with state-of-the-art and best performing algorithms on the Middlebury stereo datasets. We first addressed the practicability of quantum annealing for a stereo matching problem on real-world image sizes by trading accuracy off for the qubit availability in the QPU using a line-by-line approach. We showed that it is possible to obtain good results from the Middlebury stereo datasets and real-world image sizes. Regarding the advantages of the current quantum solution (theoretically) over the classical graph-cut stereo matching methods, we concluded that despite having time-complexity superiority over classical graph-cut methods, the class complexity of the problem is **P**, which means classical algorithms deal with such problems efficiently in polynomial time. Further research will explore more complex stereo matching problems to show the advantage of using quantum solutions over classical ones, notably by defining more complex smoothness assumption functions such as linear quadratic ones which are **NP-Hard**[4].

Acknowledgment

Many thanks to Prof. Cristian Calude and Dr. Michael Dinneen for their helpful comments. We also would like to thank William Cruz-Santos for providing their implementation codes which really helped us to understand their quantum solution.

[4]The complexity class of problems which are harder than those problems solvable by a non-deterministic Turing machine in polynomial time.

References

[1] A. Y. Vlasov, "Quantum computations and images recognition," *arXiv preprint quant-ph/9703010*, 1997.

[2] S. E. Venegas-Andraca and S. Bose, "Storing, processing, and retrieving an image using quantum mechanics," in *Quantum Information and Computation*, vol. 5105. International Society for Optics and Photonics, 2003, pp. 137–147.

[3] X.-W. Yao, H. Wang, Z. Liao, M.-C. Chen, J. Pan, J. Li, K. Zhang, X. Lin, Z. Wang, Z. Luo *et al.*, "Quantum image processing and its application to edge detection: theory and experiment," *Physical Review X*, vol. 7, no. 3, p. 031041, 2017.

[4] C. Badue, R. Guidolini, R. V. Carneiro, P. Azevedo, V. B. Cardoso, A. Forechi, L. Jesus, R. Berriel, T. M. Paixão, F. Mutz, L. de Paula Veronese, T. Oliveira-Santos, and A. F. De Souza, "Self-driving cars: A survey," *Expert Systems with Applications*, vol. 165, p. 113816, 2021.

[5] M. Abdelaal, R. M. Farag, M. S. Saad, A. Bahgat, H. M. Emara, and A. El-Dessouki, "Uncalibrated stereo vision with deep learning for 6-dof pose estimation for a robot arm system," *Robotics and Autonomous Systems*, vol. 145, p. 103847, 2021.

[6] M. Domínguez-Morales, A. Jiménez-Fernández, R. Paz-Vicente, A. Linares-Barranco, and G. Jiménez-Moreno, "Stereo matching: From the basis to neuromorphic engineering," in *Current Advancements in Stereo Vision*. IntechOpen, 2012.

[7] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International journal of computer vision*, vol. 47, no. 1, pp. 7–42, 2002.

[8] N. Lazaros, G. C. Sirakoulis, and A. Gasteratos, "Review of stereo vision algorithms: from software to hardware," *International Journal of Optomechatronics*, vol. 2, no. 4, pp. 435–462, 2008.

[9] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[10] O. Veksler, *Efficient graph-based energy minimization methods in computer vision*. Cornell University, 1999.

[11] S. R. Fanello, C. Rhemann, V. Tankovich, A. Kowdle, S. O. Escolano, D. Kim, and S. Izadi, "Hyperdepth: Learning depth from structured light without matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5441–5450.

[12] T. Taniai, Y. Matsushita, Y. Sato, and T. Naemura, "Continuous 3d label stereo matching using local expansion moves," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 11, pp. 2725–2739, 2017.

[13] W. Cruz-Santos, S. E. Venegas-Andraca, and M. Lanzagorta, "A qubo formulation of the stereo matching problem for d-wave quantum annealers," *Entropy*, vol. 20, no. 10, p. 786, 2018.

[14] C. C. McGeoch, "Adiabatic quantum computation and quantum annealing: Theory and practice," *Synthesis Lectures on Quantum Computing*, vol. 5, no. 2, pp. 1–93, 2014.

[15] N. D. Mermin, *Quantum computer science: an introduction*. Cambridge University Press, 2007.

[16] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, "Quantum computation by adiabatic evolution," *arXiv preprint quant-ph/0001106*, 2000.

[17] D. Aharonov, W. Van Dam, J. Kempe, Z. Landau, S. Lloyd, and O. Regev, "Adiabatic quantum computation is equivalent to standard quantum computation," *SIAM Review*, vol. 50, no. 4, pp. 755–787, 2008.

[18] S. Roy and I. J. Cox, "A maximum-flow formulation of the n-camera stereo correspondence problem," in *Sixth International Conference on Computer Vision*. IEEE, 1998, pp. 492–499.

[19] T. Krauss, J. McCollum, C. Pendery, S. Litwin, and A. J. Michaels, "Solving the max-flow problem on a quantum annealing computer," *IEEE Transactions on Quantum Engineering*, vol. 1, pp. 1–10, 2020.

[20] L. Ford Jr and D. Fulkerson, "Flows in networks (1962) princeton univ."

[21] Y. Dinitz, "Dinitz'algorithm: The original version and even's version," in *Theoretical computer science*. Springer, 2006, pp. 218–240.

[22] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis, "The complexity of multiterminal cuts," *SIAM Journal on Computing*, vol. 23, no. 4, pp. 864–894, 1994.