

Application of continuous time quantum walks to image segmentation

Michał Krok¹, Katarzyna Rycerz², and Marian Bubak²

¹ IBM Software Laboratory, Kraków, Poland

² AGH University of Science and Technology,
Department of Computer Science, Krakow, Poland
michal.krok@ibm.com {kzajac,bubak}@agh.edu.pl

Abstract. This paper provides the algorithm that applies concept of continuous time quantum walks to image segmentation problem. The work, inspired by results from its classical counterpart [9], presents and compares two versions of the solution regarding calculation of pixel-segment association: the version using limiting distribution of the walk and the version using last step distribution. The obtained results vary in terms of accuracy and possibilities to be ported to a real quantum device. The described results were obtained by simulation on classical computer, but the algorithms were designed in a way that will allow to use a real quantum computer, when ready.

Keywords: quantum walk · image segmentation · quantum algorithms

1 Introduction

Image segmentation [26] is one of the most important image processing tools, as it is frequently applied in an early stage of image analysis in order to extract some high-level features of the input image and significantly simplify further computations. Image segmentation is a process of dividing entire image into multiple separate segments – areas inside which pixels expose similar characteristics (e.g. intensity, hue, texture). It might be also defined as an activity of assigning each pixel of an image with a label, in a way that pixels sharing similar traits of interest are labeled alike. Image segmentation has found a broad range of applications: from a very trendy in recent years computer vision (object detection and recognition) [13] to medical diagnostics and treatment planning [14]. Despite many efforts and development of plethora of various methods, image segmentation remains an open research area, as there is still a need for inventing more efficient and more precise solutions.

A very interesting method for segmentation was invented by Grady [9]. The author proposes solution utilizing classical random walks, a very successful computational framework widely used for simulation of some natural and social processes in physics, biology and even economics [22]. The mechanism of random walk action is quite simple: a particle (walker) explores given space of possible

states by moving at each time step to a new state according to current state and given transition probabilities. The walker behavior can be useful for determining some properties of the explored space. Based on the random walk there has been developed an analogous quantum model – quantum random walks. It has been shown that quantum walks can benefit from the extraordinary properties of nanoscale world (quantum superposition, interference or entanglement) to achieve quadratic or even exponential [7] speedup over their classical counterparts for some particular problems. Therefore it seems reasonable to explore this branch of quantum computation in the quest for new, better algorithms.

The aim of this research is to elaborate and examine a new algorithm that applies concept of quantum walks to image segmentation. It is inspired by results from its classical counterpart [9]. In this paper, we present concept, implementation and results of two versions of the solution. The results were obtained by simulation on classical computer, but the algorithms were designed in a way that will allow them to be run on a real quantum computer, when the actual device will be available. This approach seems to be promising, especially in a context of recent work that describes implementation of simple quantum walk on IBM-Q hardware [4].

The rest of the document is organized as follows: section 2 discusses research concerning image segmentation and quantum walks. Section 3 presents elaborated algorithms, while section 5 examines their performance and accuracy. Finally, the effort put in this paper is summarized in section 6.

2 Related works

Image segmentation has been a carefully studied problem over the years and there has been developed a multitude of different solutions [26, 21]. One of the simplest are threshold-based algorithms, which assign a label to each pixel according to its intensity value in comparison to one or more thresholds, e.g. frequently used Otsu method [19]. Another approach that has been successfully applied is based on clustering; it determines segments by aggregating pixels in groups (clusters) according to chosen metric [17]. Deep learning and neural networks, especially convolutional neural networks, also proved to be very successful in various image processing tasks, one of which is image segmentation [6].

The interesting class are graph partitioning methods. To this category belong methods like: normalized cuts [23], intelligent scissors [18] and isoperimetric partitioning [10]. The aforementioned Grady algorithm also belongs to the last category of image segmentation solutions. As the input, the user defines (apart from an image) a set of seeds – pixels which already have assigned labels corresponding to the image segment that they belong to. The algorithm then tries to determine for each unmarked pixel, what are the probabilities of reaching each seed as the first by a random walk starting from given pixel. Each pixel is then assigned with a label of the seed that is most probable to be visited first by the walker.

With the development of quantum computing there emerged a new field of research – quantum image processing. In [24] there were presented methods for storing and retrieving as well as segmenting images in a quantum system. However, the proposed algorithm requires specifying objects in image through as set of their vertices, which rarely is a case and obtaining those vertices is a complicated task itself. Two other methods for quantum image segmentations were described in [12, 5], but these are the threshold-based solutions which do not take into account the spatial locations of pixels, which results in rather poor outcome.

Since Nayak and Vishwanath [3] showed that quantum walks on line spread quadratically faster than their classical analogues, the quantum model has been studied to find algorithms that would allow for similar speedup. The most spectacular achievement in this field was presented by Childs [7], where the author achieves exponential speedup by applying a continuous time quantum walk in a graph formed of two binary trees. There have been presented several search algorithms that can provide quadratic speedup benefiting from quantum walks e.g. works by Ambainis et al. [2] and Magniez et al. [15].

Over the recent years, there have been developed many various algorithms utilizing quantum walks. Most of the works consider rather theoretical problems, therefore it might be beneficial to elaborate an algorithm that would directly solve a practical, commonly encountered task.

3 Walk Model

In this paper we propose two versions of the algorithm for image segmentation utilizing continuous time quantum walks. In continuous time quantum walks (CTQW) transition from the current state to the next one does not take place in discrete steps, but can occur at any moment, according to a γ parameter that describes the rate of spread of the walk.

As the input, the elaborated algorithms expect following data:

- Image – two dimensional image of size $M \times N$ with each pixel p_{ij} described by two indices: $i \in \mathbb{Z}_M$ (index of the row) and $j \in \mathbb{Z}_N$ (index of the column). Each pixel can be represented as a single value of luminescence intensity or a vector of intensities in separate channels (e.g. vector of length 3 for the RGB color model).
- Labels $L = \{1, 2, 3, \dots, l\}$ – set of labels, one for each segment that the image should be divided into.
- Seeds $S = \{(p_1, l_1), (p_2, l_2), \dots, (p_s, l_s)\}$ – set of pixels p_k with already assigned label l_k from the set L ; there should be multiple seeds for each label.
- Parameter β – a parameter responsible for highlighting the stronger differences between pixels; it is used while transforming the image into a graph.
- Parameter γ – rate of spread of a continuous time quantum walk, used to construct the evolution operator.
- Parameter T – duration of the quantum walk (number of applications of evolution operator).

Firstly, the input image is converted into a weighted graph with vertices corresponding to pixels of the image. From each vertex there are drawn four edges to the neighboring pixels. Finally, each edge is assigned with a weight describing the level of similarity between adjacent pixels p_{ij} and p_{kl} . We use the formula proposed in [9]:

$$w_{ij,kl} = \begin{cases} e^{-\frac{\beta d(p_{ij}, p_{kl})}{\max d(p_{ij}, p_{kl})}}, & \text{if } p_{ij} \text{ and } p_{kl} \text{ are connected,} \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

where $d(p_{ij}, p_{kl})$ is a metric of pixel similarity defined in our paper as follows:

$$d(p_{ij}, p_{kl}) = \sum_c (p_{ij}[c] - p_{kl}[c])^2, \quad (2)$$

where c are the consecutive channels of pixels.

The main idea to tackle the problem of image segmentation is shown in the Fig.1 illustrating quantum walks for two example labels: green and red. First, a separate continuous time walk for each label is started and the walker explores the image according to its content. Then, based on which walker was the most willing to explore given region of the image, assign appropriate label to each pixel, which results in dividing the image into segments. The details of constructing the quantum walk described in this paper are presented below.

Position space. The position space for the walk is a Hilbert space of size $M \times N$; each basis state corresponds to a pixel of the image: $\{|0, 0\rangle, |0, 1\rangle, \dots, |0, M-1\rangle, |1, 0\rangle, \dots, |N-1, 0\rangle, |N-1, 1\rangle, \dots, |N-1, M-1\rangle\}$.

Initial state. The initial state of the walker is a superposition of positions of all the seeds for given label. Let the current label be $l_i \in L$ and the $S_{l_i} = \{(p_1, l_i), (p_2, l_i), \dots, (p_k, l_i)\} \subset S$ be a subset of seeds corresponding to that label. Then the initial state for walk corresponding to the i -th label has the following form:

$$|\psi_i(0)\rangle = \frac{1}{\sqrt{k}} \sum_{j=1}^k |p_j\rangle \quad (3)$$

Evolution operator. The most important element of the quantum walk is its evolution operator, that determines the behavior of the walk. The current task is to construct an operator that would encourage walker to explore image according to the weights between neighboring pixel. The desired behavior is to make the walker advance preferably over the higher weights between similar pixels and avoid crossing big gaps in pixel intensities.

The general form of the evolution operator of continuous time quantum walk has the following form (here i means imaginary unit):

$$U(t) = e^{-iHt}, \quad (4)$$

where t denotes time and H is a Hamiltonian matrix. The task is to find an appropriate Hamiltonian matrix H . It has to be a square matrix of size $MN \times$

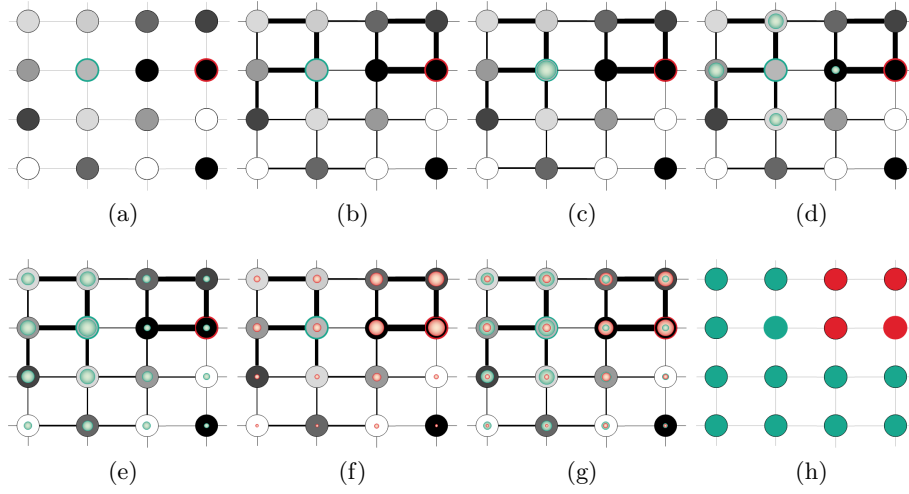


Fig. 1: Visualization of the concept outline. The algorithm input is shown in (1a) where circles symbolize pixels with their intensity and seeds are marked as two pixels with green and red ring. In (1b), image is transformed into weighted graph based on intensities of neighboring pixels (weights are denoted by the edge width). In (1c), a walk is started from a superposition of seeds with given (here green) label (colorful dots denote the probability of measuring walker at given pixel) and, in (1d), proceeds for several steps. In (1e), when the walk ends the limiting distribution (CTQW-LD) or the last state (CTQW-OS) is calculated. In (1f), similar procedure is repeated for each (here red) label. In (1g), final distributions of each walk are compared and each pixel is assigned with label of the walk that had the highest probability of being measured at given pixel (1h).

MN that would be hermitian (this ensures that the operator $U(t)$ is unitary). It also has to yield satisfying results in terms of position space exploration by the walker according to the content of the image.

The idea is to construct the Hamiltonian based on the weights matrix defined by the Equation 1 and a free real parameter γ that determines the rate of spread of the quantum walk:

$$H_{ij,kl} = \begin{cases} -\gamma w_{ij,kl}, & \text{if } i \neq k \vee j \neq l. \\ \gamma \sum_{k',l'} w_{ij,k'l'}, & \text{if } i = k \wedge j = l. \end{cases} \quad (5)$$

Please notice, that since weight matrix is symmetrical ($w_{ij,kl} = w_{kl,ij}$) and its values are real numbers, the constructed matrix H is a hermitian matrix and therefore the operator $U(t)$ is unitary.

4 Algorithms description

Based on the elaborated continuous time walk model, here are proposed two versions of the algorithm for image segmentation: with limiting distribution (CTQW-LD) and one shot version (CTQW-OS).

Limiting distribution version. Due to the application of unitary operators, quantum random walks do not converge to a stationary distribution, as it is the case with the classical random walks. Instead, it was proposed by Aharonov et al. [1] to consider *average distribution* and *limiting distribution*.

Definition 1. [1] Consider the probability distribution $\mathbf{P}_t(|\psi_0\rangle)$ on the nodes of the graph after t steps of quantum walk starting from the initial state $|\psi_0\rangle$:

$$P_t^i(|\psi_0\rangle) = \|\langle i|\psi_t\rangle\|^2. \quad (6)$$

Then the **average distribution** $\bar{P}_T(|\psi_0\rangle)$ is the mean over the distributions in each time step until T :

$$\bar{P}_T(|\psi_0\rangle) = \frac{1}{T} \sum_{t=0}^{T-1} \mathbf{P}_t(|\psi_0\rangle). \quad (7)$$

Notice that the *average distribution* can be understood as a measure of how long the quantum walker spends at each of the position states throughout the first T steps of the walk. It was shown in [1] that, for any initial state $|\psi_0\rangle$, the average distribution converges as the time approaches infinity and the limit is denoted as *limiting distribution*.

Definition 2. [1] **Limiting distribution** of the quantum random walk starting from the initial state $|\psi_0\rangle$:

$$\pi(|\psi_0\rangle) = \lim_{T \rightarrow \infty} \bar{P}_T(|\psi_0\rangle). \quad (8)$$

The *limiting distribution* is not unique, but depends on the initial state.

The version of the algorithm using limiting distribution (CTQW-LD) is shown in the Listing 1. For each label the separate quantum walk is performed. After each step of the walk the probability distribution of measuring the walker at given position is recorded. In our simulations we could directly read the walker state after each step, as the calculations were performed using matrix algebra. Also after introducing the optimization (described below) we obtained the state after given number of steps by changing the equation parameters. However, on a quantum device retrieval of the quantum state is not straightforward, This is achieved by performing so called state tomography repeating the walk for given number of steps multiple times and making a measurement which results in finding the walker at some position on the image. The number of samplings grows with the number of basis states (the image size) and the desired accuracy of the reconstructed state. This allows to retrieve the probability distribution

Algorithm 1: Image segmentation using continuous time quantum walk with limiting distribution solution (CTQW-LD)

Input: Image $A = (a_{ij}) \in V^{M \times N}$, where $V = \mathbb{R}^c$ and c is number channels for a pixel,
Set of labels L ,
List of seeds $S = \{(p_1, l_1), \dots, (p_s, l_s)\}$, where $p_k \in \mathbb{Z}_M \times \mathbb{Z}_N$ is the seed position and $l_k \in L$ is its label.

Parameters: β - boundary strengthening parameter,
 γ - rate of spread of the walker,
 T - number of steps of the walk.

Output: Segmented image $B = (b_{ij}) \in L^{M \times N}$.

```

1  $W \leftarrow \text{calculate\_weights}(A, \beta)$ ; // according to the formula 1
2  $H \leftarrow \text{construct\_hamiltonian}(W, \gamma)$ ; // based on the equation 5
3  $U \leftarrow \text{construct\_operator}(H)$ ; //  $U = U(1) = e^{-iH}$ 
4 foreach  $(l_i) \in L$  do // perform walk for each label
5    $S' \leftarrow \text{seeds\_subset}(S, l_i)$ ; // coordinates of seeds with label  $l_i$ 
6    $|\psi(0)\rangle \leftarrow \frac{1}{\sqrt{|S'|}} \sum_{p \in S'} |p\rangle$ ; // set initial state to the seed
7   for  $t \leftarrow 1$  to  $T$  do
8      $|\psi(t)\rangle \leftarrow U |\psi(t-1)\rangle$ ; // perform a move
9      $D_t \leftarrow \text{retrieve\_position}(|\psi(t)\rangle)$ ; // prob. dist. of measuring walker
      at each position
10  end
11   $LD_i \leftarrow \frac{1}{T} \sum_{t=1}^T D_t$ ; // limiting distribution of the walk
12 end
13 for  $i \leftarrow 0$  to  $M-1$  do
14   for  $j \leftarrow 0$  to  $N-1$  do
15      $k \leftarrow \text{find\_seed}(LD, i, j)$ ;
      // find seed that on average has the highest
      probability of being measured at pixel  $ij$ 
16      $B_{ij} \leftarrow l_k$ ; // assign label to the pixel
17   end
18 end

```

of the walker after each time step. Upon finishing each walk (the walk for each label) the limiting distribution of the walk is calculated (due to the fact that there is no possibility of running a walk for infinite number of steps, the limiting distribution is approximated with average distribution of T steps). Finally, the result is obtained classically, as each pixel is assigned with label of the seed, for which walker starting from that seed expressed the highest average probability of being measured at that pixel.

One shot version of the algorithm (CTQW-OS) is shown in the Listing 2. This solution is almost identical to the CTQW-LD, but instead of calculating the limiting distribution of each walk, only the distribution after last state is used.

Algorithm 2: Image segmentation using continuous time quantum walk – one shot solution (CTQW-OS)

Input: Image $A = (a_{ij}) \in V^{M \times N}$, where $V = \mathbb{R}^c$ and c is number channels for a pixel,
Set of labels L ,
List of seeds $S = \{(p_1, l_1), \dots, (p_s, l_s)\}$, where $p_k \in \mathbb{Z}_M \times \mathbb{Z}_N$ is the seed position and $l_k \in L$ is its label.

Parameters: β - boundary strengthening parameter,
 γ - rate of spread of the walker,
 T - number of steps of the walk.

Output: Segmented image $B = (b_{ij}) \in L^{M \times N}$.

```

1  $W \leftarrow \text{calculate\_weights}(A, \beta)$ ; // according to the formula 1
2  $H \leftarrow \text{construct\_hamiltonian}(W, \gamma)$ ; // based on the equation 5
3  $U \leftarrow \text{construct\_operator}(H)$ ; //  $U = U(1) = e^{-iH}$ 
4 foreach  $(l_i) \in L$  do // perform walk for each label
5    $S' \leftarrow \text{seeds\_subset}(S, l_i)$ ; // coordinates of seeds with label  $l_i$ 
6    $|\psi(0)\rangle \leftarrow \frac{1}{\sqrt{|S'|}} \sum_{p \in S'} |p\rangle$ ; // set initial state to the seed
7   for  $t \leftarrow 1$  to  $T$  do
8      $|\psi(t)\rangle \leftarrow U |\psi(t-1)\rangle$ ; // perform a move
9   end
10   $D_i \leftarrow \text{retrieve\_position}(|\psi(T)\rangle)$ ;
    // prob. dist. of measuring walker at
    // each position after last step
11 end
12 for  $i \leftarrow 0$  to  $M-1$  do
13   for  $j \leftarrow 0$  to  $N-1$  do
14      $k \leftarrow \text{find\_seed}(D, i, j)$ ;
        // find seed that has the highest probability of being
        // measured at the end of the walk at pixel  $ij$ 
15      $B_{ij} \leftarrow l_k$ ; // assign label to the pixel
16   end
17 end

```

As it will be shown in the next section, this method gives results with a bit worse segmentation accuracy, but for careful choice of parameters the difference is ommitable. However, it does not require obtaining the probability distribution after each step, which is a tremendous profit, especially if the algorithm would be executed on a real quantum device. Quantum state tomography, which is used to retrieve the quantum state of the walker is an expensive operation, but in the CTQW-OS version of the algorithm it needs to be performed only once for each label, what is a great advantage over the CTQW-LD algorithm.

Optimization. In order to improve the performance of simulations of the algorithms on classical computers there have been introduced some optimizations.

First, each walk was performed in parallel in separate processes. Additionally, it was tested that sampling the walker distribution not after each step, but for about every 100^{th} step in the range $[1, T]$ gives almost undisturbed results. Therefore, instead repeating evolution operator U , we used the numerical solver of Lindblad master equation [25].

5 Evaluation and Results

The evaluation of the proposed solutions could not, obviously, be performed on the real quantum device, as the requirements e.g. the number of qubits needed to express the walker state, the level of customization the quantum operators etc. are beyond the capabilities of currently accessible quantum computers e.g. IBM-Q [11] or Rigetti [20]. Therefore, there had to be performed a simulation of quantum computations on a classical machine. The project code has been prepared using Python language (version 3.4.3) with *QuTiP* open-source Python library for simulation of quantum systems (version 4.2.0). The program was executed on a computer with processor Intel Core i5-4210M and 16GB of RAM. We used the Berkeley Segmentation Dataset and Benchmark (BSDS500) [16] dataset which contains a multitude of photographs, with corresponding contour images performed manually by people.

The most important parameter for the proposed solution is the set of seeds that needs to be provided separately. We considered one set created manually, by careful choice of evenly spread pixels with a bit higher concentration around borders, especially the weak and unclear boundaries (these sets contained about 0.2% pixels of the whole image); and three sets that were prepared automatically by drawing uniformly a fraction of pixels from the image and assigning them with the labels based on the ground truth segmentation from BSDS500 dataset (with the concentrations at the level of 0.2%, 0.5% and 1%, respectively). There are also other three free parameters:

Parameter β – it is used during weights calculation; the greater its value, the more significant the stronger differences between pixels are. So the high β parameter makes the walker stay closer to its seed rather than disperse and explore further regions. However, the change of this parameter does not significantly influence the outcome segmentation.

Parameter γ – the rate of spread of the walker. The greater the value of this parameter, the greater is the area explored by the walker upon a single application of evolution operator. This means that one could set γ to a high value to limit the number of steps required. Indeed, increase of this parameter results in obtaining better outcome for smaller number of steps. But for relatively high values of γ , as the number of steps grows, the segmentation accuracy drops almost as rapidly as it reaches its peak value.

Parameter T – determines the number of steps (number of applications of the evolution operator). Too small values result in poor image exploration by walkers and appearance of some blank areas. Too high values result in walkers diffusion to neighboring segments that reduces the segmentation quality over time. It is

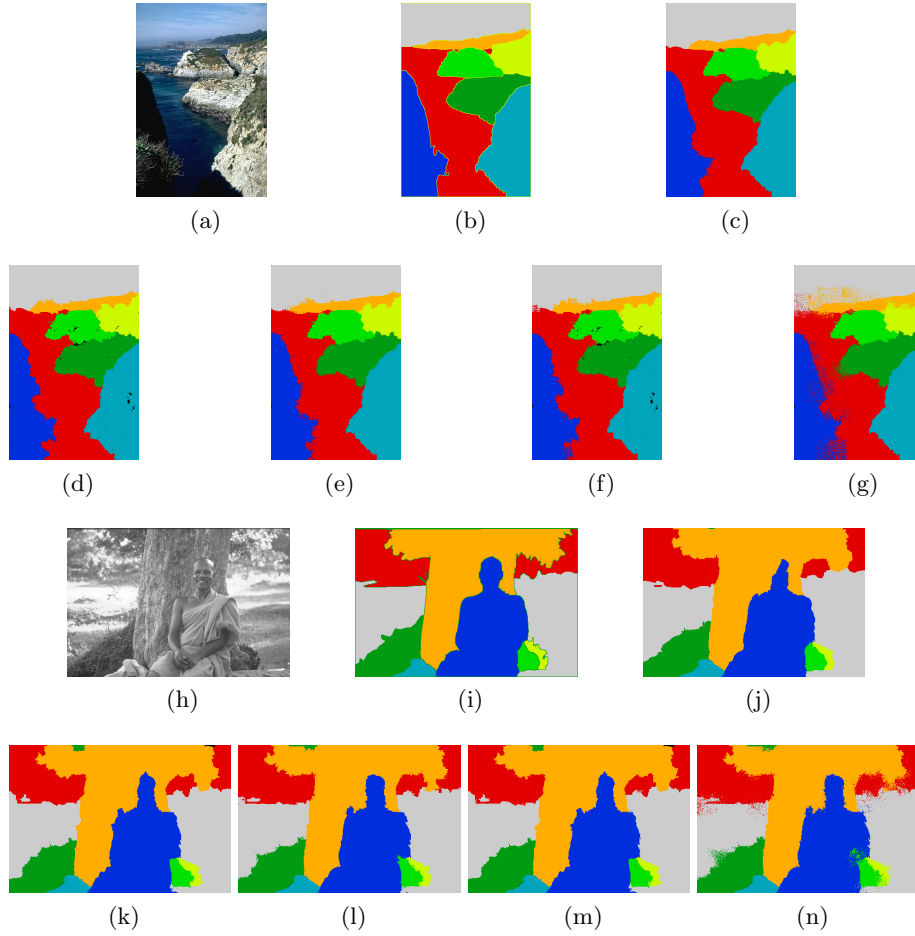


Fig. 2: Comparison of results on an example of color and grayscale test images. The original images are shown in (2a, 2h) and the manual segmentations in (2b, 2i). Results of Grady's algorithm is shown in (2c, 2j). Results of CTQW-LD with configuration: $\beta = 150.0, \gamma = 0.001, T = 5000$ is shown in (2d, 2k), CTQW-LD with the most optimal configuration: $\beta = 150.0, \gamma = 0.001, T = 20000$ in (2e, 2l), CTQW-OS with the most optimal configuration: $\beta = 150.0, \gamma = 0.001, T = 5000$ in (2f, 2m) and finally, CTQW-OS with configuration: $\beta = 150.0, \gamma = 0.001, T = 20000$ in (2g, 2n).

especially important for the CTQW-OS algorithm that strongly suffers from too high T value.

The optimal configurations for both of algorithms have been chosen as follows: $\beta = 150.0, \gamma = 0.001, T = 20000$ for CTQW-LD solution and $\beta = 150.0, \gamma = 0.001, T = 5000$ for CTQW-OS method.

In addition to the ground truth segmentation obtained from the manually created image outlines from BSDS500 dataset, we have also prepared a reference implementation of the classical random walks algorithm [9]. Each segmentation method was then compared against the ground truth images and finally there was calculated the percentage of pixels with consentaneous labels.

The representative results of proposed algorithms are presented in the Fig. 2, which shows segmentations of two sample images: a seashore landscape in color in the Fig. 2a and a man under a tree in grayscale in the Fig. 2h. Then there are reference segmentations: the Fig. 2b and the Fig. 2i are ground truth images included in BSDS500 dataset, while the Fig. 2c and the Fig. 2j were prepared using Grady's algorithm. The last four images for each photo were obtained with algorithms proposed in this paper: the Fig. 2d and Fig. 2k present results of CTQW-LD solution with configuration: $\beta = 150.0, \gamma = 0.001, T = 5000$, Fig. 2e and Fig. 2l come from CTQW-LD algorithm with optimal configuration: $\beta = 150.0, \gamma = 0.001, T = 20000$, while Fig. 2f. Fig. 2m were obtained with CTQW-OS method with optimal configuration: $\beta = 150.0, \gamma = 0.001, T = 5000$. Fig. 2g and Fig. 2n show CTQW-OS with configuration: $\beta = 150.0, \gamma = 0.001, T = 20000$. As it can be seen, 5000 steps is too little for the CTQW-LD algorithm (visible non-covered black areas in Fig. 2e and Fig. 2l). For CTQW-OS, if the walk duration is too long, the walks from adjacent segments tend to diffuse causing deterioration of the segmentation quality (Fig. 2g, Fig. 2n). As could be expected CTQW-OS is a bit inferior to CTQW-LD. Nevertheless, the difference is not obvious, due to careful choice of parameters (especially number of steps), while the gain in terms of performance is tremendous. CTQW-LD looks comparably, if not better than Grady's method (please notice the difference around the head of the man under tree). Obviously, all of these solutions are imperfect which can be clearly seen in the areas of weak and blurred margins (like, the line of horizon or leaves of the tree).

	manual seeds 0.2%	automatic seeds 0.2%	automatic seeds 0.5%	automatic seeds 1.0%
Grady ($\beta = 90.0$)	93.50	90.25	93.50	94.88
Grady ($\beta = 150.0$)	93.63	90.38	93.50	94.88
CTQW-LD ($\beta = 150.0, \gamma = 0.001, T = 5000$)	87.30	88.23	93.34	94.89
CTQW-LD ($\beta = 150.0, \gamma = 0.001, T = 20000$)	93.40	90.89	93.48	94.72
CTQW-OS ($\beta = 150.0, \gamma = 0.001, T = 5000$)	87.18	88.22	93.19	94.65
CTQW-OS ($\beta = 150.0, \gamma = 0.001, T = 20000$)	91.84	90.24	91.76	92.76

Table 1: Average accuracy (over test images) of developed algorithms as well as Grady's method for different sets of seeds.

Table 1 presents average accuracy (over test images) of developed algorithms in comparison to Grady's method for different sets of seeds. It clearly shows that the segmentation accuracy increases with the size of seeds set. Manually chosen seeds give better results than the same amount of automatically drawn ones, due

to their careful selection. More accurate of the proposed solutions is CTQW-LD algorithm, which gives almost as good results as reference algorithm or even outperforms it in some cases. The other method provides slightly worse accuracy. The last two rows show how sensitive to the walk duration is the CTQW-OS solution; the accuracy drops significantly if the number of steps is too high. Nonetheless, for optimal configuration of parameters the results of both developed methods might be considered satisfactory. The only unfavorable aspect is the quite considerable amount of seeds needed to obtain a decent segmentation accuracy. As the 0.2% seems to be a bit too little, there has to be specified about 0.5% seeds, which for image of size 320x480 means that over seven hundred of pixels need to be labeled.

The average execution times for test images of size 320x480 and optimal configurations were as follows: about 5-6 minutes for CTQW-LD and around a minute in case of CTQW-OS method. Along with the growth of γ or T the duration of the calculations increases linearly. For comparison Grady algorithm for the same samples requires about 30 seconds. This means that developed solutions are comparable both in terms of accuracy as well as performance to Grady algorithm. Of course, these are the execution times of simulation on classical computer which would not practically matter during transition to a quantum device, nevertheless the proposed results, especially CTQW-OS, are expected to do well also in quantum conditions.

6 Summary, Conclusion and Future Work

In this paper, there have been presented two novel quantum algorithms for performing image segmentation based on continuous time quantum walks: one calculating the limiting distribution of the walk and the other utilizing only its last state. Both methods have been closely examined and tested using simulation of quantum computations on a classical computer. The obtained results are satisfactory, of the same accuracy and similar performance as the reference classical Grady's [9] algorithm. A crucial difference between simulation and the actual execution of the algorithm on a quantum computer is the fact, that during simulation, there is always access to the whole quantum state. In real conditions the actual state of the particle is not known until measurement, which reduces it to one of the basis states according to probability distribution determined by the current state. So, to gain the knowledge on the walker state after a given number of steps, one needs to perform the same walk multiple times and measure it in order to reconstruct the probability distribution. In these conditions the CTQW-LD solution is not very effective in quantum case – in order to obtain the limiting distribution the experiment has to be repeated multiple times for different numbers of steps. For this reason, it would be vastly beneficial to apply the CTQW-OS algorithm, which requires the construction of the probability distribution only once per walk.

The segmentation presented in this work is a seeded-based method. User has to specify a set of pixels with assigned segment to which they belong which is

not an automatic approach. Also to achieve good results a considerable amount of seeds needs to be provided. Therefore, future work in this field might concern elaboration of either a way to limit the number of required seeds or a method for automatic specifying the seeds, similar to those currently applied in medical diagnostic images segmentation [8]. This would allow to significantly increase the user experience while using the proposed solutions for image segmentation.

The future work on this matter could consider translating the algorithm (for some small size images) on a quantum device - firstly on a quantum simulator like IBM Q simulator or Atos QLM and further to make an attempt to run it on a quantum computer (e.g. IBM Q or Rigetti). This would require disassembling the evolution operator into basic one and two qubit quantum gates, as only those are handled by current quantum devices, as well as taking into account error correction strategies to mitigate the risk of quantum state decoherence.

Acknowledgements This research was partly supported by the National Science Centre, Poland – project number 2014/15/B/ST6/05204.

References

1. Aharonov, D., Ambainis, A., Kempe, J., Vazirani, U.: Quantum Walks On Graphs. Proceedings of the thirty-third annual ACM symposium on Theory of computing - STOC '01 pp. 50–59 (2000). <https://doi.org/10.1145/380752.380758>, <http://arxiv.org/abs/quant-ph/0012090>
2. Ambainis, A., Kempe, J., Rivosch, A.: Coins Make Quantum Walks Faster (2004), <http://arxiv.org/abs/quant-ph/0402107>
3. Ashwin, N., Ashvin, V.: Quantum Walk on the Line. Tech. rep. (2000)
4. Balu, R., Castillo, D., Siopsis, G.: Physical realization of topological quantum walks on IBM-Q and beyond (2017). <https://doi.org/10.1088/2058-9565/aab823>, <http://arxiv.org/abs/1710.03615>
5. Caraiman, S., Manta, V.I.: Image segmentation on a quantum computer. Quantum Information Processing **14**(5), 1693–1715 (2015). <https://doi.org/10.1007/s11128-015-0932-1>
6. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**(4), 834–848 (2017). <https://doi.org/10.1109/TPAMI.2017.2699184>
7. Childs, A.M., Cleve, R., Deotto, E., Farhi, E., Gutmann, S., Spielman, D.A.: Exponential algorithmic speedup by quantum walk pp. 59–68 (2002). <https://doi.org/10.1145/780542.780552>, <http://arxiv.org/abs/quant-ph/0209131>
8. Fasihi, M.S., Mikhael, W.B.: Overview of Current Biomedical Image Segmentation Methods. In: Proceedings - 2016 International Conference on Computational Science and Computational Intelligence, CSCI 2016. pp. 803–808 (2017). <https://doi.org/10.1109/CSCI.2016.0156>
9. Grady, L.: Random walks for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **28**(11), 1768–1783 (2006). <https://doi.org/10.1109/TPAMI.2006.233>
10. Grady, L., Schwartz, E.L.: Isoperimetric graph partitioning for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **28**(3), 469–475 (2006). <https://doi.org/10.1109/TPAMI.2006.57>

11. Hebenstreit, M., Alsina, D., Latorre, J.I., Kraus, B.: Compressed quantum computation using a remote five-qubit quantum computer. *Physical Review A* **95**(5) (2017). <https://doi.org/10.1103/PhysRevA.95.052339>
12. Li, H.S., Qingxin, Z., Lan, S., Shen, C.Y., Zhou, R., Mo, J.: Image storage, retrieval, compression and segmentation in a quantum system. *Quantum Information Processing* **12**(6), 2269–2290 (2013). <https://doi.org/10.1007/s11128-012-0521-5>
13. Li, J., Li, X., Yang, B., Sun, X.: Segmentation-based image copy-move forgery detection scheme. *IEEE Transactions on Information Forensics and Security* **10**(3), 507–518 (2015). <https://doi.org/10.1109/TIFS.2014.2381872>
14. Ma, Z., Manuel, J., Tavares, R.S., Natal, R., Mascarenhas, T.: A Review of Algorithms for Medical Image Segmentation and their Applications to the Female Pelvic Cavity . *Computer methods of biomechanical and biomedical engineering* **13**(2), 235–246 (2010). <https://doi.org/10.1080/10255840903131878>
15. Magniez, F., Nayak, A., Roland, J., Santha, M.: Search via Quantum Walk. *SIAM Journal on Computing* **40**(1), 142 (2011). <https://doi.org/10.1137/090745854>, <http://arxiv.org/abs/quant-ph/0608026>
16. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. *Proceedings of the IEEE International Conference on Computer Vision* **2**, 416–423 (2001). <https://doi.org/10.1109/ICCV.2001.937655>
17. Mat Isa, N., Salamah, S., Ngah, U.: Adaptive fuzzy moving K-means clustering algorithm for image segmentation. *IEEE Transactions on Consumer Electronics* **55**(4), 2145–2153 (2009). <https://doi.org/10.1109/TCE.2009.5373781>, <http://ieeexplore.ieee.org/document/5373781/>
18. Mortensen, E.N., Barrett, W.A.: Interactive Segmentation with Intelligent Scissors. *Graphical Models and Image Processing* **60**(5), 349–384 (1998). <https://doi.org/10.1006/gmip.1998.0480>, <http://linkinghub.elsevier.com/retrieve/pii/S1077316998904804>
19. Otsu, N.: A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics* **9**(1), 62–66 (1979). <https://doi.org/10.1109/TSMC.1979.4310076>, <http://ieeexplore.ieee.org/document/4310076/>
20. Otterbach, J.S., Manenti, R., Alidoust, N., Bestwick, et al.: Unsupervised Machine Learning on a Hybrid Quantum Computer (2017), <http://arxiv.org/abs/1712.05771>
21. Pal, N.R., Pal, S.K.: A review on image segmentation techniques. *Pattern Recognition* **26**(9), 1277–1294 (1993). [https://doi.org/10.1016/0031-3203\(93\)90135-J](https://doi.org/10.1016/0031-3203(93)90135-J)
22. Sharma, S., Vishwamittar: Brownian motion problem: Random walk and beyond. *Resonance* **10**(8), 49–66 (2005). <https://doi.org/10.1007/BF02866746>, <http://link.springer.com/10.1007/BF02866746>
23. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(8), 888–905 (2000). <https://doi.org/10.1109/34.868688>
24. Venegas-Andraca, S.E., Ball, J.L.: Processing images in entangled quantum systems. *Quantum Information Processing* **9**(1), 1–11 (2010). <https://doi.org/10.1007/s11128-009-0123-z>
25. Weinberg, S.: Quantum mechanics without state vectors. *Phys. Rev. A* **90**, 042102 (Oct 2014). <https://doi.org/10.1103/PhysRevA.90.042102>, <https://link.aps.org/doi/10.1103/PhysRevA.90.042102>
26. Yuheng, S., Hao, Y.: Image Segmentation Algorithms Overview. *CoRR abs/1707.0* (2017), <http://arxiv.org/abs/1707.02051>