How to Select Good Neighboring Images in Depth-Map Merging Based 3D Modeling

Shuhan Shen and Zhanyi Hu

Abstract— **Depth-map merging based 3D modeling is an effective approach for reconstructing large-scale scenes from multiple images. In addition to generate high quality depth maps at each image, how to select suitable neighboring images for each image is also an important step in the reconstruction pipeline, unfortunately to which little attention has been paid in the literature untill now. This paper is intended to tackle this issue for large scale scene reconstruction where many unordered images are captured and used with substantial varying scale and view-angle changes. We formulate the neighboring image selection as a combinatorial optimization problem and use the quantum-inspired evolutionary algorithm to seek its optimal solution. Experimental results on the ground truth data set show that our approach can significantly improve the quality of the depth-maps as well as final 3D reconstruction results with high computational efficiency.**

Index Terms— **Neighboring image selection, depth-map computation, 3D modeling.**

I. Introduction

IMAGE based 3D modeling of objects and scenes is an active research field nowadays, and has emerged as a powerful tool for many applications, such as architecture heritage preservation, city-scale modeling, and so on. The ultimate goal of image based 3D modeling is to provide comparable accuracy and lower cost than relatively expensive laser scanners (LIDAR). According to [1], multiple view 3D modeling algorithms could be divided into four classes, called voxel based methods [2]–[4], surface evolution based methods [5]–[8], feature point growing based methods [9]–[13], and depth-map merging based methods [14]–[22]. Among these classes, the depth-map merging based methods have been proved to be more adapted to large-scale scenes, and the key of such methods is to generate a high-quality depth-map at each image at first, then to merge the depth-maps into a complete 3D model.

Fig. 1 shows the pipeline of a typical depth-map merging based 3D modeling system. The input of this system are multiple images captured from different positions and viewpoints, and a certain overlap between neighboring images is required. The system first uses Structure from Motion (SfM) algorithm, like Bundler [23], to determine the focal length, position and orientation of each camera, i.e. to calibrate the camera internal and external parameters. Sometimes extra information, like GPS and IMU, is available, and this could improve the calibration accuracy and efficiency [24]. After camera calibration, the system try to compute depth-map at each image followed by a refinement process to enforce depth consistency over neighboring views. Finally, all the depth-maps are back projected to 3D and merged together. The output of the system is a 3D model represented either by a dense 3D point cloud or 3D triangulated meshes.

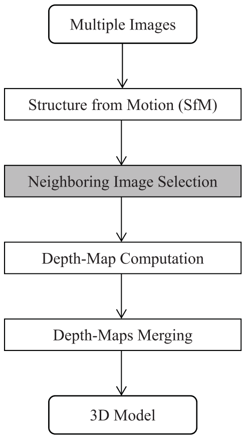


Fig. 1. The pipeline of a typical depth-map merging based 3D modeling system. This paper is focused on the optimal neighboring image selection shown in gray.

Obviously, the accuracy of the depth-map computed at each image is a key factor of a depth-map merging based 3D modeling system. A lot of research has been done to investigate accurate and efficient depth-map computation algorithms. However, how to select appropriate neighboring images for each reference image is also an important factor to which little attention has been paid till now to our knowledge. The neighboring image selection is a relatively easy task for street-side view cameras on the vehicle [25]–[27] or cameras in a controlled environment like the Middlebury benchmark data [1], but needs to be carefully designed for large-scale scenes where unordered images are captured at various locations and scales. Furukawa et al. [12] proposed a view clustering algorithm which divides an image set into overlapping view

Manuscript received July 1, 2013; revised September 26, 2013; accepted November 5, 2013. Date of publication November 12, 2013; date of current version November 28, 2013. This work was supported in part by the Natural Science Foundation of China under Grants 61105032 and 61333015, and in part by the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant XDA06030300. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Xin Li.

The authors are with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: shshen@nlpr.ia.ac.cn; huzy@nlpr.ia.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2013.2290597

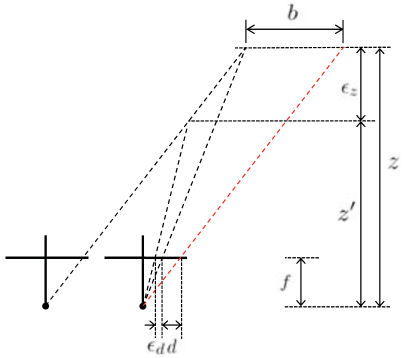


Fig. 2. The depth uncertainty of a rectified stereo image pair.

clusters, after which a feature point growing based MVS algorithm is used to reconstruct each cluster in parallel. However this image selection algorithm is not designed for depth maps. Li et al. [19] introduced a neighboring image selection algorithm by computing the angle between principle view directions and the distance between camera optical centers. This method is very simple and straightforward but only suited for regular arrayed cameras. Goesele et al. [10] computed a global score for each view and used greedy algorithm to select neighboring images. Bailer et al. [21] improved the method in [10] by modifying the view score function to increase the depth-map's accuracy. However, these two methods used greedy algorithm as the optimization tool which in most cases generates a suboptimal solution. This paper tries to investigate an optimal neighboring image selection algorithm and shows that our algorithm can significantly improve the quality of the depth-maps as well as the final 3D reconstruction results.

The rest of the paper is organized as follows. The neighboring image selection is formulated as a combinatorial optimization problem and a novel objective function is proposed in Section II. Then the Quantum-inspired Evolutionary Algorithm (QEA) is used solve this combinatorial optimization problem in Section III. Finally, experimental results on ground truth data set are presented in Section IV, followed by some concluding remarks in Section V.

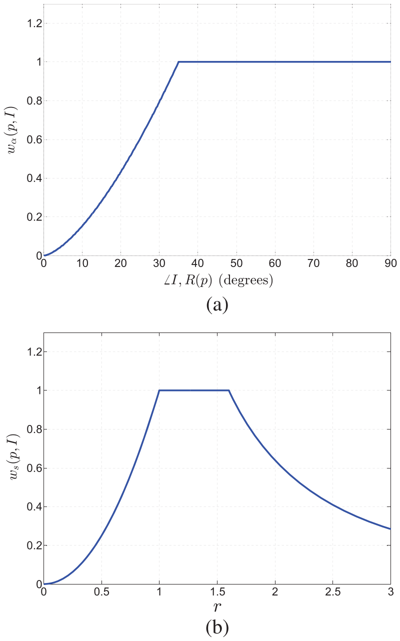
II. Formulating the Neighboring Image Selection as a Combinatorial Optimization Problem

In the depth-map merging based 3D modeling pipeline (Fig. 1), the SfM step computes the internal (focal length) and external (position and oritation) parameters of each image, as well as a set of sparse 3D points and their visibilities as a by-product. Now the question is: given the camera poses and a set of stable sparse feature points, how to define a *good* neighboring image for a reference view?

In order to answer this question, a quantitative measure is needed. As shown in Fig. 2, consider a rectified stereo image pair with focal length f and baseline b, and let d be the disparity, z be the depth of the triangulated point, ϵ_d be the correspondence error which describes the error from incorrect matches and sub-pixel accuracy of correct matches, and ϵ_z be the triangulated depth error, then the depth uncertainty can be written in terms of the disparity error as [28]:

$$\epsilon_z = z - z' = \frac{bf}{d} - \frac{bf}{d+\epsilon_d} \approx \frac{z^2}{bf} \cdot \epsilon_d \tag{1}$$

where, the "≈" step is obtained by using the first order taylor expansion at point $\epsilon_d = 0$.



(a)

(b)

Fig. 3. The graph of (a) $w_\alpha(p, I)$ and (b) $w_s(p, I)$.

Hence, the depth error ϵ_z is mainly a function of the ray intersection angle. Increasing the ray angle can decrease the depth error, but a larger angle may reduce the common visible area of the stereo pair and introduce more mismatches. In [19], [20], [22], the ray angles computed by sparse feature points from SfM are utilized to select the best neighboring view. However, as shown in [29], adding more views for multi-view triangulation could significantly reduce the depth uncertainty compared to stereo triangulation. Thus, in the neighboring image selection process it is better to select multiple neighboring images rather than selecting a single one. Here comes the second question: how to find the best neighboring images?

Intuitively, good neighboring images should have three characteristics. First, they should have sufficient ray intersection angles with the reference image at feature points according to Eq. (1) in order to guarantee the reconstruction accuracy. Second, they should have equal or higher resolution than the reference image at feature points in order to capture texture details. Third, they should sparsely and uniformly cover the visible feature points in the reference image. To achieve this, some neighboring image selection methods [10], [21] have been proposed.

Goesele et al. [10] first introduced a neighboring image selection algorithm by computing a global score for each view within a set of candidate neighboring images and used greedy algorithm to grow the neighboring images. Based on [10], Bailer et al. [21] proposed an improved selection algorithm

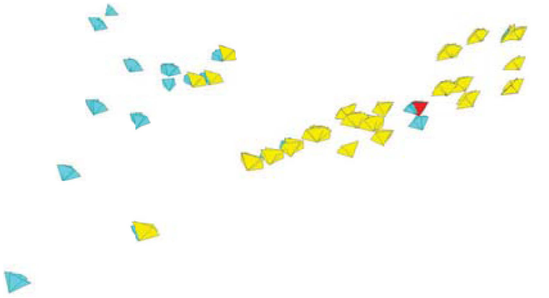


Fig. 4. A reference image R and its searching domain $\mathbf{C}$. Here, the red triangle cone represents R, yellow cones represent cameras in $\mathbf{C}$, and blue cones are other cameras. In this graph, $|\mathbf{C}| = 59$.

Algorithm 1 QEA Based Neighboring Image Selection

```
$t \leftarrow 0$
Initialize the population $Q(t)$
Obtain the observed population $P(t)$ by observing $Q(t)$
Repair $P(t)$
Evaluate $P(t)$
Store $P(t)$ into $B(t)$
**while** Termination-condition = $false$ **do**
  $t \leftarrow t + 1$
  Obtain the observed population $P(t)$ by observing $Q(t-1)$
  Repair $P(t)$
  Evaluate $P(t)$
  Store the best solutions among $B(t-1)$ and $P(t)$ into $B(t)$
  Update $Q(t)$
  Store the global best solution $\mathbf{b_g}$ among $B(t)$
  **if** Migration-condition = $true$ **then**
    Migrate $\mathbf{b_g}$ to $B(t)$
  **end if**
**end while**
```

which was reported to be able to improve the depth-map's accuracy and completeness compared to [10].

In [21], given a reference image R and a set of neighboring images $\mathbf{N}$, a score for each view $I \in \mathbf{N}$ is computed as:

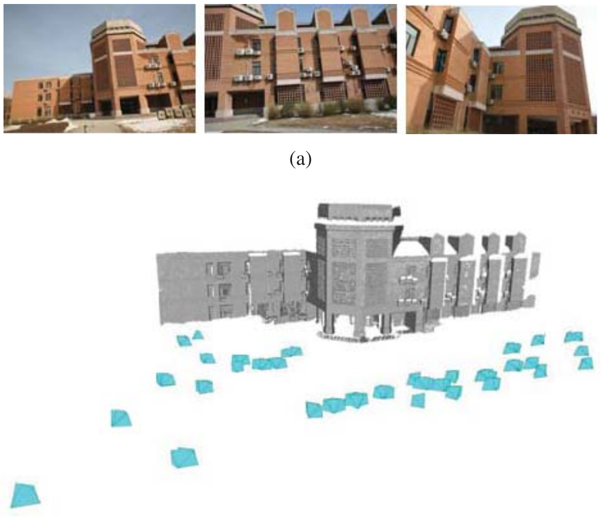
$$g_R(I) = \sum_{p \in F_R \cap F_I} \omega_\alpha(p)\omega_s(p)\omega_c(p) \tag{2}$$

where, F_X is the set of feature points visible in image X, $\omega_\alpha(p)$ is the angle weighting, $\omega_s(p)$ is the scale weighting, and $\omega_c(p)$ is the covering weighting.

The angle weighting is defined as:

$$\omega_\alpha(p) = \min\left(\frac{\angle I, R(p)}{\alpha_{\max}}, 1\right)^{1.5} \cdot \prod_{J \in \mathbf{N} \setminus I,\ p \in F_I \cap F_J} \min\left(\frac{\angle I, J(p)}{\beta_{\max}}, 1\right) \tag{3}$$

where, $\angle I, X(p)$ is the ray intersection angle at feature point p from the camera center of image I and X, $\alpha_{\max}$ is set to 35 degrees, and $\beta_{\max}$ is set to 14 degrees. This weighting favors the image whose ray intersection angle with R at feature p is bigger than $\alpha_{\max}$ and angles with other images in N are bigger than $\beta_{\max}$.



(a)

(b)

Fig. 5. Sample images, ground truth 3D model, and cameras in the data set. (a) Three sample images in the data set. (b) The ground truth 3D model with cameras.

The scale weighting is defined as:

$$\omega_s(p) = \begin{cases} 0 & r > 1.8 \\ r^2 & 1 < r \le 1.8 \\ 1 & 1/1.6 < r \le 1 \\ (\frac{1.6}{r})^2 & \text{else} \end{cases} \tag{4}$$

where $r = s_R(p)/s_I(p)$, and $s_X(p)$ is the scale at p in image X. $s_X(p)$ is computed as the diameter of a sphere centered at p whose projected diameter in the X equals the pixel spacing. Thus, $r > 1$ means the resolution of I at p is higher than R, and vice versa.

The covering weighting is defined as:

$$\omega_c(p) = \frac{r_I^*(p)}{r_I^*(p) + \sum_{J \in \mathbf{N} \setminus I,\ p \in F_I \cap F_J} r_J^*(p)} \tag{5}$$

where, $r_X^*(p) = \min(s_R(p)^2/s_X(p)^2, 1)$. This weighting favors images that sparsely cover each feature point.

Finally, given the size of the neighboring image set $\mathbf{N}$, [10] and [21] use a greedy algorithm and grow $\mathbf{N}$ by iteratively adding to $\mathbf{N}$ the view with highest score $g_R(I)$ given current $\mathbf{N}$ (initial $\mathbf{N}$ is empty). This neighboring image selection method in [21] is quite efficient, but its main drawback lies in the use of greedy algorithm which usually achieves a suboptimal solution. Besides, since the view score $g_R(I)$ defined in Eq. (2) is only designed for local optimization, using a global optimization algorithm directly on $\sum_{I \in \mathbf{N}} g_R(I)$ always performs worse than [21] as our experimental part shows. Thus a reformulation of the view selection objective function is required for global optimization.

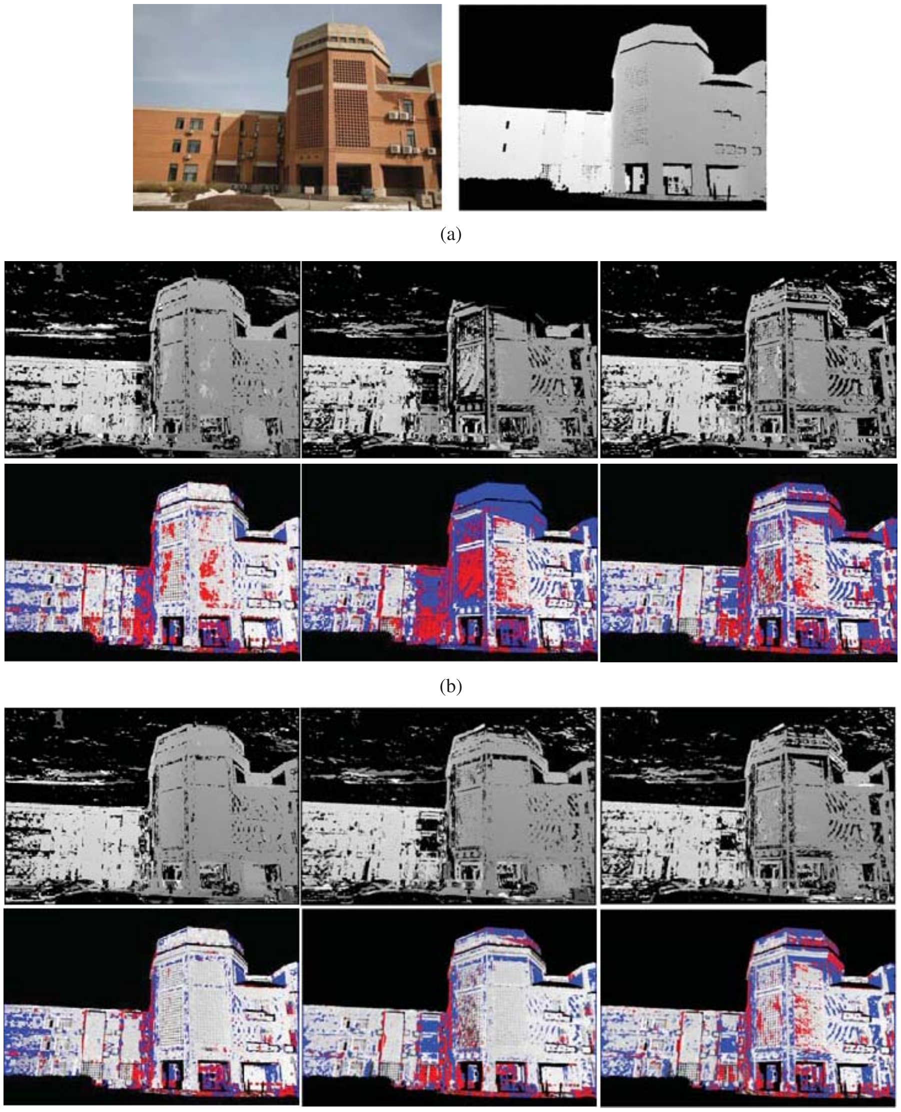
In this paper, we propose a new objective function for image selection and use global optimization method to select the neighboring image set. Given the reference image R and a set of candidate neighboring images $\mathbf{N}$, our objective function is defined as:

$$G_R(\mathbf{N}) = \sum_{p \in F_R} v_b(p)v_q(p)v_c(p) \tag{6}$$

TABLE I

PARAMETER SETTINGS OF OUR PROPOSED METHOD

Parameter	Section	Description	Value
$\alpha_{\max}$	II	Angle threshold for $w_\alpha(p, I)$ in Eq. (8)	35^o
$\beta_{\max}$	II	Angle threshold for $w_\beta(p, I, J)$ in Eq. (10)	15^o
n_{max}	III	The maximal size of $\mathbf{N}$	3 or 6
k	III	The population size of QEA	4
Termination-condition	III	The termination condition of QEA in Algorithm 1	500 generations
Migration-condition	III	The migration condition of QEA in Algorithm 1	Every 100 generations
$\Delta\theta$	III	The rotation angle in Eq. (13)	0.01π


(a)

(b)

(c)

Fig. 6. Depth-maps and error maps computed using three neighboring image selection methods for the 51st image in the data set. (a) shows the 51st image and its ground truth depth-map. (b) and (c) are the depth-maps and error maps computed using three methods with $n_{max} = 3$ and 6 respectively. In both (b) and (c), from left to right: the depth-map computed using our proposed method, the Bailer et al. Greedy method, and the Bailer et al. QEA method respectively. The top row is the depth-map, and the bottom row is the error map in which the blue pixels encode missing depth values, red pixels encode an error e larger than τ_e, and pixels with errors between 0 and τ_e are encoded in gray 255 $\sim$ 0.

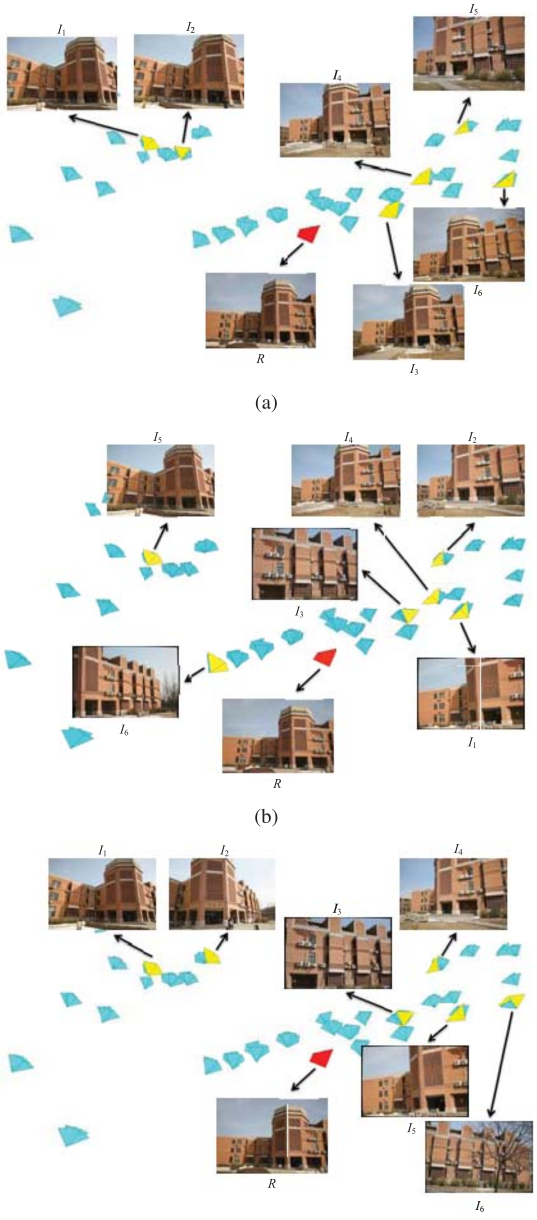
where, $v_b(p)$, $v_q(p)$ and $v_c(p)$ are three weighting functions. For each feature point p in R we define a image set $\mathbf{Q} = \{I \mid p \in F_R \cap F_I,\ I \in \mathbf{N}\}$, that is, $\mathbf{Q}$ contains all images in which p is visible. $v_b(p)$ is a boolean function as:

$$v_b(p) = \begin{cases} 1 & \mid \mathbf{Q} \mid \geq 2 \\ 0 & \text{else} \end{cases} \tag{7}$$

where, $\mid \mathbf{Q} \mid$ is the cardinality of $\mathbf{Q}$. The definition of $v_b(p)$ means that we only consider R's feature points that are visible in at least two neighboring images.

The weighting function $v_q(p)$ measures the reconstruction quality for p as an average of the visible image set $\mathbf{Q}$, as:

$$v_q(p) = \frac{\sum_{I \in \mathbf{Q}} w_\alpha(p, I) w_s(p, I)}{\mid \mathbf{Q} \mid} \tag{8}$$



(a)

(b)

(c)

Fig. 7. The reference image R (the 51st image in the data set) and its neighboring images $I_1 \sim I_6$ selected by three evaluated methods with $n_{max} = 6$. In (a), (b), and (c), the red triangle cone represents the reference camera, yellow cones represent the obtained neighboring cameras, and blue cones are other cameras.

where, $w_\alpha(p, I) = \min(\frac{\angle I, R(p)}{\alpha_{\max}}, 1)^{1.5}$ and $\alpha_{\max}$ is set to 35 degrees, and $w_s(p, I)$ is:

$$w_s(p, I) = \begin{cases} r^2 & r < 1 \\ 1 & 1 \le r \le 1.6 \\ (\frac{1.6}{r})^2 & r > 1.6 \end{cases} \tag{9}$$

where, $r = s_R(p)/s_I(p)$, it is the scale of p defined in Eq. (4).

The graph of $w_\alpha(p, I)$ and $w_s(p, I)$ is shown in Fig. 3. Thus, $v_q(p)$ favors the images that have sufficiently big ray intersection angle ($\ge 35^o$) with R and have a litter higher resolution ($1 \le r \le 1.6$) than R at feature point p.

A good neighboring image set should separate from each other and sparsely cover each feature point in the reference image. Thus, the last weighting function $v_c(p)$ measures the coverage quality of the neighboring image set, as:

$$v_c(p) = \frac{\sum_{I,J \in \mathbf{Q}} w_\beta(p, I, J)}{C_2^{|\mathbf{Q}|}} \cdot \frac{1}{|\mathbf{Q}|} \tag{10}$$

where, $w_\beta(p, I, J) = \min(\frac{\angle I, J(p)}{\beta_{\max}}, 1)$ and $\beta_{\max}$ is set to 15 degrees, and $C_2^{|\mathbf{Q}|} = \frac{|\mathbf{Q}|(|\mathbf{Q}|-1)}{2}$. Here, $v_c(p)$ favors the image set that sparsely covers p and in which each pair of images possesses big enough ray intersection angle ($\ge 15^o$) at p.

Having defined the objective function $G_R(\mathbf{N})$ for a reference image R, we could maximize this function to find its optimal neighboring image set which will be elaborated in the next section.

III. QEA Based Optimization

Let $\mathbf{M}$ denote the image set containing all the images, for each reference image $R \in \mathbf{M}$, we try to find the best neighboring image set $\mathbf{N}$ from all the remaining images $\{\mathbf{M} \backslash R\}$. To improve the efficiency, we first make a subset $\mathbf{C} \subseteq \{\mathbf{M} \setminus R\}$ which excludes all the images that are obviously *not* suited to be R's neighboring images. For each image $I \in \{\mathbf{M} \setminus R\}$, it is included into $\mathbf{C}$ if the following three conditions are satisfied: 1) the number of common visible feature points in I and R is more than 10; 2) the average ray intersection angle $\overline{\angle I, R(p)}$ over all common feature points is bigger than 5 degrees and smaller than 120 degrees; 3) the average scale $\overline{r} = \overline{s_R(p)/s_I(p)}$ over all common feature points is bigger than 0.5 and smaller than 4. This step could significantly reduce the searching domain for finding N, especially for large scale scenes where a large number of images are involved. An example of R and $\mathbf{C}$ is shown in Fig. 4. The best neighboring image set $\mathbf{N}$ is a subset of $\mathbf{C}$, and usually $|\mathbf{C}| \gg |\mathbf{N}|$. The maximal size of $\mathbf{N}$, denoted as n_{max}, is a key parameter. According to [29], increasing the size of $\mathbf{N}$ could decrease the depth uncertainty but will increase the depth-map computation burden. Thus, there is a tradeoff for setting n_{max}, and in this paper we test $n_{max} = 3$ and $n_{max} = 6$.

Given $\mathbf{C}$ and n_{max}, the best $\mathbf{N}$ for each R is computed by maximizing $G_R(\mathbf{N})$ with constraint $|\mathbf{N}| \le n_{max}$. Mathematically, the maximization of $G_R(\mathbf{N})$ over $\mathbf{C}$ could be considered as a 0-1 knapsack problem which is an NP-hard problem. The 0-1 knapsack problem is a combinatorial optimization problem which is described as: given a set of items each of which has a weight and a value, and given a knapsack with limited capacity, then select a subset of the items to maximize the profit. From a knapsack problem's view, each image in $\mathbf{C}$ is an item whose volume is 1, the capacity of the knapsack is n_{max}, the total profit of selected items is $G_R(\mathbf{N})$, and this could be formally described as:

$$\begin{array}{ll} \max & f(\mathbf{x}) \\ \text{subject to} & \sum_{i=1}^{m} x_i \le n_{max} \end{array} \tag{11}$$

where, $\mathbf{x} = (x_1, x_2, \ldots, x_m)$, m is the size of $\mathbf{C}$, i.e. $m = |C|$. x_i is 0 or 1, and $x_i = 1$ means the i-th image in $\mathbf{C}$ is selected for $\mathbf{N}$. Thus, the profit $f(\mathbf{x}) = G_R(\mathbf{N})$, where $\mathbf{N} = \{I_i \mid I_i \in \mathbf{C},\ x_i = 1\}_{i=1,\ldots,m}$.

TABLE II

NUMBERS OF CORRECT, ERROR AND MISSING DEPTH PIXELS USING THREE DIFFERENT NEIGHBORING IMAGE SELECTION METHODS FOR THE 51st IMAGE. THE NUMBER INSIDE THE BRACKET IS THE PERCENTAGE OF THE PIXELS AS OPPOSED TO THE GROUND TRUTH PIXEL COUNTS

	methods	correct pixels	error pixels	missing depth pixels
$n_{max} = 3$	Proposed Method	$5,292,544$ (78.2%)	$683,360$ (10.1%)	$788,656$ (11.7%)
	Bailer et al. Greedy	$3,266,128$ (48.3%)	$1,173,120$ (17.3%)	$2,325,312$ (34.4%)
	Bailer et al. QEA	$3,857,600$ (57%)	$1,034,672$ (15.3%)	$1,872,288$ (27.7%)
$n_{max} = 6$	Proposed Method	$5,629,088$ (83.2%)	$387,824$ (5.7%)	$747,648$ (11.1%)
	Bailer et al. Greedy	$5,001,456$ (73.9%)	$610,768$ (9.1%)	$1,152,336$ (17%)
	Bailer et al. QEA	$4,391,472$ (64.9%)	$765,728$ (11.3%)	$1,607,360$ (23.8%)

TABLE III

AVERAGE PERCENTAGES OF CORRECT/ERROR/MISSING PIXELS AS OPPOSED TO THE GROUND TRUTH PIXEL COUNTS USING THREE METHODS ON 102 IMAGES

	methods	correct	error	missing
$n_{max} = 3$	Proposed Method	67.3%	9%	23.7%
	Bailer et al. Greedy	54.6%	13.7%	31.7%
	Bailer et al. QEA	50.6%	15.1%	34.3%
$n_{max} = 6$	Proposed Method	73.7%	6.9%	19.4%
	Bailer et al. Greedy	67.3%	10.1%	22.6%
	Bailer et al. QEA	58.7%	12.1%	29.2%

Among various methods that can solve the knapsack problem, the Quantum-inspired Evolutionary Algorithm (QEA) [30] is proven to be quite suitable for the $0 - 1$ knapsack problem. QEA is characterized by its Q-bit representation for the individual, the observation process for producing a binary string from the Q-bit individual, the update process by the Q-gate, and the migration process of the Q-bit individuals. For a full description of QEA, one can refer [30] for details.

The basic unit in QEA is defined as a Q-bit as $[\alpha, \beta]^T$, where $|\alpha|^2 + |\beta|^2 = 1$. $|\alpha|^2$ and $|\beta|^2$ gives the probability that this Q-bit could be found in the $'0'$ and $'1'$ state respectively. Then an individual in QEA is given as a string of Q-bits, as:

$$\begin{bmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_m \\ \beta_1 & \beta_2 & \dots & \beta_m \end{bmatrix} \tag{12}$$

where, $|\alpha|_i^2 + |\beta|_i^2 = 1, i = 1, 2, \dots, m$, and $m = | C |$. Here $|\beta|_i^2$ is the probability that the i-th image in C is selected in N. The Q-bit individual has the advantage to represent a linear probabilistic combination of all the states, which makes the population size in QEA quite small compared to the conventional genetic algorithms.

The procedure of the QEA based neighboring image selection is outlined in Algorithm 1.

$Q(t)$ is the population, and $Q(t) = \{\mathbf{q}_1^t, \mathbf{q}_2^t, \dots, \mathbf{q}_k^t\}$. $\mathbf{q}_j^t$ is the j-th Q-bit individual in the t-th generation, where k is the population size. At $t = 0$, all probabilities in $\mathbf{q}_j^t$ are set to $\frac{1}{\sqrt{2}}$ which represents a linear combination of all possible states with the same probability ($|\alpha|^2 = |\beta|^2 = \frac{1}{2}$).

$P(t)$ is the observation population, and $P(t) = \{\mathbf{x}_1^t, \mathbf{x}_2^t, \dots, \mathbf{x}_k^t\}$. $\mathbf{x}_j^t$ is the binary solution of $\mathbf{q}_j^t$, and it is generated by selecting either 0 or 1 for each bit using the probabilities $|\alpha|^2$ and $|\beta|^2$ of each bit in $\mathbf{q}_j^t$. $\mathbf{x}_j^t$ is a binary string of length $m =| C |$, and it could be used to evaluate the profit $f(\mathbf{x})$ in Eq. (11). Since the constraint in Eq. (11) may not be satisfied for $\mathbf{x}_j^t$, a repair step is followed to randomly select n_{max} of the bits in $\mathbf{x}_j^t$ whose value is 1 and set others to 0. Then each repaired $\mathbf{x}_j^t$ in $P(t)$ is evaluated and the best solutions among $B(t-1)$ and $P(t)$ are stored into $B(t)$, where $B(t) = \{\mathbf{b}_1^t, \mathbf{b}_2^t, \dots, \mathbf{b}_k^t\}$ is the best solution population.

In QEA, traditional crossover and mutation operators do not exist. Instead, the population $Q(t)$ is updated by the rotation gate, as:

$$\begin{bmatrix} \alpha' \\ \beta' \end{bmatrix} = \begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) \\ \sin(\Delta\theta) & \cos(\Delta\theta) \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \tag{13}$$

where, $[\alpha, \beta]^T$ and $[\alpha', \beta']^T$ are the Q-bit before and after the update step respectively, and $\Delta\theta$ is a rotation angle of the Q-bit toward either 0 or 1 state depending on its sign.

For each individual $\mathbf{q}_j^t$ in $Q(t)$, $\mathbf{p}_j^t$ and $\mathbf{b}_j^{t-1}$ are respectively its observation and its best solution found till now. Suppose p_i and b_i are the i-th binary bit of $\mathbf{p}_j^t$ and $\mathbf{b}_j^{t-1}$ respectively, if $f(\mathbf{b}_j^{t-1}) > f(\mathbf{p}_j^t)$ and $p_i \neq b_i$, we should update the i-th Q-bit of $\mathbf{q}_j^t$ using Eq. (13) to rotate $\Delta\theta$ degrees toward b_i in order to increase the state b_i's observing probability. In this paper $| \Delta\theta |$ is set to 0.01π, and the sign of $\Delta\theta$ depends on $[\alpha, \beta]^T$'s quadrant.

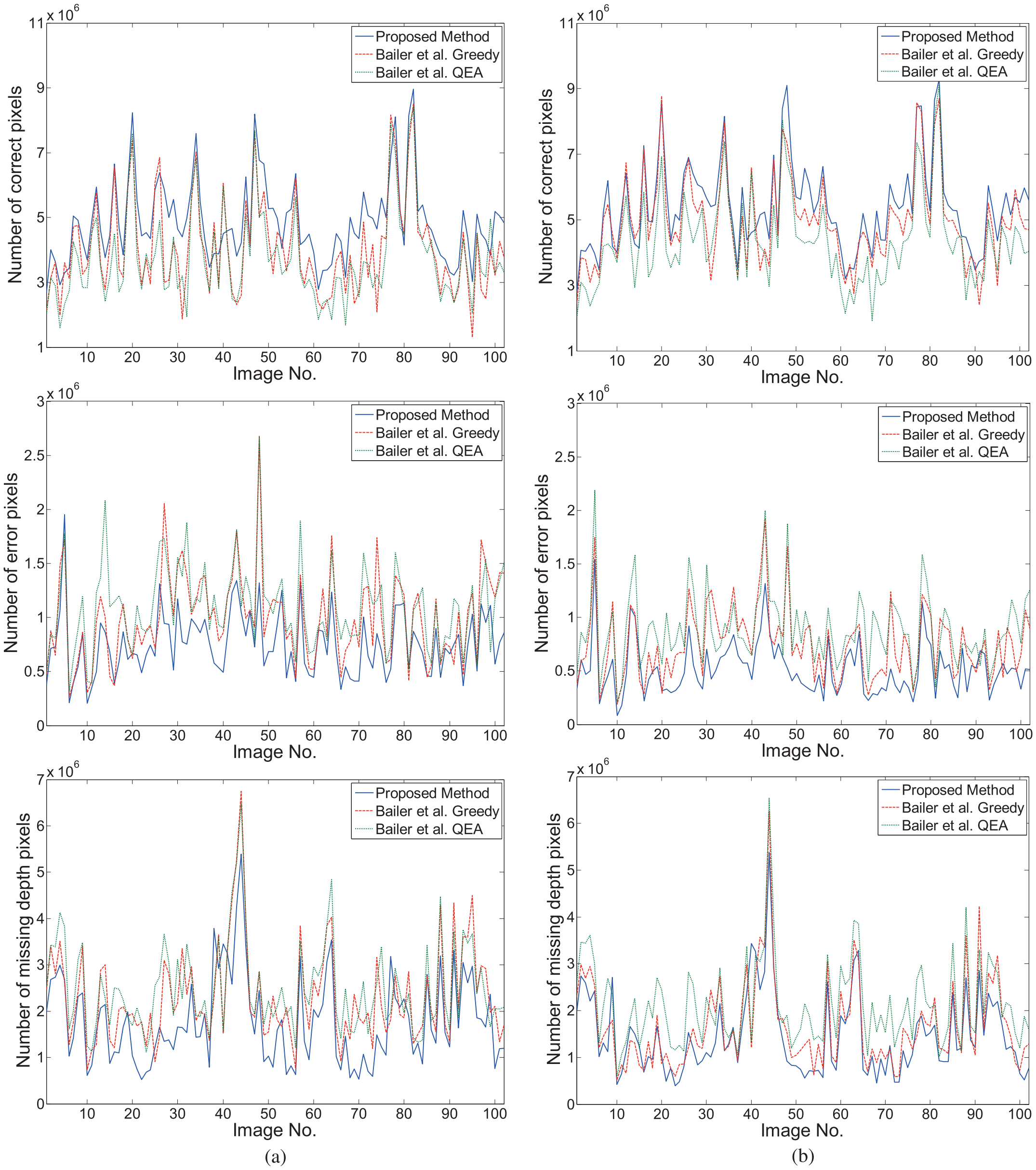
At each generation in QEA, the global best solution among $B(t)$ is stored into $\mathbf{b_g}$. When the migration condition is satisfied (usually at every certain generations), a migration step is implemented by replacing all the solutions in $B(t)$ by $\mathbf{b_g}$. Since each individual is evolved independently in QEA, the migration step plays an important role in propagating information in the population.

The QEA is running in the **while** loop in Algorithm 1 until the termination condition (usually at certain generations) is satisfied. Once the loop is finished, the best neighboring image set for R is generated from $\mathbf{b_g}$ as $\mathbf{N} = \{I_i \mid I_i \in \mathbf{C}, b_g^i = 1\}_{i=1,\dots,m}$, where $\mathbf{b_g} = (b_g^1, b_g^2, \dots, b_g^m)$.

IV. EXPERIMENTAL RESULTS

A. Experiments Description

As noted in Section I, the neighboring image selection is a relatively easy task for regular arrayed cameras like the EPFL data sets [31] or images captured in a controlled environment like the Middlebury data sets [32], but needs to be carefully designed for large-scale scenes in which unordered images are captured at various locations and scales which is the focus of this paper. In order to quantitatively evaluate our proposed neighboring image selection method, we set up a new benchmark data set which is captured in the campus of Tsinghua University. In this data set, a Riegl-LMS-Z420i laser scanner (LIDAR) is used to scan the scene. The LIDAR's



(a)

(b)

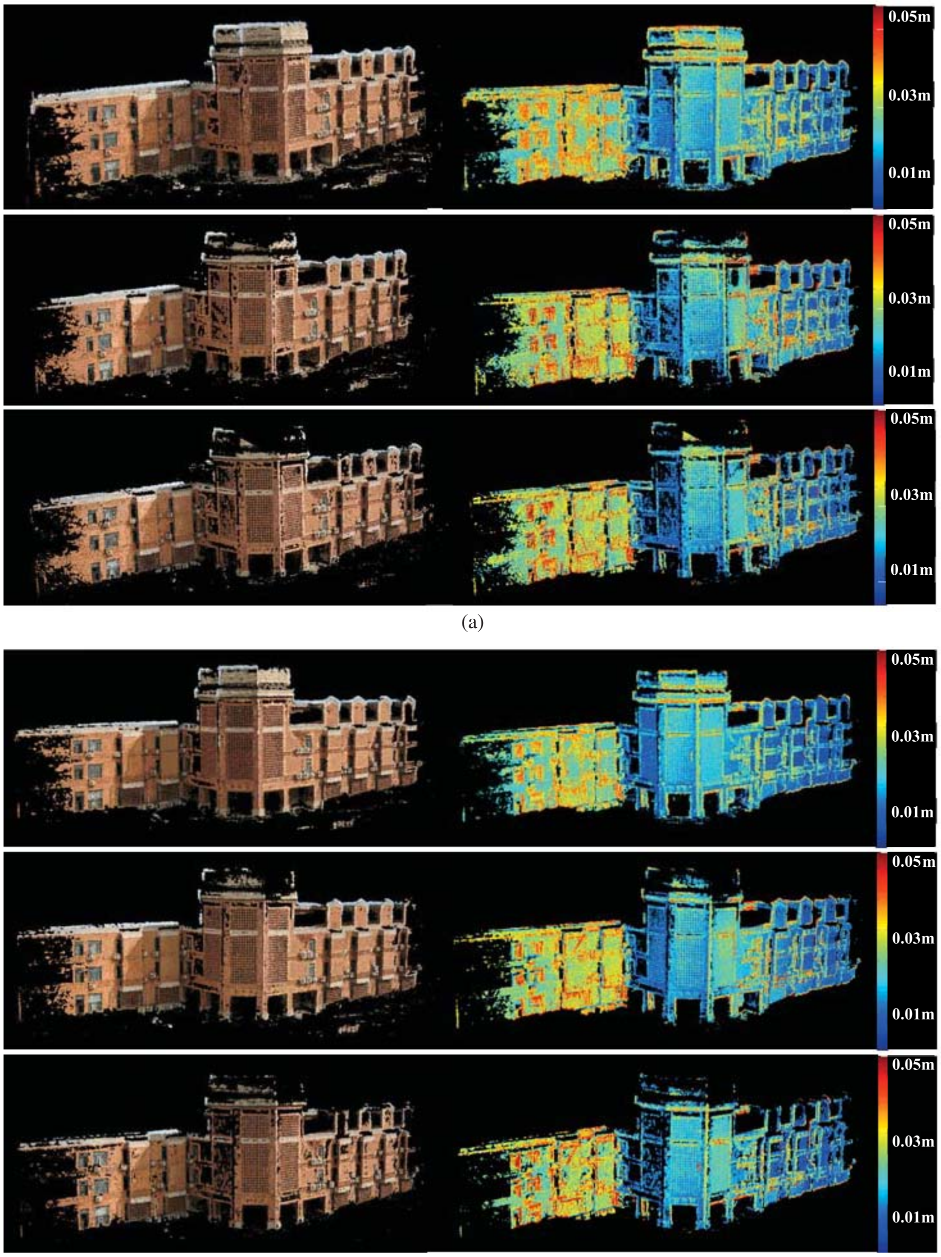
Fig. 8. Number of correct, error and missing depth pixels at each image by the three method. In both (a) and (b), from top to bottom: number of correct, error, and missing pixels at each image respectively. (102 images in total)

accuracy is $10mm@50m$, and its angular stepwidth is 0.0057 degree. The captured LIDAR data is converted to a single high-resolution triangle mesh using Poisson surface reconstruction algorithm [33] which acts as the ground truth 3D model. Together with the LIDAR data 102 images are captured with a Canon DLSR camera with a resolution of 4368×2912 pixels. Finally, the internal and external camera parameters of each image, as well as the coordinates transformation between the LIDAR data and the images, are calibrated with scene control points using the same method as [34]. This data set could be downloaded from our website,[1] and some samples of the data are shown in Fig. 5. As shown in Fig. 5b, the camera locations and their distances from the building have a large range variation across the scene.

We compared our proposed method with the method proposed by Bailer et al. [21]. The selection method in [21] uses a greedy algorithm and grows $\mathbf{N}$ by iteratively adding to $\mathbf{N}$ the view with highest score $g_R(I)$ ($g_R(I)$ is defined in Eq. (2)) as discussed in Section II. Since the QEA could be used to optimize $\sum_{I \in \mathbf{N}} g_R(I)$ directly, we also evaluate the results generated by QEA based optimization on $\sum_{I \in \mathbf{N}} g_R(I)$. We denote the original greedy method in [21] by "Bailer et al. Greedy" and the method using QEA on $\sum_{I \in \mathbf{N}} g_R(I)$ by 'Bailer et al. QEA'.

To measure the quality of the neighboring image set selected by different methods, we compute the depth-map at each reference image with its neighboring images and quantitatively compare it with the ground truth depth-map which is generated by back projecting the ground truth 3D model to each image. Here, we use the depth-map creation method in [21]

[1] http://vision.ia.ac.cn/data/



(a)

(b)

Fig. 9. 3D reconstruction results and 3D error models using three methods with $n_{max} = 3$ and 6 respectively. In both (a) and (b), from top to bottom: 3D reconstruction results (colorized point cloud rendering) and 3D error models using the proposed method, Bailer et al. Greedy, and Bailer et al. QEA, respectively.

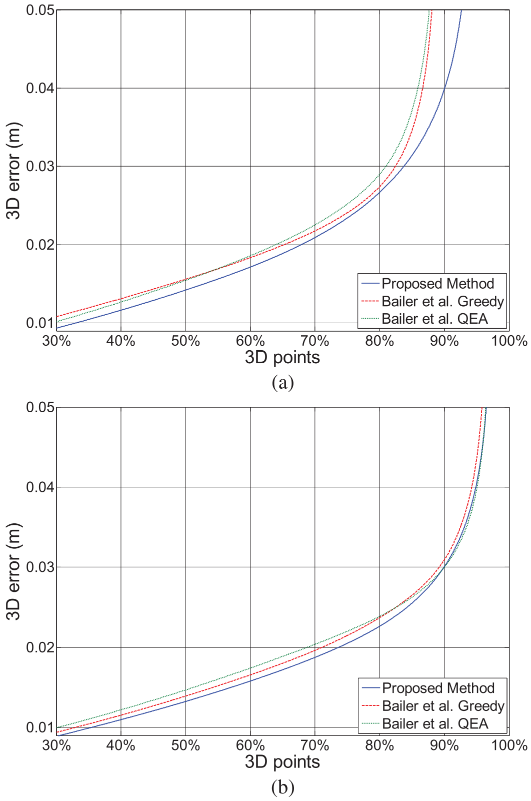
to generate the depth-maps. This method can generate high quality depth maps based on path propagation [35] between nearby pixels which is very similar to our previous work [22], and the main difference between [21] and [22] is that the method in [21] uses multiple neighboring images for depth-map computation and that in [22] uses only one neighboring image for stereo computation.

B. Parameter Settings

Our proposed method has nine parameters, and we have already discussed their value settings in Section II and III. Table I is a summary. All the experiments are implemented on a Intel 2.8GHz Quad Core CPU with 16G RAM.

Note that the parallel nature of QEA makes it well suited for parallel computing because each individual is evolved independently and information is exchanged only at the migration step (every 100 generations). In this paper, the population size of QEA is $k = 4$, thus the four individuals in our proposed method and the Bailer et al. QEA are evolved in parallel on the Quad Core CPU.

In all the experiments, the three evaluated methods use the same value of n_{max}. According to [29], increasing n_{max}



(a)

(b)

Fig. 10. 3D error with respect to the number of 3D points below the error. (a) $n_{max} = 3$. (a) $n_{max} = 6$.

can decrease the depth uncertainty but will increase the depth-map computation burden, thus in the experiments we test two different values of n_{max} (3 and 6) as shown in Table I.

C. Results

In order to quantitatively evaluate the depth-maps computed using different neighboring image selection methods, for each pixel in the image we denote the computed depth by d, and the ground truth depth by d_{gt}, then the relative depth error between the computed depth and the ground truth is defined as:

$$e = \frac{\|d - d_{gt}\|}{d_{gt}} \tag{14}$$

If the depth error e is below a threshold τ_e , the depth d is considered as correct. In this paper we set the threshold $\tau_e = 0.01$.

Fig. 6 is the depth-map of the 51st image computed using three different neighboring image selection methods. Fig. 6a shows the 51st image in the data set and its ground truth depth-map, and Fig. 6b and 6c shows the depth-maps and error-maps generated by different methods with $n_{max} = 3$ and 6 respectively. The blue pixel in the error-map represents the pixel whose depth is unavailable, called a *missing depth* pixel, the red pixel represents an error e larger than τ_e which is considered as an *error* pixel, and the gray pixel represents a *correct* pixel whose depth error $e \leq \tau_e$ and its accuracy is encoded in gray $255 \sim 0$ (brighter, more accurate). Table II compares the numbers of correct, error and missing depth pixels in Fig. 6 by the three methods.

The results show that our proposed method could generate more complete and accurate depth-maps compared with Bailer et al. Greedy and Bailer et al. QEA. When $n_{max} = 3$, the three images selected by Bailer et al. Greedy are all located at the right side of R as shown in Fig. 7b $I_1 \sim I_3$, which results in an incomplete depth-map (such as the top of the building in Fig. 6b). When $n_{max} = 6$, we found that Bailer et al. QEA works worse than Bailer et al. Greedy although the former one uses a global optimization method, which indicates that the view score $g_R(I)$ defined in Bailer et al. Greedy is only designed for local optimization and a global optimal solution on $\sum_{I \in \mathbf{N}} g_R(I)$ generated by Bailer et al. QEA does not guarantee a good solution.

Fig. 7 shows the 51st image and its neighboring images generated by the three methods with $n_{max} = 6$ in space. As shown in Fig. 7a, the six neighboring images $I_1 \sim I_6$ selected by our proposed method all have similar scales with the reference image R, $\{I_1, I_2\}$, $\{I_3, I_4\}$ and $\{I_5, I_6\}$ mainly cover the left, middle and right part of R respectively. Comparatively speaking, some neighboring images selected by Bailer et al. Greedy and Bailer et al. QEA are not appropriate, such as I_3 in Fig. 7b and I_3 and I_6 in Fig. 7c which only cover a small portion of R at a very different scale.

Besides a single image, we respectively plot the number of correct, error and missing depth pixels at each image by the three methods in Fig. 8. The average percentages of the correct/error/missing pixels as opposed to the ground truth pixel counts across 102 images are shown in Table III. The results show that our proposed method could generate more correct pixels but less error and missing depth pixels in most of the images compared with Bailer et al. Greedy and Bailer et al QEA. Once again Bailer et al. Greedy outperforms Bailer et al. QEA.

In order to evaluate the quality of final 3D models generated by different methods, we use the depth-map refinement and merging algorithm in [22] to merge all the depth-maps. The algorithm in [22] uses a depth-map refinement process to enforce the depth consistency over neighboring views and then merges all the refined depth-maps by removing redundancies, which results in a dense and uniformly spaced 3D point cloud. Here, we use the distance between a 3D point and its nearest ground-truth triangular mesh as an error measurement for each 3D point. Since some reconstructed 3D points may not have corresponding ground-truth which could result in very large 3D errors, we remove all those 3D points whose error exceed 0.05 meters. The 3D reconstruction results and 3D error models with the three methods are shown in Fig. 9. The results show that our proposed method could generate a more complete 3D model than both Bailer et al. Greedy and Bailer et al QEA, such as the top of the building in Fig. 9. Fig. 10 shows the 3D error graph with respect to the number of 3D points. It shows that our proposed method could get more accurate 3D reconstructions than the other two methods, but the improvement is not as significant as that on a single depth-map because lots of errors are later removed by the refinement step in [22].

Finally, we evaluate the computation speed of the proposed method. Thanks to the parallel structure of QEA, the proposed method in average takes 6.8 and 9.2 seconds to select neighboring images for each reference image with $n_{max} = 3$ and 6 respectively. Since the depth-map computation step averagely takes 278 seconds ($n_{max} = 3$) and 470 seconds ($n_{max} = 6$) to process an image with a resolution of 4368×2912 pixels in the data set, the runtime for image selection is almost negligible in the whole MVS pipeline.

V. Conclusion

In the depth-map merging based 3D modeling, in addition to generate high quality depth maps at each image, how to to select suitable neighboring images for each image is also an important issue to which unfortunately little attention has been paid till now in the literature. This work is focused on the optimal neighboring image selection problem. In this paper we formulate the neighboring image selection as a combinatorial optimization problem and use the quantum-inspired evolutionary algorithm to seek its optimal solution. We also create a publicly available ground truth data set in which unordered images are captured at various locations and scales. Experimental results on this ground truth data set show that our proposed algorithm could significantly improve the quality of the depth-maps as well as the final 3D reconstruction results with high computational efficiency.

References

[1] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2006, pp. 519–528.

[2] S. M. Seitz and C. R. Dyer, "Photorealistic scene reconstruction by voxel coloring," *Int. J. Comput. Vis.*, vol. 35, no. 2, pp. 151–173, Nov. 1999.

[3] G. Vogiatzis, C. Hernandez, P. H. Torr, and R. Cipolla, "Multiview stereo via volumetric graph-cuts and occlusion robust photo-consistency," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2241–2246, Dec. 2007.

[4] S. N. Sinha, P. Mordohai, and M. Pollefeys, "Multi-view stereo via graph cuts on the dual of an adaptive tetrahedral mesh," in *Proc. IEEE ICCV*, Jun. 2006, pp. 519–528.

[5] O. Faugeras and R. Keriven, "Variational principles, surface evolution, PDE's, level set methods, and the stereo problem," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 336–344, Mar. 1998.

[6] C. Hernandez and F. Schmitt, "Silhouette and stereo fusion for 3D object modeling," *Comput. Vis. Image Understand.*, vol. 96, no. 3, pp. 367–392, Dec. 2004.

[7] V. H. Hiep, R. Keriven, P. Labatut, and J.-P. Pons, "Towards high-resolution large-scale multi-view stereo," in *Proc. IEEE Conf. CVPR*, Jun. 2009, pp. 1430–1437.

[8] D. Cremers and K. Kolev, "Multiview stereo and silhouette consistency via convex functionals over convex domains," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 6, pp. 1161–1174, Jun. 2011.

[9] M. Lhuillier and L. Quan, "A quasi-dense approach to surface reconstruction from uncalibrated images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 418–433, Mar. 2005.

[10] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz, "Multi-view stereo for community photo collections," in *Proc. IEEE ICCV*, Oct. 2007, pp. 1–8.

[11] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 8, pp. 1362–1376, Aug. 2010.

[12] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski, "Towards internet-scale multi-view stereo," in *Proc. IEEE Conf. CVPR*, Jun. 2010, pp. 1434–1441.

[13] T.-P. Wu, S.-K. Yeung, J. Jia, and C.-K. Tang, "Quasi-dense 3D reconstruction using tensor-based multiview stereo," in *Proc. IEEE Conf. CVPR*, Jun. 2010, pp. 1482–1489.

[14] M. Goesele, B. Curless, and S. M. Seitz, "Multi-view stereo revisited," in *Proc. IEEE CVPR*, Jun. 2006, pp. 2402–2409.

[15] P. Merrell, A. Akbarzadeh, L. Wang, P. Mordohai, and J.-M. Frahm, "Real-time visibility-based fusion of depth maps," in *Proc. IEEE 11th ICCV*, Oct. 2007, pp. 1–8.

[16] C. Zach, T. Pock, and H. Bischof, "A globally optimal algorithm for robust TV-L^1 range image integration," in *Proc. IEEE 11th ICCV*, Oct. 2007, pp. 1–8.

[17] D. Bradley, T. Boubekeur, and W. Heidrich, "Accurate multi-view reconstruction using robust binocular stereo and surface meshing," in *Proc. IEEE Comput. Soc. Conf. CVPR*, Jun. 2008, pp. 1–8.

[18] N. D. Campbell, G. Vogiatzis, C. Hernandez, and R. Cipolla, "Using multiple hypotheses to improve depth-maps for multi-view stereo," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 1–14.

[19] J. Li, E. Li, Y. Chen, L. Xu, and Y. Zhang, "Bundled depth-map merging for multi-view stereo," in *Proc. IEEE Conf. CVPR*, Jun. 2010, pp. 2769–2776.

[20] E. Tola, C. Strecha, and P. Fua, "Efficient large-scale multi-view stereo for ultra high-resolution image sets," *Mach. Vis. Appl.*, vol. 23, no. 5, pp. 903–920, 2012.

[21] C. Bailer, M. Finckh, and H. P. Lensch, "Scale robust multi view stereo," in *Proc. ECCV*, Oct. 2012, pp. 398–411.

[22] S. Shen, "Accurate multiple view 3d reconstruction using patch-based stereo for large-scale scenes," *IEEE Trans. Image Process.*, vol. 22, no. 5, pp. 1901–1914, May 2013.

[23] N. Snavely, S. M. Seitz, and R. Szeliski, "Modeling the world from internet photo collections," *Int. J. Comput. Vis.*, vol. 80, no. 2, pp. 189–210, Nov. 2008.

[24] D. Crandall, A. Owens, N. Snavely, and D. Huttenlocher, "Discrete-continuous optimization for large-scale structure from motion," in *Proc. IEEE Conf. CVPR*, 2011, pp. 3001–3008.

[25] D. Gallup, J.-M. Frahm, P. Mordohai, Q. Yang, and M. Pollefeys, "Real-time plane-sweeping stereo with multiple sweeping directions," in *Proc. IEEE Conf. CVPR*, Jun. 2007, pp. 1–8.

[26] M. Pollefeys, D. Nister, J.-M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, *et al.*, "Detailed real-time urban 3D reconstruction from video," *Int. J. Comput. Vis.*, vol. 72, no. 2, pp. 143–167, 2008.

[27] D. Gallup, J.-M. Frahm, and M. Pollefeys, "Piecewise planar and non-planar stereo for urban scene reconstruction," in *Proc. IEEE Conf. CVPR*, Jun. 2010, pp. 1418–1425.

[28] D. Gallup, J.-M. Frahm, P. Mordohai, and M. Pollefeys, "Variable baseline/resolution stereo," in *Proc. IEEE CVPR*, Jun. 2008, pp. 1–8.

[29] M. Rumpler, A. Irschara, and H. Bischof, "Multi-view stereo: Redundancy benefits for 3D reconstruction," in *Proc. 35th Workshop Austrian Assoc. Pattern Recognit.*, May 2011, pp. 1–8.

[30] K.-H. Han and J.-H. Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," *IEEE Trans. Evol. Comput.*, vol. 6, no. 6, pp. 580–593, Dec. 2002.

[31] [Online]. Available: http://cvlabwww.epfl.ch/data/multiview/

[32] [Online]. Available: http://vision.middlebury.edu/mview/

[33] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proc. 4th Eurograph. Symp. Geometry Process.*, 2006, pp. 61–70.

[34] C. Strecha, W. von Hansen, L. V. Gool, P. Fua, and U. Thoennessen, "On benchmarking camera calibration and multi-view stereo for high resolution imagery," in *Proc. IEEE Conf. CVPR*, Jun. 2008, pp. 1–8.

[35] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," *ACM Trans. Graph.*, vol. 28, no. 3, PP. 1–24, Aug. 2009.

Shuhan Shen received the B.S. and M.S. degrees from Southwest Jiao Tong University in 2003 and 2006, respectively, and the Ph.D. degree from Shanghai Jiao Tong University in 2010. Since 2010, he has been with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, where he is currently an Assistant Professor. His research interests are image based 3D modeling.

Zhanyi Hu received the B.S. degree in automation from the North China University of Technology, Beijing, China, in 1985, and the Ph.D. degree in computer vision from the University of Liege, Belgium, in 1993. Since 1993, he has been with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, where he is currently a Professor. His research interests are in robot vision, which include camera calibration and 3D reconstruction, vision guided robot navigation. He was the Local Chair of ICCV'2005, an Area Chair of ACCV'2009, and the PC Chair of ACCV'2012.