# A Graph Matching Based Key Point Correspondence Method for Lunar Surface Images 

Yuren Zhang, Xu Yang, Hong Qiao, Zhiyong Liu, Chuankai Liu, and Baofeng Wang


#### Abstract

Key point correspondence plays an important role in lunar surface image processing. Since lunar surface images often contain obvious illumination changes, noisy points and repetitive patterns, traditional appearance based algorithms may fail when local appearance descriptors become less distinctive. In this paper, we introduce a graph matching based algorithm to tackle this problem. First, by incorporating structural information, key point sets in lunar surface images are represented by graphs. Then key point correspondence is formulated as a specific graph matching problem which aims to find a specified number of best assignments, and effectively approximately solved. Finally, an outlier assignment elimination method is proposed based on the affine invariance assumption. Simulations on both benchmark datasets and lunar surface images witness the effectiveness of the proposed method.


## I. Introduction

KEY point correspondence is often an indispensable part in lunar surface image processing and also finds applications in lunar rover operations. For instance, it plays an important role in self-localization and navigation tasks in China's first lunar rover Yutu [24], [15].

In literature, various types of algorithms have been proposed to tackle the point correspondence problem. One popular group of algorithms are based on appearance matching. Specifically, local features such as SIFT [19], bag of words [14], and shape context [2] are first extracted from two images, and then the correspondence can be found by minimizing the appearance difference between features. The problem is equivalent to a linear programming problem, for which efficient algorithms exist. Besides, there exist other algorithms for correspondence between static scenes under rigid motion [1], [10]. These methods exploit powerful constraints to reduce the search space and disambiguate the correspondence problem [9].

However, there are few algorithms dedicated for lunar surface image correspondence, and the above algorithms often fail due to some special characteristics of lunar surface images. Take Yutu rover for instance, one of its cameras could shoot forward and backward by pan-tilt rotation. As the rover moves on, images acquired may contain overlapped lunar surface area, in which extracted key points should be

[^0]matched. However, such a correspondence problem may be very challenging due to the following reasons.

- Scale and rotation transformation, even reflected view Since the lunar rover gets images from different positions and views, the correspondence between these images suffers from significant geometric transformations;
- Repetitive patterns Many lunar surface images contains similar small rocks and pits, which results in the repetitive patterns. These repetitive patterns further generate similar appearance descriptor, making appearance matching fail in lunar surface correspondence problem;
- Outlier The overlapped areas in two lunar surface images are usually parts in either images, outliers are inevitable, and there are usually outliers in both images. This makes the correspondence problem more difficult.
One main drawback of the above methods is that they focus on constructing discriminative appearance descriptors, while ignoring other useful information such as structural information. When utilizing the structural information, the correspondence problem can be well defined by graph matching, by representing key points with graph vertices, and representing relations between points with graph edges. The incorporation of constraints on geometric compatibility and spatial coherence between features could alleviate the dependence on discriminative ability of every feature.

In this paper, we introduce a graph matching based algorithm to tackle the key point correspondence problem in lunar surface images. First, key point sets are represented by graphs. In particular, the structure of the key points is described with edge features like distance and orientation. Then key point correspondence is formulated as a specific graph matching problem which aims to find a specified number of best assignments, and effectively approximately solved. Finally, an outlier assignment elimination method is proposed based on the affine invariance assumption.
From an application point of view, our method has some resemblance to [24] which for the first time utilized a graph matching based scheme to deal with the lunar surface image correspondence. The differences between the two methods are two-fold. First on algorithm, [24] adopts a probabilistic graph matching algorithm based on spectral decomposition to match all the points, and then selects best assignments by ranking assignment probability. When finding specified number of best assignments, such a two-step scheme may be inappropriate because even both steps are optimally solved, the final matching solution may not be the optimal [25]. By contrast, our method directly targets at the specified number
of best assignment, avoiding the inequivalence in the twostep scheme. Second on performance, though [25] could well tackle the geometric transformation problem, it often suffers from outliers which is common in lunar surface images. By contrast, with the help of an outlier assignment elimination step, our method is robust to outliers.

The paper proceeds as follows. Section II describes the proposed method, including the problem formulation, optimization algorithm and outlier assignment elimination method. The experimental results are given in section III. Finally section IV concludes the paper.

## II. The proposed method

In this section we first introduce the problem formulation and the objective function, then show how to optimize the problem, and finally give a outlier assignment elimination method.

## A. Problem formulation

Given a lunar surface image and key point set extracted from it. The key point set can be represented as a labeled weighted graph $G=\{V, E, l, w\}$, where $V=\{1,2, \ldots, M\}$ is the vertex set, and $E \subseteq V \times V$ is the edge set. The labeling function $l: V \rightarrow \mathbb{R}^{M \times d_{l}}$ assigns a local appearance descriptor of size $d_{l}$ to each vertex, and weighting function $w: E \rightarrow \mathbb{R}^{\|E\|_{0} \times d_{w}}$ assigns a edge feature of size $d_{w}$ to each edge. Each vertex $i \in V$ represents one key point $v_{i}$ with appearance descriptor $\mathbf{l}_{i}$ around the point as a label. Each edge $\{i, j\} \in E$ represents the connection between two vertices $i$ and $j$ with a weight vector $\mathbf{w}_{i j}$ consisting of structural descriptors. Below we introduce the formulation of the graph matching problem.

Based on the above definitions and notations, we next show how to formulate the graph matching objective function given two graphs $G^{1}=\left\{V^{1}, E^{1}, l^{1}, w^{1}\right\}$ of size $M$ and $G^{2}=\left\{V^{2}, E^{2}, l^{2}, w^{2}\right\}$ of size $N$ assuming $M \leq N$. Since there exist outliers in both lunar surface images, the goal is find specified number of best assignments between vertices in $G^{1}$ and $G^{2}$, or say $L$ best assignment where $L \leq M \leq N$ [25].

The $L$ best assignment problem can be formulated as follows:

$$
\begin{align*}
& \mathbf{x}=\arg \max \mathbf{x}^{T} \mathbf{A} \mathbf{x} \\
& \text { s.t. } \\
& \left(\mathbf{I}_{N} \otimes \mathbf{1}_{M}^{T}\right) \mathbf{x} \leq \mathbf{1}_{N}, \\
& \left(\mathbf{1}_{N}^{T} \otimes \mathbf{I}_{M}\right) \mathbf{x} \leq \mathbf{1}_{M}, \\
& \mathbf{1}_{M N}^{T} \mathbf{x}=L, \mathbf{x} \in\{0,1\}^{M N \times 1} . \tag{1}
\end{align*}
$$

where $\mathbf{1}_{N}$ is vector of 1 s of length $N$. The operator $\otimes$ denotes Kronecker product between two matrix. The symbol $<=$ denotes element-wise $<=$ between two vectors. $\mathbf{x} \in$ $\{0,1\}^{M N \times 1}$ is row-wise vectorization of the assignment matrix $\mathbf{X} \in\{0,1\}^{M \times N}$ with $\mathbf{X}_{i a}=1$ if $a$ is assigned to $i$. The constraints in (1) guarantees that $\mathbf{x}$ represents a one-to-one mapping with $L$ assignments.
$\mathbf{A} \in \mathbb{R}^{M N \times M N}$ is denoted as affinity matrix [24]. It is constructed as follows:

$$
\begin{aligned}
\mathbf{A}_{i j a b} & =\mathbf{A}_{[[i-1] N+a][(j-1) N+b]} \\
& = \begin{cases}\alpha \operatorname{Sim}_{l}\left(\mathbf{l}_{i}^{1}, \mathbf{l}_{a}^{2}\right), & \text { if } i=j, a=b \\
(1-\alpha) \operatorname{Sim}_{w}\left(\mathbf{w}_{i j}^{1}, \mathbf{w}_{a b}^{2}\right), & \text { if } i \neq j, a \neq b \\
0, & \text { otherwise }\end{cases}
\end{aligned}
$$

where $\alpha$ is used to balance the appearance similarity and pairwise consistency, $\operatorname{Sim}_{w}$ and $\operatorname{Sim}_{l}$ calculates the similarity between weights and labels respectively.

## B. Optimization method

The optimization of (1) is an NP-hard problem with factorial complexity, and thus approximate methods are necessary. In recent years, continuous methods are among the most popular approximate methods for graph matching [4], which typically involve relaxing the discrete domain to the continuous domain and obtain a continuous solution c. To finally get a discrete solution $\mathbf{x}$, a winner-take-all projection is usually adopted [11], [5], [6], [25], which, however, may introduces significant additional error [17], [25]. Besides, most current continuous methods [26], [17], [27], [12] cannot be applied to $L$ best assignment problems.

In this paper, a continuous method [25] is introduced to optimize (1), which directly targets at the $L$ best assignment problem, and project the continuous solution to discrete domain in a graduated manner. Specifically, this method first relaxes the constraints in (1) to its convex hull $\mathcal{C}$ as follows:

$$
\begin{array}{r}
\mathcal{C}=\left\{\mathbf{x} \mid\left(\mathbf{I}_{N} \otimes \mathbf{1}_{M}^{T}\right) \mathbf{x} \leq \mathbf{1}_{N},\left(\mathbf{1}_{N}^{T} \otimes \mathbf{I}_{M}\right) \mathbf{x} \leq \mathbf{1}_{M}\right. \\
\left.\mathbf{1}_{M \times N}^{T} \mathbf{x}=L, \mathbf{x} \geq 0\right\}
\end{array}
$$

Then the relaxed optimization problem is

$$
\mathbf{x}=\arg \max \mathbf{x}^{T} \mathbf{A} \mathbf{x}
$$

s.t.

$$
\mathbf{x} \in \mathcal{C}
$$

Then the method utilizes a recently proposed graduated nonconvexity and concavity procedure (GNCCP) [16] to get the continuous solution and gradually project it to the discrete domain as follows:

$$
\begin{align*}
& \max F_{\zeta}= \begin{cases}(1+\zeta) \mathbf{x}^{T} \mathbf{A} \mathbf{x}+\zeta \mathbf{x}^{T} \mathbf{x}, & \text { if }-1 \leq \zeta \leq 0 \\
(1-\zeta) \mathbf{x}^{T} \mathbf{A} \mathbf{x}+\zeta \mathbf{x}^{T} \mathbf{x}, & \text { if } 0 \leq \zeta \leq 1\end{cases} \\
& \text { s.t. } \mathbf{x} \in \mathcal{C} \tag{2}
\end{align*}
$$

As $\zeta$ increases from -1 to 1 , the objective function implicitly realizes the transition from the concave relaxation of the original objective $\mathbf{x}^{T} \mathbf{A x}$ to its convex relaxation. Accordingly, the optimization problem changes from a concave maximization problem ${ }^{1}$ to a convex maximization problem ${ }^{2}$. When finally reaching the convex relaxation, the solution is guaranteed to be in the discrete domain $\mathcal{D}$ [16]. For each specific $\zeta$, (2) is maximized by the Frank-Wolfe algorithm [7]. The overall optimization method is summarized in Algorithm 1.

```
Algorithm 1 Optimization algorithm for \(L\) best assignments
Require: Two graphs \(G^{1}\) and \(G^{2}\);
Ensure: The assignment vector \(\mathbf{x}\);
    Initialize \(x=\mathbf{1}_{M N} \frac{L}{M N}, \zeta=-1\)
    Construct the affinity matrix \(\mathbf{A}\) by (2)
    repeat
        \(\mathbf{x}_{\text {old }}=\mathbf{x}\)
        repeat
            \(\mathbf{y}=\arg \max \nabla F_{\zeta}\left(\mathbf{x}_{\text {old }}\right)^{T} \mathbf{y}\), s.t. \(\mathbf{y} \in \mathcal{C}\)
            \(\alpha=\arg \max F_{\zeta}\left(\mathbf{x}_{\text {old }}+\alpha\left(\mathbf{y}-\mathbf{x}_{\text {old }}\right)\right), 0 \leq \alpha \leq 1\)
            \(\mathbf{x}_{\text {new }}=\mathbf{x}_{\text {old }}+\alpha\left(\mathbf{y}-\mathbf{x}_{\text {old }}\right)\)
            \(\mathbf{x}_{\text {old }}=\mathbf{x}_{\text {new }}\)
        until \(\nabla F_{\zeta}\left(\mathbf{x}_{n e w}\right)^{T}\left(\mathbf{y}-\mathbf{x}_{\text {new }}\right)<\epsilon\), where \(\epsilon\) is a small
        positive constant
        \(\mathbf{x}=\mathbf{x}_{\text {new }}\)
        \(\zeta=\zeta+d \zeta\)
    until \(\zeta>1\) or \(\mathrm{x} \in \mathcal{D}\)
    return x
```

In Algorithm 1, the gradient $\nabla F_{\zeta}(\mathbf{x})$ is given as follows:

$$
\nabla F_{\zeta}(\mathbf{x})= \begin{cases}(1+\zeta)\left(\mathbf{A}+\mathbf{A}^{T}\right) \mathbf{x}+2 \zeta \mathbf{x}, & \text { if }-1 \leq \zeta \leq 0 \\ (1-\zeta)\left(\mathbf{A}+\mathbf{A}^{T}\right) \mathbf{x}+2 \zeta \mathbf{x}, & \text { if } 0 \leq \zeta \leq 1\end{cases}
$$

In the algorithm $\mathbf{y}$ is obtained by solving a linear programming problem, for which we utilize a fast approximate approach following [25] with a complexity of $O\left(M^{2} N\right)$. The step size $\alpha$ can be found by inexact line search, e.g. backtracking method [3].

## C. Outlier assignment elimination

Though the above optimization algorithm could find $L$ best assignments, there may still be mismatched points, especially when $L$ is larger than the number of ground truth assignments. Hence we introduce a outlier assignment elimination method based on angular spatial order [18]. Angular spatial order is an affine transformation invariant descriptor which is applicable in lunar surface images, because most lunar surface areas are flats or gentle slopes. Based on [18], we further use Delaunay triangulation to construct the graph and propose a new method to determine convergence. Below before introducing the outlier assignment elimination method, we first show how to adapt angular spatial order to our problem.

## D. Angular spatial order

Two subgraphs of size $L$ found by the above algorithm are denoted by $G^{s 1}$ and $G^{s 2}$, their vertex sets are respectively denoted as $V^{s 1}$ and $V^{s 2}$. For a vertex $i$ in $G^{s 1}$, a starshaped graph $S_{i}$ can be constructed around it. The leaf vertices of $S_{i}$ are adjacent vertices of $i$, denoted by $\operatorname{adj}(i)=$ $\left\{m_{1}, m_{2}, \cdots, m_{k}\right\}$, where $n_{i} \in V^{s 1}, i=1, \cdots, k$ and $k \leq L$. We also construct a star-shape graph $S_{i}^{\prime}$ around $i^{\prime}$ in $G^{s 2}$, i.e. the corresponding vertex of $i$ in $G^{s 1}$, by directly utilizing the corresponding vertices of $\operatorname{adj}(i)$ as the leaf vertices of $S_{i}^{\prime}$, denoted by $a d j^{\prime}(i)=\left\{n_{1}, n_{2}, \ldots, n_{k}\right\}$. Since $S_{i}^{\prime}$ is constructed based on the adjacency relation of


Fig. 1. Star-shaped graph and angular spatial order. (a) Star-shaped graph $S_{1}$, and angular spatial order $O_{1}=\{5,4,3,2\}$ in $G^{s 1}$ (b) Star-shaped graph $S_{1}^{\prime}$, and angular spatial order $O_{1}^{\prime}=\{4,3,5,2\}$ in $G^{s 2}$, assuming vertex 1 in $G^{s 1}$ is assigned to vertex 1 in $G^{s 2}$. Note $O_{1}$ and $O_{1}^{\prime}$ are different due to the outlier 5 .

(a) $G^{s 1}$

(b) $G^{s 2}$

Fig. 2. The angular spatial difference vector $\mathbf{d}=(0,0,2 / 3,0,1 / 2)$, and the average angular spatial difference vector $\overline{\mathbf{d}}=$ $(1 / 4,7 / 18,1 / 6,7 / 12,1 / 6)$.
$i$ in $G^{s 1}$, hence vertices in $a d j^{\prime}(i)$ are not guaranteed to be adjacent with $i^{\prime}$ in $G^{s 2}$. The angular spatial order is right the order of the leaf vertices in clockwise direction [18]. Then we can obtain the angular spatial orders for $S_{i}$ and $S_{i}^{\prime}$, which are respectively denoted by $O_{i}=\left\{n_{p_{1}}, n_{p_{2}}, \ldots, n_{p_{k}}\right\}, O_{i}^{\prime}=$ $\left\{n_{p_{1}^{\prime}}, n_{p_{2}^{\prime}}^{\prime}, \ldots, n_{p_{k}^{\prime}}\right\}$, where $p_{i}, p_{i}^{\prime} \in\{1,2, \ldots, k\}$. For example, in Fig. $1, i=1, \operatorname{adj}(1)=\{2,3,4,5\}, O_{1}=\{5,4,3,2\}$ and $O_{1}^{\prime}=\{4,3,5,2\}$.

If all the points are matched correctly, $O_{i}$ and $O_{i}^{\prime}$ should be the same because angular spatial order is invariant to affine transformation. When there is a mismatch, $O_{i}$ and $O_{i}^{\prime}$ should be different as illustrated in Fig.1. Their difference can be measured by cyclic edit distance of strings [20]. Specifically, to get the cyclic edit distance, first $O_{i}^{\prime} O_{i}^{\prime}=$ $\left\{n_{p_{1}^{\prime}}, n_{p_{2}^{\prime}}, \ldots, n_{p_{k}^{\prime}}, n_{p_{1}^{\prime}}, n_{p_{2}^{\prime}}, \ldots, n_{p_{k}^{\prime}}\right\}$ is obtained by concatenating $O_{i}^{\prime}$ with itself. Then the string edit distance between $O_{i}$ and $O_{i}^{\prime} O_{i}^{\prime}$ is computed by the insertion, deletion, and substitution operation with [20].

Since the cyclic string edit distance between $O_{i}$ and $O_{i}^{\prime}$ is bounded by $k$, i.e. the number of adjacent vertices of $i$, which may be different for other vertices, we normalize it as follows:

$$
\begin{equation*}
d_{i}=\frac{\text { CyclicDistance }\left(O_{i}, O_{i}^{\prime}\right)}{k} . \tag{3}
\end{equation*}
$$

For instance, in Fig. 1 there are outliers due to similar local appearance and $d_{1}=2 / 4=1 / 2$. The complexity of calculating a string edit distance is $O\left(k^{2}\right)$ [22]. To calculate the distance for all vertices the complexity is $O\left(L k^{2}\right)$ where $L$ is the size of the common subgraph.

## E. Outlier assignment elimination method

Based on angular spatial order, we next introduce outlier assignment elimination method. The angular spatial difference vector is denoted by $\mathbf{d}=\left(d_{1}, d_{2}, \ldots, d_{L}\right)^{T}$ where
$d_{i}, i=1, \cdots, L$ is defined in (3). First $\mathbf{d}$ is normalized by

$$
\begin{equation*}
\overline{\mathbf{d}}=\mathbf{D}_{s 1}^{-1} \mathbf{A}_{s 1} \mathbf{d} \tag{4}
\end{equation*}
$$

where $\mathbf{A}_{s 1}$ is the unweighed adjacency matrix of $G^{s 1}$ and $\mathbf{D}_{s 1}$ is the corresponding degree matrix. $\overline{\mathbf{d}}$ can be interpreted as the average angular spatial difference of each vertex's neighbors. We use $\overline{\mathbf{d}}$ rather than $\mathbf{d}$ to estimate the existence of outliers because the previous one better characterizes the matching error. An example of calculating $\mathbf{d}$ and $\overline{\mathbf{d}}$ is illustrated in Fig. 2.

To remove outlier assignments, we conduct the matching algorithm iteratively decreasing $L$ by a step size 1 until $\max [\overline{\mathbf{d}}]<\eta$, where $\eta \in(0,1]$ is a termination parameter. This whole outlier assignment elimination method is summarized in Algorithm 2.

```
Algorithm 2 Outlier assignment elimination method
    repeat
        Calculate \(L\) best assignment with Algorithm 1;
        Calculate the angular spatial order \(O_{i}\) and \(O_{i^{\prime}}^{\prime}\) for each
        pair of assigned points;
        Calculate angular spatial difference:
        \(d_{i}=\) CyclicDistance \(\left(O_{i}, O_{i}^{\prime}\right) /|\operatorname{adj}(i)|\)
        \(\overline{\mathbf{d}}=\mathbf{D}_{s 1}^{-1} \mathbf{A}_{s 1} \mathbf{d}, \mathbf{A}_{s 1}\) is the adjacency matrix of \(G^{s 1}\),
        \(\mathbf{D}_{s 1}\) is the degree matrix;
        \(L \leftarrow L-1\)
    until \(\max [\overline{\mathbf{d}}]<\eta\)
```


## III. Experimental results

Evaluation of the proposed graph matching method is conducted on benchmark datasets in Section 3.1 and Section 3.2 to examine the graph matching method together with the outlier elimination method. The performance of the proposed graph matching based method is evaluated on the lunar surface images in Section 3.3.

In the experiments, the graphs are constructed with Delaunay triangulation. The vertex similarity $\operatorname{Sim}_{l}\left(\mathbf{l}_{i}, \mathbf{1}_{a}\right)$ is defined by

$$
\operatorname{Sim}_{l}\left(\mathbf{l}_{i}, \mathbf{l}_{a}\right)=\exp \left(-\frac{\left\|\mathbf{l}_{i}-\mathbf{l}_{a}\right\|_{2}^{2}}{\delta_{l}}\right)
$$

where $\delta_{l}$ is the kernel width.
The most commonly used edge features are normalized distance $d_{i j}$ and orientation $o_{i j}$. Assume that $r_{i j}=p_{j}-p_{i}$ is the offset vector pointing from coordinate $p_{i}$ to coordinate $p_{j}$, and $e_{x}$ is the unit vector along x-axis. $d_{i j}$ and $o_{i j}$ are defined as:

$$
d_{i j}=\frac{\left|r_{i j}\right|}{\max _{i, j}\left|r_{i j}\right|}, o_{i j}=\frac{1}{\pi} \arcsin \frac{r_{i j} \times e_{x}}{\left|r_{i j}\right|}
$$

The edge similarity based on the distance descriptor and orientation descriptor $\operatorname{Sim}_{w}\left(\mathbf{w}_{i j}, \mathbf{w}_{a b}\right)$ is defined by

$$
\begin{equation*}
\operatorname{Sim}_{w}\left(\mathbf{w}_{i j}, \mathbf{w}_{a b}\right)=\exp \left(-\frac{\left\|\mathbf{w}_{i j}-\mathbf{w}_{a b}\right\|_{2}^{2}}{\delta_{w}}\right) \tag{5}
\end{equation*}
$$

where $\delta_{w}$ is the kernel width.

The algorithms included for comparison are spectral matching (SM) [11], graduated assignment (GA) [8], probabilistic spectral graph matching (PGM) [24], and Bipartite (BP) [21].

All the simulations are carried out in Matlab 2014a on 2.5 GHz CPU(two cores) and 8.00 GB RAM.

## A. Evaluation on the CMU house image dataset



Fig. 3. Comparison of the graph matching algorithms on the CMU house dataset. The baseline width is set to be (a) 10 and (b) 30 . The average accuracy and estimated $L$ are shown.

The CMU house image dataset consists of 111 frames of a house. Each of them has been manually labeled with 30 landmarks. Same number of randomly chosen outliers are added to the landmarks of both images to generate $G_{1}$ and $G_{2}$. No vertex feature is used, and the distance and orientation is used as edge feature. The similarity between edges is calculated with (5). The experiments are conducted with three setups, in which the baseline width is set to be 10,30 , and 50 respectively. The number of outliers increases from 0 to 10 with a step size of 1 . For each method, the number of inliers $L$ is estimated through the outlier elimination procedure. The number of nearest neighbors $k$ for calculating angular spatial order is set to be 7, and the threshold $\eta$ is set to be 0.4 . The lowest $L$ is set to be 20 . The accuracy is calculated as:

$$
\text { accuracy }=\frac{\# \text { CorrectMatches }}{L}
$$

The results are shown in Fig. 3. From the perspective of matching accuracy, OUR outperforms other algorithms. PGM performs slightly worse than OUR, and the $L$ estimated from PGM is less than OUR's. For GA and SM, the estimated $L$ approaches to the lower limit of 20, and the accuracy drop faster than OUR and PGM.

From the perspective of estimated L, with a small baseline width the estimated $L$ is close to the ground truth of 30 . With a larger baseline width, the deformation becomes more severe and the matching becomes harder. Accordingly, the estimated $L$ decreases. In all the cases the $L$ estimated by OUR is closest to the ground truth of 30 . The running time


Fig. 4. Running time of graph matching algorithms as a function of graph size using a logarithmic scale for both the x -axis and the y -axis.
of different graph matching algorithms is also compared as shown in Fig. 4. The size of graph varies from 10 to 30. The slope of OUR $\approx 3.1$. The slope of $\mathrm{SM} \approx 3.3$, while GA has a slope $\approx 4$.

## B. Evaluation on the Car and Motorbike dataset

The car and motorbike dataset was created in [13]. The dataset consists of 30 pairs of car images and 20 pairs of motorbike images along with ground truth labels and outliers. For each pair of images, outliers are added to the ground truth landmarks in both images to build two graphs $G_{1}$ and $G_{2}$. Similar to the CMU house image dataset, no vertex feature is used and the distance and orientation is used as edge feature. The matching accuracy is compared with the number of outliers increasing from 0 to 10 with step size of 1 . The number of nearest neighbors $k$ for calculating angular spatial order is set to be 7 , and the threshold $\eta$ is set to be 0.4. Some matching examples are shown in Fig. 6. Compared with the house dataset, point sets in the car and motorbike dataset suffers from more severe deformation and scale change which makes the matching more difficult. In both datasets, OUR outperforms other methods, which suggests the proposed method can perform well in real world images.


Fig. 5. Comparison of the graph matching algorithms on the car and motor bike dataset. (a) shows the result on the car dataset, and (b) shows the result on the motorbike dataset.

## C. Evaluation on the Lunar surface images

Finally, the proposed method is evaluated on real lunar surface images acquired by China's Yutu rover and from NASA's Apollo 15 mission ${ }^{3}$. 20 pairs of images are used, with 5 pairs acquired by Yutu rover and 15 pairs from the Apollo 15 mission. The keypoints are extracted with a contrast invariant and quasi-parameter free detector tree-base morse regions [23] . Each image is labeled with 20 ground truth points. The SIFT [19] descriptor of each keypoint is adopted as the vertex feature since it is robust against scale change, orientation, and local geometrical distortion.
The graph matching algorithms used for the matching include OUR, PGM, GA, SM, and BP. The outlier elimination procedure is used together with each of these algorithms to determine the size of common subgraph. A baseline accuracy is also shown which is derived with local features alone. Besides the 20 ground truth points, 1 to 10 outliers are added. The number of nearest neighbors $k$ for calculating angular spatial order is set to be 7 , and different thresholds are set to show the affection of the threshold on the final result. The results are shown in Fig. 7.


Fig. 7. Matching accuracy and estimated $L$ with respect to the number of outliers. The threshold $\eta=0.25$.

From Fig. 7 we can observe as follows. When the threshold is low (Fig. 7 (a,b)), the standard of outlier elimination is tight. From the perspective of accuracy, since the baseline method does not utilize structural features it performs poorly. When the number of outliers is zero, the matching accuracy is lower than 0.7 , which is a direct evidence of the existence of repetitive patterns in the lunar surface images. The performance of OUR, GA, BP and PGM is similar. OUR sightly outperforms the other methods, while the number of inliers estimated with OUR is closer to the ground truth than others. Some matching results achieved by OUR are shown in Fig. 8. With a higher threshold (Fig. 7 (c,d)) few matches are recognized as outliers. The matching accuracy drops and the estimated number of inliers tend to be larger than the ground truth.

## IV. CONCLUSIONS

The results in this paper show that key point correspondence for lunar surface images can be solved with graph matching based algorithm. A graph matching algorithm is used to solve the $L$ best assignment problem, and an outlier

[^1]

Fig. 6. Comparison of the matching results on and a pair of motorbike image. The ground truth number of inliers 30 . The estimated L and number of correctly found matches (\#correct) are shown in the brackets (\#correct/L).


Fig. 8. Typical matching samples by OUR with $\eta=0.25$. Green lines indicate right assignments, and red lines indicate incorrect assignments. The estimated L and number of correctly found matches (\#correct) are shown in the brackets (\#correct/L).
assignment elimination method is used to obtain reliable assignments. Simulations on both benchmark datasets and lunar surface images confirm the effectiveness of the proposed method.

## REFERENCES

[1] P. N. Belhumeur, "A binocular stereo algorithm for reconstructing sloping, creased, and broken surfaces in the presence of half-occlusion," in Proc. IEEE Int'l Conf. Computer Vision, 1993, pp. 431-438.
[2] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 24, no. 4, pp. 509-522, 2002.
[3] S. Boyd and L. Vandenberghe, Convex optimization. Cambridge university press, 2009.
[4] D. Conte, P. Foggia, C. Sansone, and M. Vento, "Thirty years of graph matching in pattern recognition," Int'l Journal of Pattern Recognition and Artificial Intelligence, vol. 18, no. 3, pp. 265-298, 2004.
[5] T. Cour, P. Srinivasan, and J. B. Shi, "Balanced graph matching," Proc. Advances in Neural Information Processing Systems, vol. 19, pp. 313320, 2007.
[6] A. Egozi, Y. Keller, and H. Guterman, "A probabilistic approach to spectral graph matching," IEEE Trans. Pattern Analysis and Machine Intelligence,, vol. 35, no. 1, pp. 18-27, 2013.
[7] M. Frank and P. Wolfe, "An algorithm for quadratic programming,' Naval research logistics quarterly, vol. 3, no. 1-2, pp. 95-110, 1956.
[8] S. Gold and A. Rangarajan, "A graduated assignment algorithm for graph matching," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 18, no. 4, pp. 377-388, 1996.
[9] W. E. L. Grimson, "Computational experiments with a feature based stereo algorithm," IEEE Trans. Pattern Analysis and Machine Intelligence, no. 1, pp. 17-34, 1985.
[10] V. Kolmogorov and R. Zabih, "Computing visual correspondence with occlusions using graph cuts," in Proc. IEEE Int'l Conf. Computer Vision, vol. 2, 2001, pp. 508-515.
[11] M. Leordeanu and M. Hebert, "A spectral technique for correspondence problems using pairwise constraints," in Proc. IEEE Int'l Conf. Computer Vision, vol. 2, 2005, pp. 1482-1489.
[12] M. Leordeanu, M. Hebert, and R. Sukthankar, "An integer projected fixed point method for graph matching and map inference," in Proc. Advances in Neural Information Processing Systems, 2009, pp. 11141122.
[13] M. Leordeanu, R. Sukthankar, and M. Hebert, "Unsupervised learning for graph matching," Int. J. Computer Vis., vol. 96, pp. 28-45, 2012.
[14] F. F. Li and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2005, pp. 524-531.
[15] C. K. Liu, B. F. Wang, J. S. Shen, G. S. Tang, J. Wang, and W.-H. Wan, "A positioning method of chang'e-3 rover in large-span states based on cylindrical projection of images," in Proc. World Congress on Intelligent Control and Automation, 2014.
[16] Z. Y. Liu and H. Qiao, "Graduated nonconvexity and concavity procedure for partial graph matching," IEEE Trans. Pattern Analysis and Machine Intelligence, 2014.
[17] Z. Y. Liu, H. Qiao, and L. Xu, "An extended path following algorithm for graph-matching problem," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 34, no. 7, pp. 1451-1456, 2012.
[18] Z. Liu, J. An, and Y. Jing, "A simple and robust feature point matching algorithm based on restricted spatial order constraints for aerial image registration," IEEE Trans. Geoscience and Remote Sensing, vol. 50, no. 2, pp. 514-527, 2012.
[19] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," Int'l J. of Computer Vision, vol. 60, no. 2, pp. 91-110, 2004.
[20] M. Maes, "Polygonal shape recognition using string-matching techniques," Pattern Recognition, vol. 24, no. 5, pp. 433-440, 1991.
[21] K. Riesen and H. Bunke, "Approximate graph edit distance computation by means of bipartite graph matching," Image and Vis. Computing, vol. 27, no. 7, pp. 950-959, 2009.
[22] R. A. Wagner and M. J. Fischer, "The string-to-string correction problem," Journal of the ACM, vol. 21, no. 1, pp. 168-173, 1974.
[23] Y. Xu, P. Monasse, T. Géraud, and L. Najman, "Tree-based morse regions: A topological approach to local feature detection," IEEE Trans. Image Processing, vol. 23, no. 12, pp. 5612-5625, 2014.
[24] X. Yang, C. K. Liu, Z. Y. Liu, H. Qiao, B.-F. Wang, and Z. D. Wang, "A probabilistic spectral graph matching algorithm for robust correspondence between lunar surface images," in Proc. World Congress on Intelligent Control and Automation, 2014.
[25] X. Yang, H. Qiao, and Z.-Y. Liu, "Outlier robust point correspondence based on gnccp," Pattern Recognition Letters, pp. 8-14, 2015.
[26] M. Zaslavskiy, F. Bach, and J. P. Vert, "A path following algorithm for the graph matching problem," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 31, no. 12, pp. 2227-2242, 2009.
[27] F. Zhou and F. De la Torre, "Factorized graph matching," in Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2012, pp. 127-134.


[^0]:    Yuren Zhang, Xu Yang, Hong Qiao, and Zhiyong Liu are with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, 95 Zhongguancun East Road, 100190, Beijing, China

    Chuankai Liu and Baofeng Wang are with Beijing Aerospace Flight Control Center, Beijing, China.

    This work is supported by the National Science Foundation of China (NSFC) (grants $61375005,61033011,61210009,61101221$ ) and by National Key Technology R\&D Program (grant 2012BAI34B02).

[^1]:    ${ }^{3}$ Available at http://www.lpi.usra.edu/resources/apollo/catalog/ 70mm/ mission/?15

