Accurate hierarchical stereo matching based on 3D plane labeling of superpixel for stereo images from rovers

Haichao Li, Zhi Li, Jianbin Huang, Bo Meng and Zhimin Zhang

Abstract
An accurate hierarchical stereo matching method is proposed based on continuous 3D plane labeling of superpixel for rover’s stereo images. This method can infer the 3D plane label of each pixel combined with the slanted-patch matching strategy and coarse-to-fine constraints, which is especially suitable for large-scale scene matching with low-texture or textureless regions. At every level, the stereo matching method based on superpixel segmentation makes the iteration convergence faster and avoids huge redundant computations. In the coarse-to-fine matching scheme, we propose disparity constraint and 3D normal vector constraint between adjacent levels through which the disparity map and 3D normal vector map at a coarser level are used to restrict the search range of disparity and normal vector at a fine level. The experimental results with the Chang’e-3 rover dataset and the KITTI dataset show that the proposed stereo matching method is efficiently and accurately compared with the state-of-the-art 3D labeling algorithm, especially in low-texture or textureless regions. The computational efficiency of this method is about five to six times faster than the state-of-the-art 3D labeling method, and the accuracy is better.

Keywords
Stereo matching, coarse-to-fine architecture, 3D label, superpixel segmentation

Introduction
The 3D information is the main way of robot 3D visual perception.1–2 Stereo matching is an important step of disparity calculation or 3D reconstruction in binocular vision,3 which has been widely studied. However, there are still challenges in occlusion and weak texture or textureless regions, especially for the rover’s stereo images.

In recent years, the matching method based on 3D labels has increased the accuracy of stereo matching. It can not only estimate the disparity of each pixel but also estimate the 3D normal vector of each pixel. To efficiently infer 3D labels, many methods successfully use PatchMatch,4–6 which can estimate a 3D plane at each pixel.7 In recent years, inspired by randomized search and propagation of PatchMatch,6,8 many optimization methods with belief propagation (BP)9 or graph cut (GC)9 have been proposed for efficient inference in pairwise Markov random field (MRF) with large label spaces. To the best of our knowledge, the local expansion moves (LocalExp) method10 is the state-of-the-art method using local expansion moves based on GC.
The first problem is that in the LocalExp method, three grid structures with cell sizes of $5 \times 5$, $15 \times 15$, and $25 \times 25$ pixels are used, which may lead to huge redundant computations. However, the stereo matching method based on superpixel assumes that pixels in the same superpixels belong to the same 3D surface. Therefore, we propose superpixel-based segmentation, which has low complexity and makes the iteration convergence faster. The second problem is that the LocalExp method cannot handle low texture regions well. As shown in Figure 1(a), the typical characteristic of this image is that it contains low-texture or even textureless regions. The disparity map generated with the LocalExp method (https://github.com/t-taniai/LocalExpStereo) is shown in Figure 1(b), however, there are lots of mismatches and unreliable matching points in the bottom and left side of the disparity map. The third problem is that it is very difficult to assign a suitable 3D label for each pixel from the infinite continuous label space. Therefore, the LocalExp method iteratively applies the local expansion moves using GC to update and propagate local disparity planes, which not only increases the calculation but also may get some wrong matching results. In this article, based on the original LocalExp method, we introduce a coarse-to-fine stereo matching method using 3D plane labeling of superpixels. The coarsest level improves the matching robustness especially in the low texture regions, while the other levels guarantee the robustness because they make full use of disparity constraint and normal vector constraint between two adjacent levels.

Overall, the major contributions of this work are as follows:

1. A coarse-to-fine stereo matching framework combined with 3D labels is proposed, which is especially suitable for the stereo matching of low-texture or textureless regions in large-scale real scenes.
2. A matching method based on superpixel segmentation is proposed, which makes the iteration convergence faster and avoids huge redundant computations.
3. We propose disparity constraint and normal vector constraint between two adjacent levels, through which the disparity map and 3D normal vector map at a coarser level are used to restrict the search range of disparity and normal vector at a finer level.

The remainder of the article is arranged as follows: related works are present in the second section; the proposed method is given in the third section; the experimental results and conclusion are given in the fourth and fifth sections, respectively.

**Related works**

**Coarse-to-fine matching**

A few decades ago, the coarse-to-fine stereo matching strategy has been introduced into many matching methods. This kind of method first calculates a coarse resolution disparity and then uses coarse disparity to constrain the disparity search range for calculating fine disparity.

To speed up the convergence, this strategy has been widely used in global matching methods. A hierarchical stereo matching method using dynamic programming is proposed by Van Meerbergen et al., which achieves a tremendous gain in memory and speed. A hierarchical strategy using semiglobal matching is introduced to generate a half-resolution disparity map firstly, and then, it is used for the disparity computation of the original image. A coarse-to-fine stereo matching based on the more-global matching (MGM) method is proposed by Li et al., which improves the disparity accuracy compared with the original MGM. Some BP algorithms are proposed to use a coarse-to-fine strategy for reducing the complexity. A coarse-to-fine bilateral disparity structure method based on GC is introduced to reduce the computational cost and improve the disparity accuracy. In addition to the global methods, the local methods also adopt the coarse-to-fine strategy, which mainly reduces the disparity search range for the computation of finer disparity. A stereo matching method with multiscale and multiwindow is proposed to estimate disparity for restricting the disparity search.
A confidence-based multiscale stereo matching strategy has been proposed, which can obtain higher-resolution disparity maps by processing the existing lower-resolution ones.

3D label-based stereo matching
The matching cost calculation based on 3D labels is different from that based on discrete disparities stereo matching. For each pixel in one image, lots of corresponding regions can be obtained by the slanted-plane of the pixel according to different 3D labels. Therefore, the major issue with 3D labels stereo matching is the computational complexity.

An energy function with hybrid MRF is proposed by Yamaguchi et al., which contains continuous variables for 3D slanted plane and discrete random variables for each pair of neighboring segments. A slanted plane model is optimized with simple linear iterative clustering (SLIC) segmentation based on a pre-estimated disparity map, and simultaneously, the states of occlusion and coplanarity over adjacent segment pairs are also analyzed. Bleyer et al. propose to segment the scene into planar superpixels and estimate each pixel’s optimal 3D plane during all the possible slanted planes. Olsson et al. represent second-order surface smoothness with 3D labels. Heise et al. propose to integrate PatchMatch method into a variational smoothing formulation. Li et al. use pixelwise 3D label optimization by fusing multilayer superpixels iteratively. Taniai et al. propose to use LocalExp based on GC.

Methodology
The proposed matching method constructs a hierarchical architecture and works in a coarse-to-fine scheme. The advantage of this method is that some mismatches can be solved at a coarse level, especially in the low-texture or textureless regions. Given robust disparity map at the coarser level, disparity reconstruction at the finer level can be correctly obtained based on disparity constraint and normal vector constraint.

The input two images are firstly corrected by epipolar rectification, and then, they are decomposed into a L-level pyramid. As can be seen from Figure 2, the matching process of the two images at the coarsest level L mainly consists of three steps: superpixel segmentation, iterative optimization based on superpixel, and postprocessing. From level \( L = L - 1 \) to level \( L = 1 \) (the finest level), the disparity map at the coarser level will guide the disparity search range at the finer level. For each level, the disparity map can be computed by the following four steps: coarse-to-fine
disparity constraint and 3D label constraint, superpixel segmentation, iterative optimization based on superpixel, and postprocessing. Here, postprocessing is a left-to-right consistency check.15

Superpixel-based image segmentation

The stereo images are firstly decomposed into many non-overlapping segments with SLIC algorithm.28 Figure 3(a) shows an image with about 200 SLIC superpixels. And then, as shown in Figure 3(b), we define three windows for each superpixel $S_i$. The unit window $U_i$ represents the extended window of the Minimum Enclosing Rectangle (MER) of $S_i$. The optimization window $O_i$ is an extended window, which extends $r$-pixel width around the MER of neighborhood superpixels of $S_i$. The affine transformation window $A_i$ is a rectangle window, which extends $r$-pixel width around $O_i$.

Iterative randomized optimization based on superpixel

According to the three defined windows, $U_i$, $O_i$, and $A_i$, for each $S_i$, the stereo matching based on superpixel is described as follows. Let $\mathcal{F}$ be a set of 3D labels, which correspond to disparities, and we randomly choose a pixel from $U_i$ and assign a 3D label $f \in \mathcal{F}$. It typically needs to estimate $f$ by minimizing the following function

$$E(f) = \sum_{p \in O_i} E_p(f) + \lambda \sum_{p \in O_i} \sum_{q \in N(p)} E_{pq}(f_p, f_q)$$

where $\lambda$ is a coefficient. $E_p(f)$ is the cost of pixel $p$ with label $f$ called the data term, which measures the photo-consistency between matching pixels. $E_{pq}(f_p, f_q)$ is the cost of pixel $p$ and its neighboring pixels $q \in N(p)$ by assigning labels $f_p$ and $f_q$ are called the smoothness term.

(1) Data term

For a pixel $p(p_x, p_y) \in O_i$, its disparity $d_p$ is calculated by a 3D plane $d_p = ap_x + bp_y + c$ to avoid the frontal-parallel problem. Therefore, the aim becomes to seek an optimal 3D label $f$ for every pixel in the left and right images, which can minimize the energy function $E(f)$. The data term of $p$ in the left image is then defined as

$$E_p(f) = \sum_{x \in W_p} \omega_{pq} \rho(s, w_y(s))$$

where $W_p$ is a window centered at $p$ with radius of $r$-pixel (shown in Figure 3(b)). Here, we use the same weight $\omega_{pq}$, as proposed in the LocalExp method.18 Using 3D label $(a, b, c)$, the function $\rho(s, w_y(s))$ measures the pixel dissimilarity between $s(s_x, s_y)$ in window $W_p$ of the left image and its matching point $w_y(s)$ of the right image

$$w_y(s) = s - (as_x + bs_y + c, 0)^T$$

which is defined as

$$\rho(s, w_y(s)) = (1 - \mu) \min(||I_L(s) - I_R(w_y(s))||, \tau_{col})$$

$$+ \mu \min(||\nabla I_L(s) - \nabla I_R(w_y(s))||, \tau_{grad})$$

(4)

where $\nabla I$ represents the gradient of image $I$, and $\mu$ is a factor. The two terms are truncated by $\tau_{col}$ and $\tau_{grad}$ to increase the robustness for occluded regions. $I_R(w_y(s))$ is RGB or gray value of the corresponding pixel in the right image.

As shown in Figure 4, for each affine transformation window $A_i$ in the left image, the corresponding windows in the right image can be obtained by affine transformation with different 3D labels $f = (a, b, c)$.

(2) Smoothness term

The smoothness term can be defined as the following which is the same as in the literature

$$E_{pq}(f_p, f_q) = \max(\omega_{pq}, \varepsilon) \min[\psi_{pq}(f_p, f_q), \tau_{dis}]$$

(5)

where $\psi_{pq}(f_p, f_q)$ is truncated at $\tau_{dis}$, and $\omega_{pq}$ is defined as

$$\omega_{pq} = \exp(-||I_L(p) - I_L(q)||/\gamma)$$

(6)

where $\gamma$ is a parameter, and $\varepsilon$ is a small constant value. The function $\psi_{pq}(f_p, f_q)$ penalizes the discontinuity between $f_p$ and $f_q$ in terms of disparity as

$$\psi_{pq}(f_p, f_q) = |d_p(f_p) - d_p(f_q)| + |d_q(f_q) - d_q(f_p)|$$

(7)

where the first term measures the difference between $f_p$ and $f_q$ by their disparity values at $p$, and the second term is defined similarity at $q$. 

Figure 3. (a) SLIC superpixels. (b) Unit window $U_i$ (in pink) is the extended window of the MER of $S_i$, optimization window $O_i$ (in blue) is the extended window, which extends $r$-pixel width around the MER of neighborhood superpixels of $S_i$, and the $r$-pixel extended window $A_i$ (in cyan) is the affine transformation window. MER: minimum enclosing rectangle; SLIC: simple linear iterative clustering.
We use a local expansion method similarly to the LocalExp method\textsuperscript{10} to randomly iterate optimization based on superpixels

\[ f(t+1) = \arg \min_{f_p} E\left(f'| f_p \in \{f^{(t)}_p, \alpha\}\right) \]  

for all possible labels \( \forall \alpha \in \mathcal{A} \), and \( \mathcal{A} \) is a 3D continuous space. Here, the binary variable \( f_p \) for every pixel \( p \) is assigned either by its current label \( f^{(t)}_p \) or a candidate label \( \alpha \).

Figure 4. (a) The affine transformation window and (b) its corresponding windows obtained by different 3D labels.

Algorithm 1. Iterative randomized optimization

**Input:** Current label \( f \), three windows \( U_i, O_i, A_i \).

Disparity search range of each pixel.

At level \( l \in L-1, \ldots, 1 \), we need input the reference normal vector and angle search range of each pixel.

**Output:** Updated label \( f \) (only the labels \( f_p \) at \( p \in O_i \) are updated).

1. **repeat** expansion proposer (propagation):
   2. \( \alpha \leftarrow f \), with randomly chosen \( r \in U_i \);
   3. \( f \leftarrow \arg \min E(f'| f_p \in \{f_p, \alpha\}, p \in O_i) \);
   4. until \( K_{exp} \) times;
5. **repeat** RANSAC proposer (optional):
   6. \( \alpha \leftarrow \text{RANSAC}(f, O_i) \);
   7. \( f \leftarrow \arg \min E(f'| f_p \in \{f_p, \alpha\}, p \in O_i) \);
   8. until \( K_{RANSAC} \) times;
9. **repeat** randomization proposer (refinement):
   10. \( \alpha \leftarrow f \), with randomly chosen \( r \in U_i \);
   11. \( \alpha \leftarrow \alpha + \Delta \alpha \);
   12. \( f \leftarrow \arg \min E(f'| f_p \in \{f_p, \alpha\}, p \in O_i) \);
   13. \( \Delta \alpha \leftarrow \Delta \alpha / 2 \);
   14. until \( K_{rand} \) times;

Our expansion method is also in two ways: localization and spatial propagation. By localization, we use different candidate labels \( \alpha \), instead of using the same label \( \alpha \) for all pixels. By spatial propagation, we propagate currently assigned 3D labels to the nearby pixels via GC.

Our local expansion iteratively is shown in Algorithm 1, which also includes propagation (lines 1–4), RANSAC (lines 5–8), and refinement (lines 9–14) steps similarly to the LocalExp method.\textsuperscript{10} In addition to the refinement step, the other two steps in our local expansion are the same as those in the LocalExp method.\textsuperscript{10}
The candidate label $\alpha = (a, b, c)$ of pixel $r(x, y)$ can be converted to disparity $d = ax + by + c$ and normal vector $n$. For each iteration $m$, a smaller disparity search range can be computed as

$$
\begin{align*}
    d'_m &= \max(d_{\min}, d - \Delta d) \\
    d'_m &= \min(d_{\max}, d + \Delta d)
\end{align*}
$$

where $[d_{\min}, d_{\max}]$ is the disparity search range of pixel $r$, $\Delta d = (d_{\max} - d_{\min})/2^m$, and $m = 1, 2, \cdots, K_{\text{rand}}$. We randomly select a disparity from the disparity search range $d' \in [d_{\min}, d'_{\max}]$ for each iteration $m$. In the LocalExp method, all pixels use the full disparity search range, while our method has a smaller disparity search range for each pixel. Therefore, this method has a faster convergence speed than the LocalExp method.

For normal vector, we add a random vector $\Delta' n$ of size $\|\Delta' n\|_2 = r_n$ to obtain a new normal vector $n'$, $n' \leftarrow n + \Delta' n$. Here, different from the LocalExp method, at level $l \in L - 1, \cdots, 1$, we should use the angle search range obtained from “Coarse-to-fine disparity constraint and 3D normal vector constraint” section. The angle $\theta$ between the new normal vector $n'$ and the input reference normal vector is first computed. Then, we repeatedly compute $n'$ with randomly vector $\Delta' n$, $n' \leftarrow n + \Delta' n$, until $\theta \in [\theta_{\text{min}}, \theta_{\text{max}}]$. We initialize $r_n$ by setting $r_n \leftarrow 1$, and update it by $r_n = r_n/2$ for each iteration. Finally, we convert $d'$ and $n'/|n'|$ to the plane representation $\alpha \leftarrow (a', b', c')$ to obtain a perturbed candidate label.

After that, we then update the current labels of pixels $p$ in the optimization window $O_u$, using the current labels $f_p$ or the new candidate label $\alpha$. Therefore, we obtain an improved solution with lower or equal energy as its minimum value.

**Coarse-to-fine disparity constraint and 3D normal vector constraint**

The above matching is based on continuous 3D label of superpixel, however, the large disparity search range may lead to inaccurate disparity, especially in textureless or low-texture regions. For yielding accurate disparity in those regions, based on our previous coarse-to-fine disparity constraint method proposed by Li et al., we introduce a new coarse-to-fine 3D normal vector constraint.

As can be seen from Figure 5, for each pixel $(x, y)$ at finer level-$l$, the disparity search range and the angle search range can be computed based on the following steps.

Step 1: Compute $x' = \lfloor x/2 \rfloor$ and $y' = \lfloor y/2 \rfloor$, and find the pixels with valid disparities around pixel $(x', y')$ at level-$l + 1$, which are expressed as $p_{ul}$, $p_{ur}$, $p_l$, and $p_{dl}$, and $p_{dr}$ and $p_{ul}$, as shown in Figure 5(b1) or (b2).

Step 2: Find the minimum disparity $d'_{\min}$ and the maximum disparity $d'_{\max}$ among the above pixels. Therefore, for pixel $(x, y)$ at level-$l$, the disparity search range can be defined as $[d_{\min}(x, y), d_{\max}(x, y)]$

$$
\begin{align*}
    d_{\min}(x, y) &= 2d'_{\min} - \Delta D \\
    d_{\max}(x, y) &= 2d'_{\max} + \Delta D
\end{align*}
$$

where $\Delta D$ is a given disparity margin.

Step 3: Select the normal vector of any pixel in $p_{ul}$, $p_{ur}$, $p_l$, $p_{dr}$, and $p_{dl}$ as the reference normal vector, and then, calculate the angle between the normal vector of other pixels and the reference normal vector. The minimum and maximum angles are, respectively, expressed as $\theta'_{\text{min}}$ and $\theta'_{\text{max}}$. Therefore, for pixel $(x, y)$ at level-$l$, the angle search range $[\theta_{\text{min}}(x, y), \theta_{\text{max}}(x, y)]$ can be defined as

$$
\begin{align*}
    \theta_{\text{min}}(x, y) &= \theta'_{\text{min}} - \Delta \theta \\
    \theta_{\text{max}}(x, y) &= \theta'_{\text{max}} + \Delta \theta
\end{align*}
$$

where $\Delta \theta$ is a given angle margin.

**Experimental results**

To evaluate the performance of our proposed method, two datasets are used, that is, Chang’e-3 Yutu rover dataset (http://planetary.s3.amazonaws.com/data/change3/pcam.html) and KITTI (http://www.cvlibs.net/datasets/kitti/) dataset. Our method is compared with the state-of-the-art LocalExp method.10

We use the following settings throughout the experiments. The parameters for the data term are set to $\{\epsilon, \tau_{\text{col}}, \tau_{\text{grad}-\mu}\} = \{0.01^2, 10, 2, 0.9\}$. The size of window...
$W_p$ is set to $41 \times 41$ pixels. For the smoothness term, $\{\alpha, \tau_{dist}, \varepsilon, \gamma\} = \{1, 1, 0.01, 10\}$ and eight neighbors for $N(p)$. The above parameters in our method are set the same as those in the LocalExp method.\(^{10}\)

In the LocalExp method, three structures with cell sizes of $5 \times 5$, $15 \times 15$, and $25 \times 25$ pixels are used, and the iteration numbers $\{K_{\text{prop}}, K_{\text{RANS}}, K_{\text{rand}}\}$ are set to $\{1,1,7\}$ for the first grid structure, and $\{2, 1, 0\}$ (only propagation step) for the other two grid structures.

While in our method, the number of pyramid levels is set to 3 for Chang’e-3 rover dataset and 2 for KITTI dataset. For each level, the number of superpixels is set to 1200, which will be analyzed in “Analysis of parameter and processing time” section. The sizes of three windows $U_i$, $O_i$, and $A_i$ are adaptively adjusted according to each superpixel. For each superpixel, the iteration numbers $\{K_{\text{prop}}, K_{\text{RANS}}, K_{\text{rand}}\}$ are set to $\{1, 1, 7\}$. We iterate the main loop three times (one time only with data term and two times with both data term and smoothness) in our proposed method and eight times (two times only with data term and five times with both data term and smoothness) in the LocalExp method.

In all the experiments, we only use CPU instead of GPU for comparing our method with the LocalExp method. All the experiments are executed on a laptop with Intel Core i5-8250 1.60 Hz CPU and 8-GB memory, and our codes are implemented in Microsoft Visual Studio 2017 with C++ and OpenCV library.

**Evaluation on chang’e-3 rover dataset**

The input Chang’e-3 stereo images are firstly rectified by epipolar rectification method.\(^{27}\) During the coarse-to-fine processing, we construct three-level pyramid ($L = 3$) for Chang’e-3 Yutu rover dataset. We choose five stereo pairs from the rover dataset, and the left images are shown in the top row of Figure 6, which contain many low-texture regions. The disparity search ranges of the five stereo pairs are about $[-97,-9]$, $[-40,150]$, $[-20,119]$, $[3,105]$, and $[-132,3]$ pixels, respectively.

We qualitatively evaluate our proposed method compared with the LocalExp method.\(^{10}\) As can be seen from Figure 6, the disparity map computed by our method (the third row of Figure 6) is superior to that computed by the LocalExp method (the second row of Figure 6). For example, the stereo pair-1 (shown in Figure 6(a)) has huge rock, disparity discontinuous, and many low-texture regions. We can see that our method can obtain high-precision disparity in many low-texture regions, while the LocalExp method obtains false or unreliable disparity in these regions. The other four stereo pairs (shown in Figure 6(b) to (e)) include low texture, repetitive texture, precipice, and disparity discontinuous regions, and especially, there is strong light intensity in the left of Figure 6(d). We can see that all the precipice, low-texture, or repetitive texture regions, varying light conditions, and strong light regions are perfectly reconstructed with our proposed method, while there exist some wrong disparities in these regions with the LocalExp method.

**Evaluation on KITTI dataset**

Furthermore, due to the lack of standard datasets with the ground truth of the rover’s stereo images, the accuracy of our method cannot be quantitatively evaluated. Therefore,
we choose the KITTI dataset (http://www.cvlibs.net/datasets/kitti/) created from a driving platform,29 whose imaging mode is similar to the rover imaging mode, and the sizes of images are 1226 × 370 pixels. We construct two-level pyramid ($L = 2$) for KITTI dataset with our method.

In this experiment, we use the whole 194 training image pairs with ground truth disparity maps for reflective regions available, and the evaluation metric is an error threshold 3 pixels, which is the same as the KITTI benchmark.30 We test the accuracy and efficiency of disparity maps computed by our proposed method, our improved MGM method,15 and the LocalExp method10 with KITTI dataset.

The quantitative results with different methods are listed in Table 1, which gives the average results of all the 194 training image pairs in reflective regions. In this table, Out-Noc is the percentage of erroneous pixels in nonoccluded areas. Out-All is the percentage of erroneous pixels in total. Avg-Noc is the average disparity error in nonoccluded areas. Avg-All is the average disparity error in total. It is noted that the proposed method performs better than our previous MGM method15 and the LocalExp method.10 Our method has a significant improvement in the percentage of erroneous pixels and average disparity error, and the Out-Noc and Out-All are decreased from 12.67%, 13.57% (LocalExp method) to 7.27%, 8.17% (our method) respectively, while the Avg-Noc and Avg-All are all decreased from 2.05 pixels, 2.19 pixels (LocalExp method) to 1.37 pixels, 1.53 pixels (our method). Figure 7 shows several examples of disparity maps, and Table 2 gives the corresponding quantitative results of our previous MGM, LocalExp method, and the proposed method. We can see that there are significant improvements in both percentage of erroneous pixels and average disparity error with our proposed method.

As shown in Figure 7, for low-texture or textureless regions, for example, shadowed regions, roads, and strong light conditions, the LocalExp method tends to generate errors due to nonconvergence. While in disparity maps generated by our method, the errors in these regions are almost eliminated.

**Efficiency evaluation**

We evaluate the efficiency of our method and the LocalExp method, which are both implemented with CPU. The two

| Method          | Out-Noc (%) | Out-All (%) | Avg-Noc (pixels) | Avg-All (pixels) |
|-----------------|-------------|-------------|------------------|------------------|
| Our MGM15       | 9.33        | 11.25       | 1.59             | 1.84             |
| LocalExp        | 12.67       | 13.57       | 2.05             | 2.19             |
| Our proposed    | 7.27        | 8.17        | 1.37             | 1.53             |

Figure 7. Disparity maps of five images in KITTI datasets with different methods, from left to right: Nos 0, 14, 102, 140, and 192. Disparity maps are visualized using color-map. (a) Left image, (b) ground truth pixels, (c) color map of disparity map with our previous MGM method, (d) disparity map with the LocalExp method, (e) color map of (d), (f) disparity map with our proposed method, and (g) color map of (f).
methods mainly have two computation parts: the calculations of data term of equation (3) and smoothness term of equation (5). For the two terms, they all require $O(|W|)$ of computation for each term, where $|W|$ is the window size. For our method, as shown in Figure 3, $|W|$ is the size of $O_i$ in data term, and it is the size of $A_i$ in smoothness term. For the LocalExp method, $|W|$ is the size of filtering region $M$ in data term, and it is the size of expansion region $R$ in smoothness term.

The computational complexity of our method is estimated by the sum of computation for all the superpixels at all levels

$$O(T_{Proposed}) = \sum_{levels} \sum_{n_1} \sum_{i \in \text{superpixels}} [O(|O_i|) + O(|A_i|)]$$

(12)

While the complexity of the LocalExp method is estimated by the sum of computation for all the cell sizes of $5 \times 5$, $15 \times 15$, and $25 \times 25$ pixels

$$O(T_{LocalExp}) = \sum_{S=5}^{25} \sum_{I} O(|M_S|) + O(|R_S|)$$

(13)

where $n_1$ and $n_2$ are the optimization times of our method and LocalExp method, respectively, and they include times of data term and times of smoothness term. Here, $n_1$ is set to 3 (one time only with data term and two times with both data term and smoothness term) and $n_2$ is set to 7 (two times only with data term and five times with both data term and smoothness term).

We approximate compare the computational complexity of our method with that of the LocalExp method for five stereo pairs of Yutu rover. Table 3 gives the approximately computational complexity of our method estimated by equation (12) and that of the LocalExp method estimated by equation (13). Finally, the ratio between the complexity of our method and that of LocalExp can be computed.

![Figure 8. Processing time of five stereo pairs of Chang’e-3 Yutu rover using LocalExp method and our method.](image)

As shown in Figure 9, we compare the processing time of our method with that of the LocalExp method based on...
all the stereo pairs of KITTI dataset, and our method has faster convergence and greatly reduces the computational complexity than the LocalExp method. It should be noted that for fair matching comparison, the processing time of the LocalExp method was obtained by running the downloaded C++ source code (https://github.com/t-taniai/LocalExpStereo) on our laptop with only CPU implementations. For all the 194 training images on KITTI dataset, the average processing time with the LocalExp method is about 1879 s, and the time with our SLIC method is about 350 s. By comparing the LocalExp method with our method, we can observe that our method is 5.3 times faster than the LocalExp method.

We consider that there are mainly two factors contributing to this acceleration of our method. First, our method with SLIC segmentation makes its convergence faster. Meanwhile, in these experiments, our method iterates three times in the main loop (one time only with data term and two times with data term and smoothness), which can obtain good results, while the LocalExp method iterates eight times (two times only with data term and five times with data term and smoothness). Second, in the LocalExp method, three different structures with cell sizes of $5 \times 5$, $15 \times 15$, and $25 \times 25$ pixels are used, while in our method, the number of superpixels for all levels is set to 1200.

**Analysis of parameter and processing time**

We evaluate the sensitivity of our method to some parameters. The number of pyramid levels $L$ and the number of superpixels are firstly analyzed in detail.

Figure 10 shows the performance of no. 44 image pair in KITTI dataset as an example with different pyramid levels and different number of SLIC superpixels, which contain large untextured regions and shadow regions. We evaluate the disparity maps generated with five different number of SLIC superpixels (400, 800, 1200, 1600, and 2000) and four different number of pyramid levels, that is, (4, 3, 2, and 1). The error rates Out-Noc and Out-All generated with different levels and different numbers of SLIC are shown in Figure 10(d) and (e). The error rate of using two-level is...
much lower than that of using one-level, three-level, or four-level. The Avg-Noc and Avg-All with different levels and different number of SLIC are shown in Figure 10(f) and (g). The disparity error of using two-level is much lower than that of using the other levels. Therefore, we choose two-level pyramid. Figure 10(h) gives the processing time with different levels and different number of SLIC. When the number of pyramid levels is fixed, the processing time increases almost linearly with the number of SLIC. Figure 10(i) shows the error rate of Out-Noc and Out-All with a different number of SLIC at two-level pyramid. When the number of SLIC is 1200, the disparity error rate is much lower than that of using the other number of SLIC.

And then, to further verify the relationship between optimization times and matching results, we give different quantitative results under four different optimization times, that is, one time (one time with both data term and smoothness term), two times (one time only with data term and one time with both data term and smoothness term), three times (one time only with data term and two times with both data term and smoothness term), and five times (two times only with data term and three times with both data term and smoothness term). The performance results with different iteration times under the conditions of two-level pyramid and 1200 superpixels are given in Table 4. As the main loop increases from 1 to 5, the error rates of Out-Noc and Out-All decrease from 7.80%, 8.30% to 3.14%, 3.23%, and the average disparity errors of Avg-Noc and Avg-All are getting smaller and smaller, while the processing time is getting longer. When the optimization times reach 5, not only the error rate is increased but also the processing time is increased. Therefore, the optimization times are set to three (one time only with data term and two times with both data term and smoothness term) in our method.

| Optimization times | One time | Two times | Three times | Five times |
|--------------------|----------|-----------|-------------|------------|
| Out-Noc (%)        | 7.80     | 4.71      | 3.14        | 4.91       |
| Out-All (%)        | 8.30     | 5.09      | 3.23        | 4.95       |
| Avg-Noc (pixels)   | 1.67     | 1.27      | 0.96        | 1.25       |
| Avg-All (pixels)   | 1.7      | 1.29      | 0.98        | 1.26       |
| Processing time (s)| 196      | 222       | 351         | 634        |

After that, we evaluate the sensitivity of our method to some other parameters. The coefficient $\lambda$ in equation (1), $\Delta D$ in equation (10), and $\Delta \theta$ in equation (11) is analyzed based on no. 44 image pair in KITTI dataset. We choose different values of $\lambda$, 0.1, 0.5, 1, 2, 5, and 10, and the results are given in Table 5. When $\lambda$ equals to 0.5 or 1, it can get better results. However, it consumes more time with larger $\lambda$. Therefore, we choose $\lambda = 0.5$. The results with different $\Delta D$ and $\Delta \theta$ are given in Tables 6 and 7. From Table 6, we can see that it consumes more time with $\Delta D$ increasing, while it can obtain good result beginning from $\Delta D \geq 1$. As given in Table 7, the processing time is almost the same with different values of $\Delta \theta$. When $\Delta \theta \geq 1$, we can get a slightly better result. Therefore, we choose $\Delta D = 1$ and $\Delta \theta = 1$ for all the experiments in this article.

| $\lambda$ | 0.1 | 0.5 | 1   | 2   | 5   | 10  |
|-----------|-----|-----|-----|-----|-----|-----|
| Out-Noc (%) | 10.78 | 3.14 | 3.53 | 4.47 | 8.26 | 12.91 |
| Out-All (%) | 10.70 | 3.23 | 3.60 | 5.27 | 7.99 | 14.03 |
| Avg-Noc (pixels) | 3.36 | 0.96 | 0.96 | 1.09 | 1.61 | 1.7 |
| Avg-All (pixels) | 3.34 | 0.98 | 0.98 | 1.14 | 1.87 | 1.76 |
| Processing time (s) | 296 | 351 | 386 | 511 | 563 | 598 |

| $\Delta D$ | 0 | 1 | 2 | 3 |
|------------|---|---|---|---|
| Out-Noc (%) | 3.35 | 3.14 | 3.17 | 3.2 |
| Out-All (%) | 3.43 | 3.23 | 3.25 | 3.25 |
| Avg-Noc (pixels) | 1.05 | 0.96 | 0.94 | 0.90 |
| Avg-All (pixels) | 1.07 | 0.98 | 0.97 | 0.92 |
| Processing time (s) | 330 | 351 | 399 | 431 |

| $\Delta \theta$ | 0 | 1 | 2 | 3 |
|----------------|---|---|---|---|
| Out-Noc (%) | 3.25 | 3.13 | 3.16 | 3.14 |
| Out-All (%) | 3.35 | 3.20 | 3.27 | 3.23 |
| Avg-Noc (pixels) | 0.98 | 0.97 | 0.97 | 0.96 |
| Avg-All (pixels) | 1.0 | 0.99 | 0.99 | 0.98 |
| Processing time (s) | 346 | 351 | 349 | 358 |

| $\lambda$ | 0.1 | 0.5 | 1   | 2   | 5   | 10  |
|-----------|-----|-----|-----|-----|-----|-----|
| Out-Noc (%) | 10.78 | 3.14 | 3.53 | 4.47 | 8.26 | 12.91 |
| Out-All (%) | 10.70 | 3.23 | 3.60 | 5.27 | 7.99 | 14.03 |
| Avg-Noc (pixels) | 3.36 | 0.96 | 0.96 | 1.09 | 1.61 | 1.7 |
| Avg-All (pixels) | 3.34 | 0.98 | 0.98 | 1.14 | 1.87 | 1.76 |
| Processing time (s) | 296 | 351 | 386 | 511 | 563 | 598 |

| $\Delta D$ | 0 | 1 | 2 | 3 |
|------------|---|---|---|---|
| Out-Noc (%) | 3.35 | 3.14 | 3.17 | 3.2 |
| Out-All (%) | 3.43 | 3.23 | 3.25 | 3.25 |
| Avg-Noc (pixels) | 1.05 | 0.96 | 0.94 | 0.90 |
| Avg-All (pixels) | 1.07 | 0.98 | 0.97 | 0.92 |
| Processing time (s) | 330 | 351 | 399 | 431 |

| $\Delta \theta$ | 0 | 1 | 2 | 3 |
|----------------|---|---|---|---|
| Out-Noc (%) | 3.25 | 3.13 | 3.16 | 3.14 |
| Out-All (%) | 3.35 | 3.20 | 3.27 | 3.23 |
| Avg-Noc (pixels) | 0.98 | 0.97 | 0.97 | 0.96 |
| Avg-All (pixels) | 1.0 | 0.99 | 0.99 | 0.98 |
| Processing time (s) | 346 | 351 | 349 | 358 |

**Conclusion**

An accurate stereo matching method based on continuous 3D plane label estimation has been proposed for rover’s stereo images. Unlike the previous LocalExp method using three structures with different cell sizes, we propose a coarse-to-fine hierarchical stereo matching method based on superpixels, which makes the convergence faster. The experimental results on Chang’e-3 lunar rover dataset and KITTI dataset show that, compared with the state-of-the-art 3D labeling method, our method can generate more accurate disparity maps, especially in low-texture regions of images.

Although we observe that our method is several times faster than the LocalExp method, the processing time of our method still needs several hundred seconds, which cannot meet the actual application requirements of rover. Therefore, we will implement the parallel processing of the algorithm on GPU processor to meet the speed requirements of practical applications in the future.
Acknowledgements
The authors would like to thank the editors and anonymous reviews for their valuable comments and helpful suggestions, which greatly improve the article’s quality.

Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the National Nature Science Foundation of China under grant no. 61773383.

ORCID iD
Haichao Li https://orcid.org/0000-0003-1248-2062

References
1. Sun Y, Liu M, and Meng QH. Improving RGB-D SLAM in dynamic environments: a motion removal approach. Robot Autom Syst 2017; 89: 110–122.
2. Sun Y, Liu M, and Meng QH. Active perception for foreground segmentation: an RGB-D data-based background modeling method. IEEE Trans Autom Sci Eng 2019; 16(4): 1596–1609.
3. Jianhui N. Stereo matching method for non-coding circular reference points based on motion consistency. J Comput Sci 2018; 27: 454–461.
4. Besse F, Rother C, Fitzgibbon A, et al. PMBP: PatchMatch belief propagation for correspondence field estimation. Int J Comput Vision 2014; 110(1): 2–13.
5. Lu J, Yang H, Min D, et al. Patch Match filter: efficient edge-aware filtering meets randomized search for fast correspondence field estimation. In: Proceedings of IEEE conference on computer vision and pattern recognition (CVPR), Portland, OR, USA, 23–28 June 2013, pp. 1854–1861. IEEE.
6. Barnes C, Shechtman E, Finkelstein A, et al. PatchMatch: a randomized correspondence algorithm for structural image editing. In: Proceedings of ACM SIGGRAPH (ed H Hoppe), New Orleans, LA, USA, Vol. 28, August 2009, pp. 24:1–24:11. ACM.
7. Bleyer M, Rhemann C, and Rother C. PatchMatch stereo matching with slanted support windows. In: Proceedings of the british machine vision conference, (ed J Hoey, S McKenna and E Trucco), Dundee, UK, 29 August 2011, pp. 14.1–14.11. BMVA.
8. Barnes C, Shechtman E, Goldman DB, et al. The generalized PatchMatch algorithm. In: Proceedings of european conference on computer vision, Crete, Greece, 5 September 2010, pp. 29–43.
9. Tanai T, Matsushita Y, and Naemura T. Graph cut based continuous stereo matching using locally shared labels. In: Proceedings of IEEE conference on computer vision and pattern recognition, Columbus, OH, USA, 23–28 June 2014, pp. 1613–1620.
10. Tanai T, Matsushita Y, Sato Y, et al. Continuous 3D label stereo matching using local expansion moves. IEEE Trans Pattern Anal Mach Intell 2018; 40(11): 2725–2739.
11. Marr D and Poggio T. A computational theory of human stereo vision. Proc R Soc Lond B Biol Sci 1979; 204(1156): 301–328.
12. Zhang K, Fang Y, Min D, et al. Cross-scale cost aggregation for stereo matching. IEEE Trans Circuits Syst Video Technol 2017; 27(5): 965–976.
13. Van Meerbergen G, Vergauwen M, Pollefeys M, et al. A hierarchical symmetric stereo algorithm using dynamic programming. Inter J Comput Vision 2002; 47(1–3): 275–285.
14. Hermann S and Klette R. Evaluation of a new coarse-to-fine strategy for fast semi-global stereo matching. Adv Image Video Technol 2012; 7087: 395–406.
15. Li HC, Chen L, and Li F. An efficient dense stereo matching method for planetary rover. IEEE Access 2019; 7: 48551–48564.
16. Facciolo G, de Franchis C, and Meinhardt E. MGM: a significantly more global matching for stereovision. In: British machine vision conference, Swansea, UK, 7–10 September 2015.
17. Felzenszwalb P and Huttenlocher D. Efficient belief propagation for early vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Vol. 1, Washington, DC, USA, June–July 2004, pp. 261–268. IEEE.
18. Yang Q, Wang L, Yang R, et al. Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling. IEEE Trans Pattern Anal Mach Intell 2009; 31(3): 492–504.
19. Wang YC, Tung CP, and Chung PC. Efficient disparity estimation using hierarchical bilateral disparity structure based graph cut algorithm with a foreground boundary refinement mechanism. IEEE Trans Circuits Syst Video Technol 2013; 23(5): 784–801.
20. Buades A and Facciolo G. Reliable multi-scale and multi-window stereo matching. SLAM J Imaging Sci 2015; 8(2): 888–915.
21. Li S, Chen K, Song M, et al. Robust, efficient depth reconstruction with hierarchical confidence-based matching. IEEE Trans Image Process 2017; 26(7): 3331–3343.
22. Yamaguchi K, Hazan T, McAllester D, et al. Continuous Markov random fields for robust stereo estimation, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 45–58.
23. Yamaguchi K, McAllester D, and Urtasun R. Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In: European conference on computer vision, Zurich, Switzerland, 6–12 September 2014, pp. 756–771.
24. Olsson C, Ulen J, and Boykov Y. In defense of 3D-label stereo. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Portland, OR, USA, 23–28 June 2013, pp. 1730–1737. IEEE.
25. Heise P, Klose S, Jensen B, et al. PM-Huber: PatchMatch with huber regularization for stereo matching. In: Proceedings of the IEEE international conference on computer vision, Sydney, NSW, Australia, 1–8 December 2013, pp. 2360–2367. IEEE.
26. Li L, Zhang S, Yu X, et al. PMSC: PatchMatch-based super-pixel cut for accurate stereo matching. *IEEE Trans Circuits Syst Video Technol* 2018; 28(3): 679–692.

27. Monasse P. Quasi-Euclidean epipolar rectification. *Image Process Line* 2011; 1: 187–199.

28. Achanta R, Shaji A, Smith K, et al. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans Pattern Anal Mach Intell* 2012; 34(11): 2274–2282.

29. Geiger A, Lenz P, and Urtasun R. Are we ready for autonomous driving? The KITTI vision benchmark suite. In: Proceedings of IEEE conference on computer vision and pattern recognition, Providence, RI, USA, 16–21 June 2012, pp. 3354–3361. IEEE.

30. Geiger A, Lenz P, and Urtasun R. The KITTI vision benchmark suite. http://www.cvlibs.net/datasets/kitti/eval_stereo_flow.php?benchmark=stereo