Superpixel-based graph cuts for accurate stereo matching

Liting Feng*, Kaihuai Qin
Tsinghua University, Beijing, China
*flt14@mails.tsinghua.edu.cn

Abstract. Estimating the surface normal vector and disparity of a pixel simultaneously, also known as three-dimensional label method, has been widely used in recent continuous stereo matching problem to achieve sub-pixel accuracy. However, due to the infinite label space, it’s extremely hard to assign each pixel an appropriate label. In this paper, we present an accurate and efficient algorithm, integrating patchmatch with graph cuts, to approach this critical computational problem. Besides, to get robust and precise matching cost, we use a convolutional neural network to learn a similarity measure on small image patches. Compared with other MRF related methods, our method has several advantages: its sub-modular property ensures a sub-problem optimality which is easy to perform in parallel; graph cuts can simultaneously update multiple pixels, avoiding local minima caused by sequential optimizers like belief propagation; it uses segmentation results for better local expansion move; local propagation and randomization can easily generate the initial solution without using external methods. Middlebury experiments show that our method can get higher accuracy than other MRF-based algorithms.

1 Introduction
Numerous computer vision tasks like stereo matching, optical flow can be categorized as pixel labeling problem. The general goal is to find a solution $f$ that is both spatially smooth and edge-preserving. To achieve this goal, we usually define a Markov Random Field model[15,19].

$$E = \sum_p E(f_p) + \lambda \sum_p \sum_{q \in N_p} E(f_p, f_q)$$

Finding the exact solution $f$ that minimizes the energy function is known to be NP hard. Several optimization methods like Belief Propagation(BP) and Graph Cuts(GC) have been proposed for 1D discrete disparity labels, but they can not deal with huge label space and such methods can only obtain pixel-level accuracy. Recently, the wide use of slanted patch matching has made great progress in stereo matching[3]. The disparity $f_p$ of pixel $p = (x_p, y_p)$ can be over-parameterized by a 3D vector

$$f_p = a_p x_p + b_p y_p + c_p$$

where $(a_p, b_p, c_p)$ are three parameters of a plane in pixel $p$. With such 3D label methods, we can avoid fronto-parallel assumption that commonly leads to artifacts on 1D label stereo and get sub-pixel accuracy. However, on account of the infinite label space $R^3$, it is extremely hard to assign each pixel a proper continuous 3D label via normal optimization methods.

The Patchmatch algorithm was initially introduced to compute the nearest neighbor field(NNF) between two images. Recently, many methods[4,3,6,18] integrate patchmatch into stereo matching to reduce the huge computation through sampling and neighbor propagation, each pixel’s candidate label
is randomly generated, updated and then propagated to neighbor pixels. As described in [3], Frederic combines belief propagation with resampling from patchmatch, yielding a sequential optimizer called PMBP, which has shown good performance in pairwise Markov random field minimization than Particle BP(PBP). Further in [10], considering that a labeling solution is often spatially smooth, but changes dramatically at object boundaries, Y Li proposed a super-pixel based BP algorithm(SPM-BP), taking advantage of edge-aware cost filtering and super-pixel based particle sampling, experiments show that SPM-BP speeds up PMBP by 50-100 times and get superior results on benchmark datasets. However, all these methods mentioned above are based on BP and patchmatch, in terms of energy minimization, BP is regarded as a sequential optimizer, which changes only one node state per time. In contrast, Graph Cuts improves all nodes simultaneously through interactions across nodes. To a great extent, this global property helps GC to avoid local minima[17].

To effectively use the global strength of GC and the efficiency of patchmatch, here we propose an accurate Patchmatch-based Graph Cuts algorithm. Comparing to SPM-BP, the challenge of our method is to exploit super-pixel based structure that allows for simultaneously updating multiple pixels. Later we will discuss a region group method for super-pixel images that satisfy the sub-modular requirements of Graph Cuts and perform local alpha expansion in one region.

2 Related Work

2.1 Energy minimization

Many computer vision problems can be converted to estimate some spatially varying quantities from the noisy environment. Each pixel \( p \in P \) should be assigned a label from finite solution space \( f_p \in L[5] \). For stereo matching, the labels are disparities and the solution \( f \) should be spatially smooth and consistent with observed data, we can generally formulate the problem as

\[
E(f) = E_{data}(f) + \lambda E_{smooth}(f)
\]

Where the first part \( E_{data}(f) = \sum_{p \in P} \phi_p(f_p) \), called data term or unary term, measures the consistency between matching pixels, and the second part, called smooth term, \( E_{smooth}(f) = \sum_{p \in P} \sum_{q \in N_p} E(f_p, f_q) \) penalizes discontinuity between neighboring pixel pairs. To solve such energy minimization problems, many useful discrete optimizers like BP and GC can be directly used. For BP, it minimizes energy function by message propagation on a loopy graph iteratively[10], while for GC, it defines two general classes of interaction penalty \( E(f_p, f_q) \): metric and semi-metric.

if \( \alpha = \beta \), then \( E(\alpha, \beta) = 0 \)

\[
E(\alpha, \beta) = E(\beta, \alpha) \geq 0
\]

\[
E(\alpha, \gamma) \leq E(\alpha, \beta) + E(\beta, \gamma)
\]

For any labels \( \alpha, \beta, \gamma \in L \), if it satisfies all the equation above, then \( E(f_p, f_q) \) can be regarded as metric the and it is semi-metric if \( E(f_p, f_q) \) only satisfies Equation(4) and Equation(5). In expansion moves, only metric interaction allows for \( \alpha \)-expansion [9] and then we can reduce stereo matching problem to a binary-labeling problem, which can be effectively solved by GC.

2.2 Superpixel image

As a key component of many computer vision problems, super-pixels have been widely used in stereo matching, object classification, image parsing and semantic labeling. Considering that super-pixels are generated based on continuous region of pixels having similar image data across channels, super-pixel based image representation has many benefits indeed. First, super-pixels provide a compact representation of images and carry more information than normal grids since they adhere to the nature
image edges. Second, taking super-pixel as a basic unit can avoid some redundant computation as pixels in the same super-pixel are more likely to have the same label and super-pixel based graphs have much fewer nodes than the classic pixel images. Here, we choose the SLIC algorithm[1] to decompose the input image $I$ into $K$ non-overlapping segments, i.e. $S = \{S(k)\}_{k=1}^{K}$ and $\forall m \neq n, S(m) \cap S(n) = \emptyset$. Figure 1(b) and 1(c) shows SLIC super-pixels generated with different size.

Figure 1: (a) the original input left image (b)segmented image in big size (c)segmented image in small size (d)divide superpixels in four group

2.3 Cost computation and aggregation
Unlike global optimization methods, local window-based stereo matching mainly involves three steps: (1) cost volume computation (2) cost-volume filtering and (3) Winner-Takes-All scheme to decide the final solution. For 1D label induced square patches, various matching costs can be applied, with the assumption that disparity values are constant in the small matching window. While for 3D slanted label method, the window sizes are much larger and 1D label matching costs are not suitable here, due to many challenging problems, like illumination change and projection deformation. To solve this problem, Zbontar[21] proposed a CNN model to learn a similarity measure between image patches and such method has shown more robustness and better accuracy in challenging situation. As for cost-volume filtering, given the raw cost volume $C_q(l)$ computed for a certain disparity $l$ at pixel $p$, then the filtered cost value can be computed as $\tilde{C}_q(l) = \sum_q \omega_{pq} C_q(l)$. Here, different edge-aware filtering technologies[14,11,8] can be applied and the difference between them lies primarily in defining $\omega_{pq}$. Yoon[20] performs an adaptive support window technique for cost filtering, which solves boundary issues successfully, but it costs $O(|W|)$ computation. He[7] presents a constant time filtering, which is called guided filter, to do cost-volume filtering, achieving $O(1)$ time computation cost.

3 Proposed Method
3.1 Formulation

With the slanted patch matching method, each pixel \( p \)'s label can be represented as a three-parameter vector \((a_p, b_p, c_p)\) and the disparity \( f_p \) is over-parameterized by \( f_p = a_p x + b_p y + c_p \). Therefore, our goal is to assign each pixel \( p \epsilon P \) a slanted plane that minimizes the energy function below

\[
E(f) = \sum_{p \in P} \phi_p(f_p) + \lambda \sum_{p \in P} \sum_{q \in N_p} E(f_p, f_q)
\]

Based on the disparity \( f_p \), we can get the corresponding pixel in the right image by projection function (here we suppose it has x-axis disparity only)

\[
q' = p - (a_p x + b_p y + c)
\]

As mentioned in section 2.1, the corresponding energy form consists of two parts, the data term and smooth term. Now we will discuss their formulation respectively.

1) Data Term

To compute the cost of assigning pixel \( p \) the label \( f_p \) accurately, we use convolutional neural network (CNN) proposed in [21] to predict the similarity between patches. By using the trained neural network, possible errors caused by illumination changes, slants and image noise can be effectively identified. The network architecture is shown in figure 2, and the output of network is defined to represent the matching cost

\[
C_{CNN}(p, f_p) = -s(P_L(p), P_R(p - f_p))
\]

where \( s(P_L(p), P_R(p - f_p)) \) is the similarity output of the neural network for image patches centered at pixel \( p \) in the left image and corresponding patches centered at pixel \( p - f_p \) in the right image, the minus sign converts the similarity score to matching cost. Given the matching cost, the pixel matching costs in 3D label induced window can be aggregated with adaptive weights

\[
\phi_p(f_p) = \frac{1}{N_p} \sum_{q \epsilon W_p} \omega(p, q) \min(C_{CNN}(q, f_q), \tau_{sim})
\]

Here, \( W_p \) is a square window centered at pixel \( p \) and the weight \( \omega(p, q) \) is defined as

\[
\omega(p, q) = e^{-\|l(p) - l(q)\|_1/\gamma}
\]

Here \( \gamma \) is a user-decided parameter and \( \|\cdot\|_1 \) means \( l1-norm \). \( \tau_{sim} \) is used to increase the robustness for irregular surface that cannot be modeled well by 3D labels. And the normalization factor \( N_p = \sum_{q \epsilon W_p} \omega(p, q) \).
As to smooth term, we use a second-order, curvature-based regularization term \[^{13}\] , which is defined

\[
E(f_p, f_q) = \max(\omega(p, q), \varepsilon) \min(\psi(f_p, f_q), \tau_{dis})
\]

where \(\omega(p, q)\) is a parameter that is defined in Equation (10) and \(\varepsilon\) is a small constant which helps to improve the robustness for image noise. \(\psi(f_p, f_q)\) penalizes the discontinuity between label \(f_p\) and label \(f_q\), which is defined

\[
\psi(f_p, f_q) = |d_p(f_p) - d_p(f_q)| + |d_q(f_p) - d_q(f_q)|
\]

Here, \(d_q(f_p) = a_p x_q + b_p y_q + c_p\) and \(\tau_{dis}\) is used in Equation (12) to allow discontinuity of disparity at depth edges.

3) Sub-modularity

In this part, we will give more detailed information to prove that the term \(\psi(f_p, f_q)\) satisfies the sub-modularity of \(\alpha\)-expansion in Equation (6).

\[
\psi(\alpha, \gamma)
\]

\[
= |d_p(\alpha) - d_p(\gamma)| + |d_q(\alpha) - d_q(\gamma)|
\]

\[
= \left|\left(d_p(\alpha) - d_p(\beta)\right) - \left(d_p(\gamma) - d_p(\beta)\right)\right| + \left|\left(d_q(\alpha) - d_q(\beta)\right) - \left(d_q(\gamma) - d_q(\beta)\right)\right|
\]

\[
\leq \left|\left(d_p(\alpha) - d_p(\beta)\right)\right| + \left|\left(d_p(\gamma) - d_p(\beta)\right)\right| + \left|\left(d_q(\alpha) - d_q(\beta)\right)\right| + \left|\left(d_q(\alpha) - d_q(\beta)\right)\right|
\]

\[
= \psi(\alpha, \beta) + \psi(\beta, \gamma)
\]

Till now, we have proved that \(\psi(f_p, f_q)\) has the sub-modularity. And so does the smooth term.
3.2 Local expansion

1) Local $\alpha$ expansion

In our method, we start by using SLIC algorithm to do super-pixel segmentation on the input images. After segmentation, the image is composed of K super-pixels. Given such a super-pixel based structure, we further define two types of pixels for each local $\alpha$ expansion: the representative pixel and expansion pixels (all pixels in the super-pixel except representative pixel). Then we focus on how expansion works. The expansion process involves two steps: propagation and randomization, now we discuss it in details.

a) Propagation

For super-pixel $S_k$, Firstly, we randomly select a pixel $r$ from $S_k$ as representative pixel and other pixels in $S_k$ are all called expansion pixels. Then taking $r$’s current label $(a_r, b_r, c_r)$ as candidate label, we update all expansion pixels, by choosing the current label or the candidate label. Here, we use Graph cuts $\alpha$ expansion to implement the update process, by minimizing energy function in Equation (7) with binary variables: its current label and the candidate label. We use such local expansion iteratively to get an improved result with lower or equal energy.

b) Randomization

The goal of this step is to reduce the whole energy in super-pixel $S_k$ by refining the label of representative pixel. Considering that such label like $(a_r, b_r, c_r)$ can not sample the space of all possible planes evenly, we need to convert the the original candidate label $(a_r, b_r, c_r)$ to the point plus normal vector representation $(n_x, n_y, n_z, z_0)$. In order to compute a random plane at pixel $r = (x_r, y_r)$, we first pick a random disparity $z_0$ in the range of allowed disparity values and get a point $(x_r, y_r, z_0)$ from the random plane. And then the transformation is used

$$
\begin{align*}
    a_r &= -\frac{n_x}{n_z} \\
    b_r &= -\frac{n_y}{n_z} \\
    c_r &= \frac{n_x x_r + n_y y_r + n_z z_r}{n_z}
\end{align*}
$$

Here, we define two parameters for the perturbation: $\Delta z_0^{\text{max}}$ and $\Delta n^{\text{max}}$, where $\Delta z_0^{\text{max}}$ sets a limit on the allowed change of 3D point’s z coordinate $z_0$ and $\Delta n^{\text{max}}$ defines the maximum change of normal vector $(n_x, n_y, n_z)$. We now iteratively update the candidate label $(n_x, n_y, n_z, z_0)$ by adding a random value in $[-\Delta z_0^{\text{max}}, \Delta z_0^{\text{max}}]$ to $z_0$ and adding three random values in $[-\Delta n^{\text{max}}, \Delta n^{\text{max}}]$ to normal vector $(n_x, n_y, n_z)$. If the corresponding energy is lower the candidate label are updated. And then we iteratively perform local expansion on the basis of new candidate label.

2) Mutually-disjoint group

In this part, we will discuss how to schedule local $\alpha$-expansion properly because these expansions cannot be performed simultaneously due to overlapping regions.

According to the four color theorem[2], we can always divide the super-pixels into four groups so that any two super-pixels in the same group are not adjacent in four-neighbor grid, as shown in figure1(d). This disjoint property ensures the sub-modularity and independency of local expansion moves and thus can be solved by standard GC, which has a obvious computational advantage than QPBO-GC[9]. Besides, due to the independency, local expansion moves in the same group do not interact with each other, therefore, we can perform expansion moves in a parallel manner.

3) Optimization

While the previous section presents local expansion moves, here we summarize the overall PMGC optimization process in Algorithm 1.
**Algorithm 1 Overview of PMGC Optimization**

1: Calculate different size of SLIC segmentation result $M = \{M_1, M_2, \cdots, M_n\}$.
2: Construct four disjoint group for each segmented image
3: Initialize the solution $f$ randomly.
4: Initialize perturbation factor $A_{z_0}^{max}$.
5: while not converged do
6:   for each segmentation result $m \in M$ do
7:      for each disjoint group $k=0,1,2,3$ do
8:         for each super-pixel in disjoint group $k$ do
9:            perform local expansion
10:       end for
11:   end for
12:   \[ A_{z_0}^{max} = A_{z_0}^{max} / 2, A_{n}^{max} = A_{n}^{max} / 2 \]
13: end while
14: post processing

The algorithm begins with SLIC segmentation. In order to alleviate the influence of segmentation size on local expansion moves, we try different super-pixel sizes. Besides, to guarantee the sub-modularity and independency of local expansion, we need to divide these super-pixels based images into four mutually disjoint groups and then these expansion moves can be efficiently done in parallel. At line 3, the initial solution is randomly sampled without other external methods. To sample the solution space evenly, we use the initialization form of $(n_x, n_y, n_z, z_0)$, as described before. And set $A_{z_0}^{max} = \text{maxdisp}/2$.

In the main loop from line 5 to line 14, we select one size of super-pixel based image and apply iterative local expansion moves for each super-pixel.

Finally, after the whole process, we do left-right consistency check and weighted median filtering to further improve the result, such post processing is widely employed in stereo matching field.

### 4 Experiments

In the experiments, our proposed method is evaluated on the Middlebury datasets and a detailed analysis is given. Furthermore, we compare PMGC with other MRF related methods like PMBP[3] and GC-LSL[16], which are closely related to our approach. Since the old Middlebury 2.0 benchmark, which is often tested with four image pairs named ‘Tsukuba’, ‘Cones’, ‘Teddy’ and ‘Venus’ is no longer active, we still need them to compare with other methods like PMBP, so we downloaded the four image pairs and evaluated offline.

#### 4.1 Settings

The following settings are used and kept throughout the experiments. We use a PC equipped with a 3.6GHz dual-core CPU, 8G memory and GTX 1070 graphics card. The parameters are set as \( \{\tau_{sim}, \tau_{dis}, y, \sigma, \lambda\} = \{0.5, 1, 10, 0.9, 20\} \) and the convolutional neural network is trained with all the images of Middlebury datasets. With the ground truth disparity, we extract 11×11 patches from each image pair, one positive example where the center pixels are the same point and one negative example where this is not the case. For slanted matching window, we use a larger size of 41×41 for high resolution images.

#### 4.2 Evaluation

The proposed method is evaluated on Middlebury datasets. Since other method are tested with Middlebury 2.0, here we still use the 2.0 benchmark.
Table 1: Quantitative comparison of methods on the Middlebury datasets

| Algorithm | Tsukuba all | Venus all | Teddy all | Cones all | Avg. error |
|-----------|-------------|-----------|-----------|-----------|------------|
| PMBP[3]   | 2.21        | 0.49      | 8.57      | 6.64      | 4.46       |
| PMF[12]   | 2.04        | 0.49      | 5.87      | 6.80      | 4.06       |
| PM[4]     | 2.33        | 0.39      | 8.16      | 7.80      | 4.59       |
| GC-LSL[16]| 2.73        | 0.36      | 3.77      | 7.37      | 4.19       |
| Ours      | 2.15        | 0.26      | 2.92      | 6.31      | 3.92       |

We compare our methods with other MRF model related methods in table 1. It's obvious that our method shows higher accuracy and gets best results for all tested image pairs.

Figure 3: The results of disparity map for proposed method, the datasets are Tsukuba, Venus, Cones and Teddy.

Compared with BP, one-by-one sequential optimizer, GC in contrast can update multiple pixels simultaneously, which has a fundamental advantage for global optimization. Therefore, GC-LSL shows better performance than PMBP.
Our method is most close to GC-LSL, but we use CNN for data term calculation and change normal grid to super-pixel, integrating super-pixel information with local expansion, which further improves the result.

5 Conclusion
In this paper, we propose a new algorithm to efficiently solve continuous MRF stereo model, integrating patchmatch into GC optimization. Besides, we use CNN to construct data term for more precise matching cost computation. As an efficient and accurate optimizer, our method outperforms the existing MRF related methods like PMBP and GC-LSL.

References
[1] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P. and Süsstrunk, S., 2012. SLIC superpixels compared to state-of-the-art superpixel methods. IEEE transactions on pattern analysis and machine intelligence, 34(11), pp.2274-2282.
[2] Allaire, F., 1977. Another Proof of the Four Colour Theorem. University of Calgary, Department of Mathematics and Statistics.
[3] Besse, F., Rother, C., Fitzgibbon, A. and Kautz, J., 2014. Pmbp: Patchmatch belief propagation for correspondence field estimation. International Journal of Computer Vision, 110(1), pp.2-13
[4] Bleyer, M., Rhemann, C. and Rother, C., 2011, August. PatchMatch Stereo-Stereo Matching with Slanted Support Windows. In Bmvc (Vol. 11, pp. 1-11).
[5] Boykov, Y., Veksler, O. and Zabih, R., 2001. Fast approximate energy minimization via graph cuts. IEEE Transactions on pattern analysis and machine intelligence, 23(11), pp.1222-1239.
[6] Heise, P., Klose, S., Jensen, B. and Knoll, A., 2013. Pm-huber: Patchmatch with huber regularization for stereo matching. In Proceedings of the IEEE International Conference on Computer Vision (pp. 2360-2367).
[7] He, K., Sun, J. and Tang, X., 2010, September. Guided image filtering. In European conference on computer vision (pp. 1-14). Springer Berlin Heidelberg.
[8] Hosni, A., Rhemann, C., Bleyer, M., Rother, C. and Gelautz, M., 2013. Fast cost-volume filtering for visual correspondence and beyond. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(2), pp.504-511.
[9] Lempitsky, V., Rother, C., Roth, S. and Blake, A., 2010. Fusion moves for markov random field optimization. IEEE transactions on pattern analysis and machine intelligence, 32(8), pp.1392-1405.
[10] Li, Y., Min, D., Brown, M.S., Do, M.N. and Lu, J., 2015. Spm-bp: Sped-up patchmatch belief propagation for continuous mrf. In Proceedings of the IEEE International Conference on Computer Vision (pp. 4006-4014).
[11] Lu, J., Shi, K., Min, D., Lin, L. and Do, M.N., 2012, June. Cross-based local multipoint filtering. In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on (pp. 430-437). IEEE
[12] Lu, J., Yang, H., Min, D. and Do, M.N., 2013. Patch match filter: Efficient edge-aware filtering meets randomized search for fast correspondence field estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1854-1861).
[13] Olsson, C., Ulén, J. and Boykov, Y., 2013. In defense of 3d-label stereo. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1730-1737).
[14] Paris, S., Komprubst, P., Tumblin, J. and Durand, F., 2009. Bilateral filtering: Theory and applications. Foundations and Trends® in Computer Graphics and Vision, 4(1), pp.1-73.
[15] Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M. and Rother, C., 2008. A comparative study of energy minimization methods for markov random
fields with smoothness-based priors. *IEEE transactions on pattern analysis and machine intelligence*, 30(6), pp.1068-1080.

[16] Taniai, Tatsunori, Yasuyuki Matsushita, and Takeshi Naemura. "Graph cut based continuous stereo matching using locally shared labels." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014.

[17] Woodford, O., Torr, P., Reid, I. and Fitzgibbon, A., 2009. Global stereo reconstruction under second-order smoothness priors. *IEEE transactions on pattern analysis and machine intelligence*, 31(12), pp.2115-2128.

[18] Xu, S., Zhang, F., He, X., Shen, X. and Zhang, X., 2015. PM-PM: PatchMatch with Potts model for object segmentation and stereo matching. *IEEE Transactions on Image Processing*, 24(7), pp.2182-2196.

[19] Yamaguchi, K., Hazan, T., McAllester, D. and Urtasun, R., 2012, October. Continuous markov random fields for robust stereo estimation. In *European Conference on Computer Vision* (pp. 45-58). Springer Berlin Heidelberg.

[20] Yoon, K.J. and Kweon, I.S., 2005, June. Locally adaptive support-weight approach for visual correspondence search. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on* (Vol. 2, pp. 924-931). IEEE.

[21] Zbontar, J. and LeCun, Y., 2016. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, 17(1-32), p.2.