AN EFFICIENT LOCAL METHOD FOR STEREO MATCHING USING DAISY FEATURES

Xiaoming Peng[1], *Abdesselam Bouzerdoum*[1,2], *Son Lam Phung*[1]

[1]School of Electrical, Computer and Telecommunications Engineering
The University of Wollongong, Wollongong, Australia
[2]College of Science and Engineering, Hamad Bin Khalifa University, Doha, Qatar

ABSTRACT

In this paper, a local method is proposed to estimate the visibility and disparity of pixels from a stereo pair using the DAISY feature. The problem is formulated as a joint optimization over disparity and visibility of individual pixels. The constraints on the range of disparities and the binary visibility variables are enforced by incorporating penalty terms into the cost function. Finally, the unconstrained optimization problem is solved using a Newton scheme with appropriate approximations to the Hessian matrices and gradients. The computation time of the proposed optimization method is around one minute to run for 768×512 stereo pairs using the DAISY feature descriptor in a C++ implementation.

***Index Terms*—** stereo matching, local optimization, the DAISY feature vector

1. INTRODUCTION

Depth information is very useful in computer vision tasks such as scene reconstruction and virtual touring. In this paper, we focus on the problem of depth/visibility estimation from a calibrated stereo pair, which entails finding a dense correspondence map between the two images. Generally, stereo matching methods can be roughly categorized into local (i.e., window-based) and global approaches. Stereo matching involves mostly the following four steps [1]: (1) matching cost computation, (2) matching cost aggregation, (3) disparity computation or optimization, and (4) disparity refinement. Both local and global approaches need to perform the matching cost computation step, but they differ in the treatment of smoothness constraints. Local methods make implicit smoothness assumptions by aggregating costs within a finite window. By contrast, global approaches make explicit smoothness assumptions by combining the data and smoothness terms into a single cost function, which is subsequently optimized using an iterative procedure. The most commonly used global optimization methods are energy minimization methods [2]; expectation-maximum (EM) [3] and cooperative optimization [4] are also used for this purpose.

Due to their speed and ease of implementation, local cost aggregation approaches are usually preferred in stereo matching applications over their global counterparts. For short-baseline stereo matching cases, existing methods based on pixel intensity values, such as the sum-of-absolute-differences (SAD) [5], the truncated-absolute-differences (TAD) [6], and mutual information [7], have almost reached maturity. However, these similarity measures lack robustness to large perspective distortions. Tola et al. showed that their DAISY feature descriptor outperforms pixel-difference similarity measures in the case of wide-baseline stereo matching [8]. However, since cost aggregation constitutes the major computational burden in local methods, the direct replacement of pixel-intensity-based matching costs with feature-vector-based ones can incur prohibitively high computational expense. Existing cost aggretation methods, such as bilateral filtering [9, 10], approximate weighting [6], guided filtering [11, 12], tree-based cost aggregation methods [13, 14], and the unified optimization framework presented in [15], are not readily extendable to the case of feature-vector-based similarity matching. For example, the computational expense of the similarity kernels in the bilateral filtering methods and the tree-based methods will be very high if feature-vector-based similarity measures are used. Zhang et al. proposed to pair binary masks with the BRIEF feature descriptor [16] to accelerate the cost aggregation [17]. However, their binary masks can only be paired with binary feature vectors; they are not applicable to general real-valued feature vectors. Very recently, deep learning has been used to compare image pairs for stereo matching [18]; however, deep learning relies on the availability of a large pool of annotated image pairs to learn a mapping between them.

In this paper, we propose a local optimization method for stereo matching that possesses the following three characteristics. First, the problem of local matching is formulated as a constrained optimization problem, in which the smoothness terms are incorporated explicitly into the cost function. The constrained optimization problem is then transformed into an unconstrained one. Second, similar to [3, 8], the proposed method combines both the disparity and the visibility estimation into a unified framework. As shown in Section 2.2, the visibility estimates are initialized using the quality of matches between pixels. By contrast, most local methods also utilize this quality, but in an implicit way—they remove inconsistent disparity estimates caused by poor matching pixels in the left-right and right-left cross checking procedure. Finally, an efficient Newton-type algorithm is proposed for solving the unified optimization problem, which requires only the inversion of a diagonal matrix approximating the Hessian matrix. The developed algorithm is much more efficient than global optimization methods, such as EM [3] and graph-cuts (GC) [8], and does not require intensive training as in deep learning.

2. LOCAL OPTIMIZATION METHOD FOR STEREO MATCHING USING THE DAISY FEATURE

Assuming we have a pair of *rectified* stereo images I_l and I_r such that a pixel $p = (x, y)$ in I_l has a counterpart $q = (x + d, y)$ in I_r, where d denotes the disparity. In addition to the disparity d, for each pixel p we aim to estimate its associated visibility $v(p) \in \{0, 1\}$: when $v(p) = 0$, the pixel p is occluded, otherwise, it is visible to the viewer. Let us assume that we can compute two feature descriptors $\mathbf{f}_l(p)$ and $\mathbf{f}_r(q)$ at p and q in I_l and I_r, respectively. In this paper, we use the DAISY feature descriptor, although other feature descritors could also be used.

The DAISY descriptor gets its name from its flower-like shape: the "flower" center coincides with the pixel location where the feature descriptor is being computed. The flower consists of Q rings, each containing T circles ("petals"). The flower center and each petal are described by a histogram of length H, which is the convolved orientation map computed at the flower center or a petal. Thus, a DAISY descriptor $\mathbf{f}$ contains $H \times (Q \times T + 1)$ elements; here we use $Q = 3, T = 8$, and $H = 8$, which yields a vector $\mathbf{f}$ with 200 elements.

In our case, we use $\mathbf{f}_l^k(p)$ and $\mathbf{f}_r^k(q)$ to denote the kth ℓ_2-normalized histogram of $\mathbf{f}_l(p)$ and $\mathbf{f}_r(q)$, respectively ($k = 1, 2, \ldots, K$, where $K = Q \times T + 1 = 25$). The dissimilarity between two feature vectors is measured as $\sum_k w_k^2 \|\mathbf{f}_l^k(p) - \mathbf{f}_r^k(q)\|_2^2$, where w_k is the visibility of $\mathbf{f}_l^k(p)$. The visibility w_k is bi-linearly interpolated from the visibilities of its four nearest integer neighbors, such that $w_k \in [0, 1]$. If $w_k = 0$, the kth petal of the DAISY flower in image I_l is occluded and hence makes no contribution to the dissimilarity measure. Let d_i and v_i denote the disparity and visibility of pixel $p_i \in I_l$ ($i = 1, 2, \ldots, n_l$, where n_l is the total number of pixels in I_l). Denoting $\mathbf{d} = (d_1, d_2, \ldots, d_{n_l})^T$ and $\mathbf{v} = (v_1, v_2, \ldots, v_{n_l})^T$, and assuming that the range of each element of $\mathbf{d}$ is $[d_{min}, d_{max}]$, we formulate the following constrained optimization problem:

$$\min_{\mathbf{d},\mathbf{v}} f(\mathbf{d}, \mathbf{v}), \tag{1}$$

$$\text{subject to} \begin{cases} d_i \in [d_{min}, d_{max}] \; (1a) \\ v_i \in \{0, 1\} \qquad\quad (1b) \end{cases} \quad \forall p_i \in I_l.$$

The cost function $f(\mathbf{d}, \mathbf{v})$ is defined in Eq. (2) given below, where c_o, c_d and c_v are three predefined constants, β_d and β_v are two weights, $\mathcal{N}_i$ is the 4-neighborhood of pixel $p_i \in I_l$.

$$\begin{aligned} f(\mathbf{d}, \mathbf{v}) = & \sum_{i=1}^{n_l} \sum_{k=1}^{K} \frac{1}{2} w_{i,k}^2 \|\mathbf{f}_l^k(p_i) - \mathbf{f}_r^k(q_i)\|_2^2 \\ & + \sum_{i=1}^{n_l} \frac{1}{2} (v_i - 1)^2 c_o \\ & + \beta_d \sum_{i=1}^{n_l} \frac{1}{4} \sum_{p_j \in \mathcal{N}_i} \min\Big(\frac{(d_i - d_j)^2}{2}, c_d\Big) \\ & + \beta_v \sum_{i=1}^{n_l} \frac{1}{4} \sum_{p_j \in \mathcal{N}_i} \min\Big(\frac{(v_i - v_j)^2}{2}, c_v\Big). \end{aligned} \tag{2}$$

To simplify the following discussion, we use f_1, f_2, f_3 and f_4 to denote the four terms in the above equation. The term f_1 is the data fidelity term. The second term f_2 is the occlusion term, which can be regarded as a regularization term; it prevents f_1 reaching zero due to the occlusion of all the pixels in I_l. The last two terms, f_3 and f_4, are smoothness terms, which aim to preserve discontinuities across object boundaries in the scene. Note that in the formulation of the above optimization problem, we assume that each pixel $p_i \in I_l$ is independent of other pixels in the same image, a commonly used assumption in the depth estimation literature.

The nonlinear constrained optimization problem in Eq. (1) is hard to solve because $\mathbf{v}$ contains only binary variables. As shown by Cela [19], even for the simplest case with a quadratic cost function and linear constraints, the problem is NP-hard. For this reason, we relax the constraints by incorporating penalty terms into the cost function, thereby transforming the constrained optimization problem into an unconstrained one. We use P_1 for constraint (1a), and P_2 and P_3 for constraint (1b), defined as follows:

$$P_1(d_i) = \begin{cases} \frac{(d_i - d_{min})^2}{2}, & \text{if } d_i < d_{min} \\ 0, & \text{if } d_i \in [d_{min}, d_{max}] \\ \frac{(d_i - d_{max})^2}{2}, & \text{if } d_i > d_{max} \end{cases}, \tag{3}$$

$$P_2(v_i) = \begin{cases} \frac{v_i^2}{2}, & \text{if } v_i < 0 \\ 0, & \text{if } v_i \in [0, 1] \\ \frac{(v_i - 1)^2}{2}, & \text{if } v_i > 1 \end{cases}, P_3(v_i) = v_i^2 (v_i - 1)^2.$$

The term P_1 punishes those disparity estimations that are outside the range $[d_{min}, d_{max}]$, P_2 penalizes those visibility estimations that are outside the interval $[0, 1]$, and P_3 punishes those visibility values that do not take on values of zero or one. Now the cost function $f(\mathbf{d}, \mathbf{v})$ for the reformulated unconstrained optimization problem is expanded as follows:

$$f(\mathbf{d}, \mathbf{v}) = f_1 + f_2 + f_3 + f_4 + f_5 + f_6, \tag{4}$$

where $f_5 = \gamma_d \sum_{i=1}^{n_l} P_1(d_i)$, and $f_6 = \gamma_v \sum_{i=1}^{n_l} (P_2(v_i) + P_3(v_i))$, with γ_d and γ_v two extra weights to take into account the influences of the respective penalties.

2.1. The Optimization Approach

An inspection of the terms in (4) shows that except for f_1, whose arguments include both $\mathbf{d}$ and $\mathbf{v}$, $(f_3 + f_5)$ and $(f_2 + f_4 + f_6)$ depend only on either $\mathbf{d}$ or $\mathbf{v}$. Because $f(\mathbf{d}, \mathbf{v})$ is much more sensitive to changes in $\mathbf{v}$ compared to changes in $\mathbf{d}$, the above optimization problem is not well scaled [20]. For this reason, we propose to optimize $f(\mathbf{d}, \mathbf{v})$ alternately over $\mathbf{d}$ and $\mathbf{v}$ instead of a joint optimization. Specifically, we define two sub-functions, $F(\mathbf{d}, \mathbf{v}) = f_1(\mathbf{d}, \mathbf{v}) + f_3(\mathbf{d}) + f_5(\mathbf{d})$ and $G(\mathbf{d}, \mathbf{v}) = f_1(\mathbf{d}, \mathbf{v}) + f_2(\mathbf{v}) + f_4(\mathbf{v}) + f_6(\mathbf{v})$. For a fixed $\mathbf{v}$, we have the first subproblem, $\min_{\mathbf{d}} F(\mathbf{d}, \mathbf{v})$, and for a fixed $\mathbf{d}$, we have the second subproblem, $\min_{\mathbf{v}} G(\mathbf{d}, \mathbf{v})$. We use a Newton scheme to solve both subproblems. For example, for the first subproblem we iterate the following step until convergence: $\mathbf{d}_n = \mathbf{d}_{n-1} + s\mathbf{p}_{\mathbf{d}}^{n-1}$, where s is the appropriate step length that satisfies the "Armijo condition" [20], and $\mathbf{p}_{\mathbf{d}}^{n-1} = -\left(\nabla^2 F(\mathbf{d}_{n-1}, \mathbf{v}_{n-1})\right)^{-1} \nabla F(\mathbf{d}_{n-1}, \mathbf{v}_{n-1})$, where ∇ is the gradient operator and $\nabla^2 F$ is the Hessian matrix of F. The initial values of $\mathbf{d}$ and $\mathbf{v}$ are estimated in Section 2.2. Below we briefly derive the Hessian matrices and gradients involved in f_1; the related computations for the other terms are more straightforward and are thus omitted due to space limitation. Let us denote:

$$\begin{aligned} e_{i,k} &= \frac{1}{2} w_{i,k}^2 \|\mathbf{f}_l^k(x_i, y_i) - \mathbf{f}_r^k(x_i + d_i, y_i)\|_2^2 \\ &= \frac{1}{2} w_{i,k}^2 \mathbf{v}_{i,k}^T \mathbf{v}_{i,k}, \end{aligned} \tag{5}$$

where $\mathbf{v}_{i,k} = \mathbf{f}_l^k(x_i, y_i) - \mathbf{f}_r^k(x_i + d_i, y_i)$, the derivative of $e_{i,k}$ with respect to d_i is expressed as

$$\frac{\partial e_{i,k}}{\partial d_i} = w_{i,k}^2 \frac{\partial \mathbf{v}_{i,k}}{\partial d_i} \mathbf{v}_{i,k} = -w_{i,k}^2 \frac{\partial \mathbf{f}_r^k(x_i + d_i, y_i)}{\partial d_i} \mathbf{v}_{i,k}. \tag{6}$$

We use $\mathbf{f}_r^k(x_i + d_i + 1, y_i)^T - \mathbf{f}_r^k(x_i + d_i, y_i)^T$ to approximate $\frac{\partial \mathbf{f}_r^k(x_i + d_i, y_i)}{\partial d_i}$ in the above equation. Similarly, the second derivative $\frac{\partial^2 e_{i,k}}{\partial d_i^2}$ can be computed as follows:

$$\begin{aligned} \frac{\partial^2 e_{i,k}}{\partial d_i^2} &= w_{i,k}^2 \left[(\frac{\partial \mathbf{v}_{i,k}}{\partial d_i})(\frac{\partial \mathbf{v}_{i,k}}{\partial d_i})^T + \frac{\partial^2 \mathbf{v}_{i,k}}{\partial d_i^2} \mathbf{v}_{i,k} \right] \\ &\approx w_{i,k}^2 (\frac{\partial \mathbf{v}_{i,k}}{\partial d_i})(\frac{\partial \mathbf{v}_{i,k}}{\partial d_i})^T. \end{aligned} \tag{7}$$

As in Gauss-Newton and Levenberg-Marquardt methods [20], we ignore $\frac{\partial^2 \mathbf{v}_{i,k}}{\partial d_i^2} \mathbf{v}_{i,k}$ because $(\frac{\partial \mathbf{v}_{i,k}}{\partial d_i})(\frac{\partial \mathbf{v}_{i,k}}{\partial d_i})^T$ is often more dominant. More importantly, with the approximation of $\frac{\partial^2 e_{i,k}}{\partial d_i^2}$ and the assumption that the d_i's are independent, $\frac{\partial^2 f_1}{\partial \mathbf{d}^2}$ is a diagonal matrix with $\sum_{k=1}^K \frac{\partial^2 e_{i,k}}{\partial d_i^2}$ being the ith diagonal element. Since $(\frac{\partial \mathbf{v}_{i,k}}{\partial d_i})(\frac{\partial \mathbf{v}_{i,k}}{\partial d_i})^T \geq 0$, we can almost always guarantee the positive-definiteness of $\frac{\partial^2 f_1}{\partial \mathbf{d}^2}$. Note that only when $\sum_{k=1}^K \frac{\partial^2 e_{i,k}}{\partial d_i^2}$ is zero for some pixel $p_i \in I_l$, will $\frac{\partial^2 f_1}{\partial \mathbf{d}^2}$ become positive semidefinite, which is rare for natural images. The visibility term $w_{i,k}$ is obtained with bilinear interpolation:

$$w_{i,k} = \sum_{j=1}^4 \rho_{k,j} v_j, \tag{8}$$

where v_j $(j = 1, \ldots, 4)$ is the visibility of its jth neighbor and $\rho_{k,j}$ is its interpolation coefficient. From Eq.(8), we have $\frac{\partial e_{i,k}}{\partial v_j} = \rho_{k,j} w_{i,k} \|\mathbf{v}_{i,k}\|_2^2$, and $\frac{\partial^2 e_{i,k}}{\partial v_j^2} = \rho_{k,j}^2 \|\mathbf{v}_{i,k}\|_2^2$. Similar to $\frac{\partial^2 f_1}{\partial \mathbf{d}^2}$, $\frac{\partial^2 f_1}{\partial \mathbf{v}^2}$ is a diagonal matrix with $\sum_{k \in \mathcal{K}_i} \frac{\partial^2 e_{i,k}}{\partial v_i^2}$ its ith diagonal element, where $\mathcal{K}_i$ is the set of DAISY petals that involve v_i.

2.2. Determining the Free Parameters and Initial Values

We use a stereo pair with ground truth to determine the four weights in Eq. (4). Specifically, we vary the values of the four parameters to inspect the quality of depth estimation (obtained using camera calibration parameters) for each combination, using the *average squared normalized error (ASNE)* measure defined as $\frac{\sum_i (d_{i,e} - d_{i,g})^2 / d_{i,g}^2}{M}$, where $d_{i,e}$ and $d_{i,g}$ are the estimated and ground truth depth values of the ith correctly estimated visible pixel, and M is the total number of correctly estimated visible pixels. The best combination of parameters found is $\beta_d = \beta_v = 5 \times 10^{-3}$, and $\gamma_d = \gamma_v = 5 \times 10^{-2}$. These parameters have proven to work very well in all the experiments presented in this paper.

To determine the initial estimates of $\mathbf{d}$ and $\mathbf{v}$, we establish an initial correspondence between the pixels of the stereo pair by first dividing I_l and I_r into small blocks (2×2 blocks) and then horizontally matching the blocks mutually to estimate the initial disparity values. All the pixels in the small blocks share the same disparity and visibility values. The dissimilarity between two blocks $\mathcal{B}_l$ and $\mathcal{B}_r$ (where $\mathcal{B}_l \in I_l$ and $\mathcal{B}_r \in I_r$) is defined as $\delta(\mathcal{B}_l, \mathcal{B}_r) = \frac{1}{4} \sum_{i=1}^4 \frac{1}{K} \sum_{k=1}^K \|\mathbf{f}_{l,i}^k - \mathbf{f}_{r,i}^k\|_2$, where $\mathbf{f}_{l,i}$ and $\mathbf{f}_{r,i}$ $(i = 1, \ldots, 4)$ are the DAISY features of the ith corresponding pixels in $\mathcal{B}_l$ and $\mathcal{B}_r$, and the superscript k denotes their kth histograms. Note that the range of $\delta(\mathcal{B}_l, \mathcal{B}_r)$ is $[0, 2]$. We set the visibility of all the pixels in $\mathcal{B}_l$ to be $(1 - \delta(\mathcal{B}_l, \mathcal{B}_r))_+$, where $(a)_+ = max(a, 0)$. According to Strecha et al.'s generative imaging model [3], the visible pixels in a stereo pair are generated by an inlier process which results in good matches between the pixels. Thus, a small $\delta(\mathcal{B}_l, \mathcal{B}_r)$ indicates a good match between the two blocks, and therefore a high visibility of their constituent pixels.

3. EXPERIMENTAL RESULTS

We implemented the proposed method and three other methods for comparison: an approximate cost aggregation reduction strategy [6] (denoted as "CA"), a GC-based optimization method [21], and SIFT flow [22]. Except for

Table 1. The average values of r_1, r_2 and runtime of the four methods on all the 60 stereo image pairs.

	Fountain-A r_1 (r_2)	Fountain-B r_1 (r_2)	HerzJesu r_1 (r_2)	Time (sec)
Proposed	90.7 (86.9)	87.9 (83.3)	91.9 (85.8)	61.4
CA	85.3 (82.8)	81.3 (77.7)	87.3 (81.9)	207.8
GC	87.0 (79.5)	85.0 (75.5)	89.3 (80.4)	409
SIFT flow	73.8 (73.7)	73.8 (70.0)	37.2 (69.7)	24.9

SIFT flow, which works directly on unrectified stereo pairs, all the other methods work on rectified stereo pairs using a 200 long DAISY feature. Because most stereo pairs in the Middlebury Stereo Evaluation Dataset [1] are captured with a short baseline, in our experiments we test stereo pairs with a wide baseline from a publicly available dataset (http://cvlab.epfl.ch/software/daisy). The dataset also provides ground truth of depth and occlusion maps. Specifically, we chose two data sets: the "Fountain" data set and the "HerzJesu" data set. Both datasets contain 768 × 512 gray-scale images along with camera calibration parameters. Though the DAISY descriptor is promising for wide-baseline stereo matching, Grootendorst [23] showed that the error still increases significantly as the baseline increases. For this reason, we divide the "Fountain" data set into two groups, denoted as Fountain-A and Fountain-B, with each group containing five consecutive images. For the "HerzJesu" dataset, we chose five images to form a group. For each group, each image is alternately used as the left and right image. Thus we have 20 stereo pairs per group, or 60 stereo pairs in the testdata.

Let r_1 be the ratio of the number of correctly classified visible pixels with *correct* depth estimation to the number of correctly classified visible pixels. If for a pixel $p_i \in I_l$ ($i = 1, 2, \ldots, M$), $|d_{i,e} - d_{i,g}|/d_{i,g} \leq 0.05$, we judge its depth to be correctly estimated. Let r_2 denote the ratio of correctly classified visible/invisible pixels to the total number of pixels, i.e., visibility classification accuracy. The performance of a method is measured using three criteria: the percentage of correctly estimated visible pixels, r_1 (%), the visibility classification accuracy, r_2 (%), and the runtime (s). Table 1 presents the average values of r_1, r_2, and the runtime of the four methods on the testdata. The proposed method yields the best depth estimation and visibility classification accuracy, and it is only slower than "SIFT flow", but much faster than "GC" and "CA". Figure 1 shows sample depth/visibility estimation results of the four methods. The color coding scheme used to display the images is as follows: i) pixels with missclassified visibilities are shown in red; (ii) correctly classified invisible pixels are shown in green; (iii) correctly classified visible pixels with *correct* depth estimation are shown in gray scale; (iv) the remaining pixels are shown in blue, which indicate correctly classified visible pixels with incorrect depth estimation. It can be seen from the figure that the proposed method outperforms the other three methods, with more gray-scale and green pixels recovered.

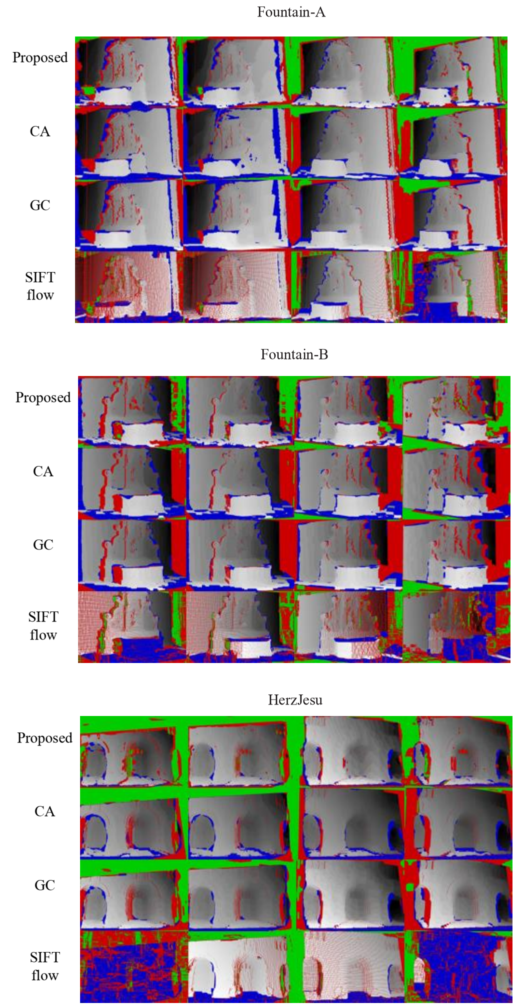

Fig. 1. Sample depth/visibility estimation results of the four methods on the testdata. For each dataset, we show the results of using image 3 as the left image and the other images as the right image. The results of a method are listed in a row.

4. CONCLUSION

This paper proposed a local optimization method to estimate pixel depth and visibility for stereo matching using the DAISY feature. The proposed method employs an efficient Newton type algorithm which requires only the inversion of a diagonal matrix as an approximation to the inverse Hessian matrix. Experimental results on a total of 60 wide-baseline stereo pairs show that our method outperforms three other benchmark methods in accuracy, and it is only slower than the SIFT flow in terms of computation time.

5. REFERENCES

[1] D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” *International Journal of Computer Vision*, vol. 47, no. 1/2/3, pp. 7–42, 2002.

[2] R. Szeliski, R. Zabih, D. Scharstein, and et. al., “A comparative study of energy minimization methods for markov random fields with smoothness-based priors,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 6, pp. 1068–1080, 2008.

[3] C. Strecha, R. Fransens, and L. Van Gool, “Combined depth and outlier estimation in multi-view stereo,” in *IEEE CVPR*, 2006, pp. 2394–2401.

[4] Z. Wang and Z. Zheng, “A region based stereo matching algorithm using cooperative optimization,” in *IEEE CVPR*, 2008, p. 10.1109/CVPR.2008.4587456.

[5] L. D. Stefano, M. Marchionni, and S. Mattoccia, “A fast area-based stereo matching algorithm,” *Image and Vision Computing*, vol. 22, pp. 983–1005, 2004.

[6] D. Min, J. Lu, and M. Do, “A revisit to cost aggregation in stereo matching: how far can we reduce its computational redundancy?,” in *IEEE ICCV*, 2011, pp. 1567–1574.

[7] H. Hirschmuller, “Stereo processing by semiglobal matching and mutual information,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328–341, 2008.

[8] E. Tola, V. Lepetit, and P. Fua, “Daisy: an efficient dense descriptor applied to wide-baseline stereo,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 815–830, 2010.

[9] K. J. Yoon and I. S. Kweon, “Adaptive support-weight approach for correspondence search,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 650–656, 2006.

[10] S. Mattoccia, S. Giardino, and A. Gambini, “Accurate and efficient cost aggregation strategy for stereo correspondence based on approximated joint bilateral filtering,” in *ACCV*, 2009, pp. 371–380.

[11] A. Hosni, C. Rhemann, M. Bleyer, and et. al., “Fast cost-volume filtering for visual correspondence and beyond,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 2, pp. 504–511, 2013.

[12] K. He, J. Sun, and X. Tang, “Guided image filtering,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, pp. 1397–1409, 2013.

[13] Q. Yang, “A non-local cost aggregation method for stereo matching,” in *IEEE CVPR*, 2012, pp. 1402–1409.

[14] X. Mei, X. Sun, W. Dong, and et al., “Segment-tree based cost aggregation for stereo matching,” in *IEEE CVPR*, 2013, pp. 313–320.

[15] K. Zhang, Y. Fang, D. Min, and et al., “Cross-scale cost aggregation for stereo matching,” in *IEEE CVPR*, 2014, pp. 1590–1597.

[16] M. Calonder, V. Lepetit, M. Ozuysal, and et al., “Brief: computing a local binary descriptor very fast,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1281–1298, 2012.

[17] K. Zhang, J. Li, Y. Li, and et al., “Binary stereo matching,” in *ICPR*, 2012, pp. 356–359.

[18] S. Zagoruyko and N. Komodakis, “Learning to compare image patches via convolutional neural networks,” in *IEEE CVPR*, 2015, pp. 4353–4361.

[19] E. Cela, *The Quadratic Assignment Problem: Theory and Algorithms*, Kluwer Academic, Dordrecht, 1998.

[20] J. Nocedal and S. J. Wright, *Numerical Optimization (2nd Edition)*, Springer Verlag, New York, 2006.

[21] Y. Boykov and V. Kolmogorov, “An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1124–1137, 2004.

[22] C. Liu, J. Yuen, and A. Torralba, “Sift flow: dense correspondence across scenes and its applications,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 978–994, 2011.

[23] D. Grootendorst, “Performance of wide-baseline matching using daisy,” *Technical Report*, 2011.