A Rotation and Scale Invariant Approach for Dense Wide Baseline Matching

Jian Gao and Fanhuai Shi[*]

Department of Control Science & Engineering, Tongji University, China
fhshi@tongji.edu.cn

Abstract. This paper proposes a new approach for dense matching of uncalibrated image pair with significant rotation and scale changes. In this approach, a modified region-based matching algorithm is combined with local invariant features like SIFT to conduct dense and reliable matching. First, sparse key point correspondences are established as reference matches; then, in dense matching step, the shape and location of support windows are normalized using SIFT structure information of those reference matches. Thus, scale and rotation changes of input images can be well handled. Experimental results from real data demonstrate that our approach can establish dense and accurate matching in wide-baseline case, which is robust to geometric transformations such as change in scale and rotation, as well as some extent of viewpoint change.

Keywords: dense matching, wide baseline, rotation and scale invariant, SIFT.

1 Introduction

Finding dense and reliable match is a challenging task in computer vision, especially when the input images have significant rotation and scale changes. Dense and accurate matching plays an essential role in many applications such as 3D reconstruction, image-based modeling and image mosaicing. Many researches have been conducted on this topic, but dense and reliable matching of two wide baseline images taken with uncalibrated cameras still remains a tough problem.

Traditional dense two-frame matching algorithms aim at computing disparity maps from image pairs with short baseline [1]. In this situation, the images are almost identical, and the search space of disparities can be reduced to one dimension. These algorithms can be classified into two classes: global and local. In the local framework, dense matching is obtained by searching for correspondences pixel by pixel, thus the similarity measure of candidate pairs is quite important. In the window-based methods, matching cost is the aggregation of pixel-level dissimilarity within a support window; they assume that pixels in the matching window have similar disparities [2].

[*] Corresponding author.

D.-S. Huang et al. (Eds.): ICIC 2014, LNCS 8588, pp. 345–356, 2014.

When the baseline is wide, for example, with the images obtained by a handheld camera, dense matching becomes quite tough. In this situation, images may be quite different, so fixed window becomes ineffective to capture the same image content.

Several approaches have been proposed to handle the wide-baseline case. Plane-sweep-based algorithms test a family of plane hypotheses and choose the best one for each pixel [3]. These approaches can simultaneously perform dense matching and compute the depth map, but the cameras are required to be fully calibrated. In the uncalibrated case, quasi-dense approaches [4, 5] based on match propagation were proposed. In these methods, new matches are propagated from reliable matches obtained in the last iteration. The idea of propagation is to use reliable match at one pixel to guide new matches of its nearby pixels. However, the denseness cannot be guaranteed, which largely depends on the image content and propagation strategies.

In this paper, we propose a new approach for dense matching. Inspired by the great success of SIFT-based descriptors [6], which was reported to have best performance [7], we combine it with the window-based method in short-baseline matching to handle the problem of rotation and scale changes. Our framework contains the following steps: First, local invariant features are used for sparse matching. Second, the support windows are normalized based on reliable sparse matching result. Then, matching score of the warped patches is calculated. Finally, the best correspondence is obtained by searching for the highest matching score, and dense matching result is achieved. Sampson error from known homography is used for evaluation.

The main contribution of the proposed method is to use normalized support windows to deal with the problem of rotation and scale changes in wide baseline matching. Our approach can establish dense and reliable pixel correspondences.

This paper is organized as follows: In section 2, our novel dense matching scheme is presented. Establishment of reliable sparse key point matching is also introduced. In section 3, the detail of dense matching is described, including support window normalization, cost aggregation and searching strategy. Some experimental results on real data are presented in section 4. Finally, this paper is concluded in section 5.

2 Algorithm Framework and Sparse Matching

In this section, we first give a brief overview of the proposed method, and then discuss the initial sparse matching step.

2.1 Framework

The proposed dense matching approach starts with the detection of sparse key points, such as SIFT key-points and image corners, which are invariant to rotation and scale changes. These robust matches are used as reference points to guide the following dense matching. In the window-based method, correspondences are established pixel by pixel. For each pixel, we assign one nearest SIFT match and several corner matches to guide the matching. Firstly, its local geometric transformation from one image to the other is estimated based on these reference matches. Secondly, its hypothesis corresponding location is directly set using this transformation. Then, pixels near this location are searched. Finally, the one with highest matching score is accepted. To deal with the rotation and scale changes, the support windows are normalized, so that it is possible to capture the same image content during searching. The flowchart of the entire algorithm is presented in Figure 1.

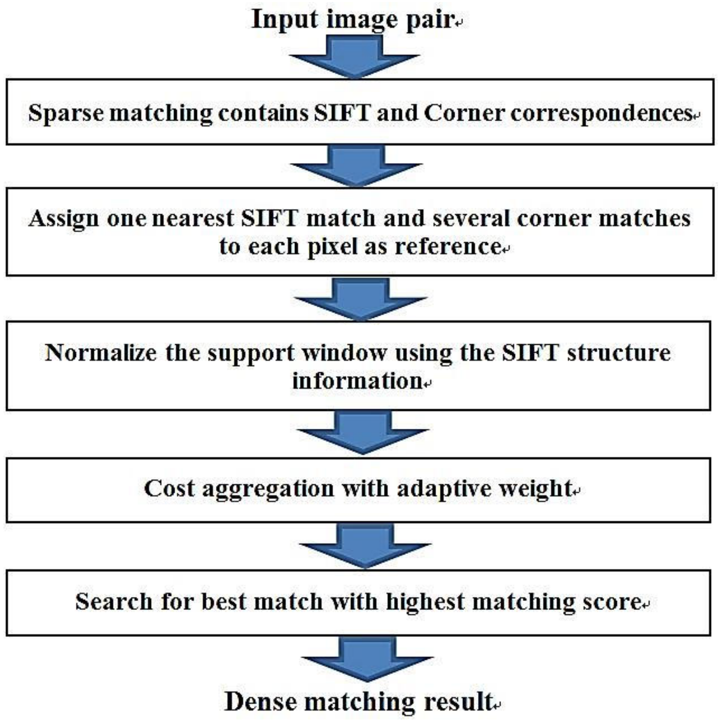
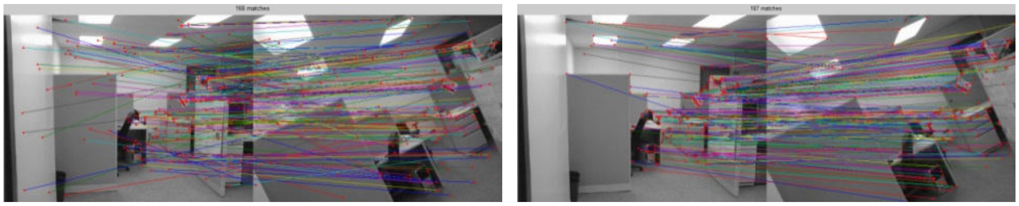

Input image pair↵
Sparse matching contains SIFT and Corner correspondences↵
Assign one nearest SIFT match and several corner matches to each pixel as reference↵
Normalize the support window using the SIFT structure information↵
Cost aggregation with adaptive weight↵
Search for best match with highest matching score↵
Dense matching result↵

Fig. 1. Flowchart of our approach

2.2 Sparse Key Point Matching

Our goal in this step is to obtain enough reliable reference matches, which need to be invariant to rotation and scale changes. The local invariant feature detectors [8] and descriptors [7] have shown good performance in sparse wide baseline matching. Hence, we use DoG detector and SIFT descriptor [6] to generate initial matches. For matching, we adopt a modified nearest neighbor strategy: both the nearest distance and the first-to-second nearest distance ratio are required to be greater than a threshold; and another requirement is that both corresponding descriptors need to be the nearest neighbor of each other. Besides SIFT matches, we extract image corners and use a robust corner matching method in [9, 10] to produce new matches, which is also invariant to scale and rotation changes. In addition, we use the epipolar constraint to refine those matches. During refinement, both the fundamental matrix F and the true correspondences are estimated iteratively to minimize the Sampson cost function [11]; a RANSAC [12] scheme is adopted to eliminate outliers.

In the test we found that our sparse matching strategy is effective to increase the number of reference matches as well as improve the reliability. An illustrating example of sparse wide baseline matching is presented in Figure 2.



(a) (b)

Fig. 2. Sparse matching result for an indoor image pair. The images are 816×612 pixels. (a): the initial SIFT match without refinement, using Vlfeat [15] library, 168 matches shown by the line linking dot; note the clear mismatches. (b): sparse matching result using our approach, 187 matches in total; note that mismatches in (a) is eliminated and new reliable matches are found.

3 Rotation and Scale Invariant Dense Matching

In window-based methods, it is implicitly assumed that pixels in the matching windows should have similar disparities [2]. When the baseline is short, this assumption is easy to meet, as the image pairs are almost identical except for a little translation in one dimension, so it is effective to search for correct correspondences by sliding the window along this dimension. However, in the wide baseline case, when geometric transformations such as rotation and scaling exist, this assumption becomes invalid. Therefore, the support windows need to be adjusted to capture the image of the same object surface. In section 3.1, we will discuss how to normalize the support windows using the sparse matching result. The definition and calculation of matching score are introduced in section 3.2. The dense matching process is introduced in section 3.3.

3.1 Normalize the Support Window

In this subsection, we discuss how to construct rotation and scale covariant support windows based on sparse matching result. First, a transformation matrix is estimated using the reference matches, and then it is used to normalize the support window.

Use SIFT Match to Estimate a Transformation Matrix

The goal of this step is to find optimal support window with adaptive shape. For an arbitrary candidate match, the support window is adjusted refer to the blob structure nearby. As each SIFT feature reliably represents the scale and orientation information of the blob, it is reasonable to take use of the SIFT matches as spatial prior to guide the adjustment. Give two image pairs I_1 and I_2. Let $p_1 = I_1(x_1, y_1)$, $p_2 = I_2(x_2, y_2)$ be an candidate match, where x_1, y_1, x_2, y_2 are image coordinates. The SIFT feature points $s_1 = \{I_1(x_{s1}, y_{s1}), \sigma_1, \theta_1\}$ and $s_2 = \{I_2(x_{s2}, y_{s2}), \sigma_2, \theta_2\}$ are the reference matches assigned to p_1 and p_2 respectively, where $x_{s1}, y_{s1}, x_{s2}, y_{s2}$ are image coordinates, and $\sigma_1, \sigma_2, \theta_1, \theta_2$ are the characteristic scale and dominant orientation of the SIFT feature. In order to achieve scale and orientation invariance, the coordinates of the candidate match and its support region should be transformed refer to the SIFT keypoint scale and orientation. We introduce a new SIFT coordinate system, in which the origin is at the SIFT keypoint (x_{s1}, y_{s1}), (x_{s2}, y_{s2}), and the horizontal axis is oriented to the dominant orientation. The relative position of p_1 in the local SIFT coordinate system is presented in Figure 3. Then, we normalize the putative correspondence p_1 and p_2 into the SIFT coordinate systems centered at (x_{s1}, y_{s1}), (x_{s2}, y_{s2}) respectively, which can be expressed as follows:

$$\tilde{p}_1 = \begin{bmatrix} \tilde{x}_1 \\ \tilde{y}_1 \end{bmatrix} = \frac{1}{\sigma_1} \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 \\ \sin\theta_1 & \cos\theta_1 \end{bmatrix}^T \begin{bmatrix} x_1 - x_{s1} \\ y_1 - y_{s1} \end{bmatrix} \quad (1)$$

$$\tilde{p}_2 = \begin{bmatrix} \tilde{x}_2 \\ \tilde{y}_2 \end{bmatrix} = \frac{1}{\sigma_2} \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 \\ \sin\theta_2 & \cos\theta_2 \end{bmatrix}^T \begin{bmatrix} x_2 - x_{s2} \\ y_2 - y_{s2} \end{bmatrix} \quad (2)$$

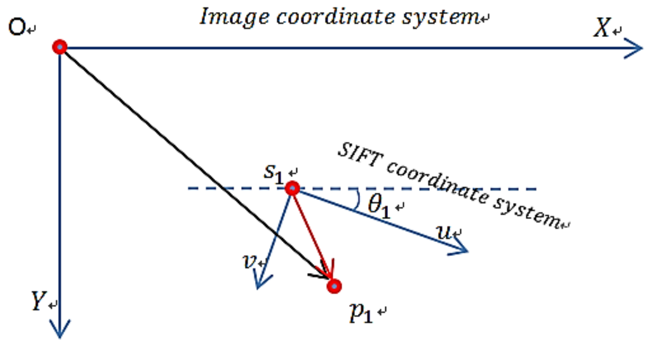


Fig. 3. Relationship between point p_1 and the local SIFT coordinate system centered at s_1; note that s_1 is the nearest SIFT feature point of p_1. $\overrightarrow{Op_1} = (x_1, y_1)^T$ and $\overrightarrow{s_1p_1} = (\tilde{x}_1, \tilde{y}_1)^T$

Suppose p_1 and p_2 are true matches, and this match is consistent with the blob structure represented by SIFT, the normalized coordinates $(\tilde{x}_1, \tilde{y}_1)^T, (\tilde{x}_2, \tilde{y}_2)^T$ are expected to be identical. Combine (1), (2) with the assumption that $(\tilde{x}_1, \tilde{y}_1)^T = (\tilde{x}_2, \tilde{y}_2)^T$, then p_2, the corresponding point of p_1 in I_2, can be predicted with a transformation matrix H. Rewrite p_1, p_2 in homogeneous coordinates $p_1 = (x_1, y_1, 1)^T$ and $p_2 = (x_2, y_2, 1)^T$; H is a local geometric transformation matrix. Then, H can be estimated as follow:

$$p_2 = H * p_1;\ \text{and}\ H = \begin{bmatrix} S*R & t \\ 0^T & 1 \end{bmatrix} \quad (3)$$

Let $\sigma = \sigma_2/\sigma_1$ and $\theta = \theta_2 - \theta_1$, thus,

$$S = \sigma,\ R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

$$t = \begin{bmatrix} t_x \\ t_y \end{bmatrix} = -S*R*\begin{bmatrix} x_{s1} \\ y_{s1} \end{bmatrix} + \begin{bmatrix} x_{s2} \\ y_{s2} \end{bmatrix} \quad (4)$$

Use Corner Match to Refine the Transformation Matrix

If a pixel p_1 is far from the nearest blob, the estimated local transformation H may be inaccurate. In our approach, we refine the transformation matrix with the help of corner matches. First, several corner matches near p_1 are selected: $c_{k1} \leftrightarrow c_{k2}$, $k = 1,2, \ldots M$. For each corner c_{k1} in I_1, we use a candidate transformation matrix H to predict its corresponding point c_{k2}' in I_2. Transfer error of this set of corner correspondences is defined as follow:

$$\text{transfer\_error}(c_{k1}, c_{k2}, H) = \sum_k d(H * c_{k1}, c_{k2})^2, \; k = 1,2, \ldots M \tag{5}$$

Then, the transformation matrix H can be optimized by minimizing this transfer error. As described above, the calculation of H is based on SIFT feature information, which includes keypoint position, characteristic scale and dominant orientation. The scale ratio $\sigma = \sigma_2/\sigma_1$ and orientation difference $\theta = \theta_2 - \theta_1$ may be not accurate due to non-rigid distortion of the local structure. So, we perturb these two parameters in a discrete set around their original value and obtain a set of transformation matrix H, and use each H to construct several pairs of corresponding patches W_1, W_2. If $p_1 \leftrightarrow p_2$ is a correct match, W_1, W_2 are expected to be identical when H is well estimated. So, the H with lowest transfer error can be accepted. Some experimental results are demonstrated in Figure 4. It is obviously shown that the image content covered by the corresponding support windows tend to be similar after normalization.

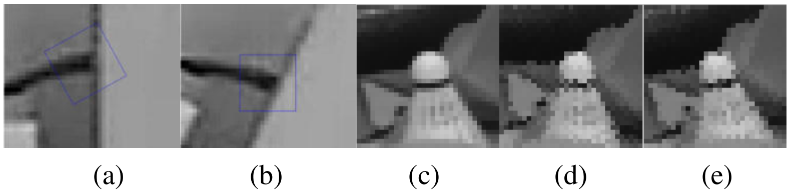


Fig. 4. Normalized support window. (a)(b) are two image patches with rotation change. The support windows can cover the same image content after normalization. (c) is a local patch in I_1, and (d),(e) are the normalized corresponding patch in I_2. The support window of (d) has not been refined using corner matches, while (e) has. Note that (c)(d) have some error in orientation. WZNCC score of (c)(d) is 0.8682, and (c)(e) is 0.9424, which is an obvious improvement.

In this way, the modified window-based method is invariant to rotation and scale changes and thus effective in wide baseline case. Following, we will discuss the similarity measure and cost aggregation strategy in the normalized window.

3.2 Matching Score

Dense matching depends on a cost function to measure the similarity of image locations. The normalized cross-correlation (NCC) and its improved version ZNCC are reported to have the best performance [13].

Suppose a pixel p_1 in the left image I_1, and its correspondence in the right image I_2 is p_2; N_{p1}, N_{p1} are pixels within the support windows, $q_1 \in N_{p1}, q_2 \in N_{p2}$ and $q_2 = q_1 - d$, $d = [d_x \quad d_y]^T$ for the disparity. Then, the similarity of two image patches centered at p_1 and p_2 can be calculated by ZNCC score:

$$\text{Score}_{ZNCC}(p_1, p_2) = \frac{\sum_{q_1 \in N_{p1}, q_2 \in N_{p2}} (I_1(q_1) - \bar{I_1}(q_1))(I_2(q_2) - \bar{I_2}(q_2))}{\sqrt{\sum_{q_1 \in N_{p1}} (I_1(q_1) - \bar{I_1}(q_1))^2 \sum_{q_2 \in N_{p2}} (I_2(q_2) - \bar{I_2}(q_2))^2}} \quad (6)$$

However, in the window-based method, pixels in the support window have expected to have similar disparities, so that they can largely support the central pixel. In fact, this local planarity assumption is difficult to achieve, as finding the optimal support region not straddle object boundary is very difficult. Yoon proposed a method called adaptive support weighted window to solve this problem [14]. It can effectively cope with occlusions and depth discontinuities near object boundaries. In our approach, we add support weight in ZNCC to measure the similarity of corresponding image patches. The weight is decided according to intensity similarity $\Delta_{c_{pq}}$ and geometric proximity $\Delta_{g_{pq}}$; γ_c and γ_p are two coefficients to keep the balance.

$$w(p, q) = \exp(-\left(\frac{\Delta_{c_{pq}}}{\gamma_c} + \frac{\Delta_{g_{pq}}}{\gamma_p}\right)) \quad (7)$$

Where $\Delta_{c_{pq}}$ represents the Euclidean distance between two colors c_p and c_q in the CIELab color space, and $\Delta_{g_{pq}}$ is the distance between p and q in the image domain:

$$\Delta_{c_{pq}} = \left|c_p - c_q\right| \ , \ \Delta_{g_{pq}} = |p - q| \quad (8)$$

Thus, our matching score WZNCC (weighted ZNCC) is:

$$\text{Score}_{WZNCC}(p_1, p_2) = \frac{\sum_{q_1 \in N_{p1}, q_2 \in N_{p2}} w_1(p_1, q_1) w_2(p_2, q_2)(I_1(q_1) - \bar{I_1}(q_1))(I_2(q_2) - \bar{I_2}(q_2))}{\sqrt{\sum_{q_1 \in N_{p1}} (w_1(p_1, q_1)(I_1(q_1) - \bar{I_1}(q_1)))^2 \ \sum_{q_2 \in N_{p2}} (w_2(p_2, q_2)(I_2(q_2) - \bar{I_2}(q_2))^2)}} \quad (9)$$

The matching score is normalized in [0, 1]. In this way, we can search for the best match according to this score. Some results are presented in Figure 5.

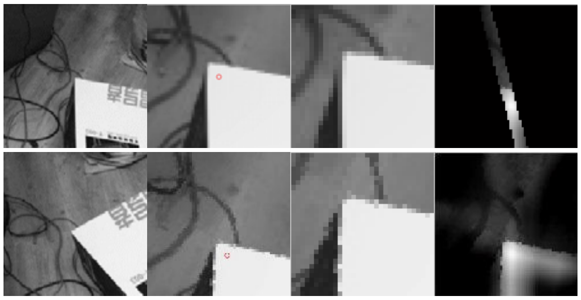

(a) (b) (c) (d)

Fig. 5. Adaptive weighted support window. (a) original image; (b) corresponding point; (c) the normalized support window; (d) WZNCC score during search, light means high score; in the upper image, search range is limited to the region near the epipolar line, with threshold 3.

3.3 Procedure of Dense Matching

In dense matching, the disparity map is obtained by searching for correspondences pixel by pixel. This is the most time consuming step, so we need to limit the search range to reduce redundancy. For each pixel on a selected reference image, the correct match in the other view is obtained by searching within a limited range. In our method, the normalized support windows need to be computed for every pixel in the neighborhood from which new matches are searched. An effective way to reduce redundancy is to limit the search space. In multiple view geometry, there is a map from a point in one image to its corresponding epipolar line in the other image, which is represented by the fundamental matrix. In other word, the confidence of a putative location is low when it is far away from the epipolar line. During search, if the distance from a candidate point to this line exceeds a threshold, it is rejected, as it violates the epipolar constraint. Then, the search range is limited to a strip region, see Figure 5(d). The dense matching procedure is presented in Algorithm 1.

Algorithm 1. Dense Matching Procedure

```
Input: Color images $I_1, I_2$; Sparse matching result
       Search range $N$, Window size $W$
Output: Dense matching result
1. Compute the fundamental matrix, refer to [11]
2. For each pixel in $I_1$, calculate its local geometric transformation matrix $H$, refer to section 3.1.
3. Initialize the corresponding position in $I_2$, form a putative match $p_1 \leftrightarrow p_{2\_initial}$
4. If $p_{2\_initial}$ is out of the image range, recorded as occlusion; return
5. Limit the search space with epipolar constraint
For each pixel within the search range in $I_2$,
    6. Extract corresponding image patches Patch1 and Patch2, warp the window refer to section 3.1.
    7. Calculate matching score according to section 3.2,
    8. If matching score is below a threshold, recorded as occlusion; return
    9. Select the one with highest score as correct match, denoted by $p_2$
    10. Add $\{p_1 \leftrightarrow p_2\}$ to Dense correspondences.
End for
```

4 Experimental Results

In this section, we present experiments on real images, which are collected from public dataset (e.g. INRIA on the web) or from our lab dataset. The latter are obtained by handheld cameras at arbitrary positions. The geometric distortion of the image pairs is apparent, such as rotation and scale changes. Firstly, we use images with significant rotation and scale changes to show the robustness of our algorithm. Dense matching results on several wide baseline image pairs are demonstrated. Then, we discuss the details of the experiment on a couple of indoor image pairs, and use the Sampson error from known homography for evaluation. We implemented the proposed method in MATLAB, and tested it on a laptop with Intel Core i5 CPU and 6GB RAM.

4.1 Test on Images with Significant Rotation and Scale Changes

The top row in Figure 6 is captured by a mobile phone, and the bottom is the image pair New York (from INRIA). We choose the first image as target, and put the corresponding pixel in the second image on it. Dense matching result is shown on the right column. The black region represents inaccurate matches, that is, the matching score is below 0.8. Scale change is apparent in the top row images, while the bottom row contains images with significant rotation change (about 100 degrees). The dense matching result is good on both cases, which demonstrates that our method is robust against rotation and scale changes.

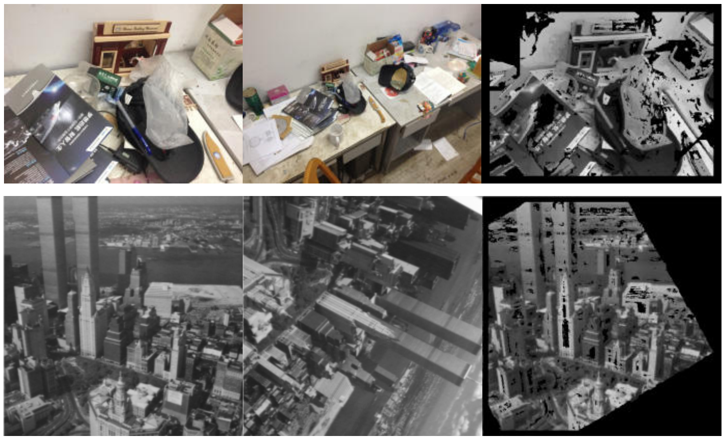


Fig. 6. Images with significant rotation and scale changes: The left two columns are original images, and the right column is dense matching results

4.2 Evaluation and Analysis

The 'office' data consists of two images with resolution 640×480 pixels taken with two uncalibrated cameras. First, we conduct sparse wide baseline matching on the image pair. Outliers have been removed using the epipolar constraint with RANSAC. Before dense matching, we use bilateral filter to remove noise. Then, for a current pixel, the support window is normalized using the sparse matching result. The window size is set to 15x15, empirically, as too small windows lead to low distinctiveness in homogenous regions, and too large windows tend to straddle depth discontinuities. In Figure 4 (c)-(e), we see that, the image content captured in the normalized support windows tend to be identical. After the refinement of the transformation matrix, the matching score

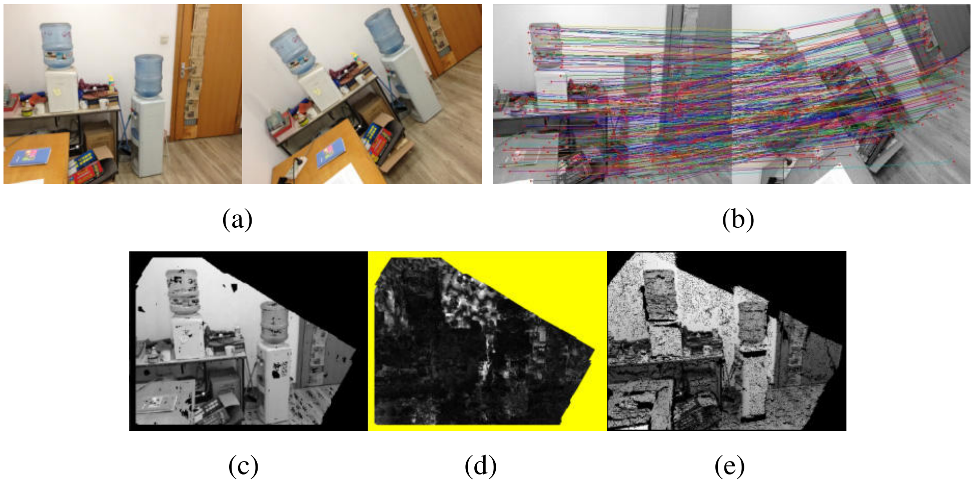


Fig. 7. Dense matching result on image data “office”. (a): original images; (b): sparse matching result; (c): our dense matching result; (d): Sampson error image, bright area means large error; (e): quasi-dense matching result [5], black region means no match.

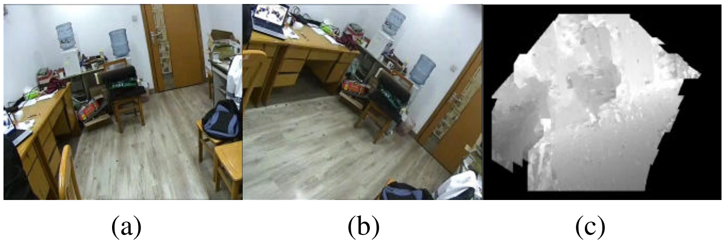


Fig. 8. Depth map from stereo-pair images. (a), (b): original images, (c): depth map.

Table 1. Percentage of good matches

Sampson distance (pixel)	<1	<2	<3	<4
%	67.57%	87.49%	95.21%	98.06%

is improved from 0.8682 to 0.9424. Search range for a candidate pixel is set to N=10 pixels. The computing time is 8.21 milliseconds per match. Correspondences with matching score less than 0.8 are rejected. Dense matching result is shown in Figure 7(c). Compared with the quasi-dense method [5], our matching result is much denser. The Sampson error [11] to the ground truth homography is shown in Table 1. Finally, in order to demonstrate the quality of the matches, we calibrated the cameras and computed a depth map of the scene, as shown in Figure 8.

5 Conclusion

In this paper, we proposed a modified window-based method for dense matching between two uncalibrated images with significant rotation and scale changes. With help of the powerful local invariant features, the support window is normalized before matching. Thus, we achieve robust matching against rotation and scale changes. In this way, our approach can deal with images obtained from handheld cameras without calibration. Experimental results on real images show that our approach is effective to obtain accurate and dense correspondences. However, our method is originally designed to handle scale and rotation changes, which may suffer from significant geometric transformations like affine changes or perspective changes. A major direction of future work is to develop an affine invariant approach to handle this problem.

Acknowledgement. This work is supported in part by National Natural Science Foundation of China under Grant No. 61175014, and the Fundamental Research Funds for the Central Universities of China.

References

1. Scharstein, D., Szeliski, R.: A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. IJCV 47(1-3), 7–42 (2002)
2. Min, D.B., Lu, J.B., Do, M.N.: A Revisit to Cost Aggregation in Stereo Matching: How Far Can We Reduce Its Computational Redundancy? In: ICCV, pp. 1567–1574 (2011)
3. Gallup, D., Frahm, J.-M., et al.: Real-time plane-sweeping stereo with multiple sweeping directions. In: CVPR, pp. 2110–2117 (2007)
4. Lhuillier, M., Quan, L.: A quasi-dense approach to surface reconstruction from uncalibrated images. TPAMI 27(3), 418–433 (2005)
5. Kannala, J., Brandt, S.S.: Quasi-dense wide baseline matching using match propagation. In: CVPR, pp. 2126–2133 (2007)
6. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV 60(2), 91–110 (2004)
7. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. TPAMI 27(10), 1615–1630 (2005)
8. Mikolajczyk, K., Tuytelaars, T., Schmid, C., et al.: A comparison of affine region detectors. IJCV 65(1-2), 43–72 (2005)
9. Shi, F., Huang, X., Duan, Y.: A Hybrid Approach for Robust Corner Matching. In: Tarn, T.-J., Chen, S.-B., Fang, G. (eds.) Robotic Welding, Intelligence and Automation. LNEE, vol. 88, pp. 169–177. Springer, Heidelberg (2011)
10. Yan, B., Shi, F., Yue, J.: An Improved Image Corner Matching Approach. In: Huang, D.-S., Bevilacqua, V., Figueroa, J.C., Premaratne, P. (eds.) ICIC 2013. LNCS, vol. 7995, pp. 472–481. Springer, Heidelberg (2013)
11. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2004)

12. Fischler, M., Bolles, R.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Comm. ACM 24(6), 381–395 (1981)
13. Hirschmueller, H., Scharstein, D.: Evaluation of Stereo Matching Costs on Images with Radiometric Differences. TPAMI 31(9), 1582–1599 (2009)
14. Yoon, K.J., Kweon, I.S.: Adaptive support-weight approach for correspondence search. TPAMI 28(4), 650–656 (2006)
15. Vedaldi, A., Fulkerson, B.: Vlfeat: An open and portable library of computer vision algorithms. In: Proc. Int. Conf. on Multimedia, pp. 1469–1472 (2010)