Research Article

International Journal of
Distributed
Sensor Networks

International Journal of Distributed Sensor Networks
2016, Vol. 12(12)

DOI: 10.1177/1550147716680826
ijdsn.sagepub.com
SAGE

An affine invariant approach for dense wide baseline image matching

Fanhuai Shi[1], Jian Gao[1,2] and Xixia Huang[3]

Abstract

Visual sensor networks have emerged as an important class of sensor-based distributed intelligent systems, where image matching is one of the key technologies. This article presents an affine invariant method to produce dense correspondences between uncalibrated wide baseline images. Under affine transformations, both point location and its neighborhood texture are changed between views, so dense matching becomes a tough task. The proposed approach tends to solve this problem within a sparse-to-dense framework. The contribution of this article is in threefolds. First, a strategy of reliable sparse matching is proposed, which starts from affine invariant features extraction and matching and then these initial matches are utilized as spatial prior to produce more sparse matches. Second, match propagation from sparse feature points to its neighboring pixels is conducted in the way of region growing in an affine invariant framework. Third, the unmatched points are handled by low-rank matrix recovery technique. Comparison experiments of the proposed method versus existing ones show a significant improvement in the presence of large affine deformations.

Keywords

Visual sensor networks, affine invariant, dense matching, wide baseline, uncalibrated images

Date received: 16 July 2016; accepted: 29 October 2016

Academic Editor: Ting Zhu

Introduction

Visual sensor networks have emerged as an important class of sensor-based distributed intelligent systems. Consisting of a large number of low-power camera nodes, visual sensor networks support a great number of novel vision-based applications, such as visual surveillance, camera calibration, three-dimensional (3D) modeling, and so on.[1] Image matching is one of the key technologies in visual sensor networks, and it is also a fundamental problem of many applications, such as 3D reconstruction, camera calibration, motion prediction, and image stitching. This problem is particularly challenging when there exist significant spatial transformations between wide baseline image pairs. The geometric deformations, such as translation, rotation, scaling, skew and stretch, can cause great matching ambiguity. So, the main difficulty is to find an invariant approach under various spatial transformations.

The simplest transformation is a small translation in one dimension. Traditional dense two-frame matching algorithms aim at computing disparity maps for these short-baseline images.[2] In this situation, the search space of disparity only contains one dimension, and matching confidence can be measured by the correlation of corresponding local patches. These algorithms

[1]Department of Control Science and Engineering, College of Electronics and Information Engineering, Tongji University, Shanghai, China
[2]Department of Computer Science and Engineering, New York University, New York, NY, USA
[3]Key Laboratory of Marine Technology and Control Engineering, Shanghai Maritime University, Shanghai, China

Corresponding author:
Fanhuai Shi, Department of Control Science and Engineering, College of Electronics and Information Engineering, Tongji University, Shanghai 201804, China.
Email: fhshi@tongji.edu.cn

can be broadly classified into two classes: global[3,4] and local.[5]

Matching becomes even more difficult in wide baseline cases. For the variation of camera poses and positions, the geometric transformation here is not only translation but also rotation and scaling, even including viewpoint distortion such as skew and stretch. Under these circumstances, images become quite different, and correspondences need to be searched in two dimensions; in addition, local texture is deformed, so matching by simply computing window correlation usually fails. Thus, dense matching tends to be a quite challenging task. There has been a large amount of work on wide baseline matching during the past decades. For example, algorithms based on plane sweep[6] first test a family of plane hypotheses and choose the best one for each pixel and then perform dense matching and depth map computation at the same time. A limitation is that cameras are required to be fully calibrated to construct those plane hypotheses. However, the camera poses and scene geometry are unavailable in many applications. Strecha et al.[7] make an evaluation of image-based 3D techniques with calibrated cameras/images.

For uncalibrated images, prevalent approaches are based on local invariant features.[8] SIFT[9] is the most popularly used feature in wide baseline matching. Difference of Gaussian (DoG) key points are first extracted and then SIFT descriptors are generated from local invariant regions centered at these key points. SIFT features are invariant to scaling and rotation and robust against various disturbances such as additional noise, change in illumination, and small affine distortion. Mikolajczyk and Schmid[10] proposed an affine invariant interest point detector. They first use an affine-adapted Harris detector to determine interest point locations and take multi-scale version of this detector for initiation. Then, the scale, location, and the neighborhood of each key point are modified by an iterative algorithm, which finally converges to an affine invariant point. Each key point is associated with an affine invariant support region, represented by a second moment matrix, which allows the generation of affine invariant descriptors for robust matching in wide baseline case. Similar local invariant features such as SURF,[11] ASIFT,[12] GLOH,[13] DAISY,[14] and Scale-Less SIFT[15] were also proposed. They are more or less invariant under different viewing conditions. However, the number of feature points is unpredictable. In low-texture regions, there may be very few features. Consequently, great matching ambiguity exists in these regions. Although sparse matching result is sufficient for object recognition, the number (or density) of reliable correspondences is quite important for applications such as 3D scene reconstruction and motion estimation.

To construct dense correspondences, a region growing framework was first presented by Otto and Chau[16] to process satellite images and was then extended by many researchers.[17–21] In this scheme, some distinctive features like corners and salient regions are first extracted and matched as seed points. Then, a correlation-based region growing step propagates them into more ambiguous regions of the images. During propagation, the obtained high-confidence matches are used to guide nearby pixels for further matching. In this matching problem, the scene is usually assumed to be composed of piecewise-smooth, Lambertian reflection and textured surfaces. In the same surface, disparity variations and local pattern deformations are relatively small, so the propagation is effective. A deficiency is that the propagation often stops at object boundaries, and a region may be lost if no reliable seed match is found in this region. So, in order to produce dense matching, the initial matching result is quite important as well as the propagation strategy.

Besides, research on dense matching for image stitching[22] or matching across different scenes is also active in recent years, and much work has been done, such as SIFT flow,[23] deformable spatial pyramid matching,[24] DAISY filter flow,[25] scales propagation prior to matching,[26] and so on. However, it is not the topic of this article.

In this work, we introduce an affine invariant method to perform dense matching between two images of the same scene. This approach has two main steps. The first step extracts and matches a sparse set of affine invariant features: seed points and their affine invariant regions. Then, these initial matches are incorporated as spatial prior to generate more candidate matches. In this way, the number of reliable seeds apparently increases. In the second step, the obtained sparse correspondences are used to initialize a dense matching propagation process. Under the assumption that disparity variations and local pattern deformations change smoothly inside a smooth region, the disparity and affine transformation parameters of obtained reliable matches can be propagated to its neighboring pixels as initial guess and then refined. During iterative region growing, high-confidence matches are used to guide nearby pixels for further matching, thus nearly dense point-to-point matching is produced. Finally, in order to handle the remaining unmatched points, low-rank matrix recovery technique is utilized to complete dense matching.

The rest of this article is organized as follows: section "Sparse matching" introduces the sparse matching technique, in which the local spatial deformation is modeled by an affine transformation matrix associated with each match. Section "Dense matching based on region growing" discusses dense matching technique based on region growing and then a strategy of handling

unmatched points is proposed in section "Handling unmatched points." Finally, experimental results on real images are presented in section "Experimental results and analysis."

Sparse matching

In this section, we introduce how to produce reliable sparse matches. First, some distinctive feature points are extracted, which should be robust over significant geometric deformations. Second, we explain how to increase the matching number as well as improve the accuracy. Each pair of correspondence is associated with an affine transformation matrix to model its local geometric deformation, which is used to guide further matching in dense matching.

Which feature to extract?

The proposed dense matching approach starts from the detection of sparse key points. Some methods extract edges or corners and then matched with correlation; others extract uniformed regions after segmentation and conduct comparison on shape or mean color. Several drawbacks are needed to be aware of, such as noise, illumination change, repetitive patterns, and geometric deformations. Recently, some local invariant features[8] have been reported to have good performance in sparse wide baseline matching. The most famous feature is SIFT, and others such as SURF, MSER, Harris-Affine, and Hessian-Affine are also widely used. We use Hessian-Affine detector to establish sparse matching, which is invariant to affine transformations. Compared with DoG detector, the response of Hessian is weak near contours and straight edges, where signal change is only in one direction. These points are less stable as their localization is more sensitive to noise or small changes in neighboring texture, so Hessian can avoid reaching maxima in these areas. An advantage over Harris detector is that Hessian detects blob-like features other than Harris corners, so the returned locations are more suitable for scale estimation as the filters for spatial and scale localization are similar. Using the Hessian-based detector, a large number of feature points can be extracted, which lead to a good coverage of the scene.

How to compare two corresponding regions?

Matching is performed by comparing the neighborhood texture of two candidate key points. There are two prevalent ways: one is to sample local image intensities around the key point and matching them using a similarity measure, for example, sum of absolute differences (SAD), sum of squared differences (SSD), and correlation functions. However, the similarity of image patches is sensitive to noise and affine changes or non-rigid deformations. Another is to compute a descriptor for the local image region. The descriptor needs to be highly distinctive and robust to variations under arbitrary viewing conditions. SIFT-based descriptors are reported to have the best performance under a variety of experimental conditions.[13]

We adopt both techniques in the proposed approach. SIFT-like descriptor is extracted in the affine invariant region associated with each feature point. This region is determined by the second moment matrix M. Suppose $M = \mu(x_0, \sigma_I, \sigma_D)$, then the boundary of the support region is an ellipse

$$(x - x_0)^T M (x - x_0) = 1 \tag{1}$$

where x_0 is the key point location. Replace M by a 2×2 symmetric matrix C, equation (1) can be written in the form of $x'^T C x' = 1$.

Let x_1, x_2 be a pair of candidate match and C_1, C_2 be the ellipses of corresponding regions, respectively. They are related by a local affine transformation A

$$C_1 = A^T C_2 A \tag{2}$$

Thus, the local affine transformation can be computed as follows

$$A = C_2^{-\frac{1}{2}} R C_1^{-\frac{1}{2}} \tag{3}$$

where R is a rotation matrix, and it is determined by the orientation of local image gradients, which is represented by the dominant direction of the SIFT descriptor. As Hessian-Affine is affine invariant feature, the local transformation A estimated by the second moment matrix is effective within the ellipse support region. For pixels on smooth surfaces, the spatial deformations do not change quickly. Therefore, the local transformation of pixels not far away from the key point center can be approximated by A. In this way, we are able to match two corresponding regions in an affine invariant way.

In order to gain invariance over affine distortions, we normalize the image patches before matching and then measure their similarity. Given patch P_1 in one image, its corresponding patch $\widetilde{P_2}$ in the other image is determined by an affine mapping

$$\widetilde{P_2} = \{q_2 | q_2 = A * q_1, q_1 \in P_1\}$$

For a correct match, P_1 and $\widetilde{P_2}$ will be two identical patches when the affine transformation is accurate. Thus, we can estimate the confidence for a candidate match by computing the similarity of P_1 and P_2. SSD is adopted as the similarity function. Considering the left–right consistency, backward matching is added. The matching cost is as follows

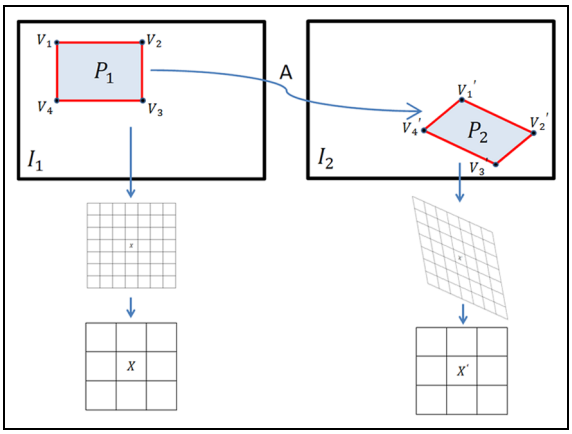

Figure 1. Illustration of affine normalization of local image patches.

$$C(P_1, P_2) = \frac{SSD\left(P_1, \widetilde{P_2}\right) + SSD\left(P_2, \widetilde{P_1}\right)}{2} \quad (4)$$

The SSD can be replaced by SAD or correlation or other similarity functions. This process is illustrated in Figure 1.

Thus, the matching cost for a candidate match at point p_1, p_2 is represented by the matching cost of corresponding image patches P_1, P_2

$$Cost(p_1, p_2) = C(P_1, P_2) \quad (5)$$

Generate more matches

For sparse matching, we expect to obtain as many matches as possible, which need to be accurate and robust against affine deformations. After the Hessian-Affine features being extracted, the 128-dimensional feature vector and another 5 parameters (2 for location of the key point and 3 for the data of symmetric second moment matrix M in equation (1)) are used for sparse matching. Traditional methods compare feature similarities and then use nearest-neighbor strategy to perform matching. First-to-second nearest-distance ratio is often used to reduce false matches. But compared with the feature points that appear in both images, the matching number is relatively low (see Figure 2 first row).

As accurate and sufficient sparse matching result has great impact on the performance of dense matching, we designed an approach to generate more matches. First, initial matches are established through the comparison of feature similarities. The best candidate match for each key point is set to its nearest neighbor, which is defined as the one with minimum Euclidean distance for the 128-dimensional SIFT-like descriptor vector. We do not eliminate false matches with first-to-second distance ratio, as not considering the spatial relationship among key points will lead to the exclusion of

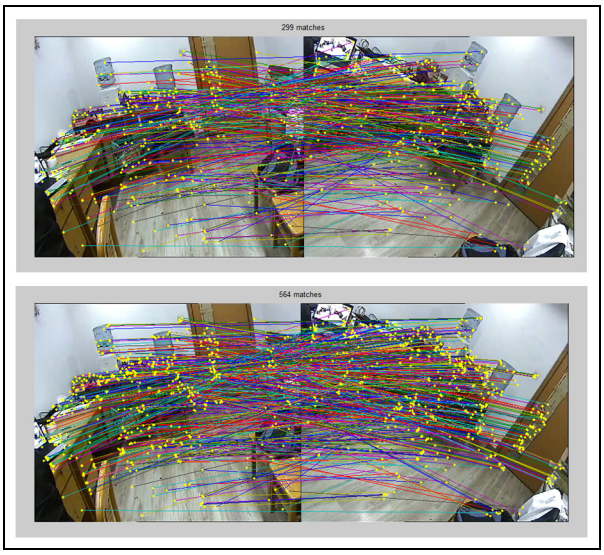

Figure 2. Sparse key points detection and matching result. There are 1997 Hessian-Affine feature points in the left image and 1699 in the right, 299 matches in the top row, and 564 matches in the bottom.

many true matches. Instead, the obtained initial matches are ranked according to matching cost, as was described in section "How to compare two corresponding regions?" A threshold is set to select some reliable candidates as good seeds. Then, we incorporate these good seeds as spatial prior to produce more matches. Considering the spatial consistency between key points, those high-confidence matches are used to guide nearby pixels for further matching. At each iteration, the best seed (with lowest matching cost) is taken out from the set "good seeds" to produce more matches. We search for new feature points near the best seed and find its putative correspondences in the other view. If the putative match is reliable, then it is added into the set "good seeds." New matches are generated iteratively, and the procedure stops when the set "good seeds" becomes empty.

In detail, suppose the coordinates of the best seed match are $p_0 = (x_0, y_0)$ and $p'_0 = (x'_0, y'_0)$, with local affine transformation A_0. First, in both images, all key points within the two regions are taken out. Reliable matching is expected to be established between these key points. The matching cost function has two components: one is the feature similarity and the other is spatial consistency. Suppose $p_1 \leftrightarrow p'_1$ is a putative match. Denote the Euclidean distance of the corresponding feature vectors as dsift, which is the first component. And the second is computed by comparing pairwise spatial consistency between $p_0 \leftrightarrow p'_0$ and $p_1 \leftrightarrow p'_1$. According to the smoothness constraint, local affine transformations change slightly on smooth surfaces. So, the corresponding position of p_1 can be predicted by $p_0 \leftrightarrow p'_0$ and A_0

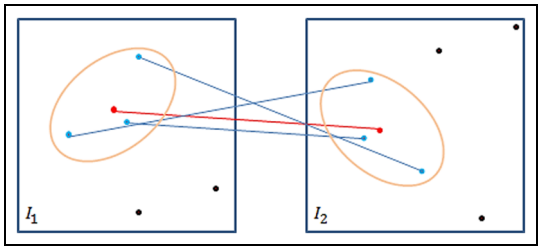

Figure 3. Generating more matches from seed match. The red point is a seed match, and blue points are new matches generated by the seed; the size of ellipses is determined by threshold D; black points too far away from the seed (out of the ellipses) are not considered.

$$\widetilde{p'_1} = p'_0 + A_0^{-1}(p_1 - p_0) \tag{6}$$

But the true match of a key point needs also to be a key point, so here is a spatial error between the predicted position and the candidate matching position

$$\text{spacial}_{\text{err}} = \left| p'_1 - \widetilde{p'_1} \right| \tag{7}$$

Thus, the matching cost is calculated as follows

$$\text{Cost}(p_1, p'_1) = \text{spacial}_{\text{err}} \cdot * \text{dsift} \tag{8}$$

Candidates with high confidence (matching cost not beyond threshold T) are added to the set "good seed," which can be used to generate more neighboring matches in the following iterations (see Figure 3).

For a "best seed," the search region where new matches are generated is determined by equation (4). On one hand, too small region is useless, as there may be no new key points existing in such area. On the other hand, if the region is too large, the local affine transformation may be inaccurate, as regions far away from the seed may violate the smoothness assumption. So, there should be a compromise in the selection of the region size.

Finally, all new generated matches need to be checked by computing local image correlation by equation (5). Matches with low correlation cannot be added to "good seeds," as the corresponding local image patches are quite dissimilar. In this way, we obtained reliable and sufficient sparse matching result, and each match is associated with a local affine transformation matrix. The detailed procedure of sparse matching can be found in the supplementary file. It is shown in the second row of Figure 2 that the number of sparse matching has been significantly increased.

Dense matching based on region growing

In the previous section, a sparse matching technique was described. Each match is associated with an affine transformation matrix, which reflects its local structure. This section will focus on the second step: dense matching, the basic idea of which is based on region growing. We incorporate region growing in an affine invariant framework for match propagation. In order to handle the remaining unmatched points, low-rank matrix recovery technique is utilized to complete the dense matching.

Region growing

Region growing is originally an approach for segmentation, in which neighboring pixels with similar properties are merged together. A region growing algorithm usually has three steps: initialization, propagation, and termination. It starts at seed points with some specific attribute and then expands to neighbor regions iteratively. During the expansion, this attribute is spread to neighboring pixels with a growing rule. If the attribute of new pixel is not eligible, this pixel will not be merged into the region. The expansion terminates when all neighbors have been handled. Our goal is to construct dense matching, but many regions are not so distinctive to perform reliable matching, for example, low-texture regions. Based on the assumption that disparity varies slightly on smooth object surfaces, matching can be propagated from high distinctive regions to low distinctive regions. Sparse matching result obtained from feature matching is often used for initialization and then matching is propagated from these reliable seeds to neighboring pixels. At each iteration, the match with highest confidence is used to guide nearby pixels for further matching, and new potential matches are accepted carefully according to a matching cost function. In this way, more and more new correspondences are constructed, and matching result becomes denser and denser. Sometimes only one seed is enough to expand to the whole image. However, region growing will stop when comes across object boundaries or occlusions, or the geometric distortion is very severe. Therefore, sufficient properly located seeds are required to prevent bad propagation.

Affine normalization

In the wide baseline situation, there exists significant geometric distortion which bring about inaccuracy when performing dense matching. In Figure 4, we search for the correspondence of Y by taking the seed match $X \leftrightarrow X'$ as reference; Y' is a wrong location due to geometric deformation, and the true match cannot be found in the surrounding pixels of Y'. So, before conducting region growing, we need to normalize the local patches to prevent bad propagation.

We assume that the geometric distortion caused by viewpoint variations changes gradually on a smooth

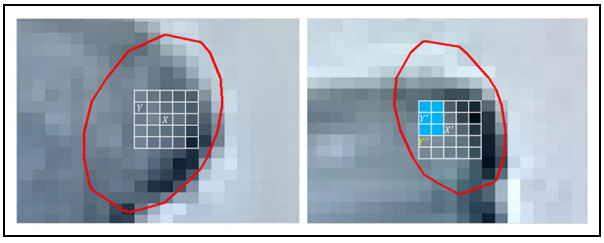

Figure 4. Example on affine deformation. $X \leftrightarrow X'$: a seed match; the red ellipse represents the associated affine invariant region. When search for correspondence of Y, Y′ is a wrong location due to affine deformation, and Y″ is correct.

surface, and it can be approximated by local affine model. For a reliable seed match, for example, $X \leftrightarrow X'$, the associated affine transformation matrix A can be used to normalize the geometric deformations of the local image region. This procedure is illustrated in Figure 1. First, a square patch P centered at X is extracted from image I_1; the four vertices of the square are denoted as "V_1, V_2, V_3, V_4." For each vertex, we transform to its corresponding location "V'_1, V'_2, V'_3, V'_4" in I_2 using A, that is

$$V'_i = X' + A^{-1}(V_i - X), \quad i = 1, 2, 3, 4 \tag{9}$$

Second, the corresponding quadrilateral P′ in I_2 is extracted. To increase the accuracy, we double the size of patches P, P′ by linear interpolation. Then, we sample these two patches to the original size. Thus, match propagation can be performed on the normalized patches. Figure 5 demonstrates some corresponding patches, where the second one is patch after affine normalized in the target frame. After normalization, corresponding local patches are almost identical if they are true matches.

Procedure of dense matching

We perform affine invariant dense matching in the way of region growing. The basic idea is to propagate reliable matching from low ambiguous regions to high ambiguous regions. Local affine invariant sparse matching is conducted first to produce seed matches for region growing. As a reliable affine match can give an initial guess of approximate disparity of their neighboring pixels, true match then can be searched from pixels adjacent to this predicted location. Matching is conducted by calculating the matching costs of normalized patches. The one with the lowest cost is selected, and if the cost is below a threshold T, then it is accepted as a new match.

When performing region growing, reliable and distinctive pixels should be propagated first, for example, feature points are the most distinctive ones, so they are matched at the beginning. During propagation, we decide the priority of seeds with regard to its distinctiveness and robustness. The distinctiveness s(p) of pixel p is defined as the color difference of its neighboring pixels. Let r_p, g_p, and b_p be the colors of a pixel p

$$\begin{aligned} n(p,q) &= 0.299|r_p - r_q| \\ &+ 0.587|g_p - g_q| + 0.114|b_p - b_q| \\ s(p) &= \text{mean}\{n(p,q), q \in N_2(p)\} \end{aligned} \tag{10}$$

Then, the priority of a seed match $p \leftrightarrow p'$ is

$$\text{prior}(p) = 0.5*(s(p) + s(p'))*(1 - \text{Cost}(p,p')) \tag{11}$$

In each loop, the match with highest priority is chosen for propagation. This suppresses propagation from unreliable matches or too uniform regions.

Region growing is based on the assumption that disparities and affine transformation change gradually on

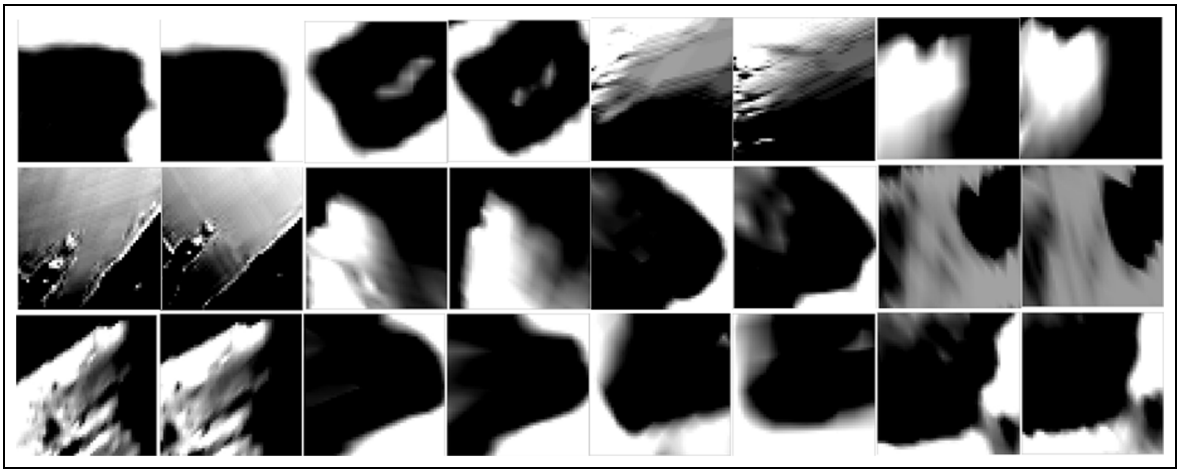

Figure 5. Corresponding patches after affine normalization.

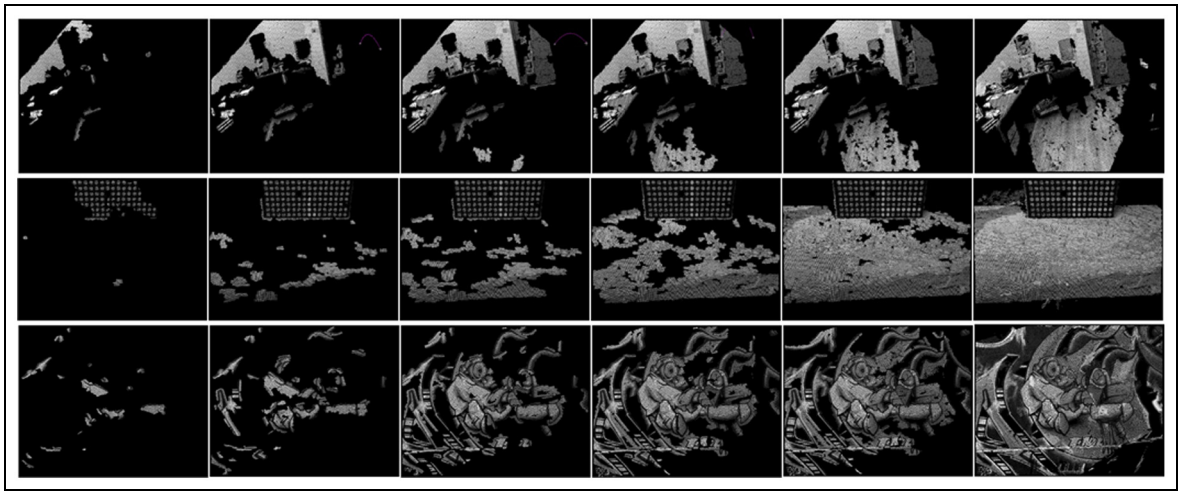

Figure 6. Examples of region growing procedure for match propagation. Black areas are unmatched regions.

smooth surfaces. However, when matching propagates through a long distance, errors will accumulate and the affine transformation inherited from the initial seeds become more and more inaccurate. So, during propagation, this matrix should be updated once a new match is accepted. Suppose the affine matrix obtained from the last iteration is 2×2 matrix A_0, A_0 is expected to be updated so that it can well reflect the affine transformation at the newly accepted match. This is a nonlinear optimization in the four-parameter search space

$$A_1 = \underset{A_0 + \delta}{\operatorname{argmin}} \operatorname{Cost}(p_1, p_2) \tag{12}$$

$\operatorname{Cost}(p_1, p_2)$ is matching cost of the newly accepted match, refer to equation (5). We exhaustively search the parameter domain in discrete steps to find the optimal affine transformation matrix. In this way, both the disparities and the affine transformation matrix are propagated during region growing. Through iterative propagation, matching becomes denser and denser. Figure 6 demonstrates some examples of region growing procedure for match propagation.

The procedure of match propagation terminates when there is no neighboring pixel to process. The dense matching procedure is presented in Algorithm 1. The input is a set of sparse matching correspondences, and the output is a quasi-dense matching Map.

Algorithm 1: algorithm for dense matching

Input: Images $\mathbf{I}, \mathbf{I}'$, *Sparse_correspondences* (location $\mathbf{x} \leftrightarrow \boldsymbol{x}'$, affine transformation matrix $\mathbf{A}$)
Output: *Map*
Map : $= \varnothing$
Seed : $=$ *Sparse_correspondences*
Compute *prior* for each match
While *Seed* $\neq \varnothing$ **do**
Draw the match with the highest *prior* from *Seed*, $s_0 := \{p_0 \leftrightarrow p'_0, \boldsymbol{A}_0, prior(p_0)\}$
Construct affine normalized patches P, P' centered at p_0, p'_0 with Eq.(9)
Region growing:
For each $p_1 \in N_2(p_0)$ and $p_1 \notin Map$
Compute its corresponding location in $\mathbf{I}'$: $p'_{1 0}$
For each potential match $p'_{1i} \in N_2(p'_{1 0})$
compute $Cost(p_1, p'_{1i})$ as Eq.(5)
End **for**
Select $p'_1 = argmin_{p_{1i}'} Cost(p_1, p'_{1i})$
If $p'_1 \notin Map$ and $Cost(p_1, p'_1) < T$, **then**
accept $p_1 \leftrightarrow p'_1$ as a new match
Update $\boldsymbol{A}_0$ to $\boldsymbol{A}_1$ with Eq.(12)
Compute $prior(p_1)$ with Eq.(11)
$s_1 := \{p_1 \leftrightarrow p'_1, \boldsymbol{A}_1, prior(p_1)\}$
$Map := Map \cup s_1$
End **if**
End **for**
End **while**
Return *Map*

Handling unmatched points

Many applications (e.g. 3D reconstruction) require dense matching result, but there remain many unmatched points in the previous quasi-dense matching Map. The "black holes" (in Figure 9(a)) are caused by two reasons: one is the low local patch similarity due to severe deformation and background interference; the other is unexpected quit of propagation. These points are scattered in the image and surrounded by those matched points. The "holes" may lead to defects in the following applications, such as manufacturing flaws and incompleteness in 3D models. Based on the character of obtained quasi-dense disparity map, we utilize low-rank matrix recovery technique to handle unmatched points and construct complete dense matching result.

Compressive sensing is a research hotpot in image processing and data analysis. According to the compressive sensing theory,[27] if the original data are low rank, and the error is sparse, then the corrupted data can be automatically correctly recovered. Low-rank matrix reconstruction[28,29] is a typical application of this theory.

Suppose the original data are $A \in R^{n_1 \times n_2}$, with rank $r \ll \min(n_1, n_2)$, and the corrupted data are X, with sparse error E, that is, $X = A + E$. Then, the original data A can be recovered from X by solving the following optimization problem

$$\min_{A,E} \text{rank}(A) + \gamma \|E\|_0, \quad \text{s.t. } A + E = X \tag{13}$$

where γ is a weighting parameter trades off the rank and sparsity of the recovered data. This problem is NP-hard, but we can transfer it to a convex surrogate under mild conditions

$$\min_{A,E} \|A\|_* + \lambda \|E\|_1, \quad \text{s.t. } A + E = X \tag{14}$$

where $\|\cdot\|_*$ is the nuclear norm and λ is a weighting parameter. Equation (14) can be solved by an augmented Lagrange multiplier algorithm.[30]

In image processing, corrupted image (X) can be decomposed to a low-rank term (A) and a sparse term (E). The original image data are required to have low rank, such as regular textures or static background in video. The sparse term often represents for errors, for example, shadows and blurs, and they are large in magnitude but sparse in spatial domain. In our problem, the wide baseline images do not always have low-rank character, so it will take risk to conduct sparse decomposition directly upon it. But according to the disparity smoothness constraint, disparities of pixels on smooth planes do not change quickly, so the disparity maps often have low rank. Based on this character, we consider disparity maps as original data, and unmatched points ("black holes" in Figure 9(a)) as errors, which are sparse in spatial domain; then, the previous quasi-dense matching result is the corrupted data need to be recovered.

Suppose $x_1 \leftrightarrow x_2$ is a pair of matched points in I and I′, then the disparity is $d = (dx, dy)^T = x_2 - x_1$.We use two disparity matrixes dX, dY to represent the quasi-dense matching result of section "Procedure of dense matching." Thus, the accurate dense matching result dX_0, dY_0 and matching error E_X, E_Y can be determined by sparse decomposition. The optimization problem can be constructed as follows

$$\min_{dX_0, E_X} \|dX_0\|_* + \lambda \|E_X\|_1, \quad \text{s.t. } dX_0 + E_X = dX \tag{15a}$$

$$\min_{dY_0, E_Y} \|dY_0\|_* + \lambda \|E_Y\|_1, \quad \text{s.t. } dY_0 + E_Y = dY \tag{15b}$$

In this way, unmatched points are handled and "black holes" are repaired. Thus, the dense matching procedure is completed.

Experimental results and analysis

In this section, we test the proposed approach on real images. In section "Sparse matching step," we show effectiveness of our sparse matching algorithm. In the second experiment (section "Dense matching step"), dense matching results of wide baseline images are demonstrated.

We implemented the proposed method in MATLAB and tested it on a laptop running Windows 7 with Intel Core i5 central processing unit (CPU) and 6-GB RAM. The images are collected from public dataset (e.g. INRIA on the web) and from our laboratory dataset. The latter are obtained by handheld cameras without calibration. Geometric distortion is significant in the test images, including large displacement, rotation, scaling, and some viewpoint changes. For evaluation, we designed some controlled experiments and calibrated the cameras used indoor; therefore, we can compute the depth map of the indoor scenes. Figure 7 lists some image pairs taken by a calibrated binocular vision system.

Sparse matching step

The first step of our method is to extract and match distinctive feature points. We extract the Hessian-Affine feature for sparse matching. SIFT-like feature vectors are computed in the affine invariant regions of each key point. Initial matching is conducted by matching these features with the nearest-neighbor strategy. However, compared with the number of feature points (e.g. in Figure 2, 1997 feature points in the left image and 1699 in the right), the number of matching points is relatively small (299 matches). So, we need to generate more seed matches for dense matching. We perform an experiment to show parameter influence in sparse matching. It reflects the effects of threshold T (matching cost) and D (region size) in finding new matches, which have both effects on matching number and accuracy. So, we need to get a balance by setting these two thresholds. Figure 8 gives matching number and accuracy with respect to D and T. The upper limit of the matching cost is set as $T = 0.2, 0.25, 0.3, 0.35,$ and 0.4 (the cost is regularized to 0–1), and the region size is set as $D = 5, 6, 7,$ and 8 pixels. Accuracy is shown as the ratio between true matches and total matching number. False matches are picked out manually.

For comparison, the original sparse matching result is also presented. Figure 8 shows that when T increases, the matching number increases and accuracy decreases,

Figure 7. Sample images taken by a calibrated binocular vision system. Top row: images taken by the left camera. Bottom: images taken by the right camera. Each column corresponds to an image pair.

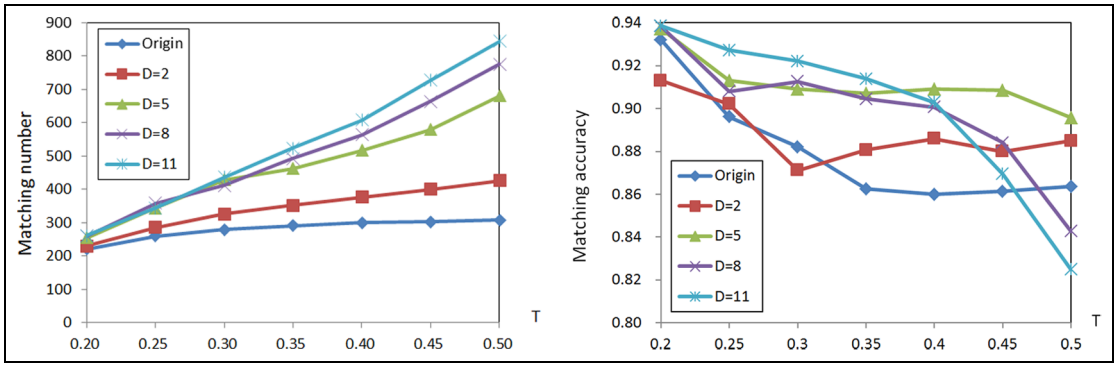

Figure 8. Matching number and accuracy with different parameters.

so we need to make a compromise. When D = 5, the accuracy is more stable; and the accuracy drops rapidly when T>0.4. So, in our experiment, we choose T = 0.4 and D = 5 to keep the optimal balance between matching number and accuracy. It is obviously shown that compared with the original matching result, the matching number is significantly increased, and the accuracy is also improved. Thus, the sparse matching is both reliable and sufficient to be used for dense matching in the next step. Although epipolar constraint can be used to eliminate mismatches[31] for the rigid scene, it is not indispensable. In the region growing framework, matching is propagated from the best match at each loop; thus, matches with low confidence have little impact on dense matching.

Dense matching step

In this subsection, we illustrate the proposed dense matching algorithm with the example of an indoor scene. For this purpose, we work with the uncalibrated images shown in Figure 2. First, 564 seed matches are obtained in the sparse matching step. Each pair of match is associated with an affine transformation matrix to describe the local structure.

During dense matching, the threshold for accepting a new match is set to T = 0.2; matches with cost above this threshold will be rejected. The propagation starts from sparse feature points, and the matched region grows gradually; in each loop, matching is propagated from the best matching point, so mismatches in previous steps are less likely to generate new matches and will not bring about great deterioration in subsequent matching. The computation time for generating a new match is 19.3 ms.

Finally, we utilize the low-rank matrix recovery technique to handle unmatched points. The quasi-dense matching result (Figure 9(a)) obtained from region growing is regarded as corrupted data. We take the disparity map in X and Y dimension as input, respectively, and then conduct low-rank and sparse decomposition. The rank of the recovered disparity matrix is 406, which

Figure 9. Comparison results (a) before and (b) after low-rank matrix recovery.

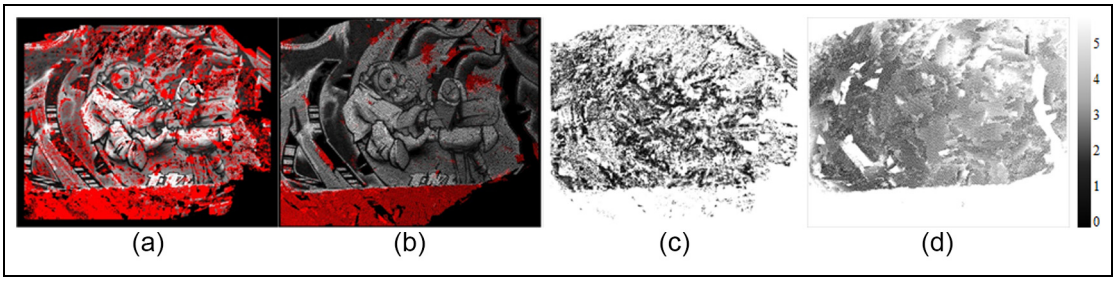

Figure 10. Matching result from the "graff" image pair: (a and c) obtained with RS method and (b and d) obtained with our method. (a) and (b) are dense matching result, and red pixels represent inaccurate matches. (c) and (d) are Sampson distance, in which bright pixels represent positions with high Sampson error and dark means low error.

is much lower than the original rank of 570. In the end, the dense matching result (Figure 9(b)) is obtained, where most unmatched points are eliminated. This result is much better than Figure 9(a).

We compared our method (denoted AF) with a recent approach,[32] which is a rotation and scale invariant dense matching algorithm (denoted RS). For an image pair, we choose the right one as reference image and the left one as target image and then put the corresponding pixel in the reference image on the target image, as illustrated in Figure 10. Compared with the RS approach, the proportion of good matches is higher in AF case. Images with known homography are used for evaluation. To examine the matching result, we compute the Sampson distance with ground truth obtained from known homography.[31] In Figure 10, the matched pixels are colored according to their Sampson distance. Those inaccurate matches (with Sampson distance above 5) are in red color and the occlusions are in black (see Figure 10(a) and (b)). The number of good matches (Sampson distance below 5) is 235,347 for RS method and 251,506 for AF. In Figure 10(c) and (d), the pixel intensities represent the Sampson error; black area means low error and white area means high error. The values of Sampson distance over 5 are suppressed

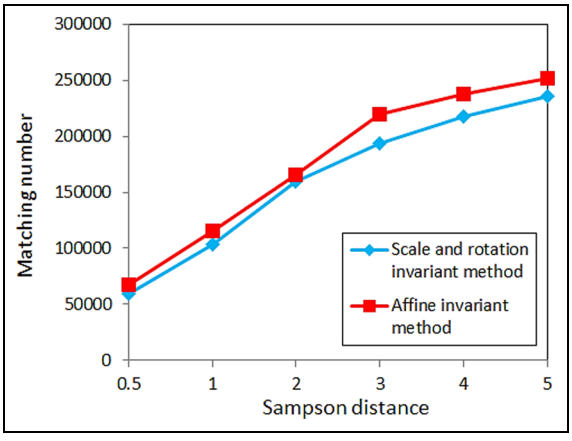

Figure 11. Matching number under several Sampson values.

to 5, with intensity value equal to 1.0. Figure 11 demonstrates that the matching result of AF is denser and more accurate than RS method. The reason is that we use Hessian-Affine features to perform sparse matching and conduct dense matching in an affine invariant way, while RS method has only rotation and scale invariance. To gain invariance in wide baseline matching, local image patches are required to be normalized to eliminate the inaccuracy introduced by geometric

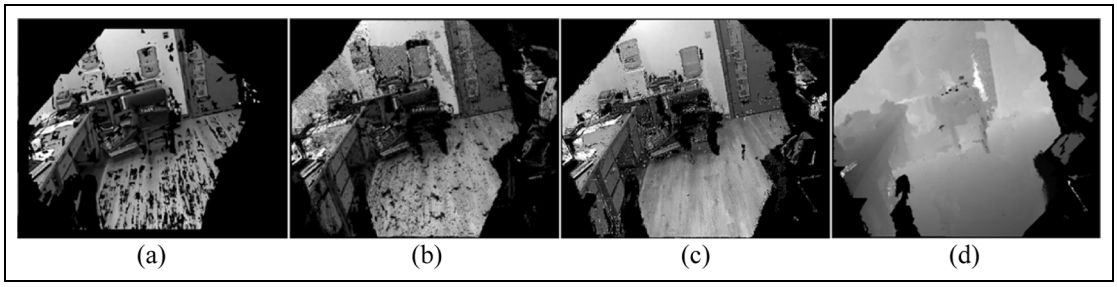

Figure 12. Comparisons of our approach with other methods: (a) RS method,[32] (b) KB method,[18] (c) our method, and (d) depth map.

Table 1. Comparing the matching speed of our approach with other methods.

	RS method[32]	KB method[18]	Our method
Average computation time per match (ms)	8.2	8.6	18.5

deformation. The affine transformation matrix in this article has four parameters, versus two parameters in RS, so our approach can describe the local geometric deformation more accurately. Thus, the dense matching result is more reliable.

We also compare our method with Kannala and Brandt's[18] quasi-dense matching algorithm, which is denoted as KB. It is shown in Figure 12 that our dense matching result is much denser. One reason is that in KB approach, propagation often stops at object boundaries during region growing, and the absence of reliable seed points leads to unmatched regions. If no match or only poor sparse matches are found in a region, this region may be lost in dense matching, because no new reliable correspondences could be established due to the poor initial seeds. To solve this problem, we present a method to generate sufficient seed matches based on the original sparse matching result. Thus, the seeds can cover more regions and the matching result is denser. In addition, matching in KB is not conducted in low distinctive regions, for example, the black area between two buckets (see Figure 12(b)). In our approach, when performing region growing, a propagating order is set according to the distinctiveness of seed matches; we first match high distinctive regions and then match low distinctive regions, using the matched points to reduce matching ambiguity. Finally, as an application, we calibrated the cameras and computed a depth map of the scene, which is shown in Figure 12(d). Moreover, Table 1 gives the comparison in matching speed with other methods. From Table 1, we can see that the running time of the proposed method is acceptable. More results can be found in the supplementary file.

Conclusion

In this article, we proposed an affine invariant approach for dense wide baseline matching. The significant geometric deformation in images introduces difficulty in dense matching. We designed a sparse-to-dense framework to handle this problem. In the stage of sparse matching, in order to get sufficient seed matches, we incorporate initial matching points as spatial prior to produce more correspondences. Dense matching is conducted by region growing, wherein matching extends from high distinctive regions to low distinctive regions, and finally completed by low-rank matrix recovery technique. The proposed approach has affine invariance and can deal with images obtained from handheld cameras without calibration. Experimental results demonstrate that the proposed approach is effective to obtain dense and reliable correspondences.

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 is supported by National Natural Science Foundation of China under grant no. 61175014 and no. 51209134.

References

1. Soro S and Heinzelman W. A survey of visual sensor networks. *Adv Multimed* 2009; 2009: 640386.

2. Scharstein D and Szeliski R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int J Comput Vision* 2002; 47(1): 7–42.
3. Kolmogorov V and Zabih R. Multi-camera scene reconstruction via graph cuts. In: Heyden A, Sparr G, Nielsen M, et al. (eds) *Computer vision: ECCV'2002*. Berlin: Springer, 2002, pp.82–96.
4. Sun J, Zheng NN and Shum HY. Stereo matching using belief propagation. *IEEE T Pattern Anal* 2003; 25(7): 787–800.
5. Min D, Lu J and Do MN. A revisit to cost aggregation in stereo matching: how far can we reduce its computational redundancy? In: *Proceedings of the 2011 IEEE international conference on computer vision (ICCV)*, Barcelona, 6–13 November 2011, pp.1567–1574. New York: IEEE.
6. Gallup D, Frahm J-M, Mordohai P, et al. Real-time plane-sweeping stereo with multiple sweeping directions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR'2007)*, Minneapolis, MN, 17–22 June 2007, pp.2110–2117. New York: IEEE.
7. Strecha C, Hansen W, Gool LV, et al. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR'2008)*, Anchorage, AK, 23–28 June 2008. New York: IEEE.
8. Tuytelaars T and Mikolajczyk K. Local invariant feature detectors: a survey. *Found Trend Comput Gr Vis* 2007; 3(3): 177–280.
9. Lowe D. Distinctive image features from scale-invariant keypoints. *Int J Comput Vision* 2004; 60: 91–110.
10. Mikolajczyk K and Schmid C. Scale & affine invariant interest point detectors. *Int J Comput Vision* 2004; 60(1): 63–86.
11. Bay H, Ess A, Tuytelaars T, et al. Speeded-up robust features. *Comput Vis Image Und* 2008; 110: 346–359.
12. Morel JM and Yu G. ASIFT: a new framework for fully affine invariant image comparison. *SIAM J Imag Sci* 2009; 2(2): 438–469.
13. Mikolajczyk K and Schmid C. A performance evaluation of local descriptors. *IEEE T Pattern Anal* 2005; 27(10): 1615–1630.
14. Tola E, Lepetit V and Fua P. DAISY: an efficient dense descriptor applied to wide-baseline stereo. *IEEE T Pattern Anal* 2010; 32(5): 815–830.
15. Hassner T, Mayzels V and Zelnik L. On SIFTs and their scales. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR'2012)*, Providence, RI, 16–21 June 2012, pp.1522–1528. New York: IEEE.
16. Otto G and Chau T. "Region-growing" algorithm for matching of terrain images. *Image Vision Comput* 1989; 7: 83–94.
17. Fraundorfer F, Schindler K and Bischof H. Piecewise planar scene reconstruction from sparse correspondences. *Image Vision Comput* 2006; 24(4): 395–406.
18. Kannala J and Brandt SS. Quasi-dense wide baseline matching using match propagation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR'2007)*, Minneapolis, MN, 17–22 June 2007, pp.2126–2133. New York: IEEE.
19. Koskenkorva P, Kannala J and Brandt SS. Quasi-dense wide baseline matching for three views. In: *Proceedings of the 2010 20th international conference on pattern recognition (ICPR'2010)*, Istanbul, 23–26 August 2010, pp.806–809. New York: IEEE.
20. Lhuillier M and Quan L. A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE T Pattern Anal* 2005; 27(3): 418–433.
21. Megyesi Z and Chetverikov D. Affine propagation for surface reconstruction in wide baseline stereo. In: *Proceedings of the 17th international conference on pattern recognition (ICPR 2004)*, Cambridge, 23–26 August 2004, pp.76–79. New York: IEEE.
22. Lin WY, Liu SY, Matsushita Y, et al. Smoothly varying affine stitching. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR'2011)*, Colorado Springs, CO, 20–25 June 2011. New York: IEEE.
23. Liu C, Yuen J and Torralba A. SIFT flow: dense correspondence across scenes and its applications. *IEEE T Pattern Anal* 2011; 33(5): 978–994.
24. Kim J, Liu C, Sha F, et al. Deformable spatial pyramid matching for fast dense correspondences. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR'2013)*, Portland, OR, 23–28 June 2013. New York: IEEE.
25. Yang H, Lin W and Lu J. Daisy filter flow: a generalized discrete approach to dense correspondences. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR'2014)*, Columbus, OH, 23–28 June 2014. New York: IEEE.
26. Tau M and Hassner T. Dense correspondences across scenes and scales. *IEEE T Pattern Anal* 2016; 38(5): 875–888.
27. Donoho DL. Compressed sensing. *IEEE T Inform Theory* 2006; 52(4): 1289–1306.
28. Candes EJ, Li XD, Ma Y, et al. Robust principal component analysis? *J ACM* 2011; 58(3): 1–37.
29. Liang X, Ren X, Zhang ZD, et al. Repairing sparse low-rank texture. In: Fitzgibbon A, Lazebnik S, Perona P, et al. (eds) *Computer vision: ECCV'2012*. Berlin: Springer, 2012, pp.482–495.
30. Lin Z, Chen M, Wu L, et al. *The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices*. UIUC technical report UILUENG-09-2214, October 2010. Champaign, IL: University of Illinois at Urbana–Champaign.
31. Hartley R and Zisserman A. *Multiple view geometry in computer vision*. Cambridge, MA: Cambridge University Press, 2004.
32. Gao J and Shi FH. A rotation and scale invariant approach for dense wide baseline matching. *Lect Notes Comput Sci* 2014; 8588: 345–356.