Structured-Patch Optimization for Dense Correspondence

Xiameng Qin, Jianbing Shen, *Senior Member, IEEE*, Xiaoyang Mao, Xuelong Li, *Fellow, IEEE*, and Yunde Jia

***Abstract*—This paper presents a new method to compute the dense correspondences between two images by using the energy optimization and the structured patches. In terms of the property of the sparse feature and the principle that nearest sub-scenes and neighbors are much more similar, we design a new energy optimization to guide the dense matching process and find the reliable correspondences. The sparse features are also employed to design a new structure to describe the patches. Both transformation and deformation with the structured patches are considered and incorporated into an energy optimization framework. Thus, our algorithm can match the objects robustly in complicated scenes. Finally, a local refinement technique is proposed to solve the perturbation of the matched patches. Experimental results demonstrate that our method outperforms the state-of-the-art matching algorithms.**

***Index Terms*—Dense correspondence, features, match, optimization, structured patch.**

I. Introduction

ESTABLISHING correspondences between image regions is one of the most fundamental tasks in image processing and multimedia applications. Most of the image matching algorithms can be classified as *feature based matching* that obtains reliable correspondences only at a sparse set of key feature points, or *dense matching* that matches all the pixels between two images. As mentioned in [20], the algorithms also can be classified as *local* or *global* in terms of the search range, which means searching in a limited spatial window or in the whole image, respectively. Furthermore, there are many multimedia applications of matching two images, such as image segmentation [16], visual tracking [12], self-similarity [14], visual recognition [50], and image classification [22], [53].

During the searching procedure, many transformations, e.g. color, scale, rotation, and even deformation, should be considered, as well as the search range is also important. Searching in a local window means much faster but less accurate, in contrast, the global searching could be more accurate but much slower. Actually, most existing methods are only taking account of part of these situations. Sparse correspondences are commonly used in many algorithms [27], [11], [15], [51] for its efficiency, and these feature based matching algorithms almost contain all the situations on feature level (e.g. SIFT [8]). Sparse correspondences are also extended for matching multi-view video sequence [47]. However, many multimedia applications [48], [21], [13], [17], [25] inherently require the dense correspondences for achieving the high quality performance. Thus, the sparse features were usually utilized to guide the matching process or as the initialization. Brox and Malik [27] added the sparse feature matching into the classical optical flow framework to handle the arbitrary large displacement motion. Simon and Seitz [15] matched the sparse features and then propagated these correspondences to the whole image. Xu *et al.* [39] expanded the sparse feature correspondences to the candidate motion field, and combined them with the classical optical flow. However, these feature based matching algorithms may be less effective in regions with weak texture [41]. Another important class of feature based matching algorithm is the graph matching methods [19], [42], [33], [40] that have been developed in recent years. All the extracted features in an image are connected to construct a graph, and then the graphs in two images are matched to find the similar objects. In general, graph matching is employed to match two images that contain the objects in the same class but have different structures, shapes, colors and so on.

In the local searching algorithms, the correspondence between two images is often formulated as an estimation of a small displacement, *e.g.*, optical flow estimation [10], [1] and stereo matching [6], [18]. However, large displacement motions of the objects will be lost if we only consider the small translations. Moreover, the matching may be failed if the input image pairs contain different objects, backgrounds, or ever different scenes. In order to overcome this problem, Barnes *et al.* [20] proposed a generalized PatchMatch correspondence algorithm to find dense and global matches by taking account of translation, scale and rotation. The approximate nearest neighbors of every pixel in one image can be found in another

Manuscript received April 15, 2014; revised January 08, 2015; accepted January 17, 2015. Date of publication January 21, 2015; date of current version February 12, 2015. This work was supported in part by the National Basic Research Program of China (973 Program) under Grant 2013CB328805, the National Natural Science Foundation of China under Grant 61272359 and Grant 61125106, JSPS KAKENHI under Grant 25540045, Grant 26240015, Grant 26560006, and Grant 25280037, the Program for New Century Excellent Talents in University under Grant NCET-11-0789, the Key Research Program of the Chinese Academy of Sciences under Grant KGZD-EW-T03, and the Specialized Fund for Joint Building Program of Beijing Municipal Education Commission. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Ebroul Izquierdo. *(Corresponding author: Jianbing Shen.)*

X. Qin, J. Shen, and Y. Jia are with Beijing Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, Beijing 100081, China (e-mail: shenjianbing@bit.edu.cn).

X. Mao is with the Interdisciplinary Graduate School of Medical and Engineering, University of Yamanashi, Kofu 400-8510, Japan.

X. Li is with the Center for Optical Imagery Analysis and Learning, State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, China.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2015.2395078

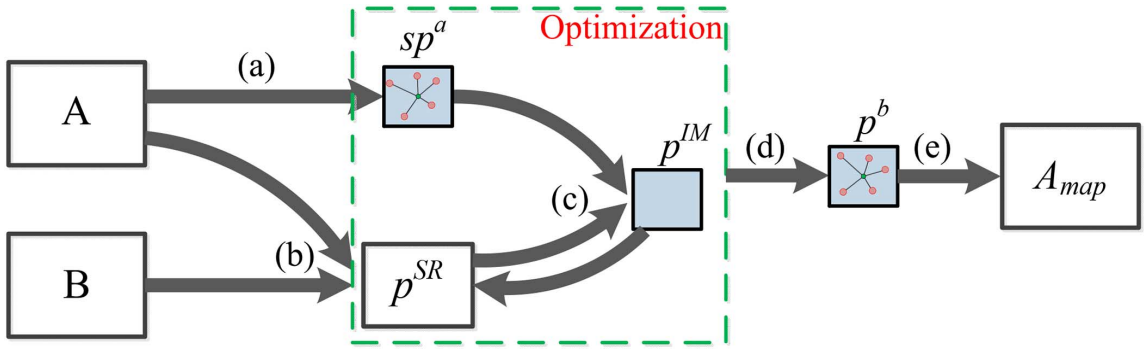

Fig. 1. Workflow of our dense matching algorithm with structured patches. The structured patch (a) is denoted by sp^a, and its corresponding search range (SR) (b) is p^{SR}. The intermediate matched patch (P^{IM}) is generated during the optimization (c) in the green dash box. After the searching in p^{SR}, we get the matched patch p^b (d). In the last step (e), the optimization process is performed for all the patches in the image, and then the local refinement is accomplished to obtain the map A_{map}.

image. Then, Korman and Avidan [32] presented a coherency sensitive hashing (CSH) matching method, which combined locality sensitivity hashing [3] and PatchMatch [20] to find corresponding patches between two images. Olonetsky and Avidan [37] improved PatchMatch by initializing a set of matching patch candidates between the two input images, and then propagated good matches to neighboring patches on the integral image. These two methods are faster and more accurate than PathMatch. However, all of these methods found the nearest neighbors in color level, the intrinsically geometric structure between the pixels was not fully utilized to limit the search range. HaCohen *et al.* [30] added the color bias and gain per channel into the transformation in [20]. Thus, their method can match the two images in different illumination, or even different background. Those patch match algorithms are also combined with the stereo matching problem together [35], [26], [43], where the constraints in stereo matching are still utilized in those combined methods.

In order to match images in complex scenes and find reliable correspondences, many image matching algorithms have been explored. Duchenne *et al.* [29] proposed a graph-matching kernel and found the matches by optimizing the Markov random field (MRF). Korman *et al.* [46] searched the corresponding template from an image for a given template, and the searching process was under 2D affine transformations. Liu *et al.* [34] used the per-pixel SIFT descriptors instead of the pixels, and presented a dense matching algorithm called SIFT Flow. The coarse-to-fine scheme and the conventional pixel-level MRF model were utilized to solve SIFT Flow. Kim *et al.* [45] built a pyramid graph between different layers of the pyramid, and then optimized this graph via belief propagation. However, the pyramid framework has an intrinsic limitation which is the zero motion assumption [41]. The results of this class of algorithms might not be well enough when large displacement motion happens without any prior information on it.

Instead, we propose a new dense matching algorithm using the structured patches in an energy optimization framework. Our new total energy function integrates the sparse features with dense pixels in patches rather than only considering one of them. Then our method does not require to search the transformations, such as color, scale, rotation and etc., explicitly during the optimal procedure. Our algorithm is different from the traditional

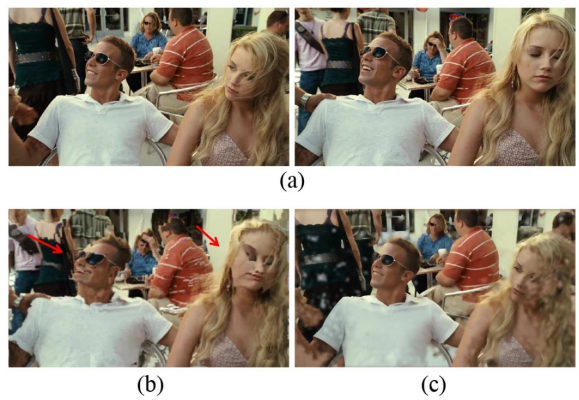

Fig. 2. Illustration of our dense matching approach. (a) is the input image pair. (b) and (c) are respectively the reconstructed image by SIFT flow [34] and our dense correspondence method. The two red arrows indicate the mismatching regions. (a) The image pair. (b) SIFT flow. (c) Our method.

feature based matching algorithms. In our algorithm, the features in patches are utilized to design a new structure to describe the patches. Thus, not only the feature descriptors are used to compute the similarity, but also the geometric structure of features is considered to restrict the matching and to speed up the searching process. Moreover, our structured patches are robust to the regions in the image with non-rigid transformations, or even with significant occlusions. As shown in Fig. 2, the input image pair contains all kinds of transformations. Our approach achieves better visual quality and finds the correct correspondences, while the result by SIFT Flow [34] introduces many mismatches. Another new design of our method is that we consider the dense information, i.e., the pixels and the scene information (e.g., GIST [4]) are integrated into our energy optimization framework to guide the matching process. Since the patches in our approach are constructed by the sparse features, the position of the matched patch might be a little perturbation around the ground truth. Thus, it is better to develop a novel local refinement algorithm to adjust the position of the matched patches.

The flow chart of our dense matching algorithm with structured patches is shown in Fig. 1. To simplify the description, we divide our algorithm into five key steps. (a) A patch from image A is structured to construct the structured patch sp^a. (b) The

map between image A and B is initialized using the scene information, and the corresponding Search Range (SR) p^{SR} is chosen with the patch sp^a. (c) The intermediate matched patch p^{IM} that corresponding with the patch sp^a is searched in the SR p^{SR}. Simultaneously, the SR is reduced according to the feedback of the intermediate matched patch p^{IM}. This optimization process is performed until the termination condition is satisfied. (d) After several iterations, the corresponding patch p^b is obtained. (e) The optimization processes are performed for all the patches in the whole image, and the local refinement is also accomplished to obtain the accurate map A_{map}.

In summary, our approach has the following three main contributions in the context of previous work on dense correspondence.

1) We propose a new dense matching algorithm that considers both the sparse and dense information in a global energy optimization framework.
2) We design a new patch structure by using the sparse features in the patch. The sparse features in the patches constraint a high percentage of approximate patches that will influence the matching.
3) We marry the global scene information and the local features so that we can deal with the deformations and occlusions during the optimizing process.

The rest of this paper is organized as follows. Section II describes our new dense correspondence algorithm with structured patches in detail. And the experimental results are shown and discussed in Sections III. We finally give the summary of the presented algorithm and future work in Section IV. Our source code will be publicly available online.[1]

II. Our Approach

Our dense matching approach simultaneously considers the sparse and dense cues, and the core idea is the iterative optimization procedure in the green dash box in Fig. 1. A new total energy optimization framework is then developed by considering both the sparse and dense information of the patch. The patches are described by the structured sparse features, and the scene information is also integrated into the energy function. Local refinement is designed to guarantee the accuracy of the matched correspondences. Assuming p^a and p^b are the corresponding patches respectively in images A and B. Then the energy between these two patches can be given by

$$E(p^a, p^b) = \eta E_F + \lambda E_P + \mu E_S \tag{1}$$

where E_F is the similarity cost between the structured patches in the corresponding images. The term E_P describes the similarity of the two patches on pixel level, and E_S ensures that the patches are in a similar sub-scene. As can be seen in Eq. (1), we integrate the sparse information with dense information not only in the construction of the structured patches (Section II-A), but also in the total energy function (E_F and $(\lambda E_P + \mu E_S)$) that will be introduced in Section II-B.

A. Structured Patch

Let $F^a = \{f_1^a, \ldots, f_n^a\}$ be the sparse features in patch p^a, and $F^b = \{f_1^b, \ldots, f_m^b\}$ in p^b. The common feature based matching methods only computed the distance between the feature descriptors. In our algorithm, however, our structured patch not only estimates the descriptor difference, but also considers the geometric structure between the features. As shown in Fig. 3, the structured patch integrates the sparse features with the dense information that is represented by the color and spatial position of the pixels in the patch. The dense weight is computed to guide the estimation of the weight among the sparse features.

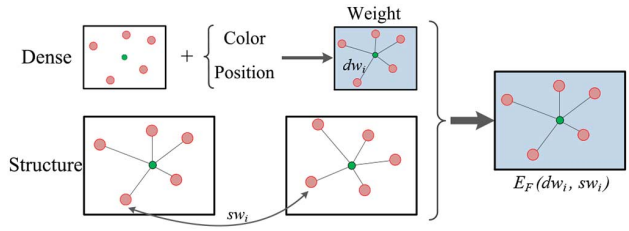

Fig. 3. Illustration of our structured patch which contains both the sparse and dense information. The weight dw_i is computed by the color and spatial position of the pixels in the patch, and the weight sw_i is the similarity between the features in the two patches. The red points represent the sparse features, and the green points indicate the center of the patches.

Dense pixels: Assuming that p_o^a denotes the center pixel of the patch p^a, and p_i^a is the pixel that corresponds to the feature f_i^a, where $i = \{1, \ldots, n\}$. We allocate a weight for every feature (f_i^a) in terms of the color difference and the spatial distance between the center pixel (p_o^a) of the patch and the pixel p_i^a. The weight that considers the spatial distance can be defined using the Laplacian kernel as follows:

$$dw_i^a(\Delta g) = exp\left(-\frac{\Delta g(p_o^a, p_i^a)}{\mathcal{L}_s^a}\right), \tag{2}$$

$$\mathcal{L}_s^a = \sum_{i=1}^{n} \Delta g(p_o^a, p_i^a) \tag{3}$$

where $\Delta g(\cdot,\cdot)$ denotes the spatial distance. The color difference is computed between the neighborhoods of the two pixels, and the corresponding weight is given by

$$dw_i^a(\Delta u) = exp\left(-\frac{\Delta u(\mathcal{N}(p_o^a), \mathcal{N}(p_i^a))}{\mathcal{L}_c}\right) \tag{4}$$

where $\mathcal{N}(p_o^a)$ and $\mathcal{N}(p_i^a)$ are respectively the neighbor pixels of the pixels p_o^a and p_i^a. In most of our experiments, the size of this neighborhood is set as 3×3. $\Delta u(\cdot,\cdot)$ denotes the color difference. $\mathcal{L}_c$ is the maximum color value of the pixel, and we set it as 255. Thus, the total weight of the feature f_i^a, which called *Dense Weight* (dw), can be defined as

$$\begin{aligned} dw_i^a &= dw_i^a(\Delta g) \cdot dw_i^a(\Delta u) \\ &= exp\left(-\left(\frac{\Delta g(p_o^a, p_i^a)}{\mathcal{L}_s^a} + \frac{\Delta u(\mathcal{N}(p_o^a), \mathcal{N}(p_i^a))}{\mathcal{L}_c}\right)\right) \end{aligned} \tag{5}$$

The weight of features f_j^b $(j = \{1, \ldots, m\})$ in patch p^b can be computed in the same way, and can be given by

$$dw_i^b = exp\left(-\left(\frac{\Delta g(p_o^b, p_i^b)}{\mathcal{L}_s^b} + \frac{\Delta u(\mathcal{N}(p_o^b), \mathcal{N}(p_i^b))}{\mathcal{L}_c}\right)\right). \tag{6}$$

[1][Online] Available: http://github.com/shenjianbing/patchoptimization

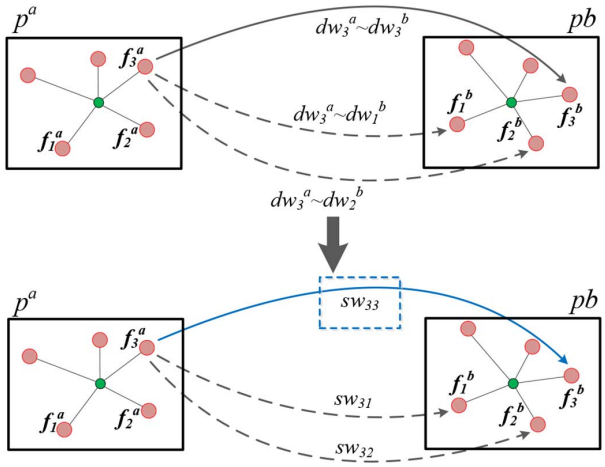

Fig. 4. Procedure to find the reliable corresponding features in the two patches. The dense weights in Eq. (5) and Eq. (6) are employed to estimate the most similar correspondences (N_{dw}), and then find the reliable one from those matches. We set $N_{dw} = 3$ in this figure as an example.

Sparse features: Another part of our structured patch is the similarity computation between the sparse features. However, unlike the traditional matching algorithms, we employ the dense information in Eq. (5) and Eq. (6) to guide the matching. As shown in Fig. 4, we first estimate the similarity of the dense weights between the sparse features in the two patches, and choose the most similar correspondences (N_{dw}) for every sparse feature in p^a. Then, the sparse feature matching is accomplished in these N_{dw} correspondences. Our sparse feature matching algorithm is much faster since we do not compute the distance between all the sparse features.

The similarity of the dense weights is computed by

$$Sdw(i,j) = \Delta Ed(dw_i^a, dw_j^b) \tag{7}$$

where $\Delta Ed(\cdot,\cdot)$ is the Euclidean distance. It is easy to compute this similarity since both dw_i^a and dw_j^b are the 1D variables. Then the N_{dw} most similar correspondences are chosen according to the similarity of the dense weights. We define the N_{dw} features in p^b as the following set:

$$\begin{aligned} Sf^i = \{ & j | j \in [1,m], \exists \alpha \in [0,1], \\ & Sdw(i,j) \le \alpha \max_{j' \in [1,m]} Sdw(i,j'), \\ & numel(Sf^i) = N_{dw} \} \end{aligned} \tag{8}$$

where $numel(\cdot)$ computes the number of the elements in Sf^i. In most of our experiments, $\alpha = 0.85$ and the variable N_{dw} is set as

$$N_{dw} = \left\lceil \frac{1}{2} \min(n,m) \right\rceil. \tag{9}$$

Thus, the reliable correspondences of sparse features can be estimated from the N_{dw} correspondences. The similarity of the sparse features, which indicated as *sparse weight* (sw), is defined as follows:

$$sw_{(i,j)} = \min_{j \in Sf^i} (\Delta d(f_i^a, f_j^b)) \tag{10}$$

where $\Delta d(\cdot,\cdot)$ is the similarity measurement between the feature descriptors, and $\Delta d(f_i^a, f_j^b) = \|f_i^a - f_j^b\|$.

Combining Eq (5), Eq. (6) and Eq. (10), we can compute the energy E_F in Eq. (1) as

$$\begin{aligned} E_F = & \sum_{c \in C(i \sim j)} |\Delta dw_1 - \Delta dw_c| \\ & + \begin{cases} \sum_i^n sw_{(i,j)}, & n \le m \\ \sum_j^m sw_{(j,i)}, & otherwise \end{cases} \\ = & \sum_{c \in [2, Nc]} \|\Delta dw_{i1} - \Delta dw_{j1}| - |\Delta dw_{ic} - \Delta dw_{jc}\| \\ & + \begin{cases} \sum_i^n \min_{j \in Sf^i} (\Delta d(f_i^a, f_j^b)), & n \le m \\ \sum_j^m \min_{i \in Sf^j} (\Delta d(f_j^b, f_i^a)), & \text{otherwise} \end{cases} \end{aligned} \tag{11}$$

where $(i1 \sim j1), (ic \sim jc) \in C(i \sim j)$, Nc is the size of the set $C(i \sim j)$. The set $C(i \sim j)$ contains the corresponding feature pairs in the two patches, which expressed as

$$\begin{aligned} C(i \sim j) = \{ & (i,j) | i \in [1,n], j \in Sf^i, \\ & \Delta d(f_i^a, f_j^b) = \min_{j' \in Sf^i} \Delta d(f_i^a, f_{j'}^b) \} \end{aligned} \tag{12}$$

where i and j are local variables. In $C(i \sim j)$, i and j denote the indexes of the features that are corresponded with each other.

As shown in Eq. (11), the energy E_F integrates the dense pixels with the sparse features. The first term of Eq. (11) contains the dense information, as well as describes the geometric smoothness of the features. The second term computes the similarity of the two sets of features using the guidance of the dense weights.

B. Dense Information

Both the pixels in the patch and scene descriptor are considered as dense information in this paper. Thus, we will introduce them together in this section.

Patch correspondences: The simple way to compute the patch similarity are L2 distance, such as the sum of the squared distance (SSD) in stereo vision [2]. However, these similarity measurements only consider the geometric information and ignore the color information. Instead, we utilize the log-chromaticity normalization to design a matching energy function, which is inspired by the adaptive normalized cross-correlation method for stereo matching in [31].

The 1D vectors of the patches are denoted as

$$\begin{aligned} V &= \{(x_1, y_1), \ldots (x_i, y_j), \ldots (x_h, y_h)\} \\ &= \{ \quad v_1, \quad \ldots v_t, \quad \ldots \quad v_{h^2} \} \end{aligned} \tag{13}$$

where $h \times h$ is the size of the patch, and $t \in [1, h^2]$. (x_i, y_j) represents the 2D coordinates of the pixel (i,j) that corresponds to the 1D coordinate v_t in the patch, where $i, j \in [1,h]$. Assuming that $\vartheta = \{a, b\}$, then the corresponding values of the red channel in the patch can be given by

$$p_{re}^{\vartheta} = \{ p_{re}^{\vartheta}(v_1), \ldots p_{re}^{\vartheta}(v_t), \ldots p_{re}^{\vartheta}(v_{h^2}) \} \tag{14}$$

Thus, the similarity measurement of the patches in the red channel is computed by

$$\begin{aligned}\epsilon_{re\_p} &= \left(\sum_{t\in[1,h^2]}\left[(p^a_{re}(v_t)-\overline{p}^a_{re}(v))\times((p^b_{re}(v_t)-\overline{p}^b_{re}(v)))\right]\right) \\ &\Big/\left(\sqrt{\sum_{t\in[1,h^2]}|p^a_{re}(v_t)-\overline{p}^a_{re}(v)|}^2\right. \\ &\left.\times\sqrt{\sum_{t\in[1,h^2]}|p^b_{re}(v_t)-\overline{p}^b_{re}(v)|}^2\right), \end{aligned} \tag{15}$$

$$\begin{aligned}p^a_{re}(v_t) &= \log\frac{a^a}{\sqrt[3]{a^a b^a c^a}} \\ &+\gamma^a\log\frac{p^a_{re}(v_t)}{\sqrt[3]{p^a_{re}(v_t)p^a_{gr}(v_t)p^a_{bl}(v_t)}},\end{aligned} \tag{16}$$

$$\begin{aligned}p^b_{re}(v_t) &= \log\frac{a^b}{\sqrt[3]{a^b b^b c^b}} \\ &+\gamma^b\log\frac{p^b_{re}(v_t)}{\sqrt[3]{p^b_{re}(v_t)p^b_{gr}(v_t)p^b_{bl}(v_t)}},\end{aligned} \tag{17}$$

where $\overline{p}^{\vartheta}_{re}(v)=\frac{1}{h^2}\sum_{t\in[1,h^2]}p^{\vartheta}_{re}(v_t)$. $p^{\vartheta}_{re}(\cdot)$, $p^{\vartheta}_{gr}(\cdot)$ and $p^{\vartheta}_{bl}(\cdot)$ are respectively the red, green and blue channel values of the patch p^{ϑ}. $a^{\vartheta}, b^{\vartheta}$ and c^{ϑ} are the global scale factors, which are defined by the illumination color, of the corresponding image. The gamma value $\gamma^{(\vartheta)}$ depends on the camera parameters, and we set it as 2.0. Thus, the final patch similarity is defined by

$$E_P=\frac{\epsilon_{re\_p}+\epsilon_{gr\_p}+\epsilon_{bl\_p}}{3} \tag{18}$$

where ϵ_{gr_p} and ϵ_{bl_p} are the similarity measurements of the green and blue channels, respectively.

Scene correspondences: Let s^a and s^b are the scene descriptors that corresponding to the patch p^a and p^b, respectively. Assume that we extract the scene descriptor in the patch p^{ϑ}_s. It is worth to explain that the patch p^{ϑ}_s is much larger than the patch p^{ϑ}. And we choose the patch size as 90×90 to extract the scene descriptor and compute the similarity. Then, the measurement of the scene descriptors can be computed as

$$E_S=\Delta d(s^a,s^b) \tag{19}$$

C. Optimization

To solve the dense matching problem, many optimized structures have been used, such as Markov random field (MRF) model [34], Belief propagation (BP) model [45] and the PatchMatch framework [20]. In terms of two important observations of the total cost in Eq. (1), we design a new structured patch based optimization framework in this sub-section. On one hand, the patches are constructed by sparse features in our algorithm, which means the patch may contain the same sparse features when we move the patch to its neighbor positions. On the other hand, the patches that we use to extract scene descriptors are large, which means that the image could be paved with a few patches without overlap. We can adjust the size of the nearest-neighbor set (NNS) so that we have a chance to search most of the regions of the image in each searching process.

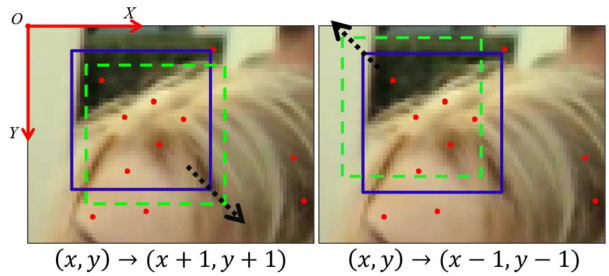

Fig. 5. The same sparse features are contained in the patch even when it moves to the nearest neighbors. The red points are the sparse features. The blue patch is the original patch. *Left*: The direction of bottom right (green patch). The image coordinate system has the origin at the upper left corner. *Right*: The direction of upper left (green patch). The displacement between blue patch and green patch is much larger than 1 pixel for explicit interpretation.

Algorithm 1: Structured patches for dense matching

Input: Image A, B
Output: Correspondences A_{map}

```
for patch = 1 → P do
    Compute the initial size K_0 using Eq. (20);
    /* Adaptive NNS for correspondence search */
    while iteration z < Z do
        Randomly choose K_0 candidate patches in the range [1, W] × [1, H];
        for NNS k = 1 → K_0 do
            Compute the similarity E(p_i^a, p_k^b) between p_i^a and p_k^b using Eq. (1);
        end
        Find the minimum E(p_i^a, p_k^b) for k ∈ [1, K_0];
        if z == 1 then
            W = H = h_s;
            else
            W = H = (1/2)^(z-1) W;
        end
        h_s = h;
        Update K_0 using Eq. (20);
    end
    Obtain the corresponding patch p_k^b of patch p_i^a;
    /* Local refinement */
    Refine the patch p_k^b (Section II-D);
end
```

As shown in Fig. 5, we move the patch to two different directions. The original patch is marked as blue, which is identified by its center pixel $p_c(x,y)$. In our experiments, the image coordinate system has its origin at the upper left corner (Fig. 5). Thus, the green patches can be denoted as $p_c(x+1,y+1)$ (*left* part of Fig. 5) and $p_c(x-1,y-1)$ (*right* part of Fig. 5), respectively. These three patches contain the same sparse features so that the similarity of the patches are barely influenced by the offsets $(x-1,y-1)$ and $(x+1,y+1)$.

Fig. 6. Comparison of the SF [34] (b), deformable spatial pyramid method (DSP) [45] (c), and our method (d). (a) is the input image pairs.

To verify this observation, we randomly choose a set of pixels and translate the match by using the offsets above. More than eighty percent of the patches contains the same sparse features as the translated patches.

The key design of our optimization is the adaptive NNS for searching. The size of the NNS is adaptively adjusted in terms of the patch size and the search range. Let $H \times W$ denote the resolution of the image, where $h_s \times h_s$ denotes the size of the patch that we used to extract the scene descriptor. As mentioned in Section II-B, h_s is set as 90 in most of our experiments. According to the aforementioned observations, we compute the initial size of the NNS as follows:

$$K_0 = \lfloor \frac{W}{h_s} \rfloor \cdot \lfloor \frac{H}{h_s} \rfloor \tag{20}$$

During the iteration, the size of the NNS will be updated according to the size of the patch and the search range. The detailed description is listed from the 4th row to the 17th row in Algorithm 2.3. It is obvious that the optimization procedure is in the computational cost of $O(ZK_0)$, where Z and K_0 are constants. Thus, the computational complexity of our approach is $O(ZK_0HW)$.

D. Local Refinement

As described in Section II-C, in most cases, the similarity measurement of the patches may not be changed if the patch is translated using the offsets (1,1) and $(-1,-1)$ (Fig. 5). This phenomena can help us to redesign the optimization procedure and improve the accuracy of the corresponding patches. As shown in Fig. 5, the final correspondence might be anyone of the three patches $(p_c(x,y), p_c(x-1,y-1)$ and $p_c(x+1,y+1))$. To deal with this problem, we perform a local refinement process around the correspondence. A patch set $\mathcal{N}(p^b)$ contains the neighbor patches around the patch p^b, which defines the refined position of patch p^b. Assuming the correspondence is $p^b(x,y)$ before local refinement, the neighbor patches $\mathcal{N}(p^b)$ can be given by $p^b(x+x_v, y+y_u)$, where $\{(x_v,y_u)|x_v \in Z, y_u \in Z, (x_v,y_u) \in [-2,2] \times [-2,2]\}$. Thus, the update of patch positions is accomplished quickly by minimizing the following energy function (18):

$$\min_{q \in \mathcal{N}(p^b)} E_P(p^a, q) \tag{21}$$

Finally, we summarize our matching method in Algorithm 1.

III. Experimental Results

In this section, the proposed dense matching algorithm with Structured-Patch optimization (SP) is performed on a variety of different image pairs. We compare with several state-of-the-art techniques of dense matching, PatchMatch (PM) [20], SIFT Flow (SF) [34] and deformable spatial pyramid method (DSP) [45], using the implementation from the authors' homepage. In order to obtain a relatively fair comparison of correspondence quality with these approaches, the optimal parameter settings are used to run the above algorithms to produce the final matching results as good as possible in our comparison experiments. The comparisons are performed on two challenging datasets: the Video Pair dataset [32] and Caltech-101 [7]. The main parameters of our method are setting as follows. The patch size h is set as 11×11 for the resolution 640×480 in our experiments, and h_s is set as 90. The setting of other parameters is mentioned where they have introduced before.

A. Comparison on Video Pair Dataset

We first compare our algorithm with SIFT Flow (SF) [34] and the deformable spatial pyramid method (DSP) [45] on the Video Pair dataset [32]. As shown in Fig. 6, we choose five pairs of images to demonstrate the effectiveness of our algorithm. The two input images are matched to find the corresponding map, and then a source image [the first column in Fig. 6(a)] is reconstructed to show the performance of different algorithms. These

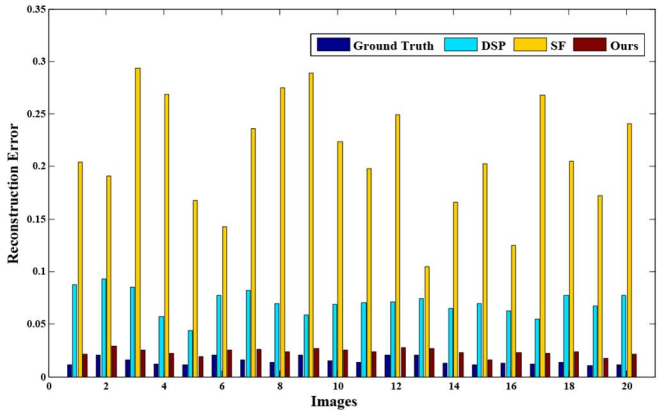

Fig. 7. Comparison of the reconstruction error by DSP [45], SF [34], and our method on the Video Pair dataset. We also compute the ground truth by the exactly dense matching method. In the histogram, twenty pairs of images are selected to estimate the reconstruction error as examples. Actually, more images have been chosen to demonstrate the accuracy of the our approach.

TABLE I
COMPARISON OF AVERAGE RECONSTRUCTION ERROR

	Ground Truth	SF	DSP	Ours
Error	0.015	0.297	0.075	0.024

images contain all the transformations such as translation, rotation, scale and deformation. It is obvious that the corresponding patches that have been matched by our algorithm contain all the transformations in Fig. 6(d). Since the spatial pyramid method has the limitation for handling the images with large displacement, many correspondences are mismatched in Fig. 6(b) and Fig. 6(c).

The reconstruction result of SIFT Flow also has severe uncorrect corresponding patches in the large displacement regions in the first and third rows of Fig. 6(b). In contrast, our method achieves more accurate reconstruction correspondence with more details as shown in Fig. 6(d).

In order to quantitatively evaluate the performance of our method, we compute the reconstruction error of the aforementioned three methods. The ground truth mapping data is directly taken from [32], where they used the exactly dense matching method to generate them. The reconstruction error is the Root-Mean-Square Error (RMSE) between the original and reconstruction images. We use I_O and I_R to denote the original image and the reconstruction one respectively, where the image contains N_p pixels. Then the reconstruction error is computed as follows:

$$R_{err} = \frac{1}{N_p}\sqrt{\sum_{i}^{N_p}(I_O(i) - I_R(i))^2}. \tag{22}$$

The comparison statistics of the reconstruction error is shown in Fig. 7. Twenty pairs of images are randomly chosen to compute the reconstruction error, and those reconstruction errors are shown as the histograms in Fig. 7. It is clear that the reconstruction error of our algorithm is more close to the ground truth, and it achieves the most accurate correspondence results. The reconstruction errors of histograms by SF [34] are much

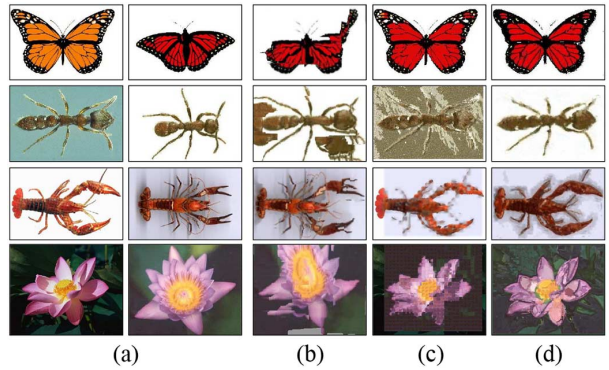

Fig. 8. Comparison results of different methods. (b) SF [34], (c) PM [20], and (d) our dense matching method. (a) is the input image pairs. Best viewed on high-resolution display.

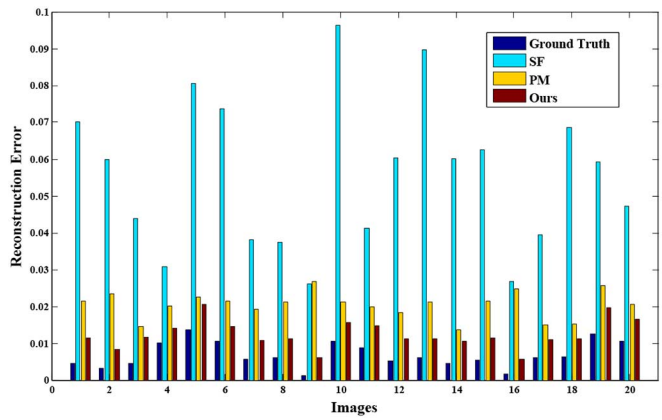

Fig. 9. Comparison of the reconstruction error by SF [34], PM [20], and our method on the dataset Caltech-101.

TABLE II
COMPARISON OF AVERAGE RECONSTRUCTION ERROR

	Ground Truth	SF	PM	Ours
Error	0.0069	0.0556	0.0204	0.0124

higher than those results by DSP [45] and our method. On one hand, the reconstruction image I_R by SF [34] is similar to the image B where A and B are the pair of input images. For instance, in the first row of Fig. 6(b), the similarity between image B and I_R is obvious, especially the background of the scene. And in the second row of Fig. 6(b), the ghost artifacts are introduced near the position of the man in the image I_R. This phenomenon exists in most of the images. On the other hand, although both SF and DSP methods estimate the correspondences in a coarse-to-fine scheme, DSP combines several layers of the pyramid to build a spatial pyramid graph. Thus, DSP can process larger displacement than SF. However, only three layers were utilized so that the common intrinsic limitation of pyramid framework still exists. In contrast, the reconstructed image by our approach achieves better visual quality and higher accuracy than the results by previous approaches. The average reconstruction errors are listed in Table I. We average the reconstruction errors of over thirty pairs of images that randomly selected from the Video Pair dataset. Our dense correspondence approach achieves the smallest reconstruction errors

Fig. 10. Comparison of color transfer using PM [20] (b), SF [34] (c), and our method (d). (a) is the input image pairs.

that are closely to those produced using the ground truth mapping, while the errors by both SF and DSP methods are much higher than those estimated using the proposed approach.

B. Comparison Results on Caltech-101

More challenging correspondence results are shown in Fig. 8, which compare the SIFT Flow (SF) [34] and PatchMatch (PM) [20] with our algorithm on the dataset of Caltech-101 [7]. Similar to the experiments in Sec. III-A, we first match the image pairs and then reconstruct the images in the first column of Fig. 8(a). As shown in Fig. 8(c), the result of PM looks fine in some situations. However, comparing with our results, the texture details of the results by PM are lost and inaccurate. Fig. 8(b) lists the reconstruction results by SF where the mismatches are obvious since the large displacement limitation of SF. Nevertheless, we also find that some local regions of the result by SF are smoothness because of the MRF connections between nearby pixels. In contrast, as shown in Fig. 8(d), our algorithm can handle the problems that could not be solved by the two methods above, and can obtain more accurate and pleasing results. In order to further quantitatively compare the reconstruction results, we randomly select twenty image pairs from the dataset to estimate the reconstruction errors. The comparison is shown by the histograms in Fig. 9, and the reconstruction errors by SF are higher than those produced by PM and our algorithm. This can be well interpreted with the following two reasons. Firstly, the mismatches in the results by SF [Fig. 8(b)] are much more than those in the results by PM [Fig. 8(c)]. In the first row of Fig. 8(b), half of the wings are missing, and the flower is deformed seriously in the last row of Fig. 8(b). Secondly, the optical flow framework was employed in SF so that the search range was limited, while PM and our algorithm search in the global range. As shown in Fig. 9, comparing with SF and PM, our algorithm obtains the smallest reconstruction errors. Moreover, the average reconstruction errors are also computed and listed in Table II. The errors are computed in the same way as those in Fig. 6 and Table I.

C. More Multimedia Applications

As illustrated in the previous sections, our approach can find dense correspondences between two images. Then many multimedia applications can be accomplished by those dense correspondences. In this subsection, we will introduce three multimedia applications which utilize the dense correspondences directly: color transfer between image pairs, exposure fusion from an input exposure sequence of multiple images, and semantic foreground segmentation by the dense correspondences. Color transfer between images is one of the most common tasks in image and video processing. The task of color transfer can be viewed as borrowing the color characteristics from another image [5]. We first match the two input images, and then fit three monotonic curves in every channel of the RGB color space. The curves are modeled by utilizing a piecewise cubic spline [30]. Fig. 10 shows the results of the color transfer produced using PM [20], SF [34] and our method.

Fig. 11. Comparison of exposure fusion using PM [20] (b), SF [34] (c) and our method (d). (a) is the input multiple exposure images. We list two images as examples because of the space limitation where the image sequences contain much more images. The red arrows indicate the ghost artifacts or under-exposure regions.

The input image pairs in Fig. 10(a) are firstly matched by different methods, and then the most reliable correspondences are selected to fit the curves. As mentioned before, Barnes *et al.* [20] computes the similarity only with the Euclidean distance, where the correspondences between the images that are matched by PM are only similar in the color space. Thus, the color transfer results by PM in Fig. 10(b) are not correct in many regions. As shown in Fig. 10(c), the results by SF are better than the results by PM since SF has utilized the similarity measurement with SIFT. However, the large displacements between the input images are not solved by SF, which makes the incorrect results of color transformation in some texture regions. In contrast, we find the reliable correspondences and successfully transfer the color characteristics between the two input images. Thus, our approach obtains much more pleasing color transformation results [see Fig. 10(d)].

The task of exposure fusion is to obtain the full dynamic range of a scene by fusing multiple exposure images into a single high-quality image while preserving the detail and color appearances. The most challenging problem is how to deal with the situations that the scene has the moving objects. In common, the multiple exposure images with moving objects are first aligned and then fused for these registered images. The difficulty and key step for exposure fusion is how to find the accurate dense correspondences with moving objects among the input images taken under different exposure settings. In this application, the dense correspondences between the images are first searched and then aligned as [36]. These registered images are finally fused using generalized random walks [52], [49]. The exposure fusion procedures can be summarized as the following four steps. The first step is to select the normal-exposure image from

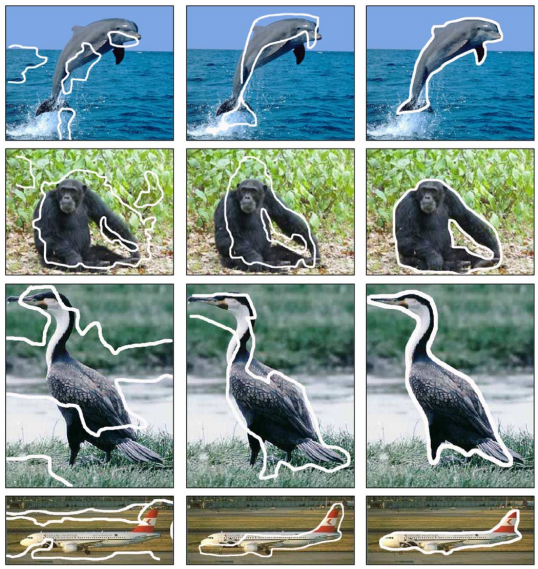

Fig. 12. Examples of the figure-ground segmentation produced using PM [20] (b), SF [34] (c), and our method (d). We randomly select four images from the Coltech-101 dataset [7] as examples because of the space limitation.

the input sequence as the reference image since the normal-exposure image contains more well-exposed and useful regions. Then, the dense correspondences between the other exposure images and the reference image are founded by using the proposed method. The third step is to reconstruct the other exposure images using this reference image. Finally, both the images that contains the reconstructed exposure images and the reference image are fused using the generalized random walks fusion method. Fig. 11 gives the comparison results produced by PM [20], SF [34] and our method, where the input image sequences are taken from [23], [38], [24], [49]. As mentioned before, PM matches the correspondence only in the color space so that the changing regions of the color can not be correctly detected, then the fusion results [Fig. 11(b)]lost many detailed color regions and textures by just fusing some parts of the input exposure images. Since the large displacement limitation of the SF method, the fused result by SF [Fig. 11(c)]contain many ghost artifacts that have been indicated by the red arrows. In contrast, our approach achieves more pleasing fusion results with vivid colors as shown in Fig. 11(d).

The figure-ground segmentation is a fundamental problem in computer vision, and its goal is to produce a binary segmentation mask of the image by separating foreground objects from their background [9]. By following the procedures in [28], [45], we first match a test image to multiple exemplar images from the same class on Caltech-101 dataset. Then we perform figure-ground segmentation with an MRF framework [45]. Fig. 12 shows the examples of the figure-ground segmentation produced using PM [20], SF [34] and our method. We match the test image to other images from the same class, which is the most important procedure during the figure-ground segmentation. Barnes *et al.* [20] computes the similarity between two images only with the Euclidean distance, while the images in the dataset Coltech-101 contain different colors and shapes. The correspondence results [Fig. 12(a) and (b)] by both SF and PM methods are not accurate, especially when the large displacement motions happened. In contrast, our method accurately find the dense correspondences between the images containing different transformations, and our approach successfully segments the foreground objects from the complicated background in Fig. 12(c).

IV. Conclusion

This paper presents a novel dense matching algorithm using structured-patch optimization by integrating the dense information with the sparse features of patches. Our approach contains all the transformations, such as translation, rotation, scale and deformation, during the optimization. In terms of the property of our structured-patches, a new energy optimization method is designed to find the correspondences. The scene information is also extracted to guide the matching process. Moreover, in order to preserve the smoothness of the edges, we propose a local refinement method to optimize the local correspondence. Comparing to the current state-of-the-art matching algorithms, our approach achieves better visual quality of dense matching results. The proposed algorithm can be extended for the other multimedia applications such as video alignment, seam carving [44], etc., in the future work. Moreover, the implementation of our algorithm can be adapted to accomplish a more efficient video processing system which can be utilized for video matching, alignment and editing.

References

[1] B. Horn and B. Schunck, "Determining optical flow," *Artificial Intell.*, vol. 16, pp. 185–203, 1981.
[2] A. Fusiello, V. Roberto, and E. Trucco, "Efficient stereo with multiple windowing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 1997, pp. 858–863.
[3] P. Indyk and R. Motwani, "Approximate nearest neighbors: Towards removing the curse of dimensionality," in *Proc. Symp. Theory Comput.*, 1998, pp. 604–613.
[4] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, 2001.
[5] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Comput. Graphics Appl.*, vol. 21, no. 5, pp. 34–41, Sep.-Oct. 2001.
[6] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. Comput. Vis.*, vol. 47, no. 1, pp. 7–42, 2002.
[7] F. Li, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2004, p. 178.
[8] D. G. Lowe, "Distinctive image feature from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[9] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," *ACM Trans. Graphics*, vol. 3, no. 3, pp. 309–314, 2004.
[10] A. Bruhn, J. Weickert, and C. Schnorr, "Lucas/Kanade meets Horn/Schunk: Combining local and global optical flow methods," *Int. J. Comput. Vis.*, vol. 61, no. 3, pp. 211–231, 2005.
[11] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.

[12] S. Avidan, "Ensemble tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 2, pp. 261–271, Feb. 2007.
[13] O. Boiman and M. Irani, "Detecting irregularities in images and in video," *Int. J. Comput. Vis.*, vol. 74, pp. 17–31, 2007.
[14] E. Shechtman and M. Irani, "Matching local self-similarities across images and videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2007, pp. 1–8.
[15] I. Simon and S. Seitz, "A probabilistic model for object recognition, segmentation, and non-rigid correspondence," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2007, pp. 1–7.
[16] S. Bagon, O. Boiman, and M. Irani, "What is a good image segment? a unified approach to segment extraction," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 30–44.
[17] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani, "Summarizing visual data using bidirectional similarity," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2008, pp. 1–8.
[18] F. Tombari, S. Mattoccia, L. D. Stefano, and E. Addimanda, "Classification and evaluation of cost aggregation methods for stereo correspondence," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2008, pp. 1–8.
[19] T. Caetano, J. McAuley, C. Li, Q. V. L. , and A. Smola, "Learning graph matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 6, pp. 1048–1058, Jun. 2009.
[20] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein, "The generalized PatchMatch correspondence algorithm," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2010, pp. 29–43.
[21] S. Battiato, A. Bruna, and G. Puglisi, "A robust block-based image/video registration approach for mobile imaging devices," *IEEE Trans. Multimedia*, vol. 12, no. 7, pp. 622–635, Nov. 2010.
[22] J. Kim and K. Grauman, "Asymmetric region-to-image matching for comparing images with generic object categories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2010, pp. 2344–2351.
[23] F. Pece and J. Kautz, "Bitmap movement detection: HDR for dynamic scenes," in *Proc. Eur. Conf. Vis. Media Prod.*, 2010, pp. 1–8.
[24] W. Zhang and W. K. Cham, "Gradient-directed composition of multi-exposure images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2010, pp. 530–536.
[25] J. Shen, X. Yang, Y. Jia, and X. Li, "Intrinsic images using optimization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2011, pp. 3481–3487.
[26] M. Bleyer, C. Rhemann, and C. Rother, "Patchmatch stereo - stereo matching with slanted support windows," in *Proc. Brit. Mach. Vis. Conf.*, 2011, pp. 1–11.
[27] T. Brox and J. Malik, "Large displacement optical flow: Descriptor matching in variational motion estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 500–513, Mar. 2011.
[28] C. Liu, J. Yuen, and A. Torralba, "Nonparametric scene parsing via label transfer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2368–2382, Dec. 2011.
[29] O. Duchenne, A. Joulin, and J. Ponce, "A graph-matching kernel for object categorization," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1792–1799.
[30] Y. HaCohen, E. Shechtman, D. B. Goldman, and D. Lischinski, "Non-rigid dense correspondence with applications for image enhancement," *ACM Trans. Graphics*, vol. 30, no. 4, pp. 70:1–70:9, 2011.
[31] Y. S. Heo, K. M. Lee, and S. U. Lee, "Robust stereo matching using adaptive normalized cross-correlation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 4, pp. 807–822, Apr. 2011.
[32] S. Korman and S. Avidan, "Coherency sensitive hashing," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1607–1614.
[33] J. Lee, M. Cho, and K. M. Lee, "Hyper-graph matching via reweighted random walks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2011, pp. 1633–1640.
[34] C. Liu, J. Yuen, and A. Torralba, "Sift flow: Dense correspondence across scenes and its applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 978–994, May 2011.
[35] F. Besse, C. Rother, A. Fitzgibbon, and J. Kautz, "PMBP: Patchmatch belief propagation for correspondence field estimation," in *Proc. Brit. Mach. Vis. Conf.*, 2012, pp. 132.1–132.11.
[36] J. Hu, O. Gallo, and K. Pulli, "Exposure stacks of live scenes with hand-held cameras," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 499–512.
[37] I. Olonetsky and S. Avidan, "TreeCANN - k-d tree coherence approximate nearest neighbor algorithm," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 602–615.
[38] J. Shen, Y. Zhao, and Y. He, "Detail-preserving exposure fusion using subband architecture," *Vis. Comput.*, vol. 28, no. 5, pp. 463–473, 2012.
[39] L. Xu, J. Jia, and Y. Matsushita, "Motion detail preserving optical flow estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 9, pp. 1744–1757, Sep. 2012.
[40] F. Zhou and F. D. la Torre, "Factorized graph matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2012, pp. 127–134.
[41] Z. Chen, H. Jin, Z. Lin, S. Cohen, and Y. Wu, "Large displacement optical flow from nearest neighbor fields," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2013, pp. 2443–2450.
[42] M. Cho, K. Alahari, and J. Ponce, "Learning graphs to match," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 25–32.
[43] P. Heise, S. Klose, B. Jensen, and A. Knoll, "PM-huber: PatchMatch with huber regularization for stereo matching," in *In Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2360–2367.
[44] J. Shen, D. Wang, and X. Li, "Depth-aware image seam carving," *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1453–1461, Oct. 2013.
[45] J. Kim, C. Liu, F. Sha, and K. Grauman, "Deformable spatial pyramid matching for fast dense correspondences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2013, pp. 2307–2314.
[46] S. Korman, D. Reichman, G. Tsur, and S. Avidan, "Fast-Match: Fast affine template matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2013, pp. 2331–2338.
[47] S. Y. Lee, J. Y. Sim, C. S. Kim, and S. U. Lee, "Correspondence matching of multi-view video sequences using mutual information based similarity measure," *IEEE Trans. Multimedia*, vol. 15, no. 8, pp. 1719–1731, Dec. 2013.
[48] M. Afonso, J. Nascimento, and J. Marques, "Automatic estimation of multiple motion fields from video sequences using a region matching based approach," *IEEE Trans. Multimedia*, vol. 16, no. 1, pp. 1–14, Jan. 2014.
[49] J. Shen, Y. Zhao, S. Yan, and X. Li, "Exposure fusion using boosting Laplacian pyramid," *IEEE Trans. Cybern.*, vol. 44, no. 9, pp. 1579–1590, Sep. 2014.
[50] L. Shao, X. Zhen, D. Tao, and X. Li., "Spatio-temporal Laplacian pyramid coding for action recognition," *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 817–827, Jun. 2014.
[51] J. Tang, L. Shao, and X. Zhen, "Robust point pattern matching based on spectral context," *Pattern Recog.*, vol. 47, no. 3, pp. 1469–1484, 2014.
[52] J. Shen, Y. Du, W. Wang, and X. Li, "Lazy random walks for super-pixel segmentation," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1451–1462, Apr. 2014.
[53] L. Shao, L. Liu, and X. Li, "Feature learning for image classification via multiobjective genetic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 7, pp. 1359–1371, Jul. 2014.

Xiameng Qin is currently pursuing the Ph.D. degree at the School of Computer Science, Beijing Institute of Technology, Beijing, China.

His current research interests include exposure fusion and image matching techniques.

Jianbing Shen (M'11–SM'12) is a Full Professor with the School of Computer Science, Beijing Institute of Technology, Beijing, China. He has authored or coauthored more than 50 refereed papers in journals and conference proceedings. His research interests include computer vision and computer graphics.

Dr. Shen has received many flagship honors including the Fok Ying Tung Education Foundation from the Ministry of Education, the Program for Beijing Excellent Youth Talents from the Beijing Municipal Education Commission, and the Program for New Century Excellent Talents in University from the Ministry of Education.