This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2595321, IEEE Transactions on Circuits and Systems for Video Technology

HSP2P: Hierarchical Superpixel-to-Pixel Dense Image Matching

Xingping Dong, Jianbing Shen, *Senior Member, IEEE*, and Ling Shao, *Senior Member, IEEE*

***Abstract*—In this paper, we propose a novel matching method to establish dense correspondences automatically between two images in a hierarchical superpixel-to-pixel (HSP2P) manner. Our method first estimates dense superpixel pairings between the two images in the coarse-grained level to overcome large patch displacements and then utilize superpixel level pairings to drive the matchings in the pixel level to obtain fine texture details. In order to compensate for the influence of color and illumination variations, we apply a regularization technique to rectify images by a color transfer function. Experimental validation on benchmark datasets demonstrates that our approach achieves better visual quality outperforming state-of-the-art dense matching algorithms.**

***Index Terms*—Dense correspondence, hierarchical superpixel-to-pixel, image reconstruction, color transfer, exposure fusion.**

I. Introduction

Establishment of dense correspondences between multiple images is one of the fundamental tasks in computer vision. It is used in many applications including 3D reconstruction [1], visual tracking [2], image segmentation [6], [5], multi-exposure fusion [7], video stylization [3], free-viewpoint video [4] and others [8], [9], [10], [11], [12], [13]. Unlike feature points based image matching approaches, every pixel in one image is assigned to the corresponding pixels in other images [14].

Depending on the size of the search window, i.e. whether a small search window [16] or the whole image [17], conventional dense matching methods could be classified as local or global [18] techniques. Most traditional dense matching algorithms, *e.g.* point-wise stereo matching and optical flow, estimate correspondences in local windows for small displacements. Such a limited search operation leads to faster processing but lower accuracy. In order to derive more reliable correspondences, Liu *et al.* [19] extracted SIFT descriptors of every pixel in the given images and applied an optical flow based dense matching procedure called SIFT Flow. Then, they imposed a pixel-level Markov Random Field (MRF) model to solve the SIFT Flow. Heo *et al.* [20] normalized the image color and formulated matching as an energy minimization problem using adaptive normalized cross-correlations, which was then optimized by graph-cuts [21]. Nevertheless, mismatches caused by local search techniques increase rapidly in case the objects undergo large displacements.

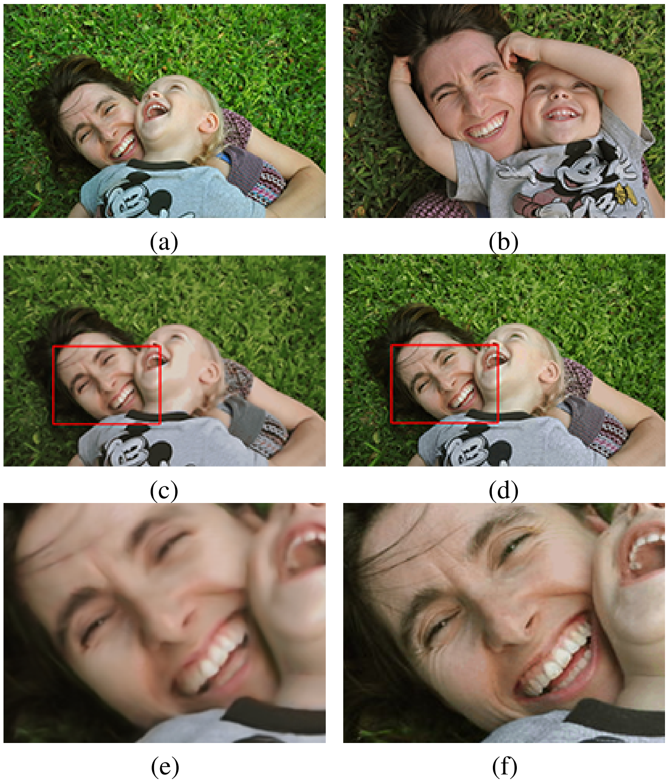


Fig. 1. An example of image reconstruction in fine detailed texture regions. (a) Target image. (b) Reference image. (c) Reconstructed target image by PatchMatch (PM) [15]. (d) Reconstructed target image by HSP2P. (e-f) Zoomed in regions. It is apparent that HSP2P captures more texture details such as wrinkle and hair.

It is feasible to employ feature matching to guide the dense matching procedure. Brox and Malik [22] presented Large Displacement Optical Flow (LDOF) that added a sparse feature matching scheme into the classical optical flow framework. Xu *et al.* [23] first matched a set of sparse features and then expanded them onto candidate motion fields. Optical flow is utilized to combine the motion fields as a last step. These feature matching guided methods improved the accuracy of dense correspondences, however, the matching on the edges of large displacement regions was not reliable as a result of the regularization effect of the incorporated motion flow stages.

In order to overcome aforementioned shortcomings, global search based methods were developed to match images depicting complex scenes. After the Generalized PatchMatch

X. Dong and J. Shen are with Beijing Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, Beijing 100081, P. R. China. (Email: {dongxingping, shenjianbing}@bit.edu.cn)

L. Shao is with the Department of Computer and Information Sciences, Northumbria University, Newcastle upon Tyne NE1 8ST, U. K. (Email: ling.shao@northumbria.ac.uk)

1051-8215 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2595321, IEEE Transactions on Circuits and Systems for Video Technology

(GPM) approach [18] in which the approximate nearest neighbors of every pixel in one image are found in the second image, Korman and Avidan [17] combined locality sensitivity hashing [24] and GPM to determine corresponding patches between two images. HaCohen *et al.* [25] presented a nonrigid dense correspondence (NRDC) algorithm that improves GPM by adding a color bias and a gain factor per channel into the transformation. Besse *et al.* [26] integrated a propagation scheme with GPM to estimate the corresponding fields of two images. Still, it is a challenge for patch based image matching algorithms to maintain the continuity of edges. Besides, the details of images in such methods might not be preserved accurately.

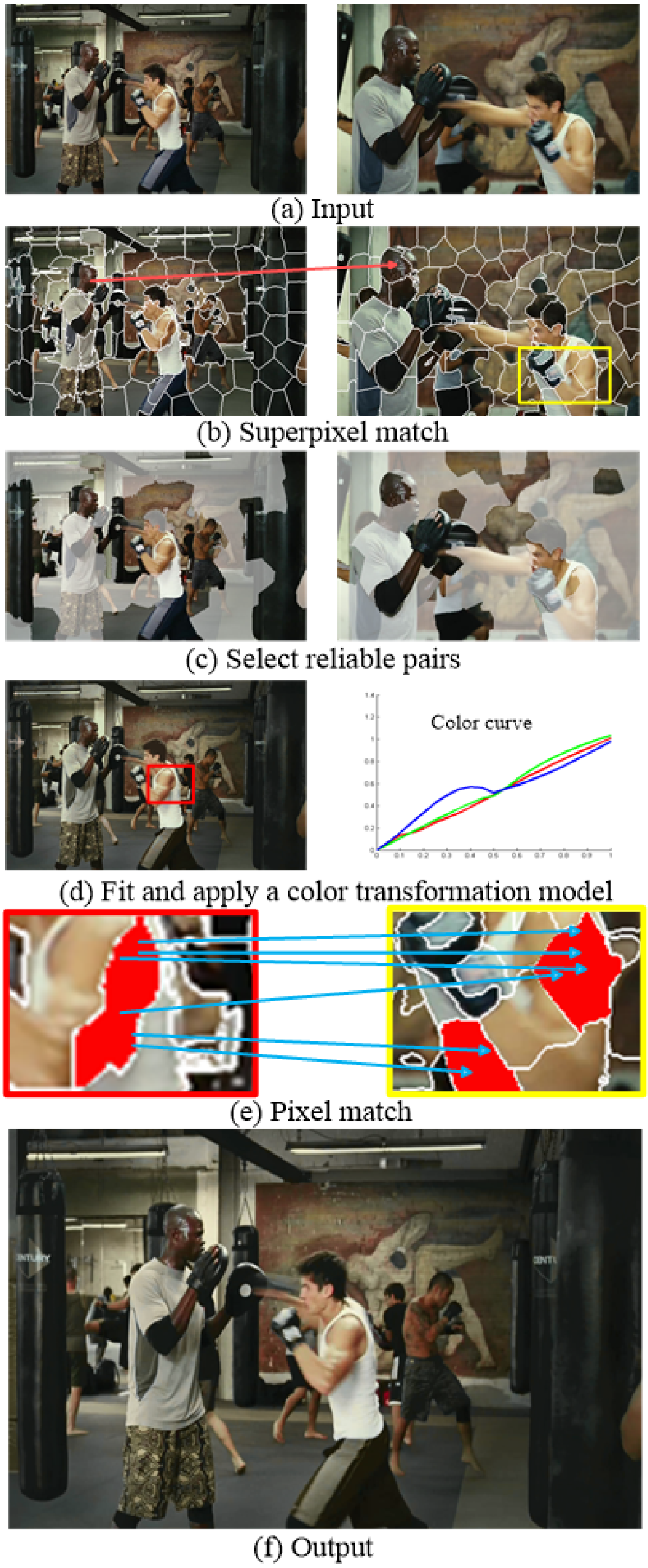
Here, we propose a dense correspondence algorithm from superpixel to pixel levels by taking advantages of global search in the superpixel level to address the large displacement problem and local search in the pixel level to reconstruct fine details. In global search, we search for the superpixel pairs between the input target image and the reference image. Compared with the grid patches, superpixels offer more reliable and consistent information of support regions. Most superpixel methods partition regions according to the similarity between adjacent pixels, and most adjacent similar pixels can be aggregated together into consistent regions. This allows similar objects to retain appearance-wise similar superpixels, which makes superpixels as good priors to establish adequate region matches. Mismatches would still exist after the initial superpixel matching process. Thus, low-level features extracted from the corresponding superpixels may be quite different due to significant geometric deformations. To solve this problem, we employ a consistency criterion to calculate the reliability of the matched superpixels. Then for reliable matches, a color transformation model is fit. Using this model, the target image is transferred to an intermediate image, which is similar to the reference image in color except for some small regions. One reason is that the color transformation model cannot completely align the real mapping between the target image and the reference image. The pixel match is used to correct this error. To this end, we first find the superpixel correspondences between the intermediate image and the reference image. For each pixel in the intermediate image, we search its nearest neighbor in the corresponding updated superpixel matches. As shown in Fig. 1, the final adjustment in the pixel level enables our HSP2P algorithm to keep better details in most images.

In summary, our approach makes two main contributions to dense correspondence and matching.

1) We present a global search scheme in the superpixel level, which overcomes the problem of large displacement between image pairs.
2) The proposed method enjoys a novel pixel level refinement, i.e. a local search strategy, to keep details for better visual performance.

II. Dense Correspondences

Given a target image A and a reference image B, for each pixel $u \in P^A$, where P^A is the pixel set of image A, our goal is to seek a transformation function $T(u) := u \rightarrow v, u \in$


(a) Input

(b) Superpixel match

(c) Select reliable pairs

(d) Fit and apply a color transformation model

(e) Pixel match

(f) Output

Fig. 2. Workflow of HSP2P. (a) is an input image pair. In (b), a superpixel directed by the red arrow is the nearest neighbor of the corresponding superpixel in the left image. (c) shows the reliable pairs and the masked regions indicate the unreliable superpixels. In (d), the left image is the result mapped by a color transformation model and the right sub-figure shows the corresponding color curve. (e) is an illustration of pixel match. The red and yellow rectangles correspond to the rectangles in (d) and (b), respectively. (f) is the reconstructed result.

1051-8215 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2595321, IEEE Transactions on Circuits and Systems for Video Technology

$P^A, v \in P^B$. Then we obtain the following transformation function:

$$T^u = \arg\min_{T} d(f_u^A, f_{T(u)}^B) \tag{1}$$

where f_u^A is the feature vector of a pixel $u \in P^A$, and $d(a,b)$ is a distance measurement between the feature vectors a and b.

One solution of this problem is to compute the nearest neighbor of A from B. It is impractical to exactly search the nearest neighbor, since the number of pixels in an image usually is huge. We propose an approximative nearest neighbor searching algorithm in images, which include four main steps:

1) superpixel match;
2) selecting reliable matched pairs;
3) color transformation fitting;
4) pixel match.

We precompute superpixels of images by an over-segmentation algorithm, then begin our main steps. First, for each superpixel in the target image A, we find its nearest neighbor in the reference image B, by minimizing the feature distance between superpixels. Second, our method HSP2P selects reliable matched superpixels according to the consistence of translations in a neighborhood. Third, we use these reliable matches to robustly fit a color transformation model. To improve the matching performance, we replace the target pixels of that predicted by the proposed color transformation model. Then the first step is repeated to update the matched pairs of superpixels. Finally, for each updated pair of matched superpixels, the nearest neighbor of each pixel in the target superpixel u is searched from the reference superpixel v. The workflow is shown in Fig. 2.

A. Superpixel match

As a preprocessing stage of the proposed algorithm, we segment these two input images into superpixels using the simple linear iterative clustering (SLIC) method [27]. The generated superpixels are relatively compact, regular and have similar sizes. However, SLIC may fail to adhere to the boundaries for some complex images, which is a common problem for most superpixels methods. We will solve this problem by pixel matching (step 4) perfectly.

After generating the superpixels, a feature vector is computed for each of them. In this paper, we use the three channels of Lab color space as the color features, and use SIFT descriptors [28] on a dense regular grid as the region features. Inside a superpixel, most pixels are similar. So we can simply use the average Lab color vector and the average SIFT vector as the features of each superpixel. The corresponding features for superpixel i are denoted as f_{lab}^i and f_{sift}^i, respectively.

To combine these two kinds of features, we define a distance function for superpixel pair (i,j) as follows:

$$D(i,j) = \alpha_1 \|f_{lab}^i - f_{lab}^j\|_2 + \alpha_2 \|f_{sift}^i - f_{sift}^j\|_2, \tag{2}$$

where α_1 and α_2 are constants, $\|\cdot\|_2$ is a L2-norm of a vector.

According to this distance function, for each superpixel $i \in S^A$ in A, where S^A is the superpixel set of image A, we compute the distances between superpixel i and all superpixels in B. And then the superpixel $j \in S^B$ with the minimum distance is chosen as the matching result of $i \in S^A$. The matching function is defined as:

$$M(i) = \arg\min_{j \in S^B} D(i,j), i \in S^A. \tag{3}$$

B. Selection of reliable pairs

After the superpixel matching stage, we obtain the initial matches, and some of them may be unreliable. Therefore, we need to select more reliable matches for the next color transformation fitting. Although we cannot individually determine which matches are unreliable, we can use groups of matches to improve robustness. Several matching works [28], [29], [25] agree that the matching error, which is produced by a coherent block region, is much lower than that of any individual pixel. Therefore, we define a consistency function to calculate a coherence error for a group of matches. The domain of function is transformed into vectors TS of a superpixel $i \in S^A$ and its neighbors set $N(i)$, which is the set of superpixels connected to superpixel i in boundaries. Given a superpixel i and its corresponding matched superpixels $M(i)$, $TS(i)$ equals $\mathbf{c}(M(i)) - \mathbf{c}(i)$, where $\mathbf{c}$ is the geometric center of a superpixel (i.e. the average coordinate of pixels in this superpixel). Then this consistency function can be formulated as follows:

$$C(i) = \frac{1}{\sum_{j \in N(i)} w_{ij}} \sum_{j \in N(i)} w_{ij} \|TS(i) - TS(j)\|_2, \tag{4}$$

where $w_{ij} = exp(-\beta \|f_{lab}^i - f_{lab}^j\|_2)$ is a weighting function, which measures the similarity between two feature vectors, and β is a constant fixed as 0.02 in our experiments. This weighting function indicates that similar adjacent superpixels should keep consistency translation. Once we get the consistency errors, we can sort them as an ascending sequence. Then the matches corresponding to the first $\tau\%$ number of superpixels are selected as the reliable matches. In our work, τ is set as 50.

C. Color fitting

The goals of our global color transformation model include two aspects. First, we want to correct some errors produced by unreliable matches. We can use this model to map the color features of all superpixels in the target image. Then these new color features are used for the superpixel match **(step 1)** to get the updated superpixel matches, which will be applied to the final step. Second, in order to produce an initial transformed image by mapping all of the colors in the target image, this color transformation model should be global and it should also recover various color differences.

In previous works, some methods have been used for the color transformation model, such as simple adjustment of mean and variance [30], histogram matching and more

1051-8215 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2595321, IEEE Transactions on Circuits and Systems for Video Technology

sophisticated statistics-based methods [31]. But these methods have some limitations, e.g. [30] cannot reproduce complex variations, [31] may fail to produce a reasonable mapping for colors that do not appear in the reliable matched superpixels in the target image.

Therefore, a parametric model is chosen to predict a meaningful mapping for those that do not appear in the input superpixels. Our algorithm fits three smooth curves, one per-channel of the Lab color space. For uniformity, we normalize the range of each channel to $[0, 1]$, respectively. Then we apply a piecewise cubic spline with 9 breaks to model each of the curves. Two break points are at the ends of the gamut range (i.e. zero and one), and 7 points are uniformly distributed along the subrange produced by reliable correspondences. For robustness to outliers, we add soft constraints to the Lab curves outside the color range. Each of the Lab curves is constrained to pass through the points $y(-0.1) = -0.1$ and $y(1.1) = y(1.1)$.

Algorithm 1 HSP2P: Dense Correspondence

```
Input: Target image $A$ and Reference image $B$;
Output: Matching result: $T(u) \in B, \forall u \in A$;
Obtain the superpixels of two images: $\{S_i^A\}$, $\{S_i^B\}$;
Extract the color feature $f_{lab}^i$ and SIFT descriptor $f_{sift}^i$;
Step 1: Superpixel match;
Find a transformation $M(i), \forall i \in A$ by using eq. 2 and eq. 3;
Step 2: Select reliable matched pairs;
Sort the matched pairs by eq. 4 and choose the first $\tau\%$ matches as reliable pairs;
Step 3: Color transformation fitting
Fit a global color transformation model;
Map color feature of each target superpixel $S_i^A$: $f_{lab}^i \rightarrow \bar{f}_{lab}^i$;
Repeat step 1 by replacing $f_{lab}^i$ of $\bar{f}_{lab}^i$ to get new matched superpixels: $M(i) \rightarrow \bar{M}(i)$;
Update target image: $A \rightarrow \bar{A}$
Step 4: Pixels match
for  each target superpixel $S_i^A$ do
    Obtain its candidate set $cand_i$ by adding its match $S_{\bar{M}(i)}^B$ and its neighbors' matches $S_{\bar{M}(j)}^B$;
    for each pixel $u \in S_i^A$ do
        Find the transformation $T(u)$ using eq. 5;
    end for
end for
```

D. Pixel level matching

After color transformation, although most of colors in the updated image are consistent with that in the reference image, some corresponding colors may have large deviation (such as the regions pointed by the red arrow in Fig. 3 (c)). This may be due to that the color transformation model cannot completely fit the real mapping function. So we need to adjust the matching result in the pixel-level. A simple strategy is to compare the distances between each pixel in the target superpixel and all pixels in the corresponding reference superpixel, and select the one with the minimum distance as the final matching result. This is practical, since the number of pixels in a superpixel is only a few hundred in our experiments. However, as mentioned before, some superpixels may fail to adhere to boundaries, which may lead to matching errors along the boundaries. Simultaneously, the unreliably matched superpixels may still exist. This may cause failed matches of all pixels in these superpixels (shown in Fig. 3).

To overcome the above problems, we expand the searching space by adding the corresponding matches of neighbors. Given a superpixel $i \in S^A$, we not only consider the pixels in the corresponding matched superpixel $M(i) \in S^B$ as the candidate searched set, but also add the pixels in the matches of its neighbors $M(j), j \in N(i)$ into the candidate set. Then, for each pixel in superpixel i, we find the pixel with the minimum distance from the candidate set as the final matching result. Here, we only use the color feature to compute the distance to reduce computation time. The formulation is defined as follow:

$$T(u) = \arg\min_{v \in cand_i} \|\bar{f}_{lab}^u - f_{lab}^v\|_2, \forall u \in S_i^A, \forall i \in S^A, \quad (5)$$

where $\bar{f}_{lab}^u$ is the Lab feature in the updated target image, and $cand_i$ represents the candidate set of superpixel i. We summarize our algorithm in **Algorithm 1**.

III. Experimental Results

In this section, we evaluate the proposed algorithm on a variety of image pairs, by comparing with several state-of-the-art techniques of dense matching, including PatchMatch (PM) [15], SIFT Flow (SF) [19], and deformable spatial pyramid method (DSP) [32]. We directly run the source codes from the authors' homepages for comparison. In order to achieve a fair comparison of correspondence quality with these methods, the advised parameter settings by the original authors are used for the above algorithms in our comparison experiments. The main parameters of our algorithm include α_1 and α_2 in equation (2). These two parameters measure the weights of different features. If most colors of an input image pair are similar, such as the images in the Video Pair dataset, we can reduce the SIFT weight α_2. For different applications, these two parameters are manually adjusted, which will be described next in detail. The other parameters are fixed as mentioned before.

In our experiments, we apply the proposed algorithm to three applications: image reconstruction, color transfer between image pairs, and exposure fusion from an input exposure sequence of multiple images.

A. Image reconstruction

Given the dense corresponding map of two input images, the target image can be reconstructed by replacing each target pixel of a corresponding pixel from the reference image. Here, we test the reconstruction performance on a challenging dataset: the Video Pair dataset [17], by comparing our algorithm with PM [15], SIFT Flow (SF) [19], and deformable spatial pyramid method (DSP) [32]. In this dataset, most of the image pairs contain high variability in geometry structure, such

1051-8215 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2595321, IEEE Transactions on Circuits and Systems for Video Technology

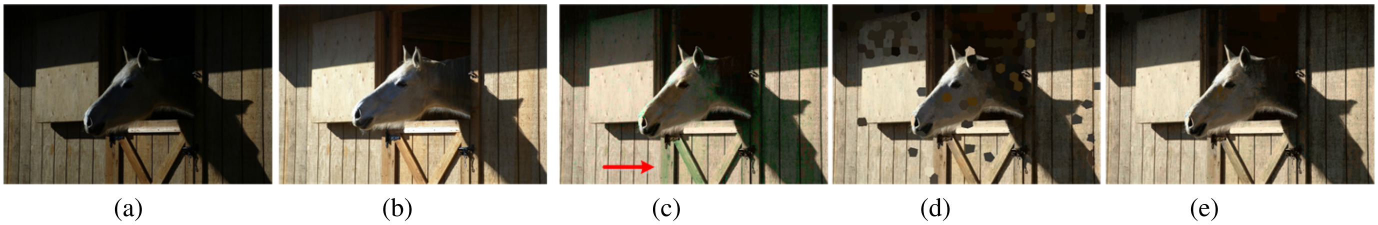


Fig. 3. An example of pixel matching. (a) is the input target image. (b) is the input reference image. (c) is the reconstructed image by the color transformation model. (d) is the result reconstructed by pixel matching without neighbor superpixels. (e) is the result of pixel matching with neighbor superpixels. Note that the color transformation model fails to map the correct color to the regions pointed by the red arrow in (c). These errors are corrected by pixel matching, as shown in (e).

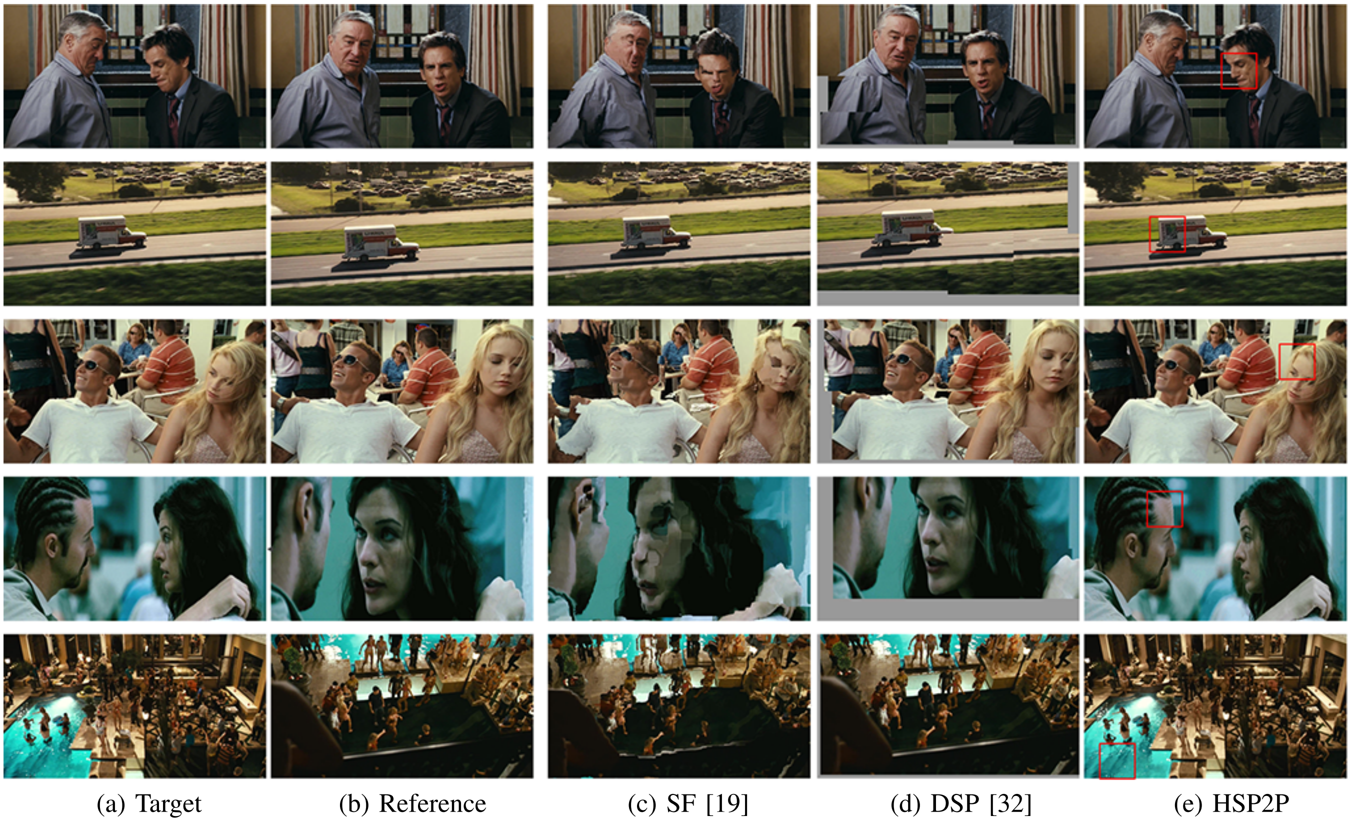


Fig. 4. Results of image reconstruction. (a) shows the input target images. (b) depicts the input reference images. (c-e) are the reconstructed images of SIFT Flow (SF) [19], deformable spatial pyramid method (DSP) [32], and HSP2P, respectively. The regions in red boxes of (e) will be enlarged in Fig. 5.

as huge deformation and large displacement. This may cause some region features (like SIFT descriptors) to be invalid. Fortunately, most of them are similar in color. Hence, we only consider the color feature by setting $\alpha_1 = 1, \alpha_2 = 0$ to reduce the matching error.

As shown in Fig. 4, five pairs of images are chosen to demonstrate the effectiveness of our algorithm. It appears that our reconstructed images have more accurate correspondences than SF [19] and DSP [32]. The reconstruction results of SF and DSP have many incorrect corresponding patches in large displacement regions. The reason may be that the methods based on spatial pyramid cannot well handle large displacement in the images. The performance between our algorithm and PM [15] is similar, but our results give more details. To compare the details, we zoom in the regions in red boxes in Fig. 4 (e), which is shown in Fig. 5. It indicates that PM fails to match correct pixels in some regions with details, such as the places pointed by the red arrows in Fig. 5. And why HSP2P performs better in these regions may be due to the exact searching strategy in a small region (the corresponding superpixels).

In order to quantitatively evaluate the performance of our algorithm, we compute the reconstruction error of the aforementioned three methods. This error is defined as the Root-Mean-Square Error (RMSE) between the target image and the reconstructed image. Denote I_A and I_R as the target RGB image and the reconstructed one, respectively. Then the reconstruction error ϵ is formulated as follows:

$$\epsilon = \frac{1}{N_p}\left(\sum_{u}^{N_p} \|I_A(u) - I_R(u)\|_2^2\right)^{0.5}, \tag{6}$$

1051-8215 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2595321, IEEE Transactions on Circuits and Systems for Video Technology

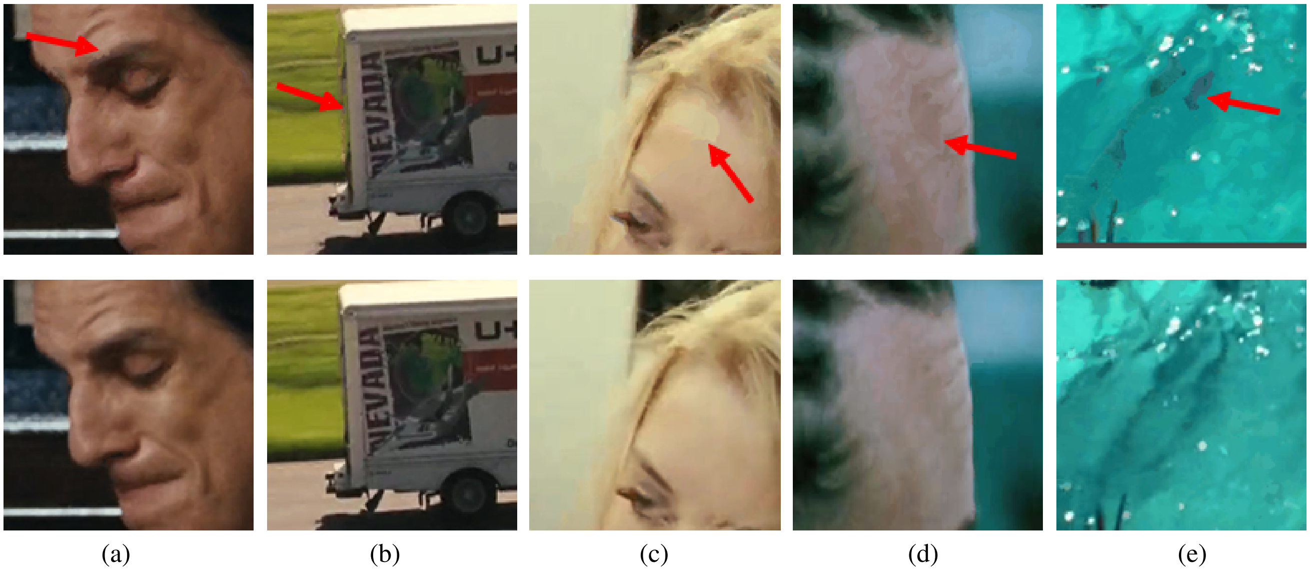


Fig. 5. Comparison with PM [15]. The first row is the zoomed regions from Fig. 4 (e). The second row is the corresponding reconstructed regions using HSP2P (best viewed in color).

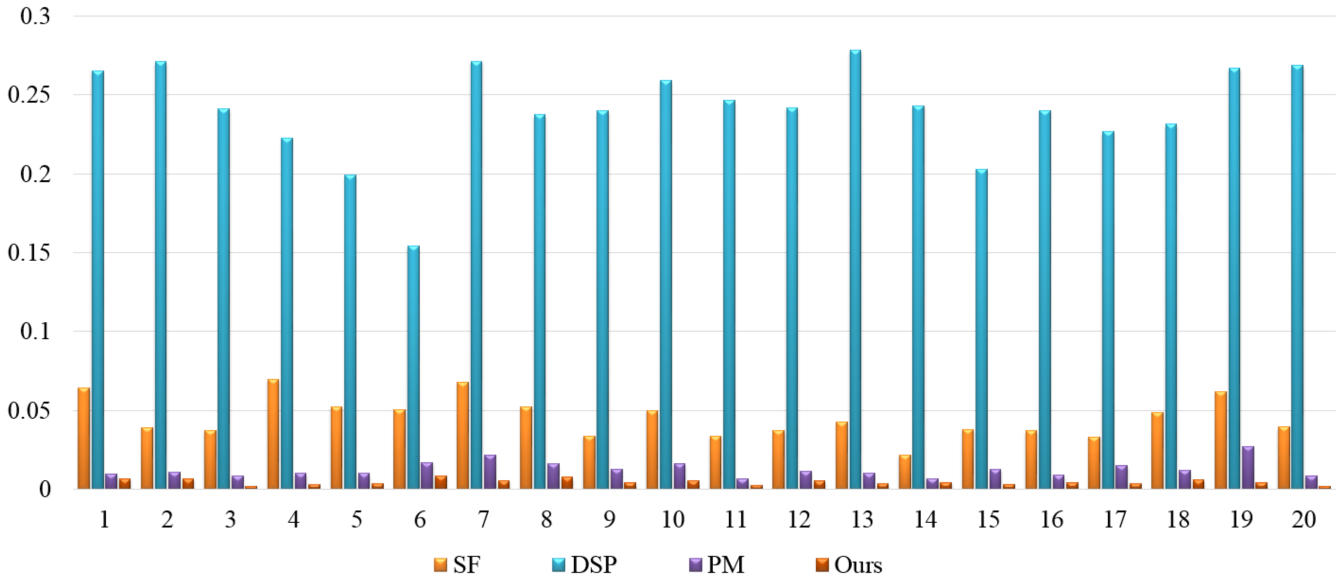


Fig. 6. Comparison of the reconstruction error by SF [19], DSP [32], PM [15] and our HSP2P on the benchmark VideoPairs dataset [17].

where N_p is the number of pixels in the target image.

The histograms of reconstruction error are shown in Fig. 6. Twenty pairs of images are randomly selected to calculate the reconstruction error of SF [19], DSP [32], PM [15] and ours. It is clear that our algorithm achieves the most accurate correspondence results, since the reconstruction error is lower than others. The reconstruction error of DSP and SF is much higher than others. One reason may be the limitation of the pyramid framework. DSP is based on the grid patch pyramid including different sizes of patches at several layers. The matching method may stop at a high layer for some images with large deformation. For example, in the third row of Fig. 4, the patches at a high pyramid layer (i.e. the big patch) are similar, such as the patch including the body of the boy, but the smaller patches at a lower layer are not similar, like the patch containing the girl's face. Then this method may only use the matches at a high layer as the final result. However, the image reconstruction requires high matching accuracy at the pixel-level i.e. the lowest level of pyramid. Thus, the reconstruction error of DSP is higher. Similarly, SF may focus on a high layer because of the multi-resolution pyramid structure.

In contrast, HSP2P and PM pay more attention to the pixel-level. So the reconstruction errors are lower. Compared with PM, HSP2P has better matching correspondences in detail as shown in Fig. 5, which leads to the lower reconstruction error.

1051-8215 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2595321, IEEE Transactions on Circuits and Systems for Video Technology

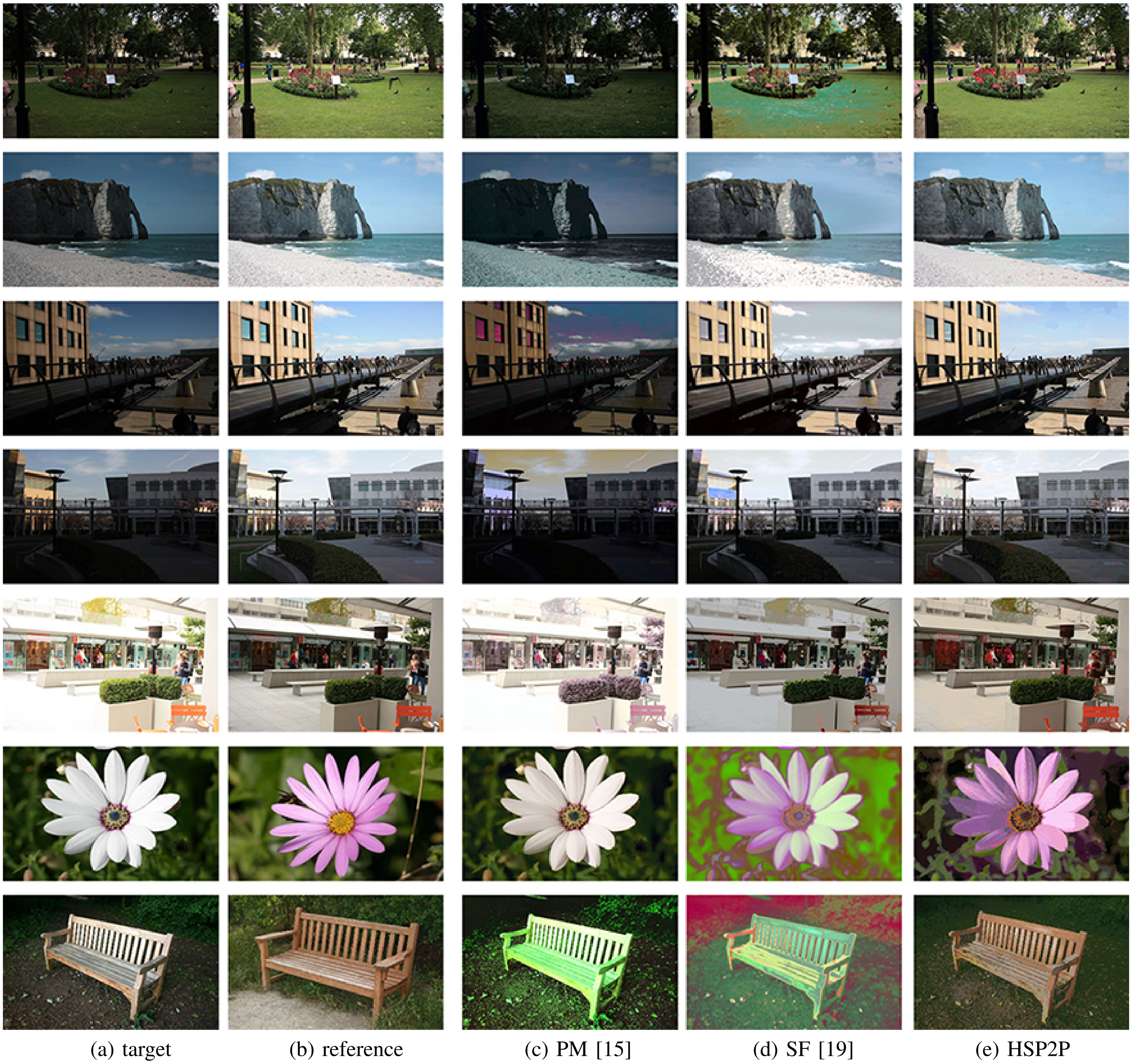


Fig. 7. Comparative results of the color transfer model. (a) Input target images. (b) Input reference images. (c-e) Transferred images of PatchMatch (PM) [15], SIFT Flow (SF) [19], and HSP2P $(\alpha_1 = 1, \alpha_2 = 5)$, respectively.

B. Color transfer

As mentioned before, our algorithm can find dense correspondences between two images. This can be used in many image applications, such as color transfer between image pairs, exposure fusion from an input exposure sequence of multiple images, and semantic foreground segmentation by the dense correspondences. Here, we use it for color transfer. This application is one of the most common tasks in image and video processing. The goal of color transfer can be viewed as borrowing the color feature from another image [30]. In this application, we increase the weight of SIFT feature by setting $\alpha_1 = 1$ and $\alpha_2 = 5$, since the color feature may not be very accurate for initial matching.

In this subsection, we compare PM [15], SF [19] and HSP2P. The results of color transfer are shown in Fig. 7. Firstly, we match the input image pairs by the above methods. Then all of these correspondences are used to fit the color transformation model proposed in Section 2.3, except for HSP2P, since this process has been contained in our method. The color transferred images of PM and SF are the mapping images using the color transformation model. As mentioned before, Barnes *et al.* [18] only compute the Euclidean distance of the color feature, where the dense correspondences between the input images may be effective in the color space. Thus, the color transfer results by PM in Fig. 7 (c) are not correct

1051-8215 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2595321, IEEE Transactions on Circuits and Systems for Video Technology

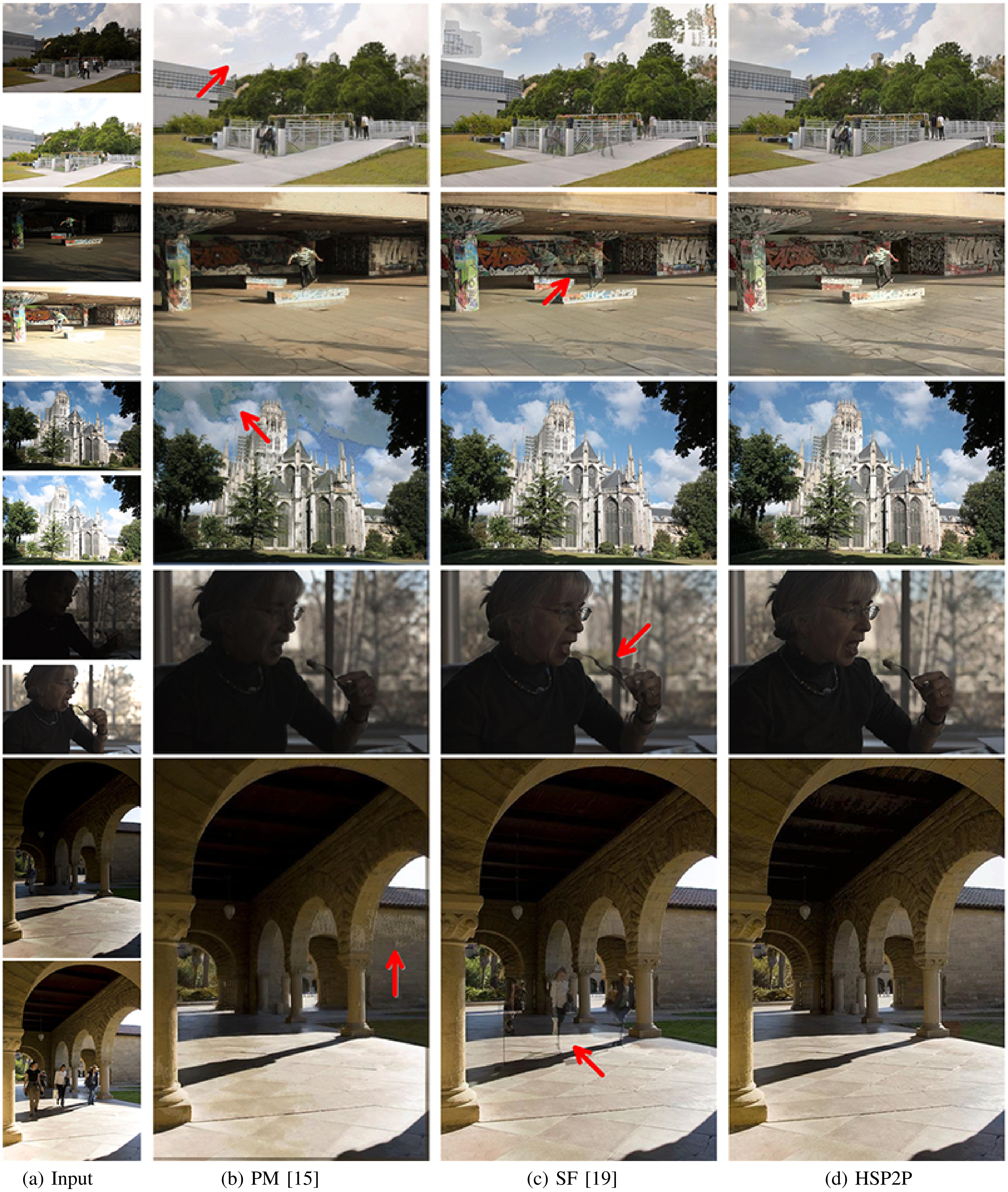


Fig. 8. Comparative results of exposure fusion. (a) Input multiple exposure images. Only Two images are listed as examples since the space is limited. In our experiments, we use three images as input. (b-d) Fusion images of PatchMatch (PM) [15], SIFT Flow (SF) [19], and HSP2P ($\alpha_1 = 1$, $\alpha_2 = 5$), respectively.

1051-8215 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2595321, IEEE Transactions on Circuits and Systems for Video Technology

in many regions. As shown in Fig. 7 (d), the results by SF are better than that by PM, since SF has utilized the similarity measurement with SIFT. However, the large displacements (such as the fifth row of Fig. 7) between the input images are not solved by SF, which makes the incorrect results of color transformation in some texture regions. And the extreme variation of luminance and exposure may make the SIFT descriptors invalid and to produce incorrect matching correspondences. 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. 7 (e)).

C. *Exposure fusion*

Another application of multimedia using our method is exposure fusion. The goal of this task is to achieve a full dynamic range of a scene which can preserve details and color appearances by fusing multiple exposure images. The most challenging problem is how to process the moving objects in the scene. A common solution is to align multiple exposure images (with moving objects) as registered images and then to fuse them. How to align these images is the most difficult and critical step for exposure fusion. This step usually is solved by searching accurate dense correspondences with moving objects among the input images taken under different exposure settings. Here, we use our method to find the dense correspondences between images and then align them as in [7]. Given these registered images, we fuse them using generalized random walk [33], [34], [35]. In summary, the exposure fusion procedures include the following four steps. The first step is to select the target image. Here, we use the normal-exposure image from the input sequence as the target image since more well-exposed and useful regions are in the normal-exposure image. Then the proposed method is applied to find the dense correspondences between the target image and other exposure images. In the third step, we respectively reconstruct this target image using the other exposure images to achieve the registered images. Finally, both the images that contain the reconstructed registered images and the target image are fused using the generalized random walk fusion method.

It is worth mentioning that the image color varies greatly among different exposure images, which leads to the color feature to be inaccurate. Thus, we reduce the weight of the color feature by setting $\alpha_1 = 1$ and $\alpha_2 = 5$ in the proposed method. We compare the proposed method with other methods of dense correspondences PM [15] and SF [19] to evaluate its performance in this application. The comparison results are shown in Fig. 8, where the input image sequences are taken from [36], [37], [38], [39]. It indicates that our approach achieves better performance with more details and well-exposed color, which is shown in Fig. 8 (d). All fusion results of PM (Fig. 8 (b)) seem to be blurred, especially the top two images in Fig. 8 (b), while our results contain more details for better visual performance because of the local search strategy. Some of them are with underexposure, such as the fourth image. Some images contain incorrect colors or textures, such as the regions indicated by red arrows. The reason for unsatisfactory performance may be that the PM method only matches the correspondences in the color space so that it fails to match the regions with high variation of colors. The fused images of SF (Fig. 8 (c)) have more normal exposure and contain more details than PM, especially the third image. However, there are many ghost artifacts in the regions with moving objects that have been indicated by the red arrows in fused results. This may be due to the limitation of large displacement in the SF method. What is more, SF fails to find the right correspondences in some flat regions, such as the cloud in the first image, which may be caused by the invalid SIFT features in these regions. While our method performs well in not only the regions with moving objects but also the flat regions. Since we use the global search to overcome the large displacement problem and combine the color and SIFT features to get more robust performance.

IV. Conclusion

We presented a novel SuperMatch method which finds the dense correspondences between two images in superpixel-pixel levels. The correspondence procedure is constructed with a new hierarchical superpixel to pixel (HSP2P) framework. The HSP2P framework first estimates the correspondences of the superpixels in the two images, and then matches the image pixels under the guidance of the corresponding superpixels. Color transfer is employed to design a color correction technique to rectify the color of the images, such that we can ignore the color of illumination variations during the matching procedure. Experimental results demonstrated that our HSP2P framework outperforms the state-of-the-art dense matching algorithms.

References

[1] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *IEEE CVPR*, vol. 1, 2006, pp. 519–528.

[2] S. Avidan, "Ensemble tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 261–271, 2007.

[3] L. Lin, K. Zeng, Y. Wang, Y.-Q. Xu, and S.-C. Zhu, "Video stylization: painterly rendering and optimization with content extraction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 4, pp. 577–590, 2013.

[4] C. Lipski, F. Klose, and M. Magnor, "Correspondence and depth-image based rendering a hybrid approach for free-viewpoint video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 6, pp. 942–951, 2014.

[5] J. Shen, Y. Du, and X. Li, "Interactive segmentation using constrained Laplacian optimization," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 24, no, 7, pp. 1088-1100, 2014.

[6] S. Bagon, O. Boiman, and M. Irani, "What is a good image segment? A unified approach to segment extraction," in *ECCV*, 2008, pp. 30–44.

[7] J. Hu, O. Gallo, and K. Pulli, "Exposure stacks of live scenes with hand-held cameras," in *ECCV*, 2012, pp. 499–512.

[8] O. Boiman and M. Irani, "Detecting irregularities in images and in video," *International Journal of Computer Vision*, vol. 74, no. 1, pp. 17–31, 2007.

[9] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani, "Summarizing visual data using bidirectional similarity," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–8.

[10] S. Battiato, A. R. Bruna, and G. Puglisi, "A robust block-based image/video registration approach for mobile imaging devices," *IEEE Transactions on Multimedia*, vol. 12, no. 7, pp. 622–635, 2010.

[11] J. Shen, X. Yang, Y. Jia, and X. Li, "Intrinsic images using optimization," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 3481–3487.

1051-8215 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2595321, IEEE Transactions on Circuits and Systems for Video Technology

[12] M. V. Afonso, J. C. Nascimento, and J. S. Marques, “Automatic estimation of multiple motion fields from video sequences using a region matching based approach,” *IEEE Transactions on Multimedia*, vol. 16, no. 1, pp. 1–14, 2014.

[13] X. Qin, J. Shen, X. Mao, X. Li, and Y. Jia, “Structured-patch optimization for dense correspondence,” *IEEE Transactions on Multimedia*, vol. 17, no. 3, pp. 295–306, 2015.

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

[15] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, “Patchmatch: a randomized correspondence algorithm for structural image editing,” *ACM Transactions on Graphics*, vol. 28, no. 3, pp. 341–352, 2009.

[16] F. Tombari, S. Mattoccia, L. D. Stefano, and E. Addimanda, “Classification and evaluation of cost aggregation methods for stereo correspondence,” in *IEEE CVPR*, 2008, pp. 1–8.

[17] S. Korman and S. Avidan, “Coherency sensitive hashing,” in *IEEE ICCV*, pp. 1607–1614, 2011.

[18] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein, “The generalized patchmatch correspondence algorithm,” in *Computer Vision-ECCV 2010*, 2010, pp. 29–43.

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

[20] Y. S. Heo, K. M. Lee, and S. U. Lee, “Robust stereo matching using adaptive normalized cross-correlation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 4, pp. 807–822, 2011.

[21] Y. Boykov, O. Veksler, and R. Zabih, “Fast approximate energy minimization via graph cuts,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[22] T. Brox and J. Malik, “Large displacement optical flow: descriptor matching in variational motion estimation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 500–513, 2011.

[23] L. Xu, J. Jia, and Y. Matsushita, “Motion detail preserving optical flow estimation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1744–1757, 2012.

[24] P. Indyk and R. Motwani, “Approximate nearest neighbors: towards removing the curse of dimensionality,” in *ACM symposium on Theory of computing*, ACM, 1998, pp. 604–613.

[25] Y. HaCohen, E. Shechtman, D. B. Goldman, and D. Lischinski, “Nonrigid dense correspondence with applications for image enhancement,” in *ACM Transactions on Graphics*, vol. 30, no. 4, 2011, p. 70.

[26] F. Besse, C. Rother, A. Fitzgibbon, and J. Kautz, “Pmbp: Patchmatch belief propagation for correspondence field estimation,” *International Journal of Computer Vision*, vol. 110, no. 1, pp. 2–13, 2014.

[27] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, “Slic superpixels compared to state-of-the-art superpixel methods,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[28] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[29] S. Cho and S. Lee, “Fast motion deblurring,” *ACM Transactions on Graphics*, vol. 28, no. 5, article no. 145, 2009.

[30] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley, “Color transfer between images,” *IEEE Computer Graphics and Applications Magazine*, vol. 21, no. 5, pp. 34–41, 2001.

[31] F. Pitié, A. C. Kokaram, and R. Dahyot, “Automated colour grading using colour distribution transfer,” *Computer Vision and Image Understanding*, vol. 107, pp. 123–137, 2007.

[32] J. Kim, C. Liu, F. Sha, and K. Grauman, “Deformable spatial pyramid matching for fast dense correspondences,” in *IEEE CVPR*, pp. 2307–2314, 2013.

[33] R. Shen, I. Cheng, J. Shi, and A. Basu, “Generalized random walks for fusion of multi-exposure images,” *IEEE Transactions on Image Processing*, vol. 20, no. 12, pp. 3634–3646, 2011.

[34] J. Shen, Y. Zhao, S. Yan, X. Li *et al.*, “Exposure fusion using boosting laplacian pyramid.” *IEEE Transactions on Cybernetics*, vol. 44, no. 9, pp. 1579–1590, 2014.

[35] J. Shen, Y. Du, W. Wang, and X. Li, “Lazy random walks for superpixel segmentation,” *IEEE Transactions on Image Processing*, vol. 23, no. 4, pp. 1451–1462, 2014.

[36] F. Pece and J. Kautz, “Bitmap movement detection: HDR for dynamic scenes,” in *Visual Media Production (CVMP), 2010 Conference on*. IEEE, 2010, pp. 1–8.

[37] W. Zhang and W.-K. Cham, “Gradient-directed composition of multi-exposure images,” in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 530–536.

[38] P. Sen, N. K. Kalantari, M. Yaesoubi, S. Darabi, D. B. Goldman, and E. Shechtman, “Robust patch-based hdr reconstruction of dynamic scenes.” *ACM Transactions on Graphics*, vol. 31, no. 6, p. 203, 2012.

[39] J. Hu, O. Gallo, K. Pulli, and X. Sun, “Hdr deghosting: How to deal with saturation?” in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2013, pp. 1163–1170.

Xingping Dong received the B.S. degree in computer science from Xiamen University in 2012. He is currently working toward the Ph.D. degree in the School of Computer Science, Beijing Institute of Technology, Beijing, China. His current research interests include random walks and image segmentation.

Jianbing Shen (M’11-SM’12) is a Professor with the School of Computer Science, Beijing Institute of Technology, Beijing, China. His research interests include computer vision and multimedia processing. He has published about 50 journal and conference papers such as *IEEE TIP*, *IEEE TCSVT*, *IEEE TCYB*, *IEEE TMM*, *IEEE CVPR*, and *IEEE ICCV*. He has also obtained many flagship honors including the Fok Ying Tung Education Foundation from Ministry of Education, the Program for Beijing Excellent Youth Talents from Beijing Municipal Education Commission, and the Program for New Century Excellent Talents from Ministry of Education. His research interests include computer vision and multimedia processing. He is on the editorial boards of *Neurocomputing*.

Ling Shao (M’09-SM’10) is a Full Professor and Head of the Computer Vision and Artificial Intelligence Group with the Department of Computer Science and Digital Technologies at Northumbria University, Newcastle upon Tyne and an Advanced Visiting Fellow with the Department of Electronic and Electrical Engineering at the University of Sheffield. His research interests include Computer Vision, Image Processing, Pattern Recognition and Machine Learning. He is an Associate Editor of *IEEE Transactions on Image Processing*, *IEEE Transactions on Circuits and Systems for Video Technology*, *IEEE Transactions on Neural Networks and Learning Systems*, and other journals. He is a Fellow of the British Computer Society, a Fellow of the IET, and a Life Member of the ACM.

1051-8215 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.