DENSE IMAGE CORRESPONDENCE UNDER LARGE APPEARANCE VARIATIONS

Linlin Liu[†] *Kok-Lim Low*[†] *Wen-Yan Lin*[⋆]

[†]Department of Computer Science, National University of Singapore
[⋆]Department of Computing, Oxford Brookes University

ABSTRACT

This paper addresses the difficult problem of finding dense correspondence across images with large appearance variations. Our method uses multiple feature samples at each pixel to deal with the appearance variations based on our observation that pre-defined single feature sample provides poor results in nearest neighbor matching. We apply the idea in a flow-based matching framework and utilize the best feature sample for each pixel to determine the flow field. We propose a novel energy function and use dual-layer loopy belief propagation to minimize it where the correspondence, the feature scale and rotation parameters are solved simultaneously. Our method is effective and produces generally better results.

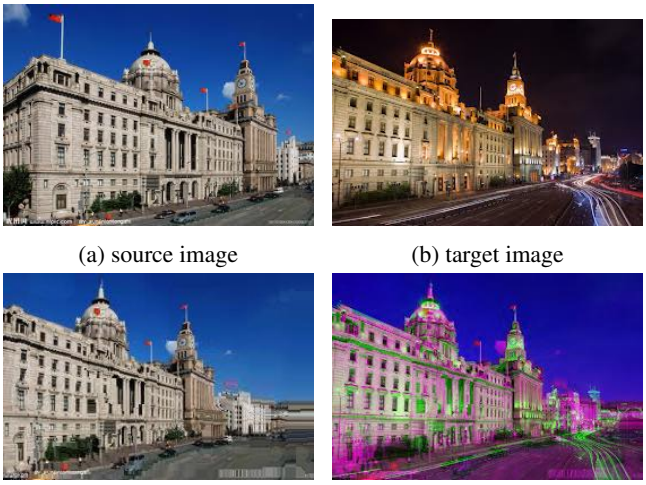
Index Terms— image registration, image matching, image motion analysis, SIFT Flow, belief propagation

1. INTRODUCTION

Image correspondence has many applications such as in structure-from-motion, image retrieval, and object recognition. Although it has been extensively studied, it remains a long standing problem in computer vision. One major challenge is the matching of images acquired under a variety of different imaging conditions, which include the variations in the camera, viewpoint, lighting, and even in the scene or object itself. These factors can result in significant appearance differences in the images of the same scene or object, and in this paper, we propose a method to compute dense correspondences between such images. Fig. 1 shows an example result from our method.

HaCohen et al. [1] fit a parametric color transfer model to handle color variation. However when color changes are irregular, their method is often unable to compute an appropriate color transfer model, which makes it unsuitable for matching images with more complex differences. Another drawback is that it does not produce sufficiently dense correspondence.

Nearest-neighbor matching of sparse features, e.g. Scale Invariant Feature Transform (SIFT) [2], does not work well for significant appearance variations. The features are designed to be discriminative. Therefore, features of the same point are often not the nearest neighbors in the feature space when its appearance changes significantly. However, as demonstrated by Lin and Liu et al. [3], such features can still be used to establish correspondences. They match two sets of sparse SIFT features by maximizing the expectation of the correspondence probability. The correspondences are parametrized by a series of geometric transformations and solved by a hierarchical approach. However, their method is complicated and computationally expensive. On the contrary, our method is much simpler and more efficient.



(a) source image (b) target image

(c) warp source image (d) overlay

Fig. 1: Matching images with large appearance variations. The source image is matched and warped to the target image. In (d), the red and blue channels of the warped source image are overlayed on the green channel of the target image for comparison.

Similar to SIFT Flow [4], our method computes dense SIFT features for the input images. However, as inspired by the work of Lin and Liu et al. [3], we avoid matching pixels at pre-defined feature scales and rotations, and instead, feature scales and rotations are treated as unknown variables that our method tries to solve for. It does this by searching for the correspondences, feature scales and rotations simultaneously.

SIFT Flow uses only one pre-defined feature scale and rotation for each pixel thus has problem with large appearance variations. Optical flow [5] is another dense correspondence

method. However its brightness constancy assumption does not hold for our problem. Lin et al. [6] compute a dense correspondence field and use it to stitch images. But large appearance variations render this algorithm ineffective.

2. ALGORITHM

Our objective is to establish correspondence for each pixel between two images. We formulate the dense correspondence using a flow-based framework. We want to find a flow field $\mathbf{w}$ between two images I_1 and I_2 such that after applying the transformation $\mathbf{w}$ to I_1, the new image $\hat{I}_1$ looks geometrically similar to I_2. The flow field $\mathbf{w}$ is defined for all pixels $\mathbf{p} = (x, y)$ and $\mathbf{w}(\mathbf{p}) = (u(\mathbf{p}), v(\mathbf{p}))$ is the flow vector at $\mathbf{p}$.

We measure similarity between two pixels using their SIFT descriptors. A SIFT descriptor can be computed at any pixel once the scale and rotation of the feature patch are given. The scale is the size of the feature patch, i.e. radius of circular patch, and the rotation is the angle that the local feature frame is rotated relative to image frame.

In an image I_i, a SIFT descriptor at a pixel $\mathbf{p}$ is denoted $\mathbf{d_i}(\mathbf{p}, s, r)$, where s and r are the scale and rotation parameters at which the descriptor is computed. Our energy function is very similar to that of SIFT Flow [4]. The main difference lies in the data term, in which we consider multiple SIFT descriptors at different scales and rotations for each pixel, while SIFT Flow considers only one descriptor for each pixel. By doing this, our energy function allows each pixel to choose the descriptor that gives the smallest matching error, whereas with only one descriptor, a pixel might miss the correct match when their descriptors differ a lot due to significant appearance variations.

Our new energy function for finding dense correspondence across two images is

$$E(\mathbf{w}, \mathbf{s_1}, \mathbf{r_1}, \mathbf{s_2}, \mathbf{r_2}) =$$

$$\sum_{\mathbf{p}} \min(\|\mathbf{d_1}(\mathbf{p}, s_1, r_1) - \mathbf{d_2}(\mathbf{p} + \mathbf{w}(\mathbf{p}), s_2, r_2)\|_1, \alpha) + \tag{1}$$

$$\sum_{(\mathbf{p},\mathbf{q})\in\xi} \min(\eta|u(\mathbf{p}) - u(\mathbf{q})|, \beta) + \min(\eta|v(\mathbf{p}) - v(\mathbf{q})|, \beta) + \tag{2}$$

$$\sum_{\mathbf{p}} \gamma(|u(\mathbf{p})| + |v(\mathbf{p})|). \tag{3}$$

$\mathbf{s_1}, \mathbf{r_1}, \mathbf{s_2}, \mathbf{r_2}$ contain the SIFT scales and rotations for all pixels in images I_1 and I_2 respectively. Note that s_1, r_1 are functions of $\mathbf{p}$, and s_2, r_2 are functions of $\mathbf{p} + \mathbf{w}(\mathbf{p})$. The energy E contains three terms: data term, smoothness term and small displacement term. $\|\cdot\|_1$ denotes L_1 norm. We use truncated L_1 norm for the data term and the smoothness term, and α and β are the truncation parameters. η and γ control the weight of the smoothness term and the small displacement term. ξ is the set of all the spatial neighborhoods.

Different from SIFT Flow, our formulation treats the scale and rotation of each SIFT feature as unknown variables. To make the problem tractable, we sample the scale and rotation parameter space and pre-compute the SIFT descriptors accordingly. To further reduce the complexity, we do this only for I_1 while for I_2 we fix s_2, r_2 to some pre-defined values. At each pixel, we impose the smoothness constraints with its four neighboring pixels.

2.1. Energy Minimization

We minimize E using dual-layer loopy belief propagation [7][8][9], which decouples the horizontal and vertical components of $\mathbf{w}$ into two layers $\mathbf{u}$ and $\mathbf{v}$ respectively. We construct a graph structure very similar to that of SIFT Flow[4]. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph. The set of vertices $\mathcal{V}$ contains two copies of set $\mathcal{P}$ where each element $\mathbf{p} \in \mathcal{P}$ represents a pixel in the image. We denote the two layers of vertices as $\mathcal{P}_1$ and $\mathcal{P}_2$, thus $\mathcal{V} = \mathcal{P}_1 \cup \mathcal{P}_2$. Let $\mathcal{E} = \mathcal{E}_1 \cup \mathcal{E}_2 \cup \mathcal{E}_{12}$, where $\mathcal{E}_1$ and $\mathcal{E}_2$ are copies of the same set of intra-layer edges that connect each pixel to its 4-neighborhood, and $\mathcal{E}_{12}$ is the set of inter-layer edges that connect each vertex $\mathbf{p}_1$ to its counterpart $\mathbf{p}_2$ in the other layer. Note that the labels for each vertex consist of not only the horizontal and vertical displacements but also the SIFT scale and rotation. We have more labels to consider than in SIFT Flow[4], which makes the data term of our model more difficult to compute.

The data term in E is encoded in the inter-layer edges in $\mathcal{E}_{12}$ as

$$\begin{aligned}\theta_{\mathbf{p_1p_2}}(u(\mathbf{p}), v(\mathbf{p})) = \min_{s_1,r_1,s_2,r_2} \min(|\mathbf{d_1}(\mathbf{p}, s_1, r_1) - \\ \mathbf{d_2}(\mathbf{p} + (u(\mathbf{p}), v(\mathbf{p})), s_2, r_2)|, \alpha), \\ \mathbf{p}_1 \in \mathcal{P}_1, \mathbf{p}_2 \in \mathcal{P}_2, \mathbf{p}_1\mathbf{p}_2 \in \mathcal{E}_{12}.\end{aligned} \tag{4}$$

We minimize over the scale and rotation variables s_1, r_1, s_2, r_2 and take the ones that give the smallest matching error in the data term. The smoothness term is encoded in the intra-layer edges as

$$\theta_{\mathbf{pq}}(u(\mathbf{p}), u(\mathbf{q})) = \min(\eta|u(\mathbf{p}) - u(\mathbf{q})|, \beta), \mathbf{p}, \mathbf{q} \in \mathcal{E}_1, \tag{5}$$

and

$$\theta_{\mathbf{pq}}(v(\mathbf{p}), v(\mathbf{q})) = \min(\eta|v(\mathbf{p}) - v(\mathbf{q})|, \beta), \mathbf{p}, \mathbf{q} \in \mathcal{E}_2. \tag{6}$$

The small displacement term is encoded in the vertices as

$$\theta_{\mathbf{p}}(u(\mathbf{p})) = \gamma|u(\mathbf{p})|, \mathbf{p} \in \mathcal{P}_1, \tag{7}$$

and

$$\theta_{\mathbf{p}}(v(\mathbf{p})) = \gamma|v(\mathbf{p})|, \mathbf{p} \in \mathcal{P}_2. \tag{8}$$

There are two kinds of messages being passed in the belief network—inter-layer message and intra-layer message. An

inter-layer message is passed between the two layers and minimizes E with respect to the data term:

$$m_{\mathbf{p}_1\mathbf{p}_2}(v(\mathbf{p})) \leftarrow \min_{u(\mathbf{p})}\{\theta_{\mathbf{p}_1\mathbf{p}_2}(u(\mathbf{p}), v(\mathbf{p})) + \sum_{(\mathbf{p},\mathbf{q})\in\xi} n_{\mathbf{q}\mathbf{p}_1}(u(\mathbf{p})) + \theta_{\mathbf{p}}(u(\mathbf{p}))\}. \quad (9)$$

We use $m_{\mathbf{p}_1\mathbf{p}_2}(v(\mathbf{p}))$ to denote the message passed from $\mathbf{p}_1$ in Layer 1 to $\mathbf{p}_2$ in Layer 2 with the belief of $\mathbf{p}_2$ having label $v(\mathbf{p})$. Message passed from Layer 2 to Layer 1 is defined in the same way. Similarly, let $n_{\mathbf{p}\mathbf{q}}(u(\mathbf{q}))$ be the belief that $\mathbf{q}$ has label $u(\mathbf{q})$. Without loss of generality, we define the intra-layer message for Layer 1 as

$$n_{\mathbf{p}\mathbf{q}}(u(\mathbf{q})) \leftarrow \min_{u(\mathbf{p})}\{\theta_{\mathbf{p}\mathbf{q}}(u(\mathbf{p}), u(\mathbf{q})) + \sum_{(\mathbf{p},\mathbf{q}')\in\xi} n_{\mathbf{q}'\mathbf{p}}(u(\mathbf{p})) + m_{\mathbf{p}_2\mathbf{p}_1}(u(\mathbf{p})) + \theta_{\mathbf{p}}(u(\mathbf{p}))\}. \quad (10)$$

We define the same for Layer 2. Intra-layer messages impose smoothness constraints among adjacent pixels. We use the same distance transform function to reduce the complexity and employ the same coarse-to-fine matching scheme as in SIFT Flow [4].

3. IMPLEMENTATION DETAILS AND RESULTS

SIFT [2] is an algorithm that detects keypoints and extracts feature descriptors. For our purpose we only use its feature extraction component. For every pixel in an image, we divide its neighborhood into a 4×4 cell array, quantize the orientations into 8 bins in each cell, and obtain a $4 \times 4 \times 8 = 128$-dimensional vector as the SIFT descriptor for a pixel. For image I_2, the cell array size (scale) is 12×12 pixels and rotation is 0. Each pixel is represented by one SIFT descriptor. For image I_1, each pixel takes 24 SIFT descriptors computed at scales $6 \times 6, 12 \times 12, 24 \times 24$ and 8 rotations from 0 to 2π. We set parameters $\beta = 200 \times 255, \gamma = 0.005 \times 255$, and $\eta = 2 \times 255$ for all the results in this paper. The data term truncation parameter α is set as the median of the matching errors for all possible correspondences. In the coarse-to-fine matching scheme, we build 4-level image pyramids. In each level, each pixel is only allowed to move in a local window. The window sizes are $21 \times 21, 15 \times 15, 13 \times 13$, and 11×11 starting from the top level. We implement our method using a hybrid of C++ and Matlab code. We run our experiments on a PC with Intel Core2 Quad CPU Q9550 @ 2.83GHz and 4GB memory. For image size 400×300, our implementation takes 5 mins to run. Computational time can be reduced dramatically by more intelligently sampling the feature space which leads to far less number of graph states.

To the best of our knowledge, there is no standard dataset for evaluating the matching of images with large appearance variations. We adopt an approach similar to Lin and Liu et al. [3]. Given two input images, we compute the flow field $\mathbf{w}$ and use it to warp the source image to the target image. Then we overlay the red and blue channels of the warped image with the green channel of the target image. We use images from Google image search.

The results on two datasets, Mount Rushmore (MR) and Golden Pavilion (GP) are presented in Fig. 2 and Fig. 3. Each dataset consists of three image pairs with large appearance variations. We compare our method with two other state-of-the-art dense correspondence algorithms—SIFT Flow [4] and the method of Lin and Liu et al. [3]. Although SIFT Flow is not specifically designed to cope with large appearance variations, it still can work to some degree. The method of Lin and Liu et al. targets exactly the same problem as ours and shows very good results. For the MR dataset shown in Fig. 2, our method gives significantly better results than SIFT Flow, which produces severe artifacts in all three cases. The method of Lin and Liu et al. fails in the second case (second column) while succeeds in the other two cases. In all cases, our method produces smaller misalignment errors, which can be seen from the overlay images. For the first case (first column) of GP in Fig. 3, we obtain comparable results as that of Lin and Liu et al. SIFT Flow produces some small artifacts at the second-level roof of the pavilion. The second case (second column) of GP exhibits extreme color changes that might have caused the other two methods to fail. The structure of the pavilion is destroyed by SIFT Flow, and the method of Lin and Liu et al. cannot compute the correct smoothly varying affine transformation. Another reason for the failure of the method of Lin and Liu et al. is that its warping function is too rigid and thus is particularly vulnerable to changes in viewpoint. While our method still does not handle very large viewpoint changes (>30 degrees), it is less sensitive to this than the method of Lin and Liu et al. because it does not force a continuous warping field onto the image pairs.

4. CONCLUSION

We have introduced a novel energy function for matching images with large appearance variations. Our method considers multiple feature samples at each pixel in a flow-based matching framework. From our experimental results, we have observed that compared to previous methods, our method achieves better or comparable results. However, our method has several limitations. Besides the limitation to handle very large viewpoint changes, for future work, we would like to also impose smoothness constraints on the scale and rotation parameters, which may be able to reduce erroneous correspondences.

Acknowledgment: This work is partly supported by Singapore MOE Academic Research Fund (Project Number: T1 251RES1104).

(a) Source images

(b) Target images

(c) Our warp

(d) Our overlay

(e) SIFT Flow's warp

(f) SIFT Flow's overlay

(g) Lin and Liu et al.'s warp

(h) Lin and Liu et al.'s overlay

Fig. 2: Mount Rushmore (MR). (a) Source images, (b) target images, (c)(d) our results, (e)(f) results of SIFT Flow, (g)(h) results of Lin and Liu et al. Images (c), (e) and (g) show the warped source images, and images (d), (f) and (h) show the overlay of the warped source images on the target images.

(a) Source images

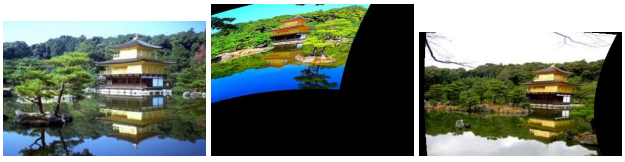
(b) Target images

(c) Our warp

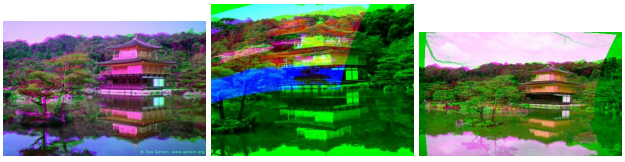
(d) Our overlay

(e) SIFT Flow's warp

(f) SIFT Flow's overlay



(g) Lin and Liu et al.'s warp



(h) Lin and Liu et al.'s overlay

Fig. 3: Golden Pavilion (GP). (a) Source images, (b) target images, (c)(d) our results, (e)(f) results of SIFT Flow, (g)(h) results of Lin and Liu et al. Images (c), (e) and (g) show the warped source images, and images (d), (f) and (h) show the overlay of the warped source images on the target images.

5. REFERENCES

[1] Yoav HaCohen, Eli Shechtman, Dan B Goldman, and Dani Lischinski, "Non-rigid dense correspondence with applications for image enhancement," *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2011)*, vol. 30, no. 4, pp. 70:1–70:9, 2011.

[2] D. G. Lowe, "Object recognition from local scale-invariant features," *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, vol. 2, pp. 1150–1157, September 1999.

[3] Wen-Yan Lin, Linlin Liu, Y. Matsushita, Kok-Lim Low, and Siying Liu, "Aligning images in the wild," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, June 2012.

[4] Ce 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, May 2011.

[5] Berthold K. P. Horn and Brian G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1-3, pp. 185–204, August 1981.

[6] Wen-Yan Lin, Siying Liu, Y. Matsushita, Tian-Tsong Ng, and Loong-Fah Cheong, "Smoothly varying affine stitching," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 345–352, June 2011.

[7] Alexander Shekhovtsov, Ivan Kovtun, and Vclav Hlav, "Efficient MRF deformation model for non-rigid image matching," *Computer Vision and Image Understanding*, vol. 112, no. 1, pp. 91–99, 2008.

[8] Kevin P. Murphy, Yair Weiss, and Michael I. Jordan, "Loopy belief propagation for approximate inference: an empirical study," *Proceedings of Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 467–475, 1999.

[9] Pedro F. Felzenszwalb and Daniel P. Huttenlocher, "Efficient belief propagation for early vision," *International Journal of Computer Vision*, vol. 70, pp. 41–54, 2006.