Dense Image Registration using Sparse Coding and Belief Propagation

Aminmohammad Roozgard, Nafise Barzigar, Samuel Cheng and Pramode Verma
School of Electrical and Computer Engineering University of Oklahoma
Tulsa, OK 74135-2512, USA

***Abstract*—Image registration as a basic task in image processing was studied widely in the literature. It is an important preprocessing step in different applications such as medical imaging, super resolution, and remote sensing. In this paper we proposed a novel dense registration method based on sparse coding and belief propagation. We used image blocks as features, and then we employed sparse coding to find a set of candidate points. To select optimum matches, belief propagation was subsequently applied on these candidate points. Experimental results show that the proposed approach is able to robustly register scenes and is competitive as compared to optical flow.**

***Index Terms*—Dense Image Registration, Sparse Coding, Belief Propagation**

I. Introduction

Image registration is the process of overlaying two or more images of the same scene taken at different times, from different viewpoints, or using different capturing modules. Registration is required in many applications including remote sensing, super resolution, and change detection. For example, in medical images, registration techniques have been used to align an MRI to a CT image [1].

The registration methods can be divided into two major classes: global registration and local (dense) registration [2], [3], [4]. In the former case, a registration method such as [3] only finds matches of a few key points and then approximates the corresponding matches of the other points based on the key point matches. A strong assumption is typically imposed on the allowed transformation from the test image to the reference image. On the other hand, dense image registration such as [4] attempts to find a matched point for each pixel of the test image directly. Unlike global registration, there is almost no restriction on the allowed transformation and thus considerably more "parameters" have to be estimated. This makes dense registration a much more challenging problem than global registration.

A classic dense registration algorithm is the Lucas-Kanade optical flow method [5]. It is commonly used to estimate the displacement field between two images with similar scenes. Since the inception of the Lucas-Kanade method, significant advancement on spatial registration methods have been observed in the last three decades [6], [7], [8]. For example, in [8], the authors studied the effect of the combined assumption of brightness constancy and gradient constancy on optical flow.

In this paper we propose a novel dense registration technique by aligning local sparse features of two images. We use the dictionary of all features to find candidate points which cover all possible movements of each pixel. Since an overcomplete dictionary is constructed directly from padding the features obtained by the reference data, no extra time is spent in training. The algorithm then finds a set of candidate points for each incoming test data point using the constructed dictionary, and finally the optimum candidates are chosen using belief propagation [9]. As we will see, the proposed methods can reconstruct the edges with high fidelity and can find match points in the background as well as foreground objects. Additionally, we achieve better results than Optical flow [8], [10] which show proposed algorithm can find the accurate location for each object that are missed by other methods. Generally, it is more sensitive to scene deformation and can not synthesize test image if there are many changes of the test image comparing to the reference image. In contrast, proposed algorithm appears to be more robust in handling complex scenes with multi

978-1-4577-1180-0/11/$26.00

objects and wide-baseline views.

The rest of this paper is structured as follows. Section II reviews the background of sparse coding. Section III introduces our proposed method. Finally in Section IV, we show and discuss our simulation results, and follow with a brief conclusion in Section V.

II. Background Of Sparse Coding

Consider a signal $y \in \mathbb{R}^M$ and a fat matrix $D \in \mathbb{R}^{M \times N}$, where we say the matrix is "fat" since $M < N$. We are interested in representing y with the column space of $D \in \mathbb{R}^{M \times N}$, i.e., finding $\alpha \in \mathbb{R}^N$ such that $y = D\alpha$. Since D is fat, α is not unique. However, if we also restrict α to be the sparsest vector to satisfy $y = D\alpha$ (i.e., α that have fewest number of non-zero elements), then in theory there is a unique solution. Sparse coding precisely considers the aforementioned problem of how to find a sparse α such that $y = D\alpha$ is satisfied.

Mathematically, we can write the problem as

$$\hat{\alpha} = arg\ min||\alpha||_0 \ \ subject \ \ to \ \ y = D\alpha. \tag{1}$$

However, this l^0 optimization problem is NP-complete [11] and thus several alternative methods have been proposed to solve it. For example, when a sufficiently sparse solution actually exists, substituting the l^1 norm for the l^0 pseudo-norm in (1) as below

$$\hat{\alpha} = arg\ min||\alpha||_1 \ \ subject \ \ to \ \ y = D\alpha \tag{2}$$

will still result in the same solution [11]. Moreover, solving this modified problem is much easier since it can be readily transformed into a linear programming problem. Besides linear programming, many other suboptimal techniques have been proposed to solve (2) including orthogonal matching pursuit [12] and gradient projection [13].

III. Proposed Method

As mentioned in Section I, in some applications we need dense registration so that for each point of the first image (*reference image*) a corresponding match point will be found on the second image (*test image*). This section describes the implementation's details of our proposed registration method. We divide the registration process into four steps as shown in Sections III-A, III-B, III-C and III-D.

A. Dense Feature Extraction and Dictionary Construction

In the first step of our proposed image registration method, we need to extract the features from the reference image $\mathcal{X}$ and the test image $\mathcal{Y}$. To achieve this purpose, we consider a patch of size $(2k+1)^2$ containing neighboring pixels around each pixel on both images, where k is a positive integer. For each pixel p_{ij} in the test image $\mathcal{Y}$, we vectorized the patch of p_{ij} to a feature vector $Y_{ij} \in \mathbb{R}^{S \times 1}$, where $S = (2k+1)^2$. A 3-D test feature image $Y \in \mathbb{R}^{M \times N \times S}$ is then constructed from Y_{ij} as follows

$$Y = \begin{bmatrix} Y_{1,1} & Y_{1,2} & \cdots & Y_{1,N} \\ Y_{2,1} & Y_{2,2} & \cdots & Y_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{M,1} & Y_{M,2} & \cdots & Y_{M,N} \end{bmatrix}. \tag{3}$$

To match the extracted features of the test image to corresponded extracted features of the reference image, we create a dictionary which contains feature vectors constructed just as the aforementioned procedure but with the reference image instead. More precisely, a dictionary $D \in \mathbb{R}^{S \times MN}$ is constructed with all possible vector $X_{ij} \in \mathbb{R}^{S \times 1}$ as D's column vectors, where X_{ij} is created in the same manner as Y_{ij} but from reference image $\mathcal{X}$ instead. Thus, we can write D as

$$D = [X_{1,1} X_{1,2} \cdots X_{1,N} X_{2,1} \cdots X_{M,N}]. \tag{4}$$

We will normalize dictionary D to guarantee the norm of each feature vector to be 1.

B. Finding Candidate Points via Sparse Coding

The goal of this step is to identify candidate patches that look most similar to an input patch in the test image. A naïve approach will be to compute the mean square error (MSE) of the input patch with each possible patch of the reference image and to select patches that have the smallest MSEs. However, the candidate patches constructed with this approach are likely to generate patches that are all concentrated in a small region since a small shift from the most similar patch generally do not decrease similarities very sharply unless for very high frequency patches. Consequently, this approach will result in patches with very low diversity.

Instead, we propose to find candidate match points using sparse coding. The intuition is that we should be able to construct a test patch out of good candidate patches (thus they correspond to a sparse coding solution) if these candidate patches are similar enough to the test patch. Moreover, since instead of simply returning the most similar patches, sparse coding outputs patches that can reconstruct the test patch through linear combination, the resulting patches of sparse coding are likely to be complementary to each other and thus provide a better diversity than the naïve solution. Mathematically, we try to solve the following sparse coding problem of finding the most sparse coefficient vectors α_{ij} such that

$$Y_{ij} = D\alpha_{ij}. \tag{5}$$

To solve (5), we employed orthogonal matching pursuit [12] in this paper. After finding the sparse representation vector α_{ij}, we pick up the n largest coefficients of α_{ij} as our n candidates for the next step.

C. Applying Belief Propagation

As described in Section III-B, we extracted n candidate points from the reference image for each point of test image. Now, we use those candidate points from the reference data as our "prior knowledge" to find the best match point for the test data. Note that in III-B, we select candidate match points only based on the local characteristic of an input pixel but ignored any geometric characteristics of the matches. For example, except for few places near object boundaries, one would expect that nearby pixels in the test image should also match to pixels that are closed to each other in the reference image. To incorporate this geometric characteristics, we model the problem by factor graph and apply belief propagation to identify the best matches similar to [3].

1) Factor Graph for Image Modeling: A factor graph is a bipartite graph which contains two types of nodes: The *Variable node*s which is assigned to the variables and *Factor node*s that show the relation between variables. In our case, we assign a variable node for each pixel on the test image and connect each pair of neighboring pixels with a factor node as in Fig. 1. Also, we introduce one extra factor node to take care of the prior knowledge obtained in the sparse coding step for each pixel of the test image.

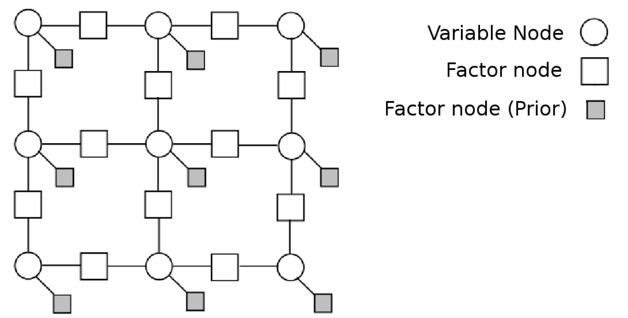

Fig. 1. A part of the test image factor graph

2) BP Algorithm: Belief propagation (BP) [9] is an approximate inference method on graphical models such as factor graphs. It was performed by passing message through the factor graph of a given problem. We apply BP on factor graph of test image with n candidate points as prior knowledge. BP changes the probability of candidate points based on the probabilities of their neighbors. Define $N(i)$ and $N(a)$ as two sets of neighbors of a variable node i and a factor node a, respectively, and denote $m_{i\to a}$ and $m_{a\to i}$ as the forward and backward messages from node i to node a. A message itself is a vectors containing current beliefs of a node mapping to all candidate pixels in the reference image. For example, $m_{a\to i}(x_i)$ can be interpreted as the belief of node a of how probable that the pixel of node i in the test image should map to location x_i in the reference image. Message updates for $m_{i\to a}$ and $m_{a\to i}$ will be based on the messages received by the incoming messages towards nodes i and a, respectively. More precisely, they are given by [9]

$$m_{i\to a}(x_i) = \prod_{b\in N(i)\backslash a} m_{b\to i}(x_i), \tag{6}$$

$$m_{a\to i}(x_i) = \sum_{x_a\backslash x_i} f(x_a) \prod_{j\in N(a)\backslash i} m_{j\to a}(x_j), \tag{7}$$

where we use $N(a)\backslash i$ to denote the neighbor of node a excluding node i.

According to our factor graph topology which each factor node is exactly connected to two variable nodes, messages from factor node to variable node can be simplified to

$$m_{a\to i}(x_i) = \sum_{x_j} f(x_i, x_j) m_{j\to a}(x_j), \tag{8}$$

where factor node a is between variable nodes i and j. In our model, the factor function $f(x_i, x_j)$, which can be interpreted as the local belief of having x_i and x_j at nodes i and j, can be used to impose the geometric constraint described earlier. Intuitively, since x_i and x_j are the corresponding mapped match points in the reference image of two neighboring pixels in the test image, we expect the probability of getting x_i and x_j decreases as their distance increases. Therefore, in this paper, we model the function of factor node between two particular variable nodes x_i and x_i as

$$f(x_i, x_j) = e^{-\frac{||x_i - x_j||_2}{\sigma^2}} \tag{9}$$

where σ^2 is a parameter to control the relative strength of the geometric constraint imposed by a neighboring node. If we increase the value of σ^2, the belief of each variable node will have less effect on its neighbors.

D. Interpreting BP Result

After applying several BP iterations, we obtain the updated probabilities for the candidate points of each pixel in the test. These probabilities can be used for the registration of the test image. In our work, we select the most probable point after the BP step as the best match point. We assume that our registration method successfully finds a match for an input point if the most probable candidate has belief larger than a threshold θ. Otherwise, we assume no best match is found. The latter can be due to failure of our method but it is also possible that a match indeed does not exist because of occlusion or boundary issues.

IV. Experimental Results

For better illustration of the results, we exchange the value of each point with the selected candidate point from the reference image if BP fails to find a match point (the probability of the selected point is more than the threshold θ as described in the last section). If the algorithm fails to find a candidate point with probability larger than θ, we use the pixel of the test image on that position. This type of points will appear when the test image is not entirely covered by the scene of the reference image.

In our experiment, we set $n = 5$, $k = 3$, $\theta = 0.5$, $\sigma^2 = 50$. We compared our proposed method with the high-accuracy optical flow estimation method by Brox *et al.* [8] with results as shown in Fig. 2 and Fig. 3. Figs. 2a and 2b illustrate a scene captured from two different angles. As some objects appeared in the test image had been occluded by other objects in the reference image, not all pixels in the test image can be matched to the reference image. Figs. 2c and 2d show the registration results using compared and our proposed methods. In Fig. 2e, we drew red lines manually to enclose the errors of optical flow in comparison to our proposed method. Clearly, for the optical flow method, the chair was split and the wall's corner location was not correct. Also, the whiteboard was stretched and the white postcard was still in its same place in the reference image. In contrast, as shown in Fig. 2f, our proposed method accurately reconstructed the chair and corner of the walls. Also, the shape and location of the whiteboard was not distorted and the postcard was registered to the true location.

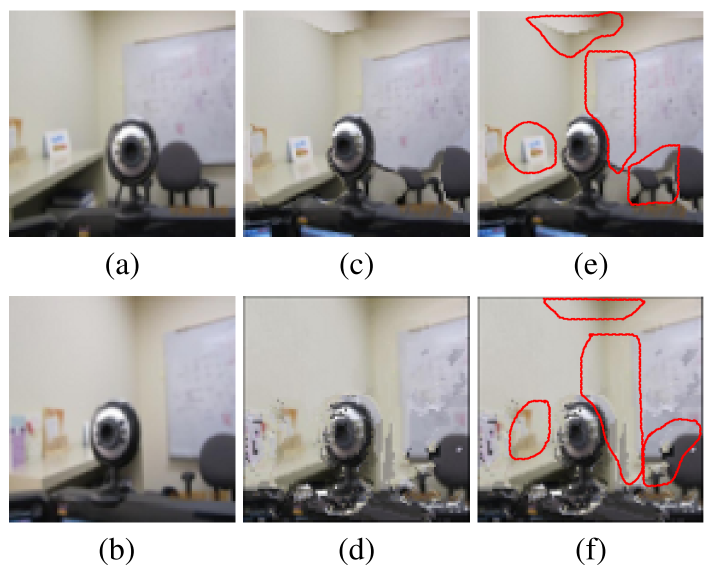

Fig. 2. a) Reference Image; b) Test Image; c) Optical flow; d) Proposed Method; e) Weaknesses in optical flow; f) Strengths of proposed method.

Fig. 3 is a simple scene that contains two cookies which had not been masked by other objects. Again, optical flow could not find the location of the objects correctly (see Fig. 3e). In particular, the centers of cookies are shifted and their boundaries are corrupted. In contrast to optical flow, the proposed method registers both the centers and the boundaries of the cookies precisely. Furthermore, our method kept the elliptical shape of both cookies with the appropriate details. Note that we did not use any shift or scale invariant features in our approach, and yet our proposed method can recognize the rotation and scaling change in the scenes. While the extracted features of our approach are similar to the optical flow features, our proposed method excels in finding the exact locations of objects and recognize the different movements of the objects.

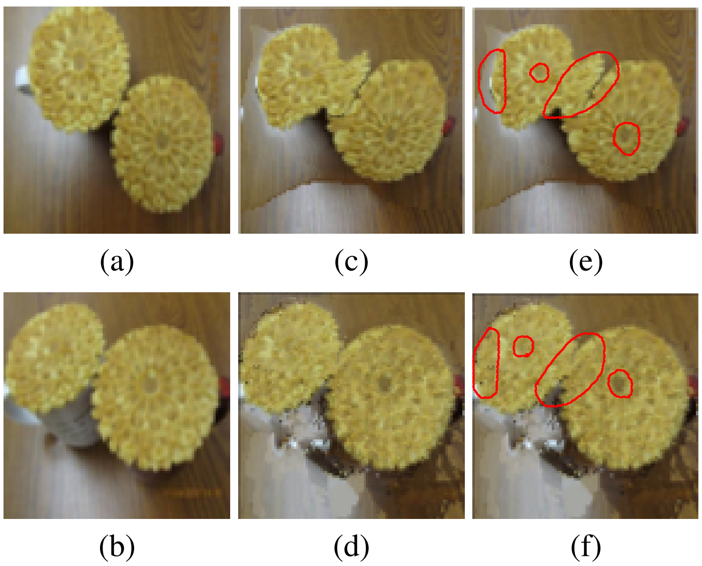

Fig. 3. a) Reference Image; b) Test Image; c) Optical flow; d) Proposed Method; e) Weaknesses in optical flow; f) Strengths of proposed method.

V. Conclusion

In conclusion, we have proposed a novel registration method based on a sparse coding and belief propagation. Our technique performs registration by first running sparse coding over an overcomplete dictionary constructed from the reference image to gather possible match candidates. Belief propagation is then applied to eliminate bad candidates and to select optimum matches. The experimental result illustrates that our proposed algorithm performs favorably compared to the high accuracy optical flow method [8], [10].

References

[1] J. Schnabel, D. Rueckert, et al., "A generic framework for non-rigid registration based on non-uniform multi-level free-form deformations," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2001*. Springer, 2010, pp. 573–581.

[2] P.F. McLauchlan and A. Jaenicke, "Image mosaicing using sequential bundle adjustment," *Image and Vision Computing*, vol. 20, no. 9-10, pp. 751–759, 2002.

[3] S. Cheng, V. Stankovic, and L. Stankovic, "Improved sift-based image registration using belief propagation," *Conf. Acoustics, Speech and Signal Proc., Taiwan*, pp. 2909–2912, 2009.

[4] B. Glocker, N. Komodakis, G. Tziritas, N. Navab, and N. Paragios, "Dense image registration through MRFs and efficient linear programming," *Medical Image Analysis*, vol. 12, no. 6, pp. 731–741, 2008.

[5] B.D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *International joint conference on artificial intelligence*. Citeseer, 1981, vol. 3, pp. 674–679.

[6] T. Pock, M. Urschler, C. Zach, R. Beichel, and H. Bischof, "A duality based algorithm for TV-L 1-optical-flow image registration," *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2007*, pp. 511–518, 2007.

[7] D. Yang, H. Li, D.A. Low, J.O. Deasy, and I.E. Naqa, "A fast inverse consistent deformable image registration method based on symmetric optical flow computation," *Physics in Medicine and Biology*, vol. 53, pp. 6143, 2008.

[8] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," *European Conf. on Computer Vision, Prague*, vol. 3024 of LNCS, pp. 25–36, 2004.

[9] F.R. Kschischang, B.J. Frey, and H.A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on information theory*, vol. 47, no. 2, pp. 498–519, 2001.

[10] C. Liu, "Beyond pixels: exploring new representations and applications for motion analysis," *Doctoral Thesis. Massachusetts Institute of Technology*, 2009.

[11] R.G. Baraniuk, "Compressive sensing," *IEEE Signal Processing Magazine*, vol. 24, no. 4, pp. 118, 2007.

[12] A. Yang, A. Ganesh, Z. Zhou, S. Sastry, and Y. Ma, "Fast l1-minimization algorithms and an application in robust face recognition: a review," in *Proceedings of the International Conference on Image Processing*, 2010.

[13] R. Maleh, AC Gilbert, and MJ Strauss, "Sparse gradient image reconstruction done faster," in *IEEE International Conference on Image Processing, ICIP*, 2007, vol. 2, pp. 77–80.