IMPROVED SIFT-BASED IMAGE REGISTRATION USING BELIEF PROPAGATION

Samuel Cheng *

University of Oklahoma
Dept. Electrical and Computer Engineering
Tulsa, OK 74135-2512

Vladimir Stanković and Lina Stanković

University of Strathclyde
Dept. Electronics and Electrical Engineering
Glasgow G1 1XW UK

ABSTRACT

Scale Invariant Feature Transform (SIFT) is a very powerful technique for image registration. While SIFT descriptors accurately extract invariant image characteristics around keypoints, the commonly used matching approach for registration is overly simplified, because it completely ignores the geometric information among descriptors. In this paper, we formulate keypoint matching as a global optimization problem and provide a suboptimum solution using belief propagation. Experimental results show significant improvement over previous approaches.

Index Terms— Image registration; belief propagation; SIFT

1. INTRODUCTION

Image registration, an important component in image processing, is widely used in numerous applications involving multiple images. Applications range from computer vision, medical imaging, to super-resolution and hyper-spectral imaging. In these applications, image registration is generally the first image processing step used to integrate/align images for further analysis. It transforms a set of images that are acquired at different times or from different angles, into the same coordinate system.

Many image registration algorithms have been proposed in the literature; however, the most popular techniques are based on Scale Invariant Feature Transform (SIFT) [1] and its variants [2, 3]. SIFT extracts distinctive features in images, that are invariant to image scale and rotation. These features can be used for reliable matching of images, robustness to noise, change in illumination, and 3D camera viewpoint. SIFT detects locations, keypoints, in the image invariant to scaling and rotation and forms descriptors based on orientation, scale and location of the keypoint. While the descriptors can accurately extract invariant image characteristics around keypoints, the commonly used matching approach for registration is overly simplified in the sense that it completely ignores the geometric information among descriptors. The main contribution of this paper is to address the insufficiency of matching algorithms in SIFT and its variants. We introduce an intuitive formulation and implement it using belief propagation (BP). Initial results show significant improvement over conventional SIFT. Note that the proposed matching approach can be directly applied to other SIFT variants as well.

The rest of the paper is organized as follows. A brief review of SIFT is presented in the next section. In Section 3, we describe our improved matching approach. Experimental results are given in Section 4. We conclude and address future work in Section 5.

2. SIFT

This section provides a brief introduction to SIFT. More details can be found in [1].

The main objective of SIFT is to identify locations, keypoints, of an image where there exist characteristics that are invariant to scaling and rotation. These characteristics are summarized by a descriptor. In a nutshell, SIFT generates keypoints through finding the extremum of difference-of-Gaussian function of an image. A candidate keypoint will be refined to subpixel level and eliminated if found to be unstable. Once a keypoint is located, a descriptor or a feature vector is generated based on orientation(s), scale and location of the keypoint.

Suppose that we have two images, Image 1 and Image 2, that need to be integrated. Denote $D_1(i)$ and $D_2(j)$ the descriptors/feature vectors of the i^{th} and the j^{th} keypoints of Image 1 and Image 2, respectively. Let $\mathbf{x}_{D1}(i)$ and $\mathbf{x}_{D2}(j)$ be the corresponding locations of the keypoints in Images 1 and 2, respectively.

The matching problem consists of finding a keypoint in Image 2 that best corresponds to each keypoint in Image 1. As described in [1], the best candidate matching can be achieved based on minimum Euclidean distance. In other words, for a descriptor $D_1(i)$, the best match $\hat{j}$ should satisfy

$$\hat{j}(i) = \text{argmin}_j ||D_1(i) - D_2(j)||_2. \tag{1}$$

Since it is possible that there does not exist a correct matching keypoint in the second image, we need to impose a condition to determine if this happens. Lowe [1] suggested to use the

*The author wants to thank his student Shuang Wang for generating some of the figures.

second best match (second minimum distance from the target descriptor) to gauge the probability of a match. Therefore, $\hat{j}(i)$ in (1) will be considered as a valid match only if

$$|D_1(i) - D_2(\hat{j}(i))||_2 < T||D_1(i) - D_2(j)||_2, \forall j \neq \hat{j}(i), \quad (2)$$

where T is a constant independent of images.

3. IMPROVED MATCHING FOR SIFT

While SIFT is highly successful in generating unique descriptive features of images, the matching method proposed by the original SIFT algorithm is overly simplified. Indeed, its locally optimized approach, that minimizes Euclidean distance, completely ignores the geometric information among different descriptors. As a result, mapped descriptors in Image 2 of a nearby descriptor in Image 1 could be far away.

To incorporate geometric information, we consider matching as a global optimization problem and introduce a penalty for keypoints that violate geometric invariance. Specifically, our penalty function, $\Phi(\mathbf{m})$, is defined as the sum of the second norms of differences between the distance from one keypoint to another in Image 1, and the distance between the corresponding mapped keypoints in Image 2. That is,

$$\Phi(\mathbf{m}) = \sum_{i \in I_{D_1}} \sum_{j \in I_{D_1}} \phi(i, j; m(i), m(j)), \quad (3)$$

where

$$\phi(i, j; i', j') = \Big\| \sqrt{\|\mathbf{x}_{D_1}(i) - \mathbf{x}_{D_1}(j)\|_2} - \sqrt{\|\mathbf{x}_{D_2}(i') - \mathbf{x}_{D_2}(j')\|_2} \Big\|_2, \quad (4)$$

and I_{D_1} is the set of descriptors for the original image and $m(.)$ are the mapped descriptor indices in Image 2. Therefore, we aim to solve a globally optimized problem as follows

$$\hat{\mathbf{m}} = \arg\min_{\mathbf{m}} \Psi(\mathbf{m}) + \lambda \Phi(\mathbf{m}) \quad (5)$$

with

$$\Psi(\mathbf{m}) = \sum_{i \in I_{D_1}} \psi(i, m(i)) \quad (6)$$

and

$$\psi(i, i') = \|D_1(i) - D_2(i')\|. \quad (7)$$

3.1. Belief Propagation

Unfortunately, the optimization problem in (5) is not convex and appears to have exponential complexity. It is possible to discard some keypoints initially as described by (2), but the number of remaining keypoints will still be over a hundred typically. Thus, an exhaustive search will require $\sim 100!$ steps and apparently is computationally intractable. Moreover, applying (2) at such an early stage may unnecessarily discard useful information. Furthermore, (2) could simply be fundamentally faulty in some cases; that is, it is possible that a match exists that does not satisfy this condition.

Instead, we propose to use Belief propagation (BP) to solve (5). BP [4], also known as message passing algorithm among many other names, is widely used in numerous signal processing applications [5, 6, 7, 8, 9]. It can be considered as an iterative algorithm to approximate the global optimum of a discrete optimization problem.

To begin with, an optimization problem is divided into a number of simpler (local) problems. At each iteration step, instead of estimating the exact optimum solution, each local problem attempts to evaluate the probability (belief) of each possible solution being optimal. These local beliefs will be exchanged among "neighboring" problems, where neighborhood is defined based on the specific problem. These beliefs will be incorporated in computing the beliefs in the next iteration for each local problem. The algorithm stops either after a fixed predefined number of iterations, or when the most probable beliefs among all local problems converge.

3.2. Descriptor Matching using Belief Propagation

Since the optimization problem does not change even if we raise the objective function exponentially, we can rewrite (5) as

$$\hat{\mathbf{m}} = \arg\max_{\mathbf{m}} \exp(-\Psi(\mathbf{m})) \exp(-\lambda \Phi(\mathbf{m})), \quad (8)$$

which is equivalent to

$$\hat{\mathbf{m}} = \operatorname*{argmax}_{\mathbf{m}} \prod_{i \in I_{D_1}} b_{Des_i}(m_i) \prod_{i \in I_{D_1}, j \in I_{D_1}} b_{Dist_{i,j}}(m_i, m_j) \quad (9)$$

where

$$b_{Des_i}(m_i) = \exp(-\psi(i, m_i)/C_{Des}), \quad (10)$$

$$b_{Dist_{i,j}}(m_i, m_j) = \exp(-\phi(i, j; m_i, m_j)/C_{Dist}), \quad (11)$$

and $C_{Des}/C_{Dist} = \lambda$. $b_{Des_i}(m_i)$ and $b_{Dist_{i,j}}(m_i, m_j)$ can be considered approximately as beliefs of the i^{th} keypoint in Image 1 being matched to the m_i^{th} keypoint in Image 2 given the descriptor information and the information that the j^{th} keypoint is being matched to the m_j^{th} keypoint, respectively.

The reformulation in (9) does not make the optimization problem tractable. However, if we relax the problem a bit and have the optimization function only include $b_{Dist_{i,j}}$ in which i and j are close together, in the neighborhood (i.e., considering $j \in \mathcal{N}(i) \subset I_{D_1}$ with $|\mathcal{N}(i)| \ll |I_{D_1}|$) the problem can be solved approximately using a type of BP algorithm known as the max-product algorithm [10]. In other words, we want to solve

$$\hat{\mathbf{m}} = \operatorname*{argmax}_{\mathbf{m}} \prod_{i \in I_{D_1}} b_{Des_i}(m_i) \prod_{i \in I_{D_1}, j \in \mathcal{N}(i)} b_{Dist_{i,j}}(m_i, m_j) \quad (12)$$

instead.

The max-product algorithm converges to the global optimum if the network graph is a tree. In general, the algorithm is suboptimal but is shown to converge to a very good solution in many applications [5, 6, 7, 8]. Before presenting our algorithm, let us define $\mu_{i,j}^{(n)}(m_j)$ as the belief of keypoint i that the correct match for keypoint j is m_j at the n^{th} iteration. This message is passed from keypoint i to keypoint j in each iteration. The algorithm is summarized as follows:

1. Initialize all messages $\mu_{i,j}^{(0)}(m_j)$ as a constant. Set $n = 1$.
2. Update messages $\mu_{i,j}^{(n)}$ iteratively for $i \in I_{D_1}$

$$\mu_{i,j}^{(n)}(m_j) \leftarrow \kappa_1 \max_{m_i} b_{Dist_{i,j}}(m_i, m_j) b_{Des_i}(m_i) \prod_{j' \in \mathcal{N}(i) \backslash j} \mu_{j',i}^{(n-1)}(m_i). \quad (13)$$

3. Compute overall beliefs

$$b_i^{(n)}(m_i) \leftarrow \kappa_2 b_{Des_i}(m_i) \prod_{j' \in \mathcal{N}(i)} \mu_{j',i}^{(n)}(m_i), \quad (14)$$

$$\hat{m}_i = \arg\max_{m_i} b_i^{(n)}(m_i) \quad (15)$$

4. $n \leftarrow n + 1$ and goto 2 until n reaches the maximum number of iterations.

κ_1 and κ_2 above are normalization constants. It is desirable to vary them such that $\sum_{m_j} \mu_{i,j}(m_j)$ and $b_i(m_i)$ are normalized to 1. Then, $b_i(m_i)$ can be physically interpreted as probability of keypoint i matching to m_i. Moreover, it becomes reasonable to discard a keypoint only when $b_i(\hat{m}_i)$ is less than some probability threshold p_{th}.

3.3. Least-Square Registration

After the matched keypoints are determined, the two images can be registered by simply computing the best projective transform to satisfy the matching points [1]. Given a set of keypoint locations (x_i, y_i) in Image 1 and the locations of corresponding mapped keypoints (x_i', y_i') in Image 2, we need to find an optimum 3×3 matrix A such that

$$A \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} x_i' \\ y_i' \\ 1 \end{bmatrix}. \quad (16)$$

The equation above can be rearranged to gather the unknowns into a column vector. Thus, we have

$$\underbrace{\begin{bmatrix} X_i & Y_i & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_i & y_i & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x_i & y_i & 1 \\ & & & \cdots & & & & & \\ & & & \cdots & & & & & \end{bmatrix}}_{B} \cdot \underbrace{\begin{bmatrix} A_{1,1} \\ A_{1,2} \\ A_{1,3} \\ A_{2,1} \\ A_{2,2} \\ A_{2,3} \\ A_{3,1} \\ A_{3,2} \\ A_{3,3} \end{bmatrix}}_{w} = \underbrace{\begin{bmatrix} x_j' \\ y_i' \\ 1 \\ \vdots \end{bmatrix}}_{c}. \quad (17)$$

The least-squares solution for A can be computed by solving the normal equation

$$w = [B^T B]^{-1} B^T c, \quad (18)$$

which can be solved efficiently and in a numerically stable manner via QR factorization.

4. EXPERIMENTS

Our SIFT implementation is based on the open source work from Vedaldi [11]. We applied our algorithm to the INRIA Graffiti dataset [12]. Fig. 2 shows example matches for original images, shown in Fig. 1, using SIFT and the proposed algorithm. There are a couple of errors shown in the original SIFT matching. In particular, the keypoints from the original image are all fairly close together. However, SIFT essentially ignores this information and matches one of the keypoints to a keypoint far away from the others. The proposed BP algorithm, however, attempts to keep distance invariance as shown in the bottom figure of Fig. 2.

The registration results using least-square registration as described in Section 3.3 for Fig. 1 are shown in Fig. 3. Figs. 4 and 5 demonstrate two more registration examples. Significant improvement is observed in the proposed algorithm result compared to the original matching approach [1].

Throughout our experiments, we have set $C_{Des} = 50,000$ and $C_{Dist} = 700$. The number of neighbors for all input keypoints, $|\mathcal{N}_i|$, is set to 5. To speed up BP, we restrict the possible matches to only ten descriptors with the highest initial (prior) matches. The threshold probability described in Section 3.2, p_{th}, is set to 0.7. The total number of BP iterations for each case is 10. The computation time of the BP algorithm varies significantly with the number of original descriptors found by the SIFT algorithm. For typical number of descriptors of around a thousand, the BP algorithm spends roughly one minute for a Pentium 4 PC. However, our current code is not yet optimized and is implemented as a MATLAB/C-MEX hybrid.

Fig. 1. Original images.

5. CONCLUSION AND FUTURE WORK

In this paper, we propose using the BP algorithm to improve traditional SIFT-based image registration. Initial results show significant improvement over the traditional approach. Our

Fig. 2. Five example matches by SIFT (top) and by the proposed algorithm (bottom). The distance constraint apparently helps to fix some potential matching errors.

Fig. 3. Registration result by SIFT (top) and by the proposed algorithm (bottom).

BP-based approach can be applied to other SIFT variants such as PCA-SIFT [2], which will be part of future work.

References

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

[2] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," in *Proc. CVPR 2004*, vol. 2. IEEE Computer Society; 1999, 2004.

[3] M. Grabner, H. Grabner, and H. Bischof, "Fast approximated sift," *Lecture Notes in Computer Science*, vol. 3851, p. 918, 2006.

[4] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. San Francisco: Morgan KauFmann Publishers, Inc., 1988.

Fig. 4. Registration result by SIFT (top) and by the proposed algorithm (bottom). . Left column shows Image 1.

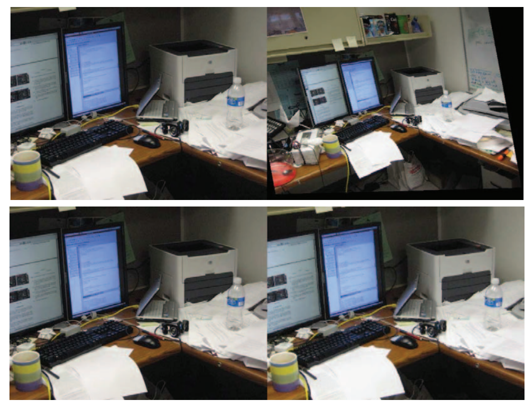

Fig. 5. Registration result by SIFT (top) and by the proposed algorithm (bottom). Left column shows Image 1.

[5] D. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, pp. 399–431, Mar. 1999.

[6] D. J. C. MacKay, R. J. McEliece, and J. F. Cheng, "Turbo decoding as an instance of pearl's belief propagation algorithm," *IEEE Journal on Selected Areas in Communication, page in press*, 1997.

[7] J. J. Hopfield and C. D. Brody, "What is a moment? transient synchrony as a collective mechanism for spatiotemporal integration," *Proceedings of the National Academy of Sciences*, p. 31567098, 2001.

[8] J. Sun, N. N. Zheng, and H. Y. Shum, "Stereo matching using belief propagation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 7, pp. 787–800, 2003.

[9] F. Kschischang, B. Frey, and H. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, pp. 498–519, Feb. 2001.

[10] D. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge, 2003.

[11] A. Vedaldi, "An open implementation of the SIFT detector and descriptor," UCLA CSD, Tech. Rep. 070012, 2007.

[12] Viewpoint change sequences. [Online]. Available: http://www.inrialpes.fr/movi/