Dense Non-Rigid Point-Matching Using Random Projections

Raffay Hamid, Dennis DeCoste
eBay Research Labs.
San Jose, CA., USA
{rhamid, ddecoste}@ebay.com

Chih-Jen Lin
National Taiwan University,
Taipei 10617, Taiwan
cjlin@csie.ntu.edu.tw

Abstract

We present a robust and efficient technique for matching dense sets of points undergoing non-rigid spatial transformations. Our main intuition is that the subset of points that can be matched with high confidence should be used to guide the matching procedure for the rest. We propose a novel algorithm that incorporates these high-confidence matches as a spatial prior to learn a discriminative subspace that simultaneously encodes both the feature similarity as well as their spatial arrangement. Conventional subspace learning usually requires spectral decomposition of the pair-wise distance matrix across the point-sets, which can become inefficient even for moderately sized problems. To this end, we propose the use of random projections for approximate subspace learning, which can provide significant time improvements at the cost of minimal precision loss. This efficiency gain allows us to iteratively find and remove high-confidence matches from the point sets, resulting in high recall. To show the effectiveness of our approach, we present a systematic set of experiments and results for the problem of dense non-rigid image-feature matching.

1. Introduction

Matching interest-points across images has been a long-standing problem in Computer Vision [23] [30]. This problem is particularly challenging as point-sets become more dense, and their spatial transformations become more non-rigid. Perturbations due to sensor noise also play a significant role to further exacerbate the problem.

Some of these challenges can be addressed by trying to maintain the spatial arrangements of corresponding points during matching. Most of the previous approaches that bring the spatial arrangement of points into account are computationally expensive, and are therefore not feasible for dense matching [2] [28] [16] [4]. Recently, Torki and Elgammal [26] proposed an efficient method for matching points in a lower-dimensional subspace, that simultaneously encodes spatial consistency and feature similarity [26] [27]. However, this method still requires exact spectral decomposition for subspace learning, which limits its efficiency improvements. In addition, the method is not robust to large amounts of non-rigid distortion or feature noise.

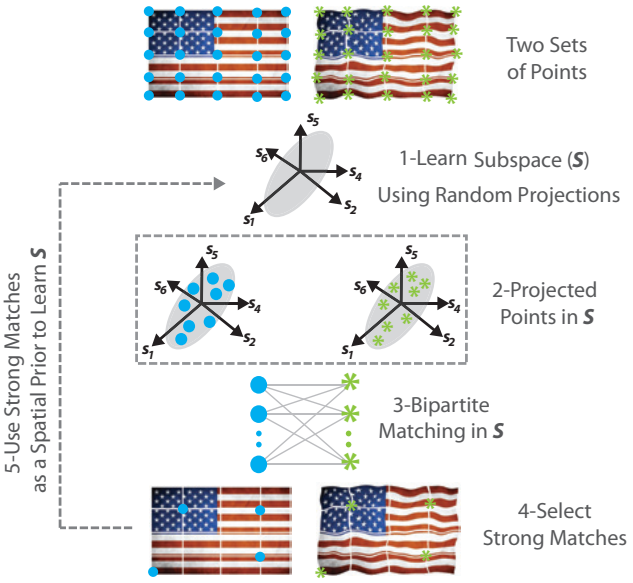

Figure 1: Given two sets of feature-points, we learn a subspace S that maintains their across-set feature similarity and within-set spatial arrangement. We use random projections to approximately learn S efficiently. We project feature-points to S, and use bi-partite graph to find their matchings. We select points with high confidence matches, and use them as a spatial prior to to learn a subspace that reduces the confusion among the remaining set of points. This process is repeated until no more points can be matched with high confidence.

In this paper, we propose a framework that improves upon the existing methods to address the issues of robustness and efficiency. Our approach has two key elements:

- *Iterative matching with spatial priors:* We propose to use the subset of high confidence matches as spatial priors, to learn a subspace that reduces the confusion among the remaining set of points. We repeat this process until no more points can be matched with high confidence. This approach provides higher robustness

in the face of noise and distortion.

- *Approximate subspace learning with random projections:* Instead of using exact spectral decomposition, we propose the use of random projections [31] for approximate subspace learning. This significantly reduces the computational complexity of subspace learning at the cost of minimal precision loss, and makes it feasible to tackle matching problems that are prohibitively expensive for existing approaches.

To show the competence of our framework, we present a comparative analysis of how different approaches perform for varying levels of feature noise and non-rigid distortion. We demonstrate that our approach outperforms alternate methods both in the face of noise and distortion, without incurring any additional costs in time complexity. The overview of our approach is illustrated in Figure 1.

We start by formalizing an approach for point matching using subspace learning in Section 2, followed by the details of our approach of iterative subspace learning using spatial priors, presented in Section 3. In Section 4 we explain how to efficiently learn approximate subspaces using random projections as opposed to exact spectral decomposition. We present our experiments and results in Section 5, and conclude our paper in Section 7.

2. Point Matching Using Subspace Learning

For dense point matching problems, it is important that not only the feature similarity of matched points is maximized, but also that their spatial arrangements are maintained. To this end, subspace learning approaches try to find a lower dimensional manifold that maintains the across-set feature similarity as well as the within-set spatial arrangement of points. The intuition here is that a matching problem based on the similarities in the learned subspace, will implicitly also take into account their spatial arrangements. The matching problem can then be expressed as a classic bipartite problem in the learned subspace, which can be solved using various methods [10] [23]. We now formally define the subspace learning problem that forms the basis of our approach described in the following sections.

2.1. Preliminaries

We follow Torki and Elgammal's formulation of the subspace learning problem [26]. Consider two[1] sets of feature points X^1 and X^2 from two images, where $X^k = \{(x_1^k, f_1^k), \cdots, (x_{N_k}^k, f_{N_k}^k)\}$ for $k \in \{1, 2\}$, and N_k denotes the number of points in the k^{th} point-set, while N denotes the total number of points in both point-sets. Each point (x_i^k, f_i^k) is defined by its spatial location in its image plane $x_i^k \in \mathbb{R}^2$, and its feature descriptor $f_i^k \in \mathbb{R}^D$, where D is the dimensionality of the descriptor.

The spatial arrangement of points X^k is encoded in a *spatial affinity* matrix denoted by $S_{i,j}^k = K_s(x_i^k, x_j^k)$. Here, $K_s(\cdot, \cdot)$ is a spatial kernel that measures the spatial proximity of points i and j in set k. Similarly, the feature similarity of point pairs across X^1 and X^2 is encoded in a *feature affinity* matrix $U_{i,j}^{p,q} = K_f(f_i^1, f_j^2)$, where $K_f(\cdot, \cdot)$ is an affinity kernel that measures the similarity of feature i in set p to feature j in set q. Note that K_s and K_f are within and across set operators respectively. A common choice for the spatial and feature kernels is $K_s(x_i^k, x_j^k) = e^{-||x_i^k - x_j^k||^2/2\sigma_s^2}$ and $K_f(f_i^1, f_j^2) = e^{-||f_i^1 - f_j^2||^2/2\sigma_u^2}$ respectively. The *bandwidth* parameters σ_s and σ_u control the importance given to spatial consistency and feature similarity respectively [26].

2.2. Subspace Learning

Let $Y^k = \{y_1^k, \cdots, y_{N_k}^k\}$ be the set of points corresponding to X^k, projected into the desired subspace. Here $y_i^k \in \mathbb{R}^d$ denotes the projected coordinates of point x_i^k, and d is the subspace dimensionality. Subspace learning can be expressed as minimizing the following objective [26]:

$$\phi(Y) = \sum_k \sum_{i,j} ||y_i^k - y_j^k|| S_{i,j}^k + \sum_{p,q} \sum_{i,j} ||y_i^p - y_j^q|| U_{i,j}^{p,q} \quad (1)$$

Here k, p, and $q \in \{1, 2\}$, and $p \neq q$. Intuitively, the first term of Equation 1 tries to keep the subspace coordinates y_i^k and y_j^k of any two points x_i^k and x_j^k close to each other based on their spatial kernel weight $S_{i,j}^k$. Also, the second term tries to minimize the distance between points y_i^p and y_j^q if the value for their feature similarity kernel $U_{i,j}^{p,q}$ is high.

Equation 1 can be re-written using one set of weights defined on the entire set of input points as

$$\phi(Y) = \sum_{p,q} \sum_{i,j} ||y_i^p - y_j^q|| A_{i,j}^{p,q} \quad (2)$$

where the matrix A is defined as:

$$A_{ij}^{p,q} = \begin{cases} S_{i,j}^k & \text{if} \quad p = q = k \\ U_{i,j}^{p,q} & \text{otherwise} \end{cases} \quad (3)$$

Here $A^{p,q}$ is the (p, q) block of A. The matrix A is an $N \times N$ weight matrix with $K \times K$ blocks, such that the (p, q) block is of size $N_p \times N_q$. The k^{th} diagonal block of A is the spatial structure kernel S^k for the k^{th} point-set. The off-diagonal (p, q) block is the descriptor similarity kernels $U^{p,q}$. The matrix A is symmetric by definition, since the diagonal blocks are symmetric, and $U^{p,q} = U^{q,p^T}$.

Equation 2 is equivalent to the *Laplacian embedding* problem for the point-set defined by the matrix A [1]. This problem is often expressed as:

$$Y^* = \arg \min_{Y^T D Y = I} \text{tr}(Y^T L Y) \quad (4)$$

[1] This formulation is extendable to multi-set problems as well [26].

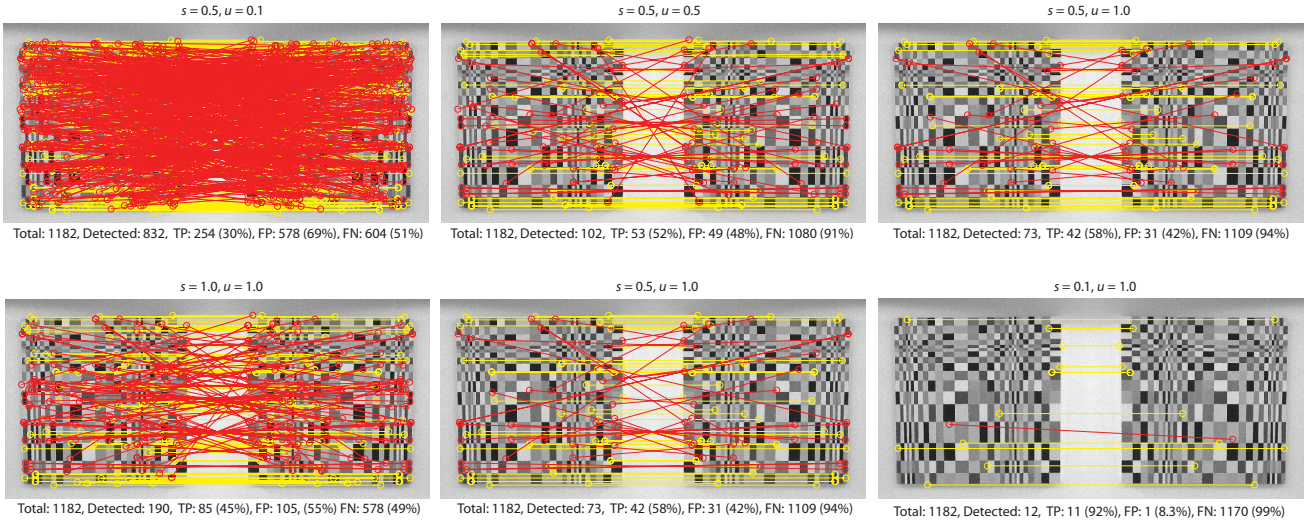

Figure 2: The top row shows the variation of the feature kernel (σ_u) from 0.1 to 1.0, with the spatial kernel (σ_s) fixed at 0.5. The bottom row shows the variation of the σ_s from 1.0 to 0.1, while σ_u being fixed at 1.0. Red and yellow lines show wrong and correct matches.

where L is the Laplacian of the matrix A defined as $L = D - A$. D is the diagonal matrix defined as $D_{ii} = \sum_j A_{i,j}$. The $N \times d$ matrix Y is the stacking of the desired subspace coordinates, such that:

$$Y = [y_1^1, \cdots, y_{N_1}^1, y_1^2, \cdots, y_{N_2}^2, y_1^K, \cdots, y_{N_K}^K] \tag{5}$$

Equation 4 is a generalized Eigenvectors problem [1]:

$$Ly = \lambda Dy \tag{6}$$

The optimal solution of Equation 4 can therefore be obtained by the smallest d generalized Eigenvectors of L. The required N subspace-embedded points Y are stacked in d vectors such that the embedding of the first point-set is the first N_1 rows followed by N_2 rows for the second point-set.

2.3. Bipartite Matching in the Learned Subspace

Once the lower-dimensional subspace has been learned, we can compute the affinity matrix G among projected point-sets Y^1 and Y^2. Following the Scott-Higgins correspondence algorithm [23], the matching problem can be expressed as finding a permutation matrix C that rearranges the rows of G to maximize its trace. When the exact permutation constraint on C is relaxed to an orthonormal matrix constraint, the permuted version of matrix G maximized trace can be computed as:

$$G^* = UIV^T \tag{7}$$

where the SVD decomposition of G is $U\Sigma V^T$. A match between a point i in set one and j in set two is found, if the entry $G^*_{i,j}$ is the maximum in the i^{th} row and j^{th} column of G^*. The value of $G^*_{i,j}$ can be interpreted as the confidence of the matching between points i and j. The overall subspace learning algorithm is summarized in Algorithm 1.

Algorithm 1 MATCHING WITH SUBSPACE LEARNING

Input: X^1, X^2, σ_s, σ_u
Output: Mappings $\mathcal{M} : X^1 \leftrightarrow X^2$

1: Compute matrices S^k, $U^{p,q}$, A, D, and L
2: Find Y using the last d generalized Eigenvectors of L
3: Compute G from pair-wise distances of Y^1 and Y^2
4: Decompose $G = U\Sigma V^T$
5: Compute $G^* = UIV^T$
6: Search rows and columns of G^* to find $\mathcal{M} : X^1 \leftrightarrow X^2$

3. Subspace Learning with Spatial Priors

The space complexity of Algorithm 1 is $O(N^2)$, which is significantly less than the $O(N^4)$ complexity for most of the previous point matching approaches that also incorporate spatial constraints [2] [28] [16] [4]. To motivate the usage of spatial priors in subspace learning, we analyze how the gain in space complexity offered by Algorithm 1 affects its matching-performance, specially when there is significant confusion between feature points. One way to do this is to analyze the precision-versus-recall tradeoff of Algorithm 1 over the entire range of the kernel bandwidths σ_u and σ_s.

3.1. Analyzing The Role of Kernel Bandwidths

We consider an image with many repetitive patterns, and flip it along its y-axis to create the second image (Figure 2). This ensures that there are many local minima in the matching space of the points from the two images. We vary either σ_s or σ_u, while keeping the threshold for selecting confident matches at a fixed value. The results are shown in Figure 2.

As shown, while varying either σ_s or σ_u, any increase in precision comes at a significant cost of a decrease in re-

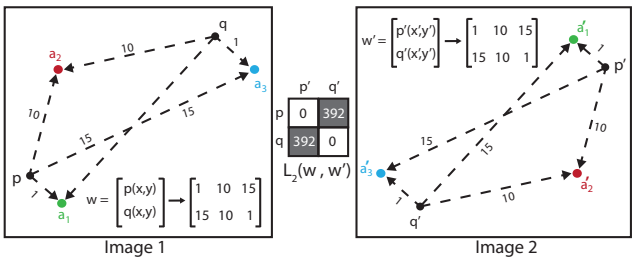

Figure 3: The figure illustrates the projection of points from their respective $2-D$ image coordinate spaces to a shared $m = 3$ dimensional space, where each dimension corresponds to how close a point is to one of the $m = 3$ anchor-points.

call. Notice however that considering precision alone, this approach can successfully find a subset of highly confident matches, which are very likely to be correct.

Based on these observations, we propose to use the maximally confident matches in order to limit the search-space of the points that initially do not have any sufficiently confident correspondence. We use this spatial prior to iteratively learn a more discriminative subspace which enables us to find and remove strong matches, until all the sufficiently strong matches are found.

3.2. Incorporating Spatial Priors

The main intuition behind how we incorporate spatial priors in subspace learning is as follows. If a point a in set 1 is confidently matched to a point b in set 2, then the neighboring points of b should be given more importance while searching for a match of any point near a. We now formalize this intuition more concretely.

Suppose we know the ground truth mappings between a subset of points $P \in X^1$ each with a mapping to another set of points $Q \in X^2$. Note that $|P| = |Q| = m$. We now compute the pairwise L_2 distances of all points in P with $\{X^1 \setminus P\}$, and points in Q with $\{X^2 \setminus Q\}$. This operation can be interpreted as a projection of points from their respective $2-D$ image coordinate spaces to a shared m dimensional space, where each dimension corresponds to how close a point is to one of the m *anchor-points* (see Figure 3).

With both point-sets $\{X^1 \setminus P\}$ and $\{X^2 \setminus Q\}$ in a shared projected space, we compute their pair-wise L_2 distances to construct an $(N_1 - m) \times (N_2 - m)$ matrix $\mathcal{D}$. To convert $\mathcal{D}$ from a distance to a similarity matrix, we apply a decaying exponential kernel $e^{-||\mathcal{D}(i,j)||/2\sigma_r^2}$, to construct a similarity matrix $\mathcal{H}$. The $(i,j)^{th}$ value in $\mathcal{H}$ encodes how much the i^{th} point in set 1 and j^{th} point in set 2 are in the vicinity of each other. We can therefore use the matrix $\mathcal{H}$ as a multiplicative kernel to boost up or down the values of feature similarity matrix $U^{p,q}$ accordingly. This results in a more strict encoding of the spatial constraints in our mapping problem, while maintaining the space complexity of $O(N^2)$.

While our approach is well-suited for interactive applications where a user could provide us with P and Q, in general we initialize P and Q as empty sets. At the start of the first iteration, all values in $\mathcal{H}$ are set equal to 1. At the end of first iteration, we select the top k most confident matches $\mathcal{M}_c$, and add them to P and Q. This procedure is repeated over multiple iterations, and is listed in Algorithm 2.

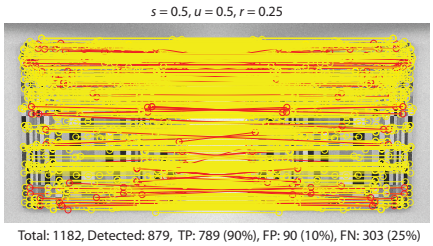

Figure 4: The matching result for the problem given in Figure 2 using iterative spatial priors (Algorithm 2).

The matching result for the problem given in Figure 2 using our iterative spatial priors scheme (Algorithm 2) is presented in Figure 4. As shown, we are able to achieve a significantly higher precision and recall rates compared to Algorithm 1 for any value of σ_s or σ_u.

4. Random Projections for Subspace Learning

We now turn our attention to the efficiency aspects of our iterative subspace learning approach for the dense point matching problem. Conventional subspace learning usually requires spectral decomposition of the point-set's pairwise distance matrix, which for larger matrices can be computationally expensive[2]. Repeating this procedure multiple times in Algorithm 2 while using an exact matrix decomposition approach would make it infeasible.

To this end, we propose to use the method of random projections [31] as an approximate way to perform spectral partitioning of matrix L. Approximate partitioning methods are well-suited for our framework since in each iteration we are interested only in high confidence matches, which are

[2]For a $1,000 \times 1,000$ problem, close to 85% of the time taken by Algorithm 1 is for subspace learning

Algorithm 2 SUBSPACES WITH SPATIAL PRIORS

Input: X^1, X^2, $P = Q = \phi$, σ_s, σ_u, σ_r, $\mathcal{M} = \phi$
Output: Mappings $\mathcal{M} : X^1 \leftrightarrow X^2$

1: Compute matrices $\mathcal{H}$, S^k, and $U^{p,q}$
2: $U^{p,q} = U^{p,q} \cdot \mathcal{H}$
3: Find $\mathcal{M}_c : X^1_{m_c} \leftrightarrow X^2_{m_c}$ using Algorithm 1
4: If $|X^1_{m_c}| = |X^2_{m_c}| = \phi$, **return**
5: $P = P \cup X^1_{m_c}$, and $Q = Q \cup X^2_{m_c}$
6: $X^1 = X^1 \setminus X^1_{m_c}$, and $X^2 = X^2 \setminus X^2_{m_c}$
7: $\mathcal{M} = \mathcal{M} \cup \mathcal{M}_c$
8: Go to step 1.

least impacted by the errors introduced by using an approximate subspace instead of an exact one [14].

4.1. From Generalized to Regular Eigenvectors

To use random projections for subspace learning, we first need to convert the generalized Eigenvector problem of Equation 6, to a regular Eigenvector problem. This can be done in the following two steps.

Step 1 - Consider the following Eigenvector problem:

$$Ax = \lambda_2 Dx \tag{8}$$

i.e.,

$$(D - L)x = \lambda_2 Dx \tag{9}$$

where λ_2 denotes the largest k generalized Eigenvectors that satisfy the constraint of Equation 9.
Solving Equation 9 leads to the following equation:

$$Lx = (1 - \lambda_2)Dx \tag{10}$$

Comparing Equations 6 and 10 gives:

$$\lambda_1 = 1 - \lambda_2 \tag{11}$$

implying that λ_1 are the smallest k Eigenvectors of L.

Step 2 - Equation 9 can be re-written as:

$$D^{-1/2}AD^{-1/2}D^{1/2}x = \lambda_2 D^{1/2}x \tag{12}$$

Denoting

$$D^{1/2}x = y \tag{13}$$

and

$$B = D^{-1/2}AD^{-1/2} \tag{14}$$

Equation 12 becomes

$$By = \lambda_2 y \tag{15}$$

Using Equation 13, we can find the largest k generalized Eigenvectors of A from the largest k regular Eigenvectors of B. In step 1 we have already shown that the largest k generalized Eigenvectors of A correspond to the smallest k generalized Eigenvectors of L. Combining step 1 and 2 lets us find the smallest k generalized Eigenvectors of L, by finding the largest k regular Eigenvectors of B.

4.2. Approximate Subspace Learning

Having converted our generalized Eigenvector problem in L and D, into a regular Eigenvector one in B, we now explain how to find the top k approximate Eigenvectors of B using random projections [14].

Given a matrix B, a target rank k, and an oversampling parameter p, we seek to construct a matrix Q such that:

$$||B - QQ'B|| \approx \min_{\text{rank}(Z)\leq k} ||B - QZ|| \tag{16}$$

Algorithm 3 FAST SVD USING RANDOM PROJECTIONS

Input: An $n \times n$ matrix B (here $n = N_1 + N_2$)
Output: Approximate rank-k SVD of B

1: Draw an $n \times k$ matrix $\Omega \sim N(0, 1)$
2: Form the $n \times k$ sample matrix $Y = B\Omega$
3: Form the $n \times k$ orthonormal matrix Q *s.t.*, $Y = QR$
4: Form the $k \times n$ matrix $Z = Q'B$
5: Find SVD of $Z : Z = \hat{U}\Sigma V'$
6: Form the matrix $U = Q\hat{U}$

where $||.||$ represents the L_2 norm operator, and $Z = Q'B$. Given such a matrix Q, we seek to find an approximate decomposition of B such that $B \approx U\Sigma V^T$, where U and V are the Eigenvectors for the row and column spaces of B respectively, while Σ are the corresponding Eigenvalues.

Recall that the standard way to decompose a rank-deficient matrix can be divided into two steps:

- **Step 1** - Use Grahm-Schmidt [13] (or an equivalent) transform to find Q, which is in fact a low-rank orthonormal bases for the range (column space) of B.
- **Step 2** - Matrix B is then projected to this low-dimensional space to form the (short and fat) matrix Z. Finally, Z is spectrally decomposed using SVD to find the low-rank U and V matrices for B.

The main computational bottleneck for such a scheme is computing Grahm-Schmidt to find Q. This is because Grahm-Schmidt requires scanning B iteratively k times, which can be computationally expensive. Following the work in [14], we now show how to use randomly generated vectors to avoid Grahm-Schmidt for finding Q.

The fundamental intuition here is that in higher dimensional spaces, randomly generated vectors are very likely to be linearly independent. One could therefore generate a linearly independent subspace Y, of rank k that spans the range of matrix B, by simply stacking k randomly generated vectors Ω, and multiplying them by B. This allows one to generate a linearly independent subspace in a single sweep of B, while fully exploiting the multi-core processing power of modern machines using BLAS 3. To produce Q, we just need to orthonormalize Y which is a much less expensive procedure than orthonormalizing B. We can now project B onto Q to generate Z, and compute its SVD to find the low-rank U and V matrices for B. The overall scheme to find the top k approximate Eigenvectors of B using random projections is listed in Algorithm 3.

5. Experiments & Results

The focus of this work is on the problem of dense non-rigid feature matching. While there are public data sets available for image feature matching problems, they either tackle

dense but affine transformations [20], or sparse but non-rigid transformations [9]. To the best of our knowledge, there are no public data-sets with ground truth available for our problem at hand. We therefore decided to simulate non-rigid transformations on our test images to have the ground-truth feature mappings, and systematically study the performance of different algorithms for dense non-rigid matching.

5.1. Simulating Non-Rigid Transformations

Given an image, we define a grid of points over it. We add random amounts of perturbations to the grid-points. The average perturbation added to the points determines how much non-rigid deformation we introduce to our input image (see Figure 5). We then use the b-spline framework proposed in [22] to morph the input image to the deformed output. We also control how much rotation and feature noise we add to make the matching task more or less difficult.

5.2. Noise Analysis

For the left image shown in Figure 5, we added 30 degrees of rotation, and added distortion of 20% of the image width. We used SIFT features [18] in this work, and perturbed their values from 5 to 15 times the norm of the average feature values. We generated 10 trials of this data, and ran different algorithms on this data. We considered 1000 points in each point-set, and used 500 dimensional subspaces.

Besides our random projection (RP) based framework (Algorithm 3), we ran the framework proposed by Torki and Elgammal [26] (TR), iterative runs of Torki and Elgammal [26] without any notion of spatial priors (N-ISP), our algorithm proposed in Algorithm 2 that incorporates spatial priors over multiple runs (ISP), and the greedy matching algorithm proposed in [19] (GR). The precision and recall curves for this set of experiments are shown in Figure 6. The average time taken, precision and recall rates for these algorithms for a fixed noise level (14) are given in Table 1.

Figure 6 shows that Torki and Elgammal [26] does very well on the precision, however it degrades very quickly as the feature noise increases. While the greedy algorithm takes less then a second to complete, its precision rate degrades quite steeply with noise. The N-ISP method takes the longest to complete, while giving poor precision performance. The best performance in terms of both precision and recall is achieved by ISP, however it takes more than twice as much time as Torki and Elgammal [26] does. The best method is our proposed Random Projections based one (RP) which takes less time than what Torki and Elgammal [26] does, and approaches our ISP algorithm in matching performance, beating all the other competitors.

5.3. Non-Rigid Distortion Analysis

To analyze the performance of the considered algorithms for different amounts of non-rigid transformations, we generated images with average non-rigid perturbation varying from 20% to 60% of the image width. Image rotation for this experiment was kept at 0° to study the effect of non-rigid transformation in isolation from rotation. The amount of feature noise for this experiment was set at 15 times the norm of average feature values. The precision and recall curves for this experiments are shown in Figure 7.

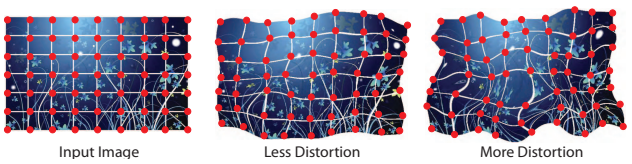

Figure 5: Given an image, we define a grid of points over it. We can control the amount of added distortion by varying the random amounts of perturbations added to each of these grid-points.

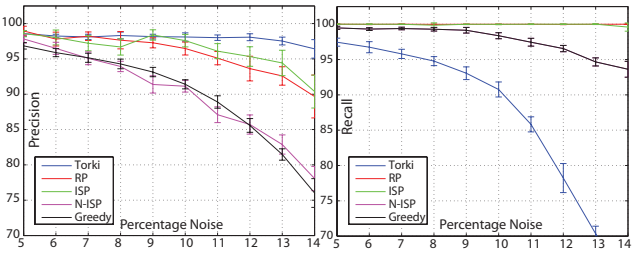

Figure 6: Noise Analysis – average precision and recall curves for 10 trials of varying amounts of feature noise.

The performance trends for both greedy and N-ISP remain similar to what is observed for the noise analysis, and they remain at the bottom of the lot. Torki and Elgammal [26] report high precision performance, however its recall gradually degrades with increasing amounts of non-rigid transformation. The ISP and RP methods give very close performance both in terms of precision as well as recall, and rank the best in the lot. Our method of RP achieves this result in time that is slightly lower than that taken by Torki and Elgammal [26]. We did similar experiments for varying amounts of rotation, and obtained similar performance trends.

5.4. Multiple Test Cases

To test the generalizability of our framework, we tried it on different images of objects which can naturally undergo non-rigid transformations (*e.g.*, garments, carpets, *etc*). The comparative results for these experiments are given in Figure 8. The behavior of the considered algorithms remains similar, with ISP and RP performing the best, while RP having a significant speed advantage.

6. Related Work

Feature matching is a well-explored problem, where points undergoing perspective [29] [15] as well as non-rigid geometric transformations have been studied [17]. The non-

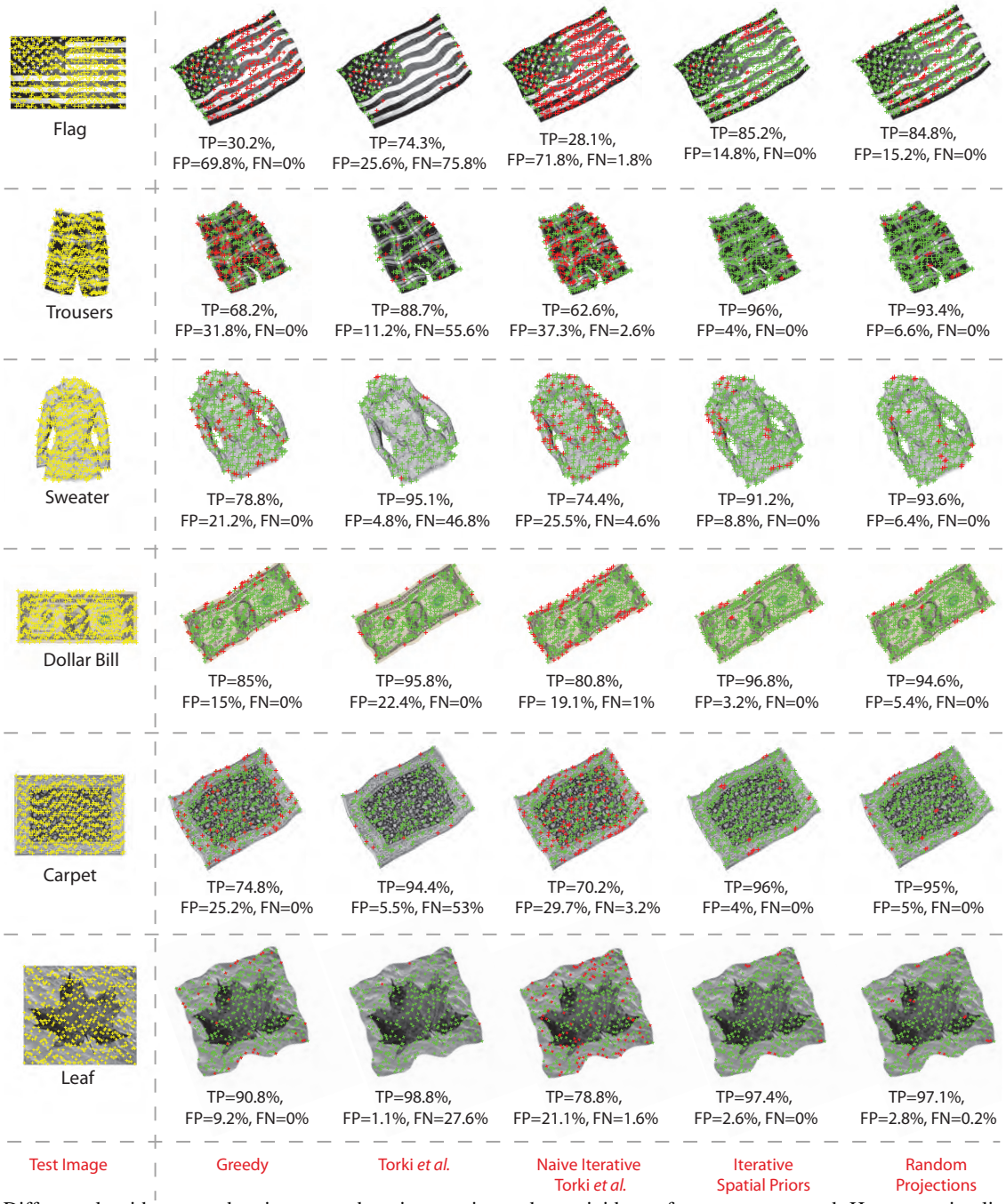

Figure 8: Different algorithms tested on images undergoing rotation and non-rigid transforms are compared. Here green implies correctly matched points, while red implies incorrect matches. TP, FP, and FN represent true positives, false positives and false negatives respectively.

rigid problems in particular have been looked at from a discrete [8] [21] as well as continuous optimization based perspectives [2] [23] [5]. In graph theoretic approaches, feature matching is framed as a graph isomorphism problem between two weighted or unweighted graphs in order to enforce edge compatibility [24] [30]. Several approaches use higher order spatial consistency constraints [7], however such constraints are not necessarily always helpful [2], and usually even linear constraints can be sufficient [26].

Graph matching algorithms usually apply spectral decomposition (*e.g.* SVD [12] [13]) to find manifold subspaces that minimize distances between corresponding points [2] [27]. Conventionally, work in manifold learning has focused on finding exact subspaces which can be computationally quite costing [6]. More recently however, there has been a growing interest in finding approximate

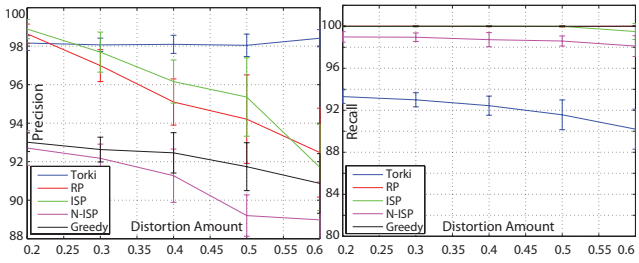

Figure 7: Distortion Analysis – average precision and recall curves for 10 trials of varying amounts of non-rigid image distortion.

	TR	RP	ISP	N-ISP	GR
T	65±5.2	62±4.3	112±4.8	182±22.9	0.85±0.08
P	96.4±1.3	89.6±3.0	90.3±2.3	78.0±1.7	76.0±2.0
R	63.6±1.1	100±0.0	99.5±0.5	93.6±1.1	100±0

Table 1: Average execution times (T), precision (P) and recall(R) of different algorithms for ten runs of tests at a noise level of 14 times the norm of average feature values. (TR) shows high precision but low recall rates. The best performance is given by ISP (Alg. 2), and RP (Alg. 3). From a computation time perspective, RP comfortably beats ISP, and is even faster than Torki.

subspaces in return for significant efficiency gains only for minimal precision loss [3] [31]. This interest has also been shared by some areas of Computer Vision [11] [25]. This work shows how approximate subspace learning techniques could be useful for the feature-matching problem.

7. Conclusions and Future Work

We presented a novel method for matching dense point-sets undergoing non-rigid transformation. Our approach iteratively incorporates high-confidence matches as a spatial prior to learn discriminative subspaces that encode both the feature similarity and their spatial arrangement. We proposed the use of random projections for approximate subspace learning that provides significant time improvements at the cost of minimal precision loss. To show the effectiveness of our approach, we presented a systematic set of comparative experiments. Currently, for each iteration of our algorithm we find a low-dimensional subspace independent of our previously computed subspaces. Going forward, we plan to use the previously computed subspace as a warm-start for the next subspace to gain more efficiency.

Acknowledgements: We would like to extend our gratitude to Maya Cakmak, Kamal Jain, and Gyanit Singh for several insightful discussions and useful feedback.

References

[1] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 2003. 2, 3

[2] T. Caetano, J. McAuley, L. Cheng, Q. Le, and A. Smola. Learning graph matching. *PAMI*, 31(6):1048–1058, 2009. 1, 3, 7

[3] E. Candes and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *Information Theory, IEEE Transactions on*, 52(12):5406–5425, 2006. 8

[4] T. Cour, P. Srinivasan, and J. Shi. Balanced graph matching. *NIPS*, 19:313, 2007. 1, 3

[5] E. Delponte, F. Isgrò, F. Odone, and A. Verri. Svd-matching using sift features. *Graphical models*, 68(5):415–431, 2006. 7

[6] D. DeMers, G. Cottrell, et al. Non-linear dimensionality reduction. *NIPS*, pages 580–580, 1993. 7

[7] O. Duchenne, F. Bach, I. Kweon, and J. Ponce. A tensor-based algorithm for high-order graph matching. *PAMI*, 2011. 7

[8] O. Duchenne, A. Joulin, and J. Ponce. A graph-matching kernel for object categorization. In *ICCV*, pages 1792–1799. IEEE, 2011. 7

[9] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *PAMI*, pages 594–611, 2006. 6

[10] F. Glover. Maximum matching in a convex bipartite graph. *Naval Research Logistics Quarterly*, 14(3):313–316, 1967. 2

[11] N. Goel, G. Bebis, and A. Nefian. Face recognition experiments with random projection. In *Proc. SPIE*, pages 426–437, 2005. 8

[12] G. Golub and C. Reinsch. Singular value decomposition and least squares solutions. *Numerische Mathematik*, 14(5):403–420, 1970. 7

[13] G. Golub and C. Van Loan. *Matrix computations*. Johns Hopkins University Press, 1996. 5, 7

[14] N. Halko, P. Martinsson, and J. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011. 5

[15] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*, volume 2. Cambridge Univ Press, 2000. 6

[16] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. In *ICCV*, 2005. 1, 3

[17] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. Freeman. Sift flow: Dense correspondence across different scenes. *Computer Vision–ECCV 2008*, pages 28–42, 2008. 6

[18] D. Lowe. Object recognition from local scale-invariant features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1150–1157. Ieee, 1999. 6

[19] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 6

[20] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *PAMI*, pages 1615–1630, 2005. 6

[21] C. Papadimitriou and K. Steiglitz. Combinatorial optimization: algorithms and complexity. 1998. 7

[22] D. Rueckert, L. Sonoda, C. Hayes, D. Hill, M. Leach, and D. Hawkes. Nonrigid registration using free-form deformations: application to breast mr images. *TMI*, pages 712–721, 1999. 6

[23] G. Scott and H. Longuet-Higgins. An algorithm for associating the features of two images. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 244(1309):21–26, 1991. 1, 2, 3, 7

[24] L. Shapiro and J. Michael Brady. Feature-based correspondence: an eigenvector approach. *ICV*, pages 283–288, 1992. 7

[25] X. Shen and F. Meyer. Analysis of event-related fmri data using diffusion maps. In *IPMI*, pages 103–137. Springer, 2005. 8

[26] M. Torki and A. Elgammal. One-shot multi-set non-rigid feature-spatial matching. In *CVPR*. IEEE, 2010. 1, 2, 6, 7

[27] M. Torki and A. Elgammal. Putting local features on a manifold. In *CVPR*, pages 1743–1750. IEEE, 2010. 1, 7

[28] L. Torresani, V. Kolmogorov, and C. Rother. Feature correspondence via graph matching: Models and global optimization. *ECCV*, pages 596–609, 2008. 1, 3

[29] S. Ullman. Aligning pictorial descriptions: an approach to object recognition. *Cognition*, 32(3):193–254, 1989. 6

[30] S. Umeyama. An eigen decomposition approach to weighted graph matching problems. *PAMI*, 10(5):695–703, 1988. 1, 7

[31] S. Vempala. *The random projection method*, volume 65. Amer Mathematical Society, 2005. 2, 4, 8