A Spectral Technique for Correspondence Problems Using Pairwise Constraints

Marius Leordeanu Martial Hebert

The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

We present an efficient spectral method for finding consistent correspondences between two sets of features. We build the adjacency matrix M of a graph whose nodes represent the potential correspondences and the weights on the links represent pairwise agreements between potential correspondences. Correct assignments are likely to establish links among each other and thus form a strongly connected cluster. Incorrect correspondences establish links with the other correspondences only accidentally, so they are unlikely to belong to strongly connected clusters. We recover the correct assignments based on how strongly they belong to the main cluster of M, by using the principal eigenvector of M and imposing the mapping constraints required by the overall correspondence mapping (one-to-one or one-to-many). The experimental evaluation shows that our method is robust to outliers, accurate in terms of matching rate, while being much faster than existing methods.

1. Introduction

There are many tasks in computer vision which require efficient techniques for finding consistent correspondences between two sets of features, such as object recognition, shape matching, wide baseline stereo, 2D and 3D registration. We propose an efficient technique that is suitable for such applications. Our method finds consistent correspondences between two sets of features, by taking in consideration both how well the features' descriptors match and how well their pairwise geometric constraints (or any other type of pairwise relationship) are satisfied. Our formulation can accommodate different kinds of correspondence mapping constraints, such as allowing a data feature to match at most one model feature (commonly used), or allowing a feature from one set to match several features from the other set (used in shape matching [1]).

The features could consist of points, lines, shape descriptors or interest points, depending on the specific application. For problems where the features are non discriminative ($e.g.$ points), it is the features pairwise geometric information that helps in finding the right correspondence. When discriminative features are extracted ($e.g.$ interest points) then both the geometry and the properties of each individual feature can be used.

Our approach avoids the combinatorial explosion inherent to the correspondence problem by taking advantage of the spectral properties of the weighted adjacency matrix M of a graph, whose nodes are the potential assignments $a = (i, i')$ and whose weights on edges measure the agreements between pairs of potential assignments (Section 2). When the two pairs of features put in correspondence by two potential assignments agree in terms of their pairwise geometry (or other type of pairwise relationship), there will be an *agreement link* (positive edge) formed between the two assignments. Otherwise there will be no link between the two assignments (edge of zero weight). We use the terms *assignment* and *correspondence* interchangebly .

Our method is based on the observation that the graph associated with M contains:

1. a main strongly connected cluster formed by the correct assignments that tend to establish agreement links (edges with positive weights) among each other. These agreement links are formed when pairs of assignments agree at the level of pairwise relationships ($e.g.$ geometry) between the features they are putting in correspondence.

2. a lot of incorrect assignments outside of that cluster or weakly connected to it, which do not form strongly connected clusters due to their small probability of establishing agreement links and random, unstructured way in which they form these links.

These statistical properties motivate our spectral approach to the problem. We start by first finding the level of *association* of each assignment with the main cluster, by inspecting the eigenvector of M corresponding to its largest eigenvalue (principal eigenvector). Then we keep rejecting the assignments of low association, until the constraints on the correspondence mapping are met (Section 3). Spectral methods are commonly used for finding the main clusters of a graph, in tasks such as segmentation [14], grouping [9],

Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05)
1550-5499/05 $20.00 © 2005 **IEEE**

[12], and change detection [11]. Shapiro and Brady [13] also proposed a spectral technique for correspondence problems, later improved by Carcassoni and Hancock [3], but their formulation is different and it applies only to matchings between point sets.

Maciel and Costeira [8] use a different formulation based on integral quadratic programming, which can be reduced to an equivalent concave minimization problem. However, the complexity of concave minimization is still non-polynomial. Berg and Malik [1] obtain a more efficient implementation that works only for the case of allowing several features from one image to match the same feature from the second image. Their approximation of the quadratic problem with $m^2 + 1$ linear programming problems is suitable for other mapping constraints as well (m is the number of features to match). We implement the approximation they used, for the case of one to one matchings, and compare it against our method. Our algorithm proves to be several orders of magnitude faster, while also being more robust to noise and outliers (Section 5).

Our work differs from existing approaches based on spectral methods or quadratic programming, in that it has a much better computational complexity, which allows it to scale much better to large data sets, while being robust to noise and outliers. Its low complexity is due to the relaxation during the optimization step of both the correspondence mapping constraints and the integral constraints on the solution. We show that this method is very robust to noise and outliers due to the spectral properties of M.

2. Problem formulation

Given two sets of features P, containing n_P data features, and Q, having n_Q model features, a *correspondence mapping* is a set C of pairs (or *assignments*) (i, i'), where $i \in P$ and $i' \in Q$. The features in P and Q that belong to some pair from C are the *inliers*. The features for which there is no such pair in C are the *outliers*. Different problems impose different kinds of *mapping constraints* on C, such as: allowing one feature from P to match at most one feature from Q, or allowing one feature from one set to match more features from the other. Our approach can accommodate different kinds of constraints.

For each candidate assignment $a = (i, i')$ there is an associated score or *affinity* that measures how well feature $i \in P$ matches $i' \in Q$. Also, for each pair of assignments (a, b), where $a = (i, i')$ and $b = (j, j')$, there is an *affinity* that measures how compatible the data features (i, j) are with the model features (i', j'). Given a list L of n candidate assignments, we store the affinities on every assignment $a \in L$ and every pair of assignments $a, b \in L$ in the n x n matrix M as follows:

1. $M(a, a)$ is the affinity at the level of individual assignments $a = (i, i')$ from L. It measures how well the data feature i matches the model feature i'. Assignments that are unlikely to be correct (due to a large distance between the descriptors of i and i') will be filtered out. Thus, each such rejection will reduce the number of rows and columns in M by one.

2. $M(a, b)$ describes how well the relative pairwise geometry (or any other type of pairwise relationship) of two model features (i', j') is preserved after putting them in correspondence with the data features (i, j). Here $a = (i, i')$ and $b = (j, j')$. If the two assignments do not agree ($e.g.$ the deformation between (i, j) and (i', j') is too large) or if they are incompatible based on the mapping constraints ($e.g$ $i = j$ and $i' \neq j'$) we set $M(a, b) = 0$. We assume $M(a, b) = M(b, a)$ without any loss of generality.

We require these affinities to be non-negative, symmetric ($M(a, b) = M(b, a)$), and increasing with the quality of the match, without any loss of generality. The candidate assignments $a = (i, i')$ from L can be seen as nodes forming an undirected graph, with the pairwise scores $M(a, b)$ as weights on the edges and the individual scores $M(a, a)$ as weights at the nodes. Then, M represents the affinity matrix of this undirected weighted graph. The number of nodes in this graph (= number of elements in L), adapts based on the actual data and it depends mainly on how discriminative the features's descriptors are. If the features are highly discriminative, such as SIFT descriptors, then only a small fraction of all possible pairs (i, i') are kept as candidate matches. In this case the size of M and the dimension of the problem search space are considerably reduced. When the features are non-discriminative (such as 2D or 3D points) and there is no a priori information about candidate matches ($e.g.$ constraints on translation), all possible pairs (i, i') can be considered as candidate assignments. In general, M is an n x n, sparse symmetric and positive matrix where $n = kn_P$, and k is the average number of candidate matches for each data feature $i \in P$. Each feature $i \in P$ will usually have a different number of candidate correspondences (i, i'), $i' \in Q$.

The correspondence problem reduces now to finding the cluster C of assignments (i, i') that maximizes the inter-cluster score $S = \sum_{a,b \in C} M(a, b)$ such that the mapping constraints are met. We can represent any cluster C by an indicator vector x, such that $x(a) = 1$ if $a \in C$ and zero otherwise. We can rewrite the total inter-cluster score as:

$$S = \sum_{a,b \in C} M(a, b) = x^T M x \tag{1}$$

The optimal solution x^* is the binary vector that maximizes the score, given the mapping constraints:

$$x^* = argmax(x^T M x) \tag{2}$$

The inter-cluster score $x^T M x$ depends mainly on three things: the number of assignments in the cluster, how interconnected the assignments are (number of links adjacent

Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05)
1550-5499/05 $20.00 © 2005 **IEEE**

to each assignment) and how well they agree (weights on the links). Previous approaches [1], [8] gave a quadratic programming formulation to the correspondence problem by embedding the mapping constraints on x in the general form of $Ax = b$. Instead, we relax both the mapping constraints and the integral constraints on x, such that its elements can take real values in $[0, 1]$. We interpret $x^*(a)$ as the *association* of a with the best cluster C^*. Since only the relative values between the elements of x matter, we can fix the norm of x to 1. Then, by the Raleigh's ratio theorem, x^* that will maximize the inter-cluster score $x^T M x$ is the principal eigenvector of M. Since M has non-negative elements, by Perron-Frobenius theorem, the elements of x^* will be in the interval $[0, 1]$. In Section 3 we describe how we use the mapping constraints to binarize the eigenvector and obtain a robust approximation to the optimum solution.

The main computational gain of our approach comes from dropping both the mapping constraints and the integral constraints during the optimization step, and using them only afterwards to binarize the eigenvector. The problem becomes one of finding the main cluster from the assignments graph and can be solved easily using the well known eigenvector technique. We show that this method is very robust, because the main cluster in the assignments graph is statistically formed by the correct assignments.

A key insight in the understanding of the statistics of M is that a pair of model features is very likely to agree (in terms of pairwise relationship between features) with the correct corresponding pair of data features. The same pair is very unlikely to agree with an incorrect pair of data features. Thus, correct assignments are expected to establish agreement links between them. At the same time, incorrect assignments are not expected to form such links, and when they do it occurs because of accidental alignments between features in the two data sets, which happen in a random, unstructured way. This suggests that the correct assignments will form a highly connected cluster with a high association score, while the wrong assignments will be weakly connected to other assignments and not form strong clusters. The larger the value in the eigenvector $x^*(a)$, the stronger the association of a with the main cluster. Since this cluster is statistically formed by correct assignments, it is natural to interpret $x^*(a)$ as the confidence that a is a correct assignment.

3. Algorithm

We propose a greedy algorithm for finding the solution to the correspondence problem. As discussed earlier, we interpret the eigenvector value corresponding to a particular assignment $a = (i, i')$ as the *confidence* that a is a correct assignment. We will refer to this value $x^*(a)$ as the $confidence$ of a. We start by first accepting as correct the assignment a^* of maximum confidence (for which the eigenvector value $x^*(a^*)$ is maximum), because it is the one we are most confident of being correct. Next we have to reject all other assignments that are in conflict with a^*, as dictated by the constraints on the correspondence mapping. In our experiments these are assignments of the form $(i, *)$ or $(*, i')$ (one feature $i \in P$ can match at most one feature $i' \in Q$ and vice-versa). Note that here one could use different constraints to find the assignments that are in conflict with a^*. We accept the next correct assignment as the one with the next highest confidence that has not been rejected and thus it is not in conflict with a^*. We continue by rejecting the assignments in conflict with the newly accepted assignment. We repeat this procedure of accepting new assignments of next highest confidence that are not in conflict with the ones accepted already, until all assignments are either rejected or accepted. This algorithm will split the set of candidate assignments in two: the set of correct assignments C^* and rejected assignments R, having the following property: every assignment from R will be in conflict with some assignments from C^* of higher confidence. Thus, no element from R can be included in C^* without having to remove from C^* an element of higher confidence.

The overall algorithm can be summarized as follows:

1. Build the symmetric non-negative n x n matrix M as described in Section 2.

2. Let x^* be the principal eigenvector of M. Initialize the solution vector x with the n x 1 zero vector. Initialize L with the set of all candidate assignments.

3. Find $a^* = argmax_{a \in L}(x^*(a))$. If $x^*(a^*) = 0$ stop and return the solution x. Otherwise set $x(a^*) = 1$ and remove a^* from L.

4. Remove from L all potential assignments in conflict with $a^* = (i, i')$. These are assignments of the form (i, k) and (q, i') for one-to-one correspondence constraints (they will be of the form (i, k) for one-to-many constraints).

5. If L is empty return the solution x. Otherwise go back to step 3.

We note that the outliers are found at steps 3 and 4. They belong to weak assignments incompatible with assignments of higher confidence, or to those that have a zero corresponding eigenvector value (step 3). Different kinds of constraints on the correspondence mapping can be used to remove the assignments conflicting with higher confidence assignments (step 4). Our approach takes advantage of the fact that these constraints are usually easy to check and it provides a simple way to enforce them as a post-optimization step. In practice our algorithm was several orders of magnitude faster than the linear programming approximation [1] to the quadratic problem, even for medium size data-sets (matching 15-20 points). In turn, the linear optimization approximation is less computationally expensive than the optimal quadratic programming approach [8].

Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05)
1550-5499/05 $20.00 © 2005 **IEEE**

In the case of rigid transformations the algorithm could be stopped after selecting enough assignments (step 3) for computing the transformation. We successfully use this modifed method on registration of $3D$ lines. However, the results in this paper do not use this idea.

4. Numerical considerations

4.1. Stability issues

It is important to verify that the principal eigenvector of the ideal matrix M^* (in which only correct assignments form pairwise links) is stable in the presence of outliers or deformations. Here we present an argument that shows that the principal eigenvector of M^* is indeed stable due to the statistics of M^*. In general, the principal eigenvector of a symmetric matrix M is robust to small perturbations $\|E\|$ if the difference between the first two largest eigenvalues of M is large [15], [10]. This difference is also known as the $eigengap$ of M. In general, our matrix M will be a slightly perturbed version of the ideal M^*, in which only the correct assignments establish pairwise agreement links among each other. In M^* there will be no accidental agreement links between wrong assignments (no accidental alignments), while the correct assignments will form a clique of pairwise agreements. Thus the off-diagonal elements $M^*(a,b)$ will have the maximum positive affinity value if a and b are correct assignments and $M^*(a,b) = 0$ otherwise. The largest eigenvalue λ_1^* of M^* will be equal to the total association score of the cluster formed by the correct assignments, while its second largest eigenvalue λ_2^* will equal the largest affinity score $M(a,a)$ of some wrong assignment a.

For a large enough number of correct assignments the eigengap ρ^* of M^* will be about the same as λ_1^*, so we can approximate $\rho^* \approx \lambda_1^*$. Moreover, the principal eigenvalue will be much larger in absolute value than all the other eigenvalues. Therefore the Frobenius norm of the ideal M^*, $\|M^*\|_F = \sqrt{\sum_i \lambda_i^{*2}}$ is slightly larger than λ_1^*, so we can approximate $\rho \approx \|M^*\|_F$. This approximation conforms to empirical observations. For example, we obtained the ideal matrix M^* experimentally, in matching sets of points (Section 5). We rotated and translated a set of points, without deforming it, to obtain the other set. The matrices obtained in this manner are very close to the ideal M^* and their ratio $\frac{\rho^*}{\|M^*\|_F}$ is approximately 1 even for small data sets (Figure 1).

Let v^* be the principal eigenvector of the ideal matrix M^* and v be the principal eigenvector of the actual matrix M, which is a slightly perturbed version of M^*: $M = M^* + E$. By Theorems $V.2.8$ from [15] and 1 from [10], if $\sqrt{2}\|E\|_F \leq \frac{\rho^*}{2}$, the perturbation of the eigenvector v^* satisfies the inequality:

$$\|v^* - v\|_2 \leq \frac{4\|E\|_F}{\rho - \sqrt{2}\|E\|_F} \tag{3}$$

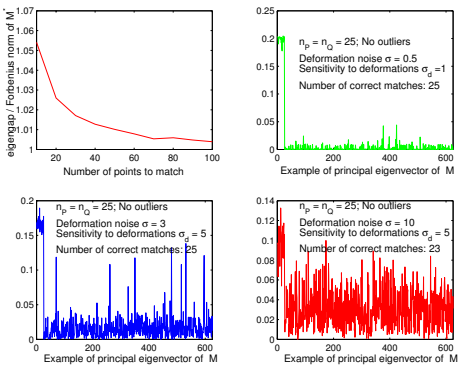

Figure 1: Top-right: $\rho^*/\|M^*\|_F$ vs.data set size. The other plots show principal eigenvectors (permuted such that the values of correct assignments show up on the first 25 entries)

In our case we approximate $\rho^* \approx \|M^*\|_F$. If we also use the inequality $\sqrt{2}\|E\|_F \leq \frac{\rho^*}{2}$, we get a rough upper bound ϵ^* on the change $\|v^* - v\|_2$ of the principal eigenvector:

$$\epsilon^* \approx 8\frac{\|E\|_F}{\|M^*\|_F} \tag{4}$$

This analysis agrees with the intuition that small perturbations $\|E\|_F$ relative to $\|M^*\|_F$ will not change significantly the direction of the principal eigenvector. In practice, even large perturbations $\|E\|_F$ that cause the formation or deletion of links in an unstructured way, will not produce higher eigenvector values for wrong assignments than for correct assignments. Only a structured perturbation, which causes wrong assignments to belong to strong clusters, can significantly affect the relative difference between the eigenvector values of correct assignments vs. wrong ones. These structured accidents happen when there is a lot of symmetry in the data, the deformations noise is high or there are a lot of outliers. Figure 1 shows how the principal eigenvector (obtained in experiments from Section 5) changes smoothly as we increase the deformation noise in the problem of matching sets of points.

In a related application on detecting structural changes Sarkar and Boyer [11] also discuss the robustness of the principal eigenvector and eigenvalue to small unstructured perturbations of matrices with similar statistics.

4.2. Complexity considerations

M is an n x n sparse matrix for which the efficient computation of its first eigenvector in step 2 is typically less than $O(n^{3/2})$. Variants of the Lanczos method, such as the one implemented by MATLAB function $eigs$ (which we used in our experiments) are very efficient for finding the principal eigenvector of large symmetric sparse matrices.

Since M is very sparse both its storage and computation can be made very efficient. Its space requirements and computation time will never reach the $O(n^2)$ complexity in practice. In our experiments (Section 5) M was on average

Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05)
1550-5499/05 $20.00 © 2005 IEEE

about 3% full. After finding the principal eigenvector, one can show that in the worst case, the number of the remaining steps is: $n + (n-1)...(n-m) = O((k-1/2)m^2)$, where $m = min(n_P, n_Q)$ and $k = n/m$. For problems where the features are very discriminative ($e.g.$, SIFT descriptors, shape context), k is expected to be very small as compared to m, since a feature from one set will have only a few possible potential matches in the other set. In the worst case, every data feature could potentially match any model feature which leads to $n = n_P n_Q$.

5. Experiments

We evaluate the robustness of our method first on the task of finding correspondences between $2D$ sets of points. This problem lets us evaluate the average performance of the algorithm for different levels of deformation and ratio of outliers to inliers. We study two main cases: when the deformation noise is added from a gaussian distribution with zero mean and equal variances on the points x-y coordinates, and when sets of points are deformed using the Thin Plate Spline model [16](TPS). The noise level is controlled by varying the variance of the gaussian distribution or the bending energy of the TPS model, and the number of outliers. We use the mapping constraint that one model feature can match at most one data feature and vice-versa.

5.1. Deformations using white noise

In the first set of experiments we generate data sets of 2D model points Q by randomly selecting n_Q^i inliers in a given region of the plane. We obtain the corresponding inliers in P by disturbing independently the n_Q^i points from Q with white gaussian noise $N(0, \sigma)$ and then rotating and translating the whole data set Q with a random rotation and translation. Next we add n_Q^o and n_P^o outliers in Q and P, respectively, by randomly selecting points in the same region as the inliers from Q and P, respectively, from the same random uniform distribution over the x-y coordinates. The range of the $x-y$ point coordinates in Q is $256\sqrt{n_Q/10}$ to enforce an approximately constant density of 10 points over a 256 x 256 region, as the number of points varies. The total number of points in Q and P are $n_Q = n_Q^i + n_Q^o$ and $n_P = n_P^i + n_P^o$. The parameter σ controls the level of deformations between the two sets, while n_P^o and n_Q^o control the number of outliers in P and Q, respectively. This is a difficult type of problem for two reasons. First, the points are non-discriminative and they can be translated and rotated arbitrarily, so any of the n_P points from P can potentially match any of the n_Q model points from Q. This maximizes the search space (the solution vector will have $n_Q n_P$ elements) and leaves the task of finding a solution entirely to the relative geometric information between pairs of points. Secondly, picking points randomly in the plane creates homogeneous data sets with an increased level of

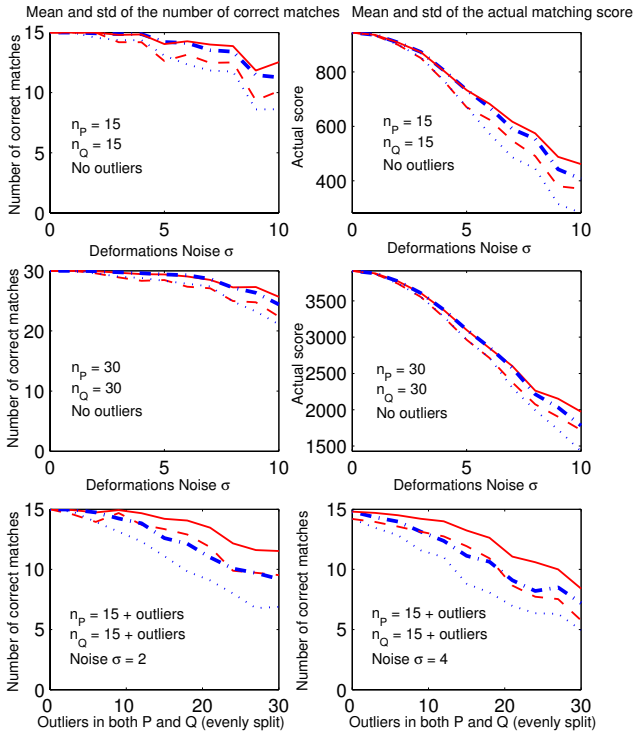

Figure 2: Performance curves for our method vs. *linprog* method. The mean performance is shown as a solid red line (our method) and a blue dash-dotted line (*linprog* method). One *std* below the mean: red dashed lines (our method), blue dotted lines (*linprog* method). First two rows: no outliers, varying deformation noise. The number of correct matches (left) and the actual scores (right) are plotted. Third row: the number of outliers in each P and Q is varied for two values of the deformation noise.

symmetry, which increases the chance of accidental agreements between wrong correspondences or between correct correspondences and wrong ones. Also, choosing outliers from the same distribution as the inliers, within the same region, increases the chance that similar geometrical relationships will be formed among outliers or between the outliers and the clean points as among the clean points only.

Since points are non-discriminative we set the score on individual assignments $M(a,a)$ to zero (we left the matching score entirely to the pairwise geometric information, since there is no information on the individual assignments). For the pairwise score $M(a,b)$ on deformations between candidate assignments $a = (i, i')$ and $b = (j, j')$ we use the pairwise distances between points:

$$M(a,b) = \begin{cases} 4.5 - \frac{(d_{ij} - d_{i'j'})^2}{2\sigma_d^2} & \text{if } |d_{ij} - d_{i'j'}| < 3\sigma_d \\ 0 & \text{otherwise,} \end{cases}$$

where the candidate assignments are $a = (i, i')$ and $b =$

Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05)
1550-5499/05 $20.00 © 2005 **IEEE**

(j, j'). d_{ij} and $d_{i'j'}$ are the Euclidean distances between the points i and j, and between their candidate matches i' and j', respectively. The parameter σ_d controls the sensitivity of the score on deformations. The larger σ_d the more deformations in the data we can accommodate, also the more pairwise relationships between wrong assignments will get a positive score. $M(a, b)$ defined above is always non-negative and increases as the deformation between candidate pairs of assignments decreases. The total score $S = x^T Mx$ increases as the number of assignment links that are below the deformation threshold of $3\sigma_d$ increases and as the sum of squared deformations on those links decreases.

Figure 2 shows the performance curves of our method vs. the linear programming approximation method [1] as we vary the noise σ from 0.5 to 10 (in steps of 0.5), the number of points to be matched: from 15 up to 30, and the number of outliers in both P and Q. We score the performances of the two methods by counting how many matches agree with the ground truth. We initially kept the sensitivity parameter fixed $\sigma_d = 5$. For our algorithm we use the $MATLAB$ function $eigs$, which implements the Implicitly Restarted Arnoldi method [5], a variant of Lanczos method. For the linear programming method we use the $MATLAB$ function $linprog$ with the $LargeScale$ option that is based on $LIPSOL$ (Linear Interior Point Solver, [17]). Both algorithms ran on the same problem sets over 30 trials for each value of the varying parameter (Figure 2). Both the mean performance curves as well as the curves one standard deviation below the mean are plotted. As expected, for large values of the deformation σ and large numbers of outliers, both algorithms start shifting smoothly from the correct matches, which indicates that some wrong assignments have established enough links to win over correct assignments. The performance of the two algorithms degrades in a similar manner, which suggests that the true optimum of the score function shifts from the ground truth as the amount of noise increases (as introduced by outliers and deformations). For lower values of the noise, both algorithms find the correct matches which indicates that the optimum of the score function coincides with the ground truth. Both algorithms prove to be robust to noise, but ours shows a slightly better robustness for larger values of noise. In the case of outliers our algorithm is clearly more robust then the *linprog* based method. Also, our method is orders of magnitude faster then $linprog$: over 400 times faster on 20 points problem sets (average time of 0.03 sec. vs 13 sec) and over 650 faster on 30 points problem sets (0.25 sec. vs 165 sec.), on a 2.4 GHz Pentium computer.

We further tested our method on the same problem, but larger data sets (50, 100 and 130 points, Figure 3), for which $linprog$ becomes too slow to make an extensive comparison. We notice that the performance of our algorithm improves as the number of points increases. This happens because as the number of data points increases, each correct assignment establishes pairwise relationships with more correct assignments. Thus it becomes more robust to deformations or presence of outliers. Our method took on average less than 9 seconds in MATLAB for 130 points problem sets on a 2.4 GHz Pentium computer.

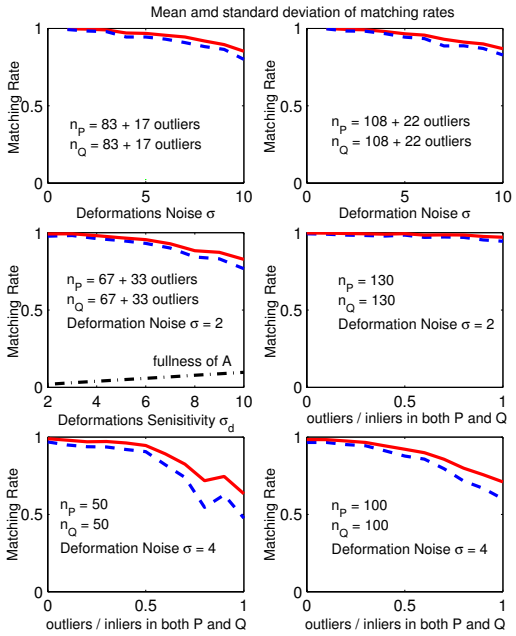

Figure 3: Average matching rates (=correctly matched inliers vs. total inliers) over 30 tests for each parameter value on the x axis. Middle-right: also plotted (black dash-dotted line) the ratio of non-zero values in M vs. total elements in M. The more deformation we allow (σ_d) the less sparse M is

We also tried to simulate real applications of matching very large number of points. We limited the number of candidate correspondences per point, by accepting as candidate matches (i, i') only points that were within a radius of 500 from each other. When generating the inliers in P from the inliers in Q, we limited the translation to 100 and the rotation around the center of mass of the points in Q to $[-\pi/9, \pi/9]$. This resulted in each point from P having on average around 100 candidate correspondences in Q, including its correct correspondence. We imposed the additional constraint on the pairwise distances score that $M(a, b) = 0$ if $d_{ij} > 200$ or $d_{i'j'} > 200$, or the angle between the directions of (i, j) and (i', j') was outside the interval $[-\pi/9, \pi/9]$. The average performances, over 30 runs, on data sets of 400, 600 and 1000 points, for a deformation noise of $\sigma = 2$ and ratio of $outliers/inliers = 50\%$ in each set P and Q, were: 97% (400 points) and 93% (both 600 and 1000 points). It took less than 30 sec. in $MATLAB$ on a 2.4 GHz Pentium computer to find the solution for 1000 points.

Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05)
1550-5499/05 $20.00 © 2005 **IEEE**

5.2. Non rigid deformations using the Thin Plate Spline Model

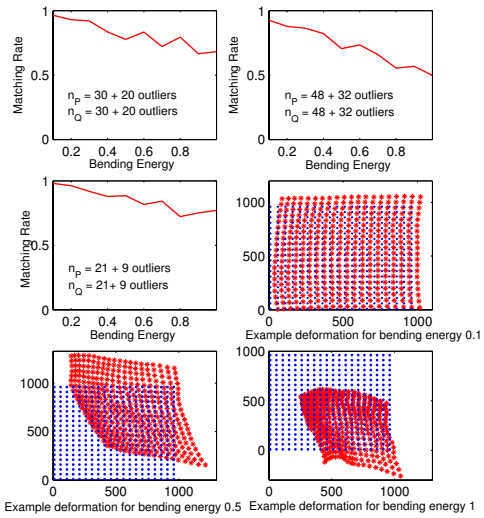

Figure 4: Top row: average matching rate of 50 points (top-left), 80 points (top-right) and 30 points (bottom-right) for different bending energy levels, over 30 trials for each energy level. Middle and bottom rows: examples of the x-y meshgrid deformation for bending energy levels: 0.1, 0.5 and 1

In this Section we test our method on deformations that follow the TPS model, for which the amount of deformation is quantified by the bending energy applied to a x-y meshgrid (Figure 4). We generated the sets Q and P as follows: we randomly pick n_Q^i inliers in Q on a 20 by 20 x-y meshgrid. Then we randomly deform the meshgrid for a given level of the bending energy and obtain the corresponding n_P^i inliers from P. We further add n_Q^o and n_P^o outliers in Q and P by randomly picking points in the plane in a region that contains the inliers. As in previous tests, we enforce the mapping constraint that one point from P must match at most one point from Q and vice-versa. The score function used before on pairwise deformations is not appropriate in this case, since the pairwise distance between far points is expected to vary more than the distance between close points. Instead, we normalize the pairwise deformations by the absolute pairwise distance $d_{i'j'}$. We also penalize changes in directions. As before, we do not use any scores on individual assignments ($M(a,a) = 0$). For pairwise deformations we use a score function that is similar in concept to the one from [1]: $M(a,b) = (1-\gamma)c_\alpha + \gamma c_d$, if $|\alpha_{ab}| < 3\sigma_\alpha$ and $|d_{ab} - 1| < 3\sigma_d$, and zero otherwise. Here, $c_\alpha = 4.5 - \frac{\alpha_{ab}^2}{2\sigma_\alpha^2}$, $c_d = 4.5 - \frac{(d_{ab}-1)^2}{2\sigma_d^2}$, $d_{ab} = \frac{d_{ij}+q}{d_{i'j'}+q}$, d_{ij} and $d_{i'j'}$ are the distances between the model features (i,j) and between the data features (i',j'), respectively, and α_{ab} is the angle between the direction of (i,j) and that of (i',j'). The term c_α penalizes changes in direction, c_d penalizes changes in the relative length, while γ weighs one term against the other. TPS is often used as a deformation model for matching shapes. Figure 5 shows a couple of examples of matching point sets sampled from natural images, using the scores defined here.

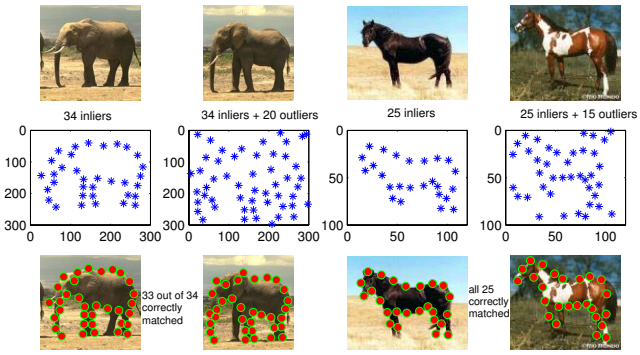

Figure 5: Correspondences between points from natural images

5.3. Recognizing objects from low resolution images

In this Section we show an application of our method to recognition of vehicles from aerial images, using the DoG feature detector and the SIFT feature descriptor, [7]. In this case, because of the low resolution images, the normal voting method for retaining the correct matches [7] cannot be applied reliably because the number of features extracted for each object is very small and their location and scale is not very stable. Moreover, the SIFT descriptors are less discriminative when applied to low resolution and low texture objects. For example, a lot of these features are extracted at the cars boundaries, and it often happens that multiple features from the same or different objects are very similar to each other. Therefore, we must allow multiple candidate matches for each feature and use the pairwise relationships between them to disambiguate the correct correspondences. We built 70 car models Q_q from 70 video sequences, in an unsupervised fashion, following the method described in [6]. The models represent constellations of clusters of SIFT features, that are grouped together, into the same model, if they co-occur with a high probability during the video sequence within a small distance from each other. Given a test image P containing an object and clutter, the task is to recognize it by trying to match it against a subset of the models built. The model Q_q^* that gives the highest correspondence score $x^T M x$ will be retrieved as the correct match. We used a pairwise score $M(a,b)$ that is similar to the ones used previously: it is high if the distance d_{ij} between the centers of the data features i and j is similar to the average distance $d_{i'j'}$ (during the training stage) between the centers of their candidate model matches i' and j' ($a = (i,i')$, $b = (j,j')$):

Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05)
1550-5499/05 $20.00 © 2005 **IEEE**

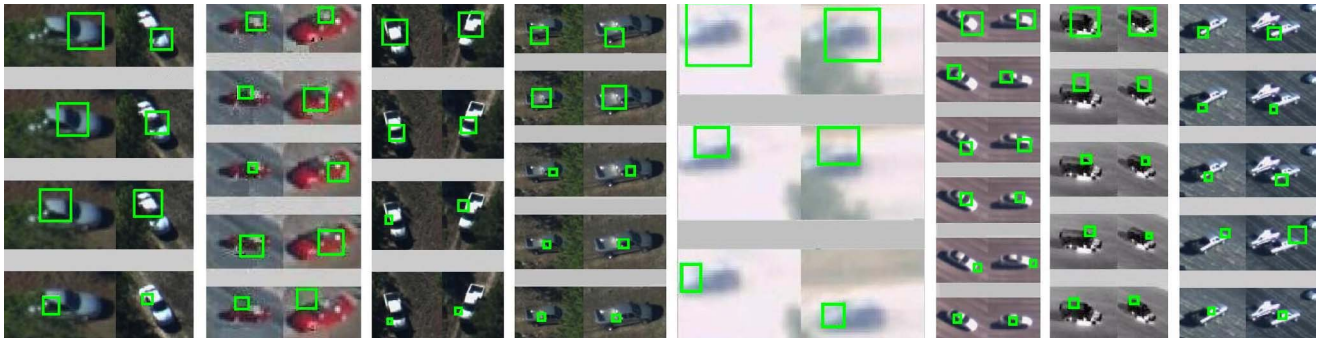

Figure 6: Examples of correspondences between data features and model features in recognizing cars from aerial images

$$M(a,b) = \begin{cases} 4.5 - \frac{(d_{ij} - d_{i'j'})^2}{2\sigma_d^2} & \text{if } |d_{ij} - d_{i'j'}| < 3\sigma_d \\ 0 & \text{otherwise,} \end{cases}$$

For the individual assignments $M(a,a)$ we used a score that is linearly decreasing with the L_2 distance between feature i and its candidate corresponding feature i'.

We matched 506 novel images against the correct model plus 10 other randomly selected models and we recognized correctly 496 images ($\approx 98\%$ recognition rate). Figure 6 shows sample matches between the features from the test images (left) and features from the models (right).

6. Conclusions

We have presented an efficient spectral solution to correspondence problems using pairwise constraints between candidate assignments. The problem formulation makes our solution suitable for a variety of vision applications from $2D$ or $3D$ registration to object recognition. Our approach takes advantage of the fact that correct assignments are likely to establish links among each other, while incorrect ones are unlikely to form these links and when they do, this happens in an unstructured, random way. Then, the recovery of the correct solution becomes a problem of detecting the main strongly connected cluster in the assignments graph. We showed in our experiments that the spectral approach is robust to noise and outliers, and it scales well with the number of features.

7. Acknowledgements

This work was sponsored in part by Grant NBCH1030013. The authors wish to thank David Tolliver for helpful discussions. The content does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

References

[1] A.C. Berg, T. Berg, J. Malik "Shape Matching and Object Recognition using Low Distortion Correspondences", TR CSD-04-1366, Dec. 2004

[2] P. J. Besl, N. D. McKay. "A method for registration of 3-d shapes" PAMI pp. 239-256, Feb 1992

[3] M. Carcassoni and Edwin R. Hancock, "Alignment using Spectral Clusters", In Proc. BMVC, 2002

[4] T. Kadir, M. Brady "Saliency, Scale and Image Description", In Proc. ICCV, pp. II: 83-105, 2001

[5] Lehoucq, R.B., D.C. Sorensen, and C. Yang, "ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods", SIAM Publications, 1998.

[6] M. Leordeanu, R. Collins, "Unsupervised Learning of Object Features from Video Sequences", Proc. CVPR 2005

[7] David G. Lowe "Object Recognition from Local Scale-Invariant Features", In Proc. ICCV, pp 1150-1157, 1999

[8] J. Maciel, J. Costeira "A Global Solution to Sparse Correspondence Problems" PAMI, February 2003

[9] S. Mahamud, L.R. Williams, K.K. Thornber, K. Xu "Segmentation of Multiple Salient Closed Contours from Real Images", PAMI, April 2003

[10] A. Y. Ng, A. X. Zheng, M.I. Jordan, "Link Analysis, Eigenvectors and Stability", In Proc. IJCAI pp. 903-910, 2001

[11] S. Sarkar, K.L. Boyer "Quantitative Measures of Change Based on Feature Organization: Eigenvalues and Eigenvectors", CVIU, July. 1998

[12] G.L. Scott, H. C. Longuet-Higgins "Feature grouping by relocalisation of eigenvectors of the proximity matrix", In Proc. BMVC, p 103-108, 1990

[13] L.S. Shapiro and J.M. Brady, "Feature-based correspondence - an eigenvector approach" Image and Vision Computing, 10, pp. 283-288, 1992

[14] J. Shi, J. Malik "Normalized Cuts and Image Segmentation", PAMI, August 2000

[15] G.W. Stewart and Ji-Guang Sun, "Matrix Perturbation Theory", Academic Press, 1990

[16] Wahba, G., "Spline Models of Observational Data", SIAM Publications, 1990

[17] Zhang, Y., "Solving Large-Scale Linear Programs by Interior-Point Methods Under the MATLAB Environment" TR96-01, University of Maryland, July 1995.

Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05)
1550-5499/05 $20.00 © 2005 **IEEE**