Unsupervised Learning for Graph Matching

Marius Leordeanu
Carnegie Mellon University
Pittsburgh, PA
mleordea@andrew.cmu.edu

Martial Hebert
Carnegie Mellon University
Pittsburgh, PA
hebert@ri.cmu.edu

Abstract

Graph matching is an important problem in computer vision. It is used in 2D and 3D object matching and recognition. Despite its importance, there is little literature on learning the parameters that control the graph matching problem, even though learning is important for improving the matching rate, as shown by this and other work. In this paper we show for the first time how to perform parameter learning in an unsupervised fashion, that is when no correct correspondences between graphs are given during training. We show empirically that unsupervised learning is comparable in efficiency and quality with the supervised one, while avoiding the tedious manual labeling of ground truth correspondences. We also verify experimentally that this learning method can improve the performance of several state-of-the art graph matching algorithms.

1. Introduction

Graph matching is an important problem in computer vision. It is becoming widely used especially in 2D shape and object matching and recognition [3], [10], [14], [17], matching articulated 3D objects [7], and unsupervised modeling of object categories [13], [9]. While there are many papers on solving it efficiently [3], [10], [6], [8], [15], [16], [18] there are only two papers published previously, to the best of our knowledge, that propose a solution for learning the optimal set of parameters for graph matching, in the context of computer vision applications [4], [11]. However, as shown by [4], [11] and also by us in this paper, learning the parameters is important for improving the matching performance.

We show for the first time how to efficiently perform unsupervised learning for graph matching in the context of object matching/recognition. Unsupervised learning for matching is important in practice, since manual labeling of correspondences can be quite time consuming. The same basic algorithm can be used in the supervised case with minimal modification, if the ground truth matches are available. We also show empirically that our learning algorithm is robust to the presence of outliers. This method is inspired from the properties of spectral matching [10], but it can improve the performance of other state-of-the-art matching algorithms as shown in our experiments.

Learning for Graph Matching The graph matching problem consists of finding the indicator vector $\mathbf{x}^*$ that maximizes a quadratic score function:

$$\mathbf{x}^* = argmax(\mathbf{x}^\mathbf{T}\mathbf{M}\mathbf{x}). \quad (1)$$

Here $\mathbf{x}$ is an indicator vector such that $\mathbf{x}_{ia} = 1$ if feature i from one image (or object model) is matched to feature a from the other image (or object model) and zero otherwise. Usually, one-to-one constraints are imposed on $\mathbf{x}$ such that one feature from one image can be matched to at most one other feature from the other image. In spectral matching $\mathbf{M}$ is a matrix with positive elements containing the pairwise score functions, such that $\mathbf{M}_{ia;jb}$ measures how well the pair of features (i, j) from one image agrees in terms of geometry and appearance (e.g. difference in local appearance descriptors, pairwise distances, angles, etc) with a pair of candidate matches (a, b) from the other. The local appearance terms of candidate correspondences can be stored on the diagonal of $\mathbf{M}$; in practice we noticed that including them in the pairwise scores $\mathbf{M}_{ia;jb}$, and leaving zeros on the diagonal gives better results; $\mathbf{M}_{ia;jb}$ is basically a function that is defined by a certain parameter vector $\mathbf{w}$. Then, learning for graph matching consists of finding $\mathbf{w}$ that maximizes the performance (w.r.t to the ground truth correspondences) of matching (as defined by Equation 1) over pairs of training images.

Our matching algorithm [10] interprets each element of the principal eigenvector $\mathbf{v}$ of $\mathbf{M}$ as the confidence that the corresponding assignment is correct. It starts by choosing the element of maximum confidence as correct, then it removes (zeroes out in $\mathbf{v}$) all the assignments in conflict (w.r.t the one-to-one mapping constraints) with the assignment chosen as correct, then it repeats this procedure until all assignments are labeled either correct or incorrect.

978-1-4244-3991-1/09/$25.00 ©2009 IEEE

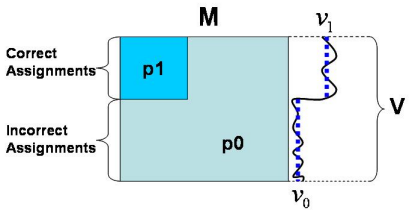

Figure 1. Pairwise scores (elements of the matching matrix $\mathbf{M}$) between correct assignments have a higher expected value p_1 than elements with at least one wrong assignment, with expected value p_0. This will be reflected in the eigenvector $\mathbf{v}$ that will have higher expected value v_1 for correct assignments than v_0 for wrong ones.

2. Theoretical Analysis

Our proposed algorithm is motivated by the statistical properties of the matrix $\mathbf{M}$ and of its leading eigenvector $\mathbf{v}$ that is used to find a binary solution to the matching problem. In order to analyze the properties of $\mathbf{M}$ theoretically, we need a few assumptions and approximations, which we validate experimentally. Each instance of the matching problem is unique so nothing can be said with absolute certainty about $\mathbf{M}$ and its eigenvector $\mathbf{v}$, nor the quality of the solution returned. Therefore, we must be concerned with the average (or expected) properties of $\mathbf{M}$ rather than the infinitely many particular cases. We propose a model for $\mathbf{M}$ (Figure 1) that we validate through experiments.

Let $p_1 > 0$ be the expected value (average value over infinitely many matching experiments of the same type) of the second-order scores between correct assignments $E(\mathbf{M}_{ia;jb})$ for any pair (ia, jb) of correct assignments. Similarly, let $p_0 = E(\mathbf{M}_{ia;jb}) \geq 0$ if at least one of the assignments ia and jb is wrong. The assumption here is that the expected values of the second order scores do not depend on the particular assignments ia or jb, but only on whether these assignments are correct or not. p_1 should be higher than p_0, since the pairs of correct assignments are expected to agree both in appearance and geometry and have strong second-order scores, while the wrong assignments have such high pairwise scores only accidentally. We expect that the higher p_1 and the lower p_0, the higher the matching rate. We also expect that this performance depends on their ratio $p_r = p_0/p_1$ and not on their absolute values, since multiplying $\mathbf{M}$ by a constant does not change the leading eigenvector. Since the model assumes the same expected value p_1 for all pairwise scores between correct assignments (and p_0 for all pairwise scores including a wrong assignment), and since the norm of the eigenvector does not matter, we can also assume that all correct assignments ia will have the same mean eigenvector confidence value $v_1 = E(\mathbf{v}_{ia})$, and all wrong assignments jb will have the same $v_0 = E(\mathbf{v}_{jb})$. The spectral matching algorithm assumes that the correct assignments will correspond to large elements of the eigenvector $\mathbf{v}$ and the wrong assignments to low values in $\mathbf{v}$, so the higher v_1 and the lower v_0 the better the matching rate. As in the case of p_r, if we could minimize during learning the average ratio $v_r = v_0/v_1$ (since the norm of the eigenvector is irrelevant) over all image pairs in a training sequence then we would expect to optimize the overall training matching rate. This model assumes fully connected graphs, but it can be verified that the results we obtain next are also valid for weakly connected graphs, as also shown in our experiments.

It is useful to investigate the relationship between v_r and p_r for a given image pair. We know that $\lambda \mathbf{v}_{ia} = \sum_{jb} \mathbf{M}_{ia;jb}\mathbf{v}_{jb}$. Next we assume that for each of the n features in the left image there are k candidate correspondences in the right image. We also approximate $E(\sum_{jb} \mathbf{M}_{ia;jb}\mathbf{v}_{jb}) \approx \sum_{jb} E(\mathbf{M}_{ia;jb})E(\mathbf{v}_{jb})$, by considering that any $\mathbf{v}_{jb}$ is *almost* independent of any particular $\mathbf{M}_{ia;jb}$, since $\mathbf{M}$ is large. The approximation is actually a $\geq$ inequality, since the correlation is expected to be positive (but very small). It follows that for a correct correspondence ia, $\lambda E(\mathbf{v}_{ia}) = \lambda v_1 \approx np_1v_1 + n(k-1)p_0v_0$. Similarly, if ia is a wrong correspondence then $\lambda E(\mathbf{v}_{ia}) = \lambda v_0 \approx np_0v_1 + n(k-1)p_0v_0$. Dividing both equations by p_1v_1 and taking the ratio of the two we obtain:

$$v_r \approx \frac{p_r + (k-1)p_rv_r}{1 + (k-1)p_rv_r}. \tag{2}$$

Solving for v_r we get:

$$v_r \approx \frac{(k-1)p_r - 1 + \sqrt{1 - (k-1)p_r + 4(k-1)p_r^2}}{2(k-1)p_r}. \tag{3}$$

It can be verified that by this equation v_r is a monotonically increasing function of p_r. This is in fact not surprising since we expect that the smaller $p_r = p_0/p_1$, the smaller $v_r = v_0/v_1$ and the more binary the eigenvector $\mathbf{v}$ would be (and closer to the binary ground truth $\mathbf{t}$), with the elements of the wrong assignments approaching 0. This approximation turns out to be very accurate in practice, as shown by our experiments in Figures 6, 7 and 8. Also, the smaller v_r, the higher the expected matching rate. One way to minimize v_r is to maximize the correlation between $\mathbf{v}$ and the ground truth indicator vector $\mathbf{t}$. However, in this paper we want to minimize v_r in an unsupervised fashion, that is without knowing $\mathbf{t}$ during training. Our proposed solution is to maximize instead the correlation between $\mathbf{v}$ and its binary version (that is, the binary solution returned by the matching algorithm). How do we know that this procedure will ultimately give a binary version of $\mathbf{v}$ that is close to the real ground truth ? We will investigate this question next.

Let $\mathbf{b}(\mathbf{v})$ be the binary solution obtained from $\mathbf{v}$, respecting the one-to-one mapping constraints, as returned by

spectral matching for a given pair of images. Let us assume for now that we know how to maximize the correlation $\mathbf{v^T b(v)}$. We expect that this will lead to minimizing the ratio $v_r^* = E(\mathbf{v}_{ia}|\mathbf{b}_{ia}(\mathbf{v}) = 0)/E(\mathbf{v}_{ia}|\mathbf{b}_{ia}(\mathbf{v}) = 1)$. If we let n_m be the number of misclassified assignments, n the number of true correct assignments (same as the number of features, equal in both images) and k the number of candidate assignments for each feature, we can obtain the next two equations: $E(\mathbf{v}_{ia}|\mathbf{b}_{ia}(\mathbf{v}) = 0) = \frac{n_m v_1 + (n(k-1) - n_m)v_0}{n(k-1)}$ and $E(\mathbf{v}_{ia}|\mathbf{b}_{ia}(\mathbf{v}) = 1) = \frac{n_m v_0 + (n - n_m)v_1}{n}$. Dividing both by v_1 and taking the ratio of the two we finally obtain:

$$v_r^* = \frac{m/(k-1) + (1 - m/(k-1))v_r}{1 - m + mv_r}, \qquad (4)$$

where m is the matching error rate $m = n_m/n$. If we reasonably assume that $v_r < 1$ (eigenvector values slightly higher on average for correct assignments than for wrong ones) and $m < (k-1)/k$ (error rate slightly lower than random) this function of m and v_r has both partial derivatives strictly positive. Since m also increases with v_r, by maximizing $\mathbf{v^T b(v)}$, we minimize v_r^*, which minimizes both v_r and the true error rate m, so the unsupervised algorithm is expected to do the right thing. In all our experiments we obtained values for all p_r, v_r, v_r^* and m very close to zero, which is sufficient in practice even if we did not necessarily find the global minimum using our gradient based method (Section 3).

One can also show that by maximizing $\mathbf{b(v)^T v}$ the solution we obtain gets closer to the solution that would be obtained if an optimal algorithm were used. Any normalized vector $\mathbf{b}$ gives a quadratic score that obeys the following optimality bound as a function of the correlation $\mathbf{b^T v}$:

$$\frac{\mathbf{b^T M(w) b}}{\mathbf{x_{opt}(w)^T M(w) x_{opt}(w)}} \geq 2(\mathbf{b^T v(w)})^2 - 1, \qquad (5)$$

where $\mathbf{x_{opt}(w)}$ is the optimal solution of Equation 1 for a given $\mathbf{w}$. Therefore, by maximizing $\mathbf{b(v)^T v}$, we maximize this lower bound and expect $\mathbf{b(v)}$ to approach the optimal solution $\mathbf{x_{opt}(w)}$. This is true for solutions returned by any approximate graph matching algorithm if we maximize instead the correlation between the eigenvector and the solution returned by that specific algorithm, which suggests that the same unsupervised learning scheme may be applied to other algorithms as well.

3. Algorithms

3.1. Supervised Learning

We want to find the geometric and appearance parameters $\mathbf{w}$ that maximize (in the supervised case) the expected correlation between the principal eigenvector of $\mathbf{M}$ and the ground truth $\mathbf{t}$, which empirically is proportional to the following sum over all training image pairs:

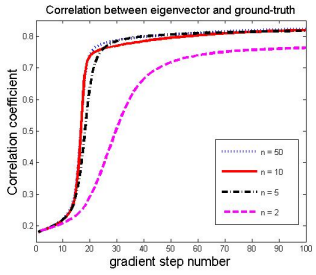

Figure 2. Experiments on the House sequence. The plots show the normalized correlation between the eigenvector and the ground truth solution for different numbers of recursive iterations n used to compute the approximative derivative of the eigenvector (averages over 70 experiments). Even for n as small as 5 the learning method converges in the same way, returning the same result.

$$J(\mathbf{w}) = \sum_{i=1}^{N} \mathbf{v}^{(i)}(\mathbf{w})^T \mathbf{t}^{(i)}, \qquad (6)$$

where $\mathbf{t}^{(i)}$ is the ground truth indicator vector for the i-th training image pair. We maximize $J(\mathbf{w})$ by coordinate gradient ascent:

$$\mathbf{w}_j^{k+1} = \mathbf{w}_j^k + \eta \sum_{i=1}^{N} \mathbf{t_i}^T \frac{\partial \mathbf{v_i}^{(k)}(\mathbf{w})}{\partial \mathbf{w}_j}. \qquad (7)$$

To simplify notations throughout the rest of the paper we use F' for the vector or matrix of partial derivatives of any vector or matrix F. One possible way of taking partial derivatives of an eigenvector of a symmetric matrix (when λ has order 1) is given in [5], in the context of spectral clustering:

$$\mathbf{v}' = (\lambda \mathbf{I} - \mathbf{M})^\dagger (\lambda' \mathbf{I} - \mathbf{M}')\mathbf{v}, \qquad (8)$$

where

$$\lambda' = \frac{\mathbf{v}^T \mathbf{M}' \mathbf{v}}{\mathbf{v}^T \mathbf{v}}. \qquad (9)$$

These equations are obtained by using the fact that $\mathbf{M}$ is symmetric and the equalities $\mathbf{v^T v'} = 0$ and $\mathbf{Mv} = \lambda \mathbf{v}$. However, this method is general and therefore does not take full advantage of the fact that in this case $\mathbf{v}$ is the principal eigenvector of a matrix with large eigengap. $\mathbf{M} - \lambda \mathbf{I}$ is large and also rank deficient so computing its pseudo-inverse is not efficient in practice. Instead, we use the power method to compute the partial derivatives to the approximate principal eigenvector: $\mathbf{v} = \frac{\mathbf{M}^n \mathbf{1}}{\sqrt{(\mathbf{M}^n \mathbf{1})^T (\mathbf{M}^n \mathbf{1})}}$. This seems to be related to [1], but in [1] the method is used for segmentation and as also pointed out by [5] it could be very unstable in that case, because in segmentation and typical clustering problems the eigengap between the first two eigenvalues is not large.

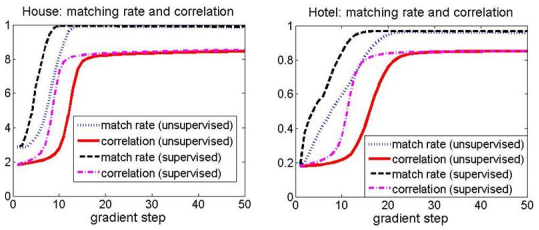

Figure 3. Supervised vs. unsupervised learning: Average match rate and correlation between the eigenvector and the ground truth over all training image pairs, over 70 different experiments (10 randomly chosen training images from the House (Hotel, respectively) sequence). The standard deviations are not significant

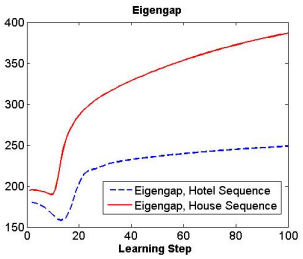

Figure 4. During unsupervised learning, the normalized eigengap (eigengap divided by the mean value in $\mathbf{M}$) starts increasing after a few iterations, indicating that the leading eigenvector becomes more and more stable. Results are on the House and Hotel datasets averaged over 70 random experiments.

Here $\mathbf{M}^n\mathbf{1}$ is computed recursively by $\mathbf{M}^{k+1}\mathbf{1} = \mathbf{M}(\mathbf{M}^k\mathbf{1})$. Since the power method is the preferred choice for computing the leading eigenvector, it is justified to use the same approximation for learning. Thus the estimated derivatives are not an approximation, but actually the exact ones, given that $\mathbf{v}$ is itself an approximation based on the power method. Thus, the resulting partial derivatives of $\mathbf{v}$ are computed as follows:

$$\mathbf{v}' = \frac{(\mathbf{M}^n\mathbf{1})'(\|\mathbf{M}^n\mathbf{1}\|) - \mathbf{M}^n\mathbf{1}/\|\mathbf{M}^n\mathbf{1}\|((\mathbf{M}^n\mathbf{1})^T(\mathbf{M}^n\mathbf{1})')}{\|\mathbf{M}^n\mathbf{1}\|^2}. \tag{10}$$

In order to obtain the derivative of $\mathbf{v}$, we first need to compute the derivative of $\mathbf{M}^n\mathbf{1}$, which can be obtained recursively:

$$(\mathbf{M}^n\mathbf{1})' = \mathbf{M}'(\mathbf{M}^{n-1}\mathbf{1}) + \mathbf{M}(\mathbf{M}^{n-1}\mathbf{1})'. \tag{11}$$

Since $\mathbf{M}$ has a large eigengap, as shown in [10], this method is stable and efficient. Figure 2 proves this point empirically. The method is linear in the number of iterations n, but qualitatively insensitive to n, as it works equally well with n as low as 5. These results are averaged over 70 experiments (described later) on 900 by 900 matrices.

To get a better feeling of the efficiency of our method as compared to Equation 8, computing Equation 4 takes 1500 times longer in Matlab (using the function $pinv$) than our method for $n = 10$ on 900 by 900 matrices used in our experiments on the House and Hotel datasets.

3.2. Unsupervised Learning

The idea for unsupervised learning (introduced in Section 2), is to maximize instead the function:

$$J(\mathbf{w}) = \sum_{i=1}^{N} \mathbf{v}^{(i)}(\mathbf{w})^T \mathbf{b}(\mathbf{v}^{(i)}(\mathbf{w})). \tag{12}$$

The difficulty here is that $\mathbf{b}(\mathbf{v}^{(i)}(\mathbf{w}))$ is not a continuous function and also it may be impossible to express in terms of $\mathbf{w}$, since $\mathbf{b}(\mathbf{v}^{(i)}(\mathbf{w}))$ is the result of the iterative greedy procedure of the spectral matching algorithm. However, it is important that $\mathbf{b}(\mathbf{v}^{(i)}(\mathbf{w}))$ is piecewise constant and has zero derivatives everywhere except for a finite set of discontinuity points. We can therefore expect that we will evaluate the gradient only at points where $\mathbf{b}$ is constant, and has zero derivatives. Also, at those points, the gradient steps will lower v_r (Equation 4) because changes in $\mathbf{b}$ (when the gradient updates pass through discontinuity points in $\mathbf{b}$), do not affect v_r. Lowering v_r will increase $\mathbf{v}^\mathbf{T}\mathbf{t}$ and also decrease m, so the desired goal will be achieved without having to worry about the discontinuity points of $\mathbf{b}$. This has been verified every time in our experiments. Then, the learning step function becomes:

$$\mathbf{w}_j^{k+1} = \mathbf{w}_j^k + \eta \sum_{i=1}^{N} \mathbf{b}(\mathbf{v_i^{(k)}}(\mathbf{w}))^T \frac{\partial \mathbf{v_i}^{(k)}(\mathbf{w})}{\partial \mathbf{w}_j}. \tag{13}$$

4. Experimental Analysis

We focus on two objectives. The first one is to validate the theoretical results from Section 2, especially Equation 3, which establishes a relationship between p_r and v_r, and Equation 4, which connects v_r^* to v_r and the error rate m. Each p_r is empirically estimated from each individual matrix $\mathbf{M}$ over the training sequence, and similarly each v_r^* and v_r from each individual eigenvector. Equation 3 is important because it shows that the more likely the pairwise agreements between correct assignments as compared to pairwise agreements between incorrect ones (as reflected by p_r), the closer the eigenvector $\mathbf{v}$ is to the binary ground truth $\mathbf{t}$ (as reflected by v_r), and, as a direct consequence, the better the matching performance. This equation also validates our model for the matching matrix $\mathbf{M}$, which is defined by two expected values, p_0 and p_1, respectively. Equation 4 is important because it explains why by maximizing the correlation $\mathbf{v}^\mathbf{T}\mathbf{b}(\mathbf{v})$ (and implicitly minimizing v_r^*) we in

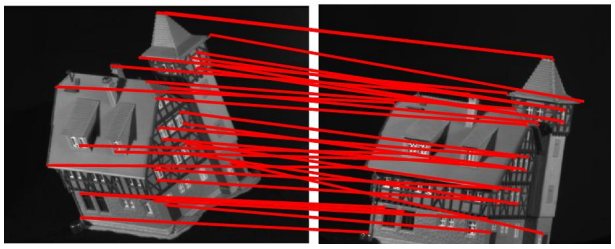

Figure 5. After learning the pairwise scores for matching, all the features in the first image are correctly matched to the features from the last image (House sequence).

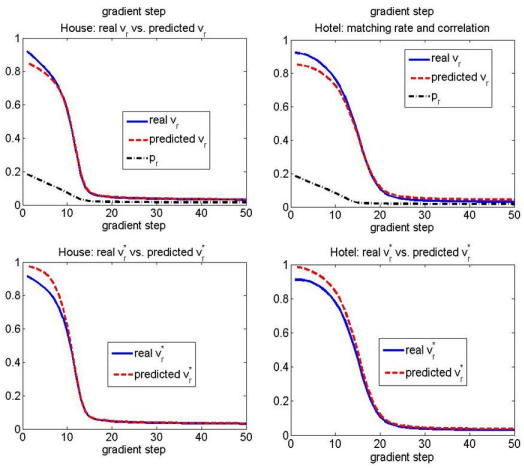

Figure 6. Learning stage: the plots show how the left hand side of Equations 3 and 4, that is v_r and v_r^*, estimated empirically from the eigenvectors obtained for each image pair, agree with their predicted values (right hand side of Equations 3 and 4). Results are averages over 70 different experiments, with insignificant standard deviations.

fact minimize v_r and the matching error m. Equation 4 basically shows why the unsupervised algorithm will indeed maximize the performance with respect to the ground truth. The results that validate our theoretical claims are shown in Figures 6, 7 and 8 on the House, Hotel, Faces, Cars and Motorbikes experiments. The details of these experiments will be explained shortly. There are a few relevant results to consider. On all 4 different experiments the correlation between $\mathbf{v}$ and the ground truth $\mathbf{t}$ increases with every gradient step even though the ground truth is unknown to the learning algorithm. The matching rate improves at the same time and at a similar rate with the correlation, showing that maximizing this correlation will also maximize the final performance. In Figure 7 we display a representative example of the eigenvector for one pair of faces, as it becomes more and more binary during training. If after the first iteration the eigenvector is almost flat, at the last iteration is very close to the binary ground truth, with all correct assignments having larger confidences than any of the wrong ones. Also, on all individual experiments both approximations from Equations 3 and 4 are becoming more and more accurate with each gradient step, from less than 10% accuracy at the first iteration to less than 0.5% at the last. In all our learning experiments we started from a set of parameters $\mathbf{w}$ that does not favor any assignment ($\mathbf{w} = 0$, which means that before the very first iteration all non-zeros scores in $\mathbf{M}$ are equal to 1). These results motivate both the model proposed for $\mathbf{M}$ (Equation 3), but also the results (Equation 4) that support the unsupervised learning scheme.

The second objective of our experiments is to evaluate the matching performance, before and after learning, on new test image pairs. The goal is to show that at testing time, the matching performance after learning is significantly better than if no learning was done.

4.1. Unlabeled correspondences

Matching Rigid Objects under Perspective Transformations We first perform experiments on two tasks that are the same as the ones in [4] and [11]. We used exactly the same image sequences (House: 110 images and Hotel: 100 images) both for training and testing and the same features, which were manually selected by the authors of [4]. As in [11] and [4], we used 5 training images for both the House and Hotel sequences, and considered all pairs between them for training. For testing we used all the pairs between the remaining images. The pairwise scores $\mathbf{M}_{ia;jb}$ are similar to [11], using shape context [2] for local appearance and pairwise distances and angles for the second-order relationships. They measure how well features (i, j) from one image agree in terms of geometry and appearance with their candidate correspondences (a, b).

More explicitly, the pairwise scores are of the type:

$$\mathbf{M}_{ia;jb} = e^{-(w_1|s_i - s_a| + w_2|s_j - s_b| + w_3 \frac{|d_{ij} - d_{ab}|}{|d_{ij} + d_{ab}|} + w_4|\alpha_{ij} - \alpha_{ab}|)}. \quad (14)$$

Learning consists of finding the vector of parameters $\mathbf{w}$ that maximizes the matching performance on the training sequence. s_a is the shape context of features a, d_{ij} is the distance between features (i, j) and α_{ij} the angle between the horizontal axis and the vector $\vec{ij}$. As in both [4] and [11] we first obtain a Delaunay triangulation and allow non-zero pairwise scores $\mathbf{M}_{ia;jb}$ if and only if both (i, j) and (a, b) are connected in their corresponding triangulation. The method of [11] is supervised and based on a global optimization scheme that is more likely to find the true global optimum than our unsupervised gradient based method. Therefore it is important to see that our unsupervised learning method matches the results from [11], while significantly outperforming [4] (Table 1).

Table 1. Matching performance on the hotel and house datasets at testing time. The same 5 training images from the House dataset and same testing images from the House and Hotel datasets were used for all three methods.

Dataset	Ours, unsup(5)	[11], sup(5)	[4], sup(5)
House	99.8%	99.8%	$< 84\%$
Hotel	94.8%	94.8%	$< 87\%$

Table 2. Comparison of average matching performance at testing time on the house and hotel datasets for 70 different experiments (10 training images, the rest used for testing). We compare the case of unsupervised learning vs. no learning. First column: unsupervised learning; Second: no learning, equal default weights $\mathbf{w}$.

Datasests	Unsup. Learning	No Learning
House+Hotel	99.14%	93.24%

Next we investigate the performance at learning and testing stages of the unsupervised learning method vs. its supervised version (when the ground truth assignments are known). We perform 70 different experiments using both datasets, by randomly choosing 10 training images (and using all image pairs from the training set) and leaving the rest of image pairs for testing. As expected we found that the unsupervised method learns a bit slower on average than the supervised one but the parameters learned are almost identical. In Figure 3 we plot the average correlation (between the eigenvectors and ground truth) and matching rate at each gradient step for all training pairs and all experiments vs. each gradient step, for both the supervised and unsupervised cases. It is interesting that while the unsupervised version tends to converge slower, after several iterations their performances (and also parameters) converge to the same values. During testing the two methods performed identically in terms of matching performance (average percentage of correctly matched features over all 70 experiments). As compared to the same matching algorithm without learned parameters the two algorithms performed clearly better (Table 2). Without learning the default parameters (elements of $\mathbf{w}$) were chosen to be all equal.

Matching Deformable 2D Shapes with Outliers The third dataset used for evaluation consists of 30 random image pairs selected from Caltech-4 Faces dataset. The experiments on this dataset are different from the previous ones for two reasons: the images contain not only faces but also a significant amount of background clutter, and, the faces belong to different people, both women and men, with different facial expressions, so there are significant non-rigid deformations between the faces that have to be matched. The features we used are oriented points sampled along contours

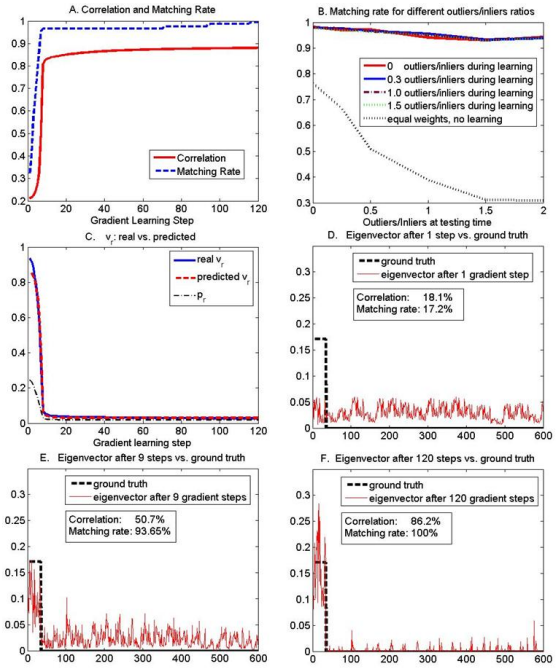

Figure 7. Results on faces: correlation between eigenvectors and ground truth, and matching rate during training (top left), matching rate at testing time, for different outliers/inliers ratios at both learning and test time (top-right), verifying Equation 3 (middle-left), example eigenvector for different learning steps. Results in the first three plots are averages over 30 different experiments.

extracted in the image in a similar fashion as in [12].The orientation of each point is the normal vector at that point to the contour where the point was sampled. The points on the faces that have to be matched (the inliers) were selected manually, while the outliers (features in the background) were selected randomly, while making sure that each outlier is not too close (15 pixels) to any other point. For each pair of faces we manually selected the ground truth (the correct matches) for the inliers only. The pairwise scores contain only geometric information about pairwise distances and angles:

$$\mathbf{M}_{ia;jb} = e^{-\mathbf{w}^T \mathbf{g}_{ia;jb}}, \tag{15}$$

where $\mathbf{w}$ is a vector of 7 parameters (that have to be learned) and $\mathbf{g_{ia;jb}} = [|d_{ij} - d_{ab}|/d_{ij}, |\theta_i - \theta_a|, |\theta_j - \theta_b|, |\sigma_{ij} - \sigma_{ab}|, |\sigma_{ji} - \sigma_{ba}|, |\alpha_{ij} - \alpha_{ab}|, |\beta_{ij} - \beta_{ab}|]$. Here d_{ij} is the distance between the features (i, j), θ_i is the angle between the normal of feature i and the horizontal axis, σ_{ij} is the angle between the normal at point i and the vector $\vec{ij}$, α_{ij} is the angle between $\vec{ij}$ and the horizontal axis and β_{ij} is the angle between the normals of i and j.

We performed 30 random experiments (see results in Figure 7) by randomly picking 10 pairs for training and

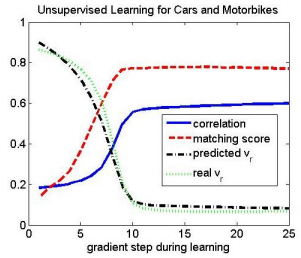

Figure 8. Correlation and matching rate w.r.t the ground truth during unsupervised learning for Cars and Motorbikes from Pascal 2007 challenge. Real and predicted v_r decrease as predicted by the model. Results are averaged over 30 different experiments

Table 3. Comparison of matching rates for 3 graph matching algorithms before and after unsupervised learning on Cars and Motorbikes from Pascal07 database, with all outliers from the right image allowed and no outliers in the left image. When no outliers were allowed all algorithms had a matching rate of over 75%, with learning moderately improving the performance.

Dataset	SM	PM	GA
Cars: No Learning	26.3%	20.9%	**31.9%**
Cars: With Learning	**62.2%**	34.2%	47.5%
Motorbikes: No Learning	29.5%	26.1%	**34.2%**
Motorbikes: With Learning	**52.7%**	41.3%	45.9%

leaving the rest 20 for testing. The results shown in Figure 7 are averages over the 30 experiments. The top-left plot shows how, as in the previous experiments, both the correlation $\mathbf{v^T t}$ and the matching performance during training improves with every learning step. At both training and testing times we used different percentages of outliers to evaluate the robustness of the method (top-right plot). The learning method is robust to outliers, since the matching performance during testing does not depend on the percentage of outliers introduced during training (the percentage of outliers is always the same in the left and the right images), but only on the percentage of outliers present at testing time. Without learning (the dotted black plot), when the default parameters chosen are all equal, the performance is much worse and degrades faster as the percentage of outliers at testing time increases. This suggests that learning not only increases the matching rate, but it also makes it more robust to the presence of outliers.

4.2. Unlabeled object classes and correspondences

In our previous experiments every pair of training images contained the same object/category, so a set of inliers exists for each such pair. Next, we evaluated the algorithm on a more difficult task: the training set is corrupted such that half of the image pairs contain different object categories. In this experiment we used cars and motorbikes from Pascal 2007, a much more difficult dataset. For each class we selected 30 pairs of images and for each pair between 30 to 60 ground truth correspondences. The features and the pairwise scores were of the same type as in the experiments on faces: points and their normals selected from pieces of contours. In Figure 9 we show some representative results after learning, with matching rates over 80%; contours are overlaid in white. During each training experiment we randomly picked 5 pairs containing cars, 5 containing motorbikes and 10 discordant pairs: one containing a car and the other one a motorbike (a total of 20 pairs for each learning experiment). For testing we used the remaining pairs of images, such that each pair contains the same object class. The learning algorithm had no knowledge of which pairs are discordant, what classes they contain and which are the ground truth correspondences. As can be seen in Figure 8 at each gradient step both the matching rate and the correlation of the eigenvector w.r.t the ground truth increases (monitored only for pairs containing the same category). The model proposed is again verified as shown by the plots of the real and ideal v_r that are almost identical. Not only that the learning algorithm was not significantly influenced by the presence of discordant pairs but it was also able to find a single set of parameters that matched well both cars and motorbikes. Learning and testing results are averaged over 30 experiments.

Using the testing image pairs of cars and motorbikes, we investigated whether this learning method can improve the performance of other graph matching algorithms. We compared spectral matching (SM) using the row/column procedure from [18] during post-processing of the eigenvector, with probabilistic matching (PM) using pair-wise constraints [18], and the well-known graduated assignment algorithm [8] (GA). The same parameters and pair-wise scores were used by all algorithms. When no outliers were allowed all algorithms had similar matching rates (above 75%) with learning moderately improving the performance. When outliers were introduced in the right image (in the same fashion as in the experiments on Faces) the performance improvement after learning was much more significant for all algorithms, with spectral matching benefiting the most (Table 3). Spectral matching with learning outperformed the other algorithms with or without learning. This indicates that the algorithm we propose is useful for other graph matching algorithms, but it might be better suited for spectral matching.

5. Conclusion

We presented an efficient way of performing both supervised and unsupervised learning for graph matching, in the context of computer vision applications. We showed that the performance of our unsupervised learning algorithm is

Figure 9. Matching results on image pairs from Pascal 2007 challenge. Best viewed in color

comparable with the one in the supervised case. The algorithm significantly improves the matching performance of several state-of-the are graph matching algorithms, which makes it widely applicable.

6. Acknowledgements

This work was supported by NSF Grant IIS0713406 and Intel Graduate Fellowship.

References

[1] F. Bach and M. Jordan. Learning spectral clustering. In *NIPS*, 2003.

[2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape context. *PAMI*, 24(4):509–522, April 2002.

[3] A. Berg, T. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. In *ECCV*, 2006.

[4] T. Caetano, L. Cheng, Q. Le, and A. J. Smola. Learning graph matching. In *ICCV*, 2007.

[5] T. Cour, J. Shi, and N. Gogin. Learning spectral graph segmentation. In *International Conference on Artificial Intelligence and Statistics*, 2005.

[6] T. Cour, P. Srinivasan, and J. Shi. Balanced graph matching. In *NIPS*, 2006.

[7] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H. Seidel, and S. Thrun. Performance capture from sparse multi-view video. In *SIGGRAPH*, 2008.

[8] S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. In *PAMI*, 1996.

[9] G. Kim, C. Faloutsos, and M. Hebert. Unsupervised modeling of object categories using link analysis techniques. In *CVPR*, 2008.

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

[11] M. Leordeanu and M. Hebert. Smoothing-based optimization. In *CVPR*, 2008.

[12] M. Leordeanu, M. Hebert, and R. Sukthankar. Beyond local appearance: Category recognition from pairwise interactions of simple features. In *CVPR*, 2007.

[13] D. Parikh and T. Chen. Unsupervised learning of hierarchical semantics of objects. In *CVPR*, 2007.

[14] X. Ren. Learning and matching line aspects for articulated objects. In *CVPR*, 2007.

[15] C. Schellewald and C. Schnorr. Probabilistic subgraph matching based on convex relaxation. In *EMMCVPR*, 2005.

[16] P. Torr. Solving markov random fields using semi definite programming. In *AIS*, 2003.

[17] L. Torresani, V. Kolmogorov, and C. Rother. Feature correspondence via graph matching: Models and global optimization. In *Tech Report, MSR-TR-2008-101*, 2008.

[18] R. Zass and A. Shashua. Probabilistic graph and hypergraph matching. In *CVPR*, 2008.