Learning Edge-Specific Kernel Functions For Pairwise Graph Matching

Michael Donoser[1]
donoser@icg.tugraz.at
Martin Urschler[2]
martin.urschler@cfi.lbg.ac.at
Horst Bischof[1]
bischof@icg.tugraz.at

[1] Institute for Computer Graphics and Vision
Graz University of Technology, Austria

[2] Ludwig Boltzmann Institute for Clinical Forensic Imaging
Graz, Austria

Abstract

In this paper we consider the pairwise graph matching problem of finding correspondences between two point sets using unary and pairwise potentials, which analyze local descriptor similarity and geometric compatibility. Recently, it was shown that it is possible to learn optimal parameters for the features used in the potentials, which significantly improves results in supervised and unsupervised settings. It was demonstrated that even linear assignments (not considering geometry) with well learned potentials may improve over state-of-the-art quadratic assignment solutions. In this paper we extend this idea by directly learning edge-specific kernels for pairs of nodes. We define the pairwise kernel functions based on a statistical shape model that is learned from labeled training data. Assuming that the setting of graph matching is a priori known, the learned kernel functions allow to significantly improve results in comparison to general graph matching. We further demonstrate the applicability of game theory based evolutionary dynamics as effective and easy to implement approximation of the underlying graph matching optimization problem. Experiments on automatically aligning a set of faces and feature-point based localization of category instances demonstrate the value of the proposed method.

1 Introduction

Graphs are frequently used as underlying representation for various computer vision tasks like object localization [18], action recognition [10] or 3D reconstruction [23]. In all these applications, nodes in general represent local features in the image and edges correspond to spatial relations between them. The goal of graph matching is to find correspondences between the nodes of two provided graphs, analyzing local descriptor similarity (unary potentials) and spatial relations between the nodes e. g. pairwise relations [15] or sometimes higher-order relations, for example analyzing triangles [14].

Graph matching has become widely used in several applications including tracking [13], shape matching [3] or object detection [18]. Various approaches for efficiently solving this NP-hard problem in an approximated manner are available, as e. g. spectral techniques [9], probabilistic methods [24] or the graduated assignment method [11]. Surprisingly, only few papers focused on the important graph potentials themselves, which have a tremendous

http://dx.doi.org/10.5244/C.26.17

influence on the quality of the obtained graph matching results. For example in [4] it turned out that solving a linear assignment problem using well chosen potentials even improves over related state-of-the-art quadratic assignment solutions. Furthermore, as shown in [7] also appropriate normalization of the different potentials is quite important.

In general, potentials used in graph matching are defined by their order. Single order (so called unary) potentials solely evaluate the single node compatibility, e. g. comparing local descriptors. A frequently used unary potential is the well known Shape Context [2] which analyzes the local distribution of the remaining graph nodes in a log polar histogram. Pairwise potentials analyze the compatibility of edges between pairs of nodes, e. g. the normalized differences between edge lengths. Such second order potentials allow rotation and translation invariant matching. Recently some effort has also been put on defining higher order potentials [14, 24] for building invariants to larger classes of transformations, for example achieving invariance to similarity transformations by considering triangle angles. By additionally analyzing the local edge orientation, even projectively invariant potentials can be designed for triangles [9].

One important challenge of using powerful potentials in graph matching is their appropriate parametrization. Mostly, these parameters are selected manually. Despite so much progress in the field of graph matching, only a few papers focused on the problem of choosing the right parameters. Caetano et al. [4] showed how to learn optimal parameters for the features used in the potentials in a supervised setting from manually labeled reference data sets. In Leordeanu et al. [17] this idea was extended to the unsupervised setting without the need for labeled data. They automatically learned weights for shape context features, edge length distances and angle distances in a pairwise graph matching setting and demonstrated a significant matching quality improvement in several settings even outperforming the supervised approach of [4]. By all means, both approaches strongly agree on the fact that learning the parameters is important for improving the matching performance.

In this paper we follow the idea of learning optimal parameters for the task of graph matching. Instead of learning fixed parameters for the features used in the potentials as done in [4, 17], we directly learn edge-specific kernel functions for each node pair in a defined reference graph structure, assuming that the setting of graph matching is a-priori known. Such a-priori knowledge is indeed available in several important applications like automated face alignment, model fitting and object localization.

We assume that we have given an annotated reference data set for our target scenario, e. g. for matching faces, we assume that several face images, including salient points (eyes, nose, ...) and their correspondences, are given as training data. Such point correspondences allow the estimation of a statistical model of shape [8] describing the distribution of point locations. Based on the learned model we can define pairwise kernel functions which are then used to solve a standard pairwise graph matching problem between a reference graph and a query graph in an improved manner.

This paper has several contributions. First, we show that statistical shape models can be easily integrated in the pairwise graph matching setting and that they enable significant improvement of matching results in several computer vision settings. We further advocate the usage of a game theory based evolutionary dynamics for efficiently solving the underlying optimization problem, and outline its similarities and advantages in comparison to the more frequently applied spectral matching methods. Finally, we demonstrate the potential of our method in the field of object localization with possible extensions to higher order potentials for achieving invariance to larger classes of transformations.

2 Overview of Method

In standard graph matching, optimal correspondences are found without any knowledge of the type of graph. The underlying idea of our novel method is to improve matching results by constraining the type of graphs considered, e. g. to faces or specific object categories. Thus, our method matches a reference graph of a known class to any query graph with the goal of finding optimal correspondences between the two graph node sets. This has several applications in computer vision like point set alignment, model fitting or localization of object instances. We only assume that the type of the graph is known in advance and that we have access to a set of labeled training images for this type. As we show in the experiments, actually only a few training samples are sufficient to significantly improve matching quality.

Our approach is divided into two main steps. First, in the training step, we learn a statistical shape model from labeled training images, obtaining a model of the location uncertainties of the graph nodes. Our model is defined by edge-specific kernel functions for every pair of nodes, which are defined in Section 3. Second, during testing, our method is an extension of standard graph matching formulated as quadratic assignment problem. The main difference is that we exploit the learned kernel functions for improving matching quality. Section 4 describes our graph matching step in detail.

3 Learning Pairwise Kernel Functions

As a first step we define our pairwise kernel functions by estimation of a statistical model of shape describing the distribution of point locations from a set of labeled training images. Our definition is inspired from statistical shape models and especially point distribution models which have seen a lot of attention over the previous years in the form of Active Shape Models and related or derived approaches [5].

We formulate our kernel functions $\mathcal{K}_{ij}$ to relate an edge connecting points i and j in our reference graph to an edge connecting points a and b in the query graph by deriving the statistics of the point locations, that are obtained from a labeled training set. We assume that we have given a set of training images, with the same number of labeled points in each image, where we require the labeled points to be corresponding over the training set. In a first step, we register all labeled points of the training set to each other, which then allows to describe the spatial distribution of each point over the training set by a Gaussian. We then define edge-specific kernel functions $\mathcal{K}_{ij}$ for measuring the spatial compatibility of any query edge (a,b) to the corresponding model edge (i,j).

Formally, we have a graph model consisting of N points with a graph node i at each point location for all training samples. We assume a densely connected graph with edges (i,j) between every pair of points in the model. Since we have to filter out differences in global orientation over the training set, we perform an iterative procedure involving Procrustes Analysis [12] to align each training shape to a mean shape as described in [6]. Procrustes Analysis minimizes the sum of squared distances $D = \sum(\mathbf{x}_i - \bar{\mathbf{x}})^2$ of each training shape to the mean shape $\bar{\mathbf{x}}$ under a similarity transformation (rotation + translation + uniform scale), where $\mathbf{x_i}$ is the location of a point i. After this procedure the training shapes are aligned to a mean shape $\bar{\mathbf{x}}$ which is centered at the origin and has unit scale. Now that all training shapes are globally aligned, the location uncertainties, due to nonlinear deformations and variations between the training samples, are modeled as Gaussian distributions $\mathcal{N}_i = (\mu_i, \Sigma_i)$, where μ_i is the mean and Σ_i is the covariance matrix of the point distribution. We derive the parameters μ_i, Σ_i for

Figure 1: Building a statistical shape model from labeled training data. Unaligned point sets (left), Procrustes aligned sets (middle) and obtained location uncertainties (right) are shown.

every point in our model by performing a Principal Component Analysis on the locations of the points in the training samples. Thus, we receive an ellipse as representation for location uncertainty as it is illustrated in Figure 1.

Finally, we define the kernel function $\mathcal{K}_{ij}(ab)$ which relates a pair of edges (i,j), (a,b) in terms of squared Mahalanobis distances defined by

$$r_1(ij,ab) = ((\mathbf{x}_a + \mathbf{v}_{ab}) - \mu_j)^T \Sigma_j^{-1} ((\mathbf{x}_a + \mathbf{v}_{ab}) - \mu_j) \,, \tag{1}$$

$$r_2(ij,ab) = ((\mathbf{x}_b + \mathbf{v}_{ba}) - \mu_i)^T \Sigma_i^{-1} ((\mathbf{x}_b + \mathbf{v}_{ba}) - \mu_i) \,, \tag{2}$$

with $\mathbf{v}_{ab}$ being the vector pointing from node a to node b. Thus, r_1 is the squared Mahalanobis distance with respect to node j, if we place our query edge (a,b) onto the model such that node a coincides with the mean location μ_i of node i and r_2 is defined equivalently. Finally, the kernel function for a graph edge connecting nodes (i,j) is defined as

$$\mathcal{K}_{ij}(ab) = \frac{r_1(ij,ab) + r_2(ij,ab)}{2} \,. \tag{3}$$

Since this kernel is simply a linear combination of squared Mahalanobis distances, the obtained matrix is guaranteed to be positive semi-definite and is thus a valid kernel function. Please note, that of course at test time only the coordinates of the query points (a,b) have to be inserted into above equations, since the mean and covariances are already fixed after training. This formulation is only valid if the distribution of the point locations follows a Gaussian model. For more complex distributions, Gaussian Mixture Models would be more appropriate. But as it is demonstrated in the experiments in Section 5 learning single Gaussian models already achieves excellent results with reduced computation time and no need to specify the number of mixture models.

4 Pairwise Graph Matching

In this paper we focus on graph matching formulated in the quadratic assignment problem (QAP) setting, where we assume that we have given a reference graph with N_1 nodes and labels $i,j,k,\ldots$ and a query graph with N_2 nodes and labels $a,b,c,\ldots$. The goal of graph matching is to find a one-to-one mapping between the two graphs defined by a binary $N_1 N_2 \times 1$ assignment vector $\mathbf{x}^*$, where $x^*_{ia} = 1$ if node i of the reference graph matches to node a of

the query graph and $x^*_{ia} = 0$ otherwise. The assignment vector is found in an optimization procedure by solving

$$\mathbf{x}^* = \underset{\mathbf{x}}{argmax}\left(\mathbf{x}^T \mathbf{A}\,\mathbf{x}\right) = \underset{\mathbf{x}}{argmax} \sum A_{ia,jb}\, x_{ia}\, x_{jb} \text{ and } \sum_i x^*_{ia} = 1, \sum_a x^*_{ia} = 1\,, \tag{4}$$

where $\mathbf{A}$ is a provided $N_1 N_2 \times N_1 N_2$ affinity matrix describing how well a pair of nodes in the reference (i, j) agrees in terms of local descriptors (unary potentials) and geometry (pairwise potentials) with a pair of nodes in the query (a, b). The affinity matrix $\mathbf{A}$ only represents pairwise potentials, therefore the unary term is mostly placed in the main diagonal or spread over the entire matrix.

We now first explain how to define the affinity matrix $\mathbf{A}$ considering the learned kernel functions in Section 4.1 and then describe in Section 4.2 how to obtain an approximate solution for the NP-hard integer optimization problem shown in Equation 4.

4.1 Affinity Matrix Using Learned Potentials

The affinity matrix used in pairwise graph matching mainly combines unary and pairwise potentials and needs parameters to be set, which is mostly done manually. One commonly used example [9] for defining the matrix using unary and pairwise potentials is

$$A_{ia,jb} = exp - \left(\mathbf{w}_1^T |\mathbf{s}_i - \mathbf{s}_a| + \mathbf{w}_2^T |\mathbf{s}_j - \mathbf{s}_b| + \mathbf{w}_3^T \frac{|d_{ij} - d_{ab}|}{d_{ij} + d_{ab}} \right), \tag{5}$$

where $\mathbf{s}_i$ denotes the 60-dimensional shape context descriptor [2] for node i, d_{ij} is the edge length between nodes i and j and $\mathbf{w}_1$,$\mathbf{w}_2$ and $\mathbf{w}_3$ are corresponding weight vectors, e. g. $\mathbf{w}_1$ is a 60-dimensional vector defining the weights for the bins in the shape context representation.

Extensions to the model in Equation 5 find optimal parameters w_i for the individual feature entries [4, 17]. In contrast, we propose to use the learned statistical shape model as described in Section 3 to define the pairwise affinity values $A_{ia,jb}$. We let the unary potentials (shape context) unchanged and replace the pairwise term (normally simply analyzing the differences in edge lengths) by our learned edge-specific kernel functions. For this reason, the affinity matrix entries are adapted to

$$A_{ia,jb} = exp - \left(\mathbf{w}_1^T |\mathbf{s}_i - \mathbf{s}_a| + \mathbf{w}_2^T |\mathbf{s}_j - \mathbf{s}_b| + \mathbf{w}_3^T \mathcal{K}_{ij}(ab) \right), \tag{6}$$

where $\mathcal{K}_{ij}(ab)$ is the learned pairwise kernel function as defined in Section 3. In such a way differences from the reference model are not equally penalized as in the standard setting, they are penalized depending on the location uncertainty learned by the statistical shape model. We always set the weights $\mathbf{w}_1$,$\mathbf{w}_2$ and $\mathbf{w}_3$ to the same value to be able to uniquely demonstrate the improved performance when using our kernel functions.

4.2 Graph Matching Optimization

After definition of the affinity matrix, any of the available approximation methods for solving the NP hard integer optimization problem shown in Equation 4 can be applied. The most common way to find the assignment vector $\mathbf{x}^*$ is to relax the integer optimization problem into the continuous domain and to apply spectral methods [16], which analyze the statistical properties of the provided affinity matrix $\mathbf{A}$. Recently, Albarelli et al. [1] demonstrated that game theory can be used to solve the quadratic assignment problem, where the problem

is also relaxed into the continuous domain but an evolutionary dynamic is applied to find the solution. We also consider game theory as an efficient and powerful field for graph matching, but in contrast to [1], who proposed an algorithm denoted as immunization and infection algorithm, we stick to the more popular replicator dynamics. Especially, we aim at highlighting similarities between spectral methods and replicator dynamics, which, although having two totally different scientific perspectives, share many common parts.

Replicator dynamics are the most popular evolutionary algorithm in the field of game theory. We apply these dynamics to solve the optimization problem defined in Equation 4 by iteratively updating the assignment vector $\mathbf{x}$ using

$$x_i^{t+1} = x_i^t \frac{(\mathbf{A}\,\mathbf{x}^t)_i}{\mathbf{x}^{t\,T}\mathbf{A}\,\mathbf{x}^t}, \tag{7}$$

where $\mathbf{x}^t$ is the assignment vector at time t. As a necessary additional constraint $\mathbf{x}$ has to lie on the simplex Δ defined as

$$\Delta = \left\{\mathbf{x} \in \mathbb{R}^{N_1 N_2} : x_i \geq 0 \quad \textit{and} \quad \mathbf{1}^T\mathbf{x} = 1\right\}, \tag{8}$$

where $\mathbf{1}$ is an $N_1 N_2$-dimensional vector of ones, i. e. $\sum x_i = 1$. The dynamics start with a random initialization Π which also has to lie on the simplex. We always initialize the dynamics by a slightly perturbed version of the barycenter of the simplex. Starting from Π, replicator dynamics find an optimal assignment vector $\mathbf{x}^*$ lying on the simplex which is a local (!) maximum of the optimization problem shown in Equation 4. Finally, the entries of the solution $\mathbf{x}^*$ represent normalized probabilities for all possible correspondences. To obtain a binary correspondence vector from the continuous vector $\mathbf{x}^*$, we apply the same strategy as proposed in [16], which iteratively chooses the most likely correspondence and then removes all contradictory correspondences by checking all assignment constraints.

The simplex Δ is invariant under the replicator dynamics formulation, which means that every trajectory starting on the simplex (random initialization) will remain on the simplex. Furthermore, the score to be maximized $\mathbf{x}^T\mathbf{A}\,\mathbf{x}$ is strictly increasing along any trajectory of the dynamics given in Equation 7. The final solution is a so-called evolutionary stable strategy (ESS), which is a stricter formulation of the well-known Nash equilibrium from game theory. For more details and convergence proofs of these dynamics see e. g. [22].

Interestingly, these dynamics have many similarities with spectral approaches for graph matching. The most popular spectral approach in this field is to apply power iteration [16] to the affinity matrix $\mathbf{A}$, a highly efficient method to find the largest eigenvector of a matrix. Power iteration, similar to replicator dynamics, is also an iterative method, which subsequently approaches the sought solution vector $\mathbf{x}^*$ starting from a random initialization. Thus, there are only two minor differences between power iteration and replicator dynamics. First, within each iteration both methods multiply the affinity matrix times the current vector, but in replicator dynamics additionally the transposed vector is multiplied times the multiplication result in an element-wise manner (see first part of Equation 7). Second, in spectral matching, optimization is performed over the set of vectors with specific properties, for example the Frobenius Norm has to be $N_2^{0.5}$ [9]. In contrast, in replicator dynamics optimization is done over the simplex Δ, representing normalized vectors whose entries sum up to 1. As also outlined e. g. in [9], a simplex normalization shows a more robust behavior.

Data / Method	*AR data set*			*IMM data set*		
	Unlearned	Learned	Impr.	Unlearned	Learned	Impr.
Spectral	88.0 ± 1.6	97.8 ± 0.2	$+9.8$	60.3 ± 3.0	79.7 ± 1.9	$+19.4$
Replicator	95.7 ± 0.7	98.7 ± 0.1	$+3.0$	69.3 ± 2.4	80.5 ± 1.4	$+11.2$

Table 1: Comparison of standard graph matching (*Unlearned*) and proposed method (*Learned*) on AR and IMM face data sets. Percentage of correct assignments and standard deviation for a 5-fold cross validation experiment is shown.

Data / Method	*AR data set*			*IMM data set*		
	Unlearned	Learned	Impr.	Unlearned	Learned	Impr.
All vs. all	88.1 ± 1.2	98.5 ± 0.3	$+10.4$	56.3 ± 1.2	81.3 ± 1.1	$+25.0$
Mean 0% out	95.6 ± 0.7	98.7 ± 0.1	$+3.1$	69.2 ± 2.4	80.5 ± 1.4	$+11.3$
Mean 100% out	82.2 ± 1.4	87.7 ± 0.4	$+5.5$	58.2 ± 2.9	64.6 ± 2.7	$+6.4$
Mean 200% out	71.5 ± 0.9	75.9 ± 0.6	$+4.4$	51.7 ± 2.3	54.4 ± 2.8	$+2.7$

Table 2: Results if using the mean model from training data as reference graph with increasing amount of outliers (*Mean* 0% to *Mean* 200%) and if using each training model as reference graph (*All vs. All*). Percentage of correct assignments and standard deviation for a 5-fold cross validation experiment using replicator dynamics are shown.

5 Experiments

As a first experiment we use our method for aligning a set of face images in Section 5.1, where we provide a direct comparison between standard graph matching and our proposed method. In Section 5.2 we further demonstrate the capability of our method for localizing category instances considering edge points as graph nodes. Analyzing simple unary potentials and our learned kernel functions allows localization of previously unseen instances.

5.1 Face Alignment

We evaluate our method on two publicly available face alignment data sets both containing ground truth correspondences: the IMM face data set [21] (240 images of 40 humans with 58 salient points) and the AR face data set [20] (550 face images with 22 salient points). We compare results obtained by standard graph matching (Equation 5 – *Unlearned*) to results when using our learned kernel functions (Equation 6 – *Learned*). All other parameters are set to the same values, to directly evaluate the achievable improvements.

As first experimental setup we used the mean point set obtained in the learning step as reference graph and compared it to all point sets of the remaining test data. As score we return the average percentage of correct assignments and the standard deviation for a 5-fold cross validation. Table 1 shows correct assignment rates for the IMM and the AR data set comparing power iteration to replicator dynamics. As can be seen replicator dynamics consistently outperforms the spectral power iteration method. Furthermore, independent of the optimization method, using our learned kernel functions clearly improves results by up to **20%** and the low standard deviation indicates good generalization properties.

To further demonstrate the influence of outliers and different reference models on the results, we additionally repeated the experiments, in the same style as described above, ex-

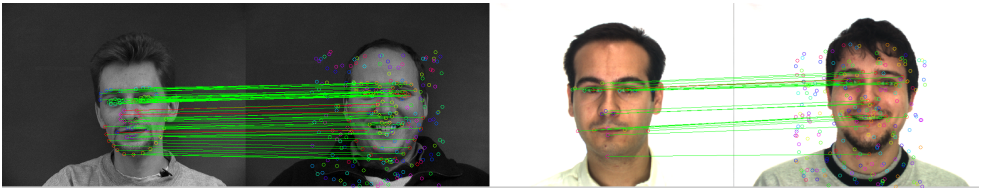

Figure 2: Graph Matching results with added outliers (100%). Left: IMM where 55 correct assignments are shown in green, whereas 3 wrong assignments are shown in red. Right: AR with 21 correct assignments from 22 (best viewed in color).

clusively using replicator dynamics with an increasing number of outliers from 0 to 200% (*Mean* 0% *outliers* to *Mean* 200% *outliers*), where 200% means that for 58 reference points we randomly added $2 \cdot 58 = 116$ outliers in the query image space in an uniform manner. Besides using the mean model as reference graph, we also made an exhaustive experiment using each point set of the training images as reference graph (*All vs. All*). Results for 5-fold cross validation are shown in Table 2. As can be seen our proposed, learned potentials again show significantly improved performance independent of the reference model and the amount of outliers. For example we achieve a 25% improvement in the all vs. all experiment, indicating that graph matching becomes much more robust against the choice of the reference model when using the learned potentials. Figure 2 illustrates two exemplary matching results for the IMM and the AR data set.

An implicit requirement of our method is the availability of labeled training samples, which might be cumbersome to obtain. Thus, it is important to analyze how many samples are required to improve matching quality and for this reason we made an additional experiment, analyzing performance when different numbers of training samples are available. We randomly selected a test set of 100 images from the AR data set and added 100% noise. From the remaining images, we randomly selected an increasing number of training images to learn our kernel functions. Figure 3 shows the percentage of correct assignments in dependence of the number of training samples over 100 runs of this experimental setup. As can be seen, actually only a few samples (more than 5) are sufficient to improve results.

5.2 Object Localization

In this experiment we demonstrate the applicability of our method for the task of object localization. This application is mainly motivated by [18], where a discriminative shape based object detection method was introduced. There, authors used simple unary potentials (edge normals, chord angles and lengths) in a pairwise graph matching setup and outperformed state-of-the-art appearance based object recognition methods.

It is also possible to use our learned kernel functions in such an object localization scenario. We again assume that we have given a set of ground truth correspondences between several objects from the same category. This labeled data is used to learn category specific kernel functions as outlined in Section 3. We then use our extended graph matching method to localize instances of the same category in test images in three subsequent steps. First, we sample equidistant points on edges detected by the Berkeley edge detector [19]. Second, we match the mean point set to the sampled point set exploiting the learned kernel functions as it is described in Section 4. Finally, we estimate a thin plate spline transformation from the correspondences and transform the coordinates of the mean model to the current image

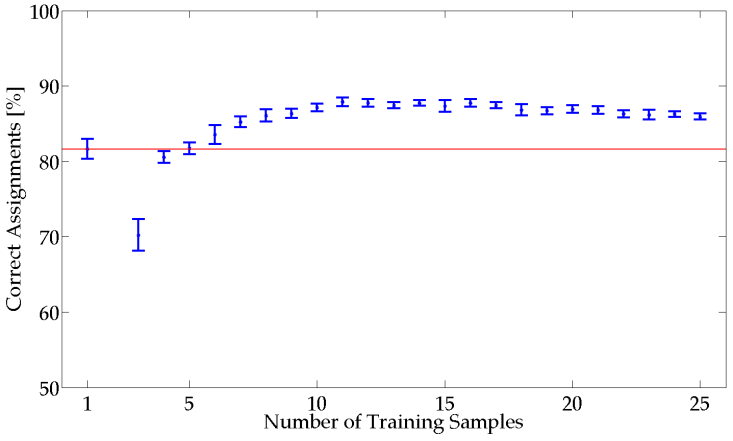

Figure 3: Matching results in dependence of number of training samples on AR data set with 100% added noise over 100 runs with randomly selected samples. Number of training samples = 1 represents the unlearned, standard graph matching model (81.7 ± 1.4 – red line). As can be seen, only a few training samples (> 5) are sufficient to improve results.

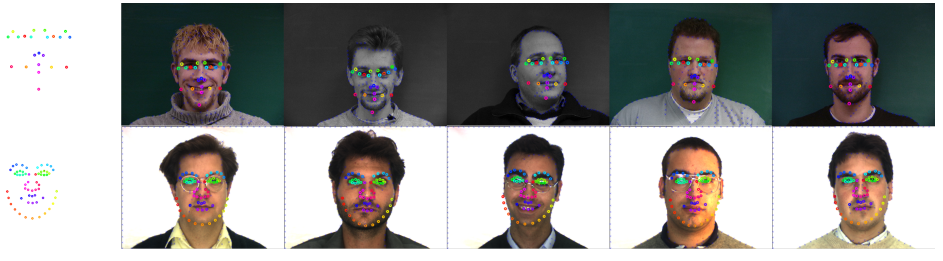

Figure 4: Face model fitting using learned pairwise potentials. First row shows learned AM model fitted to IMM images, whereas second row shows IMM model fitted to AR. Sampled edge points used in graph matching are shown as blue crosses, and highlighted points are thin plate spline (TPS) transformed model coordinates. TPS transformation is obtained by analyzing the correspondences returned from graph matching (best viewed in color).

for object localization. We used the ground truth provided by the AR and IMM data sets for learning face models. For modeling the cars we manually labeled 20 images. As unary potentials we always used angular differences between the edge point normals.

Figure 4 shows exemplary results for face model fitting, where in the first row we fitted the model obtained from the AR data set to the IMM images and in the second row the IMM model is fitted to AR images. As can be seen, despite the simplicity of the approach, reasonable localization results are provided. Of course, since the points shown are only thin plate spline transformed locations of the mean model, accuracy is limited.

Finally, Figure 5 shows results for localizing cars. For every example the obtained graph matching results and the thin plate spline transformed car model overlaid on the image are shown. As can be seen accurate localization results are achieved for different types of cars. Note, that these experiments should not demonstrate state-of-the-art localization performance, we only aim at outlining the potential of using our approach in such a scenario.

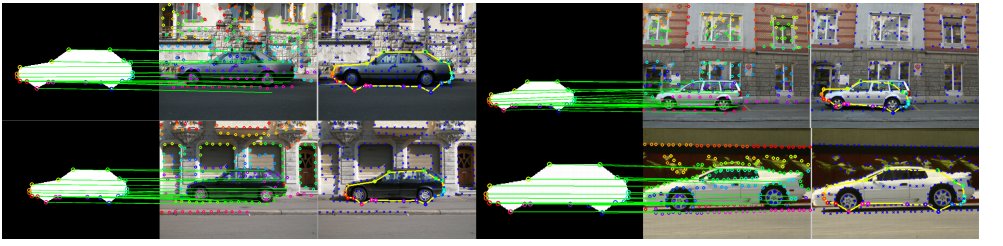

Figure 5: Object localization by graph matching to sampled edge points using learned kernel functions. Left and middle image show returned correspondences and the right image highlights the thin plate spline transformed model overlaid on the image. Blue crosses show sampled points on edges (best viewed in color).

6 Conclusion

This paper described an extension to standard graph matching methods for the case that the matching scenario, e. g. aligning faces in images, is known before. We showed how pairwise kernel functions are learned from annotated data sets and how these kernels can be used in a quadratic assignment setting to improve results. We also demonstrated that replicator dynamics are an easy and efficient method for approximating the NP hard graph matching solution, outperforming corresponding spectral methods. Experiments demonstrated the improved performance and outlined the potential for localizing category instances in images.

References

[1] A. Albarelli, S.R. Bulo, A. Torsello, and M. Pelillo. Matching as non-cooperative game. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2009.

[2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *Transactions of Pattern Analysis and Machine Intelligence (PAMI)*, 24 (4):509–522, 2002.

[3] A. C. Berg, T. L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondence. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[4] L. Caetano, L. Cheng, Q. Le, and A.J. Smola. Learning graph matching. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2007.

[5] T. Cootes, C. Taylor, D. Cooper, and J. Graham. Active Shape Models - Their Training and Application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[6] T. F. Cootes and C. J. Taylor. Statistical Models of Appearance for Computer Vision. Technical report, Imaging Science and Biomedical Engineering, University of Manchester, 2004.

[7] T. Cour, P. Srinivasan, and J. Shi. Balanced graph matching. In *Conference on Neural Information Processing Systems (NIPS)*, 2006.

[8] I. L. Dryden and K. V. Mardia. *Statistical Shape Analysis*. John Wiley & Sons, 1998.

[9] O. Duchenne, F. Bach, I. Kweon, and J. Ponce. A Tensor-Based Algorithm for High-Order Graph Matching. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2009.

[10] U. Gaur, Y. Zhu, B. Song, and A.Roy-Chowdhury. A "string of feature graphs" model for recognition of complex activities in natural video. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2011.

[11] S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. In *Transaction of Pattern Analysis and Machine Intelligence (PAMI)*, 1996.

[12] C. Goodall. Procrustes Methods in the Statistical Analysis of Shapes. *Journal Royal Statistical Society, Series B*, 53:285–339, 1991.

[13] H. Jiang and S.X. Yu. A linear solution to scale and rotation invariant object matching. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[14] J. Lee, M. Cho, and K.M. Lee. Hyper-graph matching via reweighted random walks. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[15] M. Leordeanu. *Spectral Graph Matching, Learning, and Inference for Computer Vision*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, July 2009.

[16] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2005.

[17] M. Leordeanu and M. Hebert. Unsupervised learning for graph matching. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[18] M. Leordeanu, M. Hebert, and R. Sukthankar. Beyond local appearance: Category recognition from pairwise interactions of simple features. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, 2007.

[19] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 26(5):530–549, 2004.

[20] A.M. Martinez and R. Benavente. The AR face database. Technical Report 24, CVC, June 1998.

[21] M. M. Nordstrøm, M. Larsen, J. Sierakowski, and M. B. Stegmann. The IMM face database - an annotated dataset of 240 face images. Technical report, Informatics and Mathematical Modelling, Technical University of Denmark, 2004.

[22] M. Pelillo. Replicator equations, maximal cliques, and graph isomorphism. *Neural Computation*, 11(8):1933–1955, 1999.

[23] C. Zach, M. Klopschitz, and M. Pollefeys. Disambiguating visual relations using loop constraints. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[24] R. Zass and A. Shashua. Probabilistic graph and hypergraph matching. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.