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 Vison
Graz University of Technology
Austria
[2] Ludwig Boltzmann Institute for
Clinical Forensic Imaging
Graz, Austria

Motivation Graph matching has become widely used in several computer vision applications including tracking, shape matching or object detection. Many different approaches are available for solving the NP-hard problem in an approximated manner, e. g. based on spectral techniques, probabilistic methods or graduated assignments. Surprisingly, only few papers focused on the important graph potentials themselves, which have a tremendous influence on the quality of the obtainable results. For example, it was shown in [1] that solving a linear assignment problem using well chosen potentials even improves over related state-of-the-art quadratic assignment solutions.

One important challenge of using powerful potentials in graph matching is their right parametrization, which is mostly done manually. Only a few papers focused on the problem of choosing the right parameters. Caetano et al. [1] showed how to learn optimal parameters for the features used in the potentials from manually labeled reference data sets and Leordeanu et al. [2] extended this idea to an unsupervised setting. 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, but instead of learning fixed parameters for the features used as done in [1, 2], we directly learn edge-specific kernel functions for each node pair, assuming that the setting of graph matching is a-priori known. Such a-priori knowledge is indeed available in several important computer vision applications like automated face alignment, model fitting and object localization.

Method 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 then defined by edge-specific kernel functions for every pair of nodes. Second, during testing, our method is an extension of standard graph matching formulated as quadratic assignment problem. As the main difference to standard graph matching solutions, we exploit the learned kernel functions for improving matching quality.

We define our kernel functions $\mathcal{K}_{ij}$ to relate an edge connecting points i and j in our reference graph (consisting of N_1 nodes) to an edge connecting points a and b in the query graph (N_2 nodes) by deriving statistics of the point location distributions within a labeled training set. Thus, 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. We register all labeled points of the training set to each other using Procrustes Analysis, which then allows to describe the spatial distribution of each point over the training set by a Gaussian as it is visualized in Figure 1.

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.

The goal of graph matching is to find a one-to-one mapping between two graphs, which is defined by a binary assignment vector $\mathbf{x}^* \in \mathbb{R}^{N_1 N_2}$, where $x^*_{ia} = 1$ if node i of the reference graph matches to node a of the query graph and $x^*_{ia} = 0$ otherwise and $\sum_i x^*_{ia} = 1, \sum_a x^*_{ia} = 1$. Such standard quadratic assignment problems (QAP) are solved by

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

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 and geometry with a pair of nodes in the query (a, b).

Similar to related methods we use shape context ($\mathbf{s}_i$) as local descriptor for each node i, but replace the standard analysis of the differences in edge lengths by our learned edge-specific kernel functions. For this reason, the affinity matrix entries are adapted to

$$A_{ia,jb} = exp - \left(\mathbf{w}_1^T |\mathbf{s}_i - \mathbf{s}_a| + \mathbf{w}_2^T |\mathbf{s}_j - \mathbf{s}_b| + \mathbf{w}_3^T\, \mathcal{K}_{ij}(ab)\right)\,, \tag{2}$$

where $\mathcal{K}_{ij}(ab)$ is the learned pairwise kernel function. Thus, in our setting deviations from the reference graph geometry are penalized depending on the location uncertainty as learned in the statistical shape model.

We relax the integer optimization of Equation 1 into the continuous domain and solve it using Replicator Dynamics [3], an evolutionary algorithm from the field of game theory. These dynamics iteratively update the assignment vector $\mathbf{x}$ using

$$x_i^{t+1} = x_i^t\, \frac{(\mathbf{A}\,\mathbf{x}^t)_i}{\mathbf{x}^{t\,T}\mathbf{A}\,\mathbf{x}^t}\,, \tag{3}$$

where $\mathbf{x}^t$ is the assignment vector at time t. As a necessary additional constraint $\mathbf{x}$ has to lie on the simplex $\left(\mathbf{x} \in \mathbb{R}^{N_1 N_2} : x_i \geq 0 \text{ and } \mathbf{1}^T\,\mathbf{x} = 1\right)$. Replicator dynamics return an optimal assignment vector $\mathbf{x}^*$, which is a local (!) maximum of the optimization problem shown in Equation 1.

Experiments In a first experiment we use our method to align a set of face images using the IMM and AR face data sets. We used the mean point set obtained from the training graphs (*Mean*) or each of them (*All*) as reference graph and compared it to all point sets of the remaining test data. Table 1 shows the average percentage of correct assignments, comparing our proposed, learned potentials to standard ones. As can be seen, using our learned kernel function clearly improves results by up to **25%**.

GM	*AR data set*			*IMM data set*		
	Orig.	*Learned*	*Impr.*	*Orig.*	*Learned*	*Impr.*
All	88.1	98.5	+10.5	56.3	81.3	+25.1
Mean	95.6	98.7	+3.1	69.2	80.5	+11.3

Table 1: Percentage of correct assignments for matching to the mean point model (*Mean*) or each model of the training data (*All*) using standard (*Orig.*) and our learned potentials (*Learned*).

More experiments, e. g. on evaluating the influence of the number of training samples on the matching quality and an application for feature point based localization of previously unseen category instances in images, are provided in the main paper.

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

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

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