Learning Similarities for Rigid and Non-Rigid Object Detection

Asako Kanezaki
The Univ. of Tokyo

Emanuele Rodolà
TU Munich

Daniel Cremers
TU Munich

Tatsuya Harada
The Univ. of Tokyo

Abstract

In this paper, we propose an optimization method for estimating the parameters that typically appear in graph-theoretical formulations of the matching problem for object detection. Although several methods have been proposed to optimize parameters for graph matching in a way to promote correct correspondences and to restrict wrong ones, our approach is novel in the sense that it aims at improving performance in the more general task of object detection. In our formulation, similarity functions are adjusted so as to increase the overall similarity among a reference model and the observed target, and at the same time reduce the similarity among reference and "non-target" objects. We evaluate the proposed method in two challenging scenarios, namely object detection using data captured with a Kinect sensor in a real environment, and intrinsic metric learning for deformable shapes, demonstrating substantial improvements in both settings.

1. Introduction

It has been an ultimate objective of computer vision to realize a system that can *see* the world as a human being does. These days, technology has enabled us to take advantage of rich visual information in our surroundings in the form of realistic 3D data (as captured, for instance, by consumer-level depth cameras); however, it remains an abstruse problem to make an intelligent system *see* the world and to let it know "what is where" in the real world.

In this paper, we propose an optimization method to design effective score functions for object detection tasks. For the detection step we make use of reference shape data of the target objects, and thereby obtain point-to-point *correspondences* between reference and real-world observations. Compared to conventional object detection methods which employ global features extracted from bounding boxes, this local approach is more robust to the pose variations and occlusions frequently occurring in a real environment. In order to reduce false local correspondences (mismatches), we raise the order of the problem and consider pairwise similarity terms. The resulting formulation takes the form of a graph matching problem between the graphs of the reference and observed shapes.

More formally, let (X, d_X) and (Y, d_Y) be two (compact) metric spaces with $X, Y \subset \mathrm{R}^m$, and let $C \subset X \times Y$ be a *correspondence set* between them. We formulate the matching problem as a L_p-regularized Quadratic Assignment Problem (QAP),

$$\max_{\boldsymbol{x}} \ \boldsymbol{x}^{\mathrm{T}} A \boldsymbol{x} \quad \text{s.t. } \|\boldsymbol{x}\|_p = 1 \,, \tag{1}$$

where $\boldsymbol{x} \in [0,1]^{|C|}$ represents a (weighted) cluster of matches and A is a $|C| \times |C|$ symmetric matrix representing pairwise similarity terms between matches. The problem above aims at maximizing the overall similarity among the two given metric spaces. The QAP formulation of the matching problem is rather common in the shape and graph matching communities, and in particular L_p relaxations to it have proved beneficial in a variety of settings. For instance, Leordeanu and Hebert [9] use $p = 2$ and Rodolà et al. [15] use $p = 1$. More recently, mixed norm combinations have been proposed [19, 17]. Matrix A can be regarded as the realization of some similarity function $\pi : C \times C \to [0, 1]$. Clearly, many definitions for π are possible depending on the specific problem at hand. For example, when dealing with deformable shapes we expect the similarity function to be as invariant as possible to non-rigid transformations of the data (*e.g.*, change of pose), and thus define π to take into account *intrinsic* shape quantities (such as geodesic distances) that do not depend on how the shape is embedded in Euclidean space.

To deal with an object detection task, we need to define a mechanism according to which the *score functions* are to be learned. These functions express how similar an observed object is to some target object. Here, we attempt to determine the geometric and visual properties that better characterize each object, together with their influence on the detection task. In particular, we expect these properties to depend on the *object class* to be matched. To this end, we propose a learning method that optimizes over a vector of weight parameters representing combination coefficients for *several* similarity functions.

2. Related Work

Arguably the most common method for object detection consists in the adoption of HOG descriptors [7] extracted from bounding boxes of the objects, followed by a learning step on their weights by means of a SVM. A recent, now de-facto standard approach for object detection is Deformable Part Models (DPM) [8], which also employs a combination of HOG and linear SVMs in order to determine the model parameters. Although DPM provides a flexible model for object detection, it is not sufficiently robust to handle strong posture variation or occlusions, as it assumes that the descriptor extracted from each bounding box does not change dramatically across several instances of the same object.

A second approach to object detection is given by matching-based methods. Extracting a set of correspondences among sets of features is a fundamental problem in object detection. In general, the correspondence set obtained by mere comparison of local descriptors includes numerous false corresponding points. Various methods considering the overall consistency of the solution have been proposed to eliminate these false matches, with derivations of the QAP taking the lion's share.

These formulations typically differ in the way the potentials (composing matrix A in Eq. (1)) are defined, suggesting that combinations of several similarity functions might lead to different results depending on how each similarity term is weighted relative to the others. In this view, there has recently been some interest in learning the *optimal* set of weights for the similarity functions [5, 10, 12, 14]. Caetano et al. [5] optimized these parameters by minimizing the Hamming loss between a ground-truth assignment vector $\boldsymbol{x}$ and an estimated assignment vector $\hat{\boldsymbol{x}}$. Leordeanu et al. [10] took a similar view and ran the optimization process based on a smoothed version of the objective function, leading to an increase of performance. While these methods require ground-truth correspondence sets, an unsupervised learning method [12] was recently proposed that makes use of a binarized assignment vector $\boldsymbol{x}$, obtained as a solution to a L_2-regularized QAP, in place of the ground-truth assignment vector. This approach, which adopts the estimated correspondence sets as "teaching signals", notably allows to achieve equivalent performance to the ground-truth case.

A common feature of the methods mentioned above is that they specifically attempt to improve matching performance by promoting correct matches, while at the same time restricting incorrect matches. Nevertheless, it is important to note that the obtained parameters do not necessarily lead to superior performance in object detection tasks. Although a few methods attempt indeed to "learn the graph matching" for the classification task [11, 1], they do not do so by directly minimizing the classification error. In [11], for instance, the parameters used for matching are learned independently of the object class (thus ignoring the difference in scores across all classes), given positive (correct) correspondences and negative (incorrect) correspondences in input. Brendel and Todorovic [1] learn the graph structures themselves rather than their matching parameters, whereas the similarity values of the nodes of given training graphs are fixed.

In our work, we focus on learning the similarity functions for pairwise potentials in graph matching. Intuitively, the similarity functions tell us how important each feature used in graph matching is to detect each target object. No prior information on the weight parameters of the similarity functions is given to the learning process. To detect objects, we obtain correspondence sets between points on reference shape data of target objects and observed shape data. Differently from the other methods, we place a high value on achieving proper similarity scores for each target object, rather than improving the accuracy of each correspondence set separately.

3. Method

Our objective is to learn the parameters of a score function $g : X \times Y \rightarrow \mathrm{R}$ representing the similarity value between an observed object X and a target object Y. Our approach is based on the adoption of an online learning method that, presuming target and non-target objects exist in the scene, updates the model parameters by observing training object samples one after another. Parameters are updated in such a way to make the score of the correct object higher than those of non-target objects. In the following, we describe how the training samples are obtained, the definition of the score function g, and the parameter learning step for the score function.

3.1. Training samples

In this paper, we consider generalized similarity functions π defined by the composition:

$$\pi(c_i, c_j) = \exp\left[-s\left(c_i, c_j\right)\right] , \tag{2}$$

where $s : C \times C \rightarrow \mathrm{R}$ is a (not necessarily positive) function expressing the degree of compatibility of two candidate matches c_i and c_j.

For the training process, we prepare a *reference* shape model R for each target object, together with a collection of observed shape data of the same object, which we denote by $\{O_i^{(pos)}\}_{i=1,\ldots,N}$, and observed shape data of different objects, which make up the set $\{O_j^{(neg)}\}_{j=1,\ldots,N'}$. The process then proceeds as follows. A set of K points are sampled from each reference shape R, and local descriptors computed at each point. Then, for each observation $O_i^{(pos)}$ and $O_j^{(neg)}$ we search the k_{nn} nearest neighbors (in descriptor space) to each sample point in R; by doing so, we obtain $k_{nn}K$ candidate matches for each pair of shapes

$(R, O_i^{(pos)})$ and $(R, O_j^{(neg)})$. Since in this step we are only looking at similarity of the descriptors, the correspondence sets $C_i^{(pos)}, C_j^{(neg)} \subset X \times Y$ obtained in this manner may certainly include wrong matches. These wrong correspondences are filtered out by solving problem (1); in particular, since density of the correspondence is not a concern at this point, we take the point of view of inlier selection and adopt the L_1-regularized relaxation of the QAP ($p = 1$) as proposed in [15], which allows to obtain an accurate (yet sparse) solution to the resulting QAP in an efficient manner. The solution vector $\mathbf{x} \in [0, 1]$ is then binarized in $\{0, 1\}$ by hard-thresholding. The result of this process is a collection of filtered correspondence sets between the model R and $O_i^{(pos)}$ for $i = 1, \ldots, N$, and between R and $O_j^{(neg)}$ for $j = 1, \ldots, N'$.

Note that the generality of the process allows the compatibility function s to be defined as desired. For rigid object detection tasks we adopt the Euclidean distance $d_E(a, b)$ of the pair $(a, b) \in X \times X$ and define

$$s_r((a, a'), (b, b')) \equiv |d_E(a, b) - d_E(a', b')|\,, \qquad (3)$$

where $(a, a'), (b, b') \in C$. Since rigid motions preserve Euclidean distances, we expect a correct correspondence to attain a value of zero under the function above. Similarly, for non-rigid object detection tasks we employ intrinsic (*i.e.*, isometry invariant) quantities. Namely, we consider the multi-scale diffusion (MD) metric [16] $d_M(a, b)$ and the commute-time (CT) metric [3] $d_C(a, b)$, to define

$$s_n((a, a'), (b, b')) \equiv \frac{s_M\left((a, a'), (b, b')\right) + s_C\left((a, a'), (b, b')\right)}{2}\,,$$

where s_M and s_C are defined as in (3), with the appropriate metrics.

3.2. Score function

In the specific case in which the composite function of Eq. (2) is directly replaced by the local measure of distortion $|d_X(a, b) - d_Y(a', b')|$, the quadratic form $\boldsymbol{x}^{\mathrm{T}} A \boldsymbol{x}$ encodes a notion of proximity between metric spaces X and Y, namely their Gromov-Wasserstein distance [16, 13]. In particular, the two shapes are isomorphic (*i.e.*, measure-preserving isometric) if their Gromov-Wasserstein distance equals zero. While in our current setting we replace the local distortion criterion with a *similarity* potential, it makes sense to regard the value attained by $\boldsymbol{x}^{\mathrm{T}} A \boldsymbol{x}$ for each correspondence set between R and each $O_i^{(pos)}$, $O_j^{(neg)}$ as the similarity of the corresponding underlying metric spaces. In particular, since we can only obtain a local optimum for each (relaxed) QAP, different pairs of shapes will have locally optimal correspondences of different sizes; we thus normalize the similarity values $\boldsymbol{x}^{\mathrm{T}} A \boldsymbol{x}$ by dividing them by the corresponding number of matches, which we denote by M. The baseline score $g_{<base>}$ between two objects can thus be defined as:

$$g_{<base>} \equiv \frac{2}{M(M-1)} \sum_{i=1}^{M} \sum_{j=i+1}^{M} \exp\left[-s\left(c_i, c_j\right)\right]. \qquad (4)$$

The function above, which uniformly integrates the pairwise similarity over all correspondences, can be improved by taking into account additional properties (*e.g.*, color) to help distinguishing and give an informed weighting of the correspondences. Alternatively, it is possible to integrate different similarity functions s with proper weights to define the total score function. The following subsections will describe the design of the proposed score functions for rigid and non-rigid object detection.

3.2.1 Rigid object detection (RGBD)

As noted in the previous sections, Eq. (3) is probably the most direct way to encode a similarity criterion between objects transforming in a rigid manner. However, with the recent surge in availability of consumer-level 3D scanning devices, there has been a growing interest in providing additional data together with the reconstructed geometry. Color information, when available, can be employed to drastically improve recognition results.

In Figure 1 we show a conceptual diagram of our approach for rigid object detection with RGBD data. Specifically, we operate in a quantized HSV space in which we discretize the hue value into k bins; we then define $\boldsymbol{h}_i \in [0, 1]^k$ to be an indicator vector for the point in the *reference* shape corresponding to the i-th candidate match, specifying to which bin this point belongs to. Given two matches i and j, we may then compute a matrix H for the corresponding hue values as $\boldsymbol{h}_i \boldsymbol{h}_j^{\mathrm{T}}$, and then compute a matrix H' where the non-diagonal element $H'_{mn} = H_{mn} + H_{nm} (m \neq n)$ and the diagonal element $H'_{nn} = H_{nn}$. Let $\boldsymbol{q}_{ij} \in [0, 1]^{k(k+1)/2}$ denote a vector which consists of the elements of the upper triangular portion (including diagonal components) of H'. Our objective here is to optimize the weight vector $\boldsymbol{w} \in [0, 1]^{k(k+1)/2}$ for $\boldsymbol{q}_{ij}$ with respect to each target object.

The final score function $g_r(\boldsymbol{w})$ for rigid object detection is defined as follows:

$$g_r(\boldsymbol{w}) \equiv \frac{2}{M(M-1)} \sum_{i=1}^{M} \sum_{j=i+1}^{M} \left(1 - \exp\left[\frac{-\alpha \cdot \boldsymbol{w} \cdot \boldsymbol{q}_{ij}}{s_r(c_i, c_j) + \epsilon}\right]\right)\,, \qquad (5)$$

where $\alpha > 0$ controls the shape of the exponential function, and ϵ is a small number preventing the denominator from being 0. In our experiments, we set $\alpha = 10^{-3}$ and $\epsilon = 10^{-20}$.

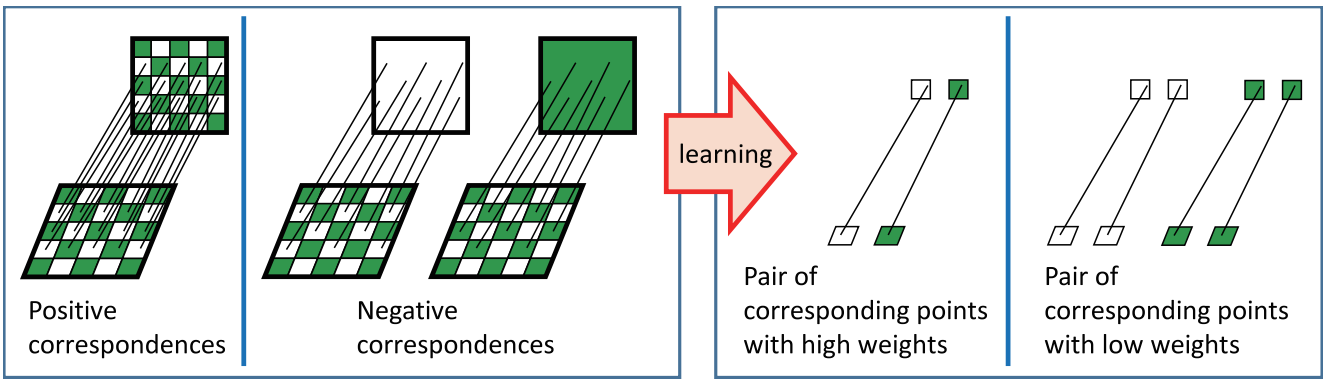

Figure 1. Conceptual diagram of the proposed learning method in a RGBD object detection scenario. Our method learns the weights for a pair of matches distinguished by color so that the total score of a correct correspondence set is higher than a wrong one. In this example, there are three different types of corresponding pairs: white-green, white-white, and green-green.

3.2.2 Non-rigid object detection

A common problem in the metric approach to matching is represented by the appropriate choice of a metric function that be invariant to a given class of deformations [13]. For example, geodesic distances are invariant to nearly-isometric deformations but are extremely sensitive to topological changes in the mesh, whereas commute-time metrics [3] are more robust to topology and global scale changes but less accurate on a local scale. In this set of experiments, we are interested in learning the best choice for an intrinsic metric (or combinations thereof) given different types of deformations of a shape. In particular, we consider two such distance functions in the definition of pairwise similarity: the multi-scale diffusion (MD) metric [16] and the commute-time (CT) metric.

Letting $\boldsymbol{w} \equiv [w_M \; w_C]^{\mathrm{T}} \in \mathrm{R}^2$ be a vector of weights and $\boldsymbol{s}(c_i, c_j) \equiv [s_M(c_i, c_j) \; s_C(c_i, c_j)]^{\mathrm{T}} \in \mathrm{R}^2$, we define the score function $g_n(\boldsymbol{w})$ for non-rigid object detection as:

$$g_n(\boldsymbol{w}) \equiv \frac{2}{M(M-1)} \sum_{i=1}^{M} \sum_{j=i+1}^{M} \exp[-\boldsymbol{w} \cdot \boldsymbol{s}(c_i, c_j)] \; . \quad (6)$$

3.3. Learning of parameters

The method we propose in this subsection allows to obtain the optimal $\boldsymbol{w}$ (appearing in Eqs. (5) and (6)) by computing a separating hyperplane on the training samples. This is similar in spirit to conventional methods such as SVM, in that we optimize $\boldsymbol{w}$ so that the value of $g(\boldsymbol{w})$ for a positive sample is high and the value attained by a negative sample is low. We do so by minimizing a quantity called *hinge loss* (Eq. (8) below), which represents the penalty incurred by training samples for being within the margin of the separating hyperplane.

Within this framework, the score function that outputs how similar an observed object is to the target object must range from $-\infty$ to $+\infty$. Therefore, we define the score function f used for training as follows:

$$\begin{aligned} f(\boldsymbol{w}, b) &\equiv \mathrm{logit}\,(g(\boldsymbol{w})) + b, \\ &= \log(g(\boldsymbol{w})) - \log(1 - g(\boldsymbol{w})) + b, \end{aligned} \quad (7)$$

where b is an offset value that is optimized together with $\boldsymbol{w}$. Letting the label of a correct (positive) set of correspondences be $y = 1$ and the label of a wrong (negative) set of correspondences be $y = -1$, the hinge loss is defined as

$$l(\boldsymbol{w}, b; (f, y)) = \begin{cases} 0 & yf(\boldsymbol{w}, b) \geq 1\,, \\ 1 - yf(\boldsymbol{w}, b) & \text{otherwise}\,. \end{cases} \quad (8)$$

We initialize $\boldsymbol{w}$ as $\boldsymbol{w}_0 = (\epsilon', \ldots, \epsilon'), \epsilon' \sim 0$, and b as $b_0 = 0$. Each time a training correspondence set is observed, these two parameters are updated accordingly. Letting $\boldsymbol{w}_t$ and b_t be the parameters obtained after the t-th update, the solutions at successive time steps are obtained by solving the projection problem

$$\{\boldsymbol{w}_{t+1}, b_{t+1}\} = \arg\min_{\boldsymbol{w}, b} \frac{1}{2}\left(\|\boldsymbol{w} - \boldsymbol{w}_t\|^2 + \|b - b_t\|^2\right) \quad (9)$$

$$\text{s.t. } l(\boldsymbol{w}, b; (f_t, y_t)) = 0. \quad (10)$$

This problem can be solved in closed-form. In particular, when $y_t f_t(\boldsymbol{w}_t, b_t) \geq 1$, we have the steady states $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t$ and $b_{t+1} = b_t$. Therefore, we can just consider the case in which $y_t f_t(\boldsymbol{w}_t, b_t) < 1$. In this case, the Lagrangian takes the form:

$$\begin{aligned} L(\boldsymbol{w}_t, b_t, \lambda) = \frac{1}{2}\|\boldsymbol{w}_{t+1} - \boldsymbol{w}_t\|^2 + \frac{1}{2}\|b_{t+1} - b_t\|^2 \\ +\lambda(1 - y_t f_t(\boldsymbol{w}_t, b_t)). \end{aligned} \quad (11)$$

Differentiating with respect to $\boldsymbol{w}_t$ and b_t and setting the derivatives to zero provides the following:

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \lambda y_t \frac{\partial f_t(\boldsymbol{w}_t, b_t)}{\partial \boldsymbol{w}_t}, \quad (12)$$

$$b_{t+1} = b_t - \lambda y_t. \quad (13)$$

Therein, $\frac{\partial}{\partial b_t} f_t(\boldsymbol{w}_t, b_t) = 1$. Plugging the above back into (11) yields

$$L(\lambda) = \frac{1}{2}\lambda^2 \left\| \frac{\partial f_t(\boldsymbol{w}_t, b_t)}{\partial \boldsymbol{w}_t} \right\|^2 + \frac{1}{2}\lambda^2 + \lambda(1 - y_t f_t(\boldsymbol{w}_t, b_t)). \tag{14}$$

Taking now the derivative of $L(\lambda)$ with respect to λ and setting it to zero gives us the following closed-form solution for the optimal λ:

$$\lambda = \frac{y_t f_t(\boldsymbol{w}_t, b_t) - 1}{\left\| \frac{\partial f_t(\boldsymbol{w}_t, b_t)}{\partial w_t} \right\|^2 + 1}. \tag{15}$$

Finally, $\boldsymbol{w}_{t+1}$ and b_{t+1} are obtained by the iterative equations:

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \frac{(y_t - f_t(\boldsymbol{w}_t, b_t))}{\left\| \frac{\partial f_t(\boldsymbol{w}_t, b_t)}{\partial \boldsymbol{w}_t} \right\|^2 + 1} \cdot \frac{\partial f_t(\boldsymbol{w}_t, b_t)}{\partial \boldsymbol{w}_t}, \tag{16}$$

$$b_{t+1} = b_t + \frac{(y_t - f_t(\boldsymbol{w}_t, b_t))}{\left\| \frac{\partial f_t(\boldsymbol{w}_t, b_t)}{\partial \boldsymbol{w}_t} \right\|^2 + 1}, \tag{17}$$

where the gradient of f is computed as:

$$\begin{aligned} \frac{\partial f(\boldsymbol{w}, b)}{\partial \boldsymbol{w}} &= \frac{\partial}{\partial \boldsymbol{w}} \left(\log(g(\boldsymbol{w})) - \log(1 - g(\boldsymbol{w})) \right) \\ &= \frac{1}{g(\boldsymbol{w})(1 - g(\boldsymbol{w}))} \frac{\partial g(\boldsymbol{w})}{\partial \boldsymbol{w}}. \end{aligned} \tag{18}$$

Note that the gradient of $g_r(\boldsymbol{w})$ is given by

$$\frac{2}{M(M-1)} \sum_{i=1}^{M} \sum_{j=i+1}^{M} \left(\frac{\alpha \cdot \boldsymbol{q}_{ij}}{s_E(c_i, c_j) + \epsilon} \cdot \exp\left[\frac{-\alpha \cdot \boldsymbol{w} \cdot \boldsymbol{q}_{ij}}{s_E(c_i, c_j) + \epsilon} \right] \right), \tag{19}$$

whereas the gradient of $g_n(\boldsymbol{w})$ becomes

$$-\frac{2}{M(M-1)} \sum_{i=1}^{M} \sum_{j=i+1}^{M} \boldsymbol{s}(c_i, c_j) \cdot \exp\left[-\boldsymbol{w} \cdot \boldsymbol{s}(c_i, c_j) \right]. \tag{20}$$

The derivation of the gradient of $g_r(\boldsymbol{w})$ and $g_n(\boldsymbol{w})$ are omitted for space reasons.

Discussion

It is particularly interesting to note that the proposed update rule for the parameters shares a connection with the Passive–Aggressive (PA) [6] online learning method of linear classifiers. Letting $\boldsymbol{x}$ be the descriptor of a training sample, PA computes the gradient of the score function $f(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x} + b$ to minimize the hinge loss, and it updates the parameters with the constraint of minimizing the L_2 distance from the current parameters. The algorithm we employ can be regarded as an extension to PA, obtained by replacing the score function $f(\boldsymbol{x})$ with Eq. (7). Note that our approach is more general than PA as we allow any differentiable score function to be adopted, whereas only linear scores can be employed with the PA method.

4. Results

4.1. Rigid object detection (RGBD)

Our first experiment is aimed at evaluating the proposed method in a rigid setting, using data captured with a Kinect sensor in a real-world environment. The captured data consists of 3D point clouds, where each point is endowed with a color attribute. The training set is composed of 10 target objects, each coming with a reference model and nine observations from as many view points (see Fig. 2). Negative samples for the non-target objects were prepared by capturing 70 scenes containing none of the target objects, and then by attempting to match each target model with these scenes (see Fig. 3 (a)). We search the $k_{nn} = 5$ nearest neighbors in RGB space[1] to 2,000 sample points in a reference model to obtain the candidate correspondence sets. For comparison, we extracted SIFT keypoints from color images and performed brute-force matching, *i.e.*, nearest neighbors in descriptor space. Then we solved the QAP problem via [15] to obtain final (sparse) correspondence sets. We set the quantization number of hue values to $k = 3$. Experiments with other values for k led to substantially similar results. We

Figure 2. Target objects used in the training dataset for similarity weight learning. Reference models of the target objects are shown in the leftmost column.

[1] Depth information is only used to obtain 3D coordinates of each point.

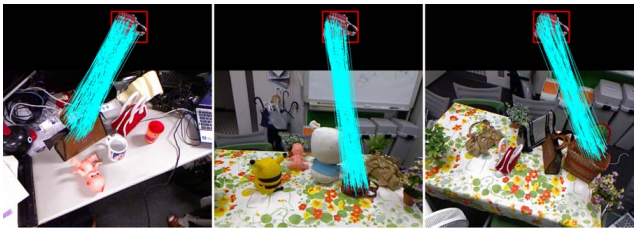

Figure 3. Exemplar correspondences between the reference model of the target object #1 and observed objects. (a) Positive correspondences in training data. (b) Correct correspondences in testing data. (c) Wrong correspondences in testing data.

terminate the learning process if the sum of the hinge loss becomes sufficiently small before $\gamma = 100$ iterations.

Quantitative evaluation was performed by capturing 120 scenes in a different environment and then computing the correspondence sets of all target objects on each of them (see Fig. 3 (a) and Fig. 3 (b)). The Precision-Recall curves and average precision values are shown in Fig. 4. The blue lines represent the results with SIFT keypoints and the baseline score function (Eq. (4)), the green lines represent the results with RGB nearest neighbor search and the baseline score function (Eq. (4)), and the red lines represent the results with RGB nearest neighbor search and the proposed score function (Eq. (5)). SIFT keypoints do not bring any clear advantage except for target objects 1 and 5, which contain textured planes. RGB nearest neighbor search with the proposed score function outperformed the baseline score in all the cases except for target object 4. The average values of precision are 0.13 with SIFT keypoints, 0.27 with RGB nearest search and the baseline score function, and 0.31 with RGB nearest search and the proposed score function.

4.2. Non-rigid object detection

In the second set of experiments we evaluate the improvements gained by adopting the proposed learning method in a non-rigid matching scenario. This setting is considerably more challenging than the previous case as the shapes are allowed to undergo non-rigid deformations. Recent attempts at introducing domain knowledge into this family of problems include [18], where the authors trained a random forest with an intrinsic shape descriptor to directly estimate dense correspondences between *complete* (*i.e.*, no partiality is allowed), previously unseen shapes. Differently from [18], in this section we demonstrate the applicability of our approach to learn an *optimal weighting* of metric functions for each class of shapes; the learned weights can then be employed within a QAP formulation to match partial, deformable shapes as in [16].

For this set of experiments we make use of the

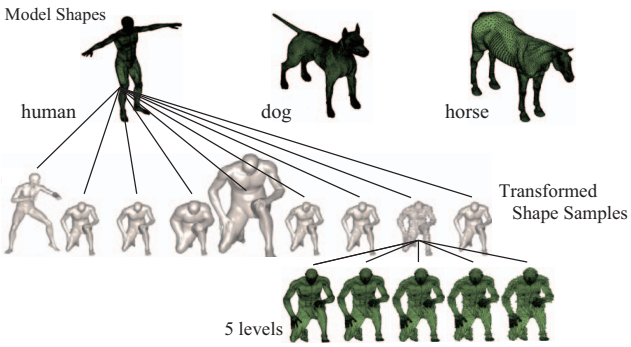

Figure 5. Datasets used in the non-rigid recognition experiments.

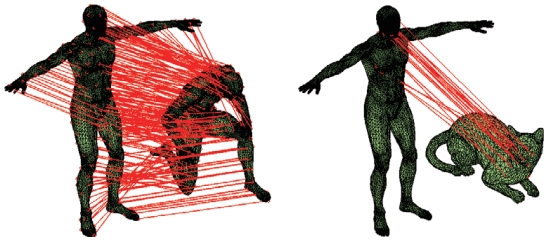

Figure 6. Examples of positive (left) and negative (right) correspondence sets.

SHREC'10 correspondence benchmark [2]. This dataset consists of three classes, namely "human", "dog", and "horse"; each class consists of one reference shape model and several data shapes transformed with nine different deformations (see Fig. 5), each coming in 5 intensities. We used one class among "human", "dog", and "horse" as the target class and used the samples in the other classes as negative samples. In each experiment, we used the samples of one type among all the 9 types of deformation.

In order to keep the problem more tractable, each shape was sampled at 200 points via Farthest Point Sampling (FPS) [3] using the extrinsic Euclidean metric (this choice is more robust to topology and partiality deformations than using an intrinsic metric). We applied FPS 10 times per shape starting from different seeds, obtaining 10 point sets per shape, and for each point in the sample sets we computed an intrinsic local descriptor, namely its Scale-Invariant Heat Kernel Signature (SI-HKS) [4]. The "positive" set of correspondences was formed by manually selecting 200 ground-truth pairs among each deformed shape $O_i^{(pos)}$ and the corresponding model R. The "negative" sets (*i.e.*, sets of matches between a deformed shape from one class and the reference model from another class) were formed by seeking the 5 nearest points in descriptor space for all the 200 sampled points on each reference shape, thus obtaining 1,000 candidate matches per set. Exemplary positive and negative correspondence sets are shown in Fig. 6.

The "human", "dog", and "horse" shapes were respec-

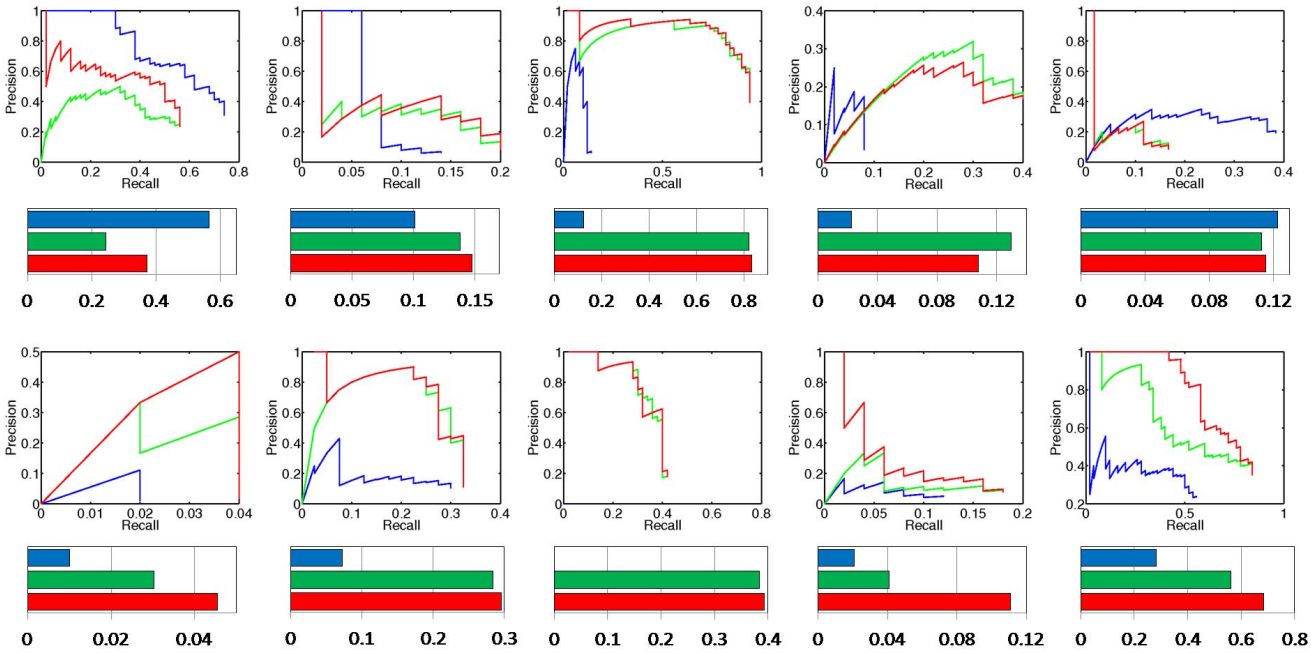

Figure 4. Precision-Recall curves and average precision values (below each Precision-Recall curve). The results of target objects from No. 1 to No. 10 are shown from the top left to the bottom right. The blue lines represent the results with SIFT keypoints and the baseline score function (Eq. (4)), the green lines represent the results with RGB nearest neighbor search and the baseline score function (Eq. (4)), and the red lines represent the results with RGB nearest neighbor search and the proposed score function (Eq. (5)).

tively used as a target object, and shapes *not* belonging to the target class were used as the negative samples. We then ran an experiment for each type of deformation. Specifically, for each experiment we constructed the training set by randomly selecting one positive and negative samples out of the 10 sets of FPS samples per shape, and we did this at deformation strengths 1, 3, and 5 (for a total of 3 positive samples and 6 negative samples in the training set). The test set was then formed by selecting all 10 sets of FPS samples per shape; we did this at the remaining deformation strengths 2 and 4, for a total of $2 \times 10 = 20$ positive samples and $2 \times 2 \times 10 = 40$ negative samples in the test set. We set the maximum number of learning iterations to $\gamma = 10,000$.

The resulting average precision values are presented in Table 1. From left to right we report the average precision values obtained by using MD only ("MD"), CT only ("CT"), uniform weights $\{0.5, 0.5\}$ ("baseline"), and learned weights ("learned"). Average precision over the whole dataset is reported in the last row; the proposed method ("learned") gave the best overall results when compared with "MD", "CT", and "baseline" alone.

Figure 7 shows the learned weights for similarity functions (MD and CT) in the deformation class "scale" and "shotnoise". Note that in the case of "scale" samples, the average precision values obtained with the CT metric are higher than those obtained with MD; likewise, the learned weight for the CT term is higher than MD. This is easily explained since CT is a fully scale-invariant metric whereas MD is only invariant to limited scale ranges. Similarly, the learned weight for the MD term is higher than CT in the case of "shotnoise" samples, whereas the average precision values obtained with the MD metric are higher than those obtained with CT. This implies, in particular, that a proper selection of the metrics could be achieved when the "scale" samples and "shotnoise" samples are used.

5. Conclusion

In this paper we proposed an optimization method for estimating the parameters that typically appear in graph-theoretical formulations of the matching problem. In particular, we restricted our attention to the object detection scenario. We formulated our method in an online learning framework, and evaluated the approach on two challenging problems, namely object detection of color 3D point clouds in a real environment, and intrinsic metric learning for deformable 3D shapes. The learning process improved the performance of object detection in both the considered scenarios. Our method can be easily extended and accommodated by considering different definitions of similarity. In particular, considering higher-order potentials and hypergraph matching scenarios are important future directions of research.

Table 1. Average precision in a non-rigid object detection task.

Deform.	Target	Average Precision MD	CT	baseline	learned
holes	human	1.00	0.94	1.00	1.00
holes	dog	0.69	0.94	0.75	1.00
holes	horse	1.00	0.96	1.00	1.00
isometry	human	1.00	1.00	1.00	1.00
isometry	dog	1.00	1.00	1.00	1.00
isometry	horse	1.00	1.00	1.00	1.00
microholes	human	1.00	1.00	1.00	1.00
microholes	dog	1.00	1.00	1.00	1.00
microholes	horse	1.00	1.00	1.00	1.00
noise	human	0.33	0.82	0.33	1.00
noise	dog	0.33	0.82	0.33	0.75
noise	horse	0.85	1.00	0.85	1.00
localscale	human	0.33	0.69	0.33	1.00
localscale	dog	0.33	0.52	0.36	1.00
localscale	horse	1.00	1.00	1.00	1.00
topology	human	0.33	0.55	0.34	1.00
topology	dog	0.68	1.00	0.82	1.00
topology	horse	1.00	1.00	1.00	1.00
sampling	human	1.00	1.00	1.00	1.00
sampling	dog	0.66	1.00	0.78	1.00
sampling	horse	1.00	1.00	1.00	1.00
scale	human	0.33	1.00	0.33	0.74
scale	dog	0.33	1.00	0.33	1.00
scale	horse	0.33	1.00	0.33	1.00
shotnoise	human	1.00	0.77	1.00	0.96
shotnoise	dog	0.85	0.75	0.77	1.00
shotnoise	horse	1.00	1.00	1.00	1.00
Average		0.76	0.92	0.77	0.98

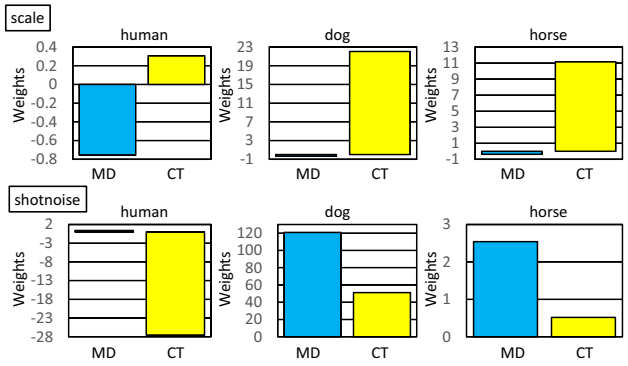

Figure 7. Learned weights for similarity functions (MD and CT) in the deformation class "scale" (top) and "shotnoise" (bottom).

References

[1] W. Brendel and S. Todorovic. Learning spatiotemporal graphs of human activities. In *Proc. IEEE ICCV*, 2011. 2

[2] A. M. Bronstein, M. M. Bronstein, U. Castellani, A. Dubrovina, et al. Shrec 2010: robust correspondence benchmark. In *Proc. EUROGRAPHICS Workshop on 3D Object Retrieval (3DOR)*, 2010. 6

[3] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. *Numerical geometry of non-rigid shapes*. Springer, 2008. 3, 4, 6

[4] M. M. Bronstein and I. Kokkinos. Scale-invariant heat kernel signatures for non-rigid shape recognition. In *Proc. IEEE CVPR*, 2010. 6

[5] T. S. Caetano, J. J. McAuley, L. Cheng, Q. V. Le, and A. J. Smola. Learning graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 31(6):1048–1058, 2009. 2

[6] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. *Machine Learning Research*, 7:551–585, 2006. 5

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. IEEE CVPR*, 2005. 2

[8] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32(9), 2010. 2

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

[10] M. Leordeanu and M. Hebert. Smoothing-based optimization. In *Proc. IEEE CVPR*, 2008. 2

[11] M. Leordeanu, M. Hebert, and R. Sukthankar. Beyond local appearance: Category recognition from pairwise interactions of simple features. In *Proc. IEEE CVPR*, 2007. 2

[12] M. Leordeanu, R. Sukthankar, and M. Hebert. Unsupervised learning for graph matching. *International Journal of Computer Vision*, 96(1):28–45, 2012. 2

[13] F. Mémoli. Gromov-Wasserstein distances and the metric approach to object matching. *Found. Comput. Math.*, 11:417–487, 2011. 3, 4

[14] D. Pachauri, M. Collins, V. Singh, and R. Kondor. Incorporating domain knowledge in matching problems via harmonic analysis. In *Proc. ICML*, 2012. 2

[15] E. Rodolà, A. Albarelli, F. Bergamasco, and A. Torsello. A scale independent selection process for 3d object recognition in cluttered scenes. *International Journal of Computer Vision (IJCV) - Special Issue on 3D Imaging, Processing and Modeling Techniques*, 19, 2012. 1, 3, 5

[16] E. Rodolà, A. M. Bronstein, A. Albarelli, F. Bergamasco, and A. Torsello. A game-theoretic approach to deformable shape matching. In *Proc. IEEE CVPR*, 2012. 3, 4, 6

[17] E. Rodolà, T. Harada, Y. Kuniyoshi, and D. Cremers. Efficient shape matching using vector extrapolation. In *Proc. BMVC*, 2013. 1

[18] E. Rodolà, S. Rota Bulò, T. Windheuser, M. Vestner, and D. Cremers. Dense non-rigid shape correspondence using random forests. In *Proc. IEEE CVPR*, 2014. 6

[19] E. Rodolà, A. Torsello, T. Harada, Y. Kuniyoshi, and D. Cremers. Elastic net constraints for shape matching. In *Proc. IEEE ICCV*, 2013. 1