Beyond Local Appearance: Category Recognition from Pairwise Interactions of Simple Features

Marius Leordeanu
Carnegie Mellon University

Martial Hebert
Carnegie Mellon University

Rahul Sukthankar
Carnegie Mellon University

Follow this and additional works at: http://repository.cmu.edu/robotics

Part of the Robotics Commons

Published In
Proceedings of CVPR.
Beyond Local Appearance: Category Recognition from Pairwise Interactions of Simple Features

Marius Leordeanu\textsuperscript{1} Martial Hebert\textsuperscript{1} Rahul Sukthankar\textsuperscript{2,1}

mleordea@andrew.cmu.edu hebert@ri.cmu.edu rahuls@cs.cmu.edu
\textsuperscript{1}Carnegie Mellon University \textsuperscript{2}Intel Research Pittsburgh

Abstract

We present a discriminative shape-based algorithm for object category localization and recognition. Our method learns object models in a weakly-supervised fashion, without requiring the specification of object locations nor pixel masks in the training data. We represent object models as cliques of fully-interconnected parts, exploiting only the pairwise geometric relationships between them. The use of pairwise relationships enables our algorithm to successfully overcome several problems that are common to previously-published methods. Even though our algorithm can easily incorporate local appearance information from richer features, we purposefully do not use them in order to demonstrate that simple geometric relationships can match (or exceed) the performance of state-of-the-art object recognition algorithms.

1. Introduction

Object category recognition is very challenging because there is no formal definition of what constitutes an object category. While people largely agree on common, useful categories, it is still not clear which are the objects’ features that help us group them into such categories. Our proposed approach is based on the observation that for a wide variety of common object categories, shape matters more than local appearance. For example, it is the shape, not the color or texture, that enables a plane to fly, an animal to run or a human hand to manipulate objects. Many categories are defined by their function and it is typically the case that function dictates an object’s shape rather than its low level surface appearance. In this paper we represent these object category models as cliques of very simple features (sparse points and their normals), and focus only on the pairwise geometric relationships between them.

The importance of using pairwise geometric constraints between simple shape features in object recognition was identified early on based on the observation that accidental alignments between oriented edges are rare and that they can be effective in pruning the search for correspondences between sets of model and input features. This is consistent with research in cognitive science hypothesizing that human category recognition is based on pairwise relationships between object parts [13]. In computer vision, this has led to the development of several promising approaches, such as interpretation trees, or alignment and verification. While successful, these techniques were limited by combinatorial issues (see [12] for a survey of early work in this area). All of these techniques were limited by their reliance on an explicit, parametric, representation of the transformation between model and input data (e.g., rigid or affine). To avoid the need for combinatorial search, global techniques that integrate pairwise geometric constraints into a global energy were also proposed (e.g., using MRF models [18]). Later approaches relaxed all of these limitations by using a non-parametric model of deformation between model data and input data. This is the case of the work of Berg \textit{et al.} [2], which uses second-order geometric relationships between features in a global optimization framework.

Our approach combines many of these ideas from early work and more recent contributions. We explicitly represent the pairwise relations between contour elements by using a large set of parameters, in contrast to the pairwise distance and angle used in previous work [2]. We show that we can achieve good recognition performance, without using local appearance features (such as those used by Berg \textit{et al.}), validating earlier observations [12] that geometric constraints alone can prune most correspondences.

We also formulate the search for consistent correspondences as a global energy optimization, with two key differences from earlier work. First, we use an efficient algorithm for finding an optimal discrete set of consistent correspondences [17], which enable us to use a large number of features. Second, we learn the parameters used in representing the geometric constraints in order to better capture the space of pairwise deformations.

Our matching-based approach builds a single abstract shape model that incorporates common features from multiple training images, as well as distinctive characteristics unique to each training image. Applying this in a semi-
supervised learning context enables us to automatically discover and remove the irrelevant clutter from training images, keeping only the features that are indeed useful for recognition. This opens up a more compact representation, which reduces the computational and memory cost, and improves generalization.

An interesting alternative to geometric matches is to classify directly based on statistics of the spatial distribution of features [10, 24, 26]. In contrast to these fully-supervised approaches, we show that simple geometric relationships can be successful in a semi-supervised setting.

The use of second-order geometric relationships enables our algorithm to successfully overcome problems often encountered by other methods:

1. During training, the algorithm is translation invariant, robust to clutter, and does not require aligned training images. This is in contrast with previous work such as [2, 11, 22, 23].
2. We efficiently learn models consisting of hundreds of fully interconnected parts (capturing both short- and long-range dependencies). Most previous work handles models only up to 30 sparsely interconnected parts, such as the star-shaped [6, 9], k-fan, [7] or hierarchical models [4, 8]. There has been work [5] handling hundreds of parts, but such models are not translation-invariant and each object part is connected only to its k-nearest neighbors.
3. We select features based on how well they work together as a team rather than individually (as is the case in [7, 23]). This gives us a larger pool of useful features, which are collectively discriminative, if not necessarily on an individual basis.

We formulate the problem as follows: given a set of negative images (not containing the object) and weakly-labeled positive images (containing an object of a given category somewhere in the image), the task is to learn a category shape model (Section 3) that can be used both for the localization and recognition of objects from the same category in novel images (Section 2). This problem is challenging because we do not have any prior knowledge about the object’s location in the training images. Also these images can contain a substantial amount of clutter that is irrelevant to the category that we want to model. All that is known at training time is that the object is present somewhere in the positive training images and absent in the negative ones.

## 2. The Category Shape Model

The category shape model is a graph of interconnected parts (nodes) whose geometric interactions are modeled using pairwise potentials inspired by Conditional Random Fields [14, 15]. The nodes in this graph are fully interconnected (they form a clique) with a single exception: there is no link between two parts that have not occurred together in the training images. These model parts have a very simple representation: they consist of sparse, abstract points together with their associated normals. Of course we could add local information in addition to their normals, but our objective is to assess the power of the geometric relationships between these simple features.

We represent the pairwise relationships by an overcomplete set of parameters. The parts as well as their geometric relationships are learned from actual boundary fragments extracted from training images. The model is a graph whose edges are abstract pairwise geometric relationships. It is a compact representation of a category shape, achieved by sharing generic geometric configurations common to most objects from a category and also by integrating specific configurations that capture different aspects or poses (Figure 1).

This section first describes the structure of the model, including, in particular, the details of the pairwise geometric relations, by showing how model parts are matched with features from an input image, which we term object localization. Then, given correspondences between model parts and image features, we show how to estimate the likelihood that the input image will contain the desired category, i.e., the recognition problem. Here we assume that a model has been learned from training data, the details of the learning procedure are described in Section 3.

### 2.1. Object Localization

We define the object localization problem as finding which feature in the image best matches each model part. We formulate it as a quadratic assignment problem (QAP), which incorporates the second-order relationships. The matching score $E$ is written as:

$$E_x = \sum_{ia,jb} x_{ia} x_{ib} G_{ia,jb}$$

Figure 1. The model is a graph whose edges are abstract pairwise geometric relationships. It integrates the geometric strategies common to most objects from a category as well as more specific configurations that capture different poses and aspects.
using logistic classifiers:

g in the context of CRFs [14] we model these potentials
g for geometric matching and recognition because it makes form an over-complete set of values, some highly dependent
depend only on those model parts that indeed belong to the
We consider that \( P(C|\mathbf{x}^*, D) \) should be a function of several factors. First, it should depend on the quality of the match (localization) as given by the pairwise potentials \( G_{ia,jb} \) for the optimal solution \( \mathbf{x}^* \). Second, it should depend only on those model parts that indeed belong to the category of interest and are discriminative against the negative class. It is not obvious which are those parts, since we learn the model in a semi-supervised fashion. For this reason we introduce the relevance parameter \( r_i \) for each part \( i \) (Section 3 explains how this is learned), which has a high value if part \( i \) is discriminative against the background, and low value otherwise. We approximate the posterior with the following logistic classifier:

\[
S(G_o, \mathbf{r}) = \frac{1}{1 + \exp(-q_0 - \mathbf{q}_r^T \mathbf{G}_o \mathbf{r})}.
\]  
(4)

The matrix \( G_o(i,j) = G_{ia,jb} \) contains all the pairwise potentials for the optimal localization \( \mathbf{x}^* \). The rows and columns of \( G_o \) corresponding to model parts not found in the image are set to 0. In Eqn. (4), each pairwise potential \( G_{ij} \) is weighted by the product \( \sigma(r_i)\sigma(r_j) \), where \( \sigma(r_i) = 1/(1 + e^{-r_i}) \).

The primary reason for passing the relevance parameters through a sigmoid function is the following: letting the relevances be unconstrained real-valued parameters would
not help us conclusively establish which parts indeed belong to the category and which ones do not. What we really want is a binary relevance variable that is 1 if the model part belongs to the category model and 0 otherwise. Having a binary variable would allow us to consider only those parts that truly belong to the object category and discard the irrelevant clutter. Our intuition is that if we squash the unconstrained relevances $r_i$ we effectively turn them into soft binary variables, and during training we force them to be either relevant ($\sigma(r_i) \approx 1$) or irrelevant ($\sigma(r_i) \approx 0$). This is exactly what happens in practice. The squashed relevances of most parts go either to 1 or 0, thus making it possible to remove the irrelevant ones ($\sigma(r_i) \approx 0$) without affecting the approximate posterior $S(G_o, r)$. An additional benefit of squashing the relevance parameters is that it damps the effect of very large or very small negative values of $r_i$, reducing overfitting without the need for an explicit regularization term.

The higher the relevances $r_i$ and $r_j$, the more $G_o(i,j)$ contributes to the posterior. It is important to note that the relevance of one part is considered with respect to its pairwise relationships with all other parts together with their relevances. Therefore, parts are evaluated based on how well they work together as a team, rather than individually. Finally, an important aspect of the approach is that we interpret the logistic classifier $S(G_o, r)$ not as the true posterior, which is impractical to compute, but rather as a distance function that is specifically tuned for classification.

### 3. Learning

The model parameters to be learned consist of: the pairwise geometric relationships $e_{ij}$ between all pairs of parts, the sensitivity to deformations $w$ (which defines the pairwise potentials), the relevance parameters $r$ and $q_0, q_1$ (which define the classification function $S$). The learning steps are summarized in Figure 3 and detailed below.

#### 3.1. Initialization

We first initialize the pairwise geometric parameters $(e_{ij})$ for each pair of model parts by simply copying them from a positive training image. Thus, our initial model will have as many parts as the first training image used and the same pairwise relationships. We initialize the rest of the parameters to a set of default values. For each part $i$ we set the default value of its $r_i$ to 0 ($\sigma(r_i) = 0.5$). The default parameters of the pairwise potentials ($w$) are learned independently as described in Section 4.

#### 3.2. Updating the Parameters

Starting from the previous values, we update the parameters by minimizing the familiar sum-of-squares error function (typically used for training neural networks) using sequential gradient descent. The objective function is differentiable with respect to $r$, $q_0$ and $q_1$ since they do not affect the optimum $x^*$ (for the other parameters we differentiate assuming fixed $x^*$):

$$J = \sum_{n=1}^{N} b_n (S(G_o^{(n)}, r) - t^{(n)})^2. \quad (5)$$

Here $t^{(n)}$ denotes the ground truth for the $n^{th}$ image (1 if the object is present in the image, 0 otherwise). The weights $b_n$ are fixed to $m_N/m_P$ if $t^{(n)} = 1$ and 1 otherwise, where $m_N$ and $m_P$ are the number of negative and positive images, respectively. These weights balance the relative contributions to the error function between positive and negative examples. The matrix $G_o^{(n)}$ contains the pairwise potentials for the optimal localization for the $n^{th}$ image.

We update the parameters using sequential gradient descent, looping over all of the training images for a fixed number of iterations; in practice this reliably leads to convergence. The learning update for any given model parameter $\lambda$ for the $n^{th}$ example has the general form of:

$$\lambda \leftarrow \lambda - \rho b_n (S(G_o^{(n)}, r) - t^{(n)}) \frac{\partial S(G_o^{(n)}, r)}{\partial \lambda}, \quad (6)$$

where $\rho$ denotes the learning rate. Using this general rule we can easily write the update rules for all of the model parameters. The pairwise potentials ($G_o$) do not depend on the parameters $r$, $q_0$, $q_1$. It follows that the optimal labeling $x^*$ of the localization problem remains constant if we only update $r$, $q_0$, $q_1$. In practice we only update $r$, $q_0$, $q_1$ and the pairwise distances $d_{ij}$, while assuming that $x^*$ does not change, thus avoiding the computationally-expensive step of matching after each gradient descent update.

#### 3.3. Removing Irrelevant Parts

As mentioned earlier, the relevance values $\sigma(r_i)$ for each part $i$ tend to converge either toward 1 or 0. This is due to the fact that the derivative of $J$ with respect to the
free relevance parameters $r_i$ is zero only when the output $S(G_o^{(n)}, r)$ is either 0 or 1, or when the relevance $\sigma(r_i)$ is either 0 or 1, the latter being much easier to achieve. This is the key factor that allows us to discard irrelevant parts without significantly affecting the output $S(G_o^{(n)}, r)$. Therefore, all parts with $\sigma(r_i) \approx 0$ are discarded. In our experiments we observe that the relevant features typically belong to the true object of interest (Figure 5).

3.4. Adding New Parts

We proceed by merging the current model with a newly-selected training image (randomly selected from the ones on which the recognition output is not close enough to 1): we first localize the current model in the new image, thus finding the subset of features in the image that shares similar pairwise geometric relationships with the current model. Next, we add to the model new parts corresponding to all of the image features that fail to match the current model parts. As before, we initialize all the corresponding parameters involving newly-added parts by copying the geometric relationships between the corresponding features and using default values for the rest. At this stage, different view-points or shapes of our category can be merged (Figure 4). The geometric configurations shared by different aspects are already in the model (that is why we first perform matching) and only the novel configurations are added (from the parts that did not match). After adding new parts we return to the stage of updating the parameters (Figure 3). We continue this loop until we are satisfied with the error rate.

The approach of adding training images one-by-one is related to incremental semi-supervised learning methods [25]. In our case, we later discard the information that is not useful for recognition (the parts with zero relevance). Removing and adding parts enables our model to grow or shrink dynamically, as needed for the recognition task.

4. Learning Pairwise Geometric Potentials

We learn a default set of parameters $w$ for the pairwise potentials independently of the object parts $(i, j)$ and the object class $C$. Learned independently, the pairwise potentials are logistic classifiers designed to model the posterior that a given pair of assignments is correct given the geometric deformation $g$, regardless of the object class $C$ or the specific parts $(i, j)$. We learn the default $w$, from a set of manually-selected correct correspondences and randomly-selected set of incorrect ones, using the iteratively re-weighted least-squares algorithm. The correspondences are selected from different databases used in the literature: CALTECH-5 (faces, motorbikes, airplanes, motorcycles, cars, leaves, background), INRIA-horses and GRAZ-02 (person, bikes, cars, background). The same set of default $w$ is used in all of our recognition experiments.

Data points $g_{ij}(a, b)$ are collected for both the positive (pair of correct correspondences) and the negative class (at least one assignment is wrong), where image feature $a$ from one image is matched to the image feature $i$ from the other image. For randomly-selected pairs of images containing the same object category, we manually select approximately 8000 pairs of correct correspondences per database (whenever the poses were similar enough to enable finding exact correspondences between the contours of the two images). We select 16000 pairs of wrong correspondences per database.

Table 1 shows how the geometry-based pairwise classifier generalizes across different object categories. This pairwise classifier operates on pairs of assignments only and it is not a classifier of object categories. In particular, this set of experiments quantifies the extent to which pairwise geometric constraints can be used to distinguish between correct and incorrect correspondences, but it does not say anything about the effectiveness of the constraints to distinguish between categories. In this set of experiments we
Table 1. The classification rates (at equal error rate) of the geometry-based pairwise classifier trained and tested on different databases. Results are averaged over 10 runs. The numbers measure the discrimination between correct and incorrect correspondences, not the category recognition performance.

<table>
<thead>
<tr>
<th>Database</th>
<th>Caltech-5 (train)</th>
<th>INRIA (train)</th>
<th>GRAZ-02 (train)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caltech-5 (test)</td>
<td>97.42%</td>
<td>97.65%</td>
<td>97.23%</td>
</tr>
<tr>
<td>INRIA (test)</td>
<td>94.66%</td>
<td>95.33%</td>
<td>94.31%</td>
</tr>
<tr>
<td>GRAZ-02 (test)</td>
<td>92.93%</td>
<td>93.73%</td>
<td>93.24%</td>
</tr>
</tbody>
</table>

Table 2. The classification rates (at equal error rate) of the geometry-based pairwise classifier vs. the local feature classifier, for three different databases, averaged over 10 runs. Note that these are not category recognition results, but the results of classifiers on pairs of assignments.

<table>
<thead>
<tr>
<th>Database</th>
<th>local only</th>
<th>geometry only</th>
<th>combined</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caltech-5</td>
<td>86.73%</td>
<td>97.42%</td>
<td>98.10%</td>
</tr>
<tr>
<td>INRIA horses</td>
<td>80.89%</td>
<td>95.33%</td>
<td>96.40%</td>
</tr>
<tr>
<td>GRAZ-02</td>
<td>83.30%</td>
<td>93.24%</td>
<td>94.74%</td>
</tr>
</tbody>
</table>

Figure 6. Original image (left) and extracted contours (right). Right: different colors correspond to different connected components.

5. Implementation Details: Grouping Edges into Contours

We first obtain the edge map by using the Pb edge detector from Martin et al. [21]. Next, we remove spurious edges using a grouping technique inspired by Mahamud et al. [20]. Then we select image features by evenly sampling them (at every 20 pixels) along edge contours, as shown in Figure 6. Since the Pb detector is rather expensive, we also experimented with using a Canny edge detector instead and we obtain similar results after grouping the edge elements into connected contour fragments.

We obtain these contours by grouping pixels that form long and smooth curves. First, we group the edges into connected components by joining those pairs of edge pixels (i, j) that are both sufficiently close (i.e., within 5 pixels) and satisfy smoothness constraints based on collinearity and proximity (thus ensuring that the components only contain smooth contours).

For each connected component we form its weighted ad-
jacency matrix $A$ such that $A(i, j)$ is positive if edge pixels $(i, j)$ are connected and 0 otherwise. The value of $A(i, j)$ increases with the smoothness between $(i, j)$. The principal eigenvalue $\lambda$ of $A$ describes the average smoothness of this component. We keep only those components that are large enough (number of pixels $> sizeThresh$) and smooth enough ($\lambda > smoothThresh$).

This step is very efficient, usually taking less than a second per image, in Matlab on a 2GHz PC. The main use of these contours is to eliminate spurious edges more reliably than by simply thresholding the output of the edge detector. It also provides a better estimate of the normal at each edge pixel by considering only neighboring edges belonging to the same contour (Figure 6).

### 6. Experiments

Tables 3 and 4 compare the performance of our method with Winn et al. [27] on the Pascal challenge training dataset (587 images). This is an interesting experiment because our method only employs geometry and ignores local appearance; in contrast, Winn et al. focus on local texture information, while ignoring the geometry. We follow the same experimental setup, splitting the dataset randomly in two equal training and testing sets. The first set of experiments uses the provided bounding box (also used by Winn et al.). We outperform the texture-based classifier by more than 10%, confirming our intuition that shape is a stronger cue than local appearance for these types of object categories. Surprisingly, bikes and motorcycles are not confused as much as one might expect despite their similar shapes. In the second set of experiments, we do not use the bounding boxes, neither for training nor testing, in order to demonstrate that our method’s ability to learn in a weakly-supervised setting. The performance drops by approximately 5%, which is significant, but relatively low considering that the objects of interest in this experiment frequently occupy only a small fraction of the image area. A more serious challenge is that several positive images for one class contain objects from other categories (e.g., there are people present in some of the motorcycle and car images). In our reported results, an image from the “motorcycle” class containing both a motorbike and a person that was classified as “person” would be treated as an error.

As mentioned earlier, the models are compact representations of the relevant features present in the positive training set. The algorithm discovers relevant parts that, in our experiments, generally belong to the true object of interest despite significant background clutter. An interesting and useful feature of our method is its ability to integrate different view-points, aspects or shapes within the same category.

---


2. For the few images in which the object was too small, we select a bounding box of 4 times the area of the original bounding box.

### Table 3. Confusion Matrix on Pascal Dataset.

<table>
<thead>
<tr>
<th>Category</th>
<th>Bikes</th>
<th>Cars</th>
<th>Motorbikes</th>
<th>People</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bikes</td>
<td>80.7%</td>
<td>0%</td>
<td>7%</td>
<td>12.3%</td>
</tr>
<tr>
<td>Cars</td>
<td>5.7%</td>
<td>88.6%</td>
<td>5.7%</td>
<td>0%</td>
</tr>
<tr>
<td>Motorbikes</td>
<td>4.7%</td>
<td>0%</td>
<td>95.3%</td>
<td>0%</td>
</tr>
<tr>
<td>People</td>
<td>7.1%</td>
<td>0%</td>
<td>0%</td>
<td>92.9%</td>
</tr>
</tbody>
</table>

### Table 4. Average multiclass recognition rates on Pascal.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Ours (bbox)</th>
<th>Ours (no bbox)</th>
<th>Winn et al. (bbox)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pascal Dataset</td>
<td>89.4%</td>
<td>84.8%</td>
<td>76.9%</td>
</tr>
</tbody>
</table>

Figure 7. Training Images (Left) and the contours on which the relevant features were found (Right) (Figure 4). This happens automatically, as new parts are added from positive images.

The computational cost of classifying a single image does not depend on the number of training images: the model is a compact representation of the relevant features in the training images, usually containing between 40 to 100 parts. The size of the model is not fixed manually; it is an automatic outcome from the learning stage.

We also compare our method with Opelt et al. [23] on the GRAZ-01 and GRAZ-02 datasets (Table 5). We run the experiments on the same training and test sets on full images (no bounding boxes were used). Opelt et al. focus mainly on local appearance and select descriptors based on their individual performance and combine them using AdaBoost. This is in contrast with our approach, since our features by themselves have no discriminative power. It is their combined configuration that makes them together discriminative.

We further test our algorithm on the INRIA (168 images) and Weizmann Horse (328 images) [3] databases, using for the negative class the GRAZ-02 background images. We do not use objects masks (nor bounding boxes) and we randomly split the images in equal training and testing sets.
Table 5. Category recognition rates (at equal error rate) on GRAZ. Dataset (People and Bikes), Shotton and INRIA horses datasets. Bounding boxes (masks) are not used.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Ours</th>
<th>Opelt et al. (1)</th>
<th>Opelt et al. (2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>People (GRAZ I)</td>
<td>82.0%</td>
<td>76.5%</td>
<td>56.5%</td>
</tr>
<tr>
<td>Bikes (GRAZ I)</td>
<td>84.0%</td>
<td>78.0%</td>
<td>83.5%</td>
</tr>
<tr>
<td>People (GRAZ II)</td>
<td>86.0%</td>
<td>70.0%</td>
<td>74.1%</td>
</tr>
<tr>
<td>Bikes (GRAZ II)</td>
<td>92.0%</td>
<td>76.4%</td>
<td>74.0%</td>
</tr>
<tr>
<td>Horses (Shotton)</td>
<td>92.02%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Horses (INRIA)</td>
<td>87.14%</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

The INRIA horse database includes significant changes in scale, pose and multiple horses present in the same image.

7. Conclusion

We demonstrate that exploiting the pairwise interactions between simple shape features enables us to match and even exceed the performance of state-of-the-art algorithms that use more complex descriptors. Our results confirm the intuition that shape is a very powerful cue for object category recognition. This paper demonstrates that by capturing shape through second-order relationships (as opposed to local, first-order descriptors), we can build flexible models that accommodate significant deformations, while still being discriminative against background clutter.

Acknowledgments

This research was performed in part for the Intelligent Robotics Development Program, one of the 21st Century Frontier R&D Program funded by the Ministry of Commerce, Industry and Energy of Korea; and in part by a summer internship at Intel Research Pittsburgh.

References