A Graph-Matching Kernel for Object Categorization

Olivier Duchenne[1,2,3] Armand Joulin[1,2,3] Jean Ponce[2,3]

[1]INRIA [2]École Normale Supérieure de Paris

Abstract

This paper addresses the problem of category-level image classification. The underlying image model is a graph whose nodes correspond to a dense set of regions, and edges reflect the underlying grid structure of the image and act as springs to guarantee the geometric consistency of nearby regions during matching. A fast approximate algorithm for matching the graphs associated with two images is presented. This algorithm is used to construct a kernel appropriate for SVM-based image classification, and experiments with the Caltech 101, Caltech 256, and Scenes datasets demonstrate performance that matches or exceeds the state of the art for methods using a single type of features.

1. Introduction

Explicit correspondences between local image features are a key element of image retrieval [30] and specific object detection [29] technology, but they are seldom used [3, 13, 16, 35] in object categorization, where bags of features (BOFs) and their variants [4, 7, 8, 10, 26, 38, 39] have been dominant. However, as shown by Caputo and Jie [6], feature correspondences can be used to construct an image comparison kernel [35] that, although not positive definite, is appropriate for SVM-based classification, and often outperforms BOFs on standard datasets such as Caltech 101 in terms of classification rates. This is the first motivation for the approach to object categorization proposed in the rest of this presentation. Our second motivation is that image representations that enforce some degree of spatial consistency–such as HOG models [8], spatial pyramids [26], and their variants, e.g. [4, 38]–usually perform better in image classification tasks than pure bags of features that discard all spatial information. This suggests adding spatial constraints to pure appearance-based matching and thus formulating object categorization as a graph matching problem where a unary potential is used to select matching features, and a binary one encourages nearby features in one image to match nearby features in the second one.

Concretely, we propose to represent images by graphs whose nodes and edges represent the regions associated with a coarse image grid and their adjacency relationships. The problem of matching two images is formulated as the optimization of an energy akin to a first-order multi-label Markov random field (MRF),[4] defined on the corresponding graphs, the labels corresponding to node assignments. Variants of this formulation have been used in problems ranging from image restoration, to stereo vision, and object recognition. However, as shown by a recent comparison [23], its performance in image classification tasks has been, so far, a bit disappointing. As further argued in the next section, this may be due in part to the fact that current approaches are too slow to support the use of sophisticated classifiers such as support vector machines (SVMs). In contrast, this paper makes three original contributions:

1. Generalizing [6, 35] to graphs, we propose in Section 2 to use the value of the optimized MRF associated with two images as a (non positive definite) kernel, suitable for SVM classification.

2. We propose in Section 3 a novel extension of Ishikawa's method [20] for optimizing the MRF which is orders of magnitude faster than competing algorithms (e.g., [23, 25, 27] for the grids with a few hundred nodes considered in this paper). In turn, this allows us to combine our kernel with SVMs in image classification tasks.

3. We demonstrate in Section 4 through experiments with standard benchmarks (Caltech 101, Caltech 256, and Scenes datasets) that our method matches and in some cases exceeds the state of the art for methods using a single type of features.

1.1. Related work

Early "appearance-based" approaches to image retrieval and object recognition, such as color histograms, eigenfaces or appearance manifolds, used global image descriptors to match images. Schmid and Mohr [30] proposed instead

[3]WILLOW project-team, Laboratoire d'Informatique de l'Ecole Normale Supérieure, ENS/INRIA/CNRS UMR 8548.

[4]As is often the case in computer vision applications, our use of the MRF notion here is slightly abusive since our formulation does not require or assume any probabilistic modeling.

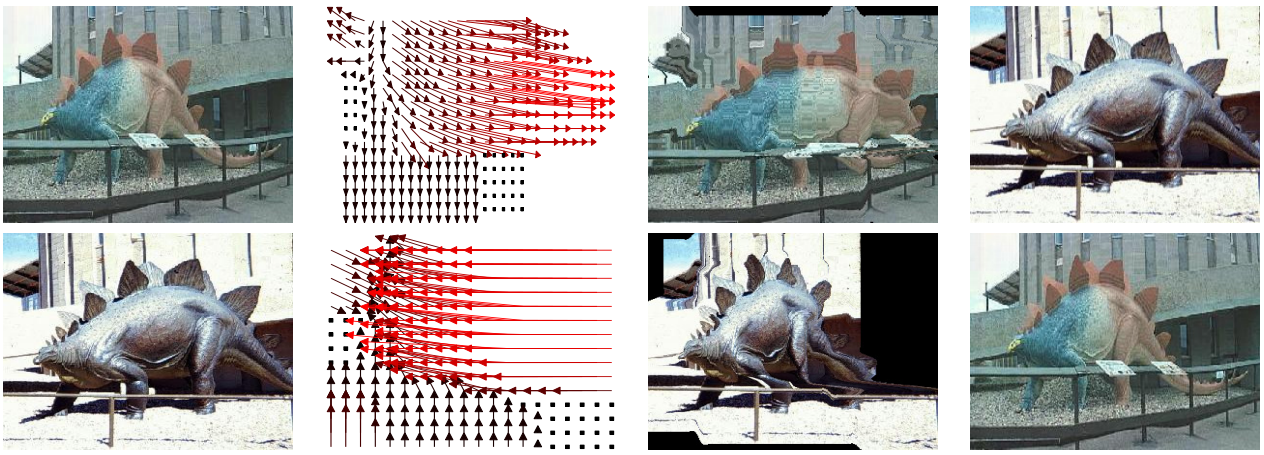

Figure 1. The leftmost picture in each row is matched to the rightmost one. The second panel shows the deformation field (displacements) computed by our matching procedure, and the third panel shows the leftmost image after it has been deformed according to that field. Since the matching process is asymmetric, our kernel is the average of the two matching scores (best seen in color).

to formulate image retrieval as a correspondence problem where local and semi-local image descriptors (jets and geometric configurations of image neighbors) are used to match individual (or groups of) interest points, and these correspondences vote for the corresponding images. A related technique was proposed by Lowe [29] to detect particular object instances using correspondences established between SIFT images descriptors, which have proven very effective for this task. Following Sivic and Zisserman [32], many modern approaches to image retrieval use SIFT and SIFT-like features, but abandon the correspondence formulation in favor of an approach inspired by text retrieval, where features are quantized using k-means to form a *bag of features* (or *BOF*)—that is, a histogram of quantized features. Pictures similar to a query image are then retrieved by comparing the corresponding histograms, a process that can be sped up by the use of inverted file systems and various indexing schemes. As noted by Jegou et al. [21], image retrieval methods based on bags of features can be seen as voting schemes between local features where the Voronoi cells associated with the k-means clusters are used to approximate the inter-feature distances. In turn, this suggests exploring alternative approximation schemes that retain the efficiency of bags of features in terms of memory and speed, yet afford a retrieval performance comparable to that of correspondence-based methods ([21] is an example among many others of such a scheme).

This also suggests that explicit correspondences between features may provide a good measure of image similarity in image categorization tasks. Variants of this approach can be found in quite different guises in the part-based constellation model of Fergus et al. [13], the naive Bayes nearest-neighbor algorithm of Boiman et al. [3], and the pyramid matching kernel of Grauman and Darrell [17]. Yet, although these techniques may give state-of-the-art results (e.g., [3]), it is probably fair to say that methods using bags of features and their variants [4, 7, 8, 10, 26, 38, 39] to train sophisticated classifiers such as support vector machines (SVMs) are dominant today in image classification and object detection tasks. This may be due, in part, to the simplicity and efficiency of the BOF model, but one should keep in mind that, as in the image retrieval domain, BOF-based approaches can be seen as approximations of their correspondence-based counterparts and, indeed, Caputo and Jie [6] have shown that feature correspondences can be used to construct an image comparison kernel [35] that, although not positive definite, is appropriate for SVM-based classification, and often outperforms BOFs on standard datasets such as Caltech 101 in terms of classification rates if not run time.

Bags of features discard all spatial information. There is always a trade-off between viewpoint invariance and discriminative power, and retaining at least a coarse approximation of an image layout makes sense for many object classes, at least when they are observed from a limited range of viewpoints. Indeed, image representations that enforce some degree of spatial consistency –such as HOG models [8], spatial pyramids [26], and their variants, e.g. [4, 38]– typically perform better in image classification tasks than pure bags of features. As noted in the introduction, this suggests adding spatial constraints to correspondence-based approaches to object categorization. In this context, several authors [2, 9, 12, 13, 14, 23, 27, 28, 31] have proposed using graph-matching techniques to minimize pairwise geometric distortions while establishing correspondences between object parts, interest points, or small image regions. The problem of matching two images is formulated as the optimization of an energy akin to a first-order multi-label MRF, defined on the corresponding graphs, the labels corresponding to node assignments

or, equivalently, to a set of discrete two-dimensional image translations. This optimization problem is unfortunately intractable for general graphs [5], prompting the use of restricted graph structures (e.g., very small graphs [13], trees [9], stars [12], or strings [23]) and/or approximate optimization algorithms (e.g., greedy approaches [14], spectral matching [27], alpha expansion [5], or tree-reweighted message passing, aka TRW-S [24, 34]).

1.2. Proposed approach

We propose in this paper to represent images by graphs whose nodes and edges represent the regions associated with a coarse image grid (about 500 regions) and their adjacency relationships. The regions are represented by the mid-level sparse features proposed in [4], and the unary potential used in our MRF is used to select matching features, while the binary one encourages nearby features in one image to match nearby features in the second one while discouraging matching nearby features to cross each other (the matching process is illustrated in Figure 1). The optimum MRF value is then used to construct a (non positive definite) kernel for comparing images (Section 2). We formulate the optimization of our MRF as a graph cuts problem, and propose as an alternative to alpha expansion [5] an algorithm that extends Ishikawa's technique [20] for optimizing one-dimensional multi-label problems to our two-dimensional setting (Section 3). This algorithm is particularly well suited to the grids of moderate size considered here: Our algorithm yields an image matching method that is empirically much faster (by several orders of magnitude) than alternatives based on alpha expansion [5], TRW-S [11, 28, 31], or the approximate string matching algorithm of [23], for our grid size at least. Speed is particularly important in kernel-based approaches to object categorization, since computing the kernel requires comparing all pairs of images in the training set. In turn, speed issues often force graph-matching techniques [2, 23] to rely on nearest-neighbor classification. In contrast, we use our kernel to train a support vector machine, and demonstrate in Section 4 classification results that match or exceed the state of the art for methods using a single type of features on standard benchmarks (Caltech 101, Caltech 256, and Scenes datasets).

2. A kernel for image comparison

2.1. Image representation

An image is represented in this paper by a graph $\mathcal{G}$ whose nodes represent the N image regions associated with a coarse image grid, and each node is connected with its four neighbors. The nodes are indexed by their position on the grid, defined as the corresponding couple of row and column indices. It should thus be clear that, when we talk of the "position" of a node n, we mean the couple $d_n = (x_n, y_n)$ formed by these indices. The corresponding "units" are not pixels but the region extents in the x (horizontal) and y (vertical) directions. For each node n in $\mathcal{G}$, we also define the feature vector F_n associated with the corresponding image region.

SIFT local image descriptors [29] are often used as low-level features in object categorization tasks. In [4], Boureau et al. propose new features which lead in general to better classification performance than SIFT. They are based on sparse coding and max pooling: Briefly, the image is divided into overlapping regions of 32×32 pixels. In each region, four 128-dimensional SIFT descriptors are extracted and concatenated. The resulting 512-dimensional vector is decomposed as a sparse linear combination of atoms of a learned dictionary. The vectors of the coefficients of this sparse decomposition are used as local sparse features. These local sparse features are then summarized over larger image regions by taking, for each dimension of the vector of coefficients, the maximum value over the region (max pooling) [4]. We use the result of max pooling over our graph regions as image features in this paper.

2.2. Matching two images

To match two images, we distort the graph $\mathcal{G}$ representing the first one to the graph $\mathcal{G}'$ associated with the second one while enforcing spatial consistency across adjacent nodes. Concretely, correspondences are defined in terms of displacements within the graph grid: Given a node n in $\mathcal{G}$, and some displacement d_n, n is matched to the node n' in $\mathcal{G}'$ such that $p_{n'} = p_n + d_n$, and we maximize the energy function

$$E_{\rightarrow}(d) = \sum_{n \in \mathcal{V}} U_n(d_n) + \sum_{(m,n) \in \mathcal{E}} B_{m,n}(d_m, d_n), \qquad (1)$$

where $\mathcal{V}$ and $\mathcal{E}$ respectively denote the set of nodes and edges of $\mathcal{G}$, d is the vector formed by the displacements associated with all the elements of $\mathcal{V}$, and U_n and $B_{m,n}$ respectively denote unary and binary potentials that will be defined below. Note that we fix a maximum displacement in each direction, K, leading to a total of K^2 possible displacements d_n for each node n. The energy function defined by Eq. (1) is thus a multi-label Markov random field (MRF) where the labels are the displacements. Typically, we set $K = 11$.

The unary potential is simply the correlation (dot product) between F_n and $F_{n'}$. The binary potential enforces spatial consistency and is decomposed into two terms. The first one acts as a spring:

$$u_{m,n}(d_m, d_n) = -\lambda \|d_m - d_n\|_1, \qquad (2)$$

where λ is the positive spring constant. We use the ℓ_1 distance to be robust to sparse distortion differences.

We focus on categorizing objects (as opposed to more general scenes) such that, for some range of viewpoints, shape variability can be represented by image displacements varying smoothly over the image, and object fragments typically cannot cross each other. We thus penalize crossing by adding a binary potential between nearby nodes such that:

$$v_{m,n}(d_m, d_n) = \begin{cases} -\mu[dx_n - dx_m]_+ & \text{if } x_n = x_m + 1 \\ & \text{and } y_n = y_m, \\ -\mu[dy_n - dy_m]_+ & \text{if } x_n = x_m \\ & \text{and } y_n = y_m + 1, \\ 0 & \text{otherwise,} \end{cases}$$

where μ a positive constant and $[z]_+ = \max(0, z)$.
The overall binary potential is thus:

$$B_{m,n}(d_m, d_n) = u_{m,n}(d_m, d_n) + v_{m,n}(d_m, d_n). \quad (3)$$

2.3. A new kernel

The two graphs $\mathcal{G}$ and $\mathcal{G}'$ play asymmetric roles in the objective function $E_{\rightarrow}(d)$. One can define a second objective function $E_{\leftarrow}(d)$ by reversing the roles of $\mathcal{G}$ and $\mathcal{G}'$. Optimizing both functions allows us to define a kernel for measuring the similarity of two images, whose value is $\frac{1}{2}(\max_{d_1} E_{\rightarrow}(d_1) + \max_{d_2} E_{\leftarrow}(d_2))$. This kernel does not satisfy the positive definitiveness criterion (this is because of the maximization of Eq. (1), see [6] for the corresponding argument in a related situation) but, by thresholding negative eigenvalues to 0, it is appropriate for constructing a kernel matrix S. This matrix is used to train a support vector machine classifier (SVM) in a one-vs-all fashion.

3. Optimization

Maximizing Eq. (1) over all possible deformations is NP hard for general graphs [5]. Many algorithms have been developed for finding approximate solutions of this problem [5, 20, 24, 34]. Alpha expansion [5] is a greedy algorithm with strong optimality properties. TRW-S [24, 34] has even stronger optimality properties for very general energy functions, but it is also known to be slower than alpha expansion [22]. Ishikawa's method [20] is a fast alternative that finds the global maximum for a well-defined family of energy functions. Since our energy function defined in Eq. (1) possesses properties very close to those of this family, we focus here on Ishikawa's method, and propose extensions to handle the specificities of our energy. Note that Ishikawa's method is one of the rare algorithms capable of solving exactly multi-label MRF.

3.1. Ishikawa's method

Ishikawa [20] have proposed a min-cut/max-flow method for finding the global maximum of a multi-label MRF whose binary potentials verify:

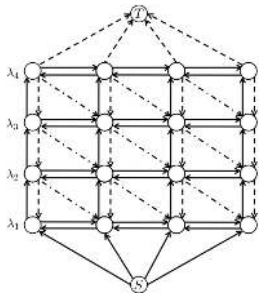

Figure 2. A graph associated with Ishikawa's method. On the horizontal axis are the nodes $n \in \{1, \ldots, 4\}$ and on the vertical axis the labels $(\lambda_j)_{1 \leq j \leq 4}$. Each Ishikawa node corresponds to a node n and a label λ_j. The plain vertical arrows represent the unary potentials $U_n(\lambda_j)$. The dashed vertical arrows represent the infinite edges. The plain horizontal arrows correspond to the binary potentials $u_{mn}(\lambda_j, \lambda_j)$ while the dash-dot arrows correspond to the non-crossing binary potentials $v_{mn}(\lambda_j, \lambda_j - 1)$.

$$B_{mn}(\lambda_m, \lambda_n) = g(\lambda_m - \lambda_n), \quad (4)$$

where g is a concave function and λ_m (resp. λ_n) is the label of the node m (resp. n). The set of labels has to be linearly ordered. The Ishikawa method relies on building a graph with one node n^λ for each pair of node n of $\mathcal{G}$ and a label λ, see Figure 2 from details. A min-cut/max-flow algorithm is performed on this graph. If it cuts the edge from $n^{\lambda-1}$ to n^λ, we assign the node n to the label λ. Ishikawa [20] proves that this assignment is optimal.

Unfortunately, our set of labels is two-dimensional and there is no linear ordering of $\mathbb{N}^2$ which keeps the induced binary potential concave. A simple argument is that for any label $d_n = (dx_n, dy_n)$, its 4-neighbor labels d_m (e.g., $d_m = (dx_n + 1, dy_n)$, $d_m = (dx_n, dy_n - 1)$...) are equally distant to d_n,i.e. $B(d_n, d_m) = c_n < 0$ for all its neighbors. Any concave function which has the same value c for 3 different points is necessarily always below c. This contradicts $B(d_n, d_n) = 0$.

3.2. Proposed method: Curve expansion

We propose in this section a generalization of Ishikawa's method capable of solving problems with two-dimensional labels.

3.2.1 Two-step curve expansion

The binary part of our MRF can readily be rewritten as:

$$B(d) = \sum_{(m,n) \in \mathcal{E}} g_x(dx_m - dx_n) + g_y(dy_m - dy_n), \quad (5)$$

where g_x and g_y are negative concave functions. For a fixed value of $dx = (dx_1, \ldots, dx_N)$, the potentials in $B(d)$ verify condition (4), and Ishikawa's method can be used to find

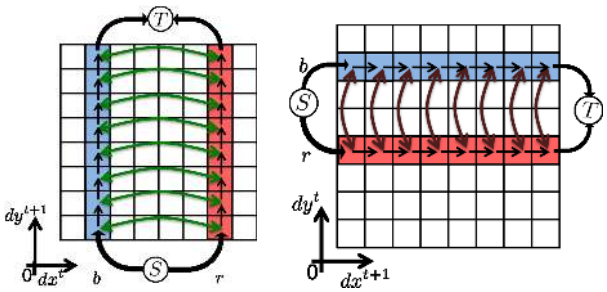

Figure 3. Vertical curve expansion (left) and horizontal curve expansion (right) with two nodes, blue ("b") and red ("r"). The grid corresponds to all possible distortions d. The blue (red) squares represent the allowed d for the curve expansion of the blue (red) node. The arrows explain the construction of Ishikawa graph: The black arrows are the unary potentials $U_n(d_n)$ and the green arrows (resp. red) are the binary potentials $B_{mn}(d_m, d_n)$ for vertical (resp. horizontal) moves, i.e., $dx^{t+1} = dx^t$ (resp. $dy^{t+1} = dy^t$). The infinite edges are omitted for clarity (best seen in color).

the optimal distortion $dy = (dy_1, \dots, dy_N)$ given dx. We thus alternate between optimizing over dy given dx ("vertical move") and optimizing over dx given dy ("horizontal move"). Figure 3 shows an example of a vertical move (left) and a horizontal move for two nodes.

More precisely, we first initialize d by computing the following upper-bound of $E_{1\to 2}$:

$$\max_{d} \sum_{n\in\mathcal{V}} U_n(d_n) + \sum_{(m,n)\in\mathcal{E}} g_x(dx_m - dx_n).$$

Since $d_n = (dx_n, dy_n)$, this can be rewritten as:

$$\max_{dx} \sum_{n\in\mathcal{V}} \max_{dy_n} \left(U_n(dx_n, dy_n)\right) + \sum_{(m,n)\in\mathcal{E}} g_x(dx_m - dx_n),$$

which can be solved optimally using Ishikawa's method. We then solve a sequence of vertical and horizontal moves:

$$dy^{t+1} \leftarrow \operatorname*{argmax}_{dy} \sum_{n\in\mathcal{V}} U_n(dx_n^t, dy_n) + \sum_{(m,n)\in\mathcal{E}} g_y(dy_m - dy_n), \quad (6)$$

$$dx^{t+1} \leftarrow \operatorname*{argmax}_{dx} \sum_{n\in\mathcal{V}} U_n(dx_n, dy_n^t) + \sum_{(m,n)\in\mathcal{E}} g_x(dx_m - dx_n).$$

The local minimum obtained by this procedure is lower than $2\left(\sqrt{N_l}\right)^{N_n}$ configurations, where N_l is the number of labels. By comparison, the minimum obtained by alpha expansion is only guaranteed to be lower than $N_l 2^{N_n}$ other configurations [5].

3.2.2 Multi-step curve expansion

The procedure proposed in the previous section only allows vertical and horizontal moves. Let us now show how to extend it to allow more complicated moves. Ishikawa's method reaches the global minimum of functions verifying

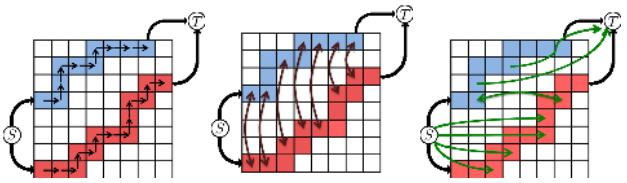

Figure 4. An example of curve expansion move for two nodes, b and r. The blue curve corresponds to the nodes $(n_b^\lambda)_{1\le\lambda\le N_l}$ obtained by applying the labels λ to the node b. On the left, we show the arrows between nodes n^λ and $n^{\lambda+1}$ representing the unary potential $U_n^{\lambda+1}$ (infinite arrows in the opposite direction have been omitted for clarity). The center (resp. right) panel represents binary potential edges between nodes b and r with labels λ_b and λ_r corresponding to vertical (resp. horizontal) displacements. A displacement which cannot be connected to the corresponding displacements of another node, is either connected to the source S or the sink T (best seen in color).

condition (4). It can be extended to more general binary terms by replacing (4) by:

$$\forall \lambda, \mu,\ B(\lambda,\mu) + B(\lambda+1,\mu+1) \le B(\lambda+1,\mu) + B(\lambda,\mu+1).$$

This is a direct consequence of the proof in [20]. With this condition, we can handle binary functions which do not only depend on pairwise label differences. This allows us to use more complicated moves than horizontal or vertical displacements. We thus propose the following algorithm: At each step t, we consider an ordered list of P_t possible distortions $\mathcal{D}^t = [\tilde{d}_1, \tilde{d}_2, ..., \tilde{d}_{P_t}]$. Given nodes n with current distortion d_n^t, we update these distortions by solving the following problem:

$$\max_{\tilde{d}_t\in\mathcal{D}^t} \sum_{n\in\mathcal{V}} U_n(d_n^t + \tilde{d}_n^t) - \lambda \sum_{(m,n)\in\mathcal{E}} \|d_n^t + \tilde{d}_m^t - (d_n^t + \tilde{d}_n^t)\|_1,$$

where $\tilde{d}_t = (\tilde{d}_1^t, \dots, \tilde{d}_N^t)$. Then the updated distortion of node n is $d_n^{t+1} \leftarrow d_n^t + \tilde{d}_n^t$.

For example the vertical move, Eq. (6), consists of distortions $\tilde{d}$ of the form $(d\tilde{x}, d\tilde{y}) = (0, k)$ for $k \in \{-(K-1)/2, \dots, (K-1)/2\}$. In practice, we construct a graph inspired by the one constructed for Ishikawa's method. An example is shown in Figure 4 with two nodes.

Note that the set of all possible distortions $\mathcal{D}$ can be different for each node, which gives N different $\mathcal{D}_n$'s. The only constraint is that all the $(\mathcal{D}_n)_{1\le n\le N}$ should be increasing (or decreasing) in y and increasing (or decreasing) in x.

4. Experiments

The proposed approach to graph matching has been implemented in C++. In this section we compare its running time to competing algorithms, before presenting image matching results and a comparative evaluation with the state of the art in image classification tasks on standard benchmarks (Caltech 101, Caltech 256 and Scenes).

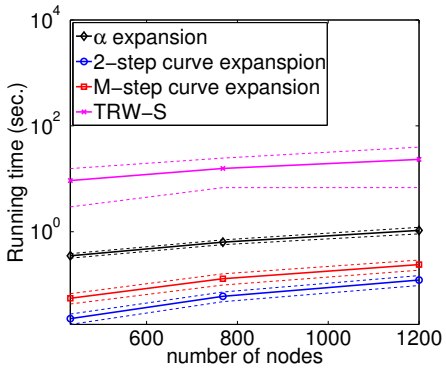

Figure 5. Comparison between the running times of TRW-S (purple), alpha expansion (black), 2-step (blue) and multi-step (red) curve expansions, for an increasing number of nodes in the grid (best seen in color, running time in log scale).

4.1. Running time

We compare here the running times of our curve expansion algorithm to the alpha expansion and TRW-S. For the alpha expansion, we use the C++ implementation of Boykov et al. [5, 25]. For TRW-S, we use the C++ implementation of Kolmogorov [24]. For the multi-step curve expansion we use four different moves (horizontal, vertical and diagonal moves). All experiments are performed on a single 2.4 gHz processor with 4 gB of RAM. We take 100 random pairs of images from Caltech 101 and run the four algorithms on increasing grid sizes. The results of our comparison are shown in Figure 5. The 2-step and multi-step curve expansions are much faster than the alpha expansion and TRW-S for grids with up to 1000 nodes or so. However, empirically, their complexity in the number of nodes is higher than alpha expansion's, which makes them slower for graphs with more than 4000 nodes.

In terms of average minimization performance, 2-step curve expansion is similar to alpha expansion, whereas the multi-step curve expansion with 4 moves, and TRW-S improve the results by respectively, 2% and 5%. However, these improvements have empirically little influence on the overall process. Indeed, for categorization, a coarse matching seems to be enough to obtain high categorization performance. Thus, the real issue in this context is time and we prefer to use the 2-step curve expansion which matches two images in less than **0.04 seconds** for 500 nodes.

Other methods have been developed for approximate graph matching [2, 23, 27], but their running time is prohibitive for our application. Berg et al. [2] match two graphs with 50 nodes in 5 seconds and Leordeanu et al. [27] match 130 points in 9 seconds. Kim and Grauman [23] propose a string matching algorithm which takes around 10 seconds for 4800 nodes and around 1 second to match 500 nodes.

Figure 6. We match the top-left image to the top-right image with different value of the crossing constant μ (on the bottom). From left to right, $\mu = 0$, .5 and 1.5.

4.2. Image matching

To illustrate image matching, we use a finer grid than in our actual image classification experiments to show the level of localization accuracy that can be achieved. We fix 30×40 grid with maximum allowed displacement $K = 15$. In Figure 6, we show the influence of the parameter μ which penalizes the crossing between matches. On the left panel of the figure, where crossings are not penalized, some parts of the image are duplicated, whereas, when crossings are forbidden (right panel), the deformed picture retains the original image structure yet still matches well the model. For our categorization experiments, we choose a value of this parameter in between (middle panel). Figure 7 shows some matching results for images of the same category, similar to Figure 1 (more examples can be seen in the supplementary material).

4.3. Image classification

We test our algorithm on the publicly available Caltech 101, Caltech 256, and Scenes datasets. We fix $\lambda = 0.1$, $\mu = 0.5$ and $K = 11$ for all our experiments on all the databases. We have tried two grid sizes: 18×24, and 24×32, and have consistently obtained better results (by 1 to 2%) using the coarser grid, so only show the corresponding results in this section. Our algorithm is robust to the choice of K (as long as K is at least 11). The value for λ and μ have been chosen by looking at the resulting matching on a single pair of images. Obviously using parameters adapted to each database and selected by cross-validation would have lead to better performance. Since time is a major issue when dealing with large databases, we use the 2-step curve expansion instead of the M-step version.

Caltech 101. Like others, we report results for two different training set sizes (15 or 30 images), and report the average performance over 20 random splits. Our results are compared to those of other methods based on graph matching [1, 2, 23] in Table 1, which shows that we obtain classification rates that are better by more than 12%. We also compare our results to the state of art on Caltech 101 in Ta-

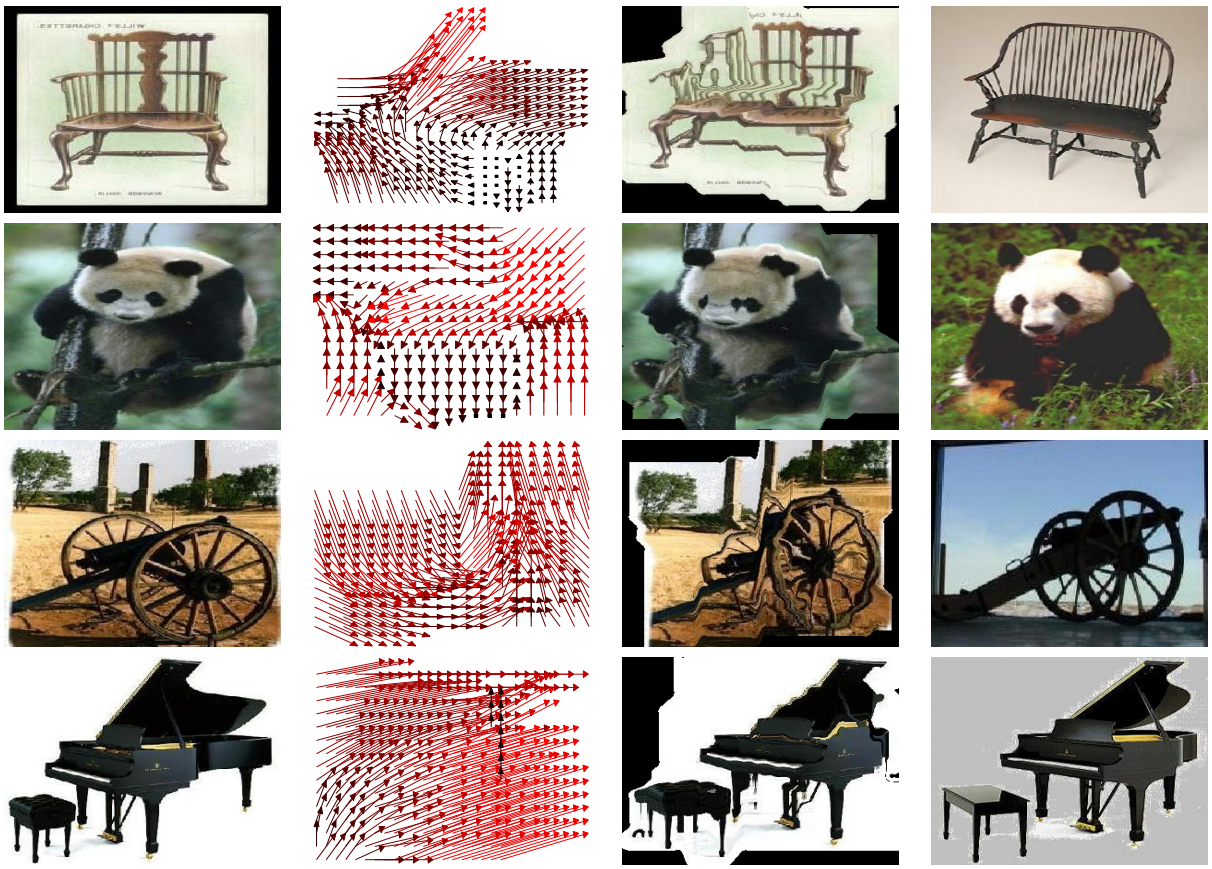

Figure 7. Additional examples of image matching on the Caltech 101 dataset. The format of the figure is the same as that of Figure 1.

ble 2. Our algorithm outperforms all competing methods for 15 training examples, and is the third performer overall, behind Yang et al. [36] and, Todorovic and Ahuja [33] for 30 examples. Note that our method is the top performer among algorithms using a single type of feature for both 15 and 30 training examples.

Caltech 256. Our results are compared with the state of the art for this dataset in Table 3. They are similar to those obtained by methods using a single feature [3, 23], but not as good as those using multiple features ([3] with 5 descriptors,[33]).

Scenes. A comparison with the state of the art on this dataset is given in Table 4. Our method is the second top performer below Boureau et al. [4]. This result is expected since it is designed to recognize objects with a fairly consistent spatial layout (at least for some range of viewpoints). In contrast, scenes are composed of many different elements that move freely in space.

Caltech101 (%)	
Graph-matching based Method	15 examples
BergMatching [2]	48.0
GBVote [1]	52.0
Kim and Grauman [23]	61.5
Ours 18×24	**75.3** $\pm$ **0.7**

Table 1. Average recognition rates of methods based on graph matching for Caltech 101 using 15 training examples. In this table as in the following ones, the top performance is shown in bold.

Caltech101 (%)			
Feature	Method	15 examples	30 examples
Single	NBNN (1 Desc) [3]	65.0 $\pm$ 1.1	-
	Boureau et al. [4]	69.0 $\pm$ 1.2	75.7 $\pm$ 1.1
	Ours 18×24	**75.3** $\pm$ **0.7**	80.3 $\pm$ 1.2
Multiple	Gu et al.[19]	-	77.5
	Gehler et al. [15]	-	77.7
	NBNN (5 Desc)[3]	72.8 $\pm$ 0.4	-
	Todorovic et al.[33]	72.0	83.0
	Yang et al. [36]	73.3	**84.3**

Table 2. Average recognition rates of state-of-the-art methods for Caltech 101.

5. Conclusion

We have presented a new approach to object categorization that formulates image matching as an energy optimization problem defined over graphs associated with a coarse image grid, presented an efficient algorithm for optimizing this energy function and constructing the corresponding image comparison kernel, and demonstrated results that match

Caltech 256 (%)		
Feature	Method	30 examples
Single	SPM+SVM [18]	34.1
	Kim et al.[23]	36.3
	NBNN (1 desc) [3]	37.0
	Ours 18×24	$38.1 \pm .6$
Multiple	NBNN (5 desc) [3]	42.0
	Todorovic et al. [33]	**49.5**

Table 3. Average recognition rates of state-of-the-art methods for the Caltech 256 database.

Scenes database (%)	
Method	100 examples
Yang et al. [37]	80.3 ± 0.9
Lazebnik et al. [26]	81.4 ± 0.5
Ours 18×24	82.1 ± 1.1
Boureau et al. [4]	$\mathbf{84.3 \pm 0.5}$

Table 4. Average recognition rates of state-of-the-art methods for the Scenes database.

or exceed the state of the art for methods using a single type of features on standard benchmarks. Our framework for image classification can readily be extended to object detection using sliding windows. Future work will include comparing this method to that of Felzenszwalb et al. [10]: The two approaches are indeed related, since they both allow deformable image models and SVM-based classification, our dense, grid-based regions taking the place of their sparse, predefined rectangular parts. Another interesting research direction is to abandon sliding windows altogether in detection tasks, by matching bounding boxes available in training images to test scenes containing instances of the corresponding objects.

References

[1] A. Berg. *Shape Matching and Object Recognition*. PhD thesis, UC Berkeley, 2005. 6, 7

[2] A. C. Berg, T. L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondence. In *CVPR*, 2005. 2, 3, 6, 7

[3] O. Boiman, E. Schechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *CVPR*, 2008. 1, 2, 7, 8

[4] Y. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. In *CVPR*, 2010. 1, 2, 3, 7, 8

[5] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23:12221239, 2001. 3, 4, 5, 6

[6] B. Caputo and L. Jie. A performance evaluation of exact and approximate match kernels for object recognition. *ELCVIA*, 8:15–26, 2009. 1, 2, 4

[7] G. Csurka, C. Bray, C. Dance, and L. Fan. Visual categorization with bags of keypoints. In *ECCV Workshop*, 2004. 1, 2

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 1, 2

[9] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61:55–79, 2005. 2, 3

[10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. In *CVPR*, 2008. 1, 2, 8

[11] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *IJCV*, 70:41–54, 2006. 3

[12] R. Fergus, P. Perona, and A. Zisserman. A sparse object category model for efficient learning and exhaustive recognition. In *CVPR*, 2005. 2, 3

[13] R. Fergus, P. Perona, and A. Zisserman. Weakly supervised scale-invariant learning of models for visual recognition. *IJCV*, 71:273–303, 2006. 1, 2, 3

[14] M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computers*, C-22:67–92, 1973. 2, 3

[15] P. Gehler and S. Nowozin. On feature combination for multiclass object classication. In *ICCV*, 2009. 7

[16] K. Grauman and T. Darrell. Pyramid match kernels: Discriminative classification with sets of image features. In *ICCV*, 2005. 1

[17] K. Grauman and T. Darrell. Pyramid Match Hashing: Sub-Linear Time Indexing Over Partial Correspondences. In *CVPR*, 2007. 2

[18] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical report, Caltech, 2007. 8

[19] C. Gu, J. Lim, P. Arbelaez, and J.Malik. Recognition using regions. In *CVPR*, 2009. 7

[20] H. Ishikawa. Exact optimization for markov random fields with convex priors. *PAMI*, 25:1333–1336, 2003. 1, 3, 4, 5

[21] H. Jegou, M. Douze, and C. Schmid. Improving bag-of-features for large scale image search. *IJCV*, 87:313–336, 2010. 2

[22] H. Y. Jung, K. M. Lee, and S. U. Lee. Toward global minimum through combined local minima. In *ECCV*, 2008. 4

[23] J. Kim and G. K. Asymmetric region-to-image matching for comparing images with generic object categories. In *CVPR*, 2010. 1, 2, 3, 6, 7, 8

[24] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *PAMI*, 28:1568–1583, 2006. 3, 4, 6

[25] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *PAMI*, 26:147–159, 2004. 1, 6

[26] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 1, 2, 8

[27] M. Leordeanu and M. Hebert. A spectral technique for for correspondance problems using pairwise constraints. In *ICCV*, 2005. 1, 2, 3, 6

[28] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman. Sift flow: Dense correspondence across different scenes. In *ECCV*, 2008. 2, 3

[29] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60:91–110, 2004. 1, 2, 3

[30] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *PAMI*, 19:530–535, 1997. 1

[31] A. Shekhovtsov, I. Kovtun, and V. Hlavac. Efficient MRF deformation model for non-rigid image matching. *CVIU*, 112:91–99, 2008. 2, 3

[32] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, 2003. 2

[33] S. Todorovic and N. Ahuja. Learning subcategory relevances for category recognition. In *CVPR*, 2008. 7, 8

[34] M. Wainwright, T. Jaakkola, and A. Willsky. Exact map estimates by (hyper)tree agreement. In *NIPS*, 2002. 3, 4

[35] C. Wallraven, B. Caputo, and A. Graf. Recognition with local features: the kernel recipe. In *ICCV*, 2003. 1, 2

[36] J. Yang, Y. Li, Y. Tian, L. Duan, and W. Gao. Group-sensitive multiple kernel learning for object categorization. In *ICCV*, 2009. 7

[37] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009. 8

[38] J. Yang, K. Yu, and T. Huang. Efficient highly over-complete sparse coding using a mixture model. In *ECCV*, 2010. 1, 2

[39] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *IJCV*, 73:213–238, 2007. 1, 2