Pattern Recognition ■ (■■■■) ■■■–■■■

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

A coarse-to-fine approach for fast deformable object detection

Marco Pedersoli [a,*,1], Andrea Vedaldi [b], Jordi Gonzàlez [a], Xavier Roca [a]

[a] Computer Vision Center and Universitat Autònoma de Barcelona, Edifici O, Campus UAB, 08193 Bellatera, Spain
[b] Department of Engineering Science, Oxford University, Oxford OX1 3PJ, UK

ARTICLE INFO

Article history:
Received 13 November 2013
Received in revised form
17 October 2014
Accepted 8 November 2014

Keywords:
Object recognition
Object detection

ABSTRACT

We present a method that can dramatically accelerate object detection with part based models. The method is based on the observation that the cost of detection is likely dominated by the cost of matching each part to the image, and not by the cost of computing the optimal configuration of the parts as commonly assumed. To minimize the number of part-to-image comparisons we propose a multiple-resolutions hierarchical part-based model and a corresponding coarse-to-fine inference procedure that recursively eliminates from the search space unpromising part placements. The method yields a ten-fold speedup over the standard dynamic programming approach and, combined with the cascade-of-parts approach, a hundred-fold speedup in some cases. We evaluate our method extensively on the PASCAL VOC and INRIA datasets, demonstrating a very high increase in the detection speed with little degradation of the accuracy.

1. Introduction

In the last few years the interest of the object recognition community has moved from image classification and orderless models such as bag-of-words [1] to sophisticated representations that can explicitly account for the location, scale, and deformation of the objects [2–5]. By reasoning about geometry instead of discarding it, these models can extract a more detailed description of the image, including the object location, pose, and deformation, and can result in better detection accuracy.

A major obstacle in dealing with deformable objects is the combinatorial complexity of the inference. For instance, in the pictorial structures pioneered by Fischler and Elschlager [6] an object is represented as a collection of P parts, connected by springs. The time required to find the optimal part configuration to match a given image can be as high as the number L of possible part placements to the power of the number P of parts, *i.e.* $O(L^P)$. This cost can be reduced to $O(PL^2)$ or even $O(PL)$ by imposing further restrictions on the model ([2], Sections 2, 3.1), but is still significant due to the large number of possible part placements L. For instance, just to test for all possible translations of a part, L can be as large as the number of image pixels. This analysis, however, does not account for several aspects of typical part based models, such as the fact that useful object deformations are not very large and that, with appearance descriptors such as histograms of oriented gradients (HOG) [7], locations can be sampled in a relatively coarse manner.

The first contribution of this paper, an extension of our prior work [8,9], is a new analysis of the cost of part based models (Section 3.1) which better captures the bottlenecks of state-of-the-art implementations such as [7,3,10]. In particular, we show that the cost of inference is likely to be dominated by the cost of *matching each part to the image* rather than by the cost of determining the optimal part configuration. This suggests that accelerating inference requires minimizing the number of times the parts are matched.

Reducing the number of part evaluations can be obtained by using a *cascade* [11], a method that rejects quickly unpromising object hypotheses based on cheaper models. For deformable part models two different types of cascades have been proposed (Sections 2, 3.1). The first one, due to Felzenszwalb et al. [12], matches parts sequentially, comparing the partial scores to learned thresholds in order to reject object locations as soon as possible. The second one, due to Sapp et al. [13], filters the part locations by thresholding marginal part scores obtained from a lower resolution model.

The second contribution of the paper is a different cascade design (Section 3.2). Similar to [11,13], our method is coarse-to-fine. However, we note that, by thresholding scores independently, standard cascades propagate to the next level clusters of nearly identical hypotheses (as these tend to have similarly high scores). Instead of thresholding, we propose to reject all but the hypothesis whose score is *locally maximal*. This is motivated by the fact that looking for a locally optimal hypothesis at a coarse resolution often predicts well the best hypothesis at the next resolution level

* Corresponding author. Tel.: +32163 21095; fax: +32163 21723.
[1] Present address: KU Leuven, ESAT-PSI-VISICS/iMinds Kasteelpark Arenberg 10, 3001, Leuven, Belgium.
E-mail address: marco.pedersoli@esat.kuleuven.be (M. Pedersoli).

http://dx.doi.org/10.1016/j.patcog.2014.11.006
0031-3203/

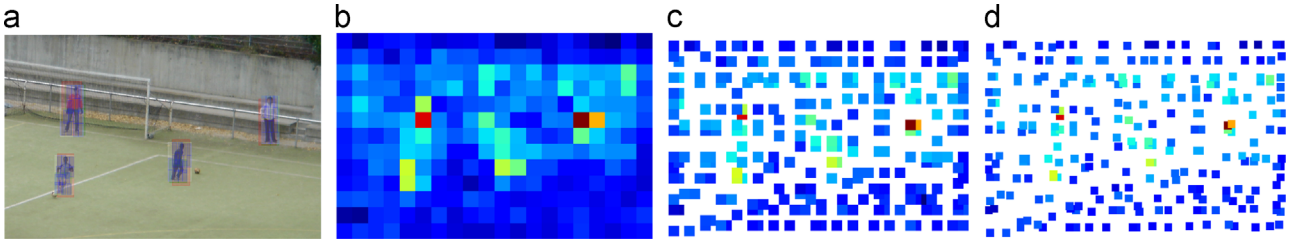

Fig. 1. Coarse-to-fine inference. We propose a method for the fast inference of multi-resolution part based models. (a) example detections; (b) scores obtained by matching the lowest resolution part (root filter) at all image locations; (c) scores obtained by matching the intermediate resolution parts, only at location selected based on the response of the root part; (d) scores obtained by matching the high resolution parts, only at locations selected based on the intermediate resolution scores. A white space indicates that the part is not matched at a certain image location, resulting in a computational saving. The saving *increases with the resolution*.

(Section 3.2). As suggested in Fig. 1, and as showed in Sections 3.2–3.4, this results in an *exponential saving, which has the additional benefit of being independent of the image content*. Experimentally, we show that this procedure can be more than ten times faster than the distance transform approach of [2,3], while still yielding excellent detection accuracy.

Compared to using global thresholds as in the cascade of parts approach of Felzenszwalb et al. [12], our method does not require fine tuning of the thresholds on a validation set. Thus it is possible to use it not just for *testing*, but also for *training* the object model, when the thresholds of the cascade are still undefined (Section 3.5). More importantly, the cascade of parts and our method are based on complementary ideas and can be combined, yielding a *multiplication the speed-up factors*. The combination of the two approaches can be more than two order of magnitude faster than the baseline dynamic programming inference algorithm [2] (Section 4).

2. Related work

In object category detection the goal is to identify and localize in images natural objects such as people, cars, and bicycles. Formally, we regard this as the problem of mapping an image $\mathbf{x}$ to a label or *interpretation* $\mathbf{y}$ that specifies whether an instance of the object is contained in the image and, if so, a bounding box enclosing it.

In order to simplify analysis as well as learning, the map $\mathbf{x} \mapsto \mathbf{y}$ is usually represented indirectly by a *scoring function* $S(\mathbf{x},\mathbf{y})$, expressing how well an interpretation $\mathbf{y}$ describes an image $\mathbf{x}$. The advantage is that the scoring function can have a simple form, often linear in a vector of parameters $\mathbf{w}$, *i.e.* $S(\mathbf{x},\mathbf{y}) = \langle \mathbf{w}, \Psi(\mathbf{x},\mathbf{y}) \rangle$. *Inferring* the interpretation $\mathbf{y}$ from the image $\mathbf{x}$ reduces then to finding which interpretations have a sufficiently large score, typically by computing the maximizer $\mathbf{y} = \arg\max_{\hat{\mathbf{y}} \in \mathcal{Y}} S(\mathbf{x},\hat{\mathbf{y}})$. Unfortunately, maximizing the scoring function is often computationally quite challenging. Next, we briefly review the main ideas that have been explored to address this issue.

Exhaustive and greedy search. If the interpretation space is sufficiently small, an inference algorithm can *score exhaustively* all interpretations $\mathbf{y} \in \mathcal{Y}$ and pick the best one. Sometimes this strategy can be applied even to continuous interpretations spaces up to discretization. A notable example are *sliding-window detectors* such as Dalal and Triggs [7]. A candidate interpretation $\mathbf{y}$ obtained from a discretized model can be further improved by a sequence of local greedy modifications, similar to gradient ascent. Unfortunately local search can easily get stuck in local optima. In less trivial cases, such as deformable part models, the interpretation space $\mathcal{Y}$ is far too complex for such simple strategies to suffice.

Sampling. By interpreting the score $S(\mathbf{x},\mathbf{y})$ as a posterior probability $p(\mathbf{y}|\mathbf{x})$ on the interpretations, inference can be reduced to the problem of drawing samples $\mathbf{y}$ from $p(\mathbf{y}|\mathbf{x})$ (because the most likely interpretations are also the ones with larger scores). Sampling ideas have been explored in the context of sliding-window object detectors in [14] demonstrating a two fold speed-ups over exhaustive search. Similar in spirit, but based on prior knowledge about the general shape of an object, are selective search [15] and objectness [16]. The main speed-up of these methods is again due to a reduced set of samples. However, in this case the samples are category independent (.i.e the same bounding boxes are used to represent different categories) so that the feature encoding can be computed only once for all categories.

Branch-and-bound. It is sometimes possible to compute efficiently upper bounds on the scores of large subsets $\mathcal{Y}' \subset \mathcal{Y}$ of interpretations at once. If a better interpretation is found somewhere else, then the whole subset $\mathcal{Y}'$ can then be removed without further consideration. *Branch-and-bound* methods apply this idea to a recursive partition of the interpretation space $\mathcal{Y}$. If the splits are balanced and the bounds sufficiently tight, these strategies can find the optimal interpretation very quickly. This idea has been popularized in the recent literature on sliding-window object detectors by Lampert and Blaschko [17].

Dynamic programming (*DP*). Sometimes interpretations are obtained by combining smaller interpretations of portions of the image. For example, in pictorial structures [6] an object is an arrangement $\mathbf{y} = (\mathbf{y}_1,\ldots,\mathbf{y}_N)$ of N object parts (*e.g.*, the head, torso, arms, and legs of a person), where $\mathbf{y}_i$ is the location of the corresponding part in the image. While there is a combinatorial number of such arrangements, in constellation models [18], the score decomposes as $S(\mathbf{y}_0,\mathbf{y}_1)+S(\mathbf{y}_0,\mathbf{y}_2)+\cdots+S(\mathbf{y}_0,\mathbf{y}_N)$, where $\mathbf{y}_0$ is a reference part connected in a star to the other parts. Hence the optimal arrangement can be obtained by finding the optimal position of each part $\arg\max_{\mathbf{y}_i} S(\mathbf{y}_0,\mathbf{y}_i)$ relative to the reference part $\mathbf{y}_0$, and then optimizing over the location of the reference. Efficient inference extends to more complex topologies such as trees and can be further improved under certain assumptions on the scores, yielding to the efficient pictorial structures of [2] (Section 3.1).

Cascades. A *cascade* considers cheaper scoring functions along with $S(\mathbf{x},\mathbf{y})$ and uses them to prune quickly unpromising interpretations $\mathbf{y}$ from consideration. Applied to an exhaustive search of the possible object locations, this yields the well-known cascade approach to sliding-window object detection [19]. The idea has been popularized by its application to AdaBoost [20–23] and has remained popular through the years, including applications to multiple kernels detectors [24]. The same idea has been applied directly to part-based models to either prune object locations by visiting only a small number of parts [12] or by finding plausible placements of the parts based on scoring functions with a lower degree of part dependencies [25] or lower resolution parameters [13]. Section 3.2 introduces an alternative coarse-to-fine cascade design. A more general analysis of other problems related with fast detection can be found in [26].

Recent methods. In parallel with the submission of this work and during the revision period several new methods for speeding

Please cite this article as: M. Pedersoli, et al., A coarse-to-fine approach for fast deformable object detection, Pattern Recognition (2014), http://dx.doi.org/10.1016/j.patcog.2014.11.006

up object detection have been presented. Dollar et al. [27] integrate the principle of locally maximal score introduced in this work for hypotheses rejection with a traditional cascade approach obtaining a noticeable gain in speed. Song et al. [28] represent the object parts of a deformable model as a sparse linear composition of a reduced set of basic parts shared among different categories. This produces an important speed-up in the convolution of the object model with the HOG features especially when dealing with several object categories. Dubout et al. [29] show how to speed-up the convolution of the object model with HOG features using the Fourier Transform. Dean et al. [30] propose a locality-sensitive hashing for the object search that can be several orders of magnitude faster than the standard HOG convolution, especially when dealing with a large number of object classes. Finally Sadeghi et al. [31] and Yan et al. [32] combine several of the previously mentioned techniques to obtain a complete deformable detector that can run at several frames per second. Further details and other relevant works will be given throughout the paper.

3. Our method

3.1. The cost of inference in deformable part models

This section studies the cost of inference in state-of-the-art models for object detection based on the notion of deformable parts, deriving key results that will be used in Section 3.2 to accelerate this process. A deformable part based model, or pictorial structure as introduced by Fischler and Elschlager [6], represents an object as collection of P parts arranged in a deformable configuration through elastic connections. Each part can be found at any of L discrete locations in the image. For instance, in order to account for all possible translations of a part, L is equal to the number of image pixels. If parts can also be scaled and rotated, L is further multiplied by the number of discrete scales and rotations, making it very large. Since even for the simplest topologies (trees) the best known algorithms for the inference of a part based model require $O(PL^2)$ operations, these models appear to be intractable. Fortunately, the distance transform technique of [2] can be used to reduce the complexity to $O(PL)$ under certain assumptions, making part models if not fast, at least practical.

The analysis so far represents the standard assessment of the speed of part based models, but it does not account for all the factors that contribute to the true cost inference. In particular, this analysis does not predict adequately the cost of state-of-the-art models such as [12] for the three reasons indicated next. First, the complexity $O(PL^2)$ reflects only the cost of finding the optimal configuration of the parts, ignoring the cost of matching each part to the image. Matching a part usually requires computing a local filter for each tested part placement. Filtering requires $O(D)$ operations where D is the dimension of the filter (this can be for instance a HOG descriptor [7] for the part). The overall cost of inference is then $O(P(LD+L^2))$. Second, depending on the quantization step δ of the underlying feature representation, parts may be placed only at a discrete set of locations which are significantly less than the number of image pixels L. For instance, [3] uses HOG features with a spatial quantization step of $\delta=8$ pixels, so that there are only L/δ^2 possible placements of a part. Third, in most cases it is sufficient to consider only *small deformations*[1] between parts. That is, for each placement of a part, only a fraction $1/c$ of placements of a sibling part are possible. All considered, the inference cost becomes

$$O\left(P\frac{L}{\delta^2}\left(D+\frac{L}{\delta^2 c}\right)\right). \tag{1}$$

Consider for example a typical pictorial structure of [3]. The part filters are composed of 6×6 HOG cells, so that each part filter has dimension $6\times 6\times 31=1116$ (where 31 is the dimension of a HOG cell). Typically the elastic connections between the parts deform by no more than 6 HOG cells in each direction. Thus the number of operations required for inferring the model is $(1116+36)PL/\delta^2$ where the first term reflects the cost of evaluating the filters, and the second the cost of searching for the best part configuration. Hence the cost of evaluating the part filters is $1116/36=31$ times larger than the cost of finding the optimal part configuration. The next section proposes a new method to reduce this cost.

3.2. Fast coarse-to-fine inference

This section proposes a new method based on a coarse-to-fine analysis of the image to speed-up detection by deformable part models. All the best performing part based models incorporate multiple resolutions [8,10]. Therefore it is natural to ask whether the multi-scale structure can be used not just for better modeling, but also to accelerate inference.

Multiple resolutions have been used in the design of a cascade for deformable part models by [13]. Here we propose an alternative design based on a principle different from global thresholding [8,9]. Consider the hierarchical part model of Fig. 2a,b similar to the one proposed by [10]. Our method starts by evaluating the root (coarser-resolution) filter at all image locations (Fig. 3). It then looks for the best placement of the root filter in, say, all 3×3 neighborhoods and propagates only this hypothesis to the next level. We call this procedure *Coarse-to-Fine* (CF) search.

The CF algorithm is justified by the fact that locally optimal placements of parts at a coarse resolution are often good predictors of the optimal part placements at the finer resolution levels. Fig. 2c shows the empirical probability that the CF procedure finds the same part locations as a globally optimal search procedure based on DP. As it can be seen, for detections with a threshold higher than -0.5 (which approximatively correspond to 80% recall), this probability is more than 70%, whereas suboptimal placements for hypotheses that have a small score are not detrimental to performance since those hypotheses would be discarded anyways. Section 4 gives more evidence of the validity of this assumption.

In order to estimate the cost of the CF search, start from the lowest resolution level $r=0$, corresponding to the root of the tree. Let this be a HOG filter of dimension $w\times h$, let L be the number of image pixels, and let δ the spatial quantization of the HOG features. Then there are L/δ^2 possible placements for the root part, evaluating which requires $Lwhd/\delta^2$ operations, where d is the dimension of a HOG cell.

At the second resolution level $r=1$, the resolution of the HOG features doubles, so that there are $4^rL/\delta^2$ possible placements of each part. Since each part is as large as the root filter and there are 4^r of those, matching all the parts requires $(4^rwhd)\times(4^rL/\delta^2)$ operations. The CF search avoids most of these computations by guiding the search based on the root filter. Specifically, of all the $4^rL/\delta^2$ placements of the root filter, we keep only the ones that have maximal response in neighborhoods of size $m\times m$, reducing the number of placements by a factor m^2. Then, for each placement of the root filter, the parts at the next resolution levels are also searched in $m\times m$ neighbors only, exploiting the fact that, in practice, deformations are bounded. Thus each higher resolution part is searched at only $m^2(L/m^2\delta^2)=L/\delta^2$ positions. Note that this is the *same number of evaluations of the root part, even though there*

[1] In sect. we will experimentally show that it is not necessary to search for deformation on the entire image. Instead, the region where to search is a fraction of the image size that for the moment we generally call c, but it will be better specified in the experimental results.

Please cite this article as: M. Pedersoli, et al., A coarse-to-fine approach for fast deformable object detection, Pattern Recognition (2014), http://dx.doi.org/10.1016/j.patcog.2014.11.006

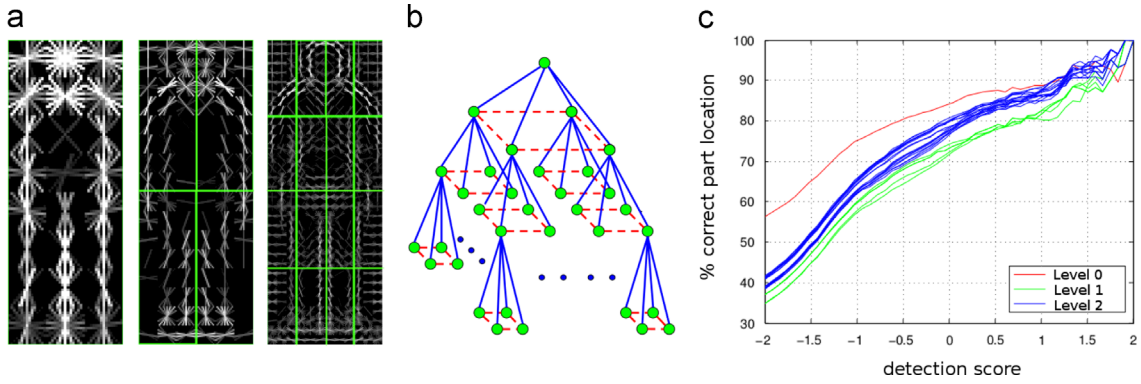

Fig. 2. Coarse-to-fine models and predictions. (a) The model is composed of a collection of HOG filters [7] at different resolutions. (b) The HOG filters form a parent–child hierarchy where connections control the relative displacement of the parts when the model is matched to an image (blue solid lines); additional sibling-to-sibling deformation constraints are enforced as well (red dashed lines). (c) Probability that the coarse-to-fine search results in exactly the same part locations as the globally optimal DP algorithm for each part of the hierarchical model. The probability is very high for highly scoring hypotheses (true positive) as desired. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

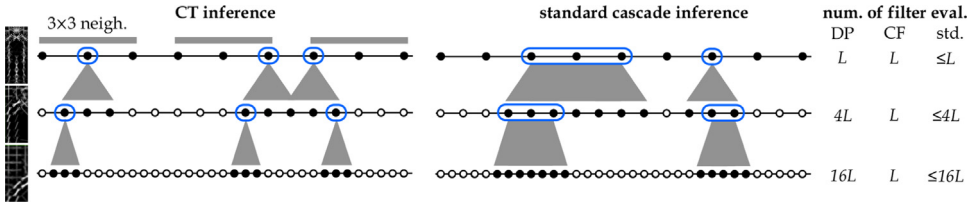

Fig. 3. Coarse-to-fine cascade designs. *Left.* Our proposed CF cascade starts by matching the coarse resolution part at a set of L discrete locations, here denoted by circles along one image axis. It then propagates to the next resolution level only the best hypotheses (marked by a rounded blue box) for each 3×3 neighborhood. As a result, parts are always evaluated at only L locations (filled circles) regardless of the resolution, yielding to a constant saving. *Right.* By contrast, a standard cascade such as [12] propagates all locations whose score is larger than a threshold (rounded blue box). This (i) tends to propagate clusters of neighbor hypothesis at once as these tend to have similar score and (ii) results in a saving that depends on the image content. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

are four times as many possible part locations at this resolution level. This is true for all the parts in the model, even the ones at higher resolutions.

Considering all levels together, the cost of evaluating naively all the part placements for the multi-resolution model is $(Lwhd)/\delta^2 \times (16^R-1)/15$ where R is the number of resolution levels in the model. The CF procedure reduces this cost to $(Lwhd)/\delta^2 \times (4^R-1)/3$. For instance, if there are $R=3$ levels the CF procedure is thirteen times faster than the standard DP approach, at least in terms of the effort required to match parts to the image.

Notice that, with this formulation, the cost is independent of m, which controls the size of the neighborhoods where parts are searched. However, in practice, we use a small value of m for the root part to avoid missing overlapping objects, and a larger one for the other resolution levels in order to accommodate larger deformations of the model which changes the expression of the cost slightly (Section 4). A more detailed analysis is presented in Sections 3.3 and 3.4.

Lateral connections. Weaknesses of the coarse-to-fine strategy can be compensated by enforcing additional geometric constraints among the parts. In particular, we add constraints among siblings, dubbed *lateral connections*, as shown in Fig. 2b (red dashed edges). This makes the motion of the siblings coherent and improves the robustness of the model. Fig. 4a,b demonstrates the importance of the lateral connections in learning a model of a human. Without lateral connections the model captures two separate human instances, but when the connections are added the model is learned properly (Section 3.4).

3.3. Object model

This section describes formally the model briefly introduced in Section 3.1. The model is a hierarchical variant of [3] (Fig. 2a,b) where parts are obtained by subdividing regularly and recursively parent parts. At the root level, there is only one part represented by a 31-dimensional HOG filter [12,7] of $w \times h$ cells. This is then subdivided into four subparts and the resolution of the HOG features is doubled, resulting in four $w \times h$ filters for the subparts. This construction is repeated to obtain sixteen parts at the next resolution level and so on. In practice, we use only three resolution levels in order to be able to detect small objects.

Let $\mathbf{y}_i$, $i=1,\ldots,P$ be the locations of the P object parts. Each $\mathbf{y}_i$ ranges in a discrete set $\mathcal{D}_i$ of locations (HOG cells), whose cardinality increases with the fourth power of the resolution level. Given an image $\mathbf{x}$, the score of the configuration $\mathbf{y}$ is a sum of appearance and deformation terms:

$$S(\mathbf{y};\mathbf{x},\mathbf{w}) = \sum_{i=1}^{P} S_{H_i}(\mathbf{y}_i;\mathbf{x},\mathbf{w}) + \sum_{(i,j)\in\mathcal{F}} S_{F_{ij}}(\mathbf{y}_i,\mathbf{y}_j;\mathbf{w}) + \sum_{(i,j)\in\mathcal{P}} S_{P_{ij}}(\mathbf{y}_i,\mathbf{y}_j;\mathbf{w})$$

where $\mathcal{F}$ are the parent–child edges (solid blue lines in Fig. 2c), $\mathcal{P}$ are the lateral connections (dashed red lines), and $\mathbf{w}$ is a vector of model parameters, to be estimated during training. The term S_{H_i} measures the compatibility between the image appearance at location $\mathbf{y}_i$ and the i-th part. This is given by the linear filter $S_{H_i}(\mathbf{y}_i;\mathbf{x},\mathbf{w}) = H(\mathbf{y}_i;\mathbf{x}) \cdot M_{H_i}(\mathbf{w})$ where $H(\mathbf{y}_i;\mathbf{x})$ is the $w \times h$ HOG descriptor extracted from the image $\mathbf{x}$ at location $\mathbf{y}_i$ and M_{H_i} extracts the portion of the parameter vector $\mathbf{w}$ that encodes the filter for the i-th part. The term $S_{F_{ij}}$ penalizes large deviations of the location $\mathbf{y}_j$ with respect to the location of its parent $\mathbf{y}_i$, which is one resolution level above. This is a quadratic cost of the type $S_{F_{ij}}(\mathbf{y}_i,\mathbf{y}_j;\mathbf{w}) = D(2\mathbf{y}_i,\mathbf{y}_j) \cdot M_{F_i}(\mathbf{w})$, where i is the parent of j, $M_{F_i}(\mathbf{w})$ extracts the deformation coefficients from the parameter vector $\mathbf{w}$, and $D(2\mathbf{y}_i,\mathbf{y}_j) = [(2x_i - x_j)^2, (2y_i - y_j)^2]$ where $\mathbf{y}_i = (x_i, y_i)$. The factor 2 maps the low resolution location of the parent $\mathbf{y}_i$ to the higher

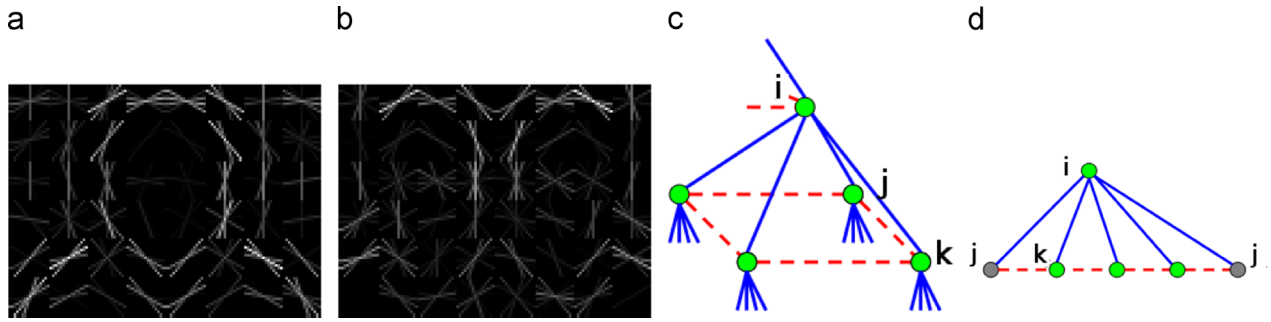

Fig. 4. Effect of lateral connections in learning a model: (a) Detail of a human model learned with lateral connections active. (b) The same model without lateral connections. *Inference on the lateral connections:* The loopy graph generated by the lateral connections is transformed into a chain by clamping the value $\mathbf{y}_i$ and then solved with DP.

resolution level of the child. Similarly, S_P penalizes sibling-to-sibling deformations and is given by $S_{P_{ij}}(\mathbf{y}_i, \mathbf{y}_j; \mathbf{w}) = D(\mathbf{y}_i, \mathbf{y}_j) \cdot M_{P_{ij}}(\mathbf{w})$. In this case the factor 2 is not used in D as sibling parts have the same resolution.

In addition to the quadratic deformation costs, the possible configurations are limited by a set of parent–child constraints of the form $\mathbf{y}_j \in \mathcal{C}_j + 2\mathbf{y}_i$. In particular, $\mathcal{C}_j + 2\mathbf{y}_i$ is a set of $m \times m$ small displacements around the parent location $2\mathbf{y}_i$. The parameter m, bounding the deformations, is discussed again in Section 3.4 in the analysis of the CF inference procedure, and its impact is evaluated in the experiments (Section 4).

As in [3,33] the model is further extended to multiple aspects in order to deal with large viewpoint variations. To this end, we stack N models $\mathbf{w}_1, \ldots, \mathbf{w}_N$, one for each aspect, into a new combined model $\mathbf{w}$. Then the inference selects both one of the n models and its configuration $\mathbf{y}$ by maximizing the score. Moreover, similar to [33], the model is extended to encode explicitly the symmetry of the aspects. Namely, each model $\mathbf{w}_k$ is tested twice, by mirroring it along the vertical axis, in order to detect the direction an object is facing.

3.4. DP and CF inference

This section analyses in detail inference with the model introduced in Section 3.3. If the hierarchical model does not have lateral connections, the structure is a tree and inference can be performed by using the standard DP technique. In detail, if part j is a leaf of the tree, let $V(\mathbf{y}_j) = S_{H_j}(\mathbf{y}_j)$, where we dropped for compactness the dependency of the score on $\mathbf{w}$ and $\mathbf{x}$. For any other part i define recursively $V(\mathbf{y}_i) = S_{H_i}(\mathbf{y}_i) + \sum_{j:\pi(j)=i} \max_{\mathbf{y}_j \in \mathcal{C}_j + 2\mathbf{y}_i} (S_{F_{ij}}(\mathbf{y}_i, \mathbf{y}_j) + V(\mathbf{y}_j))$ where $\mathbf{y}_j \in \mathcal{D}_j$ and $i = \pi(j)$ implies that i is the parent of j. Computing $V(\mathbf{y}_i)$ requires

$$|\mathcal{D}_i| \left(D + \sum_{j:\pi(j)=i} |\mathcal{C}_j| \right) \tag{2}$$

operations, where D is the dimension of a part filter and $\mathcal{C}_j$ is the set of allowable deformations given in Section 3.3. The terms $|\mathcal{C}_i|$ in the cost can be reduced to one by using the distance transform of [2], but the saving is small since $|\mathcal{C}_i|$ is small to start with. Most importantly, the distance transform is advantageous only in the case parts are tested at all locations, which is incompatible with the use of a cascade.

DP with lateral connections. The lateral connections in Fig. 4c introduce cycles and prevent a direct application of DP. However, these connections form pyramid-like structures (Fig. 4c) that can be "opened" by clamping the value of one of the base nodes (Fig. 4d). In particular, denote with i the parent node, j the child being clamped, and k the other children. Then the cost of computing the function $V(\mathbf{y}_i)$ becomes

$$|\mathcal{D}_i| \left(D + |\mathcal{C}_j| \sum_{k:\pi(k)=i, k \neq j} |\mathcal{C}_k| \right), \tag{3}$$

which is slightly higher than (2) but still quite manageable due to the small size of $\mathcal{C}_i$.

CF inference. Despite the increased complexity of the geometry of a model with lateral connections, the cost of inference is still dominated by the cost of evaluating each part filter to each image location. This cost cannot be reduced by DP; instead, we propose to prune the search top-down, by starting the inference from the root filter and propagating only the solutions which are locally the more promising. Note that, instead of using a fixed threshold to discard partial detections as done by the part based cascade [12], here pruning is performed locally and adaptively. We now describe the process in detail, and estimate its cost.

First, the root part is tested everywhere in the image, with cost $|\mathcal{D}_0|D$. Note that, since the root part resolution is coarse, $|\mathcal{D}_0|$ is relatively small. Then non-maxima suppression is run on neighbors of size $m \times m$, leaving only $|\mathcal{D}_0|/m^2$ possible placements of the root part. For each placement of the root $\mathbf{y}_0$, the parts k at the level below are searched at locations $\mathbf{y}_k \in C_k + 2\mathbf{y}_0$, which costs

$$\frac{|\mathcal{D}_0|}{m^2} \left(\sum_{k:\pi(k)=0} |\mathcal{C}_k| D + |\mathcal{C}_i| \sum_{k:\pi(k)=0, k \neq i} |\mathcal{C}_k| \right)$$

where i is the child clamped, as explained above, in order to account for the sibling connections. The dominant cost is matching the parts at $|\mathcal{D}_0|\ |\mathcal{C}_k|/m^2$ locations (if filters are memoized [12] the actual cost is a little smaller due to the fact that the same part location can be obtained from more than one root hypothesis). The process is repeated recursively, by selecting the optimum placement of each part at resolution r and using it to constrain the placement of the parts at the next resolution level $r+1$. In this way *each part is matched at most* $|\mathcal{D}_0|\ |\mathcal{C}_k|/m^2$ *times*, where $|\mathcal{C}_k|$ can be chosen equal or similar to m^2. This should be compared to the $|\mathcal{D}_k|$ comparisons of the DP approach, which grows with the fourth power of the resolution. Hence the computational saving becomes significant very quickly.

Note that, while each part location is determined by ignoring the higher resolution levels, the sibling constraints help integrating evidence from a large portion of the image and improve the localization of the parts.

Extension: CF and cascade of parts. The CF cascade can easily integrate global rejection thresholds analogous to the cascade of parts of Felzenszwalb et al. [12] resulting in a multiplication of the speed-up factors of our and their technique. In detail, one can learn thresholds to prune an object hypothesis based on the partial scores obtained by evaluating only a subset of the parts. In the experiments, a simplified version of this idea will be tested where pruning is applied after all parts at a given resolution levels have

Please cite this article as: M. Pedersoli, et al., A coarse-to-fine approach for fast deformable object detection, Pattern Recognition (2014), http://dx.doi.org/10.1016/j.patcog.2014.11.006

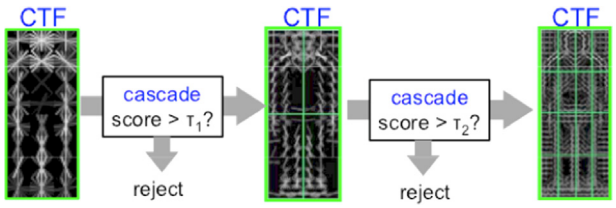

Fig. 5. Combining CF with a cascade of parts. The score at each resolution level is determined by using the fast CF inference procedure. As soon as the score up to a certain resolution level has been computed, this is compared to a threshold to discard unpromising object locations quickly. The threshold is learned on a validation set as [12].

been evaluated. We call this CF+cascade, summarize it in Fig. 5, and report its empirical performance in Section 4.

Extension: CF and DP. CF can be used as a pre-filtering step to run the standard DP algorithm at a subset of promising image locations, obtaining almost always globally optimal solutions at a fraction of the cost. In more detail, the idea is to first estimate a small set of candidate object locations using CF, and then computing the exact part placements, and hence the exact detection scores, using DP only at those locations. Since CF estimates correctly the object locations in the vast majority of the cases and since its computed scores are fairly good by themselves, retaining up to a hundred object hypothesis per image is sufficient to reconstruct the output of the globally optimal DP nearly exactly. This idea is evaluated in Section 4.

3.5. Learning

This section describes in detail the learning of the model introduced in Section 3.3 and how to leverage on the fast inference methods of Section 3.4 to do so. Learning is needed to obtain the parameters $\mathbf{w}$ of the scoring function. This uses a variant of the latent structural support vector machine (SVM) formulation of [34,33], which is also very similar to the latent SVM method of [3].

Training uses a dataset of images and the corresponding bounding box annotations for an object category of interest. Each object bounding box is initially associated with the best matching location and scale $\mathbf{y}$ for the model. This is defined as the location $\mathbf{y}$ in the HOG coordinate space for which the root filter yields maximal intersection-over-union overlap score with the object bounding box. If there are multiple model components, one for each object aspect, the one with best overlap score is selected. This defines a set of positive examples $(\mathbf{x}_i, \mathbf{y}_i)$, $i \in P$, one for each object bounding box, where $\mathbf{x}_i$ denotes the corresponding image. All the other locations that yield an overlap score of less than a threshold T with all the object bounding boxes are used as negative examples $(\mathbf{x}_i, \mathbf{y}_i)$, $i \in N$ (in the case of the CF inference, one negative per root-level neighborhood is generated instead). Note that different $\mathbf{x}_i$ can refer to the same image as detections at different locations are considered independent by learning.

From this construction, one obtains a number of negative samples far larger than the positive ones $|N| \gg |P|$, so that the data is highly unbalanced. Nevertheless, this was not found to be a problem in learning. This is due to the fact that, for the purpose of ranking, only the relative scores are important. The imbalance may result in scores that are not perfectly calibrated for binary classification, but this does not affect ranking.

Note that the ground-truth locations $\mathbf{y}_i$ are effectively unknown and the procedure just described simply suggests an initial value. During training these are consider latent variables and gradually re-estimated. Training itself optimizes the latent SVM objective function

$$E(\mathbf{w}; \{\mathbf{y}_i, i \in P\}) = \frac{\|\mathbf{w}\|^2}{2} + C \sum_{i \in P} \max\{0, 1 - S(\mathbf{y}_i; \mathbf{x}_i, \mathbf{w})\} + C \sum_{j \in N} \max\left\{0, 1 + \max_{\mathbf{y}_j} S(\mathbf{y}_j, \mathbf{x}_i, \mathbf{w})\right\}. \quad (4)$$

This trades off the usual quadratic regularizer $\|\mathbf{w}\|^2$ with a hinge-loss term for the positive samples, encouraging their score to be above 1 (the margin), and a corresponding term for the negative samples, encouraging their scores to be below -1. Note that the object location and pose $\mathbf{y}_j$ is maxed-out in the negative terms. This is possible without compromising convexity [3,34]; on the other hand, the pose parameters $\mathbf{y}_i$, $i \in P$ must be kept constant as $\mathbf{w}$ is determined to make the energy convex. Subsequently, $\mathbf{w}$ is fixed and these parameters are re-estimated by maximizing $S(\mathbf{y}_i; \mathbf{x}_i, \mathbf{w})$ and the procedure is repeated. This is known as the Concave–Convex Procedure (CCCP) and is only guarantee to find a locally optimum solution [3].

Updating the latent variables. When the latent variables $\mathbf{y}_i \in P \cup N$ are optimized, the corresponding object locations are searched in a neighborhood of their initial values. In particular, for the negative examples the object location is kept fixed (or within a root-level neighborhood with CF) while the part locations are re-estimated. This is because the goal is to have in the energy function one negative example for each candidate image location. For the positive variables $\mathbf{y}_i$ instead, the object location is adjusted in order to better align the model to the corresponding object instance. This also means that in rare cases there might be no location that, after the locations of the parts have been re-estimated, still fits the object bounding box, or that the one that does has lower score than the current setting of $\mathbf{y}_i$. This is handled below, accounting for the approximation due to the CF inference as well.

Constraint generation. The negative samples N are too many to be extracted and stored in memory. Instead, one starts with an empty set of negatives $N = \phi$ and then iteratively re-estimates $\mathbf{w}$ and searches the dataset for a batch of fresh examples N that are in margin violation (*i.e.* whose score is larger than -1), updates the model, and repeats. This procedure, which is equivalent to constraint generation [35] or mining of hard negatives [3], is guarantee to end in polynomial time provided that the set of support vectors (*i.e.* the examples violating the margin at the optimum) can fit in memory.

Using CF inference in training. Inference is used during training for two purposes: to estimate the optimal part layout $\mathbf{y}_i$, $i \in P$ for the example object instances (CCCP) and to obtain the most confusing part layout $\mathbf{y}_j$, $j \in N$ for the negative examples. The accelerated CF inference can be used to do this because, contrary to the part based cascade of [12], it does not have parameters to be learned. This fact can be used to substantially accelerate training too (see Section 4 Table 1).

Table 1
Learning and testing a model with exact and coarse-to-fine inference. The table compares learning the model without lateral connection (S_F) and with lateral connections ($S_F + S_P$) and testing it with the exact (DP) or coarse-to-fine (CF) inference algorithm. For each case, training base on the DP or CF inference is also compared.

Model	Training		Testing AP (%)	
	Method	Time (h)	DP	CF
S_F	DP	20	83.0	84.0
$S_F + S_P$	DP	22	83.4	84.0
S_F	CF	1.9	78.0	80.7
$S_F + S_P$	CF	2.2	83.5	83.5

While the CF inference has been found empirically to be quite reliable, it still returns approximated maximizers of the scoring function. From the viewpoint of the constraint generation procedure (hard negatives), this means that CF might not find all the hard negative constraints that could be determined by a globally optimal algorithm such as DP. However, such hard negatives are unlikely to be found at test time as well, so this slight relaxation of the constraints is not a problem. The estimation of the part locations for the positive examples $\mathbf{y}_i$ is more delicate as sub-optimal inference may cause the energy function to increase rather than decrease, compromising the stability of the algorithm. This problem is easily sidestepped by comparing the energy of the latent variable before and after update and retaining the latent variable configuration with lower energy.

4. Experiments

This section evaluates our method on three well known benchmarks: the INRIA pedestrians [7] and the 20 PASCAL VOC 2007 and 2009 object categories [36]. Performance is measured in terms of false positive per window (FPPI) and Average Precision (AP) according to the PASCAL VOC protocol [36]. For the VOC 2007 classes we use an object model with two components (aspects), for the VOC 2009, we use three components, while for the INRIA pedestrians we use a single one as using more did not help. The aspect ratio of each component is initialized by subdividing uniformly the aspects ratio of the training bounding boxes and taking the average in each interval.

4.1. INRIA pedestrians

Fig. 6-left compares different variants of our coarse-to-fine (CF) detector with the part based cascade of [12] by evaluating the average detection time and precision for the INRIA pedestrian dataset. Our CF search algorithm is slightly slower than the part based cascade (0.33 s vs 0.23 s per image). However, the two methods are orthogonal and can be combined to further reduce the detection time to 0.12 s, with just a marginal decrease in the detection accuracy as suggested in Section 3.2.

Fig. 6-right compares the CF detector with other published methods in terms of miss rate vs false positives per image (FPPI) rate. The CF detector obtains a detection rate of 88% at 1 FPPI, which is just a few points lower than the current state-of-the-art (91%), but uses only HOG features. In particular, due to the deformable parts and the CF inference, our detection rate is 10% better that the standard HOG detector while being much faster.

Effect of the neighborhood size m. Fig. 6-left evaluates the influence of the neighborhood size m, which controls the amount of deformation that the model allows. Compared to Section 3.2 in which the same m is chosen at all resolution levels, here this parameter is fixed to $m=3$ for the coarser resolution and changed in the range $m=3,5,7$ for the higher resolutions, to evaluate absorbing larger deformations while still being able to detect multiple close instances of the objects. While inference slows down by increasing the deformation range m, this is unnecessary as the detection performance saturates at $m=3$. Larger deformations do not change substantially the detection performance for this model, but greatly affect the inference time, which increases from 0.33 s per image for $m=3$ to almost 10 s for $m=7$.

method	time (s)	AP (%)
cascade [12]	0.23	85.6
CF	0.25	78.8
CF + siblings	0.33	84.0
CF + sib. + casc.	0.12	83.6
$m=3$	0.33	83.5
$m=5$	2.0	83.2
$m=7$	9.3	83.6

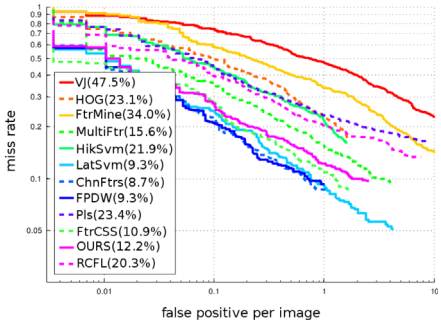

Fig. 6. Left: evaluation on the INRIA dataset. *Cascade* denotes the part based cascade of [12]. *CF*, *CF+sibling*, and *CF+sib.+casc.* denote our coarse-to-fine inference scheme trained with DP and tested respectively without sibling constraints, with sibling constraints, and combined with the cascade of [12]. Effect of the neighborhood size m on the INRIA Pedestrian dataset using CF inference. Setting m to 3 is sufficient to obtain optimal performance. Increasing the value of m does not change substantially the AP, but has a negative impact on speed. Right: Comparison to the state-of-the-art. The miss rate at 1 FPPI is reported in the legend. VJ [20], HOG [7], FtrMine [37], MultiFtr [38], HikSvm [39], LatSvm [3], ChnFtrs [40], FPDW [41], Pls [42], MultiFtr+CSS [43], RCFL [8].

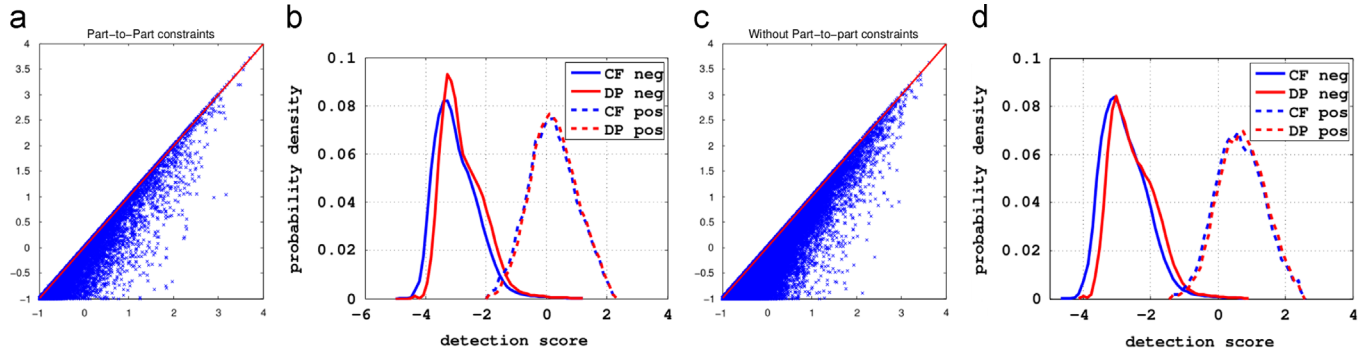

Fig. 7. Exact vs coarse-to-fine inference scores. Scatter plot (a,c) and score distributions (b,d) of the scores obtained by the exact (DP) and approximated (CF) inference algorithms: (a,b) with lateral constraints in the model, (c,d) without. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

Please cite this article as: M. Pedersoli, et al., A coarse-to-fine approach for fast deformable object detection, Pattern Recognition (2014), http://dx.doi.org/10.1016/j.patcog.2014.11.006

Table 2
Detection AP and speed on the PASCAL VOC 2007 test data. Our method has similar accuracy than other state-of-the-art methods but much faster, both in training and test.

	Plane	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table
BOW [24]	37.6	47.8	15.3	15.3	21.9	50.7	50.6	30.0	17.3	33.0	22.5
PS [3]	29.0	54.6	0.6	13.4	26.2	39.4	46.4	16.1	16.3	16.5	24.5
Hierarc. [10]	29.4	55.8	9.40	14.3	28.6	44.0	51.3	21.3	20.0	19.3	10.3
Cascade [12]	22.8	49.4	10.6	12.9	27.1	47.4	50.2	18.8	15.7	23.6	10.3
CF	27.9	54.8	10.2	16.1	16.2	49.7	48.3	17.5	17.2	26.4	21.4
	Dog	Horse	Mbike	Person	Plant	Sheep	Sofa	Train	Tv	**Mean**	**Time**
BOW [24]	21.5	51.2	45.5	23.3	12.4	23.9	28.5	45.3	48.5	32.1	≈70
PS [3]	5.0	43.6	37.8	35.0	8.8	17.3	21.6	34.0	39.0	26.8	≈10
Hierarc. [10]	12.5	50.4	38.4	36.6	15.1	19.7	25.1	36.8	39.3	29.6	≈8
Cascade [12]	12.1	36.4	37.1	37.2	13.2	22.6	22.9	34.7	40.0	27.3	<1
CF	11.4	55.7	42.2	30.7	11.4	20.9	29.1	41.5	30.0	28.9	<1

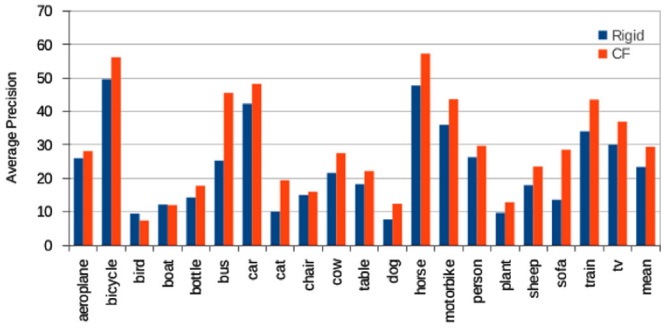

Fig. 8. Performance of Rigid and Deformable models with CF inference on the PASCAL VOC 2007. The figure reports the average precision obtained for the 20 classes by the two models.

This is probably due to two reasons. First, pedestrians are relatively rigid compared to humans in general pose. Second, although a deformation of one HOG cell in each direction for with respect to a part rest position ($m=3$) may seem small, the actual amount of deformation must be assessed in relation of the size of the root filter. If the root filter is three HOG cells wide as in our setting, then a deformation of one HOG cell corresponds to a displacement that is as large as 33% of the object size, which is substantial.

Exact and CF detection scores. Fig. 7 shows a scatter plot of the detection scores obtained on the test set of the INRIA database, where the horizontal axis reports the scores obtained by DP (exact inference) and the vertical axis the scores obtained by the CF inference algorithm. The red line represents the ideal case, where the CF inference gives exactly the same results as DP. We distinguish two cases for the analysis: (a) with lateral constraints and (c) without lateral constraints. We note two facts: First, in both cases the CF approximation improves as the detection score increases. This is reasonable because, if the object is easily recognizable, the local information drives the placement of the parts to optimal locations without much ambiguity. Second, in (a) the scatter plot is tighter than in (c), indicating that the lateral connections are in fact helping the CF inference to stay close to the ideal DP case. The same can be observed from the score distribution (b) and (d).

Training speed and detection accuracy. Table 1 evaluates the effect of using the CF and or the exact (DP) inference methods for training and testing the model. Using the CF inference method instead of the exact DP inference improves the training speed by an order of magnitude, from 20 h down to just 2. This is because the cost of training is dominated by the iterative re-estimation of the latent variables and retraining, each of which requires running inference multiple times. Note that, differently from [12] which requires tuning *after* the model has been learned, our method can be applied *while* the model is learned.

Table 3
Comparison of inference methods on PASCAL VOC 07. Models are trained using either *Exact* inference or *CF* inference and evaluated using either *Exact* inference, *CF+Ex(100,10)* where the best 100 or 10 hypothesis of CF inference are refined using exact inference, and *CF* inference.

Train	Exact				CF			
test	Exact	CF+Ex (100)	CF+Ex (10)	CF	Exact	CF+Ex (100)	CF10)	CF
Plane	32.2	32.2	32.6	28.1	29.8	30.2	30.8	27.9
Bicycle	58.4	58.1	54.5	56.2	58.6	58.4	54.2	54.4
Bird	10.7	10.7	10.7	7.4	6.4	6.5	6.5	10.2
Boat	13.9	14.1	12.5	12	16	15.9	15.5	16.1
Bottle	19.0	19.1	17.8	17.8	16.3	16.4	14.2	16.2
Bus	49.8	50.0	49.1	45.5	52.6	52.5	51.1	49.7
Car	52.0	51.7	49.1	48.2	51.2	50.8	49.4	48.3
Cat	23.1	23.1	22.0	19.5	17.1	17.0	18.1	17.5
Chair	20.3	19.3	17.7	16.0	19.2	19.2	17.2	17.2
Cow	29.4	29.7	28.3	27.5	28.6	28.2	28.2	26.4
Table	29.3	29.2	28.3	22.2	23.6	24.3	24.6	21.4
Dog	13.5	13.5	13.7	12.4	12.0	12.0	12.7	11.4
Horse	59.6	59.3	57.8	57.3	57.7	57.7	56.3	55.7
Mbike	44.5	44.3	43.2	43.6	43.1	43.0	42.5	42.2
Person	29.7	29.5	26.4	29.7	31.7	31.6	28.3	30.7
Plant	12.9	12.2	12.5	12.9	12.4	12.4	12.4	11.4
Sheep	26.2	26.2	26.1	23.5	25.2	25.1	23.8	20.9
Sofa	29.6	29.8	28.5	28.5	26.2	28.0	27.9	29.1
Train	44.0	44.2	45.2	43.5	43.0	43.2	44.0	41.5
Tv	39.2	39.5	39.4	36.9	36.8	36.6	35.6	30.0
Mean	31.9	31.8	30.8	29.4	30.4	30.5	29.7	28.9
HOG (M)	66.5	8.12	5.75	4.72	66.5	8.12	5.75	4.72
Speed-Up	1.0	8.1	11.6	14.1	1.0	8.1	11.6	14.1

A notable result from Table 1 is the fact that, for each training method (exact DP or CF) and model type (with or without lateral constraints), the accuracy never decreases, and in fact increases slightly, when the exact test procedure (DP) is substituted with the CF inference algorithm. This is probably due to the aggressive hypothesis pruning of the CF search which promotes less ambiguous detections. A second observation is that the lateral constraints are very effective and increase the AP by about 4–5% when using CF inference.

4.2. PASCAL VOC

We compare our CF model with state-of-the-arts methods on VOC 2007 using the variant with sibling constraints. Table 2 shows that the classification accuracy of the CF detector is similar to the one of state-of-the-art methods which are about an order of magnitude or more slower. The CF detector is also compared to the part-based cascade of [12], which has a similar speed. However, the results reported in [12] are generated from detectors

Please cite this article as: M. Pedersoli, et al., A coarse-to-fine approach for fast deformable object detection, Pattern Recognition (2014), http://dx.doi.org/10.1016/j.patcog.2014.11.006

trained on the VOC 2009 data, which contains twice as many training images as found in the VOC 2007 data. Note that, as explained in Section 3.5, our results are obtained using the fast CF inference during training too, reducing the training process for each class to few hours.

Rigid vs. deformable model. Fig. 8 compares a rigid and a deformable model both using CF inference. The rigid model (*rigid CF*) is a simplified version of our deformable model, where each model resolution is a rigid block without moving parts. This model is very similar to the one presented in [8]. The gain obtained by the deformable model is around 6 AP points. This shows that the increment in the model complexity due to the introduction of local deformations is worth.

CF, DP, and their combination. Table 3 evaluates the different inference methods on the PASCAL VOC 2007 data on top of models trained for each class using the exact DP inference procedure. Therefore approximations are for now factored out during training.

The most accurate detections are obtained by *Exact* (DP) inference which obtains a mAP close to 32 points. This is very close to the state-of-the-art, and probably equivalent since it does not use any post-processing such as contextual models or bounding box refinement [3]. The row labelled *HOG*(*M*) reports the number of HOG cells (in millions) that are involved in a filtering operation during inference, as this dominates the cost of inference and in fact is shown here to correlate very well with the speed of each method. The speed of DP is used as reference and speed-ups are expressed relative to it (so DP has a speed-up of 1.0).

Using CF inference the number of HOG cells that enter filtering is reduced from 66 millions to less than 5 millions, with a corresponding speed-up of more than one order of magnitude. However, the mAP decreases slightly. A trade-off between exact and approximate inference is given by the combination of CF and DP (labelled *CF+Ex*), as described in Section 3.2. Applying DP to the best 100 or 10 best hypotheses selected by CF strategy results in nearly optimal accuracy and a speed-up factor of either 8 or 11 times compared to standard DP.

CF and cascade of parts This paragraph evaluates the combination of our CF inference with a threshold-based filtering, as explained in Section 3.2. In order to simplify the visualization of the results, we set the two thresholds $\tau_1 = \tau_2 = \tau$. Setting independently the optimal value of the two thresholds can further improve the speed-up. For all VOC classes we draw the trade-off between detection speed (taking as reference exact inference computed using DP) and Average Precision (AP) achieved by varying τ.

For classes with high AP (Fig. 9a), a speed-up of more than two orders of magnitude with marginal decrease in detection accuracy is obtained. For classes with moderate AP (Fig. 9b), the speed-up achievable before the AP is reduced noticeably is smaller, but often above 100-fold. For classes with low AP (Fig. 9c), the speed-up is limited because the detection accuracy decreases abruptly as more solutions are pruned. Overall, this analysis indicates that by increasing the quality of the detector we can also expect a higher margin of gain in speed.

VOC 2009. Table 4 evaluates the CF inference on the PASCAL VOC 2009 [36]. The conclusions are analogous to the 2007 data in terms of speed-up and overall accuracy. While the PASCAL challenge results do not include the method speeds, a simple analysis of the HOG-based methods *UOCTTI* and *MIZZOU* suggests that their complexity is at least an order larger than ours.

5. Summary

We have presented a method that can substantially speed-up object detectors that use multi-resolution deformable part models.

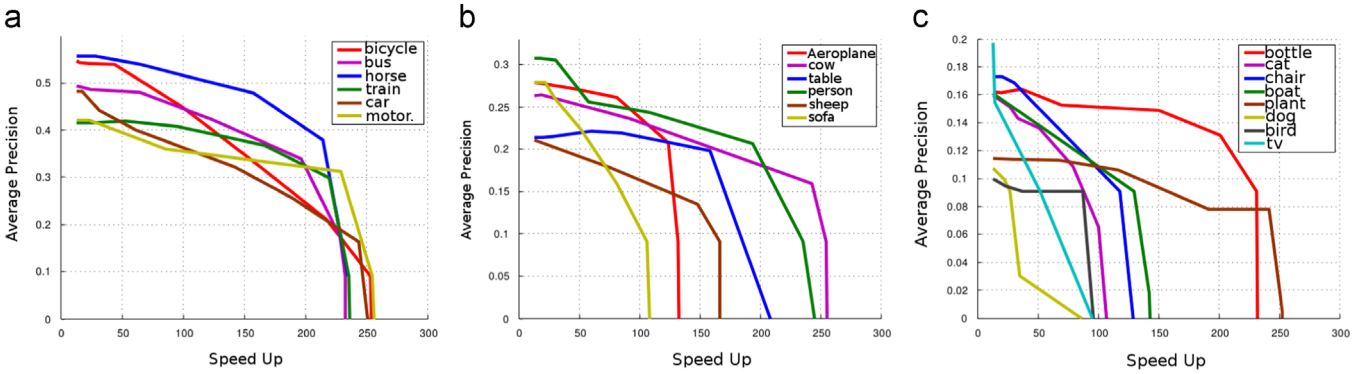

Fig. 9. Speed-Up vs AP. for classes with high AP. The figure reports the average precision vs speed-up (over the exact DP inference algorithm) for the CF detector combined with a pruning cascade on PASCAL VOC 2007 for classes with: high AP (a), medium AP (b), and low AP (c).

Table 4
Detection AP on the PASCAL VOC 2009 test data. We compare our method with the official results of the PASCAL VOC 2009 [36]. Using much less computation in both training and test, CF inference achieves results comparable to the state-of-the-art.

	Plane	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table
OXFORD	47.8	39.8	17.4	15.8	21.9	42.9	27.7	30.5	14.6	20.6	22.3
UOCTTI	39.5	46.8	13.5	15.0	28.5	43.8	37.2	20.7	14.9	22.8	8.7
MIZZOU	11.4	27.5	6.0	11.1	27.0	38.8	33.7	25.2	15.0	14.4	16.9
CF	41.3	45.5	10.9	13.6	18.3	44.0	33.3	24.2	11.7	19.1	14.9
CF+Ex(100)	41.5	46.6	11.5	15.3	20.0	44.3	35.9	23.9	13.1	20.7	15.9
	Dog	Horse	Mbike	Person	Plant	Sheep	Sofa	Train	Tv	**Mean**	
OXFORD	17.0	34.6	43.7	21.6	10.2	25.1	16.6	46.3	37.6	27.7	
UOCTTI	14.4	38.0	42.0	41.5	12.6	24.2	15.8	43.9	33.5	27.9	
MIZZOU	15.1	36.3	40.9	37.0	13.2	22.8	9.6	3.5	32.1	21.9	
CF	12.4	37.2	42.5	22.1	10.3	20.6	18.3	39.4	31.8	25.6	
CF+Ex(100)	13.4	40.4	44.1	22.4	10.7	23.4	21.9	43.4	34.3	27.1	

Please cite this article as: M. Pedersoli, et al., A coarse-to-fine approach for fast deformable object detection, Pattern Recognition (2014), http://dx.doi.org/10.1016/j.patcog.2014.11.006

This method uses a coarse-to-fine inference procedure to dramatically reduce the cost of matching object parts to the image, which dominates the cost of inference in most detectors. Compared to other speed-up techniques [12], this method does not require the learning of thresholds or other parameters, which simplifies its use during the training of the model, results in a constant speed-up regardless of the image content, and can be combined with the deformable part cascade [12] multiplying the speed-up factors. Finally, we have evaluated the coarse-to-fine search with DP rescoring, resulting in performance nearly identical to the full DP model at a fraction of the cost.

Conflict of interest

None declared.

Acknowledgments

We gratefully acknowledge J. M. Gonfaus and A. Zisserman for their suggestions and comments. Andrea Vedaldi was supported by the EU Project FP6 VIDI-Video IST-04554, ONR MURI N00014-07- 1-0182, and the Violette and Samuel Glasstone Research Fellowships in Science. The other authors would acknowledge the support of the Spanish Research Programs Consolider-Ingenio 2010: MIPRCV (CSD200700018); Avanza I+D ViCoMo (TSI-020400-2009-133); along with the Spanish projects TIN2009-14501-C02-01, TIN2009-14501-C02-02 and TIN2012-39051; and the EU Project FP7 AXES ICT- 269980.

References

[1] G. Csurka, C.R. Dance, L. Dan, J. Willamowski, C. Bray, Visual categorization with bags of keypoints, in: Proceedings of ECCV Workshop on Statistical Learning in Computer Vision, 2004, pp. 1–22.
[2] P. Felzenszwalb, D. Huttenlocher, Pictorial structures for object recognition, Int. J. Comput. Vis. 61 (1) (2004) 55–79.
[3] P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part based models, IEEE Trans. Pattern Anal. Mach. Intell. 32 (9) (2010) 1627–1645.
[4] X. Yang, H. Liu, L.J. Latecki, Contour-based object detection as dominant set computation, Pattern Recognit. 45 (5) (2012) 1927–1936.
[5] D.J. Kang, J.-E. Ha, I.-S. Kweon, Fast object recognition using dynamic programming from combination of salient line groups, Pattern Recognit. 36 (1) (2003) 79–90.
[6] M.A. Fischler, R.A. Elschlager, The representation and matching of pictorial structures, in: IEEE Trans. Comput., vol. 22, 1973.
[7] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proceedings of CVPR, 2005, pp. 886–893.
[8] M. Pedersoli, J. Gonzàlez, A. D. Bagdanov, J. J. Villanueva, Recursive coarse-to-fine localization for fast object detection, in: ECCV, 2010, pp. 280–293.
[9] M. Pedersoli, A. Vedaldi, J. Gonzàlez, A coarse-to-fine approach for fast deformable object detection, in: Proceedings of CVPR, 2011, pp. 1353–1360.
[10] L. Zhu, Y. Chen, A. Yuille, W. Freeman, Latent hierarchical structural learning for object detection, in: CVPR, 2010, pp. 1062–1069.
[11] S. Geman, K. Manbeck, D. E. McClure, Coarse-to-fine search and rank-sum statistics in object recognition, Technical report, Brown University, 1995.
[12] P. Felzenszwalb, R. Girshick, D. McAllester, Cascade object detection with deformable part models, in: CVPR, 2010, pp. 2241–2248.
[13] B. Sapp, A. Toshev, B. Taskar, Cascaded models for articulated pose estimation, in: Proceedings of ECCV, 2010, pp. 1–8.
[14] G. Gualdi, A. Prati, R. Cucchiara, Multi-stage sampling with boosting cascades for pedestrian detection in images and videos, in: Proceedings of ECCV, 2010, pp. 196–209.
[15] J. Uijlings, K. van de Sande, T. Gevers, A. Smeulders, Selective search for object recognition, Int. J. Comput. Vis. 104 (2) (2013) 154–171.
[16] B. Alexe, T. Deselaers, V. Ferrari, Measuring the objectness of image windows, IEEE Trans. Pattern Anal. Mach. Int. 34 (11) (2012) 2189–2202.
[17] C.H. Lampert, M.B. Blaschko, T. Hofmann, Beyond sliding windows: object localization by efficient subwindow search, in: Proceedings of CVPR, 2008, pp. 1–8.
[18] M. Weber, M. Welling, P. Perona, Unsupervised learning of models for recognition, in: Proceedings of ECCV, 2000, pp. 18–32.
[19] Y. Amit, D. Geman, Shape quantization and recognition with randomized trees, Neural Comput. 9 (1997) 1545–1588.
[20] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: CVPR, 2001, pp. 511–518.
[21] M. Elad, Y. Hel-Or, R. Keshet, Pattern detection using a maximal rejection classifier, Pattern Recognit. Lett. 23 (12) (2002) 1459–1471.
[22] S.-K. Pavani, D. Delgado, A.F. Frangi, Haar-like features with optimally weighted rectangles for rapid object detection, Pattern Recognit. 43 (1) (2010) 160–172.
[23] W.-H. Yun, S.Y. Bang, D. Kim, Real-time object recognition using relational dependency based on graphical model, Pattern Recognit. 41 (2) (2008) 742–753.
[24] A. Vedaldi, V. Gulshan, M. Varma, A. Zisserman, Multiple kernels for object detection, in: CVPR, 2009, pp. 606–613.
[25] D. Weiss, B. Sapp, B. Taskar, Sidestepping intractable inference with structured ensemble cascades, in: Proceedings of NIPS, 2010, pp. 2415–2423.
[26] D.K. Prasad, Survey of the problem of object detection in real images, Int. J. Image Process. 6 (6) (2012) 441–466.
[27] P. Dollár, R. Appel, W. Kienzle, Crosstalk cascades for frame-rate pedestrian detection, in: ECCV, 2012, pp. 1–8.
[28] H.O. Song, S. Zickler, T. Althoff, R. Girshick, M. Fritz, C. Geyer, P. Felzenszwalb, T. Darrell, Sparselet models for efficient multiclass object detection, in: ECCV, 2012, pp. 1–8.
[29] C. Dubout, F. Fleuret, Exact acceleration of linear object detectors, in: ECCV, 2012, pp. 301–311.
[30] T. Dean, M.A. Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan, J. Yagnik, Fast, accurate detection of 100,000 object classes on a single machine, in: CVPR, 2013, pp. 1814–1821.
[31] M. A. Sadeghi, D. Forsyth, 30hz object detection with dpm v5, in: ECCV, 2014, pp. 65–79.
[32] J. Yan, Z. Lei, L. Wen, S.Z. Li, The fastest deformable part model for object detection, in: CVPR, 2014, pp. 1–8.
[33] A. Vedaldi, A. Zisserman, Structured output regression for detection with partial occulsion, in: Proceedings of NIPS, 2009, pp. 1–8.
[34] C.-N.J. Yu, T. Joachims, Learning structural SVMs with latent variables, in: Proceedings of ICML, 2009, pp. 1169–1176.
[35] M.B. Blaschko, C.H. Lampert, Learning to localize objects with structured output regression, in: Proceedings of ECCV, 2008, pp. 1–8.
[36] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman, The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results, ⟨http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html⟩, 2009.
[37] P. Dollar, Z. Tu, H. Tao, S. Belongie, Feature mining for image classification, in: CVPR, 2007, pp. 1–8.
[38] C. Wojek, B. Schiele, A performance evaluation of single and multi-feature people detection, in: DAGM, Berlin, Heidelberg, 2008, pp. 82–91.
[39] S. Maji, A. Berg, J. Malik, Classification using intersection kernel support vector machines is efficient, in: CVPR, 2008, pp. 1–8.
[40] P. Dollár, Z. Tu, P. Perona, S. Belongie, Integral channel features, in: BMVC, 2009, pp. 1–11.
[41] P. Dollár, S. Belongie, P. Perona, The fastest pedestrian detector in the west, in: BMVC, 2010, pp. 1–11.
[42] W.R. Schwartz, A. Kembhavi, D. Harwood, L. S. Davis, Human detection using partial least squares analysis, in: CVPR, 2009, pp. 24–31.
[43] S. Walk, N. Majer, K. Schindler, B. Schiele, New features and insights for pedestrian detection, in: CVPR, 2010, pp. 1030–1037.

Marco Pedersoli is a Ph.D. student at the Computer Vision Center of Barcelona, Spain. His research is focused on localization and classification of visual object categories for image understanding.

Andrea Vedaldi is a University Lecturer (Assistant Professor) at the University of Oxford, UK. His research interests include visual detection and recognition and related machine learning methods.

Jordi Gonzàlez is an Associate Professor in Computer Science at the Computer Science Department of the Universitat Autònoma de Barcelona (UAB) and a fellow at the Computer Vision Center. The topic of his research is the cognitive evaluation of human behaviors in image sequences, or video-hermeneutics.

Xavier Roca is an Associate Professor, Director of the Computer Science and Director Department of the UAB, and fellow at the Computer Vision Center. The topic of his research is active vision and tracking.

Please cite this article as: M. Pedersoli, et al., A coarse-to-fine approach for fast deformable object detection, Pattern Recognition (2014), http://dx.doi.org/10.1016/j.patcog.2014.11.006