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

Marco Pedersoli[1] Andrea Vedaldi[2] Jordi Gonzàlez[1]

[1]Centre de Visió per Computador
Autonomous University of Barcelona, Spain
{marcopede,poal}@cvc.uab.es

[2]Department of Engineering Science
University of Oxford, UK
vedaldi@robots.ox.ac.uk

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 to be 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. Therefore accelerating detection requires minimizing the number of part-to-image comparisons. To this end 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 is complementary to the cascade-of-parts approach of [9]. Compared to the latter, our method does not have parameters to be determined empirically, which simplifies its use during the training of the model. Most importantly, the two techniques can be combined to obtain a very significant speedup, of two orders of magnitude 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.

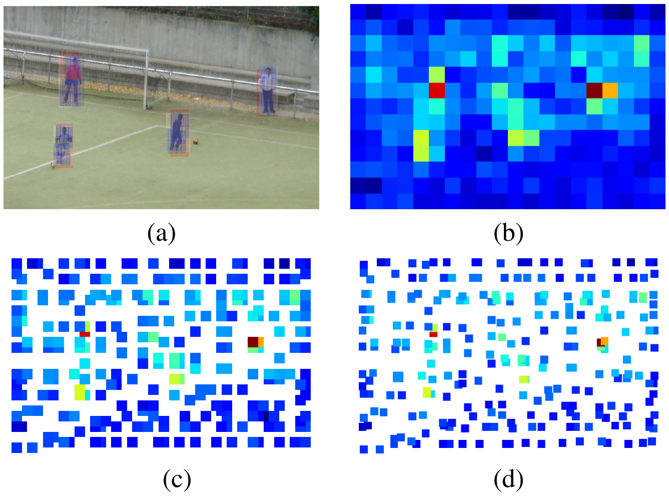


Figure 1. **Coarse-to-fine inference.** We propose a method for the fast inference of multi-resolution part based models. (a) example detections; (b) scores obtained by matching the lowest resolution part (root filter) at all image locations; (c) scores obtained by matching the intermediate resolution 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.*

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 [21, 2, 16, 28] to sophisticated representations that can explicitly account for the location, scale, and spatial configuration of the objects [11, 10]. 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 accuracy as well.

A major obstacle in dealing with geometry is the combinatorial complexity of the inference. For instance, consider the part based models (or pictorial structures) pioneered by Fischler and Elschlager [13]. The time required to estimate such a model from an 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 $O(PL)$ by imposing further restrictions on the model ([11], Sect. 2), but it is still significant due to the large number of 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 deformation bounds and discretization of the part configurations.

In Sect. 2 we reexamine the computational complexity of part based models, and show that the standard analysis does not capture the bottleneck of recent state-of-the-art models such as [3, 10, 29]. We show that, in practice, 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 a different approach to accelerating the inference of part based models that minimizes the number of times parts are matched to the image.

Guided by this observation, we propose a novel multi-resolution part based model and a corresponding coarse-to-fine inference algorithm which is extremely efficient (Fig. 1, Sect. 2,3,4). The method starts by matching the lowest resolution part, selecting for each image neighborhood only its best placement (a form of local non-maximal suppression). These locally optimal placements are then propagated recursively to the parts at higher resolution. In the process, the possible locations of the parts are constrained more and more, leaving only a few part-to-image comparisons to be computed. We show that, overall, this procedure can be ten times faster than the distance transform approach of [11, 10], while still resulting in excellent detection accuracy (Sect. 5).

Related work. Traditionally, object detection has been accelerated by the use of cascades [25, 14, 15, 7, 1, 22, 9]. Recently, for example, cascades have been applied to kernel based methods [23] resulting in models that, while very accurate, are still orders of magnitude slower than the method proposed here.

Our method accelerates part based and deformable models such as [12, 24] by reducing the number of image locations where part filters must be evaluated. The same principle has been used by the cascade of parts [9], which extends [12] directly: parts are tested sequentially and locations are discarded as soon as a partial detection score falls below a certain threshold, determined during a training phase. This avoids testing most of the parts at unpromising image locations, yielding a substantial computational saving.

Compared to the cascade of parts approach, 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. 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 [11] (Sect. 5).

Other relevant works will be cited throughout the paper.

2. Accelerating part based models

A part based model, or pictorial structure as introduced by Fischler and Elschlager [13], 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, to account for all possible translations of a part, L is equal to the number of image pixels. If parts can also scale and rotate, L is further multiplied by the number of discrete scales and rotations, making it very large. Since even for 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 [11] 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 recent part based models such as [9] 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 [3] 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, [12] uses HOG features with a spatial quantization step of $\delta = 8$ pixels, so that there are only L/δ^2 possible placements for a part. Third, in most cases it is sufficient to consider only *small deformations* 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

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

Consider for example a typical pictorial structure of [12]. The part filters are composed of 6×6 HOG cells, so that each part filter has dimension dimension $6\times6\times31 = 1{,}116$ (where 31 is the dimension of a HOG feature for a cell). Typically the elastic connections between parts deform by no more than 6 HOG cells in each direction (which is the size of a part). Thus the number of operations required for inferring the model is

$$P\frac{L}{\delta^2}(1{,}116+36) \qquad (2)$$

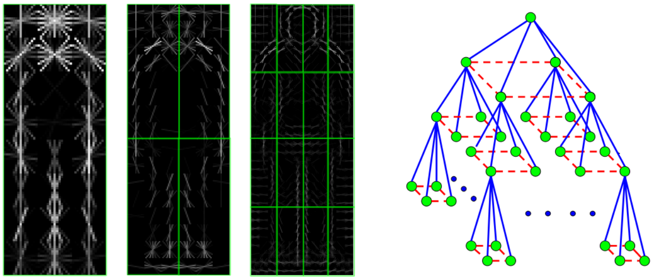

(a) (b)

Figure 2. **Hierarchical part based model of a person.** (a) The model is composed of a collection of HOG filters [3] 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).

where the first term reflects the cost of the filtering, and the second the cost of searching for the best part configuration. Hence the cost of evaluating the part filters is $1{,}116/36 = 31$ times larger than the cost of finding the optimal part configuration.

Fast coarse-to-fine inference. All the best performing part based models incorporate multiple resolutions [18, 29]. Therefore it is natural to ask whether the multi-scale structure can be used not just for better modeling, but also to accelerate inference. This idea was used by [18] for the case of rigid models; here we extend it to the case of deformable parts.

Consider for instance the hierarchical part model of Fig. 2, which is not dissimilar from the one proposed by [29]. The lowest resolution level $r = 0$ corresponds 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 $4^r L/\delta^2$ 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 $(4^r whd) \times (4^r L/\delta^2)$ operations. We propose to avoid most of these computations by guiding the search based on the root filter. Specifically, of all the $4^r L/\delta^2$ 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 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.

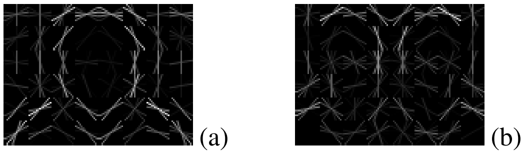


Figure 3. **Effect of lateral connections in learning a model.** (a) Detail of a human model learned with lateral connections active. (b) The same model without lateral connections.

Considering all levels together, the cost of evaluating naively all the part placements for the multi-resolution model is

$$\frac{Lwhd}{\delta^2}\frac{16^R - 1}{15} \tag{3}$$

where R is the number of resolution levels in the model. The coarse-to-fine procedure reduces this cost to

$$\frac{Lwhd}{\delta^2}\frac{4^R - 1}{3}. \tag{4}$$

For instance, if there are $R = 3$ levels the coarse-to-fine procedure is thirteen times faster than the standard Dynamic Programming (DP) approach, at least in term of the effort required to match parts to the image.

Notice that the cost is independent of m, which controls the the size of the neighborhoods where parts are searched. 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.

A more detailed analysis is presented in Sect. 3 and 4.

Lateral connections. The speed-up in our model is due to the fact that the placement of higher resolution parts is guided by the placement of lower resolution ones. This yields high computational savings, but makes inference more sensitive to partial occlusion, blurring, or other sources of noise.

This effect 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. 2 (red dashed edges). This makes the motion of the siblings coherent and improves the robustness of the model. Fig. 3 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. In Sect. 3 it will be shown that the increase in computational complexity due to the lateral connections is negligible.

3. Object model

Our model is a hierarchical variant of [10] (Fig. 2) 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 [9, 3] 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 and our root filter is small to enable relatively large displacements for the higher resolution parts.

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:

$$S(\mathbf{y}; \mathbf{x}, \mathbf{w}) = \sum_{i=1}^{P} S_{H_i}(\mathbf{y}_i; \mathbf{x}, \mathbf{w}) + \sum_{(i,j)\in\mathcal{F}} S_{F_{ij}}(\mathbf{y}_i, \mathbf{y}_j; \mathbf{w}) + \sum_{(i,j)\in\mathcal{P}} S_{P_{ij}}(\mathbf{y}_i, \mathbf{y}_j; \mathbf{w}) \qquad (5)$$

where $\mathcal{F}$ are the parent-child edges (solid blue lines in Fig. 2), $\mathcal{P}$ 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

$$S_{H_i}(\mathbf{y}_i; \mathbf{x}, \mathbf{w}) = H(\mathbf{y}_i; \mathbf{x}) \cdot M_{H_i}(\mathbf{w}) \qquad (6)$$

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

$$S_{F_{ij}}(\mathbf{y}_i, \mathbf{y}_j; \mathbf{w}) = D(2\mathbf{y}_i, \mathbf{y}_j) \cdot M_{F_i}(\mathbf{w}), \qquad (7)$$

where i is the parent of j, $M_{F_i}(\mathbf{w})$ extracts the deformation coefficients from the parameter vector $\mathbf{w}$, and

$$D(2\mathbf{y}_i, \mathbf{y}_j) = \left[(2x_i - x_j)^2, (2y_i - y_j)^2\right] \qquad (8)$$

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 resolution level of the child. Similarly, S_P penalizes sibling-to-sibling deformations and is given by

$$S_{P_{ij}}(\mathbf{y}_i, \mathbf{y}_j; \mathbf{w}) = D(\mathbf{y}_i, \mathbf{y}_j) \cdot M_{P_{ij}}(\mathbf{w}; \mathbf{y}_i). \qquad (9)$$

In this case no additional factors are needed as sibling parts have the same resolution.

In addition to the quadratic deformation costs, the possible configurations are limited by a set of additional constraints, namely parent-child constraints of the form $\mathbf{y}_j \in \mathcal{C}_j + 2\mathbf{y}_i$. In particular, $\mathcal{C}_j + 2\mathbf{y}_j$ is a set of $(2m + 1) \times (2m + 1)$ small displacements around the parent location $2\mathbf{y}_j$ (the parameter m is used again in Sect. 4 in the definition of the accelerated inference procedure, and specified in the experiments in Sect. 5).

As in [10, 24] the model is further extended to multiple aspects in order to deal with large viewpoint variations. Thus 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 (5). Moreover, similarly to [24], 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.

4. DP and coarse-to-fine inference

If the hierarchical model does not have lateral connections (i.e. $\mathcal{P} = \psi$), the structure is a tree and inference can be performed by using the standard DP technique. Namely, if part j is a tree leaf, define $V(\mathbf{y}_j) = S_{H_j}(\mathbf{y}_j)$ (here and in the following equations we drop the dependency on the parameter $\mathbf{w}$ for compactness). For any other part i define recursively

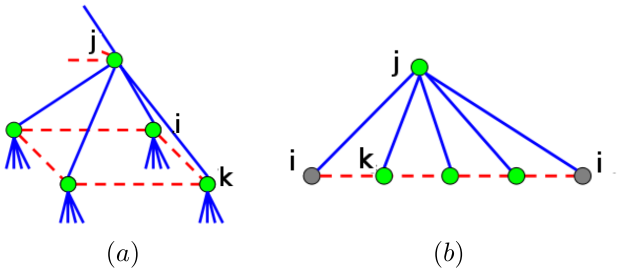
$$V(\mathbf{y}_i) = S_{H_i}(\mathbf{y}_i) + \sum_{j:\pi(j)=i} \max_{\mathbf{y}_j \in \mathcal{C}_j + 2\mathbf{y}_i} \left(S_{F_{ij}}(\mathbf{y}_i, \mathbf{y}_j) + V(\mathbf{y}_j)\right)$$

where $\mathbf{y}_j \in \mathcal{D}_j$ and $i = \pi(j)$ denotes the fact that i is the parent of j. Computing $V(\mathbf{y}_i)$ requires

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

operations, where D is the dimension of a part filter and $\mathcal{C}_j$ the deformation constraints introduced above. The terms $|\mathcal{C}_i|$ can be reduced to one by using the distance transform of [11], but the saving is small since $|\mathcal{C}_i|$ is small to start with.

DP for lateral connections. The lateral connections in Fig. 4 introduce cycles and prevent a direct application of



(a) (b)

Figure 4. **Part-to-part constraints.** The loopy graph generated by the lateral connections is transformed into a chain by clamping the value $\mathbf{y}_i$ and then solved with dynamic programming.

DP. However, these connections form pyramid-like structures (Fig. 4(a)) that can be "opened" by clamping the value of one of the base nodes (Fig. 4(b)). 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

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

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

Coarse-to-fine inference. Despite the increased complexity of the geometry, the cost of inference is still dominated by the cost of applying each part filter to each image location. This cost cannot be reduced by dynamic programming; 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 [9], 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 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 j at the level below are searched at locations $\mathbf{y}_i \in C_i + 2\mathbf{y}_0$, which costs

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

where i is the child clamped, as explained above, 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 [9] the actual cost is a little smaller due to possible interactions between nearby placements of the root part). 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. This should be compared to the $|\mathcal{D}_k|$ comparisons of the DP approach, which grow 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. This idea bears some resemblance to the Cascaded Models proposed in [19], which prune hypothesis based on the combined evidence local to a part and the best global configuration of other parts a certain resolution level, obtained by MAP inference.

Learning. In order to learn the model parameters $\mathbf{w}$ we use the latent structural SVM formulation of [24]. Inference is used during training for two purposes: to estimate the part placements for the ground truth detections (latent variable estimation) and to extract from the negative images hard negative examples [10, 24]. The coarse-to-fine inference procedure can be used to do this because, contrary to the part based cascade of [9], it does not have parameters to be learned. This yields a substantial speedup of training too.

5. Experiments

We evaluated our method on two well known benchmarks: the INRIA pedestrians [3] and the 20 PASCAL VOC 2007 object categories [8]. Performance is measured in term of Average Precision (AP) according to the PASCAL VOC protocol [8].

For the VOC classes we use an object model with two components (aspects), 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. The structural latent SVM performs multiple passes on the training data in order to extract hard negative examples and estimate the pose (part placements) for the positive examples; we limit the latent variable re-estimation passes to 8 and for each we do at most 10 rounds of retraining (selecting hard negatives).

5.1. INRIA pedestrians

Table 1 compares different variants of our coarse-to-fine (CF) detector with the part based cascade of [9] 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 (0.33s vs 0.23s per image). However, the two methods are orthogonal and can

method	det. time (s)	AP (%)
cascade [9]	0.23	85.6
CF	0.25	78.8
CF + siblings	0.33	84.0
CF + sib. + casc.	0.12	83.6

Table 1. **Accuracy and detection speed on the INRIA data.** The table reports the average precision and detection time in seconds for images in the INRIA dataset. *Cascade* denotes the part based cascade of [9]. *CF*, *CF + sibling*, and *CF + sib. + casc.* denote our coarse-to-fine inference scheme, respectively without sibling constraints, with sibling constraints, and combined with the cascade of [9]

.

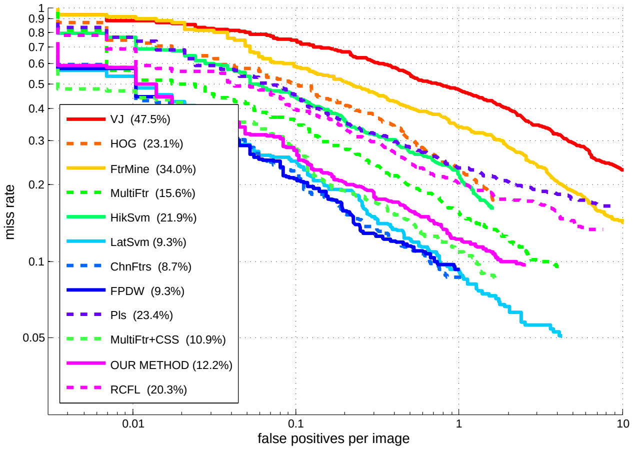


Figure 5. **Comparison to the state-of-the-art on the INRIA dataset.** The miss rate at 1 FPPI is reported in the legend. VJ [25], HOG [3], FtrMine [6], MultiFtr [27], HikSvm [17], LatSvm [10], ChnFtrs [5], FPDW [4], Pls [20], MultiFtr+CSS [26], RCFL [18].

be combined to further reduce the detection time to 0.12s, with just a marginal decrease in the detection accuracy. In fact, for simplicity our cascade implementation only prunes based on a single threshold at the intermediate resolution level; a full implementation is expected to be even faster.

Fig. 5 compares the CF detector with other published methods in term 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. Table 2 evaluates the influence of the neighborhood size m, which controls the amount of deformation that the model allows. Even though humans are in general highly deformable, pedestrians are relatively rigid, so the performance saturates for $m = 1$. Larger values of m do not change substantially the detection performance for this model, but greatly affect the inference time, which increases from 0.33s per image for $m = 1$ to almost 10s for $m = 3$.

m	1	2	3
testing AP (%)	83.5	83.2	83.6
testing time [s]	0.33	2.0	9.3

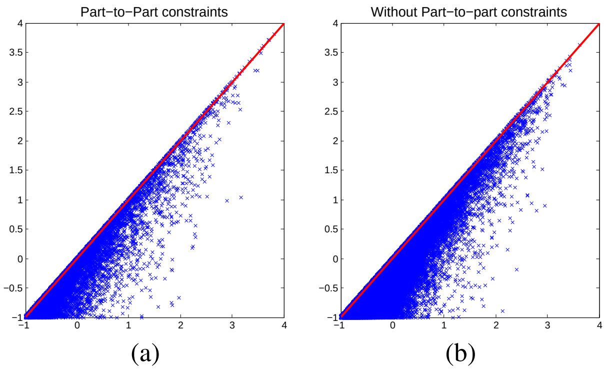
Table 2. **Effect of the neighborhood size** m. On the INRIA Pedestrian dataset setting m to 1 is sufficient to obtain optimal performance. Increasing the value of m does not change substantially the AP, but has a negative impact on speed.

Note that, although a deformation of a HOG cell ($m = 1$) may seem very small, the actual amount of deformation must be measured 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. 6 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 (b) 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 (b), indicating that the lateral connections are in fact helping the CF inference to stay close to the ideal DP case.

Training speed and detection accuracy. Table 3 evaluates the effect of using the CF and exact (DP) inference methods for training and testing the model. Using the CF inference method instead of the exact DP-based inference improves the training speed by an order of magnitude, from 20 hours 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 [9] which requires tuning *after* the model has been learned, our method can be applied *while* the model is learned.

An notable result from Table 3 is the fact that, for each training method (exact DP or CF) and model type (with or without lateral constraints), the accuracy never decreases,



(a)

(b)

Figure 6. **Exact vs coarse-to-fine inference scores.** Scatter polt of the scores obtained by the exact (DP) and approximated (CF) inference algorithms: (a) with lateral constraints in the model, (b) without.

	training		testing AP (%)	
model	method	time	DP	CF
S_F	DP	20h	83.0	84.0
$S_F + S_P$	DP	22h	83.4	84.0
S_F	CF	1.9h	78.0	80.7
$S_F + S_P$	CF	2.2h	83.5	83.5

Table 3. **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.

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% (depending on the training method). Note also that the improvement due to the lateral constraints is larger when training uses the CF inference algorithm.

5.2. PASCAL VOC data

We evaluate our CF model on the 20 classes of the PASCAL VOC 2007 data using the variant with sibling constraints. Table 4 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 base cascade of [9], which is only slightly more accurate (%1 AP better) – however the results reported in [9] are generated from detectors trained on the VOC 2009 data, which contains twice as many training images as found in the VOC 2007 data.

Finally, Fig. 7 evaluates the combination of our CF inference with the part based cascade, by reporting the trade-off of detection speed and accuracy that can be achieved by varying the pruning threshold (as indicated above, we use a simplified version of the cascade with only one threshold). For some classes such as horse, the combinations of the two methods results in a speed-up of almost two orders of magnitude (compared to the exact DP inference) with only a marginal decrease in detection accuracy.

Figure 7. **Combination of the cascade and CF inference.** The figure reports the average precision vs speed-up (over the exact DP inference algorithm) for the CF detector combined with a pruning step analogous to the one used by the part based cascade [9]. As pruning becomes more aggressive, the speed improves at the expense of the detection accuracy.

6. Conclusions

We have presented a method that can substantially speed-up object detectors based on multi-resolution deformable part models. We have shown that, for this type of models, the cost of detection is likely to be dominated by the cost of matching each part to the image, rather than by the cost of finding the optimal configuration of the parts. Based on this observation, we have proposed a new hierarchical model that, combined with a coarse-to-fine inference algorithm, can dramatically speed-up detection by reducing the number of times parts are matched to the image. While the speedup that can be obtained is similar to the one of the part based cascade [9], this method does not require the learning of thresholds or other parameters which simplify its use during the training of the model; moreover, the speed of detection does not depend on the image content. Finally, since our method is orthogonal to the part based cascade, it can be combined with the latter to obtain speedups of up to a factor 100 in some cases. In the future we plan to integrate in the coarse-to-fine architecture even more complex geometric properties of the objects, including rotations and foreshortening.

Acknowledgements. We gratefully acknowledge Josep M. Gonfaus and Andrew Zisserman for their suggestions and comments. This work was initially supported by the EU Project FP6 VIDI-Video IST-045547 and ONR MURI N00014-07-1-0182. Also, the authors 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 and TIN2009-14501-C02-02; and the EU Project FP7 AXES ICT-269980.

References

[1] L. Bourdev and J. Brandt. Robust object detection via soft cascade. In *CVPR*, 2006. 1354

	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean	Time (s)
BOW [23]	37.6	47.8	15.3	15.3	21.9	50.7	50.6	30.0	17.3	33.0	22.5	21.5	51.2	45.5	23.3	12.4	23.9	28.5	45.3	48.5	32.1	≈ 70
PS [10]	29.0	54.6	0.60	13.4	26.2	39.4	46.4	16.1	16.3	16.5	24.5	5.0	43.6	37.8	35.0	8.8	17.3	21.6	34.0	39.0	26.8	≈ 10
Hierarc. [29]	29.4	55.8	9.40	14.3	28.6	44.0	51.3	21.3	20.0	19.3	25.2	12.5	50.4	38.4	36.6	15.1	19.7	25.1	36.8	39.3	29.6	≈ 8
Cascade [9]	22.8	49.4	10.6	12.9	27.1	47.4	50.2	18.8	15.7	23.6	10.3	12.1	36.4	37.1	37.2	13.2	22.6	22.9	34.7	40.0	27.3	< 1
OUR	27.7	54.0	6.6	15.1	14.8	44.2	47.3	14.6	12.5	22.0	24.2	12.0	52.0	42.0	31.2	10.6	22.9	18.8	35.3	31.1	26.9	< 1

Table 4. **Detection AP and speed on the VOC 2007 test data**. Note that *Cascade* is trained using the VOC 2009 data which has more than two times the number of training images of VOC 2007.

[2] G. Csurka, C. R. Dance, L. Dan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Proc. ECCV Workshop on Stat. Learn. in Comp. Vision*, 2004. 1353

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, page 886893, 2005. 1354, 1355, 1356, 1357, 1358

[4] P. Dollár, S. Belongie, and P. Perona. The fastest pedestrian detector in the west. In *BMVC*, 2010. 1358

[5] P. Dollár, Z. Tu, P. Perona, and S. Belongie. Integral channel features. In *BMVC*, 2009. 1358

[6] P. Dollar, Z. Tu, H. Tao, and S. Belongie. Feature mining for image classification. In *CVPR*, June 2007. 1358

[7] M. Elad, Y. Hel-Or, and R. Keshet. Pattern detection using a maximal rejection classifier. *PRL*, 23(12):14591471, 2002. 1354

[8] M. Everingham, A. Zisserman, C. Williams, and L. V. Gool. The PASCAL visual obiect classes challenge 2007 (VOC20067) results. Technical report, Pascal Challenge, 2007. 1357

[9] P. Felzenszwalb, R. Girshick, and D. McAllester. Cascade object detection with deformable part models. In *CVPR*, 2010. 1353, 1354, 1356, 1357, 1358, 1359, 1360

[10] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 32(9), 2010. 1353, 1354, 1356, 1357, 1358, 1360

[11] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1), 2005. 1353, 1354, 1356

[12] P. F. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *Proc. CVPR*, 2008. 1354

[13] M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computer*, 22:67–92, 1973. 1353, 1354

[14] F. Fleuret and D. Geman. Coarse-to-fine face detection. *IJCV*, 41(1):85107, 2001. 1354

[15] S. Gangaputra and D. Geman. A design principle for coarse-to-fine classification. In *CVPR*, 2006. 1354

[16] S. Lazebnik and M. Raginsky. Learning nearest-neighbor quantizers from labeled data by information loss minimization. In *Proc. Conf. on Artificial Intellligence and Statistics*, 2007. 1353

[17] S. Maji, A. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. In *CVPR*, june 2008. 1358

[18] M. Pedersoli, J. Gonzàlez, A. D. Bagdanov, and J. J. Villanueva. Recursive coarse-to-fine localization for fast object detection. In *ECCV*, 2010. 1355, 1358

[19] B. Sapp, A. Toshev, and B. Taskar. Cascaded models for articulated pose estimation. In *ECCV*, 2010. 1357

[20] W. R. Schwartz, A. Kembhavi, D. Harwood, and L. S. Davis. Human detection using partial least squares analysis. In *ICCV*, 2009. 1358

[21] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proc. ICCV*, 2003. 1353

[22] J. Sochman and J. Matas. Waldboost-learning for time constrained sequential detection. In *CVPR*, 2005. 1354

[23] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *ICCV*, 2009. 1354, 1360

[24] A. Vedaldi and A. Zisserman. Structured output regression for detection with partial occulsion. In *Proc. NIPS*, 2009. 1354, 1356, 1357

[25] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, june 2001. 1354, 1358

[26] S. Walk, N. Majer, K. Schindler, and B. Schiele. New features and insights for pedestrian detection. In *CVPR*, 2010. 1358

[27] C. Wojek and B. Schiele. A performance evaluation of single and multi-feature people detection. In *DAGM*, pages 82–91, Berlin, Heidelberg, 2008. 1358

[28] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *IJCV*, 2007. 1353

[29] L. Zhu, Y. Chen, A. Yuille, and W. Freeman. Latent hierarchical structural learning for object detection. In *CVPR*, 2010. 1354, 1355, 1360