Bottom-Up & Top-down Object Detection using Primal Sketch Features and Graphical Models

Iasonas Kokkinos Petros Maragos
School of Electrical and Computer Engineering
National Technical University of Athens
{jkokkin,maragos}@cs.ntua.gr

Alan Yuille
Department of Statistics and Psychology
University of California at Los Angeles
yuille@stat.ucla.edu

Abstract

A combination of techniques that is becoming increasingly popular is the construction of part-based object representations using the outputs of interest-point detectors. Our contributions in this paper are twofold: first, we propose a primal-sketch-based set of image tokens that are used for object representation and detection. Second, top-down information is introduced based on an efficient method for the evaluation of the likelihood of hypothesized part locations. This allows us to use graphical model techniques to complement bottom-up detection, by proposing and finding the parts of the object that were missed by the front-end feature detection stage. Detection results for four object categories validate the merits of this joint top-down and bottom-up approach.

1. Introduction

During the last years part-based models for object detection [4, 30, 1, 17, 10, 16, 11, 9, 6, 2] have become increasingly popular due to their favorable properties like robustness to occlusion, parsimonious object representation and the existence of efficient detection algorithms. An approach that is gaining ground is the combination of part-based representations with the output of interest-point operators which extract salient image points in a scale-invariant manner. An important merit of such approaches is the small amount of points that have to be dealt with; this facilitates the exhaustive search over configurations for both object learning and detection while allowing the introduction of machine learning techniques.

In this work we address two problems incurred by the use of a specific interest point detector: First, the structures to which most interest point detectors [5, 19, 20, 22] respond do not necessarily match the appearance of the parts of all object categories. This is particularly prominent for articulated objects where elongated structures like legs, arms and torsos are either missed altogether or give rise to unnecessarily many interest points.

Second, the use of a threshold for the interest point detectors used at the front-end of the system can result in temperamental behavior, occasionally missing parts of the object. This is commonly treated by considering the part as being occluded, but in most cases the interest operator simply does not surpass a fixed threshold.

Our contributions in this work are twofold: for the first problem we introduce a method for extracting in a scale-invariant manner edge and ridge tokens and for the second we use top-down information to guide the search for the missed features, using an efficient method to estimate the likelihood of pose hypotheses.

In Sec. 2 we briefly discuss prior work and describe the method used for extracting ridge and edge tokens and present the object representation used for bottom-up detection in Sec. 3. After a short description of graphical model based object detection in Sec. 4 we present our top-down completion method; learning tree-structured graphical models is discussed in Sec. 5 and benchmarking results are provided in Sec. 6.

2. Primal Sketch Features

Most scale invariant interest point detectors (see e.g. [19, 5, 20, 22, 25] and references therein) detect image points where a blob/corner/junction strength measure is maximized, offering a set of reliable points for tasks like image registration, mosaicing, etc. After the work of [20, 30, 5] another important application of interest point detectors has emerged, namely their use as image descriptors used for object detection.

Representing the image using information extracted around image points defined structures is efficient and prac-

This work was supported by the Greek research program HRAKLEITOS,which is co-funded by the European Social Fund (75%) and National Resources (25%), the European NoE Muscle and NSF Grant 0413214

tical, but not necessarily optimal. The use of boundary and skeleton information for object modelling and recognition has a long tradition and is well founded [8, 26], while psychophysical evidence suggests that such feature extraction processes take place in our visual system. Two practical problems concerning the incorporation of such features in a detection system are their extraction from gray-level images and deriving a compact description from curved features.

In related work [23] edge maps are represented using point features, which are selected using a scale-invariant criterion; this representation can generate a large amount of keypoints, while it does not explicitly account for the length of the curve, which is a powerful cue for recognition. In the work of [11] edges are extracted using Canny edge detection, edge linking, bitangent point detection, and subsequently projection on a PCA basis. Scale invariance can be introduced as in [19], but this sequence of operations seems fragile, since it is susceptible to noise, (self-) occlusions and object deformations. In [21] affine covariant regions are extracted using a region-based criterion, that does not however lend itself to a transparent generative interpretation as the ridge features used herein.

In our work we build on the feature detection framework of [19, 18] to extract continuous curves corresponding to edge and ridge features in a scale-invariant manner; since we do not make a novel contribution in this direction, we refer to [18] for further details. Our contribution lies in extracting from the continuous curves of the detected features simple tokens defined at single points and thereby introducing them as inputs to an object detection system.

2.1. Edge Features

In [18] the scale-normalized edge strength operator

$$\mathcal{G}_\gamma^t = t^\gamma \left(L_x^2 + L_y^2\right), \quad L = I * G_t, \quad \gamma = 1/2 \qquad (1)$$

was introduced as a means of automatically selecting the scale at which an edge is detected. A smoothed version L of the image I, obtained by convolution with a Gaussian kernel G_t of standard deviation $\sigma = \sqrt{t}$, is used to determine the edge strength, defined in terms of the squared gradient norm. The factor t^γ makes the differential quantity scale-normalized, counteracting the decrease in the response of the differential edge operator caused by Gaussian smoothing of the input image. This renders the responses at different scales commensurate and facilitates the detection of maxima over scale of the normalized edge strength measure. The specific value of γ used is derived from analytical edge models and the constraint that $\mathcal{G}_\gamma$ is maximized at the characteristic scale of the edge. As detailed in [18], the scale selection mechanism automatically accounts for the diffuseness of the edge, choosing the scale perpendicular to the edge orientation that optimally localizes it.

2.2. Ridge Features

Ridge features can complement edge features in the detection of elongated structures, especially at areas with blurred edges but a prominent transition in intensity perpendicular to an elongated structure. Intuitively we can interpret ridge curve detection as estimating a gray-level skeleton of the image [18]; for object detection purposes 'skeletonization' or simply box-fitting is commonly implemented using a cascade of edge detection, edge linking and symmetry axis seeking operations, as e.g. in [14]. Each of these stages introduces errors, due to the complex nature of the operations involved, while requiring scale invariance would further convolve the problem.

Instead of prematurely using an edge detection system, the approach pursued in [18] utilizes the scale-normalized eigenvalues L_{pp}, L_{qq} of the smoothed image Hessian to localize ridge strength maxima simultaneously in space and scale. The quantity $\left(L_{pp}^2 - L_{qq}^2\right)^2$ quantifies the elongation of the local image structure: if the magnitude of one eigenvalue is significantly larger than the second this indicates a ridge-like structure is present at that scale. In $x - y$ coordinates the differential expression writes [18]:

$$\mathcal{N}_\gamma^t = t^{4\gamma} \left(L_{xx} + L_{yy}\right)^2 \left[\left(L_{xx} - L_{yy}\right)^2 + 4L_{xy}^2\right] \qquad (2)$$

where the decrease in derivative magnitude due to Gaussian smoothing is compensated setting $\gamma = 3/4$.

2.3. Straight Line Token Extraction

The feature strength maxima occur along continuous curves by the nature of the criteria being maximized to detect them, thereby providing suitable inputs to point linking techniques. After using the method of [24] for constructing edge and ridge chains, these are broken into straight line segments using an incremental line fitting algorithm [7]; the stopping criterion used is:

$$C = \frac{1}{l}\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(I_x(i) - M_x(i)\right)^2 + \left(I_y(i) - M_y(i)\right)^2}, \quad (3)$$

with N being the number of curve points being attributed to the straight line, and l the line's length. The second expression estimates the standard deviation of the reconstruction error between the straight-line model prediction (M_x, M_y) and the observed feature location (I_x, I_y), quantifying the accuracy of the model. Division by l yields a scale-invariant quantity, since the line's length and the standard deviation estimate scale proportionally to image scale.

Apart from its scale, location and orientation, each line token is associated with an elongation value, labelled ratio for short, which is estimated using the local structure of the image at the estimated feature scale. Specifically, the ratio

Figure 1. Ridge and edge features for the object categories considered. Left column: scale and spatial maxima of edge and ridge strength (green and blue respectively). Mid and right columns: straight ridges and edges extracted using the scale-invariant line fitting criterion. The width of the ridges and the diffuseness of edges is estimated using the local structure of the image and determines the ridge box ratios and the edge terminator lengths.

of the absolute values of the eigenvalues of the Hessian matrix $\frac{|L_{pp}|}{|L_{qq}|}$ is a contrast-invariant quantity and quantifies the elongation of the detected feature; its mean along the curve being fit by the straight line can thus quantify the width of a ridge and the diffuseness of an edge.

Further, the feature strength measures in Eqns. (1) &(2) can be used to quantify the saliency of a line segment; averaging their value along the extracted curve yields a measure that allows to tune the sensitivity of our system's front end.

As shown in Fig. 1, the curves and lines extracted capture to a large extent the dominant structure of the image, that is determined both by the contrast and the scale of the features. Still, various sources of error emerge elsewhere, like low contrast causing low feature saliency and consequently missing features, broken curve chains, or inaccurate tracking of scale and spatial maxima; a detailed treatment of such issues is left for future work.

2.4. Blob Features

Blob and junction detection can be naturally accomplished in the same framework, facilitating the construction of a primal-sketch representation with a unified approach. We use here blob features that are detected at scale and spatial maxima of the scale normalized Laplacian-of-Gaussian operator [19], which is approximately equal to the Difference-of-Gaussians used in [20]. These capture structures like eyes, nostrils, hoofs etc. and are helpful when the object category is not characterized by elongated structures.

3. Bottom-Up Object Detection

In order to initiate the top-down inference process we utilize a simple and efficient bottom-up model indicating image locations where an object could reside. This part's role in the whole system can be interpreted as an attention mechanism, rapidly indicating all potential object locations at the cost of numerous false positives.

3.1. Codebook Representation

As in [1, 17] a codebook representation is used to encode the variation in the object's appearance; in these approaches the codebook entries (CEs) are constructed by clustering the patches extracted around interest point operators based on their intensity. Contrary to these approaches we do not directly use intensity information at all; it is assumed that most of the image-related information is captured in the feature's pose and the type of the feature detected. This results in a small codebook (less than 70 entries), which allows us to learn the relationships among the CEs and exploit the object structure using few training images. Even though arguably rough, such a minimal representation can achieve satisfactory detection results for different object categories.

To learn a codebook representation we use training images where the object appears at a fixed scale and location. For each detected primitive its pose vector $\mathcal{K} = (\mathcal{X}, s, r, \theta)$ is extracted, containing its location $\mathcal{X} = (x, y)$, scale s, elongation factor r and orientation θ. To apply clustering algorithms using Euclidean distances among the pose descriptors these are transformed by taking the logarithm of s, using the embedding $\theta \rightarrow (\cos(2\theta), \sin(2\theta))$ of the orientation in $\Re^2$ and subsequently normalizing the features by their standard deviation.

Ridge, edge and blob features are separately clustered using the k-means algorithm; the number of clusters per feature type is set to a small number – 60. After k-means the EM algorithm is used to refine the estimates of the codebook centers and variances; we constrain the distribution P_k of each cluster to factorize over the individual pose elements: $P_k(\mathcal{K}) = P_{k,\mathcal{X}}(\mathcal{X})P_{k,s}(s)P_{k,r}(r)P_{k,\theta}(\theta)$ and model the individual distributions using Gaussian densities.

3.2. Codebook subset selection

As in [3] a subset of the CEs is chosen based on how well they individually perform on object detection: the detection scenario is simulated where the location of the object is unknown and the truncated pose vector $\mathcal{K}'_i = (s_i, r_i, \theta_i)$ of an extracted feature is used to decide whether it can be attributed to a codebook entry.

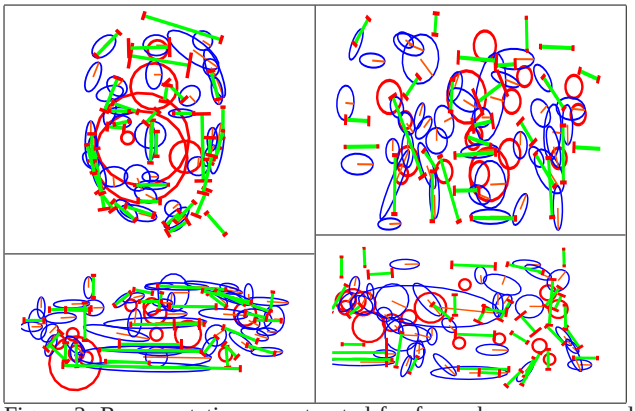

Figure 2. Representations constructed for faces, horses, cars and cows after the codebook subset selection stage; red circles denote blob features.

A background distribution P_B is constructed to account for the statistics of generic images and the behavior of the interest point detector. This is built by aggregating the interest point features extracted from all training images and constructing a distribution $P_B(\mathcal{K}') = P_{B,s}(s)P_{B,r}(r)P_{B,\theta}(\theta)$ using kernel density estimation, with the bandwidth estimated using cross-validation. The i-th extracted feature is then assigned to the k-th CE based on the ratio $L_i = \frac{P_{k,s}(s)P_{k,r}(r)P_{k,\theta}(\theta)}{P_B(\mathcal{K}'_i)}$. The prior probability ratio is incorporated in the threshold used for assignment and is allowed to vary.

For a specific value of the threshold used, k of the features being assigned to the codebook come from the background or are coincidentally somewhere else on the object (e.g. a lip being matched to an eyebrow) and the rest n are actually due to the matched CE. To determine this the full pose vector is introduced and the entries for which $P_{k,\mathcal{X}}(\mathcal{X}) < e^{-2}$ are retained as true hits. One can thus associate to each CE a precision and a recall value $P = \frac{n}{n+k}$ $R = \frac{n}{N}$, with N being the number of objects in the training set. The precision and recall values are combined in the F-measure: $F = \frac{2}{P^{-1}+R^{-1}}$ which is a common index for detector evaluation; varying the threshold the maximal F-measure is estimated and used to indicate the CEs that are better suited for detection, while simultaneously providing an optimal threshold value.

In Fig. 2 the CEs selected using the previous procedure for four different object categories are shown; we see that most salient structures are accounted for in the codebook, while typically several entries correspond to different poses of the same object part, like the front legs of the horse.

3.3. Generation of Candidate Locations

The decisions of the individual detectors can be combined using a probabilistic model that simultaneously accounts for the distributions of all CEs; the simplest possible model is a naive-Bayes model, where no interactions among the CEs are considered and all are directly related to the 'root' node, i.e. the location, O, of the object. For a specific candidate object location (x, y) detection relies on the likelihood ratio $R = \frac{P(I|O_{x,y})}{P(I|B)}$ of the hypotheses that the image I is due to (i) an object O being at location (x, y) and (ii) the background B. Assuming an assignment $\mathcal{H}$ of features to CEs is given and using the factorization of the likelihood term over the individual CEs due to the naive-Bayes model, we can write under the assumptions of [4]:

$$R_{\mathcal{H}} = \prod_{k\in H} \frac{P(f_k^{\mathcal{H}}|h_k, O_{x,y})P(h_k|O_{x,y})}{P(f_k^{\mathcal{H}}|h_k, B)P(h_k|B)} \prod_{k\in M} \frac{1-P(h_k|O_{x,y})}{1-P(h_k|B)}.$$

H is the subset of CEs having found a hit, M the missed ones, $f_k^{\mathcal{H}}$ the pose descriptor of the feature assigned to CE k by $\mathcal{H}$ and h_k is a binary variable indicating CE k has found a hit. The application of this expression is hindered in practice by the need to search over all locations and all possible assignments of features to codebooks to find the locations of high interest. Instead, we invert the top-down flow of dependence, and use the detected keypoints to detect image regions where an object could reside.

For a given image the primal sketch features are estimated and their truncated pose features are matched to the CEs, as described in the previous section. For each CE the threshold value that yields its optimal F-measure is used to make the decision whether to assign a feature to a CE or not. For every possible matching of a feature f to a CE k the full distribution $P_k(f)$ is used to determine the image locations (x, y) for which the ratio $\frac{P(f_k|h_k,O_{x,y})P(h_k|O_{x,y})}{P(f_k|h_k,B)P(h_k|B)}$ is larger than one. The evidence in favor of an object at those points is multiplied by this factor and the locally maximal responses obtained by accumulating contributions from all possible matchings propose potential object locations. This detection method is similar the codebook-voting approach of [17]; however, the Gaussian densities used do not necessitate the Mean-Shift clustering algorithm used therein, while using a generative model for the features allows the principled construction of likelihood expressions.

Specifically, since the constraint that each feature is matched to a single CE is not necessarily satisfied, overcounting of evidence may occur, resulting in sub-optimal decisions. This problem is traded off by the speed with which this stage is accomplished, but can be easily eliminated once the top-down model is initiated at the proposed location. Assigning the features to the single CE for which the maximal increase in the log-ratio is achieved directly yields an improvement in performance, as shown in the experimental results section.

4. Incorporation of Top-Down information

The generated object hypotheses typically rely on a small set of matched CEs with the rest being considered

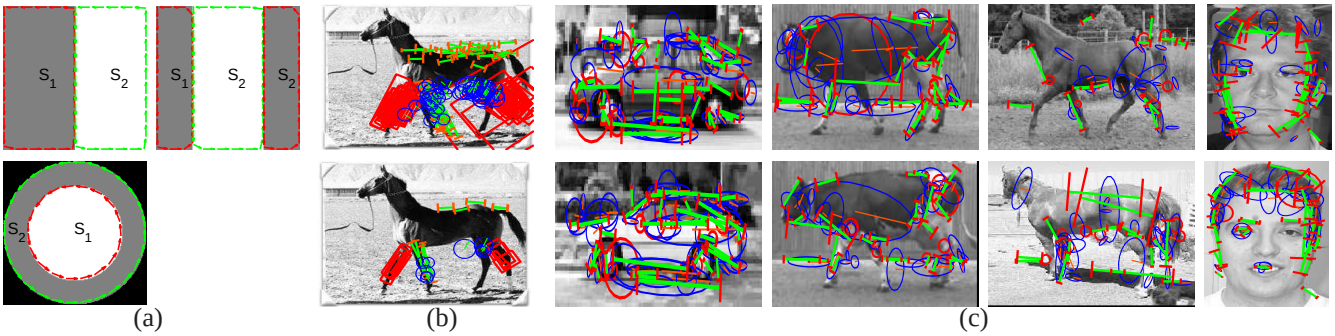

Figure 3. Incorporation of top-down information using efficient particle likelihood estimation: (a) Templates for edge, ridge and blob features and tangent vectors to template subpart borders. Area integrals over the subpart supports, S_1, S_2 can be replaced with curvilinear ones, drastically reducing computation time. The density of the tangent vectors is proportional to the integration step and controls the tradeoff between speed and accuracy. (b) Top: samples from the messages sent from the object-center node to part nodes. Bottom: the 5 highest ranking samples for each node. (c) Top-down part matches obtained by thresholding the probability of observing an object part.

as missed. This is primarily due to i) the inherently over-complete representation provided by a codebook ii) the extracted features not being assigned to the CE during the matching stage, due to the log-ratio being below the CE's threshold iii) inefficiencies of the feature extraction stage on its own. The last two problems can be alleviated using top-down information to recover the missing features: the matched CEs can propose potential locations for the missed ones, which can be evaluated using the available image information.

This approach has been pursued initially in [3, 17], using exemplars instead of a generative model; formulating the problem in the framework of generative models reduces the computational and representational complexity of the top-down stage, since no object-specific fragment/boundary dictionaries are needed and makes a link with the general probabilistic approach to vision. In [13] a syntactical approach is used in a similar setting, but a static set of primal-sketch tokens is used [12], while the focus is on parsing generic structures rather than detecting objects.

In a graphical model formulation, using the observation potential expression $\Phi_i(\mathcal{P}_i) = P(I|\mathcal{P}_i)$ to express the likelihood of the image observations I conditioned on the pose of the $\mathcal{P}_i$-th node, the optimal configuration $\mathcal{P} = (\mathcal{P}_1, \ldots, \mathcal{P}_N)$ can be found as the maximizer of:

$$P(I|\mathcal{P}) = \prod_i \Phi_i(\mathcal{P}_i) \prod_{(i,j)\in\mathcal{C}} \Psi_{i,j}(\mathcal{P}_i, \mathcal{P}_j). \quad (4)$$

The product in the second term is over the cliques of the tree distribution, and for the naive-Bayes model we are currently employing relates the object's location with all the CEs, i.e. $\Psi_{i,O}(\mathcal{P}_i, \mathcal{P}_O) = P(\mathcal{P}_i|\mathcal{P}_O)$ with $\mathcal{P}_O$ being the pose of the object center node. In the following section we shall explore the potential of using distributions defined over subsets of CEs alone, ignoring the location of the object.

When as in [9, 6, 11] the whole image is used for detection instead of a small set of points, $\Phi_i(\mathcal{P}_i)$ is evaluated at all discretized values of the pose vector $\mathcal{P}_i$, of part i. Belief propagation-type algorithms [9] can then be used to estimate the marginals $P(\mathcal{P}_i|I)$. In brief, each node i sends a message $m_{i,j}$ to its neighbors $\mathcal{N}(i)$, using the messages it receives from neighboring nodes:

$$m_{i,j}(\mathcal{P}_j) = \sum_{\mathcal{P}_i} \Phi_i(\mathcal{P}_i)\Psi_{i,j}(\mathcal{P}_i, \mathcal{P}_j) \prod_{k\in\{\mathcal{N}(i)\setminus j\}} m_{k,i}(\mathcal{P}_k, \mathcal{P}_i). \quad (5)$$

The belief for each potential location of node i is estimated as the product of the incoming messages with the potential function Φ_i, $B_i(\mathcal{P}_i) = \Phi_i(\mathcal{P}_i) \prod_{j\in\mathcal{N}(i)} m_{i,j}(\mathcal{P}_i)$ and for a tree-structured graph $B_i(\mathcal{P}_i)$ will equal $P(\mathcal{P}_i|I)$ upon convergence of the message passing operation. The major bottleneck is the evaluation of the summation in Eq. (5): when orientation, scale, and location are used the pose space becomes huge, and even though efficient algorithms can be used to speed it up [9], the algorithm is still slow for practical applications.

In [4, 30, 10] where only interest points are used it is not possible to recover from misses of the interest operator, while the alternative of reducing the threshold results in a combinatorial explosion. An approach lying somewhere in between is that of [16], where a set of features is extracted initially and then instead of exhaustive search, loopy belief propagation is used; this results in computational savings, but it is still not possible to recover missed features.

In our case, on the one hand we know in advance the features that gave rise to the hypothesized object location, which leads to closed form expressions for the messages sent to their neighboring nodes, since Gaussian pairwise functions are used; on the other, the summation over all the possible poses of the missed features is computationally demanding. This operation can be speeded up since the summand in Eq. (5) is above zero only where the product of incoming messages is above zero. This is a case that is well suited to the application of particle filtering methods, avoiding the evaluation of the likelihood function over the whole pose space. Instead of a full-blown particle filtering

method [27, 28], we approximate products of the form $\sum_{\mathcal{P}_i} \Phi_i(\mathcal{P}_i) \prod_j m_{i,j}(\mathcal{P}_i)$ as follows:

(i) analytically estimate $P_{in}(\mathcal{P}_i) = \prod_j m_{i,j}(\mathcal{P}_i)$
(ii) draw N samples S_n from $P_{in}(\mathcal{P}_i)$
(iii) form the Monte Carlo approximation:

$$\sum_{\mathcal{P}_i} \Phi_i(\mathcal{P}_i) \prod_{j \in \mathcal{N}(i)} m_{i,j}(\mathcal{P}_i) \simeq \sum_{i=1}^{N} l_n d(P - S_n) \qquad (6)$$

using Dirac functions centered at the samples and the likelihoods $l_n = \Phi(S_n) = P(I|S_n)$ as weights.
(iv) Approximate the distribution in terms of a Gaussian distribution minimizing the Kullback Leibler divergence to the Monte Carlo approximation, $-\sum_n l_n \log \frac{P(S_n)}{l_n}$.
(vi) Analytically estimate the messages sent to neighbors.

Apart from step iii) all other steps are straightforward and can be efficiently implemented; in what follows we present a method to efficiently estimate the likelihood terms, which facilitates the practical application of this approach.

4.1. Efficient Likelihood estimation

For the estimation of $\Phi_i(S_n) = P(I|S_n)$ one can either use a static set of bottom-up features F as in [27] favoring some parts of the pose space over others and associate $P(I|S_n) \simeq P(F|S_n)$ or one using a generative model, reconstruct 'on demand' the image based on the pose hypothesis S_n, estimate the reconstruction error and derive a corresponding likelihood expression. Pursuing the latter approach, we derive our likelihood expressions by building a simple template for each type of features as shown in Fig. 3, modelling the image around the feature patch in terms of two (or generally K) constant intensity values.

The generative model used assumes that within the support T_{S_n} of the template patch corresponding to particle S_n the image is generated by corrupting the template with white Gaussian noise. Given the generative model parameters, F, namely the constant intensity values within the template subparts and the standard deviation σ of the noise process, the likelihood of the observations writes:

$$P(I|S_n) = \prod_{i \in T_{S_n}} P(I_i|F), \quad P(I_i|F) = N(I_i - c_i, \sigma) \quad (7)$$

where c_i is the template model's intensity prediction at point i. Taking the logarithm of the above expression and setting its derivative with respect to the model parameters to zero, their maximum likelihood estimates are:

$$\begin{aligned} c_k &= \sum_{i \in S_k} I_i/|S_k|, \quad k = 1 \dots K \\ \sum_k |S_k|\sigma^2 &= \sum_k \sum_{i \in S_k} (I_i - c_k)^2 = \sum_i I_i^2 - \sum_k c_k^2 |S_k| \end{aligned}$$

where K is the number of constant sub-models used, S_k is the set of pixels belonging to the k-th template subpart and $|\cdot|$ stands for set cardinality.

All the summations in the above expressions can be seen as discretizations of area integrals of $I(x,y)$,$I^2(x,y)$,$1(x,y)$ over the template subpart domains; our method circumvents area integrals using curvilinear ones, based on Stoke's theorem:

$$\iint_S f(x,y)dxdy = \int_{\partial S} Pdx + Qdy = \int_0^l (P,Q) \cdot \mathcal{T} ds.$$

$\mathcal{T}$ is the unit norm tangent vector to the curve, l is the curve's length, and Q and P is a pair of functions such that $\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} = f(x,y)$. An obvious pair is the set of 'integral images' [29] $Q(x,y) = \frac{1}{2}\int_0^x f(x,y)dx$, $\quad P(x,y) = -\frac{1}{2}\int_0^y f(x,y)dy$. Since all the quantities involved in the likelihoods can be replaced by the corresponding curvilinear expressions, this allows us to rapidly estimate the likelihood of each hypothesized pose using summations over the template subpart borders instead of summing over its whole support. Further, the spacing of the points on which the curvilinear integral is estimated can be adjusted at will, allowing to control the tradeoff between speed and accuracy in the estimation of the quantities involved.

In practice featureless regions result in low reconstruction errors and large weights in Eq. (6); instead of using an ad-hoc prior on the model parameters to penalize smooth regions, the discriminative quantity $p_i = \frac{P(I|S_i)}{P(I|S_i)+P(I|C)}$ is used as weight instead. This combines the likelihood of the observations given the particle's pose $P(I|S_i)$ with a complementary, constant intensity model $P(I|C)$, thereby focusing on strong features.

As shown in Fig. 5, using top-down information we can recover the parts of the object that have been missed during the initial detection stage; the probability of observing a part is estimated using the particle S_i for which the posterior p_i is maximized. During detection improved results are obtained by treating this quantity as a feature, and modelling its activity on positive and negative training images using nonparametric distributions.

5. Graphical Model Construction

Using a naive-Bayes graphical model it is assumed that the pose of each CE depends only on the pose of the object; this can be an impediment to accurate model construction, especially for articulated objects [14, 27, 9], where the tree-like structure of the dependencies cannot be exploited. When constructing a codebook-based representation from the bottom-up the exploration of such dependencies is infeasible, unless large amounts of data are available. Most of the codebook pairs, triples etc. will not be simultaneously active on images, thereby rendering the construction of joint probability distributions problematic. However, using the top-down filling-in method we can recover most of the parts that have been missed and enrich the observation

set. Further, the smooth probabilities of having observed a part can be incorporated in the estimation of pairwise potentials and the selection of cliques for structure learning.

Efficient algorithms for inference with graphical models [9, 6] are feasible for distributions of the form:

$$P(\mathcal{P}) = \frac{\prod_{C_i \in \mathcal{C}} P_C(\mathcal{P}_C)}{\prod_{S_i \in \mathcal{S}} P_S(\mathcal{P}_S)} \quad (8)$$

where $\mathcal{C}$ is the set of maximal cliques of the graphical model and $\mathcal{S}$ the set of separators, namely variables shared by the cliques. For tree-structured distributions where only binary relations are encoded in the maximal cliques, the Minimum Spanning Tree algorithm can be used to recover the optimal graph connecting all nodes as in [9].

Specifically, using Gaussian functions for the individual and joint distributions, the gain in log-likelihood achieved by considering the joint distribution of nodes i and j is:

$$C_{i,j} = \sum_{(\mathcal{P}_i, \mathcal{P}_j) \in \mathcal{P}_{ij}} \log \frac{P(\mathcal{P}_i, \mathcal{P}_j)}{P(\mathcal{P}_i) P(\mathcal{P}_j)} = cN \log \frac{|\Sigma_i||\Sigma_j|}{|\Sigma_{i,j}|} \quad (9)$$

with $\mathcal{P}_{ij}$ the set of observed (i, j) node pairs, N its cardinality, $\Sigma_{i,j}$ and Σ_i, Σ_j the maximum likelihood estimates of the joint and individual covariance matrixes respectively and c a common proportionality factor.

Since we are not using manually annotated data, two issues that emerge are that i) the covariance matrixes are estimated using observations of varying quality due to the potentially inaccurate part matches and ii) we have a soft measure of the number of observed pairs, since the top-down stage provides a probability measure. We have observed that compared to thresholding these probabilities, better results are obtained by incorporating them in the parameter and pair count estimates. Specifically, the parameters of the distributions are estimated by maximizing the weighted log-likelihood of the observed pose vectors, $\sum_i p_i \log P(\mathcal{P}_i)$ which yields:

$$\mu_i = \frac{\sum_i l_i \mathcal{P}_i}{\sum_i l_i}, \Sigma_i = \frac{\sum_i l_i (\mathcal{P}_i - \mu_i)}{\sum_i l_i} \quad (10)$$

while the quantity N in Eq. (9) is replaced by $\sum_i l_i$. For pairwise distributions $\mathcal{P}_i$ amounts to the feature vector formed by joining the pose vectors of both codebook entries and l_i is the product of the individual probabilities of observing each part of the pair. As shown in Fig. 4, the learnt graphical model structure is intuitively appealing, since most edges are between neighboring parts, correctly capturing the dependency of their pose vectors.

After recovering the graph structure two set of clique parameters are estimated, one using foreground (as before) and the other background images. During detection, the log likelihood ratio of the recovered parts under the two hypotheses is used to discriminate between valid (foreground) and coincidental (background) configurations. Using the graphical models of this section instead of the naive-Bayes ones used earlier for the top-down task is straightforward, but we have not compared their performance yet.

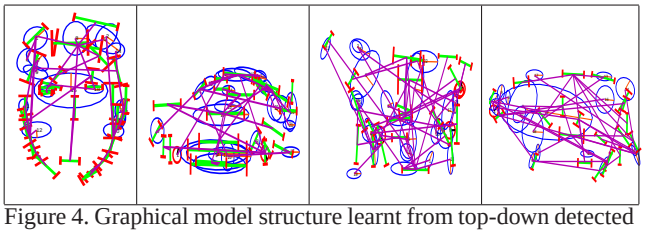

Figure 4. Graphical model structure learnt from top-down detected features for 40 CEs; background CEs are omitted for simplicity.

6. Experimental Results

To validate the merit of each proposed step we focus on the detection of cars from [1]; in related work, e.g. [1, 17, 11, 2], precision-recall (P-R) curves are provided for this dataset, allowing a direct comparison. For the other categories similar conclusions are drawn from the P-R results, but they cannot be directly compared to other work.

For cars and faces we use the datasets and validation methods of [1, 10, 11] and 100 training images, while for cows and horses we use the datasets of [3, 17] and 50 images for training and 50 for testing. Car images are rescaled by a factor of 2.5, and flipped to have the same direction during training, while faces are normalized so that the distance between the eyes is 50 pixels and a 50×30 box is used to label a detected face a true hit.

For the results shown in Fig. 5 (a) we disentangle nonmaximum suppression [1] from the performance of the detector using a 5×5 window, which explains the apparently low precision values. Each of the proposed steps introduces an improvement in performance, with the final fused top-down and bottom-up results being considerably better than the purely bottom-up; as in [15] we combine two classifiers by adding their values, after normalizing them to be commensurate. The improvement in performance gained by introducing more particles and decreasing the integration step is demonstrated in Fig. 5(b); for the final results we use the highest accuracy method which takes 40 sec per hypothesis on a 1.4 Ghz. PC while the other two cases need 7 and 15 sec. In Fig. 5(c) where we compare our results after nonmaximum suppression to those of other authors, our method is found to rank among the best and being systematically inferior only to the current state-of-the-art [17], where intensity information, a larger codebook and a fixed scale interest operator are used. Flipping the images during training may result in some small improvement in performance, but this should be negligible given that the cars in [1] and the learned models are almost symmetrical. Still, we consider it is more important that excellent detection results are obtained by combining bottom-up with top-down

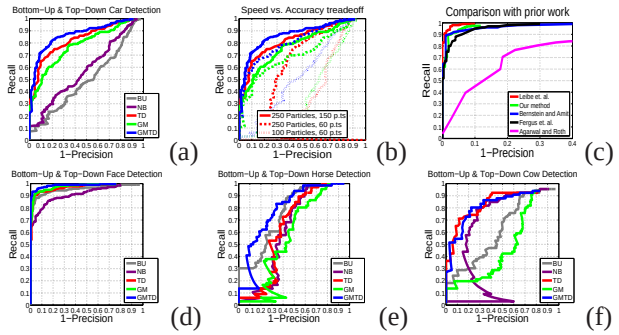

Figure 5. Top-row: (a) Improvements in performance introduced at different processing stages: BU: bottom-up, NB: naive Bayes, TD: Top-down filled-in features, GM: Graphical model, GMTD: combination of GM and TD. (b) Accuracy vs. performance tradeoff; colors encode the detection method used. (c) Comparison with other methods, after nonmaximum suppression. Bottom-row: Comparative results for the other categories.

information *without* the complex appearance descriptors of [20, 11], using a small codebook representation and efficient algorithms. In the bottom row P-R curves are provided for the rest categories, where for faces excellent localization is achieved. For horses and cows the results are not as good but this can be attributed to the combination of a small training set and a more complex object structure.

7. Conclusions

In this work primitive geometrical structures, like straight edges/ridges and blobs have been used for object representation; this has enabled the construction of simple templates for the related structures and the efficient estimation of the likelihood of arbitrary feature poses. The incorporation of top-down information is thus enabled, resulting in systematically improved detection results.

We are currently pursuing methods to introduce more complex appearance information than constant profiles, while keeping the computations simple and efficient. 2-D geometric patterns like corners, T-junctions or more general primal sketch primitives [12] can also be incorporated in the same framework, allowing for their efficient exploitation in top-down processes, while improving the performance of both detection streams. At the higher level, we are interested in learning more compact structure models for complex objects. The efficient and robust extraction of simple image primitives could facilitate the use of syntactical approaches [13], while the top-down filling-in method can be used to iteratively refine the learned graphical model.

References

[1] S. Agarwal and D. Roth. Learning a Sparse Representation for Object Detection. In *ECCV*, 2002.

[2] E. J. Bernstein and Y. Amit. Part-Based Statistical Models for Object Classification and Detection. In *CVPR*, 2005.

[3] E. Borenstein and S. Ullman. Class-Specific, Top-Down Segmentation. In *ECCV*, 2002.

[4] M. Burl, M. Weber, and P. Perona. A Probabilistic Approach to Object Recognition using Local Photometry and Global Geometry. In *ECCV*, 1998.

[5] C. Schmid and R. Mohr. Local grayvalue invariants for object retrieval. *IEEE Trans. PAMI*, 19(5), 1997.

[6] D. Crandall, P. Felzenszwalb, and D. Huttenlocher. Spatial Priors for Part-Based Recognition using Statistical Models. In *CVPR*, 2005.

[7] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2003.

[8] D. Hoffman and W. Richards. Parts of Recognition. *Cognition*, 18:65–96, 1985.

[9] P. Felzenszwalb and D. Huttenlocher. Pictorial Structures for Object Recognition. *IJCV*, 2005.

[10] R. Fergus, P. Perona, and A. Zisserman. Object Class Recognition by Unsupervised Scale-Invariant Learning. In *CVPR*, 2003.

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

[12] C.-E. Guo, S.-C. Zhu, and Y. N. Wu. A Mathematical Theory of Primal Sketch and Sketchability. In *ICCV*, 2003.

[13] F. Han and S. C. Zhu. Bottom-Up/Top-Down Image Parsing by Attribute Graph Grammar. In *ICCV*, 2005.

[14] S. Ioffe and D. A. Forsyth. Probabilistic Methods for Finding People. *IJCV*, 43(1):45–68, 2001.

[15] I. Kokkinos and P. Maragos. An Expectation Maximization Approach to the Synergy between Object Categorization and Image Segmentation. In *ICCV*, 2005.

[16] M. P. Kumar, P. H. S. Torr, and A. Zisserman. Extending Pictorial Structures for Object Recognition. In *BMVC*, 2004.

[17] B. Leibe, A. Leonardis, and B. Schiele. Combined Object Categorization and Segmentation with an Implicit Shape Model. In *ECCV*, 2004. SLCV workshop.

[18] T. Lindeberg. Edge Detection and Ridge Detection with Automatic Scale Selection. *IJCV*, 30(2), 1998.

[19] T. Lindeberg. Feature Detection with Automatic Scale Selection. *IJCV*, 30(2), 1998.

[20] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 60(2), 2004.

[21] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust Wide Baseline Stereo from Maximally Stable Extremal Regions. In *BMVC*, 2002.

[22] K. Mikolajczyk and C. Schmid. Scale and Affine Invariant Interest Point Detectors. *IJCV*, 60(1), 2004.

[23] K. Mikolajczyk, A. Zisserman, and C. Schmid. Shape recognition with edge-based features. In *BMVC*, 2003.

[24] R. Nevatia and K. Babu. Linear feature extraction and description. *CGIP*, 13(3):257–269, July 1980.

[25] F. Schaffalitzky and A. Zisserman. Multi-View Matching for Unordered Image Sets. In *ECCV*, 2002.

[26] K. Siddiqi and B. Kimia. Parts of Visual Form: Computational Aspects. *IEEE Trans. PAMI*, 17:239–251, Mar. 1995.

[27] L. Sigal, S. Bhatia, S. Roth, M. J. Black, and M. Isard. Tracking loose-limbed people. In *CVPR*, 2004.

[28] E. Sudderth, A. Ihler, W. Freeman, and A. Willsky. Nonparametric belief propagation. In *CVPR*, 2003.

[29] P. Viola and M. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. In *CVPR*, 2001.

[30] M. Welling, M. Weber, and P. Perona. Unsupervised Learning of Models for Recognition. In *ECCV*, 2000.