Excerpt chapter from Synthesis lecture draft:
Visual Recognition
Kristen Grauman and Bastian Leibe.

Chapter 7

Part-Based Category Models

The previous chapters introduced object categorization approaches that were based on unordered sets of features (as in the case of *bag-of-visual-words* methods) or that incorporate only weak spatial constraints (as *e.g.* in the case of the *pyramid match kernel*). In this chapter, we now want to examine the question how to incorporate more detailed spatial relations into the recognition procedure and how the resulting object representation can be efficiently learned from training data.

For this, we draw parallels to the specific object recognition techniques presented in Chapter 4. Back then, we were concerned with establishing exact correspondences between the test image and the model view in order to verify if the matched features occurred in a consistent geometric configuration. As the exact appearance of the model object was known, the extracted features could be very specific, and accurate transformation models could be estimated.

When moving from specific object recognition to object categorization, however, the task becomes more difficult. Not only may the object appearance change due to intra-category variability, but the spatial layout of category objects may also undergo a certain variation. Thus, we can no longer assume the existence of exact correspondences. As shown in Figure 7.1, we can however still often find local object fragments or parts with similar appearances that occur in a similar spatial configuration. The basic idea pursued in this chapter is therefore to learn object models based on such parts and their spatial relations.

Figure 7.1: While the global object appearance may undergo significant variation inside a category, the appearance and spatial relationship of local parts can often still give important cues. This provides a strong motivation for using part-based models *(BL: Figure from Rob Fergus)*.

7.1 Object Categorization with Part-Based Models

Many part-based models have been proposed in the literature. The idea to represent objects as an assembly of parts and flexible spatial relations reaches back to Fischler & Elschlager's work in 1973 [FE73]. While this early work started from a set of hand-defined part templates, most recent approaches try to also learn the part appearance from training data. This implies that the learning algorithm itself should be able to *select* which local object regions to represent and it should be able to *group* similar local appearances into a common part representation. An optimal solution to the selection problem would imply a search over a huge search space. The development of local invariant features however provides an efficient alternative which has proven to work well in practice. Consequently, all part-based models discussed in the following are based on local features.

Once the parts have been defined, the next question is how to represent their spatial relationship. This choice reflects the mutual independence assumptions we want to make about relative part locations, and it directly affects the number of parameters needed to fully specify the resulting model, as well as the complexity of performing inference using this model.

Various spatial models have been proposed over the years. Figure 7.2 gives

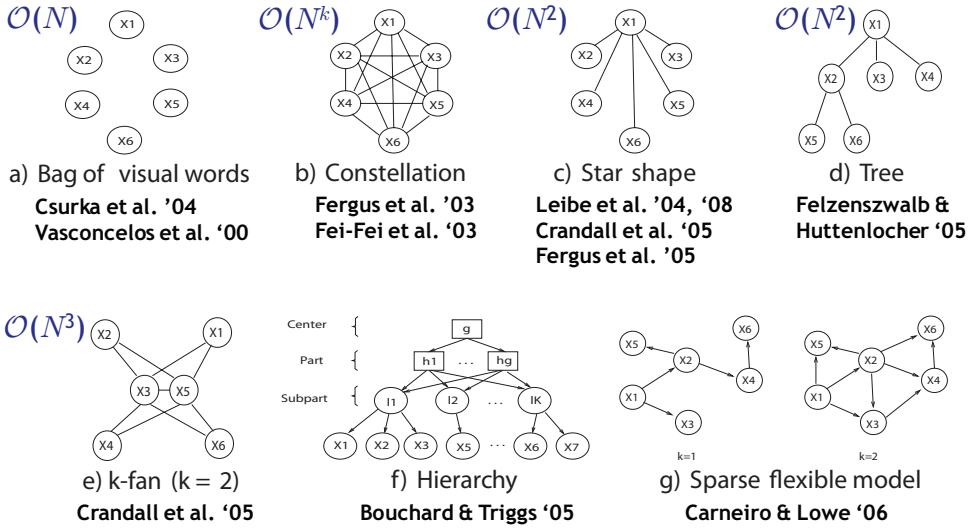

Figure 7.2: Overview over different part-based models investigated in the literature. In this chapter, we focus on two models from this list: the Constellation Model and the Star Model *(BL: Figure adapted from Carneiro* et al. *[CL06]).*

an overview over the most popular designs. The simplest model is a *Bag of Visual Words*, as described in Chapter 5 and shown in Fig. 7.2(a). This model does not encode any geometric relations and is listed just for completeness. At the other extreme is a fully connected model, which expresses pairwise relations between any pair of parts. This type of model has become known as a *Constellation Model* and has been used in [FZP03, FFFP03]. A downside of the full connectivity is that such a model requires an exponentially growing number of parameters as the number of parts increases, which severely restricts its applicability for complex visual categories.

A compromise is to combine the parts in a *Star Model* (Fig. 7.2(c)), where each part is only connected to a central reference part and is independent of all other part locations given an estimate for this reference part. Such a representation has been used in the *Implicit Shape Model* [LLS04, LLS08], as well as in several other approaches [CFH05, FPZ05, OPZ06a]. The advantage of this model is its computational efficiency: exact inference can be performed in $\mathcal{O}(N^2)$ (compared to $\mathcal{O}(N^k)$ for a k-part Constellation model), and more efficient approximations can be devised based on the ideas of the *Generalized Hough Transform* [LS03b, LLS04, LLS08] or the *Generalized Distance Transform* [FH05b]. The idea of the Star Model can be readily generalized to a *Tree Model* (Fig. 7.2(d)), where each part's location is only dependent on the location of its parent. This type of model is used in the *Pictorial Structures* framework by Felzenszwalb & Huttenlocher [FH05b] and has led to efficient algorithms for human pose estimation.

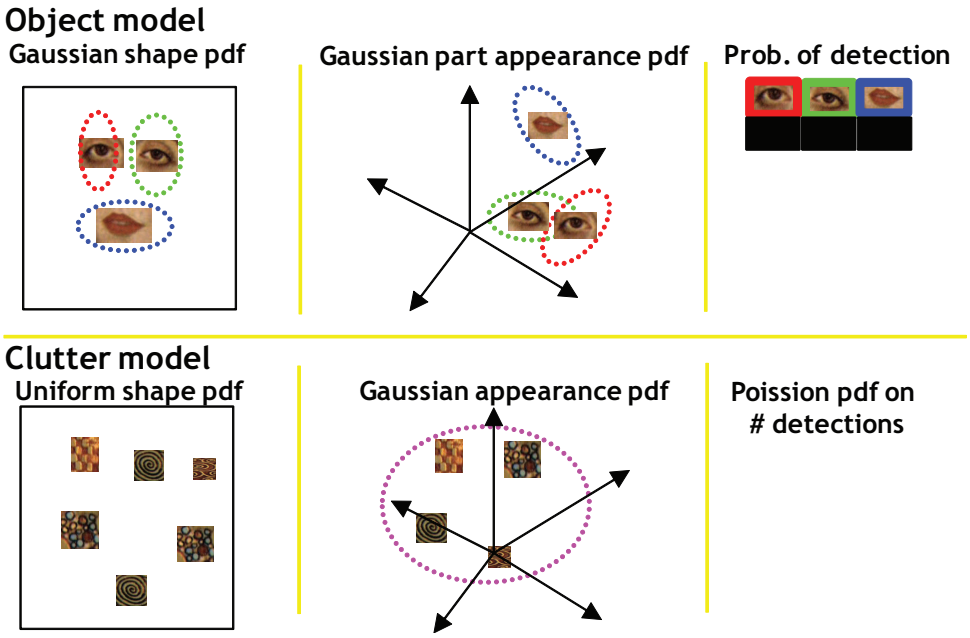

Figure 7.3: Visualization of the different components of the Constellation Model (see text for details) *(BL: Figure from Fergus* et al. *[FZP03]).*

Finally, the above ideas can be generalized in various other directions. The *k-fan Model* [CFH05] (Fig. 7.2(e)) spans a continuum between the fully-connected Constellation Model and the singly-connected Star Model. It consists of a fully-connected set of k reference parts and a larger set of secondary parts that are only connected to the reference parts. Consequently, its computational complexity is in $\mathcal{O}(N^{k+1})$. A similar idea is employed in the *Hierarchical Model* (Fig. 7.2(f)) by Bouchard & Triggs [BT05], which contains a (star-shaped) layer of object parts, each of which is densely connected to a set of bottom-level local feature classes. Finally, we want to mention the *Sparse Flexible Model* (Fig. 7.2(g)) proposed by Carneiro & Lowe [CL06], where the geometry of each local part depends on the geometry of its k nearest neighbors, allowing for flexible configurations and deformable objects.

In the following, we will focus on two models from this list which have been widely used in the literature: the Constellation Model and the star-shaped Implicit Shape Model. We will introduce the basic algorithms behind those approaches and discuss their relative strengths and weaknesses.

7.2 The Constellation Model

The *Constellation Model* by [WWP00c, WWP00a] and [FZP03] was introduced for unsupervised learning of object categories. It represents objects by estimating a joint appearance and shape distribution of their parts. Thus, object parts can be characterized either by a distinct appearance or by a distinct location on the object. As a result, the model is very flexible and can even be applied to objects that are only characterized by their texture.

The Constellation model can best be introduced by first considering the recognition task. Given a learned object class model with P parts and parameters θ, the task is to decide whether a new test image contains an instance of the learned object class or not. For this, N local features are extracted with locations $\mathbf{X}$, scales $\mathbf{S}$, and appearances $\mathbf{A}$. The Constellation model now searches for an assignment $\mathbf{h}$ of features to parts in order to make a Bayesian decision R [FZP03]:

$$R = \frac{p(\text{Object}|\mathbf{X},\mathbf{S},\mathbf{A})}{p(\text{No object}|\mathbf{X},\mathbf{S},\mathbf{A})} \approx \frac{p(\mathbf{X},\mathbf{S},\mathbf{A}|\theta)p(\text{Object})}{p(\mathbf{X},\mathbf{S},\mathbf{A}|\theta_{bg})p(\text{No object})}\ , \tag{7.1}$$

where the likelihood factorizes as follows

$$\begin{aligned} p(\mathbf{X},\mathbf{S},\mathbf{A}|\theta) &= \sum_{h\in\mathcal{H}} p(\mathbf{X},\mathbf{S},\mathbf{A},\mathbf{h}|\theta) \\ &= \sum_{h\in\mathcal{H}} \underbrace{p(\mathbf{A}|\mathbf{X},\mathbf{S},\mathbf{h},\theta)}_{Appearance}\underbrace{p(\mathbf{X}|\mathbf{S},\mathbf{h},\theta)}_{Shape}\underbrace{p(\mathbf{S}|\mathbf{h},\theta)}_{Rel.\ Scale}\underbrace{p(\mathbf{h}|\theta)}_{Other}\ . \end{aligned} \tag{7.2}$$

That is, we represent the likelihood as a product of separate terms for appearance, shape, relative scale, and other remaining influences. Figure 7.3 shows a visualization of those different components. In each case, a separate model is learned for the object class and for the background.

Briefly summarized, the first term represents each part's appearance independently by a Gaussian density in a 15-dimensional appearance space obtained by PCA dimensionality reduction from 11×11 image patches. This is compared against a single Gaussian density representing the background appearance distribution. The shape term models the joint Gaussian density of the part locations within a hypothesis in a scale-invariant space. The corresponding background clutter model assumes features to be spread uniformly over the image. The scale model is again given by a Gaussian density for each part relative to a common reference frame, also compared against a uniform background distribution. Finally, the last term takes into account both the number of features detected in the image (modeled using a Poisson distribution) and a probability table for all possible occlusion patterns if only a subset of the object parts could be observed.

The classification score is computed by marginalizing over all $|\mathcal{H}| \subseteq \mathcal{O}(N^P)$ possible assignments of features to parts. This marginalization makes it possible to represent an entire category by a relatively small number of parts. It effectively removes the need to make hard assignments at an early stage – if two features provide an equally good support for a certain part, both will contribute substantially to the total classification result. At the same time, the exponential complexity of the marginalization constitutes a major restriction, since it limits the approach to a relatively small number of parts.

7.2.1 Learning Procedure

The Constellation Model has been designed with the goal of learning with weak supervision. That is, neither the part assignments, nor even object bounding boxes are assumed to be known – only the image labels (target category or background) are provided. Given such a training dataset, the goal of the learning procedure is to find the maximum likelihood estimate for the model parameters $\hat{\theta}_{ML}$, *i.e.* the parameter setting that maximizes the likelihood for the observed data $\mathbf{X}, \mathbf{S}, \mathbf{A}$ from all training images.

This is achieved using the *expectation maximization* (EM) algorithm. Starting from a random initialization, this algorithm converges to a (locally optimal) solution by alternating between two steps. In the E-step, it computes an expectation for the part assignments given the current value of θ. The M-step then updates θ in order to maximize the likelihood of the current assignment. Since the E-step involves evaluating the likelihood for each of the N^P possible feature-part assignments, efficient search methods are needed to keep the approach computationally feasible. Still, the authors report training times of 24-36 hours for a single-category model trained on 400 class images in their original paper [FZP03]. This is partially also due to the large number of parameters required to specify the fully-connected model (according to [FZP03], a 5-part model needs 243 parameters and a 6-part model already requires 329 parameters), which in turn impose a requirement on the minimum training set size. Those constraints together restrict the original approach to a small set of only 5-6 parts.

7.2.2 Example Results

Figure 7.4 shows the learned representations and recognition results on two different object categories. The first category, motorbikes, has a clearly defined structure. Consequently, the learned model contains well-defined appearances and compact spatial locations for all object parts (as visible from the small covariance ellipses

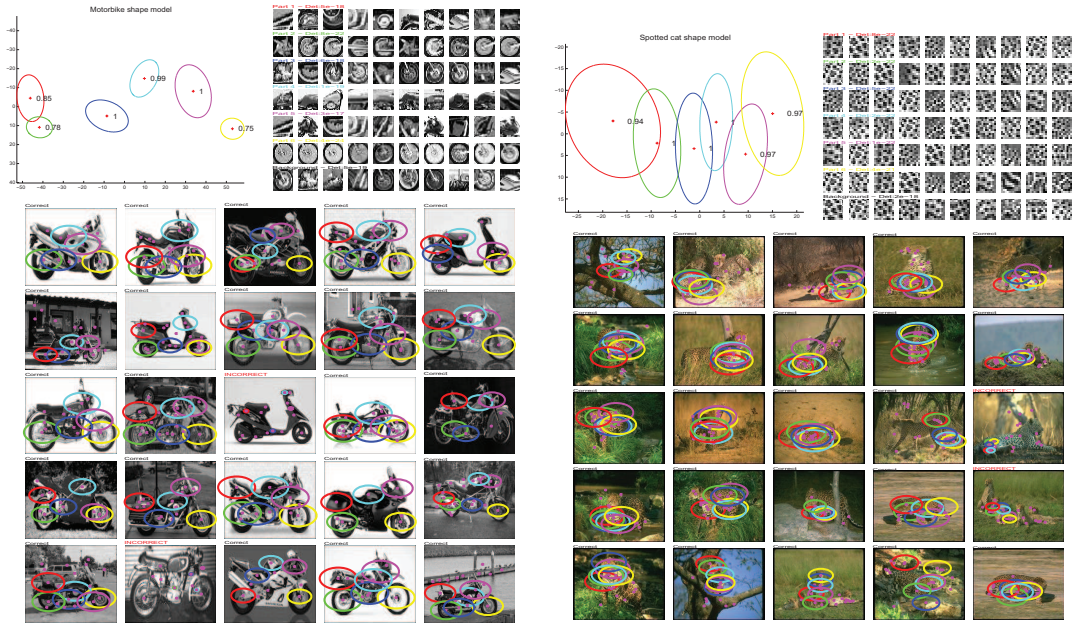

Figure 7.4: Results of the Constellation Model on two object categories: motorbikes and spotted cats. The top row shows the learned representations for spatial relations and appearance; the bottom row contains recognition results on images from the test set, visualizing the best-scoring part assignment *(BL: Figure from Fergus* et al. *[FZP03]).*

in the upper left plot of the figure). It can also be seen that the parts are consistently found in corresponding locations on the test images, showing that the learned representation really makes sense. In contrast, the second category, "spotted cats" contains significant variability from different body poses and viewing angles. As a result, the Constellation Model focuses on the repeatable texture as most distinctive feature and keeps only very loose spatial relations. This ability to adapt to the requirements of different categories, automatically weighting the contribution of appearance versus spatial features as needed for the task at hand, is an important property of the Constellation Model.

7.2.3 Discussion and Extensions

The Constellation model was historically one of the first successful part-based models for object categorization. It therefore had a big impact and helped shape the field for the next years. In addition, it initiated a research competition for the best spatial representation and introduced one of the first realistic benchmark datasets for this task. Many of the above-mentioned restrictions were addressed in follow-up work,

e.g. in the later papers by [FFFP03, FPZ05]. As research progressed, it became clear that the full connectivity offered by the original Constellation Model was both not required and could not be taken advantage of given the usual training set sizes that were investigated. Instead, star-shaped and tree-shaped spatial models were deemed more promising, as they require a far smaller number of parameters and are more efficient to evaluate. Consequently, Fergus *et al.* themselves proposed an updated version of their model incorporating such a star topology [FPZ05].

7.3 The Implicit Shape Model (ISM)

The fully-connected shape model described in the previous section is a very powerful representation, but suffers from a high computational complexity. in this section, we now examine a recognition approach that builds upon a much simpler spatial representation, namely a Star Model in which each part's location only depends on a central reference part. Given this reference position, each part is treated independently of the others. Thus, the object shape is only defined implicitly by the information which parts agree on the same reference point. This motivates the name of the approach: *Implicit Shape Model* (ISM) [LS03b, LLS04, LLS08].

Together with the change in the spatial model, the ISM approach also takes on a different philosophical interpretation of what object properties are represented. The Constellation model aims to represent a relatively small number of (less than 10) *semantically meaningful parts*, with the tacit assumption that each object of the target category should contain those parts. The parts may undergo appearance variations and may occur in varying spatial configurations, but a majority of them should always be present and if any part cannot be found, it should be explicitly flagged as "occluded". In contrast, the ISM does not try to model semantically meaningful parts, but instead represents objects as a collection of a large number (potentially 1000s) of *prototypical features* that should ideally provide a dense cover of the object area. Each such prototypical feature has a clearly defined, compact appearance and a spatial probability distribution for the locations in which it can occur relative to the object center. In each test image, only a small fraction of the learned features will typically occur—*e.g.*, different features will be activated for a dark and a brightly colored car—but their consistent configuration can still provide strong evidence for an object's presence.

7.3.1 Learning Procedure

In contrast to the Constellation Model, the ISM requires labeled training examples. In the least case, the labels should include a bounding box for each training object,

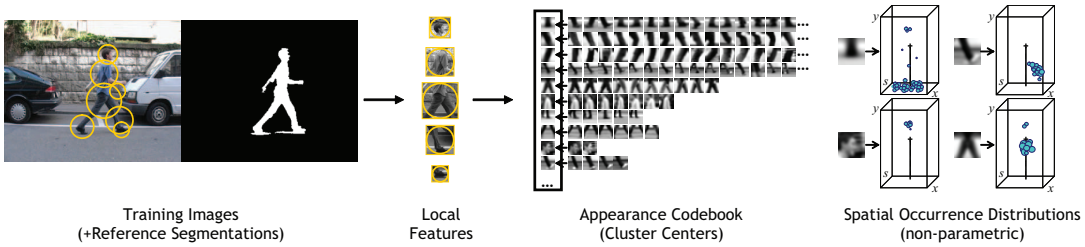

Figure 7.5: Visualization of the ISM training procedure (Figure from [LLS08]).

so that the training algorithm knows the object location and scale. However, in order to take full advantage of the ISM's capabilities, the training examples should also include a reference segmentation, *e.g.* given by a polygonal object boundary as available in the LabelMe database [RTMF08] or by a pixel-wise figure-ground map. If such a training segmentation is available, the ISM can then infer a top-down segmentation for each test image as a result of the recognition procedure. This requirement of a training segmentation may sound like a big restriction. However, it only has to be provided for relatively small training sets (50-150 examples per visual category are usually sufficient), and the recognition results are significantly improved as a consequence.

The full ISM training procedure is visualized in Figure 7.5. The first step is to build up a visual vocabulary (the *appearance codebook*) from scale-invariant local features that overlap with the training objects, using any of the methods presented in Chapter 5, Section 5.2.1. Next, the ISM learns a *spatial occurrence distribution* for each visual word. For this, we perform a second pass over all training images and match the extracted features to the stored vocabulary using a soft-matching scheme (*i.e.*, activating all visual words within a certain distance threshold). For each visual word, the ISM stores a list of all positions and scales at which this feature could be matched, relative to the object center. This results in a non-parametric probability density representation for the feature position given the object center. The key idea behind the following recognition procedure is then that this distribution can be inverted, providing a probability distribution for the object center location given an observation of the corresponding feature.

In addition to the non-parametric distribution, the ISM also stores a reference figure-ground mask for each occurrence entry, as extracted from the segmented training image at the feature position. This mask is then later on used for inferring a top-down segmentation.

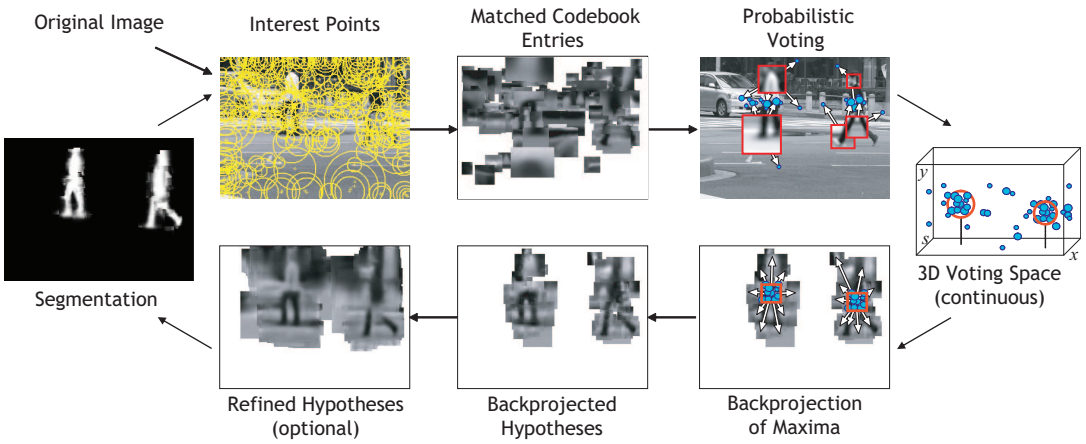

Figure 7.6: Visualization of the ISM recognition procedure (Figure from [LLS08]).

7.3.2 Recognition Procedure

The ISM recognition procedure follows the idea of the Generalized Hough Transform (*c.f.* Section 4.3.2), but extends this technique to model the uncertainty inherent in recognizing an object category [LLS08]. An overview is shown in Figure 7.6. Given a new test image, the ISM again extracts local features and matches them to the visual vocabulary using soft-matching. Each activated visual word then casts votes for possible positions of the object center according to its learned spatial distribution, whereupon consistent hypotheses are searched as local maxima in the voting space.

In order to model the uncertainty of the object category, the Hough voting step is formulated in a probabilistic manner. The contribution of a feature f observed at location ℓ to the object category o_n at position $\mathbf{x}$ is expressed by a marginalization over all matching visual words $\mathcal{C}_i$:

$$p(o_n, \mathbf{x}|f, \ell) = \sum_i \underbrace{p(o_n, \mathbf{x}|\mathcal{C}_i, \ell)}_{Hough\ vote} \underbrace{p(\mathcal{C}_i|f)}_{Matching\ prob.} \quad . \tag{7.3}$$

The first term corresponds to the stored occurrence distribution for visual word $\mathcal{C}_i$, which is weighted by the second term, the probability that feature f indeed corresponds to this visual word. In practice, this second term is usually set to $\frac{1}{|\mathcal{C}^\star|}$, where $|\mathcal{C}^\star|$ corresponds to the number of matching visual words. Thus, each image feature casts an entire distribution of weighted votes.

As another difference to the standard GHT, the votes are stored in a continuous 3D voting space for the object position $\mathbf{x} = (x, y, s)$. Maxima in this space are efficiently found using Mean-Shift Mode Estimation [CM02] with a scale-adaptive

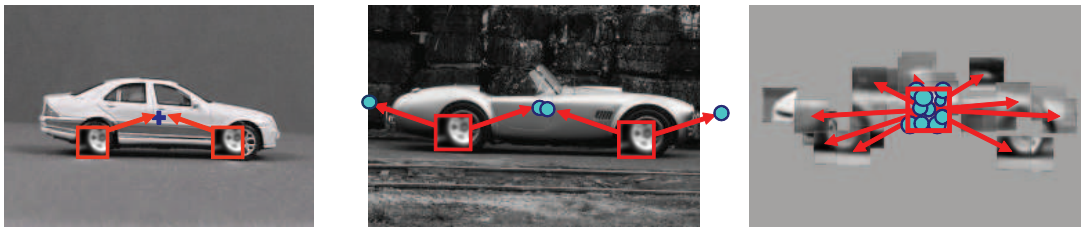

Figure 7.7: Visualization of the basic idea employed in the ISM. (left) During training, we learn the spatial occurrence distribution of each visual word relative to the object center. (middle) For recognition, we use those learned occurrence distributions in order to cast probabilistic votes for the object center in an extension of the Generalized Hough Transform. (right) Once a maximum in the voting space has been found, we can backproject the contributing votes in order to get the hypothesis's support in the image.

kernel K:

$$\hat{p}(o_n, \mathbf{x}) = \frac{1}{V_b(\mathbf{x}_s)} \sum_k \sum_j p(o_n, \mathbf{x}_j | f_k, \ell_k) K\left(\frac{\mathbf{x} - \mathbf{x}_j}{b(\mathbf{x}_s)}\right) , \tag{7.4}$$

where b is the kernel bandwidth and V_b its volume. Both are adapted according to the scale coordinate $\mathbf{x}_s$, such that the kernel radius always corresponds to a fixed fraction of the hypothesized object size. This way, the recognition procedure is kept scale invariant [LS04, LLS08].

The search procedure can be interpreted as kernel density estimation for the position of the object center. It should be noted, though, that the ISM voting procedure does not conform to a strict probabilistic model, since the vote accumulation implies a summation of probabilities instead of a product combination, as would be required. This issue is more closely examined in the recent work by Lehmann *et al.* [LLV09], where a solution is proposed motivated by a duality to sliding-window detection approaches.

Once a hypothesis has been selected, all features that contributed to it are backprojected to the image, thereby visualizing the hypothesis's support. This backprojected information is later on used to infer a top-down segmentation. The main ideas behind the ISM recognition procedure are again summarized in Figure 7.7.

7.3.3 Top-Down Segmentation

The backprojected support already provides a rough indication where the object is in the test image. As the sampled features still contain background structure, this is

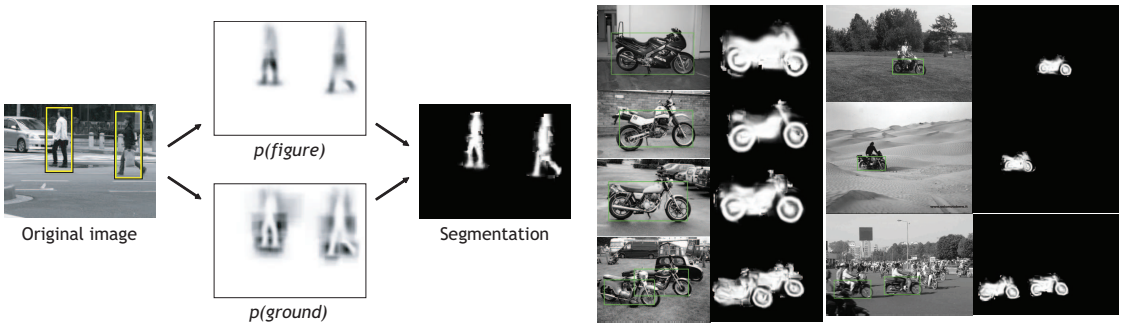

Figure 7.8: (left) Visualization of the ISM top-down segmentation procedure. (right) Example detection and segmentation results on motorbikes (Figures from [LLS08]).

however not a precise segmentation yet. The ISM therefore performs an additional step in order to infer a pixel-wise figure-ground segmentation from the recognition result. This step uses the stored local figure-ground masks that were obtained when learning from segmented training images.

In detail, we are interested in the probability that a pixel $\mathbf{p}$ is *figure* or *ground* given the object hypothesis, *i.e.* $p(\mathbf{p}=\mathit{figure}|o_n,\mathbf{x})$. This probability can be obtained by marginalizing over all features containing this pixel and then again marginalizing over all vote contributions from those features to the selected object hypothesis [LS03b, LLS08]:

$$p(\mathbf{p}=\mathit{figure}|o_n,\mathbf{x}) = \tag{7.5}$$
$$\underbrace{\sum_{(f_k,\ell_k)\ni\mathbf{p}}}_{\substack{\text{all contributing}\\ \text{features containing}\\ \text{pixel } \mathbf{p}}} \quad \underbrace{\sum_{i}}_{\substack{\text{all visual}\\ \text{words matched}\\ \text{to feature } f_k}} \quad \underbrace{p(\mathbf{p}=\mathit{fig.}|o_n,\mathbf{x},\mathcal{C}_i,\ell_k)}_{\substack{\text{Stored f/g mask}\\ \text{for each vote}}} \underbrace{p(o_n,\mathbf{x}|f_k,\ell_k)}_{\mathit{Vote\ weight}} \frac{p(f_k,\ell_k)}{p(o_n,\mathbf{x})} \ .$$

In this formulation, $p(\mathbf{p}=\mathit{fig.}|o_n,\mathbf{x},\mathcal{C}_i,\ell_k)$ denotes the stored figure-ground masks for the votes contributed by feature (f_k,ℓ_k). The priors $p(f_k,\ell_k)$ and $p(o_n,\mathbf{x})$ are assumed to be uniform. This means that for every pixel, we effectively build a weighted average over all local segmentation masks stemming from features containing that pixel, where the weights correspond to the features' contribution to the object hypothesis. The final object segmentation is then obtained by computing the likelihood ratio of the *figure* and *ground* probability maps for every pixel, as shown in Figure 7.8.

As shown by Thomas *et al.* [TFL+07, TFL+09], this top-down segmentation procedure can be further generalized to also infer other kinds of meta-data annotations. This includes discrete properties such as part labels, continuous values such

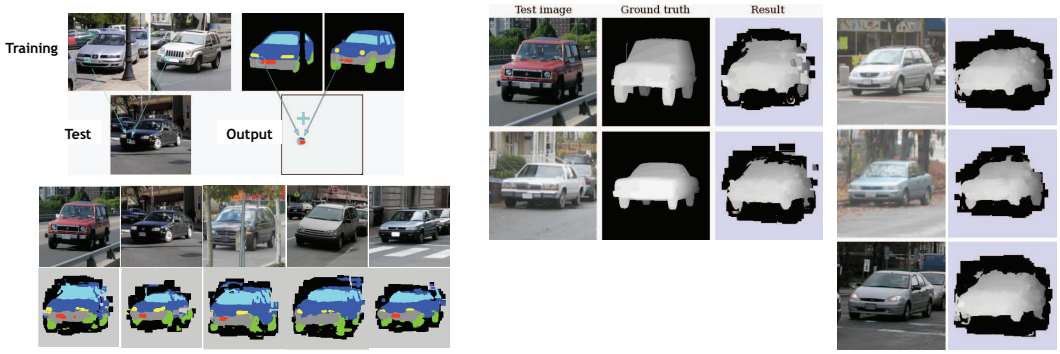

Figure 7.9: As shown by Thomas *et al.* [TFL+07, TFL+09], the ISM top-down segmentation procedure can be generalized to also infer other kinds of meta-data, such as part labels, depth maps, or surface orientations.

as depth maps, as well as vector-valued data such as surface orientations. Some example results are shown in Figure 7.9.

7.3.4 Hypothesis Verification

Finally, the extracted object hypotheses are verified in a model selection procedure, which selects the set of hypotheses that together best explain the image content. Briefly stated, this procedure expresses the score of a hypothesis as the sum over its per-pixel $p(\textit{figure})$ probability map. If two hypotheses overlap, then they compete for pixels, as each pixel can only be explained by a single hypothesis. Thus, each pair of hypotheses incurs an interaction cost that is subtracted from their combined scores. A new hypothesis is therefore only selected if it can draw sufficient support from an otherwise as-yet-unexplained image region. This step is important to obtain robust detection performance and significantly improves the recognition results. For details, we refer to [LLS08].

Some example detection results with this verification procedure are shown in Figure 7.10. The ISM and its extensions were also successfully applied to a variety of other object categories, in particular for detecting pedestrians [LSS05] and cars [LLS08, LCCV07]. The recognition performance however hinges on the availability of a sufficient number of input features to cover the target objects. For this reason, later experiments were often based on a combination [LMS06] of several different interest region detectors [LCCV07]. For example, the chair detection results shown in Figure 7.10(right) were obtained through a combination of *Harris-Laplace*, *Hessian-Laplace*, and *DoG* interest regions.

Figure 7.10: Example recognition and segmentation results of the ISM.

7.3.5 Discussion and Extensions

The ISM provides a successful example for a part-based recognition approach based on a Star Model. The success of this simple representation may first seem surprising, since it imposes no further constraints on relative part locations other than that they should be consistent with a common object center. Clearly, this is quite a weak constraint, but its good performance in practice can be explained by the large number of local features that contribute to an object hypothesis. If those features overlap, they are no longer truly independent, and consistent responses are enforced this way. This property is also used by the ISM top-down segmentation stage, which further reinforces consistency between overlapping local segmentations. Still, it may happen that additional, spurious object parts are associated to a hypothesis simply because they are also consistent with the same object center. This may particularly become a problem for articulated objects, as shown in Figure 7.11. Experience however shows that such effects can usually be removed by a further hypothesis verification stage enforcing more global constraints (as done *e.g.* in [LSS05]).

Since its inception, a number of extensions have been proposed for the basic ISM algorithm. Those include adaptations for rotation-invariant voting [MLS06], multi-cue combination [LMS06, SLMS05], multi-category recognition [MLS06, LCCV07], multi-viewpoint detection [SLMS06, TFL$^+$06], discriminative verification [FLCS05, GL09, MM09], and articulated pose estimation [ARS08]. We refer to the extensive literature for details.

7.4 Concluding Remarks

In this chapter, we have discussed two popular part-based models for object categorization. Several other models should also be mentioned, but could not be expanded

Figure 7.11: An important restriction of the star topology model used in the ISM is that no higher-level spatial relations between features are encoded. Each local feature that is consistent with the same object center may contribute to a hypothesis. For articulated objects, this may lead to additional body parts being associated to an object hypothesis (Figure from [LSS05]).

upon due to space constraints. Of particular interest here is the *Pictorial Structures* approach by Felzenszwalb & Huttenlocher [FH05b], which implements a Tree Model. This model has become popular for articulated body pose analysis (*e.g.* in [RFZ07, FMZ08, ARS08]).

As mentioned before, part-based models have the advantage that they can deal with object shapes that are not well-represented by a bounding box with fixed aspect ratio. They have therefore often be applied for recognizing deformable object categories. In contrast, it has been observed that, given sufficient training data, the discriminative sliding-window approaches from Chapter 2 (*e.g.* [DT05b, FMR08]) often perform better for recognizing mostly rectangular object categories such as faces or front/back views of pedestrians. The best choice of representation therefore again depends on the application.