Multi-Scale Object Detection by Clustering Lines *

Björn Ommer Jitendra Malik
Computer Science Division, EECS
University of California at Berkeley
{ommer, malik}@eecs.berkeley.edu

Abstract

Object detection in cluttered, natural scenes has a high complexity since many local observations compete for object hypotheses. Voting methods provide an efficient solution to this problem. When Hough voting is extended to location and *scale, votes naturally become lines through scale space due to the local scale-location-ambiguity. In contrast to this, current voting methods stick to the location-only setting and cast point votes, which require local estimates of scale. Rather than searching for object hypotheses in the Hough accumulator, we propose a weighted, pairwise clustering of voting lines to obtain globally consistent hypotheses directly. In essence, we propose a hierarchical approach that is based on a sparse representation of object boundary shape. Clustering of voting lines (CVL) condenses the information from these edge points in few, globally consistent candidate hypotheses. A final verification stage concludes by refining the candidates. Experiments on the ETHZ shape dataset show that clustering voting lines significantly improves state-of-the-art Hough voting techniques.*

1. Introduction

Category-level object detection in cluttered natural scenes requires matching object models to the observations in the scene. The two leading approaches to this problem are sliding windows, *e.g*. [34, 9], and voting methods, which are based on the Hough transform [19]. Sliding windows scan over possible locations and scales, evaluate a binary classifier, and use post-processing such as non-max suppression to detect objects. The computational burden of this procedure is daunting although various techniques have been proposed to deal with the complexity issue, *e.g*. cascaded evaluation [34], interest point filtering, or branch-and-bound [21]. In contrast to this, Hough voting [19] parametrizes the object hypothesis (*e.g*. the location of the object center) and lets each local part vote for a point in hypothesis space. Since it was first invented [19], the Hough transform has been generalized to arbitrary shapes [1], used for instance registration [25], and employed for category recognition [23, 15, 30, 31].

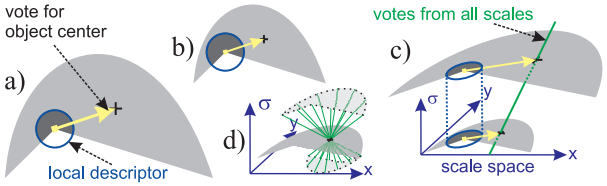

Figure 1. Voting in scale space with scale-location-ambiguity. The circle indicates the spatial support of a local feature. Based on the descriptor, the difference of object scale between a) and b) is not detectable. Thus, a local feature casts votes for the object center on all scales, c). These votes lie on a line through scale space, since the position of the center relative to a feature varies with object scale as expressed in Eq. 7. Without noise, all voting lines from an object intersect in a single point in scale space, d). For all other scales this point is blurred as indicated by the dotted outline.

Although various techniques for local estimation of scale have been proposed in earlier years (*e.g*. [25, 20]), the latest high performance recognition systems do not rely much on local scale estimates and rather sample features densely on a large range of scales [4, 10]. It is interesting that so many approaches use SIFT features computed at multiple scales, when the SI in SIFT stands for scale invariance!

We believe that inherently object scale is a global property, which makes local scale estimates unreliable and, thus, leads to a ***scale-location-ambiguity*** illustrated in Fig. 1. When the Hough transform is extended to provide hypotheses for location *and* scale, each local feature casts votes that form lines through scale space rather than just a single point as in current voting methods [24, 30, 14, 15], see Fig. 1. Since all points on a voting line are statistically dependent, they should agree on a single object hypothesis rather than being treated as independent votes. In this setting, finding consistent object hypotheses naturally leads to a formulation as a pairwise clustering of voting lines. Clustering avoids a local search through hypothesis space [23] and the pairwise setting circumvents having to specify the

* This work was supported by ONR MURI N00014-06-1-0734

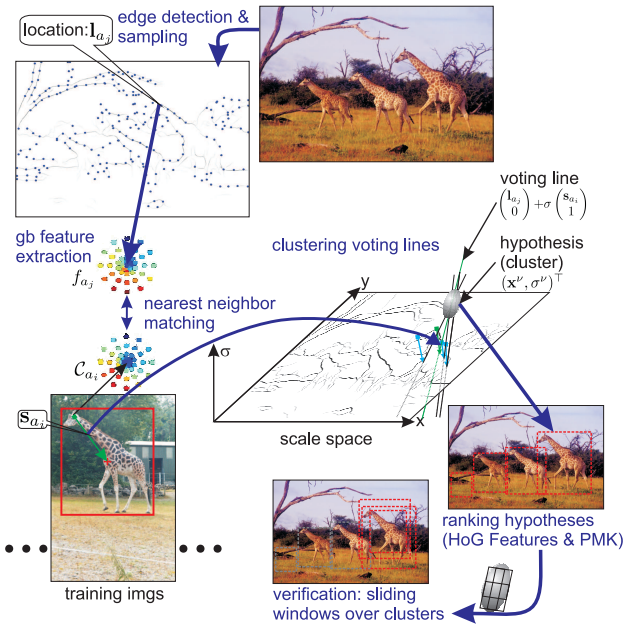

Figure 2. Outline of the processing pipeline

number of objects ahead of time. Moreover, clustering voting lines deals with the large number of false positives [17] which point votes produce and that hamper the commonly used local search heuristics such as binning [24].

Our Approach

Let us now briefly summarize the relevant processing steps for recognition, Fig. 2. To detect objects in a novel image, a probabilistic edge map is computed (we use [26]). Edge pixels are then uniformly sampled and represented using local features from a **single** scale (we utilize geometric blur [3]). Each descriptor is mapped onto similar features from training images which vote for an object hypothesis in scale space, *i.e*. object location and scale.

Without local scale estimates, each point in a query image casts votes for object hypotheses that form lines through scale space. Ideally, all points on an object would yield lines that intersect in a single point. Due to intra-category variation, and background clutter, the points of intersection are, however, degraded into scattered clouds. Finding these clusters becomes difficult since their number is unknown (it is the number of objects in the scene) and because the assignment of votes to objects is not provided (segmentation problem). To address these issues we frame the search for globally consistent object hypotheses as a weighted, pairwise clustering of local votes without scale estimates.

To find globally consistent hypotheses all the voting lines are grouped using weighted agglomerative clustering. The resulting clusters constitute the candidate hypotheses (on the order of ten per image). Centered at these candidate locations, object descriptors (multi-scale grid of histograms of gradients [22]) are computed and classified using a SVM. The classification probabilities are then used to obtain the final ranking of the hypotheses. Ultimately, a verification stage concludes the approach by evaluating object descriptors in the local neighborhood of the candidate hypotheses to take account of the uncertainty in each of the clusters.

In Sect. 4 we show experimental results on the challenging ETHZ shape dataset, which features large variations in scale. The evaluation demonstrates that we are able to detect objects with a scale variation of roughly 2.5 octaves although our features come from only a single scale. This is a significant advantage over approaches such as sliding windows [9], that require a dense sampling of scales with step widths as low as a $1/5$ octave.

To deal with the large amount of information in a scene, we follow a hierarchical approach. In successive stages, the number of entities significantly decreases (from pixels over curves to hypotheses), while individual entities are becoming more global, *i.e*. they capture information that is backed up by an increasing number of image pixels. Starting with 10^5 – 10^6 image pixels, boundary contours are detected which consist of 10^3 – 10^4 points. From the boundary contours points are sampled to produce 10^2 voting lines. Clustering then yields on the order of 10 hypotheses per image and the verification stage finally selects the correct hypotheses. Thus, our approach reduces the set of candidate hypotheses by several orders of magnitude more than search strategies such as branch-and-bound [21]. For same sized images, [21] gives an upper bound of 20,000 hypotheses that remain after branch-and-bound compared to the 20 hypotheses our approach has to check.

2. Voting Approaches to Recognition

A wide range of object models have been proposed to represent objects based on local measurements in an image. These models differ in the amount and complexity of (spatial) relationships they establish between the local measurements. These range from bag-of-features approaches [8] and latent topic models [32] without spatial constraints, to more involved spatial models such as star graph shaped models [23, 30], k-fans [7], compositional models [28, 29], and latent scene models [33]. Rich spatial relationships have been represented by joint models of all parts such as constellation models [12], pictorial structures [11], shape matching [2] and by regular grid like models that act as comparably rigid templates of objects [22, 9]. We focus on voting approaches since they are effective in dealing with the complexity of object models.

We sample semi-local features [3] uniformly along the contours. Based on some local relatedness measure, these local observations could be grouped as proposed by Ferrari *et al*. [14, 15]. Such a bottom-up approach is, however, susceptible to produce groupings that are globally inconsistent—at least when extended groupings are formed,

e.g. points might be grouped along an object contour that passes into a shadow contour in the background. Therefore, we follow a clustering approach that considers all local observations *jointly* and we explicitly model their individual uncertainty w.r.t. object scale. That way, we avoid having to make local decisions concerning groupings or scale.

2.1. Probabilistic Hough Voting

Probabilistic, part-based models (*e.g.* [12, 24]) combine potentially large numbers of local features in a single model by establishing statistical dependencies between the parts and the object hypothesis, *e.g.* by modeling the probabilities for relative location of parts to the object center. Leibe et al. [23] propose a Hough voting scheme to obtain candidates for object hypotheses. The Hough accumulator $\mathcal{H}^{\text{ha}}$ approximates a probability distribution $p(c, \mathbf{x}, \sigma)$ over scale space—here c denotes the category of an object hypothesis and $\mathbf{x}, \sigma$ are its location and scale in scale space. Local parts, which are represented by feature vectors $f_j \in \mathbb{R}^N$ and detected at image location $\mathbf{l}_j \in \mathbb{R}^2$ and scale $\sigma_j \in \mathbb{R}$, are assumed to be independent,

$$\mathcal{H}^{\text{ha}}(c, \mathbf{x}, \sigma) \propto \sum_j p\big(\mathbf{x}, \sigma \big| c, f_j, \mathbf{l}_j, \sigma_j\big)\, p\big(c \big| f_j, \mathbf{l}_j, \sigma_j\big). \quad (1)$$

Let $\mathcal{C}_i$ denote the i-th training sample or the i-th codebook vector, depending on whether a nearest-neighbor approach [4] or vector quantizion is used. Moreover, $\mathcal{C}_i$ has a shift $\mathbf{s}_i \in \mathbb{R}^2$ from the object center in the respective training image. All training images are assumed to be scale normalized, *i.e.* they have been rescaled so that objects are the same size. Now we can marginalize over $\mathcal{C}_i$ and $\mathbf{s}_i$ to obtain

$$\begin{aligned}\mathcal{H}^{\text{ha}}(c, \mathbf{x}, \sigma) &\propto \sum_{j,i} p\big(\mathbf{x}, \sigma \big| c, \mathcal{C}_i, \mathbf{s}_i, f_j, \mathbf{l}_j, \sigma_j\big) \\ &\quad \times P\big(c \big| \mathcal{C}_i, \mathbf{s}_i, f_j, \mathbf{l}_j, \sigma_j\big) p\big(\mathcal{C}_i, \mathbf{s}_i \big| f_j, \mathbf{l}_j, \sigma_j\big) \quad (2)\\ &= \sum_{j,i} p\big(\mathbf{x}, \sigma \big| c, \mathbf{s}_i, \mathbf{l}_j, \sigma_j\big)\; P\big(c \big| \mathcal{C}_i\big)\; p\big(\mathcal{C}_i \big| f_j\big) \quad (3)\\ &= \sum_{j,i} p\big(\mathbf{x} - \mathbf{l}_j - \sigma_j \mathbf{s}_i, \sigma - \sigma_j \big| c\big)\; P\big(c \big| \mathcal{C}_i\big)\; p\big(\mathcal{C}_i \big| f_j\big) \quad (4)\end{aligned}$$

The main assumption in this derivation is that only relative shifts of local parts from the object center are informative, not the absolute positions on their own.

2.2. Finding Candidate Hypotheses is Problematic

The main problem with $\mathcal{H}^{\text{ha}}$ is that we are interested in maximizers of a complex, continuous function that is highly non-concave. Furthermore, this continuous function in the infinite scale space has to be obtained by interpolation based on a comparably small set of points—the votes. Let K denote the kernel function, $b(\sigma)$ is the adaptive kernel bandwidth, $V_b(\sigma)$ is a scale-dependent normalization, and $d : \mathbb{R}^3 \times \mathbb{R}^3 \mapsto \mathbb{R}$ denotes a distance function in scale space. Then (4) can be approximated using the balloon density estimator [6] to obtain

$$\begin{aligned}\mathcal{H}^{\text{ha}}(c, \mathbf{x}, \sigma) \approx \frac{1}{V_b(\sigma)} \sum_{j,i} K\left(\frac{d\Big[(\mathbf{x}, \sigma)^\top; (\mathbf{l}_j + \sigma_j \mathbf{s}_i, \sigma_j)^\top\Big]}{b(\sigma)}\right) \\ \times P\big(c \big| \mathcal{C}_i\big)\, p\big(\mathcal{C}_i \big| f_j\big)\;. \quad (5)\end{aligned}$$

After this interpolation the second problem concerns finding candidate hypotheses $\mathcal{S}_c$ for category c,

$$\mathcal{S}_c := \big\{(\mathbf{x}^1, \sigma^1)^\top, (\mathbf{x}^2, \sigma^2)^\top, \ldots\big\}\;. \quad (6)$$

The common approach is to threshold $\mathcal{H}^{\text{ha}}(c, \mathbf{x}, \sigma)$ and find its local maxima. Due to the complexity of this objective function (many local maxima), exact optimization is computationally intractable. Therefore, [24] presents a heuristic approach that discretizes scale space and searches over a discrete grid of bins. Thresholding of $\mathcal{H}^{\text{ha}}(c, \mathbf{x}, \sigma)$ discards irrelevant bins and a successive refinement procedure localizes the hypotheses more accurately within the bins.

2.3. Characteristics of Our Approach

We present an alternative voting approach that directly obtains candidate hypotheses. The standard approach, which we have summarized above, takes discrete votes, *interpolates* to obtain a function over a continuous space, *discretizes* this space, and *searches* for local maxima. Our approach takes the voting lines and *clusters* them in scale space to directly obtain candidate hypotheses. We differ from Hough voting techniques such as [23] and the more recent max-margin version [27] in the following aspects (see App. A for a detailed comparison):

1. Scale estimation based on complete object hypotheses *instead of* ill-posed local scale estimation or search over scales. As a consequence, votes are straight lines in scale space instead of points. Uncertainty is overcome by finding concerted, global hypotheses based on all votes rather than based on local estimates.
2. Accurate "voting lines" are used directly *instead of* blurring "vote points" with adaptive kernels.
3. Direct clustering of voting lines (global optimization) *instead of* search for local maxima and thresholding in vote space.

Moreover, we match local features using approximate nearest neighbors rather than using vector quantization.

3. Vote Clustering with Global Scale Estimation

3.1. Voting Lines

The scale of objects in images is a global property which renders local scale estimation notoriously brittle. Only by

combining all the local observations in an image we can expect to obtain robust scale estimates. The uncertainty in scale σ_j of local features f_j affects the voting procedure: rather than voting for points in scale space, each local feature now votes for a line through scale space.

Let f_{a_j} be a local feature at location $\mathbf{l}_{a_j}$ in a query image that is mapped to a similar feature $\mathcal{C}_{a_i}$ from the training set. An efficient method for finding these related features are approximate nearest neighbor techniques [4]. A match is then represented by the vector $a = (a_i, a_j)^\top \in \mathbb{N}^2$ and we refer to it as the *voting line* a or simply just as a *vote*. Moreover, $\mathcal{C}_{a_i}$ has a shift $\mathbf{s}_{a_i} \in \mathbb{R}^2$ from the object center in the respective training image. As illustrated in Fig. 1 c), the votes for the object center $(\mathbf{x}, \sigma)^\top$ are lines through scale space, parametrized by the unknown object scale σ,

$$\begin{pmatrix}\mathbf{x}\\ \sigma\end{pmatrix} = \underbrace{\begin{pmatrix}\mathbf{l}_{a_j}\\ 0\end{pmatrix}}_{=\widehat{\mathbf{l}}_{a_j}} + \sigma \underbrace{\begin{pmatrix}\mathbf{s}_{a_i}\\ 1\end{pmatrix}}_{=\widehat{\mathbf{s}}_{a_i}} . \tag{7}$$

3.2. Candidate Hypotheses by Clustering Voting Lines

Rather than discretizing and searching through the continuous voting space, the following presents an approach that directly retrieves object hypotheses. Ideally, all the voting lines (7) intersect in a single point in scale space for each object as illustrated in Fig. 1 d). However, due to intra-class variations these points of intersection are rather blurred point clouds and mismatches and votes from the background yield considerable, additional clutter. An additional problem arises since the number of objects is unknown. All these challenges motivate a pairwise clustering approach [18]. In contrast to central clustering which requires the number of centroids to be given, the pairwise setting requires only a threshold on the distance between different prototypes. This threshold is directly accessible: To evaluate whether the bounding box $\mathcal{B}^{\text{hyp}}$ defined by the hypothesis $(\mathbf{x}, \sigma)^\top$ for category c is correct we use the standard PASCAL VOC [10] criterion. This requires that the intersection of a predicted hypothesis with the ground truth is greater than half the union of both

$$\text{bounding box } \mathcal{B}^{\text{hyp}} \text{ correct} \Leftrightarrow \frac{A(\mathcal{B}^{\text{hyp}} \cap \mathcal{B}^{\text{gt}})}{A(\mathcal{B}^{\text{hyp}} \cup \mathcal{B}^{\text{gt}})} > \frac{1}{2} . \tag{8}$$

Obviously, multiple matches onto the same ground truth object also count as mismatches. Therefore, the threshold for pairwise clustering is given by the minimal overlap between hypotheses as specified in the PASCAL criterion (8). Otherwise multiple, strongly overlapping hypotheses would be produced, which would lead to additional false positives.

Let us now define the cost function for our setting of weighted pairwise clustering. The goal is to compute assignments $\mathbf{M}_{a\nu}$ of votes a (specified by (7)) to one of K hypotheses $(\mathbf{x}^\nu, \sigma^\nu)^\top$. The matrix $\mathbf{M} \in \{0,1\}^{N\times K}$ captures many-to-one assignments, *i.e.*, $\sum_\nu \mathbf{M}_{a\nu} = 1$. Let $\mathbf{D}_{ab}$ denote the distance in scale space between two votes and w_{ac} is the weight given to vote a when the object is from class c. The goal is then to find assignments $\mathbf{M}$ that minimize the cost function

$$\mathcal{H}^{\text{pc}}(c, \mathbf{M}) := \sum_\nu \sum_a \mathbf{M}_{a\nu} w_{ac} \frac{\sum_b \mathbf{M}_{b\nu} \mathbf{D}_{ab}}{\sum_b \mathbf{M}_{b\nu}} . \tag{9}$$

The underlying rationale is that the fraction to the right computes the average dissimilarity between votes a and hypotheses ν. $\mathcal{H}^{\text{pc}}$ sums these individual contributions up. A classical way for obtaining approximate solutions to this cost function is hierarchical, agglomerative clustering [18]. Therefore, we use a weighted version of *Ward's Method*: Given $\mathbf{D}_{ab}$ and w_{ac}, Ward computes $\mathbf{M}_{a\nu}$ by grouping votes using a minimum variance approach. The weights w_{ac} are the same as in (4),

$$w_{ac} := P\big(c\big|\mathcal{C}_{a_i}\big)\, p\big(\mathcal{C}_{a_i}|f_{a_j}\big) . \tag{10}$$

The discrete probability distribution to the left is estimated using a discriminative approach, *i.e.* an SVM with probabilistic output (LibSvm [5] with rbf-kernel). The second distribution captures how far a matched training sample $\mathcal{C}_{a_i}$ is from the observed feature f_{a_j}. The Gibbs distribution represents this distance in terms of a probability

$$p\big(\mathcal{C}_{a_i}|f_{a_j}\big) = \frac{\exp\left(-\|f_{a_j} - \mathcal{C}_{a_i}\|\right)}{\sum_{a_i} \exp\left(-\|f_{a_j} - \mathcal{C}_{a_i}\|\right)} . \tag{11}$$

3.3. Distances between Voting Lines

For two votes a and b, there exists one point in scale space (an object hypothesis) that is most consistent with both.[1] This is the point that is closest to both lines and σ denotes its scale. The distance between the two votes a and b is the distance between their voting lines,

$$\mathbf{D}_{ab} := \left| \left\langle \frac{\widehat{\mathbf{s}}_{a_i} \times \widehat{\mathbf{s}}_{b_i}}{\|\widehat{\mathbf{s}}_{a_i} \times \widehat{\mathbf{s}}_{b_i}\|}, \widehat{\mathbf{l}}_{a_j} - \widehat{\mathbf{l}}_{b_j} \right\rangle \right| \cdot \frac{1}{\sigma} \tag{12}$$

This distance is normalized with the scale σ of the predicted object hypothesis, since large distances on coarser scales correspond to smaller distances on finer scales. The scale $\sigma = \frac{\sigma_a + \sigma_b}{2}$ of the point that is closest to both votes is

$$(\sigma_a, \sigma_b) = \operatorname*{argmin}_{\sigma_a, \sigma_b} \left\| \widehat{\mathbf{l}}_{a_j} + \sigma_a \widehat{\mathbf{s}}_{a_i} - \widehat{\mathbf{l}}_{b_j} - \sigma_b \widehat{\mathbf{s}}_{b_i} \right\| \tag{13}$$

$$\Leftrightarrow \sigma_a \widehat{\mathbf{s}}_{a_i} - \sigma_b \widehat{\mathbf{s}}_{b_i} = \widehat{\mathbf{l}}_{b_j} - \widehat{\mathbf{l}}_{a_j} - \left\langle \frac{\widehat{\mathbf{s}}_{a_i} \times \widehat{\mathbf{s}}_{b_i}}{\|\widehat{\mathbf{s}}_{a_i} \times \widehat{\mathbf{s}}_{b_i}\|}, \widehat{\mathbf{l}}_{b_j} - \widehat{\mathbf{l}}_{a_j} \right\rangle \frac{\widehat{\mathbf{s}}_{a_i} \times \widehat{\mathbf{s}}_{b_i}}{\|\widehat{\mathbf{s}}_{a_i} \times \widehat{\mathbf{s}}_{b_i}\|} \tag{14}$$

Given the three equations in (14) with two unknowns, a closed form solution can be computed analytically—we skip it here merely for the length of this formula.

[1] In the special case of two parallel voting lines, consistent hypotheses form a line parallel to these voting lines rather than a single point.

3.4. From Clusters to Object Hypotheses

Agglomerative clustering yields an assignment matrix $\mathbf{M}_{a\nu}$ as a solution to the weighted, pairwise clustering problem of (9). Given the assignments, the object hypothesis that corresponds to each cluster can be computed as a weighted average of all the assigned votes,

$$\begin{pmatrix} \mathbf{x}^\nu \\ \sigma^\nu \end{pmatrix} = \sum_a \frac{\mathbf{M}_{a\nu} w_{ac}}{\sum_a \mathbf{M}_{a\nu} w_{ac}} \left(\widehat{\mathbf{l}}_{a_j} + \sigma_a \widehat{\mathbf{s}}_{a_i} \right) . \qquad (15)$$

Rather than computing local scale estimates and searching over scales, all voting lines directly vote for a concerted hypothesis of object scale. App. A concludes this presentation by comparing our algorithm with Hough voting.

3.5. Ranking Candidate Hypotheses

After clustering voting lines, the resulting object hypotheses have to be ranked. We investigate two ranking schemes.

A) Ranking using Weights w_{ac}: The score $\zeta_\nu^{\mathrm{M}w}$ for cluster ν is the weighted sum over the weights of all voting lines which are assigned to cluster ν,

$$\zeta_\nu^{\mathrm{M}w} := \sum_a \mathbf{M}_{a\nu} w_{ac} \frac{\sum_b \mathbf{M}_{b\nu} \mathbf{D}_{ab}}{\sum_b \mathbf{M}_{b\nu}} . \qquad (16)$$

B) PMK Ranking: A cluster ν receives a ranking score ζ_ν^{PMK} by applying a SVM classifier to the corresponding image window. We use the pyramid match kernel (PMK) [16, 22] with histograms of oriented gradients as features [2] Positive examples for a class are the groundtruth bounding boxes which are rescaled to the average bounding box diagonal length of the class to assure that all samples are on the same scale. To obtain negative samples for the SVM, we run our voting method on the positive training images and pick the false positive hypotheses. In the spirit of classifier boosting this approach collects difficult negative samples and we do not require any negative training images.

3.6. Verification of Candidate Hypotheses

Ideally each cluster would just be a single point in scale space but intra-class variations and noise lead to scattered clusters. In the verification stage, the SVM classifier from Sect. 3.5 is run in sliding window mode over a grid of locations around each cluster centroid. The grid is obtained by rotating it according to the principal components of the vote cloud for the respective cluster. The size of the grid is two standard deviations in each direction and each direction is regularly divided into 5 cells. The verification stage refines the candidate hypotheses by performing a local search around each candidate from the first round.

[2]The implementation is as follows: Histograms have 9 orientations. Image windows are sampled on 4 regular grids of different resolution. Successive ones are one octave apart from each other and are weighted with a scale-dependent factor of 2^{l-1}, where $l = 1$ is the coarsest scale.

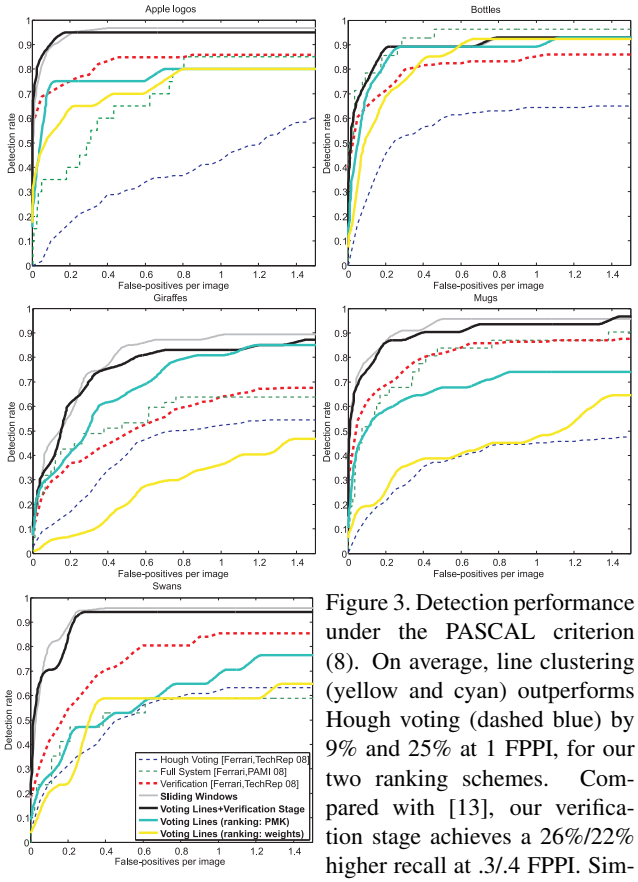

Figure 3. Detection performance under the PASCAL criterion (8). On average, line clustering (yellow and cyan) outperforms Hough voting (dashed blue) by 9% and 25% at 1 FPPI, for our two ranking schemes. Compared with [13], our verification stage achieves a 26%/22% higher recall at .3/.4 FPPI. Similarly, we observe a 20%/17% improvement over [15].

4. Experimental Evaluation

To evaluate our approach, we choose the challenging ETHZ Shape Dataset containing five diverse object categories with 255 images in total. All categories feature significant scale changes and intra-class variation. Images contain one or more object instances and have significant background clutter. We use the latest experimental protocol of Ferrari *et al*. [15]: to train our detector on a category, we use half the positive examples for that class. No negative training images are used and we test on all remaining images in the dataset (like [15] we average the results over five random splits). The detection performance is measured based on the rigid PASCAL criterion given in (8).

4.1. Performance of Clustering Voting Lines

Fig. 3 compares our vote clustering approach with the method of Ferrari *et al*. [15] by plotting recall against false positives per image (fppi). It turns out that our line clus-

tering with the PMK ranking procedure from Sect. 3.5 significantly outperforms the Hough voting used by [15]. The average gain is 25% and even the simple method proposed in Eq. 16 achieves a gain of more than 9%, *c.f*. Tab. 1. Interestingly, our vote clustering alone performs better than Hough voting together with the verification stage in [15] on giraffes and bottles. Here, clustering provides hypotheses that are already comparably accurate and do not require significant model refinement by the verification stage. Furthermore, we compare our results to [27] who cast the Hough voting in a discriminative, maximum-margin setting. Tab. 1 shows that our vote clustering with the PMK ranking provides an improvement of approximately 18% and even ranking by a simple summation over all our discriminatively learned weights yields a gain of 2.1%. These performance improvements underline the effectiveness of modeling local votes as lines and clustering them to provide globally consistent hypotheses.

4.2. Results of the Combined Detector

Although the main goal of this paper is to improve the hypothesis generation of voting methods, we also investigate the combined detector consisting of vote clustering and a verification stage that performs model refinement Fig. 3. Compared to KAS [13], our approach improves the average detection rate by 22% to 26% at .4 and .3 fppi, respectively (Tab. 1). Similarly, we obtain a gain between 17% and 20% over the full system of [15]. For completeness, Tab. 1 also gives the results of [27], although they follow a different experimental protocol: they use twice as many training samples (positive images and an equal number of negative ones) and they expand the voting space to also include aspect ratio (this is crucial for the giraffes). Taking these aspects into account, it is fair to say that our verification stage performs at least comparable to theirs. Finally, we compare our combined detector to a sliding window detector. Therefore, the SVM classifier from Sect. 3.6 is run in sliding window mode over the whole image (step width of 8 pixels) and over all scales (successive scales differ by a factor of $2^{1/8}$). This exhaustive search procedure evaluates on the order of 10^4 hypotheses per image whereas the vote clustering retrieves an average number of 20 candidates. Consequently, we observe a gain in computational speed that is between two and three orders of magnitude. The central part of voting using line clustering has negligible running time (on the order of a second) compared to the other processing steps (computing geometric blur features etc.). This significant reduction of computational complexity results in a 1.5% lower recall, which we believe is tolerable given the significant reduction of the candidate set.

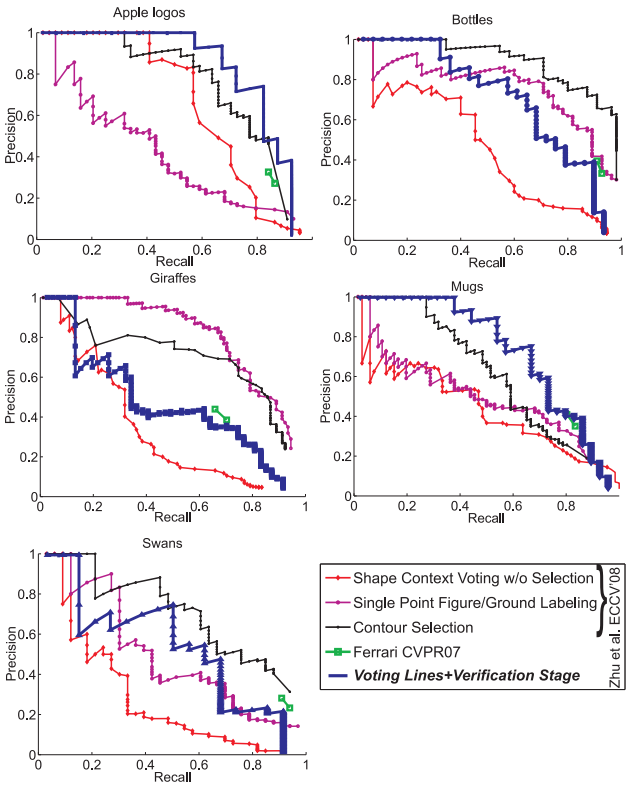

Figure 4. Comparing our full system, (trained only on bounding boxes from positive iamges) against the approach of Zhu *et al.* (trained on manually drawn shape models). [15, 14] have shown that training on bounding boxes is significantly harder—they observed a 13% lower recall at .4 FPPI.

4.3. Comparing against a Supervised Approach

All our approach requires are positive training images where the objects are marked with a bounding box. Let us now compare this weakly supervised approach with the contour context selection of Zhu *et al*. [35] who require a manually drawn shape model for each category. The experiments by Ferrari *et al*. [14] show that learning from bounding boxes is a significantly harder problem: they report a 9% lower recall at 0.4 FPPI for the bounding box training (using positive and negative training images) as opposed to manually drawn models (76.8% vs. 85.3%). When using only positive samples [15] as we also do, the gap increases to 13% (72.0% vs. 85.3%). Fig. 4 compares the precision/recall of our full approach (line clustering and verification stage) against the different methods of [35]. Unfortunately we cannot compare the average precision as we are lacking the actual values of [35]. However, a visual inspection of the curves indicates that our method performs better on apple logos and mugs, but it is outperformed on the other three categories. Since we are dealing with a significantly harder task, an improvement on two out of three categories is noteworthy.

Category	*Voting Stage (FPPI = 1.0 FPPI)*				*Verification Stage (FPPI = 0.3 / 0.4)*				
	w_{ac} **rank** Eq. 16	**PMK rank** Sect. 3.5	Hough [15]	M^{2}HT [27]	**Verification** Sect. 3.6	Sliding Windows	KAS [13]	Full system [15]	M^{2}HT+ IKSVM [27]
Apples	80.0	80.0	43.0	85.0	95.0 / 95.0	95.8 / 96.6	50.0 / 60.0	77.7 / 83.2	95.0 / 95.0
Bottles	92.4	89.3	64.4	67.0	89.3 / 89.3	89.3 / 89.3	92.9 / 92.9	79.8 / 81.6	92.9 / 96.4
Giraffes	36.2	80.9	52.2	55.0	70.5 / 75.4	73.9 / 77.3	49.0 / 51.1	39.9 / 44.5	89.6 / 89.6
Mugs	47.5	74.2	45.1	55.0	87.3 / 90.3	91.0 / 91.8	67.8 / 77.4	75.1 / 80.0	93.6 / 96.7
Swans	58.8	68.6	62.0	42.5	94.1 / 94.1	94.8 / 95.7	47.1 / 52.4	63.2 / 70.5	88.2 / 88.2
Average	63.0	78.6	53.3	60.9	87.2 / 88.8	88.9 / 90.1	61.4 / 66.8	67.2 / 72.0	91.9 / 93.2

Table 1. Comparing the performance of various methods. Detection rates (in [%]) are measured using the PASCAL criterion (8). The approach of [27] is not directly comparable, since they are using twice the amount of training samples we use (additionally to the positive samples an equally sized subset of negative images). Our clustering of voting lines (using PMK ranking) yields a 25% higher performance than the Hough voting in [15] and a 18% gain over max-margin Hough voting [27].

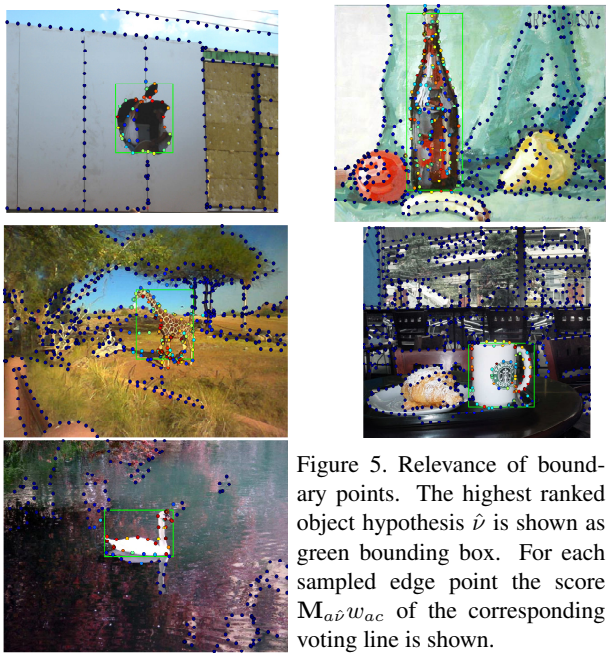

Figure 5. Relevance of boundary points. The highest ranked object hypothesis $\hat{\nu}$ is shown as green bounding box. For each sampled edge point the score $\mathbf{M}_{a\hat{\nu}} w_{ac}$ of the corresponding voting line is shown.

4.4. Relevance of Boundary Points

Our approach infers candidate hypotheses by establishing a sparse object representation based on voting lines at subsampled edge points. Fig. 5 shows the sample points and plots the highest ranked bounding box hypothesis. After clustering the voting lines, each voting line assigned to this hypothesis $\hat{\nu}$ receives a positive weight $\mathbf{M}_{a\hat{\nu}} w_{ac}$. We plot this score for all the sample points. Interestingly, nearly all of the background points are suppressed since their voting lines are clustered into other candidate hypotheses. Similarly, the reflection of the swan on the water is also discarded because it is inconsistent with the predicted hypothesis. Moreover, we obtain a lot of high votes on the object boundary. However, the bottle image also features strong responses on the front of the bottle where many training samples have shown the labeling of the bottle. Our algorithm appears to have learned these regions which are rich in contours. In conclusion the plots show that our approach is effective in segregating figure from background.

5. Discussion and Conclusion

We have presented a simple yet effective approach for combining large quantities of local object votes into globally consistent object hypotheses. The basis for this method is to explicitly model the scale-location-ambiguity of local image observations leading to voting lines through scale space. To estimate object location and scale jointly based on all the uncertain local votes, we have proposed a pairwise clustering of voting lines. The approach condenses a large number of local votes into a few relevant hypotheses, thereby reducing the number of candidates by three orders of magnitude compared to the leading sliding window paradigm. The line clustering procedure has shown a significant performance gain over existing Hough voting methods for object detection.

A. Hough Voting vs. Clustering Voting Lines

Alg. 1 summarizes the main processing steps of probabilistic Hough voting as described by Leibe et al. in [24]. The approach of clustering voting lines is then presented in Alg. 2. A comparison of both procedures reveals that Alg. 1 goes a detour by taking a discrete set of votes, interpolating them in the continuous voting space, and discretizing that space again to search through it. Alg. 2 clusters the original votes to obtain the hypotheses directly. Thereby, a search through the voting space is not needed. Moreover, the proposed approach does not require local scale estimates but computes object scale based on the correspondence of all votes.

Algorithm 1 Detecting objects from category c in a query image using probabilistic Hough voting.

1 *Map* features $f_j \mapsto$ Codebook vectors $\mathcal{C}_i$
2 $\sigma_j \leftarrow$ Local *scale estimation* at locations $\mathbf{l}_j$
3 *Vote* into Hough accumulator
4 *Interpolate* votes using Kernel density estimation
5 *Threshold* votes & *discretize* voting space into bins
6 $\left\{(\mathbf{x}^\nu, \sigma^\nu)^\top\right\}_\nu \leftarrow$ *Search* for local maxima

Algorithm 2 Detecting objects from category c in a query image by clustering voting lines.

1 *Map* features $f_{a_j} \mapsto$ nearest training samples $\mathcal{C}_{a_i}$
2 $\mathbf{D}_{ab} \leftarrow$ *Distances* between votes (12)
3 $\mathbf{M}_{a\nu} \leftarrow$ Agglomerative *clustering* of $\mathbf{D}_{ab}$ to solve (9)
4 $\left\{(\mathbf{x}^\nu, \sigma^\nu)^\top\right\}_\nu \leftarrow$ *Apply* (15) to $\mathbf{M}_{a\nu}$

References

[1] D. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2), 1981.

[2] A. C. Berg, T. L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondence. In *CVPR*, pages 26–33, 2005.

[3] A. C. Berg and J. Malik. Geometric blur for template matching. In *CVPR*, pages 607–614, 2001.

[4] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification.CVPR'08.

[5] C.-C. Chang and C.-J. Lin. *LIBSVM: A library for support vector machines*, 2001.

[6] D. Comaniciu, V. Ramesh, and P. Meer. The variable bandwidth mean shift and data-driven scale selection. In *ICCV*, pages 438–445, 2001.

[7] D. J. Crandall, P. F. Felzenszwalb, and D. P. Huttenlocher. Spatial priors for part-based recognition using statistical models. In *CVPR*, pages 10–17, 2005.

[8] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *ECCV. Workshop Stat. Learn. in Comp. Vis.*, 2004.

[9] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005.

[10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results. http://www.pascal-network.org/challenges/VOC/voc2008/workshop.

[11] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1):55–79, 2005.

[12] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, pages 264–271, 2003.

[13] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid. Groups of adjacent contour segments for object detection. *PAMI*, 30(1):36–51, 2008.

[14] V. Ferrari, F. Jurie, and C. Schmid. Accurate object detection with deformable shape models learnt from images. In *CVPR*, 2007.

[15] V. Ferrari, F. Jurie, and C. Schmid. From images to shape models for object detection. Technical Report 6600, INRIA, Grenoble, 2008.

[16] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV*, 2005.

[17] W. Grimson and D. Huttenlocher. On the sensitivity of the hough transform for object recognition. *PAMI*, 12(3):255–274, 1990.

[18] T. Hofmann and J. M. Buhmann. Pairwise data clustering by deterministic annealing. *PAMI*, 19(1):1–14, 1997.

[19] P. Hough. Method and means for recognizing complex patterns. *U.S. Patent 3069654*, 1962.

[20] T. Kadir and M. Brady. Saliency, scale and image description. *IJCV*, 45(2), 2001.

[21] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *CVPR*, 2008.

[22] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, pages 2169–2178, 2006.

[23] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV. Workshop Stat. Learn. in Comp. Vis.*, 2004.

[24] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *IJCV*, 77(1-3):259–289, 2008.

[25] D. Lowe. Object recognition from local scale-invariant features. In *ICCV*, pages 1150–1157, 1999.

[26] M. Maire, P. Arbelaez, C. Fowlkes, and J. Malik. Using contours to detect and localize junctions in natural images. In *CVPR*, 2008.

[27] S. Maji and J. Malik. Object detection using a max-margin hough transform. In *CVPR*, 2009.

[28] B. Ommer and J. M. Buhmann. Learning the compositional nature of visual objects. In *CVPR*, 2007.

[29] B. Ommer, T. Mader, and J. Buhmann. Seeing the objects behind the dots: Recognition in videos from a moving camera. *IJCV*, 83(1):57–71, 2009.

[30] A. Opelt, A. Pinz, and A. Zisserman. Incremental learning of object detectors using a visual shape alphabet. In *CVPR*, pages 3–10, 2006.

[31] J. Shotton, A. Blake, and R. Cipolla. Contour-based learning for object detection. In *ICCV*, 2005.

[32] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering objects and their localization in images. In *ICCV*, pages 370–377, 2005.

[33] E. B. Sudderth, A. B. Torralba, W. T. Freeman, and A. S. Willsky. Learning hierarchical models of scenes, objects, and parts. In *ICCV*, pages 1331–1338, 2005.

[34] P. A. Viola and M. J. Jones. Robust real-time face detection. *IJCV*, 57(2):137–154, 2004.

[35] Q. H. Zhu, L. M. Wang, Y. Wu, and J. B. Shi. Contour context selection for object detection: A set-to-set contour matching approach. In *ECCV*, 2008.