INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Groups of Adjacent Contour Segments for Object Detection

Vittorio Ferrari — Loic Fevrier — Frederic Jurie — Cordelia Schmid

N° 5980

September 2006

Thème COG

apport de recherche

ISSN 0249-6399 ISRN INRIA/RR--5980--FR+ENG

Groups of Adjacent Contour Segments for Object Detection

Vittorio Ferrari , Loic Fevrier , Frederic Jurie , Cordelia Schmid

Thème COG — Systèmes cognitifs
Projet LEAR

Rapport de recherche n° 5980 — September 2006 — 31 pages

Abstract: We present a family of scale-invariant local shape features formed by chains of k connected, roughly straight contour segments (kAS), and their use for object class detection. kAS are able to cleanly encode pure fragments of an object boundary, without including nearby clutter. Moreover, they offer an attractive compromise between information content and repeatability, and encompass a wide variety of local shape structures. We also define a translation and scale invariant descriptor encoding the geometric configuration of the segments within a kAS, making kAS easy to reuse in other frameworks, for example as a replacement or addition to interest points.

We demonstrate the high performance of kAS within a simple but powerful sliding-window object detection scheme. Through extensive evaluations, involving eight diverse object classes and more than 1400 images, we 1) study the evolution of performance as the degree of feature complexity k varies and determine the best degree; 2) show that kAS substantially outperform interest points for detecting shape-based classes; 3) compare our object detector to the recent, state-of-the-art system by Dalal and Triggs [4].

Key-words: local features, shape descriptors, object detection

A software implementation is available at `lear.inrialpes.fr/software`

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe, 38334 Montbonnot Saint Ismier (France)
Téléphone : +33 4 76 61 52 00 — Télécopie +33 4 76 61 52 52

1 Introduction

In the last few years, the problem of recognizing object classes has received growing attention, in both variants of whole image classification [3, 5, 10, 14, 15], and object localization [2, 4, 16, 31]. The majority of existing methods use local image patches as basic features. While these work well for some object classes, such as motorbikes and cars, other classes are defined by their *shape*, and are therefore better represented by *contour* features (e.g. horses, or mugs). In spite of their substantial scope, only comparably few works [2, 13, 22, 29] have tackled the class-level localization problem using contour features.

In this paper we present a family of local contour features, and their application for detecting and localizing objects. These features are small groups of *connected*, approximately straight contour segments, called k *adjacent segments*, or kAS. The segments in a kAS form a path of length k through a network of contour segments covering the image [9]. Essentially, two segments are connected in the network if they are adjacent on the same edgel-chain, or if one is at the end of an edgel-chain directed towards the other segment (section 3). The larger the number k of segments in a kAS, the more complex the local shape structures it can capture. $1AS$ are just individual segments, while $2AS$ include L shapes, and $3AS$ can form C, F and Z shapes (figures 2, 3). Along with the kAS features, we propose a low dimensional, translation+scale invariant descriptor designed to encode the geometric properties of the segments composing a kAS, and their relative locations.

kAS have a several attractive properties. First, as both kAS and their descriptors cover solely short chains of connected segments, they have the ability to cover pure portions of an object boundary, without including clutter edges which very often lie in the vicinity. Second, for a sensible range of k, kAS have intermediate complexity, which makes them detectable repeatably while being informative at the same time. Third, connectedness is a natural grouping criterion to form kAS. It avoids the need for defining a 'grouping scale' or a 'grouping neighborhood' for a segment, and effectively constrains the features to be chains of segments, which are more likely of lying entirely on a boundary. Finally, kAS are complete local invariant features: each has a well defined location and scale, an invariant descriptor, and is detected based only on local properties of a single image. Hence, they can be reused effortlessly in a variety of recognition and image matching frameworks as a replacement or addition to interest points (such as [2, 5, 10, 16, 30]).

We demonstrate the power and flexibility of kAS within an object detection framework which brings together several successful ideas presented before. Following the 'bag of features' paradigm [3, 14, 34], we construct a codebook of *kAS types*, each capturing a different kind of local shape structure (figures 2 and 3). An image window is subdivided into tiles [4, 15] and each is described by a separate bag of kAS. In this fashion the window representation is composed of several bags of kAS spatially localized within the window. Adding this layer of spatial organization improves the discriminative power compared to a standard orderless bag of features over the entire window. We first train a classifier from example object and background windows, and then localize previously unseen instances in test images via a multi-scale sliding-window mechanism [4, 31] coupled with the classifier. Our method is rendered computationally efficient by organizing all image kAS in an Integral Histogram [24], which is a recently developed datastructure supporting the rapid computation of multidimensional histograms.

During an extensive evaluation, involving eight diverse object classes and over 1400 images (section 6), we study several aspects of kAS. First, we analyze the object detection performance while varying k, thereby shedding light on the relation between repeatability and informativeness as k increases. Second, for each k, we vary the resolution of the window tiling, allowing to observe the trade-off between adding localization information and reducing tolerance to spatial variations within the class. Interestingly, we find the optimal window tiling to relate to the complexity of the features (k), with simpler features preferring finer tiling. Moreover, we thoroughly compare the performance of kAS against interest points, and against the state-of-the-art object detection technique by Dalal and Triggs [4]. Their work is particularly relevant because it follows a similar detection framework (sliding-windows, tiles), but it applies different descriptors to the window tiles (simpler histograms of gradient orientations). Finally, we experiment with the application of kAS with different k at the same time, and with the combination of interest points and kAS.

2 Related works

In the following we first review object detection techniques based on contour features, for which kAS offer an alternative, and then present works on the perceptual grouping of contours, upon which kAS build.

Contour features for object class detection Selinger and Nelson [27] detect *key curves*: long segments of an edgel-chain comprised between two high curvature points. A key curve's size and orientation defines a square image patch, which is then described using all edgels falling within it. These edge patches attempt to strike a winning trade-off: be local, and hence bring robustness to occlusion and clutter, while also complex enough to be distinctive to some degree, enabling to match individual features, and opening the door to computationally efficient indexing schemes. However, these patches are likely to include clutter edgels lying near the object boundary, which corrupt their descriptors and makes them difficult to put in correspondence.

Selinger and Nelson's recognition system was demonstrated in controlled laboratory conditions, with clean images containing modest amounts of clutter, and mostly on the task of recognizing specific objects. Jurie and Schmid [13] were among the first to propose local contour features for the detection of object *classes*, and to test their system on real, cluttered images. Their scale-invariant feature detector responds to circular arcs of edgels, which are described by the spatial distribution of points in a thin annular neighborhood of the circle. This attempts to exclude clutter from the descriptor by avoiding encoding points inside the circle. As one limitation, circular arcs only cover a fairly restricted class of shapes.

In their very recent works, Shotton et al. [29] and Opelt et al. [22] independently propose to construct contour fragments tailored to a specific class. The idea is to explicitly construct fragments to occur frequently on positive training images of a class, while seldom in negative ones. Both works employ boosting to select fragments from a large pool of candidates, but differ in the way these candidates are constructed (random rectangles sampled from training segmentation masks in [29], whereas [22] grows fragments starting from random contour points, and optimizes their length so as to maximize Chamfer matching score and accuracy of object centroid prediction in validation images). Although they can be more discriminative for the learned class, this kind of fragments are harder to reuse within other recognition or image matching frameworks, compared to generic features, depending only on local properties of individual images. Moreover, the fragments of [29] are not scale invariant and need segmented training images to be produced, which further limits their applicability.

Berg et al. [2] offer an alternative view on contour-based object recognition, casting the problem as deformable shape matching. Instead of counting on sophisticated local features, they simply take individual edgels (with a Geometric Blur neighborhood descriptor), and put them in correspondence between pairs of images with a powerful non-rigid point matching algorithm based on Integer Quadratic Programming. The method obtains impressive results on the challenging Caltech101 database. One disadvantage is that it reduces recognition to matching pairs of training and test images, and doesn't infer from the training images a single model summarizing common properties shared by different instances of the class. Besides, it would be interesting to inject kAS in their framework, as replacecement for individual edgels, and observe whether this would lead to improved performance.

Dalal and Triggs [4] considerably advanced the state-of-the art in human detection, by designing the Histogram of Oriented Gradients (HoG) descriptor, and carefully optimizing it over a large dataset containing thousands of humans in unconstrained poses. In their recognition framework image windows are subdivided in tiles and each one is described by a HoG. A simple sliding-window mechanism then allows to localize objects. Photometric normalization within multiple overlapping blocks of tiles makes the method particularly robust to lighting variations. Notice that HoG descriptors are only defined within a given subwindow, they don't have a concept of location and scale. Hence, they need to be associated to some external feature detector before being applicable within frameworks not based on sliding-windows.

Perceptual grouping Perceptual grouping of contours has a long history in computer vision [6, 12, 17, 18, 25, 26, 28, 32]. The crucial idea behind these works is that pieces of contour related by some *perceptually salient* property are more likely to belong to the same object. The perceptual properties exploited include convexity [12], co-circularity [32], connectedness [26, 28], parallelism [18], and proximity[18].

One major area of application for perceptual grouping is image segmentation, in which the task is to group together all elements belonging to individual, unspecified objects [6, 12, 32]. Moreover, perceptual grouping played an important role in the recognition of specific objects under varying viewpoint, particularly in the 80s and 90s. The focus was mainly on planar objects [26] and polyhedra [11, 18].

The kAS features are motivated by the same general intuitions of earlier perceptual grouping works, and are most related the ideas of Rothwell [25, 26], who advocated for the importance of connectedness and topological relations. We believe that connectedness is a fundamental, powerful driving force which is currently still underexploited in computer vision. In this paper, connectedness is brought to the domain of object *class* detection, and is exploited to define modern local invariant features: image elements with a well defined location, a scale and an invariant descriptor, ready to be used in many recent matching and recognition schemes.

3 k adjacent segments (kAS)

3.1 Contour Segment Network

We summarize here the technique of [9] to build the *contour segment network* (CSN) of the image, on which we will detect our kAS features.

Edgels are detected by the excellent Berkeley natural boundary detector [20], and then chained. The resulting edgel-chains are linked at their discontinuities, i.e. two edgel-chains c_1 and c_2 are linked if c_2 passes near an endpoint of c_1, and if the ending of c_1 is directed towards c_2 (figure 1a). Informally, if c_1 would be extended a bit, it would meet c_2. These links are useful in two ways: they record that a contour might continue over the gap between two edgel-chains, and allow to capture junctions (L-junctions, T-junctions, and higher order junctions involving several edgel-chains).

The edgel-chains are partitioned into roughly straight contour segments. The idea is to organize these segments in a network, by connecting them along the edgel-chains, and across their links (figure 1a). Since every edgel-chain can be linked to several others, the CSN is a complex branching structure. Intuitively, two segments are connected if the edgels provide evidence that they might be adjacent along some object contour, even when they are physically separated by a (small) gap, or when forming a junction. The key property of the CSN is to include paths going along the contours of the imaged objects [9], which motivates kAS features.

3.2 Detecting kAS

The principal contribution of this paper is to propose a family of local features: paths of length k through the CSN. More formally, a group of k segments is a kAS iff they can be ordered so that the i-th segment is connected in the CSN to the $(i+1)$-th one, for $i \in \{1, k-1\}$. Hence we call them k *adjacent segments*, and refer to their length k as *degree*. As k grows, kAS can form more and more complex local shape structures: individual segments for $k=1$; L shapes and 2-segment T shapes for $k=2$; C, Y, F, Z shapes, 3-segment T shapes, and triangles for $k=3$ (figures 2, 3). The dimensionality of kAS descriptors also grows with k (next section), and we treat kAS of different degrees as different feature types, all united in one family by a shared crucial property: to be sequences of *connected* segments.

Connectedness provides a natural criterion for grouping segments into kAS. It avoids arbitrary definitions of the neighborhood of a segment, and constrains kAS to be chains

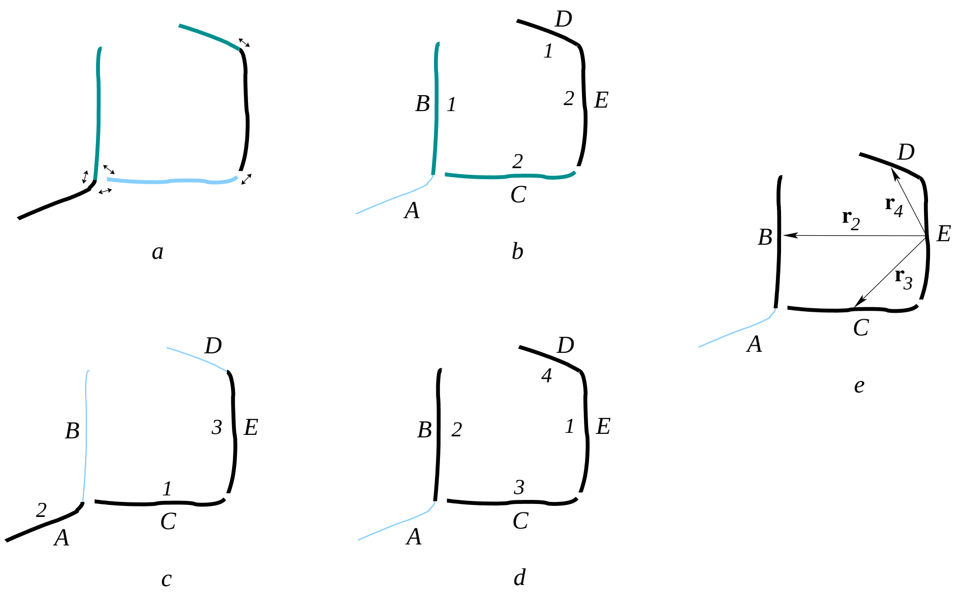

Figure 1: *a) Three edgel-chains, with five segments and their inter-connections (arrows) in the network. b) Two detected 2AS (B, C) and (D, E). The order of each segment in the descriptor is marked next to it. Notice that (A, B), (A, C), (C, E) are also detected, though not displayed because they overlap with (B, C) and (D, E). c) A 3AS (C, A, E). d) A 4AS (E, B, C, D). e)* $\mathbf{r}_i$ *vectors involved in the descriptor for the 4AS in d).*

of segments. Compared to the broader class of groups of 'nearby' segments, they have higher chances to lie entirely on a portion of the object boundary. The features of [13] instead, include disconnected sets of edgels which happen to be located along part of a circle. Besides, the key curves of [27] are based on individual edgel-chains, and hence are less robustly detected in real images than kAS, which bridge gaps between edgel-chains.

kAS can be detected by a depth-first search started from every segment, followed by the elimination of equivalent paths (two different paths involving the same segments constitute the *same* kAS). This is computationally cheap for the small values of k corresponding to local features (about $k \leq 5$). We disregard higher values of k because they result in large scale structures, too specific to a particular image or object instance, and in an excessive number of detected features (several thousands already for $k = 5$). More precisely, the number of kAS in an image containing n segments grow quickly with k, as can be understood by the following observations. On average, each segment is connected to two to three other, because T and higher orders junctions occur less frequently than simple 1-to-1 connections. As a consequence, as k grows, the number of paths of length k passing through a given branching point increases quickly. In practice, while the average number of 2AS is only about $1.5n$, the number of 3AS is $4n$, that of 4AS is $10n$, and there are more than $20n$ 5AS !

As k increases, features increase in complexity. On the one hand, they become more and more informative, while on the other they gradually get less and less repeatable across different images and object instances. Additionally, the number of non-boundary features (or mixed features covering partly boundary and partly clutter) also grows with k, actually faster than pure boundary ones, leaving a lower signal-to-noise ratio. Hence, for rather low values of k, kAS have an attractive intermediate complexity, offering a convenient compromise: simple enough to be detected repeatably, yet complex enough to capture informative local object structures. In section 6, we confirm these intuitions experimentally, and determine that 2AS perform best.

3.3 Describing kAS

In order to compare different kAS, we need a numerical descriptor. As first step, it is important to order the kAS segments $\{s_i\}_{i=1..k}$ in a repeatable manner, so that similar kAS have the same order. We select as first segment the one with midpoint closest to the centroid of all midpoints $\{m_i = (x_i, y_i)\}_{i=1..k}$ (when several segments have similar distances to the centroid, we pick the first one according to the order defined below). As we will see in the descriptor below, this centermost segment is the natural choice as reference point for measuring the relative location of the other segments. The remaining segments take up positions 2 through k, and are ordered from left to right, according to their midpoint. If two segments s_i, s_j have similar x coordinate, i.e. $(x_i - x_j) \leq 0.2\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$, then they are ordered from top to bottom. Note that this order is stable, as no two segments can have similar location in both x and y. Example orderings can be seen in figure 1b-d.

Once the order established, a kAS is a list $P = (s_1, s_2, \ldots, s_k)$ of segments. Let $\mathbf{r}_i = (r_i^x, r_i^y)$ be the vector going from the midpoint of s_1 to the midpoint of s_i. Furthermore, let θ_i, $l_i = \|s_i\|$ be the orientation and length of s_i. The descriptor of P is composed of $4k - 2$ values [1] (figure 1e):

$$\left(\frac{r_2^x}{N_d}, \frac{r_2^y}{N_d}, \ldots, \frac{r_k^x}{N_d}, \frac{r_k^y}{N_d}, \quad \theta_1, \ldots, \theta_k, \quad \frac{l_1}{N_d}, \ldots, \frac{l_k}{N_d} \right) \tag{1}$$

The distance N_d between the two farthest midpoints is used as normalization factor, making the descriptor scale-invariant (hence, both the kAS features and their descriptors are scale-invariant). While segment lengths are known to be often inaccurate, and each is based only on part of the kAS, the distance between the farthest midpoints makes a better choice for a reliable estimate of the kAS scale. In addition to a kAS scale, we also define its *location* to be the geometric center of the midpoints of its segments. Exact definitions of scale and location are useful when using kAS in higher level algorithms, such as in our sliding-window object detection scheme (next sections).

The proposed descriptor considers the segments as completely straight, so as to capture only the relevant information of the geometric configuration they form, and not the varying

[1] The case $k = 1$ makes exception. The descriptor is composed only of θ_1, and the scale of $1AS$ is defined as l_1.

details of the weak curvature along them. Moreover, we stress that only the k segments are described, and not other nearby edgels. In this fashion, we can cleanly encode a portion of an object boundary, without including inner/outer clutter (unlike [27]).

With its $4k - 2$ dimensions, the descriptor is also very compact. Indeed, since the intrinsic dimensionality of k straight segments is $4k$, and the dimensionality of the desired scale+translation invariance space is 3, the lowest dimensionality of a complete descriptor is $4k - 3$. The only redundant degree of freedom we encode is embedded within the relative location vectors $\{\mathbf{r}_i\}_{i=2..k}$, and factoring it out would require representing them in a more complicated way.

Interestingly, the kAS descriptor is of different nature than conventional local textured feature descriptors. While the latter encode the appearance of the pixel patch covered by the feature, the kAS descriptor encodes the geometric properties of the segments (orientation and length), and of their spatial arrangement ($\{\mathbf{r}_i\}_{i=2..k}$).

If desired, the descriptor can be easily made rotation-invariant, at the cost of some distinctiveness, by measuring $\{\theta_i\}_{i=2..k}$ relative to θ_1, rotating $\{\mathbf{r}_i\}_{i=2..k}$ by θ_1, and removing θ_1 from the descriptor. In addition, for $k \geq 3$ one can design descriptors with even higher degrees of invariance (affine, projective) to be used, e.g. for wide-baseline stereo [30], although we do not investigate this possibility further in this paper.

3.4 Comparing kAS

We define here a measure $D(a, b)$ of the dissimilarity between two kAS P^a, P^b

$$D(a,b) = w_r \sum_{i=2}^{k} \|\mathbf{r}_i^a - \mathbf{r}_i^b\| + w_\theta \sum_{i=1}^{k} D_\theta\left(\theta_i^a, \theta_i^b\right) + \sum_{i=1}^{k} \left|\log\left(l_i^a / l_i^b\right)\right| \tag{2}$$

where the first term is the difference in the relative locations of the segments, $D_\theta \in [0, 1]$ measures the difference between segment orientations, normalized by π, and the last two terms account for the difference in lengths. As segment lengths are often inaccurate, we give higher weight to the two other terms of the comparison measure: in all our experiments $w_r = 4$, $w_\theta = 2$. All $\mathbf{r}_i$ and all lengths are normalized as in equation (1).

4 Constructing the kAS codebook

In the previous section we have introduced the kAS features. Before using them for object class detection (next section), we construct a codebook (or 'visual vocabulary' [3]) of feature types by clustering a set of training kAS according to their descriptors (a different codebook is generated for each k). In addition to revealing the frequency at which feature types occur, the codebook is convenient because it relieves the need for explicitly comparing every test image features to every feature from the the training images. Instead, comparison to much fewer feature types suffice. Codebook representations have become popular through several recent works [3, 5, 14, 15, 16],

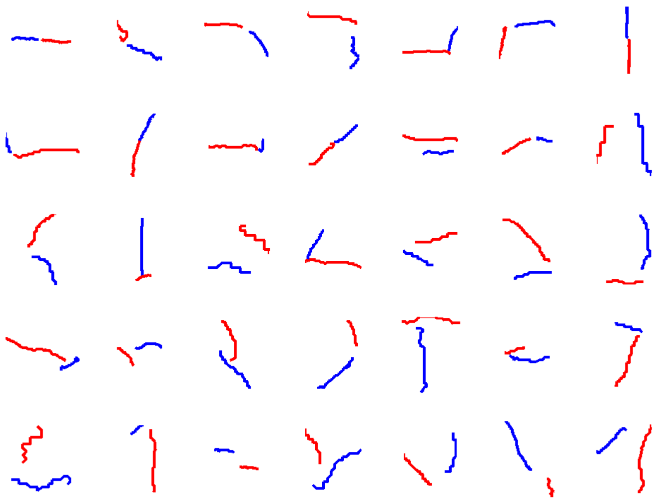

Figure 2: *The 35 most frequent 2AS types from the codebook we use in all experiments, constructed from 10 outdoor images (5 positive and 5 negative images from the INRIA horses dataset, section 6)*

Let $\mathcal{G}$ be a complete graph whose nodes are the training kAS, and arcs are weighed by $d - D(a, b)$. We partition $\mathcal{G}$ into cliques so as to maximize the sum of intra-clique weights, using the clique-partitioning (CP) approximation algorithm of [8]. Each resulting clique is a cluster of similar kAS.

The choice of CP instead of K-means, commonly used for building visual codebooks, is appropriate in our context where the dissimilarity measure D makes the descriptor space circular (D_θ terms). Moreover, the parameter d is easy to set, because it represents a rough indication of the acceptable intra-cluster dissimilarity (akin to the kernel-width in mean-shift clustering [14]). K-means instead requires the number of clusters as input, which is unknown apriori and varies from dataset to dataset. Several experiments indicate that the exact choice of d has little impact on the overall system performance (section 6).

For each cluster, we select as a representative the kAS with the lowest sum of dissimilarities to all others (i.e. the one closest to the cluster center). The final codebook $\mathcal{C}$ is the collection of these representative kAS, the *kAS types*.

When constructing codebooks from different image sets, we observed that the kAS types occurring with a significant frequency were very similar. This confirms the intuition that kAS are generic features (certainly for the low values of k we consider). Hence, for each k we build a single codebook from 10 images and use it for all object classes in our experiments (section 6).

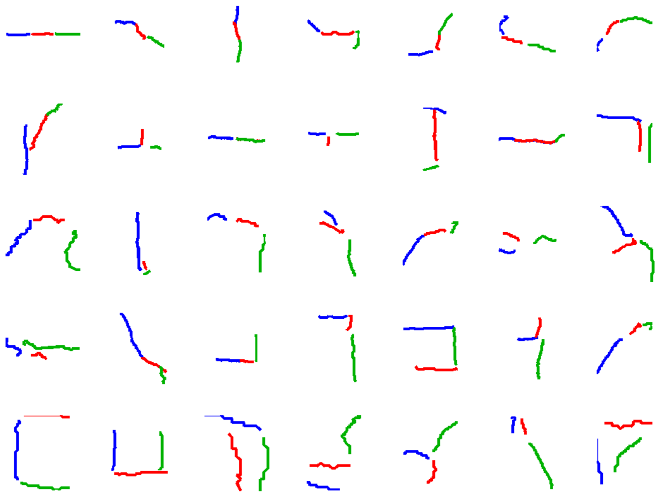

Figure 3: *The 35 most frequent 3AS from our codebook.*

Figure 2 show the 35 most frequent types in the 2AS codebook. As we can see, they have quite natural shapes: two collinear segments, L structures, and small T-junctions. Figure 3 displays the 35 most frequent 3AS types. They form more complex structures than 2AS: C, Y, F, Z shapes, larger T shapes, and triangles.

5 Object class detection

In this section we present a scheme for detecting objects based on kAS. We first train a classifier to distinguish windows covering objects of a certain class from any other window, and then apply it for localizing novel instances in previously unseen test images, based on a sliding window mechanism. As in the large majority of modern works, we build a detector for a single viewpoint.

5.1 Training

The training data includes positive images, containing instances of the class annotated by a bounding-box (figure 4), and negative images.

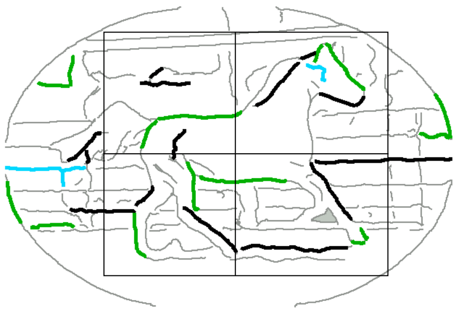

Figure 4: *a positive training example, with bounding-box, tiling, and a few kAS superimposed ($k = 2$, $T = 4$).*

Window descriptor. To produce a useful classifier, we need a numerical window descriptor which is accurate enough to separate positive examples from negative, yet flexible enough to accommodate for class variability. When these goals are met, test windows on novel object instances will have descriptors closer to the positive training set than to the negative, and the classifier can succeed.

A straightforward option would be the kAS histogram, counting how many kAS of each type there are inside the window, which is a simple bag of features representation. However, we can obtain better discriminative power by also encoding the spatial layout of the kAS in the window descriptor. We subdivide each window into a set of *tiles* $\mathcal{B}$, and compute a separate kAS histogram for each tile (figure 4). The concatenation of all histograms yields the $|\mathcal{B}| \cdot |\mathcal{C}|$ - dimensional window descriptor.

The tiling pattern $\mathcal{B}$ automatically adapts to the training data as follows. First, the system computes the mean dimensions of the positive training windows (width M_w and height M_h). Next, it allocates a total of T tiles, choosing the number of tiles along each dimension so as to make them as square as possible: round($\sqrt{TM_w/M_h}$) along the width, and round($\sqrt{TM_h/M_w}$) along the height. The parameter $T = |\mathcal{B}|$ controls the resolution of the tiling. M_w, M_h will later be used again when searching for objects in new test images, to set the aspect-ratio of the sliding window to the one best fitting the training examples.

When computing the kAS histograms, rather than assigning each kAS to the single closest type, it is soft-assigned to all types within dissimilarity d. More precisely, each kAS P distributes a total sum P_s among the types it is assigned to, in inverse proportions to the dissimilarity to the types' representative kAS. This makes the representation of a window less sensitive to the exact shape of the kAS it contains, and to the exact codebook types. This leads to smoother models, which better generalize to novel object instances, and to a more accurate, stable evaluation of test image windows (next subsection). In addition to a kAS' shape, we also consider its *relevance*: the total contribution of kAS P to a histogram is the average strength of its edgels $P_s \in [0, 1]$. We experimentally observed a considerable improvement over treating edgels as binary features (as also noticed by [4, 9]).

Our window descriptor is a valuable choice for object class detection. It is distinctive, because it records *which* local shape structures (kAS) it contains, and roughly *where* they appear. At the same time, it is flexible thanks to the coarse tiling, and the continuous assignment of kAS to types. Much of the power of our representation comes from organizing the image edges over two levels of spatial arrangements: contour segments within the kAS, and then the kAS within the overall object.

As the tiling resolution T increases, the spatial localization of kAS grows stronger, resulting in a more informative descriptor, but also a more rigid one, accommodating for less spatial variability of the class. Hence, there should be some optimal T, bringing the best trade-off between accuracy of localization information and tolerance to intra-class variation. Interestingly, our experiments show the optimal T to decrease with increasing k (section 6). With $k = 1$ the features are so uninformative that the window descriptor needs to be augmented with fine-grained localization to be distinctive. While k grows, kAS become more complex, and the added value of localization gradually diminishes. In addition, we found the optimal T for interest points described by SIFT to be lower than that of any kAS we explored ($k \leq 4$). Since SIFT descriptors of an image patch are richer features (and have a descriptor of much higher dimensionality), this further confirms the above subtle relation between feature complexity and localization resolution.

SVM classifier training. The window descriptor is computed for each positive training example, and for a number of negative examples collected by sampling windows of size $M_w \times M_h$ over each negative training image. In our experiments, windows are sampled every 50 pixels horizontally and vertically, typically resulting in thousands of negative windows.

All window descriptors are finally used to train a two-class linear SVM. Since negative windows are much more numerous, the positive window descriptors are replicated to correct the imbalance.

Figures 5 and 6 show a few kAS automatically selected by the SVM for a few classes (i.e. the 'kAS type + tile' combinations corresponding to the highest weighed window descriptor dimensions). Among the large number of KAS composing each example, several lie on the object boundary, and are picked up by the SVM as local shape structures common to multiple training examples.

Using multiple kAS degrees at once. Our framework includes the possibility to use multiple degrees of kAS at the same time (e.g. $2AS$ and $3AS$). In this case the different sets of kAS are treated separately: there is a codebook and tiling resolution for each value of k. Window descriptors obtained for different k are then concatenated to give a large descriptor which is fed to the SVM.

Using kAS of different degrees at the same time is an interesting option. Some characteristic object elements might be extremely simple (like the straight line on top of a comb, for which $k = 1$ is good), while others might be more sophisticated local structures (like a C-

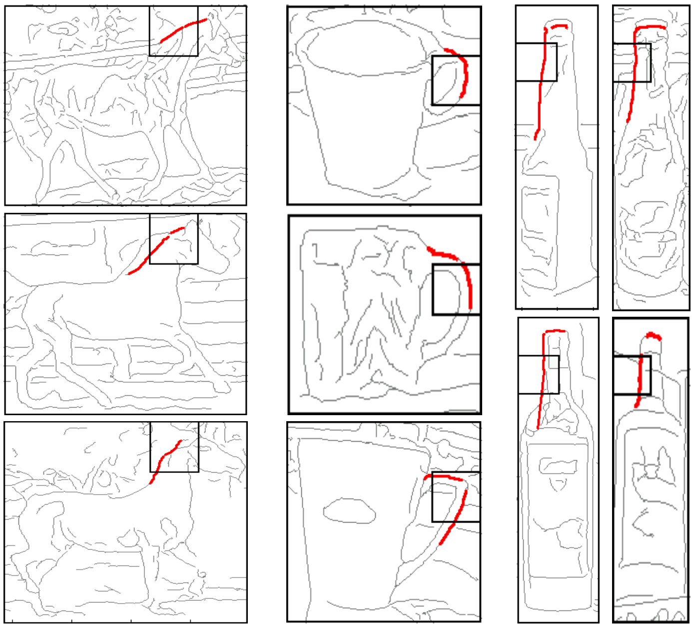

Figure 5: *one of the top dimensions selected by the SVM for three classes ($k = 2$)*

shaped mug handle, for which $k = 3$ is good). Hence, using multiple degrees simultaneously offers the SVM a larger, more diverse pool of parts to choose from.

5.2 Testing

Having trained a linear SVM window classifier, we can detect and localize novel object instances in a test image using a simple sliding-window mechanism [31, 4]. We slide a window of aspect-ratio M_w/M_h over the image at multiple scales [2], compute the window descriptor at each location/scale and evaluate it with the SVM. This provides a 3D response map, whose local maxima give candidate object detections. The final set of detections are obtained after a last polishing: if two candidate detections overlap considerably, we filter out the weaker one.

This sliding window technique requires computing the histogram of kAS types within a large number of image windows (tiles). We achieve this efficiently by using an Integral Histogram [24] representation (IH). After building an IH where each dimension correspond

[2] This is implemented simply by resizing the window to contain a varying portion of the image. It is not necessary to rescale the image, because the kAS features themselves automatically adapt to image structures of different scales. In all our experiments the sliding step is 10 pixels is each direction, while the scale step is $2^{\frac{1}{4}}$. We consider all scale levels where the window's longer side is more than 50 pixels and still fits in the image.

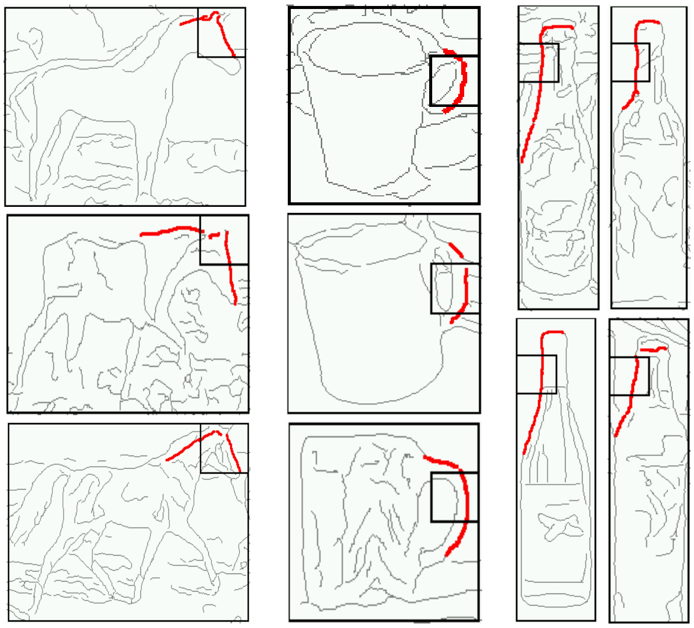

Figure 6: *one of the top dimensions selected by the SVM for three classes ($k = 3$)*

to a kAS type, it is possible to compute the histogram of kAS types in *any* window in $3|\mathcal{C}|$ operations, independently of the total number of kAS in the image and of the number of kAS in the window. The cost of building the kAS IH is low, and it is done only once for an image (as we adpot a single codebook for each k). The main share of the cost is the computation of the soft-assignments the image kAS to types, which must be done anyway.

Our object detection procedure is very fast. After preprocessing (from edge detection to the kAS IH), it takes about 1 second to detect all instances in our C++ implementation on a standard workstation. Preprocessing takes longer, due the accurate, but slow, Berkeley edge detector (a few minutes). However, it only needs to be done once, so the cost is amortized when searching for several classes, or when using kAS of multiple degrees at the same time.

6 Experimental evaluations

6.1 Datasets and protocol

We present extensive experimental evaluations, involving several existing datasets, over 8 diverse shape-based object classes, for a total of more than 1400 test images. Here we briefly introduce these datasets, while the following sections report the experiments.

INRIA horses [13] This challenging dataset consists of 170 images containing one or more horses, seen from the side, and 170 images without horses. Horses appear at several scales, and against cluttered backgrounds. We employ the first 50 positive and 50 negative images for training, and the remaining $120 + 120$ images for testing.

This dataset plays a special role in our evaluations, as we optimize the two free parameters of our detection system on it (the window tiling resolution T and the clustering threshold d). The optimal setting established on this dataset is then used on all others. No tuning is applied to any other dataset, so the exact same system is run on all datasets.

Weizmann-Shotton horses [29] Shotton et al. [29] propose another horse detection dataset, composed of 327 positive images containing exactly one horse each, and 327 negative images. The positive images are derived from a dataset previously released by the Weizmann institute for evaluating image segmentation algorithms. In order to carry out proper comparisons, we follow the protocol of [29] strictly by using their scale-normalized images, and running our system at a single scale [3]. As in [29], the first 50 positive and 50 negative images are used for training, the other $277 + 277$ for testing (figure 8).

ETHZ shape classes [9] This dataset features five diverse classes (bottles, swans, mugs, giraffes, apple logos), over a total 255 images collected from the web by Ferrari et al. [9]. It is the most challenging one we report on, as the objects appear in a wide range of scales, there are considerable intra-class shape variations, and many images are severely cluttered, with the objects comprising only a fraction of it (figure 10).

We train one detector per class, using the first half of the available positive images (there are 40 for apple-logos, 48 for bottles, 87 for giraffes, 48 for mugs, and 32 for swans). As negative training images, an equal number is taken, with each of the other 4 classes contributing 1/4 of them. For example, the training images for the bottle detector are 24 bottle images, plus 6 images from each of the other classes, totaling 24 negative training images. All other images are used for testing, so each class is searched for in images from *every* class.

Caltech 101 [7] The last source of data we consider are three shape-based classes from the well-known Caltech-101 database [7]: anchors, chairs, and cups $(42, 62, 57$ positive images respectively). Although most images contain only limited clutter, the dataset offers substantial intra-class variation (figure 10). As for the ETHZ Shape Classes, we evaluate one class at the time. We use the first half of the positive images for training, as well as an equal number of negative images from the Caltech-101 background set. The test set consists of the remaining positive images, plus the same number of negative ones.

Evaluation criterion Performance is evaluated by plotting detection-rate (DR) versus the incidence of false-positives (false-positives per image, FPPI) while varying the detection

[3] sliding a window of fixed dimensions $M_w \times M_h$. In all other experiments the system is run at multiple scales as detailed in section 5.2.

threshold. We prefer these DR/FPPI plots over precision/recall ones for several reasons. FPPI has a clearer interpretation than precision, which is entangled with detection-rate. Moreover, FPPI is independent of the number of negative test images, and DR/FPPI plots are easier to read, because they increase monotonously.

Comparisons between different methods is mainly based on two points on the DR/FPPI plot, at 0.3 and 0.4 FPPI. These are especially relevant because they correspond to a rather low, but not extremely low, FP rate (around 1 FP every 3 images). Only on the Shotton horses dataset we report precision/recall plots, and compare methods based on equal-error rates, because [29] published their results in that form. Hence, average detection-rates at a particular FPPI rate refer to means computed over 9 datasets, excluding Shotton horses.

For all datasets and methods, a detection is counted as correct if its bounding-box overlaps more than 20% with the ground-truth bounding-box, *and* vice-versa. Any other detection is counted as a false-positive.

6.2 Degree of complexity of kAS

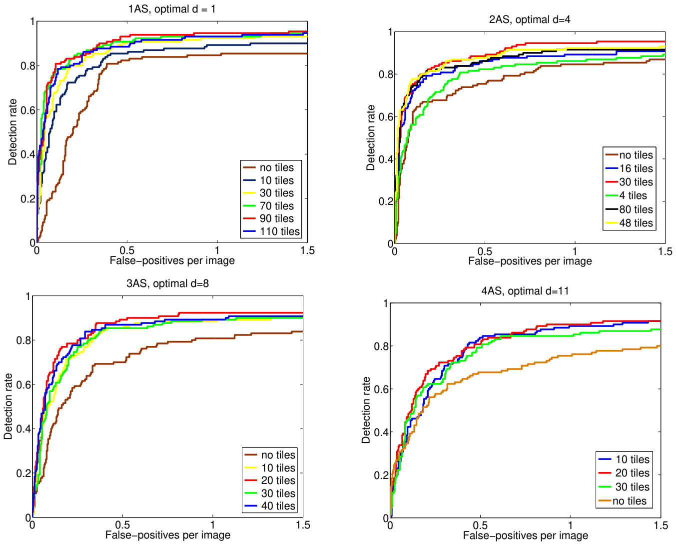

Figure 7: *Impact of window tiling resolution T at the optimal clustering threshold d. The optimal T is $T = 90$ for 1AS, $T = 30$ for 2AS, and $T = 20$ for 3AS and 4AS.*

Impact of tiling and clustering threshold Before comparing the performance of kAS of different degrees of complexity on all 10 datasets, we first optimize T, d for each k separately on the INRIA horses dataset ($k \in \{1, 2, 3, 4\}$). For several pairs of T, d, we reprocess the dataset and obtain a DR/FPPI curve. Figure 7 shows the impact of the tiling resolution,

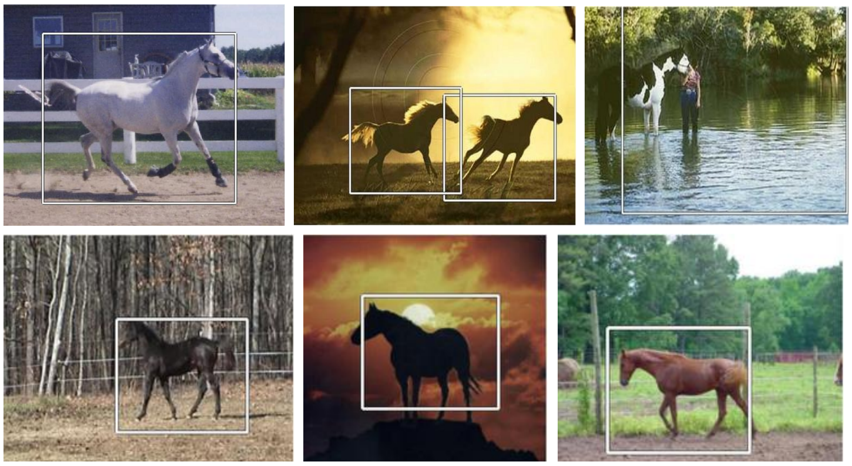

Figure 8: *Top row: detections at 0.4 FPPI for the INRIA horses dataset. Rightmost image shows a missed detection and a false-positive. Bottom: detections at the equal-error rate for the Shotton horses dataset.*

while keeping d fixed at the optimum. From the plots it clearly appears that subdividing the window into tiles makes a substantial difference for all k. Compared to a single bag-of-feature representation (no tiles), at 0.3 FPPI the optimal tiling brings improvements ranging from 20% ($k = 1$) to 13% ($k = 4$) detection-rate.

It is intriguing to observe that the optimal value of T decreases with increasing k. This confirms experimentally the subtle relation discussed in section 5.1: as the features grow more complex and hence informative, a coarser spatial localization is sufficient, while at the same time a lower T yields better tolerance to intra-class variations. Individual segments benefit most from a very fine subdivision of 90 tiles, whereas the saturation point for localization information is already reached at 20 tiles for $3AS$. Moreover, also the gain brought by tiling reduces as the features become more complex, because the added value of localization gradually diminishes (at 0.3 FPPI, it is of 20%,16%,16%,13% detection-rate for k=1,2,3,4 respectively).

Varying the clustering threshold d has a smaller impact. Nevertheless, we observe the number of clusters corresponding to the optimal d to increase with k $(4, 127, 255, 397$ for $k = 1, 2, 3, 4$ respectively). This makes sense, because as the features becomes more complex, they can assume a wider variety of shapes. In particular, just 4 clusters are necessary for $k = 1$, corresponding roughly to four orientations separated by 90 degrees.

Following these observations, all further experiments are performed with the optimal parameters T, d for every k.

Degree k We applied our object detection scheme to all 10 datasets, for the four degrees of kAS complexity we explore ($k \in \{1, 2, 3, 4\}$). Although there is no single degree producing

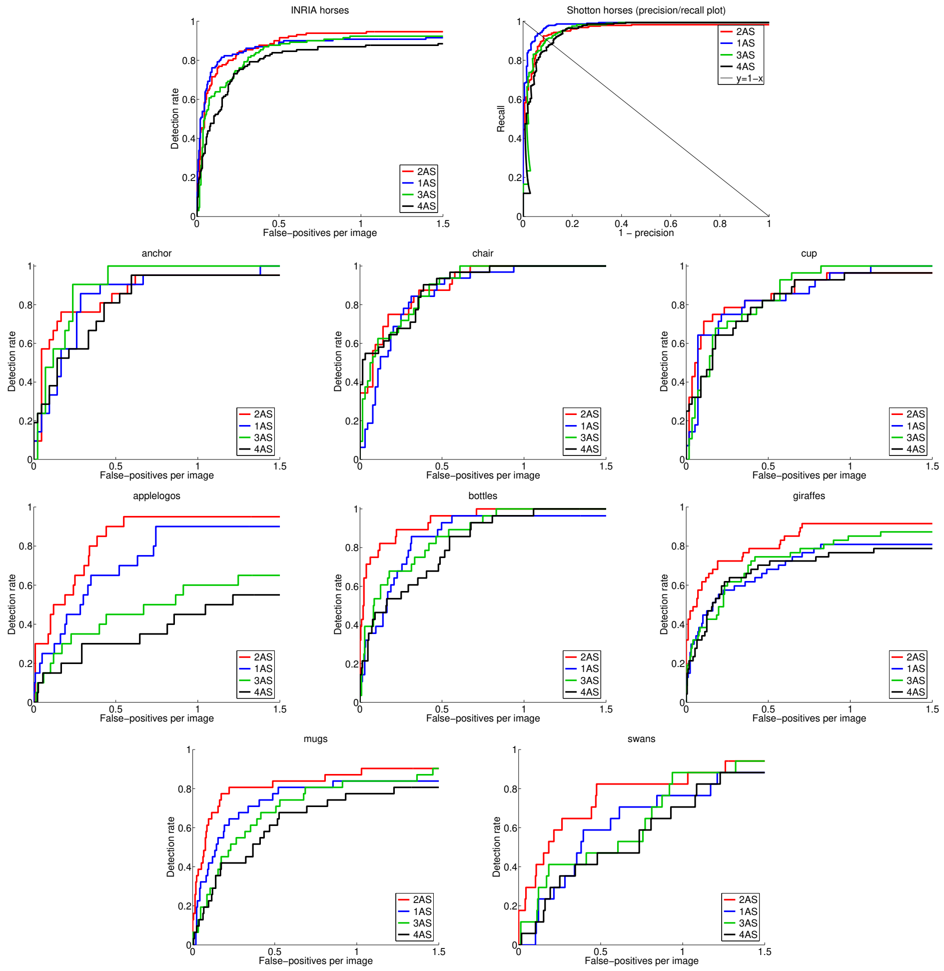

Figure 9: *Performance of kAS, for varying degrees of complexity k. Top row: the two horses datasets we consider: INRIA horses, and Shotton horses. Second row: the three classes from Caltech 101. Third row: the five ETHZ Shape Classes. All performance comparison figures in the paper follow this layout.*

the best results on all datasets, $2AS$ perform best overall (figure 9). The $2AS$ plot is above all others on 5 datasets, and its average detection-rate at 0.3 FPPI is 76.7%, versus 69.4%,64.1%,56.5% of $1AS$,$3AS$,$4AS$ respectively (table 3). Hence, we concluder that 2AS are the kAS with the best intermediate complexity, offering the optimal compromise between being informative, repeatable, and generating a good ratio of pure boundary features versus mixed/clutter features (as discussed in subsection 3.2). In the remainder of the paper, $2AS$

is the reference kAS for comparison to other methods, and will be referred to as PAS (*pairs* of adjacent segments).

As inspecting table 3 reveals, ranking kAS according to average detection-rate at 0.3, or 0.4 FPPI, the following order appears: $2AS > 1AS > 3AS > 4AS$. This ranking is well confirmed by the overall relative heights of the DR/FPPI plots (clearest on mugs, applelogos). Surprisingly, the second best kAS are individual segments. Much of the reason is in the great impact of tiling on $1AS$, where very fine-grained localization compensates for the feature's lack of distinctiveness (figure 7). Nevertheless, PAS do better, confirming it's advantageous to consider groups of connected segments as features for object detection. Moreover, PAS, as well as kAS of higher degrees, are more reusable in other systems, where the discriminative power of individual features is more important, or where a feature correspondence must generate a higher order transformation than just translation (e.g. recognition systems using feature transformations [16, 10], or for matching features between two images [21, 30]).

In absolute performance terms, PAS work consistently well on all classes (detection-rates between 79% and 88% at 0.4 FPPI), with the exception of swans. This is especially remarkable when considering the low number of positive training images used in many datasets (e.g. 24 for bottles, see table 1). PAS achieve particularly high performance on Shotton-horses, with 91.7% precision-recall equal error-rate, in line with the state-of-the-art approach [29] (92.1%), while $1AS$ do even better, with 93.5%. Moreover, in contrast to their work, our method doesn't need any segmented training image (only bounding-boxes), and can detect objects at multiple scales. The striking performance of $1AS$ on this dataset (the only one where they beat PAS) might be explained by the very low resolution of the images (horses are about 100 pixel wide), which favors simpler features.

We can also draw a loose comparison to [13], on the INRIA horses datasets. Numerically, PAS' performance is close to their work (e.g. PAS do 70.0% at 0.066 FPPI, which corresponds to 86.1% precision, while [13] reports 70.3% recall at 87.7% precision). However, an exact comparison is not possible, as the authors of [13] have lost details of the particular test set on which results were reported. We adopt here the official release of the dataset, which should come quite close. As a reference, we also mention that [9] obtains a similar level of performance as PAS on the ETHZ Shape Classes, although the two methods are not directly comparable since [9] inputs hand-drawings as models.

In order to further strengthen our understanding of PAS' performance, and properly set it in the context of alternative methods, in the following we perform in-depth comparisons to interest points, used within our object detection framework, and to the system of Dalal and Triggs [4].

6.3 Comparison to interest points

Interest point (IP) detectors respond to local pixel patterns with certain special properties (e.g. cornerness) and produce local features widely used for object class detection [10, 16, 5]. IP descriptors capture the appearance of the image patches surrounding them. In order to support the claim that kAS are better suited to represent shape-based classes, we replace them by IPs in our object detection framework, and reprocess all 10 datasets. We experiment

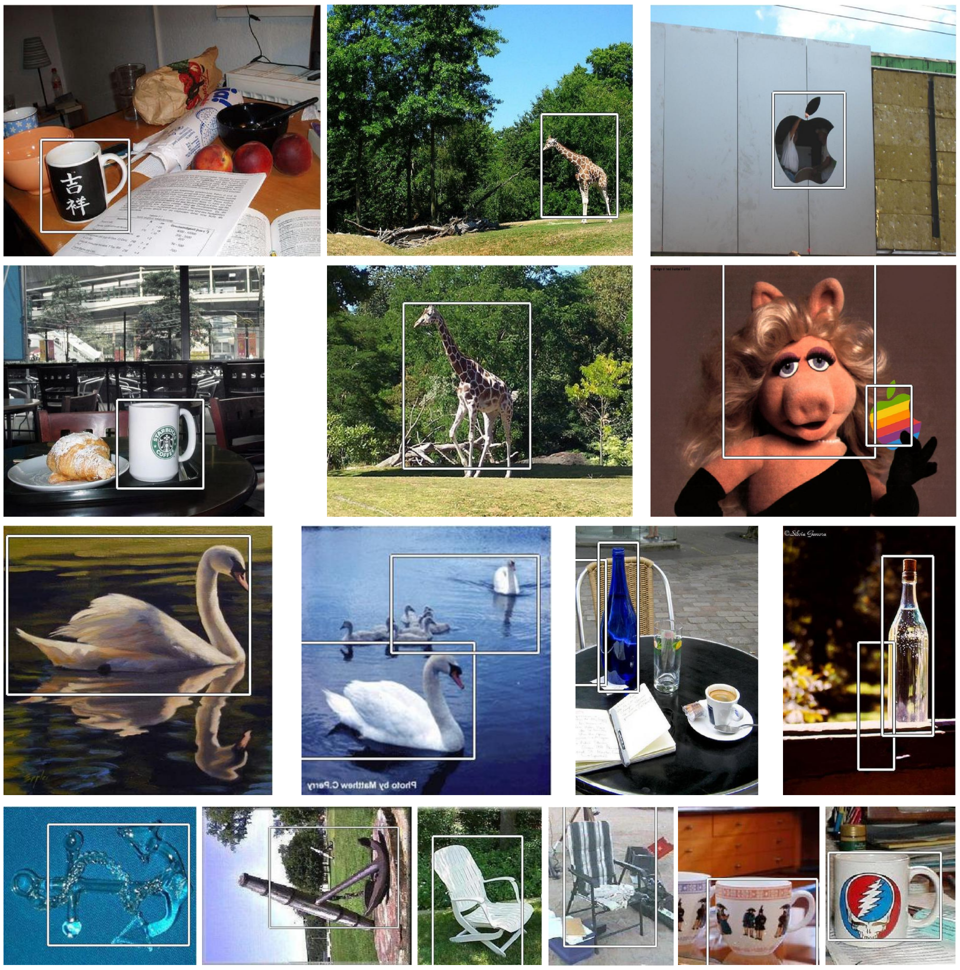

Figure 10: *All detections at 0.4 FPPI on some example images. Top 3 rows: ETHZ shape classes. Bottom row: Caltech 101.*

with three of the most widespread scale-invariant IPs: Harris-Laplace [21], LoG [19], and DoG [19]. All IPs are described by the extremely popular 128-dimensional SIFT [19].

Codebooks, tiling, and number of clusters The number of IP per image is about 1000 to 2000, larger than that of kAS (for $k \leq 4$). In addition, we want to experiment with IP codebooks built from more than the 10 images used for kAS (details below). As a result, the total number of IPs to be clustered can grow beyond what CP can handle. Since CP builds a pairwise dissimilarity matrix, memory consumption limits the number of features

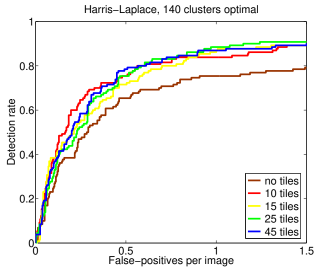

Figure 11: *Optimizing T, keeping the number of clusters fixed at* 140 *(which is the optimum in the tested range* $100 - 300$*).*

to about 15000, while in several cases there are more than 50000 IPs. Hence, we build IP codebooks using k-means. Notice how the CP parameter d is now replaced by explicitly providing the number of clusters.

As done before for kAS, we optimize the number of tiles and of codebook clusters on INRIA horses (figure 11). The selected optimal T is 10, which confirms the trend observed on kAS: the richer the feature, the lower the value. Figure 11 shows the evaluation on Harris-Laplace, but similar optimal values are obtained for DoG or LoG.

The rationale behind using a single kAS codebook from a small set of 10 images is that the features are simple and generic enough. However, this might not hold for IPs. Since they are based on texture, and the SIFT descriptor captures an entire image patch, quite different codebooks might result from different image sets (e.g. giraffes versus horses). Therefore, we experimented with three kinds of codebooks, on the five ETHZ Shape Classes. The first is computed from the same 10 images used for kAS, the second is specific to a single class (computed from the same images used to train the SVM), and the last is based on images from all five classes (computed from all images used to train all 5 SVMs). From the results we obtained, it indeed appears that class-specific IP codebooks perform moderately better on average. Hence, all experiments below are performed with class-specific codebooks.

Performance The plots in figure 12 and the average detection-rates in table 3 clearly show that PAS substantially outperform all tested IPs. Only on two datasets IPs achieve a moderately better performance than PAS (Harris-Laplace on swans, and DoG on cup). Besides, we notice IP's uneven performance across different classes (compare DoG on cup and bottles). PAS' performance instead, is quite stable. Finally, it's worth noting that on giraffes, for which both shape *and* texture are characteristic, the results of PAS and the best IP are very close (especially in the range 0.3-0.4 FPPI).

Beyond PAS, one can compare kAS in general to IPs. In terms of average detection-rate at $0.3 - 0.4$ FPPI, all explored kAS do considerably better than any of the tested IPs (table 3). Only the performance levels of $4AS$ and Harris-Laplace are similar.

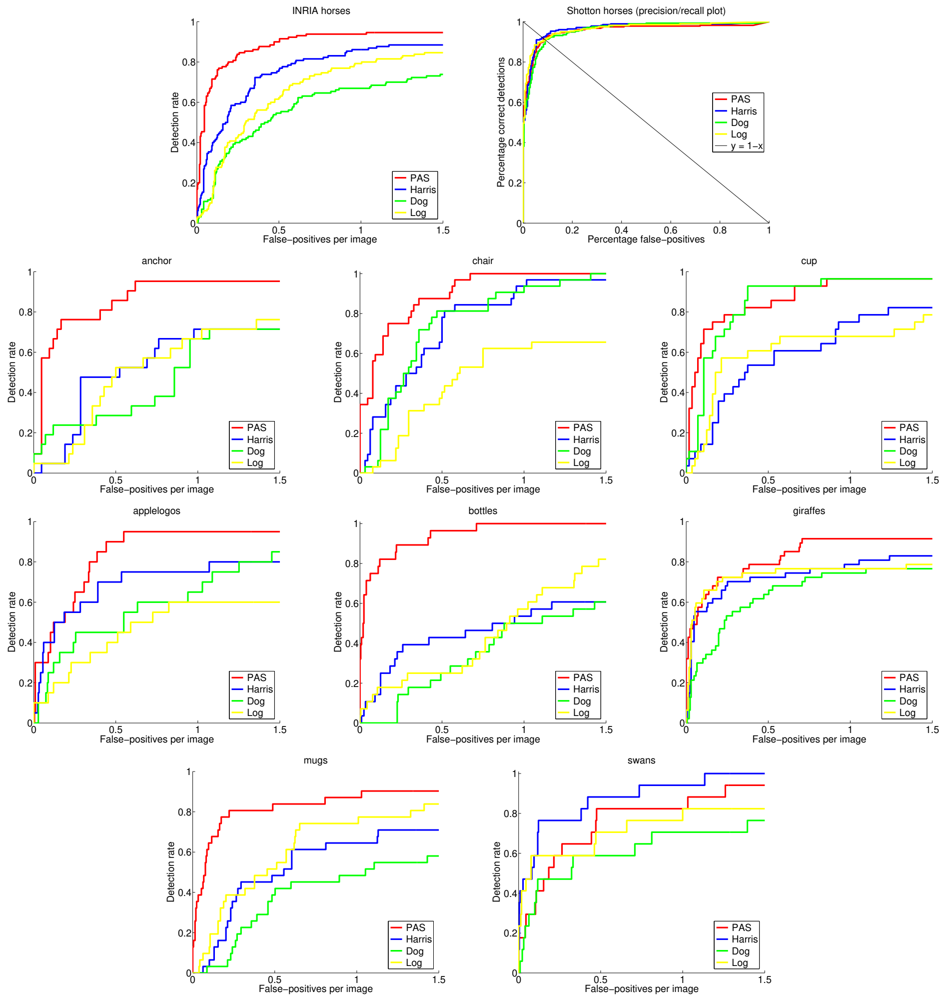

Figure 12: *Performance of IPs, compared against PAS.*

Inspecting the data, it is also possible to rank features among IPs. Although no single one works best on all datasets, on average Harris-Laplace stands out, followed by DoG and LoG, which are at about the same level (table 3).

In conclusion, these experiments confirm that kAS are more appropriate features than IPs for shape-based classes.

6.4 Combining multiple kAS degrees

Even though PAS are better than kAS of other degrees on most datasets, 1AS and $3AS$ win on Shotton horses and anchors respectively. This suggests that using all three 1AS/2AS/3AS simultaneously might give an even better detector.

As explained in section 5, we integrate kAS of multiple degrees by concatenating window descriptors computed separately for each k, and then training a single SVM from them. Each degree uses its own codebook and optimal tiling resolution. In this fashion, the SVM can choose from a very large pool of different local shape structures and tile combinations.

The results can be seen in table 2. For most datasets, performance is very similar to PAS. On Shotton horses instead, $\{123\}AS$ achieves an excellent 94.2% precision-recall equal-error rate, which is significantly better than any of $1AS$, $2AS$, or $3AS$. On swans however, we register a considerable performance drop wrt to PAS (from 64.7% down to 47.1% at 0.3 FPPI). Moreover, the high performance of $3AS$ on anchors (90.5% at 0.3 FPPI) is not reproduced by $\{123\}AS$ (76.2% at 0.3 FPPI).

Although it seems surprising that adding features can lower performance, this could be due to overfit. Indeed, the dimensionality of the combined window descriptor is much higher than that of a single kAS degree, while the number of training examples remains the same. To corroborate this, notice how the performance drops occur on the datasets with fewest training examples (swans, 2x16 training examples, and anchor, 2x21 examples), while the largest improvement happens on the dataset with most examples (Shotton horses, 2x50 examples, see table 1). Hence, although in our experiment $\{123\}AS$ performs on average slightly below PAS (table 3), they remain a promising option for datasets with many training images.

Finally, we have tried to reduce the dimensionality of the window descriptor with a simple feature selection scheme. We sort the SVM hyperplane coefficient and keep the first N, so that they sum up to a certain percentage of the total mass. Remarkably, when keeping as few as 50% of the total mass, detection-rate only decreases of 2.5% at 0.3 FPPI on INRIA horses. This corresponds to retaining only 18% dimensions, therefore computing a fraction of all 'kAS type + tile' combinations, yielding a speedup of factor 5.

6.5 Combining PAS and Harris-Laplace

Following the same approach as in the previous subsection, we have combined PAS and Harris-Laplace, as they are the best members of their respective feature families. This seems an exciting possibility, because PAS and Harris-Laplace exploit *complementary* image properties (contour and texture). Hence, the hope is to obtain a more generic object detector, which might autonomously determine which kind of feature is more appropriate for (part of) a given object.

The PAS+Harris detector does better than either component alone on giraffes and Shotton horses (table 2). This is particularly meaningful since giraffes are defined by both shape and texture, so we expect their combination to reinforce the detector. Moreover, giraffes and Shotton horses are the only two classes on which PAS and Harris-Laplace work about equally well. PAS+Harris now achieves an impressive 95.7% precision-recall equal-error-rate on Shotton horses, considerably above the 92.1% of [29]. Furthermore, on INRIA horses, chair, and swans, PAS+Harris exhibit the desired behavior: its performance aligns with the better of either PAS or Harris-Laplace. However, on the remaining 5 classes PAS+Harris only performs somewhere inbetween PAS and Harris-Laplace. Again, a probable reason is overfit, and we observe a clear correlation between the performance improvement/loss of PAS+Harris and the number of training images in a class (table 1). To confirm this further, we run tests with several randomized splits of the images in training and test subsets, and observed that the performance variations of PAS+Harris are far greater than those of either feature alone.

6.6 Comparison to Dalal and Triggs [4]

We conclude our series of evaluations by comparing against the object detection technique by Dalal and Triggs [4], which is currently the state-of-the-art in human detection, and has proven very competitive on other classes as well [33]. Like ours, their object detector is based on sliding a window subdivided into tiles, but uses histograms of gradient orientations as descriptors.

In an effort to perform a fair comparison, we discussed with the authors of [4], who recommended the following operations. First of all, we used the official software released by the authors. Moreover, we rescaled all training windows to make the longest side 100 pixels, which is about the resolution their system is tuned on [4]. Finally, following the protocol applied for PAS, we optimized the two most important parameters on INRIA horses. These are the preprocessing applied before computing gradients, and the block normalization scheme applied after collecting HoG descriptors. The difference between the best and the worst combination of preprocessing and normalization turned out to be moderate: 4.5% detection-rate at 0.3 FPPI. Nevertheless, we processed all datasets with the best combination: converting to Lab color space as preprocessing, and normalizing descriptors by the square root of the L1 norm (see [4] for details).

The results are displayed in figure 13 (the system of [4] is marked as HoG). Our detector achieves a substantially higher performance on 6 of the 10 datasets, while on mugs and applelogos the two methods are about equally good, and HoG obtains better results on applelogos and swans. In terms of average detection-rate at 0.3 FPPI, PAS leads with a considerable margin of 20% (table 3).

The HoG curves abruptly stop growing after a rather low FPPI rate, due to the system returning no detection on several images. We tried to counter this by altering a parameter controlling the minimal score for windows to enter the non-maxima suppression stage, but it only resulted in lower curves (as also expected by Dr. Dalal). Besides, we point out that explicitly comparing performance at the point where HoG stops growing only makes a difference for cup. PAS still leads on 6 datasets, and draws on mugs. Finally, even if fully growing curves could be produced, the trend in DR/FPPI plots is for the slope to decrease with increasing FPPI. Hence, even in an optimistic extrapolation, the HoG curves would still remain below PAS' ones for 6 datasets.

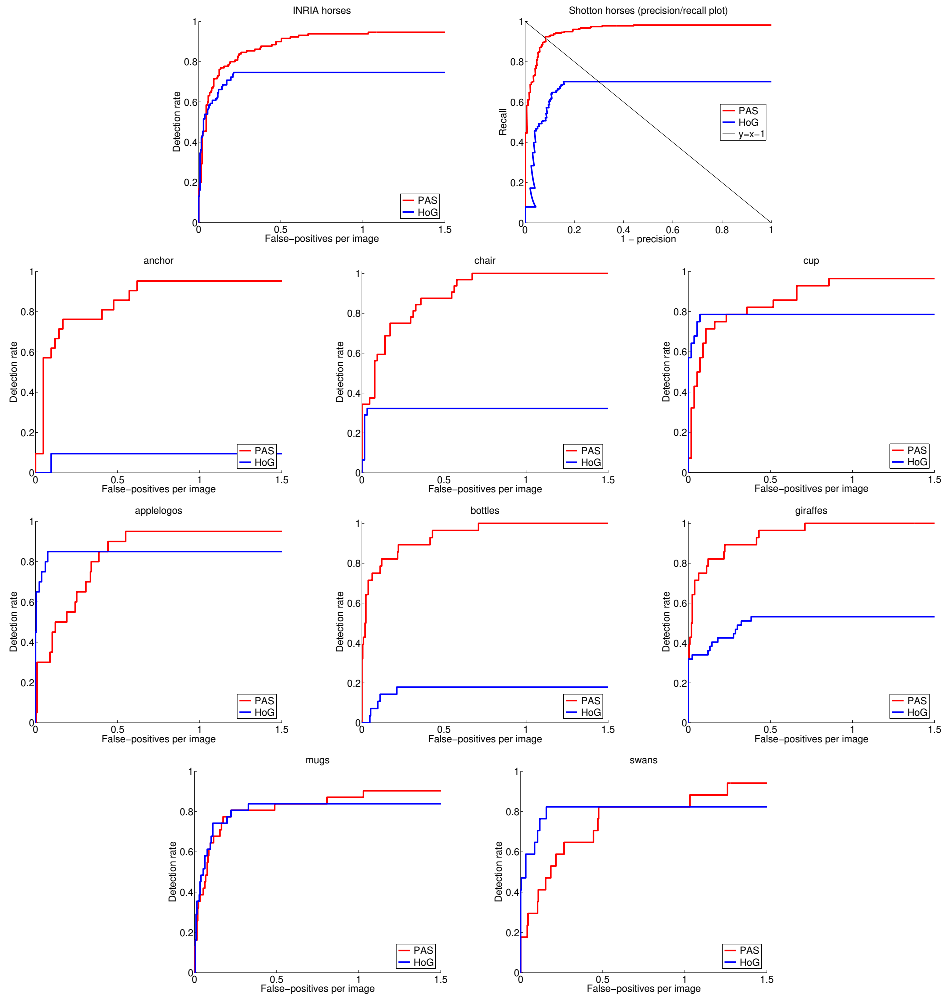

Figure 13: *Comparison between our PAS-based detector and Dalal and Triggs [4] HoG-based ones.*

In conclusion, our detection system compares favorably to [4] in our experiments, which further consolidates PAS as excellent features for object detection.

7 Conclusions

We have introduced the kAS family of local contour features and their application for object detection. kAS are designed to be capable of covering pure portions of an object boundary,

Table 1: *Number of positive+negative training and testing images for all datasets.*

	INRIA	Anchor	Chair	Cup	Applelogo	Bottle	Giraffe	Mug	Swan	Shotton
Train	50+50	21+21	31+31	29+29	20+20	24+24	44+44	24+24	16+16	50+50
Test	120+120	21+21	31+31	28+28	20+215	24+207	43+167	24+207	16+223	277+277

Table 2: *Detection rates at 0.3 and 0.4 FPPI for all features we tested, as well as for the object detection system of [4] (marked as HoG). The 'Shotton' column reports recall at equal error-rate for the Shotton horses dataset. It is included here for homogeneity of presentation.*

Detection rate	INRIA	Anchor	Chair	Cup	Applelogo	Bottle	Giraffe	Mug	Swan	Shotton
PAS (0.3 FPPI)	85.4	76.2	78.1	78.6	65.0	89.3	72.3	80.6	64.7	91.7
1AS (0.3 FPPI)	86.2	85.7	81.2	75.0	55.0	78.6	59.6	67.7	35.3	93.5
3AS (0.3 FPPI)	79.2	90.5	75.0	71.4	35.0	67.9	61.7	54.8	41.2	90.3
4AS (0.3 FPPI)	76.9	57.1	71	71.4	30.0	60.7	63.8	41.9	35.3	90.3
Harris (0.3 FPPI)	63.1	47.6	50.0	42.9	60.0	39.3	70.2	45.2	76.5	91.0
Dog (0.3 FPPI)	42.3	23.8	53.1	78.6	45.0	17.9	55.3	22.6	47.1	89.2
Log (0.3 FPPI)	50.8	23.8	31.2	57.1	30.0	25.0	72.3	38.7	58.8	90.3
{1,2,3}AS (0.3 FPPI)	86.2	76.2	77.4	82.1	70.0	85.7	68.1	80.6	47.1	94.2
PAS + Harris (0.3 FPPI)	84.6	61.9	83.9	67.9	60.0	57.1	80.9	51.6	82.4	95.7
HoG (0.3 FPPI)	74.6	9.5	32.3	78.6	85.0	17.9	48.9	80.6	82.4	70.1
PAS (0.4 FPPI)	87.7	81.0	87.5	82.1	85.0	89.3	78.7	80.6	64.7	91.7
1AS (0.4 FPPI)	86.9	90.5	84.4	82.1	65.0	85.7	61.7	71.0	58.8	93.5
3AS (0.4 FPPI)	85.4	90.5	84.4	75.0	40.0	78.6	72.3	64.5	41.2	90.3
4AS (0.4 FPPI)	80.8	71.4	90.3	78.6	30.0	64.3	68.1	51.6	41.2	90.3
Harris (0.4 FPPI)	73.8	47.6	62.5	53.6	70.0	39.3	72.3	45.2	82.4	91.0
Dog (0.4 FPPI)	49.2	28.6	71.9	92.9	45.0	17.9	59.6	29.0	58.8	89.2
Log (0.4 FPPI)	56.2	38.1	34.4	60.7	35.0	25.0	74.5	48.4	58.8	90.3
{1,2,3}AS (0.4 FPPI)	87.7	85.7	87.1	82.1	80.0	85.7	74.5	83.9	58.8	94.2
PAS + Harris (0.4 FPPI)	87.7	71.4	90.3	75.0	75.0	64.3	80.9	64.5	82.4	95.7
HoG (0.4 FPPI)	74.6	9.5	32.3	78.6	85.0	17.9	53.2	83.9	82.4	70.1

Table 3: *All features and combinations tested, ranked according to their detection rates at 0.3 and 0.4 FPPI, averaged over all datasets but Shotton (for which we evaluate in terms of precision/recall).*

Method	Average DR at 0.3 FPPI	Avearge DR at 0.4 FPPI
PAS	76.7	81.8
{1,2,3}AS	74.8	80.6
PAS + Harris	70.0	76.8
1AS	69.4	76.2
3AS	64.1	70.2
4AS	56.5	64.0
HoG	56.6	57.5
Harris	55.0	60.7
Dog	42.9	50.3
Log	43.1	47.9

without including nearby spurious edgels. Moreover, they can form a wide variety of local shape structures, combine informativeness and repeatability, and constitute complete, scale-invariant local features ready to be used in many recognition or image matching frameworks [4].

We have demonstrated kAS within a sliding-window object detector, where windows are subdivided into tiles, each described by a bag of kAS. Extensive evaluations brought several interesting conclusions. First, the optimal number of tiles decreases with increasing complexity k, as a result of the shifting trade-off between the resolution of localization information versus the rigidity of the representation. Second, PAS perform better than other kAS, as they bring the best compromise between distinctiveness and repeatability, while also yielding a good proportion of pure boundary features. Third, kAS work substantially better than interest points for shape-based classes, and, finally, our PAS-based object detection system compares favorably to the state-of-the-art method [4].

The use of kAS in this paper is limited to a rather simple detection framework, through which we analysed their properties and performance. However, we expect kAS to be useful in other systems and tasks, with possibly other behaviors. For example, although in our analysis $2AS$ worked best, kAS of higher complexity are attractive when the localization constraints are weaker or absent and hence the discriminative power of individual features might become more important [16, 5, 2], or when higher degrees of geometric invariance are required (e.g. image matching [30, 21]). Besides, since our object detector is restricted to a single viewpoint, it is unclear how well kAS would work in a multi-view setting. Finally, effective ways to combine appearance features with kAS remain to be investigated. One option would be to integrate both tightly, by augmenting kAS with appearance information (e.g. by describing color or texture properites on either side of a L-shaped PAS).

[4] We have released an executable to detect and describe kAS on `lear.inrialpes.fr/software`

References

[1] E. Bart, E. Byvatov, and S. Ullman, View-Invariant Recognition Using Corresponding Object Fragments, *ECCV*, 2004.

[2] A. Berg, T. Berg and J. Malik, *Shape Matching and Object Recognition using Low Distortion Correspondence*, CVPR, 2005.

[3] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, Visual Categorization with Bags of Keypoints, *SLCV workshop in conjunction with ECCV*, 2004.

[4] N. Dalal, and B. Triggs, Histograms of Oriented Gradients for Human Detection, *CVPR*, 2005.

[5] G. Dorko, and C. Schmid, Selection of Scale-Invariant Parts for Object Class Recognition, *ICCV*, 2003.

[6] F. Estrada, and A. Jepson Perceptual Grouping for Contour Extraction, *ICPR*, 2004.

[7] L. Fei-Fei, R. Fergus, and P. Perona, Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories, *GMBV workshop of CVPR*, 2004. Caltech 101 database: www.vision.caltech.edu/Image_Datasets/Caltech101/Caltech101.html

[8] V. Ferrari, T. Tuytelaars, and L. Van Gool, *Real-time Affine Region Tracking and Coplanar Grouping*, CVPR, 2001.

[9] V. Ferrari, T. Tuytelaars, and L. Van Gool, Object Detection by Contour Segment Networks, ECCV, 2006; Dataset: www.vision.ee.ethz.ch/∼ferrari

[10] R. Fergus, P. Perona, and A. Zisserman, *Object class recognition by unsupervised scale-invariant learning*, CVPR, 2003.

[11] P. Gros, Matching and Clustering: Two Steps Toward Automatic Object Modeling in Computer Vision, IJRR, 2005

[12] D. Jacobs, *Robust and Efficient Detection of Convex Groups*, PAMI, 18:1, 1996.

[13] F. Jurie, and C. Schmid, Scale-invariant shape features for recognition of object categories, *CVPR*, 2004. Dataset: lear.inrialpes.fr/data

[14] F. Jurie, and B. Triggs, Creating Efficient Codebooks for Visual Recognition, *ICCV*, 2005.

[15] S. Lazebnik, C. Schmid, and J. Ponce, Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories, CVPR, 2006.

[16] B. Leibe, and B. Schiele, Scale-Invariant Object Categorization using a Scale-Adaptive Mean-Shift Search, *DAGM*, 2004.

[17] D. Lowe, T. Binford, *Perceptual organization as a basis for visual recognition*, AAAI, 1983 .

[18] D. Lowe, *Three-Dimensional Object Recognition from Single Two-Dimensional Images*, AI Journal, 1987.

[19] D. Lowe, Distinctive Image Features from Scale-Invariant Keypoints *IJCV*, 2004

[20] D. Martin, C. Fowlkes and J. Malik, *Learning to detect natural image boundaries using local brightness, color, and texture cues*, PAMI, 26(5):530-549, 2004.

[21] K. Mikolajczyk, C. Schmid Indexing based on scale invariant interest points, *ICCV*, 2001.

[22] A. Opelt, A. Pinz, and A. Zisserman, A Boundary-Fragment-Model for Object Detection, ECCV, 2006.

[23] A. Pope, and D. Lowe, Learning Object Recognition Models from Images, *Early Visual Learning*, 1996.

[24] F. Porikly, Integral Histogram: A Fast Way to Extract Histograms in Cartesian Spaces, *CVPR*, 2005.

[25] C. Rothwell, J. Mundy, W. Hoffman, and V.-D. Nguyen, Driving Vision by Topology, *IEEE Intl. Symposium on Computer Vision*, 1995.

[26] C. Rothwell, A. Zisserman, D. Forsyth, and J. Mundy, Planar Object Recognition using Projective Shape Representation, *IJCV*, 1995.

[27] A. Selinger, R. Nelson, *A Cubist approach to Object Recognition*, ICCV, 1998.

[28] A. Shashua, and S. Ullman, Structural Saliency: The Detection of Globally Salient Structures Using a Locally Connected Network, *ICCV*, 1998.

[29] J. Shotton, A. Blake, and R. Cipolla, Contour-Based Learning for Object Detection, *ICCV*, 2005.

[30] T. Tuytelaars and L. Van Gool, Matching Widely Separated Views based on Affine Invariant Regions *IJCV*, 2004.

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

[32] M. Zerroug, and R. Nevatia, Volumetric Descriptions from a Single Intensity Image *IJCV*, 1996.

[33] PASCAL Challenge 2006 Visual Object Classes: www.pascal-network.org/challenges/VOC/voc2006

[34] J. Zhang, M. Marszalek, S. Lazebnik, C. Schmid, Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study, *IJCV*, 2006 (to appear).

Contents

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399