IDIAP RESEARCH REPORT

# Stationary Features and Cat Detection

François Fleuret [1] Donald Geman [2]

IDIAP–RR 07-56

October 25, 2007

submitted for publication

[1] IDIAP Research Institute, Centre du Parc, av. des Prés-Beudin, 20, Case Postale 592, 1920 Martigny, Switzerland, fleuret@idiap.ch

[2] Johns Hopkins University, Clark Hall 302A, 3400 N. Charles Street, Baltimore, MD 21218, USA, geman@jhu.edu

**IDIAP Research Institute** **www.idiap.ch**

Av. des Prés–Beudin 20 P.O. Box 592 1920 Martigny – Switzerland
Tel: +41 27 721 77 11 Fax: +41 27 721 77 12 Email: info@idiap.ch

IDIAP Research Report 07-56

# Stationary Features and Cat Detection

François Fleuret Donald Geman

October 25, 2007

submitted for publication

**Abstract.** Most discriminative techniques for detecting instances from object categories in still images consist of looping over a partition of a pose space with dedicated binary classifiers. The efficiency of this strategy for a complex pose, i.e., for fine-grained descriptions, can be assessed by measuring the effect of sample size and pose resolution on accuracy and computation. Two conclusions emerge: i) fragmenting the training data, which is inevitable in dealing with high in-class variation, severely reduces accuracy; ii) the computational cost at high resolution is prohibitive due to visiting a massive pose partition.

To overcome data-fragmentation we propose a novel framework centered on pose-indexed features which assign a response to a pair consisting of an image and a pose, and are designed to be stationary: the probability distribution of the response is always the same if an object is actually present. Such features allow for efficient, one-shot learning of pose-specific classifiers.

To avoid expensive scene processing, we arrange these classifiers in a hierarchy based on nested partitions of the pose as in previous work, which allows for efficient search. The hierarchy is then "folded" for training: all the classifiers at each level are derived from one base predictor learned from all the data. The hierarchy is "unfolded" for testing: parsing a scene amounts to examining increasingly finer object descriptions only when there is sufficient evidence for coarser ones. In this way, the detection results are equivalent to an exhaustive search at high resolution. We illustrate these ideas by detecting and localizing cats in highly cluttered greyscale scenes.

# Contents

# 1 Introduction

This work is about a new strategy for supervised learning designed for detecting and describing instances from semantic object classes in still images. Conventional examples include faces, cars and pedestrians. We want to do more than say whether or not there are objects in the scene; we want to provide a description of the pose of each detected instance, for example the locations of certain landmarks. More generally, pose could refer to any properties of object instantiations which are not directly observed; however, we shall concentrate on geometric descriptors such as scales, orientations and locations.

The standard discriminative approach, meaning based on classifiers induced from training data, is to learn a pose-specific binary classifier and apply it many times ([16, 15, 20, 11]). Usually, there is an outer loop which visits certain locations and scales with a sliding window, and a purely learning-based module which accommodates all other sources of variation and predicts whether or not a sub-window corresponds to a target.

Parsing the scene in this manner already exploits knowledge about transformations which preserve object identities. In particular, translating and scaling the training images to a reference pose allows for learning a base classifier with all the training examples, a trivial example of "data-aggregation." The alternative, namely learning a great many separate classifiers each dedicated to a sub-population of objects with highly constrained poses, would require a massive amount of training data due to "data-fragmentation." Therefore, to date, the discriminative, data-aggregation approach has been applied almost exclusively to learning rather coarse geometric descriptions, such as a facial landmark and in-plane orientation.

Evidently, performance improves with having either more, or more richly annotated, training data; with allowing for more computation; and by targeting a more homogeneous population or coarser resolution. This inevitably leads to trade-offs. For example, in order to make learning more efficient by reducing variation it may be necessary to build several classifiers dedicated to increasingly more constrained sub-populations, which fragments the training data and increases the amount of computation. Also, it may be difficult to exploit rich ground-truth if the number of samples is limited, and there is an obvious trade-off between the efficacy of the individual classifiers and the overall computation at any target resolution. These trade-offs are clearly seen for cascades ([20, 21]): at a high true positive rate, reducing false positives could only come at the expense of considerable computation due to dedicating the cascade to a highly constrained pose, hence increasing dramatically the number of classifiers to train and evaluate in order to parse the scene.

We propose an experiment to quantify these trade-offs and then a framework to avoid being obliged to make them. Consider partitioning the space of poses at different resolutions or granularities into cells of about the same size. For each partition, build a binary classifier for each cell. There are two experimental variables besides the resolution of the partition: the data may be either fragmented or aggregated during training and the overall cost of executing all the classifiers may or may not be equalized. Not surprisingly, the best performance occurs with aggregated training at high resolution, but the on-line computational cost is formidable.

The framework we propose rests on two core ideas. One, which is not new, is to control online computation by using a hierarchy of classifiers corresponding to a recursive partitioning of the pose space, i.e., parameterizations of increasing complexity. A richer parametrization is considered only when "necessary", meaning the object hypothesis cannot be ruled out with a simpler one (see, e.g., [5, 19]). (Note that cascades are efficient for a similar reason - they are coarse-to-fine in terms of background rejection.) However, hierarchical organization alone is unsatisfactory because it does not solve the data-fragmentation problem. Unless data can be synthesized to generate many dedicated sets of positive samples, one set per node in the hierarchy, the necessity of training a classifier for every node leads to massive data fragmentation, hence small node-specific training sets, which cripples performance.

The second idea, the new one, is to avoid data-fragmentation by using pose-specific classifiers trained with "stationary features", a generalization of the underlying implicit parametrization of the

features by a scale and a location in all the discriminative learning techniques mentioned earlier. Each stationary feature is "pose-indexed" in the sense of assigning a numerical value to each combination of an image and a pose (or subset of poses). The desired form of stationarity is that, for any given pose, the joint *distribution* of the responses of the features over images containing an object at that pose does not depend on the pose. Said another way, if an image and an object instance at a given pose are selected, and only the responses of the stationary features are provided, one cannot guess the pose.

Given that objects are present, a stationary feature evaluated at one pose is then the "same" as at any other, but not in a literal, point-wise sense as functions, but rather in the statistical, population sense described above. In particular, stationary features are not "object invariants" in the deterministic sense of earlier work ([13]) aimed at discovering algebraic and geometric image functionals whose actual values were invariant with respect to the object pose. Our aim is less ambitious: our features are only "invariant" in a statistical sense.

Surprisingly, this does not appear to have been done before (or at least explicitly formulated and analyzed) even though stationarity is all that is needed to aggregate data and yet preserve the standard properties of a training set. This makes it possible, and effective, to analytically construct an entire family of pose-specific classifiers – all those at a given level of the hierarchy – using one base classifier induced from the entire training set. In effect, each pose-specific classifier is a "deformation" of the base classifier. Hence the number of classifiers to train grows linearly, not exponentially, with the depth of the pose hierarchy. This is what we call a folded hierarchy of classifiers.

In practice, this formulation encompasses standard strategies based on translating and scaling Haar wavelets or edge detectors to compute the response of a classifier for a given location and scale. But it also opens the way for training classifiers based on checking consistency among parts or deformations of parts instead of relying exclusively on their marginal appearance. Such a capability is indeed exploited by the detector we designed for finding cats and greatly improves the performance compared to individual part detection. This gain is shown in Figure 12, the main result of the paper, which compares ROC curves for a coordinated search for body parts (HB detector) with two separate ones (H|B detector).

In § 2, we summarize previous, related work on object detection in still images. Our notation and basic ideas are formally introduced in § 3, highlighting the difference between transforming the signal and the features. In § 4 we describe the actual implementation of this framework for detecting cats. The motivating experiment is presented in § 5, in which we substantiate our claims about the forced trade-offs when conventional approaches are applied to estimating a complex pose. Embedding pose-indexed classifiers in a hierarchy is described in § 6 and our main experiments on cat detection are reported in § 7. Finally, some concluding remarks appear in § 8.

# 2 Related Work

We characterize other work in relation to the two basic components of our detections strategy: explicit modeling of a hidden pose parameter, as in many generative and discriminative methods, and formulating detection as a controlled "process of discovery" during which computation is invested in a highly adaptive and unbalanced way depending on the ambiguities in the data.

## 2.1 Hidden variables

A principal source of the enormous variation in high-dimensional signals (e.g., natural images) is the existence of a hidden state which influences many components (e.g., pixel intensities) simultaneously, creating complex statistical dependencies among them. Still, even if this hidden state is of high dimension, it far simpler than the observable signal itself. Moreover, since our objective is to interpret the signal at a semantic level, much of the variation in the signal is irrelevant.

In fact, conditioning on the value of the hidden state, which means, in practice, testing for the presence of a target with a given pose, often leads to very simple, yet powerful, statistical models,

by exploiting the increased degree of independence among the components of the signal. This means decisions about semantic content can be based on directly aggregating evidence (naive Bayes). The problem is computational: there are many possible hidden states.

The extreme application of this conditioning paradigm is template matching: if the pose is rich enough to account for all non-trivial statistical variation, then even a relatively simple metric can capture the remaining uncertainty, which is basically noise. But this requires intense online computation to deform images or templates many times. One motivation of our approach is to avoid such online, global image transformations.

Similarly, the purest learning techniques, such as boosting ([20]) and convolution neural networks ([11]), rely on an algorithmic search through a subset of possible scales and locations in the image plane; that is, coarse scale and coarse location are not learned. Nor is invariance to illumination, usually handled at the feature level. However, invariance to other geometric aspects of the pose, such as rotation, and to fine changes in scale and translation, are accommodated implicitly, i.e., during classifier training.

On the contrary, "Part and Structure" models and other generative (model-based) approaches aim at more complex representations in terms of properties of "parts" ([12, 17, 3]). However, tractable learning and computation often require strong assumptions, such as conditional independence in appearance and location. In some cases, each part is characterized by the response of a feature detector, and the structure itself – arrangement of parts – can either be captured by a complex statistical model, incurring severe computation in both training and testing, or by a simple model by assuming conditional independence among part locations given several landmarks, which can lead to very efficient scene parsing with the use of distance transforms. It is not clear which of these techniques extends to highly articulated and deformable objects. For instance, modeling parts of cats (heads, ears, paws, tails, etc.) may be exceedingly difficult due to the low resolution and high variation in their appearance, and in the spatial arrangements among them. Compositional models ([10, 22, 14, 2]) appear promising.

### 2.2 A process of discovery

We do not regard the hidden pose as a "nuisance" parameter, secondary to detection itself, but rather as part of what it means to "recognize" an object. In this regard, we share the view expressed in [10], [3] and elsewhere that scene interpretation should go well beyond pure classification towards rich annotations of the instantiations of the individual objects detected.

In particular, we envision detection as an organized process of discovery, as in [1], and we believe that computation is a crucial issue and should be highly concentrated. Hierarchical techniques, which can accomplish focusing, are based on a recursive partitioning of the pose space (or object/pose space), which can be either ad-hoc ([9, 5]) or learned ([19, 7]). There is usually a hierarchy of classifiers, each one trained on a dedicated set of examples – those carrying a pose in the corresponding cell of the hierarchy. Often, in order to have enough data to train the classifiers, samples must be generated synthetically, which requires a sophisticated generative model. Our work is also related to the hierarchical template-matching ([8]), and to the cascade of classifiers in [20] and [21].

Relative to the tree-based methods, we use the stationary features to aggregate data and build only one base classifier per level in the hierarchy, from which all other classifiers are defined analytically. Finally, the fully hierarchical approach avoids the dilemma of cascades, namely the sacrifice of selectivity if the pose space is coarsely explored and the sacrifice of computation if it is finely explored, i.e., the cascades are dedicated to a very fine subset of poses.

## 3 Stationary Features and Classifiers

We regard the image as a random variable $I$ assuming values in $\mathcal{I}$. The set of possible poses for an object appearing in $I$ is $\mathcal{Y}$. We only consider geometric aspects of pose, such as the sizes of well-defined

| |
|---|
| $\mathcal{Y}$, the pose space |
| $\mathcal{Y}_1, \ldots, \mathcal{Y}_K$, a partition of the pose space $\mathcal{Y}$ |
| $\mathcal{Z}$, a $W \times H$ pixel lattice |
| $\mathcal{I} = \{0, \ldots, 255\}^{\mathcal{Z}}$, a set of gray-scale images of size $W \times H$ |
| $I$, a random variable taking values in $\mathcal{I}$ |
| $Y_k$, a Boolean random variable indicating if there is a target in $I$ with pose in $\mathcal{Y}_k$ |
| $\mathbf{Y} = (Y_1, \ldots, Y_K)$ |
| $T$, the number of training images, each with or without targets |
| $\mathcal{T} = \left\{\left(I^{(t)}, \mathbf{Y}^{(t)}\right)\right\}_{1 \leq t \leq T}$, the training set |
| $f_k : \mathcal{I} \to \{0, 1\}$, a predictor of $Y_k$ based on the image |
| $\xi : \mathcal{I} \to \mathbb{R}^N$, a family of base image features |
| $\psi : \{1, \ldots, K\} \times \mathcal{I} \to \mathcal{I}$, an image transformation intended to normalize a given pose |
| $\mathbf{X} : \{1, \ldots, K\} \times \mathcal{I} \to \mathbb{R}^Q$, a family of pose-indexed features |
| $\mathbf{X}(k)$, the r.v. corresponding to $\mathbf{X}(k, I)$ |
| $g : \mathbb{R}^Q \to \{0, 1\}$, a predictor trained from all the data |

Table 1: Notation

parts and the locations of distinguished points. Let $\mathcal{Y}_1, \ldots, \mathcal{Y}_K$ be a partition of $\mathcal{Y}$. We are interested in partitions of varying granularities, ranging from rather coarse resolution (small $K$) to rather fine resolution (larger $K$). Finally, for every $k = 1 \ldots K$, let $Y_k$ be a Boolean random variable indicating whether or not there is a target in $I$ with pose in $\mathcal{Y}_k$. The binary vector $(Y_1, \ldots, Y_K)$ is denoted $\mathbf{Y}$.

In the case of merely detecting and localizing an object of fixed size in a gray-scale image of size $W \times H$, natural choices would be $\mathcal{I} = [0, 1]^{WH}$ and $\mathcal{Y} = [0, W] \times [0, H]$, the image plane itself; that is, the pose reduces to one location. If the desired detection accuracy were 5 pixels, then the pose cells might be disjoint $5 \times 5$ blocks and $K$ would be approximately $\frac{WH}{25}$. On the other hand, if the pose accommodated scale and multiple points of interest, then obviously the same accuracy in the prediction would lead to a far larger $K$, and any detection algorithm based on looping over pose cells would be highly costly.

We denote by $\mathcal{T}$ a training set of images labeled with the presences of targets

$$\mathcal{T} = \left\{\left(I^{(t)}, \mathbf{Y}^{(t)}\right)\right\}_{1 \leq t \leq T}, \tag{1}$$

where each $I^{(t)}$ is a full image, and $\mathbf{Y}^{(t)}$ is the Boolean vector indicating the pose cells occupied by targets in $I^{(t)}$. We write

$$\xi : \mathcal{I} \to \mathbb{R}^N,$$

for a family of $N$ image features such as edge detectors, color histograms, Haar wavelets, etc. These are the "base features" $(\xi_1, \ldots, \xi_N)$ which will be sampled to generate our stationary feature vector. We will write $\xi(I)$ when we wish to emphasize the mapping and just $\xi$ for the associated random variable. The dimension $N$ is sufficiently large to account for all the variations of the feature parameters, such as locations of the receptive fields, orientations and scales of edges, etc.

In the next section, § 3.1, we consider the problem of "data-fragmentation", meaning that specialized predictors are trained with subsets of the positive samples. Then, in § 3.2, we formalize how fragmentation has been conventionally avoided in simple cases by normalizing the signal itself; we then propose in § 3.3 the idea of pose-indexed, stationary features, which avoids global, signal normalization both offline and online and opens the way for dealing with complex pose spaces.

### 3.1 Data fragmentation

Without additional knowledge about the relation between $\mathbf{Y}$ and $I$, the natural way to predict $Y_k$ for each $k = 1 \dots K$ is to train dedicated classifier

$$f_k : \mathcal{I} \rightarrow \{0, 1\}$$

with the training set

$$\left\{ \left( I^{(t)}, Y_k^{(t)} \right) \right\}_{1 \leq t \leq T}$$

derived from $\mathcal{T}$. This corresponds to generating a single sample from each training scene, labeled according to whether or not there is a target with pose in $\mathcal{Y}_k$. This is *data-fragmentation*: training $f_k$ involves only those data which exactly satisfy the pose constraint; no synthesis or transformations are exploited to augment the number of samples available for training. Clearly, the finer the partitioning of the pose space $\mathcal{Y}$, the fewer positive data points are available for training each $f_k$.

Such a strategy is evidently foolhardy in the standard detection problems where the pose to be estimated is the location and scale of the target since it would mean separately training a predictor for every location and every scale, using as positive samples only full scenes showing an object at that location and scale. The relation between the signal and the pose is obvious and normalizing the positive samples to a common reference pose by translating and scaling them is the natural procedure; only one classifier is trained with all the data. However, consider a face detection task for which the faces to detect are known to be centered and of fixed scale, but are of unknown out-of-plane orientation. Unless 3D models are available, from which various views can be synthesized, the only course of action is data-fragmentation: partition the pose space into several cells corresponding to different orientation ranges and train a dedicated, range-specific classifier with the corresponding positive samples.

### 3.2 Transforming the signal to normalize the pose

As noted above, in simple cases the image samples can be normalized in pose. More precisely, both training and scene processing involve normalizing the image through a pose-indexed transformation

$$\psi : \{1, \dots, K\} \times \mathcal{I} \rightarrow \mathcal{I}.$$

The "normalization property" we desire with respect to $\xi$ is that the conditional probability distribution of $\xi(\psi(k, I))$ given $Y_k = 1$ be the same for every $1 \leq k \leq K$.

The intuition behind this property is straightforward. Consider for instance a family of edge detectors and consider again a pose consisting of a single location $z$. In such a case, the transformation $\psi$ applies a translation to the image to move the center of pose cell $\mathcal{Y}_k$ to a reference location. If a target was present with a pose in $\mathcal{Y}_k$ in the original image, it is now at a reference location in the transformed image, and the distribution of the response of the edge detectors in that transformed image does not depend on the initial pose cell $\mathcal{Y}_k$.

We can then define a new training set

$$\left\{ \left( \xi \left( \psi(k, I^{(t)}) \right), Y_k^{(t)} \right) \right\}_{1 \leq k \leq K, 1 \leq t \leq T}$$

with elements residing in $\mathbb{R}^N \times \{0, 1\}$. Due to the normalization property, and under mild conditions, the new training set indeed consists of independent and identically distributed components (see the discussion in the following section). Consequently, this set allows for training a classifier

$$g : \mathbb{R}^N \rightarrow \{0, 1\}$$

from which we can analytically define a predictor of $Y_k$ for any $k$ by

$$f_k(I) = g\left(\xi(\psi(k, I))\right)$$

This can be summarized algorithmically as follows: In order to predict if there is a target in image $I$ with pose in $\mathcal{Y}_k$, first normalize the image with $\psi$ so that a target with pose in $\mathcal{Y}_k$ would be moved to a reference pose cell, then extract features in that transformed image using $\xi$, and finally evaluate the response of the predictor $g$ from the computed features.

### 3.3 Stationary features

The pose-indexed, image-to-image mapping $\psi$ is computationally intensive for any non-trivial transformation. Even rotation or scaling induces a computational cost of $O(WH)$ for every angle or scale to test during scene processing, although effective shortcuts are often employed. Moreover, this transformation does not exist in the general case. Consider the two instances of cats shown in Figure 1. Rotating the image does not allow for normalizing both the head and the body orientations at the same time, and designing a non-affine transformation to do so would be unlikely to produce a realistic cat image as well as be computationally intractable when done many times. Finally, due to occlusion and other factors, there is no general reason *a priori* for $\psi$ to even exist.

Instead, we propose a different mechanism for data-aggregation based on pose-indexed features which directly assign a response to a pair consisting of an image and a pose cell and which satisfy a stationarity requirement. This avoids assuming the existence of a normalizing mapping in the image space, not to mention executing such a mapping many times online.

*A* **stationary feature vector** *is a pose-indexed mapping*

$$\mathbf{X} : \{1, \ldots, K\} \times \mathcal{I} \to \mathbb{R}^Q,$$

*with the property that the probability distribution*

$$P(\mathbf{X}(k) = \mathbf{x} \,|\, Y_k = 1), \quad \mathbf{x} \in \mathbb{R}^Q \tag{2}$$

*is the same for every* $k = 1, \ldots, K$*, where* $\mathbf{X}(k)$ *denotes the random variable* $\mathbf{X}(k, I)$.

The idea can be illustrated with two simple examples, a pictorial one in Figure 2 and a numerical one in § 3.4.

In practice, the relationship with $\xi$, the base feature vector, is simply that the components of the feature vector $\mathbf{X}(k)$ are chosen from among the components of $\xi$; the choice depends on $k$. In this case, we can write

$$\mathbf{X}(k) = (\xi_{\pi_1(k)}, \xi_{\pi_2(k)}, \ldots, \xi_{\pi_Q(k)}).$$

where $\{\pi_1(k), \ldots, \pi_Q(k)\} \subset \{1, \ldots, N\}$ is the ordered selection for index $k$. The ordering matters because we want (2) to hold and hence there is a correspondence among individual components of $\mathbf{X}(k)$ from one pose cell to another.

**Note:** We shall refer to (2) as the "stationarity" or "weak invariance" assumption. As seen below, this property justifies data-aggregation in the sense of yielding an aggregated training set satisfying the usual conditions. Needless to say, however, demanding that this property be satisfied exactly is not practical, even arguably impossible. In particular, with our base features, various discretizing effects come into play, including using quantized edge orientations and indexing base features with rectangular windows. Even designing the pose-indexed features to approximate stationarity by appropriately selecting and ordering the base features is non-trivial; indeed, it is the main challenge in our framework. Still, utilizing pose-indexed features which are even approximately stationary will turn out to be very effective in our experiments with cat detection.

The contrast between signal and feature transformations can be illustrated with the following commutative diagram: Instead of first applying a normalizing mapping $\psi$ to transform $I$ in accordance with a pose cell $k$, and then evaluating the base features, we directly compute the feature responses as functions of both the image and the pose cell.

Figure 1: Normalizing simultaneously the head and the body orientation cannot be done at the image level.

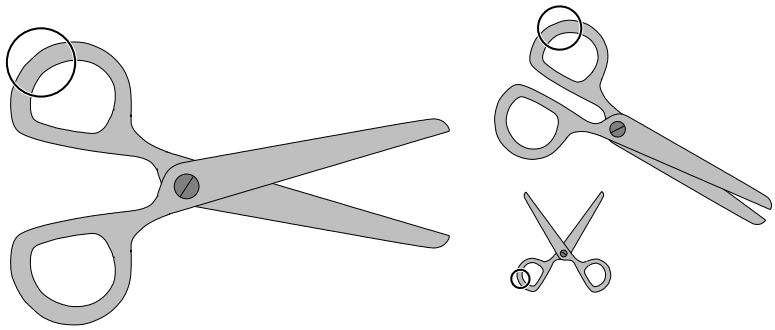
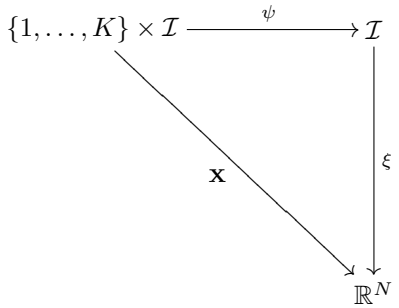


Figure 2: An idealized example of stationary features. The pose of the scissors could be the locations of the screw and the two tips, in which case one might measure the relative frequency a particular edge orientation inside in a disc whose radius and location, as well as the chosen orientation, depends on the pose. If properly designed, the response statistics have a distribution which is invariant to the pose when in fact a pair of scissors is present. See § 3.3.




Once provided with $\mathbf{X}$, a natural training set consisting of $TK$ samples is provided by

$$\mathcal{T}_{agg} = \left\{ \left( \mathbf{X}^{(t)}(k), Y_k^{(t)} \right) \right\}_{1 \leq t \leq T,\, 1 \leq k \leq K} . \tag{3}$$

Under certain conditions, the elements of this training set will satisfy the standard assumption of being independent and identically distributed. One condition, the key one, is stationarity, but technically three additional conditions would be required: i) property (2) extends to conditioning on $Y_k = 0$; ii) the "prior" distribution $P(Y_k = 1)$ is the same for every $k = 1, \ldots, K$; iii) for each $t$, the samples $\mathbf{X}^{(t)}(k), k = 1, \ldots, K$, are independent. The first condition says that the background distribution of the pose-indexed features is spatially homogeneous, the second that all pose cells are *a priori* equally likely and the third, dubious but standard, says that the image data associated with different pose cells are independent despite some overlap.

It therefore makes sense to train a predictor $g : \mathbb{R}^Q \to \{0, 1\}$ using the training set (3). We can then *define*

$$f_k(I) = g(\mathbf{X}(k, I)), \quad k = 1, \ldots, K. \tag{4}$$

Notice that the family of classifiers $\{f_k\}$ is also "stationary" in the sense that conditional distribution of $f_k$ given $Y_k = 1$ does not depend on $k$.

### 3.4 Toy example

We can illustrate the idea of stationary features with a very simple roughly piecewise constant, one-dimensional signal $I(n), n = 1, ..., N$. The base features are just the components of the signal itself: $\xi(I) = I$. The pose space is

$$\mathcal{Y} = \left\{ (\theta_1, \theta_2) \in \{1, \ldots, N\}^2,\, 1 < \theta_1 < \theta_2 < N \right\}$$

and the partition is the finest one whose cells are individual poses $\{(\theta_1, \theta_2)\}$; hence $K = |\mathcal{Y}|$. For simplicity, assume there is at most one object instance, so we can just write $Y = (\theta_1, \theta_2) \in \mathcal{Y}$ to denote an instance with pose $(\theta_1, \theta_2)$. For $\mathbf{u} = (u_1, ..., u_N) \in \mathbb{R}^N$, the conditional distribution of $I$ given $Y$ is

$$\begin{aligned} P(I = \mathbf{u} \,|\, Y = (\theta_1, \theta_2)) &= \prod_n P(I(n) = u_n \,|\, Y = (\theta_1, \theta_2)) \\ &= \prod_{n < \theta_1} \phi_0(u_n) \prod_{\theta_1 \leq n \leq \theta_2} \phi_1(u_n) \prod_{\theta_2 < n} \phi_0(u_n) \end{aligned}$$

where $\phi_\mu$ is a normal law with mean $\mu$ and standard deviation 0.1. Hence the signal fluctuates around 0 on the "background" and around 1 on the target, see Figure 3.

We define a four-dimensional pose-indexed feature vector taking the values of the signal at the extremities of the target, that is

$$\mathbf{X}((\theta_1, \theta_2), I) = (I(\theta_1 - 1), I(\theta_1), I(\theta_2), I(\theta_2 + 1)).$$

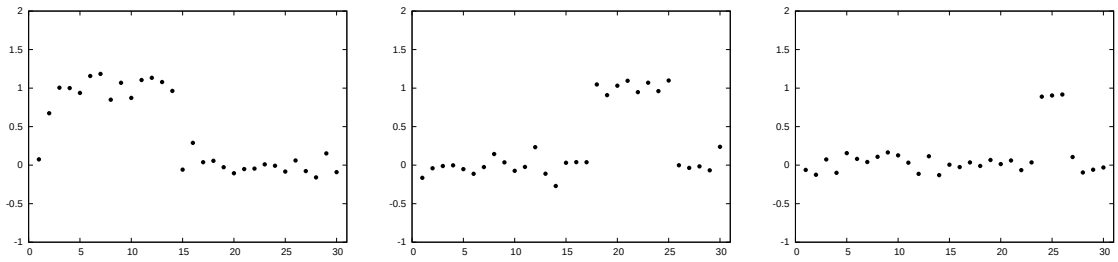


Figure 3: Examples of toy scenes

Clearly,

$$P(\mathbf{X}(\theta_1,\theta_2) = (x_1,x_2,x_3,x_4) \,|\, Y_{\theta_1,\theta_2} = 1) = \phi_0(x_1)\phi_1(x_2)\phi_1(x_3)\phi_0(x_4)$$

which is not a function of $\theta_1, \theta_2$. Consequently, $\mathbf{X}$ is stationary and the common law in (2) is $\phi_0 \times \phi_1 \times \phi_1 \times \phi_0$.

# 4 Cat Detection

In this section we present specific examples of poses, base features and stationary features for the case of detecting cats. The original training images and available ground truth are described in § 4.1. Then, in § 4.2, we define a family of highly robust, base image features based on counting edge frequencies over rectangular areas, and in § 4.3 we propose a way to index such features with the pose of a cat defined by its head and belly locations. The final subsection, § 4.4, provides details about the classifier we induce from our stationary features and how we are able to train with several million negative samples by sub-sampling from them at every step of a classical AdaBoost procedure ([6]).

## 4.1 Cat images and poses

The cat images were randomly sampled from the web site RateMyKitten[1]; we are grateful to Harrison Page for providing us with this remarkable source of image data. Images of cluttered scenes without cats, mostly home interiors, were sampled from various web sites. The complete database we are using has 2,330 images containing a total of 1,988 cats.

For each experiment, we split this database at random into a training set containing 85% of the images, and a test set containing the other 15%.

Each cat was manually annotated with one circle roughly outlining the head, from which the head size (diameter) and head location (center) are derived, and one point placed more or less, quite subjectively, at the center of mass, which we have referred to as the "belly" location. Hence, the pose of a cat takes the form $(h, b, s)$ where $h$ is the location in the image plane $\mathcal{Z}$ of the center of the head, $b$ is the belly location and $s$ is the head diameter.

Finally, all the experiments are performed at one scale range: Each cat scene has been successively rescaled by factor of $(\frac{1}{2})^{\frac{s}{5}}$ until the cat head size (diameter) falls below 50 pixels. This yields a scale range of $[43, 50]$ pixels. Larger cats would be detected by repeatedly downsampling by the same factor. Scenes with no cat are rescaled with a scaling factor picked at random according to the scale distribution on the cats.

## 4.2 Base image features

An image is pre-processed by computing, at every location $z \in \mathcal{Z}$, the responses of eight edge-detectors similar to those proposed in [1] (see Figures 4 and 5), but at three different scales, ending up with 24

[1] `http://www.ratemykitten.com`

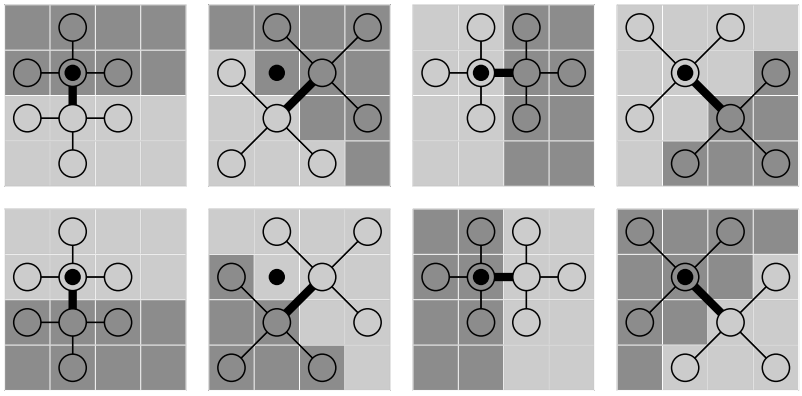


Figure 4: Our edge-detectors: For each of four orientations and two polarities, an edge is detected at a certain location (the dark circle) if the absolute difference between the intensities of the two pixels linked by the thick segment is greater than each of the six intensity differences for pixels connected by a thin segment.

Boolean features $e_1(z), \ldots, e_{24}(z)$ corresponding to four orientations and two polarities. In addition, we add a variance-based binary test $e_0(z)$ which responds positively if the variance of the gray levels in a $16 \times 16$ neighborhood of $z$ exceeds a fixed threshold. Our features are based on counting the responses of any of these detectors over a rectangular areas ([4]), which can be done in constant time by using 25 integral images ([18]).

From these edge maps and the raw gray levels we define the following three types of image features:

1. **Edge proportion:** The proportion of an edge type in a rectangular window. Given a rectangular window $W$ and an edge type $\lambda \in \{0, 1, \ldots, 24\}$, the response is the number of pixels $z$ in $W$ for which $e_\lambda(z) = 1$, divided by the total number of pixels in $W$ if $\lambda = 0$ or by the number of pixels in $W$ for which $e_0(z) = 1$ if $\lambda > 0$.

2. **Edge orientation histogram distance:** Given again two rectangular windows $W_1$ and $W_2$, and a scale $s$, the response is the $L^1$ norm between the empirical eight-bin histograms of orientations corresponding to the eight edge detectors at scale $s$.

3. **Gray-scale histogram distance:** Given two rectangular windows $W_1$ and $W_2$, the response is the $L^1$ norm between the sixteen-bin empirical histograms of gray-scales for the two windows.

The rational behind the features of type 1 is to endow the classifiers with the ability to check for the presence of certain pieces of outlines or textures. The motivation for types 2 and 3 is to offer the capability of checking for similarity in either edge or gray-scale statistics between different parts of the image, typically to check for a silhouette in the case of very blurry contours. Some examples of features actually picked during the training are shown in Figure 13.

### 4.3 Indexing features by pose

As formalized in § 3.3, a pose-indexed feature is a real-valued functional of both a pose cell and an image. The features described in the previous section are standard functionals of the image alone. Since the response of any of them depends on counting certain edge types over rectangular windows in the image, we construct our family of pose-indexed features indirectly by indexing both the edge types and the window locations with the pose cell.

Precisely, for any pose cell index $k$, we compute the average head location $h$ and the average belly location $b$ of the pose cell $\mathcal{Y}_k$, and we add to the parameterization of each window two binary flags to

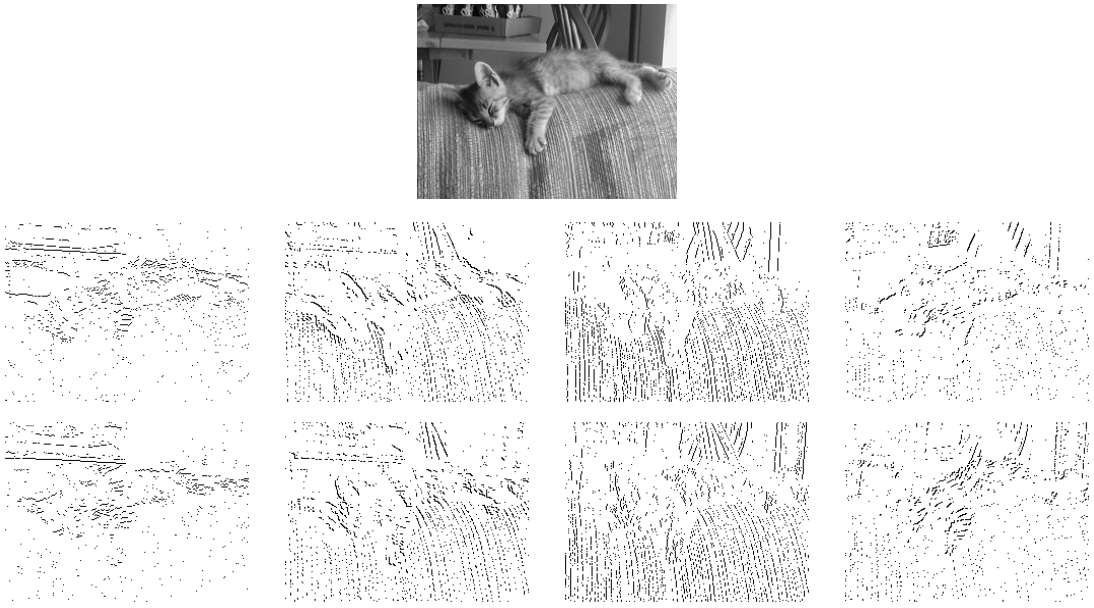
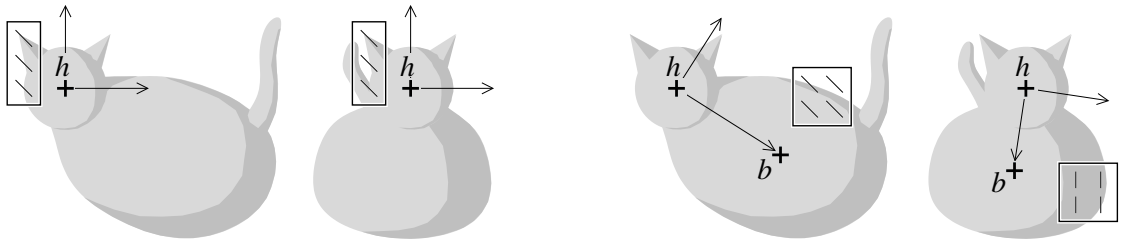


Figure 5: Result of the edge detection. Each one of the eight binary images in the two bottom rows corresponds to one orientation of the edge detectors of Figure 4.




(a) H-frame (b) HB-frame

Figure 6: Registration of the pose-indexed features for cats. Given the center $(h, b)$ of a pose cell, we define two registered frames: the H-frame is relative to $h$ alone and is neither scaled nor rotated according to the belly location. The HB-frame is relative to both $h$ and $b$; its basis vectors are $\vec{hb}$ and an orthogonal vector of fixed length. Hence the HB-frame rotates, and can be compressed in the head-belly axis but not orthogonally. Similarly, the edge type can be defined relatively to the $\vec{hb}$ orientation. The boxes show how a window defined in one of these frames is translated when the pose changes, and the short lines in the windows show the edge type the feature actually counts.

indicate whether the window location and the edge type are defined in a frame relative to the head location $h$ alone - the H-frame - or in a frame relative to both the head and the belly location $(h, b)$ - the HB-frame. See figure 6.

Windows defined in the H-frame simply translate with the head similarly to classical features in face detection, and hence do not depend on the belly location. In contrast, windows defined in the HB-frame move when the belly location moves, even if the head is fixed, and hence can, for instance, remain centered on the bulk of the body, or roughly at the interface between the body and the background.

Similarly, the edge type counted in a window relative to the H-frame does not depend on $b$ and hence is constant, while the one relative to the HB-frame is rotated according to the orientation of $\vec{hb}$, as show on the left of Figure 6.

Combining both reference frames in the same feature vector allows for adapting the classifier to the specific statistics of a cat pose, in particular the very loose dependence between the head and body orientations.

### 4.4 Classifiers: boosted trees and asymmetric weighting by sampling

Our objective is to train a classifier $g$, a standard mapping from $\mathbb{R}^Q$ into $\{0, 1\}$, from a training set of the form (3). The classifiers we used are linear combinations of short binary trees built from thresholded features.

At each step of boosting, a tree is trained based on the current sample weights, choosing the split at each node of the tree which minimizes the weighted error for the two-class decision. (Recall that $Y_k^{(t)}$ is the binary label for the $Q$-dimensional sample feature vector $\mathbf{X}^{(t)}(k)$ for $k = 1, \ldots, K, t = 1, \ldots, T$.) Not every feature is scored, meaning the optimal threshold and corresponding weighted error rate are computed. We only score a random sample of 1,000 of the $Q$ features and minimize the error rate over these. Sampling is done by randomly choosing a feature type (recall from § 4.2 there are three types), window locations, edge types, and for both the locations and the edge types one of the two registration modes (H-frame or HB-frame).

Special attention must be given to the characteristics of the populations of interests. In such a detection problem, the prior distribution is very skewed, with an extremely low probability of the presence of a target at a pose picked at random. This makes the corresponding classification problem highly unbalanced. Also, any tractable sampling of the negative population is still too small to account for the negative sub-population which lives close to positive examples.

A popular approach to this dilemma is to build a cascade of classifiers, each trained with all the positive examples and with a fresh sample of negative examples which survive the filtering of the previous classifiers in the cascade ([20, 21]). In this way the sampling is eventually concentrated on the "difficult" negative samples. This is similar in practice to what boosting itself is intended to do, namely ignore easily classified samples and concentrate on the difficult ones. We avoid the complexity of tuning such a cascade by using all the negative examples at every step through an asymmetric, sampling-based version of standard boosting. This provides an excellent approximation to the exact weighting for a fraction of the computational cost.

When picking the optimal split at a certain node, we approximate the weighted error with an error computed over all positive samples and a *random subset of negative samples* drawn according to the current boosting weights. Hence, we keep the response of the strong classifier up-to-date on the $S = KT \simeq 10^7$ samples, but we pick the optimal weak learners at every step based on $M \simeq 10^4$ samples.

More precisely, at a certain iteration of the boosting procedure, let $\omega_s$ denote the weight of sample $s = 1, \ldots, S$, let $Y_s \in \{0, 1\}$ be its true class, and let

$$\omega_{neg} = \sum_s \omega_s \mathbf{1}_{\{Y_s=0\}}$$

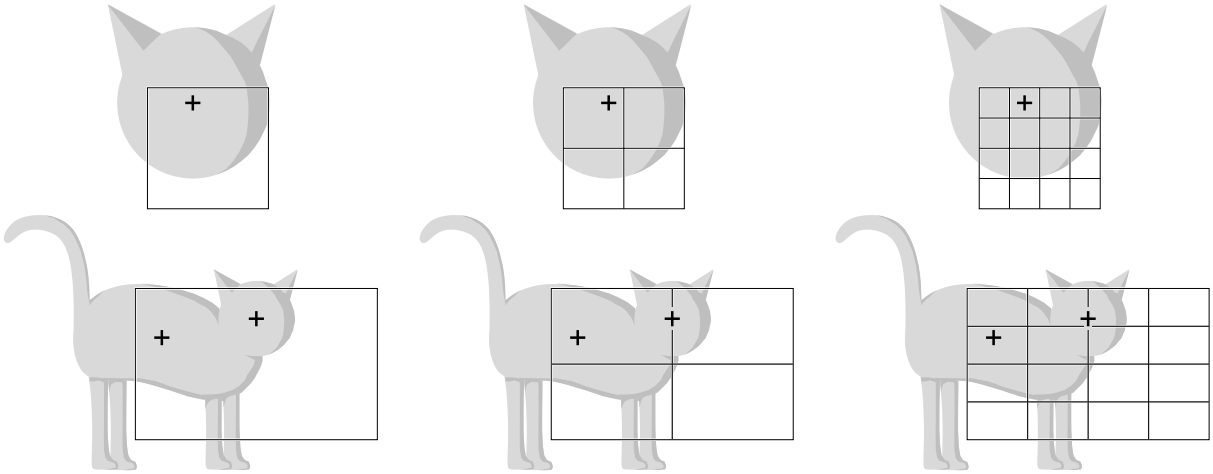


Figure 7: Top row: The head experiments involve predicting if a window of size $40 \times 40$ contains the center of a cat head. The head can be detected at three spatial resolutions, corresponding to partitioning the window into 1, 4 or 16 cells. Bottom row: The head-belly experiments involve predicting if a $5 \times 5$ square contains the center of a cat head whose belly is roughly centered in a window of size $160 \times 100$. Similarly, three resolutions in the location of the belly are considered.

be the total weight of the negative samples. We sample independently $M$ indices $S_1, \ldots, S_M$ in $\{1, \ldots, S\}$ according to the negative sample density

$$P(S_m = s) = \frac{\omega_s \mathbf{1}_{\{Y_s=0\}}}{\omega_{neg}}, \quad m = 1, \ldots, M.$$

Then we re-weight the training samples as follows:

$$\omega'_s = \begin{cases} \omega_s & \text{if } Y_s = 1 \\ \omega_{neg} \dfrac{\|\{m : S_m = s\}\|}{M} & \text{otherwise} \end{cases}$$

This can be seen as an approximation to the distribution on the full training set obtained by

- keeping all positive samples with their original weights,
- selecting a random subset of negative samples according to their original weights, and giving them a uniform weight.

Since this sub-sampling is done at every boosting step, any sample for which the classifier response is strongly incorrect will eventually be picked. In our experiments we sample four negative examples for every positive example. From a computational perspective this sampling is negligible as it only accounts for about 1% of the total training time. Finally, the coefficients for the weak learners in the final classifier are estimated with the true error rates computed with all the training samples.

# 5 Motivational Experiment: Quantifying Trade-offs

We present two series of experiments designed to study the impact on accuracy of data-fragmentation, with and without controlling for total online computation. In both series the goal is to predict the presence of a target with high accuracy in pose. The experimental settings and experimental results are presented in § 5.1 and in § 5.2, respectively.

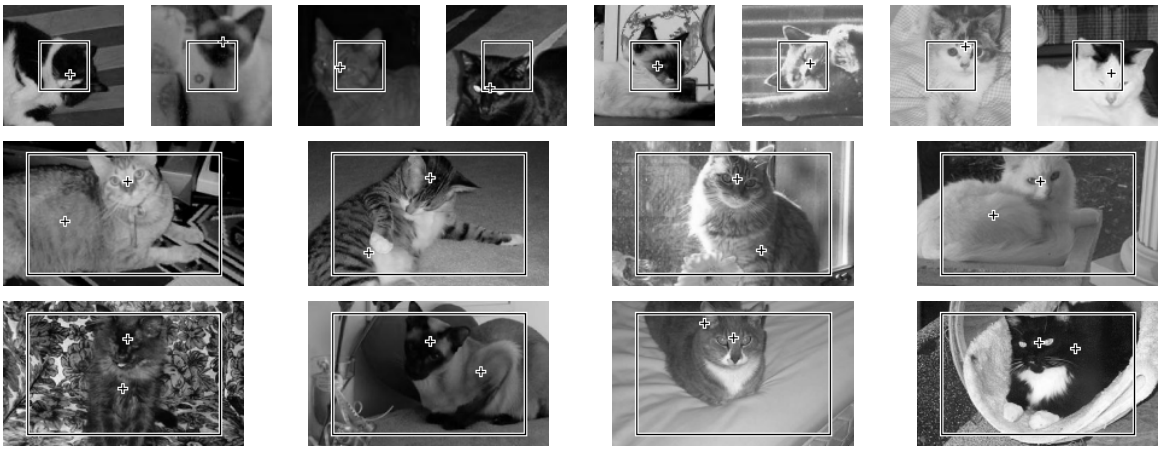


Figure 8: A few positive examples picked uniformly at random. Top row: samples from the head experiments. Bottom two rows: samples from the head-belly experiments. The crosses depict the head and belly centers provided on the training data. The boxes show the admissible pose domain $\mathcal{Y}$.

## 5.1 Settings

Since training with fragmentation is not feasible for any complete partition of a complex pose space at a realistic resolution, the images we consider in our experiments have been cropped from the original data set so that the pose space $\mathcal{Y}$ is already strongly constrained.

1. **Head experiments**: In the first series of experiments the target pose is the center of the cat head, constrained to $\mathcal{Y} = [-20, 20] \times [-20, 20]$. It is this pose space that will be investigated at different resolutions. The training set is constructed as follows: for *every* original image, and *every* non-overlapping $40 \times 40$ square, we crop a $100 \times 100$ image centered on the square. The total number of samples $T$ is hence equal to the number of non-overlapping $40 \times 40$ squares in the $2,327$ scenes. The top row of Figure 8 shows a few of these cropped scenes with a target.

2. **Head-Belly experiments**: In the second series of experiments the pose is the *pair of locations* $(h, b)$ for the head and belly, constrained to $\mathcal{Y} = ([0, 5] \times [0, 5]) \times ([-80, 80] \times [-20, 80])$. For every non-overlapping $5 \times 5$ square in every original image, we crop a $200 \times 140$ image centered at the square. It is only deemed a positive sample if it contains a cat head in the $5 \times 5$ square *and* the belly location is in the corresponding admissible $160 \times 100$ rectangle. (Ninety-five percent of the cats in the dataset satisfy this constraint on $(h, b)$.) The two bottom rows of Figure 8 show a few of these cropped scenes containing a target.

In both series, our objective is to compare the performance of classifiers trained when the data are fragmented or aggregated and when the computational cost is equalized or not. More precisely, we consider three partitions of $\mathcal{Y}$ into $K = 1$, 4 and 16 pose cells, as show in Figure 7. For instance, in the head experiment, for $K = 4$ (Figure 7, top-center) the $40 \times 40$ window is divided into four subwindows of size $20 \times 20$, namely $\mathcal{Y}_1 = [-40, 0] \times [-40, 0]$, $\mathcal{Y}_2 = [0, 40] \times [-40, 0]$, $\mathcal{Y}_3 = [-40, 0] \times [40, 0]$ and $\mathcal{Y}_4 = [0, 40] \times [0, 40]$.

In each series, we build four detection systems. Three of them are trained under data-fragmentation at the three considered resolutions, namely $K = 1$, $K = 4$ or $K = 16$ pose cells as depicted in Figure 7. The fourth classifier is trained with the pose-indexed, stationary features at the finest resolution $K = 16$. The stationary features are based on the H-frame alone for the head experiments, and on both the H-frame and the HB-frame for the head-belly experiments.

The computational cost for evaluating one such classifier is proportional to the number of stumps it combines. In the particular case of boosting, a classifier combining only a fixed number of weak learners

| | | Without cost equalization | | | With cost equalization | | |
|---|---|---|---|---|---|---|---|
| | | Fragmented | | Aggregated | Fragmented | | Aggregated |
| **TP** | $K = 1$ | $K = 4$ | $K = 16$ | $K = 16$ | $K = 4$ | $K = 16$ | $K = 16$ |
| 0.95 | 18.1 | 18.1 | 17.3 | 12.5 | 23.8 | 32.0 | 18.1 |
| 0.90 | 10.9 | 7.2 | 8.9 | 3.5 | 11.3 | 18.1 | 11.9 |
| 0.80 | 3.7 | 3.2 | 2.5 | 1.1 | 6.3 | 8.8 | 5.4 |
| 0.70 | 2.0 | 1.8 | 1.0 | 0.4 | 2.6 | 5.3 | 3.3 |
| 0.60 | 0.9 | 0.5 | 0.4 | 0.1 | 1.4 | 3.6 | 1.9 |
| 0.50 | 0.5 | 0.2 | 0.2 | 0.1 | 0.6 | 1.9 | 1.0 |

(a) Head experiments

| | | Without cost equalization | | | With cost equalization | | |
|---|---|---|---|---|---|---|---|
| | | Fragmented | | Aggregated | Fragmented | | Aggregated |
| **TP** | $K = 1$ | $K = 4$ | $K = 16$ | $K = 16$ | $K = 4$ | $K = 16$ | $K = 16$ |
| 0.95 | 173.1 | 177.5 | 686.0 | 80.4 | 379.4 | 980.7 | 1065.1 |
| 0.90 | 65.0 | 64.0 | 189.7 | 22.8 | 97.3 | 548.0 | 519.2 |
| 0.80 | 13.9 | 11.3 | 34.5 | 3.8 | 31.5 | 231.8 | 209.3 |
| 0.70 | 4.4 | 2.2 | 9.7 | 1.0 | 11.9 | 93.3 | 87.4 |
| 0.60 | 1.1 | 0.7 | 1.5 | 0.3 | 4.4 | 44.4 | 47.2 |
| 0.50 | 0.4 | 0.2 | 0.4 | 0.0 | 2.3 | 17.7 | 19.2 |

(b) Head-belly experiments

Table 2: Results of Experiment I: this table gives the average number of false alarms per scene of size $640 \times 480$ at several true positive rates, with or without cost equalization, and with or without fragmentation of the data.

is still effective, and hence, unlike many discriminative methods, computation is easy to control. This motivates a very simple strategy to equalize the cost among experiments: We simply control the total number of feature evaluations. Hence, if the classifier at resolution $K = 1$ is constructed from $M$ features, then each classifier for $K = 4$ uses $M/4$ features and each one for $K = 16$ uses $M/16$ features. We take $M = 1024$.

As a measure of performance, we estimate the number of false alarms for any given true positive rate. In order to compare results across resolutions, the labeling of detections as true or false positives occurs at the coarsest resolution. For simplicity, for the head-belly case, we only score the estimated head location.

## 5.2 Results

The results in Table 2 clearly demonstrate the gain in performance in constraining the population provided there is no fragmentation of the data. In the head experiments, even with fragmentation, higher resolution almost always results in fewer errors. This is not true for the head-belly experiments, where sixteen pose cells do worse than four, with or without cost equalization, which can be explained to some extent by the lower variation in the appearance of cat heads than full cat bodies, and hence fewer samples may be sufficient for accurate head detection.

As expected, without controlling the on-line computational cost, aggregation with stationary features is more discriminating than the fragmented classifiers in both experiments and at any true positive rate, reducing the false positive rate by a factor of at least three. Still, the performance of the classifiers when cost is equalized shows the influence of computation in this framework: at the finest

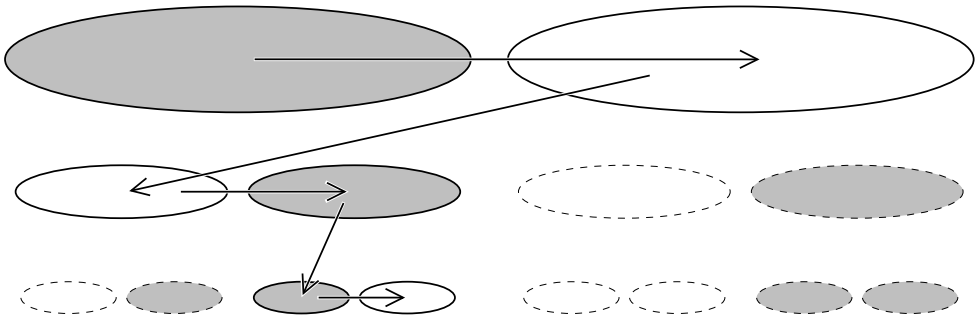


Figure 9: Hierarchical detection. Each ellipse on stands for a pose cell $\mathcal{Y}_k^{(d)}$, $k = 1, \ldots, K_d$, $d = 1, \ldots, D$. Here, $D = 3$ and $K_1 = 2$, $K_2 = 4$, $K_3 = 8$. Gray ellipses correspond to pose cells whose $f_k^{(d)}$ respond positively, and dashed ellipses correspond to pose cells whose classifiers are not evaluated during detection. As shown by the arrows, the algorithm ignores all sub-cells of a cell whose classifier responds negatively.

resolution, the number of false alarms in the head experiments increases by a factor greater than four at any true-positive rate, and by two orders of magnitude in the head-belly experiments.

These results also demonstrate the pivotal role of computation if we are to extend this approach to a realistically fine partition of a complex pose space. Consider an image of resolution $640 \times 480$ and a single scale range for the head. Obtaining an accuracy in the locations of the head and the belly of five pixels requires more than $7 \times 10^6$ pose cells. Investing computation *uniformly* among cells is therefore hopeless, and argues for an adaptive strategy able to distribute computation in a highly special and uneven manner.

The conclusions drawn can be summarized in two key points:

1. **The need for data-aggregation**: *Dealing with a rich pose by training specialized predictors from constrained sub-populations is not feasible, both in terms of offline computation and sample size requirements. Aggregation of data using stationary features appears to be a sound strategy to overcome the sample size dilemma as it transfers the burden of learning to the design of the features.*

2. **The need for adaptive search**: *If fragmentation can be avoided and a single classifier built from all the data and analytically transformed into dedicated classifiers, the computation necessary to cover a partition of a pose space of reasonable accuracy is not realistic if the effort is uniformly distributed over cells.*

As indicated, stationary features provide a coherent strategy for dealing with data-aggregation but do not resolve the computational dilemma resulting from investigating many possible poses during scene processing. Hierarchical representations largely do, as explained in the next section.

# 6 Folded Hierarchies

We have proposed above to normalize the samples through a family of pose-indexed features in order to avoid fragmentation of the data. Since one classifier can be built for any partition of the pose space, and no longer for every cell of such a partition, neither the cost of learning nor the required size of the training set grows linearly with the number $K$ of pose cells in the partition. However, one main drawback remains: We must still have to visit all the pose cells online, which makes the scene processing cost itself linear in $K$.

A natural strategy to address computational cost is an hierarchical search strategy based upon a recursive partitioning of $\mathcal{Y}$. As in previous work ([5, 7]), there is a succession of nested partitions of increasing resolution and a binary classifier assigned to each cell. Given such a hierarchy, the detection process is adaptive: a classifier is evaluated for a certain pose cell only if all the classifiers for its ancestor cells have been evaluated and responded positively.

**Note:** This is *not* a decision tree, both in terms of representation and processing. The hierarchy recursively partitions the space of hidden variables not the feature space, and the edges from a node to its children do not represent the possible values of a node classifier. Moreover, during processing, a data point may traverse many branches at once and reach no leaves or many.

The crucial difference with previous work is that, using stationary features, only one classifier must be trained for each level, not one classifier for each cell. In essence, the hierarchy is "folded" (like a fan) for training: The entire learning strategy described in § 3 is repeated for each level in the hierarchy. This is quite straightforward and only summarized below.

Consider a sequence of partitions of $\mathcal{Y}$

$$\left\{\mathcal{Y}_1^{(d)}, \ldots, \mathcal{Y}_{K_d}^{(d)}\right\}, \quad 1 \leq d \leq D,$$

for which any cell $\mathcal{Y}_k^d$ for $k = 1, \ldots, K_{d+1}$, is a (disjoint) union of cells at the next level $d+1$. Consequently, we can identify every $\mathcal{Y}_k^{(d)}$ with the node of a multi-rooted tree: A leaf node for $d = D$ and an internal node otherwise. A three-level hierarchy is shown in Figure 9.

Given such a pose hierarchy, we can construct a scene parsing algorithm aimed at detecting all instances of objects at a pose resolution corresponding to the finest partition. Again, the processing strategy is now well-known. (A slight modification allows one to build an ROC curve by aggregating the responses along each branch; see § 7.) This algorithm has the desirable property of concentrating computation on the ambiguous and challenging pose-image pairs.

Let $Y_k^{(d)}$ denote a Boolean random variable indicating whether or not there is a target in $I$ with pose in $\mathcal{Y}_k^{(d)}$ and let $\mathbf{X}^{(d)}$ denote a pose-indexed feature vector adapted to the partition $\{\mathcal{Y}_1^{(d)}, \ldots, \mathcal{Y}_{K_d}^{(d)}\}$. For each level $d$, we train a classifier $g^{(d)}$ exactly as described in § 3.3, and define a predictor of $Y_k^{(d)}$ by

$$f_k^{(d)}(I) = g^{(d)}\left(\mathbf{X}_k^{(d)}(I)\right).$$

The threshold in $g^{(d)}$ is determined by the requirement that each $f_k^{(d)}$ has a very low false-negative rate:

$$P\left(f_k^{(d)} = 0 \,\middle|\, Y_k^{(d)} = 1\right) \simeq 0, \quad \forall k, d$$

The hierarchy is "unfolded" for testing and the predictors are evaluated in an adaptive way by visiting the nodes (cells) according to breadth-first "coarse-to-fine" search. A classifier is evaluated if and only if all its ancestors along the branch up to its root have been evaluated and returned a positive response. In particular, once a classifier at a node responds negatively, none of the descendant classifiers are ever evaluated. The result of the detection process is the list of leaves which are reached and respond positively. In this way, pose cells corresponding to obvious non-target regions such as flat areas are discarded early in the search and the computation is invested the ambiguous areas, e.g., parts of images with "cat-like" shape or texture.

# 7 Main Experiment: Full Scene Parsing

We now present our results on parsing full, cluttered scenes in order to detect instances of cats. Since we process the scene at only one scale range, the pose is defined by the head and the belly locations. Hence, the search through scales is handled externally and not integrated into the hierarchy as in [5] and elsewhere, where a much larger scale range (e.g., of the form $(s, 2s)$) is accommodated by adding

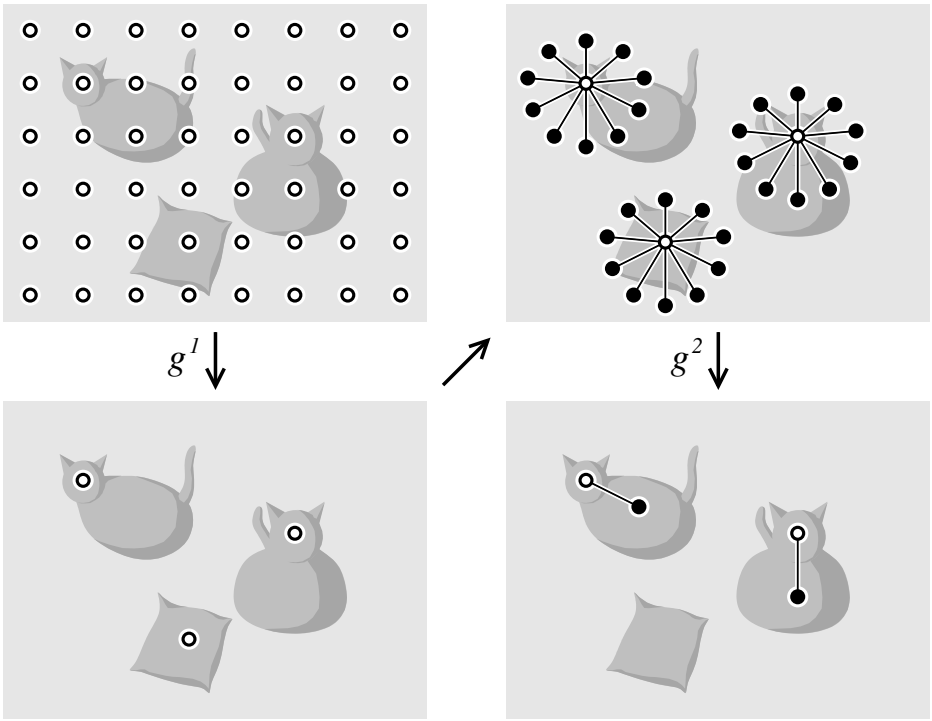



Figure 10: Parsing a scene with a two-level hierarchy to find cats: First, a classifier $g^{(1)}$ is evaluated over a sublattice of possible head locations and all alarms above a very low threshold are retained. Then a classifier $g^{(2)}$ is evaluated for each pair of head-belly locations on a sublattice consistent with the retained head alarms and with observed statistics about joint head-belly locations. For clarity, the depicted discretization of the pose space is idealized, and coarser than in the actual experiments.

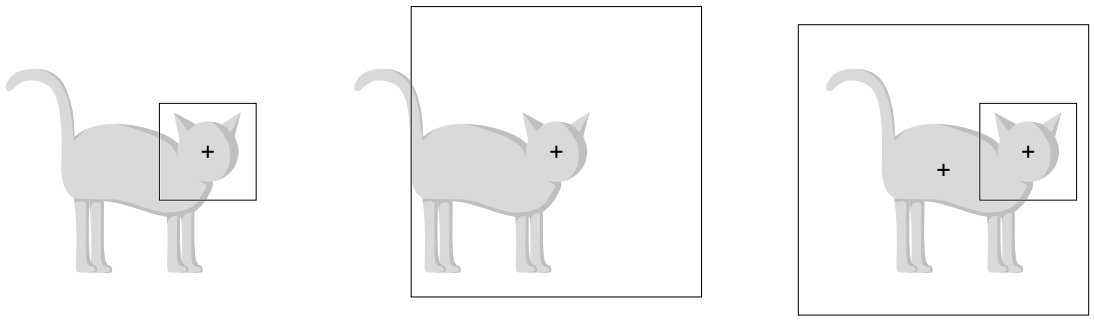
levels associated with a recursive partitioning of the base range. In these approaches, a base detector is trained for the base scale range which represents the smallest anticipated (or "detectable") size of the objects. Then, in order to find instances at larger scales, the image is downsampled several times and processed with the base detector.

We will consider two levels of description: localizing the head alone and localizing both the head and the belly, our principal objective. In the former case we shall consider two strategies: building a head detector on a restricted field of view, and building a head detector which attempts to use the body data as well, either i) in the spirit of separate "parts" or ii) genuinely in combination by using a two-level folded hierarchy and stationary features. Detecting both locations can be done either by i) or ii).

## 7.1 Hierarchy of poses

We consider pairs of locations $(h, b)$ consistent with the relative locations seen on the training set. For instance, this discards pairs $(h, b)$ which are very far apart (relative to the head size). However, these pairs may be very close together, for example when the body is behind the head, or very far apart, for example when the cat is stretched out. Hence the full pose space is $\mathcal{Y} \subset \mathcal{Z}^2$. We propose a hierarchy with $D = 2$ levels in order to concentrate on folded learning with stationary features. The precise specification of allowable pairs will be made in defining the pose partition for second level of the hierarchy.

The first level $\{\mathcal{Y}_1^{(1)}, \ldots, \mathcal{Y}_{K_d}^{(1)}\}$ is based on partitioning the head locations into disjoint $5 \times 5$ squares. Each cell $\mathcal{Y}_k^{(1)}, k = 1, \ldots, K_1$ in the first level contains all poses $(h, b)$ with $h$ restricted to a



$\mathbf{H}^{narr}$ detector $\mathbf{H}^{wide}$ detector $\mathbf{H}|\mathbf{B}$ detector

Figure 11: The three "standard" detectors, in particular not based on stationary features

fixed square, hence is of the form $([5i, 5(i+1)) \times [5j, 5(j+1))) \times \mathcal{B}(i,j)$, where $\mathcal{B}(i,j)$ is the set of belly locations consistent with a head location in $([5i, 5(i+1)) \times [5j, 5(j+1)))$.

A cell $\mathcal{Y}_k^{(2)}, k = 1, \ldots, K_2$ in the second level contains all poses $(h, b)$ with $h$ restricted to one $5 \times 5$ square and $b$ restricted to another $5 \times 5$ square, hence is of the form $([5i, 5(i+1)) \times [5j, 5(j+1))) \times ([5i', 5(i'+1)) \times [5j', 5(j'+1)))$.

The top-left illustration in Figure 10 depicts one open circle for each cell of the first level of the hierarchy, and the top-right illustration depicts each $\mathcal{B}$ as a circle of black dots connected to an open circle for every first-level cell kept alive during processing the first level. More specifically, as shown in Figure 10, the algorithmic process corresponding to this two-level hierarchy is as follows:

1. The first stage loops over a sublattice of possible head locations in the scene, evaluates the response of the appropriate first-level classifier and retains all alarms using a very low (i.e., conservative) threshold determined by keeping 99% of cat heads on a validation set.

2. The second stage visits each location tagged by the first stage, scans a sublattice of all "consistent" belly locations (all those actually observed on training images) and evaluates an appropriate second-level classifier for every such candidate pair of locations.

## 7.2 Detectors

Detecting cat heads is similar to the well-studied problem of detecting frontal views of human faces. As stated earlier, if the pose reduces to a single position, data-aggregation is straightforward by translating either whole images or features. Still, it is a logical first step in trying to find cats since the head is clearly the most stable landmark and the part of the cat with the least variation, assuming of course the head is visible, which is the case with our data (for the same reason that family photographs display the faces of people). Moreover, comparing the performance of varying strategies (field of view, "checking" for the belly separately, demanding "consistency", etc.) provides some insight on the nature of the problem and serves as a simple way of demonstrating the power of the base feature set and the asymmetrical weighting by sampling. It does not, however, expose the full strength of the folded hierarchy; for that we need to address the harder task of accurately estimating $(h, b)$ for the visible cats, our core objective, and for which we will compare our pose-indexed method with a more standard parts-based detector.

We consider the following four detectors:

- $\mathbf{H}^{narr}$ is trained by boosting 500 trees of maximum depth five based on features constrained to a "narrow" $80 \times 80$ field of view centered on the head. At the reference range of scales, this basically captures the head alone.

- **H**$^{wide}$ is similar to H$^{narr}$, but the field of view is of size of 240 × 240, again centered on the head, enabling the features to probe the body shape.

- **H|B** is a standard parts detector, implemented adaptively. The "|" between H and B indicates that the two part detectors are trained separately. The first detector is H$^{narr}$, but with only 250 trees. The second is similar to H$^{wide}$, also combines 250 trees, but training is registered to $b$ instead of $h$. Adaptive scene processing is described below.

- **HB** is the hierarchical detector described in § 4 based on the two-level hierarchy and folded learning. The difference with H|B is that HB uses stationary features in the second level in order to take into account the position of the head in searching for the belly; the features are parametrized by both locations.

For both H|B and HB, all candidate head locations (5 × 5 squares) are rejected for which the H$^{narr}$ does not exceed a very conservative threshold, estimated on 10% of the positive samples. Then, every surviving candidate is scored by the sum of the response of the head detector and the maximum response of the belly detector over all admissible (basically, nearby) belly locations. Both levels use 250 trees.

## 7.3 Results

All the settings (number of training samples, number of features used for optimization, etc.) are kept identical in all the experiments.

### 7.3.1 Head detection

For this task, let us call an "alarm" every 5 × 5 window for H$^{narr}$ and H$^{wide}$, and every surviving candidate 5 × 5 window for the two-stage detectors H|B and HB. The "score" of an alarm is just the response of the classifier for H$^{narr}$ and H$^{wide}$ and the sum of the two responses (see above) for the H|B and HB detectors.

An alarm is labeled a true positive if and only if it falls within 25 pixels of a true head location. The alarms are post-processed with a very simple clustering consistent with this designation. All alarms are ranked according to their score, visited in order, and for each alarm we discard all other alarms (necessarily with a lower score) falling within a radius of 25 pixels.

Perhaps the first observation about the error rates in Table 3 is that all the methods are doing something reasonable on a very difficult problem. The HB detector does provide an improvement over both the H$^{wide}$ and H|B detectors, and a slight improvement over the H$^{narr}$ detector. Of course, the H|B and HB detectors both provides additional information, namely an estimate of the full pose $(h, b)$

As shown in the table, the performance of H$^{narr}$, which just looks at the head, cannot be improved simply by increasing the field of view. Indeed, H$^{wide}$ does worse: widening the field increases the number of features to choose from and training with boosting is actually less efficient, at least if the total number of trees is held fixed. More importantly, combining features sampled in a larger area cannot account for the complex statistical coupling between different parts of the cat.

### 7.3.2 Head-belly detection

An $(h, b)$ detection is labeled a true positive if the estimated head location is within 25 pixels of the true head location *and* the estimated belly location is within 50 pixels of the true belly location. As in the head-detection task, the alarms are post-processed with a crude clustering: We visit the alarms in the order of the response of the detector in the second level, and for each alarm we remove all others whose head locations is within 25 pixels of the one visited.

The error rates in Table 4 and Figure 12 demonstrate the power of conditioning on the full pose. Using stationary features to build classifiers dedicated to fine cells, allows the search for one part to be informed by the location of the other, and allows for consistency checks. This is more discriminating

| TP | $\mathbf{H}^{narr}$ | $\mathbf{H}^{wide}$ | H\|B | HB |
|---|---|---|---|---|
| 0.95 | 8.41 (1.42) | 11.15 (2.08) | 12.31 (3.34) | 6.65 (1.95) |
| 0.90 | 3.22 (0.70) | 4.50 (0.53) | 6.39 (0.80) | 3.24 (0.79) |
| 0.80 | 0.92 (0.14) | 1.46 (0.25) | 2.00 (0.37) | 0.84 (0.12) |
| 0.70 | 0.35 (0.06) | 0.61 (0.10) | 0.74 (0.11) | 0.34 (0.04) |
| 0.60 | 0.15 (0.04) | 0.27 (0.04) | 0.30 (0.05) | 0.15 (0.04) |
| 0.50 | 0.10 (0.03) | 0.12 (0.03) | 0.10 (0.03) | 0.08 (0.03) |

Table 3: Average number of false alarms per scene of size $640 \times 480$ vs. true positive rate on the head detection task with clustering. The figures between parenthesis are the empirical standard deviation of the estimates.

| TP | H\|B Detector | HB Detector |
|---|---|---|
| 0.85 | n.a. | 11.29 (3.61) |
| 0.80 | 17.23 (3.87) | 5.07 (1.08) |
| 0.70 | 11.40 (2.89) | 1.88 (0.32) |
| 0.60 | 7.80 (1.98) | 0.95 (0.22) |
| 0.50 | 5.41 (1.62) | 0.53 (0.13) |

Table 4: Average number of false alarms per scene of size $640 \times 480$ vs. true positive rate on the $(h, b)$ detection task with head-based clustering. The figures between parenthesis are the empirical standard deviation of the estimates.

than checking for individual parts separately. Indeed, the error rates are cut be a factor of roughly three at very high true-positive rates and by almost a factor of ten at lower true-positive rates. Since the detection criterion requires both the estimated head and the belly locations to be close to the true ones, the presence of a strong false-alarm may kill a true alarm at any threshold during the clustering, accounting for the maximum true positive rate of 85% for the HB detector, and 80% for the H|B detector.

Specific examples how features selected in the second-level of the HB classifier exploit the full pose can be seen in Figure 13, bottom diagrams.These features allow HB to check for highly discriminating properties of the data, such as the continuity of appearance between the head and the belly, or discontinuities in the direction orthogonal to the head-belly axis.

More then two-thirds of the false positives are located on or very near cats; see the top images of Figure 15. Such false positives are exceedingly difficult to filter out. For instance, a false head detection lying around or on the body will be supported by the second-level classifier because the location of the true belly will usually be visited.

# 8 Conclusion

We have presented a novel detection algorithm for objects with a complex pose. Our main contribution is the idea of stationary, pose-indexed features, a variation on deformable templates without whole image transforms. This makes it possible to train pose-specific classifiers without clustering the data, and hence without reducing the number of training examples. Moreover, combining simultaneous training with a sequential exploration of the pose space overcomes the main drawback of previous coarse-to-fine strategies, especially for going beyond scale and translation. Unlike in earlier variations, graded, tree-structured representations can now be learned efficiently because there is only one classifier to train per level of the hierarchy rather than one per node.

We have illustrated these stationary features by detecting cats in cluttered still images. We chose boosting with edge and intensity counts, but any base learning algorithm and any flexible base feature

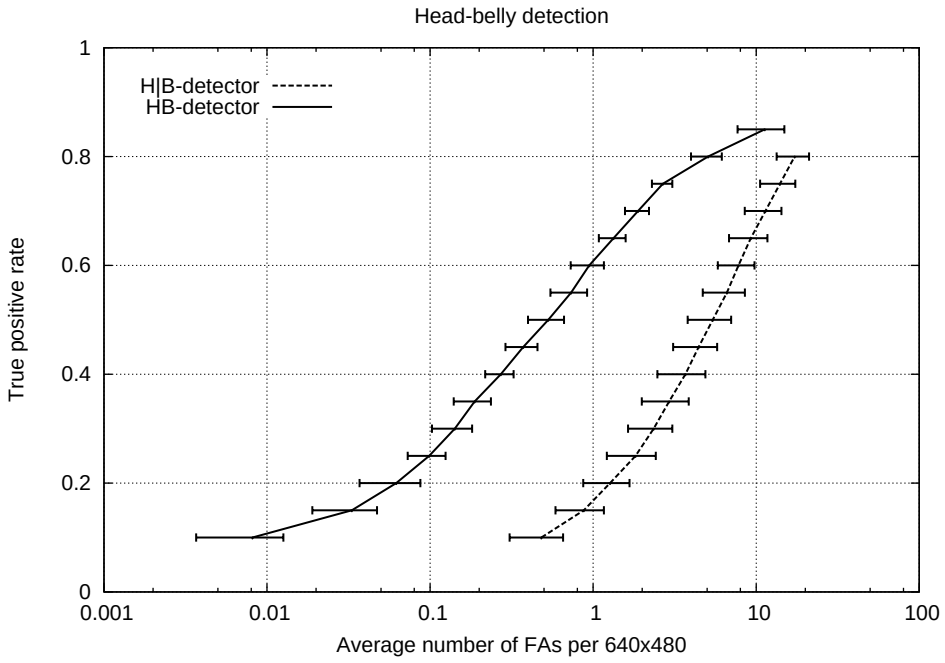



Figure 12: ROC curves for the head-belly detection.

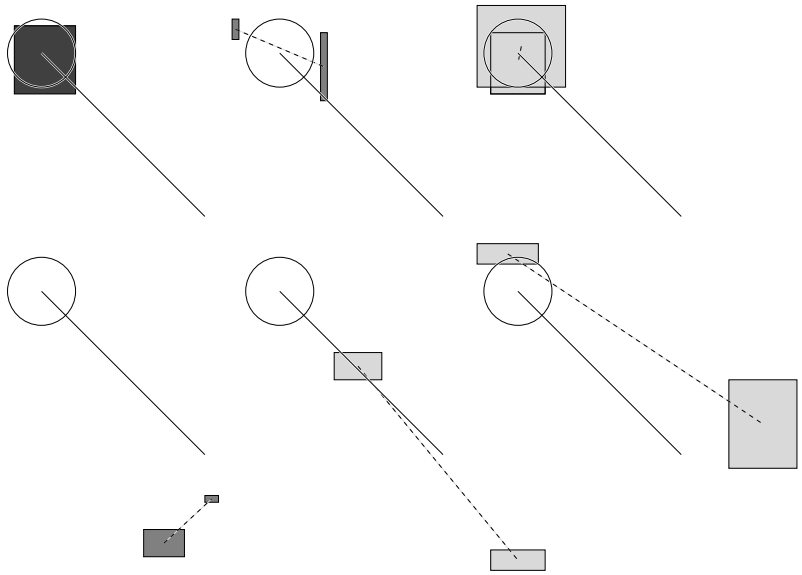


Figure 13: Examples of stationary features selected at the root of trees for the classifiers $g^{(1)}$ (top row) and $g^{(2)}$ (bottom row). The circle shows the scale and location of the cat head, and the line segment points to the estimated belly location. A dark gray rectangle indicates the window of an edge-proportion feature, a pair of medium gray rectangles the two windows in a comparison of gray-scale histograms, and a pair of light gray rectangles the two windows in a comparison of edge orientation histograms.

set could be used. The resulting algorithm is a two-stage process, first visiting potential head locations alone and then examining additional aspects of the pose consistent with and informed by candidate head locations.

In principle, our approach can deal with very complex detection problems in which multiple objects of interest are parametrized by a rich hidden pose. However, two basic limitations must first be addressed. The first is the design of adequate stationary features. Whereas difficult, this is far simpler than the search for full geometric invariants. Since the hidden state is explicitly examined in traversing the hierarchy, there is no need to integrate over all possible values of the hidden quantities. The second difficulty is the labeling of a large training set with rich ground truth. One way to tackle this problem is by exploiting other information available during training, for instance temporal consistency if there are motion data.

# 9 Acknowledgment

This work was partially supported by the National Science Foundation under grants NSF 0427223 and NSF 0625687, and by the Office of Naval Research under contract N00014-07-1-1002.

Figure 14: Detection results with stationary features and a folded two-level hierarchy on scenes picked uniformly at random in the test set. Circles and discs show respectively the estimated head and belly locations.

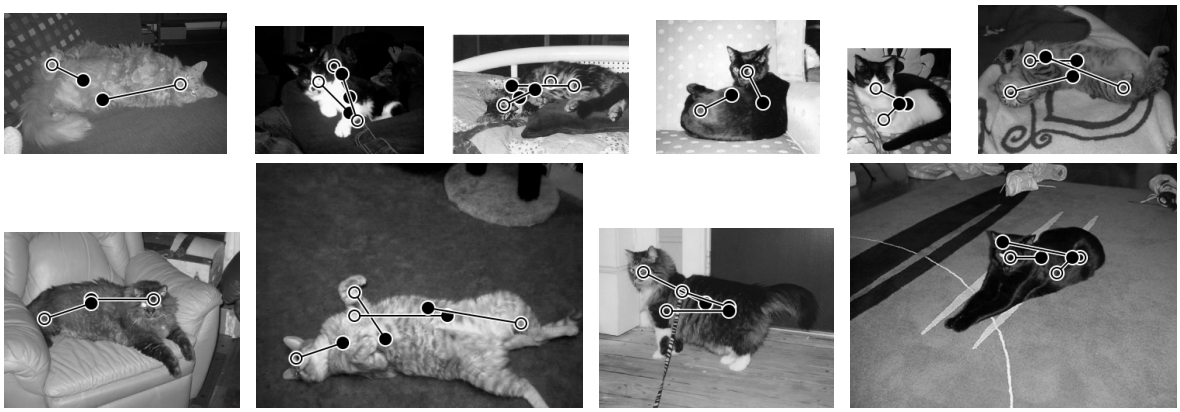


Figure 15: Examples of selected false alarms for a true-positive rate of 90%; alarms are clustered by non-maximum suppression. Circles and dark disks show respectively the estimated head and belly locations.

## References


[1] Y. Amit, D. Geman, and B. Jedynak. Efficient focusing and face detection. In *Face Recognition: From Theory to Applications*. Springer Verlag, 1998. 5, 11

[2] Y. Amit and A. Trouve. POP: Patchwork of parts models for object recognition. *International Journal of Computer Vision*, 75(2):267–282, November 2007. 5

[3] D. J. Crandall and D. P. Huttenlocher. Weakly supervised learning of part-based spatial models for visual object recognition. In *European Conference on Computer Vision*, pages 16–29, 2006. 5

[4] X. Fan. *Learning a hierarchy of classifiers for multi-class shape detection*. PhD thesis, Johns Hopkins University, 2006. 12

[5] F. Fleuret and D. Geman. Coarse-to-fine face detection. *International Journal of Computer Vision*, 41(1/2):85–107, 2001. 3, 5, 19

[6] Y. Freund and R. E. Schapire. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5):771–780, 1999. 11

[7] S. Gangaputra and D. Geman. A design principle for coarse-to-fine classification. In *Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1877–1884, 2006. 5, 19

[8] D. Gavrila. Multi-frame hierarchical template matching using distance transforms. In *International Conference on Pattern Recognition*, 1998. 5

[9] S. Geman, K. Manbeck, and E. McClure. Coarse-to-fine search and rank-sum statistics in object recognition. Technical report, Brown University, 1995. 5

[10] S. Geman, D. F. Potter, and Z. Chi. Composition systems. *Quarterly of Applied Mathematics*, LX:707–736, 2002. 5

[11] Y. LeCun, F. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Conference on Computer Vision and Pattern Recognition*. IEEE Press, 2004. 3, 5

[12] F. Li, R. Fergus, and P. Perona. A Bayesian approach to unsupervised one-shot learning of object categories. In *International Conference on Computer Vision*, volume 2, page 1134, 2003. 5

[13] J. Mundy and A. Zisserman, editors. *Geometric Invariance in Computer Vision*. MIT Press, 1992. 4

[14] B. Ommer, M. Sauter, and J. M. Buhmann. Learning top-down grouping of compositional hierarchies for recognition. In *Conference on Computer Vision and Pattern Recognition*, 2006. 5

[15] C. Papageorgiou and T. Poggio. A trainable system for object detection. *International Journal of Computer Vision*, 38(1):15–33, June 2000. 3

[16] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–28, 1998. 3

[17] H. Schneiderman and T. Kanade. Object detection using the statistics of parts. *International Journal of Computer Vision*, 56(3):151–177, 2004. 5

[18] P. Simard, L. Bottou, P. Haffner, and Y. LeCun. Boxlets: a fast convolution algorithm for neural networks and signal processing. In *Neural Information Processing Systems*, volume 11, 1999. 12

[19] B. Stenger, A. Thayananthan, P. Torr, and R. Cipolla. Model-based hand tracking using a hierarchical bayesian filter. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1372–1384, 2006. 3, 5

[20] P. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004. 3, 5, 14

[21] J. Wu, S. C. Brubaker, M. D. Mullin, and J. M. Rehg. Fast asymmetric learning for cascade face detection. Technical Report GIT-GVU-05-27, Georgia Institute of Technology, 2005. 3, 5, 14

[22] S. Zhu and D. Mumford. *A stochastic grammar of images*, volume 2 of *Foundations and Trends in Computer Graphics and Vision*, pages 259–362. Now Publishers, 2006. 5