Fast Feature Pyramids for Object Detection

Piotr Dollár, Ron Appel, Serge Belongie, and Pietro Perona

Abstract—Multi-resolution image features may be approximated via extrapolation from nearby scales, rather than being computed explicitly. This fundamental insight allows us to design object detection algorithms that are as accurate, and considerably faster, than the state-of-the-art. The computational bottleneck of many modern detectors is the computation of features at every scale of a finely-sampled image pyramid. Our key insight is that one may compute finely sampled feature pyramids at a fraction of the cost, without sacrificing performance: for a broad family of features we find that features computed at octave-spaced scale intervals are sufficient to approximate features on a finely-sampled pyramid. Extrapolation is inexpensive as compared to direct feature computation. As a result, our approximation yields considerable speedups with negligible loss in detection accuracy. We modify three diverse visual recognition systems to use fast feature pyramids and show results on both pedestrian detection (measured on the Caltech, INRIA, TUD-Brussels and ETH datasets) and general object detection (measured on the PASCAL VOC). The approach is general and is widely applicable to vision algorithms requiring fine-grained multi-scale analysis. Our approximation is valid for images with broad spectra (most natural images) and fails for images with narrow band-pass spectra (e.g. periodic textures).

Index Terms—visual features, object detection, image pyramids, pedestrian detection, natural image statistics, real-time systems

1 INTRODUCTION

Multi-resolution multi-orientation decompositions are one of the foundational techniques of image analysis. The idea of analyzing image structure separately at every scale and orientation originated from a number of sources: measurements of the physiology of mammalian visual systems [1], [2], [3], principled reasoning about the statistics and coding of visual information [4], [5], [6], [7] (Gabor, DOGs, and jets), harmonic analysis [8], [9] (wavelets), and signal processing [9], [10] (multirate filtering). Such representations have proven effective for visual processing tasks such as denoising [11], image enhancement [12], texture analysis [13], correspondence [14], motion flow [15], [16], attention [17], boundary detection [18] and recognition [19], [20], [21].

It has become clear that such representations are best at extracting visual information when they are over-complete, i.e. when one oversamples scale, orientation and other kernel properties. This was suggested by the architecture of the primate visual system [22], where striate cortical cells (roughly equivalent to a wavelet expansion of an image) outnumber retinal ganglion cells (a representation close to image pixels) by a factor ranging from $10^2$ to $10^4$. Empirical studies in computer vision provide increasing evidence in favor of overcomplete representations [23], [24], [25], [21], [26]. Most likely the robustness of these representations with respect to changes in viewpoint, lighting, and image deformations is a contributing factor to their superior performance.

To understand the value of richer representations, it is instructive to examine the reasons behind the breathtaking progress in visual category detection during the past ten years. Take, for instance, pedestrian detection. Since the groundbreaking work of Viola and Jones (VJ) [27], [28], false positive rates have decreased two orders of magnitude. At 80% detection rate on the INRIA pedestrian dataset [21], VJ outputs over 10 false positives per image (FPPI), HOG [21] outputs $\sim$1 FPPI, and more recent methods [29], [30] output well under 0.1 FPPI (data from [31], [32]). In comparing the different detection schemes one notices the representations at the front end are progressively enriched (e.g. more channels, finer scale sampling, enhanced normalization schemes); this has helped fuel the dramatic improvements in detection accuracy witnessed over the course of the last decade.

Unfortunately, improved detection accuracy has been accompanied by increased computational costs. The VJ detector ran at $\sim$15 frames per second (fps) over a decade ago, on the other hand, most recent detectors require multiple seconds to process a single image as they compute richer image representations [31]. This has practical importance: in many applications of visual recognition, such as robotics, human computer interaction, automotive safety, and mobile devices, fast detection rates and low computational requirements are of the essence.

Thus, while increasing the redundancy of the representation offers improved detection and false-alarm rates, it is paid for by increased computational costs. Is this a necessary trade-off? In this work we offer the hoped-for but surprising answer: no.

We demonstrate how to compute richer representations without paying a large computational price. How is this possible? The key insight is that natural images have fractal statistics [7], [33], [34] that we can exploit to reliably predict image structure across scales. Our anal-
ysis and experiments show that this makes it possible to inexpensively estimate features at a dense set of scales by extrapolating computations carried out expensively, but infrequently, at a coarsely sampled set of scales.

Our insight leads to considerably decreased run-times for state-of-the-art object detectors that rely on rich representations, including histograms of gradients [21], with negligible impact on their detection rates. We demonstrate the effectiveness of our proposed fast feature pyramids with three distinct detection frameworks including integral channel features [29], aggregate channel features (a novel variant of integral channel features), and deformable part models [35]. We show results for both pedestrian detection (measured on the Caltech [31], INRIA [21], TUD-Brussels [36] and ETH [37] datasets) and general object detection (measured on the PASCAL VOC [38]). Demonstrated speedups are significant and impact on accuracy is relatively minor.

Building on our work on fast feature pyramids (first presented in [39]), a number of systems show state-of-the-art accuracy while running at frame rate on $640 \times 480$ images. Aggregate channel features, described in this paper, operate at over 30 fps while achieving top results on pedestrian detection. Crosstalk cascades [40] use fast feature pyramids and couple detector evaluations of nearby windows to achieve speeds of 35-65 fps. Benenson et al. [30] implemented fast feature pyramids on a GPU, and with additional innovations achieved detection rates of over 100 fps. In this work we examine and analyze feature scaling and its effect on object detection in far more detail than in our previous work [39].

The rest of this paper is organized as follows. We review related work in §2. In §3 we show that it is possible to create high fidelity approximations of multiscale gradient histograms using gradients computed at a single scale. In §4 we generalize this finding to a broad family of feature types. We describe our efficient scheme for computing finely sampled feature pyramids in §5. In §6 we show applications of fast feature pyramids to object detection, resulting in considerable speedups with minor loss in accuracy. We conclude in §7.

2 Related Work

Significant research has been devoted to scale space theory [41], including real time implementations of octave and half-octave image pyramids [42], [43]. Sparse image pyramids often suffice for certain approximations, e.g. [42] shows how to recover a disk’s characteristic scale using half-octave pyramids. Although only loosely related, these ideas provide the intuition that finely sampled feature pyramids can perhaps be approximated.

Fast object detection has been of considerable interest in the community. Notable recent efforts for increasing detection speed include work by Felzenszwalb et al. [44] and Pedersoli et al. [45] on cascaded and coarse-to-fine deformable part models, respectively, Lampert et al.’s [46] application of branch and bound search for detection, and Dollár et al.’s work on crosstalk cascades [40]. Cascades [27], [47], [48], [49], [50], coarse-to-fine search [51], distance transforms [52], etc., all focus on optimizing classification speed given precomputed image features. Our work focuses on fast feature pyramid construction and is thus complementary to such approaches.

An effective framework for object detection is the sliding window paradigm [53], [27]. Top performing methods on pedestrian detection [31] and the PASCAL VOC [38] are based on sliding windows over multiscale feature pyramids [21], [29], [35]; fast feature pyramids are well suited for such sliding window detectors. Alternative detection paradigms have been proposed [54], [55], [56], [57], [58], [59]. Although a full review is outside the scope of this work, the approximations we propose could potentially be applicable to such schemes as well.

As mentioned, a number of state-of-the-art detectors have recently been introduced that exploit our fast feature pyramid construction to operate at frame rate including [40] and [30]. Alternatively, parallel implementation using GPUs [60], [61], [62] can achieve fast detection while using rich representations but at the cost of added complexity and hardware requirements. Zhu et al. [63] proposed fast computation of gradient histograms using integral histograms [64]; the proposed system was real time for single-scale detection only. In scenarios such as automotive applications, real time systems have also been demonstrated [65], [66]. The insights outlined in this paper allow for real time multiscale detection in general, unconstrained settings.

3 Multiscale Gradient Histograms

We begin by exploring a simple question: given image gradients computed at one scale, is it possible to approximate gradient histograms at a nearby scale solely from the computed gradients? If so, then we can avoid computing gradients over a finely sampled image pyramid. Intuitively, one would expect this to be possible, as significant image structure is preserved when an image is resampled. We begin with an in-depth look at a simple form of gradient histograms and develop a more general theory in §4.

A gradient histogram measures the distribution of the gradient angles within an image. Let $I(x, y)$ denote an $m \times n$ discrete signal, and $\partial I/\partial x$ and $\partial I/\partial y$ denote the discrete derivatives of $I$ (typically 1D centered first differences are used). Gradient magnitude and orientation are defined by: $M(i, j)^2 = \partial_x I(i, j)^2 + \partial_y I(i, j)^2$ and $O(i, j) = \arctan \left( \frac{\partial_y I(i, j)}{\partial_x I(i, j)} \right)$. To compute the gradient histogram of an image, each pixel casts a vote, weighted by its gradient magnitude, for the bin corresponding to its gradient orientation. After the orientation $O$ is quantized into $Q$ bins so that $O(i, j) \in \{1, Q\}$, the $q^{th}$ bin of the histogram is defined by: $h_q = \sum_{i, j} M(i, j) \mathbf{1}[O(i, j) = q]$, where $\mathbf{1}$ is the indicator function. In the following everything that holds for global histograms also applies to local histograms (defined identically except for the range of the indices $i$ and $j$).
3.1 Gradient Histograms in Upsampled Images

Intuitively the information content of an upscaled image is similar to that of the original, lower-resolution image (upsampling does not create new structure). Assume $I$ is a continuous signal, and let $I'$ denote $I$ upscaled by a factor of $k$: $I'(x, y) = I(x/k, y/k)$. Using the definition of a derivative, one can show that $\frac{\partial^2 I'}{\partial x^2}(i,j) = \frac{1}{k^2} \frac{\partial^2 I}{\partial x^2}(i/k, j/k)$, and likewise for $\frac{\partial^2 I'}{\partial y^2}$, which simply states the intuitive fact that the rate of change in the upscaled image is $k$ times slower the rate of change in the original image. While not exact, the above also holds approximately for interpolated discrete signals.

Let $M'(i,j) \approx \frac{1}{M} M([i/k],[j/k])$ denote the gradient magnitude in an upscaled discrete image. Then:

$$\sum_{i=1}^{kn} \sum_{j=1}^{km} M'(i,j) \approx \sum_{i=1}^{kn} \sum_{j=1}^{km} \frac{1}{M} M([i/k],[j/k])$$

$$= k^2 \sum_{i=1}^{n} \sum_{j=1}^{m} M(i,j) = k \sum_{i=1}^{n} \sum_{j=1}^{m} M(i,j)$$

Thus, the sum of gradient magnitudes in the original and upscaled image should be related by about a factor of $k$. Angles should also be mostly preserved since $\frac{\partial^2 I'}{\partial x^2}(i,j) / \frac{\partial^2 I'}{\partial y^2}(i,j) \approx \frac{\partial I}{\partial x}(i/k, j/k) / \frac{\partial I}{\partial y}(i/k, j/k)$. Therefore, according to the definition of gradient histograms, we expect the relationship between $h_q$ (computed over $I$) and $h'_q$ (computed over $I'$) to be: $h'_q \approx k h_q$. This allows us to approximate gradient histograms in an upscaled image using gradients computed at the original scale.

Experiments: One may verify experimentally that in images of natural scenes, upscaled using bilinear interpolation, the approximation $h'_q \approx k h_q$ is reasonable. We use two sets of images for these experiments, one class specific and one class independent. First, we use the 1237 cropped pedestrian images from the INRIA pedestrians training dataset [21]. Each image is 128 × 64 and contained a pedestrian approximately 96 pixels tall. The second image set contains 128×64 windows cropped at random positions from the 1218 images in the INRIA negative training set. We sample 5000 windows but exclude nearly uniform windows, i.e. those with average gradient magnitude under 0.01, resulting in 4280 images. We refer to the two sets as ‘pedestrian images’ and ‘natural images’, although the latter is biased toward scenes that may (but do not) contain pedestrians.

In order to measure the fidelity of this approximation, we define the ratio $r_q = h'_q / h_q$ and quantize orientation into $Q = 6$ bins. Figure 1(a) shows the distribution of $r_q$ for one bin on the 1237 pedestrian and 4280 natural images given an upsampling of $k = 2$ (results for other bins were similar). In both cases the mean is $\mu \approx 2$, as expected, and the variance is relatively small, meaning the approximation is unbiased and reasonable.

Thus, although individual gradients may change, gradient histograms in an upscaled and original image will be related by a multiplicative constant roughly equal to the scale change between them. We examine gradient histograms in downsampled images next.

![Diagram](image_url)
The information content of an upsampled image is roughly the same as that of the original image, information is typically lost during downsampling. However, we find that the information loss is consistent and the resulting approximation takes on a similarly simple form.

If \( I \) contains little high frequency energy, then the approximation \( h_q' \approx kh_q \) derived in §3.1 should apply. In general, however, downsampling results in loss of high frequency content which can lead to measured gradients undershooting the extrapolated gradients. Let \( I' \) now denote \( I \) downsampled by a factor of \( k \). We expect that \( h_q \) (computed over \( I \)) and \( h_q' \) (computed over \( I' \)) will satisfy \( h_q' \leq h_q/k \). The question we seek to answer here is whether the information loss is consistent.

Experiments: As before, define \( r_q = h_q'/h_q \). In Figure 1(b) we show the distribution of \( r_q \) for a single bin on the pedestrian and natural images given a downsampling factor of \( k = 2 \). Observe that the information loss is consistent: \( r_q \) is normally distributed around \( \mu \approx .34 < .5 \) for natural images (and similarly \( \mu \approx .33 \) for pedestrians). This implies that \( h_q' \approx \mu h_q \) could serve as a reasonable approximation for gradient histograms in images downsampled by \( k = 2 \).

In other words, similarly to upsampling, gradient histograms computed over original and half resolution images tend to differ by a multiplicative constant (although the constant is not the inverse of the sampling factor). In Figure 2 we show the quality of the above approximations on example images. The agreement between predictions and observations is accurate for typical images (but fails for images with atypical statistics).

3.2 Gradient Histograms in Downsampled Images

While the information content of an upsampled image is roughly the same as that of the original image, information is typically lost during downsampling. However, we find that the information loss is consistent and the resulting approximation takes on a similarly simple form.

If \( I \) contains little high frequency energy, then the approximation \( h_q' \approx kh_q \) derived in §3.1 should apply. In general, however, downsampling results in loss of high frequency content which can lead to measured gradients undershooting the extrapolated gradients. Let \( I' \) now denote \( I \) downsampled by a factor of \( k \). We expect that \( h_q \) (computed over \( I \)) and \( h_q' \) (computed over \( I' \)) will satisfy \( h_q' \leq h_q/k \). The question we seek to answer here is whether the information loss is consistent.

Experiments: As before, define \( r_q = h_q'/h_q \). In Figure 1(b) we show the distribution of \( r_q \) for a single bin on the pedestrian and natural images given a downsampling factor of \( k = 2 \). Observe that the information loss is consistent: \( r_q \) is normally distributed around \( \mu \approx .34 < .5 \) for natural images (and similarly \( \mu \approx .33 \) for pedestrians). This implies that \( h_q' \approx \mu h_q \) could serve as a reasonable approximation for gradient histograms in images downsampled by \( k = 2 \).

In other words, similarly to upsampling, gradient histograms computed over original and half resolution images tend to differ by a multiplicative constant (although the constant is not the inverse of the sampling factor). In Figure 2 we show the quality of the above approximations on example images. The agreement between predictions and observations is accurate for typical images (but fails for images with atypical statistics).

3.3 Histograms of Normalized Gradients

Suppose we replaced the gradient magnitude \( M \) by the normalized gradient magnitude \( \hat{M} \) defined as \( \hat{M}(i,j) = M(i,j)/(\overline{M}(i,j) + 0.005) \), where \( \overline{M} \) is the average gradient magnitude in each 11 \( \times \) 11 image patch (computed by convolving \( M \) with an L1 normalized 11 \( \times \) 11 triangle filter). Using the normalized gradient \( \hat{M} \) gives improved results in the context of object detection (see §6). Observe that we have now introduced an additional nonlinearity to the gradient computation; do the previous results for gradient histograms still hold if we use \( \hat{M} \) instead of \( M \)?

In Figure 1(c) we plot the distribution of \( r_q = h_q'/h_q \) for histograms of normalized gradients given a downsampling factor of \( k = 2 \). As with the original gradient histograms, the distributions of \( r_q \) are normally distributed and have similar means for pedestrian and natural images (\( \mu \approx .26 \) and \( \mu \approx .27 \), respectively). Observe, however, that the expected value of \( r_q \) for normalized gradient histograms is quite different than for the original histograms (Figure 1(b)).

Deriving analytical expressions governing the scaling properties of progressively more complex feature types would be difficult or even impossible. Instead, in §4 we describe a general law governing feature scaling.
4 Statistics of Multiscale Features

To understand more generally how features behave in resampled images, we turn to the study of natural image statistics [7], [33]. The analysis below provides a deep understanding of the behavior of multiscale features. The practical result is a simple yet powerful approach for predicting the behavior of gradients and other low-level features in resampled images without resorting to analytical derivations that may be difficult except under the simplest conditions.

We begin by defining a broad family of features. Let $\Omega$ be any low-level shift invariant function that takes an image $I$ and creates a new channel image $C = \Omega(I)$ where a channel $C$ is a per-pixel feature map such that output pixels in $C$ are computed from corresponding patches of input pixels in $I$ (thus preserving overall image layout). $C$ may be downsampled relative to $I$ and may contain multiple layers $k$. We define a feature $f_\Omega(I)$ as a weighted sum of the channel $C = \Omega(I)$:

$$f_\Omega(I) = \sum_{i,j,k} w_{i,j,k} C(i,j,k).$$

Numerous local and global features can be written in this form including gradient histograms, linear filters, color statistics, and others [29]. Any such low-level shift invariant $\Omega$ can be used, making this representation quite general.

Let $I_s$ denote $I$ at scale $s$, where the dimensions $h_s \times w_s$ of $I_s$ are $s$ times the dimensions of $I$. For $s > 1$, $I_s$ (which denotes a higher resolution version of $I$) typically differs from $I$ upscaled by $s$, while for $s < 1$ an excellent approximation of $I_s$ can be obtained by downsampling $I$. Next, for simplicity we redefine $f_\Omega(I_s)$ as:

$$f_\Omega(I_s) \equiv \frac{1}{h_s w_s k} \sum_{i,j,k} C_s(i,j,k) \text{ where } C_s = \Omega(I_s).$$

In other words $f_\Omega(I_s)$ denotes the global mean of $C_s$ computed over locations $ij$ and layers $k$. Everything in the following derivations based on global means also holds for local means (e.g. local histograms).

Our goal is to understand how $f_\Omega(I_s)$ behaves as a function of $s$ for any choice of shift invariant $\Omega$.

4.1 Power Law Governs Feature Scaling

Ruderman and Bialek [33], [67] explored how the statistics of natural images behave as a function of the scale at which an image ensemble was captured, i.e. the visual angle corresponding to a single pixel. Let $\phi(I)$ denote an arbitrary (scalar) image statistic and $E[\cdot]$ denote expectation over an ensemble of natural images. Ruderman and Bialek made the fundamental discovery that the ratio of $E[\phi(I_s)]$ to $E[\phi(I_1)]$, computed over two ensembles of natural images captured at scales $s_1$ and $s_2$, respectively, depends only on the ratio of $s_1/s_2$ and is independent of the absolute scales $s_1$ and $s_2$ of the ensembles.

Ruderman and Bialek’s findings imply that $E[\phi(I_s)]$ follows a power law:

$$E[\phi(I_{s_1})]/E[\phi(I_{s_2})] = (s_1/s_2)^{-\lambda_\phi},$$

Every statistic $\phi$ will have its own corresponding $\lambda_\phi$. In the context of our work, for any channel type $\Omega$ we can use the scalar $f_\Omega(I)$ in place of $\phi(I)$ and $\lambda_\Omega$ in place of $\lambda_\phi$. While Eqn. (3) gives the behavior of $f_\Omega$ w.r.t. to scale over an ensemble of images, we are interested in the behavior of $f_\Omega$ for a single image.

We observe that a single image can itself be considered an ensemble of image patches (smaller images). Since $\Omega$ is shift invariant, we can interpret $f_\Omega(I)$ as computing the average of $f_\Omega(I^k)$ over every patch $I^k$ of $I$ and therefore Eqn. (3) can be applied directly for a single image. We formalize this below.

We can decompose an image $I$ into $K$ smaller images $I^1 \ldots I^K$ such that $I = [I^1 \ldots I^K]$. Given that $\Omega$ must be shift invariant and ignoring boundary effects gives $\Omega(I^k) = \Omega(I^k[I^1 \ldots I^K]) \approx \Omega(I^k[I^1] \ldots [I^K])$, and substituting into Eqn. (2) yields $f_\Omega(I) \approx \sum f_\Omega(I^k)/K$. However, we can consider $I^1 \ldots I^K$ as a (small) image ensemble, and $f_\Omega(I) \approx E[f_\Omega(I^k)]$ an expectation over that ensemble. Therefore, substituting $f_\Omega(I_{s_1}) \approx E[f_\Omega(I^k_{s_1})]$ and $f_\Omega(I_{s_2}) \approx E[f_\Omega(I^k_{s_2})]$ into Eqn. (3) yields:

$$f_\Omega(I_{s_1})/f_\Omega(I_{s_2}) = (s_1/s_2)^{-\lambda_\Omega} + \mathcal{E},$$

where we use $\mathcal{E}$ to denote the deviation from the power law for a given image. Each channel type $\Omega$ has its own corresponding $\lambda_\Omega$, which we can determine empirically.

In §4.2 we show that on average Eqn. (4) provides a remarkably good fit for multiple channel types and image sets (i.e. we can fit $\lambda_\Omega$ such that $E[\mathcal{E}] \approx 0$). Additionally, experiments in §4.3 indicate that the magnitude of deviation for individual images, $E[\mathcal{E}^2]$, is reasonable and increases only gradually as a function of $s_1/s_2$.

4.2 Estimating $\lambda$

We perform a series of experiments to verify Eqn. (4) and estimate $\lambda$ for numerous channel types $\Omega$.

To estimate $\lambda_\Omega$ for a given $\Omega$, we first compute:

$$\mu_s = \frac{1}{N} \sum_{i=1}^{N} f_\Omega(I_s^i)/f_\Omega(I_1^i),$$

for $N$ images $I^i$ and multiple values of $s < 1$, where $I_s^i$ is obtained by downsampling $I_1^i = I$. We use two image ensembles, one of $N = 1237$ pedestrian images and one of $N = 4280$ natural images (for details see 2). Let $F(s) = E[\phi(I_s)]$. We can rewrite the observation by saying there exists a function $R$ such that $F(s_{12}) = F(s_1)/F(s_2) = R(s_1/s_2)$. Applying repeatedly gives $F(s_{12})/F(1) = R(s_1), F(1)/F(s_2) = R(1/s_2)$, and $F(s_{12})/F(1) = R(s_1/s_2)$. Therefore $R(s_1/s_2) = R(s_1)R(1/s_2)$. Next, let $R'(s) = R(e^s)$ and observe that $R'(s_1 + s_2) = R'(s_1)R'(s_2)$ since $R(s_1 + s_2) = R(s_1)R(s_2)$. If $R'$ is also continuous and non-zero, then it must take the form $R'(s) = e^{-\lambda s}$ for some constant $\lambda$ [68]. This implies $R(s) = R'(\ln(s)) = e^{-\lambda \ln(s)} = s^{-\lambda}$. Therefore, $E[\phi(I_s)]$ must follow a power law (see also Eqn. (9) in [67]).
(c) difference of gaussians (DoG)

(e) local standard deviation

(f) HOG [21]

Fig. 3. Power Law Feature Scaling: For each of six channel types we plot $\mu_s = s^{-\lambda_s} + E[\mathcal{E}]$, Our goal is to fit $\lambda_s$ accordingly and verify the fidelity of Eqn. (4) for various channel types $\Omega$ (i.e. verify that $E[\mathcal{E}] = 0$).

For each $\Omega$, we measure $\mu_s$ according to Eqn. (5) across three octaves with eight scales per octave for a total of 24 measurements at $s = 2^{-\frac{1}{4}}, \ldots, 2^{-\frac{24}{4}}$. Since image dimensions are rounded to the nearest integer, we compute and use $s' = \sqrt{h_w w_s} / hw$, where $h \times w$ and $h_w \times w_s$ are the dimensions of the original and downsampled images, respectively.

In Figure 3 we plot $\mu_s$ versus $s'$ using a log-log plot for six channel types for both the pedestrian and natural images. In all cases $\mu_s$ follows a power law with all measurements falling along a line on the log-log plots, as predicted. However, close inspection shows $\mu_s$ does not start exactly at 1 as expected: downsampling introduces a minor amount of blur even for small downsampling factors. We thus expect $\mu_s$ to have the form $\mu_s = a_\Omega s^{-\lambda_s}$, with $a_\Omega \neq 1$ as an artifact of the interpolation. Note that $a_\Omega$ is only necessary for estimating $\lambda_s$ from downsampled images and is not used subsequently. To estimate $a_\Omega$ and $\lambda_s$, we use a least squares fit of $\log_2(\mu_s) = a'_{\Omega} - \lambda_s \log_2(s')$ to the 24 measurements computed over natural images (and set $a_\Omega = 2^{a'}$). Resulting estimates of $\lambda_s$ are given in plot legends in Figure 3.

There is strong agreement between the resulting best-fit lines and the observations. In legend brackets in Figure 3 we report expected error $|E[\mathcal{E}]| = |\mu_s - a_\Omega s^{-\lambda_s}|$

3. Figure 3 generalizes the results shown in Figure 1. However, by switching from channel sums to channel means, $\mu_{1/2}$ in Figures 3(a) and 3(b) is 4x larger than $\mu$ in Figures 1(b) and 1(c), respectively.

for both natural and pedestrian images averaged over $s$ (using $a_\Omega$ and $\lambda_s$ estimated using natural images). For basic gradient histograms $|E[\mathcal{E}]| = 0.018$ for natural images and $|E[\mathcal{E}]| = 0.037$ for pedestrian images. Indeed, for every channel type Eqn. (4) is an excellent fit to the observations $\mu_s$ for both image ensembles.

The derivation of Eqn. (4) depends on the distribution of image statistics being stationary with respect to scale; that this holds for all channel types tested, and with nearly an identical constant for both pedestrian and natural images, shows the estimate of $\lambda_s$ is robust and generally applicable.

4.3 Deviation for Individual Images

In §4.2 we verified that Eqn. (4) holds for an ensemble of images; we now examine the magnitude of deviation from the power law for individual images. We study the effect this has in the context of object detection in §6.

Plots of $f_{\Omega}(I_{s1}) / f_{\Omega}(I_{s2})$ for randomly selected images are shown as faint gray lines in Figure 3. The individual curves are relatively smooth and diverge only somewhat from the best-fit line. We quantify their deviation by defining $\sigma_s$ analogously to $\mu_s$ in Eqn. (5):

$$\sigma_s = \text{stddev}[f_{\Omega}(I_{s}) / f_{\Omega}(I_{s})] = \text{stddev}[\mathcal{E}],$$

where ‘stddev’ denotes the sample standard deviation (computed over $N$ images) and $\mathcal{E}$ is the error associated with each image and scaling factor as defined in Eqn. (4).

In §4.2 we confirmed that $E[\mathcal{E}] = 0$, our goal now is to understand how $\sigma_s = \text{stddev}[\mathcal{E}] \approx \sqrt{E[\mathcal{E}]^2}$ behaves.
In Figure 4 we plot $\sigma_s$ as a function of $s$ for the same channels as in Figure 3. In legend brackets we report $\sigma_s$ for $s = \frac{1}{2}$ for both natural and pedestrian images; for all channels studied $\frac{1}{2} < .2$. In all cases $\sigma_s$ increases gradually with increasing $s$ and the deviation is low for small $s$. The expected magnitude of $\mathcal{E}$ varies across channels, for example histograms of normalized gradients (Figure 4(b)) have lower $\sigma_s$ than their unnormalized counterparts (Figure 4(a)). The trivial grayscale channel (Figure 4(d)) has $\sigma_s = 0$ as the approximation is exact.

Observe that often $\sigma_s$ is greater for natural images than for pedestrian images. Many of the natural images contain relatively little structure (e.g. a patch of sky), for such images $f_\Omega(I)$ is small for certain $\Omega$ (e.g. simple gradient histograms) resulting in more variance in the ratio in Eqn. (4). For HOG channels (Figure 4(f)), which have additional normalization, this effect is minimized.

4.4 Miscellanea

We conclude this section with additional observations.

**Interpolation Method:** Varying the interpolation algorithm for image resampling does not have a major effect. In Figure 5(a), we plot $\mu_{1/2}$ and $\sigma_{1/2}$ for normalized gradient histograms computed using nearest neighbor, bilinear, and bicubic interpolation. In all three cases both $\mu_{1/2}$ and $\sigma_{1/2}$ remain essentially unchanged.

**Window Size:** All preceding experiments were performed on $128 \times 64$ windows. In Figure 5(b) we plot the effect of varying the window size. While $\mu_{1/2}$ remains relatively constant, $\sigma_{1/2}$ increases with decreasing window size (see also the derivation of Eqn. (4)).

**Upsampling:** The power law can predict features in higher resolution images but not upsampled images. In practice, though, we want to predict features in higher resolution as opposed to (smooth) upsampled images.

**Robust Estimation:** In preceding derivations, when computing $f_\Omega(I_{s_1})/f_\Omega(I_{s_2})$ we assumed that $f_\Omega(I_{s_2}) \neq 0$. For the $\Omega$’s considered this was the case after windows of near uniform intensity were excluded (see §3.1). Alternatively, we have found that excluding $I$ with $f_\Omega(I) \approx 0$ when estimating $\lambda$ results in more robust estimates.

**Sparse Channels:** For sparse channels where frequently $f_\Omega(I) \approx 0$, e.g., the output of a sliding-window object detector, $\sigma$ will be large. Such channels may not be good candidates for the power law approximation.

**One-Shot Estimates:** We can estimate $\lambda$ as described in §4.2 using a single image in place of an ensemble ($N = 1$). Such estimates are noisy but not entirely unreasonable; e.g., on normalized gradient histograms (with $\lambda \approx 0.101$) the mean of 4280 single image estimates of $\lambda$ if 0.096 and the standard deviation of the estimates is 0.073.

**Scale Range:** We expect the power law to break down at extreme scales not typically encountered under natural viewing conditions (e.g. under high magnification).
we have computed \( C = \Omega(I) \); can we predict \( C_s = \Omega(I_s) \) at a new scale \( s \)? **Top:** the standard approach is to compute \( C_s = \Omega(R(I,s)) \), ignoring the information contained in \( C = \Omega(I) \). **Bottom:** instead, based on the power law introduced in §4, we propose to approximate \( C_s \) by \( R(C,s) \cdot s^{-\lambda_0} \). This approach is simple, general, and accurate, and allows for fast feature pyramid construction.

## 5 Fast Feature Pyramids

We introduce a novel, efficient scheme for constructing feature pyramids. First, in §5.1 we outline an approach for scaling feature channels. Next, in §5.2 we show its application to constructing feature pyramids efficiently and we analyze computational complexity in §5.3.

### 5.1 Feature Channel Scaling

We propose an extension of the power law governing feature scaling introduced in §4 that applies directly to channel images. As before, let \( I_s \) denote I captured at scale \( s \) and \( R(I,s) \) denote I resampled by \( s \). Suppose we have computed \( C = \Omega(I) \); can we predict the channel image \( C_s = \Omega(I_s) \) at a new scale \( s \) using only \( C \)?

The standard approach is to compute \( C_s = \Omega(R(I,s)) \), ignoring the information contained in \( C = \Omega(I) \). Instead, we propose the following approximation:

\[
C_s \approx R(C,s) \cdot s^{-\lambda_0} \tag{7}
\]

A visual demonstration of Eqn. (7) is shown in Figure 6.

Eqn. (7) follows from Eqn. (4). Setting \( s_1 = s \), \( s_2 = 1 \), and rearranging Eqn. (4) gives \( f_\Omega(I_s) \approx f_\Omega(I)^{s^{-\lambda_0}} \). This relation must hold not only for the original images but also for any pair of corresponding windows \( w_s \) and \( w \) in \( I_s \) and \( I \), respectively. Expanding yields:

\[
\frac{1}{|w_s|} \sum_{i,j \in w_s} C_s(i,j) \approx \frac{1}{|w|} \sum_{i,j \in w} C(i,j)^{s^{-\lambda_0}}
\]

\[
C_s \approx R(C,s)^{s^{-\lambda_0}}
\]

The final line follows because if for all corresponding windows \( C'/|w_s| \approx \sum C'/|w| \), then \( C' \approx R(C,s) \).

On a per-pixel basis, the approximation of \( C_s \) in Eqn. (7) may be quite noisy. The standard deviation \( \sigma_s \) of the ratio \( f_\Omega(I_s)/f_\Omega(I) \) depends on the size of the window \( w_s \): \( \sigma_s \) increases as \( w \) decreases (see Figure 5(b)). Therefore, the accuracy of the approximation for \( C_s \) will improve if information is aggregated over multiple pixels of \( C_s \).

### 5.2 Fast Feature Pyramids

A feature pyramid is a multi-scale representation of an image \( I \) where channels \( C_s = \Omega(I_s) \) are computed at every scale \( s \). Scales are sampled evenly in log-space, starting at \( s = 1 \), with typically 4 to 12 scales per octave (an octave is the interval between one scale and another with half or double its value). The standard approach to constructing a feature pyramid is to compute \( C_s = \Omega(R(I,s)) \) for every \( s \), see Figure 7 (top).

The approximation in Eqn. (7) suggests a straightforward method for efficient feature pyramid construction. We begin by computing \( C_s = \Omega(R(I,s)) \) at just one scale per octave (\( s \in \{1, 1.5, 2, \ldots\} \)). At intermediate scales, \( C_s \) is computed using \( C_s \approx R(C_{s'}, s/s')(s/s')^{-\lambda_0} \) where \( s' \) is the nearest scale for which we have \( C_{s'} = \Omega(I_{s'}) \), see Figure 7 (bottom).

Computing \( C_s = \Omega(R(I,s)) \) at one scale per octave provides a good tradeoff between speed and accuracy. The cost of evaluating \( \Omega \) is within 33% of computing \( R(I) \) at the original scale (see §5.3) and channels do not need to be approximated beyond half an octave (keeping error low, see §4.3). While the number of evaluations of \( R \) is constant (evaluations of \( R(I,s) \) are replaced by \( R(C,s) \)), if each \( C_s \) is downsampled relative to \( I_s \), as described in §5.1, evaluating \( R(C,s) \) is faster than \( R(I,s) \).
Fig. 8. Overview of the ACF detector. Given an input image \( I \), we compute several channels \( C = \Omega(I) \), sum every block of pixels in \( C \), and smooth the resulting lower resolution channels. Features are single pixel lookups in the aggregated channels. Boosting is used to learn decision trees over these features (pixels) to distinguish object from background. With the appropriate choice of channels and careful attention to design, ACF achieves state-of-the-art performance in pedestrian detection.

Alternate schemes, such as interpolating between two nearby scales \( s' \) for each intermediate scale \( s \) or evaluating \( \Omega \) more densely, could result in even higher pyramid accuracy (at increased cost). However, the proposed approach proves sufficient for object detection (see §6).

5.3 Complexity Analysis
The computational savings of computing approximate feature pyramids is significant. Assume the cost of computing \( \Omega \) is linear in the number of pixels in an \( n \times n \) image (as is often the case). The cost of constructing a feature pyramid with \( m \) scales per octave is:

\[
\sum_{k=0}^{\infty} n^2 2^{-2k/m} = n^2 \sum_{k=0}^{\infty} \left( \frac{4}{m} \right)^k = \frac{n^2}{1 - 4^{-1/m}} \approx \frac{mn^2}{\ln 4} \quad (8)
\]

The second equality follows from the formula for a sum of a geometric series; the last approximation is valid for large \( m \) (and follows from l'Hôpital's rule). In the proposed approach we compute \( \Omega \) once per octave (\( m = 1 \)). In the total cost is \( \frac{4}{3}n^2 \), which is only 33% more than the cost of computing single scale features. Typical detectors are evaluated on 8 to 12 scales per octave [31], thus according to (8) we achieve an order of magnitude savings over computing \( \Omega \) densely (and intermediate \( C_s \) are computed efficiently through resampling afterward).

6 APPLICATIONS TO OBJECT DETECTION
We demonstrate the effectiveness of fast feature pyramids in the context of object detection with three distinct detection frameworks. First, in §6.1 we show the efficacy of our approach with a simple yet state-of-the-art pedestrian detector we introduce in this work called Aggregated Channel Features (ACF). In §6.2 we describe an alternate approach for exploiting approximate multiscale features using integral images computed over the same channels (Integral Channel Features or ICF), much as in our previous work [29], [39]. Finally, in §6.3 we approximate HOG feature pyramids for use with Deformable Part Models (DPM) [35].

6.1 Aggregated Channel Features (ACF)
The ACF detection framework is conceptually straightforward (Figure 8). Given an input image \( I \), we compute several channels \( C = \Omega(I) \), sum every block of pixels in \( C \), and smooth the resulting lower resolution channels. Features are single pixel lookups in the aggregated channels. Boosting is used to train and combine decision trees over these features (pixels) to distinguish object from background and a multiscale sliding-window approach is employed. With the appropriate choice of channels and careful attention to design, ACF achieves state-of-the-art performance in pedestrian detection.

Channels: ACF uses the same channels as [39]: normalized gradient magnitude, histogram of oriented gradients (6 channels), and LUV color channels. Prior to computing the 10 channels, \( I \) is smoothed with a \([1 2 1]/4\) filter. The channels are divided into \( 4 \times 4 \) blocks and pixels in each block are summed. Finally the channels are smoothed, again with a \([1 2 1]/4\) filter. For \( 640 \times 480 \) images, computing the channels runs at over \( 100 \) fps on a modern PC. The code is optimized but runs on a single CPU; further gains could be obtained using multiple cores or a GPU as in [30].

Pyramid: Computation of feature pyramids at octave-spaced scale intervals runs at ~75 fps on \( 640 \times 480 \) images. Meanwhile, computing exact feature pyramids with eight scales per octave slows to ~15 fps, precluding real-time detection. In contrast, our fast pyramid construction (see §5) with 7 of 8 scales per octave approximated runs at nearly 50 fps.

Detector: For pedestrian detection, AdaBoost [69] is used to train and combine 2048 depth-two trees over the \( 128 \cdot 64 \cdot 10/16 = 5120 \) candidate features (channel pixel lookups) in each \( 128 \times 64 \) window. Training with multiple rounds of bootstrapping takes ~10 minutes (a parallel implementation reduces this to ~3 minutes). The detector has a step size of 4 pixels and 8 scales per octave. For \( 640 \times 480 \) images, the complete system, including fast pyramid construction and sliding-window detection, runs at over 30 fps allowing for real-time uses (with exact feature pyramids the detector slows to 12 fps).

Code: Code for the ACF framework is available online. For more details on the channels and detector used in ACF, including exact parameter settings and training framework, we refer users to the source code.

Accuracy: We report accuracy of ACF with exact and fast feature pyramids in Table 1. Following the methodology of [31], we summarize performance using the log-average miss rate (MR) between \( 10^{-2} \) and \( 10^{0} \) false positives per image. Results are reported on four pedestrian datasets: INRIA [21], Caltech [31], TUD-Brussels [36] and

Table 1

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Shapelet</td>
<td>82</td>
<td>91</td>
<td>95</td>
<td>91</td>
<td>91</td>
<td>90</td>
<td>91</td>
<td>91</td>
<td>90</td>
</tr>
<tr>
<td>VJ [27]</td>
<td>72</td>
<td>95</td>
<td>95</td>
<td>90</td>
<td>88</td>
<td>88</td>
<td>88</td>
<td>88</td>
<td>88</td>
</tr>
<tr>
<td>PoseNetV2</td>
<td>80</td>
<td>86</td>
<td>88</td>
<td>92</td>
<td>87</td>
<td>87</td>
<td>87</td>
<td>87</td>
<td>87</td>
</tr>
<tr>
<td>HistSvm</td>
<td>43</td>
<td>73</td>
<td>83</td>
<td>72</td>
<td>68</td>
<td>68</td>
<td>68</td>
<td>68</td>
<td>68</td>
</tr>
<tr>
<td>HOG [21]</td>
<td>46</td>
<td>68</td>
<td>78</td>
<td>64</td>
<td>64</td>
<td>64</td>
<td>64</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>HogLsp [75]</td>
<td>39</td>
<td>68</td>
<td>82</td>
<td>55</td>
<td>61</td>
<td>61</td>
<td>61</td>
<td>61</td>
<td>61</td>
</tr>
<tr>
<td>MF [74]</td>
<td>56</td>
<td>68</td>
<td>73</td>
<td>60</td>
<td>59</td>
<td>59</td>
<td>59</td>
<td>59</td>
<td>59</td>
</tr>
<tr>
<td>PLS [75]</td>
<td>40</td>
<td>62</td>
<td>71</td>
<td>55</td>
<td>57</td>
<td>57</td>
<td>57</td>
<td>57</td>
<td>57</td>
</tr>
<tr>
<td>MF+CSS [76]</td>
<td>25</td>
<td>61</td>
<td>60</td>
<td>61</td>
<td>52</td>
<td>52</td>
<td>52</td>
<td>52</td>
<td>52</td>
</tr>
<tr>
<td>MF+Motion</td>
<td>–</td>
<td>51</td>
<td>55</td>
<td>60</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>LatSvmV2</td>
<td>20</td>
<td>63</td>
<td>70</td>
<td>51</td>
<td>51</td>
<td>51</td>
<td>51</td>
<td>51</td>
<td>51</td>
</tr>
<tr>
<td>FPDW [39]</td>
<td>21</td>
<td>57</td>
<td>63</td>
<td>60</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>ChnFtrs [29]</td>
<td>22</td>
<td>56</td>
<td>60</td>
<td>57</td>
<td>49</td>
<td>49</td>
<td>49</td>
<td>49</td>
<td>49</td>
</tr>
<tr>
<td>Crosstalk</td>
<td>19</td>
<td>54</td>
<td>58</td>
<td>52</td>
<td>46</td>
<td>46</td>
<td>46</td>
<td>46</td>
<td>46</td>
</tr>
<tr>
<td>VeryFast</td>
<td>16</td>
<td>–</td>
<td>–</td>
<td>55</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>MultiResC</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>ICF-Exact §6.2</td>
<td>18</td>
<td>48</td>
<td>53</td>
<td>50</td>
<td>42</td>
<td>42</td>
<td>42</td>
<td>42</td>
<td>42</td>
</tr>
<tr>
<td>ICF §6.2</td>
<td>19</td>
<td>51</td>
<td>55</td>
<td>56</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
</tr>
<tr>
<td>ACF-Exact §6.1</td>
<td>17</td>
<td>43</td>
<td>50</td>
<td>50</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>ACF §6.1</td>
<td>17</td>
<td>45</td>
<td>52</td>
<td>51</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
</tr>
</tbody>
</table>

**6.2 Integral Channel Features (ICF)**

Integral Channel Features (ICF) [29] are a precursor to the ACF framework described in §6.1. Both ACF and ICF use the same channel features and boosted classifiers; the key difference between the two frameworks is that ACF uses pixel lookups in aggregated channels as features while ICF uses sums over rectangular channel regions (computed efficiently with integral images).

Accuracy of ICF with exact and fast feature pyramids is shown in Table 1. ICF achieves state-of-the-art results: inferior to ACF but otherwise outperforming most competing approaches. The MR of ACF averaged over the four datasets is 42% with exact feature pyramids and 45% with fast feature pyramids. The gap of 3% is larger than the 1% gap for ACF but still small. With fast feature pyramids ICF runs at ~16 fps, see Figure 10. ICF is slower than ACF due to construction of integral images and more expensive features (rectangular sums computed via integral images versus single pixel lookups). For more details on ICF, see [29], [39]. The variant tested here uses identical channels to ACF.

Detection performance with fast feature pyramids under varying settings is shown in Figure 12. The plots mirror the results shown in Figure 11 for ACF. The key result is given in Figure 12(a): when approximating 7 of 8 scales per octave, the MR for ICF is .169 which is virtually identical to the MR of .166 obtained using the exact feature pyramid. Even approximating 15 of every 16 scales increases MR only somewhat. Constructing the channels without correcting for power law scaling, or using an incorrect value of λ, results in markedly decreased performance, see Figure 11(b). Finally, we observe that at least 8 scales per octave must be used for good performance (Figure 11(c)), making the proposed scheme crucial for achieving detection results that are both fast and accurate.

**Pyramid parameters**: Detection performance on INRIA [21] with fast feature pyramids under varying settings is shown in Figure 11. The key result is given in Figure 11(a): when approximating 7 of 8 scales per octave, the MR for ACF is .169 which is virtually identical to the MR of .166 obtained using the exact feature pyramid. Even approximating 15 of every 16 scales increases MR only somewhat. Constructing the channels without correcting for power law scaling, or using an incorrect value of λ, results in markedly decreased performance, see Figure 11(b). Finally, we observe that at least 8 scales per octave must be used for good performance (Figure 11(c)), making the proposed scheme crucial for achieving detection results that are both fast and accurate.
Fig. 10. Log-average miss rate (MR) on the INRIA pedestrian dataset [21] versus frame rate on 640 × 480 images for multiple detectors. Method runtimes were obtained from [31], see also [31] for citations for detectors A-L. Numbers in brackets indicate MR/fps for select approaches, sorted by speed. All detectors that run at 5 fps and higher are based on our fast feature pyramids; these methods are also the most accurate. They include: (M) FPDW [39] which is our original implementation of ICF, (N) ICF [§6.2], (O) ACF [§6.1], (P) crosstalk cascades [40], and (Q) the VeryFast detector from Benenson et al. [30]. Both (P) and (Q) use the power law governing feature scaling described in this work; the additional speedups in (P) and (Q) are based on improved cascade strategies, multi-resolution models and a GPU implementation, and are orthogonal to the gains achieved by using approximate multiscale features.

Fig. 11. Effect of parameter setting of fast feature pyramids on the ACF detector [§6.1]. We report log-average miss rate (MR) averaged over 25 trials on the INRIA pedestrian dataset [21]. Orange diamonds denote default parameter settings: 7/8 scales approximated per octave, $\lambda \approx .17$ for the normalized gradient channels, and 8 scales per octave in the pyramid. (a) The MR stays relatively constant as the fraction of approximated scales increases up to 7/8 demonstrating the efficacy of the proposed approach. (b) Sub-optimal values of $\lambda$ when approximating the normalized gradient channels cause a marked decrease in performance. (c) At least 8 scales per octave are necessary for good performance, making the proposed scheme crucial for achieving detection results that are both fast and accurate.

Fig. 12. Effect of parameter setting of fast feature pyramids on the ICF detector [§6.2]. The plots mirror the results shown in Figure 11 for the ACF detector, although overall performance for ICF is slightly lower. (a) When approximating 7 of every 8 scales in the pyramid, the MR for ICF is .195 which is only slightly worse than the MR of .176 obtained using exact feature pyramids. (b) Computing approximate channels with an incorrect value of $\lambda$ results in decreased performance (although using a slightly larger $\lambda$ than predicted appears to improve results marginally). (c) Similarly to the ACF framework, at least 8 scales per octave are necessary to achieve good results.
Our insight ultimately relies on the fractal structure of much of the visual world. By investigating the statistics of natural images we have demonstrated that the behavior of image features can be predicted reliably across scales. Our calculations and experiments show that this makes it possible to estimate features at a given scale inexpensively by extrapolating computations carried out at a coarsely sampled set of scales. While our results do not hold under all circumstances, for instance, on images of textures or white noise, they do hold for images typically encountered in the natural world.

In order to validate our findings we studied the performance of three end-to-end object detection systems. We found that detection rates are relatively unaffected while computational costs decrease considerably. This has led to the first detectors that operate at frame rate while using rich feature representations.

Our results are not restricted to object detection nor to visual recognition. The foundations we have developed should readily apply to other computer vision tasks where a fine-grained scale sampling of features is necessary as the image processing front end.

7 Conclusion

Improvements in the performance of visual recognition systems in the past decade have in part come from the realization that finely sampled pyramids of image features provide a good front-end for image analysis. It is widely believed that the price to be paid for improved performance is sharply increased computational costs. We have shown that this is not necessarily so. Finely sampled pyramids may be obtained inexpensively by extrapolation from coarsely sampled ones. This insight decreases computational costs substantially.
Datasets/CaltechPedestrians/.

. Lades, J. Vorbruggen, J. Buhmann, J. Lange, C. von der M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zis-
D. G. Lowe, “Object recognition from local scale-invariant fea-
N. Dalal and B. Triggs, “Histograms of oriented gradients for hu-
P. Felzenszwalb, R. Girshick, and D. McAllester, “Cascade object de
P. Felzenszwalb, R. Girshick, and D. McAllester, “Cascade object de
P. Felzenszwalb, R. Girshick, and D. McAllester, “Cascade object de
M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zis-
P. Felzenszwalb, R. Girshick, and D. McAllester, “Cascade object de
P. Felzenszwalb and D. Huttenlocher, “Efficient matching of pict-
P. Felzenszwalb and D. McAllester, “Efficient matching of pict-
P. Felzenszwalb and D. McAllester, “Efficient matching of pict-

[14] D. Jones and J. Malik, “Computational framework for determin-
Piotr Dollar received his masters degree in computer science from Harvard University in 2002 and his PhD from the University of California, San Diego in 2007. He joined the Computational Vision lab at Caltech as a postdoctoral fellow in 2007. Upon being promoted to senior postdoctoral fellow he realized it time to move on, and in 2011, he joined the Interactive Visual Media Group at Microsoft Research, Redmond, where he currently resides. He has worked on object detection, pose estimation, boundary learning and behavior recognition. His general interests lie in machine learning and pattern recognition and their application to computer vision.

Ron Appel is completing his PhD in the Computational Vision lab at Caltech, where he currently holds an NSERC graduate award. He received his bachelor's and masters degrees in 2006 and 2008 in electrical and computer engineering from the University of Toronto, and co-founded ViewGenie Inc., a company specializing in intelligent image processing and search. His research interests include machine learning, visual object detection, and algorithmic optimization.

Serge Belongie received a BS (with honor) in EE from Caltech in 1995 and a PhD in EECS from Berkeley in 2000. While at Berkeley, his research was supported by an NSF Graduate Research Fellowship. From 2001-2013 he was a professor in the Department of Computer Science and Engineering at UCSD. He is currently a professor at Cornell NYC Tech and the Cornell Computer Science Department. His research interests include Computer Vision, Machine Learning, Crowdsourcing and Human-in-the-Loop Computing. He is also a co-founder of several companies including DigitalPersona, Anchovi Labs (acquired by Dropbox) and Orpix. He is a recipient of the NSF CAREER Award, the Alfred P. Sloan Research Fellowship and the MIT Technology Review “Innovators Under 35” Award.

Pietro Perona graduated in Electrical Engineering from the Università di Padova in 1985 and received a PhD in Electrical Engineering and Computer Science from the University of California at Berkeley in 1990. After a postdoctoral fellowship at MIT in 1990-91 he joined the faculty of Caltech in 1991, where he is now an Allen E. Puckett Professor of Electrical Engineering and Computation and Neural Systems. His current interests are visual recognition, modeling vision in biological systems, modeling and measuring behavior, and Visipedia. He has worked on anisotropic diffusion, multiresolution-multi-orientation filtering, human texture perception and segmentation, dynamic vision, grouping, analysis of human motion, recognition of object categories, and modeling visual search.