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 data sets) 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] (Gabors, 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], stereoscopic 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 overcomplete, 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 10^3. Empirical studies in computer vision provide increasing evidence in favor of overcomplete representations [21], [23], [24], [25], [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 percent detection rate on the INRIA pedestrian data set [21], VJ outputs over 10 false positives per image (FPPI), HOG [21] outputs ~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 ~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 tradeoff? 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

- P. Dollár is with the Interactive Visual Media Group at Microsoft Research, One Microsoft Way, Redmond, WA 98052.
  E-mail: pdollar@microsoft.com.
- R. Appel and P. Perona are with the Department of Electrical Engineering, California Institute of Technology, Pasadena, CA.
  E-mail: {appel, perona}@caltech.edu.
- S. Belongie is with Cornell NYC Tech and the Cornell Computer Science Department.

image structure across scales. Our analysis 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 (ICF) [29], aggregate channel features (a novel variant of integral channel features), and deformable part models (DPM) [35]. We show results for both pedestrian detection (measured on the Caltech [31], INRIA [21], TUD-Brussels [36] and ETH [37] data sets) 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 640×480 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 remainder of this paper is organized as follows. We review related work in Section 2. In Section 3 we show that it is possible to create high fidelity approximations of multiscale gradient histograms using gradients computed at a single scale. In Section 4 we generalize this finding to a broad family of feature types. We describe our efficient scheme for computing finely sampled feature pyramids in Section 5. In Section 6 we show applications of fast feature pyramids to object detection, resulting in considerable speedups with minor loss in accuracy. We conclude in Section 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 [27], [53]. 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 Section 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: $M(i, j)^2 = \frac{\partial I}{\partial x}(i, j)^2 + \frac{\partial I}{\partial y}(i, j)^2$ and $O(i, j) = \arctan\left(\frac{\partial I}{\partial y}(i, j) / \frac{\partial I}{\partial x}(i, j)\right)$. 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 qth 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 upsampled image is similar to that of the original, lower-resolution image (upsampling does not create new structure). Assume I is a
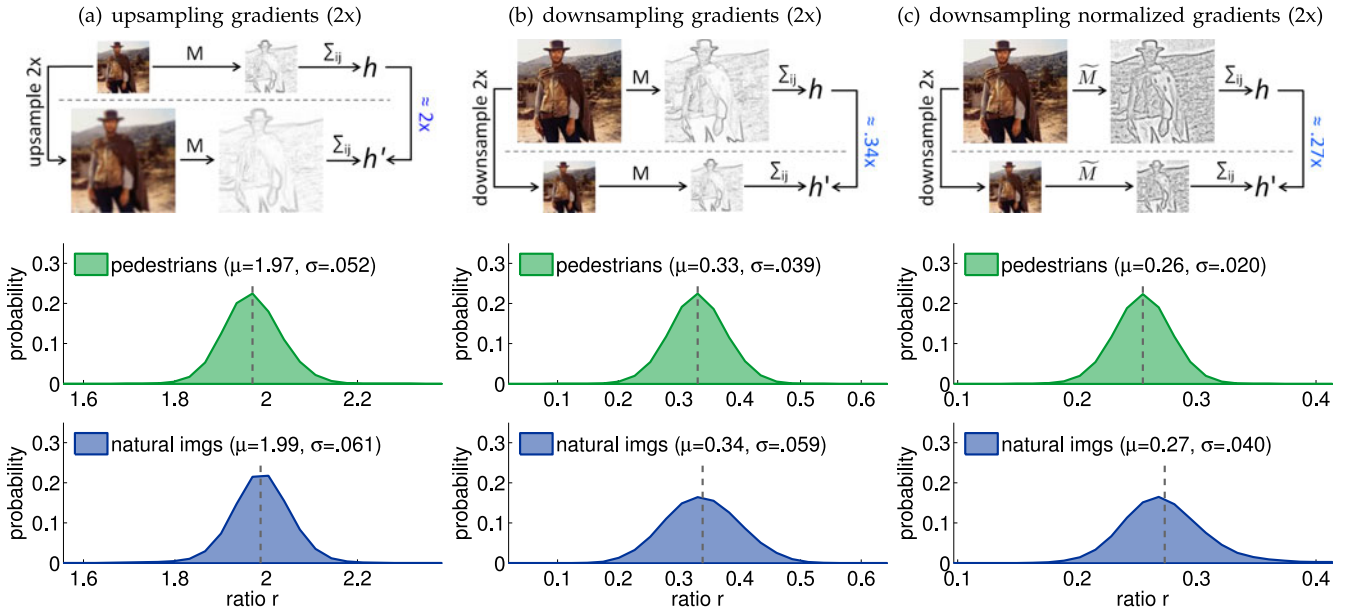
Fig. 1. Behavior of gradient histograms in images resampled by a factor of two. *(a) Upsampling gradients*. Given images I and I' where I' denotes I upsampled by two, and corresponding gradient magnitude images M and M', the ratio $\Sigma M / \Sigma M'$ should be approximately 2. The middle/bottom panels show the distribution of this ratio for gradients at fixed orientation over pedestrian/natural images. In both cases the mean $\mu \approx 2$, as expected, and the variance is relatively small. *(b) Downsampling gradients*. Given images I and I' where I' denotes I *downsampled* by two, the ratio $\Sigma M / \Sigma M' \approx 0.34$, not 0.5 as might be expected from (a) as downsampling results in loss of high frequency content. *(c) Downsampling normalized gradients*. Given *normalized* gradient magnitude images $\widetilde{M}$ and $\widetilde{M}'$, the ratio $\Sigma \widetilde{M} / \Sigma \widetilde{M}' \approx 0.27$. Instead of trying to derive analytical expressions governing the scaling properties of various feature types under different resampling factors, in Section 4 we describe a general law governing feature scaling.

continuous signal, and let I' denote I upsampled by a factor of k: $I'(x, y) \equiv I(x/k, y/k)$. Using the definition of a derivative, one can show that $\frac{\partial I'}{\partial x}(i, j) = \frac{1}{k} \frac{\partial I}{\partial x}(i/k, j/k)$, and likewise for $\frac{\partial I'}{\partial y'}$, which simply states the intuitive fact that the rate of change in the upsampled 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 $M'(i, j) \approx \frac{1}{k} M(\lceil i/k \rceil, \lceil j/k \rceil)$ denote the gradient magnitude in an upsampled discrete image. Then:

$$
\begin{aligned}
\sum_{i=1}^{kn} \sum_{j=1}^{km} M'(i, j) &\approx \sum_{i=1}^{kn} \sum_{j=1}^{km} \frac{1}{k} M(\lceil i/k \rceil, \lceil j/k \rceil) \\
&= k^2 \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{1}{k} M(i, j) = k \sum_{i=1}^{n} \sum_{j=1}^{m} M(i, j).
\end{aligned}
\tag{1}
$$

Thus, the sum of gradient magnitudes in the original and upsampled image should be related by about a factor of k. Angles should also be mostly preserved since $\frac{\partial I'}{\partial x}(i, j) / \frac{\partial I'}{\partial y}(i, j) \approx \frac{\partial I}{\partial x}(i/k, j/k) / \frac{\partial I}{\partial y}(i/k, j/k)$. 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 upsampled image using gradients computed at the original scale.

Experiments. One may verify experimentally that in images of natural scenes, upsampled 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 $1{,}237$ cropped pedestrian images from the INRIA pedestrians training data set [21]. Each image is 128×64 and contains a

pedestrian approximately 96 pixels tall. The second image set contains 128×64 windows cropped at random positions from the $1{,}218$ images in the INRIA negative training set. We sample $5{,}000$ windows but exclude nearly uniform windows, i.e., those with average gradient magnitude under 0.01, resulting in $4{,}280$ 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. Fig. 1a shows the distribution of r_q for one bin on the $1{,}237$ pedestrian and $4{,}280$ 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 upsampled 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.

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 $h'_q \approx k h_q$ derived in Section 3.1 should apply. In general, however, downsampling results in loss of high
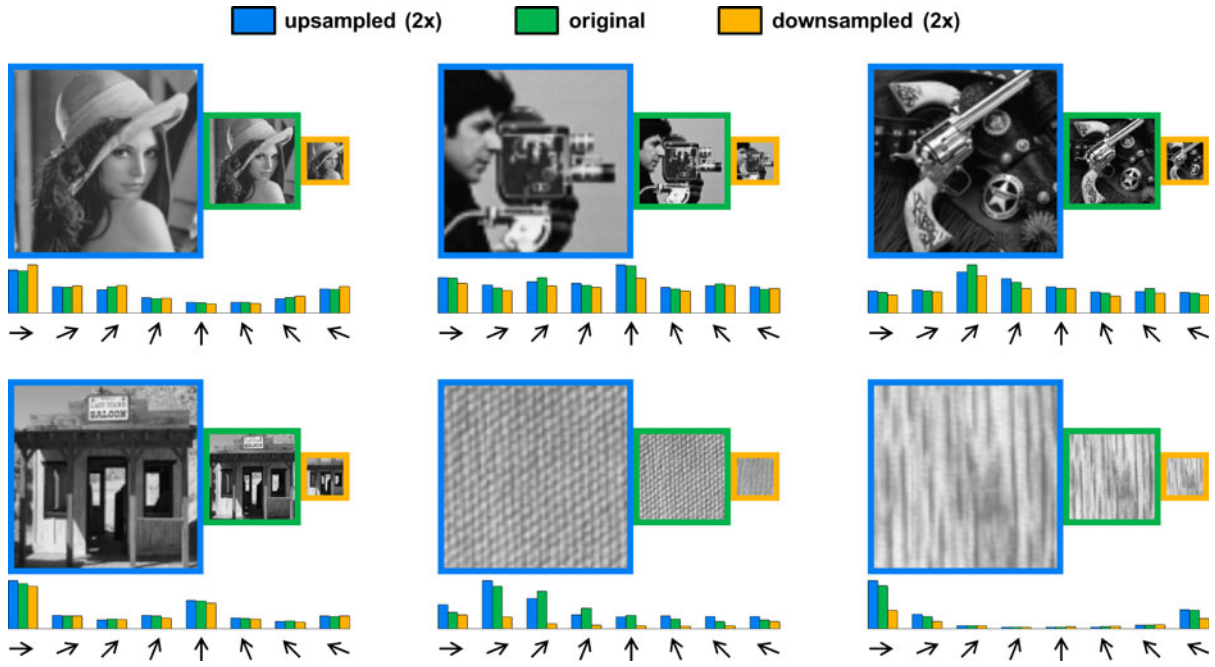
Fig. 2. *Approximating gradient histograms in images resampled by a factor of two.* For each image set, we take the original image (green border) and generate an upsampled (blue) and downsampled (orange) version. At each scale we compute a gradient histogram with eight bins, multiplying each bin by 0.5 and $1/0.34$ in the upsampled and downsampled histogram, respectively. Assuming the approximations from Section 3 hold, the three normalized gradient histograms should be roughly equal (the blue, green, and orange bars should have the same height at each orientation). For the first four cases, the approximations are fairly accurate. In the last two cases, showing highly structured Brodatz textures with significant high frequency content, the downsampling approximation fails. The first four images are representative, the last two are carefully selected to demonstrate images with atypical statistics.

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 Fig. 1b 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 $\mu \approx 0.34 < 0.5$ for natural images (and similarly $\mu \approx 0.33$ 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 Fig. 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 $\widetilde{M}$ defined as $\widetilde{M}(i,j) = M(i,j)/(\overline{M}(i,j) + 0.005)$, where $\overline{M}$ is the average gradient magnitude in each 11×11 image patch (computed by convolving M with an L1 normalized 11×11 triangle filter). Using the normalized gradient $\widetilde{M}$ gives improved results in

the context of object detection (see Section 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 $\widetilde{M}$ instead of M?

In Fig. 1c 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 ($\mu \approx 0.26$ and $\mu \approx 0.27$, respectively). Observe, however, that the expected value of r_q for normalized gradient histograms is quite different than for the original histograms (Fig. 1b).

Deriving analytical expressions governing the scaling properties of progressively more complex feature types would be difficult or even impossible. Instead, in Section 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 Ω 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 $C = \Omega(I)$: $f_\Omega(I) = \sum_{ijk} w_{ijk} C(i,j,k)$. 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 Ω 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 upsampled 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[1]

$$f_\Omega(I_s) \equiv \frac{1}{h_s w_s k} \sum_{ijk} C_s(i,j,k) \text{ where } C_s = \Omega(I_s). \quad (2)$$

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 Ω.

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 $E[\phi(I_{s_1})]$ to $E[\phi(I_{s_2})]$, 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[2]:

$$E[\phi(I_{s_1})]/E[\phi(I_{s_2})] = (s_1/s_2)^{-\lambda_\phi}. \quad (3)$$

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

We observe that a single image can itself be considered an ensemble of image patches (smaller images). Since Ω 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 Eq. (3) can be applied directly for a single image. We formalize this below.

We can decompose an image I into K smaller images $I^1 \dots I^K$ such that $I = [I^1 \cdots I^K]$. Given that Ω must be shift invariant and ignoring boundary effects gives $\Omega(I) = \Omega([I^1 \cdots I^K]) \approx [\Omega(I^1) \cdots \Omega(I^K)]$, and substituting into Eq. (2) yields $f_\Omega(I) \approx \Sigma f_\Omega(I^k)/K$. However, we can consider $I^1 \cdots I^K$ 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 Eq. (3) yields:

$$\boxed{f_\Omega(I_{s_1})/f_\Omega(I_{s_2}) = (s_1/s_2)^{-\lambda_\Omega} + \mathcal{E},} \quad (4)$$

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

In Section 4.2 we show that on average Eq. (4) provides a remarkably good fit for multiple channel types and image sets (i.e., we can fit λ_Ω such that $E[\mathcal{E}] \approx 0$). Additionally, experiments in Section 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 λ

We perform a series of experiments to verify Eq. (4) and estimate λ_Ω for numerous channel types Ω.

To estimate λ_Ω for a given Ω, we first compute:

$$\mu_s = \frac{1}{N} \sum_{i=1}^{N} f_\Omega(I^i_s)/f_\Omega(I^i_1), \quad (5)$$

for N images I^i and multiple values of $s < 1$, where I^i_s is obtained by downsampling $I^i_1 = I^i$. We use two image ensembles, one of $N = 1,237$ pedestrian images and one of $N = 4,280$ natural images (for details see Section 3.1). According to Eq. (4), $\mu_s = s^{-\lambda_\Omega} + E[\mathcal{E}]$. Our goal is to fit λ_Ω accordingly and verify the fidelity of Eq. (4) for various channel types Ω (i.e., verify that $E[\mathcal{E}] \approx 0$).

For each Ω, we measure μ_s according to Eq. (5) across three octaves with eight scales per octave for a total of 24 measurements at $s = 2^{-\frac{1}{8}}, \dots, 2^{-\frac{24}{8}}$. Since image dimensions are rounded to the nearest integer, we compute and use $s' = \sqrt{h_s w_s / hw}$, where $h \times w$ and $h_s \times w_s$ are the dimensions of the original and downsampled images, respectively.

In Fig. 3 we plot μ_s versus s' using a log-log plot for six channel types for both the pedestrian and natural images.[3] In all cases μ_s follows a power law with all measurements falling along a line on the log-log plots, as predicted. However, close inspection shows μ_s does not start exactly at 1 as expected: downsampling introduces a minor amount of blur even for small downsampling factors. We thus expect

---

1. The definition of $f_\Omega(I_s)$ in Eq. (2) differs from our previous definition in [39], where $f(I,s)$ denoted the channel *sum* after resampling by 2^s. The new definition and notation allow for a cleaner derivation, and the exponential scaling law becomes a more intuitive power law.

2. Let $F(s) = E[\phi(I_s)]$. We can rewrite the observation by saying there exists a function R such that $F(s_1)/F(s_2) = R(s_1/s_2)$. Applying repeatedly gives $F(s_1)/F(1) = R(s_1)$, $F(1)/F(s_2) = R(1/s_2)$, and $F(s_1)/F(s_2) = R(s_1/s_2)$. 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 λ [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 Eq. (9) in [67]).

3. Fig. 3 generalizes the results shown in Fig. 1. However, by switching from channel sums to channel means, $\mu_{1/2}$ in Figs. 3a and 3b is $4\times$ larger than μ in Figs. 1b and 1c, respectively.
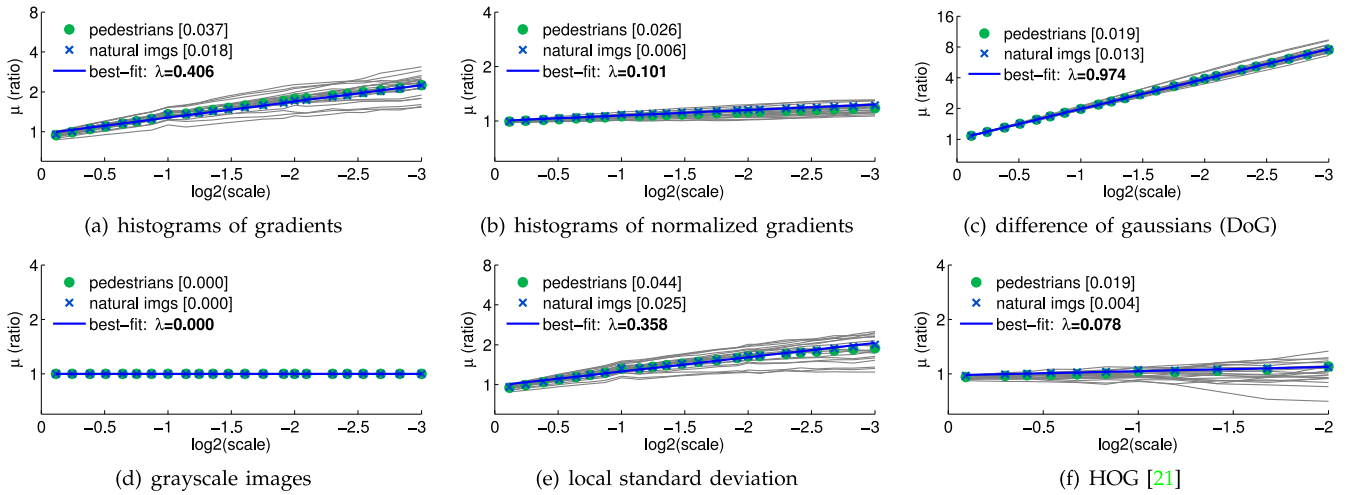
Fig. 3. *Power law feature scaling.* For each of six channel types we plot $\mu_s = \frac{1}{N}\sum f_\Omega(I_s^i)/f_\Omega(I_1^i)$ for $s \approx 2^{-\frac{1}{8}}, \ldots, 2^{-\frac{24}{8}}$ on a log-log plot for both pedestrian and natural image ensembles. Plots of $f_\Omega(I_{s_1})/f_\Omega(I_{s_2})$ for 20 randomly selected pedestrian images are shown as faint gray lines. Additionally the best-fit line to μ_s for the natural images is shown. The resulting λ and expected error $|E[\mathcal{E}]|$ are given in the plot legends. In all cases the μ_s follow a power law as predicted by Eq. (4) and are nearly identical for both pedestrian and natural images, showing the estimate of λ is robust and generally applicable. The tested channels are: (a) histograms of gradients described in Section 3; (b) histograms of normalized gradients described in Section 3.3; (c) a difference of gaussian (DoG) filter (with inner and outer σ of 0.71 and 1.14, respectively); (d) grayscale images (with $\lambda = 0$ as expected); (e) pixel standard deviation computed over local 5×5 neighborhoods $C(i,j)=\sqrt{E[I(i,j)^2]-E[I(i,j)]}$; (f) HOG [21] with 4×4 spatial bins (results were averaged over HOG's 36 channels). Code for generating such plots is available (see chnsScaling.m in Piotr's Toolbox).

μ_s to have the form $\mu_s = a_\Omega s^{-\lambda_\Omega}$, with $a_\Omega \neq 1$ as an artifact of the interpolation. Note that a_Ω is only necessary for estimating λ_Ω from downsampled images and is not used subsequently. To estimate a_Ω and λ_Ω, we use a least squares fit of $\log_2(\mu_{s'}) = a'_\Omega - \lambda_\Omega \log_2(s')$ to the 24 measurements computed over natural images (and set $a_\Omega = 2^{a'_\Omega}$). Resulting estimates of λ_Ω are given in plot legends in Fig. 3.

There is strong agreement between the resulting best-fit lines and the observations. In legend brackets in Fig. 3 we report expected error $|E[\mathcal{E}]| = |\mu_s - a_\Omega s^{-\lambda_\Omega}|$ for both natural and pedestrian images averaged over s (using a_Ω and λ_Ω 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 Eq. (4) is an excellent fit to the observations μ_s for both image ensembles.

The derivation of Eq. (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 λ_Ω is robust and generally applicable.

4.3 Deviation for Individual Images

In Section 4.2 we verified that Eq. (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 Section 6.

Plots of $f_\Omega(I_{s_1})/f_\Omega(I_{s_2})$ for randomly selected images are shown as faint gray lines in Fig. 3. The individual curves are relatively smooth and diverge only somewhat from the best-fit line. We quantify their deviation by defining σ_s analogously to μ_s in Eq. (5):

$$\sigma_s = \text{stdev}[f_\Omega(I_s^i)/f_\Omega(I_1^i)] = \text{stdev}[\mathcal{E}], \qquad (6)$$

where 'stdev' 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 Eq. (4). In Section 4.2 we confirmed that $E[\mathcal{E}] \approx 0$, our goal now is to understand how $\sigma_s = \text{stdev}[\mathcal{E}] \approx \sqrt{E[\mathcal{E}^2]}$ behaves.

In Fig. 4 we plot σ_s as a function of s for the same channels as in Fig. 3. In legend brackets we report σ_s for $s = \frac{1}{2}$ for both natural and pedestrian images; for all channels studied $\sigma_{1/2} < .2$. In all cases σ_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 (Fig. 4b) have lower σ_s than their unnormalized counterparts (Fig. 4a). The trivial grayscale channel (Fig. 4d) has $\sigma_s = 0$ as the approximation is exact.

Observe that often σ_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 Ω (e.g., simple gradient histograms) resulting in more variance in the ratio in Eq. (4). For HOG channels (Fig. 4f), 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 Fig. 5a, 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 128×64 windows. In Fig. 5b we plot the effect of varying the window size. While $\mu_{1/2}$ remains relatively constant,
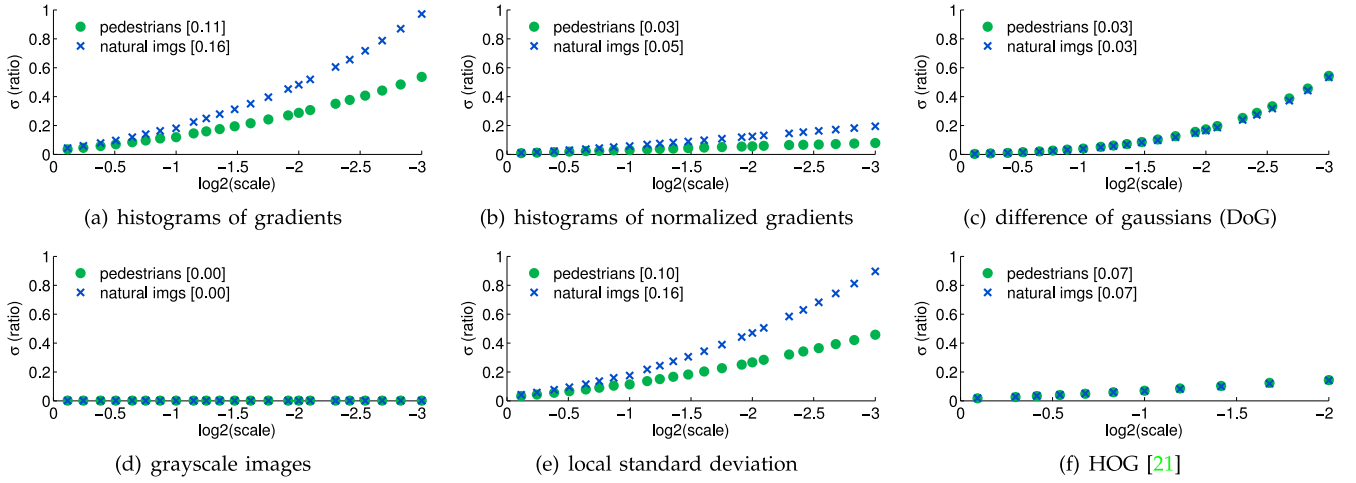
(a) histograms of gradients   (b) histograms of normalized gradients   (c) difference of gaussians (DoG)

(d) grayscale images   (e) local standard deviation   (f) HOG [21]

Fig. 4. *Power law deviation for individual images*. For each of the six channel types described in Fig. 3 we plot σ_s versus s where $\sigma_s = \sqrt{E[\mathcal{E}^2]}$ and $\mathcal{E}$ is the deviation from the power law for a single image as defined in Eq. (4). In brackets we report $\sigma_{1/2}$ for both natural and pedestrian images. σ_s increases gradually as a function of s, meaning that not only does Eq. (4) hold for an *ensemble* of images but also the deviation from the power law for *individual* images is low for small s.

$\sigma_{1/2}$ increases with decreasing window size (see also the derivation of Eq. (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 Ω's considered this was the case after windows of near uniform intensity were excluded (see Section 3.1). Alternatively, we have found that excluding I with $f_\Omega(I) \approx 0$ when estimating λ 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, σ will be large. Such channels may not be good candidates for the power law approximation.

One-shot estimates. We can estimate λ as described in Section 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 4,280 single image estimates of λ 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).

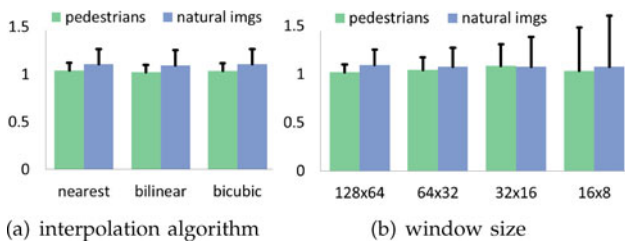(a) interpolation algorithm   (b) window size

Fig. 5. *Effect of the interpolation algorithm and window size on channel scaling.* We plot $\mu_{1/2}$ (bar height) and $\sigma_{1/2}$ (error bars) for normalized gradient histograms (see Section 3.3) . (a) Varying the interpolation algorithm for resampling does not have a major effect on either $\mu_{1/2}$ or $\sigma_{1/2}$. (b) Decreasing window size leaves $\mu_{1/2}$ relatively unchanged but results in increasing $\sigma_{1/2}$.

5 FAST FEATURE PYRAMIDS

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

5.1 Feature Channel Scaling

We propose an extension of the power law governing feature scaling introduced in Section 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:

$$C_s \approx R(C, s) \cdot s^{-\lambda_\Omega}. \tag{7}$$

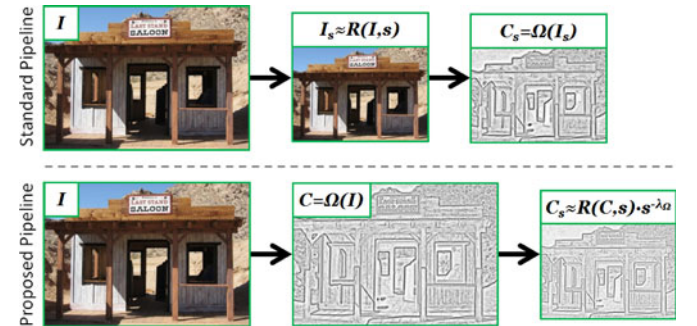A visual demonstration of Eq. (7) is shown in Fig. 6.

Fig. 6. *Feature channel scaling.* Suppose 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 Section 4, we propose to approximate C_s by $R(C, s) \cdot s^{-\lambda_\Omega}$. This approach is simple, general, and accurate, and allows for fast feature pyramid construction.

Eq. (7) follows from Eq. (4). Setting $s_1 = s$, $s_2 = 1$, and rearranging Eq. (4) gives $f_\Omega(I_s) \approx f_\Omega(I)s^{-\lambda_\Omega}$. 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:

$$f_\Omega(I_s^{w_s}) \approx f_\Omega(I^w)s^{-\lambda_\Omega}$$

$$\frac{1}{|w_s|}\sum_{i,j \in w_s} C_s(i,j) \approx \frac{1}{|w|}\sum_{i,j \in w} C(i,j)s^{-\lambda_\Omega}$$

$$C_s \approx R(C,s)s^{-\lambda_\Omega}.$$

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

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

A simple strategy for aggregating over multiple pixels and thus improving robustness is to downsample and/or smooth C_s relative to I_s (each pixel in the resulting channel will be a weighted sum of pixels in the original full resolution channel). Downsampling C_s also allows for faster pyramid construction (we return to this in Section 5.2). For object detection, we typically downsample channels by $4\times$ to $8\times$ (e.g., HOG [21] uses 8×8 bins).

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 four 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 Fig. 7 (top).

The approximation in Eq. (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 ($s \in \{1, \frac{1}{2}, \frac{1}{4}, \ldots\}$). At intermediate scales, C_s is computed using $C_s \approx R(C_{s'}, s/s')(s/s')^{-\lambda_\Omega}$ where $s' \in \{1, \frac{1}{2}, \frac{1}{4}, \ldots\}$ is the nearest scale for which we have $C_{s'} = \Omega(I_{s'})$, see Fig. 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 Ω is within 33 percent of computing $\Omega(I)$ at the original scale (see Section 5.3) and channels do not need to be approximated beyond half an octave (keeping error low, see Section 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 Section 5.1, evaluating $R(C, s)$ is faster than $R(I, s)$.

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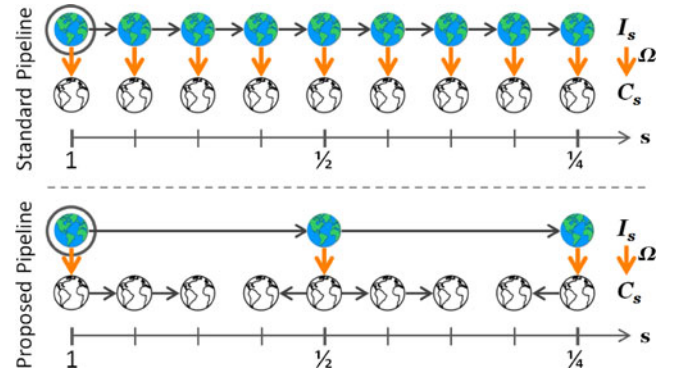

Fig. 7. *Fast feature pyramids*. Color and grayscale icons represent images and channels; horizontal and vertical arrows denote computation of R and Ω. *Top.* The standard pipeline for constructing a feature pyramid requires computing $I_s = R(I, s)$ followed by $C_s = \Omega(I_s)$ for every s. This is costly. *Bottom.* We propose computing $I_s = R(I, s)$ and $C_s = \Omega(I_s)$ for only a sparse set of s (once per octave). Then, at intermediate scales C_s is computed using the approximation in Eq. (7): $C_s \approx R(C_{s'}, s/s')(s/s')^{-\lambda_\Omega}$ where s' is the nearest scale for which we have $C_{s'} = \Omega(I_{s'})$. In the proposed scheme, the number of computations of R is constant while (more expensive) computations of Ω are reduced considerably.

5.3 Complexity Analysis

The computational savings of computing approximate feature pyramids is significant. Assume the cost of computing Ω 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

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

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 Ω once per octave ($m = 1$). The total cost is $\frac{4}{3}n^2$, which is only 33 percent more than the cost of computing single scale features. Typical detectors are evaluated on eight to 12 scales per octave [31], thus according to (8) we achieve an order of magnitude savings over computing Ω 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 Section 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 Section 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 Section 6.3 we approximate HOG feature pyramids for use with *deformable part models* [35].

6.1 Aggregated Channel Features

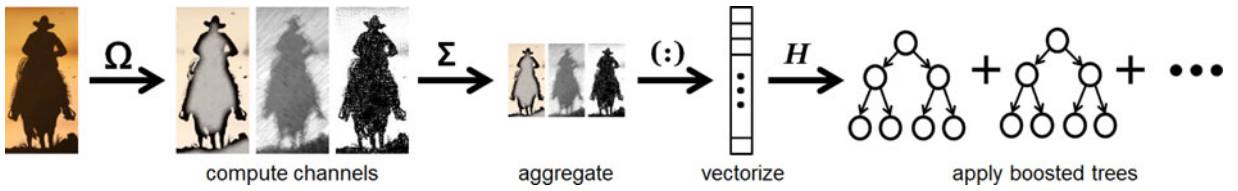The ACF detection framework is conceptually straightforward (Fig. 8). Given an input image I, we compute several

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.

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 (six 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 4×4 blocks and pixels in each block are summed. Finally the channels are smoothed, again with a [1 2 1]/4 filter. For 640×480 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 640×480 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 Section 5) with seven of eight scales per octave approximated runs at nearly *50 fps*.

Detector. For pedestrian detection, AdaBoost [69] is used to train and combine 2,048 depth-two trees over the $128 \cdot 64 \cdot 10/16 = 5{,}120$ candidate features (channel pixel lookups) in each 128×64 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 four pixels and eight scales per octave. For 640×480 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.[4] 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 data sets: INRIA [21], Caltech [31], TUD-Brussels [36] and ETH [37].

4. Code: http://vision.ucsd.edu/ pdollar/toolbox/doc/index.html.

MRs for 16 competing methods are shown. ACF outperforms competing approaches on nearly all datasets. When averaged over the four data sets, the MR of ACF is 40 percent with exact feature pyramids and 41 percent with fast feature pyramids, a negligible difference, demonstrating the effectiveness of our approach.

Speed. MR versus speed for numerous detectors is shown in Fig. 10. ACF with fast feature pyramids runs at ~32 fps. The only two faster approaches are Crosstalk cascades [40] and the VeryFast detector from Benenson et al. [30]. Their additional speedups are based on improved cascade strategies and combining multi-resolution models with a GPU implementation, respectively, and are orthogonal to the gains achieved by using approximate multiscale features. Indeed, all the detectors that run at 5 fps and higher exploit the power law governing feature scaling.

Pyramid parameters. Detection performance on INRIA [21] with fast feature pyramids under varying settings is shown in Fig. 11. The key result is given in Fig. 11a: when approximating seven of eight scales per octave, the MR for ACF is 0.169 which is virtually identical to the MR of 0.166 obtained

TABLE 1
MRs of Leading Approaches for Pedestrian Detection on Four Data Sets

	INRIA [21]	Caltech [31]	TUD [36]	ETH [37]	MEAN
Shapelet [70]	82	91	95	91	90
VJ [27]	72	95	95	90	88
PoseInv [71]	80	86	88	92	87
HikSvm [72]	43	73	83	72	68
HOG [21]	46	68	78	64	64
HogLbp [73]	39	68	82	55	61
MF [74]	36	68	73	60	59
PLS [75]	40	62	71	55	57
MF+CSS [76]	25	61	60	61	52
MF+Motion [76]	–	51	55	60	–
LatSvmV2 [35]	20	63	70	51	51
FPDW [39]	21	57	63	60	50
ChnFtrs [29]	22	56	60	57	49
Crosstalk [40]	19	54	58	52	46
VeryFast [30]	**16**	–	–	55	–
MultiResC [77]	–	48	–	–	–
ICF-Exact §6.2	18	48	53	**50**	42
ICF §6.2	19	51	55	56	45
ACF-Exact §6.1	17	**43**	**50**	**50**	**40**
ACF §6.1	17	45	52	51	41

For ICF and ACF exact and approximate detection results are shown with only small differences between them. For the latest pedestrian detection results please see [32].
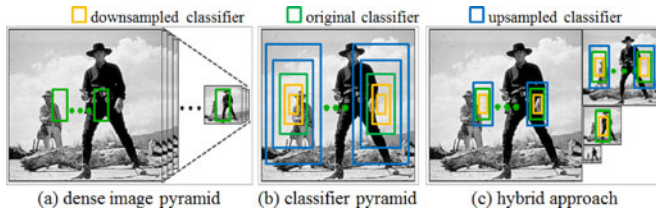
Fig. 9. (a) A standard pipeline for performing multiscale detection is to create a densely sampled feature pyramid. (b) Viola and Jones [27] used simple shift and scale invariant features, allowing a detector to be placed at any location and scale without relying on a feature pyramid. (c) ICF can use a hybrid approach of constructing an octave-spaced feature pyramid followed by approximating detector responses within half an octave of each pyramid level.

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 Fig. 11b. Finally, we observe that at least eight scales per octave must be used for good performance (Fig. 11c), making the proposed scheme crucial for achieving detection results that are both fast and accurate.

6.2 Integral Channel Features

Integral channel features [29] are a precursor to the ACF framework described in Section 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 ICF averaged over the four data sets is 42 percent with exact feature pyramids and 45 percent with fast feature pyramids. The gap of 3 percent is larger than the 1 percent gap for ACF but still small. With fast feature pyramids ICF runs at ∼16 fps, see Fig. 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 Fig. 12. The plots mirror the results shown in Fig. 11 for ACF. The key result is given in Fig. 12a: when approximating seven of eight scales per octave, the MR for ICF is 2 percent worse than the MR obtained with exact feature pyramids.

The ICF framework allows for an alternate application of the power law governing feature scaling: instead of rescaling channels as discussed in Section 5, one can instead rescale the *detector*. Using the notation from Section 4, rectangular channel sums (features used in ICF) can be written as $Af_{\Omega}(I)$, where A denotes rectangle area. As such, Eq. (4) can be applied to approximate features at nearby scales and given integral channel images computed at one scale, detector responses can be approximated at nearby scales. This operation can be implemented by rescaling the detector itself, see [39]. As the approximation degrades with increasing scale offsets, a hybrid approach is to construct an octave-spaced feature pyramid followed by approximating detector responses at nearby scales, see Fig. 9. This approach was extended in [30].

6.3 Deformable Part Models

Deformable part models from Felzenszwalb et al. [35] are an elegant approach for general object detection that have consistently achieved top results on the PASCAL VOC challenge [38]. DPMs use a variant of HOG features [21] as their image representation, followed by classification with linear SVMs. An object model is composed of multiple parts, a root model, and optionally multiple mixture components. For details see [35].

Recent approaches for increasing the speed of DPMs include work by Felzenszwalb et al. [44] and Pedersoli et al. [45] on cascaded and coarse-to-fine deformable part
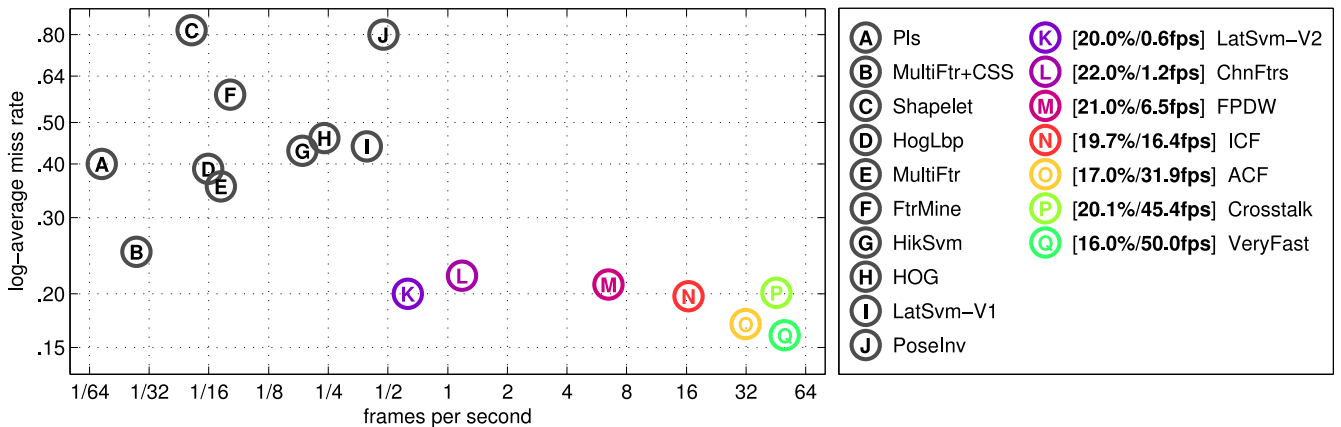

Fig. 10. Log-average miss rate on the INRIA pedestrian data set [21] versus frame rate on 640×480 images for multiple detectors. Method runtimes were obtained from [31], see also [31] for citations for detectors A-L. Numbers in brackets indicate MR/fps for select approaches, sorted by speed. *All detectors 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 [Section 6.2], (O) ACF [Section 6.1], (P) crosstalk cascades [40], and (Q) the Very-Fast 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.
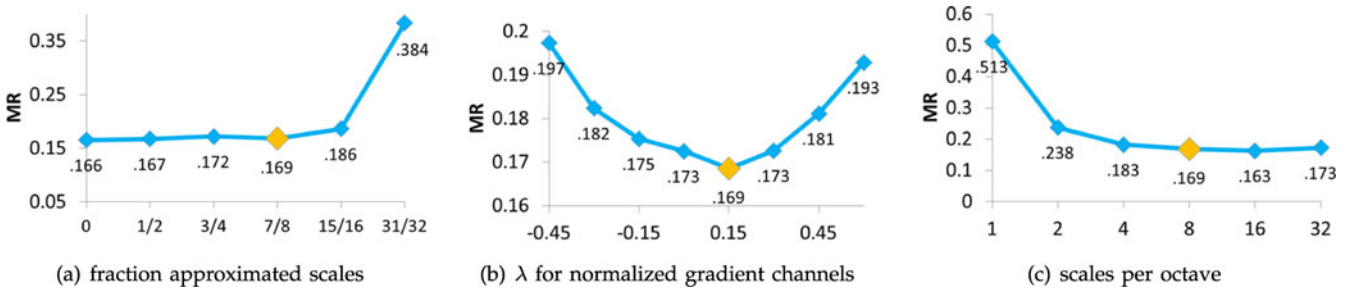
Fig. 11. *Effect of parameter setting of fast feature pyramids on the ACF detector* [Section 6.1]. We report log-average miss rate averaged over 25 trials on the INRIA pedestrian data set [21]. Orange diamonds denote default parameter settings: 7/8 scales approximated per octave, $\lambda \approx 0.17$ for the normalized gradient channels, and eight 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 λ when approximating the normalized gradient channels cause a marked decrease in performance. (c) At least eight scales per octave are necessary for good performance, making the proposed scheme crucial for achieving detection results that are both fast and accurate.

models, respectively. Our work is complementary as we focus on improving the speed of pyramid construction. The current bottleneck of DPMs is in the classification stage, therefore pyramid construction accounts for only a fraction of total runtime. However, if fast feature pyramids are coupled with optimized classification schemes [44], [45], DPMs have the potential to have more competitive runtimes. We focus on demonstrating DPMs can achieve good accuracy with fast feature pyramids and leave the coupling of fast feature pyramids and optimized classification schemes to practitioners.

DPM code is available online [35]. We tested pre-trained DPM models on the 20 PASCAL 2007 categories using exact HOG pyramids and HOG pyramids with nine of 10 scales per octave approximated using our proposed approach. Average precision (AP) scores for the two approaches, denoted DPM and ~DPM, respectively, are shown in Table 2. The mean AP across the 20 categories is 26.6 percent for DPMs and 24.5 percent for ~DPMs. Using fast HOG feature pyramids only decreased mean AP 2 percent, demonstrating the validity of the proposed approach.

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.

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.
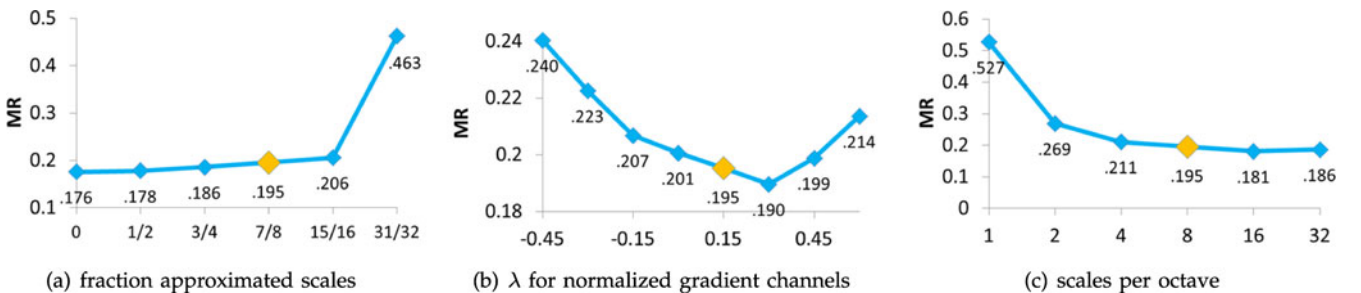

Fig. 12. *Effect of parameter setting of fast feature pyramids on the* ICF *detector* [Section 6.2]. The plots mirror the results shown in Fig. 11 for the ACF detector, although overall performance for ICF is slightly lower. (a) When approximating seven of every eight scales in the pyramid, the MR for ICF is 0.195 which is only slightly worse than the MR of 0.176 obtained using exact feature pyramids. (b) Computing approximate channels with an incorrect value of λ results in decreased performance (although using a slightly larger λ than predicted appears to improve results marginally). (c) Similarly to the ACF framework, at least eight scales per octave are necessary to achieve good results.

TABLE 2
Average Precision Scores for Deformable Part Models
with Exact (DPM) and Approximate ($\sim$DPM)
Feature Pyramids on PASCAL

	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow
DPM	26.3	59.4	2.3	10.2	21.2	46.2	52.2	7.9	15.9	17.4
$\sim$DPM	24.1	54.7	1.6	9.8	20.0	42.1	50.1	8.0	13.8	16.7

|	table	dog	horse	moto	person	plant	sheep	sofa	train	tv
DPM	10.9	2.9	53.4	37.6	38.2	4.9	16.6	29.7	38.2	40.8
$\sim$DPM	8.9	2.5	49.4	38.3	36.0	4.2	14.9	24.4	35.8	35.0

ACKNOWLEDGMENTS

REFERENCES

[1] D. Hubel and T. Wiesel, "Receptive Fields and Functional Architecture of Monkey Striate Cortex," *J. Physiology*, vol. 195, pp. 215-243, 1968.
[2] C. Malsburg, "Self-Organization of Orientation Sensitive Cells in the Striate Cortex," *Biological Cybernetics*, vol. 14, no. 2, pp. 85-100, 1973.
[3] L. Maffei and A. Fiorentini, "The Visual Cortex as a Spatial Frequency Analyser," *Vision Research*, vol. 13, no. 7, pp. 1255-1267, 1973.
[4] P. Burt and E. Adelson, "The Laplacian Pyramid as a Compact Image Code," *IEEE Trans. Comm.*, vol. 31, no. 4, pp. 532-540, Apr. 1983.
[5] J. Daugman, "Uncertainty Relation for Resolution in Space, Spatial Frequency, and Orientation Optimized by Two-Dimensional Visual Cortical Filters," *J. Optical Soc. Am. A: Optics, Image Science, and Vision*, vol. 2, pp. 1160-1169, 1985.
[6] J. Koenderink and A. Van Doorn, "Representation of Local Geometry in the Visual System," *Biological Cybernetics*, vol. 55, no. 6, pp. 367-375, 1987.
[7] D.J. Field, "Relations between the Statistics of Natural Images and the Response Properties of Cortical Cells," *J. Optical Soc. Am. A: Optics, Image Science, and Vision*, vol. 4, pp. 2379-2394, 1987.
[8] S. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 647-693, July 1989.
[9] P. Vaidyanathan, "Multirate Digital Filters, Filter Banks, Polyphase Networks, and Applications: A Tutorial," *Proc. IEEE*, vol. 78, no. 1, pp. 56-93, Jan. 1990.
[10] M. Vetterli, "A Theory of Multirate Filter Banks," *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. 35, no. 3, pp. 356-372, Mar. 1987.
[11] E. Simoncelli and E. Adelson, "Noise Removal via Bayesian Wavelet Coring," *Proc. Int'l Conf. Image Processing (ICIP)*, vol. 1, 1996.
[12] W.T. Freeman and E.H. Adelson, "The Design and Use of Steerable Filters," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, no. 9, pp. 891-906, Sept. 1991.
[13] J. Malik and P. Perona, "Preattentive Texture Discrimination with Early Vision Mechanisms," *J. Optical Soc. Am. A*, vol. 7, pp. 923-932, May 1990.
[14] D. Jones and J. Malik, "Computational Framework for Determining Stereo Correspondence from a Set of Linear Spatial Filters," *Image and Vision Computing*, vol. 10, no. 10, pp. 699-708, 1992.
[15] E. Adelson and J. Bergen, "Spatiotemporal Energy Models for the Perception of Motion," *J. Optical Soc. Am. A*, vol. 2, no. 2, pp. 284-299, 1985.
[16] Y. Weiss and E. Adelson, "A Unified Mixture Framework for Motion Segmentation: Incorporating Spatial Coherence and Estimating the Number of Models," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 1996.
[17] L. Itti, C. Koch, and E. Niebur, "A Model of Saliency-Based Visual Attention for Rapid Scene Analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254-1259, Nov. 1998.
[18] P. Perona and J. Malik, "Detecting and Localizing Edges Composed of Steps, Peaks and Roofs," *Proc. Third IEEE Int'l Conf. Computer Vision (ICCV)*, 1990.
[19] M. Lades, J. Vorbruggen, J. Buhmann, J. Lange, C. von der Malsburg, R. Wurtz, and W. Konen, "Distortion Invariant Object Recognition in the Dynamic Link Architecture," *IEEE Trans. Computers*, vol. 42, no. 3, pp. 300-311, Mar. 1993.
[20] D.G. Lowe, "Object Recognition from Local Scale-Invariant Features," *Proc. Seventh IEEE Int'l Conf. Computer Vision (ICCV)*, 1999.
[21] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2005.
[22] R. De Valois, D. Albrecht, and L. Thorell, "Spatial Frequency Selectivity of Cells in Macaque Visual Cortex," *Vision Research*, vol. 22, no. 5, pp. 545-559, 1982.
[23] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, "Gradient-Based Learning Applied to Document Recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.
[24] M. Riesenhuber and T. Poggio, "Hierarchical Models of Object Recognition in Cortex," *Nature Neuroscience*, vol. 2, pp. 1019-1025, 1999.
[25] D.G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int'l J. Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.
[26] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet Classification with Deep Convolutional Neural Networks," *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2012.
[27] P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2001.
[28] P. Viola, M. Jones, and D. Snow, "Detecting Pedestrians Using Patterns of Motion and Appearance," *Int'l J. Computer Vision*, vol. 63, no. 2, pp. 153-161, 2005.
[29] P. Dollár, Z. Tu, P. Perona, and S. Belongie, "Integral Channel Features," *Proc. British Machine Vision Conf. (BMVC)*, 2009.
[30] R. Benenson, M. Mathias, R. Timofte, and L. Van Gool, "Pedestrian Detection at 100 Frames per Second," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2012.
[31] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian Detection: An Evaluation of the State of the Art," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 743–761, 2012.
[32] www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/, 2014.
[33] D.L. Ruderman and W. Bialek, "Statistics of Natural Images: Scaling in the Woods," *Physical Rev. Letters*, vol. 73, no. 6, pp. 814-817, Aug. 1994.
[34] E. Switkes, M. Mayer, and J. Sloan, "Spatial Frequency Analysis of the Visual Environment: Anisotropy and the Carpentered Environment Hypothesis," *Vision Research*, vol. 18, no. 10, pp. 1393-1399, 1978.
[35] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part Based Models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627-1645, Sept. 2010.
[36] C. Wojek, S. Walk, and B. Schiele, "Multi-Cue Onboard Pedestrian Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2009.
[37] A. Ess, B. Leibe, and L. Van Gool, "Depth and Appearance for Mobile Scene Analysis," *Proc. IEEE 11th Int'l Conf. Computer Vision (ICCV)*, 2007.
[38] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes (VOC) Challenge," *Int'l J. Computer Vision*, vol. 88, no. 2, pp. 303-338, June 2010.
[39] P. Dollár, S. Belongie, and P. Perona, "The Fastest Pedestrian Detector in the West," *Proc. British Machine Vision Conf. (BMVC)*, 2010.
[40] P. Dollár, R. Appel, and W. Kienzle, "Crosstalk Cascades for Frame-Rate Pedestrian Detection," *Proc. 12th European Conf. Computer Vision (ECCV)*, 2012.

[41] T. Lindeberg, "Scale-Space for Discrete Signals," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 3, pp. 234-254, Mar. 1990.

[42] J.L. Crowley, O. Riff, and J.H. Piater, "Fast Computation of Characteristic Scale Using a Half-Octave Pyramid," *Proc. Fourth Int'l Conf. Scale-Space Theories in Computer Vision*, 2002.

[43] R.S. Eaton, M.R. Stevens, J.C. McBride, G.T. Foil, and M.S. Snorrason, "A Systems View of Scale Space," *Proc. IEEE Int'l Conf. Computer Vision Systems (ICVS)*, 2006.

[44] P. Felzenszwalb, R. Girshick, and D. McAllester, "Cascade Object Detection with Deformable Part Models," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2010.

[45] M. Pedersoli, A. Vedaldi, and J. Gonzalez, "A Coarse-to-Fine Approach for Fast Deformable Object Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2011.

[46] C.H. Lampert, M.B. Blaschko, and T. Hofmann, "Efficient Subwindow Search: A Branch and Bound Framework for Object Localization," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2129-2142, Dec. 2009.

[47] L. Bourdev and J. Brandt, "Robust Object Detection via Soft Cascade," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2005.

[48] C. Zhang and P. Viola, "Multiple-Instance Pruning for Learning Efficient Cascade Detectors," *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2007.

[49] J. Šochman and J. Matas, "Waldboost—Learning for Time Constrained Sequential Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2005.

[50] H. Masnadi-Shirazi and N. Vasconcelos, "High Detection-Rate Cascades for Real-Time Object Detection," *Proc. IEEE 11th Int'l Conf. Computer Vision (ICCV)*, 2007.

[51] F. Fleuret and D. Geman, "Coarse-To-Fine Face Detection," *Int'l J. Computer Vision*, vol. 41, no. 1/2, pp. 85-107, 2001.

[52] P. Felzenszwalb and D. Huttenlocher, "Efficient Matching of Pictorial Structures," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2000.

[53] C. Papageorgiou and T. Poggio, "A Trainable System for Object Detection," *Int'l J. Computer Vision*, vol. 38, no. 1, pp. 15-33, 2000.

[54] M. Weber, M. Welling, and P. Perona, "Unsupervised Learning of Models for Recognition," *Proc. European Conf. Computer Vision (ECCV)*, 2000.

[55] S. Agarwal and D. Roth, "Learning a Sparse Representation for Object Detection," *Proc. European Conf. Computer Vision (ECCV)*, 2002.

[56] R. Fergus, P. Perona, and A. Zisserman, "Object Class Recognition by Unsupervised Scale-Invariant Learning," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2003.

[57] B. Leibe, A. Leonardis, and B. Schiele, "Robust Object Detection with Interleaved Categorization and Segmentation," *Int'l J. Computer Vision*, vol. 77, no. 1-3, pp. 259-289, May 2008.

[58] C. Gu, J.J. Lim, P. Arbeláez, and J. Malik, "Recognition Using Regions," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2009.

[59] B. Alexe, T. Deselaers, and V. Ferrari, "What Is an Object?" *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2010.

[60] C. Wojek, G. Dorkó, A. Schulz, and B. Schiele, "Sliding-Windows for Rapid Object Class Localization: A Parallel Technique," *Proc. 30th DAGM Symp. Pattern Recognition*, 2008.

[61] L. Zhang and R. Nevatia, "Efficient Scan-Window Based Object Detection Using GPGPU," *Proc. Workshop Visual Computer Vision on GPU's (CVGPU)*, 2008.

[62] B. Bilgic, "Fast Human Detection with Cascaded Ensembles," master's thesis, MIT, Feb. 2010.

[63] Q. Zhu, S. Avidan, M. Yeh, and K. Cheng, "Fast Human Detection Using a Cascade of Histograms of Oriented Gradients," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2006.

[64] F.M. Porikli, "Integral Histogram: A Fast Way to Extract Histograms in Cartesian Spaces," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2005.

[65] A. Ess, B. Leibe, K. Schindler, and L. Van Gool, "Robust Multi-Person Tracking from a Mobile Platform," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 31, no. 10, pp. 1831-1846, Oct. 2009.

[66] M. Bajracharya, B. Moghaddam, A. Howard, S. Brennan, and L.H. Matthies, "A Fast Stereo-Based System for Detecting and Tracking Pedestrians from a Moving Vehicle," *Int'l J. Robotics Research*, vol. 28, pp. 1466-1485, 2009.

[67] D.L. Ruderman, "The Statistics of Natural Images," *Network: Computation in Neural Systems*, vol. 5, no. 4, pp. 517-548, 1994.

[68] S.G. Ghurye, "A Characterization of the Exponential Function," *The Am. Math. Monthly*, vol. 64, no. 4, pp. 255-257, 1957.

[69] J. Friedman, T. Hastie, and R. Tibshirani, "Additive Logistic Regression: A Statistical View of Boosting," *Annals of Statistics*, vol. 38, no. 2, pp. 337-374, 2000.

[70] P. Sabzmeydani and G. Mori, "Detecting Pedestrians by Learning Shapelet Features," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2007.

[71] Z. Lin and L.S. Davis, "A Pose-Invariant Descriptor for Human Detection and Segmentation," *Proc. 10th European Conf. Computer Vision (ECCV)*, 2008.

[72] S. Maji, A. Berg, and J. Malik, "Classification Using Intersection Kernel SVMs Is Efficient," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2008.

[73] X. Wang, T.X. Han, and S. Yan, "An HOG-LBP Human Detector with Partial Occlusion Handling," *Proc. IEEE Int'l Conf. Computer Vision (ICCV)*, 2009.

[74] C. Wojek and B. Schiele, "A Performance Evaluation of Single and Multi-Feature People Detection," *Proc. 30th DAGM Symp. Pattern Recognition (DAGM)*, 2008.

[75] W. Schwartz, A. Kembhavi, D. Harwood, and L. Davis, "Human Detection Using Partial Least Squares Analysis," *Proc. IEEE 12th Int'l Conf. Computer Vision (ICCV)*, 2009.

[76] S. Walk, N. Majer, K. Schindler, and B. Schiele, "New Features and Insights for Pedestrian Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2010.

[77] D. Park, D. Ramanan, and C. Fowlkes, "Multiresolution Models for Object Detection," *Proc. 11th European Conf. Computer Vision (ECCV)*, 2010.

Piotr Dollár received the master's degree in computer science from Harvard University in 2002 and the PhD degree from the University of California, San Diego in 2007. He joined the Computational Vision Lab at the California Institute of Technology Caltech as a postdoctoral fellow in 2007. Upon being promoted to a senior postdoctoral fellow he realized it was time to move on, and in 2011, he joined the Interactive Visual Media Group at Microsoft Research, Redmond, Washington, where he currently resides. He has worked on object detection, pose estimation, boundary learning, and behavior recognition. His general interests include machine learning and pattern recognition, and their application to computer vision.
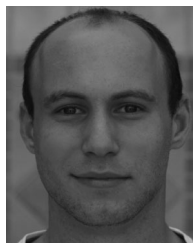
Ron Appel received the bachelor's and master's degrees in electrical and computer engineering from the University of Toronto in 2006 and 2008, respectively. He is working toward the PhD degree in the Computational VisionLab at the California Institute of Technology, Caltech, where he currently holds an NSERC Graduate Award. He cofounded 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 the BS (with honor) in electrical engineering from the California Institute of Technology, Caltech in 1995 and the PhD degree in electrical engineering and computer science from the University of California at Berkeley in 2000. While at the University of California at Berkeley, his research was supported by the US National Science Foundation (NSF) Graduate Research Fellowship. From 2001 to 2013, he was a professor in the Department of Computer Science and Engineering at the University of California, San Diego (UCSD). He is currently a professor at Cornell NYC Tech and the Cornell Computer Science Department, Ithaca, New York. His research interests include computer vision, machine learning, crowdsourcing, and human-in-the-loop computing. He is also a cofounder of several companies including Digital Persona, Anchovi Labs (acquired by Dropbox) and Orpix. He is a recipient of the US National Science Foundation (NSF) CAREER Award, the Alfred P. Sloan Research Fellowship, and the MIT Technology Review "Innovators Under 35" Award.

Pietro Perona received the graduate degree in electrical engineering from the Università di Padova in 1985 and the PhD degree in electrical engineering and computer science from the University of California at Berkeley in 1990. After a postdoctoral fellowship at MIT in 1990-1991 he joined the faculty of the California Institute of Technology, Caltech in 1991, where he is now an Allen E. Puckett professor of electrical engineering and computation and neural systems. His current interests include visual recognition, modeling vision in biological systems, modeling and measuring behavior, and Visipedia. He has worked on anisotropic diffusion, multiresolution-multiorientation filtering, human texture perception and segmentation, dynamic vision, grouping, analysis of human motion, recognition of object categories, and modeling visual search.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.