The Fastest Pedestrian Detector in the West

Piotr Dollár[1]
pdollar@caltech.edu
Serge Belongie[2]
sjb@cs.ucsd.edu
Pietro Perona[1]
perona@caltech.edu

[1] Dept. of Electrical Engineering
California Institute of Technology
Pasadena, CA, USA

[2] Dept. of Computer Science and Eng.
University of California, San Diego
San Diego, CA, USA

Abstract

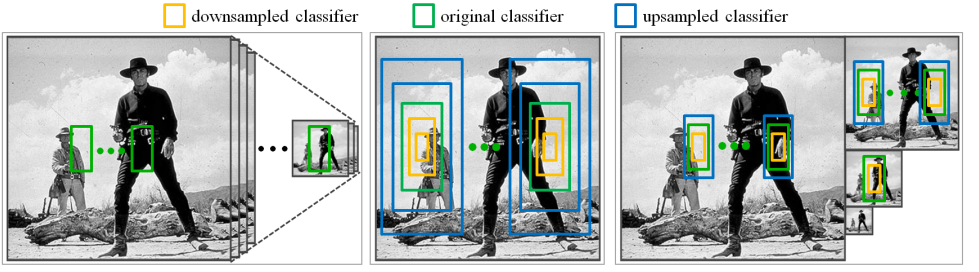
We demonstrate a multiscale pedestrian detector operating in near real time (∼6 fps on 640x480 images) with state-of-the-art detection performance. The computational bottleneck of many modern detectors is the construction of an image pyramid, typically sampled at 8-16 scales per octave, and associated feature computations at each scale. We propose a technique to avoid constructing such a finely sampled image pyramid without sacrificing performance: our key insight is that for a broad family of features, including gradient histograms, the feature responses computed at a single scale can be used to approximate feature responses at nearby scales. The approximation is accurate within an entire scale octave. This allows us to decouple the sampling of the image pyramid from the sampling of detection scales. Overall, our approximation yields a speedup of 10-100 times over competing methods with only a minor loss in detection accuracy of about 1-2% on the Caltech Pedestrian dataset across a wide range of evaluation settings. The results are confirmed on three additional datasets (INRIA, ETH, and TUD-Brussels) where our method always scores within a few percent of the state-of-the-art while being 1-2 orders of magnitude faster. The approach is general and should be widely applicable.

1 Introduction

Significant progress has been made in pedestrian detection in the last decade. While both detection and false alarm figures are still orders of magnitude away from human performance and from the performance that is desirable for most applications, the rate of progress is excellent. False positive rates have gone down two orders of magnitude since the groundbreaking work of Viola and Jones (VJ) [28, 29]. At 80% detection rate on the INRIA pedestrian dataset [6], VJ outputs 10 false positives per image (fppi), HOG [6] outputs 1 fppi, and the most recent methods [2, 14], through a combination of richer features and more sophisticated learning techniques, output just .1 fppi (error rates as reported in [8]).

The increase in detection accuracy has been paid for with increased computational costs. The VJ detector ran at roughly 15 frames per second (fps) on 384×288 video nearly a decade ago, while the detectors recently evaluated on the Caltech Pedestrian dataset range in time from 1-30 *seconds per frame* on 640×480 video on modern hardware [8]. In many applications of pedestrian detection, including automotive safety, surveillance, robotics, and human machine interfaces, fast detection rates are of the essence.

BMVC 2010 doi:10.5244/C.24.68



(a) dense image pyramid (b) classifier pyramid (c) proposed hybrid approach

Figure 1: In many applications, detection speed is as important as accuracy. **(a)** A standard pipeline for performing modern multiscale detection is to create a densely sampled image pyramid, compute features at each scale, and finally perform sliding window classification (with a fixed scale model). Although effective; the creation of the feature pyramid can dominate the cost of detection, leading to slow multiscale detection. **(b)** Viola and Jones [28] utilized shift and scale invariant features, allowing a trained detector to be placed at any location and scale without relying on an image pyramid. Constructing such a *classifier pyramid* results in fast multiscale detection; unfortunately, most features are *not* scale invariant, including gradient histograms, significantly limiting the generality of this scheme. **(c)** We propose a fast method for approximating features at multiple scales using a sparsely sampled image pyramid with a step size of an entire octave and within each octave we use a classifier pyramid. The proposed approach achieves nearly the same accuracy as using densely sampled image pyramids, with nearly the same speed as using a classifier pyramid applied to an image at a single scale.

We present a method (Figure 1) for significantly decreasing the run-time of *multiscale* object detectors that utilize multiple feature types, including gradient histograms, with very minor decreases to their detection accuracy. Specifically, we show an application to multiscale pedestrian detection that results in nearly real time rates on 640x480 images: about 6 fps for detecting pedestrians at least 100 pixels high and 3 fps for detecting pedestrians over 50 pixels. The resulting method achieves state-of-the-art results, being within 1-2% detection rate of the highest reported results across four datasets (Caltech Pedestrians [8], INRIA [6], ETH [11], and TUD-Brussels [33]).

We show that it is possible to create high fidelity approximations of multiscale gradient histograms using gradients computed at a single scale in §2, and develop a more general theory applicable to various feature types in §3. In §4 we show how to effectively utilize these concepts for fast multiscale detection, and in §5 we apply them to pedestrian detection, resulting in speedups of 1-2 orders of magnitude with little loss in accuracy.

1.1 Related Work

One of the most successful approaches for object detection is the sliding window paradigm [23, 28]. Numerous other detection frameworks have been proposed [2, 15, 20, 30], and although a full review is outside the scope of this work, the approximations we develop could potentially be applicable to such approaches as well. For pedestrian detection [8, 10], however, the top performing methods are all based on sliding windows [6, 7, 14, 22, 31], and each of these methods utilizes some form of gradient histograms. Through numerous strategies, including cascades [28], coarse-to-fine search [17], distance transforms [13], branch and bound search [19], and many others, the classification stage can be made quite fast. Nevertheless, even when highly optimized, just constructing gradient histograms over a finely sampled image pyramid takes a minimum of about one second per 640×480 image [7, 14]; thus this becomes a major bottleneck for all of the pedestrian detectors listed above.

One of the earliest attempts to create real time detectors that utilize gradient histograms was the method of [36] (based on integral histograms [25]). However, the proposed system was real time for *single scale* detection only (recent methods [7, 14] achieve similar speeds and with higher accuracy); in this work we are interested in real time *multiscale* detection. Another recent approach used a fast coarse-to-fine detection scheme but sacrificed detection of small pedestrians [35] (which are of crucial importance in many applications [8, 10]). A number of fast systems have been proposed [3, 12]; however, a detailed overview is outside of the scope of this work. Finally, a number of groups have recently ported HOG to a parallel implementation using GPUs [4, 32, 34]; such efforts are complementary to our own.

Significant research has also been devoted to scale space theory [21], including real time variants [5, 9]. Although only loosely related, a key observation is that half octave pyramids are often sufficient for accurate approximations of various types. This suggests that fine scale sampling may also be unnecessary for object detection.

2 Approximating Multiscale Gradient Histograms

We seek to answer the following question: *given gradients computed at one scale, is it possible to approximate gradient histograms at a different scale?* 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 (even if the gradients themselves change). We begin with an in depth look at a simple form of gradient histograms below and develop a more general theory in §3.

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 $O(i,j) = \arctan\big(\frac{\partial I}{\partial y}(i,j)/\frac{\partial I}{\partial x}(i,j)\big)$. 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. Local histograms with rectangular support are frequently used, these can be defined identically except for the range of the indices i and j. In the following everything that holds for global histograms also applies to local histograms.

2.1 Gradient Histograms in Upsampled Images

Intuitively the information content of an upsampled image is the same as that of the original image (upsampling does not create new image structure). Assume I is a continuous signal, and let I_k denote an upscaled version of I by a factor of k: $I_k(x,y) \equiv I(x/k, y/k)$. Using the definition of a derivative, one can show that $\frac{\partial I_k}{\partial x}(i,j) = \frac{1}{k}\frac{\partial I}{\partial x}(i/k, j/k)$, and likewise for $\frac{\partial I_k}{\partial y}$, 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. Under mild smoothness assumptions, the above also holds (approximately) for discrete signals. Let $M_k(i,j) \approx \frac{1}{k}M(\lceil i/k \rceil, \lceil j/k \rceil)$ denote the gradient magnitude in the upsampled discrete image. Then:

$$\sum_{i=1}^{kn}\sum_{j=1}^{km} M_k(i,j) \approx \sum_{i=1}^{kn}\sum_{j=1}^{km} \frac{1}{k}M(\lceil i/k \rceil, \lceil j/k \rceil) = k^2 \sum_{i=1}^{n}\sum_{j=1}^{m} \frac{1}{k}M(i,j) = k\sum_{i=1}^{n}\sum_{j=1}^{m} M(i,j)$$

Thus, the sum of gradient magnitudes in the pair of images is related by a factor of k. Gradient angles are also preserved since $\frac{\partial I_k}{\partial x}(i,j)\big/\frac{\partial I_k}{\partial y}(i,j) \approx \frac{\partial I}{\partial x}(i/k, j/k)\big/\frac{\partial I}{\partial y}(i/k, j/k)$. Therefore,

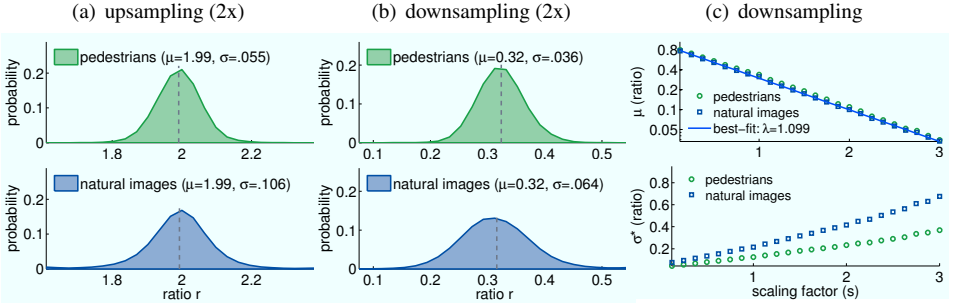


Figure 2: Behavior of gradient histograms in resampled images, see text for details.

according to the definition of gradient histograms given above, the relationship between h_q (computed over I) and h'_q (computed over I_k) is simply: $h'_q \approx kh_q$. We can thus approximate gradient histograms in an upsampled image using gradients computed at the original scale.

Experiments: We demonstrate that the quality of the approximation $h'_q \approx kh_q$ in real images, upsampled using a standard bilinear interpolation scheme, is quite high. We used two sets of images for these experiments. First, we used the 1237 cropped pedestrian images from the INRIA pedestrians training dataset. Each image was 128×64 and contains a pedestrian approximately 96 pixels tall. The second set of images contains 5000 128×64 windows cropped at random positions from the 1218 images in the INRIA negative training set. We refer to the two sets of images as 'pedestrian images' and 'natural images', although the latter set is biased toward windows that may (but do not) contain pedestrians.

In order to measure the fidelity of this approximation, we define the ratio $r_q = h'_q/h_q$, quantizing orientation into $Q = 6$ bins. Figure 2(a) shows the distribution of r_q for one bin on the 1237 pedestrian (top) and 5000 natural (bottom) 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 fairly small, meaning the approximation is unbiased and reasonable.

2.2 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. Below, we demonstrate the nontrivial finding that the information loss is relatively consistent, and furthermore show that we can compensate for it to a large extent in a straightforward manner.

If I contains little high frequency energy, then the approximation $h'_q \approx kh_q$ should apply. In general, however, downsampling results in loss of high frequency content and its gradient energy. Let I_k now denote I *downsampled* by a factor of k. We expect the relationship between h_q (computed over I) and h'_q (computed over I_k) to have the form $h'_q \leq h_q/k$. 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 2(b) we show the distribution of r_q for a single bin on the pedestrian (top) and natural (bottom) images given a downsampling factor of $k = 2$. Observe that the information loss is consistent: r_q is normally distributed around $\mu = .32 < .5$. This implies that $h'_q \approx .32h_q$ could serve as a reasonable approximation for gradient histograms in images downsampled by $k = 2$. We seek to understand how this relation arises and extend the above to all values of k.

Figure 3 shows the quality of the above approximations on example images.

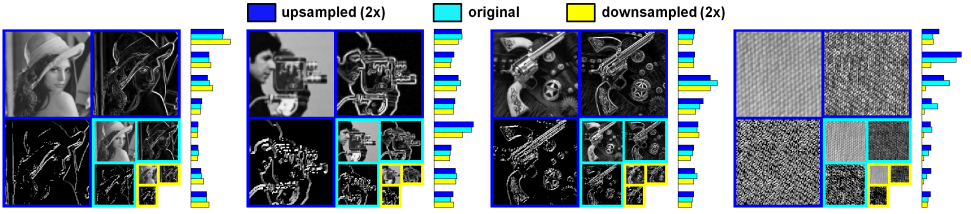


Figure 3: *Approximating gradient histograms in resampled images.* For each image set, we take the original image (cyan border) and generate an upsampled (blue) and downsampled (yellow) version. Shown at each scale are the image (center), gradient magnitude (right), and gradient orientation (bottom). At each scale we compute a gradient histogram with 8 bins, normalizing each bin by .5 and $.32^{-1}$ in the upsampled and downsampled histogram, respectively. Assuming the approximations developed in §2 hold, the three normalized gradient histograms should be roughly equal. For the first three cases, the approximations are fairly accurate. In the last case, showing a highly structured Brodatz texture with significant high frequency content, the downsampling approximation fails entirely.

3 Approximating Multiscale Features

In order to understand more generally how information behaves in resampled images, we turn to the study of natural image statistics [16, 27]. While we analytically derived an expression for predicting gradient histograms in upsampled images, there is no equivalent derivation for downsampled images. Instead, an analysis of natural images statistics allows us to approximate gradient histograms and numerous additional features in resampled images.

We begin by defining a broad family of features. Let Ω be any shift-invariant function that takes an image $I(i,j)$ and creates a new *channel* image $C = \Omega(I)$, where C is a registered map of the original image. Output pixels in C are typically computed from corresponding patches of input pixels in I (thus preserving overall image layout). We define a feature as the weighted sum of a channel C: $f(I) = \sum_{ij} w_{ij} C(i,j)$. Numerous local and global features can be written in this form including gradient histograms, linear filters, color statistics, and countless others [7]. For simplicity, we assume $f(I) = \sum_{ij} C(i,j)$ is the global sum of a channel and refer to the result as the *channel energy*; in the following everything that holds for the channel energy also holds for local weighted features. Finally, we write $f(I,s)$ to denote the channel energy computed over I after downsampling by a factor of 2^s.

3.1 Exponential Scaling Law

Ruderman and Bialek [26, 27] showed that various statistics of natural images are independent of the scale at which the images were captured, or in other words, the statistics of an image are independent of the scene area corresponding to a single pixel. In the context of our work, we expect that on average, the difference in channel energy between an image and a downsampled version of the image is independent of the scale of the original image and depends only on the relative scale between the pair of images. In other words the expectation over natural images of $f(I,s_1)/f(I,s_2)$ should depend only on $s_1 - s_2$. We can formalize this by assuming there exists a function $r(s)$ such that $f(I,s_1)/f(I,s_2) \approx r(s_1 - s_2)$ and $E[f(I,s_1)/f(I,s_2)] = E[f(I,s_1)]/E[f(I,s_2)] = r(s_1 - s_2)$ for all s_1, s_2. One can then show that $r(s_1 + s_2) = r(s_1)r(s_2)$; if r is also continuous and non-zero, then it must take the form $r(s) = e^{-\lambda s}$ for some constant λ [18]. Therefore, $E[f(I,s_1)/f(I,s_2)]$ must have the form:

$$E[f(I,s+s_0)/f(I,s_0)] = e^{-\lambda s} \tag{1}$$

Each channel type has its own corresponding λ, determined empirically. In §3.2 we show that (1) provides a remarkably good fit to our data for multiple channel types and image sets.

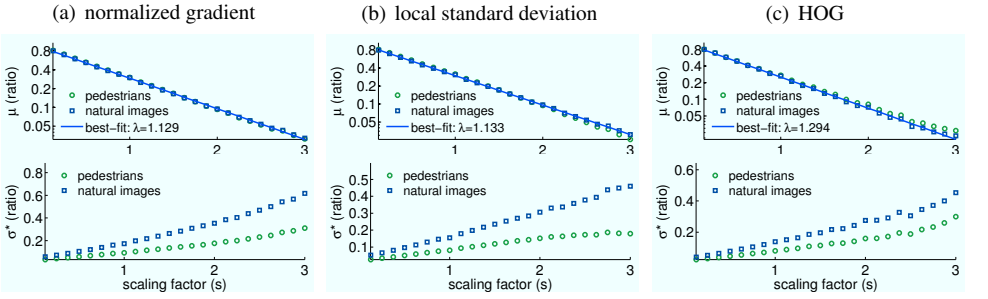


Figure 4: Behavior of channel energy in downsampled images. Top: $E[r(I,s)]$ as a function of s; in each case $ae^{-\lambda s}$ provides an excellent fit for both pedestrian and natural images. Bottom: standard deviation of $r^*(I,s) = r(I,s)/ae^{-\lambda s}$ increases slowly as a function of s. The channel types plotted are: **(a)** gradient histogram channels computed over locally L_1 normalized gradients $M'(i,j) = M(i,j)/(E[M(i,j)]+.1)$ (with E computed over a 9×9 image neighborhood); **(b)** standard deviation of intensity values computed in each local 9×9 neighborhood $C(i,j) = E[I(i,j)^2] - E[I(i,j)]$; and **(c)** HOG [8] (each channel is a single gradient orientation at one of four normalizations).

Although (1) holds for an ensemble of images, the equivalent relation should also hold for individual images. Given $f(I,0)$, we propose to approximate $f(I,s)$ by:

$$\boxed{f(I,s) \approx f(I,0)e^{-\lambda s}} \tag{2}$$

Experiments in §3.3 indicate that the quality of the approximation in (2) is very good, for both natural *and* pedestrian images. Although the quality of the approximation degrades with increasing value of s, it does so only gradually and proves effective in practice.

Equations (1) and (2) also hold for *upsampled* images (details omitted). However, λ for upsampling and downsampling will typically be different even for the same channel type (as in the case of gradient histograms, see §2). In practice, though, we want to predict channel energy in *higher resolution* images (to which we may not have access) as opposed to (smooth) upsampled images. For this one should use the same λ as for downsampling.

3.2 Estimating λ

We perform a series of experiments to verify (1) and estimate λ for four different channel types. Define $r(I,s) \equiv f(I,s)/f(I,0)$. To estimate λ, we first compute $\mu_s = E[r(I,s)]$ for $s = \frac{1}{8}, \frac{2}{8}, \ldots, \frac{24}{8}$. For the gradient histogram channels defined previously as $C(i,j) = M(i,j)\mathbf{1}[O(i,j) = q]$, the 24 resulting values μ_s for both pedestrian and natural images are shown in Figure 2(c), top. Observe that μ_s does not start near 1 as expected: bilinear interpolation smooths an image somewhat even when using a single pixel downsampling rate ($s = \varepsilon$), in which case $E[r(I,\varepsilon)] \approx .88$. We thus expect μ_s to have the form $\mu_s = ae^{-\lambda s}$, with $a \approx .88$ as an artifact of the interpolation. To estimate λ and a we use a least squares fit of $\lambda s = \ln(a) - \ln(\mu_s)$ to the 24 means computed over natural images, obtaining $\lambda = 1.099$ and $a = .89$. The agreement between the resulting best-fit curve and the observed data points is excellent: the average squared error is only 1.8×10^{-5}. The best-fit curve obtained from natural images was also a very good fit for pedestrian images, with average error 3.2×10^{-4}.

We repeat the above experiment for the three additional channel types, results are shown in Figure 4, top. For every channel type (1) is an excellent fit to the observations μ_s for both natural and pedestrian images (with a different λ for each channel). The derivation of (1) depends on the distribution of image statistics being stationary with respect to image scale; that this holds for pedestrian images in addition to natural images, and with nearly an identical constant, implies the estimate of λ is robust and generally applicable.

3.3 Approximation Accuracy

We have shown that (1) is an excellent fit for numerous channel types over an ensemble of images; we now examine the quality of the approximation in (2) for individual images. To do so, we define the quantity $r^*(I,s) = r(I,s)/ae^{-\lambda s}$, with a and λ set to the estimates obtained previously. $r^*(I,s)$ is normalized such that $E[r^*(I,s)] \approx 1$; for an individual image, $r^*(I,s) \approx 1$ implies the approximation (2) is accurate. In Figures 2(c) and 4, bottom, we plot the standard deviation σ^* of $r^*(I,s)$. Low standard deviation implies low average error, and for $s \leq 1$ (downsampling by at most two), $\sigma^* < .2$ for all channel types studied. In general, σ^* increases gradually and not too steeply as a function of s. The best evidence for the validity of the approximation, however, is that its use does not significantly degrade the performance of pedestrian detection, as we will show in §5.

4 Fast Multiscale Detection

In numerous tasks, including sliding window detection, the same features are computed at multiple locations and scales. In the context of the channel features discussed in §3, this can be performed efficiently assuming $C = \Omega(I)$ is *shift* and *scale* invariant, respectively. Ω is *shift* invariant if computing Ω on a translated version of an image I is the same as translating the channel $\Omega(I)$; likewise Ω is *scale* invariant if computing Ω on a resampled version of I is the same as resampling $\Omega(I)$. Shift invariance allows for fast single scale detection; scale invariance allows for fast multiscale detection. Most Ω used in computer vision are shift invariant but *not* scale invariant; nevertheless, the approximations developed in §3 can give us nearly the same speed as true scale invariance.

Most modern detectors utilize features which are not scale invariant, such as gradient histograms. This includes all top performing pedestrian detectors [6, 7, 14, 22, 31] evaluated on the Caltech Pedestrian dataset [8]. Without scale invariance, the standard approach is to explicitly construct an *image pyramid* [21] and perform detection at each scale separately, see Figure 1(a). To detect objects larger than the model scale the image is downsampled; conversely, to detect smaller objects the image is upsampled. At each resulting scale features are recomputed and single-scale detection applied. Typically, detectors are evaluated on 8-16 scales per octave [8], and even when optimized, just constructing gradient histograms over a finely sampled image pyramid can take over one second per 640×480 image [7, 14].

Given shift and scale invariance, fast multiscale detection is possible through the construction of a *classifier pyramid*, which involves rescaling a single detector to multiple scales, see Figure 1(b). The Viola and Jones detector (VJ) [28] was built using this idea. Utilizing integral images, Haar like features (differences of rectangular sums) [24] at any position and scale can be computed with a fixed number of operations [28]. To compute the detector at different spatial locations, the support of the Haars simply needs to be shifted; to compute the detector at different scales, the Haar features need to be shifted, their dimensions adjusted, and a scaling factor introduced to account for their change in area, but otherwise no additional changes are needed. During detection the integral image needs to be computed only once at the original scale, and since the classifier is fast through the use of integral images and cascades, so is the resulting overall multiscale detection framework.

Viola and Jones demonstrated a detection framework using scale invariant features that achieves real time multiscale detection rates; however, scale invariant features tend to be quite limited. On the other hand, using richer features, such as gradient histograms, leads to large increases in accuracy but at the cost of much slower multiscale detection. Instead,

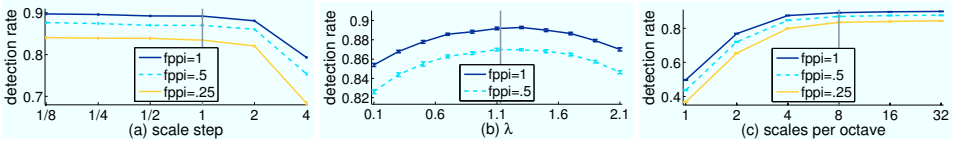


Figure 5: Performance under various parameter choices in the construction of *FPDW*; evaluated using 25 trials on the INRIA full image dataset using. The vertical gray lines denote default parameter values used in all other experiments (scale step of 1, $\lambda = 1.129$ as predicted in §3.2, 8 scales per octave). **(a)** Detection rate decreases very slowly as the scale step size for the image pyramid increases up to 2 octaves, demonstrating the applicability of (2) over a wide scale range. **(b)** Sub-optimal values of λ for the normalized gradient channels causes a decrease in performance. **(c)** At least 8 scales per octave are necessary in the classifier pyramid for good performance.

we propose to construct a classifier pyramid using features which are *not* scale invariant, utilizing the approximation given in (2). As the quality of the approximation degrades with increasing distance in scale, we utilize a *hybrid approach*: we construct a sparsely sampled image pyramid with a step size of one octave and within each octave we use a classifier pyramid, see Figure 1(c). In essence, the proposed approach achieves the speed of a classifier pyramid with the accuracy of an image pyramid. Implementation details are given in §5.

4.1 Complexity Analysis

The computational savings of using the hybrid approach over a densely sampled image pyramid can be significant. Assume the cost of computing features is linear in the number of pixels in an $n \times n$ image (as is often the case). Typically the image pyramid is sampled using a fixed number of m scales per octave, with each successive image in the pyramid having side length $2^{1/m}$ times that of the previous image. The cost of constructing the pyramid is:

$$\sum_{k=0}^{\infty} n^2 2^{-2k/m} = n^2 \sum_{k=0}^{\infty} (4^{-1/m})^k = \frac{n^2}{1-4^{-1/m}} \approx \frac{mn^2}{\ln 4} \tag{3}$$

The second equality follows from the formula for a sum of a geometric series; the last approximation is valid for large m (and follows by a subtle application of l'Hôpital's rule). In the hybrid approach we use one scale per octave ($m = 1$). 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 $m = 8$ to 16 scales per octave [8], thus according to (3) we expect an order of magnitude savings by using the proposed hybrid approach (more if upsampled images are used).

5 Experiments

For the following experiments, we use the *ChnFtrs* detector described in [7]. The detector is relatively fast and achieves good performance across a number of pedestrian datasets and scenarios, as described on the Caltech Pedestrian Dataset website [1, 3]. *ChnFtrs* spends most of its computation constructing the feature pyramid, making it an excellent candidate for our fast detection scheme. No re-training was necessary for this work; instead, we rescale a pre-trained detector [7] using the approximation in (2) (details below). We refer to our fast multiscale variant of *ChnFtrs* as the 'Fastest Pedestrian Detector in the West' (*FPDW*).

The *ChnFtrs* detector is a generalization of VJ: instead of constructing an integral image and extracting Haar-like features over just the original intensity image I, multiple channels $C = \Omega(I)$ are used, including gradient magnitude, normalized gradient histogram and LUV color channels. For additional details see [7]. To resample the detector the Haars need

(a) Caltech Pedestrian Data (peds. ≥ 100 pixels) (b) Caltech Pedestrian Data (peds. ≥ 50 pixels))

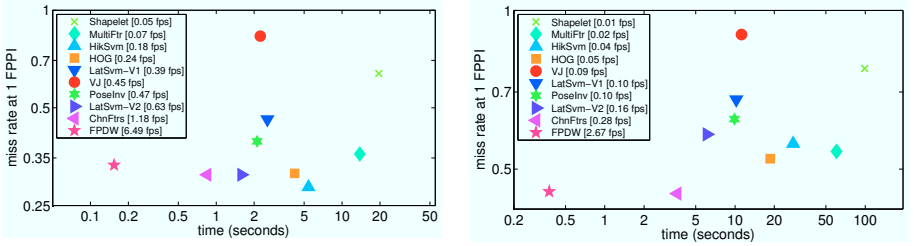
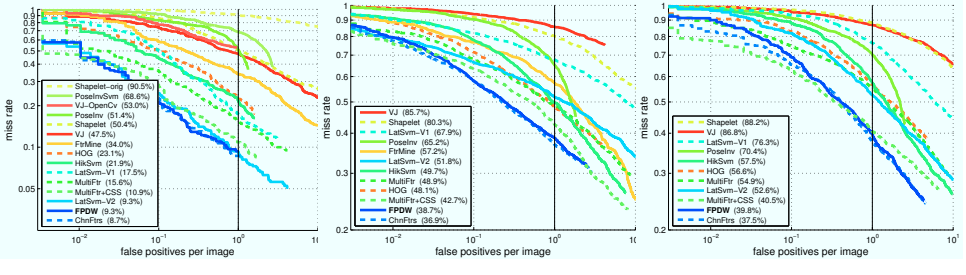


Figure 6: Time versus detection rate at 1 false positive per image on 640×480 images from the Caltech Pedestrian Dataset [9]. Timing methodology and code may be obtained from [10]. Run times of all algorithms are normalized to the rate of a single modern machine, hence all times are directly comparable. (Note that the VJ implementation used did not utilize scale invariance and hence its slow speed). *FPDW* obtains a speedup of about 10-100 compared to competing methods with a detection rate within a few percent of best reported performance.

to be repositioned, their dimensions adjusted accordingly, and finally, according to (2), the output of each Haar must be multiplied by a channel specific scaling factor $e^{\lambda s}$ ($s > 0$ for downsampling, $s < 0$ for upsampling). For features computed over the gradient channels, which were L_1 normalized, $\lambda = 1.129$ is used (see Figure 4(a)). Color channels, like intensity channels, are scale invariant and $\lambda = \ln(4)$ is used to compensate for the change in feature area during resampling. Finally, as most Haars can't be resized exactly (due to quantization effects), a multiplicative factor can also used to compensate for changes in area.

As the approximation (2) degrades with increasing scale offsets, our hybrid approach is to construct an image pyramid sampled once per octave and use the classifier pyramid within half an octave in each direction of the original detector. Details of how this and other choices in the construction of *FPDW* affect performance are shown in Figure 5. We emphasize that re-training was not necessary and all other parameters were unchanged from *ChnFtrs*.

Overall, *FPDW* is roughly 5-10 times faster than the *ChnFtrs* detector; detailed timing results are reported in Figure 6. Computing the feature pyramid is no longer the bottleneck of the detector; thus, if desired, additional speedups can now be achieved by sampling fewer detections windows (although at some loss in accuracy). Finally, in Figure 7 we show full-image results on three datasets [6, 8, 33]. In all cases the detection rate of *FPDW* is within 1-2% of the top performing algorithm, and always quite close to the original *ChnFtrs* classifier, all while being 1-2 orders of magnitude faster than competing methods.



(a) INRIA Results (b) Caltech 'Reasonable' Results (c) TUD-Brussels Results

Figure 7: The 'Fastest Pedestrian Detector in the West' (*FPDW*) is obtained by rescaling *ChnFtrs* to multiple target scales. Results on three datasets are shown (plot legends are ordered by miss rate at 1 false positive per image – lower is better). In all cases the detection rate of *FPDW* is within a few percent of *ChnFtrs* while being 1-2 orders of magnitude faster than all competing methods. Evaluation scripts, detector descriptions, additional results (including on the ETH dataset [11]) and under varying conditions) are all available online at [10].

6 Conclusion

The idea motivating our work is that we can approximate features, including gradient histograms, at nearby scales from features computed at a single scale. To take advantage of this, we proposed a hybrid approach that uses a sparsely sampled image pyramid to approximate features at intermediate scales. This increases the speed of a state-of-the-art pedestrian detector by an order of magnitude with little loss in accuracy. That such an approach is possible is not entirely trivial and relies on the fractal structure of the visual world; nevertheless, the mathematical foundations we developed should be readily applicable to other problems.

Acknowledgments: P. P. was supported by ONR MURI Grant #N00014-06-1-0734. S. B. was supported in part by NSF CAREER Grant #0448615 and ONR MURI Grant #N00014-08-1-0638.

References

[1] www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/.

[2] S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *ECCV*, 2002.

[3] 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. *IJRR*, 2009.

[4] B. Bilgic. Fast human detection with cascaded ensembles. Master's thesis, MIT, February 2010.

[5] J. L. Crowley, O. Riff, and J. H. Piater. Fast computation of characteristic scale using a half-octave pyramid. In *International Conference on Scale-Space Theories in Computer Vision*, 2002.

[6] N. Dalal and B. Triggs. Histogram of oriented gradient for human detection. In *CVPR*, 2005.

[7] P. Dollár, Z. Tu, P. Perona, and S. Belongie. Integral channel features. In *BMVC*, 2009.

[8] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *CVPR*, 2009.

[9] R. S. Eaton, M. R. Stevens, J. C. McBride, G. T. Foil, and M. S. Snorrason. A systems view of scale space. In *ICVS*, 2006.

[10] M. Enzweiler and D. M. Gavrila. Monocular pedestrian detection: Survey and experiments. *PAMI*, 2009.

[11] A. Ess, B. Leibe, K. Schindler, and L. Van Gool. A mobile vision system for robust multi-person tracking. In *CVPR*, 2008.

[12] A. Ess, B. Leibe, K. Schindler, and L. Van Gool. Robust multi-person tracking from a mobile platform. *PAMI*, 31(10):1831–1846, 2009.

[13] P. Felzenszwalb and D.P. Huttenlocher. Efficient matching of pictorial structures. In *CVPR*, 2000.

[14] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI - to appear*, 2010.

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

[16] D. J. Field. Relations between the statistics of natural images and the response properties of cortical cells. *Journal of the Optical Society of America A*, 4:2379–2394.

[17] F. Fleuret and D. Geman. Coarse-to-fine face detection. *IJCV*, 41(1-2):85–107, 2001.

[18] S. G. Ghurye. A characterization of the exponential function. *The American Mathematical Monthly*, 64(4):255–257, 1957.

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

[20] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segm. *IJCV*, pages 259–289, 2008.

[21] T. Lindeberg. Scale-space for discrete signals. *PAMI*, 12(3):234–254, 1990.

[22] S. Maji, A.C. Berg, and J. Malik. Classification using intersection kernel SVMs is efficient. In *CVPR*, 2008.

[23] C. Papageorgiou and T. Poggio. A trainable system for object detection. *IJCV*, 38(1):15–33, 2000.

[24] C. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *ICCV*, 1998.

[25] F. Porikli. Integral histogram: A fast way to extract histograms in cartesian spaces. In *CVPR*, 2005.

[26] D. L. Ruderman. The statistics of natural images. *Network: Computation in Neural Systems*, 5 (4):517–548, 1994.

[27] D. L. Ruderman and W. Bialek. Statistics of natural images: Scaling in the woods. *Physical Review Letters*, 73(6):814–817, Aug 1994.

[28] P. Viola and M. Jones. Fast multi-view face detection. In *CVPR*, 2001.

[29] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *CVPR*, 2003.

[30] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *ECCV*, 2000.

[31] C. Wojek and B. Schiele. A performance evaluation of single and multi-feature people detection. In *DAGM*, 2008.

[32] C. Wojek, G. Dorkó, A. Schulz, and B. Schiele. Sliding-windows for rapid object class localization: A parallel technique. In *DAGM*, 2008.

[33] C. Wojek, S. Walk, and B. Schiele. Multi-cue onboard pedestrian detection. In *CVPR*, 2009.

[34] L. Zhang and R. Nevatia. Efficient scan-window based object detection using gpgpu. In *Visual Computer Vision on GPU's*, 2008.

[35] W. Zhang, G. J. Zelinsky, and D. Samaras. Real-time accurate object detection using multiple resolutions. In *ICCV*, 2007.

[36] Q. Zhu, M.C. Yeh, K.T. Cheng, and S. Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *CVPR*, 2006.