Integral Channel Features – Addendum

Piotr Dollár[1]
pdollar@caltech.edu
Zhuowen Tu[2]
zhuowen.tu@loni.ucla.edu
Pietro Perona[1]
perona@caltech.edu
Serge Belongie[3]
sjb@cs.ucsd.edu

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

[2] Lab of Neuro Imaging
University of CA, Los Angeles
Los Angeles, CA, USA

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

1 Addendum

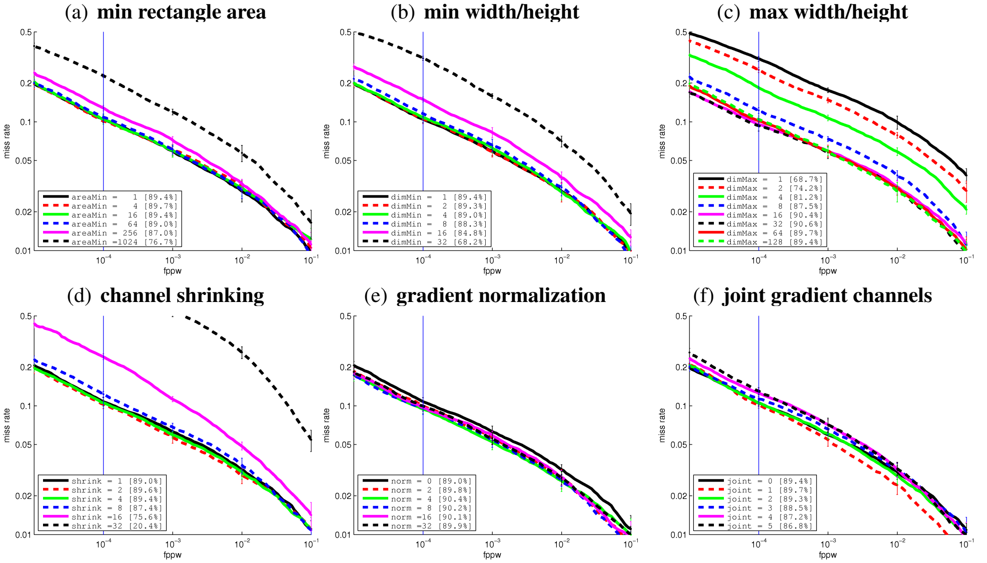
This document is meant to serve as an addendum to [1], published at BMVC 2009. The purpose of this addendum is twofold: (1) to respond to feedback we've received since publication and (2) to describe a number of changes, especially to the non-maximal suppression, that further improve performance. The performance of our updated detection increases 5% to over 91% detection rate at 1 false positive per image on the INRIA dataset, and similarly on the Caltech Pedestrian Dataset, while overall system runtime for multiscale detection decreases by $1/3$ to just under 1.5s per 640×480 image.

We begin by rectifying an important omission to the related work. Levi and Weiss had an innovative application of integral images to multiple image channels quite early on, demonstrating good results on face detection from few training examples [4]. This work appears to be the earliest such use of integral images, indeed the authors even describe a precursor to integral histograms. Many thanks to Mark Everingham for sending us this reference.

1.1 Channels and Features: Additional Experiments

We start by repeating all our original experiments over ten random trials to ensure the results are statistically significant (in our original presentation only a single trial was used for each experiment). As before we use use false positive per window (fppw) curves for these experiments and switch to per image measures when comparing to other methods in the literature. In each trial we vary the pool of random features (the training and testing data is kept constant). Results averaged over the ten trials, along with standard error bars, are shown in Fig. 4 toward the end of this document. They are qualitatively unchanged from before (compare to Fig. 4 in [1]) and we observe that the curves are quite stable.

Feature size: Recall that the candidate features are generated by randomly choosing both the channel index and rectangle. In our original experiments we enforced that random rectangles have area of at least 25 pixels to avoid the feature pool from being dominated by small rectangles. It turns out that allowing smaller rectangles has no impact on performance,



(a) **min rectangle area** (b) **min width/height** (c) **max width/height** (d) **channel shrinking** (e) **gradient normalization** (f) **joint gradient channels**

Figure 1: Additional evaluation of channel features, see text for details.

see Fig. 1(a). To achieve good performance, only intermediate size rectangles are necessary: in Fig. 1(b) and Fig. 1(c) results are shown where the minimum and maximum rectangle dimension, respectively, was limited to a given size. Removing all small rectangles (width or height below 8 pixels) or large rectangles (width or height above 16 pixels) did not affect performance; however, removing intermediate size rectangles severely degraded results. Note that these results are generally consistent with the work of Ullman *et al.* [7] who showed that visual features of intermediate complexity are necessary for accurate classification.

Shrinking channels: As small features are not necessary for good performance, we hypothesized that we could shrink the channels prior to computing the integral images without loss of accuracy. Indeed, shrinking the channels by a factor of 4 along each dimension does not adversely affect performance, see Fig. 1(d). The resulting speedup of the integral channel computation is significant, the overall system runtime for multiscale detection decreases by $1/3$ to just under 1.5s per 640×480 image. We emphasize that we shrink the channels rather than the original image (please refer to the related discussion on pre and post-smoothing).

Gradient normalization: Gradient normalization plays an important role in many image descriptors, including HOG and SIFT. We implemented a very simple and fast L1 normalization scheme: the gradient magnitude at each pixel is divided by the sum of the gradient magnitude in the surrounding window. Results for various window sizes are shown in Fig. 1(e); although the effect is small, some normalization (norm>0) is slightly better then none.

Joint gradient channels: Inspired by classic work in texture recognition of Haralick *et al.* [2], a number of recent papers have experimented with higher order pixel statistics for pedestrian detection [3, 6]. Haralick *et al.* [2] captured texture statistics by computing co-occurrence statistics between pairs of nearby pixels, pooling the results over the image, and deriving texture measures from the resulting co-occurrence matrices. Schwartz *et al.* [6] used these co-occurrence features directly for pedestrian detection and addressed reducing their dimensionality prior to SVM training. In [3], the authors propose using gradient local auto-correlation (GLAC) features for capturing joint gradient statistics.

We attempted to incorporate the GLAC features [3] minus the spatial pooling step into our integral channel framework. Unfortunately their use did not improve performance (see Fig. 1(f)). Nevertheless, we give details below for completeness. The first step is to compute gradient magnitude and orientation, quantized into D bins (we use $D = 6$ throughout). The computation is identical as for the standard gradient histograms. Next, at each location x, we take the two pixels at locations $x+d$ and $x-d$ for a fixed direction d and store the minimum of their gradient magnitudes at location x in one of D^2 channels, as determined by their joint orientation. Actually, as in [3], we used bilinear interpolation to place the gradient magnitude in up to four channels. Also, as in [3], we used four direction d: $(0,r),(r,0),(r,r),(-r,r)$ for a fixed radius r, resulting in $4D^2 = 144$ image channels. To reduce memory usage we shrunk all channels by a factor of 4 (see above). Results for various setting of the radius are shown in Fig. 1(f). As stated above performance does not improve when the joint channels are included and actually begins to degrade for large radii. It is not clear why the joint features are not effective, lack of a sophisticated normalization scheme may have played a role.

1.2 Full Image Results: Additional Experiments

For all subsequent experiments we use a re-trained classifier with 6 parameters changed from the default configuration in order to maximize performance. We used: (1) LUV color channels, (2) pre-smoothing with $r = 1$, (3) 2000 weak classifiers, (4) 30,000 candidate features for training, (5) channels shrunk by a factor of 4, and (6) gradient normalization with a radius of 4. The resulting classifier achieves over 93% detection rate at 10^{-4} fppw, and about 90% at 10^{-5} fppw. We refer to the resulting method as *ChnFtrs*. Note that only (5) and (6) are different from the setting used for the final classifier in [1] and per-window performance is only slightly improved. The new classifier is, however, about 33% faster.

1.2.1 Non-Maximal Suppression

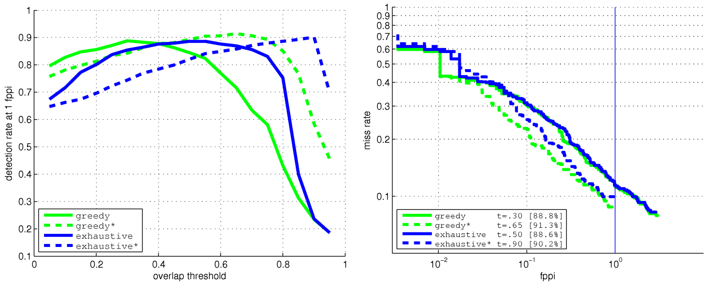


Figure 2: Evaluation of non-maximal suppression, see text for details.

A number of minor tweaks to the NMS raised full-image performance from about an 86% detection rate at 1 false positive per image (fppi) to over 91%. First, we found that cropping the extra padding around the detected bounding boxes prior to NMS rather then after improves performance slightly. Our procedure for NMS is to suppresses the less confident of every pair of detections that overlap sufficiently according to the PASCAL criteria [5]. In the first variant all pairs of detections are considered ('exhaustive'). The second variant is similar, except detections are processed in order of decreasing score, and, unlike in the first, once a detection is suppressed it can no longer suppress weaker detections ('greedy').

In addition, we observed that many false positives are caused by our detector firing on sub-regions of a pedestrian (*e.g.*, the legs). To allow two detections BB_1 and BB_2 at nearby spatial position but different scales to interact we alter the PASCAL overlap criteria from:

$$\frac{\mathrm{area}(BB_1 \cap BB_2)}{\mathrm{area}(BB_1 \cup BB_2)} \quad \text{to:} \quad \frac{\mathrm{area}(BB_1 \cap BB_2)}{\min(\mathrm{area}(BB_1), \mathrm{area}(BB_2))}. \tag{1}$$

We refer to the variants of NMS using the latter overlap criteria as ‘exhaustive*’ and ‘greedy*’.

Results are shown in Fig. 2. As can be seen in the left panel, the overlap must be set quite differently for the four NMS variants. The ‘exhaustive’ NMS variants are more aggressive then the ‘greedy’ variants, and the ‘*’ variants are more aggressive then the non-‘*’ variants, hence they require the largest overlaps. Performance varies smoothly with the overlap making it easy to select. In the right panel we show performance for each NMS variant for reasonable choices of the overlap. The ‘greedy’ and ‘exhaustive’ variants perform very similarly, as do the ‘greedy*’ and ‘exhaustive*’ variants. The ‘greedy’ variants are faster, so they are preferred. The best performance of 91% detection at 1 fppi is achieved by ‘greedy*’ with an overlap set to .65.

1.2.2 Full Image Results

We conclude by repeating all the full image results using the updated detector. Results are shown in Fig. 3; we refer to our method as *ChnFtrs*. For details see discussion in Section 4.2 in [1]. Note that up to date and more complete results are maintained on the website for the Caltech Pedestrian Dataset[1].

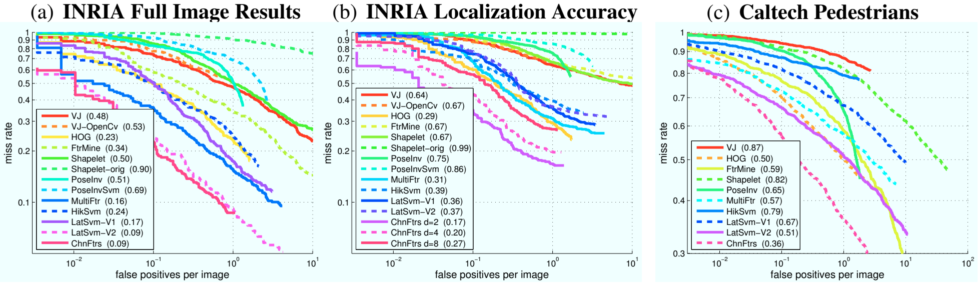

(a) **INRIA Full Image Results** (b) **INRIA Localization Accuracy** (c) **Caltech Pedestrians**

Figure 3: Full image results, see text for details.

References

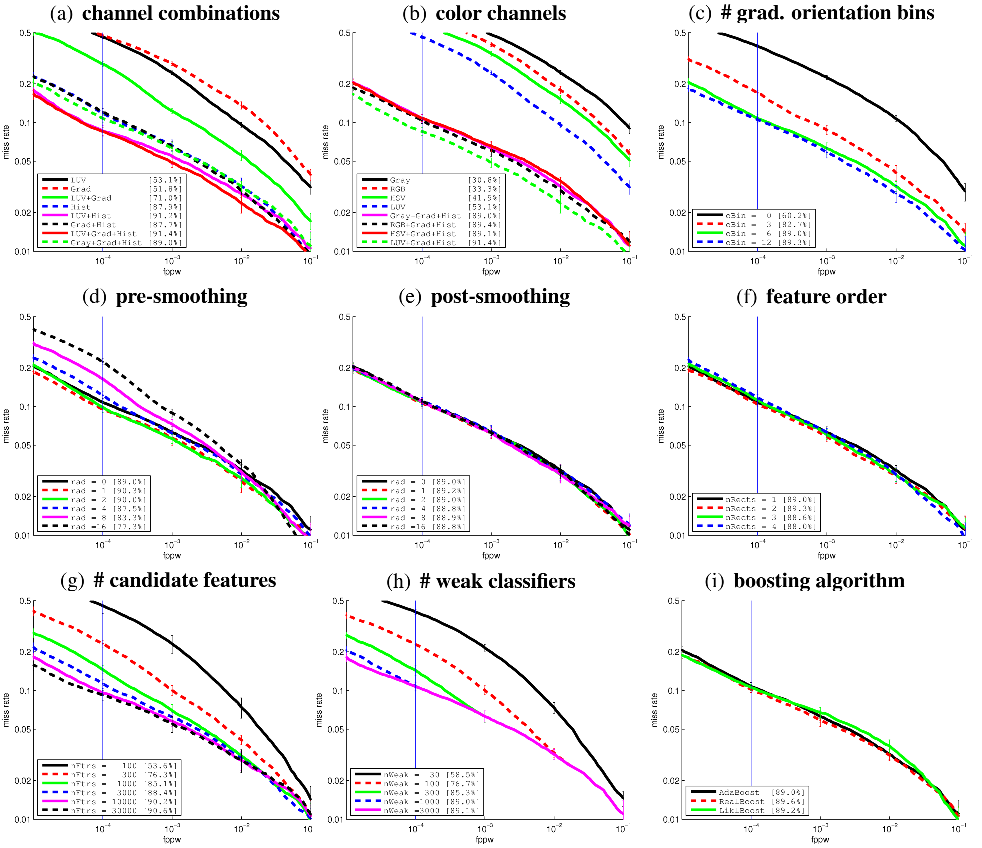
[1] P. Dollár, Z. Tu, P. Perona, and S. Belongie. Integral channel features. In *BMVC*, 2009.

[2] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(6):610–621, 1973.

[3] T. Kobayashi and N. Otsu. Color image feature extraction using color index local autocorrelations. In *ICASSP*, 2009.

[4] K. Levi and Y. Weiss. Learning object detection from a small number of examples: The importance of good features. In *CVPR*, 2004.

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



(a) **channel combinations**
(b) **color channels**
(c) **# grad. orientation bins**
(d) **pre-smoothing**
(e) **post-smoothing**
(f) **feature order**
(g) **# candidate features**
(h) **# weak classifiers**
(i) **boosting algorithm**

Figure 4: Evaluation of channel features, see text for details.

[5] J. Ponce, T.L. Berg, M. Everingham, D. Forsyth, M. Hebert, S. Lazebnik, M. Marszałek, C. Schmid, C. Russell, A. Torralba, C. Williams, J. Zhang, and A. Zisserman. Dataset issues in object rec. In *Towards Category-Level Object Rec.*, pages 29–48. Springer, 2006.

[6] W.R. Schwartz, A. Kembhavi, D. Harwood, and L.S. Davis. Human detection using partial least squares analysis. In *ICCV*, 2009.

[7] S. Ullman, M. Vidal-Naquet, and E. Sali. Visual features of intermediate complexity and their use in classification. *Nature Neuroscience*, 5(7):682–687, July 2002.