Appears in: Proc. IEEE Int’l. Conf. on Image Processing, San Antonio, TX, Oct 2007.

A MACHINE LEARNING FRAMEWORK FOR ADAPTIVE COMBINATION OF SIGNAL DENOISING METHODS

David K. Hammond and Eero P. Simoncelli

Center for Neural Science and Courant Institute of Mathematical Sciences
New York University

We present a general framework for combination of two distinct local denoising methods. Interpolation between the two methods is controlled by a spatially varying decision function. Assuming the availability of clean training data, we formulate a learning problem for determining the decision function. As an example application we use Weighted Kernel Ridge Regression to solve this learning problem for a pair of wavelet-based image denoising algorithms, yielding a “hybrid” denoising algorithm whose performance surpasses that of either initial method.

Index Terms**—** Image Processing, Image Denoising, Machine Learning, Kernel Ridge Regression

1. INTRODUCTION

The image denoising problem consists of recovering the true content of a digital image that has been corrupted by noise. Under the common assumption of additive noise, one may view the denoising process as partitioning a given noisy signal into an estimate of the desired clean signal and the residual noise. Performing this separation relies upon describing and exploiting the differences between signal and noise.

Natural photographic images typically contain highly localized oriented features such as edges formed by occlusion boundaries, as well as localized non-oriented features such as T-junctions. In contrast, noise processes are typically homogeneous across the image and do not contain significant local structure. Local image features are poorly captured by global denoising methods based on classical power spectral models, such as the global Wiener filter. This observation motivates the use of denoising algorithms that act locally, where each signal coefficient is estimated using a local neighborhood of noisy data. Many recent denoising methods can be viewed as such local filtering operations.

Images are also often inhomogeneous, with different local structure in different areas such as smooth nearly constant regions with sky or blank walls, edge regions, and texture regions that may or may not be oriented. Local filtering behavior that is appropriate for signal content in one region may be inappropriate in another region. This has motivated research in developing locally adaptive denoising methods that modulate their filtering behavior based on the local image content [1, 2].

Using ideas from machine learning, we study the problem of constructing a new denoising algorithm by combining two distinct local denoising functions. Machine learning techniques have been applied to image denoising before. Several authors have used support vector regression techniques to directly estimate clean coefficients from noisy coefficients [3, 4]. Lin and Yu have used an SVM classifier to adaptively switch between applying a median filter and the identity filter for removing impulse noise [5].

In this work, we introduce a locally adaptive decision function that determines how the two base denoising estimates are to be combined at each location. This decision function is then learned from example data, where we assume we have access to an “example” clean image whose statistical and structural properties are similar to the image to be denoised. We apply this to a pair of state-of-the-art wavelet-based image denoising algorithms, yielding a hybrid image denoising method with better overall performance.

2. LOCAL DENOISING FUNCTIONS

Considering the noisy signal as a vector $y \in \mathbb{R}^N$, any denoising algorithm may be viewed as a function $f : \mathbb{R}^N \to \mathbb{R}^N$ where $\hat{x} = f(y)$ is the estimate of the clean signal. It is a common practice to first transform the image with a multiscale wavelet transform and do the denoising calculations in the space of wavelet coefficients. The wavelet transform of an image consists of coefficient subbands corresponding to different spatial scales and orientations. We define a generalized wavelet neighborhood, or patch, as a set of coefficients close to each other in space, scale and orientation. Given this notion of a generalized wavelet neighborhood, we define a local denoising function to be a function $g : \mathbb{R}^d \to \mathbb{R}^n$ mapping a patch of d wavelet coefficients to an estimate of a group of $n < d$ coefficients, typically at the center of the patch. Applying this procedure to overlapping patches and estimating the center coefficients yields a complete estimator for all of the wavelet coefficients, which may be inverted to give the denoised image.

3. HYBRID DENOISER FORM

Given a set of two local denoising functions g_1, g_2 with the same input and output dimensionality, we seek to combine them into a single hybrid denoising function g_h. Introducing the decision function h, we write the hybrid estimate for a noisy patch $y \in \mathbb{R}^d$ as

$$g_h(y) = h(y)g_1(y) + (1 - h(y))g_2(y)$$

The decision function h should determine for each patch which of the two base denoising methods is more reliable. As h is a function of the patch itself, it is spatially adaptive. If the initial denoisers have been optimized for distinct local signal content, one may view the output of h as classifying each patch into the natural domain for either g_1 or g_2. Allowing h to take arbitrary real values avoids a hard decision for each patch and permits the hybrid denoiser g_h to interpolate smoothly between the outputs of the base denoising functions. Fitting h from data is then a regression problem.

3.1. Generation of training data

We wish to learn the function h that will yield good performance for the resulting hybrid denoiser. Let $y \in \mathbb{R}^d$ and $x^c \in \mathbb{R}^n$ denote a noisy wavelet patch and corresponding clean center coefficients. We assume that these are drawn from some fixed unknown distribution $\mathbf{D}(y, x^c)$ that is determined by the statistics of the signal and noise processes. We measure the performance of h by the expected squared error for the corresponding hybrid denoiser g_h, given by

$$E_{(y,x^c)}\left[||h(y)g_1(y) + (1-h(y))g_2(y) - x^c||^2\right]$$

In practice we must learn h from a finite set of training examples $\{(y_i, x_i^c)\}_{i=1}^m$. Define the error for the ith training sample as

$$E(h_i, i) = ||x_i^c - (h_i g_1(y_i) + (1-h_i)g_2(y_i))||^2$$

which is a quadratic polynomial in h_i. Next, we define the target value h_i^* for $h(y_i)$ to be the argmin of $E(h_i, i)$, which yields

$$h_i^* = \frac{-(g_1(y_i) - g_2(y_i)) \cdot (g_2(y_i) - x_i^c)}{||g_1(y_i) - g_2(y_i)||^2}$$

The pairs $\{(y_i, h_i^*)\}_{i=1}^m$ then form the training data set for learning the decision function h.

3.2. Training Data Weights

One important issue for learning h is that the same amount of error in h for different patches will contribute differently to the error for the hybrid denoiser. For patches where the output of the two base denoisers g_1 and g_2 are either very similar or close to zero, large changes in h will yield only small changes in the output of g_h. Conversely, for image regions where the outputs of the base denoisers are substantially different, small changes in h lead to large changes in g_h and in these regions it is more important for h to be correct.

Appropriate weightings for the training examples can be found by expanding the error of the hybrid denoiser g_h on the training set, the so-called empirical loss, in terms of the target values h_i^*. The empirical loss is

$$\hat{E}_h = \sum_{i=1}^m E(h(y_i), i)$$

Expanding $E(h_i, i)$ about its minimum gives

$$E(h(y_i), i) - E(h_i^*, i) = ||g_1(y_i) - g_2(y_i)||^2 \, (h(y_i) - h_i^*)^2$$

Summing over i and setting $\rho_i = ||g_1(y_i) - g_2(y_i)||^2$ yields

$$\hat{E}_h = \sum_{i=1}^m \rho_i (h(y_i) - h_i^*)^2 + C$$

where the constant $C = \sum E(h_i^*, i)$ does not depend on h. The ρ_i define the weights for each training data instance.

4. WEIGHTED KERNEL RIDGE REGRESSION

We have written the empirical loss as a weighted sum, where the weights are easily calculated from the training data and the base denoisers g_1 and g_2. Incorporating these weights into the data-fidelity term for the Kernel Ridge Regression algorithm gives a learning method that respects the relative importance of the different training data points. Standard Kernel Ridge Regression is described in detail in [6], and the weighted version has been used in [7].

Weighted Ridge Regression without the use of Kernels is equivalent to performing linear weighted least squares with a quadratic regularization term. Assuming a linear form for the decision function $h(x) = w^T \cdot x$, the Weighted Rigde Regression algorithm choses w to minimize the weighted Ridge loss

$$L(w) = \sum_{i=1}^m \rho_i (w \cdot y_i - h_i^*)^2 + \alpha \, ||w||^2$$

where α is a learning parameter controlling the regularization.

This optimization problem is soluble in closed form. Introducing the data matrix $\mathbf{Y}$, the vector of target values $\mathbf{H}$, and the diagonal matrix P with $P_{ii} = \rho_i$, we can write

$$L(w) = \alpha w^T w + (\mathbf{H} - \mathbf{Y}w)^T P (\mathbf{H} - \mathbf{Y}w)$$

Setting the gradient of L to zero yields the linear weighted Ridge Regression solution for the decision function

$$h(x) = w^T \cdot x = \mathbf{H^T} P \mathbf{Y} \left(\alpha I_d + \mathbf{Y}^T P \mathbf{Y}\right)^{-1} x$$

where I_d is an identity matrix of dimension d.

Like many algorithms in machine learning, Ridge regression may be "Kernelized" by examining the form of the solution of the linear version and noting that the training data appear only through their dot products. Replacing these dot products with a Kernel function $K(y_1, y_2)$ yields a nonlinear version of the algorithm that implicitly maps the input data into a higher, possibly infinite dimensional, space before performing weighted Ridge Regression. Applying the matrix identity $(I + AB)^{-1}A = A(I + BA)^{-1}$, one may rewrite

$$h(x) = \mathbf{H}^T (\alpha I_m + P\mathbf{Y}\mathbf{Y}^T)^{-1} P \mathbf{Y} x$$

As the i, j entry of $\mathbf{Y}\mathbf{Y}^T$ is $y_i \cdot y_j$, we replace it by $\mathbf{K}$ where $\mathbf{K}_{i,j} = K(y_i, y_j)$. Similarly, we replace $\mathbf{Y}x$ by the mx1 vector $\mathbf{k}(x)$ that has ith entry $K(y_i, x)$. With this notation, the weighted Kernel Ridge Regression solution is given by

$$h(x) = \mathbf{H}^T (\alpha I_m + P\mathbf{K})^{-1} P \mathbf{k}(\mathbf{x})$$

5. APPLICATION TO IMAGE DENOISING

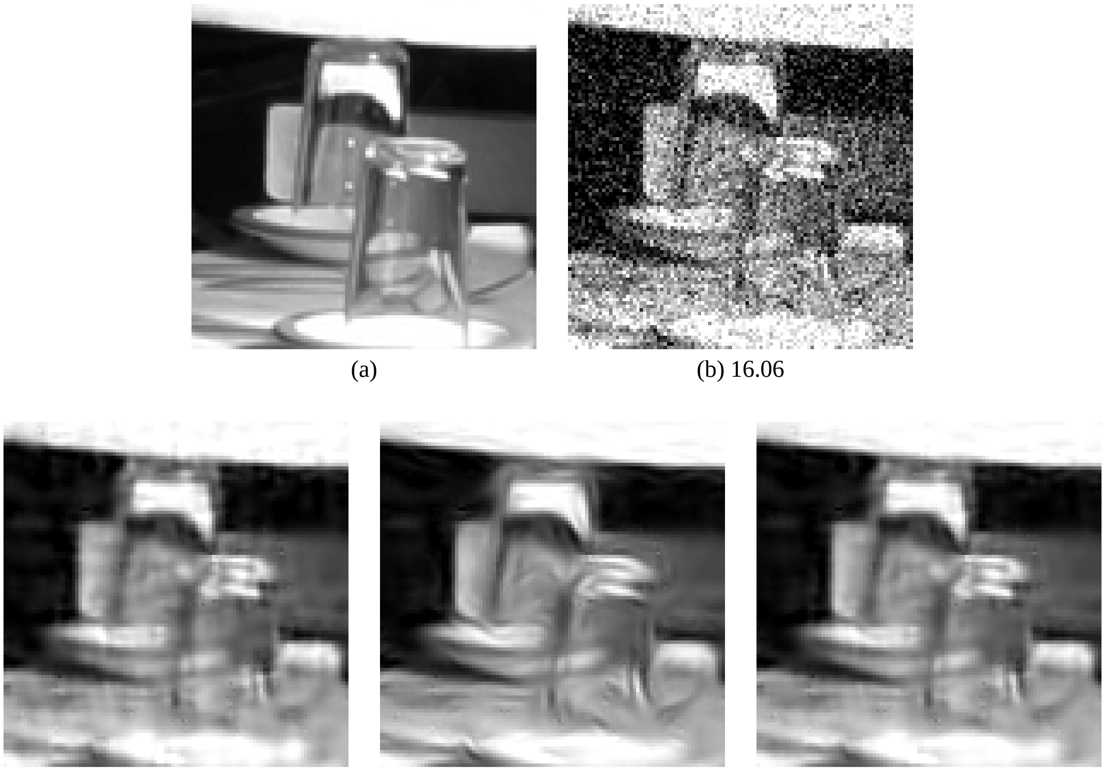
As an example application, we apply these techniques to a pair of image denoising methods based on the Gaussian Scale Mixture (GSM) and Orientation Adapted Gaussian Scale Mixture (OAGSM) models [8, 9] The OAGSM method in general performs well in strongly oriented regions such as edges, but creates oriented artifacts in non-oriented texture regions of the image. Conversely, the GSM often performs better in texture regions and T-junction or corner regions. This complementary nature of the strengths and weaknesses of the two algorithms allows the hybridization to give improvement.

5.1. Image Representation

Both the GSM and OAGSM denoising methods used in this work employ the Steerable Pyramid (SP) representation as a front end transform [10]. The SP transform is an overcomplete multiscale wavelet-type transform where the filters are oriented derivative operators at multiple scales. The SP transform with J orientations and K levels decomposes an image into a single scalar highpass band, K sets of J dyadically subsampled oriented bandpass bands and a residual lowpass band. For this work we use the two band (J=2) pyramid with K=3 scales, in which case the bandpass filters are first derivative filters in the x and y directions, and the coefficients of the oriented bands define the components of the image gradient at multiple scales.

5.2. Oriented and non-oriented denoisers

Both the GSM and OAGSM are Gaussian mixture models for local coefficient patches. Each patch x is described as a zero mean Gaussian with covariance $C(\vec{\tau})$ when conditioned on one or more hidden variables $\vec{\tau}$. In the GSM case $\vec{\tau}$ consists of a single scalar multiplier z and $C(z) = zC_{NOR}$ for a fixed covariance C_{NOR} that is estimated at each scale. For the OAGSM the hidden variables consist of both a scalar multiplier z and a rotation angle θ. One may write


(a) (b) 16.06

(c) 26.12 (d) 26.32 (e) 26.79

Fig. 1. Results with PSNR values - 100x100 detail from image 5, noise σ=40 : (**a**) Original (**b**) Noisy (**c**) GSM (**d**) OAGSM (**e**) Hybrid

$C(z,\theta) = zC_{ORI}(\theta)$ where $C_{ORI}(\theta)$ is the covariance for a fixed Gaussian process that has been rotated by θ. This is equivalent to modeling an image patch as

$$x = \sqrt{z}R(\theta)u$$

where $u \sim N(0, C_{ORI}(0))$ and $R(\theta)$ is an operator that rotates the patch by θ.

Noisy patches y are then modeled as $y = x + n$ where the noise n is a zero mean Gaussian with covariance C_n. In this paper the noise is considered to be Gaussian white noise of known covariance in the pixel domain. The noise covariance for each subband is shaped by the transform but is easily computed from the SP filters.

For both the GSM and OAGSM models, the covariance matrices C_{NOR} and $C_{ORI}(\theta)$ can be computed from the noisy image data once the noise covariance is known. For the GSM model, C_{NOR} may be calculated simply by taking the sample covariance of vectorized noisy patches, and subtracting off the noise covariance C_n. For the OAGSM model, the oriented covariances $C_{ORI}(\theta)$ are computed by first measuring the dominant orientation of each noisy patch, and "rotating out" each patch by its dominant orientation. Rotating these orientation-normalized patches by a fixed value of θ, taking their sample outer product, and subtracting off C_n then gives the oriented covariances $C(\theta)$.

The dominant orientation of each coefficient patch is calculated directly from the two-band SP coefficients. Viewing the coefficients as a set of two-dimensional image gradient vectors $\{v_i\}_{i=1}^{d}$, the dominant orientation of the patch is

$$\phi^* = \underset{\phi}{\operatorname{argmax}} \sum_{i=1}^{d} (w(\phi) \cdot v_i)^2 = \underset{\phi}{\operatorname{argmax}}\, w(\phi)^T M w(\phi)$$

where $w(\phi)$ is the unit vector $(\cos(\phi), \sin(\phi))^T$, and M is the "orientation response matrix" $\sum v_i v_i^T$. The ratio $\nu = \lambda_1/\lambda_2$ of the eigenvalues of this matrix M measures how strongly oriented each patch is. By restricting the patches used for estimating of $C_{ORI}(\theta)$ to those having ν above a certain threshold ν^*, we can further specialize the OAGSM denoising algorithm for highly oriented regions. This "winnowing" of the oriented patches helps the resulting hybrid denoiser, even though the overall performance of the base OAGSM denoiser suffers slightly.

Both of these models have the property that when conditioned on the hidden variables, the signal and noise are zero mean Gaussian with covariances $C(\vec{\tau})$ and C_n, respectively. This allows a simple closed form for the Bayesian Minimum Mean Square Estimator (MMSE) of the original patch $\vec{x}$

$$\hat{x}(y;\vec{\tau}) = C(\vec{\tau})(C(\vec{\tau}) + C_n)^{-1} y$$

The full MMSE estimator is obtained by integrating the conditional estimators over the hidden variables, weighted by the probability of the hidden variables given noisy data, yielding

$$\hat{x}(y) = \int \hat{x}(y;\vec{\tau}) p(\vec{\tau}|y) d\vec{\tau}$$

where the weighting $p(\vec{\tau}|y)$ may be calculated from the prior on $\vec{\tau}$. While both methods provide an estimate of the entire patch, only the estimate of the center coefficient is kept.

6. RESULTS

The hybrid denoising procedure was applied to a collection of five 256x256 pixel test images that were corrupted with pseudorandom

	Im 1	**Im 2**	**Im 3**	**Im 4**	**Im 5**
Noisy	28.10	28.10	28.10	28.10	28.10
GSM	34.47	34.05	35.64	32.26	34.00
OAGSM	34.51	34.38	35.79	31.44	34.13
Hybrid	**34.78**	**34.61**	**36.17**	**32.65**	**34.90**
Gain	0.27	0.23	0.38	0.39	0.77

Noisy	22.08	22.08	22.08	22.08	22.08
GSM	31.65	30.97	31.55	28.59	29.90
OAGSM	31.83	31.62	31.56	28.29	30.17
Hybrid	**32.13**	**31.81**	**32.01**	**29.02**	**30.81**
Gain	0.30	0.18	0.45	0.43	0.64

Noisy	16.06	16.06	16.06	16.06	16.06
GSM	28.88	27.59	27.75	25.62	26.12
OAGSM	29.03	28.38	27.58	25.88	26.32
Hybrid	**29.35**	**28.39**	**28.10**	**26.03**	**26.79**
Gain	0.32	0.02	0.35	0.14	0.47

Table 1. Table of PSNR values for denoising results, starting from 3 different noise levels **top** σ = 10 **middle** $\sigma = 20$ **bottom** $\sigma = 40$

Gaussian white noise. Original image pixel values ranged between 0 and 255. For these numerical experiments, training and test image pairs were generated from a single 512x512 image by taking two non overlapping 256x256 subimages.

The noisy training and test images, as well as the clean training image were decomposed using the Steerable Pyramid representation with 3 scales and 2 orientation bands. 5x5 patches including one pair of "parent" coefficients at the coarser scale are used, so each patch may be viewed as a vector in $\mathbb{R}^{52}$. The noisy training image was denoised with both the OAGSM and GSM denoising methods, and at each location in space and scale the decision function target value h_i^* was computed, as described in section (3.1). OAGSM covariances were formed using winnowing, with threshold ν^* set to the 85^{th} percentile value at each scale. Distinct decision functions h were learned for each image and at each scale. To form training features, the noisy patches were extracted at each of the 3 scales. These noisy patches were then rotated by their dominant orientation and the 52 coefficients of these rotated noisy patches were taken as the feature vectors for the learning problems. At each image scale, the rotated patch features were scaled by a divisive constant to lie in the range [-1,1]. This gave 65536 training examples for at the first scale, 16384 training examples at the second scale and 4096 training examples at the third scale. Due to high computational cost, the training examples at the first and second scale were pruned to the 5000 with largest weights.

Gaussian kernels of the form $K(x_i, x_j) = e^{-\gamma||x_i - x_j||^2}$ were used for the weighted Kernel Ridge Regression. Different values of the learning parameters α and γ were used for different image scales and noise levels, however the same parameters were used across the different images. The learning parameters were selected by four-fold cross-validation on a single training image, using the 3000 training examples with greatest weights at each scale. Cross-validation was done for each point of a logarithmically spaced grid with $\alpha = [2^0, 2^1, ..., 2^{10}]$ and $\gamma = [2^{-5}, 2^{-4}, ..., 2^5]$ and the parameters yielding the lowest cross-validation error were selected.

Both the OAGSM and GSM denoising estimates were computed for the noisy test images. Each test image patch was then rotated by its dominant orientation and rescaled according to the divisive constants calculated during training. The learned decision function for the appropriate scale and noise level was then evaluated on these rotated patches, and used to combine the OAGSM and GSM estimates to give a hybrid estimator for each of the 3 scales. Denoising results reported by PSNR are given in table 1, with the hybrid method showing significant improvement over the GSM and OAGSM methods. Details for a single image are shown in Figure 1.

7. DISCUSSIONS / CONCLUSIONS

We have developed a general framework for combining two local denoising methods and applied the method to the GSM and OAGSM image denoising algorithms. The resulting hybrid denoiser shows noticeable improvement in both signal-to-noise ratio and visual quality. The OAGSM method tends to introduce oriented artifacts in image regions that are not oriented, as well as in T-junction regions. These artifacts are noticeably reduced in the hybrid denoised results.

Although the OAGSM and GSM base denoising methods used in this paper were in fact quite similar in their internal details, the only requirement for hybridization was that they operated on the same dimension of input patches, and produced the same dimension output center coefficient estimates. Throughout the signal processing literature, a large number of very different denoising methods have been developed, many of which have different strengths and weaknesses. The general learning-based combination methodology presented here may allow significant improvement for denoising a wide class of signals by combining different well developed methods already in existence.

8. REFERENCES

[1] S Grace Chang, Bin Yu, and Martin Vetterli, "Spatially adaptive wavelet thresholding with context modelling for image denoising," *IEEE Transactions on Image Processing*, vol. 9, no. 9, pp. 1522–1531, 2000.

[2] C R Jung and J Scharcanski, "Adaptive image denoising and edge enhancement in scale-space using the wavelet transform," *Pattern Recognition Letters*, vol. 24, pp. 965–971, 2003.

[3] H Cheng, Q Yu, J Tian, and J Liu, "Image denoising using wavelet and support vector regression," in *Proc. of the 3rd International Conf on Image and Graphics*, 2004, pp. 43–46.

[4] B Sun, D Huang, and H Fang, "Lidar signal denoising using least-squares support vector machine," *IEEE Signal Processing Letters*, vol. 12, pp. 101–104, 2005.

[5] Tzu-Chao Lin and Pao-Ta Yu, "Adaptive Two-Pass Median Filter Based on Support Vector Machines for Image Restoration," *Neural Comp.*, vol. 16, no. 2, pp. 333–354, 2004.

[6] Nello Cristianini and John Shawe-Taylor, *An Introduction to Support Vector Machines and other kernel-based learning methods*, Cambridge University Press, 2000.

[7] G Saon, "A nonlinear speaker adaptation technique using kernel ridge regression," in *IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2006.

[8] J Portilla, V Strela, M Wainwright, and E P Simoncelli, "Image denoising using a scale mixture of Gaussians in the wavelet domain," *IEEE Trans. Image Processing*, vol. 12, no. 11, pp. 1338–1351, November 2003.

[9] D Hammond and E P Simoncelli, "Image denoising with an orientation-adaptive gaussian scale mixture model," in *Proc. of ICIP*, 2006.

[10] E P Simoncelli and W T Freeman, "The steerable pyramid: A flexible architecture for multi-scale derivative computation," in *Proc. of ICIP*, Washington, DC, October 1995, vol. III, pp. 444–447.