Nonlinear Blind Sensor Fusion and Identification

Sam T. Roweis
Department of Computer Science, University of Toronto
Toronto, Canada M5S 3G4
roweis@cs.toronto.edu

***Abstract*— When several uncharacterized sensors measure the same unknown signal or image, we would like to simultaneously combine the measurements into an estimate of true source (fusion) and learn the properties of the individual sensors (identification). This paper presents a model in which sensors perform (time-invariant) linear filtering followed by pointwise nonlinear squashing with additive noise and shows how, given several such noisy nonlinear observations, it is possible to recover the true signal and also estimate the sensor parameters. The setup assumes that both the linear filtering and the nonlinear squashing are spatially (temporally) invariant, but does not make any prior assumptions (such as smoothness, sparsity or heavily tailed marginals) about the signal being recovered and thus is appropriate for a variety of source distributions, such as astronomical images, speech signals and hyperspectral satellite data which may violate one or more standard prior assumptions. An efficient estimation algorithm minimizes the sum of squared errors between the predicted sensor outputs and the sensor readings actually observed, using an efficient procedure, isomorphic to the backpropagation algorithm. The setup can be thought of as learning the weights *and* unknown common input for several one-layer neural networks given their outputs.**

I. Simultaneous Identification and Fusion of Multiple Sensors

In this paper I study the problem of combining information from several noisy sensors, all measuring the same signal. For example, consider several simultaneous recordings of the same audio stream, each one taken by a microphone that has some unknown (but time invariant) filtering and distortion properties. Can we recover a "clean" recording of the original audio and simultaneously identify the properties of the microphones? Similarly, if we have several images each of which has been blurred, edge detected, or otherwise filtered, can we estimate both the original image and the properties of the instruments or channels that delivered the altered versions? This problem, which is one of simultaneous sensor fusion and sensor identification, is quite general. But if we make some fairly strong assumptions about the properties of the sensors, then we can make substantial analytic progress.

Sensor fusion and identification is related to, but not exactly the same as, several other problems of interest in statistical signal processing and machine learning. In blind deconvolution, super-resolution and adaptive denoising, we are given a single sensor measurement and asked to recover the original signal (and perhaps also to estimate the sensor properties). This is a much more ill posed problem which requires making some a priori assumptions about the statistics of the signals to be recovered (and possibly the sensor properties as well). In Independent Components Analysis (ICA)[5] we are given multiple measurements but we are also trying to determine several underlying signals or sources. Since the objective function for ICA is typically some estimate of independence, prior assumptions are once again required about the marginal statistics of the sources. Furthermore, the concept of independence as a driving contrast function does not apply when there is only a single underlying source to be recovered.

II. Mathematical formulation: regression with shared unknown input

To begin, assume that we observe M sensor signals of length T samples each. The entire setup is also applicable to two-dimensional signals (images); we can also handle the case in which the observed signals are different lengths/sizes. Each observed signal (image) $\boldsymbol{y}^m = [y_1^m, \ldots, y_t^m, \ldots, y_T^m]$ is modeled as a linearly filtered version of some unknown true signal $\boldsymbol{s} = [s_{2-K}, \ldots, s_1, \ldots, s_t, \ldots, s_T]$ followed by a pointwise nonlinearity and additive noise:

$$\begin{aligned} z_t^m &= \sum_{k=0}^{K-1} w_k^m s_{t-k} \\ \hat{y}_t^m &= \alpha_m f(b_m + z_t^m) + \beta_m \\ p(y_t^m|\hat{y}_t^m) &= \mathcal{N}(y_t^m|\hat{y}_t^m, \sigma_m^2) \end{aligned}$$

where $\mathcal{N}(x|\mu, \sigma^2)$ is a Gaussian distribution on x with mean μ and variance σ^2.

The linear filters are assumed to be time (space) invariant and to have compact support K; this is a crucial regularization in the model. We do not use either periodic boundary conditions or zero padding in the linear filtering so that the observed sensor signals $\hat{\boldsymbol{y}}^m$ are $K{-}1$ samples shorter than the true signals $\boldsymbol{s}$. Figure 1 illustrates the generative model.

We model the nonlinearity using a fixed squashing function $f(\cdot)$ (such as the tanh or sigmoid) with scaling and offsets applied both before and after the squashing. Because these squashing functions both asymptote at large inputs and behave linearly for small inputs, when combined with input and output scaling and offset they allow us to capture quite a wide range of monotonic nonlinearities, including saturation, clipping, thresholding around a specific value, as well as simple linear behaviour.

By assuming temporal (spatial) invariance of the filters and pointwise action of the nonlinearity, we are able to learn in this multiple sensor model, even in the complete absence of

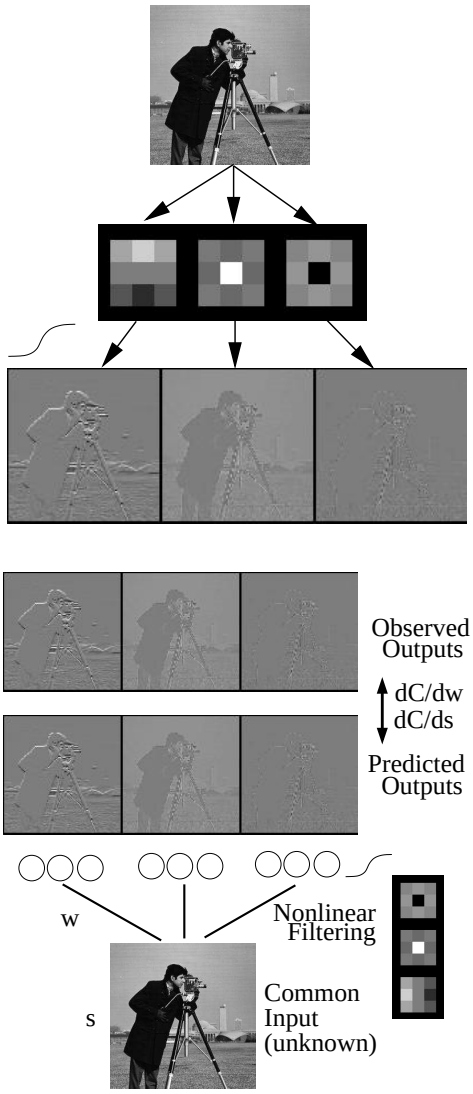

Fig. 1. Generative model (top) creates each observed sensor signal by (spatially homogeneous) linear filtering using a compact kernel, followed by pointwise nonlinear squashing, followed by additive noise. Learning (bottom) is like training an ensemble of networks with a common unknown input and restricted weights.

priors on the unknown signal (such as sparsity, smoothness, or heavy tailed marginals). The objective function is simply the log likelihood of the observed sensor signals given the hypothesized true signal and sensor parameters. Maximum likelihood learning in this model is equivalent to minimizing the sum of squared errors between our predicted sensor outputs $\hat{y}_t^m$ and the sensor readings y_t^m we actually observed:

$$C = \frac{1}{2}\sum_{mt}(\hat{y}_t^m - y_t^m)^2/\sigma_m^2 \tag{1}$$

with respect to the unknown signal $\boldsymbol{s}$, and the filtering parameters $\{\boldsymbol{w}^m, \alpha_m, \beta_m, b_m\}$ of each sensor.

From this it can be seen that this problem is isomorphic to that of training an ensemble of nonlinear regressors, all of which share the same missing input. (Of course, it is also possible to estimate the noise levels σ_m^2 at the same time as fitting the other model parameters, although we do not give the details here.)

Gradient equations for multiple nonlinear sensor model

$$\begin{aligned}
\frac{\partial C}{\partial s_t} &= \sum_m \alpha_m \sum_{k=0}^{K-1} d_{t+k}^m w_k^m \\
\frac{\partial C}{\partial w_k^m} &= \alpha_m \textstyle\sum_t d_t^m s_{t-k} \\
\frac{\partial C}{\partial b_m} &= \alpha_m \textstyle\sum_t d_t^m \\
\frac{\partial C}{\partial \alpha_m} &= \frac{1}{\alpha_m}\sum_t (\hat{y}_t^m - y_t^m)(\hat{y}_t^m - \beta_m) \\
\frac{\partial C}{\partial \beta_m} &= \textstyle\sum_t (\hat{y}_t^m - y_t^m)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial C}{\partial \boldsymbol{s}} &= \sum_m \alpha_m \left(\boldsymbol{d}^m \otimes \boldsymbol{w}^m\right) \\
\frac{\partial C}{\partial \boldsymbol{w}^m} &= \alpha_m \left(\boldsymbol{d}^m * \boldsymbol{s}\right) \\
\frac{\partial C}{\partial b_m} &= \alpha_m \left(\mathbf{1}^\top \boldsymbol{d}^m\right) \\
\frac{\partial C}{\partial \alpha_m} &= \frac{1}{\alpha_m}(\hat{\boldsymbol{y}}^m - \boldsymbol{y}^m)^\top(\hat{\boldsymbol{y}}^m - \beta_m) \\
\frac{\partial C}{\partial \beta_m} &= \mathbf{1}^\top(\hat{\boldsymbol{y}}^m - \boldsymbol{y}^m)
\end{aligned}$$

Fig. 2. On the right we have provided the vector versions of the equations: $\boldsymbol{a} * \boldsymbol{b}$ is used to denote convolution (without zero padding or circular boundary conditions), $\boldsymbol{a} \otimes \boldsymbol{b}$ to denote correlation (i.e. convolution with the time-reversed second argument) and $\mathbf{1}$ is a stencil of all ones having length T. On the left, sums over t run from 1 to T and sums over m run from 1 to M. For simplicity we have assumed equal output noise variances $\sigma_m^2 = \text{const}$.

There are many possible approaches to minimizing the above objective, but most of them require computing the gradient of C with respect to the unknown signal $\boldsymbol{s}$ and the sensor parameters $\{\boldsymbol{w}^m, \alpha_m, \beta_m, b_m\}$. To compute this vector of derivatives, it is useful to define an intermediate "scaled difference signal" $\boldsymbol{d}^m$:

$$d_t^m = (\hat{y}_t^m - y_t^m) f'(z_t^m + b_m) \tag{2}$$

which is extended to $t = (2-K)\ldots(T+K-1)$ by defining $d_t^m = 0$ for $t < 1$ and $t > T$. (Here $f'(x)$ denotes the derivative of f evaluated at x.) Given this difference signal, the gradient computations are all linear, as given below.

Many of the intermediate quantities used in computing $\hat{y}_t^m$ can be reused in the computation of the gradients, in just the same way that backpropagation stores intermediate quantities

- Initialize $C = 0$, $\partial C/\partial \boldsymbol{s} = \mathbf{0}$
- Loop over $m = 1 \ldots M$
  1) Compute $\boldsymbol{z}^m = \boldsymbol{w}^m * \boldsymbol{s}$
  2) Overwrite z_t^m in place: $z_t^m \leftarrow f(z_t^m + b_m)$
  3) Compute $\boldsymbol{d}^m = \alpha_m \boldsymbol{z}^m + \beta_m - \boldsymbol{y}^m$
  4) Increment $C \mathrel{+}= (\boldsymbol{d}^m)^\top \boldsymbol{d}^m$
  5) Output $\partial C/\partial \alpha_m = (\boldsymbol{d}^m)^\top \boldsymbol{z}^m$
  6) Output $\partial C/\partial \beta_m = \mathbf{1}^\top \boldsymbol{d}^m$
  7) Overwrite in place: $d_t^m \leftarrow d_t^m (1 - (\bar{z}_t^m)^2)$
  8) Output $\partial C/\partial b_m = \alpha_m (\mathbf{1}^\top \boldsymbol{d}^m)$
  9) Output $\partial C/\partial \boldsymbol{w}^m = \alpha_m \, (\boldsymbol{d}^m * \boldsymbol{s})$
  10) Increment $\partial C/\partial \boldsymbol{s} \mathrel{+}= \alpha_m \, (\boldsymbol{d}^m \otimes \boldsymbol{w}^m)$

Fig. 3. Pseudo-code for efficient gradient computation in the multiple sensor model. This example is one-dimensional, uses the derivative of the tanh nonlinearity (in step 7) and assumes all output noise variances are equal. Until step 7, $\boldsymbol{d}^m$ holds intermediate quantities different from its definition in eq(2). The computation is very similar to backpropagation in several networks all sharing the same unknown input.

to efficiently compute the learning signal in multilayer perceptron regression machines. A reasonably efficient procedure for computing the value of the cost and its gradient is given above, using $f(\cdot) = \tanh(\cdot)$ as an example. Also, since many of the derivatives above can be expressed as convolutions or correlations, they may be more efficiently computed in a transform domain using, for example, the FFT (taking care to correctly respect or correct for the lack of periodic boundary conditions and zero padding).

III. Extensions to the basic model

The equations presented above assume one-dimensional signals, equal sized observations and known output noises and filter sizes. It is possible to extend the basic model to account for a variety of more complex situations as outlined briefly below.

Two dimensional signals & different sized observations: Everything above goes through trivially for two dimensions, i.e. for images. The relationship between convolution and correlation must be carefully defined in two dimensions: for the equations above to be correct correlation must be defined as convolution with the second argument vertically and horizontally flipped (not with the transpose of the second argument). As before, the difference image $\boldsymbol{d}^m$ is extended by zero padding its perimeter.

The above formulation also goes through essentially unchanged if some of the sensors do up/downsampling, i.e. produce outputs of size shorter (smaller) or longer (larger) than the true signal. The equations defining the sensor outputs y_t^m are simply defined by making the linear filtering include a resampling operation on the true signal; similarly for the summations used to compute the derivatives. The result of this resampling plus filtering is the size of the observed sensor output. If all channels do some degree of downsampling, then at best we can estimate the signal at the horizontal/vertical resolution of the sensor with the highest horizontal/vertical Nyquist rate (and maybe even not that well); the setup in this paper cannot automatically do super-resolution.

Learning output noise variances and filter orders: The output noise level σ_m^2 of each sensor can easily be learned as part of the gradient procedure by computing the derivative of the cost with respect to its logarithm: $\partial C/\partial \log \sigma_m^2$. It is also possible to estimate a nonhomogeneous noise level at different temporal (spatial) locations but in that case care must be taken to avoid overfitting.

The only structural parameter that needs to be specified is the filter order K. This parameter controls the capacity or complexity of the model we are fitting. Although the above formulation assumes K to be the same for all m, there is essentially no difficulty in rederiving the setup with a different order for each sensor.

If K (or K_m) is known then the above equations completely specify the optimization to be performed. If not, then K needs to be estimated using, for example by a discrete search over K using cross-validation as a selection criterion, although not in the naive way. To implement cross-validation one usually removes some sensor samples t' from the summation in the cost function when computing the gradients and uses the prediction error on those samples as a validation signal to estimate the filter order. However, in this case one should set aside a different set t'_m from each sensor, or else the underlying signal $\boldsymbol{s}$ will not be determined at all for the set t'. Furthermore, selecting holdout samples completely at random selecting samples does not work as well, because overfitting effects of higher filter orders do not manifest themselves as severely when validation points are scattered and isolated as when they are in a large block that is not part of the training signal. In preliminary experiments, I have had some success with setting aside a continuous block of samples (of order 10% of the signal length) from each sensor as a validation set, but ensuring that these validation blocks are non-overlapping from sensor to sensor.

Another possibility is to perform learning with a large maximum filter order K_m but to penalize filter energy away from the centre of the stencil in order to encourage compact support as much as possible. This requires some way of setting the penalty hyperparameters, which in certain analytically tractable cases can be performed with empirical Bayes (ML type II) methods.

Registration: One serious limitation of the current model is that it requires all signals to be exactly temporally/spatially aligned. While this may be appropriate for, e.g. multiple exposures of the sky in astronomical images, microphones in close proximity or tripod mounted cameras there are often situations in which sensors introduce registration shifts. Small constant shifts can be dealt with by learning with a relatively large filter size K; the model then has some freedom as to where inside the stencil it places energy. Extending the model to include learning large shifts is mathematically possible but in practice gradient methods do not work well for discovering registrations larger than the highest frequencies in the signal. One possibility is

to compute the gradient of the sensor offsets using a blurred version of the signal, in which the gradients have effects over a much larger extent.

IV. Experiments with Images and Speech

I have applied the model described above to fuse and identify several examples of synthetic (known) and real (unknown) sensors. All of the experiments reported used the tanh function as the nonlinearity (although the sigmoid would have been exactly equivalent) and conjugate gradients as an optimization routine.

First, the model was applied to artificial sensors synthesized by applying exactly the generative model process to input images. Images were filtered using K=3x3 kernels followed by nonlinear pixel squashing and additive Gaussian noise. The kernels were chosen to perform common image processing operations such as blurring and edge detection. Figure 4 shows three examples, with the original image, the true filters and the observed sensors on the top of each panel and the recovered image, estimated filters and approximate sensor outputs on the bottom of each panel.

The model was also applied to real signal sets for which the true sensor characteristics are unknown. As an illustration of the fact that a special case of this model is exposure correction, we processed several sets of multiple exposure frames using a K=1x1 kernel that did not allow any spatial filtering. (Although the experiment has also been performed with K=3x3 and 5x5 kernels which learn mostly delta-function like peaks, often offset from the centre inducing slight registration alignments.) Figures 5-6 show the original input images along with the estimated true scene and recovered exposure curves learned by the model for each sensor. The estimated true image contains information not simultaneously visible in any of the input scenes, and the exposure curves show reasonable estimates of underexposure and overexposure with saturation at high and low intensities.

Finally, I have applied the model to speech signals consisting of simultaneous recordings of the same audio scene by several microphones. The data contains wide field microphones, which are used as the inputs (unknown sensors) as well as a close-talking microphone worn by the speaker which is used as the "true" or reference signal for comparison. (Relative delays have been removed so that the signals are temporally aligned to within a few samples.) Figure 7 shows two wide field recordings (inputs) as well as the recovered clean audio signal and the reference microphone as a comparison.

V. Discussion: Identifiability and Linear-Noiseless Case

Model Identifiability:

There is a trivial way in which the model presented above is not exactly identifiable: there exists a scale and offset degeneracy between the output amplitude (intensity) and the unknown true signal amplitude. In particular, the signal can be multiplied by a constant and amplitude shifted, and all of the α_m, β_m can be appropriately rescaled and translated to exactly cancel this. To break such a degeneracy, we always constrain the unknown signal to have zero mean and unit variance. This is easy to enforce during the gradient computations by removing the mean of the gradient and rescaling the signal and filters after each update. For large filter sizes, there is potentially also a similar temporal (spatial) shift degeneracy.

However, there is a potentially more serious non-identifiability related to the richness of the sensors and the signal. The signal must have significant power at all frequencies and the sensors must collectively cover frequency space with some spectral overlap between them. (For a purely linear model, these concepts can be formalized in terms of the rank of the lagged data matrix and the common zeros in the z-transform of the filters; but for this nonlinear model the equivalent formal characterization of identifiability is unknown.)

Linear Noiseless Sensors:

In the special case where f is linear, the problem studied in this paper reduces to the problem of blind linear channel identification, which has been studied previously in the signal processing literature[3], [2], [4]. This literature has tended to focus on the linear algebraic structure of the problem, identifying various matrix pencils, subspaces and Sylvester type conditions that must be satisfied for identifiability. It is important to notice that even in the absence of the nonlinearity $f(\cdot)$, this problem still represents a bilinear form in $\boldsymbol{s}$ and $\{\boldsymbol{w}^m\}$. One possible strategy is to minimize the cost C using coordinate descent, by first fixing the unknown signal $\boldsymbol{s}$ and solving for all of the filters $\{\boldsymbol{w}^m\}$ using standard deconvolution and then fixing the filters and solving for the signal using linear least squares. Even in a purely linear setting this approach is not computationally attractive, since at each iteration deconvolution must find the values of the filter coefficients by solving a large system of linear equations with the constraint of a Toeplitz structured coefficient matrix. Solving such systems efficiently is tricky and numerically poorly conditioned; even if done correctly the resulting iterations are usually slower than direct optimization of the cost using, for example, conjugate gradients.

If in addition to assuming linearity we also make the extremely strong and unrealistic assumption that there is no noise at the sensors, it is possible to reformulate the optimization as a single system of linear equations (as opposed to a bilinear cost to be minimized). This is achieved by appealing to the associative-commutative property of the convolution, and noticing that in the absence of noise, $\boldsymbol{y}^m * \boldsymbol{w}^n = \boldsymbol{y}^n * \boldsymbol{w}^m$ exactly for all pairs m, n[3]. Enforcing all of these constraints results in a very large, but linear system of equations for $\boldsymbol{w}^m$ in terms of $\pm\boldsymbol{y}^m$. In principle, this system could be solved for the $\boldsymbol{w}^m$ and then subsequently these could be used to find $\boldsymbol{s}$. However, this setup can be extremely poorly conditioned unless many pairs of sensors with significant spectral overlap exist; furthermore it cannot be extended to nonlinear estimation. Thus, this "trick" for reducing the bilinear cost function down to a linear problem is mainly of theoretical rather than practical interest.

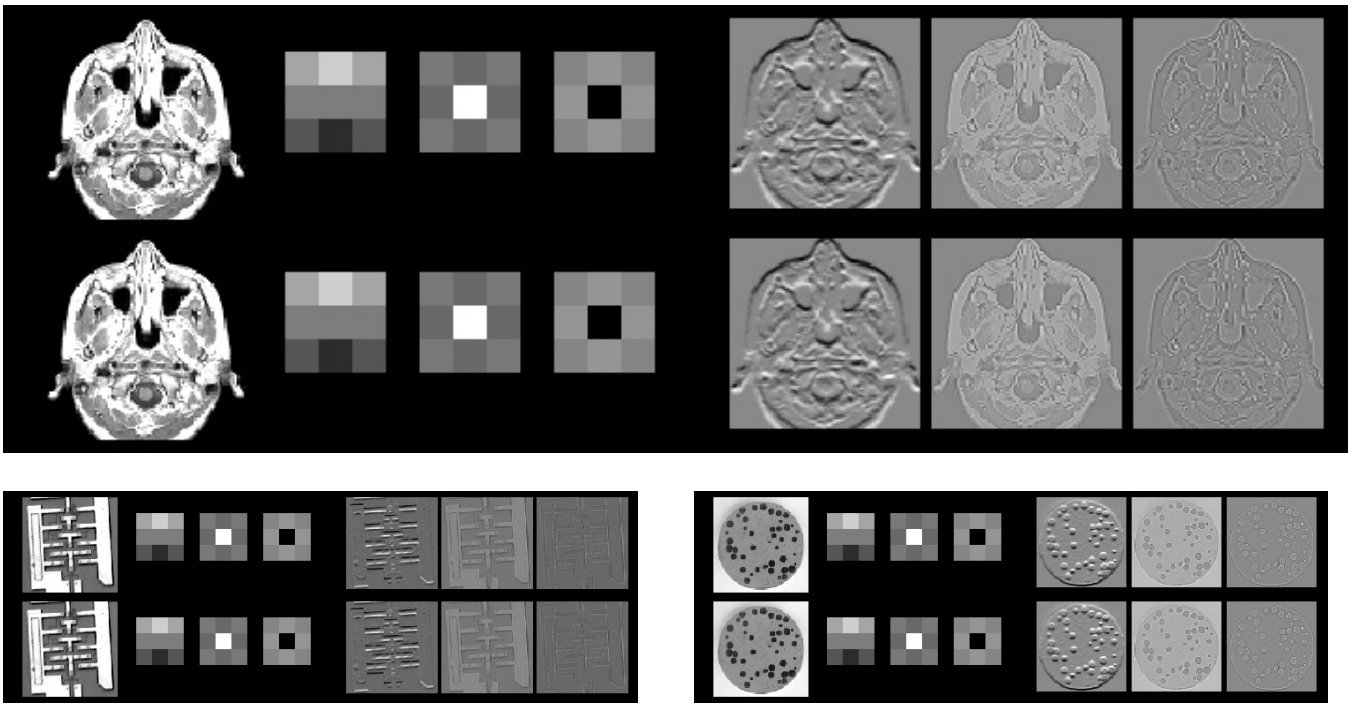

Fig. 4. Synthetic sensors. Images were filtered using K=3x3 kernels followed by nonlinear pixel squashing and additive Gaussian noise. In each panel, the top row shows (from left to right) the original image, true filters and observed sensors; the bottom row shows the recovered image, estimated filters and approximate sensor outputs. The scale degeneracy has been removed for this display by scaling everything so that the recovered image has the same minimum and maximum as the true image. For the top example, the RMS intensity of the true image is 0.58, and the RMS of the residual (innovation) is $<$0.02; other SNR rates are similar.

VI. Conclusions

I have presented a simple model for explaining multiple observations of a single signal, taken by sensors with unknown properties. By assuming that the sensors perform spatially invariant linear filtering (using compact support kernels) followed by pointwise nonlinearity and additive Gaussian noise it is possible to formulate the simultaneous estimation of the true unknown signal — sensor fusion – and learning of the sensor properties — identification — as a minimization problem analogous to training an ensemble of one-layer neural networks with a shared but unknown common input.

By modeling the nonlinear squashing using a flexible monotonic transformation we can include linear models as a special case, as well as capture saturation and clipping effects when they are present. By restricting the linear filtering to have a kernel size of 1, we can also estimate scalar response (exposure) curves independent of filtering effects. Once the sensor properties have been learned they can be used to calibrate single readings or to simulate plausible sensor readings given a novel input.

Results on real and synthetic data show that the model is capable of learning both a high quality estimate of the true signal as well as identifying the unknown sensor properties.

References

[1] *Radiometric Self Calibration*, T. Mitsunaga & S. K. Nayar, ICCV, 1999.

[2] *On subspace methods for blind identification of single-input multiple-output FIR systems*, K. Abed-Meraim, J.F. Cardoso, A.Y. Gorokhov, P. Loubaton & E. Moulines, IEEE Trans. Signal Processing v.45, 1997.

[3] *A Least-Squares Approach to Blind Channel Identification*, G. Xu, H. Liu, L. Tong & T. Kailath, IEEE Trans. Signal Processing v.43, 1995.

[4] *Subspace methods for the blind identification of multichannel FIR filters*, E. Moulines, P. Duhamel, J.F. Cardoso & S. Mayrargue, ICASSP v.4, 1994.

[5] *An information maximisation approach to blind separation and blind deconvolution*, A.J. Bell & T.J. Sejnowski, Neural Computation v7(6), 1995.

Acknowledgments

Thanks to Allan Jepson and Miguel Carreira-Perpinan for detailed comments on draft versions. STR is supported in part by the LEARN project of IRIS and by NSERC Canada.

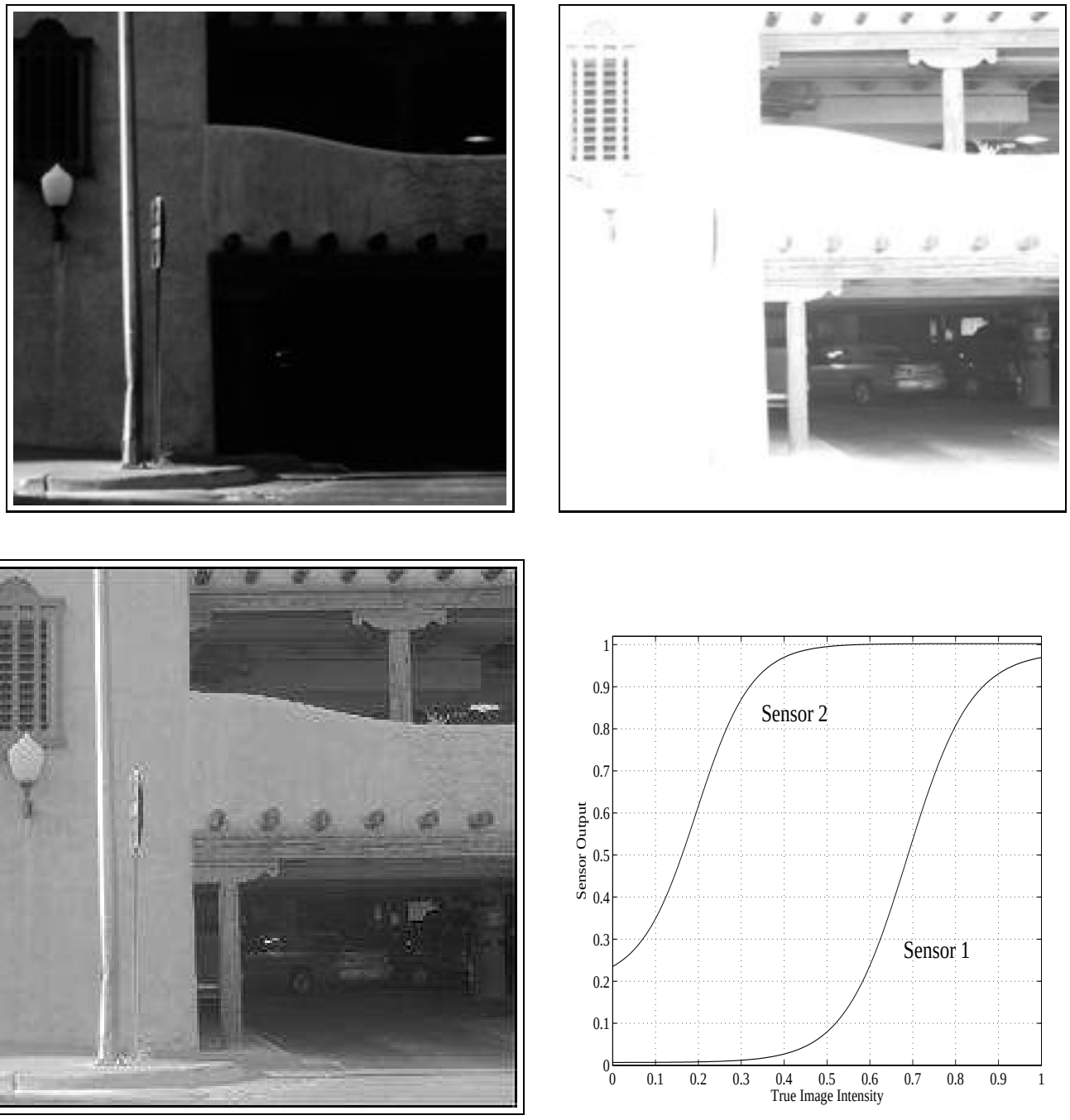

Fig. 5. Exposure correction using a K=1x1 kernel. The two original input images appear in the top row, and the estimated true scene below them on the left. The estimated true image contains information not simultaneously visible in any of the inputs. On the bottom right are the learned exposure curves for each sensor showing camera output as a function of true image intensity. Images were downloaded from the RASCAL[1] page at `http://www1.cs.columbia.edu/CAVE/tomoo/RRHomePage/rrslrr.html`

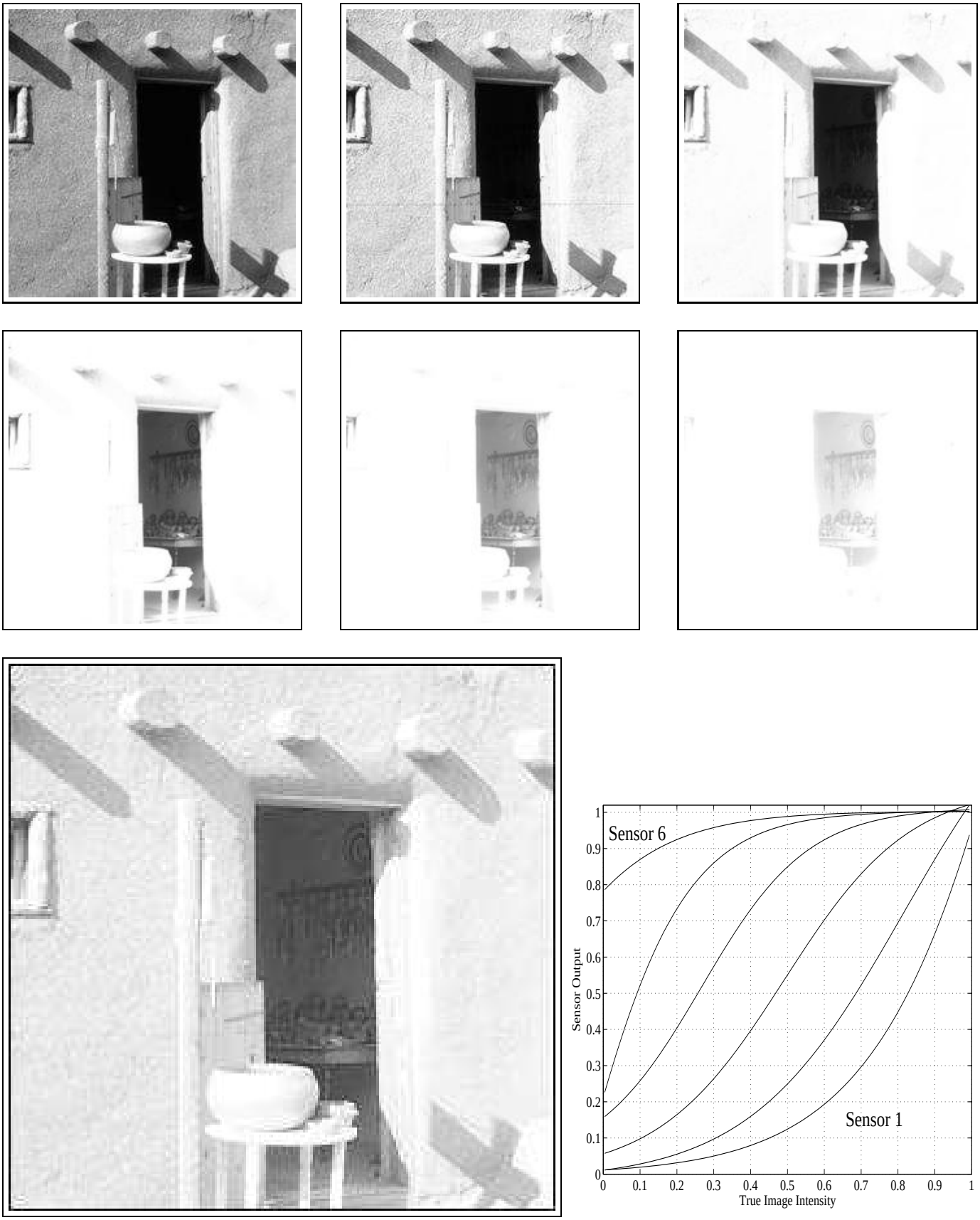

Fig. 6. Simultaneous exposure correction and blur estimation using a K=3x3 kernel. The six original input images appear in the first two rows. The estimated true scene appears in the third row on the left, and contains information not simultaneously visible in any of the inputs. On the bottom right are the learned exposure curves for each sensor showing camera output as a function of true image intensity; spatial kernels (not shown) which estimate the blurring properties of each exposure are also learned. Images were downloaded from the RASCAL[1] page at `http://www1.cs.columbia.edu/CAVE/tomoo/RRHomePage/rrslrr.html`

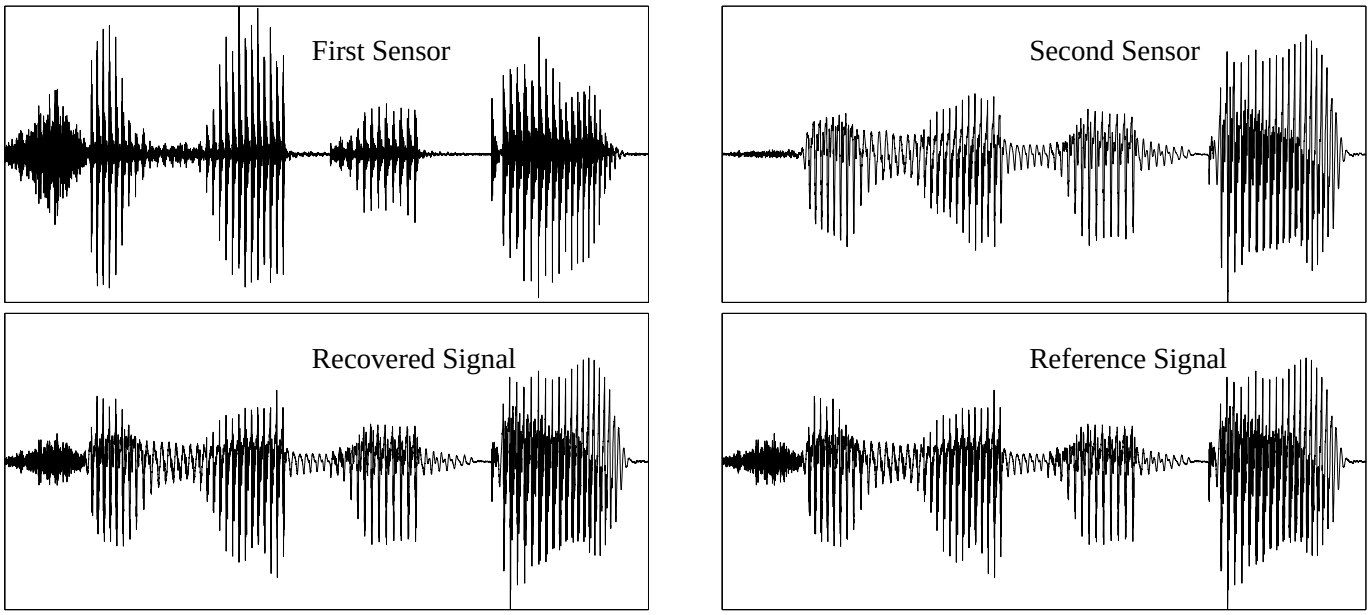

Fig. 7. Microphone fusion. Audio data containing simultaneous recordings of the same audio scene by several microphones are processed by the model. Wide field microphones are used as inputs from the unknown sensors (top); the true signal is estimated (bottom left) and compared to a close-talking microphone worn by the speaker as a reference (bottom right). Original signals were downsampled to 12.5kHz sample rate and filter taps of $K = 11$ (about 1ms) were learned. For display purposes the recovered signal has been sign flipped and scaled to match the amplitude of the reference signal.