STATISTICAL AND INFORMATION-THEORETIC METHODS FOR SELF-ORGANIZATION AND FUSION OF MULTIMODAL, NETWORKED SENSORS

John W. Fisher III
Martin J. Wainwright
Erik B. Sudderth
Alan S. Willsky

MASSACHUSETTS INSTITUTE OF TECHNOLOGY, USA

Abstract

The appeal of distributed sensing and computation is matched by the formidable challenges it presents in terms of estimation and communication. Applications range from military surveillance to collaborative office environments. Despite the attractiveness of exploiting networks of low-power and low-cost sensors, how to do so is a difficult problem. In this paper, we adopt a statistical viewpoint of such networks, and identify three key challenges. The first is to develop principled methods for low-level fusion of sensors measuring different modalities. We discuss an information-theoretic approach to sensor fusion, and present experimental results using audio and video data. The core component of this method is the learning of a nonparametric joint statistical model for the sensing modes. Secondly, we discuss how one might apply such a sensor fusion algorithm to acquire the relative geometry of a network of sensors using passively-sensed data. Specifically, we show how the fusion method previously developed can be used to find correspondences between pairs of long-baseline sensors. Finding such correspondences is, in general, the starting point for recovering the geometry. Finally, we discuss two iterative algorithms for performing inference on graphical models with cycles. Such models provide a flexible framework for constructing globally consistent statistical models from a set of local interactions. Importantly, the algorithms that we present allow information to be transmitted and processed in a distributed manner.

1 Introduction

The idea of deploying large numbers of networked, multimodal sensors to monitor, analyze, and adapt to the characteristics of an unknown environment has broad appeal in a rich variety of contexts. These applications include military surveillance, collaborative environments for virtual meetings, and "intelligent rooms" where embedded sensors provide an untethered interface to communications and computational systems. As is often the case, the raw data provided by increasingly complex sensory systems greatly exceeds our current understanding of how to effectively transform that data into useful information.

As a result, there are a number of basic challenges that must be met if the promise of distributed sensing is to be realized. Fundamental to many of these problems is the fact that in isolation, each of the available sensor data streams is of limited value. For example, in concepts being developed for military surveillance, large numbers of very inexpensive sensors of differing modalities – acoustic, seismic, infrared, optical, magnetic, pressure, temperature, etc. – might be scattered throughout a region of interest. Each individual sensor is extremely myopic, providing limited information about the environment and objects in its immediate vicinity. As a result, extracting useful information of the form needed by decision-making and planning systems requires the fusion of multiple signals into a coherent environment model. Similarly, in an intelligent room scenario in which multiple persons simultaneously interact with different data sources, automatic audio/video environmental responses must be constructed separately for each individual. This requires a system that can disentangle superimposed audio signals, and properly attribute them to individuals present in the room. In either case, data from each individual sensor is of limited value; indeed, the challenge is to determine relations among the signals from different sensors, and then exploit such relations to perform sensor fusion.

Another important challenge, present in many distributed sensing contexts, arises from constraints on computation and communication. For example, in military contexts, most of the sensing nodes may be extremely simple (i.e., they are "throw-away" sensors that are deployed only for a single mission). Such sensors may only be able to communicate through local wireless connections, perhaps augmented with connections to more powerful platforms. Similarly, embedded sensors in the home may not be coordinated through centralized processing; even if there is such a central node, much of the communication and computation is likely to take place in a distributed manner. Such contexts require distributed algorithms for passing information so as to maintain consistent statistical information throughout the network. Indeed, in a centralized system in which all sensed signals are available at a common point (e.g., as in a smart room), the vast amounts of data necessitate methods for fusion with computational complexity that scales well

The International Journal of High Performance Computing Applications, Volume 16, No. 3, Fall 2002, pp. 337-353

with the dimensionality of both the data and the desired output information.

In this paper, we provide an overview of several projects aimed at addressing the challenges we have just described. In particular, we describe three research efforts:

Multi-modal Sensor Fusion. Our first line of research addresses the fusion of signals from differing modalities in the absence of any explicit prior information about relationships among the signals. More precisely, we do not assume knowledge of the mechanisms by which the underlying causes interact to give rise to the multi-modal measurements. The specific context in which we describe this work is that of audio-video fusion. The methods we describe are based on exploiting the notion of mutual information in order to identify linear projections of the signals that are related to a common cause, without explicitly modeling or identifying that cause.

Calibration of Multi-modal Sensors from Passively Sensed Data. A critical problem in many distributed sensor systems is caused by uncertainty in the locations or calibration of the various sensors. Such uncertainties can lead to substantial errors in the fusion of data collected by these sensors. Thus, the development of methods for performing such calibration is critical if these data are to be exploited correctly. We describe preliminary work on a particular version of this problem, namely that of identifying multi-modal correspondences for the purpose of ultimately recovering the relative geometry of a set of passive sensors using the methodology described in the first section. Specifically, we consider co-located audio-video sensors and how to fuse their complementary information in order to locate their source using passive sensing.

Information Propagation in Networks with Cycles. As indicated previously, a major challenge in distributed fusion is the development of algorithms for propagating information throughout the network in order to produce consistent statistical descriptions at each local region. Accordingly, we devote the third section to describing two new iterative algorithms for estimating unknown variables in a network based on a set of noisy or incomplete observations. In this setting, the network itself is a *graphical model:* that is, at each node lies a random variable, and the graphical structure of the network represents dependencies (and more importantly, conditional independencies) among these variables. The problem of estimation or inference in such graphical models is the focus of considerable research in a variety of fields, including artificial intelligence [22, 24], image processing [14, 15], and the decoding of error correcting codes [18, 21]. For acyclic graphs or trees, there exist well-known and very efficient algorithms for performing inference. However, for graphs with cycles of any substantial size, these same problems are intractable. Although there exist methods for obtaining approximate solutions (e.g., belief propagation [24]) in graphs with cycles, such algorithms are not guaranteed to converge. Moreover, even when they do converge, the accuracy of the resulting approximation can vary substantially.

In this paper, we describe some new iterative algorithms for graphs with cycles. These algorithms are characterized by a common computational engine – that of exploiting very efficient algorithms to perform a sequence of exact calculations on acyclic graphs embedded within the original network. As we discuss, the behavior of the resulting algorithms is at least as good as that of previous methods for relatively easy problems; in addition, these new methods perform much better for many problems in which previous methods fail.

The presentations we provide of these three topics are, of necessity, overview in nature, and we refer the reader to more complete descriptions of each of these [11, 12, 27–31].

2 Information Theoretic Audio-Video Fusion

In this section we describe an information theoretic approach for signal-level sensor fusion. An obstacle to signal-level fusion of disparate signal types is the lack of simple joint statistical models (e.g., jointly Gaussian). For modalities whose relationship is complicated, fusion is not a straightforward task. For example, the optimal predictor from one sensor to another might be nonlinear, or the joint statistics might be multi-modal. In this section, we describe and justify an information-theoretic approach applicable to such problems. Although we demonstrate our approach using audio and video sensors, the technique is not restricted to these modalities, and is quite general.

A critical question is whether, in the absence of an adequate parametric model for joint measurement statistics, can one integrate measurements in a principled way, incorporating all available knowledge about statistical uncertainties. A nonparametric statistical estimation framework provides one attractive solution to this problem. In such approaches, we appeal to the information-theoretic notions of mutual information (MI) and minimum conditional entropy, which are equivalent to the principles of maximum a posteriori (MAP) and maximum likelihood (ML) in the parametric framework. We suggest an approach for learning maximally informative joint subspaces for multimedia signal analysis. The technique is a natural application of the learning method described

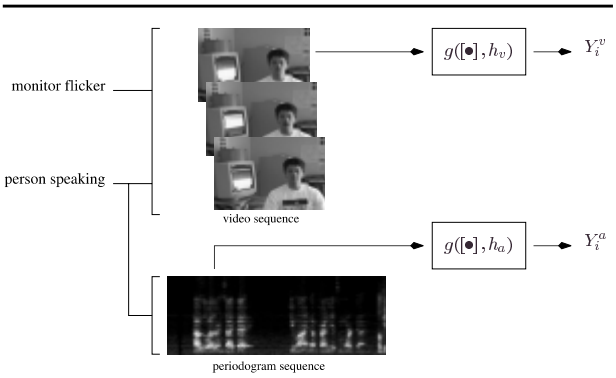


Fig. 1 Multi-modal fusion approach: Audio-video example in which the independent causes are the person speaking and the monitor flickering (due to asynchronous sampling). Note that the functions $g([\bullet], h_v)$ and $g([\bullet], h_a)$ may have different functional forms.

in [10, 12, 13], which is an entropy/MI optimization method for differentiable maps.

Fusion problems may be further complicated by the dimensionality of the signals. The video signals in the experiments we present are of high dimension (e.g. 360×240 pixels) while the audio signals are sampled at 24 Khz. The fusion approach makes use of a few seconds of data. Consequently, relative to the dimensionality of the signals, we have a small number of samples. High dimensionality is addressed by working in a subspace of the original measurements.

The approach is illustrated notionally in Figure 1. Given a video and audio signal collected at the same time, we treat the video frames as samples of a random variable and, likewise, the audio frames (windowed spectra computed every 1/30 seconds) as samples of another random variable. These variables correspond to $X^v \in \Re^{N_v}$ and $X^a \in \Re^{N_a}$ in the graphical model of Figure 2(a). These samples are passed through functions $Y_i^v = g(X_i^v, h_v)$ and $Y_i^a = g(X_i^a, h_a)$ where $Y_i^v \in \Re^{M_v}$ and $Y_i^a \in \Re^{M_a}$ have reduced dimensionality (i.e. $M_v << N_v$, $M_a << N_a$). For the experimental results presented here, the functions are linear projections where h_v and h_a are the coefficients of the linear projection. The dimensionality of the projections is such that each video and audio frame is reduced to a scalar. While our experiments map video and audio frames to a one-dimensional statistic (each) using a linear projection, the method itself is, in principle, extensible to any differentiable function and higher output dimension.

The goal of the approach is to choose the projection coefficients to optimize our fusion criterion. Specifically, the fusion criterion is the mutual information between the projections of the audio and video data defined in three equivalent ways as:

$$I(Y^a; Y^v) = h(Y^a) + h(Y^v) - h(Y^a, Y^v) \quad (1)$$

$$= h(Y^a) - h(Y^a | Y^v) \quad (2)$$

$$= h(Y^v) - h(Y^v | Y^a) \quad (3)$$

where $h(\,)$ is differential entropy [7] of the random variable Y with density $p_Y(y)$. Differential entropy is defined as

$$h(p_Y) = h(Y) = -\int_{\Omega_Y} p_Y(y) \log(p_Y(y)) dy \quad (4)$$

Entropy quantifies uncertainty in terms of the volume occupied by a random variable (as opposed to moments, which capture the spread of a density.[1])

Intuitively, one can think of this criterion as designing features which summarize the common information in X^a and X^v. The underlyis notion is that of a maximally informative subspace: the variable Y^a summarizes information about X^v that is contained in X^a and Y^v summarizes information about X^a that is contained in X^v. The challenge of using such a criterion is that mutual information is an integral functional of a density. Furthermore, we can only infer that density from samples. Consequently one needs an approximation to entropy (and by extension mutual information), an (implicit) estimate of the density, and an efficient means to compute the gradient with respect to the mapping coefficients (h_a and h_v). We refer the reader to [10, 13] where the approach is described in detail.

A brief description of the algorithm is as follows. We estimate the density in the low-dimensional *output space* using the Parzen density estimator [23], defined as

$$\hat{p}_Y(y) = \frac{1}{N\sigma} \sum_j k\left(\frac{y - y_j}{\sigma}\right) \quad (5)$$

where y can be either y^a, y^v when estimating their marginal densities, $k(y)$ is a kernel and must be a valid pdf (in our case a unit-variance Gaussian), $\{y_j\}$ are samples of the random variable, and N is the number of samples. Joint densities are similarly estimated using measurement pairs:

$$\hat{p}_{Y^a, Y^v}(y^a, y^v) = \frac{1}{N\sigma^2} \sum_j k\left(\frac{y^a - y_j^a}{\sigma}\right) k\left(\frac{y^v - y_j^v}{\sigma}\right) \quad (6)$$

The Parzen density estimate is chosen because it has the capacity to model densities with complex structure. Furthermore, it has desirable L_1 convergence properties [9].

Next we replace the integrand of (4) ($p \log p$) with a second-order Taylor series approximation (expanded about the uniform density) obtaining the following relationship between the approximation and the true entropy of the estimated density.

$$\hat{h}(\hat{p}) = h(\hat{p}) + D(\hat{p} \,||\, p_u) - \int_{\Omega_y} \frac{1}{2p_u(y)} (\hat{p}(y) - p_u(y))^2 \, dy \quad (7)$$

where $p_u(\,)$ is the uniform density over the support of the output space (constrained to lie in a unit hyper-cube) and $D(\hat{p} \,||\, p_u)$ is the Kullback-Leibler divergence between the densities $\hat{p}(\,)$ and $p_u(\,)$ [19].

This particular choice of entropy approximation and density estimate lead to a closed form gradient of MI with respect to the projection coefficients which can be computed by evaluating a *finite* number of functions at a *finite* number of points in the output space (see [13]). The update term for the *individual* entropy terms in (1) of sample y_i at iteration k as a function of y_i's at iteration $k-1$ is (note the opposite sign on the third term)

$$\Delta y_i^{(k)} = b_r(y_i^{(k-1)}) - \frac{1}{N} \sum_{j \neq i} \kappa_a(y_i^{(k-1)} - y_j^{(k-1)}, \Sigma) \quad (8)$$

$$b_r(y_i)_j \approx \frac{1}{d} \left(\kappa\left(y_i + \frac{d}{2}, \Sigma\right)_j - \kappa\left(y_i - \frac{d}{2}, \Sigma\right)_j \right) \quad (9)$$

$$\kappa_a(y, \Sigma) = \kappa(y, \Sigma) * \kappa'(y, \Sigma)$$
$$= -(2^{M+1} \pi^{M/2} \sigma^{M+2})^{-1} \exp\left(-\frac{y^T y}{4\sigma^2}\right) y \quad (10)$$

where y_i denotes a sample of either Y^a or Y^v, $M = M_a$, M_v or $M_a + M_v$ depending on which entropy term in equation (1) is being completed. Both $b_r(y_i)$ and $\kappa_a(y_i, \sigma)$ are vector-valued functions (M-dimensional) and d is the support of the output (i.e. a hyper-cube with volume d^M). The notation $b_r(y_i)_j$ indicates the jth element of $b_r(y_i)$. Adaptation consists of the update rule above followed by a modified least squares solution for h_v and h_a until a local maximum is reached. In the experiments that follow $M_v = M_a = 1$ with 150 to 300 iterations.

2.1 STATISTICAL JUSTIFICATION

While mutual information as a fusion criterion has intuitive appeal a natural question is when is it appropriate for fusion. One case arises in the context of the directed graph of Figure 2(a) which corresponds to the (statistically) independent cause model where the joint density of the variables (A, B, C, X^a, X^v) has the form:

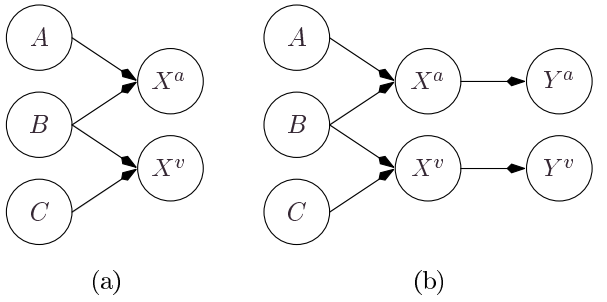


Fig. 2 (a) Graph of the independent cause model, and (b) the extension to the graph when the projections are considered.

$$p(A, B, C, X^a, X^v) = p(A) \times p(B)\, p(C)\, p(X^a \mid A, B)\, p(X^v \mid B, C)$$

where (A, B, C) are the independent causes of the observation variables (X^a, X^v). In Figure 1(a) the causes would be (as we shall see) the person speaking and the monitor flickering. The image sequence depends on both of those "causes" while the audio signal depends only on the person speaking. Figure 2(b) represents the extension to the graph when we add our projections.

Addressing the question of fusion criterion, consider the graphical models of Figure 3 which can be derived using graphical manipulations (or equivalently Bayes' rule) from the independent cause model. They show that information about X^a is conveyed through the *joint* statistics of the causes A and B. A similar statement can be made about X^v. As a result we cannot, in general, disambiguate the influences that A, B, and C have on the measurement X^a and X^v.

However, suppose decompositions of the measurements X^a and X^v *exist* such that the following joint densities can be written:

$$P(A, B, X^a) = P(A)P(B)P(X_A^a \mid A)P(X_B^a \mid B)$$

$$P(B, C, X^v) = P(B)P(C)P(X_B^v \mid B)P(X_C^v \mid C)$$

where $X^a = [X_A^a, X_B^a]$ and $X^v = [X_B^v, X_C^v]$. An example for our specific application would be segmenting the video image (or filtering the audio signal). Under this assumption, the model simplifies to the graph shown in Figure 4(a); from this simplified graph, we can extract the Markov chain which contains elements related only to B. Figure 4(b) shows equivalent graphs of the extracted Markov chain; panel 4(c) shows these same Markov chains with the projections and fusion variables. In these Markov chains, there is no longer any influence due to A or C.

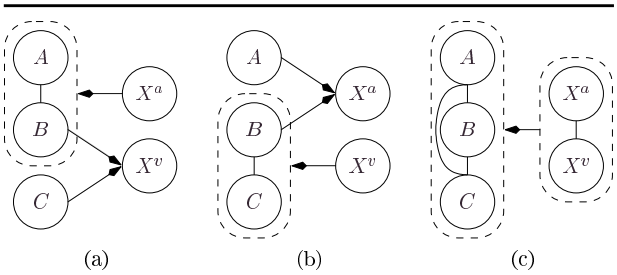


Fig. 3 Dependency graphs derived from the independent cause model (a)–(c) show the causal dependency induced by observing (a) X^a, (b) X^v, and (c) both X^a and X^v.

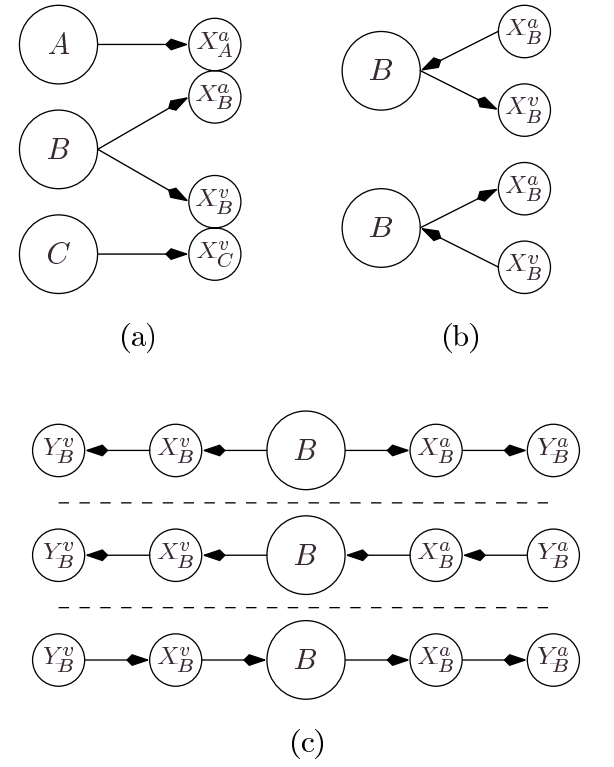


Fig. 4 (a) Modified directed graph when decomposition exists. (b) Extracted Markov chain and equivalent graphs. (c) Extracted Markov chain with projection variables.

However, this does not say how much X^a or X^v depend on one another, or their common cause *B*.

There exist more general decompositions that lead to this type of decoupling. Finding such a decomposition may be nontrivial. However, given the decomposition property, we can demonstrate why the fusion criterion is appropriate. Using the data processing inequality [7] applied to the Markov chains of Figure 4(c), the following inequalities hold:

$$I(Y_B^a; Y_B^v) \le I(X_B^a; Y_B^v) \le I(B; Y_B^v) \le I(X_B^v; Y_B^v) \quad (11)$$

$$I(Y_B^v; Y_B^a) \le I(X_B^v; Y_B^a) \le I(B; Y_B^a) \le I(X_B^a; Y_B^a) \quad (12)$$

and consequently

$$I(Y^a, Y^v) \le I(Y^a, B) \quad (13)$$

$$I(Y^a, Y^v) \le I(Y^v, B) \quad (14)$$

So, by maximizing the mutual information $I(Y^a, Y^v)$, we must necessarily increase the mutual information between Y^a and *B* and Y^v and *B*. The implication is that this method of fusion discovers the underlying cause of the observations, so that the joint density of $P(Y^a, Y^v)$ is strongly related to *B*, without explicitly modeling *B*. Furthermore, with an approximation, we can optimize this criterion without estimating the separating function directly. In fact, learning the separating functions is an implicit part of the adaptation [13]. In the event that a perfect decomposition does not exist, it can be shown that the method will approach a "good" solution in the Kullback-Leibler sense. In the collaborative signal processing domain, such fusion would allow multiple signals to be broken down into their constituent associated parts.

2.2 FUSION OF AUDIO-VIDEO DATA: EMPIRICAL RESULTS

We now present experimental results fusing audio/video data. In all cases the video signals in our experiments have dimension 360×240 pixels taken at 30 frames per second while the audio signals are sampled at 24 KHz. Audio signals are converted to a spectral representation by computing periodograms at the video rate using a window of length 2/30 seconds. Approximately three to four seconds of data are used in each case.

In each experiment video and audio frames are projected to a scalar. The projected variables are defined as

$$Y^a = h_a^T X^a \text{ audio projection} \quad (15)$$

$$Y^v = h_v^T X^v \text{ video projection} \quad (16)$$

where h_a and h_v are one-dimensional vectors of appropriate dimension. Processing is performed on samples of these random variables.

In this experiment we illustrate the utility of the approach for localization of the speaker in a video

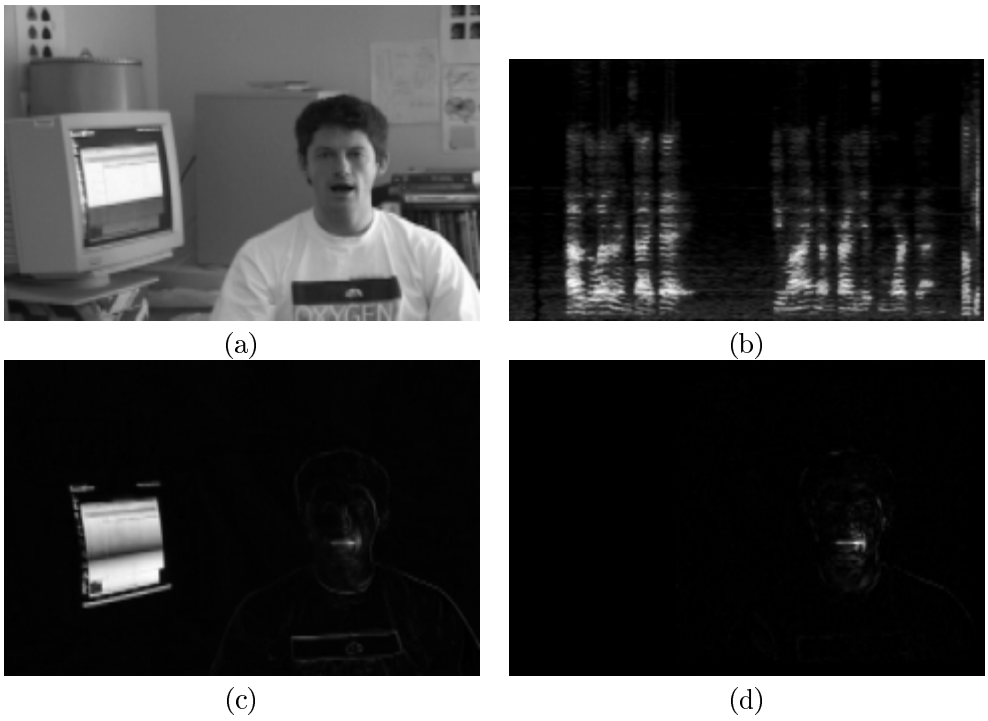


Fig. 5 Example of video localization using fusion approach: (a) one frame from 3 second sequence, (b) image of periodogram sequences (horizontal is time, vertical is frequency), (c) image of pixel standard deviations, and (d) image of learned video projection.

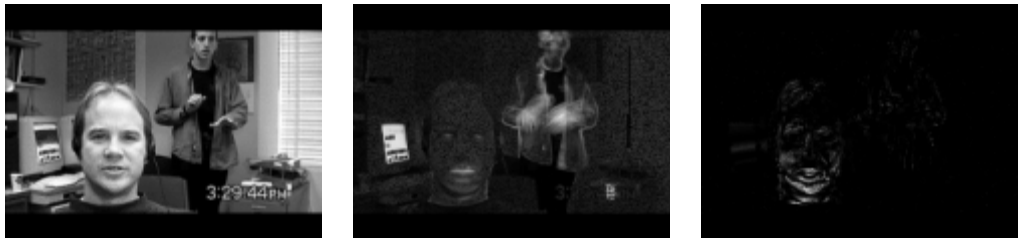
sequence using the audio signal. Figure 5(a) shows a single video frame from one sequence of data while 5(b) shows the sequence of periodograms computed from an audio signal recorded at the same time. In the figure there is a single speaker and a video monitor. Thoughout the sequence the video monitor exhibits significant flicker. Figure 5(c) shows an image of the pixel-wise standard deviations computed over the video sequence. As can be seen the energy associated with changes due to monitor flicker is significantly greater than that due to the speaker. However, one would not expect the changes in the monitor to be related statistically to the voice of the speaker. One would expect the changes due to the speakers lip motion to be related to the audio, but an exact model may not be available. The algorithm itself is agnostic with regard to the exact nature of the cause of the audio/video signal. It merely searches for projections which exhibit relatedness as measured by mutual information. This approach would be expected to work on other motion/sound pairs so long as the motion/sound pairs were related.

As we have described, we learn a projection of both the video and audio measurements such that the projections have high mutual information. Figure 5(d) shows the magnitude of the video projection coefficients after running the algorithm. As can be seen the projection coefficients have high magnitude in the region of the speakers lips and insignificant magnitude elsewhere, consistent with the independent cause model. Again, the utility lies in the fact the we do not place any express constraints on the form of the relationship of the projections, merely that they have high mutual information as estimated by our algorithm.

An example from a second video sequence is shown in Figure 6. In this sequence there is a single speaker (foreground), a monitor flickering (left) and an additional person in the background (moving their arms). This example is more difficult than the previous due to the presence of additional motion distractors which exhibit higher energy as measured by frame-to-frame pixel differences than changes due to the motion of the speaker's mouth.

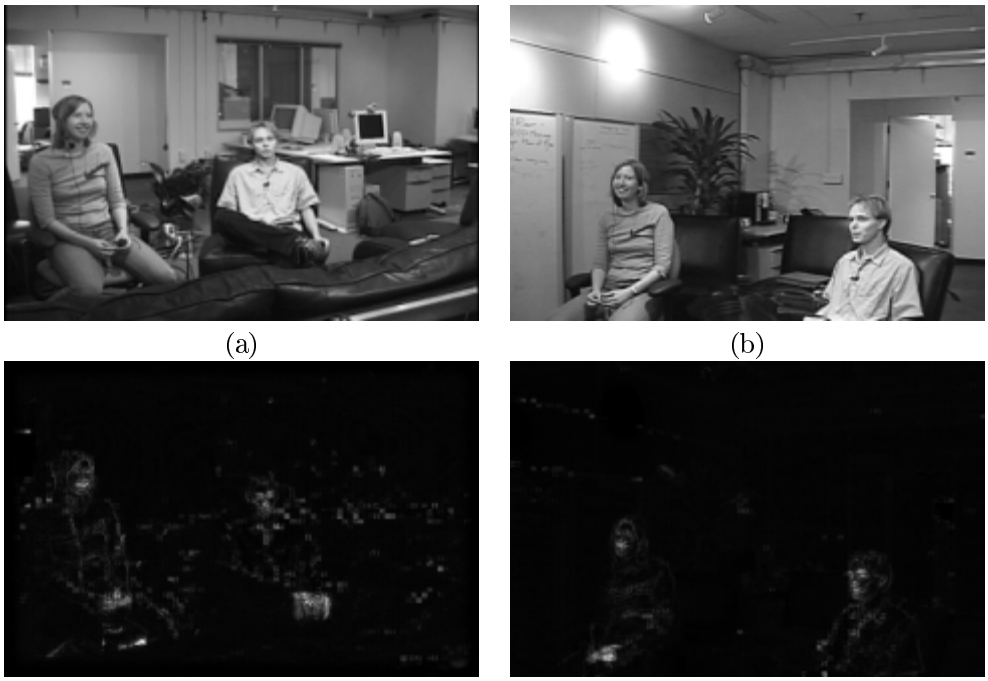
3 Array Geometry from Passively Sensed Data

Unknown sensor geometry can pose a significant challenge for distributed sensing. In the presence of sensor location uncertainties, data fusion becomes problematic. This problem is particularly acute when careful sensor placement is neither possible nor practical. If none of the sensors are active, calibration (when possible) relies



(a) (b) (c)

Fig. 6 Example of video localization using fusion approach: (a) one frame from 2.5 second sequence, (b) image of pixel standard deviations, and (c) image of learned video projection.



(a) (b) (c) (d)

Fig. 7 Audio/video data was collected from two viewpoints with two speakers in the scene. (a) and (b) show one frame from each of the sequences, while (c) and (d) are images of the sequence pixel standard deviations.

on passively sensed data. In particular, a basic requirement is the necessity to associate measurements taken from different locations which have a long baseline separation.

Using the basic method from the previous section, we discuss an approach which might be used for long-baseline stereo camera calibration. In this particular case we consider two cameras with co-located microphones. As before video and audio data are captured synchronously. Figures 7(a and b) show a frame from each of the cameras. Figures 7(c and d) show the pixel standard deviations for each of the sequences.

In typical stereo camera calibration methods, correspondences in each image are used to find a homography between the two viewpoints (i.e. a transformation that maps points in the scene taken from one viewpoint to the scene taken from the other viewpoint). There are various methods for finding correspondences, but most

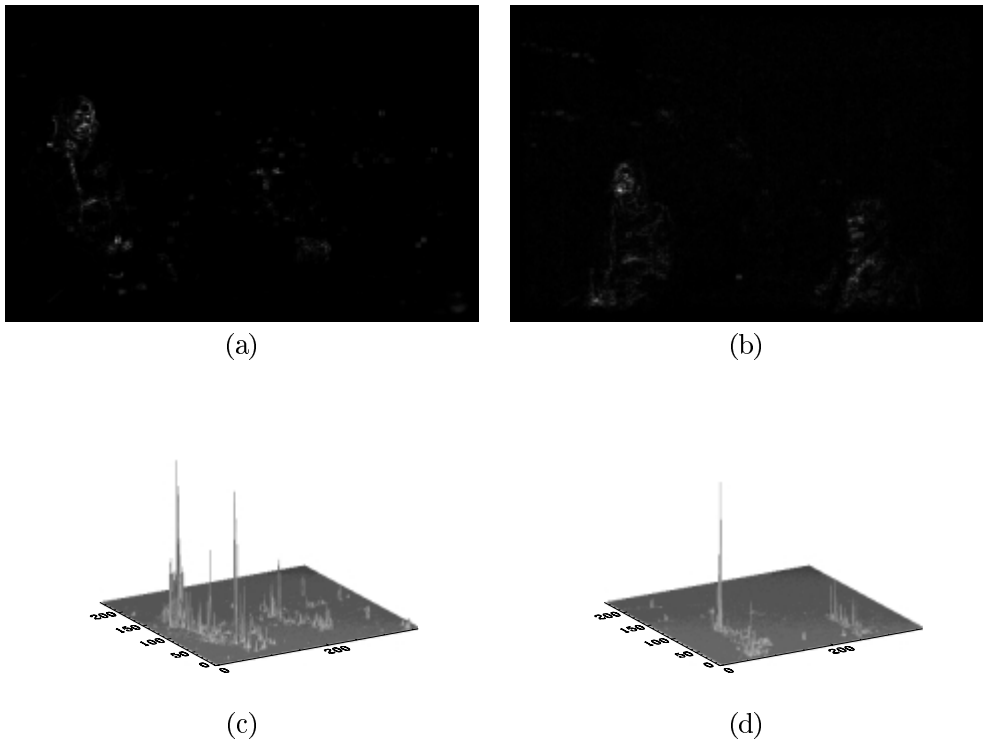


Fig. 8 Magnitude image of resulting projections for speaker on left for the camera on left (a) and right (b). A surface plot of the two projections are shown in (c) and (d). The largest peak in both plots corresponds to the mouth area of the subject on the left.

use a local correlation approach. That is, small image patches from one viewpoint are correlated with an image taken from the other viewpoint. This method works reasonably well for short baselines. However, as the baseline increases, correlation–based approaches degrade significantly. We propose an alternative method for finding correspondences based on the learning algorithm of the previous section. That is, the common variation in the image sequences associated to a common audio signal will provide correspondence locations between the viewpoints. In the data presented, the subjects in the scene speak at different times. Breaking the sequence into two segments (one for the person on the left, and one for the person on the right), we learn two projections (one for each camera) that relate their common audio signal as in the previous section. This is done for each segment of data. As we shall see, the projections (or the maximal point of the projection) provide fairly good correspondence to the individual speakers mouths. While this is a very specific example, it illustrates the notion of using complementary properties of multi-modal sensors.

As can be seen in Figures 8 and 9, for this set of data, the peak magnitude of the learned projection provides a good correspondence for the individuals' mouth locations. Of course, significantly more experimentation will be necessary, but these preliminary results show promise for the idea conceptually and are a good example of leveraging joint properties of sensing modes for signal processing.

We should note that in order to complete the calibration more than two correspondences are necessary. As a practical matter, stationary sensors can acquire additional correspondences over time as people (or objects) enter and leave the scene.

4 Inference techniques for graphs with cycles

We now bring our attention to another important and challenging problem: namely, given a network of nodes with constraints on computation and communication, how does one distribute information so as to perform statistical inference in an efficient manner? In this section, we describe two new approaches to the problem of estimation or inference on graphs with cycles. As discussed in the introduction, such inference problems arise

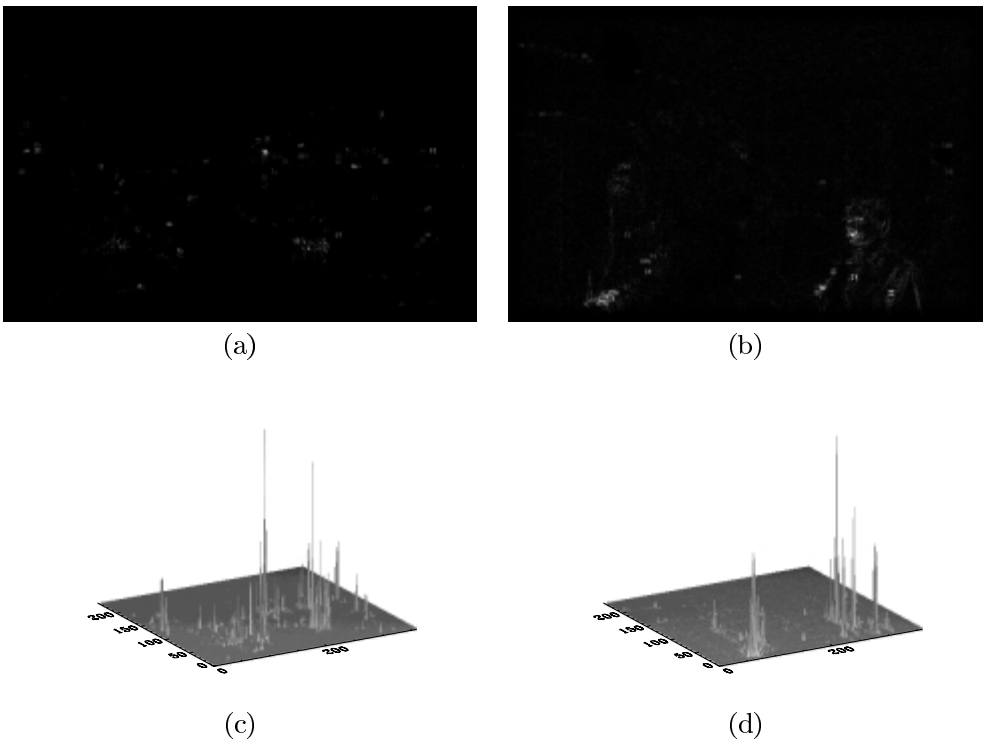


Fig. 9 Magnitude image of resulting projections for speaker on right for the camera on left (a) and right (b). A surface plot of the two projections are shown in (c) and (d). The largest peak in both plots corresponds to the mouth area of the subject on the right.

in a wide range of applications. In the context of sensor networks, the prototypical example is illustrated in Figure 10(a). Each sensor in the distributed network observes the activities occurring in a local area which overlaps the observation domains of its immediate neighbors. The objective is to fuse the information from all of the sensors to obtain a consistent, and preferably optimal, estimate of the observed environmental variables over the entire region. In some problems, these variables correspond to random fields representing quantities defined over the entire domain (e.g. pressure or temperature). In others, they may represent discrete objects whose number, characteristics, locations, and interrelationships we wish to estimate. In either case, the resulting problem can be cast abstractly in graphical terms as depicted in Figure 10(b).

Each node, or vertex, of the graph in Figure 10(b) represents a different random variable in the original distributed sensing problem. Some nodes represent the observations recorded at each sensor, while others correspond to the unobserved, or hidden, environmental variables we would like to estimate. The edges between nodes specify their statistical interrelationships. For example, signals recorded by acoustic sensors may be the

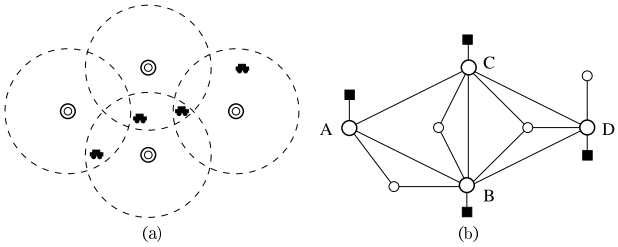


Fig. 10 (a) Notional sensor network in which field of view overlaps for "neighboring" sensors. (b) Corresponding graphical model, where circles represent unobserved (hidden) random variables and squares represent noisy observations.

superposition of the sounds generated by several different environmental sources. In this case, there are edges between the node representing the sensor and all of the nodes corresponding to hidden variables that directly influence the sensor. Alternatively, edges between hidden variables can capture known or suspected relationships among these variables. For example, pressure and

temperature distributions have spatial correlations that can be modeled with such edges. Similarly, edges could model hypotheses concerning the relationships between objects, allowing phenomena like coordinated navigation to be captured.

Given such a graphical model, our objective is to infer the behavior of variables throughout the network from imperfect, uncertain observations of a subset of the nodes. Typically, we also wish to perform this inference in a network–constrained manner, where all information is exchanged through a series of local message–passing operations between neighboring nodes. As indicated in the introduction, if the graph is acyclic, there exist efficient optimal inference algorithms which scale linearly with problem size (as measured by the number of nodes) and satisfy the network constraint. For graphs with cycles, however, exact inference is known to be NP hard [6]. Moreover, in virtually every context in which graphical inference arises, there are compelling reasons to consider graphs with cycles. In particular, note that removing a single node or link from an acyclic graph will cause the nodes to become disconnected. Therefore, to model sensor networks which are robust to the failure of individual components, we must consider graphs with cycles.

Is this section, we discuss a pair of novel statistical inference algorithms for graphs with cycles. Each algorithm generates a sequence of iterates by solving a series of modified problems on acyclic, or tree–structured, graphs embedded within the original graph. The first technique, called the *embedded trees* algorithm [31], is designed for exact inference on graphs in which the variables are jointly Gaussian. We also discuss a family of *tree-based reparameterization* algorithms for approximate inference, with particular emphasis on discrete–valued random variables. More detailed and extended descriptions of these methods and their properties can be found in [27–31].

4.1 GRAPHICAL MODELS

Graphical models derive their power from the fact that their graphical structure directly specifies the Markovian structure, or conditional independencies, of the underlying random variables. In particular, conditioned on the values of any set of nodes, disjoint subsets of the graph which are separated by those nodes are independent.[2] For example, in Figure 10(b), nodes A and D are conditionally independent given nodes B and C. The conditional independencies specified by a given graph in turn constrain the probability distribution of the variables in the graph. These constraints are made precise by the Hammersley–Clifford Theorem [3], which asserts that any valid distribution on a graphical model can be compactly encoded using the structure of the graph itself.

The Hammersley–Clifford Theorem is stated in terms of cliques, which are sets of nodes in which every node is *directly* connected to every other node in the clique. For example, in Figure 10(b) any pair of nodes connected by an edge forms a clique, and certain node triples such as $\{A, B, C\}$ and $\{B, C, D\}$ form cliques. However, $\{A, B, C, \mathrm{D}\}$ is *not* a clique because there is no edge between nodes A and D. Let $\mathcal{C}$ be the set of all cliques in the graph. Then, a positive distribution $p(\mathbf{x})$ defined on the set of hidden nodes $\mathbf{x}$ satisfies the conditional independencies implied by the graph if and only if it can be written in the factorized form

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(x_C) \tag{17}$$

where x_C is the set of random variables in clique C, Z is a normalization constant, and $\psi_C(x_C)$ is an arbitrary function taking positive values, called a compatibility function or clique potential.

From the Hammersley–Clifford Theorem, we know that the distribution $p(\mathbf{x})$ of the hidden variables must factorize into local potential terms as in equation (17). Let $\mathbf{y}$ be the set of observations made by all of the sensors. Under the typical assumption that each observation y_s is of a single hidden node x_s, the conditional distribution $p(\mathbf{x}|\mathbf{y})$ will retain the same functional form as (17), and hence the same graphical structure. This structure can be exploited by inference algorithms, which must compute functions of $p(\mathbf{x}|\mathbf{y})$. Some inference problems, such as computing the MAP estimate $\hat{\mathbf{x}}_{MAP}$ maximizing $p(\hat{\mathbf{x}}|\mathbf{y})$, involve global optimization over the entire graph. Many others, however, reduce to a set of local estimates for which the core challenge is computing the marginal distributions $p(x_s|\mathbf{y})$ for some or all of the nodes s in the graph.

In principle, one could calculate these single-node marginals by integrating or summing over all possible configurations of the other hidden variables. However, such a brute force approach is, in general, intractable. For example, given a set of n nodes taking discrete values from an alphabet of size m, the direct approach would require $\mathcal{O}(m^n)$ operations. Alternatively, if the hidden nodes and observations were jointly Gaussian, the exact conditional distribution could be found by solving a system of linear equations. For n hidden nodes, each representing a vector Gaussian random variable of dimension d, this would require $\mathcal{O}(n^3 d^3)$ operations. In either case, direct costs are intractable for values of n arising in many practical applications.

For acyclic or tree-structured graphs, there exist extremely efficient techniques for inference. A key property of trees is that nodes in a tree can be partially ordered in terms of their distance from a root node. Exploiting this

partial ordering leads to direct recursive inference algorithms which generalize dynamic programming–based [2] two–pass smoothing algorithms [16] for inference on Markov chains. In the discrete case, such algorithms require only $\mathcal{O}(nm^2)$ to compute single node marginals. In the Gaussian case, generalizations of the Kalman filter [5] require $\mathcal{O}(nd^3)$ operations.

For graphs with cycles, exact algorithms that scale linearly with n are generally not available. As a result, there is considerable interest and activity [4, 17] in developing approximate inference methods that work directly on the original graphical structure and are thus well–matched to network–constrained estimation problems. One of the best known, and most widely studied, approaches is the message–passing algorithm known as belief propagation [24]. In its standard form, belief propagation (BP) consists of an iterative sequence of local updates in which each node performs local computations and sends the results to its immediate neighbors. For tree–structured graphs, BP is similar to the two–pass tree inference algorithms mentioned above, and it converges to the exact optimal solution in a finite number of steps.

Much of the contemporary interest in BP, however, lies in its application to graphs with cycles, where it has been found to perform well for certain subclasses of graphs with cycles. In particular, it has received a great deal of attention in the coding theory literature [18, 21], where it provides the decoding procedure for the capacity-approaching turbo codes and low–density parity check codes.[3] For other graphs, however, BP may yield very poor approximations, or even fail to converge at all. In the Gaussian case, it can be shown that if BP converges, it always computes the correct conditional means, but incorrect error covariances [26, 33]. In the discrete case, recent work [1, 25, 32, 34] has yielded some insight into the dynamics and convergence properties of BP. Nonetheless, there remains much to be understood about the behavior of this algorithm, and more generally about other (perhaps superior) approximation algorithms.

In the following sections, we briefly describe two new network–constrained algorithms that improve on some of the limitations of belief propagation. The updates of belief propagation are purely *local*, in that at each iteration each node exchanges information only with its neighbors. In contrast, both of the techniques that we describe are based on the idea of isolating a spanning tree embedded within the original graph with cycles, and then performing exact calculations on this substructure. Since the tree is chosen to span the graph, the associated updates are *global*, in the sense that information from each node propagates throughout the graph within a single iteration. Despite the global nature of these updates, the computational cost is equivalent to or cheaper than BP, since we can make use of efficient tree algorithms.

4.2 TREE-BASED INFERENCE FOR GAUSSIAN PROCESSES

In this section, we examine inference techniques for graphs where the variables are jointly Gaussian. Let $\mathbf{x} = [x_1^T \; x_2^T \ldots x_n^T]^T$ be a vector containing the hidden variables, where $\mathbf{x} \sim \mathcal{N}(0, P)$. In the Gaussian case, the constraints on the clique potentials implied by the Hammersley–Clifford Theorem translate into a sparse structure for the inverse covariance matrix P^{-1}. If it is partitioned into blocks according to the hidden variable dimensions, the $(s,t)^{th}$ block can be nonzero only if there is an edge between nodes s and t.

Let $\mathbf{y} = C\mathbf{x} + \mathbf{v}$, $\mathbf{v} \sim \mathcal{N}(0, R)$, be a vector of noisy observations consisting of independent measurements y_s of individual nodes x_s. This implies that both C and R are block diagonal. We are interested in $p(x_s \mid \mathbf{y}) \sim \mathcal{N}(\hat{\mathbf{x}}_s, \hat{P}_s)$, the conditional distributions of the hidden variables at each node given *all* of the observations. Standard formulas exist for the computation of $p(\mathbf{x}|\mathbf{y}) \sim \mathcal{N}(\hat{\mathbf{x}}, \hat{P})$:

$$[P^{-1} + C^T R^{-1} C]\hat{\mathbf{x}} = C^T R^{-1} \mathbf{y} \tag{18}$$

$$\hat{P} = [P^{-1} + C^T R^{-1} C]^{-1} \tag{19}$$

The conditional means $\hat{\mathbf{x}}_s$ are simply subvectors of $\hat{\mathbf{x}}$, while the error covariances $\hat{P}_s$ are the block diagonal elements of $\hat{P}$.

4.2.1 Calculation of Conditional Means using Embedded Trees. For a Gaussian process on a graph, the operation of removing edges corresponds to modifying the inverse covariance matrix. Specifically, we apply a matrix splitting

$$P^{-1} + C^T R^{-1} C = P^{-1}_{\text{tree}(t)} - K_t + C^T R^{-1} C$$

where K_t is a symmetric cutting matrix chosen to ensure that $P^{-1}_{\text{tree}(t)}$ corresponds to a valid tree-structured inverse covariance matrix. This matrix splitting allows us to consider defining a sequence of iterates $\{\hat{\mathbf{x}}^n\}$ by the recursion:

$$\left[P^{-1}_{\text{tree}(t(n))} + C^T R^{-1} C\right]\hat{\mathbf{x}}^n = K_{t(n)}\hat{\mathbf{x}}^{n-1} + C^T R^{-1} \mathbf{y}$$

Here $t(n)$ indexes the embedded tree used in the n^{th} iteration. For general graphs, there are a huge number of potential cutting matrices $K_{t(n)}$. For example, Figure 11 shows two of the many spanning trees embedded in a nearest–neighbor grid.

When the matrix $(P^{-1}_{\text{tree}(t(n))} + C^T R^{-1} C)$ is invertible, it is possible to solve for the next iterate $\hat{\mathbf{x}}^n$ in terms of data $\mathbf{y}$ and the previous iterate $\hat{\mathbf{x}}^{n-1}$. Thus, given some starting point $\hat{\mathbf{x}}^0$, we can generate a sequence of iterates $\{\hat{\mathbf{x}}^n\}$ by the recursion

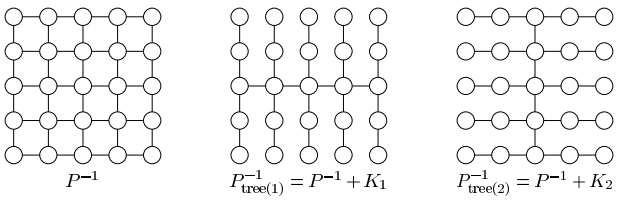


Fig. 11 Embedded trees produced by two different cutting matrices K_i for a nearest-neighbor grid (observation nodes not shown).

$$\hat{\mathbf{x}}^n = J^{-1}_{t(n)}[K_{t(n)}\hat{\mathbf{x}}^{n-1} + C^T R^{-1}\mathbf{y}] \qquad (20)$$

where $J_{t(n)} \triangleq (P^{-1}_{\text{tree}(t(n))} + C^T R^{-1} C)$. By comparing equation (20) to equation (18), it can be seen that computing the n^{th} iterate corresponds to a linear-Gaussian problem, which can be solved efficiently and directly with standard tree algorithms [5].

Due to the linearity of the ET iterations, their convergence is easily analyzed. Assuming the $J_{t(n)}$ matrices are invertible, algebraic manipulation of equation (20) shows that for any starting point, $\hat{x}$ is the unique fixed point of the recursion. The error $e^n \triangleq \hat{\mathbf{x}}^n - \hat{\mathbf{x}}$ at the n^{th} iteration obeys the dynamics

$$e^n = \left[\prod_{j=1}^{n} J^{-1}_{t(j)} K_{t(j)}\right] e^0 \qquad (21)$$

One natural implementation of the ET algorithm cycles through the embedded trees in some fixed order, say $t = 1, \ldots, T$. In this case, the convergence of the algorithm can be analyzed in terms of the spectral radius of the product matrix $\mathbf{E} \triangleq \Pi_{j=1}^{T} J_j^{-1} K_j$. In particular, if $\rho(\mathbf{E}) > 1$ then the algorithm will not converge, whereas if $\rho(\mathbf{E}) < 1$, then $(\hat{x}^n - \hat{x}) \xrightarrow{n\to\infty} 0$ geometrically at rate $\gamma \triangleq \rho(\mathbf{E})^{1/T}$.

Although $\rho(\mathbf{E})$ completely defines the convergence behavior, for large problems we cannot explicitly compute this quantity. The challenge is then to find guidelines for choosing cutting matrices K which produce rapidly convergent iterations. Empirically, we find that cuts which remove weak edges and modify the diagonal entries of $(P^{-1} + C^T R^{-1} C)$ as little as possible generally converge fastest. In addition, *much* faster convergence rates are typically found by cycling through multiple embedded trees. Intuitively, this happens because using multiple trees allows the immediate reinstatement of constraints that were neglected on previous iterations. For theoretical analyses and substantial experimentation supporting these observations, see [27, 31].

4.2.2 Calculation of error covariances using embedded trees. Although there exist a variety of iterative algorithms, such as belief propagation, for computing the conditional mean of a linear-Gaussian problem, none of these methods correctly compute error covariances at each node. We show here that the ET algorithm can efficiently compute these covariances in an iterative fashion. For many distributed sensing applications, these error statistics are as important as the estimates.

We assume for simplicity in notation that $\hat{\mathbf{x}}^0 = 0$ and then expand equation (20) to yield that for any iteration $\hat{\mathbf{x}}^n = [F_n + J^{-1}_{t(n)}] C^T R^{-1} \mathbf{y}$, where the matrix F_n satisfies the recursion

$$F_n = J^{-1}_{t(n)} K_{t(n)} [F_{n-1} + J^{-1}_{t(n-1)}] \qquad (22)$$

with the initial condition $F_1 = \mathbf{0}$. It is straightforward to show that whenever the recursion for the conditional means in equation (20) converges, then the matrix sequence $\{F_n + J^{-1}_{t(n)}\}$ converges to the full error covariance $\hat{P}$.

Moreover, the cutting matrices K are typically of low rank, say $\mathcal{O}(Ed)$ where E is the number of cut edges and d is the dimension of the hidden variables. On this basis, it can be shown that each F_n can be decomposed as a sum of $\mathcal{O}(Ed)$ rank 1 matrices. Directly updating this low–rank decomposition of F_n from that of F_{n-1} requires $\mathcal{O}(nE^2 d^5)$ operations. However, an efficient restructuring of this update requires only $\mathcal{O}(nEd^4)$ operations [27]. The diagonal blocks of the low–rank representation may be easily extracted and added to the diagonal blocks of $J^{-1}_{t(n)}$, which are computed by standard tree smoothers. All together, we may obtain these error variances in $\mathcal{O}(nEd^4)$ operations per iteration. Thus, the computation of error variances will be particularly efficient for graphs where the number of edges E that must be cut is small compared to the total number of nodes n.

4.2.3 Comparison to other techniques. Consider again the estimation formulas given in equation (18). In the Gaussian case, computing the conditional mean $\hat{\mathbf{x}}$ is equivalent to taking the product of the inverse of a sparse matrix with a vector. A variety of extremely efficient techniques for this problem are available in the numerical linear algebra literature [8]. Of these, the conjugate gradients (CG) method is one of the most effective, so it will be used to provide a comparison point for the performance of the embedded trees algorithm. Note, however, that like BP, CG does not provide the correct error covariances. In addition, CG iterations do *not* decompose into the local structure needed for network–constrained estimation.

We have applied the ET algorithm to a variety of graphs, ranging from single cycle graphs to nearest-neighbor grids. Figure 12(a) compares the rates of convergence for embedded trees (ET), belief propagation

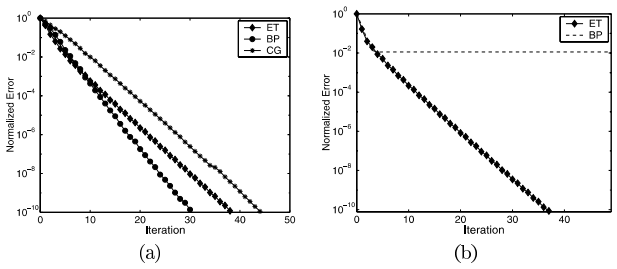


Fig. 12 (a) Convergence rates for computing conditional means (normalized L^2 error). (b) Convergence rate of ET algorithm for computing error variances, compared to the approximate error variances calculated by the BP algorithm.

(BP), and conjugate gradients (CG) on a nearest-neighbor grid. ET and BP have per iteration costs of $\mathcal{O}(nd^3)$, while CG is $\mathcal{O}(nd^2)$. The ET algorithm employed two embedded trees analogous to those shown in Figure 11. In accordance with theoretical results, the ET algorithm converges geometrically. For graphs with inhomogeneous potentials, all three algorithms typically have similar convergence rates. However, for these tests we did not attempt to optimize the trees used by ET. Figure 12(b) shows that in contrast to CG and BP, the ET algorithm can also be used to compute the error variances, where the convergence rate is again geometric. Note that the approximate error variances calculated by the BP iteration are much less accurate than the asymptotically exact variances produced by the ET algorithm.

4.3 TREE-BASED REPARAMETERIZATION FOR DISCRETE PROCESSES

In other work [28–30], we have shown that the same idea of performing exact computations over trees embedded within a graph with cycles can be applied fruitfully to discrete processes as well. Our work provides a new conceptual view of various algorithms for approximate estimation, including belief propagation (BP). The basic idea is to seek a *reparameterization* of the distribution that yields factors which correspond, either exactly or approximately, to the desired marginal distributions. If the graph is tree-structured (i.e., acyclic), then there exists a unique reparameterization specified by exact marginal distributions over cliques. For a graph with cycles, we consider the idea of iteratively reparameterizing different parts of the distribution, each corresponding to an acyclic subgraph. We show that the usual parallel message-passing BP can be interpreted in exactly this manner, in which each reparameterization takes place over a pair of neighboring nodes. More generally, we consider updates in which reparameterization is performed over arbitrary spanning trees, which we refer to as *tree-based reparameterization* (TRP) algorithms. At one low level, these more global updates can be viewed as a tree-based schedule for message-passing.

Viewing approximate estimation as reparameterization leads to a number of new conceptual insights. First of all, we derive an intuitive characterization of BP fixed points: they must be consistent, in a suitable sense to be defined, over *all* trees embedded within the original graph with cycles. Secondly, we establish that all iterates of TRP/BP algorithms, as well as their fixed points, obey an intrinsic invariance: namely, although the updates alter the local compatibility functions, the distribution on the graph with cycles is left unchanged. These two results enable us to make a contribution to an important open problem – viz., characterizing the approximation error that arises in applying BP to a graph with cycles. Some results have been obtained in certain special cases: Weiss [32] for the single cycle, and Richardson [25] for turbo codes. We derive an exact expression for the difference between the BP approximations and the actual marginals on an arbitrary graph with cycles. Moreover, we derive computable bounds on this error, which help to illustrate factors controlling approximation accuracy. More details of the work described here can be found in the papers [28–30].

4.3.1 Inference in trees as reparameterization. To understand the notion of tree-based reparameterization, recall that as shown in equation (17), any probability distribution defined by a graphical model decomposes as a product of functions, each involving only maximal cliques of the graph. In general, determining the marginal distributions of subsets of variables from such representations is a daunting task [6]. However, such factorized representations are far from unique. This suggests the possibility of finding factorizations of the probability distribution in which individual factors correspond, either exactly or approximately, to the desired marginal distributions.

The lack of uniqueness in the parameterization of $p(\mathbf{x})$ is illustrated in Figure 13. Shown in (a) is the original parameterization in terms of compatibility functions ψ_{st} and observation functions ψ_s. For such a tree-structured graph, it is well-known that the distribution $p(\mathbf{x})$ can be reparameterized in terms of the exact joint marginals $p_{st}(x_s, x_t)$ and single node marginals $p_s(x_s)$, as illustrated in Figure 13(b). This result generalizes the representation of a discrete-time Markov chain as the product of an initial distribution and successive one-step transitions. (Consider, for instance, the simple Markov chain formed by nodes 1 and 2.) Alternatively, it can be considered a special case of the factorization of distributions specified by the junction tree representation [20].

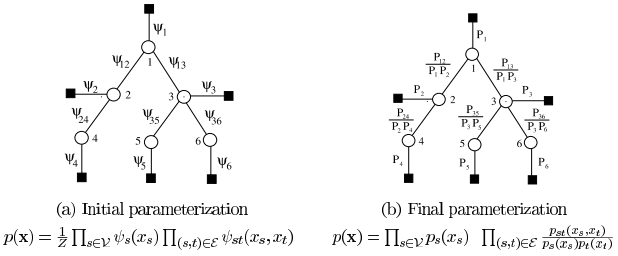


Fig. 13 Non-uniqueness in the parameterization of a distribution $p(\mathbf{x})$ on a graphical model. (a) The original parameterization is given in terms of compatibility functions $\{\psi_s, \psi_{st}\}$. (b) The desired parameterization is specified in terms of the exact marginal distributions $p_{st}(x_s, x_t)$ and $p_s(x_s)$ on the graph with cycles.

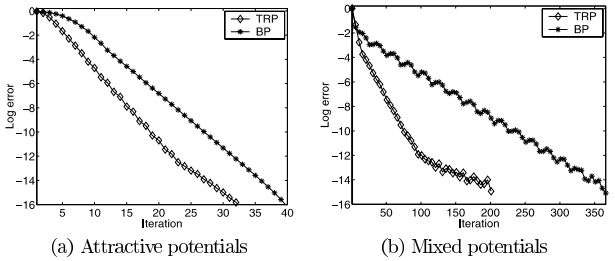


Fig. 14 Convergence plots of log error versus iteration number n for the 7×7 grid under two conditions.

4.3.2 Tree-based reparameterization for graphs with cycles. Thus, exact graphical inference algorithms for trees can be viewed as reparameterizing the distribution $p(\mathbf{x})$ in terms of the exact marginals. From this viewpoint arises the idea of iterative algorithms for graphs with cycles in which, at each iteration, a partial reparameterization is performed over a collection of factors corresponding to a tree embedded within the original graph. In particular, the first step of any iteration is to decompose the original distribution as a product $p(\mathbf{x}) = p^i(\mathbf{x}) r^i(\mathbf{x})$, where p^i denotes a distribution over the embedded tree, and r^i denotes a set of residual terms. The second step is to perform reparameterization on the embedded tree, leaving fixed the potentials on edges not in the tree. Finally, we can choose some other embedded tree, and repeat the procedure.

This sequence of operations can be formulated precisely as updating a vector $\mathbf{T} = \{T_s, T_{st}\}$ of *pseudomarginals* on each node and edge of the graph. Ultimately, we seek pseudomarginals that are locally consistent (i.e., for which the local marginal T_{st} on edge (s, t) agrees with the single node marginals T_s and T_t.) A key property is that these updates can be viewed as reparameterizations, since (as with exact estimation on a tree), they simply express the full distribution $p(\mathbf{x})$ in an alternative form (but do not alter it). This invariance has a number of important theoretical consequences.

As we show in [29], belief propagation can be reformulated as a procedure of exactly this type, where each reparameterization takes place over the extremely simple subgraph formed by a pair of neighboring nodes. More generally, the reparameterization perspective leads to a new class of algorithms, which we refer to as *tree-reparameterization* (TRP) algorithms. At each iteration, an entire spanning tree of the original graph is reparameterized simultaneously, thereby propagating information globally across all nodes of the graph. We prove that the fixed points of the reparameterization algorithms described in [29] coincide with those of BP.

Nonetheless, we find that global updates lead to some important practical advantages. In particular, one might expect such an algorithm to have better convergence properties than the purely local two-node updates of BP. Indeed, experimentation with TRP supports this conclusion: when applied to problems for which BP converges, TRP converges at least as quickly, and for many problems often much more quickly. Figure 14 gives a sample empirical comparison of BP versus a TRP algorithm using two spanning trees, as applied to a binary process defined on a 7×7 grid. The condition shown in (a) corresponds to attractive potentials that encourage equality between neighboring random variables; the mixed condition consists of a mixture of attractive and repulsive potentials. In both cases, the TRP algorithm using two spanning trees (those shown in Figure 11) converges more quickly than BP, with lower computational cost and storage. The advantage is especially marked in the harder problem case shown in (b). Moreover, in addition to faster convergence documented here, the global updates of TRP using spanning trees has another and perhaps more important advantage – namely, we find that it converges in many cases where BP does not.

4.3.3 Conceptual insights into approximate inference. The reparameterization perspective also provides new theoretical insights. First of all, it leads to a new characterization of fixed points of algorithms like TRP/BP. Recall that each iteration entails reparameterizating the full distribution $p(\mathbf{x})$ in terms of pseduomarginals T_{st} and T_s obtained from calculations over an embedded tree. Suppose that this sequence converges to a fixed set of functions T_{ts}^* and T_s^*, which leads to the situation illustrated in Figure 15(a). We prove that this fixed point T^* must be consistent on each embedded tree contained within the original graph with cycles. In particular, if we remove edges (4, 5) and (5, 6) from the graph shown in Figure 15(a), then we obtain the spanning tree shown

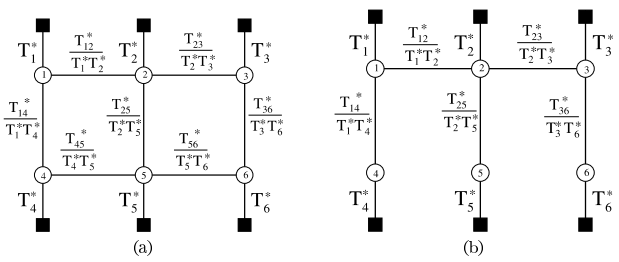


Fig. 15 Parameterization upon convergence (a) and tree-based consistency condition (b).

in (b). The functions T_s^* (for all $s \in \mathcal{V}$) and T_{st}^* must be the *exact* marginal distributions for this tree-structured distribution. More remarkably, this property must hold for any acyclic structure embedded within the original graph. So a similar condition would have to hold if (for instance) we removed edges (1, 2) and (2, 3) from the graph in (a) to obtain a different embedded tree.

Another fundamental property of the reparameterization updates is that they leave invariant the full distribution on the graph with cycles. That is, if the original parameterization was given as $p(\mathbf{x}) = \frac{1}{Z}\prod_s \psi_s(x_s)\prod_{(s,t)}\psi_{st}(x_s, x_t)$, then upon convergence, a TRP algorithm will simply have found an alternative parameterization of the form $p(\mathbf{x}) = \frac{1}{Z'}\prod_s T_s^*(x_s)\prod_{(s,t)} \frac{T_{st}^*}{T_s^*(x_s)T_t^*(x_t)}$. This invariance has a number of important consequences, including geometric insight into the reparameterization updates; consequences for the exactness of algorithms like TRP and BP; implications for BP in the Gaussian setting; and an exact expression and bounds on the approximation error. This last result is especially important, in that it leads to conceptual insight into the cases where TRP/BP are expected to perform well or poorly. We refer the reader to the papers [28–30] for full details.

5 Discussion

In this paper, we have described research in several important areas pertaining to networks: information extraction, sensor fusion, and information propagation. The research we have described is part of an ongoing effort aimed at establishing a foundation for an integrated theory and methodology for such networks. A common theme is the need for algorithms that retain the capacity for modeling complex relationships while minimizing computation.

We presented a general framework for fusing signals from disparate sensor modalities based on concepts from information theory. In addition, we described a practical algorithm which embodies these principles. While the experimental results focused on audio/video data, such an approach is likely to work on other modalities as well. Moreover, using the previously described approach to fusion, we developed a method for finding correspondences for sensors separated by long baselines. Experimental results were given for a two camera system. Although these results are preliminary, they highlight a basic challenge for randomly placed sensor arrays: namely, the need to either recover a relative sensor geometry from passively sensed data or, failing that, to develop algorithms that remain robust to sensor uncertainty. This problem is particularly challenging when the sensor array contains mixed mode sensors. An additional issue is the scalability of the algorithm. As the number of sensors increases it becomes difficult to perform joint fusion in an optimal manner, which motivates work on inference techniques for graphs with cycles.

We discussed two new approaches for performing inference on graphs with cycles. First, we presented the embedded trees algorithm for exact inference in graphs where the variables are Gaussian. Brute force approaches to this problem are intractable for sufficiently large graphs. Instead, we exploited the fact that any graph with cycles has a large number of trees embedded within it, and that tree-structured problems can be solved efficiently. We showed that an exact solution to the original problem on the graph with cycles can be obtained by solving an appropriately constructed sequence of tree problems. Unlike other methods (e.g., belief propagation), the embedded trees algorithm computes exactly both the means and error covariances. Our second approach showed that similar ideas can be applied for inference problems involving discrete processes. Here the general problem is NP hard [6], which motivates the analysis of approximate rather than exact methods. We presented the tree-based reparameterization framework, which provides a new conceptual view of a large class of algorithms for approximate inference including belief propagation. Among the important theoretical insights are a new characterization of fixed points, and analysis of the error in the approximation for an arbitrary graph with cycles.

There are a variety of important and interesting open problems which bridge the boundaries between the different research areas discussed above. Traditional approaches to heterogeneous sensor fusion (as in Section 2) assume that the relative sensor geometry is known so that each sensor's measurements can be correctly registered. However, as described in Section 3, this fusion procedure may in fact enhance our ability to estimate and account for relative geometry. For example, robust coherent processing of distributed acoustic sensors requires reducing or accommodating errors in sensor location. However, if we also have video sensors to which the acoustic sensors can be fused, we can use stereo processing of the

video data to improve our estimation of acoustic sensor location, which may in turn improve their coherent exploitation.

A second area for future work is that of performing multimodal fusion, as in Section 2, in a network–constrained manner. Distributed fusion procedures must account for the fact that each sensor has only a limited domain of observation, overlapping that of nearby sensors, and also has only limited communication connectivity. To develop efficient network fusion algorithms, we must extend network-constrained inference methods, such as those described in Section 4, to accommodate the types of inference problems arising in multimodal fusion. One particularly important challenge is the problem of distributed learning. Specifically, the method described in Section 2 involved learning how the observables from two sensors are related, while the methods in Section 4 assume a known probabilistic model for sensor interrelationships. Building network-constrained algorithms for learning such models adaptively is thus of great importance.

Finally, it is interesting to consider problems in which the sensor network is itself an embedded system within a larger sensing and global awareness system. For example, in military contexts, distributed ground sensors may represent only one component of a system that also includes conventional sensing systems such as airborne radars and imaging devices. In such cases, there are typically higher-level graphical models of the military situation, describing how the activities of different objects are related over space and time. Such graphical models are well matched to the framework described in Section 4. In addition, this more global perspective raises another important direction for research, namely that of sensor cueing or control. For example, particular sensors may have multiple modes of operation and be able to directly focus their attention on areas of specific interest. In this situation, this flexibility should be exploited by directing these sensors to perform those measurements which best reduce the uncertainty in estimates of particularly important environmental variables. Implementing this distributed control in a network–constrained manner represents, in a sense, a "dual" to network–constrained inference problems.

NOTES

1. Note that two random variables can have the same variance (second central moment), but very different entropies.

2. These relationships are valid for graphs with undirected edges. There is a related theory for graphical models with directed edges which leads to a different set of conditional independence relationships. For a more detailed introduction to graphical models, see [20, 24].

3. In the coding community, belief propagation is known as the sum-product algorithm.

BIOGRAPHIES

John W. Fisher III received a Ph.D. degree in electrical and computer engineering from the University of Florida, Gainesville, Florida in 1997. He is currently a research scientist in the Artificial Intelligence Laboratory and affiliated with the Laboratory for Information and Decision Systems at the Massachusetts Institute of Technology. Prior to joining the Massachusetts Institute of Technology he was affiliated with the University of Florida, as both a faculty member and graduate student since 1987, during which time he conducted research in the areas of ultra-wideband radar for ground penetration and foliage penetration applications, radar signal processing, and automatic target recognition algorithms. His current area of research focus includes information theoretic approaches to signal processing, multi-modal data fusion, machine learning and computer vision. He is a member of IEEE and SPIE.

Martin Wainwright is currently a postdoctoral associate in EECS and Statistics at the University of California, Berkeley. He received his Ph.D. in Electrical Engineering and Computer Science from Massachusetts Institute of Technology in January 2002. His research interests include graphical models, machine learning, network information theory, and combinatorial optimization.

Erik Sudderth is a Ph.D. student in the department of electrical engineering and computer science at the Massachusetts Institute of Technology, where he received the M.S. degree in 2002. He received the B.S. degree (summa cum laude) in electrical engineering from the University of California at San Diego in 1999. His research interests include statistical modeling and machine learning, and their application to such fields as computer vision, remote sensing, and error correcting codes.

Dr. Alan Willsky joined the M.I.T. faculty in 1973 and is currently the Edwin S. Webster Professor of Electrical Engineering. He is a founder and member of the board of directors of Alphatech, Inc. and a member of the US Air Force Scientific Advisory Board. He has received several awards including the 1975 American Automatic Control Council Donald P. Eckman Award, the 1979 ASCE Alfred Noble Prize and the 1980 IEEE Browder J. Thompson Memorial Award. Dr. Willsky has held visiting positions in England and France and various leadership positions in the IEEE Control Systems Society (which made him a Distinguished Member in 1988). He has delivered numerous keynote addresses and is co-author of the undergraduate text Signals and Systems. His research interests are in the development and application of advanced methods of estimation and

statistical signal and image processing. Methods he has developed have been successfully applied in a variety of applications including failure detection, surveillance systems, biomedical signal and image processing, and remote sensing.

REFERENCES

[1] Aji, S. M., Horn, G., and McEliece, R. On the convergence of iterative decoding on graphs with a single cycle. In *Proceedings IEEE Intl. Symp. on Information Theory*, page 276, Cambridge, MA, 1998.

[2] Bertsekas, D. *Dynamic programming and stochastic control*, volume 1. Athena Scientific, Belmont, MA, 1995.

[3] Besag, J. Spatial interaction and the statistical analysis of lattice systems. *J. Roy. Stat. Soc. Series B*, 36:192–236, 1974.

[4] Besag, J. and Green, P. J. Spatial statistics and Bayesian computation. *J. R. Stat. Soc. B*, 55(1):25–37, 1993.

[5] Chou, K., Willsky, A., and Nikoukhah, R. Multiscale systems, Kalman filters, and Riccati equations. *IEEE Trans. AC*, 39(3):479–492, March 1994.

[6] Cooper, G. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42:393–405, 1990.

[7] Cover, T. M. and Thomas, J. A. *Elements of Information Theory*. John Wiley & Sons, Inc., New York, 1991.

[8] Demmel, J. *Applied numerical linear algebra*. SIAM, Philadelphia, 1997.

[9] Devroye, L. *A Course in Density Estimation*, volume 14 of *Progress in Probability and Statistics*. Birkhauser, Boston, 1987.

[10] Fisher, J. and Principe, J. Unsupervised learning for nonlinear synthetic discriminant functions. In Casasent, D. and Chao, T., editors, *Proc. SPIE, Optical Pattern Recognition VII*, volume 2752, pages 2–13, 1996.

[11] Fisher III, J. W., Darrell, T., Freeman, W. T., and Viola, P. Learning joint statistical models for audio-visual fusion and segregation. In *Advances in Neural Information Processing Systems 13*, 2000.

[12] Fisher III, J. W., Ihler, A. T., and Viola, P. A. Learning informative statistics: A nonparametric approach. In Solla, S. A., Leen, T. K., and Müller, K.-R., editors, *Advances in Neural Information Processing Systems 12*, 1999.

[13] Fisher III, J. W. and Principe, J. C. A methodology for information theoretic feature extraction. In A. Stuberud, editor, *Proceedings of the IEEE International Joint Conference on Neural Networks*, 1998.

[14] Freeman, W. T., Pasztor, E. C., and Carmichael, O. T. Learning low-level vision. *Intl. J. Computer Vision*, 40(1):25–47, 2000.

[15] Frey, B., Koetter, R., and Petrovic, N. Very loopy belief propagation for unwrapping phase images. In *NIPS 14*. MIT Press, 2001.

[16] Jazwinski, A. H. *Stochastic processes and filtering theory*. Academic Press, New York, 1970.

[17] Jordan, M., Ghahramani, Z., Jaakkola, T. S., and Saul, L. An introduction to variational methods for graphical models. In *Learning in graphical models*, pages 105–161. MIT Press, 1999.

[18] Kschischang, F. and Frey, B. Iterative decoding of compound codes by probability propagation in graphical models. *IEEE Sel. Areas Comm.*, 16(2):219–230, February 1998.

[19] Kullback, S. *Information Theory and Statistics*. John Wiley and Sons, New York, 1959.

[20] Lauritzen, S. L. *Graphical models*. Oxford University Press, Oxford, 1996.

[21] McEliece, R., McKay, D., and Cheng, J. Turbo decoding as an instance of Pearl's belief propagation algorithm. *IEEE Jour. Sel. Communication*, 16(2):140–152, February 1998.

[22] Murphy, K., Weiss, Y., and Jordan, M. Loopy-belief propagation for approximate inference: An empirical study. In *Uncertainty in Artificial Intelligence*, volume 9, 1999.

[23] Parzen, E. On estimation of a probability density function and mode. *Ann. of Math Stats.*, 33:1065–1076, 1962.

[24] Pearl, J. *Probabilistic reasoning in intelligent systems*. Morgan Kaufman, San Mateo, 1988.

[25] Richardson, T. The geometry of turbo-decoding dynamics. *IEEE Trans. Info. Theory*, 46(1):9–23, January 2000.

[26] Rusmevichientong, P. and Van Roy, B. An analysis of belief propagation on the turbo decoding graph with Gaussian densities. *IEEE Trans. Info. Theory*, 47(2): 745–765, Feb. 2001.

[27] Sudderth, E. Embedded trees: Estimation of Gaussian processes on graphs with cycles. Master's thesis, Massachusetts Institute of Technology, February 2002.

[28] Wainwright, M. J. *Stochastic processes on graphs with cycles: geometric and variational approaches*. PhD thesis, MIT, Laboratory for Information and Decision Systems, January 2002.

[29] Wainwright, M. J., Jaakkola, T., and Willsky, A. S. Tree-based reparameterization for approximate estimation on graphs with cycles. LIDS Tech. report P-2510: available at http://ssg.mit.edu/group/mjwain/mjwain.shtml, May 2001.

[30] Wainwright, M. J., Jaakkola, T., and Willsky, A. S. Tree-based reparameterization for approximate inference on loopy graphs. In *NIPS 14*. MIT Press, 2002.

[31] Wainwright, M. J., Sudderth, E. B., and Willsky, A. S. Tree-based modeling and estimation of Gaussian processes on graphs with cycles. In *Advances in Neural Information Processing Systems 13*, pages 661–667. MIT Press, 2001.

[32] Weiss, Y. Correctness of local probability propagation in graphical models with loops. *Neural Computation*, 12:1–41, 2000.

[33] Weiss, Y. and Freeman, W. T. Correctness of belief propagation in Gaussian graphical models of arbitrary topology. *Neural Computation*, 13:2173–2200, 2001.

[34] Yedidia, J., Freeman, W. T., and Weiss, Y. Generalized belief propagation. In *NIPS 13*, pages 689–695. MIT Press, 2001.