In *Energy Minimization Methods in Computer Vision and Pattern Recognition* at Nice, September, 2001. Figueiredo, Zerubia, Jain, Eds., *Lecture Notes in Computer Science*, Springer.

A Markov Process Using Curvature for Filtering Curve Images

Jonas August and Steven W. Zucker

Center for Computational Vision and Control
Departments of Electrical Engineering and Computer Science
Yale University
51 Prospect Street, New Haven, CT 06520
{jonas.august,steven.zucker}@yale.edu

Abstract. A Markov process model for contour curvature is introduced via a stochastic differential equation. We analyze the distribution of such curves, and show that its mode is the Euler spiral, a curve minimizing changes in curvature. To probabilistically enhance noisy and low contrast curve images (e.g., edge and line operator responses), we combine this curvature process with the curve indicator random field, which is a prior for ideal curve images. In particular, we provide an expression for a nonlinear, minimum mean square error filter that requires the solution of two elliptic partial differential equations. Initial computations are reported, highlighting how the filter is curvature-selective, even when curvature is absent in the input.

1 Introduction

Images are ambiguous. One unpleasant consequence of this singular fact is that we cannot compute contours without making assumptions about image structure. Inspired by Gestalt psychology [16], most previous work has defined this structure as good continuity in orientation, that is to say, curves with varying orientation—high curvature—are rejected, and, conversely, straighter curves are enhanced. This is naturally phrased in terms of an energy functional on curves that minimizes curvature. In this paper, we present a stochastic model that instead aims to enforce good continuation in curvature, and thus minimizes *changes* in curvature.

To understand why we believe that good continuation in curvature is important, imagine the situation of a bug trying to "track" the contour in Fig. 1. Suppose the bug is special in that it can only "search" for its next piece of contour in a cone in front of it centered around its current predicted position and direction (i.e., orientation with polarity) [24, 6]. This strategy is appropriate so long as the contour is relatively straight. However, when the bug is on a portion of the contour veering to the right, it will constantly waste time searching to the left, perhaps even mistracking completely if the curvature is too large. In estimation terms, the errors of our searching bug are correlated, a tell-tale clue that the assumption that the contour is straight is biased. A good model would only lead to an unavoidable "uncorrelated" error. We present a Markov process that models not only the contour's direction, but also its local curvature.

It may appear that one may avoid these problems altogether by allowing a higher bound on curvature. However, this forces the bug to spend more time searching in a larger cone. In stochastic terms, this larger cone is amounts to asserting that the current (position, direction) state has a weaker influence on the next state; in other words, the prior on contour shape is weaker (less peaked

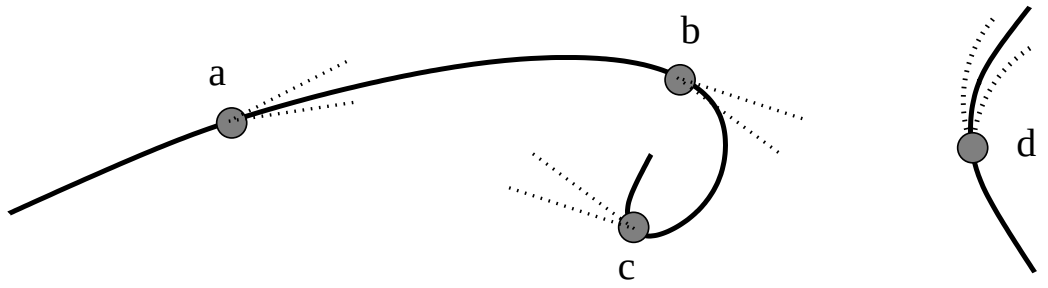


Fig. 1. Mistracking without curvature. A bug (grey dot) attempts to track the contour, "looking" in the cone of search directions centered around its current direction. At point (a), the curve is straight and the bug is successful, although at (b), curve is veering to the right and the bug can barely still track. At (c), the curvature is so high that tracking fails. A better model would explicitly include the curvature of the contour, giving rise to a "bent" search cone (d) for the bug. The same difficulty arises in contour enhancement, which is the application considered in this paper.

or broader). But a weaker prior will be less able to counteract a weak likelihood (high noise): it will not be robust to noise. Thus we must accept that *good continuation models based only on contour direction are forced to choose between allowing high curvature or high noise*; they cannot have it both ways.[1]

Although studying curvature is hardly new in vision, modeling it probabilistically is. In [3, 25, 15] and [8, 361 ff.], measuring curvature in images was the key problem. In [14], curvature is used for smooth interpolations, following on the work on elastica in [20, 10] and later [18]. The closest work in spirit to this is relaxation labeling [26], several applications of which include a deterministic form of curvature [19, 11]. Markov random fields for contour enhancement using orientation [17] and co-circularity [9, 21] have been suggested, but these have no complete stochastic model of individual curves. The explicit study of stochastic but direction-only models of visual contours was initiated by Mumford [18] and has been an extended effort of Williams and co-workers [22, 23].

This paper is organized as follows. We first introduce a curvature random process and its diffusion equation; we then present example impulse responses, which act like the "bent" search cones. Second, we relate the mode of the distribution for the curvature process to an energy functional on smooth curves. Next we review a model of an ideal curve image (e.g., "perfect" edge operator responses) called the curve indicator random field (CIRF), which was introduced in [2] and theoretically developed in [1], but in the context of Mumford's direction process [18]. Here we apply the CIRF to the curvature process, and then report a minimum mean square error filter for enhancing image contours. We conclude with some example computations.

2 A Brownian Motion in Curvature

Recall that a planar curve is a function taking a parameter $t \in \mathbb{R}$ to a point $(x(t), y(t))$ in the plane $\mathbb{R}^2$. Its direction is defined as $\theta = \arg(\dot{x} + \sqrt{-1}\,\dot{y})$, where the dot denotes differentiation with respect

[1] Observe in the road-tracking examples in [6] how all the roads have fairly low curvature. While this is realistic in flat regions such as the area of France considered, others, more mountainous perhaps, have roads that wind in the hillsides.

to the arc-length parameter t ($\dot{x}^2 + \dot{y}^2 = 1$ is assumed). Curvature κ is equal to $\dot{\theta}$, the rate of change of direction.

Now we introduce a Markov process that results from making curvature a Brownian motion. Let $R(t) = (X, Y, \Theta, K)(t)$ be random[2], with realization $r = (x, y, \theta, \kappa) \in \mathbb{R}^2 \times \mathbb{S} \times \mathbb{R}$. Consider the following stochastic differential equation:

$$\dot{X} = \cos\Theta, \qquad \dot{Y} = \sin\Theta, \qquad \dot{\Theta} = K, \qquad dK = \sigma dW,$$

where $\sigma = \sigma_k$ is the "standard deviation in curvature change" (see §3) and W denotes standard Brownian motion. The corresponding Fokker-Plank partial differential equation (PDE), describing the diffusion of a particle's probability density, is

$$\begin{aligned} \frac{\partial p}{\partial t} &= \frac{\sigma^2}{2}\frac{\partial^2 p}{\partial \kappa^2} - \cos\theta\frac{\partial p}{\partial x} - \sin\theta\frac{\partial p}{\partial y} - \kappa\frac{\partial p}{\partial \theta} \\ &= \frac{\sigma^2}{2}\frac{\partial^2 p}{\partial \kappa^2} - (\cos\theta, \sin\theta, \kappa, 0)\cdot\nabla p, \end{aligned} \tag{1}$$

where $p = p(x, y, \theta, \kappa, t) = p(R(t) = r | R(0) = r_0)$, the conditional probability density that the particle is located at r at time t given that it started at r_0 at time 0. Observe that this PDE describes probability transport in the $(\cos\theta, \sin\theta, \kappa, 0)$-direction at point $r = (x, y, \theta, \kappa)$, and diffusion in κ. An extra decay term [18, 22] is also included to penalize length (see §3). We have solved this parabolic equation by first analytically integrating the time variable and then discretely computing the solution to the remaining elliptic PDE. Details will be reported in [1]. See Fig. 2 for example time-integrated transition densities.

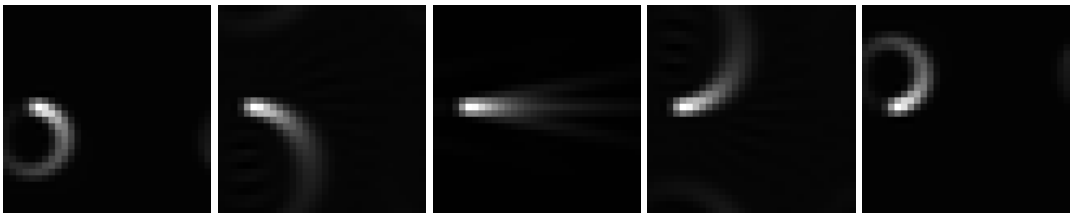


Fig. 2. Curvature diffusions for various initial curvatures. For all cases the initial position of the particle is an impulse centered vertically on the left, directed horizontally to the right. Shown is the time integral of the transition density of the Markov process in curvature (1), integrated over direction θ and curvature κ; therefore, the brightness displayed at position (x, y) indicates the expected time that the particle spent in (x, y). (Only a linear scaling is performed for displays in this paper; no logarithmic or other nonlinear transformation in intensity is taken.) Observe that the solution "veers" according to curvature, as sought in the Introduction. Contrast this with the straight search cone in Fig. 3. The PDE was solved on a discrete grid of size $32 \times 32 \times 32 \times 5$, with $\sigma_k = 0.01$ and an exponential decay of characteristic length $\lambda = 10$ (see §3 for length distribution).

[2] Capitals will often be used to denote random variables, with the corresponding letter in lower case denoting a realization. However, capitals are also used to denote operators later in the paper.

3 What is the Mode of the Distribution of the Curvature Random Process?

To get more insight into our random process in curvature, consider one of the simplest aspects of its probability distribution: its mode. First, let us consider the situation for Mumford's direction-based random process in[3] $\mathbb{R}^2 \times \mathbb{S}$, or

$$\dot{X} = \cos\Theta, \qquad \dot{Y} = \sin\Theta, \qquad d\Theta = \sigma_\kappa \, dW,$$

where σ_κ is the "standard deviation in curvature" and W is standard Brownian motion. This process has the following Fokker-Plank diffusion equation:

$$\frac{\partial p}{\partial t} = \frac{{\sigma_\kappa}^2}{2}\frac{\partial^2 p}{\partial \theta^2} - \cos\theta\frac{\partial p}{\partial x} - \sin\theta\frac{\partial p}{\partial y}, \tag{2}$$

where $p = p(x, y, \theta, t)$ is the transition density for time t. As Mumford has shown [18], the mode of the distribution of this direction process is described by *elastica*, or planar curves that minimize the following functional:

$$\int (\alpha\kappa^2 + \beta)dt, \tag{3}$$

where α and β are nonnegative constants. With such an elegant expression for the mode of the Mumford process, we ask: Is there a corresponding functional for the curvature process? If so, what is its form?

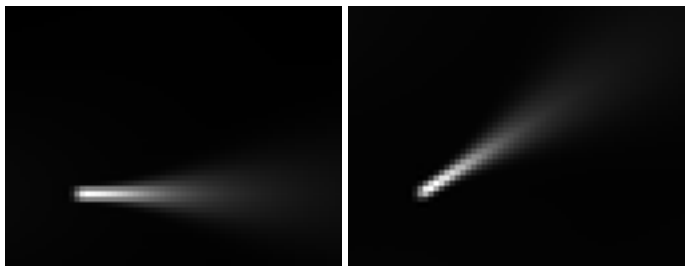


Fig. 3. Solutions to Mumford's diffusion. Equation (2) was integrated over time and then solved for a slightly blurred impulse on a $80 \times 80 \times 44$ grid, with parameters $\sigma_\kappa = 1/24$, $\lambda = 100$, and at discrete directions 0 (left) and 4 (right). Depicted is the integral over θ, cropped slightly. The method used [1] responds accurately at all directions. Note that these responses are straight, analogous the search cone described in the Introduction. Given their initial direction, particles governed by the direction process move roughly straight ahead, in contrast to those described by our curvature process (Fig. 2).

To answer these questions, we follow a line of analysis directly analogous to Mumford [18]. First, we discretize our random curve into N subsections. Then we write out the distribution and observe the discretization of a certain integral that will form our desired energy functional.

[3] $\mathbb{R}^2 \times \mathbb{S}$ is also called (x, y, θ)-space, the unit tangent bundle, and orientation space [13].

Suppose our random curve from the curvature process has length T, distributed with the exponential density $p(T) = \lambda^{-1}\exp(-\lambda^{-1}T)$, and independent of the shape of the contour. Each step of the N-link approximation to the curve has length $\Delta t := T/N$. Using the definition of the t-derivatives, for example,

$$\dot{X} = \frac{dX}{dt} = \lim_{N\to\infty} \frac{X_{i+1} - X_i}{T/N},$$

we can make the approximation $X_{i+1} \approx X_i + \Delta t\dot{X}$. Recalling the stochastic differential equation (1), we therefore let the curvature process be approximated in discrete time by

$$X_{i+1} = X_i + \Delta t\cos\Theta_i, \qquad Y_{i+1} = Y_i + \Delta t\sin\Theta_i, \qquad \Theta_{i+1} = \Theta_i + \Delta t K_i,$$

where $i = 1, \ldots, N$. Because Brownian motion has independent increments whose standard deviation grows with the square root $\sqrt{\Delta t}$ of the time increment Δt, the change in curvature for the discrete process becomes

$$K_{i+1} = K_i + \sqrt{\Delta t}\,\epsilon_i,$$

where $\{\epsilon_i\}$ is an independent and identically distributed set of 0-mean, Gaussian random variables of standard deviation $\sigma = \sigma_{\dot{\kappa}}$. Let the discrete contour be denoted by

$$\Gamma_N = \{(X_i, Y_i, \Theta_i, K_i) : i = 0, \ldots, N\}.$$

Given an initial point $p_0 = (x_0, y_0, \theta_0, \kappa_0)$, the probability density for the other points is

$$p(\Gamma_N|p_0) = \lambda^{-1}\exp(-\lambda^{-1}T)\cdot(\sqrt{2\pi}\sigma)^{-N}\exp\left(-\sum_i \frac{\epsilon_i^2}{2\sigma^2}\right),$$

which, by substitution, is proportional to

$$\exp\left[-\sum_i \frac{1}{2\sigma^2}\left(\frac{\kappa_{i+1}-\kappa_i}{\Delta t}\right)^2 \Delta t - \lambda^{-1}T\right].$$

We immediately recognize $\frac{\kappa_{i+1}-\kappa_i}{\Delta t}$ as an approximation to $\frac{d\kappa}{dt} = \dot{\kappa}$, and so we conclude that

$$p(\Gamma_N|p_0) \to p(\Gamma|p_0) \propto e^{-E(\Gamma)} \text{ as } N \to \infty,$$

where the energy $E(\Gamma)$ of (continuous) curve Γ is

$$E(\Gamma) = \int (\alpha\dot{\kappa}^2 + \beta)dt, \tag{4}$$

where $\alpha = (2\sigma^2)^{-1}$ and $\beta = \lambda^{-1}$.

Maximizers of the distribution $p(\Gamma)$ for the curvature random process are planar curves that minimize of the energy functional $E(\Gamma)$. Such curves are known as Euler spirals, and have been studied recently in [14]. A key aspect of the Euler spiral functional (4) is that it penalizes *changes* in curvature, preferring curves with slowly varying curvature. In contrast, the elastica functional (3) penalizes curvature itself, and therefore allows only relatively straight curves, to the dismay of the imaginary bug of the Introduction.

4 Filtering and the Curve Indicator Random Field

Given our stochastic shape model for contours, we now introduce the *curve indicator random field* (CIRF), which naturally captures the notion of an ideal curve image, and provides a precise definition for the kind of output we would like from an edge operator, for example. Roughly, this random field is non-zero-valued along the true contours, and zero-valued elsewhere. The actually measured edge/line map is then viewed as an *imperfect* CIRF, corrupted by noise, blur, etc. The goal of filtering, then, is to estimate the true CIRF given the imperfect one. For completeness, we review the theory of the CIRF now; proofs and more details can be found in [1].

4.1 Definitions

For generality, we shall define the curve indicator random field for any continuous-time Markov process $R_t, 0 \leq t < T$ taking values in a finite (or at most countable) set $\mathcal{I}$ of cardinality $|\mathcal{I}|$. As in §3, the random variable T is exponentially-distributed with mean value $\lambda > 0$, and represents the length of a contour. To ensure the finiteness of the expressions that follow, we assume $\lambda < \infty$. Sites or states within $\mathcal{I}$ will be denoted i and j. (Think of $\mathcal{I}$ as a discrete approximation to the state space $\mathcal{R} = \mathbb{R}^2 \times \mathbb{S} \times \mathbb{R}$ where the curvature random process takes values.) Let $\mathbb{1}\{condition\}$ denote the (indicator) function that takes on value 1 if *condition* is true, and the value 0 otherwise. With these notations we define the *curve indicator random field* V *for a single curve* to be

$$V_i := \int_0^T \mathbb{1}\{R_t = i\} dt, \qquad \forall i \in \mathcal{I}.$$

Observe that V_i is the (random) amount of time that the Markov process spent in state i. In particular, V_i is zero unless the Markov process passed through site i. In the context of Brownian motion or other symmetric processes, V is variously known as the occupation measure or the local time of R_t [4, 5].

Generalizing to multiple curves, we pick a random number $\mathcal{N}$, Poisson distributed with average value $\bar{\mathcal{N}}$. We then choose $\mathcal{N}$ independent copies $R_{t_1}^{(1)}, \ldots, R_{t_{\mathcal{N}}}^{(\mathcal{N})}$ of the Markov process R_t, with independent lengths $T_1, \ldots, T_{\mathcal{N}}$, each distributed as T. To define the (multiple curve) CIRF, we take the superposition of the single-curve CIRFs $V^{(1)}, \ldots, V^{(\mathcal{N})}$ for the $\mathcal{N}$ curves.

Definition 1. *The curve indicator random field U is defined to be*

$$U_i := \sum_{n=1}^{\mathcal{N}} V_i^{(n)} = \sum_{n=1}^{\mathcal{N}} \int_0^{T_n} \mathbb{1}\{R_{t_n}^{(n)} = i\} dt_n, \qquad \forall i \in \mathcal{I}.$$

Thus U_i is the total amount of time that all of the Markov processes spent in site i. Again, observe that this definition satisfies our desiderata for an ideal edge/line map: (1) non-zero value where the contours are, and (2) zero-value elsewhere. The probability distribution of U will become our prior for inference.

4.2 Statistics of the Curve Indicator Random Field

Probabilistic models in vision and pattern recognition have been specified in a number of ways. For example, Markov random field models [7] are specified via clique potentials and Gaussian models

are specified via means and covariances. Here, instead of providing the distribution of the curve indicator random field itself, we report its moment generating functional, from which all moments can be computed straightforwardly.

Before doing so, we need to develop more Markov process theory. We first define the inner product $(a,b) = \sum_{i\in\mathcal{I}} a_i b_i$. The *generator* of the Markov process R_t is the $|\mathcal{I}| \times |\mathcal{I}|$ matrix $L = (l_{ij})$, and is the instantaneous rate of change of the probability transition matrix $P(t) = (p_{ij})(t)$ for R_t. For the curvature process, we let L be a discretization of the partial differential operator on the right hand side of (1), or

$$L \approx \frac{{\sigma_\kappa}^2}{2}\frac{\partial^2}{\partial\kappa^2} - \cos\theta\frac{\partial}{\partial x} - \sin\theta\frac{\partial}{\partial y} - \kappa\frac{\partial}{\partial\theta} \qquad \text{(curvature process)},$$

and for the direction process, L is the discretization of the corresponding operator in (2), or

$$L \approx \frac{{\sigma_\kappa}^2}{2}\frac{\partial^2}{\partial\theta^2} - \cos\theta\frac{\partial}{\partial x} - \sin\theta\frac{\partial}{\partial y} \qquad \text{(direction process)}.$$

To include the exponential distribution over T (the lifetime of each particle), we construct a *killed* Markov process with generator $Q = L - \lambda^{-1}I$. (Formally, we do this by adding a single "death" state $\Cap$ to the discrete state space $\mathcal{I}$. When $t \geq T$, the process enters $\Cap$ and it cannot leave.) Slightly changing our notation, we shall now use R_t to mean the killed Markov process with generator Q. The Green's function matrix $G = (g_{ij})$ of the Markov process is the matrix $\int_0^\infty e^{Qt}dt = \int_0^\infty P(t)e^{-t/\lambda}dt$, where $P(t) = e^{Lt}$ (e^A denotes the matrix exponential of matrix A). The (i,j)-entry g_{ij} in the Green's function matrix G represents the expected amount of time that the Markov process spent in j before death, given that the process started in i. One can show that $G = -Q^{-1}$. Given a vector $c \in \mathbb{R}^{|\mathcal{I}|}$ having sufficiently small entries c_i, we define the Green's function matrix $G(c)$ with spatially-varying "creation" c as the Green's function matrix for the killed Markov process with extra killing $-c$, i.e., having generator $Q(c) := Q + \operatorname{diag} c$, where $\operatorname{diag} c$ is a diagonal matrix with the entries of vector c along the diagonal; in particular, $G(c) = -Q(c)^{-1} = -(Q + \operatorname{diag} c)^{-1}$.

Recalling that each Markov process $R^{(1)}_{t_1}, \ldots, R^{(\mathcal{N})}_{t_\mathcal{N}}$ is distributed as R_t, let the joint distribution of the initial and final states of R_t be

$$\mathbb{P}\{R_0 = i, R_{T-} = j\} = \mu_i g_{ij} \nu_j, \qquad \forall i,j \in \mathcal{I},$$

where $\mu = (\mu_i)$ and $\nu = (\nu_i)$ are vectors in $\mathbb{R}^{|\mathcal{I}|}$ weighting initial and final states, respectively, and $f(T-)$ is the limit of $f(t)$ for t approaching T from below. Therefore, vectors μ and ν must satisfy the *normalization constraint* $(\mu, G\nu) = 1$. For general purpose contour enhancement, we typically have no a-priori preference for the start and end locations of each contour, and so we set these vectors proportional to the constant vector $\mathbf{1} = (1, \ldots, 1)$. For example, by setting $\mu = |\mathcal{I}|^{-1}\mathbf{1}, \nu = \lambda^{-1}\mathbf{1}$, the normalization constraint is satisfied.

The following key theoretical result is used in this paper but is developed in [1], and is most closely related to the work of Dynkin [4].

Proposition 1. *The moment generating functional of the curve indicator random field U is*

$$\mathbb{E}\exp(c,U) = \exp(\mu, \bar{\mathcal{N}}(G(c) - G)\nu).$$

While this result may seem abstract, it is actually very useful. Let G^* denote the transpose of G. From Prop. 1 we obtain the first two cumulants of the CIRF [2]:

Corollary 1. *Suppose the* $\mu = |\mathcal{I}|^{-1}\mathbf{1}, \nu = \lambda^{-1}\mathbf{1}$. *The mean of the curve indicator random field* U *is* $\mathbb{E}U_i = \bar{\mathcal{N}}\lambda|\mathcal{I}|^{-1}, \forall i \in \mathcal{I}$. *The covariance matrix of* U *is* $\operatorname{cov} U = \bar{\mathcal{N}}\lambda|\mathcal{I}|^{-1}(G + G^*)$.

Several "columns" of the covariance matrix for the curvature process are illustrated in Fig. 4, by taking its impulse response for several positions, directions and curvatures. In addition, using Prop. 1

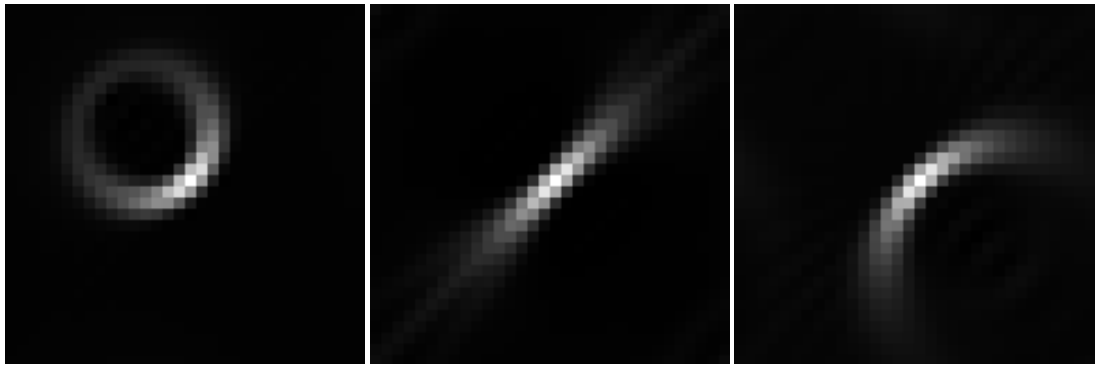


Fig. 4. Impulse responses of the covariance matrix for the curve indicator random field for the curvature process. Impulses are located at the center of each image, directed at discrete direction 4 out of 32, with 5 curvatures. Parameters are $\sigma_\dot{\kappa} = 0.01, \lambda = 10$.

one can compute the higher-order cumulants of U, and they are generally *not* zero, which shows that *the curve indicator random field is non-Gaussian* [1]. Despite that, its moment generating functional has a tractable form that we shall directly exploit next.

5 Minimum Mean Square Error Filtering

Instead of the unknown random field U, what we actually observe is some realization m of a random field M of (edge or line) *measurements*. Given m, we seek that approximation $\tilde{u}$ of U that minimizes the mean square error (MMSE), or

$$\tilde{u} \triangleq \arg\min_u \mathbb{E}_m ||u - U||^2,$$

where $\mathbb{E}_m$ the denotes taking an expectation conditioned on the measurement realization m. It is well-known that the posterior mean is the MMSE estimate

$$\tilde{u} = \mathbb{E}_m U,$$

but in many interesting, non-Gaussian, cases this is extremely difficult to compute. In our context, however, we are fortunate to be able to make use of the moment generating functional to simplify computations.

Before developing our MMSE estimator, we must define our likelihood function $p(M|U)$. First let H_i be the binary random variable taking the value 1 if one of the contours passed through (or

"hit") site i, and 0 otherwise, and so H is a binary random field on $\mathcal{I}$. In this paper we consider conditionally independent, local likelihoods: $p(M|H) = p(M_1|H_1)\cdots p(M_{|\mathcal{I}|}|H_{|\mathcal{I}|})$. Following [6, 24], we consider two distributions over measurements at site i: $p_{\text{on}}(M_i) := p(M_i|H_i = 1)$ and $p_{\text{off}}(M_i) := p(M_i|H_i = 0)$. It follows [24] that $\ln p(M|H) = \sum_i \ln(p_{\text{on}}(M_i)/p_{\text{off}}(M_i))H_i$. Now let τ be the average amount of time spent by the Markov processes in a site, given that the site was hit; observe that U_i/τ and H_i are therefore equal on average. This suggests that we replace H with U/τ above to generate a likelihood in U, in particular,

$$\ln p(M|U) \approx \sum_i c_i U_i = (c, U), \text{ where } c_i = c_i(M_i) = \tau^{-1} \ln \frac{p_{\text{on}}(M_i)}{p_{\text{off}}(M_i)}.$$

As shown in [1], the posterior mean in this case becomes

$$\mathbb{E}_m U_i \approx \frac{\partial}{\partial c_i}(\mathbb{E}\exp(c, U))(c) = \bar{\mathcal{N}} f_i b_i, \qquad \forall i \in \mathcal{I}, \tag{5}$$

where $f = (f_1, \ldots, f_{|\mathcal{I}|})$ is the solution to the forward equation

$$(Q + \operatorname{diag} c)f + \nu = 0 \tag{6}$$

and $b = (b_1, \ldots, b_{|\mathcal{I}|})$ is the solution to the backward equation

$$(Q^* + \operatorname{diag} c)b + \mu = 0. \tag{7}$$

Note that equations (6) and (7) are linear systems in $\mathbb{R}^{|\mathcal{I}|}$; however, since $Q = L - \lambda^{-1}$, where the generator L is the discretization of an (elliptic) partial differential operator, we view the forward and backward equations as (linear) elliptic partial differential equations, by replacing L with its undiscretized counterpart, and c, f, b, μ, and ν with corresponding (possibly generalized) functions on a continuum, such as $\mathbb{R}^2 \times \mathbb{S}$ for the direction process, or $\mathbb{R}^2 \times \mathbb{S} \times \mathbb{R}$ for the curvature process.

Observe that two nonlinearities arise in this posterior mean in equation (5). First, there is a product of the forward and backward solutions.[4] Second, although both the forward and backward equations are linear, they represent nonlinear mappings from input (c) to output (f or b). For example, it follows that $f = (I - G\operatorname{diag} c)^{-1} G\nu = \sum_{k=0}^{\infty} (G \operatorname{diag} c)^k G\nu$, i.e., f is a polynomial—and thus nonlinear—function of the input c.

5.1 Example Computations

We have implemented a preliminary version of the CIRF posterior mean filter. For our initial experimentation, we adopted a standard additive white Gaussian noise model for the likelihood $p(M|U)$. As a consequence, we have $c(m) = \gamma_1 m - \gamma_2$, a simple transformation of the input m, where γ_1 and $\gamma_2 = 3$ are constants. The direction-dependent input m was set to the result of logical/linear edge and line operators [12]. The output of the logical/linear operator was linearly interpolated to a many directions as necessary. For direction-only filtering, this input interpolation was sufficient, but not for the curvature-based filtering, as *curvature was not directly measured in the image*; instead, the directed logical/linear response was simply copied over all curvature values, i.e., the input m was

[4] This is analogous to the source/sink product in the stochastic completion field [22].

constant as a function of curvature. The limit of the invertibility of $Q + \text{diag}\, c$ and its transpose was used to set γ_1 [1]. The technique used to solve the forward and backward CIRF equations for the curvature process in (x, y, θ, κ) will be reported in [1], and are a generalization of the method used for the forward and backward CIRF equations for the Mumford (direction) process in (x, y, θ) [1], which is also used here. Parameter settings were $\bar{\mathcal{N}} = 1, \mu = |\mathcal{I}|^{-1}\mathbf{1}, \nu = \lambda^{-1}\mathbf{1}$. The direction-process based CIRF was solved on a grid the size of the given image but with 32 directions. For the curvature-based CIRF filtering, very few curvatures were used (3 or 5) in order to keep down computation times in our unoptimized implementation. Unless we state otherwise, all filtering responses (which are fields over either discrete (x, y, θ)-space or discrete (x, y, θ, κ)-space) are shown summed over all variables except (x, y).

For our first example, we considered a blood cell image (Fig. 5, top). To illustrate robustness, noise was added to a small portion of the image that contained two cells (top left), and was processed with the logical/linear edge operator at the default settings[5]. The result was first filtered using the CIRF posterior mean based on Mumford's direction process (top center). Despite using two very different bounds on curvature, the direction-based filtering cannot close the blood cell boundaries appropriately. In contrast, the CIRF posterior mean with the curvature process (top right) was more effective at forming a complete boundary. To illustrate in more detail, we plotted the filter responses for the direction-based filter at $\sigma_\kappa = 0.025$ for 8 of the 32 discrete directions in the middle of Fig. 5. The brightness in each of the 8 sub-images is proportional to the response for that particular direction as a function of position (x, y). Observe the over-straightening effect shown by the elongated responses. The curvature filter responses were plotted as a function of direction and curvature (bottom). Despite the input having been constant as a function of curvature, the result shows *curvature selectivity*. Indeed, one can clearly see in the $\kappa > 0$ row (Fig. 5, bottom) that the boundary of the top left blood cell is traced out in a counter-clockwise manner. In the $\kappa < 0$ row, the same cell is traced out in the opposite manner. (Since the parameterization of the curve is lost when forming its image, we cannot know which way the contour was traversed; our result is consistent with both ways.) The response for the lower right blood cell was somewhat weaker but qualitatively similar. Unlike the direction-only process, the curvature process can effectively deal with highly curved contours.

For our next example, we took two sub-images of a low-contrast angiogram (top of Fig. 6; sub-images from left and top right of original). The first sub-image (top left) contained a straight structure, which was enhanced by our curvature-based CIRF filter (summed responses at top right). The distinct responses at separate directions and curvatures show curvature selectivity as well, since the straight curvature at 45° had the greatest response (center). The second sub-image (bottom left) of a loop structure also produced a reasonable filter response (bottom right); the individual responses (bottom) also show some curvature selectivity.

As argued in the Introduction, the bug with a direction-only search cone would mistrack on a contour as curvature builds up. To make this point computationally, we consider an image (top left of Fig. 7) of an Euler spiral extending from a straight line segment.[6] Observe that the contour curvature begins at zero (straight segment) and then builds up gradually. To produce a 3-dimensional input to our direction-based filter, this original (2-d) image was copied to all directions (i.e., $m(x, y, \theta) =$

[5] Code and settings available at http://www.ai.sri.com/~leei/loglin.html.

[6] We used formula (16.7) of Kimia et al [14], and created the plot in Mathematica with all parameters 0, except $\gamma = 0.1$ (Kimia et al's notation). The resulting plot was grabbed, combined with a line segment, blurred with a Gaussian, and then subsampled.

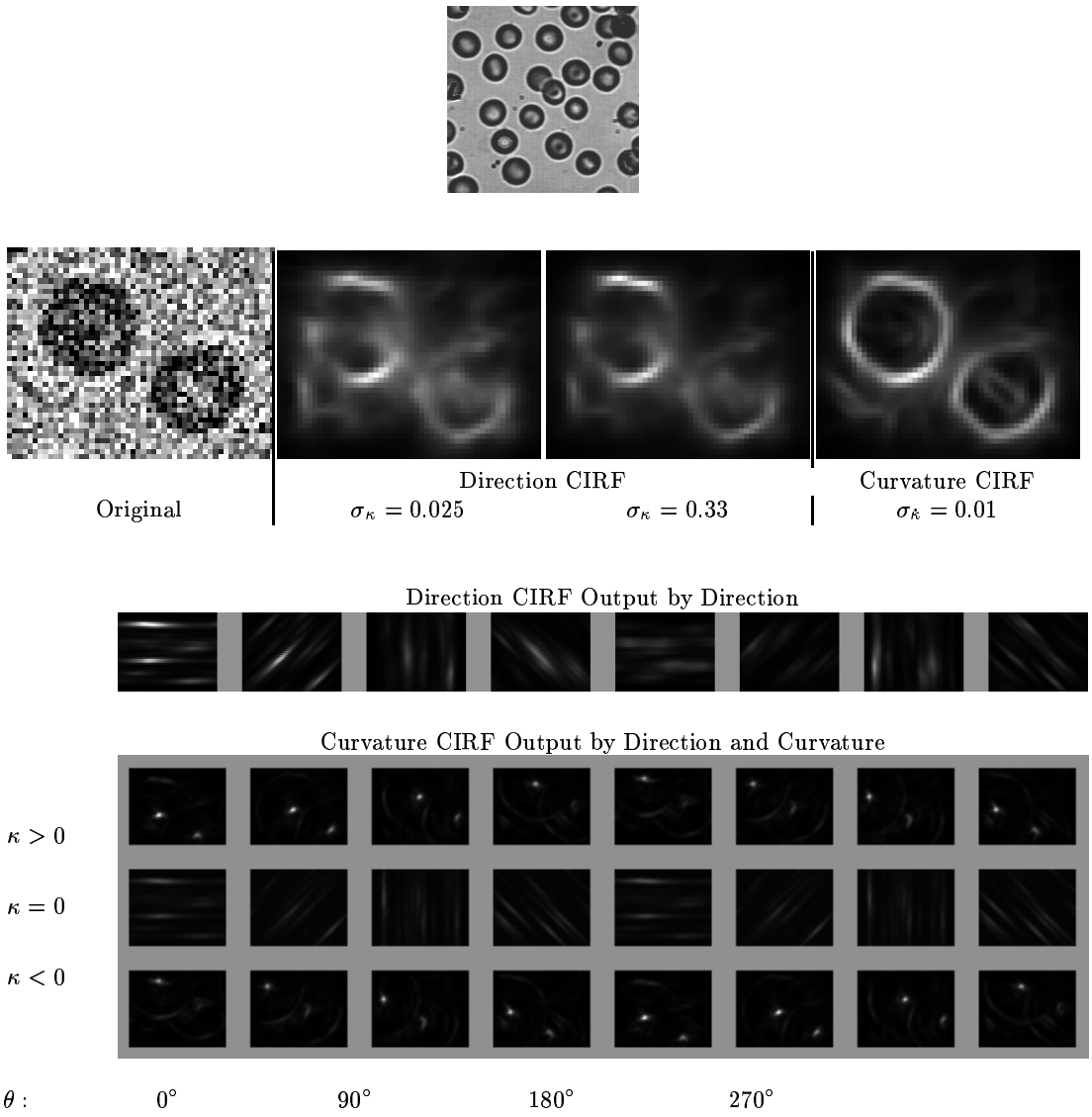
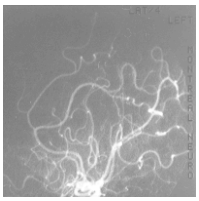
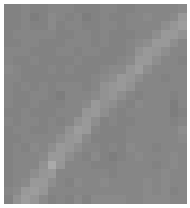
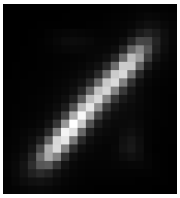


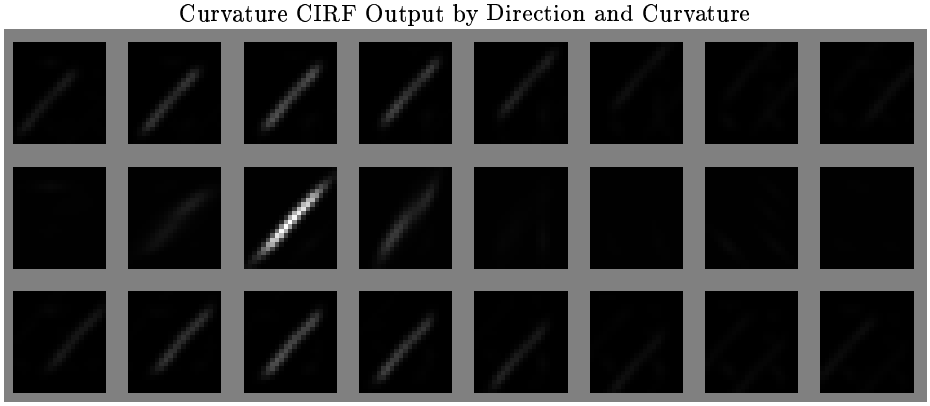
Fig. 5. Curvature filtering of a blood cell image (see text).

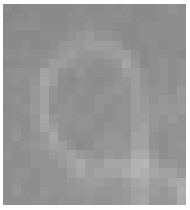


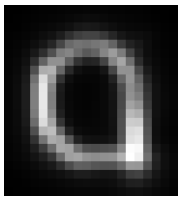




Curvature CIRF Output by Direction and Curvature







Curvature CIRF Output by Direction and Curvature

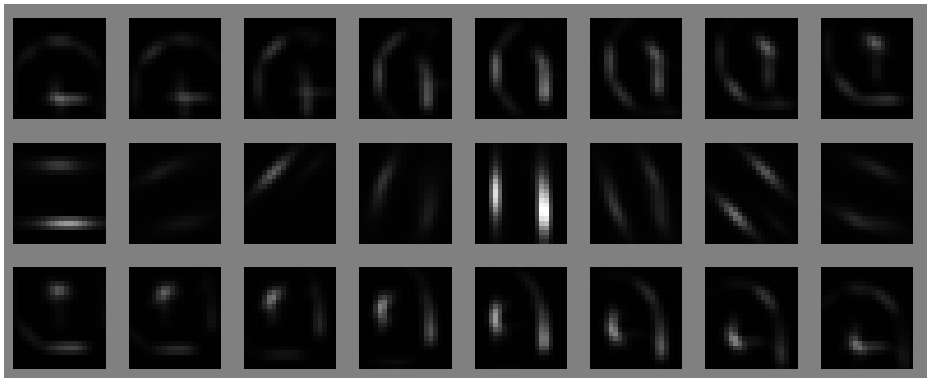


Fig. 6. Curvature filtering for an angiogram (see text).

image(x, y), for all θ). Similarly, the image was copied to all directions and curvatures to produce a 4-d input to the curvature-based filter. For this test only, our 2-dimensional outputs were produced by taking, at each position (x, y), the maximum response over all directions (for the direction-based filtering) or over all directions and curvatures (for the curvature-based filtering). The direction-based CIRF posterior mean (with parameters $\sigma_\kappa = 0.025, \lambda = 10$, with 64 directions) was computed (center), showing an undesirable reduction in response as curvature built up. The curvature-based CIRF posterior mean (right, with parameters $\sigma_{\dot{\kappa}} = 0.05, \lambda = 10, 64$ directions, and 7 curvatures $(0, \pm 0.05, \pm 0.1, \pm 0.15)$) shows strong response even at the higher curvature portions of the contour. To test robustness, 0-mean Gaussian noise of standard deviation 0.4 was added (bottom left) to the image (0 to 1 was the signal range before adding noise). The results (bottom center and right) show that the curvature-based filter performs better in high curvature regions despite noise.

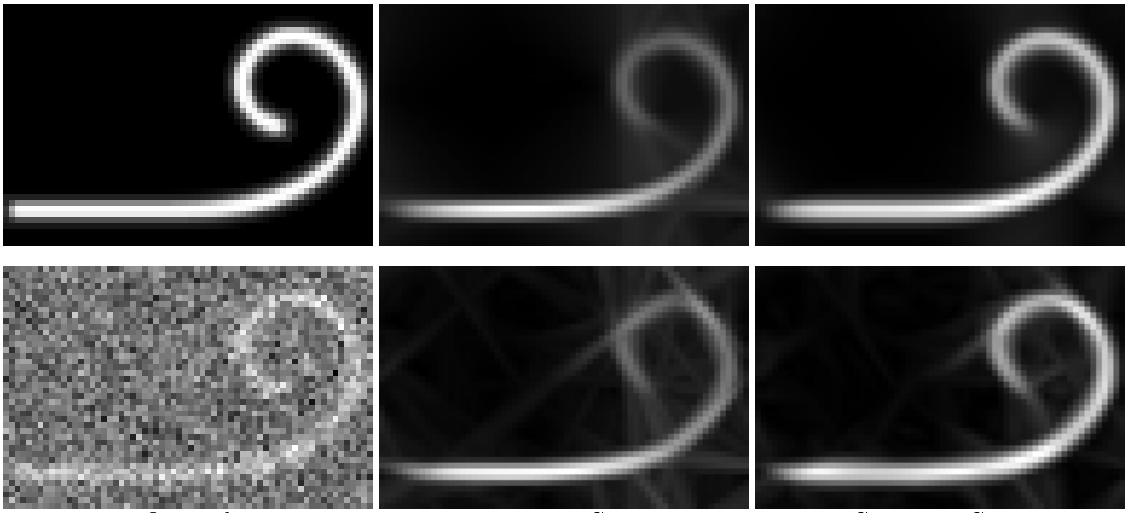


Fig. 7. Filtering an Euler spiral without noise (top) and with noise (bottom). The original images are on the left; the result after filtering using the curve indicator random field based on Mumford's direction-based Markov process (center) and our curvature-based Markov process (right). Notice that the direction CIRF result tends to repress the signal at high curvatures, while the curvature process has more consistent performance, even at higher curvatures. See text for details.

Computations were conveniently performed using the Python scripting language with numerical extensions in a GNU/Linux environment.

6 Conclusion

In this paper we introduced a new stochastic model for contour curvature to more faithfully capture the shape of image curves. Whereas most contour models penalize large curvatures, our curvature

Markov process allows highly curving contours, and only penalizes changes in curvature. The curvature process can be used directly in the curve indicator random field [2, 1] to construct a prior for curve images. To enhance noisy images of contours, we present a nonlinear filter (details in [1]) that approximates the posterior mean of the curve indicator random field. Our initial computations show that the filter responds well along smooth contours, even those having large curvature.

Acknowledgements

We thank Patrick Huggins, Athinodoros Georghiades, Ohad Ben-Shahar, and the reviewers for their helpful comments. This work was supported by AFOSR.

References

1. J. August. *The Curve Indicator Random Field*. PhD thesis, Yale University, 2001.
2. J. August and S. W. Zucker. The curve indicator random field: curve organization via edge correlation. In K. Boyer and S. Sarkar, editors, *Perceptual Organization for Artificial Vision Systems*, pages 265–288. Kluwer Academic, Boston, 2000.
3. A. Dobbins, S. W. Zucker, and M. S. Cynader. Endstopped neurons in the visual cortex as a substrate for calculating curvature. *Nature*, 329(6138):438–441, 1987.
4. E. B. Dynkin. Markov processes as a tool in field theory. *Journal of Functional Analysis*, 50:167–187, 1983.
5. E. B. Dynkin. Gaussian and non-gaussian fields associated with markov processes. *Journal of Functional Analysis*, 55:344–376, 1984.
6. D. Geman and B. Jedynak. An active testing model for tracking roads in satellite images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(1):1–14, 1996.
7. S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.
8. G. H. Granlund and H. Knutsson. *Signal Processing for Computer Vision*. Kluwer Academic, Dordrecht, 1995.
9. L. Herault and R. Horaud. Figure-ground discrimination: A combinatorial optimization approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):899–914, 1993.
10. B. K. P. Horn. The Curve of Least Energy. *ACM Transactions on Mathematical Software*, 9:441–460, 1983.
11. L. A. Iverson. *Toward Discrete Geometric Models for Early Vision*. PhD thesis, McGill University, Montreal, 1994.
12. L. A. Iverson and S. W. Zucker. Logical/linear operators for image curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(10):982–996, 1995.
13. S. N. Kalitzin, B. M. ter Haar Romeny, and M. A. Viergever. Invertible orientation bundles on 2d scalar images. In *Proc. Scale-Space '97*, LICS, pages 77–88. Springer, 1997.
14. B. B. Kimia, I. Frankel, and A.-M. Popescu. Euler spiral for shape completion. In K. Boyer and S. Sarkar, editors, *Perceptual Organization for Artificial Vision Systems*, pages 289–309. Kluwer Academic, Boston, 2000.
15. J. J. Koenderink and W. Richards. Two-dimensional curvature operators. *J. Opt. Soc. Am. A*, 5(7):1136–1141, 1988.
16. K. Koffka. *Principles of Gestalt Psychology*. Harcourt, Brace & World, New York, 1963.
17. J. L. Marroquin. A markovian random field of piecewise straight lines. *Biological Cybernetics*, 61:457–465, 1989.
18. D. Mumford. *Algebraic Geometry and Its Applications*, chapter Elastica and Computer Vision, pages 491–506. Springer-Verlag, 1994.

19. P. Parent and S. W. Zucker. Trace inference, curvature consistency, and curve detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(8):823–839, August 1989.
20. S. Ullman. Filling-in gaps: The shape of subjective contours and a model for their generation. *Biological Cybernetics*, 25:1–6, 1976.
21. S. Urago, J. Zerubia, and M. Berthod. A markovian model for contour grouping. *Pattern Recognition*, 28(5):683–693, 1995.
22. L. Williams and D. Jacobs. Stochastic completion fields: A neural model of illusory contour shape and salience. *Neural Computation*, 9(4):837–858, 1997.
23. L. Williams, T. Wang, and K. Thornber. Computing stochastic completion fields in linear-time using a resolution pyramid. In *Proc. of 7th Intl. Conf. on Computer Analysis of Images and Patterns*, Kiel, Germany, 1997.
24. A. L. Yuille and J. M. Coughlan. Fundamental bounds of bayesian inference: Order parameters and phase transitions for road tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(2):160–173, 2000.
25. S. W. Zucker, A. Dobbins, and L. Iverson. Two stages of curve detection suggest two styles of visual computation. *Neural Computation*, 1:68–89, 1989.
26. S. W. Zucker, R. Hummel, and A. Rosenfeld. An application of relaxation labelling to line and curve enhancement. *IEEE Trans. Computers*, C-26:393–403, 922–929, 1977.