Project no. FP6-507752

MUSCLE

Network of Excellence
Multimedia **U**nderstanding through **S**emantics, **C**omputation and **LE**arning

Computation Intensive Methods: State of the Art

Due date of deliverable: 31.09.2004
Actual submission date: 10.10.2004

Start date of project: 1 March 2004 Duration: 48 Months

Workpackage: 7
Deliverable: D7.1

Editors:
Simon Wilson and Rozenn Dahyot
Department of Statistics
Trinity College Dublin
Dublin 2, Ireland

Revision 1.0

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	x
PP	Restricted to other programme participants (including the commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Keyword List: Intensive Computation, MCMC, Multimedia Understanding, etc.

Contents

Introduction

This State of the Art report fulfills the requirements of Deliverable 7.1 of the EU-funded Network of Excellence MUSCLE.

It is part of the programme of work for Workpackage 7 “Computation Intensive Methods”. There are 2 objectives of the report:

1. To provide some background information and references for the major areas of research in computational methods for multimedia data;

2. To describe the current state of activities of those members of MUSCLE that are conducting research that uses or develops computation intensive methods.

To this end, the report is divided into two chapters. In the first chapter you will find descriptions of the major areas of research. These are a mixture of lists of references and articles by MUSCLE members. The second chapter describes current work by members of the MUSCLE consortium. This chapter is also a mixture of articles and references to work. The work described in this chapter is continuing and will form part of the research efforts of the MUSCLE consortium over the next 42 months.

Simon Wilson
Trinity College Dublin, August 2004.

Chapter 1

Current State of the Art in Computational Methods for Multimedia Understanding

1.1 Markov Chain Monte Carlo

1.1.1 Introduction to MCMC

Author: Simon Wilson, Trinity College Dublin Ireland

It is 20 years since the first application of Markov chain Monte Carlo (MCMC) methods to multimedia data [41]. Since then they have seen wide application to problems in audio, video and image analysis; indeed, it is probably reasonable to assert that MCMC methods have been successfully used in all principal areas of multimedia data analysis. Combined with the large increases in computing power over the last 20 years, they have revolutionised what can be achieved in many areas.

The next section of this chapter is a more technical description of the main ideas of MCMC. For those who are looking for introductory texts to the field, a good place to start is [14]. This paper is aimed at a statistical audience. There is a very nice review of Monte Carlo methods, including MCMC, by [82], available from the author's webpage.

There are several books on MCMC methods. An early book of applications, including some in image processing, is [42]. In [106], the main methods of MCMC are described from a more theorectical perspective. Applying MCMC to problems in image analysis is the subject of [115]. For those using MCMC with Bayesian methods, [38] is full of examples (although none in multimedia data unfortunately) and good advice on the practicalities of implementing MCMC methods. The latter book describes several diagnostics for checking that the MCMC method has converged and is mixing well, both very important aspects of any MCMC algorithm ; [19] is a review of the principal ideas. Another important idea is the "reversible jump" sampler, for use in problems where the dimensionality of the distribution being explored is unknown; see [47].

Finally, there is the "exact" MCMC approach of [92]. While this in principle solves all the problems associated with convergence and mixing, it has proved difficult to apply in all but a few selected situations. There are not yet any applications of it to multimedia data.

1.1.2 Stochastic Algorithms

Author: Michal Haindl, UTIA Czech Republic

Most multidimensional probability densities (e.g., random fields) parameter estimates and their realization cannot be directly computed except for few treatable cases since the underlying space is too large. Therefore, static Monte Carlo methods have to be replaced by dynamic simulation of feasible Markov chains with required limit distribution (Markov Chain Monte Carlo methods - MCMC). The MCMC methods provide direct approximations of required probabilities rather than the usual indirect (e.g., asymptotic distribution fitting) alternatives. The principle of MCMC is to construct such an ergodic transition kernel of a Markov chain $\tilde{p}(Y^t \to Y^{t+1})$, so that $\tilde{p}(Y^t \to Y^{t+1})$ is the probability associated with a transition from $Y^t \to Y^{t+1}$ and has the required limit distribution $p(Y)$. If the transition probability does not depend on the step t then the corresponding Markov chain is called homogeneous otherwise it is the inhomogeneous Markov chain. Then for single chain realizations $Y^1, Y^2, \ldots, Y^n$ the ergodic theorem implies [12] that for any seed Y^1, the random sequence

$$\bar{f}_n = \frac{1}{n}\sum_{i=1}^{n} f(Y^i)$$

converges almost surely to $E\{f\}$

$$E\{f\} = \sum_{Y} f(Y)p(Y) \tag{1.1}$$

as $n \to \infty$. Hence the mean value (1.1) is approximated by the empirical (ergodic) average $\bar{f}_n$ obtained for sufficiently long run of the chain $Y^1, Y^2, \ldots, Y^n$. The MCMC methods use the fact that

$$p(Y_A \,|\, Y_{(A)}) \propto p(Y) \qquad\qquad A \subset I$$

where (A) denotes set complement of A. Hence if two realizations $Y, \acute{Y}$ fulfil $Y_{(A)} = \acute{Y}_{(A)}$ then

$$\frac{p(\acute{Y}_A \,|\, \acute{Y}_{(A)})}{p(Y_A \,|\, Y_{(A)})} = \frac{p(\acute{Y})}{p(Y)} \quad . \tag{1.2}$$

Using (1.2) we do not need to know the normalization constant Z which is usually intractable analytically as well as numerically. Another considerable simplification is usually possible if the involved densities are of the product type.

Stochastic algorithms use dynamic Monte Carlo techniques to generate required random field (RF) realizations (equilibrium states). These algorithms can be also used for generating the ground states of RF, i.e.,

$$\Omega_{\min} = \{\tilde{Y} : Q(\tilde{Y}) = \min_{Y} Q(Y)\} \tag{1.3}$$

or simply for finding a minimum of any function $Q(Y)$ if we introduce an additional parameter to the Gibbs random field (GRF), called temperature. This parameter is decreased according

to a special schedule (simulated annealing). Similarly it is possible to solve the constraint optimization problem:

$$\tilde{\Omega}_{\min} = \{\tilde{Y} : \ Q(\tilde{Y}) = \min_{Y, B(Y)=b} Q(Y)\} \ , \tag{1.4}$$

if the energy function is redefined as

$$\tilde{Q}(Y) = \frac{Q(y) + \lambda B(Y)}{T}$$

and the normalization constant is

$$Z_T = \sum_{Y \in \Omega} \exp\{-\tilde{Q}(Y)\}$$

then

$$\lim_{\lambda \to \infty, T \to 0} p(Y; T, \lambda) = p(Y^g) \ .$$

In contrast to deterministic algorithms, stochastic algorithms permit changes that can decrease the posterior distribution (objective function) as well and so to avoid being stick in a local maxima. Hence these algorithms theoretically guarantee convergence towards the global optimum of a highly non-linear non-convex objective function irrespectively of an initial system state. The computational complexity of stochastic algorithms depends on spatial and measurements quantization (e.g., spectral resolution). The computational complexity increases for a fixed index set size $n = card\{I\}$ approximately linearly with the number of quantization levels. Geman and others [39] proposed an approximation to restrict the configuration space by ignoring quantization levels which do not occur in $P(Y_r|Y_s \ \forall s \in I_r)$, where I_r is the first order hierarchical neighbourhood. This approximation decreases number of operations necessary to generate a sample in order of magnitude [39].

The MCMC algorithms can be used to simulate complicated multivariate distributions. Suppose we need to simulate a multivariate continuous distribution $p(Y) \propto \exp\{-Q(Y)\}$ and we assume the existence the vector of partial derivatives $\nabla Q(Y)$. Then the stochastic differential equation $dY^t = -\nabla Q(Y^t)dt + \sqrt{2}dw^t$, where w^t is standard $n-$dimensional Brownian motion defines a continuous-time Langevin diffusion which has the required stationary distribution $p(Y)$ [12]. The MCMC replacement is

$$\acute{Y} \sim \mathcal{N}(Y^{t-1} - a\nabla Q(Y^{t-1}), 2aI_n) \ ,$$

where $a > 0$ is a small constant.

The general formulation for the sampling task can include observed data X, unobserved data Y, a vector of parameters θ and hyperparameters ϕ, respectively. The posterior density is then

$$p(\phi, \theta, Y \,|\, X) \propto p(X, Y \,|\, \theta)\, p(\theta \,|\, \phi)\, p(\phi)$$

with the following corresponding conditionals

$$
\begin{aligned}
p(\theta_r \,|\, \theta_{(r)}, \phi, Y, X) &\propto p(X, Y \,|\, \theta)\, p(\theta_r \,|\, \theta_{(r)}, \phi) \\
p(\phi_r \,|\, \theta, \phi_{(r)}, Y, X) &\propto p(\theta \,|\, \phi)\, p(\phi_r \,|\, \phi_{(r)}) \\
p(Y_r \,|\, \theta, \phi, Y_{(r)}, X) &\propto p(X, Y \,|\, \theta) \quad .
\end{aligned}
$$

The convergence of stochastic algorithms makes only weak demands on the manner in which pixels are visited. Any asynchronous method of updating deterministic or stochastic is acceptable, provided that each pixel is visited infinitely often. The procedure may even be synchronous to the extent that no two pixels which are neighbours should be simultaneously updated. A complete parallelization of a single site updating is wrong because the obtained sequence of configurations converges in distribution to a different distribution than the required one (usually even not having a Gibbs representation). If the joint probability $p(Y)$ is highly multimodal a single-site updating algorithm will have very slow convergence. A possible remedy can be to run the algorithm several times from different starting points and to combine these results. However such a coherent combination is a problem itself. Another possibility is the introduction of auxiliary variables to design simple Markov chains that can make substantial changes to many components at once. In these methods (see for example [11]) a variable Y is augmented by one or more additional variables X and a Markov chain is constructed that alternates between two types of transitions: X is drawn from $p(X|Y)$ and $\acute{Y}$ is generated given X, Y in such a way that the detailed balance for $p(Y|X)$ is preserved. Difficult question is to decide if a stochastic algorithm run was sufficiently long or if it is more efficient to run one long single-chain run or to combine results from several independent chains running in parallel. Useful bounds on rates of convergence are not known in general. For Gaussian distributions are available results on relationship between the target distribution correlation structure and the Gibbs sampler convergence rate [5].

The following stochastic algorithms are particular instances of a class of dynamic systems known also as stochastic cellular automata (SCA) and technically they are regular Markov chains whose invariant distribution is the Gibbs measure of the corresponding MRF. Their asymptotic behaviour was intensively studied and is relatively well understood. Their transient dynamic behaviour (convergence rate), however, is much less understood.

1.1.2.1 Metropolis Algorithm

Given the state Y^{t-1} of a MRF, another random configuration $\acute{Y}$ is chosen [87] such that $\acute{Y}_{(A)} = Y^{t-1}_{(A)}, \acute{Y}_A \neq Y^{t-1}_A$ where $A \subset I$. The transition kernel $\tilde{p}(Y^{t-1}_A \rightarrow \acute{Y}_A; Y^{t-1}_{(A)})$ is chosen to be symmetric in $Y^{t-1}_A, \acute{Y}_A$. The ratio

$$
\alpha = \frac{p(\acute{Y})}{p(Y^{t-1})}
$$

is computed. If $\alpha > 1$, or $Q(\acute{Y}) < Q(Y^{t-1})$ then $Y^t = \acute{Y}$, otherwise the transition is made with probability α. A variable ξ is selected from a standardized uniform distribution, if $\xi \leq \alpha$ then $Y^t = \acute{Y}$, otherwise $Y^t = Y^{t-1}$. Because a less favourable configuration is not automatically rejected the algorithm can escape from local minima.

A modification of the algorithm is the Exchange algorithm [20], where $\acute{Y}$ is obtained from Y^{t-1} by exchanging values of two randomly chosen pixels. The exchange algorithm keeps the overall distribution (intensity histogram) of RF values fixed. The disadvantage of this method is its sensitivity to initial configuration [18]. In the "single-flip" [40] algorithm, $\acute{Y}$ is obtained from Y^t by changing the value of one randomly chosen pixel $(card\{A\} = 1)$. The exchange algorithm similarly as the "single-flip" algorithm does the change with probability

$$\frac{\alpha}{1+\alpha}.$$

Some other algorithms of the Metropolis-Hastings type can be found in [102] and the convergence theorems in [114].

1.1.2.2 Gibbs Sampler

The Gibbs sampler [40] (also known as stochastic relaxation or the heath bath method) generates realizations from a given MRF using a relaxation technique similar to the Metropolis algorithm. The Gibbs sampler is a special case of the simulated annealing (1.1.2.3) with a fixed temperature. The sampler is also the Metropolis algorithm with zero rejection probability. The transition kernel $\tilde{p}(Y_A^{t-1} \rightarrow \acute{Y}_A; Y_{(A)}^{t-1})$ is independent of Y_A^{t-1}. Gibbs samplers usually require univariate updates $(card(A) = 1$. The stationary configuration Y^0 is arbitrary. By repeatedly visiting all sites (for example by raster scanning) we always replace one pixel with a value generated from the local characteristic of Gibbs distribution:

$$p(Y^t) = p(Y_r^t \,|\, Y_s^{t-1}\ \forall s \neq r)\, p(Y_s^{t-1}\ \forall s \neq r)\ \ , \tag{1.5}$$

where

$$p(Y_r^t \,|\, Y_s^{t-1}\ \forall s \neq r) = \frac{\exp\{-Q_r^t(Y)\}}{\tilde{Z}}\ \ . \tag{1.6}$$

Convergence of the algorithm (usually slower than the Metropolis algorithm convergence rate) is assured by the relaxation theorem [40]. The Gibbs sampler can be used to find a minimal energy state as stated in the annealing theorem [40].

For highly correlated components of a RF the Gibbs sampler convergence can be very slow. If however such a correlated variables are blocked together and drawn from a multivariate conditional distribution the convergence speed can be significantly improved.

1.1.2.3 Simulated Annealing

Simulated annealing iteratively samples from conditional distribution of each variable, while a control parameter, the temperature, is varied according a special schedule from high to low values. At low temperature values the algorithm samples from the most probable configurations. Let us introduce a new parameter T (temperature) into a GRF

$$p_T(Y) = \frac{1}{Z_T}\exp\{-\frac{1}{T}Q(Y)\}\ \ . \tag{1.7}$$

We get so called Boltzmann distribution, frequently used in statistical physics. It is easy to show that

$$\lim_{T\to\infty} p_T(Y) = \frac{1}{|\Omega|} \qquad \forall Y \in \Omega \tag{1.8}$$

where Ω is the set of all possible configurations of Y, and if

$$\Omega_{\min} = \{Y^g \in \Omega : \ Q(Y^g) \leq Q(Y) \ \ \forall Y \in \Omega\}$$

then

$$\lim_{T\to 0^+} p_T(Y) = \begin{cases} \frac{1}{|\Omega_{\min}|} & \text{for } Y \in \Omega_{\min} \\ 0 & \text{otherwise} \end{cases} . \tag{1.9}$$

Hence the probability mass function of GRF becomes uniform over all states as $T \to \infty$ and uniform over all global minima (Y^g) of its energy function as $T \to 0^+$, respectively. Global minima are also called the ground states of Y.

The simulated annealing algorithm is as follows:

1. Select an initial temperature T_0 and randomly chose Y^0.
2. At step k perturb Y^k, i.e., $Y^{k+1} = Y^k + \triangle Y$ and compute $\triangle Q = Q(Y^{k+1}) - Q(Y^k)$.
3. If $\triangle Q < 0$, accept the change. If $\triangle Q > 0$, accept the change only with probability $p(\triangle Y) = \exp\{-\frac{\triangle Q}{T_k}\}$.
4. If there is a considerable drop in energy, or enough iterations, lower the temperature $T_i = T_{i-1} - \triangle T$.
5. If energy becomes stable and temperature is very low, stop; otherwise go to the step 2.

The theorems [61, 17] state conditions for the simulated annealing algorithm convergence.

The simulated annealing differs from the Metropolis algorithm or the Gibbs sampler in using samples at time k generated from $P_{T_k}(Y)$ instead from $P(Y)$ $(T_k = 1 \ \ \forall k)$. The initial state to the simulated annealing is often of critical importance [13]. If there is not available any better initial state this state is assigned at random. Instead of visiting all sites as in the standard simulated annealing, the modified simulated annealing [121] visits only so called unstable sites, i.e., sites with the criterion function value smaller than some threshold $(Y_r^* < T)$. The second option for a site r to become stable is if $Y_r = Y_s \ \forall s \in I_r$, i.e., if the site r has the same value as all sites in its neighbourhood I_r. If the number of unstable sites decreases at each iteration, the amount of computation for the simulated annealing process can be significantly reduced [121].

Simulated annealing may converge very slowly, hence several modifications of the simulated annealing were developed to get faster algorithms. Instead of standard cooling $T_k = \frac{C}{\ln k}$, it is possible to use faster cooling $T_k = k$ or $T_k = \alpha^k$ $(\alpha = 1.01, \alpha = 1.05)$ however the convergence is not guaranteed any more. Alternatively it is possible to use fast visiting schemes (e.g., chequer-board) updating sets of sites or synchronous updating.

There is also a deterministic analogy to the simulated annealing called continuation methods [88]. These methods also involves the variation of a control parameter during the iterations which generates a sequence of cost functions. The original cost function is increasingly, closely approximated and is asymptotically reached at the final values of the control parameter.

The simulated annealing algorithm can be implemented also in the parallel version for its convergence theorem see reference [93].

1.1.3 Deterministic Algorithms

The stochastic algorithms guarantee to reach an optimal solution. Unfortunately while their asymptotic properties are specified in corresponding theorems, very little is known about their convergence rate and to reach the global optimum is usually expensive. Deterministic algorithms like the Highest Confidence First (HCF) [93], Maximum Marginal Probability method (MMP) [28, 83], the Lagrange-Hopfield method (LH) [76], or the Iterated Conditional Modes algorithm (ICM) [10] on the other hand are efficient but they stop at a local optimum. These algorithms can be used as a computationally inexpensive approximation of the stochastic algorithms.

1.1.4 URLs

www.mrc-bsu.cam.ac.uk/bugs/

1.2 Particle Filtering

Authors: Ercan Kuruoglu, ISTI, CNR, Pisa, Italy, and introduction by Simon Wilson, Trinity College Dublin Ireland

Particle filtering is particularly suited to multimedia applications where the data arrive as a sequence i.e. audio and video. It is closely related to the idea of importance sampling. Good reference papers for its application to Bayesian inference with multimedia data are [27] and [26]. These papers contain extensive bibliographies. A recent tutorial on the approach for tracking is [6]; this paper also contains an extensive list of references. The following are also references for the use of particle filtering in multimedia applications [44, 45, 111, 112, 34, 110, 43].

The groups within MUSCLE that are conducting research with particle filters are: Bilkent University (Turkey), ISTI-CNIR (Italy) and the University of Cambridge (UK). They are being applied to problems in source separation and object tracking.

1.2.1 The State Space Formulation

Filtering is the problem of estimating the hidden variables (called *states*) of a system, as a set of observations becomes available on-line: the introduction of a *state space* formulation is a fundamental step, because it allows to deal with non-stationarity, as it will be shown later.

In many real-world data analysis applications, prior knowledge about the unknown quantities to be estimated is available, and this information can be exploited to formulate Bayesian models: prior distributions for the unknown quantities and likelihood functions that relate these quantities to the observations. Then, all inference on the unknown quantities is based on the posterior distribution obtained from Bayes' theorem.

It is possible to express the model in terms of a *state equation* and an *observation equation*:

$$\alpha_t = f_t(\alpha_{t-1}, \mathbf{v}_t);$$
$$\mathbf{y}_t = h_t(\alpha_t, \mathbf{w}_t).$$

The state equation evaluates the state sequence: α_t is the state at current step t, f_t is a possibly nonlinear function, α_{t-1} is the state at the previous step, and $\mathbf{v}_t$ is called dynamic noise process. The observation equation is characterized by a nonlinear function h_t, and both the current state α_t and the observation noise realisation $\mathbf{w}_t$ at time step t are taken into account to generate the observation $\mathbf{y}_t$.

The *Kalman filter* (KF) is an extension of the Wiener filter, and it was presented by R. E. Kalman in 1961 [30]: this filter derives an exact analytical expression to compute the evolving sequence of the posterior distributions, when the data are modelled by a linear Gaussian state-space model. The obtained posterior density at every time step is Gaussian, hence parametrized by a mean and a covariance.

The best known algorithm that allows a non-Gaussian and nonlinear model is the *Extended Kalman filter* (EKF) [21], based upon the principle of linearising the measurements and evolution models using Taylor series expansions. Unfortunately, this procedure may lead to poor representations of both the non-linear functions and the probability distributions of interest, so the filter can diverge.

The more recent *Unscented Kalman Filter* (UKF) is founded on the intuition that it is better to approximate a Gaussian distribution, than approximating arbitrary non-linear functions [72]. Also this approach has, however, a limitation, that is it does not apply to general non-Gaussian distributions.

A new technique to solve the general filtering problem is introduced in this chapter: this approach, named *particle filtering*, uses sequential Monte Carlo methods, and it was introduced for the first time in automatic control by Handschin and Mayne [29] at the end of the 60's, but it has been overlooked until the early 90's because of the low computational power available. The renewed interest in these methods brought to success in tracking problems (see [97] for a general review), and very recently it has been applied also to perform source separation ([1], [15], [81]). Sequential Monte-Carlo particle filters are able to solve time or space varying mixing problems, and allow for a complete representation of the posterior distribution of the states, so that any statistical estimates (mean, variance, and so on...) can be computed.

1.2.2 The Bayesian Approach

This section is a brief overview of the Bayesian approach, which is the basis of Particle Filtering. Given a set of observations $\mathbf{y}$ and the set of unknown sources α, we consider the *posterior* distribution

$$p(\alpha|\mathbf{y}) = \frac{p(\mathbf{y}|\alpha)p(\alpha)}{p(\mathbf{y})}$$

where

$$p(\mathbf{y}) = \int p(\mathbf{y}|\alpha)p(\alpha)d\alpha$$

and where $p(\mathbf{y}|\alpha)$ denotes the *likelihood* and $p(\alpha)$ denotes the *prior* distribution. In order to keep the the same notation used in literature [3], we use α_t to denote both the random variable and its realisation. Consequently, we express continuous probability distributions using $p(d\alpha_t)$ instead of $\Pr(\alpha_t \in d\alpha_t)$, and discrete distributions using $p(\alpha_t)$ instead of $\Pr(\alpha_t = \alpha_t)$.

Given the posterior distribution, optimum estimators can be obtained, most notably the *Minimum Mean Squared Error* (MMSE) and the *Maximum A Posteriori* (MAP) estimates of α:

$$\hat{\alpha}_{MMSE} = \int \alpha p(\alpha|\mathbf{y})d\alpha;$$

$$\hat{\alpha}_{MAP} = \arg\max_{\alpha} p(\alpha|\mathbf{y}).$$

The aforementioned filters (KF, EKF, UKF) rely on various assumptions to ensure mathematical tractability. Unfortunately, real data sets are often very complex, typically high dimensional, nonlinear, nonstationary and non-Gaussian: except in some simple cases, the integration (MMSE) or the optimisation (MAP) of the posterior are not analitically tractable. Moreover, classical optimisation methods need good initialisations and are sensitive to local minima.

On-line simulation based *Sequential Monte Carlo* (SMC) methods are a set of simulation-based approaches which use variates from the posterior, and provide an attractive solution to compute the posterior distribution of interest: at each time step, the posterior distribution is approximated by a set of *particles* generated by an *importance distribution* $\pi(\alpha|\mathbf{y})$, chosen such that it is easy to sample, and whose support is assumed to include that of $p(\alpha|\mathbf{y})$, as shown in the next section.

1.2.3 Monte Carlo Particle Filters

1.2.3.1 Problem Statement

As stated before, we usually cannot obtain an analytic expression for the posterior distribution: this is the reason why we have to resort to stochastic simulation. The unobserved signal (hidden state) α_t is modelled as a Markov process of initial distribution $p(\alpha_0)$ and transition equation $p(\alpha_t|\alpha_{t-1})$:

$$\begin{aligned} p(\alpha_0) \qquad & \text{for } t = 0, \\ p(\alpha_t|\alpha_{t-1}) \qquad & \text{for } t = 1,2,3,\cdots. \end{aligned}$$

We denote by $\alpha_{0:t} \stackrel{\triangle}{=} \{\alpha_0,\cdots,\alpha_t\}$ and $\mathbf{y}_{1:t} \stackrel{\triangle}{=} \{\mathbf{y}_1,\cdots,\mathbf{y}_t\}$ the signals and the observations respectively, up to step t.

Our objective is to estimate recursively in time the posterior distribution $p(\alpha_{0:t}|\mathbf{y}_{1:t})$, its associated features (including the marginal distribution $p(\alpha_t|\mathbf{y}_{1:t})$, known as the *filtering distribution*), and the expectations

$$I(f_t) = E_{p(\alpha_{0:t}|\mathbf{y}_{1:t})}\{f_t(\alpha_{0:t})\} \stackrel{\triangle}{=} \int f_t(\alpha_{0:t})p(\alpha_{0:t}|\mathbf{y}_{1:t})\mathrm{d}\alpha_{0:t}$$

for some function of interest, like the mean of the sources, or their covariance.

At any time t, the posterior distribution is given by Bayes' theorem:

$$p(\alpha_{0:t}|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_{1:t}|\alpha_{0:t})p(\alpha_{0:t})}{\int p(\mathbf{y}_{1:t}|\alpha_{0:t})p(\alpha_{0:t})\mathrm{d}\alpha_{0:t}}.$$

A recursive formula for this joint distribution can be obtained as follows:

$$p(\alpha_{0:t+1}|\mathbf{y}_{1:t+1}) = p(\alpha_{0:t}|\mathbf{y}_{1:t})\frac{p(\mathbf{y}_{t+1}|\alpha_{t+1})p(\alpha_{t+1}|\alpha_t)}{p(\mathbf{y}_{t+1}|\mathbf{y}_{1:t})}.$$

The marginal distribution $p(\alpha_t|\mathbf{y}_{1:t})$ also satisfies the following recursive equations (prediction and update respectively):

$$\begin{aligned} p(\alpha_t|\mathbf{y}_{1:t-1}) &= \int p(\alpha_t|\alpha_{t-1})p(\alpha_{t-1}|\mathbf{y}_{1:t-1})\mathrm{d}\alpha_{t-1}; \\ p(\alpha_t|\mathbf{y}_{1:t}) &= \frac{p(\mathbf{y}_t|\alpha_t)p(\alpha_t|\mathbf{y}_{1:t-1})}{\int p(\mathbf{y}_t|\alpha_t)p(\alpha_t|\mathbf{y}_{1:t-1})\mathrm{d}\alpha_t}. \end{aligned}$$

Monte Carlo integration methods have the great advantage of not being subject to any linearity or Gaussianity constraints on the model, and they also have appealing convergence properties. The basic idea is that a large number of samples drawn from the required posterior distribution is sufficient to approximate the posterior distribution itself, and to approximate the integrals appearing in the "prediction and update" equations mentioned before.

1.2.3.2 Importance Sampling

Assume that $N >> 1$ random samples $\{\alpha_{0:t}^{(i)};\ i = 1, \cdots, N\}$, called *particles* (hence the term *particle filters*), have been generated from the posterior $p(\alpha_{0:t}|\mathbf{y}_{1:t})$: a Monte Carlo approximation of this function is thus given by:

$$p_N(d\alpha_{0:t}|\mathbf{y}_{1:t}) = \frac{1}{N}\sum_{i=1}^{N}\delta_{\alpha_{0:t}^{(i)}}(d\alpha_{0:t}),$$

where $\delta_{\alpha_{0:t}^{(i)}}(d\alpha_{0:t})$ denotes the delta-Dirac mass located in $\alpha_{0:t}^{(i)}$. The following estimate of the function of interest $I(f_t)$ can be obtained straightforwardly by:

$$I_N(f_t) = \int f_t(\alpha_{0:t})p_N(d\alpha_{0:t}|\mathbf{y}_{1:t}) = \sum_{i=1}^{N} f_t\left(\alpha_{0:t}^{(i)}\right).$$

Unfortunately, it is usually impossible to sample efficiently from the posterior distribution at any step t, since it is, in general, multivariate, non-standard, and only known up to a proportionality constant. A classical solution consists of using the *importance sampling* method [70], which introduces an arbitrary *importance function* (also referred to as the *proposal distribution* or the

importance sampling distribution) $\pi(\alpha_{0:t}|\mathbf{y}_{1:t})$. Provided that the support of $\pi(\alpha_{0:t}|\mathbf{y}_{1:t})$ includes the support of $p(\alpha_{0:t}|\mathbf{y}_{1:t})$, we get the identity

$$I(f_t) = \frac{\int f_t(\alpha_{0:t}) w(\alpha_{0:t}) \pi(\alpha_{0:t}|\mathbf{y}_{1:t}) d\alpha_{0:t}}{\int w(\alpha_{0:t}) \pi(\alpha_{0:t}|\mathbf{y}_{1:t}) d\alpha_{0:t}},$$

where $w(\alpha_{0:t})$ is known as the *importance weight*:

$$w(\alpha_{0:t}) = \frac{p(\alpha_{0:t}|\mathbf{y}_{1:t})}{\pi(\alpha_{0:t}|\mathbf{y}_{1:t})}.$$

Consequently, it is possible to obtain a Monte Carlo extimate of $I(f_t)$ using N particles $\{\alpha_{0:t}^{(i)};\ i = 1, \cdots, N\}$ sampled from $\pi(\alpha_{0:t}|\mathbf{y}_{1:t})$:

$$\bar{I}_N(f_t) = \frac{\frac{1}{N}\sum_{i=1}^{N} f_t\left(\alpha_{0:t}^{(i)}\right) w\left(\alpha_{0:t}^{(i)}\right)}{\frac{1}{N}\sum_{j=1}^{N} w\left(\alpha_{0:t}^{(i)}\right)} = \sum_{i=1}^{N} f_t\left(\alpha_{0:t}^{(i)}\right) \tilde{w}_t^{(i)},$$

where the *normalised importance weights* $\tilde{w}_t^{(i)}$ are given by:

$$\tilde{w}_t^{(i)} = \frac{w\left(\alpha_{0:t}^{(i)}\right)}{\sum_{j=1}^{N} w\left(\alpha_{0:t}^{(i)}\right)}.$$

This integration method can be interpreted as a sampling method, where the posterior distribution is approximated by:

$$\bar{p}_N(d\alpha_{0:t}|\mathbf{y}_{1:t}) = \sum_{i=1}^{N} \tilde{w}_t^{(i)} \delta_{\alpha_{0:t}^{(i)}}(d\alpha_{0:t}).$$

It is clear that importance sampling needs all the data set $\mathbf{y}_{1:t}$ before estimating $p(\alpha_{0:t}|\mathbf{y}_{1:t})$. That makes this method not adequate for recursive estimation, because, whenever new data $\mathbf{y}_{t+1}$ become available, the importance weights over the entire state sequence need to be recomputed. As the complexity of this operation increases with long sequences, recursive techniques for overcoming this problem have been studied.

1.2.3.3 Sequential Importance Sampling

Our aim is to estimate the posterior density function $p(\alpha_{0:t}|\mathbf{y}_{1:t})$ without modifying the past simulated trajectories $\{\alpha_{0:t-1}^{(i)};\ i = 1, \cdots, N\}$. This means that the importance function $\pi(\alpha_{0:t}|\mathbf{y}_{1:t})$ has to admit $\pi(\alpha_{0:t-1}|\mathbf{y}_{1:t-1})$ as marginal distribution, which happens when the importance function is restricted to be of the general form:

$$\begin{aligned}\pi(\alpha_{0:t}|\mathbf{y}_{1:t}) &= \pi(\alpha_{0:t-1}|\mathbf{y}_{1:t-1})\pi(\alpha_t|\alpha_{0:t-1},\mathbf{y}_{1:t}) \\ &= \pi(\alpha_0)\prod_{k=1}^{t}\pi(\alpha_k|\alpha_{0:k-1},\mathbf{y}_{1:k}).\end{aligned}$$

This importance distribution allows the importance weights to be avaluated recursively:

$$\tilde{w}_t \propto \tilde{w}_{t-1}\frac{p(\mathbf{y}_t|\alpha_t^{(i)})p(\alpha_t^{(i)}|\alpha_{t-1}^{(i)})}{\pi(\alpha_t^{(i)}|\alpha_{0:t-1}^{(i)},\mathbf{y}_{1:t})}.$$

The only constraints on the selection of the importance function are those that have been mentioned so far. It follows that a wide choice for $\pi(\alpha_{0:t}|\mathbf{y}_{1:t})$ is allowed.

1.2.3.4 Selection

Unfortunately, for the importance distributions of the form specified before, a degeneracy phenomenon may occur: after a few iterations, all but one of the normalised importance weights are very close to zero. This happens because the variance of the importance weights can only increase (stochastically) over time, as demonstrated in [1]. As a result of the degeneracy phenomenon, it is indispensable to include one more step (called *selection*) in the particle filter algorithm. The purpose of this procedure is to discard the particles with low importance weights, and to multiply the particles having high importance weights: the idea is that of associating with each particle (say $\tilde{\alpha}_{0:t}^{(i)} : i = 1,\cdots,N$) a number of "children" $N_t^{(i)}$, such that $\sum_{i=1}^{N} N_t^{(i)} = N$, in order to obtain N new particles $\{\alpha_{0:t}^{(i)} : i = 1,\cdots,N\}$. For each particle, if $N_t^{(j)} = 0$, then $\tilde{\alpha}_{0:t}^{(j)}$ is discarded, otherwise it has $N_t^{(j)}$ children at step $t+1$. More formally, the weighted empirical distribution $\bar{p}_N(d\alpha_{0:t}|\mathbf{y}_{1:t}) = \sum_{i=1}^{N}\tilde{w}_t^{(i)}\delta_{\alpha_{0:t}^{(i)}}(d\alpha_{0:t})$ is replaced by the unweighted measure

$$p_N(d\alpha_{0:t}|\mathbf{y}_{1:t}) = \frac{1}{N}\sum_{i=1}^{N} N_t^{(i)}\delta_{\alpha_{0:t}^{(i)}}(d\alpha_{0:t}),$$

where $N_t^{(i)}$ is the number of offsprings associated to the particle $\alpha_{0:t}^{(i)}$. After the selection step, all the importance weights are divided by N; since they do not depend on any past values of the normalised importance weights, all information regarding the old importance weights is discarded.

There is a variety of selection schemes, including *Residual Resampling* [37], *Stratified Sampling* [37], and *Multinomial Sampling*, also known as SIR (*Sampling Importance Resampling*) [71]: all of them can be implemented in a number of operations which is proportional to the number of particles N, and their aim is to provide the coefficients $N_t^{(i)}$ such that $p_N(d\alpha_{0:t}|\mathbf{y}_{1:t})$ is close to $\bar{p}_N(d\alpha_{0:t}|\mathbf{y}_{1:t})$, in the sense that, for any function f_t,

$$\int f_t(\alpha_{0:t})p_N(d\alpha_{0:t}|\mathbf{y}_{1:t}) \approx \int f_t(\alpha_{0:t})\bar{p}_N(d\alpha_{0:t}|\mathbf{y}_{1:t}),$$

according to different criteria.

As we can see in figure 1.1, the filtering density is approximated by an adaptive stochastic grid. This is a direct consequence of the Monte Carlo approach, where the particles interact with each other randomly in time, and either give birth to children, or die out, depending on the magnitude of their weights.

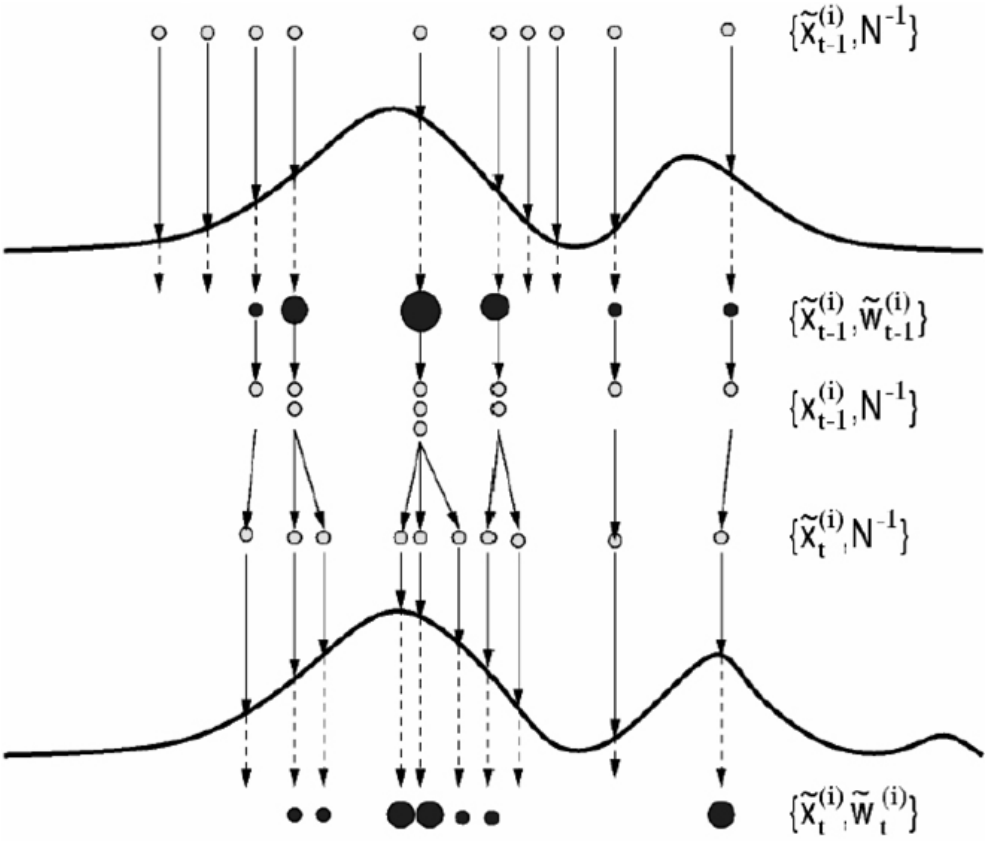

Figure 1.1: Particle Filtering - The adaptive stochastic grid and the selection step, when the density approximation is propagated from step $t-1$ to step t ([25]).

1.2.3.5 The Algorithm

Now it is possible to describe in outline the general particle filter algorithm: as stated before, the recursive Monte Carlo filter operates on N particles $\{\alpha_{0:t}^{(i)} : i = 1, \cdots, N\}$, given at step $t-1$, and distributed approximately according to $p(\alpha_{0:t-1}|\mathbf{y}_{1:t-1})$. The algorithm has a structure that can be divided into two main blocks, and it proceeds as follows at step t:

- Sequential Importance Sampling Step:
  - For $i = 1, \cdots, N$, sample

$$\tilde{\alpha}_t^{(i)} \sim \pi(\alpha_t|\alpha_{0:t-1}^{(i)}, \mathbf{y}_{1:t})$$

    and set

$$\tilde{\alpha}_{0:t}^{(i)} = (\alpha_{0:t-1}^{(i)}, \tilde{\alpha}_t^{(i)});$$

– For $i = 1, \cdots, N$, evaluate the importance weights, up to a normalising constant:

$$w_t^{(i)} \propto \frac{p(\mathbf{y}_t|\tilde{\alpha}_{0:t}^{(i)}, \mathbf{y}_{1:t-1}) p(\tilde{\alpha}_t^{(i)}|\tilde{\alpha}_{t-1}^{(i)})}{\pi(\tilde{\alpha}_t^{(i)}|\tilde{\alpha}_{0:t-1}^{(i)}, \mathbf{y}_{1:t})};$$

– For $i = 1, \cdots, N$, normalise the importance weights:

$$\tilde{w}_t^{(i)} = \frac{w_t^{(i)}}{\sum_{j=1}^{N} w_t^{(j)}}.$$

- Selection Step:

  – Discard / multiply particles $\{\tilde{\alpha}_{0:t}^{(i)} : i = 1, \cdots, N\}$ with low / high normalised importance weights to obtain N particles:

$$\{\alpha_{0:t}^{(i)} : i = 1, \cdots, N\}.$$

Although the computational complexity at each t step is proportional to N, the above algorithm is parallelisable, so that efficient implementation may be achieved by using parallel processors. It is worth mentioning that there is an unlimited number of choices for the implementation of this algorithm, as we have a lot of freedom both in the choice of the importance distribution and of the selection schemes.

1.2.4 Implementation

In this section, the particle filter is implemented considering the source separation problem: our aim is to operate a Bayesian source separation of the different independent components, given a set of observation, providing MMSE estimators of each source through the knowledge of the approximations of the posterior distributions computed by the particle filter. We allow the sources to have non-Gaussian distributions; the mixing-system is assumed to be non-stationary, and we also take space-varying noise into account.

1.2.4.1 Model Specification

Before illustrating the implementation of the particle filter algorithm, we introduce the model we will follow ([1]).

We consider instantaneous mixing of independent sources, each one modelled as a mixture of a known number of Gaussian components: this model is very flexible, generic and was adopted by various researchers in the literature (e.g. [31, 60, 32]).

The mixing-system is assumed to be non-stationary, and we also take space-varying noise into account. Let n be the number of sensors: each of the n observations $\mathbf{y}$ will be represented as a row vector of t elements, where t is the number of the pixels we are taking into account. The number of independent sources α is m (of course, each one is represented as a row vector of t elements).

The general model for the observations is thus, at time t:

$$\mathbf{y}_{1:n,t} = \mathbf{H}_t \alpha_{1:m,t} + \mathbf{w}_{1:n,t}$$

where $\mathbf{y}_{1:n,t}$, $\alpha_{1:m,t}$ and $\mathbf{w}_{1:n,t}$ are column vectors, representing the n observations, the m sources and the n additive noise samples at time t respectively. The $n \times m$ real valued mixing matrix $\mathbf{H}_t$ varies in t, and we can re-parametrise it into a vector $\mathbf{h}_t = vec\{\mathbf{H}_t\}$ so that $[\mathbf{h}_t]_{n(j-1)+1} = h_{i,j,t}$.

Now we are able to express the model in terms of state equation and observation equation, in this way:

$$\mathbf{h}_t = \mathbf{A}_t \mathbf{h}_{t-1} + \mathbf{v}_t$$

$$\mathbf{y}_{1:n,t} = \mathbf{C}_t \mathbf{h}_t + \mathbf{w}_{1:n,t}$$

where $\mathbf{A}_t$ and $\mathbf{C}_t$ are $(nm \times nm)$ and $(n \times nm)$ real valued matrices respectively, and $\mathbf{w}_t$ is a $(n \times 1)$ real vector. $\mathbf{C}_t$ can be expressed in terms of the source signal vector $\alpha_{1:m,t}$, as $\mathbf{C}_t = \alpha_{1:m,t}^T \bigotimes \mathbf{I}_n$. In absence of further prior information, we assume $\mathbf{A}_t = \mathbf{I}_{nm}$, and of course $\mathbf{C}_t$ is unknown, as it consists of the source signals to be estimated. The distributions of the dynamic noise $\mathbf{v}_t$ and the observation noise $\mathbf{w}_t$ are assumed to be i.i.d. and mutually independent: $\mathbf{v}_t \sim \mathcal{N}(0, \sigma_v)$ and $\mathbf{w}_t \sim \mathcal{N}(0, \sigma_w)$, with obvious notation. The introduction of the state equation allows to deal with non-stationary mixing matrices, as the coefficients of $\mathbf{h}$ can be updated at every step.

In this formulation there is a scaling ambiguity, as we can multiply $\mathbf{H}_t$ by a non-zero constant c and divide the sources $\alpha_{1:m,t}$ by c and obtain the same observations: in order to solve this ambiguity, we constrain $\mathbf{H}_t$ to have constant unity diagonal for $\mathbf{H}_t$ square ($m = n$), or set the diagonal of sub-matrix $\mathbf{H}^a_{m \times m}$ to unity if $n > m$:

$$\mathbf{H}_t = \begin{bmatrix} \mathbf{H}^a_{m \times m} \\ \mathbf{H}^b_{(n-m) \times m} \end{bmatrix}$$

1.2.4.2 Model of the Sources

As the m sources are statistically independent of one another:

$$p(\alpha_{1:m,t}) = \prod_{i=1}^{m} p(\alpha_{i,t}).$$

Moreover, we can model each source by a finite mixture of Gaussians, so:

$$p(\alpha_{i,t} | \mu_{i,j,t}, \sigma^2_{i,j,t}) = \sum_{j=1}^{q_i} \rho_{i,j} \mathcal{N}(\alpha_{i,t}; \mu_{i,j,t}, \sigma^2_{i,j,t}); \sum_{j=1}^{q_i} \rho_{i,j} = 1,$$

where $\rho_{i,j}$ is the weight of the j^{th} Gaussian component of the i^{th} source, and q_i is the number of Gaussian components for the i^{th} source.

Now we will consider a hidden variable z_i which takes on a finite set of values $Z_i = \{1, \cdots, q_i\}$, so that we can denote the distribution of $\alpha_{i,t}$ as if at time t only the j^{th} Gaussian component is active, with probability $\rho_{i,j}$:

$$p(\alpha_{i,t} | z_{i,t} = j) = \mathcal{N}(\alpha_{i,t}; \mu_{i,j}, \sigma^2_{i,j})$$

At time t let $\mathbf{z}_{1:m,t} \triangleq [z_{1,t} \cdots z_{m,t}]^T$. Given that the sources are statistically independent of one another, $\alpha_{1:m,t}$ have distributions:

$$p(\alpha_{1:m,t}|\mathbf{z}_{1:m,t}) = \mathcal{N}(\alpha_{1:m,t}; \mu(\mathbf{z}_{1:m,t}), \Gamma(\mathbf{z}_{1:m,t})),$$

where

$$\mu(\mathbf{z}_{1:m,t}) = [\mu_{1,z_{1,t}}, \cdots, \mu_{m,z_{m,t}}]^T$$

and

$$\Gamma(\mathbf{z}_{1:m,t}) = diag\{\sigma^2_{1,z_{1,t}}, \cdots, \sigma^2_{m,z_{m,t}}\}.$$

It is possible to describe the discrete probability distribution of $\mathbf{z}_{1:m,t}$ using the i.i.d. model: in this case, the indicators of the states $z_{i,t}$ have identical and independent distributions. If we want to introduce temporal correlation beween the samples of a particular source, we have to consider the first-order Markov model case, where the vector of the states evolves as a homogeneous Markov chain for $t > 1$:

$$p(\mathbf{z}_{1:m,t} = \mathbf{z}_l|\mathbf{z}_{1:m,t-1} = \mathbf{z}_j) = \prod_{i=1}^{m} p(z_{i,t} = [\mathbf{z}_l]_i|z_{i,t-1} = [\mathbf{z}_j]_i) = \prod_{i=1}^{m} \pi^{(i)}_{j,l},$$

where $\pi^{(i)}_{j,l}$ is an element of the $q_i \times q_i$ real valued *transition matrix* for the states of the i^{th} source, denoted by $\pi^{(i)}$. The state transition can be thus parametrised by a set of m transition matrices $\pi^{(i)}$, $i \in \{1, \cdots, m\}$.

Given the observations $\mathbf{y}_t$ (assuming that the number of sources m, the number of Gaussian components q_i for the i^{th} source, and the number of sensors n are known), we would like to estimate all the following unknown parameters of interest, grouped together:

$$\theta_{0,t} = [\alpha_{1:m,0:t}, \mathbf{z}_{1:m,0:t}, \{\mu_{i,j,0:t}\}, \{\sigma^2_{i,j,0:t}\}, \{\pi^{(i)}_{0:t}\}, \{\sigma^2_{\mathbf{w}_{1:n,0:t}}\}],$$

where we recall that $\alpha_{1:m,0:t}$ are the sources, $\mathbf{z}_{1:m,0:t}$ is the matrix of the indicator variables which determines which Gaussian component is active at a particular time for each source, $\{\mu_{i,j,0:t}\}$ and $\{\sigma^2_{i,j,0:t}\}$ are the means and the variances of the j^{th} Gaussian component of the i^{th} source, $\{\pi^{(i)}_{0:t}\}$ is the transition matrix for the evolution of $z_{i,0:t}$, and $\{\sigma_{\mathbf{w}_{1:n,0:t}}\}$ represents the standard deviation of the observation noise.

1.2.4.3 Rao-Blackwellisation

In our case, referring to the model of the sources defined before, we want to estimate the wide set of unknown parameters grouped together in

$$\theta_{0,t} = [\alpha_{1:m,0:t}, \mathbf{z}_{1:m,0:t}, \{\mu_{i,j,0:t}\}, \{\sigma^2_{i,j,0:t}\}, \{\pi^{(i)}_{0:t}\}, \{\sigma^2_{\mathbf{w}_{1:n,0:t}}\}],$$

and we have also to consider that the mixing matrix is both space-varying and not precisely known. The particles we should deal with will be thus $\{(\mathbf{h}^{(i)}_{0:t}, \theta^{(i)}_{0:t}) : i = 1, \cdots, N\}$, generated according to $p(\mathbf{h}_{0:t}, \theta_{0:t}|\mathbf{y}_{1:t})$. An empirical estimate of this distribution is given by

$$\bar{p_N}(d\mathbf{h}_{0:t}, \theta_{0:t}|\mathbf{y}_{1:t}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{h_{0:t}{}^{(i)}}, \delta_{\theta_{0:t}{}^{(i)}}(d\mathbf{h}_{0:t}, d\theta_{0:t}),$$

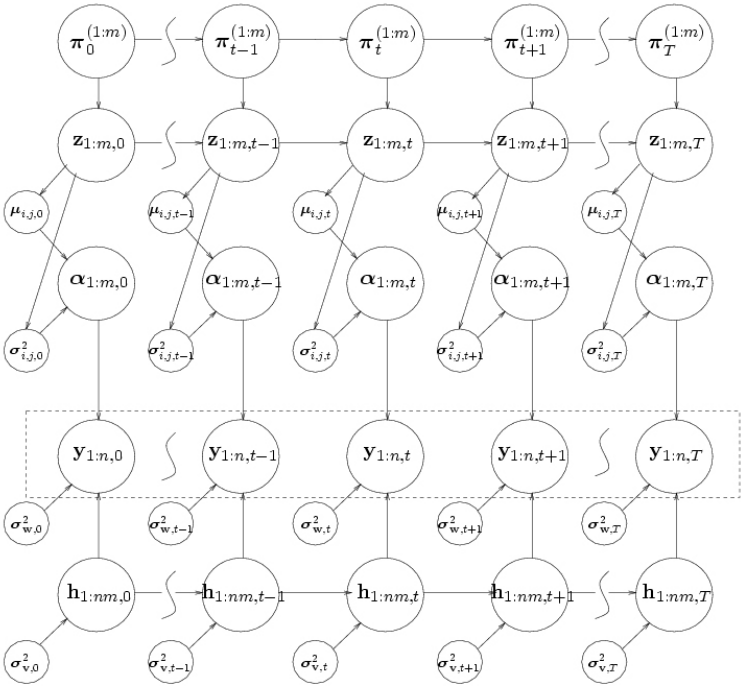

Figure 1.2: Particle Filtering - Graphical representation of the model ([1]).

and, as a corollary, an estimate of $p(\mathbf{h}_t, \theta_t | \mathbf{y}_{1:t})$ is

$$\bar{p_N}(d\mathbf{h}_t, \theta_t | \mathbf{y}_{1:t}) = \frac{1}{N} \sum_{i=1}^{N} \delta_{h_t{}^{(i)}}, \delta_{\theta_t{}^{(i)}}(d\mathbf{h}_t, \theta_t).$$

It is possible to reduce the problem of estimating $p(\mathbf{h}_t, \theta_{0:t} | \mathbf{y}_{1:t})$ to a simpler one of sampling from $p(\theta_{0:t} | \mathbf{y}_{1:t})$. In fact,

$$p(\mathbf{h}_t, \theta_{0:t} | \mathbf{y}_{1:t}) = p(\mathbf{h}_t | \theta_{0:t}, \mathbf{y}_{1:t}) p(\theta_{0:t} | \mathbf{y}_{1:t}).$$

Given an approximation of $p(\theta_{0:t} | \mathbf{y}_{1:t})$, an approximation of $p(\mathbf{h}_t | \theta_{0:t}, \mathbf{y}_{1:t})$ may straightforwardly be obtained considering the following state space model for each particle:

$$\begin{aligned} \mathbf{h}_t^{(i)} &= \mathbf{A}_t \mathbf{h}_{t-1}^{(i)} + \mathbf{v}_t^{(i)}; \\ \mathbf{y}_{1:n,t} &= \mathbf{C}_t \mathbf{h}_t^{(i)} + \mathbf{w}_{1:n,t}^{(i)}; \end{aligned}$$

where we recall that $\mathbf{C}_t$ can be expressed in terms of the source signal vector $\alpha_{1:m,t}$, as $\mathbf{C}_t = \alpha_{1:m,t}^T \bigotimes \mathbf{I}_n$. The posterior distribution of the state $\mathbf{h}_t$ given the observations $\mathbf{y}_{1:n,t}$ can be recursively estimated in closed form using the Kalman filter [30].

This technique, called *Rao-Blackwellisation* [36], leads to better results, as we are reducing the size of the parameter set to be estimated by marginalising out the mixing coefficients $\mathbf{h}_t$ using the Kalman filter, so that the only distribution we have to estimate by particle filtering is $p(\theta_{0:t} | \mathbf{y}_{1:t})$.

1.2.4.4 Prior Distribution as Importance Function

Referring to the approach defined before, the samples used to estimate the posterior density functions of the parameters of interest have to be drawn from an importance distribution of the general form

$$\pi(\theta_{0:t}|\mathbf{y}_{1:t}) = \pi(\theta_{0:t-1}|\mathbf{y}_{1:t-1})\pi(\theta_t|\theta_{0:t-1},\mathbf{y}_{1:t})$$

The best strategy is to choose, at step t, the importance distribution that minimises the variance of the importance weights, given $\theta_{0:t-1}$ and $\mathbf{y}_{1:t}$. In [2] we find the proof that the importance distribution we are looking for is:

$$\pi(\theta_t|\theta_{0:t-1},\mathbf{y}_{1:t}) = p(\theta_t|\theta_{0:t-1},\mathbf{y}_{1:t}).$$

From Bayes' rule, the optimal importance distribution may be expressed as

$$p(\theta_t|\theta_{0:t-1},\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|\theta_{0:t},\mathbf{y}_{1:t-1})p(\theta_t|\theta_{t-1})}{p(\mathbf{y}_t|\theta_{0:t-1},\mathbf{y}_{1:t-1})},$$

being

$$p(\mathbf{y}_t|\theta_{0:t-1},\mathbf{y}_{1:t-1}) = \int p(\mathbf{y}_t|\theta_{0:t},\mathbf{y}_{1:t-1})p(\theta_t|\theta_{0:t-1})d\theta_t.$$

Unfortunately it is not easy to sample directly from the optimal importance distribution, and the above integral cannot be evaluated analitically, since $p(\mathbf{y}_t|\theta_{0:t},\mathbf{y}_{1:t-1})$ is a complex possibly non-linear function of θ_t. This is the reason why the following sub-optimal method will be employed throughout, taking the importance distribution at step t to be the *prior distribution*:

$$\pi(\theta_{0:t}|\mathbf{y}_{1:t}) = p(\theta_{0:t}) = p(\theta_0)\prod_{k=1}^{t} p(\theta_k|\theta_{0:k-1}).$$

In this case, the importance weights can be computed recursively by

$$\tilde{w}_t^{(i)} \propto \tilde{w}_{t-1}^{(i)} p(\mathbf{y}_t|\theta_t^{(i)})$$

whose evaluation requires only one step of the Kalman filter for each particle. Now it is convenient to factorize the prior importance function:

$$
\begin{aligned}
p(\theta_t|\theta_{t-1}) &= p(\alpha_{1:m,t}, \mathbf{z}_{1:m,t}, \pi_t, \{\mu_{i,j,t}\}, \{\sigma^2_{i,j,t}\}, \{\sigma^2_{\mathbf{w},K,t}\}| \\
&\quad \alpha_{1:m,t-1}, \mathbf{z}_{1:m,t-1}, \pi_{t-1}, \{\mu_{i,j,t-1}\}, \{\sigma^2_{i,j,t-1}\}, \{\sigma^2_{\mathbf{w},K,t-1}\}) \\
&= p(\alpha_{1:m,t}, \mathbf{z}_{1:m,t}, \pi_t, \{\mu_{i,j,t}\}, \{\sigma^2_{i,j,t}\}| \\
&\quad \alpha_{1:m,t-1}, \mathbf{z}_{1:m,t-1}, \pi_{t-1}, \{\mu_{i,j,t-1}\}, \{\sigma^2_{i,j,t-1}\}) \times \\
&\quad p(\{\sigma^2_{\mathbf{w},K,t}\}|\{\sigma^2_{\mathbf{w},K,t-1}\})
\end{aligned}
$$

If now we consider a new variable, $\tilde{\theta}_t$, which excludes the observation noise variance, we obtain

$$
\begin{aligned}
p(\tilde{\theta}_t|\tilde{\theta}_{t-1}) &= p(\alpha_{1:m,t}, \mathbf{z}_{1:m,t}, \pi_t, \{\mu_{i,j,t}\}, \{\sigma^2_{i,j,t}\}|\tilde{\theta}_{t-1}) \\
&= p(\alpha_{1:m,t}|\mathbf{z}_{1:m,t}, \{\mu_{i,j,t}\}, \{\sigma^2_{i,j,t}\}) \times \\
&\quad p(\{\mu_{i,j,t}\}|\{\mu_{i,j,t-1}\}, \mathbf{z}_{i,t}) \times \\
&\quad p(\{\sigma^2_{i,j,t}\}|\{\sigma^2_{i,j,t-1}\}, \mathbf{z}_{i,t}) \times \\
&\quad p(\mathbf{z}_{1:m,t}|\mathbf{z}_{1:m,t-1}, \pi_t) \times \\
&\quad p(\pi_t|\pi_{t-1}).
\end{aligned}
$$

This hierarchical structure allows us to obtain an approximation of the distribution of the sources exploiting the particles generated from the distributions of the other parameters, sampling subsequently from $p(\pi_t|\pi_{t-1})$, $p(\mathbf{z}_{1:m,t}|\mathbf{z}_{1:m,t-1}, \pi_t)$, $p(\{\sigma^2_{i,j,t}\}|\{\sigma^2_{i,j,t-1}\}, \mathbf{z}_{i,t})$, $p(\{\mu_{i,j,t}\}|\{\mu_{i,j,t-1}\}, \mathbf{z}_{i,t})$, and finally obtain the particles of the distribution $p(\alpha_{1:m,t}|\mathbf{z}_{1:m,t}, \{\mu_{i,j,t}\}, \{\sigma^2_{i,j,t}\})$.

1.2.5 Work in the literature

There are a few publications in the literature on the use of particle filters for separation. In particular, Everson and Roberts [81] have implemented a simplified version of particle filtering where they consider only the mixing matrix to be nonstationary. The examples they give consider the case when the mixing is piece-wise stationary with abrupt changes in their values at the end of stationary periods. They assume generalised Gaussian distributions for the sources. Their implementation is especially of interest for speech on mobile stations. Andrieu and Godsill [15] instead consider a parametric AR model for the sources (with the audio signals in mind) and provide a particle filtering scheme general enough to model convolutional mixing. They report simulations on synthetic sources. Their particle filtering use is mor general and they adopt an importance function which can be described as a hybrid one. It is not the prior distribution nor the posterior which is the optimal but somewhere in between. The insight into this choice is interesting. Ahmed et al. [1] instead adopt a Gaussian mixture model for the sources and persent a hierachical model which fully exploits Bayesian formulation. The description in this report closely followed their formulation. Th eproblem in this approach is the choice of the importance function (they chose the prior pdf). Recent work is by Costagli et al. who adopted this formulation for a practical source separation problem namely the separation of the independent components in astrophysical images [80].

1.3 Method of Mixtures

Author: J. Grim, UTIA Czech Republic

1.3.1 Finite Mixture Models

In the last decades the finite distribution mixtures became increasingly popular as a flexible tool to estimate unknown (possibly multimodal) probability density functions or discrete distributions in multidimensional spaces. The finite mixture models are applicable to pattern recognition [56, 55, 54], cluster analysis [59, 116], data mining [51, 52], texture modelling [53] and to many other problems which can be solved by estimating unknown probability distributions or densities [86]. Finite mixtures can also be used to design biologically interpretable neural networks [55, 54, 58, 63, 104].

The finite mixture models can be viewed as a reasonable compromise between the parametric and nonparametric approaches to estimating probability distributions. The standard parametric estimates are advantageous mainly because of a small number of parameters to be identified. Nevertheless, the assumption of a simple parametric form of the estimated distribution (e.g. normal) may be too restrictive. If the assumed parametric model is not adequate, the resulting solution may be poor.

On the other hand, the nonparametric estimation methods do not assume any specific type of the estimated probability distribution but they are computationally awkward especially in case of a large sample size. Typically, in case of the nonparametric kernel estimates we can guarantee the asymptotically unbiased and consistent density estimate at any continuity point of the estimated density, however, we have to store all available data vectors.

Let us recall that the kernel estimate with a normal kernel function can be viewed formally as a normal mixture with uniform weights. In this sense finite mixture models provide a convenient estimation method that occupies the full range between the classical parametric methods and the nonparametric kernel estimates. They have much of the flexibility of the nonparametric methods while keeping the advantage of a limited number of the parameters included.

1.3.2 EM Algorithm

The key point of finite mixture models is a widely applicable method to compute the maximum-likelihood estimates of mixtures by means of the iterative EM algorithm. The EM algorithm increases the maximized likelihood function in each iteration without any control parameters like step size. In the last two decades nearly all applications of finite mixture models make use of the EM algorithm. The EM algorithm can be applied in multidimensional spaces to continuous, discrete and mixed-type data and to different types of components. There are several monographs on estimating mixtures by means of EM algorithm (cf. [33], [107], [84], [85]), the most recent by McLachlan and Peel [86].

In case of finite mixtures the log-likelihood function is known to have local maxima nearly always. As the achieved local maximum is starting-point dependent, the quality of the estimates may be influenced by the chosen number of components and the initial parameter values. In view of these facts the literature on EM algorithm is more or less dominated by the problem

of a proper choice of the number of mixture components and by the optimization of the initial parameters.

The tendency to local maxima is more probable in case of a small number of components, small data sets and/or high-dimensional spaces. There is no generally accepted way to solve the problem of the local optimality. A frequently proposed idea is to repeat the computation from sufficiently many different starting points and to choose the highest local maximum as the best solution (cf. [86], [107]). Obvious disadvantage of such a method is the computational complexity. The problem of locally optimal parameter estimates can be avoided by using Bayes estimation approach. Unfortunately, some of the underlying steps have to be solved by means of approximation techniques. Mixture estimation in a Bayesian framework became feasible by using posterior simulation via the recently developed Markov chain Monte Carlo method (cf. [86, Chapter 4.]).

Let us remark that in the context of cluster analysis (cf. [16], [59]) the true number of components is to be decided. One possibility to infer the proper number of components from data is to use different likelihood ratio test statistics (cf. [86, Chapter 6.]).

There are also different approaches concerning a suitable choice of the initial parameter values. If there is no prior knowledge the starting point is usually chosen randomly or by using simple clustering techniques. In the paper [57] we have proposed to initialize the EM algorithm by evaluating the modes of an optimized kernel estimate.

Another frequently discussed problem with the EM algorithm is the slow convergence in the final iterations. Different acceleration methods are discussed in McLachlan and Peel [86, pp. 70-73], however, the acceleration procedures usually do not guarantee the valuable monotone convergence of EM algorithm.

1.3.3 Approximation Problem

In applications it is useful to distinguish between the mixture identification problem (as discussed in Sec. 1.2) when the properly chosen number of components is essential and between the practical problem of approximation of unknown distributions by means of mixtures when only the approximation accuracy is the primary goal. Approximation problems often arise in a technical environment when the data sets are produced automatically by some electronic measurements (e.g. recognition of digitized numerals or characters, segmentation of digitized images, texture modelling etc.). In such cases we usually have a large number of multidimensional measurements presumably with a multi-modal distribution.

The EM algorithm is known to have a tendency to suppress the weights of superfluous components. This mechanism is not strong enough to control the mixture complexity reliably but it is sufficient to avoid excessive number of components. In case of mixtures with a large number of components ($M = 10^1$-10^2) we obtain usually a sigmoidal distribution of component weights, i.e. there is usually a large part of components (10–20 %) having very low weights. Obviously, these components can be omitted without affecting the approximation accuracy too much. In this sense we may conclude from the experiments that the initial number of components does not play an essential role in approximation problems.

Similar conclusions can be justified also by simple heuristic considerations. One can easily imagine that there are many different possibilities to fit a mixture of many components to a large

number of multi-dimensional measurements – each possibility corresponding to a local maximum of the related likelihood function. As it can be expected, the considered local maxima are usually not very different and therefore the corresponding mixture models achieve a comparable approximation accuracy. In view of the above properties of the practical approximation problems the proper initialization of mixtures with many components is less important since there are no great differences between the individual local maxima. In case of randomly initialized mixtures we may expect a comparable quality of the resulting approximations in repeated experiments.

There is also another important aspect of the considerations above: the form of the components playing a role similar to kernel functions is also less relevant and can be defined e.g. as a product of univariate distributions. The assumption of product components corresponds to the model of conditional independence which has some advantages as an approximation tool:

- any marginal distribution of a product mixture is easily available (unlike many parametric models)
- the product mixtures can be estimated from incomplete data directly (instead of estimating missing values (cf. [24]) the mixture is always reduced to the subspace of the currently available values)
- the EM algorithm simplifies computationally (e.g. no matrix inversion is necessary in case of normal mixtures)
- the computational stability of EM Algorithm increases (no risk of singular matrices in case of normal mixtures)
- the product components support a structural modification of the finite mixture model [56] (application of the structural mixtures to classification and to some other problems is dimension-independent)
- the product mixtures can be interpreted as neural network models in biological terms (cf. [55], [54]).

In view of the obvious analogy with kernel estimates the approximation accuracy of the conditional independence models can be arbitrarily increased by increasing the number of components. Let us remark that the mixtures of products of general univariate discrete distributions (defined by a vector of probabilities) are not identifiable (e.g. mixtures of multivariate Bernoulli distributions as a special case [59]). This circumstance may facilitate the convergence of EM Algorithm to the global maximum but it is undesirable in cluster analysis.

1.3.4 Historical comments

The problem of estimation of finite mixtures has been first formulated by Pearson in 1897 but only since the sixties there is the effectively applicable iterative EM algorithm for computing the maximum likelihood estimates of mixture parameters. When omitting the iteration index the EM iteration equations may be easily obtained by algebraically rearranging the corresponding likelihood equations for mixtures. It appears that using this heuristic idea Hasselblad first

derived the EM iteration scheme ([62], [109]). The method has been studied and modified also by Behboodian [69], Kale [73], Day [22], Wolfe [116], Peters and Walker [91] and others. It has been observed in experiments that the EM algorithm increases the maximized likelihood function in each iteration. However, there is no proof of this monotone property in the above papers.

It appears that the first proof of the monotone convergence of EM algorithm is due to Schlesinger [98] (cf. Grim [50]). The result of Schlesinger concerning the monotone convergence of the EM procedure to some possibly local maximum has been reported by Isaenko and Urbakch [68] and also in full detail in the book of Ajvazjan *et al.* [4].

At present the standard reference on EM algorithm is the paper of Dempster et.al. [24] who introduced the name EM algorithm and demonstrated its wide applicability in different fields. Unfortunately, finite mixtures are mentioned in this paper only as a special case in the framework of incomplete data problem which is not very intuitive from the point of view of estimating mixtures. A more tractable presentation of EM algorithm for mixtures can be found e.g. in Titterington et.al. [107] or in the more recent monograph of McLachlan [86]. The paper of Dempster et.al. [24] contains an error [1] which has been first noticed by Wu [119] who also analyzed the convergence properties of EM algorithm in detail.

1.3.5 Available Software

There is special software for estimating mixtures available on internet. However, the implementation of EM algorithm is relatively easy and therefore it is recommendable to write own procedure especially if the underlying problem is specific in a way. The possibility to modify the EM iteration equations may be more important than the advantage of a well designed external procedure. The following list of software products with a brief description and availability address can be found in the monograph [86]:

EMMIX (McLachlan, Peel, Adams, and Basford, 1999).
A general tool to fit mixtures of normal components by maximum likelihood via the EM algorithm to continuous multivariate data.
http://www.maths.uq.edu.au/∼gjm/emmix/emmix.html

AUTOCLASS (Cheeseman and Stutz, 1996)
Adopts a Bayesian approach to fit mixtures of normal or uniform distributions to continuous multivariate variables, and mixtures of Bernoulli distributions to discrete data. The program is also able to handle missing data and the case of an unspecified number of components.
http://ic∼www.arc.nasa.gov/ic/projects/bayes∼group/people/cheeseman/

BINOMIX (Erdfelder, 1993)
Fits a mixture of binomial or beta-binomial distributions using the EM algorithm.
http://www.psychologie.uni-bonn.de/∼erdfel_e/comp/binomix.html

C.A.MAN (Boehning, Schlattmann, and Lindsay, 1992,1998)
Fits mixtures of normal (with equal or unequal variances), Poisson, geometric, binomial, exponential, or Laplace univariate distributions, using one of four fitting methods, including the EM algorithm. It also has a semi-parametric method to estimate an appropriate number of components. The maximum number

[1] an incorrect inequality in Eq. (3.14), p. 8

of data points that the PC version can analyze is 500.
http://ftp.ukbf.fu∼berlin.de/sozmed/caman.html

MCLUST/EMCLUST (Fraley and Raftery, 1999)
A software package for hierarchical clustering on the basis of mixtures of normal components under various parameterizations of the component-covariance matrices. The EM algorithm is used in the fitting process and BIC is used for the determination of the number of components. It has the option to include an additional component in the model for background (Poisson) noise.
http://www.stat.washington.edu/fraley/software.html

MGT (Jones and McLachlan, 1990b)
A subroutine for fitting a mixture of univariate normal distributions to binned and truncated data.
http://www.stat.cmu.edu/apstat/

MIX (Macdonald and Pitcher, 1979)
Works with univariate binned data, with a maximum of 80 bins and 15 components. The program will fit mixtures of normal, log-normal, gamma, exponential, or Weibull components.
http://icarus.math.mcmaster.ca/peter/mix/mix.html

MIXBIN (Uebersax)
Fits mixtures of binomial distributions via an EM-type algorithm and also gives the asymptotic standard errors by inverting the observed information matrix. The likelihood ratio test statistics AIC and BIC are also computed.
http://members.xoom.com/jsuebersax/papers.html

PROGRAM FOR GOMPERTZ MIXTURES (McLachlan et al., 1997)
An algorithm for fitting mixtures of two Gompertz distributions to censored survival data.
http://www.stat.ucla.edu/journals/jss/v02.i07

MPLUS (B. Muthén and L. Muthén)
A statistical modelling program that includes tools to fit latent class models. The criteria AIC and BIC are used for model selection.
http://www.statmodel.com/

MULTIMIX (Jorgensen and Hunt, 1996)
Adopts the location model to fit mixtures to mixed continuous and categorical variables.
ftp://ftp.math.waikato.ac.nz/pub/maj/

NORMIX (Wolfe, 1965, 1967, 1970)
Fits mixtures of normals or Bernoulli distributions from specified initial values of the parameters or from initial partitions obtained by various hierarchical clustering methods.
http://alumnus.caltech.edu/∼wolfe/

SNOB (Wallace and Dowe, 1994)
It allows the fitting of mixtures of discrete distributions (multistate Bernoulli or categorical), normal (with diagonal covariance matrices), Poisson, and von Mises distributions. The input data can contain missing values and the number of components can be estimated.
http://www.cs.monash.edu.au/∼dld/snob.html

BAYESIAN MODELLING AND MARKOV CHAIN SAMPLING (Neal, 1999)
Allows the fitting of mixtures via a Bayesian approach.

http://www.cs.toronto.edu/∼radford/fbm.software.html

1.4 Genetic Algorithms

Author: Nahum Kyriati, Tel Aviv University

Many challenges in engineering and science boil down to optimization problems. Consider the basic problem of finding the maximum of a function within a search space. The case of unimodal objective functions is well understood and standard solutions exist, see e.g. [78]. Finding the global maximum of a multimodal function is much more difficult. It is easy to show that, in the worst case, the global maximum of a general function defined on continuous support cannot be found in finite time [108]. If the search space is discrete, global optimization can in principle be accomplished using exhaustive search. The feasibility of exhaustive search depends on the size of the search space, the cost of objective function evaluation, the available computing power and the time constraint. In the context of global optimization research, the interesting problems are those for which exhaustive search is not a viable option. These "expensive" optimization problems are common, showing up in diverse application domains, from aerospace engineering to financial planning, and from oil exploration to modem adaptation.

Various approaches to global optimization have been suggested [108]. Note that the utility of any global optimization method is problem dependent [117]. This means that global optimization algorithms should be tuned to the specific problems to which they are applied, and must utilize apriori knowledge to the fullest extent. Genetic algorithms [64, 113, 101] have been successfully applied to highly important optimization tasks. In a genetic algorithm, a set of candidate locations iteratively evolves using operators inspired by the principles of crossover, mutation and selection of the fittest. In many useful cases, the candidates may rapidly converge to the global maximum. Holland's schema theorem, studies of deceptive problems and the analysis of neighborhood structures and landscapes provide some insight as to the fruitfulness of applying genetic search to classes of objective functions, see [120].

Thanks to significant research efforts, especially in the last decade, genetic algorithms now have a solid theoretical basis. Many interesting results are rooted in the Markov chain model of genetic algorithms [90]. For example, it has been shown that the canonical genetic algorithm will never converge to the global optimum, but variants of the algorithm that always maintain the best solution in the population (elitist schemes) will converge to the global optimum [96]. See also [105, 23, 94, 118, 77].

In standard genetic algorithms, once a maximum is found, candidates representing smaller local maxima are suppressed and eventually disappear. This is not always desirable. In many important problems, significant local maxima represent useful solutions, or solution strategies, that are valuable altern a tives to the global maximum. For example, in routing design, a secondary maximum might represent a backup operational scheme. In computer vision, the multiple maxima of a multimodal function could correspond to several objects, or image primitives, that should be all detected [95, 79]. Indeed, genetic algorithms aimed at finding all significant local maxima of a given objective function have been suggested. Some are sequential, i.e., determine local maxima one by one [9]. Others are parallel search methods (niching algorithms) in which candidates representing a multitude of local maxima can coexist throughout the iterative process [46]. We focus on the latter, and loosely refer to them as multi-modal genetic algorithms, in the sense that several modes should be simultaneously detected. Theoretical analyses of niching algorithms, e.g. [65, 66] elucidate the fast convergence to niching equilibrium, the long niche loss times, and the dependence of these characteristics on the fitness landscape, i.e., on the properties of the objective function.

Whenever a genetic algorithm is called into operation, the right time to terminate the search must also be decided. This problem is obvious, and its importance cannot be overstated. Stopping too early means that the solution is inadequate; stopping too late implies that valuable resources are wasted. The lack of good stopping rules was widely recognized as a major deficiency in genetic algorithms, see e.g. [49]. Given its significance, the stopping problem has received surprisingly little attention. Practitioners use simplistic rules, such as "stop after k iterations". More sophisticated rules, such as "stop when there has been no significant improvement in the last k iterations" have been suggested [35, 100] in the context of restart scheduling, i.e., periodic re-initialization of the search to alleviate the problem of premature convergence. Specifically, performance data from previous runs on similar problems is used in [35] to generate the best possible solution given a fixed amount of time. Adaptive changing of the threshold number of generations according to the diversity of the fitness values in the population is proposed in [100].

Advances in convergence analysis of genetic algorithms explain and quantify aspects of genetic algorithm behavior that are closely related to the stopping problem [7, 48]. Notably, [8] provides a bound on the number of iterations needed to achieve a level of confidence that a genetic algorithm has seen all strings, and hence an optimal solution. Stopping rules for the elitist genetic algorithm model are presented in [89].

When dealing with "expensive" global optimization problems, thorough exploration of the search space cannot be carried out. This means that chances to find the global optimum itself are low, and one must be satisfied with some "good" (but not necessarily optimal) solution that can be found reasonably fast. In such cases, the potential benefit from continued search should be weighted against the cost of additional probing. The key to analysis along these lines is in estimating of the expected utility of additional search. A Bayesian approach was taken in [67]. The cost distribution of the last generation is used as prior distribution, and after creation of the new distribution a posterior distribution is derived by Bayes' formula. Based on these estimations, one can decide whether to stop or continue.

Given the fundamental difficulty of global optimization, and the complexity of the genetic search mechanism , the theoretical achievements in genetic algorithm research are admirable. However, the all-important stopping problem, encountered in any challenging application of genetic algorithms, has received much less attention than other aspects of evolutionary optimization. From the practitioner's point of view, there is a gap between the few theoretical results that are available and the need for clear stopping rules. This is especially true for multi-modal genetic search: our survey of the literature revealed n o results on optimal stopping of niching algorithms.

This document has several roots. One is our interest in global optimization problems in computer vision, e.g. [75], especially those for which constraints on the objective function can be used to guarantee convergence to the global maximum [103]. The other is our work on optimal stopping of voting in the probabilistic Hough Transform, a poll-based pattern recognition method [74, 122, 99].

Bibliography

[1] Ahmed A. *Signal Separation*. PhD thesis, Signal Processing Group, Department of Engineering, University of Cambridge, U.K., 2000.

[2] Doucet A. *Monte Carlo Algorithms for Bayesian Estimation of Hidden Markov Models*. PhD thesis, University Paris-Sud, Orsay, France, 1997.

[3] Doucet A., De Freitas J. F. G., and Gordon N. J. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.

[4] S.A. Ajvazjan, Z.I. Bezhaeva, and O.V. Staroverov. *Classification of Multivariate Observations (in Russian)*. Statistika, Moscow, 1974.

[5] Y. Amit and U. Grenander. Comparing sweep strategies for stochastic relaxation. *J. Multivar. Anal.*, 37:197–222, 1991.

[6] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50:174–188, 2002.

[7] H. Aytug and G.J. Koehler. Stopping criteria for finite length genetic algorithms. *ORSA Journal of Computing*, 96:195–201, 1996.

[8] H. Aytug and G.J. Koehler. New stopping criterion for genetic algorithms. *European Journal of Operations Research*, 126:662–674, 2000.

[9] D. Beasley, D.R. Bull, and R.R. Martin. A sequential niche technique for multimodal function optimization. *Evolutionary Computation*, 1:101–125, 1993.

[10] J Besag. On the statistical analysis of dirty pictures. *J. Royal statistical soc. B.*, 48:259–302, 1986.

[11] J. Besag and P. Green. Spatial statistics and bayesian computation. *Journal of the Royal Statistical Society, Series B*, B-55:25–37, 1993.

[12] J. Besag, P. Green, D. Higdon, and K. Mengersen. Bayesian computation and stochastic systems. *Statistical Science*, 10:3–66, 1995.

[13] C. Bouman and B. Liu. Multiple resolution segmentation of textured images. *IEEE Trans. Pattern Anal. Mach. Int.*, PAMI–13:99–113, 1991.

[14] S. Brooks. Markov chain Monte Carlo and its applications. *The Statistician*, 47:69–100, 1998.

[15] Andrieu C. and Godsill S.J. A particle filter for model based audio source separation. In *International Workshop on Independent Component Analysis and Blind Signal Separation, ICA 2000*, Helsinki, Finland, 2000.

[16] M.A. Carreira-Perpignan and S. Renals. Practical identifiability of finite mixtures of multivariate bernoulli distributions. *Neural Computation*, 12:141–152, 2000.

[17] T.S. Chiang and Y. Chow. On the convergence rate of annealing processes. *SIAM J. Control and Optimization*, 1987.

[18] F.S. Cohen. *Modeling and Application of Stochastic Processes*, chapter Markov random fields for image modelling and analysis. Kluwer Academic Publishers, Boston, 1986.

[19] M. K. Cowles and B. P. Carlin. Markov chain Monte Carlo convergence diagnostics: a comparitive review. *Journal of the American Statistical Association*, 91:883–904, 1996.

[20] G.R. Cross and A.K. Jain. Markov random field texture models. *IEEE Trans. Pattern Anal. Mach. Int.*, PAMI–5:25–39, 1983.

[21] Anderson B. D. and Moore J. M. *Optimal Filtering*. Prentice-Hall, New Jersey, 1979.

[22] N.E. Day. Estimating the components of a mixture of normal distributions. *Biometrika*, 56:463–474, 1969.

[23] K.A. DeJong, W.M. Spears, and D.F. Gordon. Using Markov chains to analyze GAFOs. In *Proc. Foundations of Genetic Algorithms (FOGA '94)*, volume 1, pages 115–137, 1994.

[24] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Stat. Soc.*, B 39:1–38, 1977.

[25] Van der Merwe R., Doucet A., De Freitas N., and Wan E. The unscented particle filter. *Advances in Neural Information Processing Systems*, 13:584–590, 2000.

[26] A. Doucet, J.F.G. de Freitas, and N.J. Gordon. An introduction to sequential Monte Carlo methods. In A. Doucet, J.F.G. de Freitas, and N.J. Gordon, editors, *Sequential Monte Carlo methods in practice*. Spinger-Verlag, New York, 2001.

[27] A. Doucet, S.J. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10:197–208, 2000.

[28] R.C. Dubes and A.K. Jain. Random fields models in image analysis. *J.of Appl. Statistics*, 16:131–164, 1989.

[29] Handschin J. E. and Mayne D. Q. Monte carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering. *International Journal of Control, 9*, pages 547–559, 1969.

[30] Kalman R. E. A new approach to linear filtering and prediction problems. *Transactions of the ASME, Journal of Basic Engineering*, 83:95–107, 1961.

[31] Kuruoğlu E. E., Bedini L., Paratore M. T., Salerno E., and Tonazzini A. Source separation in astrophysical maps using independent factor analysis. *Neural Networks*, 16:479–491, 2003.

[32] Moulines E., Cardoso J. F., and Gassiat E. Maximum likelihood for blind separation and deconvolution of noisy signals using mixture models. In *Proceedings of ICASSP '97*, volume 5, pages 3617–3620, 1997.

[33] B.S. Everitt and D.J. Hand. *Finite Mixture Distributions*. Chapman & Hall, London, 1981.

[34] W. Fong, S. J. Godsill, A. Doucet, , and M. West. Monte carlo smoothing with application to speech enhancement. *IEEE Trans. on Signal Processing*, 50(2):438–449, February 2002.

[35] A.S. Fukunaga. Restart scheduling for genetic algorithms. In *Proc. Parallel Problem Solving from Nature V (Lecture Notes in Computer Science)*, volume 1498, New York, 1998. Springer.

[36] Casella G. and Robert C. P. *Monte Carlo Statistical Methods*. Springer Texts in Statistics, 1999.

[37] Kitagawa G. Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5:1–25, 1996.

[38] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian data analysis*. Chapman and Hall, London, 1995.

[39] D. Geman, G. Reynolds, and C. Yang. *Markov Random Field*, chapter Stochastic Algorithms for Restricted Image Spaces and Experiments in Deblurring. Academic Press, 1993.

[40] S. Geman and D. Geman. Stochastic relaxation , gibbs distributions and bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Int.*, 6(11):721–741, November 1984.

[41] S. Geman and D. Geman. Stochastic relaxation, Gibb's distributions and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Machine Intell.*, PAMI-6:721–741, 1984.

[42] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. *Markov chain Monte Carlo in practice*. Chapman and Hall, London, 1996.

[43] S. J. Godsill, A. Doucet, and M. West. Maximum a posteriori sequence estimation using monte carlo particle filters. *Ann. Inst. Stat. Math.*, 53(1):82–96, March 2001.

[44] S. J. Godsill and J. Vermaak. Models and algorithms for tracking using trans-dimensional sequential monte carlo. In *IEEE proceedings of the International Conference on Acoustics Speech and Signal Processing (ICASSP)*, Montreal, May 2004.

[45] S.J. Godsill, A. Doucet, and M. West. Monte carlo smoothing for non-linear time series. *Journal of the American Statistical Association*, 50:438–449, 2004.

[46] D.E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In *Proc. Int. Conf. On Genetic Algorithms*, volume 1, pages 41–49, 1987.

[47] P. J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82:711–732, 1995.

[48] D. Greenhalgh and S. Marshall. Convergence criteria for genetic algorithms. *SIAM J. Computing*, 30:269–282, 2000.

[49] J.J. Grefenstette. Predictive models using fitness distributions of genetic operators. In D. Whitley, editor, *Foundations of Genetic Algorithms 3*. Morgan Kaufmann, San Mateo, 1995.

[50] J. Grim. On numerical evaluation of maximum - likelihood estimates for finite mixtures of distributions. *Kybernetika*, l8(3):173–190, 1982.

[51] J. Grim. Knowledge representation and uncertainty processing in the probabilistic expert system PES. *International Journal of General Systems*, 22(2):103–111, 1994.

[52] J. Grim, P. Boček, and P. Pudil. Safe dissemination of census results by means of interactive probabilistic models. In *Proceedings of the ETK-NTTS 2001 Conference, Hersonissos (Crete)*, volume 2, pages 849–856. European Communities 2001, June 18-22 2001.

[53] J. Grim and M. Haindl. Texture modelling by discrete distribution mixtures. *Computational Statistics and Data Analysis*, 41(3-4):603–615, 2003.

[54] J. Grim, P. Just, and P. Pudil. Strictly modular probabilistic neural networks for pattern recognition. *Neural Network World*, 13(6):599–615, 2003.

[55] J. Grim, J. Kittler, P. Pudil, and P. Somol. Multiple classifier fusion in probabilistic neural networks. *Pattern Analysis & Appl.*, 7(5):221–233, 2002.

[56] J. Grim, P. Pudil, and P. Somol. Recognition of handwritten numerals by structural probabilistic neural networks. In H. Bothe and R. Rojas, editors, *Proceedings of the Second ICSC Symposium on Neural Computation*, pages 528–534, Wetaskiwin, May 2000. ICSC.

[57] J. Grim, P. Somol, J. Novovičová, P. Pudil, and F.J. Ferri. Initializing normal mixtures of densities. In A. K. Jain, S. Venkatesh, and B.C. Lovell, editors, *Proceedings of the 14th International Conference on Pattern Recognition*, pages 886–890, Los Alamitos, August 1998. IEEE.

[58] J. Grim, P. Somol, P. Pudil, and P. Just. Probabilistic neural network playing a simple game. In M. Gori and S. Marinai, editors, *Artificial Neural Networks in Pattern Recognition*, pages 132–138, Florence, Italy, September 2003. University of Florence.

[59] M. Gyllenberg, T. Koski, E. Reilink, and M. Verlaan. Non-uniqueness in probabilistic numerical identification of bacteria. *Journal of Applied Probability*, 31:542–548, 1994.

[60] Attias H. Independent factor analysis. *Neural Computation*, 11:803–851, 1999.

[61] B. Hájek. A tutorial survey of theory and application of simulated annealing. In *Proc. 27th IEEE Conf. Decision and Control*, pages 755–760, 1985.

[62] V. Hasselblad. Estimation of prameters for a mixture of normal distributions. *Technometrics*, 8:431–444, 1966.

[63] S. Haykin. *Neural Networks: a comprehensive foundation*. Morgan Kaufman, San Mateo CA, 1993.

[64] J.H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.

[65] J. Horn. Finite markov chain analysis of genetic algorithms with niching. In *Proc. International Conference on Genetic Algorithms*, volume 1, pages 110–117, 1993.

[66] J. Horn and D.E. Goldberg. A timing analysis of convergence to fitness sharing equilibrium. In *Proc. Parallel Problem Solving from Nature V (Lecture Notes in Computer Science)*, volume 1498, New York, 1999. Springer.

[67] M. Hulin. An optimal stop criterion for genetic algorithms: A Bayesian approach. In *Proc. International Conference on Genetic Algorithms (ICGA)*, volume 1, pages 135–143, 1997.

[68] O.K. Isaenko and K.I. Urbakh. *Decomposition of probability distribution mixtures into their components (in Russian)*. Theory of probability, mathematical statistics and theoretical cybernetics, Vol. 13. VINITI, Moscow, 1976.

[69] Behboodian J. On a mixture of normal distributions. *Biometrika*, 57(1):215–217, 1970.

[70] Geweke J. Bayesian inference in econometric models using monte carlo integration. *Econometrica, 24*, pages 1317–1399, 1989.

[71] Gordon N. J., Salmond D. J., and Smith A. F. M. Novel approach to non-linear / non-gaussian bayesian state estimation. pages 107–113, 1993.

[72] Julier S. J. and Uhlmann J. K. A new extension of the kalman filter to nonlinear systems. In *Proceedings of AeroSence: the 11th International Symposium on Aerospace/Defence Sensing, Simulation and Controls, Vol. Multi Sensor Fusion, Tracking and Resource Management II*, 1997.

[73] B.K. Kale. On the solution of likelihood equations by iteration processes: The multiparametric case. *Biometrika*, 49:479–486, 1962.

[74] N. Kiryati, Y. Eldar, and A.M. Bruckstein. A probabilistic hough transform. *Pattern Recognition*, 24:303–316, 1991.

[75] N. Kiryati and Y. Gofman. Detecting symmetry in grey level images: the global optimization approach. *International Journal of Computer Vision*, 29:29–45, 1998.

[76] S.Z. Li, K.L. Chan, and H. Wang. Bayesian image restoration and segmentation by constrained optimization. In *Proc. CVPR-96*, pages 1–6, San Francisco, 1996.

[77] J.A. Lozano. Genetic algorithms: Bridging the convergence gap. *Theoretical Computer Science*, 229:11–22, 1999.

[78] D.G. Luenberger. *Linear and Nonlinear Programming*. Addison-Wesley, New York, 1984.

[79] E. Lutton and P. Martinez. A genetic algorithm with sharing for the detection of 2d geometric primitives in images. In *Artificial Evolution (Lecture Notes in Computer Science)*, volume 1063, pages 287–303, New York, 1996. Springer.

[80] Costagli M. and Kuruoglu E.E. Astrophysical source separation using particle filters. In *International Workshop on Independent Component Analysis and Blind Signal Separation*, Granada, Spagna, 2004.

[81] Everson R. M. and Roberts S. J. *Particle Filters for Non-stationary ICA.* Advances in Independent Components Analysis, M. Girolami (Ed.) 23-41, Springer, 2000.

[82] D.J.C. Mackay. Introduction to Monte Carlo methods. Technical report, Department of Physics, Cambridge University, 1998.

[83] J. Marroquin, S. Mitter, and T. Poggio. Probabilistic solution of ill-posed problems in computational vision. *J. Am. Stat. Assoc.*, 82(397):76–89, 1987.

[84] G.J. McLachlan and K.E. Basford. *Mixture Models: Inference and Application to Clustering*. Marcel Dekker, New York, 1988.

[85] G.J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. John Wiley & Sons, New York, 1997.

[86] G.J. McLachlan and D. Peel. *Finite Mixture Models*. John Wiley & Sons, New York, Toronto, 2000.

[87] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equations of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087–1091, 1953.

[88] A. Morgan. *Solving Polynomial Systems Using Continuation for Engineering and Scientific Problems*. Prentice-Hall, 1987.

[89] C.A. Murthy, D. Bhandari, and S.K. Pal. Epsilon-optimal stopping time for genetic algorithms. *Fundamenta Informaticæ*, 35:91–111, 1998.

[90] A.E. Nix and M.D. Vose. Modeling genetic algorithms with Markov chains. *Annals of Mathematics and Artificial Intelligence*, 5:79–88, 1992.

[91] B.C. Peters and W.A. Coberly. The numerical evaluation of the maxmimum-likelihood estimate of mixture proportions. *Comm. Statist. A - Theory Methods AS*, 12:1127–1135, 1976.

[92] J. G. Propp and D. B. Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms*, 9:223–252, 1996.

[93] Chellapa R. and A. Jain, editors. *Markov Random Fields*. Academic Press, 1993.

[94] Y. Rabinovich and A. Wigderson. Techniques for bounding the convergence rate of genetic algorithms. *Random Structures Algorithms*, 14:111–138, 1999.

[95] G. Roth and M.D. Levine. Geometric primitive extraction using a genetic algorithm. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16:901–905, 1994.

[96] G. Rudolph. Convergence analysis of canonical genetic algorithms. *IEEE Transactions on Neural Networks*, 5:96–101, 1994.

[97] Arulampalam M. S., Maskell S., Gordon N., and Clapp T. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.

[98] M.I. Schlesinger. Relation between learning and self-learning in pattern recognition (in russian). *Kibernetika (Kiev)*, (2):81–88, 1968.

[99] D. Shaked, O. Yaron, and N. Kiryati. Deriving stopping rules for the probabilistic Hough transform by sequential analysis. *Computer Vision and Image Understanding*, 63:512–526, 1996.

[100] H. Shimodaira. Methods for reinitializing the population to improve the performance of a diversity-control-oriented genetic algorithm. *IEICE Trans. Inf. and Syst.*, E84-D:1745–1755, 2001.

[101] M. Sipper. *A Brief Introduction to Genetic Algorithms*. http://www.cs.bgu.ac.il/ sipper/ga.html.

[102] A.F.M. Smith and G.O. Roberts. Bayesian computation via gibbs sampler and related markov chain monte carlo methods. *J. Royal Stat. Soc.*, B–55:3–23, 1993.

[103] M. Soffer and N. Kiryati. Guaranteed convergence of the Hough transform. *Computer Vision and Image Understanding*, 69:119–134, 1998.

[104] L.R. Streit and T.E. Luginbuhl. Maximum likelihood training of probabilistic neural networks. *IEEE Trans. on Neural Networks*, (5):764–783, 1994.

[105] J. Suzuki. A Markov chain analysis on a genetic algorithm. In *Proc. International Conference on Genetic Algorithms*, volume 1, pages 146–153, 1993.

[106] M. A. Tanner. *Tools for statistical inference: methods for the exploration of posterior distributions and likelihood functions*. Springer-Verlag, New York, Third edition, 1996.

[107] D.M. Titterington, A.F.M. Smith, and U.E. Makov. *Statistical analysis of finite mixture distributions*. John Wiley & Sons, Chichester, New York, 1985.

[108] A. Törn and A. Žilinkas. Global optimization. In *Lecture Notes in Computer Science*, volume 350, New York, 1989. Springer.

[109] Hasselblad V. Estimation of finite mixtures of distributions from the exponential family. *Journal of Amer. Statist. Assoc.*, 58:1459–1471, 1969.

[110] J. Vermaak, C. Andrieu, A. Doucet, and S. J. Godsill. Particle methods for bayesian modelling and enhancement of speech signals. *IEEE Trans. on Speech and Audio Processing*, 10(3):173–185, 2002.

[111] J. Vermaak, S. J. Godsill, and A. Doucet. Radial basis function regression using trans-dimensional sequential monte carlo. In *IEEE Workshop on Statistical Signal Processing*, 2003.

[112] J. Vermaak, S. J. Godsill, and A. Doucet. *Sequential Bayesian kernel regression*. In Advances in Neural Information Processing Systems 16, Cambridge, MA. MIT Press,, 2003.

[113] D. Whitley. A genetic algorithm tutorial. Technical Report CS-93-103, Dept. of Computer Science, Colorado State University, 1993.

[114] G. Winkler. *Image Analysis, Random Fields and Dynamic Monte Carlo Methods*. Springer, Berlin, 1991.

[115] G. Winkler. *Image analysis, random fields and dynamic Monte Carlo methods*. Springer-Verlag, Berlin, First edition, 1995.

[116] J.H. Wolfe. Pattern clustering by multivariate mixture analysis. *Multivariate Behavioral Research*, 5:329–350, 1970.

[117] D.H. Wolpert and W.G. MacReady. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1:67–82, 1997.

[118] A.H. Wright and Y. Zhao. Markov chain models of genetic algorithms. In *Proceedings of the Genetic and Evolutionary Computation (GECCO) Conference*, pages 734–742, 1999.

[119] C.F.J. Wu. On the convergence properties of the EM algorithm. *Ann. Statist.*, 11:95–103, 1983.

[120] X. Yao. An overview of evolutionary computation. *Chinese Journal of Advanced Software Research*, 3:12–29, 1996.

[121] C. Yim, A.C. Bovik, and J.K. Aggarwal. Bayesian range segmentation using focus cues. In *Proc. ICPR-96*, pages 482–486, Vienna, 1996.

[122] A. Ylä-Jääski and N. Kiryati. Adaptive termination of voting in the probabilistic circular hough transform. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16:911–915, 1994.

Chapter 2

Current State of Work in the Network

2.1 Exact Metropolis Hastings sampling for marked point processes using a C++ library

Author: Mari-Colette van Lieshout, CWI

In a joint report (CWI, Research Report PNA-R0403, June 2004 [96]) with R.S. Stoica of the University Jaume I, we document MPPLIB, a C++ library for marked point processes developed at CWI, and illustrate its use by means of a new exact Metropolis-Hastings simulation algorithm. The paper is a follow up of earlier work about generalising exact simulation algorithms for point processes to the case of object processes.

2.2 Moving object detection in wavelet compressed video

Author: B. Ugur Toreyin, Bilkent University

2.2.1 Introduction

In many surveillance systems, the video is compressed intra-frame only without performing motion compensated prediction due to legal reasons. Courts do not accept predicted image frames as legal evidence in many countries [32]. As a result, a typical surveillance video is composed of a series of individually compressed image frames. In addition, many practical systems have built-in VLSI hardware image compressors directly storing the compressed video data coming from several cameras into a hard-disc. The main reason for this is that standard buses used in PC's cannot handle the raw multi-channel video data.

In this paper, it is assumed that the video data is available in wavelet compressed format. In many multi-channel real-time systems, it is not possible to use uncompressed video due to available bus and processor limitations. The proposed moving object detection algorithm compares the Wavelet Transform (WT) of the current image with the WTs of the past image frames to detect motion and moving regions in the current image without performing an inverse wavelet transform operation. Moving regions and objects can be detected by comparing the wavelet transforms of the current image with the wavelet transform of the background scene which can be estimated from the wavelet transforms of the past image frames. If there is a significant difference between the two wavelet transforms then this means

that there is motion in the video. If there is no motion then the wavelet transforms of the current image and the background image ideally should be equal to each other or very close to each other due to quantization process during compression. Stationary wavelet coefficients belong to the wavelet transform of the background. This is because the background of the scene is temporally stationary [4, 22, 32, 40, 65]. If the viewing range of the camera is observed for some time, then the wavelet transform of the entire background can be estimated as moving regions and objects occupy only some parts of the scene in a typical image of a video and they disappear over time. On the other hand, pixels of foreground objects and their wavelet coefficients change in time. Non-stationary wavelet coefficients over time correspond to the foreground of the scene and they contain motion information. A simple approach to estimate the wavelet transform of the background is to average the observed wavelet transforms of the image frames. Since moving objects and regions occupy only a part of the image they can conceal a part of the background scene and their effect in the wavelet domain is cancelled over time by averaging.

Any one of the space domain approaches [4, 22, 40, 61, 65, 94] for background estimation can be implemented in wavelet domain providing real-time performance. For example, the background estimation method in [22] can be implemented by simply computing the wavelet transform of both sides of their background estimation equations.

2.2.2 Hybrid Algorithm for Moving Object Detection

Background subtraction is commonly used for segmenting out objects of interest in a scene for applications such as surveillance. There are numerous methods in the literature [4, 22, 32, 40, 65]. The background estimation algorithm described in [22] uses a simple IIR filter applied to each pixel independently to update the background and use adaptively updated thresholds to classify pixels into foreground and background. This is followed by some post processing to correct classification failures. Stationary pixels in the video are the pixels of the background scene because the background can be defined as temporally stationary part of the video. If the scene is observed for some time, then pixels forming the entire background scene can be estimated because moving regions and objects occupy only some parts of the scene in a typical image of a video. A simple approach to estimate the background is to average the observed image frames of the video. Since moving objects and regions occupy only a part of the image, they conceal a part of the background scene and their effect is cancelled over time by averaging. Our main concern is real-time performance of the system. In Video Surveillance and Monitoring (VSAM) Project at Carnegie Mellon University [22] a recursive background estimation method was developed from the actual image data. Let $I_n(x,y)$ represent the intensity (brightness) value at pixel position (x,y) in the n^{th} image frame I_n. Estimated background intensity value at the same pixel position, $B_{n+1}(x,y)$, is calculated as follows:

$$B_{n+1}(x,y) = \begin{cases} aB_n(x,y) + (1-a)I_n(x,y) & \text{if } (x,y) \text{ is non-moving} \\ B_n(x,y) & \text{if } (x,y) \text{ is moving} \end{cases} \tag{2.1}$$

where $B_n(x,y)$ is the previous estimate of the background intensity value at the same pixel position. The update parameter a is a positive real number close to one. Initially, $B_0(x,y)$ is set to the first image frame $I_0(x,y)$. A pixel positioned at (x,y) is assumed to be moving if the brightness values corresponding to it in image frame I_n and image frame I_{n-1}, satisfy the following inequality:

$$|I_n(x,y) - I_{n-1}(x,y)| > T_n(x,y) \tag{2.2}$$

where $I_{n-1}(x,y)$ is the brightness value at pixel position (x,y) in the $(n-1)^{st}$ image frame I_{n-1}. $T_n(x,y)$ is a threshold describing a statistically significant brightness change at pixel position (x,y). This threshold

is recursively updated for each pixel as follows:

$$T_{n+1}(x,y) = \begin{cases} aT_n(x,y) + (1-a)(c|I_n(x,y) - B_n(x,y)|) & \text{if } (x,y) \text{ is non-moving} \\ T_n(x,y) & \text{if } (x,y) \text{ is moving} \end{cases} \tag{2.3}$$

where c is a real number greater than one and the update parameter a is a positive number close to one. Initial threshold values are set to an experimentally determined value. As it can be seen from (3), the higher the parameter c, higher the threshold or lower the sensitivity of detection scheme. It is assumed that regions significantly different from the background are moving regions. Estimated background image is subtracted from the current image to detect moving regions. In other words all of the pixels satisfying:

$$|I_n(x,y) - B_n(x,y)| > T_n(x,y) \tag{2.4}$$

are determined. These pixels at (x,y) locations are classified as the pixels of moving objects.

2.2.3 Moving Object Detection in Wavelet Domain

Above arguments and the methods proposed in [61], [94] are valid in compressed data domain as well, [65]. In [65], DCT domain data is used for motion detection in video. In our case, a wavelet transform based coder is used for data compression. The wavelet transform of the background scene can be estimated from the wavelet coefficients of past image frames, which do not change in time, whereas foreground objects and their wavelet coefficients change in time. Such wavelet coefficients belong to the background because the background of the scene is temporally stationary. Non-stationary wavelet coefficients over time correspond to the foreground of the scene and they contain motion information. If the viewing range of the camera is observed for some time, then the wavelet transform of the entire background can be estimated because moving regions and objects occupy only some parts of the scene in a typical image of a video and they disappear over time.

Let B be an arbitrary image. This image is processed by a single stage separable Daubechies 9/7 filterbank and four quarter size subband images are obtained. Let us denote these images as $LL(1), HL(1), LH(1), HH(1)$ [3]. In a Mallat wavelet tree, $LL(1)$ is processed by the filterbank once again and $LL(2), HL(2), LH(2), HH(2)$ are obtained. Second scale subband images are the quarter size versions of $LL(1)$. This process is repeated several times in a typical wavelet image coder. A three scale wavelet decomposition of an image is shown in Fig. 1.

Let D_n represent any one of the subband images of the background image B_n at time instant n. The subband image of the background D_{n+1} at time instant $n+1$ is estimated from D_n as follows:

$$D_{n+1}(i,j) = \begin{cases} aD_n(i,j) + (1-a)J_n(i,j) & \text{if } (i,j) \text{ is non-moving} \\ D_n(i,j) & \text{if } (i,j) \text{ is moving} \end{cases} \tag{2.5}$$

where J_n is the corresponding subband image of the current observed image frame I_n. The update parameter a is a positive real number close to one. Initial subband image of the background, D_0, is assigned to be the corresponding subband image of the first image of the video I_0. In Equations (1)-(4), (x,y)'s correspond to the pixel locations in the original image, whereas in (5) and in all the equations in this section, (i,j)'s correspond to locations of subband images' wavelet coefficients.

A wavelet coefficient at the position (i,j) in a subband image is assumed to be moving if

$$|J_n(i,j) - J_{n-1}(i,j)| > T_n(i,j) \tag{2.6}$$

where $T_n(i,j)$ is a threshold recursively updated for each wavelet coefficient as follows:

$$T_{n+1}(i,j) = \begin{cases} aT_n(i,j) + (1-a)(b|J_n(i,j) - D_n(i,j)|) & \text{if } (i,j) \text{ is non-moving} \\ T_n(i,j) & \text{if } (i,j) \text{ is moving} \end{cases} \tag{2.7}$$

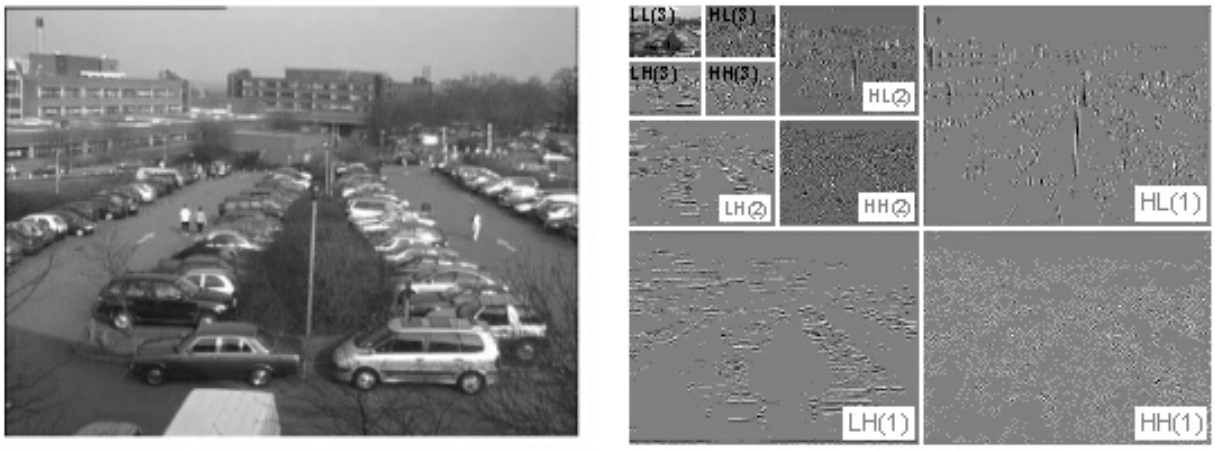

Figure 2.1: Original image and its corresponding three levels of the wavelet tree consisting of subband images (luminance data is shown)

where b is a real number greater than one and the update parameter a is a positive real number close to one. Initial threshold values can be experimentally determined. As it can be seen from the above equation, the higher the parameter b, higher the threshold or lower the sensitivity of detection scheme. Estimated subband image of the background is subtracted from the corresponding subband image of the current image to detect the moving wavelet coefficients and consequently moving objects as it is assumed that the regions different from the background are the moving regions. In other words, all of the wavelet coefficients satisfying the inequality

$$|J_n(i,j) - D_n(i,j)| > T_n(i,j) \tag{2.8}$$

are determined.

It should be pointed out that there is no fixed threshold in this method. A specific threshold is assigned to each wavelet coefficient and it is adaptively updated according to (2.7).

Once all the wavelet coefficients satisfying the above inequalities are determined, locations of corresponding regions on the original image are determined. If a single stage Haar wavelet transform is used in data compression then a wavelet coefficient satisfying (8) corresponds to a two by two block in the original image frame I_n. For example, if $(i,j)^{th}$ coefficient of the subband image $HH_n(1)$ (or other subband images $HL_n(1)$, $LH_n(1)$, $LL_n(1)$) of I_n satisfies (8), then this means that there exists motion in a two pixel by two pixel region in the original image, $I_n(k,m)$, $k = 2i, 2i-1, m = 2j, 2j-1$, because of the subsampling operation in the discrete wavelet transform computation. Similarly, if the $(i,j)^{th}$ coefficient of the subband image $HH_n(2)$ (or other second scale subband images $HL_n(2)$, $LH_n(2)$, $LL_n(2)$) satisfies (8) then this means that there exists motion in a four pixel by four pixel region in the original image, $I_n(k,m), k = 4i, 4i-1, 4i-2, 4i-3$ and $m = 4j, 4j-1, 4j-2, 4j-3$. In general, a change in the l^{th} level wavelet coefficient corresponds to a 2^l by 2^l region in the original image.

Visioprime [99] designed a video processing system which feeds the compressed video data in Aware Inc.'s Motion Wavelet format to our system [42]. It uses Daubechies' 9/7 biorthogonal wavelet. In this biorthogonal transform, the number of pixels forming a wavelet coefficient is larger than four but most of the contribution comes from the immediate neighborhood of the pixel $I_n(k,m) = (2i, 2j)$ in the first level wavelet decomposition, and $(k,m) = (2^l i, 2^l j)$ in the l^{th} level wavelet decomposition, respectively. Therefore, in this paper, we classify the immediate neighborhood of $(2i, 2j)$ in a single stage wavelet

Table 2.1: Comparison of motion detection methods with videos having large moving objects. All videos are captured at 10 fps

Large Object Videos	Object	Subband Domain Method	VSAM	GMM
VIDEO-1	MAN1	15	15	16
	MAN2	19	19	19
VIDEO-2	MAN1	13	13	13
	MAN2	74	74	74
	MAN3	164	164	164
VIDEO-3	MAN1	15	15	15
	MAN2	20	20	21

decomposition or in general $(2^l i, 2^l j)$ in the l^{th} level wavelet decomposition as a moving region in the current image frame, respectively. Determining the moving pixels of the corresponding regions, the union of them on the original image is formed to locate the moving object(s) in the video. These pixels are processed by a region growing algorithm to include the pixels located at immediate neighborhood of them. This region growing algorithm checks whether the following condition is met for these pixels:

$$|J_n(i+m, j+m) - D_n(i+m, j+m)| > K\, T_n(i+m, j+m) \tag{2.9}$$

where $m = \pm 1$, and $0.8 < K < 1$, $K \in \mathbf{R}^+$. If this condition is satisfied, then that particular pixel is also classified as moving. After this classification of pixels, moving objects are formed and encapsulated by their minimum bounding boxes.

2.2.4 Experimental Results

The above algorithm is implemented using C++ 6.0, running on a 1500 MHz Pentium 4 processor. The PC based system can handle 16 video channels captured at 5 frames per second in real-time. Each image fed by the channels has the frame size of PAL composite video format, which is 720 pixel by 576 pixel.

The video data is available in compressed form. Only the lowest resolution part of the compressed video bit-stream is decoded to obtain the low-low, low-high, high-low, and high-high coefficients which are used in moving object detection. Higher resolution wavelet sub-images are not decoded.

The performance of our algorithm is tested using 65 different video sequences. These sequences have different scenarios, covering both indoor and outdoor videos under various lighting conditions containing different video objects with various sizes. Some example snapshots are shown in Fig. 2. In a typical surveillance system, 16 video channels are displayed in a monitor simultaneously as shown in Fig. 3. The size of each video window is 256 by 192 in a 1024 by 768 monitor. Therefore, there is no need to reconstruct full-resolution images during regular screening. If the security guard wants to take a look at one of the video channels more carefully then one needs to decode the entire bit-stream of that particular channel and synthesize the full-resolution image using the reconstruction filter-bank of the wavelet transform. Otherwise there is no need to fully decompress 16 channels.

The moving regions are also detected by using two different background subtraction methods over 180 by 144 size images. They are the hybrid method of VSAM [22] and the method based on modeling the background using Gaussian Mixture Models (GMM) [85]. The low-resolution 180 by 144 images can be obtained from the 2^{nd} low-low of the wavelet pyramid and the composite image shown in Fig. 3

Table 2.2: Comparison of motion detection methods with videos having small moving objects. Toy car videos and Crowded parking lot video are captured at 15 fps and 5 fps, respectively

Small Object Videos	Object	Subband Domain Method	VSAM	GMM
Toy Cars-1	CAR1	40	40	40
	CAR2	65	65	65
	CAR3	75	75	76
Toy Cars-2	CAR1	70	70	71
	CAR2	78	78	78
Crowded Parking Lot-3	MAN1	12	4	6
	COUPLE1	14	11	11
	WOMAN1	16	5	6
	WOMAN2	18	17	17
	COUPLE2	29	29	29
	CAR1	34	34	35
	CAR2	94	94	95

Table 2.3: Comparison of motion detection methods in a parking lot at night. This video is captured at 3 fps

Video	Object	Subband Domain Method	VSAM	GMM
Dark Parking Lot	MAN1	67	67	67
	MAN2	233	233	235
	MAN3	603	602	604

Table 2.4: Frame numbers of some outdoor videos at which false alarms occur when leaves of the surrounding trees move with the wind. Indoor videos yield no false alarms

Videos	Subband Domain Method	VSAM	GMM
OUTDOOR-1	72, 81, 86, 91	51, 61, 72, 81, 91	69, 72, 83
OUTDOOR-2	420, 440, 462,481,497	419, 432, 449, 463, 480, 498	422, 481, 487, 500
OUTDOOR-3	No false alarms	No false alarms	No false alarms
INDOOR-1	No false alarms	No false alarms	No false alarms
INDOOR-2	No false alarms	No false alarms	No false alarms
INDOOR-3	No false alarms	No false alarms	No false alarms

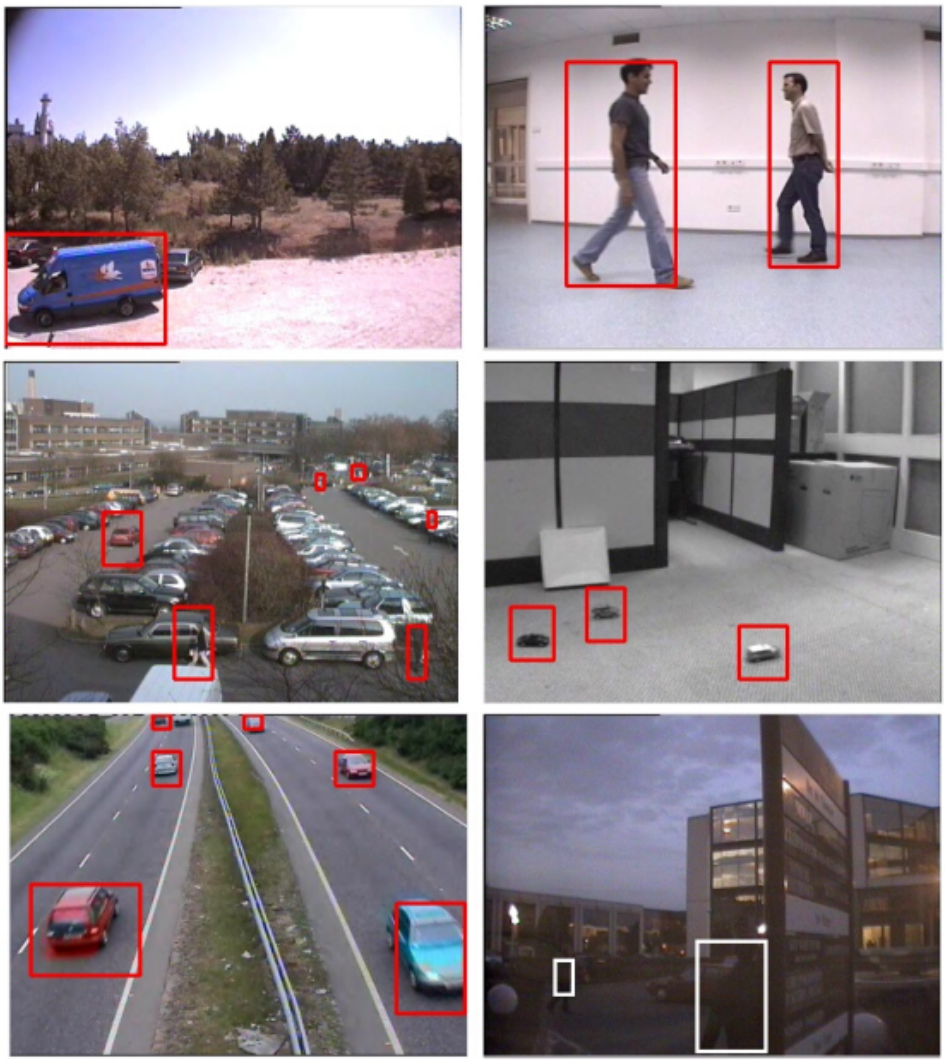

Figure 2.2: Detection results of subband domain method with various object sizes and lighting conditions both indoor and outdoor

can be populated by these images. The size of 2^{nd} low-low images are close to the allocated window size of 256 by 192 in Fig. 3.

Some moving object detection results are shown in Fig. 4 and 5. In all of the sequences, the regions obtained by the methods using 180 by 144 low-low data are tighter than the ones detected by using subband images as expected. This is natural because there is a compromise between location and scale as we go up in the wavelet pyramid, i.e., smaller size images are used in our method. However, this is not important in a video surveillance system because smaller size images are displayed in a regular monitor during 16-channel simultaneous screening. The important issue is to detect moving objects and produce alarms for the security guard watching the surveillance system because guards may get dizzy quickly and may ignore events taking place in front of his or her eyes without an automatic motion detection based alarm system.

Moving objects of various sizes are successfully detected by all three methods as summarized in Tables 1-3. The numbers listed in these tables are the frame numbers of frames in which detection took

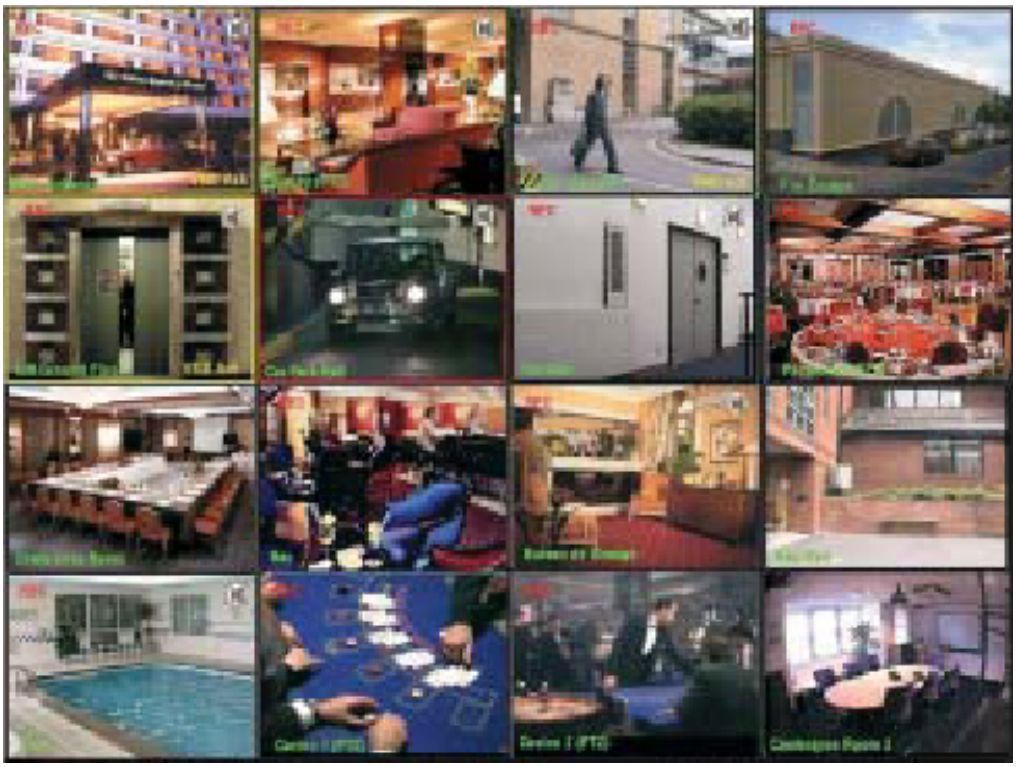

Figure 2.3: A typical surveillance system monitoring 16 channels simultaneously

place. For example, MAN2 object in VIDEO-1 sequence in Table 1 is detected at the 19^{th} frame in all three methods, namely our method utilizing the subband data only, the methods of VSAM [22] and GMM [85].

Motion detection results in videos containing objects with sizes ranging from 20 by 20 to 100 by 100 objects are presented in Table 1. Such large moving objects are detected at the same time by all three methods.

In Table 2, motion detection results of the algorithms with videos containing objects having sizes comparable to 8 pixel by 8 pixel are presented. In these videos, there is not much difference in terms of time delay between the methods. Smaller size objects are not important in a surveillance system with well-placed cameras because they may be moving tree leaves, small birds, animals etc.

Table 3 presents a comparison of the methods with a video called “dark parking lot”. In this video, the system is tested in a parking lot at night. The three methods raise alarms at around the same time instants.

Time performance analysis of the methods are also carried out. The methods of VSAM and GMM are implemented using videos with frame-size of 180 by 144. This image data is extracted from the low-low image of the 2^{nd} level wavelet transform. Our method uses all the coefficients in the 4^{th} level subband image, including low-low, high-low, low-high and high-high subimages. Performance results show that subband domain method is by far the fastest one. Our method processes an image in 1.1*msec*, whereas ordinary VSAM method processes an image in 3.1*msec* and the time for the GMM-based background estimation approach to process an image takes about 28*msec*, on average. It is impossible to process 16 video channels consisting of 180 by 144 size images simultaneously using the VSAM and GMM-based motion detection methods in a typical surveillance system implemented in a PC.

Figure 2.4: Detection results for subband domain(left), VSAM(middle) and GMM based methods in a parking lot sequence recorded in the day-time

Figure 2.5: Detection results for subband domain(left), VSAM(middle) and GMM based methods in a parking lot sequence recorded at night

False alarms occur in all three methods due to leaves and tree branches moving in the wind, etc. In indoor surveillance applications, neither of the three methods produce false alarms. On the other hand, in outdoor surveillance applications, the GMM based method have the least false alarm performance among the three methods studied in this paper as shown in Table 4. This is because it uses color information and it models possible background scenarios in a probabilistic framework. As a result, it is more robust against periodic motion such as motion of leaves. However, it still exhibits false alarms. The GMM based method can be also implemented in the wavelet domain. However, even the wavelet domain implementation is computationally too costly compared to other methods.

Motion sensitivity of our subband domain method can be adjusted to detect any kind of motion in the scene, by going up or down in the wavelet pyramid and playing with the parameter b in equation (2.7). However, by going up to higher resolution levels in the pyramid, the processing time per frame of the subband domain method approaches to that of the ordinary background subtraction method of VSAM. Similarly, false alarms may be reduced by increasing b in (2.7) at the expense of delays in actual alarms.

2.2.5 Conclusion

We developed a method for detecting motion in wavelet compressed video using only subband domain data without performing inverse wavelet transform. Our results assure us that the motion detection performance of the wavelet domain method is almost the same as methods using actual pixel data for motion detection. This is an expected result because subband domain data contains all the necessary information to reconstruct the actual image.

The main advantage of the proposed method compared to regular methods is that it is not only computationally efficient but also it solves the bandwidth problem associated with video processing

systems. It is impossible to feed the pixel data of 16 video channels into the PCI bus of an ordinary PC in real-time. However, compressed video data of 16 channels can be handled by an ordinary PC and its buses, hence real-time motion detection can be implemented by the proposed algorithm.

2.3 Higher order active contours

Author: Ian Jermyn, INRIA-ARIANA

2.3.1 Introduction

The task of image processing algorithms is to assert propositions about images, propositions that typically concern not the images themselves as collections of numbers, but the 'scene' of which the image is a representation. Amongst the many varieties of propositions one can make, one of the most common consists of those formed using the predicate 'The volume that projects to region R in the image domain has properties P'. The central quantity of interest is then the probability distribution $\Pr(\langle R,P\rangle|I,K)$ over these propositions given the image data I and all prior knowledge K. This distribution describes our knowledge of the propositions, from which, if required, point estimates of R and P can be extracted. Attempts to assert such propositions thus have to construct, if only implicitly, a probability distribution on the space of regions, $\Pr(R)$, which may depend on other, known or unknown parameters and the data.

An important issue in such a construction is whether the image domain Ω is regarded as a subset of $\mathbb{Z}^2$ or $\mathbb{R}^2$. The first case includes Markov random fields and various graph-based approaches, while the second is more or less coterminous with active contours, the subject of interest here. In this second case, the technical difficulties involved in constructing probability distributions on the infinite-dimensional space of regions leads to two different approaches. The first avoids an explicitly probabilistic description, and instead defines a 'energy' functional and attempts to minimize it. With some reservations, this can be regarded as the computation of a MAP estimate using the negative logarithm of the probability density. The second approach reduces the dimensionality of the space involved *a priori*, either by restricting attention to finite-dimensional, easily parameterizable subspaces (*e.g.* splines), or by defining a probability distribution on a quotient space defined by certain functionals of the region (*e.g.* low-order moments). In either case, the output is generated by minimizing a functional over some space of regions, or equivalently, in the cases of relevance here, of boundaries. Our work proposes new models of active contours based on the first of the above two approaches, and we begin by looking at previous work in this area.

2.3.1.1 Linear energies

The original paper on active contours was by [48], although the energies used were not well-defined as functionals on regions, since they were parameterization dependent. If the parameterization is taken to be arc length, then the energy used is the sum of boundary length and the integral of boundary curvature, plus the negative of the integral of image gradient magnitude. 'Balloon forces' (a constant pressure, which can be viewed as generated by adding the region area to the energy) were introduced by [21] to improve the stability of results by 'pushing' the region boundary past shallow local minima caused by weak image gradients. 'Geometric' or 'geodesic' active contours [57, 16, 17, 50] removed the parameterization dependence of the early models by using as energy the length of the boundary in a non-Euclidean metric on Ω determined by the image. Most of these energies were written as the integrals of functions over the boundary of the region, but [18], [66], and [44], among others, introduced integrals of functions over the interior to facilitate the description of region properties and to reduce sensitivity to noise and clutter.

All the above energy functionals, both prior and data terms, are representable as algebraic combinations of single integrals over the boundary of the region or over its interior. Such integrals represent *linear* or *twisted linear* functionals on the spaces of 1-boundaries and 2-chains [13]. Chains are equivalence classes of formal linear combinations of differentiable embeddings of rectangles, *e.g.* the interval (1-chains) or the unit square (2-chains). 'Boundaries' in a generalized sense are then defined by the action of a boundary operator ∂ taking n-chains to $(n-1)$-chains. In the plane, 1-boundaries (1-chains in the image of ∂) are equivalent to closed 1-chains (those in the kernel of ∂, and thus without boundary). Consequently, we will reserve the term 'boundary' for the geometric boundary of a region, and use the word 'closed' to indicate boundaries in this generalized sense.

The utility of these formal objects is to characterize properties of curves and curve functionals in algebraic terms. A functional on chains is 'linear' in the standard sense: given a linear combination of chains $\alpha C_1 + \beta C_2$, the value of the functional is given by the same linear combination of the values of the two chains:

$$E(\alpha C_1 + \beta C_2) \quad = \quad \alpha E(C_1) + \beta E(C_2) \ . \tag{2.10}$$

Note that by definition two embeddings C_1 and C_2 with the same domain D represent the same chain if $C_2 = C_1\varepsilon$, for some diffeomorphism $\varepsilon : D \to D$. Functionals defined on the space of embeddings must therefore be invariant under diffeomorphisms in order to project to well-defined functionals on chains. This invariance requirement means that differential forms are the natural language in which to represent such functionals. Linear functionals on 1-chains thus take the form

$$E(C) \quad = \quad \int_{\partial R} A = \int_{\mathrm{dom}\, C} C^* A = \int dp\, \vec{\mathbf{t}}(p) \cdot A \ , \tag{2.11}$$

where A is a 1-form on Ω; $v \cdot A$ denotes the evaluation ('inner product') of the 1-form A on the vector v; $\mathrm{dom}\, C$ is the domain of C; C^* is pullback by C; p is a coordinate on $\mathrm{dom}\, C$, and $\vec{\mathbf{t}}(p) = C'(p)$ is the tangent vector to C at p. Using the generalized Stokes theorem, such functionals can be rewritten as integrals over R. Equally importantly, since in two dimensions every 2-form is closed and in the plane every closed form is exact because the cohomology is trivial, the reverse is true. For every 2-form F there exists a 1-form A_F such that $F = dA_F$, meaning that every energy of the form $\int_R F$, where R is a region, or more generally a 2-chain, can be rewritten as

$$\int_R F \quad = \quad \int_R dA_F = \int_{\partial R} A_F \ . \tag{2.12}$$

The area of the interior of a closed 1-chain provides one example of this process. In this case, $F = \star_g \mathbb{I}$, where $\mathbb{I}$ is the function identically equal to one everywhere. In an Euclidean metric, this becomes

$$E(C) \quad = \quad \frac{1}{2} \int dp\, \vec{\mathbf{t}}(p) \times C(p) = \int dp\, \frac{\partial x}{\partial p} y(C(p)) \ , \tag{2.13}$$

where (x, y) are Euclidean coordinates. In consequence of equation (2.12), linear energies of the form (2.11) encompass all the forms of region energies in the literature. They are also used by [46] as part of a 'ratio energy', and by [97] to find 'flux maximizing flows'.

Rather than define twisted linear functionals in general, we simply give the form appropriate to our context:

$$E(C) \quad = \quad \int_{\mathrm{dom}\, C} \star_{C^* g}\, C^* f = \int dp\, |\vec{\mathbf{t}}(p)|_g \ f(C(p)) \ . \tag{2.14}$$

where g is a metric on Ω, and f is a function (0-form) on Ω. $C^* g$ is therefore the metric on $\mathrm{dom}\, C$ induced by C; and $\star_{C^* g}$ is the associated Hodge operator. $|v|_g$ is the norm of the vector v in the metric g. The form

of functional in equation (2.14) encompasses the remainder of the models mentioned above, including geometric and geodesic active contours, and most others that have appeared in the literature. A particular example is boundary length, in the metric g, which is given by $f = \mathbb{I}$.

In the particular case of prior terms, much more can be said. Prior terms should be Euclidean invariant in general. This forces f to be constant, g to be Euclidean, and A to calculate the interior area. Thus there are only two linear prior terms compatible with Euclidean invariance: length and area.

2.3.1.2 Shape modelling

The limitation of the functionals described above is that they incorporate only local interactions. In the case of a finite-dimensional vector space X, this is clear. Linear functionals $x \cdot a$, $x \in X$, $a \in X^*$, lead to exponential probability distributions, $\Pr(x|a) \propto \exp(-x \cdot a)$. In any basis, this takes the form $\exp(-\sum_i x^i a_i) = \prod_i \exp(-x^i a_i)$. Thus x^i and x^j are independent for all $i \neq j$. The same is true in a function space, where a linear functional is represented by the integral of the function against a measure, $\int d\mu(p) f(p)$.

For linear functionals on the space $C_1(\Omega)$ of 1-chains in Ω, the situation is similar, except that it is important to realize that the equivalent of the indices i or the points p in this case are the tangent vectors $\vec{\mathbf{t}}(p)$. The situation for twisted linear functionals is complicated by the fact that the metric on $\mathrm{dom} C$ is induced by the embedding. The result is that the functional incorporates local interactions in the sense of a function space, where local functionals may be integrals of derivatives of the function at each point as well as of its value. This notion of locality is closely related to the property of Markovianity. In the discrete case, the dependence on derivatives means that interactions take place within fixed size neighbourhoods, as in a Markov random field. In addition, because the degree of the derivatives involved is typically small, the neighbourhoods are small. Thus for equation (2.14), the interaction is between 'neighbouring pairs' of tangent vectors.

The result of this limitation is impoverished modelling, especially in the prior terms. Any two boundaries that share length and area are equiprobable from the point of view of these models. Looking at the energy minima confirms this impression. It is well known that gradient descent using the length leads to evolution by curvature and that this evolution moves the boundary towards a circle that then shrinks and disappears. The limitation imposes itself equally on data terms, although there the lack of Euclidean invariance allows a wider variety of terms.

In order to get around this limitation, various approaches have been taken to the incorporation of more sophisticated information. This has usually been done within the second of the two frameworks outlined in section 2.3.1: the use of an *a priori* finite-dimensional space. [54] represent shapes as signed distance functions, and use a Gaussian distribution on the principal components of variation around the mean distance function acquired from training data as a shape prior. [25] modify the Mumford-Shah functional to incorporate statistical shape knowledge. They use an explicit parameterization of the contour as a closed spline curve, and learn a Gaussian probability distribution for the spline control point vectors. The statistical prior restricts the contour deformations to the subspace of learned deformations. [67] propose a functional that can account for the global and local shape properties of the target object. A prior shape model is built using aligned training examples. A probabilistic framework uses the shape image and the variability of shape deformations as unknown variables. They seek a global transformation and a level set representation that maximizes the posterior probability given the prior shape model. [19] define an energy functional depending on the gradient and the average shape of the target object. The prior shape term evaluates the similarity of the shape of the contour (modulo scale, rotation and translation) to that of the reference shape through the computation of a distance function using the Fast Marching method of [80]. Finally, [33] define shape descriptors with Legendre moments and introduce a geometric prior

in the framework of region-based active contours, with a quadratic distance function between the set of moments of the contour and the set of moments of the reference object.

2.3.1.3 New models

What the above models have in common, is that they are looking for a single instance of a specific shape in an image. Given one or more training examples, and a shape representation, a 'mean' shape is computed. The evolution of the contour is then constrained by this 'mean' shape and the possible deformations around this shape. This is effective in some circumstances, but these approaches rapidly become restrictive if there are several instances of the shape to detect in the image, or if the regions to be extracted cannot be defined as small variations around a 'mean' shape.

Consider the example of 'networks'. These possess complex geometric properties in common (they are composed of 'arms' of roughly parallel sides, perhaps of varying width, joined together in various ways), but their variability cannot be reduced to perturbations of a template shape parameterized by a few quantities. Nevertheless it is clearly important from a modelling point of view to incorporate the geometrical properties that they share; what might be called their 'family resemblance'.

With the aim of modelling such families, and of extending the expressive power of active contour models more generally by introducing a coherent way to construct functionals of increasing complexity, we propose a new class of active contour models. These generalize the linear models of section 2.3.1.1 to *higher-order polynomial energies* on the space of 1-chains. These models describe arbitrarily long-range interactions between subsets of points in the boundary: quadratic energies describe interactions between pairs of points, cubic energies between triples, and so on. These interactions in their turn allow the incorporation of non-trivial geometric information into prior terms, and in particular the description of shape families such as networks. If used as data terms, they allow the description of more complex relations between the region and the image.

The new energies require new minimization techniques. The basic methodology is still gradient descent, but its implementation is significantly harder. Higher-order energies lead to non-local forces: the speed of a point in the boundary depends on the whole of the boundary and not just on its infinitesimal neighbourhood. The computation of the evolution thus involves integrals over the boundary. We use a level set approach to the problem, and extend standard methods to handle non-local forces in a way similar to, but necessarily more precise than, that used for incompressible flows.

In section 2.3.2, we present higher-order active contour energies in detail. In section 2.3.3, we describe the extended level set method we use to minimize the energy. In section 2.3.4, we introduce image terms, apply a quadratic energy functional to the extraction of road networks, and present results on real images. We conclude in section 2.3.5.

2.3.2 Higher-order energies

The new models make use of the linear structure of the chain space, which allows us to go beyond linear functionals to polynomial functionals in a clear and structured way. This can be thought of as a coherent way of generating functionals of increasing complexity, or as the expansion of an arbitrary functional. Such functionals have not been considered before, and their use constitutes a major generalization of the active contour approach.

Since polynomials are sums of monomials, it is sufficient to construct these. A monomial function of order n on a vector space V is the composition of three maps:

$$V \xrightarrow{\Delta_n} V^n \xrightarrow{\otimes} V^{\otimes n} \xrightarrow{E} \mathbb{R} \tag{2.15}$$

where: Δ_n is the diagonal map from V to its n-fold Cartesian product V^n; $\otimes$ is the projection from this latter space to the n-fold tensor product of V, $V^{\otimes n}$; and E is a linear functional on the latter. Note that setting $V = \mathbb{R}$ gives normal monomials, ax^n, $x \in \mathbb{R}$. In our context, $V = \mathcal{C}_1(\Omega)$, the space of 1-chains in Ω, and our task boils down to constructing linear functionals E on tensor products of $\mathcal{C}_1(\Omega)$ with itself. Fortunately, $\mathcal{C}_1(\Omega)^{\otimes n}$ is a subspace of $\mathcal{C}_n(\Omega^n)$, the space of n-chains in Ω^n, so that a linear functional on the latter is also a linear functional on the former. Linear functionals on the latter are easy to create however. One can proceed in several ways, one of which is analogous to equation (2.14), while another is analogous to equation (2.11). We do not describe all the possibilities here for lack of space, but instead focus on the latter. Given an n-form F on Ω^n, we pull it back to the domain of $C^{\otimes n}$ and integrate it, using the analogue of equation (2.11):

$$E(C) \quad = \quad \int_{(\partial R)^n} F = \int_{(\mathrm{dom}C)^n} (C^{\otimes n})^* F \quad . \tag{2.16}$$

What is then required is an n-form on Ω^n. In what follows, we will focus on the quadratic case, $n = 2$, both for clarity and because this is what we will use in the application later in the paper.

2.3.2.1 Quadratic energies

In the case $n = 2$, equation (2.16) becomes

$$E(C) \quad = \quad \int_{(\partial R)^2} F = \int_{(\mathrm{dom}C)^2} (C \otimes C)^* F \quad . \tag{2.17}$$

The product structures of $C \otimes C$ and $(\mathrm{dom}C)^2$ mean that this functional can always be written (in terms of coordinates (p, p') on $(\mathrm{dom}C)^2$) as

$$E(C) \quad = \quad \int\int dp\, dp'\, \vec{\mathbf{t}}(p) \cdot F(C(p), C(p')) \cdot \vec{\mathbf{t}}(p') \quad , \tag{2.18}$$

where $F(x, x')$, for each $(x, x') \in \Omega^2$, is a matrix. The operator F allows us to model a non-trivial interaction between different contour points. Note that this interaction is not Markov, even if the value of F tends to zero rapidly with increasing distance between its arguments. Since the interaction is mediated by the embedding rather than the embedded space, interactions can occur between arbitrarily separated pieces of the contour if they approach each other in Ω. Note also that the force derived from equation (2.17) is non-local, the force at a point being determined by an integral over the contour.

For prior terms, when the 2-form F does not depend on the image, we require the energy to be Euclidean invariant. This results in the form

$$E(C) \quad = \quad -\int\int dp\, dp'\, \vec{\mathbf{t}}(p) \cdot \vec{\mathbf{t}}(p')\, \Psi(|C(p) - C(p')|) \quad , \tag{2.19}$$

where $|x - y|$ is the Euclidean distance between points x and y in Ω. The function Ψ weights the interactions between different points of the curve according to their distance, and must be chosen carefully since it defines the geometrical content of the model. Ψ can be chosen to tend to zero as x tends to infinity, since adding a constant to Ψ adds zero to the energy. It should also be chosen so that the integral converges.

It is clear from equation (2.19) that even in the quadratic case, the use of higher-order energies opens up a much wider range of modelling possibilities than previously possible. With linear energies, only two prior terms existed; now there is a whole function space full. Note also that unlike the shape models described in section 2.3.1.2, the new energies incorporate Euclidean invariance naturally without requiring the estimation of position or rotation, since they are not mixture models over these variables. Note, however, that this does not constrain the minimum energy configurations to be Euclidean invariant, although the set of such minima will be; the symmetry is 'broken' in general.

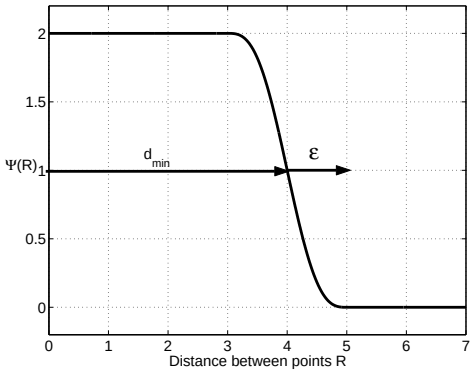

Figure 2.6: The function Ψ

2.3.2.2 An example of a geometric quadratic energy

In this section, we study a particular case of an Euclidean invariant quadratic energy. We will use this particular case later on to model road networks, but we use it here to illustrate the possibilities inherent in higher-order energies.

The energy we choose takes the form

$$E_g(C) \quad = \quad \mathcal{L}(C) + \alpha \mathcal{A}(C) - \beta \int \int dp\, dp'\, \vec{\mathbf{t}} \cdot \vec{\mathbf{t}}'\, \Psi(R(p,p')) \ , \tag{2.20}$$

where $\mathcal{L}$ is the length of the boundary in the Euclidean metric on Ω, an energy of the form (2.14); and $\mathcal{A}$ is the area of its interior, an energy of the form (2.11). $R(p,p') = |C(p) - C(p')|$ is the Euclidean distance between $C(p)$ and $C(p')$. The length term acts as a regularizer. The area term is introduced to control the expansion of the region. The Euclidean invariant quadratic term, of the form (2.19), introduces the interactions. We choose the following form for the function Ψ:

$$\Psi(x) \quad = \quad \begin{cases} 1 \ \text{ if } \ x < d_{\min} - \varepsilon \ , \\ 0 \ \text{ if } \ x > d_{\min} + \varepsilon \ , \\ \frac{1}{2}(1 - \frac{x - d_{\min}}{\varepsilon} - \frac{1}{\pi} \sin(\pi \frac{x - d_{\min}}{\varepsilon})) \ \text{ otherwise} \ . \end{cases} \tag{2.21}$$

This function is shown in figure 2.6, where the parameters $d_{\min}$ and ε are also illustrated. A point p on the contour interacts with other points within a certain distance $d_{\min} + \varepsilon$, measured in Ω. The function Ψ is always positive, and so from equation (2.20), the quadratic part of the energy is a minimum when the points interacting with one another have parallel tangent vectors. The quadratic energy thus favours straight lines. On the other hand, for pairs of points with antiparallel tangent vectors, the quadratic part of the energy is zero unless the points approach closer than a distance of $d_{\min} + \varepsilon$, when it starts to increase rapidly. The quadratic energy therefore acts as a softened 'hard-core' potential, preventing the points from approaching much closer than $d_{\min}$.

The energy in equation (2.20) is minimized using gradient descent. Thus the contour evolution is determined by

$$\frac{\partial C}{\partial t} \quad = \quad -\frac{\delta E}{\delta C}(C) \ , \tag{2.22}$$

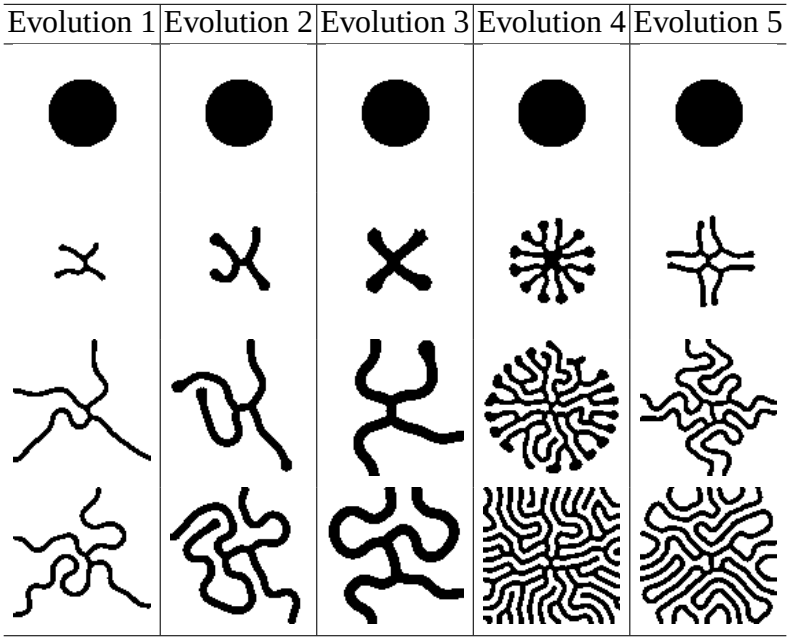

Figure 2.7: Examples of gradient descent using the energy in equation (2.20). The first three columns correspond to different values of $d_{\min}$, while the last two correspond to different values of α.

where $\delta E/\delta C$ is the functional derivative of E with respect to C.[1] The resulting descent equation is then

$$\hat{\mathbf{n}}\cdot\dot{C}(p) \quad = \quad -\kappa(p)-\alpha+2\beta\int dp'\ (\hat{\mathbf{R}}(p,p')\cdot\hat{\mathbf{n}}(p'))\ \Psi'(R(p,p'))\ , \tag{2.23}$$

where $\hat{\mathbf{R}}(p,p') = (C(p)-C(p'))/|C(p)-C(p')|$. The component of $\partial C/\partial t$ along the normal has been taken, movement along the tangent direction being equivalent to a diffeomorphism of the domain of C, and thus irrelevant. The energy minima that result consist of elongated structures ('arms') of a fixed minimum width that tend to elongate. The arms are mutually repulsive, so that they distribute themselves over the domain Ω, and have a limited branching number.

Figure 2.7 shows examples of evolutions starting from a circle using equation (2.23). All the evolutions show the formation of fingered structures with parallel-sided arms of constant width. The width is controlled by the parameter $d_{\min}$ in the Ψ function, and the first three rows of figure 2.7 show evolutions for different values of this parameter ($d_{\min} = 3,5,7$); the fingers formed are indeed of the correct width. The last two rows illustrate the role of the parameter α. In the fourth row, $\alpha = 0.05$, while $\alpha = 0.1$ in the fifth row. The larger the value of α, the fewer the number of arms that form at the beginning of the evolution.

The growth away from a circle towards a labyrinthine structure with elongated 'arms' can be understood as follows. A linear analysis of the stability of the circle to small sinusoidal perturbations shows that above a certain angular wavelength, the perturbations, rather than being damped back to zero, are

[1] Note that strictly speaking, $\delta E/\delta C$ are the components of a 1-form on the space of boundaries, and that to generate a gradient we should map it to the tangent bundle using a metric. By using it as is, we are effectively assuming that the metric on the space of boundaries is Euclidean in the point basis; this is common practice. The choice of a metric is difficult; there are good arguments for saying that it should be determined not *a priori*, but by the measurements we intend to make, that is, by the likelihood in Bayes' theorem [45].

amplified, their size and their spatial frequency around the initial circle being controlled by the Ψ function. Thus instead of smoothing all irregularities, as in the linear case, this energy allows some of them to develop, and hence encourages complex shapes. An uncontrollable instability at all frequencies is prevented by the fact that the 'bumps' corresponding to two peaks in the sinusoid cannot approach closer than $d_{\min}$. Once created, the bumps elongate into arms with parallel sides, thus decreasing the energy, although this nonlinear behaviour can no longer be described within the linear approximation used to study stability. In an infinite domain it seems likely that the energy is not bounded below, and that the arms will continue to grow and to ramify indefinitely. In a finite domain such as an image, this cannot happen due to the repulsion between the arms.

The experiments serve to illustrate the greater complexity of information contained within quadratic energies as compared to linear energies, and to show that the specific energy in equation (2.20) is well-suited to modelling network structures.

2.3.3 Minimization of the energy

It is important to realize that although the linear space of chains is useful for constructing and describing functionals, the minimization of the energy takes place not over the space of (closed) 1-chains, but over the space of region boundaries.

In order to minimize the energy, we use gradient descent, evolving the contour using the level set framework introduced by [64]. Level set representations handle changes of topology naturally, are parameter free, and allow the simple expression of geometrical quantities like curvature. Instead of evolving the contour itself, a function of higher dimension ϕ is used to represent the contour, and the function is evolved. The representing function ϕ is defined as the signed distance function to the contour C:

$$\phi(x) = \pm d(x, C) \ , \tag{2.24}$$

where the plus (minus) sign is chosen if the point x lies inside (outside) the contour. The inverse of this map from contours to functions is

$$C = \{x | \phi(x) = 0\} \ . \tag{2.25}$$

Supposing that the contour changes with time t, differentiating the defining equation for ϕ gives

$$\phi(C(p,t),t) = 0 \Rightarrow \nabla\phi|_{(C(p,t),t)} \cdot \frac{\partial C}{\partial t}(p,t) + \left.\frac{\partial \phi}{\partial t}\right|_{(C(p,t),t)} = 0 \ . \tag{2.26}$$

If the contour propagates along the outward normal direction with speed F, *i.e.* $\hat{\mathbf{n}} \cdot \dot{C}(p) = F[C](p)$, then the level set function on the contour must obey

$$\dot{\phi} \quad = \quad -\nabla\phi \cdot F\hat{\mathbf{n}} = F\nabla\phi \cdot \nabla\phi/|\nabla\phi| = F|\nabla\phi| \ . \tag{2.27}$$

Note that in principle, the rest of ϕ should evolve as given in equation (2.24). In practice however, since the exact evolution of ϕ away from the contour is of no consequence, other recipes are used. In particular, it is possible to apply the expression for F to each level set, and evolve the function ϕ accordingly, reinitializing if necessary to maintain equation (2.24).

As can be seen from equation (2.23), the evolution equations derived from quadratic energies contain nonlocal terms, and this creates new difficulties. Following the procedure of applying the expression for F to every level set is impractical, since it means extracting the level set belonging to each point of the discretized version of Ω and integrating over it. In order to construct the speed at all points of Ω, we

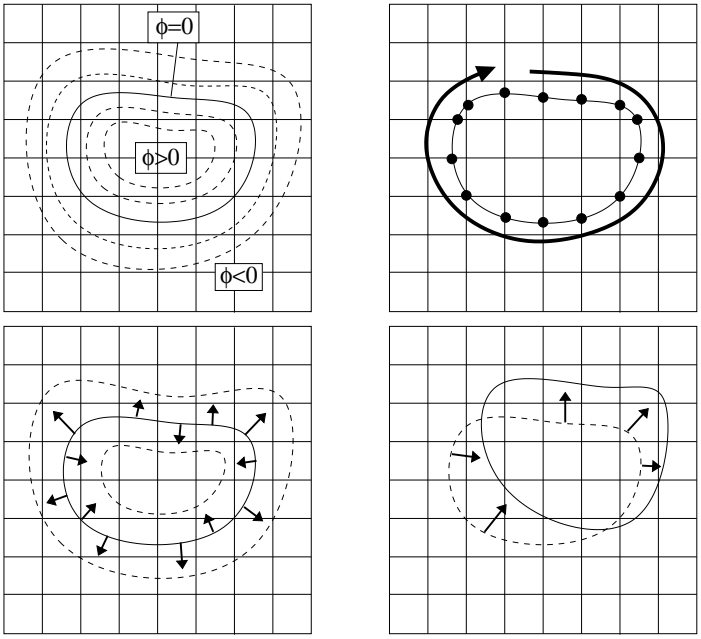

Figure 2.8: The four evolution steps. Top-left, step (1): (re)initialization; top-right, step (2): contour extraction and computation of the speed on-contour; bottom-left, step (3): extension of the speed to the Narrow Band; bottom-right, step (4): evolution of ϕ.

therefore use the technique of 'extension velocities' [1]. This leaves the problem of calculating the speed on the zero level set. To do this, we first extract the zero level set using ENO interpolation [82] and contour tracing [68], and then compute the speed using numerical integration over the contour. The level set function is thus evolved in four steps. First ϕ is (re)initialized, then the zero level set is extracted and the speed computed. The speed is then extended from the zero level set to Ω, and finally ϕ is updated. Figure 2.8 depicts the four steps. In the next few subsections, we describe these steps in more detail.

2.3.3.1 (Re)initialization

In order to (re)initialize ϕ as a signed distance function, we use the approach described by [91], where the PDE

$$\begin{aligned} \phi(p,0) &= \phi_0 \ , \\ \phi_t &= \text{sign}(\phi_0)\ (1-|\nabla\phi|) \end{aligned} \tag{2.28}$$

is solved for this purpose. We found, however, that the zero level set moved during the numerical solution of this equation, an effect which manifested itself as a loss of area when we attempted to simulate an area-preserving flow for example. This is a recognized problem, to which [90] have proposed a solution. A local area conservation constraint is imposed by modifying equation (2.28) in each cell Ω_{ij} of Ω to

$$\phi_t = \text{sign}(\phi_0)\ (1-|\nabla\phi|) + \lambda_{ij} H'(\phi)|\nabla\phi| \ , \tag{2.29}$$

where

$$\lambda_{ij} = \frac{-\int_{\Omega_{ij}} H'(\phi)\text{sign}(\phi_0)\ (1-|\nabla\phi|)}{\int_{\Omega_{ij}} H'(\phi)^2|\nabla\phi|} \ . \tag{2.30}$$

The initial condition for equation (2.29) is the current value of ϕ, except for the initialization of the evolution, when ϕ is set to $+1$ inside the contour and -1 outside.

2.3.3.2 Contour extraction and computation of F on the contour

In order to compute accurately the speed F on the zero level set, we first locate the intersections of this set with the discrete grid using Essentially Non Oscillatory (ENO) interpolation [82], as described in table 2.3.3.2.

Table 2.5: ENO interpolation algorithm.

1. Construction of a first-order polynomial $P_{j+1}^{f,1}(x)$ and initialization of the first window for the point j ($l=1$):

$$P_{j+1/2}^{f,l}(x) = f[x_j] + f[x_j, x_{j+1}](x - x_j)$$
$$k_{min}^{l} = j$$

2. $l = l+1$.

3. If $P_{j+1}^{f,l-1}(x)$ and k_{min}^{l-1} are defined:

$$P_{j+1/2}^{f,l}(x) = P_{j+1/2}^{f,l-1}(x) + c^l \prod_{i=k_{min}^{l-1}}^{i=k_{min}^{l-1}+l-1}(x - x_j)$$

where

$$c^l = \begin{cases} b^l & \text{si } |a^l| \geq |b^l| \\ a^l & \text{sinon} \end{cases}$$

$$k_{min}^{l} = \begin{cases} k_{min}^{l-1} - 1 & \text{si } |a^l| \geq |b^l| \\ k_{min}^{l-1} & \text{sinon} \end{cases}$$

and

$$a^l = f[x_{k_{min}^{l-1}}, \cdots, x_{k_{min}^{l-1}+l}]$$
$$b^l = f[x_{k_{min}^{l-1}-1}, \cdots, x_{k_{min}^{l-1}+l-1}]$$

f[...] represents the Newton divided differences.

After interpolation, the boundary is extracted using the contour tracing algorithm shown in table 2.7 [68]. At each step, we start from the current point and consider six possible directions for the next point. These directions are adapted to the different possible configurations, as shown in figure 2.9. We obtain an ordered set of points $\{C(p_i);\, i = 1, \ldots, n\}$ representing the boundary.

In fact, the situation is a little more complicated than this, because some configuration are ambiguous, as illustrated in the bottom part of figure 2.9. To deal with these, it is necessary to adopt a convention: either the interior or the exterior, but not both, can have subcellular width. We choose the former.

Table 2.6: Newton divided differences.

$$
\begin{aligned}
f[x_j] &= f(x_j) \\
f[x_j, x_{j+1}] &= \frac{f[x_{j+1}] - f[x_j]}{x_{j+1} - x_j} \\
f[x_j, \cdots, x_{j+2}] &= \frac{f[x_{j+1}, x_{j+2}] - f[x_j, x_{j+1}]}{x_{j+2} - x_j} \\
f[x_j, \cdots, x_{j+k}] &= \frac{f[x_{j+1}, \cdots, x_{j+k}] - f[x_j, \cdots, x_{j+k-1}]}{x_{j+k} - x_j}
\end{aligned}
$$

Table 2.7: Tracing algorithm.

1. Choose a starting point A in a set of points R. Set current point $C = A$ and search direction $S = 6$.
2. While C is different from A or *first*$= 1$, do steps 3 to 9.
3. *found*$= 0$.
4. While *found*$= 0$, do steps 5 to 8, at most 3 times.
5. If B, the neighbour $(S-1)$ of C is in R; $C = B$, $S = S-2$, *found*$= 1$.
6. Else, if B, the neighbour S of C is in R, $C = B$ and *found*$= 1$.
7. Else, if B, the neighbour $(S+1)$ of C is in R, $C = B$ and *found*$= 1$.
8. Else $S = S+2$.
9. *first*$= 0$.

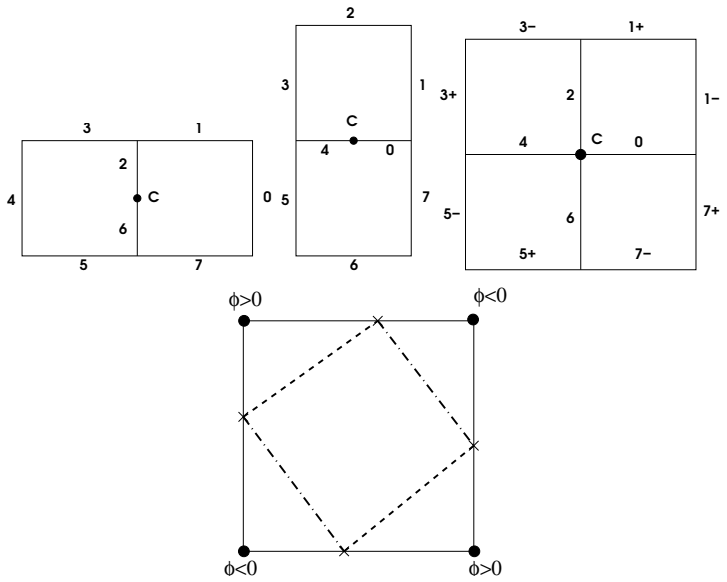

Figure 2.9: Top: the three configurations encountered in the contour tracing algorithm. Bottom: an ambiguous configuration.

Having extracted the boundary, and after interpolating the necessary values from the grid, we compute the speed F for each extracted point by performing a numerical integration over the contour.

2.3.3.3 Computation of F on all points of the domain

The speed is needed for all points of Ω. As mentioned at the beginning of this section, in order to do this efficiently, we use the method of 'extension velocities', as proposed by [1]. To initialize the process, the grid points closest to the extracted boundary inherit the speed of the closest extracted boundary point. We then solve the PDE

$$F_\tau + \text{sign}(\phi)\frac{\nabla\phi}{|\nabla\phi|}\cdot\nabla F = 0 \ .$$

The solution of this equation satisfies $\nabla\phi\cdot\nabla F = 0$, which means that the variation of F along the normal to the level sets is null. Thus every level set evolves with the same speed, and the distance between each level set is preserved in principle.

2.3.3.4 Evolution of ϕ

In practice, it is not necessary to compute the evolution of the level set function over the whole of Ω. Computational efficiency can be increased by restricting computation to a band around the zero level set, known as the 'Narrow Band' [81], defined by $\phi(x,y) < t$, where t is a threshold. When the zero level set comes too close to the edge of the Narrow Band, the level set function is reinitialized as described above, and the Narrow Band is reconstructed.

2.3.4 Application: line network extraction

Automatic detection of line networks, and especially of road networks, in satellite and aerial imagery has been studied for the last fifteen years at least. Motivated by the increasing rate of data acquisition and the growing importance of geographic information systems, a wide variety of methods have been developed to attack this problem. Despite all this attention, extraction of line networks remains a challenge because of the great variability of the objects concerned, and the consequent difficulty in their characterization. The intensity of a road can vary significantly from one road to another, for example, while the presence of trees and buildings ('geometric noise') in high resolution data can obscure the network; junctions can be highly complex; networks do not possess exactly the same properties in rural and urban areas; and so on.

We can distinguish different categories of methods for detecting line networks. Some aim at extracting the network as a one-dimensional object, whereas others extract the network as a region. In addition, methods may restrict the network topologies that can be found. The first category includes methods for finding the optimal path between two endpoints, either defined by the user or found automatically. [30], for example, combine the results of applying several specially designed operators into an array of costs inversely related to the likelihood of the presence of a road, and then find an optimal path through this array. [59] define a path cost depending on the contrast, grey-level and curvature along a path between two endpoints, and then minimize the cost using dynamic programming. [37] propose a tree search method for road tracking based on reducing as much as possible the uncertainty in the road position. [95] first generate a number of candidate line segments using two different line detectors. The segments are then connected together using a Markov random field defined on a graph with vertices the segments, thus allowing more complex topologies. [88] and [52] model thin networks, including roads, as ensembles of line segments embedded in the image domain. Marked point processes (with line segments as marks) control network parameters such as connectivity and curvature via interactions between the segments.

All these methods find a connected set of points or segments, but do not extract the borders of the road (although this is certainly possible with marked point processes). With increase of image resolution, the width of networks in images can become significant, and it then makes more sense to consider the network as a region. Methods have been proposed that specifically take into account the width of the roads to be extracted. [5] propose an automatic approach that first finds MAP estimates of the road configuration in small windows using dynamic programming, and then combines these window estimates, again using dynamic programming. The model used explicitly includes the road borders. [62] introduce 'ziplock snakes'. From an initial and a final point, forces derived from the image are progressively used to adjust the position of the active contour. The endpoints are positioned on either side of the road, and both borders of the road are extracted, but the topology is limited to linear structures. [34] and [53] model roads using 'ribbon snakes', active contours with a certain width associated to each point, again extracting linear structures. The method we propose lies in this second category: we extract the network as a region of arbitrary topology.

2.3.4.1 Proposed model

The model has to take into account two fundamental aspects of the entity to be detected: the geometry and the radiometry, corresponding to prior and likelihood terms. The energy thus contains two parts:

$$E(C) \quad = \quad E_g(C) + \lambda E_i(C) \ , \tag{2.31}$$

where λ balances the contributions of the geometric part E_g and the data part E_i. The geometric part E_g is given by equation (2.20), and is described in section 2.3.2.2.

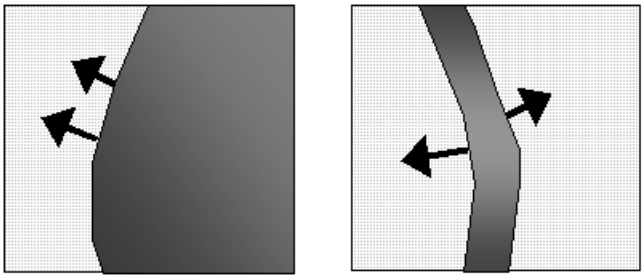

Figure 2.10: The two configurations favoured by the quadratic image term.

The image part E_i is composed of two terms:

$$\begin{aligned} E_i(C) &= \int_{\partial R} \star dI - \int_{(\partial R)^2} (\Psi \circ R)(dI \star dI') \\ &= \int dp\, \hat{\mathbf{n}} \cdot \nabla I - \int\int dp\, dp'\, \vec{\mathbf{t}} \cdot \vec{\mathbf{t}}'\, (\nabla I \cdot \nabla I')\, \Psi(R(p,p'))\ , \end{aligned} \tag{2.32}$$

where we use primed and unprimed variables to designate quantities evaluated at points p (or $C(p)$) and p' (or $C(p')$) respectively. The first, linear term has the form (2.11), while the quadratic term takes the general form (2.17).

The linear term favours situations in which the outward normal is opposed to the image gradient, or in other words, in which the road is lighter than its environment. When this is the case, it also favours larger gradients under the contour. The second term is an example of a quadratic data term: it describes a relation between the contour and the data that cannot be incorporated into a linear functional. Its effect is to favour the two situations illustrated in figure 2.10. First, it favours configurations in which pairs of points whose tangent vectors are parallel and that are not too distant from each other (*i.e.* points on the same side of a road) lie on image gradients that point in the same direction and are large. Second, it favours configurations in which pairs of points whose tangent vectors are antiparallel (*i.e.* points on opposite sides of a road) lie on image gradients that point in opposite directions and are large. This latter is important, as it allows the model to capture the *joint* behaviour exhibited by the opposing sides of a road.

The energy in equation (2.31) is minimized using gradient descent implemented via level sets as described in section 2.3.3. The resulting descent equation is

$$\begin{aligned} \hat{\mathbf{n}} \cdot \frac{\partial C}{\partial t} &= -\kappa - \lambda \nabla^2 I - \alpha + 2\lambda \int dp'\, (\nabla I' \cdot \nabla \nabla I \cdot \hat{\mathbf{n}}')\, \Psi(R(p,p')) \\ &+ 2 \int dp'\, (\hat{\mathbf{R}} \cdot \hat{\mathbf{n}}')(\beta + \lambda \nabla I \cdot \nabla I')\, \Psi'(R(p,p'))\ . \end{aligned} \tag{2.33}$$

2.3.4.2 Experimental results

We tested the above model on real satellite and aerial images. Two such images are shown in the first column of figure 2.11. The images present several difficulties. There are regions of high gradient corresponding to the borders of fields rather than to roads, and fields also exhibit parallel sides. In the first image, there is a discontinuity in the road. The gradient descent procedure and results are shown in the second to fifth columns of figure 2.11. In both images, the roads are perfectly extracted.

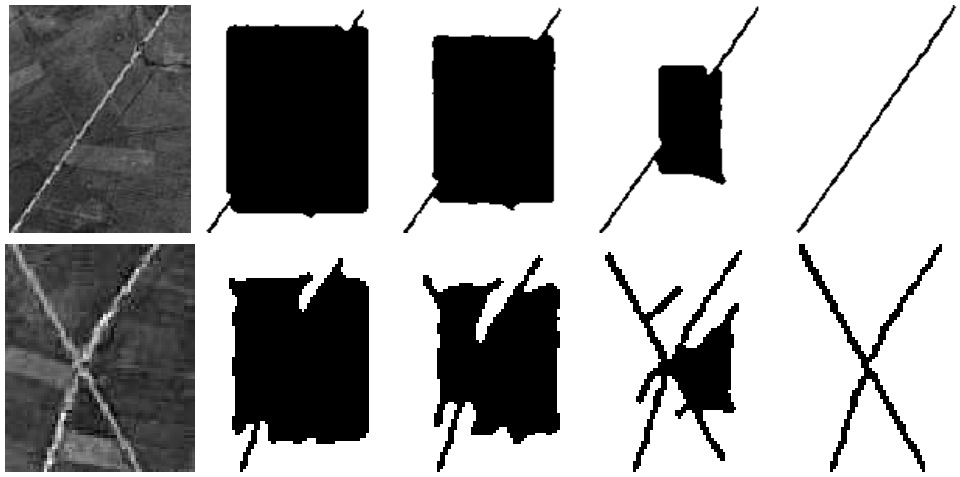

Figure 2.11: Gradient descent on the two SPOT satellite images in the first column.

Figure 2.12 shows another result on a larger, more complex piece of the same satellite image. The result is not perfect but very encouraging. We are able to detect both straight and 'windy' portions of the network, and areas where the road width varies significantly.

The data term, although it takes into account some aspects of the appearance of road networks in images, can nevertheless be improved. For instance, isolated edges are occasionally detected. In the next section, we add another image term to our model, more specific to the radiometry of a line network.

2.3.4.3 A more specific image term

Consider a function G on Ω that is representative of the entity to detect, in our case the line network. For instance, it could be the log probability that each point of Ω belongs to the network. Then one can define the following energy of the form (2.11) and (2.12):

$$E(C) \quad = \quad -\int_R \star G = -\int_{\partial R} A_G = -\int_{\mathrm{dom}\, C} C^* A_G \ , \tag{2.34}$$

where $dA_G = \star G$. The functional derivative is given by

$$\frac{\delta E}{\delta C(p)} = G(C(p))\hat{\mathbf{n}}(p) \ . \tag{2.35}$$

In the following subsections, we describe two ways of constructing a suitable function G. The first method uses oriented filtering, whereas the second uses hypothesis tests.

Oriented filtering Define the function

$$\mathcal{F}_\theta \quad = \quad (\hat{\mathbf{v}}_\theta \cdot \nabla)^2 N_\sigma \ ,$$

where N_σ is a rotationally symmetric Gaussian with standard deviation σ, and $\hat{\mathbf{v}}_\theta$ is the unit vector in direction θ. Then G is given by

$$G(x) \quad = \quad Q(\min_{\theta \in \Theta}(\mathcal{F}_\theta * I(x))) \ ,$$

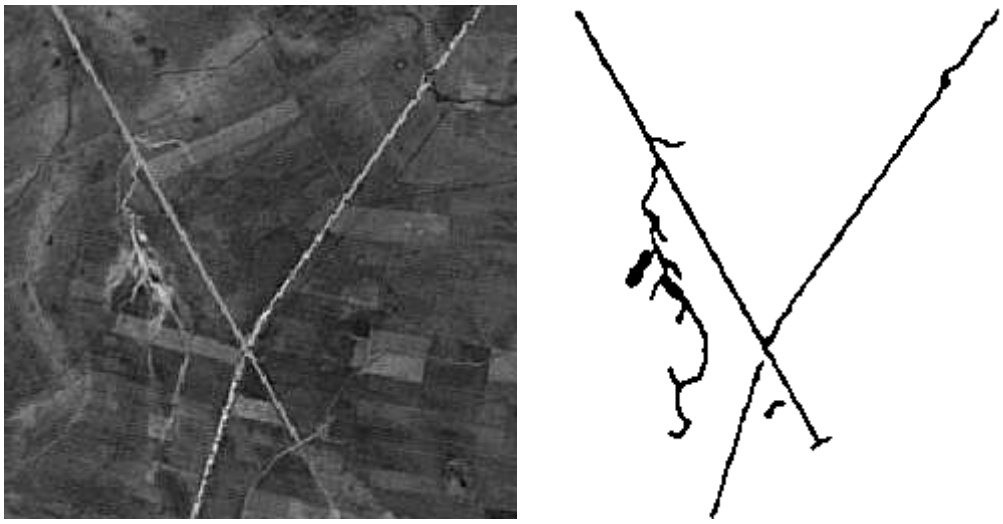

Figure 2.12: Result on a larger piece of the SPOT image.

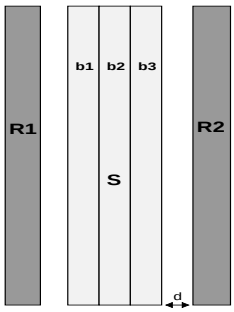

Figure 2.13: Mask for Student tests.

where $*$ indicates convolution. The rotations are chosen from the set $\Theta = \{0, \frac{\pi}{8}, \ldots, \frac{7\pi}{8}\}$. The function Q maps the values into the interval $[-1,1]$:

$$Q(x) = \begin{cases} 1 & \text{if } x < s_1 \ , \\ 1 - 2\frac{x-s_1}{s_2-s_1} & \text{if } s_1 \leq x \leq s_2 \ , \\ -1 & \text{if } x > s_2 \ , \end{cases} \tag{2.36}$$

where s_1 and s_2 are two thresholds, chosen empirically.

Hypothesis tests [52] used Student t-tests for line network detection. Here we adapt their approach to our context. We suppose that roads are homogeneous and contrasted with respect to their environment. A t-test on sets of pixels from inside a potential road will test the homogeneity criterion, while a t-test on sets of pixels from inside and outside a potential road will test the contrast criterion. In order to compute the test, we use the mask shown in figure 2.13.

The Student t-test computes

$$\text{t-test}(x,y) = \frac{|\overline{x} - \overline{y}|}{\sqrt{\frac{\sigma_x}{n_x} + \frac{\sigma_y}{n_y}}} \ ,$$

Figure 2.14: Aerial image. (Image ©IGN.)

where $\bar{\cdot}$, σ and n represent respectively the mean, the standard deviation, and the number of observations. When the result of the test is above a certain threshold, we can consider that the two sets of pixels belong to different populations (implicitly, Gaussian with different means and variances). Given a mask location and orientation, (x,θ), we test the homogeneity and contrast criteria by computing the quantity

$$T_\theta(x) \quad = \quad Q\left(\frac{H_2(S)}{\min\{1,H_1(S)\}}\right) \quad .$$

where

$$\begin{aligned} H_1(S) &= \max_{j,k\in\{1,\ldots,n_b\},j\neq k} [\text{t-test}(b_j,b_k)] \\ H_2(S) &= \min_{l\in\{1,2\}} [\text{t-test}(R_l,S)] \quad . \end{aligned}$$

The function G is then defined by

$$\begin{aligned} \theta_{\max}(x) &= \arg\max_{\theta\in\Theta} |T_\theta(x)| \\ G(x) &= T_{\theta_{\max}(x)}(x) \quad . \end{aligned}$$

2.3.4.4 Experimental results

We add this new energy (2.34) to the model (2.31), and test the model on the high-resolution aerial image shown in figure 2.14. The image presents several difficulties because of high gradients that do not correspond to sides of roads and because of occlusions due to the presence of trees next to the road network. We obtain two extraction results corresponding to the two functions G above. The results are similar, and are shown in figure 2.15.

The main part of the network is extracted, and field borders and other geometric noise are eliminated. In the top-right in one result, a road encircling a house is extracted as a solid area. This happens because 'holes' cannot form in the centre of a region with the current formulation. The main problem, however, is that occlusions due to trees disrupt the network. We are currently addressing this issue using a quadratic energy that causes two road 'tips' to attract one another, and thus close such gaps.

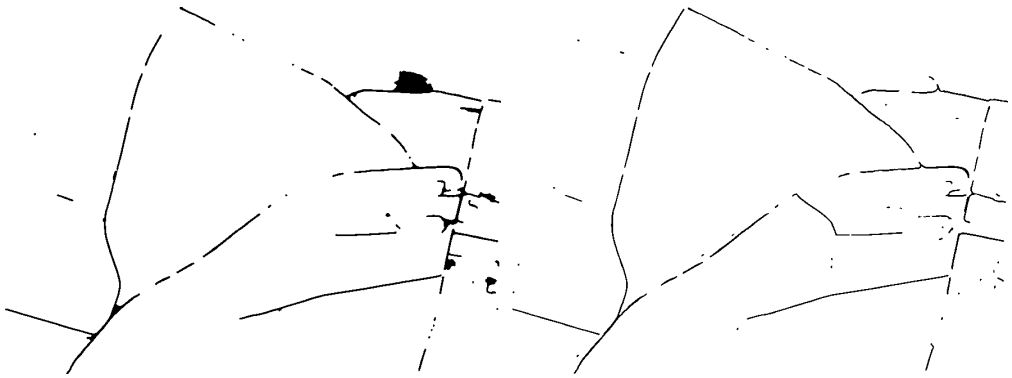

Figure 2.15: Results of extraction with the two functions G.

2.3.4.5 Initialization

Initialization is an issue for many algorithms, and in particular for gradient descent methods. The results may depend heavily on the initialization chosen, and indeed a number of the methods used for the detection of roads rely on an initialization very close to the network. In all the results shown, however, the region used to initialize the gradient descent was chosen to be a rounded rectangle lying just inside Ω. This is possible because greater specificity in the model eliminates many candidate contours from consideration, removing local minima and allowing the boundary to avoid being trapped as easily. In all experiments, the gradient descent was run until it converged.

2.3.5 Conclusions

We have introduced a new class of active contour energy functionals. These energies are polynomial on the space of 1-chains, in contrast to classical energies, which are linear. The new energies enable the introduction of arbitrarily long-range interactions between sets of contour points, and thus the incorporation of sophisticated geometric information, in the sense that the energy minima are not circles but families of complex shapes. We studied a particular form of quadratic energy whose minima consist of fingered structures with parallel sides. Using this energy as a base, we designed an energy functional for the detection of roads in satellite and aerial imagery and tested it on real data. Simulations prove the efficiency of the model and illustrate the effect of the incorporation of non-trivial geometrical interactions between points of the contour. Algorithmically, these models presented new challenges also, in particular the need for a maximum of precision in the calculation of the speed and the evolution of the contour.

Our immediate future work is focused on the solution of the problems mentioned in connection with figure 2.14, where occlusions disrupted the network. We have designed a quadratic ‘gap closure’ force that overcomes the repulsion introduced by the existing quadratic term in certain circumstances, leading road ‘tips’ to attract one another and fill in gaps in the network, something that is impossible using classical techniques. Incorporating such a force into an energy framework is challenging, as it involves higher-order derivatives that create numerical difficulties. We are currently working on resolving these.

Many open questions and research directions remain to be explored: higher-than-quadratic functionals; the extension to surfaces; a probabilistic formulation; parameter and model estimation; new level set techniques; improving computational efficiency; and applications to other domains, in particular to medical imagery.

2.4 Index structures for image search by content

Author: Valérie Gouet, INRIA-Imedia / CEDRIC-Vertigo

In applications involving multimedia databases, similarity queries are very important. This type of query consists in searching for all objects in the database which are similar to a given object. The approaches which are currently used to solve similarity search problems are mostly feature-based solutions. The basic idea is to extract characteristic features from the multimedia objects, map them into a high-dimensional feature space and search, in this space, objects with feature vectors similar to a query one. For example, when dealing with image retrieval, the content of an image can be described by a feature vector which can be a color histogram [14] or a shape descriptor [58], or by several feature vectors like approaches based on points of interest [38] or on regions of interest [29].

For an efficient similarity search, it is necessary to store the feature vectors in a high-dimensional index structure. Such structures must efficiently support the different kinds of queries that may be encountered in multimedia databases: (a)*Point queries*, which consist in finding in the database the points that are identical to the query one; (b)*Range queries*, which involve similarity measures often called $\varepsilon - similarity$. Here the searched objects have a similarity to a given object below a given threshold ε; (c)*Nearest neighbor queries*, which involve similarity measures often called $NN - similarity$. Here the interesting objects are the ones which are the most similar with respect to the searched object. *k-NN queries* refer to the first *k* objects the most similar.

Effects in high-dimensional spaces: the "curse of dimensionality"

Some non-intuitive mathematical effects can be observed when the dimensionality of the data space increases. Generally speaking, the problem is that important parameters such as volumes and area depend exponentially on the dimensions of the space. For example, the volume of a hypercube grows exponentially with increasing dimension and constant length edge. It produces effects concerning the shape and location of the index partitions. For example, a typical index partition in high-dimensional space will span the majority of the data space in most dimensions and only be split in a few dimensions. Another point is that assuming uniformity, a reasonable range query corresponds to a hypercube having a huge extension in each dimension. For more precisions and demonstrations, the reader can consult the survey [12].

2.4.1 Tree-based approaches

Such approaches are based on the principle of hierarchical clustering of the data space. They structurally come from the B^+-tree structure [7, 23]. The feature vectors are stored in data nodes such that spatially adjacent vectors are likely to reside in the same node. The high-dimensional access methods are mainly designed for secondary storage. Consequently, the encountered approaches consist in determining the data nodes such that they fit exactly into the pages of secondary storage. Moreover, for efficient query processing, it is important that the data are well clustered into the pages. A page region is assigned to each page, it is a subset of the feature space. such regions depend on the index structures. We briefly revisit here the most important approaches based on trees for multi-dimensional indexing:

The R-tree family. [2] R-trees [39] have been originally been designed for spatial databases, i.e. for the management of 2-dimensional objects. This class of approach uses bounding rectangles as page

[2] R-tree for Rectangle tree.

regions. They represent multidimensional intervals of feature space and are minimal approximations of the enclosed clusters of points. Such a space partitioning is neither complete nor disjoint. The R^+-tree [89, 79] is an overlap free variant of the R-tree. The split algorithm guarantees no overlap by using a forced-split strategy. The R^*-tree [8] is an extension of the R-tree where several optimizations have been proposed, the most known one consisting in minimizing overlap between page regions. The R-tree and R^*-tree have been also used for high-dimensional indexing. However, studies [11, 101] have shown a deterioration of the performances of the R^*-tree for high-dimensional spaces, due to overlap which increases with such dimensions.

The X-tree. [3] This approach [11] is an extension of the R^*-tree, which is designed for the management of high-dimensional objects. It extends the R^*-tree by two concepts: overlap-free split according to a split history and supernodes with an enlarged page capacity.

The SS-tree. [4] This approach [101] uses spheres as page regions instead of rectangles. Here the spheres do not correspond to minimum bounding spheres: the centroid of the data is used as center and the minimum radius is chosen such that all clustered objects are included in the sphere. Spheres are theoretically superior to volume-equivalent bounding rectangles because the corresponding Minkowski sum (which expresses the selectivity of a bounding volume [12]) is smaller. But the problems of spheres is that it is difficult to design an overlap-free split and that bounding rectangles have a smaller volume in high-dimensional spaces. In conclusion, the SS-tree outperforms the R^*-tree but does not reach the performance of the X-tree.

The SR-tree. [5] This structure [49] can be viewed as a combination of the R^*-tree and the SS-tree. Page regions are built by considering intersections of rectangles and spheres. The motivation for using such a combination is that spheres are better suited for processing nearest neighbor queries and range queries using a L_2 metric. On the other hand, spheres are difficult to produce much overlap in splitting. A combination of these two primitives may overcome both disadvantages. The reported performance results show that the SR-tree outperforms the SS-tree and R^*-tree structures, but no evaluation has been done with the X-tree.

The M-tree. [6] This approach [20] is one of the first that also try to reduce the CPU cost of distance computations. It can be applied on any metric space. The basic structure of the M-tree is similar to the R-tree one. It uses the triangular inequality principle and some pre-computed distances to reduce the number of distances to compute during the search. Its relevance depends on the cost ratio between a distance computation and distances comparison [15].

2.4.2 Other approaches

We revisit here new approaches for efficient database access. On the contrary of the ones described below, most of them have been developed with the aim of taking the problems associated to high-dimensional spaces into account.

Space Filling Curves. Many variants exist for space filling curves, like Z-ordering [60, 63], Hilbert Curve [43, 47] or Gray Codes [28]. For an overview, see [73]. These approaches are mappings from a d-dimensional data space into a one-dimensional data space. Distances are not exactly

[3] X-tree for eXtended Node tree.
[4] SS-tree for Similarity Search tree.
[5] SR-tree for Sphere Rectangle tree.
[6] M-tree for Metric tree.

preserved but points that are close to each other in the original space, are likely to be close to each other in the embedded space. Therefore, these mappings are called distance-preserving mappings.

The Pyramid-tree. This technique [10] can be considered as an index structure that maps d-dimensional point into a one-dimensional space and uses a B^+-tree to index the embedded space. The partitioning strategy is optimized for range queries on high-dimensional spaces. Its main advantage is to generate a number of cells that grows linearly with the dimensions, instead of exponentially as with the traditional approaches.

The VA-File. [7] The VA-File technique [100] is not an index structure, but a compression technique. The authors suggest to accelerate a sequential scan by the use of data compression. The basic idea consists in filtering the data by considering a quantized version of the feature vectors during the search. The quantized points are loaded in memory and are sequentially scanned during the search. Candidates which cannot be pruned are refined, i.e. their exact coordinates are called from the disk.

2.5 CBIR with SVM using kernels with compact support

Author: Hichem Sahbi, INRIA-IMEDIA

2.5.1 Coarse-to-fine image classification

Many large scale classification problems suffer from the computational overhead due to the huge amount of data to be processed and the need to achieve high precision at the detriment of using complex classifiers. Object detection is one of these applications, which has been widely investigated during the last three decades (see for instance [72, 76, 98, 31]) but at this time, there is as yet no solutions with performances comparable to humans' both in precision and speed. High precision is now technically achieved by building systems which learn from lots of data in order to reduce test errors. In most cases, the increase in precision is achieved at the expense of a degradation in run-time performance and in major applications high precision is demanded, so dealing with computation to reduce processing time is now a problem with hard constraints.

Recent studies started to address this issue and introduced alternatives for specific applications, for instance *coarse-to-fine* processing. Indeed, many problems in computer vision have been solved efficiently using coarse-to-fine processing such as object detection, filtering, edge detection, motion estimation, image registration, matching, compression, noise reduction, binocular disparity estimation [36, 6, 2, 71, 83] and in other close areas like speech processing [27]. In object detection, this approach proceeds by rapidly focusing on a particular targeted object in a scene using some statistically common characteristics (global appearance, generic pose constraints, etc.) and by considering that the major part of the scene contains background dominant information which may be rejected quickly.

In the context of face detection, Fleuret and Geman [31] developed a fast and coarse-to-fine face detector based on simple edge configurations and a hierarchical training platform. Their approach considers a nested family of classifiers each one trained on a population of faces with particular pose constraints in order to achieve a negligible miss-detection rate. For a given scene, simple and uniform structures are

[7] VA-File for Vector Approximation File.

rejected using few tests in the hierarchy while more complex and rare structures, for instance textured areas, require more processing. Consequently, the overall average cost to process a scene was dramatically reduced. Using the same idea, Viola and Jones [98] proposed a real-time and accurate face detection algorithm. The main strengths of this algorithm were the use of a new image representation referred to as the integral image, in order to reduce the overhead due to computing the response of wavelet filters, combined with a cascade of classifiers in order to speedup the global face classifier.

Sahbi et al [75, 74] introduced a new computational model for object detection and scene interpretation based a tree-structured network of support vector machines (SVM). This model makes it possible to design a hierarchy of classifiers in order to minimize the overall computational cost of classification under the fact that the a priori distribution of a particular object versus background is unbalanced. For large size classification problems such as face detection, the design of the hierarchical detector, ensures fast processing of most of the dominant patterns (such as background) using cheap SVMs, and fine processing only in the areas containing rare patterns (faces) and similar structures using more expensive and accurate SVMs, thereby resulting in an efficient classifier.

In [41], the authors introduced a method to speed-up object classification that uses support vector machines. At the top of their hierarchy, the authors use a simple and fast linear classifier which analyzes the whole image and rejects quickly large parts of the background. At other levels, slow but more accurate linear classifiers are used in order to perform the final detection. The authors apply feature reduction to the top level classifiers by choosing relevant image features according to a measure derived from statistical learning theory. This makes it possible to speedup these classifiers. In [70], the authors used a cascade of SVMs in order to speedup face detection. The cost of the SVMs (defined as the number of support vectors) is constrained to increase throughout different steps of the cascade. First, a complex SVM classifier is trained by solving a classical quadratic programming problem, then each classifier in the cascade is built using the reduced set technique [77]. Again, the major part of a scene is rejected using cheap classifiers at the early stages of the cascade while face and face-like structures are classified using more steps. In their work, [71] make their face detector more flexible and rapid using coarse-to-fine and this by training a coarse neural network classifier which detects faces whose locations are given into a 10×10 pixel block. Thus, this detector can be moved in steps of 10 pixels across the image, and still detects all faces present within these tolerances. [71] use also a finer neural network classifier in the regions of 10×10 pixels for which the coarse detector responds positively. To reduce the effort of scanning these 10×10 blocks, the author trains a neural network which returns the coordinates of the candidate face inside a given 10×10 block. These coordinates are used to extract the face window and to validate the face hypothesis using a fine classifier.

2.6 CBIR using decision-theoretic approaches

Authors: Simon Wilson and Georgios Stefanou, Trinity College

2.6.1 Introduction

In the previous deliverables we have described our extensions to the Bayesian content based image retrieval (CBIR) system *PicHunter* of [24]. In this short report we discuss how the Bayesian paradigm may be extended to another aspect of the relevance feedback process, namely the decision as to which images to display at each iteration. This is a decision problem, and as such it is to be solved within the

Bayesian paradigm by the methods of decision theory.

This report is organised as follows. In Section 2.6.2 we describe the Bayesian learning algorithm for CBIR. In Section 2.6.3 we describe the decision theory solution to the display strategy problem. Section 2.6.4 concludes with some examples.

2.6.2 A Brief Description of a Bayesian CBIR system

Our approach is an extension of [24]. We consider a database of images $I = \{T_1, \ldots, T_N\}$. The objective is to determine the "target" image $T \in I$ that the user requires. T could be a specific known image in the database or more generally that image in I which best satisfies the user's subjective search criteria. The determination of T is accomplished by displaying a set of N_D images from I, from which the user picks one that best satisfies what is being looked for. The system uses this information to select another image set, from which the user picks one, and so on. We define $D_i \subseteq I$ to be the set of displayed images at the ith iteration of this process, and $A_i \in D_i$ to be the image picked, also known as the user action. We define $H_t = \{D_1, A_1, D_2, A_2, \ldots, D_t, A_t\}$ to be the history of displayed images and user actions up to the tth iteration.

The learning algorithm is based around the user model for the probability of which image a user picks from D_k:

$$P(A_k \,|\, D_k, T = T_i, \sigma, F) = \frac{\exp\left(-d_F(A_k, T_i)/\sigma\right)}{\sum_{T_j \in D_k} \exp\left(-d_F(T_j, T_i)/\sigma\right)}, \tag{2.37}$$

where σ is a precision parameter and d_F is a normalised distance measure in the set of image features F. In this case we have 3 sets of features: global colour, texture and segmentation features, so $F \in \{\text{GC, TX, SG}\}$.

The unknowns are T, the precision parameter σ and the feature set F. Given H_t, our knowledge about these unknowns is given by the posterior distribution:

$$P(T, \sigma, F \,|\, H_t) \propto \left(\prod_{k=1}^{t} P(A_k \,|\, D_k, T, \sigma, F) \right) P(T) P(\sigma) P(F), \tag{2.38}$$

where $P(T)$, $P(\sigma)$ and $P(F)$ are prior distributions that we assume are uniform: $P(T = T_i) = N^{-1}, i = 1, \ldots, N$, $P(\sigma) = 1, 0 \leq \sigma \leq 1$ and $P(F) = 1/3, F \in \{\text{GC,TX,SG}\}$.

Of interest in this report is the marginal posterior distribution of T:

$$P(T = T_i \,|\, H_t) = \int_0^1 \sum_{F \in \{\text{GC,TX,SG}\}} P(T_i, \sigma, F \,|\, H_t),\ i = 1, \ldots, N. \tag{2.39}$$

2.6.3 Deciding the Next Display Set $\mathbf{D_{t+1}}$

The question that this report addresses is the following. Based on $P(T = T_i \,|\, H_t)$, which set of images D_{t+1} should be displayed next? This is a decision problem — we must decide which subset of I of size N_D to display — and within the Bayesian paradigm, such problems are solved by decision theory.

We define a utility $U(D, T)$ that is the "worth" of picking the set D to display when the target image is T. Since T is unknown, we compute for each possible D the expected utility with respect to $P(T = T_i \,|\, H_t)$:

$$\mathcal{U}(D) = \sum_{i=1}^{N} U(D, T_i) P(T = T_i \,|\, H_t). \tag{2.40}$$

The optimal set to display is that D which maximises expected utility:

$$D_{t+1} = \arg \max_{\substack{D \subseteq I \\ |D|=N_D}} \mathcal{U}(D). \tag{2.41}$$

2.6.3.1 The Most Probable Display Scheme

The most obvious display scheme is to display those N_D images with the highest posterior probability. We observe that if we define

$$U_I(D,T) = \begin{cases} 1, & \text{if } T \in D, \\ 0, & \text{otherwise,} \end{cases}$$

then

$$\mathcal{U}_I(D) = \sum_{T_i \in D} P(T = T_i \,|\, H_t)$$

which is clearly maximised by those images with highest probability. We call this the indicator utility.

2.6.3.2 Other Display Strategies

A property of the most probable display scheme is that it tends to quickly display images in a small region of the feature space, clustered about the user actions, and ignores all images outside it. While this may be ultimately what is needed during a query, it may be more worthwhile to display images that maximise information to the system, at least in the early stages of the query process. We propose 2 utilities to model this idea.

Variance Utility We display a set of images that are widely dispersed in feature space. We can use the variance of the distances between images in D and T to define a measure of dispersion, thus

$$U_V(D,T) = \frac{1}{N_D - 1} \sum_{T_i \in D} (d(T_i,T) - \overline{d})^2,$$

where $d(T_i,T)$ is a normalised distance measure in feature space and

$$\overline{d} = \sum_{T_i \in D} d(T_i,T)/N_D$$

is the mean distance of images in D to T.

Entropy Utility A measure of information is the reduction in entropy in the distribution of T by selecting a particular display set. So we can define a utility based on the negative expected entropy of the posterior of T from picking an image in D, expectation over the images in D:

$$U_E(D,T) = -\sum_{A_j \in D} \mathcal{E}(A_j,D)\, P(A_j \,|\, D,T), \tag{2.42}$$

where

$$\mathcal{E}(A_j,D) = -\sum_{i=1}^{N} P(T = T_i \,|\, A_j,D) \log(P(T = T_i \,|\, A_j,D))$$

is the entropy of the posterior distribution of T given that A_j is picked from D (following Equations 2.38 and 2.39) and

$$P(A_j \,|\, D,T) = \frac{\exp(-d(A_j,T)/\overline{\sigma})}{\sum_{T_j \in D} \exp(-d(T_j,T)/\overline{\sigma})}$$

is the likelihood term as in Equation 2.37 but using the distance measure d over the entire feature space and $\overline{\sigma}$ is the posterior mean of σ.

2.6.3.3 Optimisation Methods

Because $\mathcal{U}(D)$ is separable in each element of D, the optimal D for the indicator utility can be easily computed. This is not the case if one moves to using the variance or entropy utilities. Evaluation of the expected utility for all possible D is not an option as the number is large i.e. for $N = 1000$ and $N_D = 6$ we have about 1.37×10^{15} possible subsets. For these, we have to resort to methods that are not guaranteed to find the optimal. We propose two Monte Carlo optimisation schemes.

Random Generation We randomly generate without replacement K subsets $D^1, \ldots, D^K$. Then we let

$$D_{t+1} = \arg\max_{k=1}^{K} \mathcal{U}(D^k). \tag{2.43}$$

Each element of a set D can be simulated from any distribution on I; obvious choices are the uniform and $P(T \mid H_t)$. In this paper we choose the latter.

Simulated Annealing For simulated annealing, we define a "neighbour" of a subset D to be another subset with one different image. A simulated annealing algorithm then runs as follows:

1. Define an initial temperature T_0, a final temperature $T_{\min}$ and a cooling schedule $T_1, T_2, \ldots$. Randomly generate without replacement a set D^0, using $P(T \mid H_t)$. Let $k = 0$.
2. While $T_k > T_{\min}$
   - $k = k + 1$.
   - Select at random one image in D^{k-1} and replace with another image in $I - D^{k-1}$, randomly generated according to $P(T \mid H_t)$. Call this new set D^{new}.
   - With probability $\min\{1, \exp((\mathcal{U}(D^{\text{new}}) - \mathcal{U}(D^{k-1}))/T_k)\}$, let $D^k = D^{\text{new}}$ else $D^k = D^{k-1}$.
3. $D_{t+1} = D^k$.

Initial and final temperatures were decided on by using the methods of [51] and [56]. We looked at several different cooling schedules, and found that inverse linear ($T_k = a/(1+bk)$) performed best. The choice of $P(T \mid H_t)$ to generate D^0 and D^{new} can be changed, to for example the uniform, but we found that the method was not particularly sensitive to this choice. Finally, our definition of neighbour can be made more or less strict, by for example allowing two changes for D^{new} or, conversely, only favouring replacement of one image in D^{new} that is close in feature space to that image replaced. However we found that our choice was a compromise between a too small and too large change that offered a good accept rate.

Finally we note that computation time is limited in a live implementation of either optimisation scheme, so typically we can compute only a small number of expected utilities.

2.6.4 Examples

As an illustration of the method, we have a simulated database of only $N = 15$ images, each with only 2 features, for which queries are implemented by displaying $N_D = 3$ images. This is clearly an unrealistically small example but it has the advantages of allowing us to display what happens in feature space

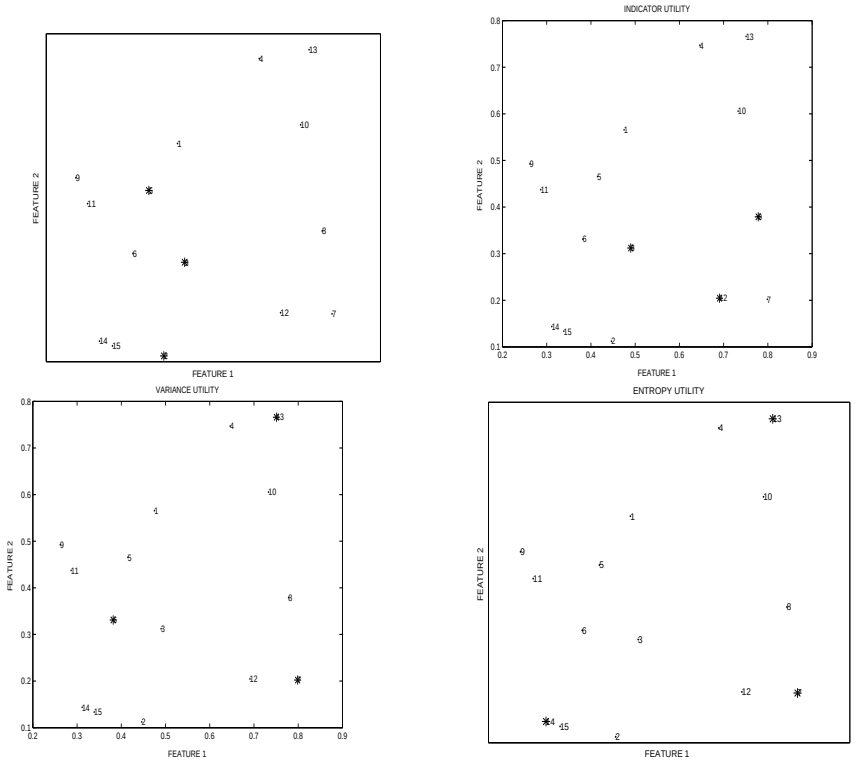

Figure 2.16: Feature space plots for the selection of D_2 for a simple database of 15 images given $D_1 = \{T_2, T_3, T_5\}$ and $A_1 = T_3$. Images in D_2 are highlighted by *.

and, since there are only 455 possible subsets of size 3, to compute the exact optimal subset under all 3 utilities and compare with the results obtained by random sampling and simulation.

Figure 2.16 shows an example of the system where one image A_1 is picked from an initial display set D_1, and the resulting choice of D_2 according to the 3 utilities. Upper left of the figure are the 15 images in feature space with $D_1 = \{T_2, T_3, T_5\}$ highlighted. Image 3 is selected. We then see that, under U_I, we have $D_2 = \{T_3, T_8, T_{12}\}$, that is images close to that selected. For U_V we have $D_2 = \{T_6, T_7, T_{13}\}$ and for U_E we have $D_2 = \{T_7, T_{13}, T_{14}\}$, that is images that are widely separated in feature space are chosen. To explore the effectiveness of the optimisation methods, we repeated this experiment 1000 times, computing the optimal D_2 according to the random generation method and simulated annealing. For the random subset generation, we simulated $K = 100$ subsets. For the simulation annealing we used an inverse linear cooling schedule $T_k = a/(1+bk)$ with a and b chosen so that the final temperature was reached in 100 iterations, thus both methods took the same time to compute. The results are compared with the exact calculation in Table 2.8 and we see that both non-exact methods are sub-optimal, but nevertheless do manage on average to find subsets with expected utility close to the optimal. Random generation appears to do slightly better than simulated annealing.

Finally, we move to the BAL database. In this case we cannot enumerate all possible subsets and so D_2 under the variance and entropy utilities is computed by the two optimisation schemes only. Table 2.9 compares the optimisation methods over 10 runs for an example from this database where D_1 consists of 6 images; note that we only have the exact result for the indicator utility. From the results for the indicator utility it appears that both optimisation methods can be significantly sub-optimal. From all 3 utilities it appears that random generation performs better than simulated annealing.

Utility	Exact Computation over all subsets	Random Generation of 100 subsets	Simulated Annealing 100 iterations
Indicator	0.2643	0.2624	0.2591
Variance	0.2299	0.2264	0.2246
Entropy	-2.6869	-2.6879	-2.6883

Table 2.8: The average of the expected utility for D_2 over 1000 runs for the 3 utility functions and three computation methods. All runs use the example of Figure 2.16.

Utility	Exact Computation over all subsets	Random Generation of 100 subsets	Simulated Annealing 100 iterations
Indicator	0.0157	0.0106	0.0100
Variance	—	0.7767	0.7583
Entropy	—	-6.9642	-6.9647

Table 2.9: The average of the expected utility for D_2 over 10 runs for the 3 utility functions and three computation methods using the BAL database.

Figure 2.17 is an example from a database of paintings from the Bridgeman Art Library, using sets of 6 images. The upper display set is D_1, then the two lower sets are D_2 under the indicator and entropy utilities. They clearly show the large effect that the choice of utility has on the search process.

2.6.5 Conclusion

We have described a decision-theoretic approach to the problem of display set strategy in content-based image retrieval systems. The notion of utility is, we believe, a useful and intuitive way to quantify display strategy objectives. One is free to define any utility function at all, as long as computational issues can be successfully addressed.

It remains to say that the two new utilities that we have proposed — variance and entropy — are primarily of use in the early stages of a query, when the objective is to learn as much as possible about the user's target. Ultimately, one will want to resort to a utility that displays images close to the target, such as the indicator. An obvious way to do this is to consider a utility that is a convex weighted combination of the indicator utility with one of the other two, with the weight on the indicator utility increasing to 1 with the iteration, for example at the tth display set:

$$U(D,T) = \alpha_t U_I(D,T) + (1-\alpha_t)U_E(D,T),$$

with $0 \leq \alpha_t \leq 1$ and $\alpha_t \to 1$, and the entropy utility normalised from that in Equation 2.42 so that it lies in $[0,1]$ like $U_I(D,T)$. This is the subject of current work.

2.6.6 Acknowledgements

The images in the final figure are courtesy of the Bridgeman Art Library, London.

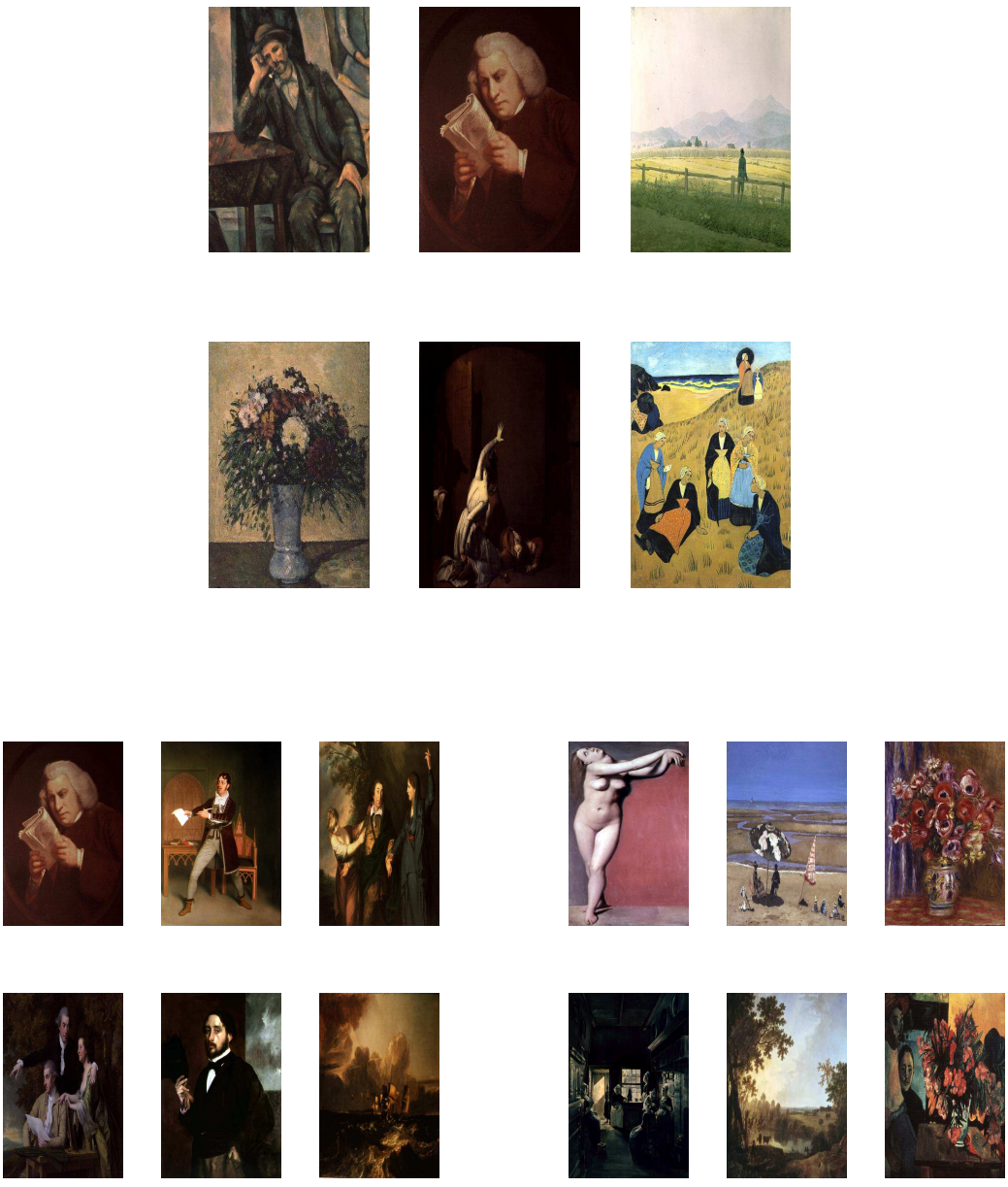

Figure 2.17: An example of display set selection with paintings. At the top are six images for D_1. The male portrait (centre of top row) is selected. Below left is D_2 under the indicator utility. Below right is D_2 under the entropy utility.

2.7 State of the Art on Computation Intensive Methods in Video Outdoor Surveillance Systems

by László Havasi (havasi@digitus.itk.ppke.hu) and Tamás Szirányi (SzTAKI)

In this section we give an up-to-date overview about methods that require high computational cost in video surveillance systems. The most important fact is several of the below methods should run simultaneously in real-time to solve a complex multi-camera surveillance task.

2.7.1 Preprocessing

2.7.1.1 Image Deblurring

[9] Motion blur due to camera motion can significantly degrade the quality of an image. Since the path of the camera motion can be arbitrary, deblurring of motion blurred images is a hard problem. Previous methods to deal with this problem have included blind restoration of motion blurred images, optical correction using stabilized lenses, and special CMOS sensors that limit the exposure time in the presence of motion. Take advantage of the fundamental tradeoff between spatial resolution and temporal resolution to construct a hybrid camera that can measure its own motion during image integration. The acquired motion information is used to compute a point spread function (PSF) that represents the path of the camera during integration. This PSF is then used to deblur the image. The method can be extended beyond the case of global camera motion to the case where individual objects in the scene move with different velocities.

Note:
This deblurring method is not real time, but it is needed in case of moving cameras.

2.7.1.2 Background modeling

[87] A common method for real-time segmentation of moving regions in image sequences involves background subtraction, or thresholding the error between an estimate of the image without moving objects and the current image. The numerous approaches to this problem differ in the type of background model used and the procedure used to update the model. This method implements modeling each pixel as a mixture of Gaussians and using an on-line approximation to update the model. The Gaussian distributions of the adaptive mixture model are then evaluated to determine which are most likely to result from a background process. Each pixel is classi?ed based on whether the Gaussian distribution which represents it most effectively is considered part of the background model. This results in a stable, real-time outdoor tracker which reliably deals with lighting changes, repetitive motions from clutter, and long-term scene changes. This system has been run almost continuously for 16 months, 24 hours a day, through rain and snow.

Note:
In our expreiments this method can run with 10-15 FPS on medium sized images, but it causes heavy CPU usage.

2.7.1.3 Shadow detection

[69] Moving shadows need careful consideration in the development of robust dynamic scene analysis

systems. Moving shadow detection is critical for accurate object detection in video streams since shadow points are often misclassified as object points, causing errors in segmentation and tracking. Many algorithms have been proposed in the literature that deal with shadows. The paper presents a comprehensive survey of moving shadow detection approaches, two of them are statistical and two are deterministic.

Note:
The shadow detection is an important task in surveillance systems, for instance traffic counters and indoor action recognition. The cleared input images are goint to increase the accuracy of the following methods.

2.7.2 Feature extraction

2.7.2.1 Edge detection

[26] This statistical model was designed for the gradient vector field of the gray level in images. Moreover, the model contains a global constrained Markov model for contours in images that uses this statistical model for the likelihood. The model is amenable to an Iterative Conditional Estimation (ICE) procedure for the estimation of the parameters; the model also allows segmentation by means of the Simulated Annealing (SA) algorithm, the Iterated Conditional Modes (ICM) algorithm, or the Modes of Posterior Marginals (MPM) Monte Carlo (MC) algorithm. This yields an original unsupervised statistical method for edge-detection, with three variants. The tests indicate that the model and its estimation are valid for applications that require an energy term based on the log-likelihood ratio. Besides edge-detection, this model can be used for semiautomatic extraction of contours, localization of shapes, non-photo-realistic rendering; more generally, it might be useful in various problems that require a statistical likelihood for contours.

Note:
The edge map is the input of several detection and recognition methods. This fact means that the quality of edge map should be as good as possible. The statistical methods generate pretty clear edge maps, but the computation time is exremly high.

2.7.2.2 Symmetry extraction

[35] The availability of large 3D datasets has made volume thinning essential for compact representation of shapes. The density of the skeletal structure resulting from the thinning process depends on the application. Current thinning techniques do not allow control over the density and can therefore address only specific applications. The paper describes an algorithm which uses a thinness parameter to control the thinning process and thus the density of the skeletal structure.

Note:
The symmetry map is very characteristic to the objects and shapes. It leads several applications on recognition and classification. Nevertheless, the calculation of the ideal symmetry map is far away from real-time use.

2.7.3 Feature understanding/ Symmetry based shape recognition

[78] This paper presents a novel framework for the recognition of objects based on their silhouettes. The main idea is to measure the distance between two shapes as the minimum extent of deformation necessary for one shape to match the other. Since the space of deformations is very high-dimensional, three steps

are taken to make the search practical: 1) define an equivalence class for shapes based on shock-graph topology, 2) define an equivalence class for deformation paths based on shock-graph transitions, and 3) avoid complexity-increasing deformation paths by moving toward shock-graph degeneracy. Despite these steps, which tremendously reduce the search requirement, there still remain numerous deformation paths to consider. The proposed approach gives intuitive correspondences for a variety of shapes and is robust in the presence of a wide range of visual transformations. The recognition rates on two distinct databases of 99 and 216 shapes each indicate highly successful within category matches (100 percent in top three matches), which render the framework potentially usable in a range of shape-based recognition applications.

2.7.4 Motion classification

2.7.4.1 Activity detection by co-occurrence statistics

[86] The paper presents a visual monitoring system that passively observes moving objects in a site and learns patterns of activity from those observations. For extended sites, the system will require multiple cameras. Thus, key elements of the system are motion tracking, camera coordination, activity classification, and event detection. This paper focuses on motion tracking and show how one can use observed motion to learn patterns of activity in a site. Motion segmentation is based on an adaptive background subtraction method that models each pixel as a mixture of Gaussians and uses an on-line approximation to update the model. The Gaussian distributions are then evaluated to determine which are most likely to result from a background process. This yields a stable, real-time outdoor tracker that reliably deals with lighting changes, repetitive motions from clutter, and long-term scene changes. While a tracking system is unaware of the identity of any object it tracks, the identity remains the same for the entire tracking sequence. The system leverages this information by accumulating joint co-occurrences of the representations within a sequence. These joint co-occurrence statistics are then used to create a hierarchical binary-tree classification of the representations. This method is useful for classifying sequences, as well as individual instances of activities in a site.

Note:
The statistical based events detection methods are very promising, but in our experiments with similar motion statistics the acquisition is memory and CPU demanding in high resolution images.

2.7.4.2 Pedestrian detection

[84] This paper presents an unsupervised learning algorithm that can derive the probabilistic dependence structure of parts of an object (a moving human body in our examples) automatically from unlabeled data. The distinguished part of this work is that it is based on unlabeled data, i.e., the training features include both useful foreground parts and background clutter and the correspondence between the parts and detected features are unknown. We use decomposable triangulated graphs to depict the probabilistic independence of parts, but the unsupervised technique is not limited to this type of graph. In the new approach, labeling of the data (part assignments) is taken as hidden variables and the EM algorithm is applied. A greedy algorithm is developed to select parts and to search for the optimal structure based on the differential entropy of these variables.

Note:
There isn't polynomial method for graph decomposition, so this effective method cannot run in real-time in case of complex scene.

2.7.5 Camera calibration/A statistical approach

[55] Monitoring of large sites requires coordination between multiple cameras, which in turn requires methods for relating events between distributed cameras. This paper tackles the problem of automatic external calibration of multiple cameras in an extended scene, that is, full recovery of their 3D relative positions and orientations. Because the cameras are placed far apart, brightness or proximity constraints cannot be used to match static features, so the method instead applies planar geometric constraints to moving objects tracked throughout the scene. By robustly matching and ?tting tracked objects to a planar model, the scene's ground plane is aligned across multiple views and decomposed the planar alignment matrix to recover the 3D relative camera and ground plane positions. The method does not require synchronized cameras, and the paper shows that enforcing geometric constraints enables us to align the tracking data in time. In spite of noise in the intrinsic camera parameters and in the image data, the system successfully transforms multiple views of the scene 's ground plane to an overhead view and recovers the relative 3D camera and ground plane positions.

[92][93] A method presented for groundplane estimation from image-pairs even if unstructured environment and motion. In a typical outdoor multi-camera system the observed objects might be very different due to the noise coming from lighting conditions, camera positions. Static features such as color, shape, and contours cannot be used for image matching in these cases. In the paper a method is proposed for matching partially overlapping images captured by video cameras. Using co-motion statistics, which is followed by outlier detection and a nonlinear optimization, does the matching. The described robust algorithm finds point correspondences in two images without searching for any structures and without tracking any continuous motion. Real-life outdoor experiments demonstrate the feasibility of this approach.

Note:
Our statistical calibration method is very similar to the above method, but our method is more general. The disadvantage of these methods is the extremely high memory usage and CPU power to update the memory.

Bibliography

[1] D. Adalsteinsson and J. A. Sethian. The fast construction of extension velocities in level set methods. *J. Comp. Phys.*, 148:2–22, 1999.

[2] Yali Amit and D. Geman. A computational model for visual selection. *Neural Computation.*, 11(7):1691–1715, 1999.

[3] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies. Image coding using wavelet transform. *IEEE Transactions on Image Processing*, 1:205–220, 1992.

[4] M. Bagci, Y. Yardimci, and A. E. Cetin. Moving object detection using adaptive subband decomposition and fractional lower order statistics in video sequences. *Elsevier, Signal Processing*, pages 1941–1947, 2002.

[5] M. Barzohar and D. Cooper. Automatic finding of main roads in aerial images by using geometric-stochastic models and estimation. *IEEE Trans. Patt. Anal. Mach. Intell.*, 18(7):707–721, 1996.

[6] Roberto Battiti and Christof Koch. Computing optical flow across multiple scales: a coarse-to-fine approach. *International Journal of Computer Vision*, 6(2):133–145, 1991.

[7] R. Bayer and E. McCreight. Organization and maintenance of large ordered indices. *Acta Informatica*, 1(3):173–189, 1977.

[8] N. Beckmann, H.P. Kriegel, R. Schneider, and B. Seeger. The r*-tree: an efficient and robust access method for points and rectangles. In *ACM SIGMOD Int. Conf. on Management of Data*, pages 322–331, Atlantic City, NJ, 1990.

[9] Moshe Ben-Ezra and Shree K. Nayar. Motion based motion deblurring. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):689–699, June 2004.

[10] S. Berchtold, C. Bohm, and H.P. Kriegel. The pyramid-technique: towards indexing beyond the curse of dimensionality. In *ACM SIGMOD Int. Conf. on Management of Data*, pages 142–153, Seattle, WA, 1998.

[11] S. Berchtold, D. Keim, and H.P. Kriegel. The x-tree: An index structure for high-dimensional data. In *22th Int. Conf. on Very Large Databases*, pages 28–39, Bombay, India, 1996.

[12] C. Bohm, S. Berchtold, and D.A. Keim. Searching in high-dimensional spaces - index structures for improving the performance of multimedia databases. *ACM Computing Survey*, 33(3):322–373, 2001.

[13] A. Bossavit. Applied differential geometry: A compendium, 2002. http://www.icm.edu.pl/edukacja/mat/Compendium.php.

[14] S. Boughorbel, N. Boujemaa, and C. Vertan. Histogram-based color signatures for image indexing. In *Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU'2002*, 2002.

[15] B. Braunmuller, M. Ester, H.P. Kriegel, and J. Sander. Efficiently supporting multiple similarity queries for mining in metric databases. In *16th Int. Conf. on Data Engineering*, pages 256–270, San Diego, CA, 2000.

[16] V. Caselles, F. Catte, T. Coll, and F. Dibos. A geometric model for active contours. *Numerische Mathematik*, 66:1–31, 1993.

[17] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *Int'l J. Comp. Vis.*, 22(1):61–79, 1997.

[18] T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Trans. Im. Proc.*, 10-2:266–277, 2001.

[19] Y. Chen, S. Thiruvenkadam, H. D. Tagare, F. Huang, D. Wilson, and E.A. Geiser. On the incorporation of shape priors into geometric active contours. *Proc. IEEE Workshop VLSM*, pages 145–152, 2001.

[20] P. Ciaccia, M. Patella, and P. Zezula. M-tree: An efficient access method for similarity search in metric spaces. In *23th Int. Conf. on Very Large Databases*, pages 426–435, Greece, 1997.

[21] L. D. Cohen. On active contours and balloons. *CVGIP: Image Understanding*, 53:211–218, 1991.

[22] R. T. Collins, A. J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt, and L. Wixson. A system for video surveillance and monitoring: Vsam final report. *Technical Report CMU-RI-TR-00-12*, 1998.

[23] D. Comer. The ubiquitous b-tree. *ACM Computing Survey*, 11(2):121–138, 1979.

[24] I. J. Cox, M. L. Miller, T. P. Minka, T. V. Papathomas, and P. N. Yianilos. The Bayesian image retrieval system, PicHunter: theory, implementation and psychophysical experiments. *IEEE Transactions on Image Processing*, 9:20–37, 2000.

[25] D. Cremers, C. Schnorr, and J. Weickert. Diffusion-snakes: combining statistical shape knowledge and image information in a variational framework. *Proc. IEEE Workshop VLSM*, pages 137–144, 2001.

[26] Francois Destrempes and Max Mignotte. A statistical model for contours in images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):626–639, May 2004.

[27] R. Duraiswami, D. Zotkin, and L. Davis. Active speech source localization by a dual coarse-to-fine search. *in icassp*, 2001.

[28] C. Faloutsos. Gray codes for partial match and range queries. *IEEE Trans. on software Engineering*, 14:1381–1393, 1988.

[29] J. Fauqueur and N. Boujemaa. New image retrieval paradigm : Logical composition of region categories. In *IEEE International Conference on Image Processing (ICIP'2003)*, Barcelona, Spain, 2003.

[30] M. A. Fischler, J. M. Tenenbaum, and H. C. Wolf. Detection of roads and linear structures in low-resolution aerial imagery using a multisource knowledge integration technique. *Comp. Graph. and Im. Proc.*, 15:201–223, 1981.

[31] F. Fleuret and D. Geman. Coarse-to-fine visual selection. *In International Journal of Computer Vision*, 41(2):85–107, 2001.

[32] G. L. Foresti, P. Mahonen, and C. S. Regazzoni. *Multimedia Video-Based Surveillance Systems: Requirements, Issues and Solutions*. Kluwer, 2000.

[33] A. Foulonneau, P. Charbonnier, and F. Heitz. Geometric shape priors for region-based active contours. *Proc. IEEE ICIP.*, 3:413– 416, 2003.

[34] P. Fua and Y. G. Leclerc. Model driven edge detection. *Mach. Vis. and Appl.*, 3:45–56, 1990.

[35] N. Gagvani and D. Silver. Parameter controlled volume thinning. *Graphical Models and Image Processing*, 61(3):149–164, 1999.

[36] J. C. Gee and D. R. Haynor. Rapid coarse-to-fine matching using scale-specific priors. *in Proc. SPIE Medical Imaging: Image Processing, M. H. Loew and K. M. Hanson, eds., Bellingham, WA:SPIE*, 2710, 1996.

[37] D. Geman and B. Jedynak. An active testing model for tracking roads in satellite images. *IEEE Trans. Patt. Anal. Mach. Intell.*, 18:1–14, 1996.

[38] V. Gouet and N. Boujemaa. Object-based queries using color points of interest. In *IEEE Workshop on Content-Based Access of Image and Video Libraries (CBAIVL 2001)*, pages 30–36, Kauai, Hawaii, USA, 2001.

[39] A. Guttman. R-trees: a dynamic index structure for spatial searching. In *ACM SIGMOD Int. Conf. on Management of Data*, pages 47–57, Boston, MA, 1984.

[40] I. Haritaoglu, D. Harwood, and L. Davis. W4: real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:809–830, 2000.

[41] B. Heisele, T. Serre, S. Mukherjee, and T. Poggio. Feature reduction and hierarchy of classifiers for fast object detection in video images. *In: Proceedings of 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2:18–24, 2001.

[42] Aware Inc., 40 Middlesex Turnpike, Bedford, Massachusetts, and 01730 URL:www.aware.com. *MotionWavelets^{TM} real-time software video codec*, 1999.

[43] H. Jagadish. Linear clustering of objects with multiple attributes. In *ACM SIGMOD Int. Conf. on Management of Data*, pages 332–342, Atlantic City, NJ, 1990.

[44] S. Jehan-Besson, M. Barlaud, and G. Aubert. DREAM2S: Deformable regions driven by an Eulerian accurate minimization method for image and video segmentation. *Int'l J. Comp. Vis.*, 53:45–70, 2003.

[45] I. H. Jermyn. On Bayesian estimation in manifolds. Research Report 4607, INRIA, France, November 2002.

[46] I. H. Jermyn and H. Ishikawa. Globally optimal regions and boundaries as minimum ratio cycles. *IEEE Trans. Patt. Anal. Mach. Intell. (Special Section on Graph Algorithms and Computer Vision)*, 23(10):1075–1088, October 2001.

[47] I. Kamel and C. Faloutsos. On packing r-trees. In *Second international conference on Information and knowledge management*, pages 490–499, Washington, DC, 1993.

[48] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Int'l J. Comp. Vis.*, pages 321–331, 1988.

[49] N. Katayama and S. Satoh. The sr-tree: An index structure for high-dimensional nearest neighbor queries. In *ACM SIGMOD Int. Conf. on Management of Data*, pages 369–380, 1997.

[50] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi. Gradient flows and geometric active contour models. *Proc. ICCV*, pages 810–815, 1995.

[51] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[52] C. Lacoste, X. Descombes, and J. Zerubia. A comparative study of point processes for line network extraction in remote sensing. Research Report 4516, INRIA, France, August 2002.

[53] I. Laptev, T. Lindeberg, W. Eckstein, C. Steger, and A. Baumgartner. Automatic extraction of roads from aerial images based on scale space and snakes. *Mach. Vis. and Appl.*, 12:23–31, 2000.

[54] M. E. Leventon, W. E. L. Grimson, and O. Faugeras. Statistical shape influence in geodesic active contours. *Proc. IEEE CVPR*, 1:316–322, 2000.

[55] L.Lee, R.Romano, and G.Stein. Monitoring activities from multiple video streams: Establishing a common coordinate frame. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):758–767, July 2000.

[56] M. Lundy and A. Mees. Convergence of an annealing algorithm. *Mathematical Programming*, 34:111–124, 1986.

[57] R. Malladi, J. A. Sethian, and B. C. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Trans. Patt. Anal. Mach. Intell.*, 17:158–175, 1995.

[58] R. Mehrotra and J. Gary. Feature-index-based similar shape retrieval. In *Third Working Conference on Visual Databases Systems*, 1995.

[59] N. Merlet and J. Zerubia. New prospects in line detection by dynamic programming. *IEEE Trans. Patt. Anal. Mach. Intell.*, 18(4):426–431, 1996.

[60] G. Morton. *A computer oriented geodetic database and a new technique in file sequencing*. IBM Ltd., USA, 1966.

[61] S. Naoi, H. Egawa, and M. Shiohara. Image processing apparatus. *U.S. Patent 6,141,435*, 2000.

[62] W. M. Neuenschwander, P. Fua, L. Iverson, G. Székely, and O. Kubler. Ziplock snakes. *Int'l J. Comp. Vis.*, 25(3):191–201, 1997.

[63] J. Orenstein. A comparison of spatial query processing techniques for native and parameter spaces. In *ACM SIGMOD Int. Conf. on Management of Data*, pages 326–336, 1990.

[64] S. Osher and J. A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations. *J. Comp. Phys.*, 79:12–49, 1988.

[65] I. B. Ozer and W. Wolf. A hierarchical human detection system in (un)compressed domains. *IEEE Transactions on Multimedia*, 4:283–300, 2002.

[66] N. Paragios and R. Deriche. Geodesic active regions: A new framework to deal with frame partition problems in computer vision. *Journal of Visual Communication and Image Representation*, 13:249–268, 2002.

[67] N. Paragios and M. Rousson. Shape priors for level set representations. *Proc. ECCV*, pages 78–92, 2002.

[68] T. Pavlidis. *Algorithms for Graphics and Image Processing*, chapter 7. Computer Science Press, Inc., 1982.

[69] Andrea Prati, Ivana Mikic, Mohan M. Trivedi, and Rita Cucchiara. Detecting moving shadows: Algorithms and evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):918–923, July 2003.

[70] Sami Romdhani, Philip Torr, Bernhard Schölkopf, and Andrew Blake. Computationally efficient face detection. *Proc. ICCV*, pages 695–700, 2001.

[71] H. Rowley. Neural network-based face detection. *PhD Thesis, Carnegie Mellon University*, 1999.

[72] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *In IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, 1998.

[73] H. Sagan. *Space Filling Curves*. Springer, Berlin/Heidelberg/New york, 1994.

[74] H. Sahbi. *Coarse-to-Fine Support Vector Machines for Hierarchical Face Detection*. PhD thesis, Versailles University, 2003.

[75] H. Sahbi, D. Geman, and N. Boujemaa. Face detection using coarse-to-fine support vector classifiers. *In Proceedings of the IEEE International Conference on Image Processing.*, pages 925–928, 2002.

[76] H. Schneiderman and T. Kanade. Object detection using the statistics of parts. *International Journal of Computer Vision*, 2002.

[77] B. Schölkopf, P. Knirsch, A. Smola, and C. Burges. Fast approximation of support vector kernel expansions and an interpretation of clustering as approximation in feature spaces. *In Proceedings of Levi M. Schanz R.-J. Ahlers and F. May editors, Mustererkennung DAGM-Symposium Informatik aktuell*, pages 124–132, 1998.

[78] Thomas B. Sebastian, Philip N. Klein, and Benjamin B. Kimia. Recognition of shapes by editing their shock graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):550–572, May 2004.

[79] T. Sellis, N. Roussopoulos, and C. Faloutsos. The r+-tree: a dynamic index for multi-dimensional objects. In *13th Int. Conf. on Very Large Databases*, pages 507–518, Brighton, GB, 1987.

[80] J. A. Sethian. Fast marching methods. *SIAM Rev.*, 41-2:199–235, 1996.

[81] J. A. Sethian. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Geometry Fluid Mechanics, Computer Vision and Materials Science*. Cambridge University Press, 1999.

[82] K. Siddiqi, B. B. Kimia, and C-W. Shu. Geometric shock-capturing ENO schemes for subpixel interpolation, computation and curve evolution. *Graphical Models and Image Processing*, 59:278–301, 1997.

[83] J. Sobottka and I. Pittas. Segmentation and tracking of faces in color images. *In Proceedings of the International Conference on Automatic Face and Gesture Recognition.*, pages 236–241, 1996.

[84] Yang Song, Luis Goncalves, , and Pietro Perona. Unsupervised learning of human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):814–827, July 2003.

[85] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 246–252, 1999.

[86] Chris Stauffer and W. Eric L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, August 2000.

[87] Chris Stauffer and W.E.L Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*, 1999.

[88] R. Stoica, X. Descombes, and J. Zerubia. A Markov point process for road extraction in remote sensed images. Research Report 3923, INRIA, March 2000. To appear in the Int'l J. Comp. Vis., May 2004.

[89] M. Stonebraker, T. Sellis, and E. Hanson. An analysis of rule indexing implentation in database systems. In *Int. Conf. on Expert Database Systems*, 1986.

[90] M. Sussman and E. Fatemi. An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow. *SIAM J. Sci. Comp.*, 20(4):1165–1191, 1997.

[91] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible 2-phase flow. *J. Comp. Phys.*, 114:146–159, 1994.

[92] Z. Szlávik, L. Havasi, and T. Szirányi. Estimation of common groundplane based on co-motion statistics. In *ICIAR*, 2004.

[93] Z. Szlávik, L. Havasi, and T. Szirányi. Image matching based on co-motion statistics. In *3DPVT*, 2004.

[94] Y. Taniguchi. Moving object detection apparatus and method. *U.S. Patent 5,991,428*, 1999.

[95] F. Tupin, H. Maitre, J-F. Mangin, J-M. Nicolas, and E. Pechersky. Detection of linear features in SAR images: Application to road network extraction. *IEEE Trans. Geoscience and Remote Sensing*, 36(2):434–453, 1998.

[96] M N M van Lieshout and R S Stoica. Exact metropolis-hastings sampling for marked point processes using a c++ library. Technical report, Research Report PNA-R0403, CWI, Amsterdam, June 2004.

[97] A. Vasilevskiy and K. Siddiqi. Flux maximizing geometric flows. *IEEE Trans. Patt. Anal. Mach. Intell.*, 24(12):1565–1578, December 2002.

[98] P. Viola and M. Jones. Robust real-time object detection. (to appear). *Int'l J. Comp. Vis.*, 2002.

[99] visiOprime Ltd., 30 St Johns Road, St Johns, Woking, Surrey, and GU21 7SA URL:www.visioprime.com.

[100] R. Weber, H.J. Schek, and S. Blott. A quantitative analysis and performance study for similarity search methods in high-dimensional spaces. In *24th Int. Conf. on Very Large Databases*, New york, NY, 1998.

[101] D. White and R. Jain. Similarity indexing with the ss-tree. In *12th Int. Conf. on Data Engineering*, New Orleans, LA, 1996.