Frozen-State Hierarchical Annealing

by

Wesley R. Campaigne

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2012

© Wesley R. Campaigne 2012

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

There is significant interest in the synthesis of discrete-state random fields, particularly those possessing structure over a wide range of scales. However, given a model on some finest, pixellated scale, it is computationally very difficult to synthesize both large and small-scale structures, motivating research into hierarchical methods.

This thesis proposes a frozen-state approach to hierarchical modelling, in which simulated annealing is performed on each scale, constrained by the state estimates at the parent scale. The approach leads significant advantages in both modelling flexibility and computational complexity. In particular, a complex structure can be realized with very simple, local, scale-dependent models, and by constraining the domain to be annealed at finer scales to only the uncertain portions of coarser scales, the approach leads to huge improvements in computational complexity. Results are shown for synthesis problems in porous media.

Acknowledgements

There are many people without whom I never would have been able to complete this thesis. First and foremost, I would like to thank my supervisor, Prof. Paul Fieguth, for his patience, support, and guidance over the years. Recognition is also due to Simon Alexander, whose Ph.D. research in hierarchical annealing laid the foundation for the research presented here. Prof. Marios Ioannidis provided the porous media images, and was always a source of enthusiasm.

I would also like to thank my family — in particular, my parents and my grandmother — for their generous support. Through many obstacles, they were unwavering in their belief of what I could accomplish and their willingness to help me reach my goals.

Table of Contents

List of Figures **xi**

1 Introduction **1**

1.1 Thesis Outline . . . 3

2 Background **5**

2.1 Random Fields . . . 5

2.1.1 Markov Random Fields . . . 6

2.1.2 Gibbs Random Fields . . . 6

2.1.3 Gibbs and Metropolis Sampling . . . 8

2.1.4 Simulated Annealing . . . 11

2.2 Hierarchical Methods . . . 13

2.2.1 Hierarchical Representation of Images . . . 14

2.2.2 Hierarchical Random Fields . . . 15

2.2.3 Hierarchies and Computation . . . 15

2.2.4 Hierarchical Annealing . . . 16

2.3 Porous Media . . . 17

3 Frozen-State Hierarchical Annealing **19**

3.1 Regular Hierarchical Annealing . . . 19

3.2 Defining Frozen-State . . . 20

4 Modelling **25**

4.1 Hierarchical Modelling . . . 25

4.2 Local Histogram Energy Function . . . 27

4.3 Methodology . . . 29

4.4 Evaluation of Modelling . . . 31

5 Computation **37**
5.1 Annealing Cooling Schedules 37
5.2 Annealing and Computational Complexity 38
5.3 Computational Complexity of Frozen-State Method 39
  5.3.1 Memory-Efficient Data Structures 40
  5.3.2 Highly Local Energy Model 40
  5.3.3 Propagation of State Information 41
  5.3.4 Tuning of Annealing Cooling Schedules 41
  5.3.5 Reduced Number of Sampled Pixels 42

6 Results and Discussion **47**

7 Conclusions and Recommendations **57**
7.1 Summary of Contributions 57
7.2 Extensions and Future Work 58

References **59**

List of Figures

1.1 Two example images of porous media 2

2.1 Relationship between temperature, energy, and probability within a Gibbs distribution 12

3.1 The principle of ternary decomposition 22

3.2 Ternary downsampling of large and small spheres training data 22

4.1 Example of histogram model data 28

4.2 Illustration highlighting which pixels enter into the energy function . . . 30

4.3 Triangles toy problem 32

4.4 Scale dependance of different models 34

4.5 Ternary hierarchical sampling results for large and small spheres images . 36

5.1 Energy versus iteration and versus computations for a large spheres result 43

5.2 Number of simulated pixels as a function of scale 45

6.1 Comparison of a full-size sampling result with the original image for sintered glass beads 48

6.2 Comparison of a full-size sampling result with the original image for carbonate rock 49

6.3 Comparison of the number and fraction of pixels to simulate for the results in Figs. 6.1 and 6.2 50

6.4 A comparison the proposed ternary-hierarchical method with existing techniques 51

6.5 A zoom-in comparison of actual data and two sets of hierarchical results 52

6.6 Two-model hidden field example with a sinusoidal boundary 54

6.7 Hidden fields containing vertical bands combined with hybrid models to create a left-to-right gradient 55

6.8 Hidden fields containing circular bands combined with hybrid models to create a middle-to-edges gradient 56

Chapter 1

Introduction

The synthesis of large, binary random fields has become an area of substantial interest, particularly so in the study of porous media [31, 37], materials characterized by complex, multiscale, binary structures, for which two examples are shown in Figure 1.1.

The essential challenge is how to construct a model for a given field in order to artificially synthesize further random samples, for two reasons:

1. Although it would be preferable to study *physical* samples, there is considerable time and expense associated with sample preparation and high resolution imaging. Furthermore aspects of sample preparation (cutting, polishing, exposure to air) may alter the original sample. Worst of all, the ultimate goal is to study three dimensional samples, for which thousands of repeated cutting/polishing passes are completely impractical, and three-dimensional imaging by MRI or tomographic methods so far yields samples at only very low resolution.
2. In order to analyze the macroscopic, aggregate behaviour of a material, we need multiple, large samples to study. To determine the permeability or conductivity of the medium with respect to a fluid, detailed simulations of the interactions between the fluid, the pore structures, and the medium must be performed; it is impossible to derive such values purely from the fluid's properties. These calculations must be done in aggregate across a sample; it does not make sense to discuss point-wise value of such properties. Furthermore, the concern is simulating how a given fluid interacts with the medium *on average*, not just with respect to a single sample. Arriving at a meaningful average requires simulating over *many* samples — ideally, one would conduct a Monte Carlo simulation [25] using a large set of sample images — and acquiring such a large set through physical imaging is impractical.

The problem is further complicated by the requirement that the samples be large, in order to express the behaviour on all of the scales. If the synthesis is fine-scale, but too small to show the large-scale structures or the full diversity of features, the conclusions one might draw from it are worthless. At the same time, if the resolution is not high enough — if the finest-scale features are not clearly visible in the synthesized image — simulating

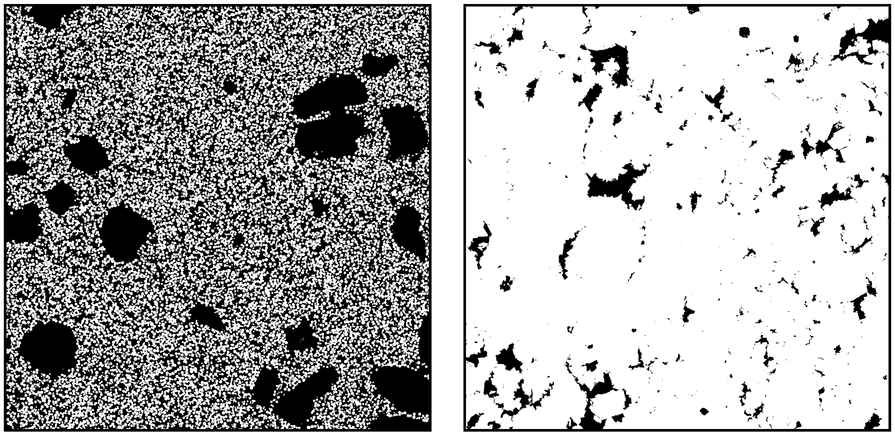

(a) Sintered glass beads (b) Carbonate rock

Figure 1.1: Two examples of large 8192 × 8192 microscopic images having complex, multiscale structure.

the fluid–medium interactions becomes impossible; for example, to model water going through a bucket of sand, the pixels need to be fine enough to show the individual grains of sand and the spaces between the grains.

A great number of approaches and methods have been proposed [36, 37, 39], however all of these methods suffer from either limits on modelling complexity, such that subtle features of the porous medium cannot be represented and synthesized by the selected model, or a limit on computational complexity, such that computational complexity limits the size or accuracy of the produced sample. In this thesis we report on an approach which offers tremendous improvements in both modelling and computational complexity, allowing rapid synthesis of huge multiscale two-dimensional samples, and offering the possibility of large-scale three-dimensional synthesis.

The standard approach to image sampling is based on simulated annealing [20], in which we repeatedly visit the pixels of a random field, using the Metropolis or Gibbs [14] sampler to update each pixel on the basis of an energy function E and an annealing temperature T_k.

In most cases, a given pixel in a random field most strongly interacts with its local neighbours, and therefore it is exceptionally difficult (i.e., slow) to synthesize structures large in size relative to the local neighbourhood.

In response to this observation, we are motivated to model a binary field in some sort of hierarchical representation, for reasons of computational efficiency, to be able to synthesize large structures in an easier fashion at a coarser scale. There are two fundamental ways in which to construct a hierarchy:

1. **Top-down,** in which the hierarchy begins at a coarse scale, with the coarse-scale elements repeatedly refined at finer scales.

2. **Bottom-up,** in which the hierarchy begins at the finest, pixellated scale, and where some sort of grouping or aggregation leads to coarser representations.

For discrete-state Markov Chain Monte Carlo (MCMC) problems [42], the bottom-up approach is considerably more common (such as Swendsen-Wang [34] and region-grouping [38] methods), in contrast to continuous-state image processing problems, in which top-down approaches dominate (wavelets [11, 22, 44, 45], Laplacian pyramids [6] etc.). There are two reasons for this distinction:

1. In the continuous-state case, a coarse scale can represent a smooth, low-resolution image, which is then nudged and refined towards finer scales. However a discrete hierarchy does not allow for a smoothly-varying representation or for small refinements in state value from scale to scale.
2. In a top-down representation, a coarse-scale state element represents some square subset of the finest-scale domain. Since the finest scale will normally not be made up of piecewise-constant squares, the imposition of a regular grid from a top-down structure is not a natural fit, as opposed to the adaptive, irregular regions produced by bottom-up approaches.

Despite the above limitations and liabilities, this thesis proposes a top-down hierarchical approach for the modelling and synthesis of binary random fields.

We need to be clear that existing methods such as wavelet image synthesis [29], random fields synthesis using Fast Fourier transforms [10], and fast texture rendering methods from the computer graphics literature are all effective in their fields, efficiently producing rendered images satisfying aesthetic requirements, but which cannot however be argued to quantitatively satisfy a scientific discrete-state model. In scientific image synthesis we require a verifiable model, therefore heuristic image synthesis and enhancement methods are inappropriate.

There are two key contributions in this thesis, one for each of modelling and computational complexity. First, a top-down hierarchy gives us a regular grid on a sequence of scales; the random field on each scale can be modelled by a scale-dependent model. Since even large-scale phenomena are local on a sufficiently coarse scale, it is possible to use relatively simple, *local* models on *all* scales to represent complex behaviour. Next, we propose a "frozen-state" approach, by which confident portions of coarser scales are frozen in place and cannot be modified at finer scales. This has advantages both in modelling, preventing finer-scale models from undoing or eroding structures put into place at coarser scales, and also a great advantage computationally, in that at any given scale only those "unfrozen" state elements need to be simulated. With these advantages, the proposed method is able to synthesize huge 8192 × 8192 images possessing multi-scale structures on regular computers in a few hours of computation time.

1.1 Thesis Outline

The following chapters are organized as follows: Background material is discussed in Chapter 2. The frozen-state hierarchy, introduced above, will be discussed in Chapter 3.

Modelling considerations, including an examination of a simple, scale-dependent local model, will be discussed in Chapter 4. Computational aspects follow in Chapter 5. Throughout the thesis, incremental results are presented; a set of large-scale results based on actual data are presented and examined in greater depth in Chapter 6. Finally, conclusions and recommendations are presented in Chapter 7.

Chapter 2

Background

This chapter outlines the background material that provides the foundation for the work presented in the following chapter. The focus is principally divided between overviews of random fields and of hierarchical methods. A brief introduction to the study of porous media is also included.

2.1 Random Fields

A random field is a way to represent a set of random variables arranged on a lattice [10]. Specifically, given a lattice L, then

$$\mathbf{x} = \{x_i \in \Psi | i \in L\} \tag{2.1}$$

is a random field, where each x_i is the random variable corresponding to site i, and Ψ represents the range of values these variables can take. In principle, the lattice can be any collection of discrete points, in any number of dimensions. The values within the field, Ψ, is similarly flexible: Ψ can be discrete or continuous, and finite or infinite.

As each element is a random variable, we can therefore say x_i obeys some probability measure $p(x_i)$, and there exists a joint probability measure $p(\mathbf{x})$ for the field as a whole. The form of $p(\mathbf{x})$ depends on the form of $p(x_i)$, which in turn depends on Ψ: if Ψ is a continuous set, then $p(x_i)$ and $p(\mathbf{x})$ denote probability density functions; if Ψ is discrete, then $p(x_i)$ and $p(\mathbf{x})$ are probability distributions. As with many other topics in probability, the tools and approaches one can use are often constrained by this difference.

This thesis is focused on random fields where Ψ is discrete and finite, and the lattice is a two-dimensional regular array

$$L = \{(i, j) | 1 \leq i \leq n_1, 1 \leq j \leq n_2\}. \tag{2.2}$$

This can be thought of as a two-dimensional image composed of random, discrete-valued pixels

$$\mathbf{x} = \{x_{i,j} | (i, j) \in L\}. \tag{2.3}$$

The primary challenge in random fields is the representation and manipulation of the joint distribution $P(\mathbf{x})$. For all but the most trivial cases $P(\mathbf{x})$ is too complex to be computed explicitly. Instead, we rely on methods and formulations that represent $P(\mathbf{x})$ implicitly such that we can examine or manipulate it in an efficient manner.

2.1.1 Markov Random Fields

A Markov random field (MRF) [10, 14, 42] can be viewed as a generalization of Markov chains; specifically, an MRF asserts a random field that behaves according to a generalized version of the conditional decoupling property of Markov chains, i.e., the statistics governing an element are determined by its neighbouring region, and are conditionally independent from the rest of the field.

Specifically, given each site i has a neighbourhood structure $\{\mathcal{N}_i\}$, we can state that $\mathbf{x}$ is a Markov random field if

$$p(x_i|\{x_j, j \in L \setminus i\}) = p(x_i|\{x_j, j \in \mathcal{N}_i\}), \forall i \in L, \tag{2.4}$$

provided that, also $\forall i \in L$, a site is not its own neighbour,

$$i \notin \mathcal{N}_i \tag{2.5}$$

and neighbours have a reciprocal relationship,

$$j \in \mathcal{N}_i \Leftrightarrow i \in \mathcal{N}_j. \tag{2.6}$$

The usefulness of MRFs stems from how this decoupling provides a means of handling the large, complex joint distributions inherent in random fields; the ensemble of these much simpler conditional distributions *implicitly* represents the same statistics. We can use iterative methods that manipulate them in such a way that the results converge on values that are consistent with the joint distribution, without ever needing to compute it.

Technically, the formulation of MRFs applies equally well to both discrete and continuous state random fields; however, most methods apply only to continuous MRFs, particularly the ubiquitous Gauss-Markov random field. For discrete problems, almost universally, the Gibbs random field approach is preferred.

2.1.2 Gibbs Random Fields

A Gibbs random field (GRF) [4, 10, 14, 42] is another formulation that, like Markov random fields, expresses the statistics governing the field as the implicit result of a set of simpler interactions. GRFs were originally developed as a tool for statistical mechanics [8, 41], where their use continues to be widespread. Even when applied in other domains, such as image processing [10, 14], this origin in physics can be seen in how it expresses the distribution of the field implicitly as a function of ‘temperature’ and ‘energy’.

Specifically, Gibbs random fields behave according to the Gibbs distribution,

$$p(\mathbf{x}) = \frac{1}{\mathbb{Z}} \exp\big(- E(\mathbf{x})/T \big), \tag{2.7}$$

given some energy function, $E(\mathbf{x})$, that measures of the 'energy' of state x. In physics, this is typically calculated as the sum of the energy of all pairwise interactions between neighbouring particles; in image processing, it is usually the sum of some measure that relates the value of each pixel to those of its neighbours. $\mathbb{Z}$ is known as the partition function, and normalizes the expression into a probability by taking the sum of energies of all possible configurations, i.e.,

$$\mathbb{Z} = \sum_{\mathbf{x}'} \exp\big(- E(\mathbf{x}')/T \big), \tag{2.8}$$

for discrete-valued $\mathbf{x}$, or the corresponding integral if the elements of $\mathbf{x}$ take continuous values. Lastly, the parameter T (where $T > 0$) describes the temperature for the field. When modelling a thermodynamic system, the probability of the system having a certain configuration can only be answered given the temperature of the system, as changes in temperature alter the distribution — low temperatures tend be structured and have a probability space dominated by the low-energy configurations, while high temperatures are more chaotic and show an increase in the probability of higher-energy configurations. This ability to use T to tune the relationship between energy and probability distributions is also useful to non-physics applications and, as will be discussed in Section 2.1.4, is key to simulated annealing methods for sampling and optimizing GRFs [10, 14].

One particularly notable instance of a GRF (and perhaps the simplest non-trivial one) is the classic Ising model, which was used to understand the behaviour of magnetic domains in ferromagnetic materials [8, 21, 42]. In its simplest form in two dimensions, this model is

$$E(\mathbf{x}) = -J \sum_{i,j} (x_{i,j} x_{i+1,j} + x_{i,j} x_{i,j+1}) \tag{2.9}$$

where each $x_{i,j}$ has a value in $+1, -1$, representing the binary magnetic spin of an atom in the lattice. $J > 0$ acts as a coupling constant, corresponding to the strength of an external magnetic field. Thus, the entire model derives from the simple question of whether sites agree or disagree with the states of their immediate neighbours. Despite this seeming simplicity, the Ising model exhibits a range of complex behaviours, particularly with regard to phase-transitions, and its properties are still active areas of research 90 years after its introduction [23, 30].

Although very simple energy functions can give rise to complex phenomena, they need not be that simple. A key strength of the Gibbs representation is the flexibility in defining $E(\mathbf{x})$: under Equation 2.7, essentially *any* real function will define a valid probability distribution $p(\mathbf{x})$, without a need to ensure any kind of validity constraints are satisfied (e.g., positive definiteness of the covariance matrix). This gives it excellent flexibility as a modelling tool. The challenge, then, is in finding an energy function $E(\mathbf{x})$ that implies the the desired kind of $p(\mathbf{x})$.

As mentioned above, the vast majority of energy functions have some notion of locality. Strictly speaking, the full definition of a Gibbs random field includes such a constraint, and requires that the energy function be expressible as a sum of the energies of the local interactions for each unique subset of jointly neighbouring sites. In this manner, Gibbs and Markov random fields can be very similar; in fact, the Hammersley–Clifford theorem establishes that, given certain minor constraints, Gibbs and Markov random fields are equivalent [4, 10].

While this equivalence is often useful, there is nothing about the general formulation of the Gibbs distribution that *requires* $E(x)$ be composed exclusively of local interactions. One can quite easily define a non-local energy function that involves some global statistics of the state of the field. A model based on such a function would have no clear Markovian equivalent, but would be easily defined and manipulated as a Gibbs random field. Models of this sort are explored in depth in [10] and [37].

The main challenge in handling Gibbs distributions is their reliance on the normalizing constant, $\mathbb{Z}$. Except in for a few extremely simple instances where analytic solutions have been devised, calculating $\mathbb{Z}$ is intractable. Consequently, it is impossible to directly answer questions related to the full joint distribution, or to assess its plausibility.

We can, however, easily calculate the *relative* likelihood of two states,

$$\frac{p(\mathbf{x}_1)}{p(\mathbf{x}_2)} = \frac{\exp\big(-E(\mathbf{x}_1)/T\big)/\mathbb{Z}}{\exp\big(-E(\mathbf{x}_2)/T\big)/\mathbb{Z}} = \exp\left(\frac{-[E(\mathbf{x}_1) - E(\mathbf{x}_2)]}{T}\right), \tag{2.10}$$

and this connection between differences in energy and relative likelihoods forms the basis for all practical computational methods for Gibbs random fields and their applications.

Specifically, the two fundamental operations one tends to perform using Gibbs random fields are

1. the generation of random samples from the distribution, and
2. optimization in the energy function to find a minimum-energy state.

The former notion is central to a huge proportion of the literature on Markov Chain Monte Carlo (MCMC) methods [10, 18, 37, 46], while the latter can be used to find maximum-likelihood estimates in the model, $\hat{\mathbf{x}}$, given some observations or constraints [42] — a very common objective in image processing, with applications in areas such as image reconstruction [14], enhancement [43], and segmentation [40]. The basic methods for these operations are outline in the next two sections.

2.1.3 Gibbs and Metropolis Sampling

The most fundamental methods for producing a sample $\mathbf{x}$ from a Gibbs distribution $p(\mathbf{x})$ are the Metropolis sampler [10, 18, 20, 24] (also known as the Metropolis–Hastings algorithm) and the related Gibbs sampler [10, 14]. Both algorithms function by iterating over the elements of $\mathbf{x}$ (typically with some randomness to the ordering) and resampling the value at each element. Through these resamplings, both algorithms will produce, under certain conditions, a sequence of states of $\mathbf{x}$ that will provably converge on the true distribution $p(\mathbf{x})$.

In Metropolis sampling, a new state $\bar{\mathbf{x}}$ is proposed based on the current state $\mathbf{x}$ (e.g., by randomly selecting a site $k \in L$ and a new value for it $\bar{x}_k \in \Psi \setminus x_i$). The probability that this proposal is accepted is

$$\Pr(\mathbf{x} \Rightarrow \bar{\mathbf{x}}) = \begin{cases} 1 & \text{if } p(\bar{\mathbf{x}}) > p(\mathbf{x}) \\ \frac{p(\bar{\mathbf{x}})}{p(\mathbf{x})} & \text{if } p(\bar{\mathbf{x}}) \leq p(\mathbf{x}) \end{cases} \tag{2.11}$$

which, in the case of $\mathbf{x}$ being a Gibbs random field (and therefore $p(\mathbf{x})$ being of the form given in Equation 2.7), becomes

$$\Pr(\mathbf{x} \Rightarrow \bar{\mathbf{x}}) = \begin{cases} 1 & \text{if } E(\bar{\mathbf{x}}) < E(\mathbf{x}) \\ \exp\left(\frac{-[E(\bar{\mathbf{x}}) - E(\mathbf{x})]}{T}\right) & \text{if } E(\bar{\mathbf{x}}) \geq E(\mathbf{x}) \end{cases} \tag{2.12}$$

The full algorithm for the Metropolis sampler is expressed in Algorithms 1 and 2. Algorithm 3 gives the special case simplification of Algorithm 2 for when $\mathbf{x}$ is a Gibbs random field.

Gibbs sampling is conceptually very similar to Metropolis sampling. The difference lies in how new states are generated: whereas the Metropolis sampler has discrete steps for proposing a *specific* new state and then probabilistically choosing whether to accept it, the Gibbs sampler integrates the two components. Specifically, in Gibbs sampling, we choose just the site k, and *directly* sample from the marginal distribution for x_k conditioned on the rest of $\mathbf{x}$:

$$\Pr(x_k \to \alpha | \mathbf{x}_{\setminus k}) = \frac{p(\mathbf{x} \setminus x_k = \alpha)}{\mathbf{x}_{\setminus k}} = \frac{p(\mathbf{x} \setminus x_k = \alpha)}{\sum_{\beta \in \Psi} p(\mathbf{x} \setminus x_k = \beta)}. \tag{2.13}$$

Notice how both samplers operate on a ratio of likelihoods which, when applied to a Gibbs distribution, results in the cancellation of the partition function $\mathbb{Z}$ (as shown in Equation 2.10), and thus makes the distribution computable.

Algorithm 1 Metropolis Sampler

```
Randomly initialize $\mathbf{x}_{(0)}$
$t \leftarrow 0$
repeat
    Randomly select site $k \in L$,
    Randomly select new value $\psi$ for site $k$, where $\psi \in \Psi, \psi \neq x_{k,(t)}$
    Propose a new state $\bar{\mathbf{x}}$ by copying $\mathbf{x}_{(t)}$ but substituting in value $\psi$ at site $k$
    $\mathbf{x}_{(t+1)} \leftarrow$ SingleMetropolis($\mathbf{x}_{(t)}$, $\bar{\mathbf{x}}$, $p()$)
    $t \leftarrow t + 1$
until stopping criteria are satisfied
```

Algorithm 2 Single Metropolis Sample

```
function SingleMetropolis($\mathbf{x}$, $\bar{\mathbf{x}}$, $p()$)
    $A \leftarrow p(\bar{\mathbf{x}})/p(\mathbf{x})$
    if $A > 1$ then
        $\hat{\mathbf{x}} \leftarrow \bar{\mathbf{x}}$
    else
        $r \leftarrow$ UniformRandom()
        if $r < A$ then
            $\hat{\mathbf{x}} \leftarrow \bar{\mathbf{x}}$
        else
            $\hat{\mathbf{x}} \leftarrow \mathbf{x}$
        end if
    end if
    return $\hat{\mathbf{x}}$
end function
```

Algorithm 3 Single Metropolis Sample for Gibbs Random Fields

```
function SingleMetropolis($\mathbf{x}$, $\bar{\mathbf{x}}$, $E()$, $T$)
    $e \leftarrow E(\mathbf{x})$
    $\bar{e} \leftarrow E(\bar{\mathbf{x}})$
    $r \leftarrow$ UniformRandom()
    if $r < \exp\left(\frac{e-\bar{e}}{T}\right)$ then
        $\hat{\mathbf{x}} \leftarrow \bar{\mathbf{x}}$
    else
        $\hat{\mathbf{x}} \leftarrow \mathbf{x}$
    end if
    return $\hat{\mathbf{x}}$
end function
```

Gibbs sampling is generally regarded as faster to converge [13, 42], as it considers all possible values $x_k \in \Psi$ each time site k is visited. This also makes Gibbs sampling more limited; if Ψ is large, calculating the denominator of Equation 2.13 may be unpractical. One can also implement more elaborate state-proposal methods that alter multiple elements simultaneously [34, 40]: with the Metropolis sampler, the decision whether to accept or reject such proposals is no different from the basic single-change case, while the Gibbs sampler may become problematic with an analogous proposal scheme due to its need to iterate over the full set of possible new values.

2.1.4 Simulated Annealing

Metropolis and Gibbs sampling both allow us to produce random samples from a Gibbs distribution $p(\mathbf{x})$, however, in very many cases, we are interesting in only the tiny subset of the *most likely states* of $\mathbf{x}$, and therefore we seek to find $\mathbf{x}$ that maximizes $p(\mathbf{x})$ (and therefore also minimizes the energy function $E(\mathbf{x})$). A near universal approach to this is the method of simulated annealing, first developed by Kirkpatrick, Gelatt, and Vecchi [20], and later introduced to image processing contexts by Geman and Geman [14], where it has since seen widespread use.

In the annealing process of metallurgy, a metal is heated to high temperature then cooled at a controlled rate to toughen the sample. At high temperatures, existing crystalline formations are erased; as it cools, it recrystallizes. The size of the crystal domains is a function of the cooling rate: slower cooling allows the crystals to form more gradually, enabling the formation of larger domains, and thus a lower overall energy state.

Simulated annealing works in much the same fashion. Recall that, on their own, the Metropolis and Gibbs samplers mimic thermodynamic distributions operating at a fixed temperature. The Gibbs distribution's temperature parameter T has the effect of tuning the relative likelihoods of different energy states. Figure 2.1 illustrates the effect of temperature on energy and probability: higher temperatures increase uniformity in the probability space, while at lower temperatures, the differences in likelihood caused by variations in energy become more pronounced. This tuning by temperature has a significant effect on the output of the Metropolis and Gibbs samplers, where the state transitions are entirely dependent on these relative likelihoods.

Simulated annealing simply extends the Metropolis and Gibbs samplers through observing this relationship between cooling and convergence on a low-energy state. By beginning the sampling at a high temperature, we start the sampling with, effectively, an almost uniformly random $p(\mathbf{x})$. As T is slowly decreased, changes to $\mathbf{x}$ that *increase* the energy gradually become less likely over time, and the state of $\mathbf{x}$ converges on a subset of low-energy states.

The change in value of T as a function of time is known as the annealing schedule. When T is decreased sufficiently slowly, it can be guaranteed that the resulting $\mathbf{x}$ has globally minimal energy [14], but the temperature schedules which guarantee this outcome are far too slow to be used in practice. Instead, T is normally decreased more rapidly (a process known as quenching), and a more approximate optimum is found

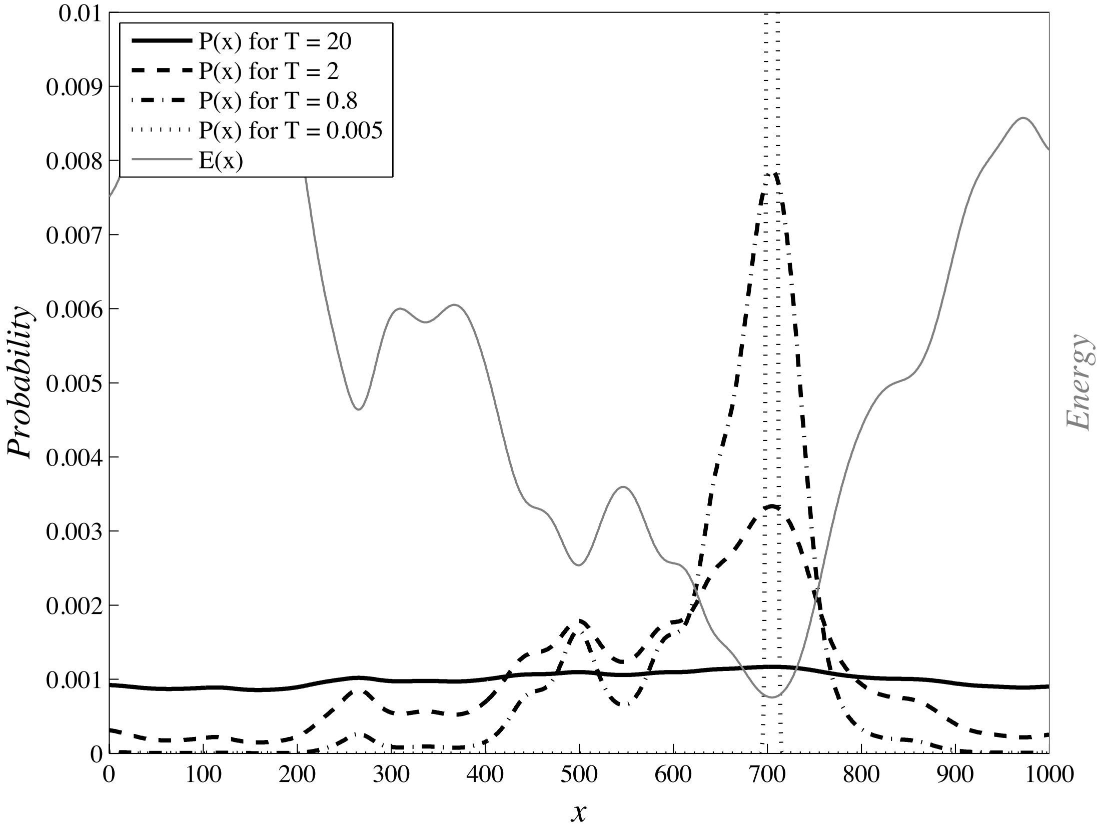


Figure 2.1: The Gibbs probability distributions at various temperatures for an arbitrary one-dimensional energy function $E(x)$. Note how as $T \to \infty$, the significance of $E(x)$ decreases, and $P(x)$ approaches uniformity. As T decreases, peaks form in $P(x)$ that correspond to minima in $E(x)$. As $T \to 0$, $P(x)$ becomes a singularity at the global minimum of $E(x)$, and 0 elsewhere.

[14, 20, 26]. The overall procedure for simulated annealing using the Metropolis sampler is given in Algorithm 4.

The only elements needed to make practical use of simulated annealing are a temperature schedule and an energy function, both of which can be quite simple. This ease of use has led to widespread applications of simulated annealing, both as a sampling method [1, 10, 36, 46] and a stochastic optimization method [14, 40, 42].

2.2 Hierarchical Methods

A wide variety of hierarchical approaches [6, 19, 22, 28] have been proposed as a means of ameliorating the computational issues of representing and modelling very large images. It would be impossible to survey a significant fraction in this thesis, and so for our purposes we will only give a quick overview.

Hierarchical methods serve two basic purposes: they enable modelling improvements and computational acceleration. These two topics will be discussed in Chapters 4 and 5 respectively.

The fundamental idea behind hierarchies in modelling is that it is difficult to simultaneously describe local details and large-scale behaviour. Representing both these aspects simultaneously in an image requires very large, high-resolution images composed of many pixels — too many pixels to be able to easily describe their interrelationships. Consequently, from a modelling perspective, decomposing the problem into pieces is an attractive idea.

There exists a completely separate section of the literature that is focused on the acceleration of computation through the use of a hierarchical approach [5, 15, 17, 47]. For instance, random fields methods on very large domains can suffer from very long computational times. For these types of problems, there is something attractive about solving a reduced-order problem at the coarse scale of a hierarchy as a way of initializing or approximating a fine-scale solution. In these instances, the hierarchy is not used to

Algorithm 4 Simulated Annealing with the Metropolis Sampler

```
Randomly initialize $\mathbf{x}_{(0)}$
$t \leftarrow 0$
$T \leftarrow T_{(0)}$
repeat
    Randomly select site $k \in L$,
    Randomly select new value $\psi$ for site $k$, where $\psi \in \Psi, \psi \neq x_{k,(t)}$
    Propose a new state $\bar{\mathbf{x}}$ by copying $\mathbf{x}_{(t)}$ but substituting in value $\psi$ at site $k$
    $\mathbf{x}_{(t+1)} \leftarrow$ SingleMetropolis($\mathbf{x}_{(t)}$, $\bar{\mathbf{x}}$, $E()$, $T$)
    $t \leftarrow t + 1$
    $T \leftarrow T_{(t)}$
until stopping criteria are satisfied
```

examine a coarse representation; rather, it serves only as a shortcut towards achieving a certain state at the finest scale.

2.2.1 Hierarchical Representation of Images

A great deal of work on hierarchical representation of images has been done for two-dimensional images (that is, two-dimensional sets of pixels), although the focus for much of that work has been on real-valued images (in other words, digital photographs) and not necessarily random fields. These kinds of hierarchical transformations may be used for the purposes of image classification, image segmentation, image modelling, image representation, among others. In terms of modelling, there exist a wide variety of non-local features to analyze and represent spacial structure. Examples of these methods include multi-scale local binary patterns [28], wavelet methods [22, 33], Gabor filters [19], Laplacian and Gaussian pyramids [6].

Each of these methods, fundamentally, is a different answer to the question: given a fine-scale image $I_{x,y}$, what is an appropriate way of defining the coarser-scale representations? We shall let I^s be used to denote the representation of image I at scale s, with s increasing for progressively coarser scales, and $s = 1$ denoting the finest scale.

Most approaches define, either implicitly or explicitly, some kind of decimation or approximation operator, A, such that $I^s = A(I^{s-1})$ — implying that repeated applications of this operator gives us repeatedly coarser fields.

Perhaps the simplest and most obvious approach to defining A is subsampling by using as a simple majority vote among adjacent pixels with a tie-breaking mechanism. Gaussian pyramids create a hierarchy by defining A as a convolution with a low-pass filter followed by a downsampling:

$$I_G^s = \text{downsample}(H * I_G^{s-1}). \tag{2.14}$$

The scales of a Laplacian pyramid are formed by taking the difference between adjacent scales in a Gaussian pyramid,

$$I_L^s = I_G^s - \text{interpolate}(I_G^{s+1}), \tag{2.15}$$

which has the effect of decomposing I by a series of band-pass filters, eliminating much of the redundant information from the hierarchy. Methods based on Gabor filters do not explicitly construct I^s but instead decompose the image into coefficients for any array of filters that depend on scale *and* orientation; the information at scale s is divided amongst the filters that encompass the corresponding frequencies.

Wavelets, like the Laplacian pyramid, are motivated by a desire to create a hierarchy where each I^s contains *only* the information for scale s. Clearly, if one can compute I^s from I^{s-1}, then the two are not independent, and not a basis. The wavelet domain asks the question, how can we construct a scale-dependent basis? We again start with a multi-resolution image pyramid based on an approximation operator,

$$I^s = A(I^{s-1}) = A(A(I^{s-2})) = \ldots, \tag{2.16}$$

as we did for the Laplacian/Gaussian case, but from that we want to construct an orthogonal set, in order to get a basis. To do so, it turns out we need another special operator — a detail or differencing operator, D, that is orthogonal to A — such that $W^s = D(I^s)$ is orthogonal to $I^{s+1} = A(I^s)$, and $I^s = W^s + I^{s+1}$. Iterating this relationship gives us a full decomposition,

$$I^s = W^s + W^{s+1} + \cdots + W^{s+n-1} + I^{s+n}, \tag{2.17}$$

wherein each W term is orthogonal to the rest, and they collectively form a multiresolution basis.

2.2.2 Hierarchical Random Fields

The types of hierarchical representations possible for random fields are primarily determined by whether the elements of the field take continuous or discrete values.

For continuous-valued random fields, the techniques for representing images can often offer inspiration. One can, for instance, create a coarse representation of a field by simply averaging together elements of the fine scale — effectively downsampling it. Using multigrid methods [5, 17], one can take the additional step of subtracting out the coarse representation from the finer scale, and represent the field as hierarchical set of averages and differences, similar to wavelets. Despite these similarities, continuous random field models are still difficult to handle hierarchically, because the act of changing scale alters the statistics governing the field. For instance, downsampling an Markov random field does *not* give an MRF; the resulting lower-resolution field is close to but does not quite satisfy the MRF requirement for conditional independence.

The hierarchical representation of discrete random fields poses additional problems. Since the field is discrete, it is not possible to use an average of finer-scale values as a coarse-scale representation. There is substantial literature on the topic of Renormalized Ising models [8, 12, 41] — that is, Ising models at reduced resolutions. This offers one example of how to construct a lower-resolution discrete field. In these models, the resulting low-resolution field *is* itself another Ising random field, however, it is *not* a low-resolution version of the initial field. Gidas formulated what the energy function looks like for the reduced-resolution random field [16], however a problem remains: much as the subsampling of a MRF is not Markov, there is a corresponding relationship for Gibbs random fields. The end result is that we go from modelling only local relationships at high-resolution to having to model non-local behaviours at the subsampled resolution, which is a much more complicated problem.

The question of modelling representation of random fields will be looked at in Chapter 4.

2.2.3 Hierarchies and Computation

The use of hierarchical methods for algorithmic acceleration has long history within the field of preconditioning methods [15, 47]. If one needs to solve a large linear system, it turns out, by performing a change of basis of that system into a hierarchical basis, one can solve the problem much faster.

This process takes a problem that is mathematically stiff (that is, slow to converge) and accelerates its convergence. This is extremely similar, conceptually, to the work done in this thesis; much like solving linear systems, simulated annealing or Gibbs sampling on a large flat domain are algorithmically stiff, and by recasting that domain onto a hierarchy and doing sampling hierarchically one can achieve far faster convergence.

One famous example of this is in the topic of orthogonal transformations and the choice between the Fast Fourier Transform (FFT) and the wavelet transformation. Computing the FFT is a $O(n \log n)$ operation, while the wavelet transform requires only $O(n)$ computations; the $\log(n)$ computation difference between these two stems exactly from the fact that the wavelet transform is hierarchical in nature, and its computational complexity decreases geometrically with scale and therefore the overall complexity grows only linearly in problem size.

Another example is multigrid methods [5, 17], which solve linear systems by iteratively moving the solver between the fine scale system and coarser, subsampled versions of it. This allows the solver to make large-scale changes at a coarse scale where they are local and require significantly fewer iterations, the results of which it then projects back down the hierarchy. By doing some fraction of the work at a coarser scale, the amount of work remaining at a finer scale is much less, and because there are fewer pixels at a coarser scale, there is an overall reduction in complexity.

2.2.4 Hierarchical Annealing

In practice, the methods described in Section 2.1 run into limits because of computational complexity, particularly when modelling highly multiscale phenomena. Chapter 5 contains a detailed discussion of these limitations; the core problem is that iterative methods operating on single pixels require an unreasonably large number of iterations to converge for models containing complex structures that are thousands of pixels in size.

Hierarchical annealing [1, 2, 7, 10] offers a way around many of these limitations by asking the question: how can we perform annealing such that structures of each scale can be created at a resolution where they are relatively small and local and would require relatively few iterations. The method creates a hierarchy of subsampled versions of the finest-scale modelling problem, and takes a top-down approach to sampling from them. Starting from an arbitrarily coarse scale and moving the finest, each scale is annealed according to an appropriately rescaled version of the model. The result from each scale is then upscaled and used to initialize the next finer scale. By using a suitably tuned schedule of temperatures, the annealer at fine scales will preserve the already-created, coarser-scale structures while filling in the finer structures made possible by the greater resolution of the finer scale. Consequently, at the finest scale, even though the annealing may still involve simulating millions of pixels, relatively few iterations are needed because the larger structure are already present in the image — information only needs to flow distances of a few pixels, to create those fine, local details that are visible only at the finest scale. This method will be discussed in more detail in Section 3.1, and further examination of the computational complexity of hierarchical annealing can be found in Chapter 5.

As mentioned in Chapter 1, there are alternative, bottom-up approaches that involve building ever-changing groups of finest-scale pixels and manipulating their states collectively. These methods (in particular the Swenson–Wong algorithm [34]) have certain attractive properties but will not be used in this thesis.

2.3 Porous Media

The definition of a porous medium is, roughly, any material in which a solid phase is permeated by a network of pore structures through which fluids are able to flow [37]. This definition applies to a huge range of both natural and artificial materials. The nature of the pore structures in each, and the resulting interactions with fluids, have significant effects for many areas of research:

1. Earthen materials such as sand, soil, and clay, whose porous features determine the dynamics of groundwater contamination and subsurface flow;
2. Construction materials, including cement and concrete, where porous properties can address questions such as cracking and material lifetime;
3. Biological materials such as bone and cartilage, where understanding the porous behaviours could lead to medical improvements, such as in drug delivery;
4. Plus many others, such as wood, skin, and different types of rock.

In short, the often random, microscopic structures in a material can have major implications in its macroscopic properties. The study of porous media is principally to understand these relationships. Generally speaking, these structures and the resulting interactions between fluid and solid are far too complex for analytic solutions, however, Monte Carlo simulations provide an alternative: given high-resolution images showing the pore structures, model and simulate the physics of these interactions, and from the model measure the emergent properties or phenomena of interest.

This poses a problem: if these Monte Carlo simulations are to produce robust, meaningful results, they must be run over large sets of data. Unfortunately, acquiring two- and three-dimensional images at sufficient size and resolution to capture the pore structures is often quite expensive, even for individual images; the cost of acquiring them at the volumes needed for the simulations puts it out of the question.

Instead, the research community has turned to image synthesis [36, 37, 46]. The idea here is to take a single binary image of a physical sample — wherein a value of 0 indicates pore and 1 indicates a solid phase — then infer a model of the pore structures from this image, such that one can generate random samples from that model, as needed, to serve as input for the Monte Carlo simulations. A related question is one of posterior sampling, which seeks to combine low-resolution observations with high-resolution models of the phenomena to create plausible realizations that are consistent with both.

What are these pore structures we are modelling? Most porous media are fractal-like, showing structures on a wide range of length scales. Thus, they are a natural fit for the hierarchical models being proposed in this thesis. However, most established models in the literature of porous media only operate at the pixel level, at the finest scale. Two of the most common models are based on the two-point correlation distribution and the chordlength distribution [36, 37, 46].

The two-point correlation model seeks to recreate the two-point correlation distribution of some training image, which expresses the correlation between pixels in an image as a function of the offset between them. If $\langle\cdot\rangle$ denotes a spatial average, then the two-point correlation of an image $x_{i,j}$ is given by

$$S(r_1, r_2) = \langle x_{i+r_1,j+r_2} x_{i,j} \rangle. \tag{2.18}$$

The two-point correlation model's energy function is simply a norm of the difference between this distribution in the sampled image and the target distribution (taken from the training image, which will be denoted using $\hat{\cdot}$):

$$E = \sum_{r=1}^{r_{max}} \left\| \hat{S}(0,r) - S(0,r) \right\| + \left\| \hat{S}(r,0) - S(r,0) \right\|. \tag{2.19}$$

Notice that only the correlations along strictly vertical and horizontal offsets are used; this is done to reduce computations. Similarly, the l_2–norm is generally used, although other norms are possible.

The chordlength model operates quite similarly; it seeks to match the chordlength distribution of the target image. The chordlength distribution is the distribution of contiguous "runs" of pixels of a given value, in a given direction, as a function of the length of the run. Let us denote these probability mass functions as $\mathcal{L}_{h,z}$ and $\mathcal{L}_{v,z}$ for the respective horizontal and vertical distributions of chordlength for pixels of value z. The energy function can thus be defined as

$$E = \sum_{z \in Z} \sum_{d \in \{h,v\}} \sum_{r=1}^{r_{max}} \left\| \hat{\mathcal{L}}_{d,z}(r) - \mathcal{L}_{d,z}(r) \right\|. \tag{2.20}$$

Note that Z (the phases, or pixel values, for which chordlength is constrained) can vary; we could use $Z = \{0\}$, $Z = \{1\}$, or even $Z = \{0, 1\}$ (which is known as dual chordlength).

Chapter 3

Frozen-State Hierarchical Annealing

This chapter introduces the key idea of the thesis, one in which large-scale structures become "frozen" at coarse scales, leading to significant benefits in improved modelling and reduced computational complexity.

3.1 Regular Hierarchical Annealing

For reasons of multi-scale modelling and computational efficiency, we wish to work on a discrete-state, top-down hierarchy. That is, if

$$x_{i,j} \in \{0,1\},\ (i,j) \in L \tag{3.1}$$

is some given, two-dimensional binary random field on lattice L, then we construct

$$x^s_{i,j},\ (i,j) \in L^s \ \text{ such that }\ L^1 \equiv L, x^1 \equiv x \tag{3.2}$$

that is, where scale $s = 1$ represents the finest scale. At each scale we will suppose the existence of some energy function E^s, implying a Gibbs prior [14] on x^s:

$$p\left(x^s\right) = \frac{1}{Z} e^{-E^s(x^s)} \tag{3.3}$$

If we define an interpolation operator

$$x^s \longleftarrow \rho^s\left(x^{s+1}\right), \tag{3.4}$$

then a top-down simulation proceeds as shown in Algorithm 5, whereby we begin with a coarse grid, having relatively few elements, and proceed, scale by scale, to finer grids. Related approaches have been proposed in the past [1, 2, 3] but suffer from two particular problems:

Algorithm 5 Basic Top-Down Annealing

```
Randomly initialize $x_0^M$
$s \leftarrow M$
while $s \geq M$ do
    Iteratively anneal $x_1^s, x_2^s, \ldots, x_q^s$ at temperatures $T_1^s, T_2^s, \ldots$
    if $s > 1$ then
        $x_0^{s-1} \leftarrow \rho^{s-1}(x_q^s)$
        $s \leftarrow s - 1$
    end if
end while
```

1. From a computational perspective, the algorithm ends with some number of MCMC passes on x^0, the finest scale. Because the number of elements in x^0 is the same as in x, the computational complexity of the entire hierarchical algorithm is likely to be dominated by the finest scale, and the degree of improvement over a regular, non-hierarchical method will depend on the number of iterations needed for convergence.

2. Secondly, from a modelling perspective, the more troubling question is how to prevent the MCMC model on some scale from undoing the structures created by the models on coarser scales. Such a structure undoing can occur for at least two reasons:

    - **Annealing Schedule:** if the MCMC annealing schedule begins at too warm a temperature, then the sampling at some scale s can, in one pass, randomize the entire domain and undo all coarser-scale results. On the other hand, if the MCMC annealing schedule begins at too cold a temperature, the model at the current scale is prevented from making any changes, leading to misconvergence.
    - **Model Locality:** the purpose of the model E^s at each scale s is to model those phenomena *local to that scale*. Thus the model at one scale may be quite different from that at another. Therefore if the model at scale s is permitted to fully converge, the converged result is likely to be quite different from the converged result at another scale. Therefore not only the annealing schedule, but also the permitted number of iterations, must be tuned at each scale.

3.2 Defining Frozen-State

On the basis of the above criticisms, we introduce a novel frozen, ternary-state hierarchy, which highly effectively deals with both of the computational and modelling issues above. As opposed to the binary state of (3.1), we introduce an intermediate "grey" value

$$x_{i,j}^s \in \{0, g, 1\}, \ (i,j) \in L^s \tag{3.5}$$

The scale-to-scale interpolation operator ρ is unchanged from before, however the top-down simulation, shown in Algorithm 6, differs crucially from the previous approach in Algorithm 5, in that *only those pixels whose parents at scale* $s+1$ *are grey* may be changed or updated by the MCMC sampling at scale s, whereas children of parent elements which are "0" or "1" remain unchanged (frozen). Proposing such a ternary model has a huge impact on computational complexity and modelling:

- *Computationally,* now only a (possibly small) fraction of state elements at a given scale are involved in the MCMC sampling process. In many cases, the fraction of pixels which need sampling at the finest scale is a tiny fraction of the total.
- For *modelling,* because coarser "white" and "black" structures are forcibly preserved, therefore there is no need to tune the annealing schedule or number of iterations, *greatly* simplifying the annealing step at each scale, relative to previous approaches.

An illustration of the ternary hierarchy is shown in Figure 3.1. A given binary, finest-scale image is repeatedly sub-sampled to construct the hierarchy. The subsampling needs to be the inverse of the interpolation operator, preserving black and white regions only where the entire subset is black or white, respectively:

$$x_{i,j}^{s+1} = \begin{cases} 0 & \text{If all children of } x_{i,j}^{s+1} \text{ at scale } s \text{ are ``0''} \\ 1 & \text{If all children of } x_{i,j}^{s+1} \text{ at scale } s \text{ are ``1''} \\ g & \text{Otherwise} \end{cases} \tag{3.6}$$

A more comprehensive illustration is shown in Figure 3.2, which shows the coarsification of two images, one containing relatively large spheres, and the other small spheres. We immediately observe some striking behaviours with corresponding observations:

1. The representation becomes increasingly "grey" at coarser scales, as the size of the coarse scale pixels begins to exceed the size of finest-scale structures.

Algorithm 6 Ternary, Frozen-State Top-Down Annealing

```
Initialize $x_0^{M+1} \leftarrow$ grey
$q \leftarrow 0$
$s \leftarrow M$
while $s > 0$ do
    $x_0^s \leftarrow \rho^s(x_q^{s+1})$
    $G^s \leftarrow$ grey subset of $x_0^s$
    $x_1^s(G^s) \leftarrow$ random initialization
    Iteratively anneal $x_2^s(G^s), x_3^s(G^s), \ldots, x_q^s(G^s)$ at temperatures $T_1^s, T_2^s, \ldots$
    $s \leftarrow s - 1$
end while
```

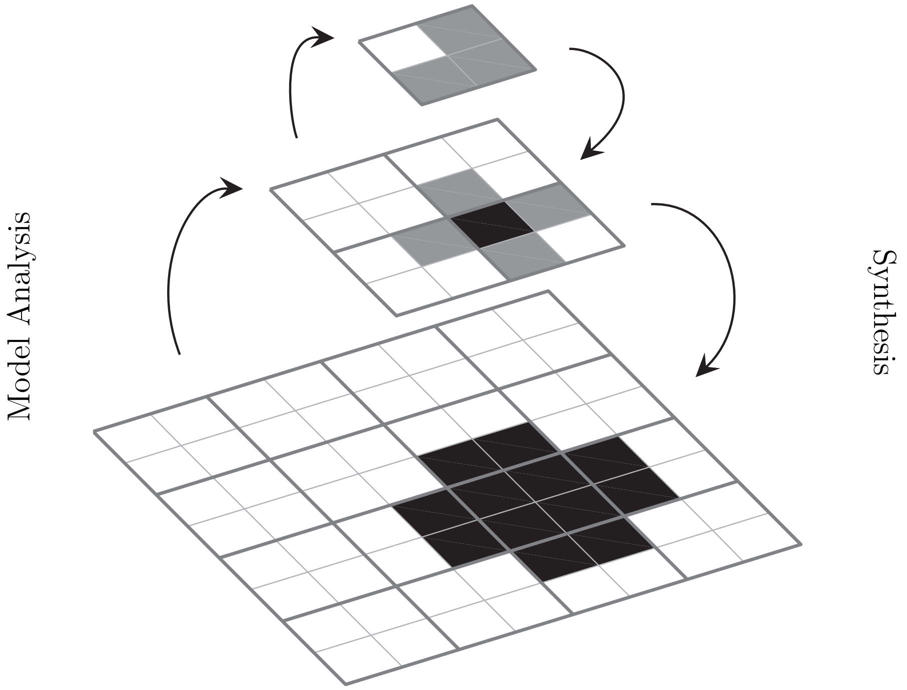

Model Analysis
Synthesis

Figure 3.1: The principle of ternary decomposition: a given binary image can be represented at coarser scales, such that each element on a coarser scale is selected on the basis of whether its descendants are *all* black, *all* white, or mixed.

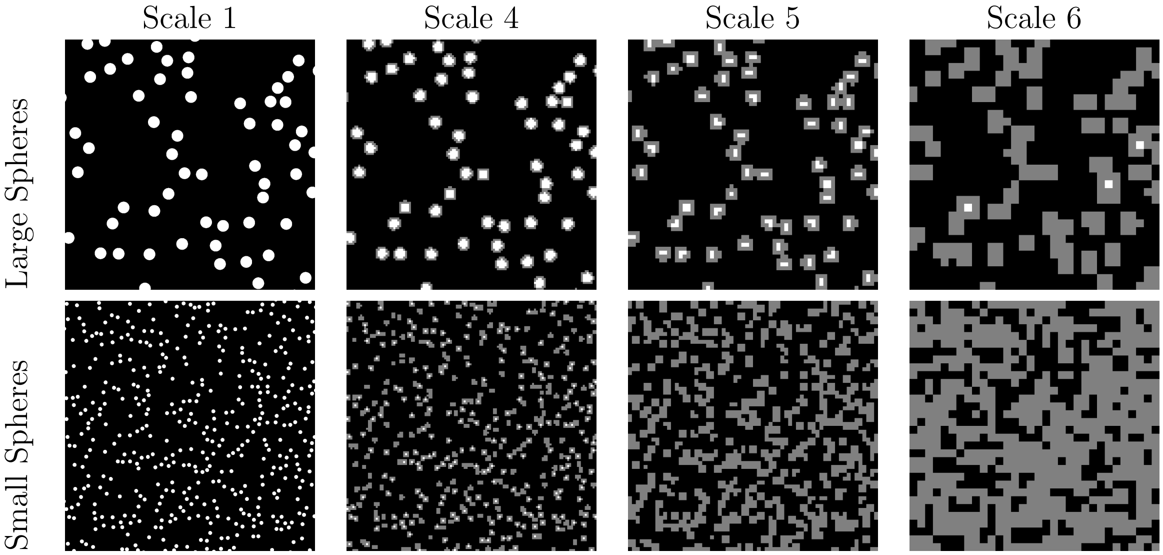

Scale 1
Scale 4
Scale 5
Scale 6
Large Spheres
Small Spheres

Figure 3.2: Two illustrations of downsampling a given binary image per the ternary representation of Figure 3.1. At coarser scales the downsampled image will become increasingly "grey", as the size of individual coarse pixels exceed the characteristic lengths of structures at the finest scale.

⇒ *Therefore the model is easily initialized as all-grey at some sufficiently coarse scale.*

2. The appearance of white seed pixels, forming the centres of finest scale spheres, takes place on a scale dependant on the sphere size. Thus the large spheres appear on scales 5 to 6, whereas small spheres appear on scale 4.

   ⇒ *The model is inherently scale sensitive, and is naturally able to model scale-dependent structures, as evidenced by the very different statistical patterns in the two rows in Figure 3.2.*

3. At fine scales, the grey pixels lie only on the interface between white and black.

   ⇒ *In most cases the interface between white and black will be only a tiny fraction of the total number of pixels. Therefore only a small fraction of the total number of pixels need to be simulated at the fine scales, precisely those scales having large numbers of pixels.*

This chapter has only demonstrated the method in principle and given a brief overview of the gains it offers. Subsequent chapters will examine these aspects in greater detail; specifically, Chapter 4 discusses modelling, Chapter 5 examines the computational gains, and Chapter 6 presents some large scale results.

Chapter 4

Modelling

In modelling binary images, we want to be able to generate random samples that look like they could plausibly be other portions of the same material that was used to train the model. Thus, the model should encode the more abstract properties, such as the general shape, size, and distribution of the structures the images contains, but the model should not over-learn the training image: if the generated samples are too similar to the original, they will be useless to the Monte Carlo simulations for which they are intended.

In general, a huge number of models has been proposed, however, many of these are unsuitable for the particular uses we have in mind: this research is focused on a set of images in porous media (such as those examined in [36, 37, 39, 46]), which exhibit relatively complex fractal-like properties over a variety of scales. Despite this focus on porous media, the work presented here is likely to be of interest to researchers in other fields, as there are a wide variety of other applications with similar properties.

The multi-scale properties of our porous media images suggest the use of a hierarchical model. The literature on scale space theory and wavelet-based models is extensive, and although wavelets are reasonable for real-life imagery, their use is less appropriate in scientific imagery, and especially *discrete* state scientific imagery, as wavelets are inherently continuous.

4.1 Hierarchical Modelling

Because of the wide range in the scales of the features the model must encode, we require either a flat model capable of encoding very large-scale information, or a hierarchy such that these large-scale features can be expressed by a local model at a coarse scale. For a flat model, one can approach the problem directly in the spatial domain, or one could use a change of basis such that non-local interactions become possible. For hierarchical models, one must choose what, exactly, the hierarchy is doing: if a sample is generated as one moves from coarse to finer scales, does the previous scale's result initialize the next scale, or does it constrain it? This thesis is focused on exploring the latter of these four options.

Methods for hierarchically annealing samples of porous media using the hierarchy to initialize finer scales has been the subject of previous research [1, 2, 3]. In this approach, the purpose of the hierarchy is primarily to to bypass the extremely long computation times needed to generate large structures (discussed in more depth in Section 5.2); one can create the larger structures at a coarse scale, then use this to initialize the finer-scale annealing with the structures already in place. At finer scales, however, the annealer is still free to explore the full range of the model. This approach has some limitations:

- The energy model used at each scale must contain enough implicit information about coarser scales to prefer to preserve the coarse-scale structures.
- One must be very careful in determining the annealing temperature schedule to avoid either randomly eroding the coarse-scale structures (by starting a finer scale at too high a temperature) or freezing unwanted attributes of the coarse scale results in place (by starting a finer scale at too low a temperature).
- The number of pixels at the finest scale remains a computational bottleneck: although it is initialized with a state that one might deem nearly converged (and in which the vast majority of the pixels are unlikely to ever change), the annealer continues to visit every pixel.

A constraint-oriented hierarchical model, in contrast, fixes coarse-scale attributes and limits the finer scale annealing to only create those finer scale details. This approach addresses the limitations of initialization-oriented models:

- It can explicitly separate the model into a series of scale-dependent energy models representing *only* those features relevant at the respective scale; consequently, these energy models can be highly local in nature.
- One need not be as concerned about annealing schedules, since the constraint eliminates the possibility of eroding coarser-scale results.
- The computational efficiency in annealing can be improved by simply ignoring those regions that are constrained.

That said, a constraint-oriented model faces its own limitations: Because the coarse-scale results are immutable, any errors or artefacts developed at coarse scales become frozen in place; they cannot be changed to accommodate structures that converge at finer scales. An example of a problem of this nature will be seen in Section 4.4.

Chapter 3 established the principle of a ternary hierarchy, such that at any given scale white and black structures are frozen or fixed in place, with only a grey subset remaining to be determined. These grey pixels may become black or white, and therefore frozen at finer scales, or may remain grey.

What is required is a statistical model which is compatible with describing the behaviour of a subset of grey pixels in a hierarchy. The key insight is that the hierarchical structure allows a great simplification to the model: because coarser scales have modelled and created any structures which were apparent at those larger scales, at some intermediate scale the statistical model is responsible only for *relatively local* phenomena. That is, *non-local* phenomena, which are large relative to the current scale, will already have been modelled in a *local* way on some coarser scale.

A variety of energy models are possible, including well-established models such as correlation or chordlength [36, 37, 46], which were discussed in Section 2.3. However, such non-local models have a cost: they either have a very limited ability to represent finer aspects of the shape and edges of image structures, or they require a high level of computational complexity to encode both detailed local and non-local properties. The scale-separation provided by the frozen-state hierarchy frees us from this trade-off; instead, we can choose a comparatively simple model dependent on *only* local patterns.

4.2 Local Histogram Energy Function

Ignoring Ising/Potts models, which are unable to represent even local patterns or textures, the simplest possible approach, known as a local binary pattern [27, 37] (a generalization of co-occurrence matrices [9]), is essentially a preservation of the joint histogram of a given pixel and its neighbours.

That is, if we define the neighbourhood of $x^s_{i,j}$ to be

$$\mathcal{N}^s_{i,j} \subset L^s, \tag{4.1}$$

then our model needs to store the conditional probabilities

$$\Pr(x^s_{i,j} | x^s_{k,l}, \forall (k,l) \in \mathcal{N}^s_{i,j}). \tag{4.2}$$

The model we have adopted is a modification of the local histogram model discussed in [1, 2]. Based on our tests, it is adequate to define the neighbourhood to be a 3×3 region, containing the eight pixels surrounding a central pixel, from which it follows that the model at each scale s is represented by $3^9 = 19683$ probabilities, which can be plotted as a histogram. An exception takes place at the finest scale, which is forced to be binary, in which case $2^9 = 512$ probabilities are stored.

An illustration of such scale-dependent histograms can be seen in Figure 4.1. The horizontal axis is somewhat arbitrary as it contains, in no significant order, the 19683 possible permutations of nine ternary pixels. After a brief study, however, a very clear structure emerges from the seemingly irregular histograms. At the finest scale (top), the vast majority of the nine pixels are either all black (left) or all white (right). At the fourth scale, most of the domain is still all-black, however significant probabilities can be seen associated with individual white (seed) pixels. As we proceed to coarser scales the white pixels disappear entirely, the likelihood of all-black regions decreases, and grey becomes more prevalent.

Let $\hat{H}^s$ represent the learned histogram model at scale s, and $H(x)$ an operator returning the histogram of x. We wish to develop an energy function, for use in annealing,

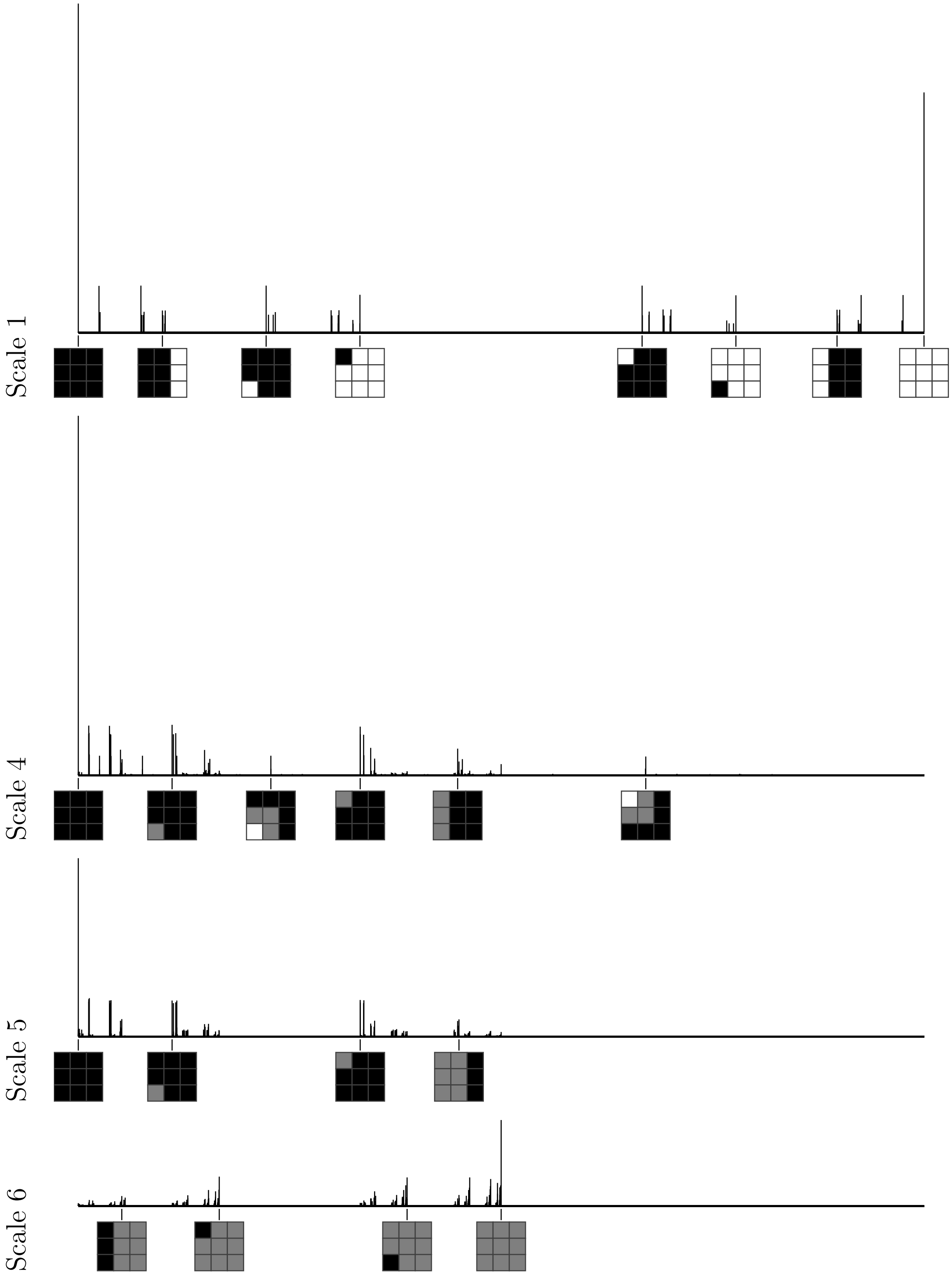


Figure 4.1: The histogram model data for selected scales from the small spheres training image in Figure 3.2 (page 22). The bar height indicates the relative probability of a neighbourhood pattern, and the patterns for certain peak bars are shown below the horizontal axis. Notice how dramatically the histograms change between scales 4 and 6, corresponding with the phase transition in Figure 3.2 from discrete sphere-like structures to predominantly grey as the sphere size drops below the resolution of a single pixel.

which penalizes the difference or inconsistency between a simulated field x^s and its target histogram:

$$E^s(x^s) = \|H(x^s) - \hat{H}^s\| \tag{4.3}$$

We propose a quadratic penalty,

$$E^s(x^s) = \sum_i \frac{(H_i(x^s) - \hat{H}_i^s)^2}{\sqrt{\hat{H}_i} + \epsilon}, \tag{4.4}$$

where the denominator serves to allow for greater flexibility in large histogram entries, and ϵ is a small term to prevent division by zero for entries of zero in $\hat{H}$.

A slight modification to (4.3) is needed, to fully separate the scales. Specifically, those portions of x^s that are frozen (i.e., structures created at scales coarser than s), should not be involved in the considerations at scale s, and therefore those portions of the field should not be included in the calculated histogram.

Therefore, if

$$G^{s+1} = \{i | x_i^{s+1} = g\} \tag{4.5}$$

represents the grey elements of the parent field x^{s+1}, then

$$G_\rho^s = \rho^s(G^{s+1}) \tag{4.6}$$

represents those elements of x^s which are un-frozen, and are eligible for annealing. However because a given pixel contributes to the histogram entries of its eight neighbours, and we seek to include in the histogram all the changeable parameters — and *only* the changeable parameters — for the current scale, we need to additionally include in the histogram any frozen elements that neighbour unfrozen ones. Let us refer to this expanded set containing G_ρ^s plus elements that are neighbour to an element of G_ρ^s as

$$\bar{G}_\rho^s = G_\rho^s \cup \{i | i \in \mathcal{N}_j^s, j \in G_\rho^s\}. \tag{4.7}$$

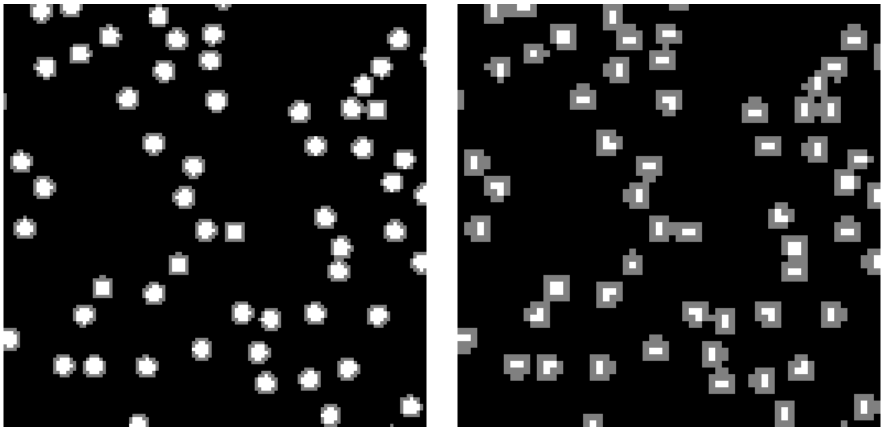
This relationship is illustrated in Figure 4.2. Both the target and observed histograms are computed over $\bar{G}_\rho^s$, and (4.3) is modified to become

$$E^s(x^s) = \left\| H\left(x^s(\bar{G}_\rho^s)\right) - \hat{H}^s \right\| \tag{4.8}$$

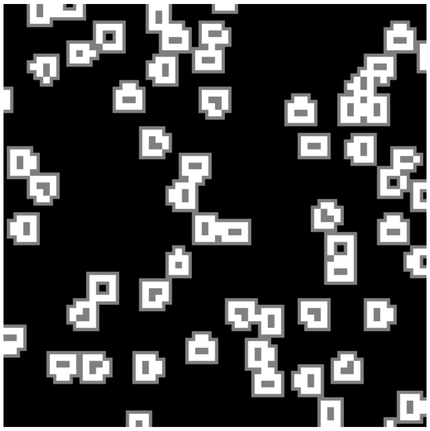
At this point we have an energy function $E^s(x^s)$ as a function of scale s, based on the scale-dependent histogram learned from training images, as in Figure 3.2.

4.3 Methodology

The overall use of the proposed method, in practice, is to build a model from some training image, y, by iteratively coarsening it according to (3.6) until we reach a scale where y^s is entirely undefined ("grey"), since beyond this scale there is nothing further



(a) Mid-Scale Image x^s (b) Parent Scale x^{s+1}



(c) Counted Pixels
$\bar{G}^s_\rho = G^s_\rho$ (white) $\cup \{i | i \in \mathcal{N}^s_j, j \in G^s_\rho\}$ (grey)

Figure 4.2: In both training and annealing, to determine which sites enter into the energy function at a given scale x^s (panel (a)), we must first identify the pixels that have a grey parent in x^{s+1} (panel (b)) — when annealing, these are the pixels that may be changed; when training, these are the pixels that contain new information not present in the parent scale. These locations are denoted G^s_ρ, and are shown as white in panel (c). In the energy function, we wish to include exactly the sites whose local neighbourhood pattern can change. Therefore, the histogram includes the entries for both the sites in G^s_ρ, where the central pixel can change values, and sites that are neighbour to G^s_ρ, where at least one of the neighbour pixels can change values (shown as grey pixels in (c)). This combined set of sites — the non-black pixels of (c) — is denoted $\bar{G}^s_\rho$.

to model. We define this coarsest scale as $M + 1$. For each scale, $s \in 1, 2, \ldots, M$, we need to acquire and store a target histogram for that scale, $\hat{H}^s$, using

$$\hat{H}^s = H\left(y^s(\bar{G}^s_\rho)\right). \tag{4.9}$$

The histogram set $\{\hat{H}^1, \hat{H}^2, \ldots, \hat{H}^M\}$ forms the model input into the random sampler. Using these target histograms in conjunction with (4.8) as the energy function, we can then run Algorithm 6 (see page 21) to produce random samples through simulated annealing.

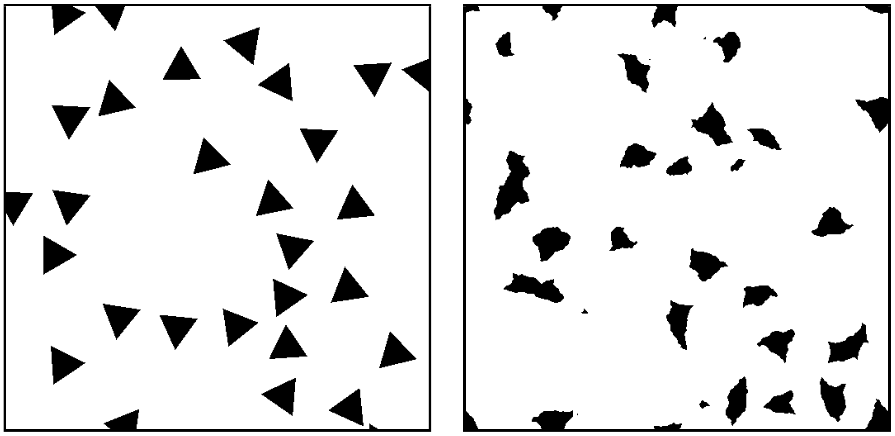
4.4 Evaluation of Modelling

Having established all of the requisite components for modelling and synthesizing images, we can use toy problems to evaluate the capabilities of the method.

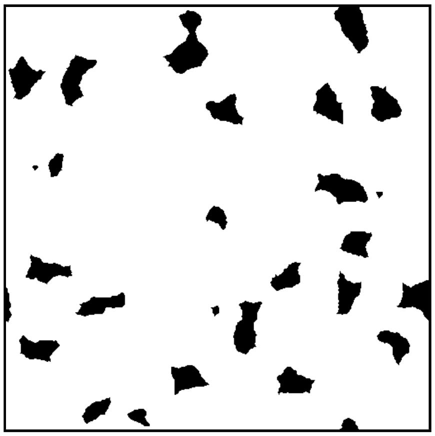
Figure 4.3 contrasts a training image containing only perfect equilateral triangles with an image synthesized from the trained model. The hierarchical nature of the model is able to capture the size and density of the features and, to a certain degree, a pointed ‘triangularity’. The perfectly straight edges, however, are not captured. Such precise straightness is a phenomenon that is evident only at the finest scale within the ternary model, and on a single scale the highly local histogram energy model can only see a 3×3 window. Despite this, the hierarchy manages to encode a degree of straightness to the edges: with each projection to a finer scale, the rules governing frozen-state constrain the edge to remain within a certain window, while the energy model tries to assert what local straightness it sees. The resulting meandering can be viewed as a sort of random walk within that progressively narrowing window. In general, effective ways to model long, narrow features non-parametrically is an unsolved problem: because of the narrowness, they disappear at coarser scales in a hierarchy, while at finer scales, the models are too local to fully capture the length and structure of the feature.

There are a few very small black structures (< 5 pixels in area) also visible in Figure 4.3(b); their existence is a partially a result of the constraint that a pixel which is grey at a certain coarse scale *must* contain a mix of black and white pixels at the finest scale. If, in moving from a coarse to finer scale, the edge is placed close to the core of the triangle, a grey pixel could be ‘orphaned’ from the rest of the structure as it develops. Although the sampler would generally prefer to eliminate that by setting the region entirely white, it cannot because of this constraint. Relaxing the constraint at fine scales, as shown in Figure 4.3(c), reduces this issue, at the expense of losing a top-down consistency of grey states in the annealing result. These ‘orphaned pixel’ artefacts were less prevalent with other data sets, and the results elsewhere in the thesis do not make use of this relaxation.

Figures 4.3(b) and (c) both show a few anomalous structures that are larger than the ‘orphaned pixels’ but distinctly smaller than the triangles of the training image. These structures are similarly orphaned features that became separated from larger structures at a coarser scale. Their existence points again to the difficult of modelling long, narrow



(a) Training Image (b) Result



(c) Result with grey constraint relaxed at scales 1–3

Figure 4.3: This toy problem demonstrates several of the strengths and limitations of the frozen state method when used with the local histogram energy function. Notice how the model is able to do an excellent job recreating the density and size of the triangle features, including, to a degree, their triangular natures. However, the highly-local histogram function is unable to see the long, straight edges — a property that is present only at the finest scales, and thus also invisible to the local models on the hierarchy. Relaxing the constraint that the children of a grey pixel *cannot* be all black or all white eliminates some small artifacts present in the standard frozen-state result.

features hierarchically; during the annealing, the ancestors to these structures became separated from larger structures at a scale coarse enough that the annealer was still evolving many triangle features completely in grey (without a black pixel to anchor to), but fine enough that these grey regions are larger than any single neighbourhood. Consequently, the energy model at that scale is unable to "see" the cases where the created grey structures aren't connected to something larger.

In general, because large-scale and small-scale structures emerge at different scales, and with a *different* model learned at each scale, we do *not* require a single model to capture all structures simultaneously.

Indeed, because the ternary hierarchy freezes large-scale structures determined by coarser scales, the model at a given scale is responsible *only* for the *new* structures at the given scale — those previously unresolved at coarser scales. Therefore the model at a scale is in no way responsible to preserve larger-scale structures, as they are frozen, nor does it even need to be consistent with the coarser models. This enables the use of arbitrarily high starting temperatures are each scale. The model at each scale is iterated to full convergence. Because each scale is fully converged, the temperature schedule in no way needs to be finely tuned.

Furthermore, in the ternary representation it is only the descendants of grey pixels from the parent scale which need modelling, as all white and black regions have been committed and frozen. As suggested by the images in Figure 3.2, we frequently observe large regions of grey having a simple structure (coarse scales), *or* small grey regions with complex structure, but only rarely do we have complex, non-local behaviour for grey. Therefore at every scale it is a comparatively simple, *local* model that is needed.

Such model locality in hierarchical representations has been observed in the past. In particular, a past approach to hierarchical sampling [3] asserted a chordlength model [36, 37], in which the model tests the distribution of the lengths of black and white chords. The chordlength model was truncated to test only chords up to a length of n pixels, and the reconstructed samples were found to be very nearly independent of n, for $n > 10$ pixels. In other words, only the local portions of the model were significant and instrumental in sampling.

This near-independence, in turn, suggests that there is a highly complementary relationship between hierarchical models and simple, local energy functions. Figure 4.4 shows the probability of various pixel configurations (black, white, black next to grey, white next to grey). As opposed to a single-scale model, which needs to *explicitly* account for structure size by encoding the size of an object into the model, in a hierarchical model the size of an object is represented *implicitly* on the basis of the scale at which it appears. In Figure 4.4 we see the emergence and disappearance of of various configurations, such that the probabilities for large spheres lie one to two scales coarser than those for small spheres.

Although such an implicit representation may not seem impressive, imagine constructing a statistical model allowing a mixture of large and small spheres. A single-scale model becomes rather complex, possibly involving hidden layers or additional labelling, whereas hierarchically the sum of large and small sphere models leads to the creation of

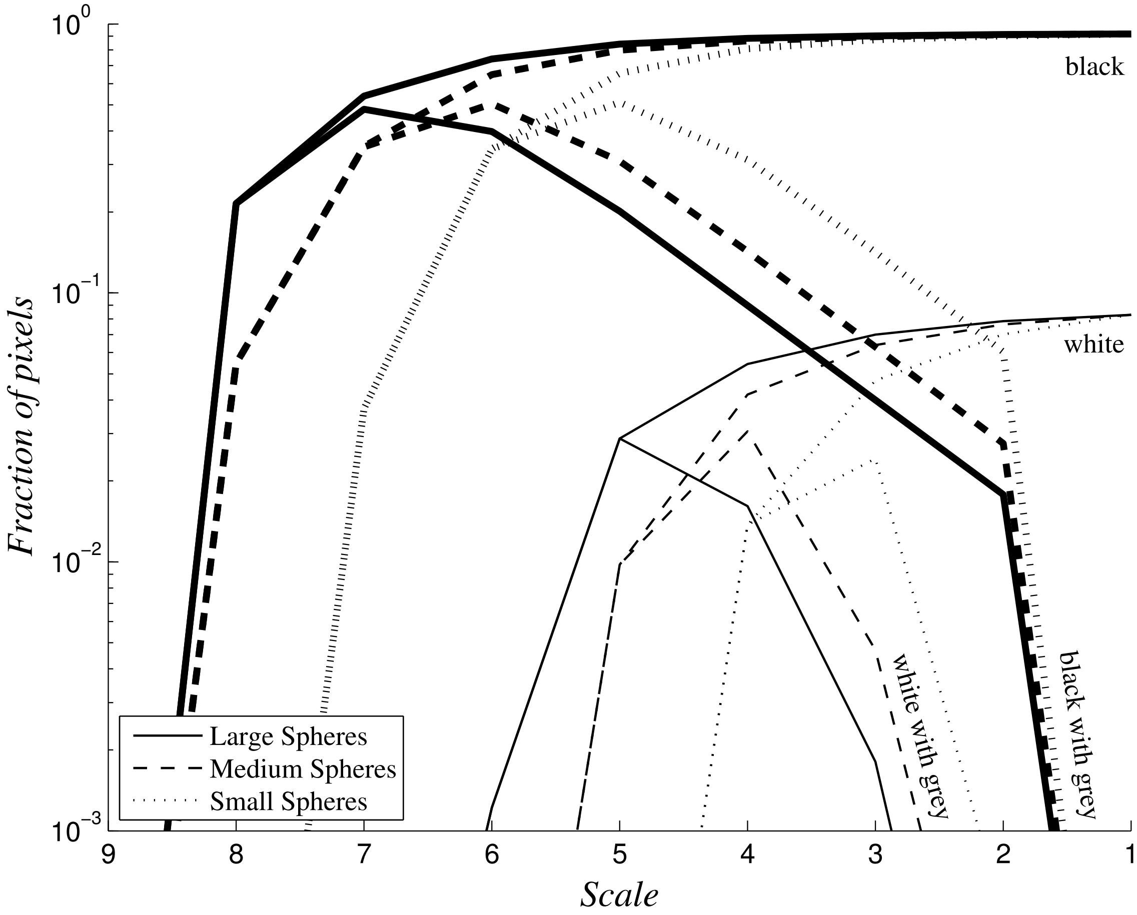


Figure 4.4: Scale dependance of different models: Consider the three examples of Figure 3.2, with small, medium, and large spheres respectively. For each image, the fraction of black, white, black next to grey, and white next to grey pixels are plotted. Since all three images have the same overall porosity, the fractions of blank and white pixels converge to the same values at finer scales. However the real power of scale-dependent modelling is revealed at the intermediate scales, where the appearance of black or white pixels, and the later disappearance of grey pixels, takes places at scales which are clearly dependent on the underlying image (at coarser scales for larger structures, and at finer scales for smaller ones).

seed pixels at both large and small scales, thus leading to the interesting, heterogeneous result shown in Figure 4.5. Such a hybrid model can be formed, in general, as long as the two structures being mixed are scale-resolved, since their behaviour is actually asserted by separate models on separated scales.

The strength of the ternary approach, on the basis of modelling, has been argued in this chapter and is illustrated in Figure 4.5. For fractal-like morphologies, such as those encountered in porous media, a scale-dependent model is essential; however, single-scale annealing has even greater computational limitations, to be examined in the following chapter.

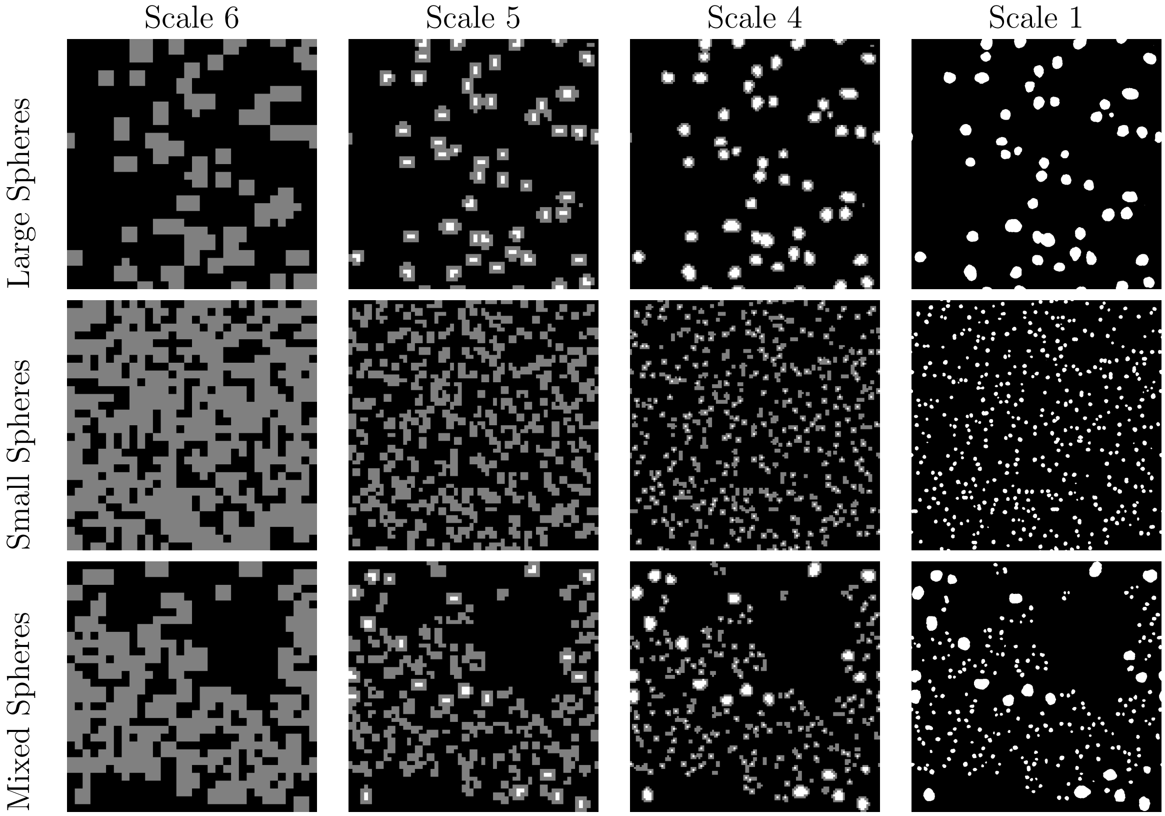

Scale 6
Scale 5
Scale 4
Scale 1
Large Spheres
Small Spheres
Mixed Spheres

Figure 4.5: Ternary hierarchical sampling results, with models learned from the large and small spheres images of Figure 3.2. The rather complex image of mixed spheres in the bottom row has a trivial model, nothing more than the average of the models from the top two rows. The reader can observe large spheres being seeded in scale 5 in the top and bottom rows, and the small spheres seeded in scale 4 in the bottom two rows. Such a heterogeneous, hybrid model is a huge advantage of the hierarchical representation.

Chapter 5

Computation

For the modelling of complex, multiscale porous phenomena, a major issue with existing sampling methods has been the huge computational time needed to produce satisfactory results [1]. The images produced should be fine enough to resolve the smallest-scale features yet also big enough to observe the largest, which means they must have very high pixel resolutions — requiring unreasonably long computation times to produce even remotely meaningful results [1]. Ideally, researchers on porous materials would like to be able to generate full three-dimensional volumes, but the curse of dimensionality has made this almost inconceivable given current limitations. This chapter explores why previous methods have had such high computational complexity, and discusses how the frozen-state method is largely able to avoid them.

5.1 Annealing Cooling Schedules

As discussed in Section 2.1.4, the idea underlying the use of simulated annealing for sampling is that we wish to randomly select as our sample one state of the random field from the set of states that exist at the energy function's global minimum. It has been proven that a logarithmic cooling schedule would, in principle, guarantee that a global minimum is reached, however the extreme number of iterations required by such schedules make their use infeasible [14, 26]. In practice, exponential cooling schedules of the form

$$T(t) = T_0 \alpha^t \tag{5.1}$$

are often used [14, 20, 26], where the initial temperature T_0 and cooling decay factor α can each be determined either in advance or by using a heuristic. Similarly, the stopping criterion can be based on some fixed value — such as a maximum number of iterations, or reaching a target energy state — or something more complex, like a heuristic to detect whether the annealing has reached a (practically) converged state. The goal, in any case, is to start the simulation at a high enough temperature to explore the full solution space (i.e., $T_0 \gg \Delta E$ such that practically all moves are accepted), then cool for long enough, and at a slow enough rate, such that the method will reliably produce a satisfactory result.

While higher starting temperatures and slow cooling rates do increase the likelihood of a successful convergence, they do so at the cost of requiring more iterations to reach the same final temperature; there is no easy solution to this tradeoff.

For all of the results presented in this thesis, an exponential cooling schedule was used with decay factors ranging between 0.99975 at the coarsest scale and 0.992 at the finest. At each scale, the annealing was started at a super-critical temperature and continued until full convergence, which was defined very conservatively as 25 iterations with no state changes. The formal convergence properties of the histogram energy function are unknown.

5.2 Annealing and Computational Complexity

In general, the computational complexity of pixel-based annealing methods is the product of three factors:

1. ϕ, the cost of computing the change in energy for each proposed change in pixel values,
2. N, the number of pixels being iterated over (typically the full size of the sample to be generated), and
3. I, the number of iterations (typically determined by the cooling schedule).

We can express this in scale-dependent fashion as

$$Complexity \propto \sum_{s} \phi \cdot N_s \cdot I_s \tag{5.2}$$

Traditional uses of annealing methods [31, 36, 37, 46] have tended to regard the first two factors, ϕ and N, as relatively fixed: the first is determined by the choice of energy model, while the second is determined by the image size, leaving the third factor — the number of iterations — as the only area where significant reductions are feasible. Consequently, much of the literature on simulated annealing methods has focused on ways to accelerate the computations by reducing the number of iterations needed to achieve satisfactory results [1, 26, 35].

A key reason why existing methods have been unusably slow relates to the number of iterations required to produce large structures (as measured in pixels). In an iterative local model, the evolution of a structure requires that the pixels that compose it achieve a coherent state; state information must be able propagate across the structure. The length scales involved are often much larger than the neighbourhood size, so such propagation is not instantaneous. Conditional independence between two pixels, e.g., non-neighbouring elements in a Markov random field, does *not* imply that the pixels are uncorrelated with each other; generally there are significant correlations over regions much larger than the neighbourhood size, and it is these correlations that must be allowed to propagate. Consequently, the communication between two points depends on the sequence of stochastic state changes across the chains of neighbouring pixels that connect the two points. This implies that the information flow is a form of random walk; it has been shown that to communicate across a distance d pixels (and thus to create structures of that size) requires roughly $O(d^2)$ iterations as a minimum [40].

This computational complexity is especially problematic in sampling images which contain structures across a broad range of length scales: simulating at a scale fine enough to resolve the small structures means that the largest structures (and the images themselves) require a huge number of pixels. If we are simulating an $n \times n$ image and $d \sim n$ (that is, the largest structures are roughly the same size as the image), then we require at least $I = O(n^2)$ iterations, and each of those iterations involves $N = n^2$ pixel-samplings, meaning that the whole process would require $O(\phi \cdot n^4)$ computations. In the case of porous media, such as those shown in Figure 1.1 (page 2), such structures can be thousands of pixels wide, requiring millions of iterations, and trillions of computations.

Even when one allows for the necessary number of iterations, the creation of large structures is subject to energetic feasibility constraints [10, 40]. Suppose one has an energy function that allows for disjoint discs of small and large sizes, but disallows medium-sized ones. If one starts annealing at a state that has only small discs, one will never get large ones: because normal, single-scale annealing makes only incremental changes, it must necessarily pass through intermediate stages containing of medium-sized discs before it can arrive at large ones. However, the disallowance of medium-sized discs creates an “energy barrier”; it is extremely unlikely that the sampler will perform the sequence of pixel state changes needed to grow or combine the small discs into large ones when those necessary intermediate states are not tolerated by the model. With finer resolutions (or larger structures), this likelihood plummets further, as more and more pixels must be unfavourably (and improbably) changed to bridge the gulf separating the permitted sizes. This sort of scale disparity is present in many of the natural phenomena we seek to model. One could conceivably ‘cheat’ by altering a model to add such intermediate states, but doing so would not be simple, and would undermine our goal of drawing proper random samples from the original model of the observed phenomenon.

Hierarchical methods can avoid both of these problems: if we make the pixel size proportional to size of structure we seek to induce, then at that scale those structures will be local and can be induced by just a few pixel-value changes. By proceeding through the hierarchy from coarse to fine scales, one can give this opportunity to structures of all sizes.

5.3 Computational Complexity of Frozen-State Method

As the frozen-state method uses simulated annealing, its computational complexity is also governed by the same basic factors explored in Section 5.2; however, several of its properties offer substantial advantages over previous hierarchical and flat annealing. In many ways, these differences parallel the properties affecting its modelling characteristics, as discussed in Chapter 4. Recall that the following properties exist when sampling using the frozen-state hierarchy:

1. At each scale, it is only *local* structures which need to be synthesized;
2. Structures created at coarse scales are frozen and cannot be eroded at finer scales;
3. In the ternary representation, only the descendants of grey pixels from the parent scale need sampling, since all white and black regions have been committed and frozen.

These properties, in turn, lead to some very compelling computational advantages:

1. The ternary values lend themselves directly to memory-efficient data structures;
2. A highly local energy model is sufficient;
3. State information does not need to propagate long distances;
4. The method does not need carefully tuned annealing cooling schedules;
5. The number of pixels to sample at each scale is modest.

We will explore each of these advantages.

5.3.1 Memory-Efficient Data Structures

Although not a focus of the research presented in this thesis, it is worth noting that the frozen-state hierarchy can be viewed as a tree-like structure in which grey pixels are non-leaf nodes, and white and black pixels are leaf nodes at the scale at which the became frozen. From this perspective, the hierarchy is very similar to a region quadtree representation [32]. An implementation could use this compacted form of representation to simulate extremely large sparse fields.

5.3.2 Highly Local Energy Model

Within the frozen-state hierarchy, only relatively local structures ever need to be synthesized. Any features that are larger than (i.e., non-local to) the current scale have already been created and *frozen* before the current scale is simulated; recall from Figure 4.2 how nonlocal structures are, in fact, entirely *removed* from the energy model for finer scales. Conversely, local grey pixels act as a placeholder to indicate areas where features finer than the current scale may be present — the method is not concerned with the details of those finer features, but rather with where they ought to eventually be, relative to the current scale. This locality across the entire hierarchy enables the use of highly local energy models, which are generally simpler to compute; thus the per-pixel computational cost, ϕ, is reduced compared to models that must incorporate statistics calculated over larger neighbourhoods.

With the histogram energy model, for instance, computing the energy change for a proposed single-pixel state change is a function only of the neighbourhood size: for a neighbourhood containing q pixels, the single-pixel state change will affect, at most, $2(q+1)$ entries in the histogram, and the change in energy can be computed using just those entries.

5.3.3 Propagation of State Information

Since larger-scale features are already fixed in place, we are more insulated from the concern that there must be a large number of iterations (i.e., a slow annealing cooling schedule) to allow for changes in one location to propagate.

Within a frozen-state hierarchy, the maximum range across which state information *could* need to propagate is explicitly bounded by the size of contiguous areas being simulated, i.e., the dimensions of grey areas from the preceding scale; outside of those areas the state is fixed and they can be no state changes. Notice in the images of Figures 3.2 and 4.5 how limited these grey zones are: often, they are only a handful of pixels wide at all but the coarsest scales. Consequently, we do not experience a need for $I_s = O(d^2)$ iterations; instead, a much smaller (and relatively constant) number of iterations at each scale will suffice. Simulating only *local* structures leads to a huge reduction in the required iteration count, and therefore relatively quick annealing schedules can be used.

5.3.4 Tuning of Annealing Cooling Schedules

Because large-scale structures from coarser scales are immutable, the current scale is *unable* to destroy those structures, therefore there is no problem in starting with a hot annealing temperature. The implementation of this method does not, therefore, rely on many repeated runs to tune annealing parameters. Each scale can be annealed independently from some supercritical temperature until convergence. With this flexibility, one can use a wider variety of established heuristics to estimate these temperatures and adaptive methods for accelerating the schedule.

Figure 5.1 illustrates the type of robust and relatively fast annealing schedules that can be used, by plotting the temperature alongside the corresponding changes in energy over time. The upper panel of Figure 5.1 uses iterations (specifically, the number of annealing iterations over the image) as its horizontal axis, while the lower panel shows the same data an against an axis that has been normalized for the number of computations (i.e., the product of iterations with the number of pixels visited in each iteration). In Figure 5.1's energy plots, at every scale, the start and end segments have roughly horizontal slopes, which demonstrates how annealing can be started at a supercritical temperature and continued until convergence for *each* scale. At coarse scales, the energy is more volatile; the small number of pixels means the value of each individual pixel makes a much more significant contribution to the energy function. Because the frozen-state hierarchy effectively makes the results at coarse scales fixed for all finer scales, relatively slow schedules are used at coarse scales to ensure optimal convergence. And yet, despite these slow schedules, the number of computations used at the coarsest scales is barely visible when shown relative to the finer scales, because of the extremely small number of pixels. Notice how, in contrast, relatively few iterations are required at fine scales: a benefit that follows directly from the hierarchical nature of our modelling approach. Slower fine-scale schedules were experimented with, but it was found that the faster cooling shown here had no effect on the final energies reached. This freedom to use accelerated cooling schedules at fine scales (which one can infer from the increasingly steep slopes of the temperature lines in the upper panel of Figure 5.1) helps to lessen the effect that the growing size of the image has on the number of computations required, in a manner that is independent of (but in addition to) the frozen-state method's reductions to the number of sampled pixels.

5.3.5 Reduced Number of Sampled Pixels

The fourth and most significant computational advantage is how the frozen-state method reduces the number of pixels that need to be sampled. The computational complexity of simulated annealing sampling techniques is always a function of N, the number of elements that need to be simulated — that is, the number of pixels in the image being generated. As discussed in Section 3.1, existing hierarchical annealing techniques still experience this limitation: the hierarchy is used to more quickly *initialize* the state at the finest scale (reducing the number of iterations there) but the whole image is still simulated. Supposing the finest-scale image is $n \times n$ pixels in size, then the number of pixels being iterated over at each scale is therefore

$$N_s = \frac{n^2}{4^{s-1}}. \tag{5.3}$$

(Alternately, we could say that each finer scale simulates quadruple the number from the parent scale.) Consequently, each finer scale possesses more elements to simulate than *all previous scales combined*, and the overall computational complexity for these hierarchical techniques is still dominated by the simulation of the finest scale.

Under the ternary representation of the frozen-state method, it is only the descendants of grey pixels from the parent scale which need sampling; all white and black regions have been committed and frozen. Consequently, while the image size at scale s is still $\frac{n^2}{4^{s-1}}$, the number of pixels simulated is some fraction β_s of that,

$$N_s = \beta_s \frac{n^2}{4^{s-1}}, \tag{5.4}$$

and β_s decreases monotonically with decreasing s — going down the hierarchy, as the total number of pixels increases, the *proportion* of pixels that must be sampled *decreases*

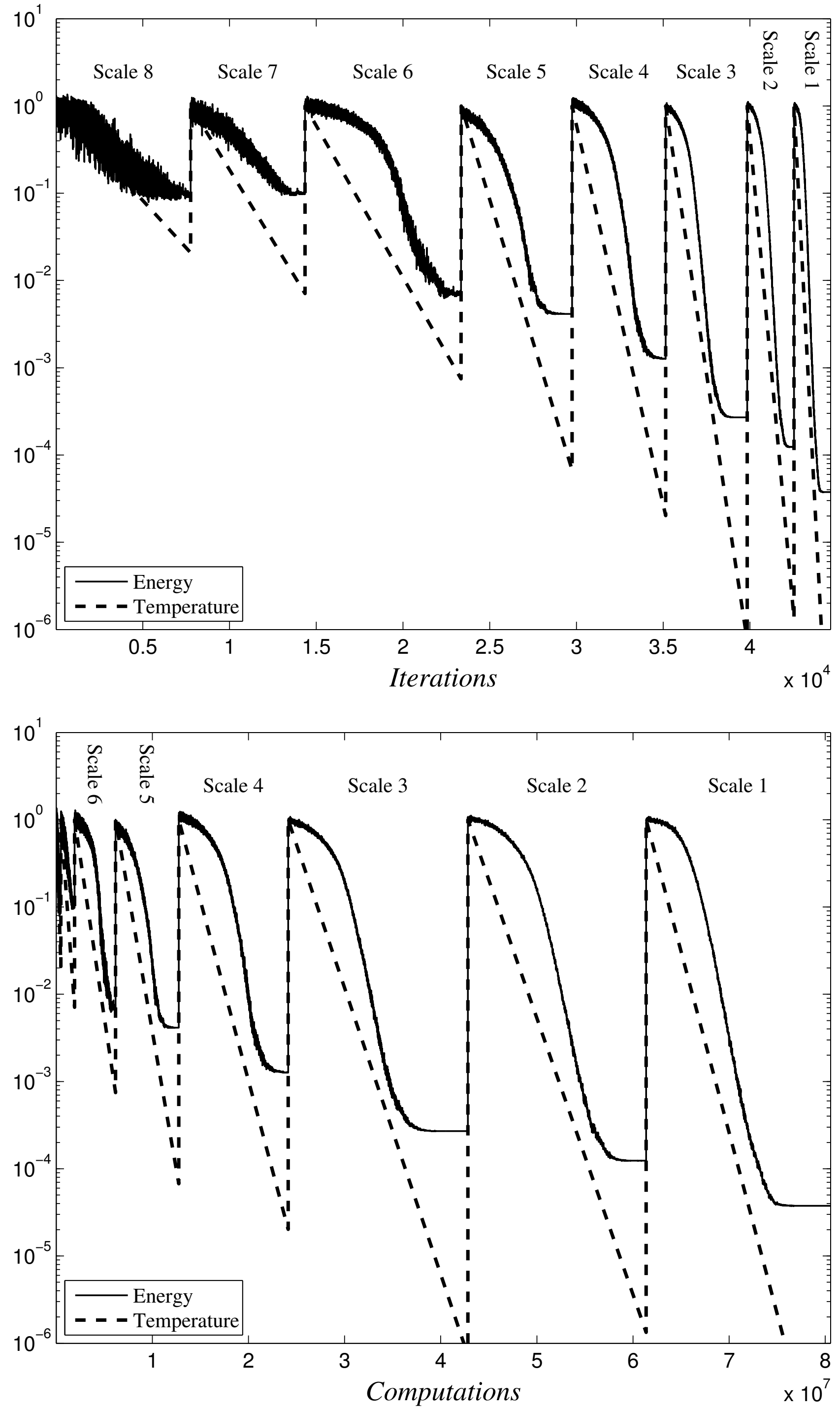
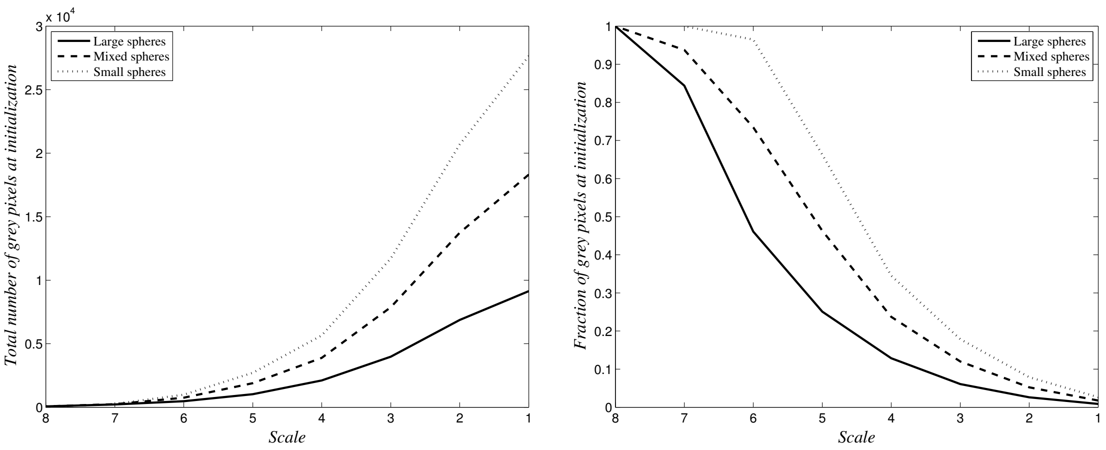


Figure 5.1: The energy behaviour for the "large spheres" result of Figure 4.5. The starting values of energy and temperature at each scale have been normalized. As scale proceeds from coarse to fine, the computational cost of each annealing iteration across the image increases while the importance an individual pixel has on the overall result decreases. Thus, to balance computation time where it is most needed, the cooling rate is increased for finer scales.

monotonically. Notice how in the images of Figures 3.2 and 4.5, the grey pixels fill a declining fraction of the random field as we move to finer scales. At coarse scales, grey dominates and $\beta_s \approx 1$, but the total number of pixels (and thus the cost per iteration) is very modest there, and thus this poses little concern. At the finest scales, the number of annealed pixels is reduced to a tiny fraction of the total; β_1 is typically less than 0.05. Indeed, the behaviour of β_s for all three "spheres" test cases can be seen in Figure 5.2(b), and the fraction of pixels to sample reduces to 1% to 5% at the finest scale. Therefore at the finest scale we actually simulate only a *tiny* fraction of the pixels, a major reduction in complexity from non-hierarchical and other non-frozen standard hierarchical approaches, which still simulate the entire finest scale (i.e., $N_1 = n^2$). The bottom panel of Figure 5.1 shows the modest fraction of the computational complexity spent at the finest scale, despite the fact that the finest scale possesses more pixels than all other scales combined. In practice, we have observed that as the simulation moves from coarse to fine scales, the number of annealed pixels N_s tends to converge on a linear growth rate — observe in Figure 5.2(a) the relatively constant slopes from scales 4 to 1 — which is a drastic reduction from the 4^{-s} exponential growth rate for the overall number of pixels.

The overall reduction in complexity is the product of the reduction in iterations and the reduction in the fraction of pixels to be sampled. The consequence of the above improvements is enormous, relative to standard single-scale annealing. Problems which take days or weeks to converge with standard annealing, or which require carefully tuned sampling patterns and energy functions to make convergence possible at all, converge in minutes to hours with our proposed approach.



(a) Number of pixels to simulate (b) Fraction of pixels to simulate

Figure 5.2: The number and fraction of pixels to simulate as a function of scale, for the small/large/mixed spheres results of Fig. 4.5. The fraction of pixels being simulated decreases to a tiny proportion of the whole as the scale becomes finer. Although the overall fraction of white pixels is the same in all three, they are more condensed for the large spheres — fewer pixels lie on the boundaries between black and white — and consequently its fraction decreases more quickly. The curves for the mixed spheres result is, unsurprisingly, consistently halfway between the other two.

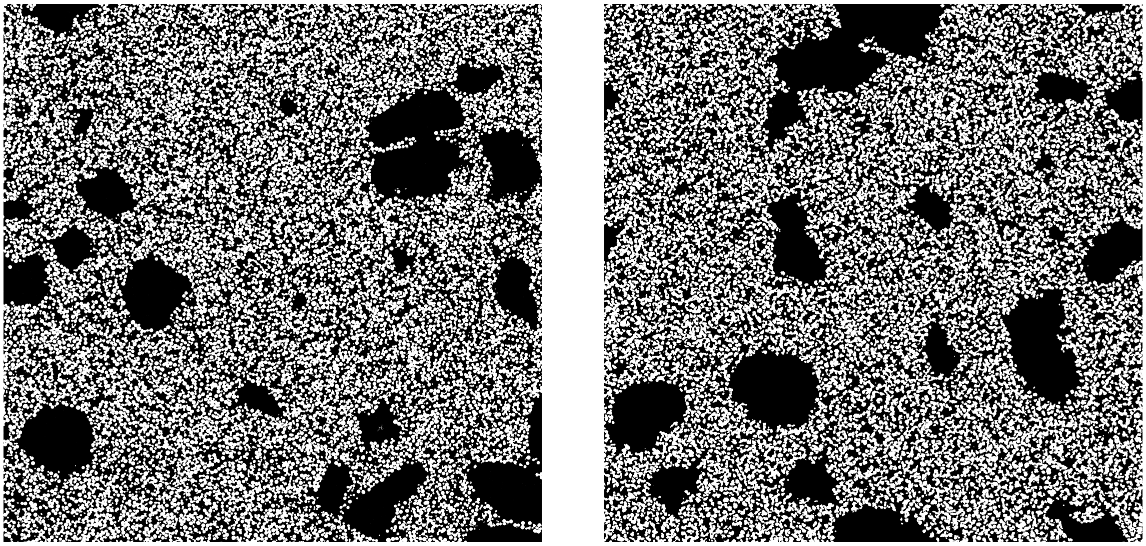
Chapter 6

Results and Discussion

We have simulated 64-million pixel (8192×8192) images on standard PCs in two hours, opening the possibility for far larger images, or large three-dimensional simulations on larger workstations. Figures 6.1 and 6.2 compare two such 64-million pixel samples with their training images. The similarity between the sampled and original microscopic images is stunning, preserving much of the structure on multiple scales. Comparing the two-point correlation and chordlength distributions of the samples with the originals (Figs. 6.1(c,d) and 6.2(c,d)) offers an objective measure for the quality of result, as the models do not explicitly contain either distribution. Figure 6.3 shows how these two samples compare with each other in the number of pixels to simulate — and, thus, the overall computational effort needed — as a function of scale. Because the sintered glass beads have a much greater pore-solid interface than the carbonate rock, the computational burden is somewhat greater in the glass-bead case. In general, the time required to generate a sample is dominated by the number of pixels to simulate at the finest scale.

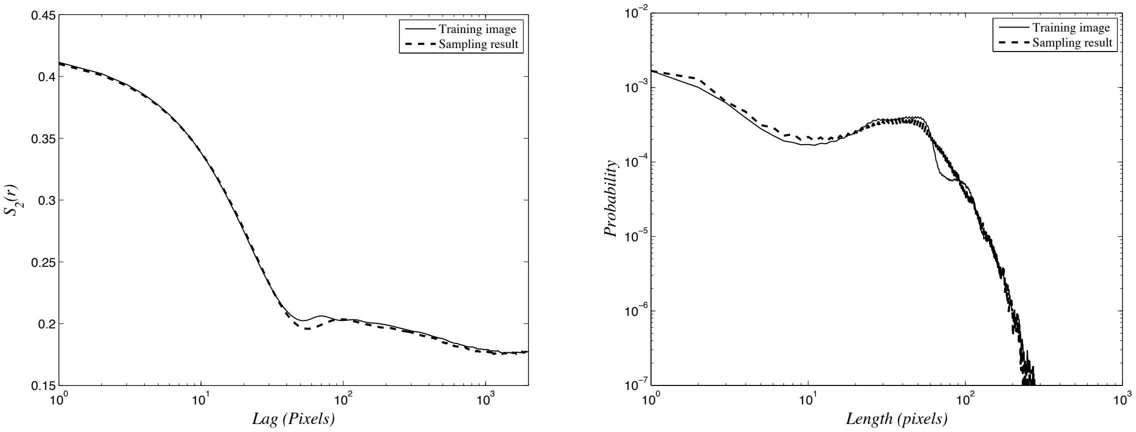
Two other examples give explicit comparisons with past methods. Figure 6.4 compares our proposed frozen-hierarchical approach with regular flat annealing and with a previously-proposed hierarchical approach [1]. Even after *days* of computing time, the flat annealers are, for all intents and purposes, unable to converge. Past hierarchical approaches are far more convincing in their convergence, however their sensitivity to annealing schedule means that fine scales have the ability to destroy coarse-scale structure, and therefore there is a constant challenge in parameter tuning, leading to pixellation and noise effects. The flat annealing method's failure to converge is directly related to the discussion of computational complexity and annealing in Section 5.2, and is plainly due to the large size of the image coupled with a requirement for $O(d^2)$ iterations. The flat annealing could potentially *never* converge on the desired structures, regardless of the number of iterations, due to large size of the desired structures and the energy barrier created by the relative lack of small- and intermediate-sized structures in the training image.

Next, to give a more comprehensive comparison to the results of [1], Figure 6.5 gives a zoom-in comparison of the performance of the hierarchical methods, relative to the original 8192×8192 sample. The strength of hierarchical modelling is clear here, in



(a) Original (training) image

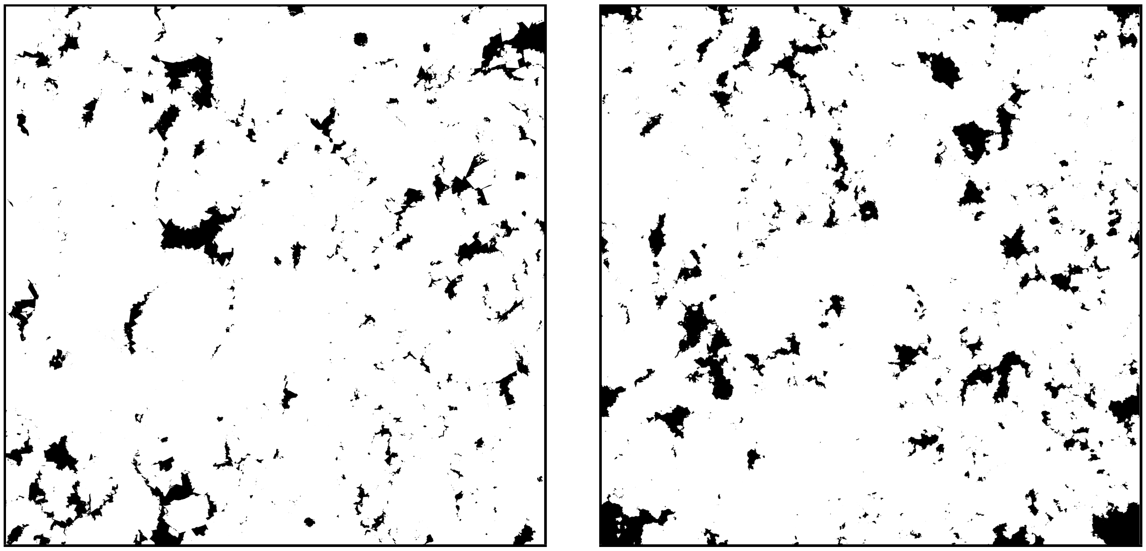
(b) Sampling result (8192 × 8192, 14 hours)



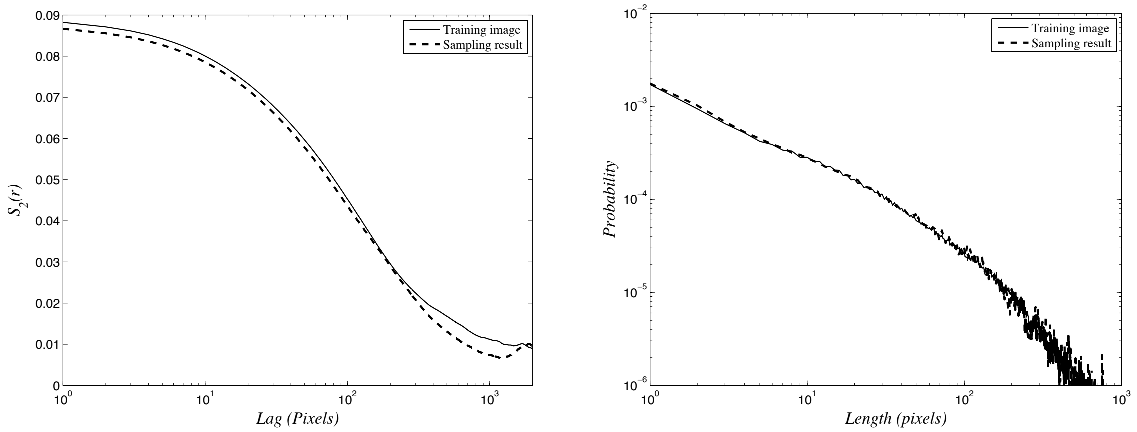
(c) Two-point correlation distribution (white-to-white pixels)

(d) Chordlength distribution (white pixels)

Figure 6.1: Comparison of a full-size sampling result with the original image for sintered glass beads. The morphology of the simulated image captures behaviour over an incredibly wide range of scales, as confirmed by both the two-point correlation and chordlength distributions. The only notable deviation occurs at a chordlength of roughly 70 pixels, which corresponds to the maximum diameter of the glass beads: a hard limit which is observable only at the finest scale with a wide spatial view — precisely the type of property that a hierarchical model composed of local, scale-dependent observations is unable to capture.

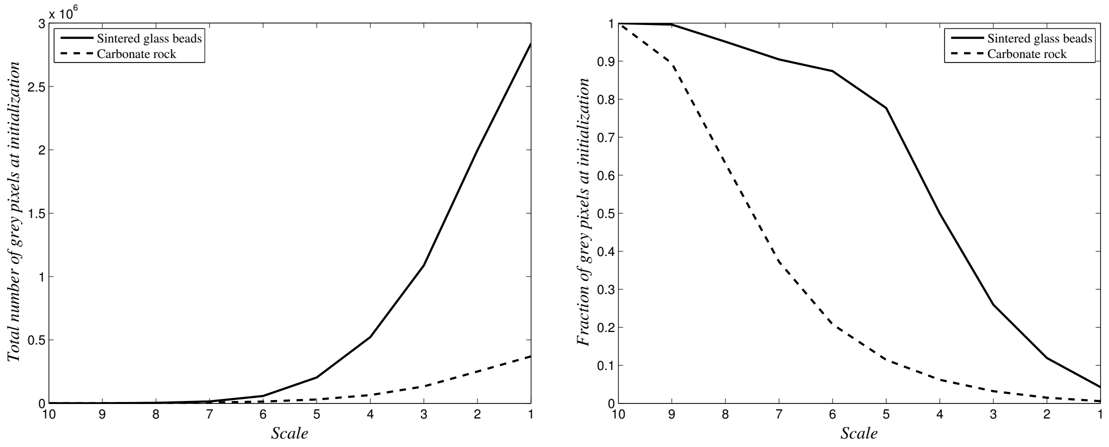


(a) Original (training) image (b) Sampling result (8192 × 8192, 2 hours)



(c) Two-point correlation distribution (black-to-black pixels) (d) Chordlength distribution (black pixels)

Figure 6.2: Comparison of a full-size sampling result with the original carbonate rock image. The sample is nearly indistinguishable from the original unless examined at full resolution. The sample and original two-point correlation and chordlength distributions match well within the expected degree of variability for other physical samples of this medium.



(a) Number of pixels to simulate (b) Fraction of pixels to simulate

Figure 6.3: Comparison of the number and fraction of pixels to simulate for the results in Figs. 6.1 and 6.2. The greater sparsity of the carbonate rock leads to a much sharper drop-off in the fraction of pixels to simulate as the scale becomes finer.

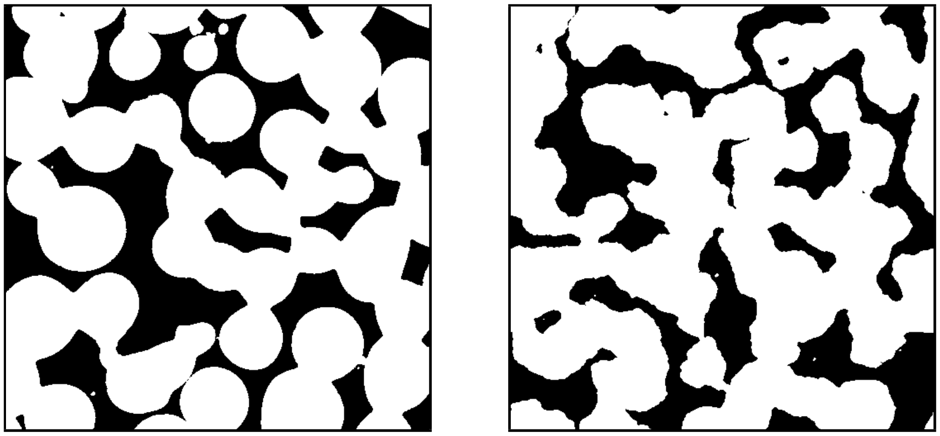
that the details of fine-scale boundaries require a completely different model from the large-scale morphology, and the hierarchical approach offers such scale-dependent models. Careful comparison of the original and the result reveal that, at the finest scale, the piecewise-flat faceting visible in the original image is not precisely captured, for the same reasons as were discussed in Section 4.4 regarding the triangles in Figure 4.3. However, the "islanding" effect — the tiny fragments of pores that appear near the edges of the larger pore structures — is recreated quite well. The high-level morphology is definitely accurately reproduced, thanks to scale-dependent modelling.

Figure 6.6 shows the potential of the proposed model to be used in more complex, nonstationary, multi-model settings. Rather than averaging two models, as was done in Figure 4.5, here we create two complementary models, and use a predetermined hidden layer to partition the physical space of the image as being represented by one model or the other. The hidden layer is simply a binary value indicating whether the corresponding pixel is to be evaluated against the white-on-black model histogram, or the black-on-white, and there is a sinusoidal pattern running down the centre of the image separating the former from the latter. Thus, the two halves could almost be viewed as separately annealed images, except for three key aspects:

1. They are always able to "see" into each other along the boundary between them. Specifically, along the boundary, pixels from one domain enter into the neighbourhoods, and thus into the histogram energy function, of the other domain. Because of the highly local nature of the histogram energy function, this overlap is limited to roughly one pixel in width.

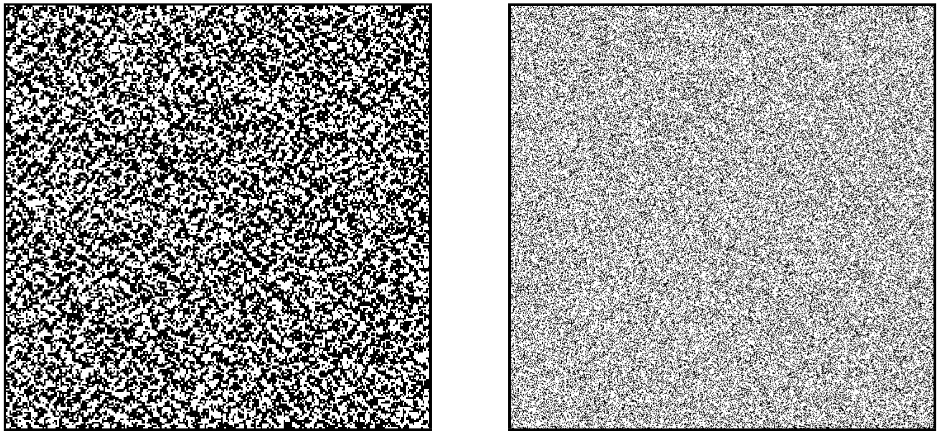
2. The two halves are annealed concurrently, so changes in one along the boundary are sensed by, and able to co-evolve with, the annealing of the other.

3. Both of the preceding aspects are true at *all scales* in the hierarchical annealing.



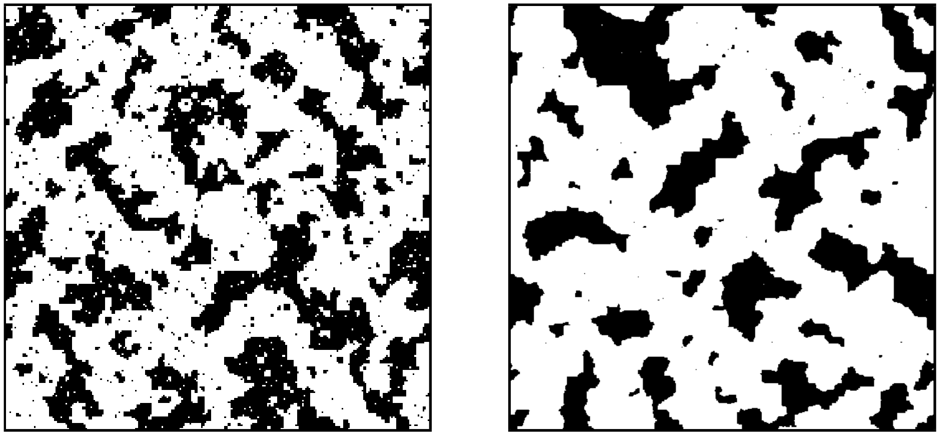
(a) Training Image

(b) Ternary-Hierarchical (Histogram, 2 min, 20 sec)



(c) Flat Annealing (Chord-Length, 3 days)

(d) Flat Annealing (Chord-Length + Two-Point, 3 days)



(e) Binary-Hierarchical (Chord-Length, 5 min)

(f) Binary-Hierarchical (Chord-Length + Two-Point, 15 min)

Figure 6.4: A comparison of our proposed ternary-hierarchical method (b) with regular, flat annealing (c, d) and regular binary-hierarchical annealing [1] (e, f). The frozen-state approach outperforms the others by a wide margin, both in terms of computational complexity and in terms of reproduced morphology.

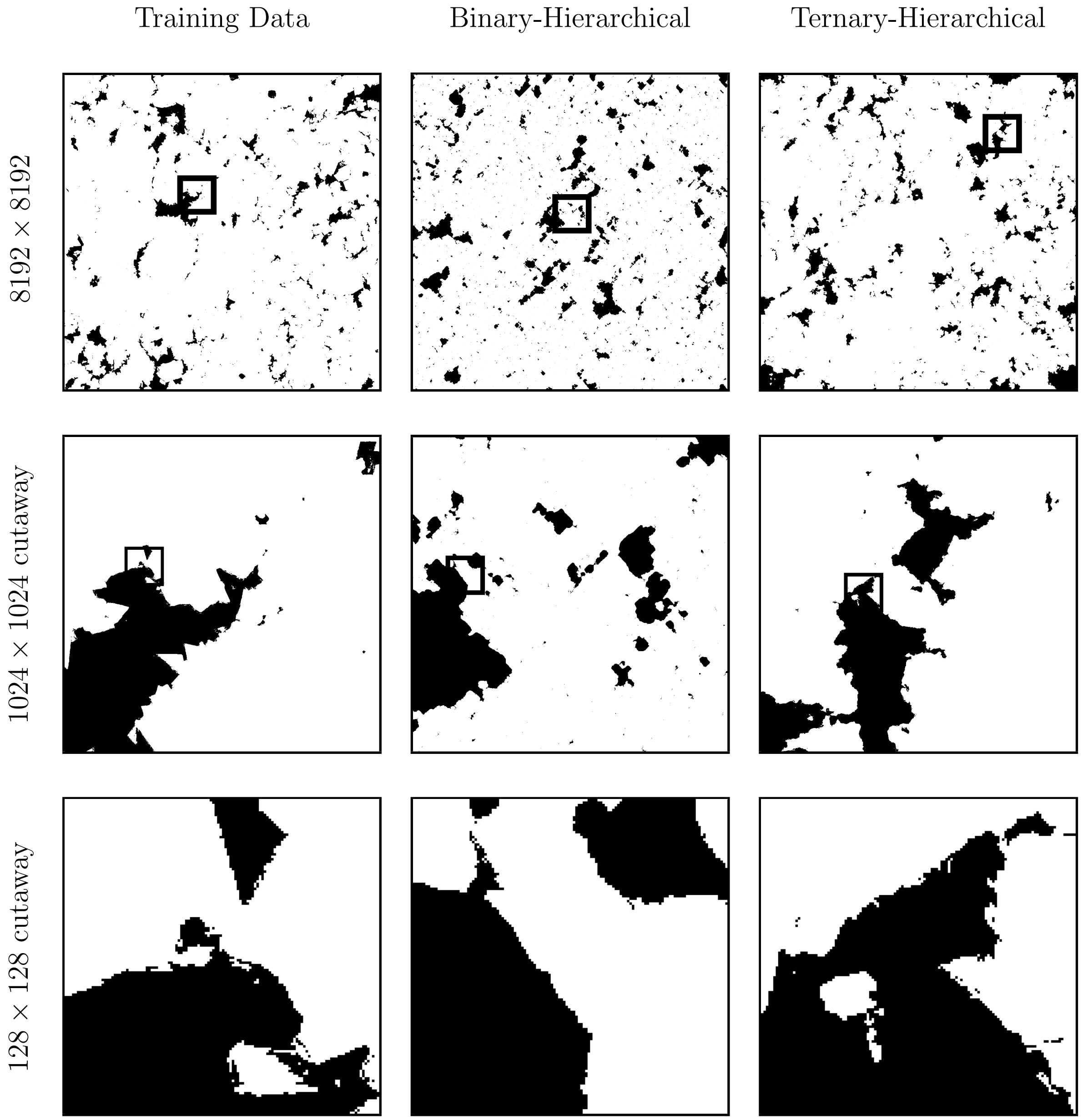

Training Data
Binary-Hierarchical
Ternary-Hierarchical
8192 × 8192
1024 × 1024 cutaway
128 × 128 cutaway

Figure 6.5: A zoom-in comparison of actual data and two sets of hierarchical results for a real carbonate rock example, showing how structures are faithfully reproduced at a variety of scales. The binary-hierarchical results are taken from [1].

The final point is key to the remarkably organic appearance of the structures along the boundary at the finest scale. Because the annealing is done hierarchically, this boundary region (what each model sees into the other) scales in size; the boundary is always a width of one pixel at the current scale, and as the pixels become finer, so too does the boundary. The end result has a very natural appearance, rather than the abrupt shift that one might expect. There was nothing specific in the annealing *telling* it to prefer grey along the boundary (as seen in the result at coarse scales); it was this overlap and the contradicting nature of the two models that forced them to compromise in this manner.

Figures 6.7 and 6.8 combine the averaging seen in Figure 4.5 with the hidden layer partitioning of Figure 6.6. Both images are composed of hidden bands; each band is controlled separately by a model that is composed of a weighted average between the large and small spheres models. In Figure 6.7, the leftmost band is simply a large spheres model, and the band to its right is an 80%/20% average between the large and small spheres models respectively, followed by 60%/40%, and so forth, ending with pure small spheres on the right. Figure 6.8 proceeds in an analogous manner, moving in concentric rings from the outer edge to the centre. Whereas the hidden layer of Figure 6.6 was composed of binary values specifying which of two models the corresponding pixel belonged, in these figures the hidden layer contains integers specifying which band contains each pixel. Functionally, all of these forms of hidden layers are the same, and the shift from one to two to more models brings only a linear increase in memory requirements (to store additional histograms) and no real change in computational complexity. The success of this method for asserting a gradient between two original disparate features (recall that the training images each contained *only* large or *only* small spheres) is self-evident; the results show the intended shift without any visible discontinuities along the band boundaries.

Some limitations to this approach are also visible. Note that, within the model, the behaviour with each band is still homogenous, and to some degree this is visible in the plotted results. The size of the bands offers a trade-off in this matter: narrower bands would allow one to more specifically assert the gradient, but would reduce the domain size for each band (as each band is evaluated against its own target histogram, independent of all the others), potentially limiting the variability within the band. One can also see towards the middle of Figure 6.7 how the large spheres model has encoded within it the sort of large gaps it expects between the spheres, and the averaging method still recreates these gaps even when combined with the small sphere model. In principle, the model is behaving correctly, but one can imagine situations where one would want the small spheres to fill those gaps more uniformly, and where this behaviour would be undesirable.

The substantial advances offered by the proposed method should make it of keen interest in a variety of annealing problems. Certainly for binary porous media, the huge potential of a method able to simulate 100-million pixel domains is to perform such simulation in three dimensions, which is the subject of future work.

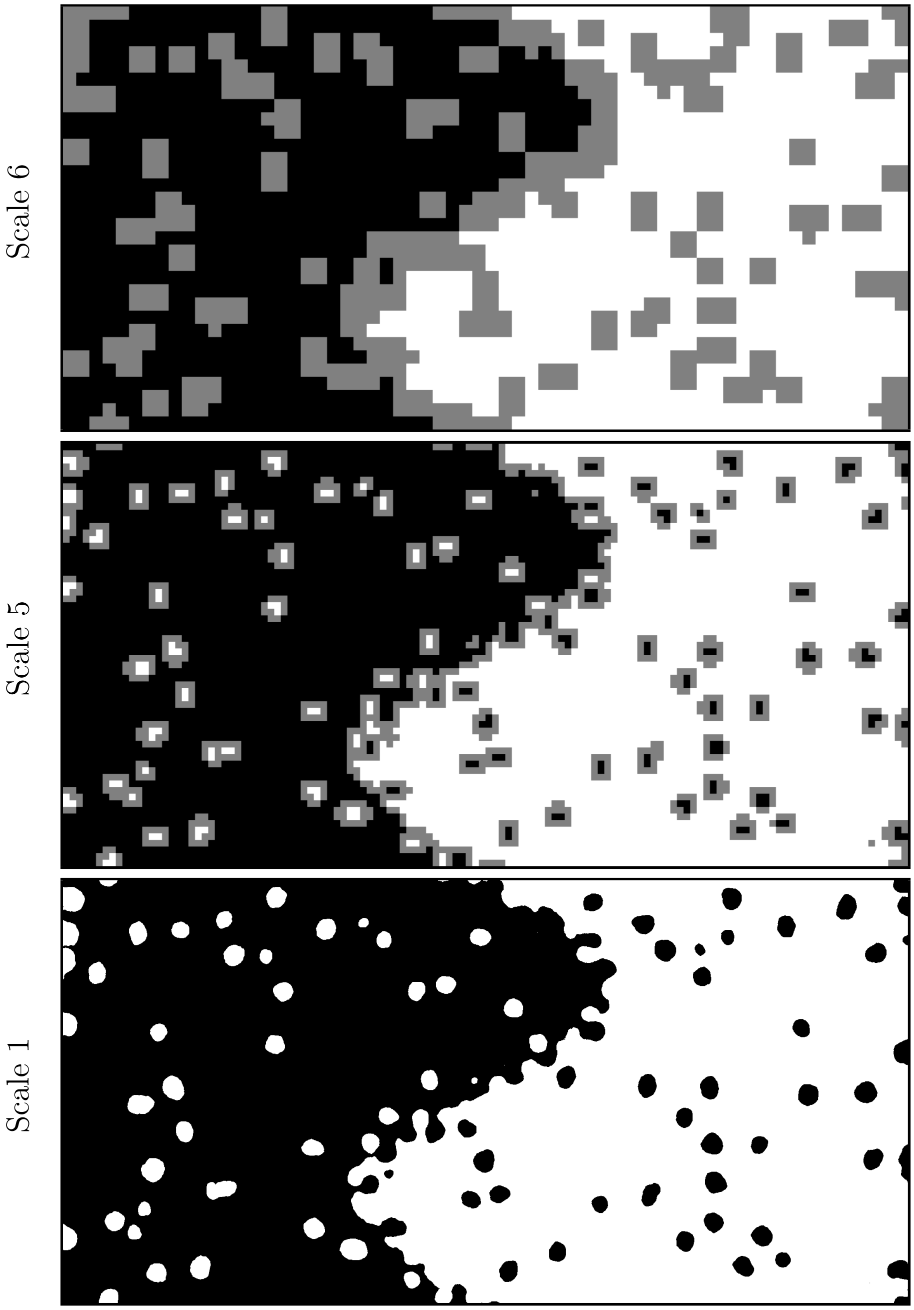


Figure 6.6: Three scales are shown from a two-model example, a white-circles model on the left and its complement (black circles on white) to the right. The presence of a sinusoidal boundary separating the two model regions is evident, but discerning its exact path is difficult: both models were able to accommodate what they were able to see on the other side of the boundary, and shape their portion of it into something consistent with itself and continuous with the other side.

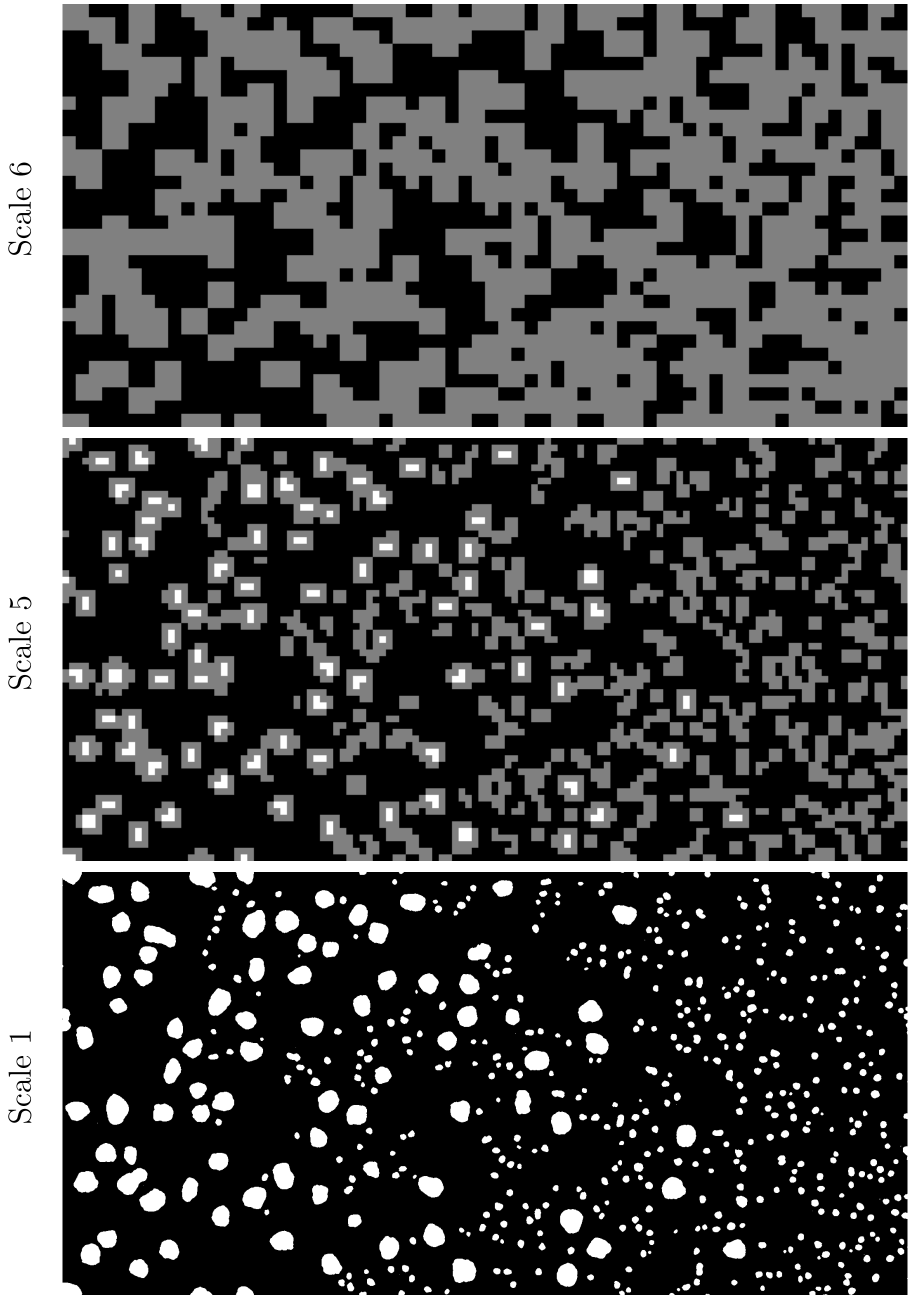


Figure 6.7: A hidden field is defined, separating the image into six vertical bands, where each band contains a differently-weighted mix of large and small sphere models. The resulting model transitions from a pure large spheres model on the left, to a pure small spheres model on the right.

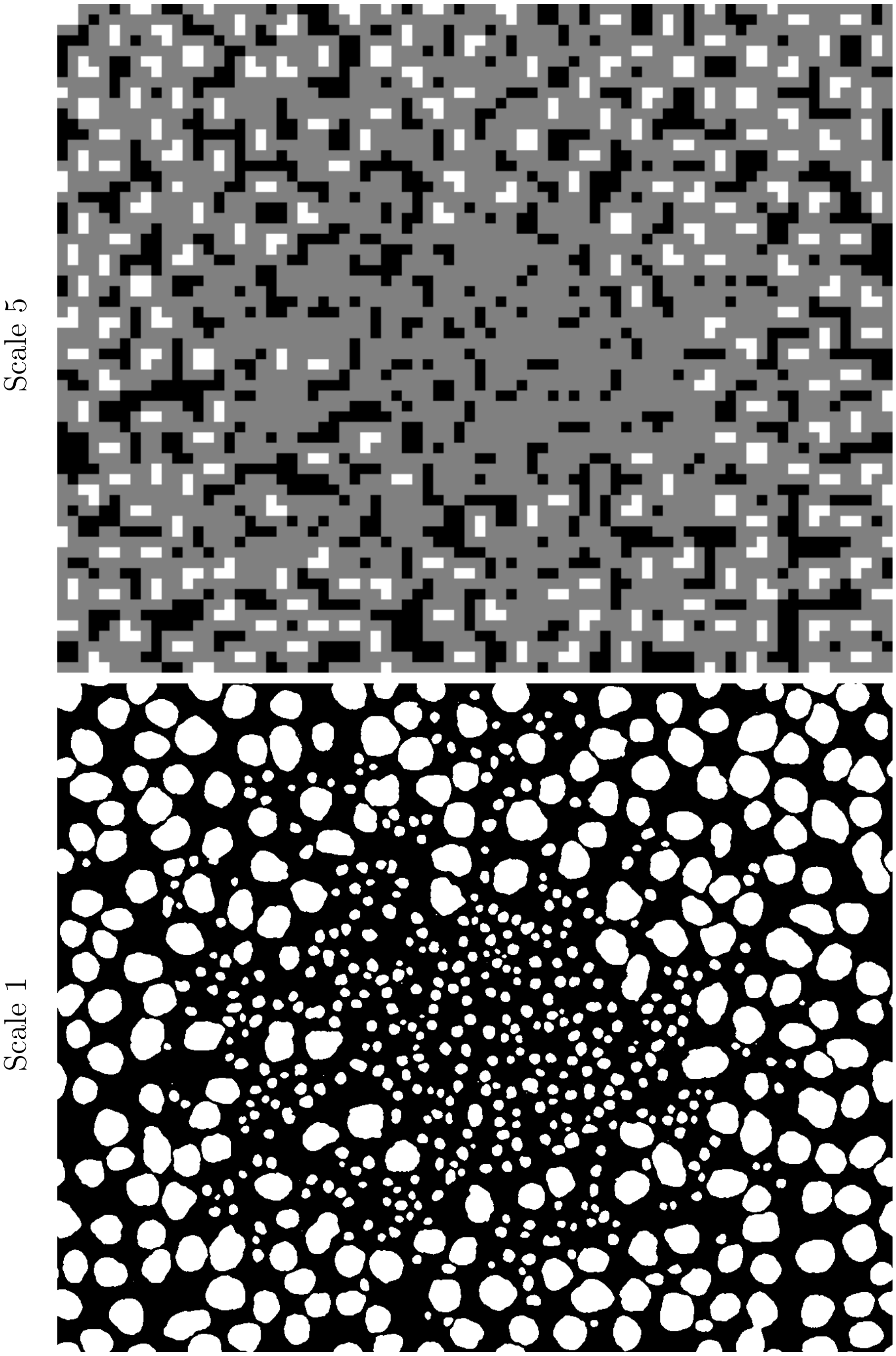


Figure 6.8: Similar to Fig. 6.7, here a hidden field separates the image into five concentric circular bands, where each band contains a differently weighted mix of large and small sphere models. The model transitions from small spheres in the centre to large spheres on the edges.

Chapter 7

Conclusions and Recommendations

The main conceptual contributions of this thesis appear in Chapter 3, which introduced the frozen-state concept, and in Chapter 4, which combined the frozen-state method with a local energy model to produce the technique analyzed in subsequent Chapters 5 and 6.

7.1 Summary of Contributions

The specific contributions of this work are as follows:

- The ternary frozen state model, defined in Section 3.2.
- Adapting the local histogram energy function for use with frozen state hierarchical annealing (Section 4.2).
- Together, frozen state and the local histogram energy function create a model composed of explicitly scale-dependent components. This separation of scales improves the quality of results and makes sampling from the model more robust compared to previous hierarchical methods (by removing the limitations discussed in Section 3.1).
- The proposed method also offers at least two orders of magnitude reduction in computational complexity over existing hierarchical methods without reducing the quality of the results. Although the exact gains are problem-dependant, the number of pixels simulated at finest scale — the dominating computational factor in previous hierarchical methods — is typically reduced to 1% to 5%, as demonstrated in Figures 5.2 and 6.3. This makes possible the synthesis of 8192×8192 images in as little as two hours, with equal or better quality results, as shown in Figure 6.5. Figure 6.4 compares the proposed method with flat annealing methods: in two minutes, the frozen-state approach produced a high-quality result, while the flat methods were not remotely close to converging even after three days.

- The techniques explored here have a degree of generalizability lacking in previous methods: Figure 4.5 demonstrates the ability to combine distinct models to produce a hybrid result expressing features from both. Figure 6.6 explores the potential use of hidden fields, and shows the method's robustness against boundary artefacts. Figures 6.7 and 6.8 show how the combined application of those two abilities can create useful results that could not be produced by previous methods.

- These techniques — specifically, the computational gains, the scale-dependent nature of the energy model, and the ability to efficiently represent sparse fields — open the potential for 3D modelling in areas where it was previously intractable.

7.2 Extensions and Future Work

In the course of this research, two main avenues for future work were identified.

The first is in modelling. The methods described in this thesis are still limited by the use of a single hierarchy as the model. What aspects of a random field can we not capture with a single hierarchy? How might one use hidden fields to model non-stationary or non-isotropic phenomenon?

A second goal is to extend the technique disclosed in this thesis to three dimensions. Computationally, the 8192^2 pixel images generated require a similar order of computations as a volumetric image with 256^3 elements, and the frozen state concept naturally lends itself to well oct-tree-like sparse data structures. However, other concerns also need to be addressed: the energy function must be adapted to accommodate or avoid the curse of dimensionality that it currently faces. Sparsity in the 3D training data also becomes a concern. Could such a 3D object be approximated using 2D random fields, and would the result be meaningful? Asserting certain properties of the porous medium structure, such as isotropy, and incorporating them into the model could help overcome some of these obstacles, however the validity of such assertions (and the feasibility of exploiting them) would need to be examined on a case-by-case basis.

References

[1] S.K. Alexander, P. Fieguth, M.A. Ioannidis, and E.R. Vrscay. Hierarchical annealing for synthesis of binary porous media images. *Mathematical Geosciences*, 41(4):357–378, 2009.

[2] S.K. Alexander, P. Fieguth, and E.R. Vrscay. Hierarchical annealing for random image synthesis. In *Fourth International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, volume 2683 of *Lecture Notes in Computer Science*, pages 194–210. Springer, 2003.

[3] S.K. Alexander, P. Fieguth, and E.R. Vrscay. Hierarchical annealing of porous media. In *SIAM Conference on Mathematical and Computational Issues in the Geosciences*, Avignon, 2005.

[4] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *J. Royal Statistical Society B*, 36(2):192–236, 1974.

[5] J. Bramble. *Multigrid Methods*. Wiley, 1993.

[6] P.J. Burt and E.H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communication*, 31(4):532–540, 1983.

[7] W.R. Campaigne, P. Fieguth, and S.K. Alexander. Frozen-state hierarchical annealing. In *International Conference on Image Analysis and Recognition*, volume 4141 of *Lecture Notes in Computer Science*, pages 41–52. Springer, 2006.

[8] D. Chandler. *Introduction to Modern Statistical Mechanics*. Oxford University Press, 1987.

[9] L.S. Davis, S.A. Johns, and J.K. Aggarwal. Texture analysis using generalized cooccurrence matrices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1:251–259, 1979.

[10] P. Fieguth. *Statistical Image Processing and Multidimensional Modeling*. Springer, 2011.

[11] M. Figueiredo and R. Nowak. Wavelet-based image estimation: An empirical bayes approach using jeffreys' noninformative prior. *IEEE Transactions on Image Processing*, 10(9):1322–1331, 2001.

[12] M.E. Fisher. Renormalization of critical exponents by hidden variables. *Physical Review*, 176(1):257–272, 1968.

[13] A. Frigessi, P. Di Stefano, C. Hwang, and S. Sheu. Convergence rates of the gibbs sampler, the metropolis algorithm and other single-site updating dynamics. *J. Royal Statistical Society B*, 55(1):205–219, 1993.

[14] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

[15] A. George and J. Liu. *Computer Solution of Large Sparse Positive Definite Systems.* Prentice-Hall, 1981.

[16] B. Gidas. A renormalization group approach to image processing problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(2):164–180, 1989.

[17] W. Hackbusch. *Multi-Grid Methods and Applications.* Springer, 1985.

[18] W.K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[19] A.K. Jain and F. Farrokhnia. Unsupervised texture segmentation using gabor filters. *Pattern Recognition*, 24(12):1167–1186, 1991.

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

[21] W. Lenz. Beiträge zum verständnis der magnetischen eigenschaften in festen körpern. *Physikalische Zeitschrift*, 21:613–615, 1920.

[22] S. Mallat. A theory of multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.

[23] V.V. Mangazeev, M.Y. Dudalev, V.V. Bazhanov, and M.T. Batchelor. Scaling and universality in the two-dimensional ising model with a magnetic field. *Physical Review E*, 81, 2010.

[24] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[25] N. Metropolis and S. Ulam. The monte carlo method. *Journal of the American Statistical Association*, 44(247):335–341, 1949.

[26] Y. Nourani and B. Andresen. A comparison of simulated annealing cooling strategies. *Journal of Physics A: Mathematical and General*, 31:8373–8385, 1998.

[27] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29:51–59, 1996.

[28] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2002.

[29] J. Portilla and E.P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision*, 40(1):49–71, 2000.

[30] T. Preis, P. Virnau, W. Paul, and J.J. Schneider. Gpu accelerated monte carlo simulation of the 2d and 3d ising model. *Journal of Computational Physics*, 228(12):4468–4477, 2009.

[31] M.G. Rozman and M Utz. Efficient reconstruction of multiphase morphologies from correlation functions. *Physical Review E*, 63, 2001.

[32] H. Samet. The quadtree and related hierarchical data structures. *ACM Computing Surveys*, 16(2):187–260, 1984.

[33] G. Strang. Wavelets and dilation equations: A brief introduction. *SIAM Review*, 31(4):614–627, 1989.

[34] R.H. Swendsen and J.S. Wang. Nonuniversal critical dynamics in monte carlo simulations. *Physical Review Letters*, 58(2):86–88, 1987.

[35] H. Szu and R. Hartley. Fast simulated annealing. *Physics Letters A*, 122(3-4):157–162, 1987.

[36] M. Talukdar, O. Torsaeter, and M. Ioannidis. Stochastic reconstruction of particulate media from two-dimensional images. *Journal of Colloid and Interface Science*, 248(2):419–428, 2002.

[37] S. Torquato. *Random Heterogeneous Materials: Microstructure and Macroscopic Properties*. Springer, 2001.

[38] A. Tremeau and N. Borel. A region growing and merging algorithm to color segmentation. *Pattern Recognition*, 30(7):1191–1203, 1997.

[39] A. Tsakiroglou and M. Ioannidis. Dual-porosity modelling of the pore structure and transport properties of a contaminated soil. *European Journal of Soil Science*, 59:744–761, 2008.

[40] S. Wesolkowski. *Stochastic Nested Aggregation for Images and Random Fields*. PhD thesis, Dept. of Systems Design Engineering, University of Waterloo, 2007.

[41] K. Wilson. Problems in physics with many scales of length. *Scientific American*, 241:158–179, 1979.

[42] G. Winkler. *Image analysis, Random Fields, and Markov Chain Monte Carlo Methods.* Springer, second edition, 2003.

[43] A. Wong, A. Mishra, W. Zhang, P. Fieguth, and D.A. Clausi. Stochastic image denoising based on markov-chain monte carlo sampling. *Signal Processing*, 91(8):2112–2120, 2011.

[44] G. Wornell and A. Oppenheim. Estimation of fractal signals from noisy measurements using wavelets. *IEEE Transactions on Signal Processing*, 40:611–623, 1992.

[45] M. Yaou and W. Chang. Fast surface interpolation using multiresolution wavelet transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(7):673–688, 1994.

[46] C.L.Y. Yeong and S. Torquato. Reconstructing random media. *Physical Review E*, 57:495–506, 1998.

[47] H. Yserentant. Two preconditioners based on the multilevel splitting of finite element spaces. *Numerische Mathematik*, 58(2):163–184, 1990.