An Algorithm for Identifying and Tracking Structures in Time-Varying Physical Fields

Iván Ordóñez
Dept. of Computer and Information Science
The Ohio State University
2015 Neil Avenue
Columbus, OH 43210
Email: iordonez@cis.ohio-state.edu

Feng Zhao
Dept. of Computer and Information Science
The Ohio State University
2015 Neil Avenue
Columbus, OH 43210
and
Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304
Email: fz@cis.ohio-state.edu

September 21, 1998

Abstract

Data sets modeling spatially distributed physical phenomena often exhibit regions of uniformity. Example of such data sets include spatial patterns in certain Diffusion-Reaction systems, weather maps, and pressure distribution in regions of sea.

This paper describes an algorithm for identifying and tracking coherent regions in spatial fields produced by a class of diffusion-reaction systems. Such systems describe a number of physical, chemical, and biological phenomena, and their study may shed light on how nature constructs and evolves structures that exhibit a high degree of regularity. The algorithm adaptively samples a spatial field using a particle system, aggregates the sampled points into a neighborhood graph, classifies the structure into coherent regions, and tracks the regions over time to produce a qualitative description of the temporal evolution of the field. Because the adaptive sampling grid varies smoothly with the temporal evolution of the underlying field, the algorithm is able to efficiently track the corresponding objects over successive time frames by minimally updating the grid.

1 Introduction

Spatial data sets typically arise when spatially distributed physical fields are discretized for the purpose of simulation or analysis. Some examples of the spatial fields are weather maps, isothermal regions in the sea, or pattern formations in natural systems, such as convection rolls or diffusion-reaction systems. The analysis of these discrete data sets is essential in scientific visualization, modeling, or generating qualitative interpretations. Because of the recent advances in computational methods and hardware, there have been increasing interest in automated means for mining large spatial data sets. In particular, the Spatial Aggregation (SA) approach [12] provides a framework for the identification of structures in spatially-distributed fields. This paper extends SA to address problems that vary over time, and develops an algorithm for reasoning about temporal and causal structures. Our ultimate goal is to automatically generate a taxonomy of behavior classes for diffusion-reaction phenomena and infer from data high-level laws that govern the behaviors of these systems. Although this paper uses the diffusion-reaction fields to illustrate the main idea, the algorithm is applicable to a class of problems that exhibit coherent spatial regions.

Regions of uniformity arise in a physical field because of continuities of properties such as intensity or temperature or pressure [12]. A human would have little trouble describing such events in qualitative terms. For example, a person observing heating water might say: "after several seconds of apparent rest, four convective rolls arose spontaneously; they rock back and forth slowly, as if oscillating with respect to a fixed point." Such a description recognizes several events: the existence of coherent objects, their persistence through time, and their gradual change. Any attempt to study complex phenomena of this kind would require the automatic generation of such high-level descriptions from raw data.

One way such regions of uniformity interact is to form regular structures, the so-called patterns. An interesting example of pattern formation is found in biological systems. The emergence of shape in growing organisms is a phenomenon that demands adequate explanation, particularly because it is clear that such growth is not controlled by any central agent. Turing was the first to propose a model [9] to account for the phenomenon of *morphogenesis,* and it makes use of the fact that substances that diffuse through an inert medium and react with each other tend to form patterns of highly organized shapes under appropriate circumstances. The emergence of patterns is hardly limited to biological systems. In his study of complex patters, Pearson [8] describes problems such as the formation of snowflakes, cloud streets, sand ripples and convective rolls in hydrodynamic experiments. This demonstrates that pattern formation is a phenomenon of great scientific interest.

There are several standard algorithms that identify uniform regions in data sets. Naive region growing, for example, starts with arbitrary points in the sample and builds regions by aggregating other points that share similar features around the seeds. The marching cubes algorithm [6], on the other hand, finds iso-surfaces more efficiently by applying a clever heuristic, but it has to consider explicitly all the ways a surface may intersect a cube. All of these algorithms depend on the level of discretization in the data set, and, more importantly, they are developed primarily for static fields. When the field varies over time, as is the case with the diffusion-reaction phenomena, it is necessary to re-compute the structures at every time step in the standard algorithms, and then establish correspondence among structures across time. Yip [11] described an algorithm to extract topologically consistent 3D structures in the domain of fluid dynamics. The algorithm identifies objects of interest in a fluid field and tracks their evolution. Yip's work inspired us to develop an adaptive sampling algorithm that can efficiently track object evolution.

In this paper a structure-identifying algorithm that does not suffer from the drawbacks of naive region growing and other standard methods will be introduced. It is based on the central idea that qualitative structures of a spatial field can be constructed from an adaptive spatial subdivision rather than directly from a regularly discretized field. This adaptive subdivision changes as the field changes, but the identity of

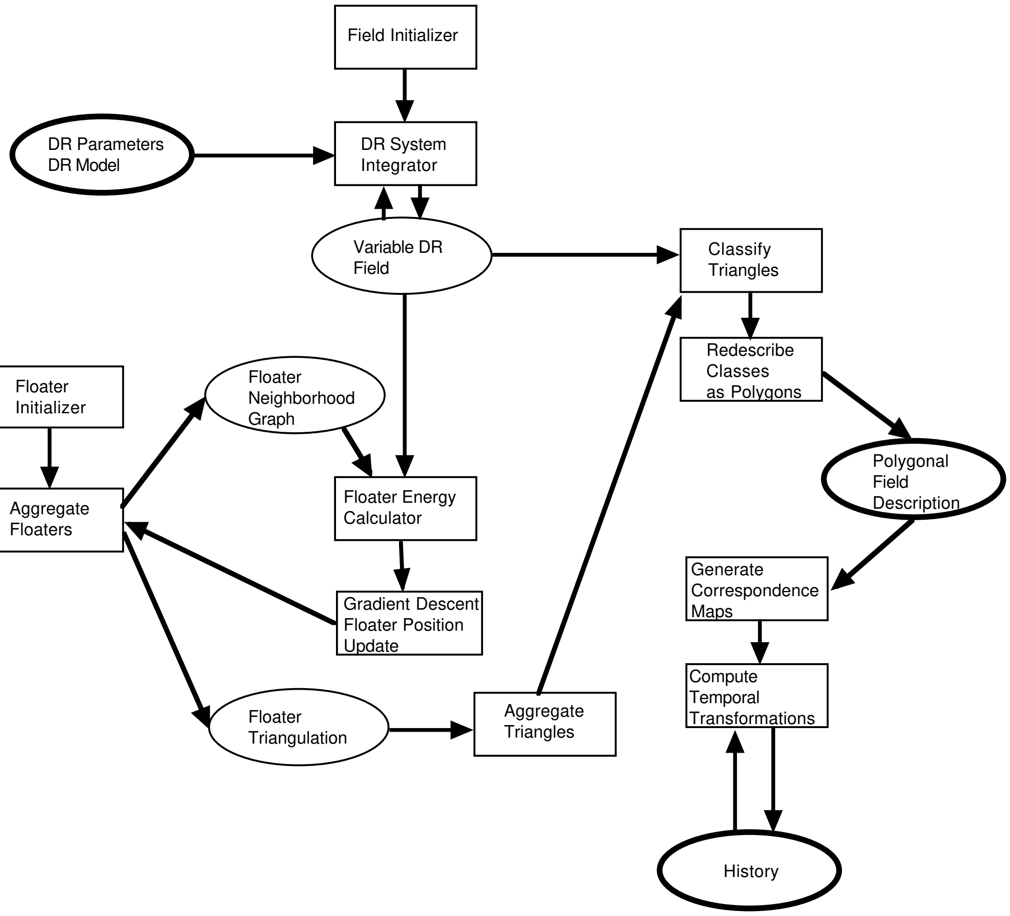

Figure 1: A flowchart for the main components of the identification and tracking algorithm. Boxes represent operations, while ovals indicate objects.

its structural components is persistent. The persistence of these components simplifies the correspondence between successive temporal snapshots. We will illustrate the algorithm using the 2D reaction-diffusion fields. The algorithm is applicable to higher dimensional problems, although the computational complexity of such problems remains to be addressed.

Our algorithm comprises several elements, as shown in Figure 1. The system simulator takes as input a field model, its parameters, and a set of initial conditions, and numerically integrates the model to generate a continuously varying field. After initialization, the particle system uses the field gradient information to form a sampling mesh. The triangle elements in the particle mesh are aggregated to form polygons describing regions of uniformity. Tracking these polygonal objects over time, the algorithm generates a high-level description of the history of the objects.

The rest of the paper is organized as follows. Section 2 introduces pattern formation in several diffusion-reaction systems. Sections 3 and 4 present the algorithm for adaptively subdividing the field. Section 5 describes the extraction of polygonal description. Section 6 describe temporal correspondence of objects. Section 7 discusses computational experiments that have been completed.

2 Pattern Formation in Diffusion-Reaction Systems

A typical instance of time-varying fields is the set of phenomena known as Diffusion-Reaction. These phenomena are of great scientific importance, because they are associated with the problem of morphogenesis, first addressed by Turing [9]. We will pick a particularly interesting instance, where noticeable patterns emerge and vary in seemingly unpredictable ways, and develop an algorithm to automate the extraction of such patterns.

Diffusion-reaction systems model the dispersive behavior of concentrations of chemicals. In [3] we find a generic derivation of a general Diffusion-Reaction model, summarized as follows: Let $c(x,t)$ be the density function of a substance in a certain medium, where x is a vector in the domain space and t is time; let $Q(x,t)$ be the rate of creation of the substance, i.e. the rate at which its concentration increases; and let $J(x,t)$ be the flux density, i.e. the rate at which the substance crosses a unit of area. We can relate these three functions by noticing that in any particular volume, the instantaneous change in substance concentration must be the instantaneous creation rate minus the flow through the border of this volume. From this, through application of the divergence theorem we obtain a general balance law:

$$\frac{\partial c}{\partial t} = -\nabla \cdot J + Q. \tag{1}$$

We specify this model further through the application of Fick's law, which states that in common diffusion the flux is proportional to the gradient of the density:

$$J = -D\nabla c, \tag{2}$$

where D is the diffusion rate. Replacing above, we obtain the general Diffusion-Reaction equation:

$$\frac{\partial c}{\partial t} = D\nabla^2 c + Q. \tag{3}$$

We are interested in cases where there are two reactants which diffuse independently but whose reaction components are coupled together. In such cases, our general model is given by

$$\begin{aligned} \frac{\partial u}{\partial t} &= D_u \nabla^2 u + f_u(u,v) \\ \frac{\partial v}{\partial t} &= D_v \nabla^2 v + f_v(u,v) \end{aligned} \tag{4}$$

where u and v are the concentrations of the two reactants and D_u and D_v are their diffusion rates.

2.1 The Gray-Scott Model of Glycolysis

The phenomenon of glycolysis is found in virtually all living organisms. It probably evolved before the existence of an oxygen-rich atmosphere, because it allows the extraction of energy from sugar in the absence of free oxygen.

A glycolysis process has the purpose of generating ATP (adenosin-triphosphate), the main cellular energy "currency", from the reduction of glucose. This process is possible due to the fact that glucose is a highly

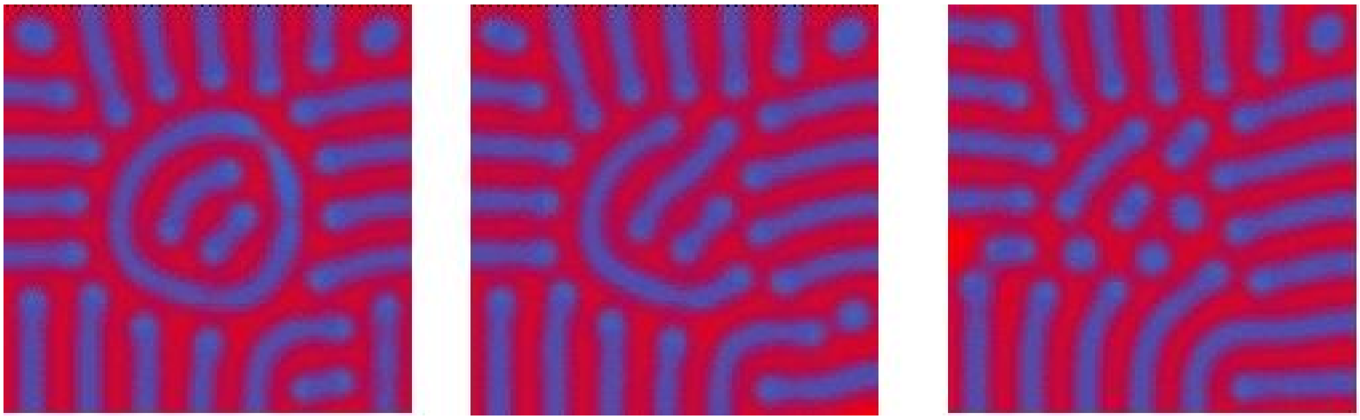

Figure 2: Three successive snapshots of a pattern in the Gray-Scott model of Equations 5 for values $D_u = 2E-5$, $D_v = 1E-5$, $F = 0.03$, $k = 0.06$.

energetic molecule. The liberation of some energy from atomic rearranging allows to store this energy in the form of ATP molecules. All steps in the glycolysis process are mediated by enzymes, as follows: Starting with a glucose molecule, glycolysis begins by exposing the molecule to phosphorylation (the acquisition of two phosphorus atoms) through the intervention of two ATP molecules. The ATPs lose one phosphorus each, so they are reduced to ADP (adenosin-diphosphate).

The transformed glucose molecule is then split into two, and each half (a dehoxyacetone phosphate) is phosphorylized again using an inorganic phosphate (not ATP). Then, through two separate dephosphorylization steps, two ATP molecules are made in combination with ADT, leaving lactate as a remainder. Though glycolysis consumes two ATP molecules, it produces four, thus yielding a net gain of two ATP molecules.

As seen above, glycolysis is an autocatalytic process, because ATP is used to produce more ATP. A model of glycolysis introduced by Gray and Scott was first studied from a computational perspective by Pearson [8]. It is a simplified model, which abstracts away the details of this complex phenomenon, and considers two reactants, U and V, which interact in an irreversible autocatalytic process: $U + 2V \rightarrow 3V$. Because the reaction speeds are proportional to the molarity of the reactants, it follows that U is lost at a rate proportional to uv^2, while V is gained proportionally to the same rate. To compensate for the loss, U is fed continuously into the system at a rate F. On the other hand, V is assumed to be consumed by some other process at a constant rate k. Finally, both U and V are removed by the feed process, so we are left with the following equations:

$$\begin{aligned} \frac{\partial u}{\partial t} &= D_u \nabla^2 u - uv^2 + F(1-u) \\ \frac{\partial v}{\partial t} &= D_v \nabla^2 v + uv^2 - (F+k)v. \end{aligned} \tag{5}$$

This model is of interest because it exhibits a variety of behaviors unlike anything observed before in theoretical or numerical studies.

In Figure 2 we show three successive snapshots for a Gray-Scott system, for a particular set of parameters. Colors code the concentration of the substances: red indicates high concentration of substance **U** and low concentration of substance **V**, whereas blue indicates the opposite (if this paper is not printed in color, red regions are seen as dark grey, and blue regions as light grey). It should be pointed out that different combinations of the four parameters D_u, D_v, F and k produce very different behaviors in the formation of patterns for this systems. These behaviors range from the formation of stable hexagonal patterns to fast-changing bodies in perpetual chaotic motion.

Pearson has argued that since glycolysis occurs inside the cell, it is possible that patterns such as the above could form within the cell. Furthermore, he observes that the process of mitosis, by which cells divide, requires the formation of a bipolar structure known as the mitotic spindle, which

> ...consists of two organizing centers called centrosomes from which microtubule fibers emanate. The replicated chromosomes migrate to the spindle equator and are then separated by forces exerted by the microtubule on the chromosomes. These forces are generated by the action of the ATP engines dynein and kinesin, which convert chemical potential energy to work. The mitotic spindle forms not only in cells with artificial chromosomes, but also in cells from which the entire nucleus has been completely removed. Moreover, microtubules are observed to organize into a coherent network in cell fragments containing no centrosomes or chromosomes. Thus it seems much more likely that the formation of the ubiquitous spindle is governed by simple physical processes such as chemical reactions and diffusion than by complex genetic mechanisms. [7]

Hence it seems very likely that, as Turing suggested, there is a strong link between the qualitative behavior of Diffusion-Reaction systems and the formation of structures in living organisms.

2.2 The Ferrocyanide-Iodate-Sulphite (FIS) Reaction

Lee et al. [5] have studied a phenomenon in inorganic chemistry that has a model similar to glycolysis and is feasible to carry out in a laboratory:

> The laboratory chemical patterns form in a thin transparent gel (0.4 mm thick, 22 mm diameter) whose bottom surface is in contact with a well stirred reservoir (3.0 ml volume) that is continuously refreshed with the reagents of the reaction. Transitions are studied by changing the input ferrocyanide concentrations with other parameters held fixed ... chemical patterns ... are viewed with a video camera in reflected light in a band centered at 420 nm where ferrocyanide absorbs strongly.

The FIS reaction exhibits two stable states [4], one with low pH (about 4) and the other with high pH (about 7). The model studied by Lee et al. follows the equations

$$\begin{aligned} \frac{\partial u}{\partial t} &= D_u \nabla^2 u - uv^2 + A(1-u) \\ \frac{\partial v}{\partial t} &= D_v \nabla^2 v + uv^2 - Bv, \end{aligned} \tag{6}$$

where A and B represent the strengths of the coupling of U and V to the reservoir. Notice that if we make $A = F$ and $B = F + k$, we have the same model as Gray-Scott's. Hence we can think of Figure 2 as representing a FIS reaction, where blue regions represent the low pH state, and red regions represent the high pH state. From now on, we shall refer to the FIS and the Gray-Scott models interchangeably.

3 Description Through Iso-Surfaces

The existence of coherent structures in space implies that there are regions of more or less uniform characteristics. For example, in the Gray Scott model, each region clearly belongs to one of two classes, low or high pH, depicted as red or blue. The fewer the number of classes observed, the higher the order perceived by an observer. As the number of classes increases, uniform regions become smaller, and in the limit there is nothing but a random field. For this reason, local uniformity is one of the main features to look for when studying patterns. Once regions of uniformity are identified, we can study their characteristics, such as topology, spatial distribution, and temporal behavior.

Cluster boundaries are associated with field regions of high gradient. To identify such regions we need to construct iso-surfaces, continuous zones of equal field value. The ratio of field value change to the distance between iso-surfaces gives an estimation of the gradient. Therefore, a field that is characterized by more or less uniform regions that vary smoothly is well described by iso-surfaces that sample evenly-spaced field values. All the problems described above —fluid dynamics analysis, weather maps, Diffusion-Reaction fields— are well suited to description through iso-surfaces. Therefore we study temporal variations in fields through the study of geometric and topological change in iso-surfaces.

We note that in Diffusion-Reaction systems the stable regions typically constitute solutions to the differential equations that describe the systems. Boundaries between regions tend to be very sharp (the field gradient is high), so choosing a threshold value to separate the regions is not particularly difficult. Also, small variations in this threshold value do not significantly affect the behavior of classification algorithms.

4 Constructing an Adaptive Subdivision

We want to divide the field into regions whose size corresponds to the degree of uniformity of the field: wherever the gradient is high, regions should be small, and wherever the gradient is small, regions should be large. This way, we have high resolution only where it is needed, and we can later aggregate the regions to form bodies whose boundaries approximate iso-surfaces that correspond to cluster boundaries. Furthermore, we would like our subdivision to be re-usable, meaning that with minimal recomputing efforts, we can update it to reflect changes in the field. For this reason, the components of the subdivision should be persistent, as they will allow us to identify time-varying spatial structures in the field. Now we proceed to explain how we achieve this.

The main idea is that of sampling the field by means of a system of particles that float more or less freely; we call the particles *floaters*. These floaters will move as the field changes, always keeping an accurate sampling; they do it following a modification of an algorithm proposed in [10]: floaters repel each other and modify their distribution and density to compensate for under or oversampling. Whereas Witkin and Heckbert's algorithm is meant to sample implicit surfaces, namely those described by an implicit mathematical equation, we adapt it to be used on explicit fields. We do this by treating the field as an n-dimensional hypersurface in $n+m$ dimensions, where the field is a mapping from $\Re^n \rightarrow \Re^m$. In our particular case, a 2D Diffusion-Reaction field, we will use the field value of one of the reactants as a third dimension (in our case, the other field value is highly correlated to the first, so we can ignore it). The floaters interact with each other by computing their distances along this 3D surface.

To make floaters repel each other, following Witkin and Heckbert, we employ a Gaussian energy function that increases when floaters get close together. For any two floaters i and j, their mutual energy is

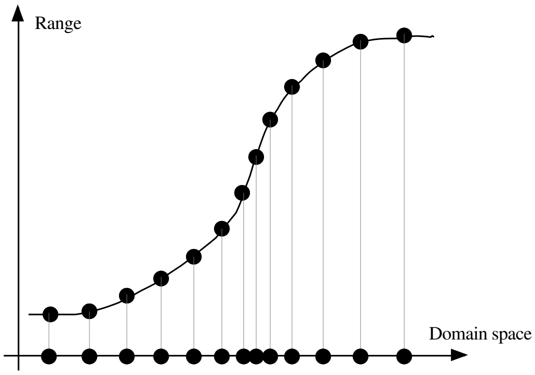

Figure 3: The projection of floater positions into the domain of the field produces high floater density in high gradient regions.

$$E_{ij} = \alpha e^{-\frac{|r_{ij}|^2}{2\sigma^2}}, \tag{7}$$

where α is a global constant, r_{ij} is the distance between floaters i and j, and the energy for each floater is given by

$$E_i = \sum_{j=1}^{n} E_{ij}. \tag{8}$$

Floaters are assigned a velocity that is negatively proportional to the gradient of energy, such that their local energy is minimized. Notice that the energy of a floater is a measure of how close it is to other floaters. By minimizing this energy, floaters seek to be as far away from each other as possible, and they can only do this by placing themselves in a position approximately equidistant to all their neighbors. In all, this algorithm amounts to using local search to minimize the total energy of the floaters. In a flat surface, a global solution would be a uniform hexagonal grid.

When the surface changes so that its area varies locally, the above scheme will leave too many floaters in contracting regions, and too few in expanding ones. To solve this problem two measures are adopted: first, instead of using a constant, global value for σ, each floater is assigned an adaptive σ_i that grows when the floater is in sparsely populated regions and shrinks when it is in crowded areas. And second, floaters whose σ_i grows too much are allowed to generate offspring, and where it has shrunk too much, they are allowed to die. This ensures an even distribution of floaters throughout the surface. The value of $sigma_i$ is varied using a simple delta-rule that varies it in proportion to the deviation of the floater's energy from a fixed "optimal" energy, so that when the energy is higher than the optimal σ shrinks, and when it is lower, σ grows.

We use the positions of the floaters to construct a spatial subdivision. To do this, we project the positions of the floaters to the domain, which has the effect of producing a spatial distribution where floater density is approximately proportional to field gradient, as illustrated in Figure 3. The subdivision is computed by dividing the space into simplices whose vertices are the floaters; in other words, we need to solve the n-dimensional triangulation problem. Further, we wish that the simplices be small and non-sharp, so we

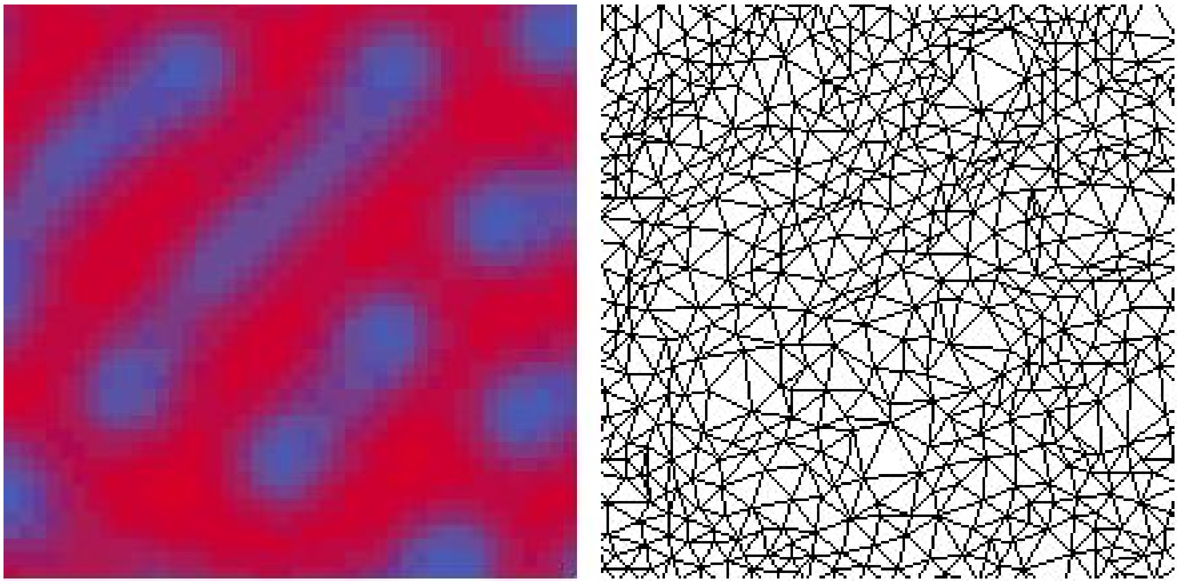

Figure 4: Patterns in a field and their floater representation.

use a Delaunay triangulation, which offers the added advantage that it can be computed efficiently in the 2D case. Also, this triangulation is a superset of the closest neighbor graph, and therefore it captures the notion of spatial locality: local variations in a floater's position cause changes in the triangulation that do not propagate beyond the immediate vicinity of the floater.

We also introduce an optimization to Witkin and Heckbert's algorithm in the following fashion: First, we notice that we have an explicit representation of a neighborhood relationship between floaters (defined by the edges of the triangulation). Next, we see that the energy contribution of surrounding floaters decreases exponentially with the distance. Hence we can approximate the calculation of the floater energy considering only each floater's immediate neighbors. In average, this reduces the complexity of the energy computation algorithm from $O(n^2)$ to $O(n)$. This optimization extends to higher dimensions as long as there is an efficient means to construct a neighborhood relationship.

In Figure 4 a snapshot of a Diffusion-Reaction field is shown next to a spatial subdivision (a Delaunay triangulation) built on a set of floaters that sample the field. The triangulation follows the structure of the field, a fact that may not be obvious by visual inspection, but which will become apparent once we annotate our triangles using their average field values, as will be explained in the next section.

As the field varies in time, so does the position of the floaters, which in turn causes the spatial subdivision to change: some edges cease to exist and some new ones arise at every time step. However, given the assumption that the underlying field changes slowly, the vast majority of edges and triangles are preserved, even though their shape is slightly changed. Herein lies the main advantage of the approach being described: since most structures are preserved across time, the need to use statistical methods to perform correlations decreases substantially. Moreover, it is not necessary to reconstruct the spatial subdivision at every time step; it is sufficient to update the parts that have varied. Because of the local nature of the Delaunay triangulation, these updates do not propagate far.

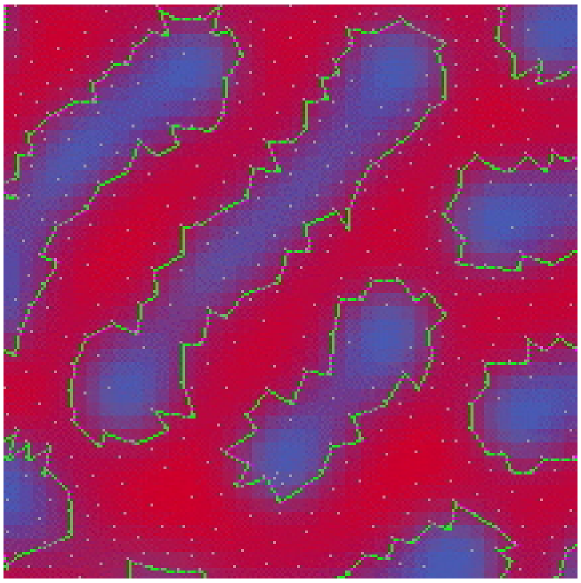

Figure 5: The aggregation of segments of the spatial subdivision yields an explicit representation of field structures.

5 Aggregating Substructures

The extraction of structures from the spatial subdivision is analogous to a pixel-based region growth algorithm, with the difference that the element of aggregation is not the pixel, but the triangle. In this way, instead of using pixel values as the criterion to aggregate elements, we use the average field value inside each triangle. In Figure 5 we show the result of carrying out this process in the snapshot shown in Figure 4. Since only two classes of structures are present, we only outline one kind.

Notice that the polygonal representation of the blue structures are somewhat lacking in smoothness. This is due to the fact that the overall density of floaters in the example is rather coarse. However, considering the relatively low pixel resolution in the underlying field, this polygonal representation is adequate, as it approximates the irregular shape of an iso-line surrounding the blue bodies. Also notice that by varying a single threshold value, it is possible to extract several iso-linear representations from a single spatial subdivision. However, no significant variation will be introduced in the obtained iso-lines unless the threshold is varied so much that it gets close to the average value of one of the structures.

The re-description of a connected group of triangles as a single polygonal body is of critical importance to achieve topologically correct super-structures. The algorithm to achieve this is straightforward: start with a set of triangles and proceed as follows:

```
initialize polygon P as empty
for every triangle t in the list
  remove from P those edges that are also in t;
  insert into P the remaining edges of t
end for
identify all connected regions in P, labeling them as separate borders
output P
```

Notice that the above algorithm is applicable to higher dimension spaces; however, the existence of a linear order for borders of 2D objects makes the implementation of this algorithm very efficient for the 2D case, as it can be handled by means of cut-and-paste string operations. Also, the above algorithm identifies all borders in objects. An object with more than one border has hollow regions in it. The border representation is, thus, topologically complete. Although in 2D it exhibits no significant topological advantage over, say, a stick diagram, in 3D or higher it permits to extract features such as the existence of knots or holes; besides, it also approximates geometric information such as volumes and surfaces.

6 Dealing with Time

As time elapses and the field changes, floaters adjust their positions, and sometimes they split or die, to maintain an accurate reflection of the field structure. Change is slow, however, and most changes are floater displacements, which carry with them edges and triangles. Correspondence through time is almost always straightforward: immediate for floaters, since their identities are persistent, and recursively constructible for edges, triangles and triangle groups. For instance, for line segments, we proceed as follows: for time steps t and $t+1$ and floaters p and q, we consider the following cases:

- if at t a line segment s joined p and q and at $t+1$ a segment s' joins p and q, we identify s and s';
- if at t a line segment s joined p and q and at $t+1$ it happens that p and q are not adjacent, we say that segment s died at time $t+1$;
- if at t it is the case that p and q are not adjacent and at $t+1$ a segment s' joins p and q, we say that segment s' was born at time $t+1$.

We can identify triangles in a similar fashion. However, we should relax the concept of identity for polygons, to allow for minor alterations. We say that two polygons are identified if the identity of most of their components (the triangles or the floaters inside of the polygon) has been established.

To generate a history of objects as defined by the above rules, we need an efficient method to record change, and to keep track of elements and their life extensions. We achieve this through the implementation of several ideas:

1. Instead of recomputing the triangulation at every time step, we only update it wherever floaters have moved beyond a certain threshold. To do this, we simply remove the floater from the triangulation and insert it back in its new position. This is attained through the implementation of incremental methods for insertion and deletion, inspired on De Floriani's algorithm [2].
2. The overall history of triangulation elements is initialized by registering all elements during the first time step; from that point on, on every update step, after a compressed update list has been generated, it is run through to annotate changes: elements that are destroyed are annotated as dead in the overall list, while those that are created are inserted as new; in each case, a death or birth time stamp are recorded.
3. The final result is a comprehensive compendium of transformation events, which can be indexed either spatially (by element location) or temporally (by birth or death date).

The above ideas work well for edges and triangles, but they must be somewhat extended for aggregated polygons: we want to preserve identities even when the polygons suffer structural change, namely the loss

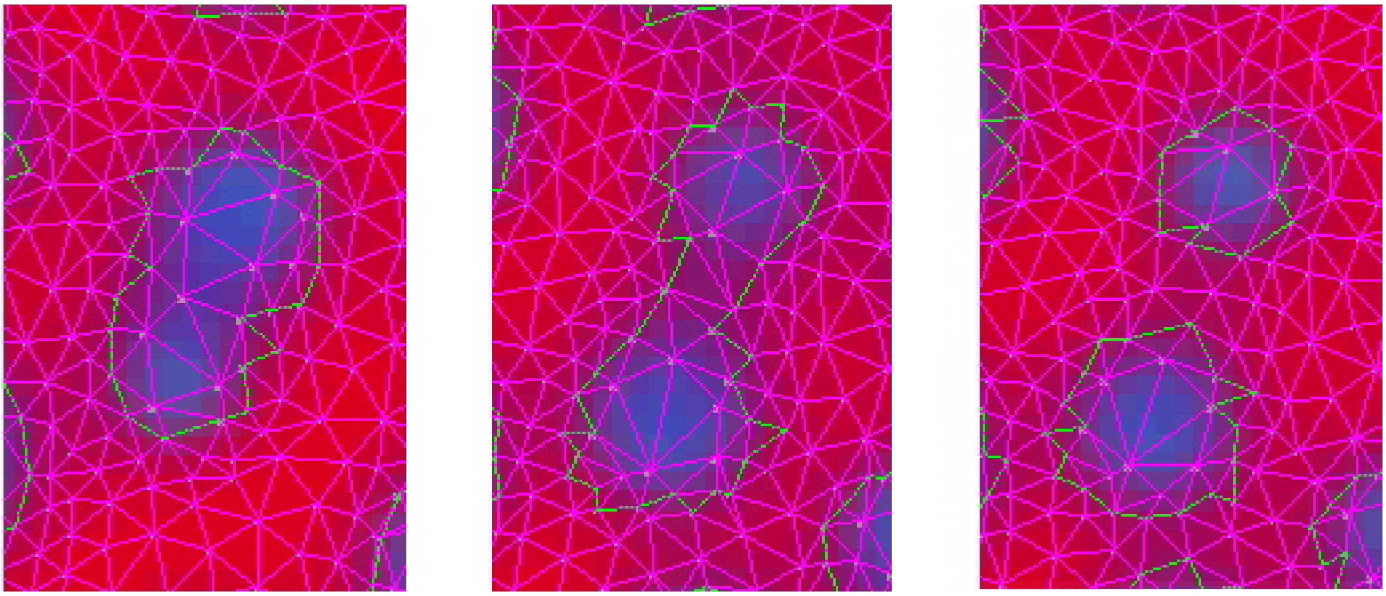

Figure 6: A fission transformation. Stills are approximately 100 generations apart.

or gain of a few triangles or border edges. This only requires to elaborate on the comparison algorithm used to identify entries in the global history. For edges, we looked for the identity of the two endpoints, and for triangles, we required identity of the three vertices. In contrast, two polygons will be said to be identified if it is possible to establish identification among the majority of their elements.

However, there are more events that need to be recorded in a history of aggregated objects:

- Changes in shape: objects may suffer elongations or other distortions in shape, without necessarily suffering any significant structural change. Events of this kind must be noted by examining alterations in geometric relationships among border elements. To do this, we use spatial aggregation on the border elements, and later we examine changes in neighborhood relations.
- Changes in topology: objects may gain or lose holes. These events are considerably simpler to notice in 2D than shape changes, since they involve clear, discrete changes in structure. In higher dimensions, however, there are considerably more cases in need of examination.
- Catastrophic changes in structure: objects may collide with other objects, or split into several smaller objects. Such events would be annotated in the history as death events followed by birth events; however, a subtler record can also be established by examining what happens to the components of the dying object: if most of them are included in a new object then we infer collision, and if most of them are separated into disjoint sets, we infer fission.
- Catastrophes due to noise: if an object has existed for a long interval for t to $t+k$ and is born again at time $t+k+2$, to continue existing for long, our above scheme would actually consider it to be two distinct objects. Hence we extend identity to include objects that disappear for short times.

In Figure 6 we exhibit a sequence that represents a fission transformation. To formalize the above ideas, we need to introduce a few definitions, where α is a real such that $0 < \alpha < 1$.

Definition 6.1 *Given finite sets S and U such that $U \subset S$, we say that U is an α-majority of S iff $|U| > \alpha|S|$.*

Definition 6.2 *Given two finite sets S_1 and S_2, we say that S_1 contains an α-majority of S_2 iff there is a subset U of S_1 such that U is an α-majority of S_2.*

Definition 6.3 *Two finite sets S_1 and S_2 are said to be α-identifiable iff S_1 contains an α-majority of S_2 and S_2 contains an α-majority of S_1.*

Definition 6.4 *A set of sets $\{S_1, S_2, \cdots, S_n\}$ is said to be an α-decomposition of a finite set S iff S and $\bigcup_{i=1}^{n} S_i$ are α-identifiable.*

These definitions allow us to establish various temporal correspondences among polygons given the sets of their embedded triangles or floaters: we fix a parameter α, and we say that two polygons P_1 and P_2 consecutive in time are identifiable if their triangle or floater sets are α-identifiable. Similarly, if a polygon P is followed by several polygons whose triangles or floaters constitute an α-decomposition of P, we say that a fission has occurred, unless one of the new polygons is α-identifiable with P, in which case we say that the other polygons have separated from P. If the above process is reversed in time, we call it a fusion or an assimilation, respectively.

Notice that to compute temporal correspondences between polygons is efficient, in the sense that we do not have to check every pair of polygons across time steps; the persistence of floaters and the use of maps from floaters to the objects that contain them allow us to determine which polygons to compare in almost constant time (or rather, the time it takes to query each map).

The above methodology provides us with a simple way to track topological change; the next step will be to combine this kind of tracking with geometric change, which will give a complete picture of all that is qualitatively observable about a time-varying field that exhibits patterns.

7 Computational Experiments

The experiments were implemented in C++, using a C++ implementation of the Spatial Aggregation Language (SAL) [1]. They were run on 150 MHz and 210 MHz PowerMac computers with the 604 and 604e PowerPC processors, respectively; runs using approximately 1600 floaters and a field grid of 100×100 had approximate speeds of 6 and 3 generations per second, respectively. We now provide the actual methods and parameters used in our experimental settings.

7.1 The Diffusion-Reaction System Simulation

The Gray-Scott Equations were simulated using Euler's integration method with a variable time step Δt that was made dependent on the error ϵ: if the maximum absolute deviation of the field after one iteration with respect to the field carried forward two successive iterations at half the time step was larger than ϵ, then the time step was split in half, until the deviation was under ϵ. Conversely, if the deviation was exceedingly small (less than 0.1ϵ) then Δt was allowed to grow by 10%. The value of ϵ used was 0.01.

The field was simulated using arrays from 100×100 to 256×256, mapping 0.01 spatial units per array element. The field was initialized with all values $u = 1, v = 0$ except for a center square (whose sides were a tenth of the total side length), where values were $u = 0.3, v = 0.5$. Random noise was added to break symmetries. A typical set of parameter values used for the equation is $D_u = 2E-5$, $D_v = 1E-5$, $F = 0.03$, $k = 0.06$.

7.2 Floaters

Floaters are initially placed on a hexagonal grid. Although in the long term the initial number of floaters is unimportant, given that they reproduce until achieving optimal density, we typically started with approximately 1000 floaters. Parameter σ also adapts, depending on the density, but we found that a good initial value was 0.01. The global change rate β for floaters was set to 0.2, and the adaptation rate for σ was $\rho = 0.01$, where a simple linear feedback loop is used to control σ. The optimal energy value was placed at 0.08; this value determines the floater density, since floaters that are closer to each other exhibit higher energies. Floaters are only considered fit to be split or destroyed when they are adequately energized (defined as having at least 30% of the optimal energy) and in equilibrium, where equilibrium is defined as having a velocity smaller than $(\gamma/\sigma)^2$ where the equilibrium factor $\gamma = 0.01$. Of these floaters, those whose σ is under a minimum radius (0.001) are destroyed, and those above a maximum radius (0.045) are split into two, giving each of the new two a σ which is 71% of the original.

7.3 Triangles

A Delaunay triangulation is constructed using the floaters as vertices. The triangulation is updated as needed when the floaters move. To aggregate substructures, the average field value inside each triangle is computed; to do this, a weighted sum of all pixel values inside the triangle is divided by the area of the triangle; the weights are 1 for pixels that are completely inside the triangle, and less than 1 for those traversed by the sides of the triangle; a square clipping algorithm is used to determine those weights. Triangles whose average field value is at most 0.58 are considered to be blue, and all others are considered red; this value was chosen because it is a middlepoint between equilibrium states for the field.

7.4 Temporal Identification

Identification across time was carried out using a value of $\alpha = 0.5$, which corresponds to the usual concept of majority.

A typical output of the history-generation algorithm follows. Birth and death events, as well as fissions, separations, assimilations and fusions are registered as events. In the example, we start with a single body, labeled 0, which generates several other bodies. We start from generation 1510.

```
On generation 1526, body 19 separated from 0 (born 107).
On generation 1527, body 20 separated from body 0 (born 107).
On generation 1528, body 20 was assimilated by body 0 (born 107).
On generation 1531, body 21 separated from body 19 (born 1526).
On generation 1531, body 22 separated from body 0 (born 107).
On generation 1532, body 23 separated from body 19 (born 1526).
On generation 1533, bodies 27, 28, 29, 30 resulted from
the fission of body 0 (born 107).
```

Figures 7, 8 and 9 illustrate the events recorded by the above history fragment.

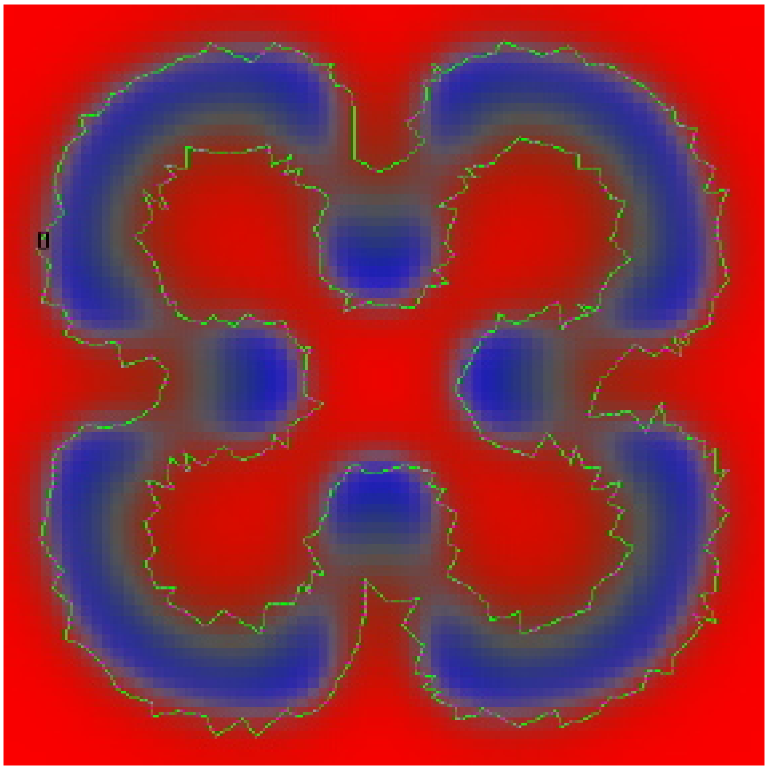

Figure 7: On generation 1510, body 0 is about to suffer several transformations.

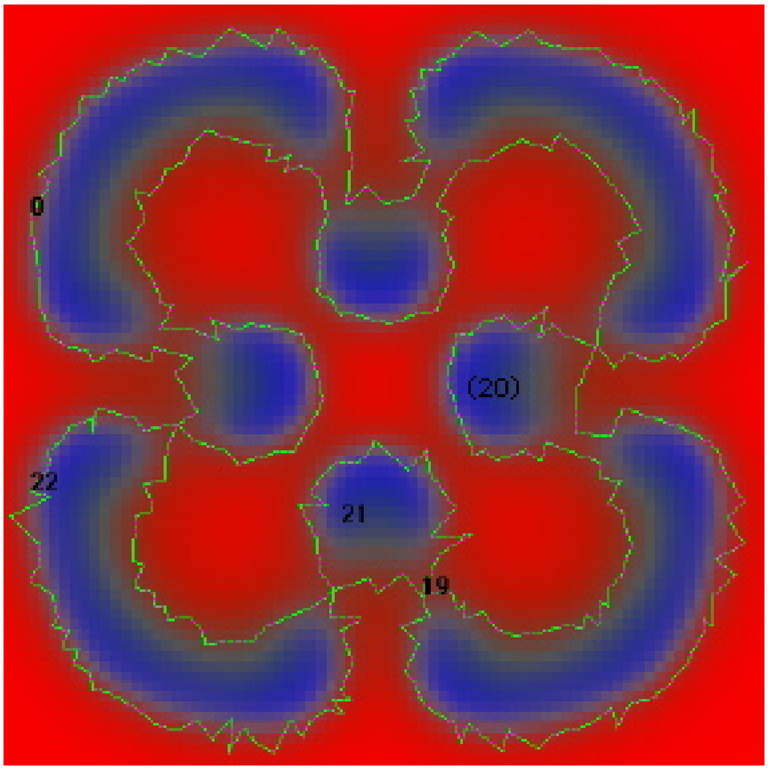

Figure 8: From generation 1526 to 1531, bodies 19, 20 and 22 separate from 0, and body 21 separates from 19. Body 20 is quickly assimilated back by body 0. The picture was taken after the assimilation, so the label for body 20 was added manually, between parentheses

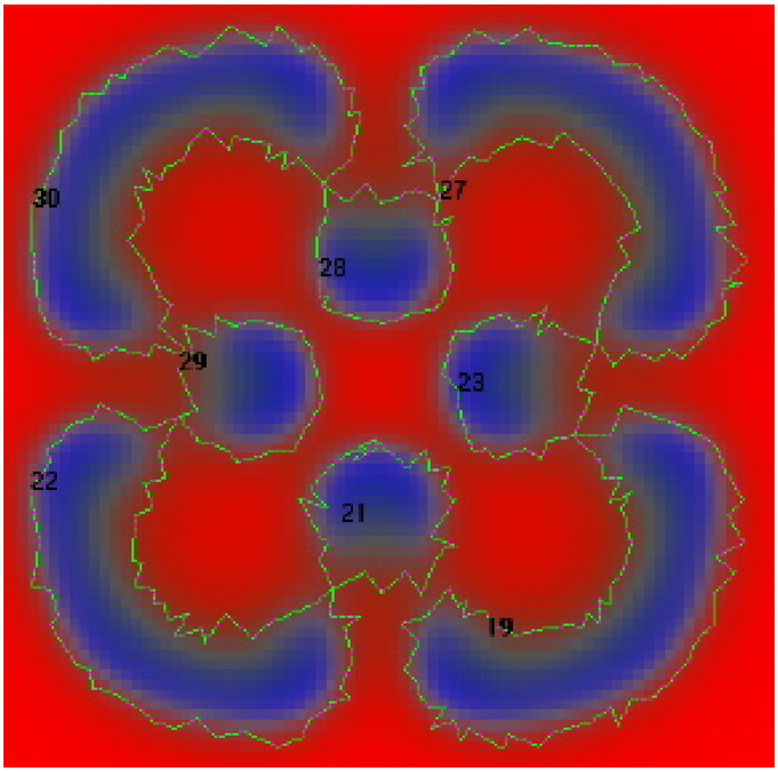

Figure 9: From generation 1532 to 1533, body 23 separates from 19, and then body 0 fissions abruptly into four different bodies.

8 Future Work

We plan to develop methods to efficiently aggregate temporal structures extracted from histories. This idea is based on the fact that time-varying patterns consistently exhibit coherent behavior: for example, certain Diffusion-Reaction patterns in the Gray-Scott system consist of more or less stationary objects that grow until reaching a certain size, and then fission. Temporal aggregation would be able to extract such a higher-level interpretation by noticing the existence of similar chains of events throughout the field's history.

9 Summary

We have presented an algorithm for adaptively sampling spatial fields and extracting iso-surfaces in the fields. The algorithm subdivides the field into triangular regions according to the sampling particle system. Triangles in the subdivision are aggregated to form regions of uniformity. The sampling is adaptive because the density of particles is a function of the magnitude of field gradient. The subdivision is updated as the field varies over time, and temporal correspondence is implied because of the persistence of the particle system.

References

[1] Christopher Bailey-Kellogg, Feng Zhao, and Kenneth Yip. Spatial aggregation: language and applications. In *Proceedings of AAAI*, 1996.

[2] Leila De Floriani. Surface representations based on triangular grids. *The Visual Computer*, 3:27–50, 1987.

[3] Peter Grindrod. *The Theory and Applications of Reaction-Diffusion Equations*. Oxford Applied Mathematics and Computing Science. Oxford University Press, Inc., Oxford University Press. Walton Street, Oxford OX2 6DP, Great Britain, second edition, 1996.

[4] D. Haim, G. Li, Q. Ouyang, William D. McCormick, Harry L. Swinney, A. Hagberg, and E. Meron. Breathing spots in reaction-diffusion systems. *Physical Review Letters*, 77(190), 1996.

[5] Kyoung-Jin Lee, William D. McCormick, John E. Pearson, and Harry L. Swinney. Experimental observation of self-replicating spots in a reaction-diffusion system. *Nature*, 369(6477):215–218, 1994.

[6] W. E. Lorensen and H. E. Cline. A high resolution 3d surface construction algorithm. *Computer Graphics*, 21(4), 1987.

[7] John Pearson, Gary Doolen, Brosl Hasslacher, Bill Reynolds, Werner Horsthemke, Kyoung Lee, Bill McCormick, Harry Swinney, and Silvina Ponce-Dawson. Chemical pattern formation. *Parallel Computing Research*, 2(3), July 1994.

[8] John E. Pearson. Complex patterns in a simple system. *Science*, 261, July 1993.

[9] Alan Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society (B)*, 237:37–72, 1952.

[10] Andrew P. Witkin and Paul S. Heckbert. Using particles to sample and control implicit surfaces. In *Computer Graphics, SIGGRAPH '94 Proceedings*, 1994.

[11] Kenneth Yip. Structural inferences from massive datasets. In *Proceedings of IJCAI*, 1997.

[12] Kenneth Yip and Feng Zhao. Spatial aggregation: Theory and applications. *Journal of Artificial Intelligence Research*, 5, 1996.