Chapter 1

PATTERN GENERATION USING LEVEL SET BASED CURVE EVOLUTION

Amit Chattopadhyay[1] and Dipti Prasad Mukherjee[2]
[1]*RuG, University of Groningen, Groningen, The Netherlands*
[2]*Indian Statistical Institute, Kolkata, India*
Email:{amit@cs.rug.nl, dipti@isical.ac.in}

Patterns are being generated in nature through biological and chemical processes. In this paper we are proposing artificial pattern generation technique using curve evolution model. Given a closed curve in 2D space the curve is deformed under a set of constraints derived from biological and physical pattern generation models. In the proposed approach the reaction-diffusion and shape optimization models are used to derive the constraints for curve evolution. The evolution of curve is implemented in level set framework as the level set based curve evolution supports change in topology of the closed contour. The proposed model is used to generate number of patterns and also successfully tested for reconstructing partially occluded patterns.

1.1. Introduction

Patterns generated in nature often enchant us. However reproduction of such patterns for realistic rendering of a physical object or for animation is a major research challenge in image processing and computer graphics. Natural patterns are so diverse that it is impossible to describe and generate them in a single mathematical framework. This motivates researchers to propose different pattern generation models. There are many pattern generation models in literature.[4,7,11] In this work we utilize reaction-diffusion model of Turing[11] and shape optimization model typically used for analyzing strength of materials.[3]

The reaction-diffusion model, proposed by Turing[11] and which is based on reaction and diffusion of chemicals, can be used to explain biological patterns, for example the spots of Cheetah or Leopard, patterns on the skin of Giraffe etc. Meinherdt[4] has extended Turing's reaction-diffusion model to generate patterns like stripes of Zebra. Recently, Murray has used netlike structure generation model,[7] which can as well be used for pattern generation. In a related context reaction-diffusion model is also extended for fingerprint and natural texture generation and for solving pattern disocclusion (when part of the pattern is missing) problem.[1]

The motivation behind our work is to design an alternative model of pattern

generation using level set framework.[8] Level set based curve evolution is a well-researched topic and has wide application ranging from image restoration to image segmentation, tracking etc.[2,5,8] Since a topology adaptive closed curve can be evolved in the level set paradigm, and can be converged to a desired shape depending on the constraint to the curve evolution, level set based curve evolution is adapted in this paper as the framework for pattern generation. In this approach we have used constraints from reaction-diffusion model to evolve the level set function for curve evolution. Similar to reaction-diffusion model, shape optimization technique can also be used to drive the evolving curve or level set function for pattern generation. Optimization of shape (that is the distribution of the material density within the shape) under different physical conditions, for example, a rectangular piece of material subjected to a pre-designed stress, also generates patterns.[3] The boundary of the shape expressed in level set function is deformed to generate a particular pattern. The contribution of this paper is in demonstrating the use of level set paradigm in generating patterns utilizing both these biological (reaction-diffusion) and physical (shape optimization) models.

In Section 1.2, we briefly review the level set model of curve evolution followed by the description of reaction-diffusion and shape optimization models that eventually drives the curve for pattern generation. The proposed level set based curve evolution scheme for pattern generation is described in Section 1.3. The results and applications related to pattern disocclusion are presented in Section 1.4 followed by conclusions.

1.2. Background

The understanding of level set based curve evolution is the prerequisite for understanding our proposed pattern generation model. Level set based curve evolution is briefly introduced in the next section. As explained in the last section the constraints for curve evolution come from the traditional model of biological pattern generation using reaction-diffusion and shape optimization schemes. These topics are introduced in Sections 1.2.2 and 1.2.3 respectively.

1.2.1. *Level set model of curve evolution*

A closed curve $c(s)$ embedded in a 2D image matrix $I \subset \mathrm{Z}^2$, can be evolved with respect to time t along any direction vector decomposed into normal and tangential components. However, since curve evolution along tangential component is essentially a re-parameterization of the curve[8] and since we are interested only in the deformation of curve shape and not in parameterization of the curve, the equation of curve evolution (with respect to time t) can be expressed as,

$$.\frac{\partial \vec{c}(s)}{\partial t} \approx \beta\, \vec{\mathrm{N}}(t), \tag{1.1}$$

where β is the speed of deformation of $\vec{c}(s)$ along $\vec{\mathrm{N}}(t)$, the normal to the curve $\vec{c}(s)$. The problem of generating a pattern can be posed as detecting the position of $\vec{c}(s)$ at specific time steps when $\vec{c}(s)$ is continuously being deformed along $\vec{\mathrm{N}}(t)$. In the proposed approach, reaction-diffusion and shape optimization based models supply the requisite constraint to monitor β. The initial curve is specified by $\vec{c}(s)$ at $t = 0$ and the iterative evolution of the curve terminates when $\vec{c}(s)$ evolves into a desired pattern. We now define the curve evolution in level set domain.

It is a common practice to define the level set function φ as the signed distance function[8,9] such that $\varphi(x, y) > 0$ if (x, y) is outside $c(s)$, $\varphi(x, y) < 0$ if (x, y) is inside $c(s)$ and $\varphi(x, y) = 0$ if (x, y) is on $c(s)$. The element of the image matrix I having m and n numbers of rows and columns respectively, is (x, y), $0 \leq x < \mathrm{m}$, $0 \leq y < \mathrm{n}$. Therefore, by definition $c(s)$ is embedded in the *zero level set* of φ at any time instant t;

$$.\varphi(c(s), t) = 0. \tag{1.2}$$

The zero level set is the intersection of the level set function (assuming the signed distance values of φ are plotted along z-axis) and the plane at $z = 0$. Differentiating 1.2 with respect to t and using 1.1, the evolution of signed distance function φ is given by[8,9]:

$$.\frac{\partial \varphi}{\partial t} = -\beta \vec{N} \nabla \phi = -\beta \|\nabla \phi\| \ . \tag{1.3}$$

The equivalent numerical approximation is given by $\varphi_{ij}^{n+1} = \varphi_{ij}^{n} - \Delta t \beta \left|\nabla \varphi_{ij}^{n}\right| = 0$ where φ_{ij}^{n} and φ_{ij}^{n+1} are level set functions at (i, j) location at iteration n and $(n+1)$ respectively and Δt is the time step. Since, β is the speed of deformation of $c(s)$ along $\vec{\mathrm{N}}(t)$ and $c(s)$ is embedded in φ, we design β to deform φ and the modified shape of $c(s)$ is obtained from the zero level set of the deformed φ. In context of pattern generation, β controls the deformation of φ such that after certain time $c(s)$ takes the shape of a desired pattern. So the art of pattern generation using level set method is the art of constructing suitable velocity field, which evolves the level set function to give a particular pattern. Throughout this paper our objective is to design β based on the reaction-diffusion and shape optimization based pattern generating process. In the next section we introduce reaction-diffusion model.

1.2.2. *Reaction-diffusion model*

Observing patterns generated through biological process, for example, patterns of Zebra, Jaguar, Leopards etc, Alan Turing is the first to articulate an explanation of how these patterns are generated in nature.[11] Turing observed that patterns could arise as a result of instabilities in the diffusion of morphogenetic chemicals in the animals' skins during the embryonic stage of development. The basic form of a simple reaction-diffusion system is to have two chemicals (call them a and b)

that diffuse through the embryo at different rates and then react with each other to either build up or break down the chemicals a and b. Following are the equations showing the general form of a two chemical reaction-diffusion system in 1D.[11]

$$.\frac{\partial a}{\partial t} = F\left(a, b\right) + D_a \nabla^2 a, and \tag{1.4}$$

$$.\frac{\partial b}{\partial t} = G\left(a, b\right) + D_b \nabla^2 b. \tag{1.5}$$

The equation 1.4 conveys that the change of concentration of a at a given time depends on the sum of the local concentrations of a and b, $F(a, b)$ and the diffusion of a from places nearby. The constant D_a defines how fast a is diffusing, and the Laplacian $\nabla^2 a$ is a measure of how high the concentration of a is at one location with respect to the concentration of a nearby in a local spatial neighbourhood. If nearby places have a higher concentration of a, then $\nabla^2 a$ is positive and a diffuses towards the center position of the local region. If nearby places have lower concentrations, then $\nabla^2 a$ is negative and a diffuses away from the center of the local region. The same analogy holds for the chemical b as given in 1.5.

The key to pattern formation based on reaction-diffusion is that an initial small amount of variation in the concentrations of chemicals can cause the system to be unstable initially and then to be driven to a stable state in which the concentrations of a and b vary across a boundary. A typical numerical implementation of **??**due to[12] is given as:

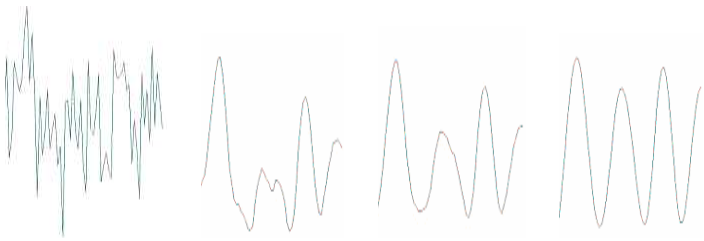
$$.\Delta a_i = s(16 - a_i b_i) + D_a(a_{i+1} + a_{i-1} - 2a_i), and \tag{1.6}$$

$$.\Delta b_i = s(a_i b_i - b_i - \xi_i) + D_b(b_{i+1} + b_{i-1} - 2b_i). \tag{1.7}$$

In an array of cells the concentration of chemical a (b) in i, (i+1) and (i-1) locations are given by a_i (b_i), a_{i+1} (b_{i+1}) and a_{i-1} (b_{i-1}) respectively. The value of ξ_i is the source of slight irregularities in chemical concentrations at ith location. Fig. 1.1 illustrates the progress of concentration of chemical b across an array of 60 cells as its concentration varies over time. Initially the values of a_i and b_i are set to 4 for all the cells in the array. The value of ξ_i is perturbed around 12±0.05. The diffusion constants are set to $D_a = 0.25$ and $D_b = 0.0625$, which means a diffuses more rapidly than b and we take reaction constant s as 0.03125.

The numerical scheme of 1.6 and 1.7 can easily be extended for 2D grid where a matrix of cells are defined in terms of 4 or 8 neighbourhood connectivity. The two-chemical model of Turing is extended to five chemical systems by Meinherdt[4] for periodic stripe generation.

As discussed in the introduction, attractive pattern can also be generated through shape optimization when the shape is subjected to certain physical constraints. We present this concept in the next section.



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

Fig. 1.1. 1D example of reaction-diffusion. (a): Initial concentration of chemical b. (b) – (d): Concentrations of b after every 4000 iterations.

1.2.3. *Shape optimization*

The problem of shape optimization is often referred as structural shape optimization where an optimized structure is obtained as the original shape is subjected to certain pre-defined load. Through shape optimization process, the mass of the shape is redistributed within the shape boundary (also referred as design domain) optimally to counter the effect of load and support to the shape. This optimal mass distribution is what we perceive as a pattern. In one sense it is a user defined pattern as the extent and position of load and support to the shape or structure is user selectable. Consider a design domain or a shape as shown in Fig. 1.2. Under a given load and support, the mass of the shape is redistributed as shown in Fig. 1.3. The optimized shape boundary is always constrained within the initial design domain. The pattern generated in Fig. 1.3 is what interests us and we show in subsequent section that it is possible to pose this problem as curve evolution problem.

It is a standard practice to assume that the shape under consideration is a collection of finite elements to find stresses and displacements of individual elements and consequently the entire shape.[13] Utilizing the displacement information of individual element, the method of moving asymptotes (MMA) finds the optimal mass distribution within the design domain.

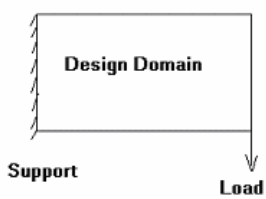


Fig. 1.2. Design domain with support and load.

Considering the top left corner of the design domain as origin and the displacements of ith element u_i due to load F_i at the ith element of the design domain, the work done or compliance C is expressed as force *times* displacement $C = F^T U$ after

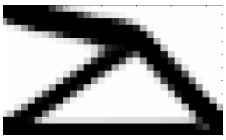


Fig. 1.3. Optimized shape as the desired pattern.

arranging displacement and forces of all elements in vectors U and F respectively. Given K as the global stiffness matrix of the discretized design domain $F = KU$, compliance can be written as, $C = F^T U = U^T KU$.

Considering that the design domain consists of N number of unit elements each having material densities x, the mass m of the shape is given by,

$$.m = x_1 v_1 + x_2 v_2 + \ldots + x_N v_N, \tag{1.8}$$

where v is the element volume. The distribution of this material density x due to different load and support arrangements to the design domain is what we perceive as desired pattern. Therefore, the objective of this derivation is to find the solution of x under different load and support conditions.

The total stiffness of the shape using finite element method is given by,

$$.K = x_1 k_0 + x_2 k_0 + \ldots + x_N k_0, \tag{1.9}$$

where k_0 is the stiffness of individual element. The above model suggests that the stiffness of each element is proportional to the density of the material the element contains. The effect of material density on the stiffness value can be penalized by introducing penalization power p as:

$$.K = x_1^p k_0 + x_2^p k_0 + \ldots + x_N^p k_0. \tag{1.10}$$

The objective of the topology optimization problem is to minimize the compliance $\min_x : C(x) = U^T KU = \sum_{i=1}^{N} (x_i)^p u_i^T k_0 u_i$ subject to constraints $(V(x)/V_0) = f$, $F = KU$, $0 < x_{\min} \leq x \leq 1$. The index i stands for the ith element and x_{min} is a vector of minimum material densities (non-zero to avoid singularity). The predefined volume fraction f is defined as the ratio of volume $V(x)$ at a given instant (that is, $V(x)$ is the volume at a particular material density x which is changing with time) and the initial volume V_0. This optimization problem can be solved using different approaches such as optimality criteria (OC) method or using the Method of Moving Asymptotes (MMA).[3] Following[3] the heuristic updating scheme for the design variable can be formulated as

$$.x_i^{new} = \begin{cases} \max(x_{\min}, x_i - \theta) & \text{if } x_i B_i^\eta \leq \max(x_{\min}, x_i - \theta) \\ x_i B_i^\eta & \text{if } \max(x_{\min}, x_i - \theta) < x_i B_i^\eta < \min(1, x_i + \theta) \\ \min(1, x_i + \theta) & \text{if } \min(1, x_i + \theta) \leq x_i B_i^\eta \end{cases} \tag{1.11}$$

where x_i^{new} is the updated design vector, θ is a positive constant which is the limit of change of the design vector. The parameter η is a numerical damping coefficient and B_i is found from the optimality condition $B_i = (-(\partial C/\partial x_i)/\lambda(\partial V/\partial x_i))$

where λ is a Lagrangian multiplier evaluated from well-known bi-sectioning algorithm. The element sensitivity (i.e. the change in compliance with respect to the change in design variable) of the objective function is found as $(\partial C/\partial x_i) = -p\,(x_i)^{p-1}\,u_i^T\,k_0\,u_i$. In order to ensure existence of solutions to the topology optimization problem some restrictions on the resulting design are usually introduced.[10]

For Fig. 1.2, the design space is discretized into 32x20 elements whose left side is fixed (as support) and unit force is applied at the position (30,20). The initial volume fraction and penalization power are taken as 0.5 and 3 respectively. The optimized shape following 1.11 is shown in Fig. 1.3.

From the pattern generation point of view we investigate how patterns as in Fig. 1.3 can be generated using level set curve evolution method. As discussed earlier, the reaction diffusion based approach or optimized shape boundary technique should be implemented to guide the evolution of level set function. This is explained next.

1.3. Proposed Methodology

So far we have investigated reaction-diffusion and shape optimization based pattern generation. The point is whether these techniques can be unified in the level set framework. Alternately, the challenge is to develop β of 1.3 which is to be motivated from either reaction-diffusion approach or shape optimization approach. This is taken up next.

1.3.1. *Reaction-diffusion influenced curve evolution*

In reaction-diffusion system, two unstable chemicals have different levels of density distribution. A stable pattern is formed when two chemicals and the interface between them describe stable configurations. For implementation using level set function, the interface between the chemicals at stable state should be given by the zero level set. The proposed model should evolve the level set function with a velocity such that the chemicals or the resulting interface between the chemicals goes to a stable state. One of the preconditions to generate stable state is that the energy corresponding to the system should be minimum. The energy term corresponding to a reaction-diffusion system of two chemicals with densities a and b can be expressed as,[7]

$$.E(t) = \frac{1}{2}\int_{\Pi} \|\nabla w\|^2\, dx, \tag{1.12}$$

where the norm $\|\nabla w\|^2 = |\nabla a|^2 + |\nabla b|^2$ and Π is the domain of reference. The variable w can be visualized as the surface of average concentrations of the chemicals a and b put together. The domain of reference represents the surface plane of chemicals at $z = 0$ where the chemical concentrations are being varied as the deformation of zero level set.

The initial boundary condition is given as $(n.\nabla)w = 0$ on $\partial\Pi$. The normal unit vector n is defined on the boundary $\partial\Pi$ of reference domain Π. The initial boundary condition $(n.\nabla)w = 0$ implies that the rate of variation of concentration w along the normal to the boundary $\partial\Pi$ of reference domain is zero. The initial condition $w(x,0) = w_0(x)$ on Π gives the concentration of the chemical w in Π at time t=0. To find the gradient descent direction so that energy defined in 1.12 is minimized, we get

$$.\frac{\partial E}{\partial t} = \frac{1}{2}\frac{\partial}{\partial t}\int_{\Pi}(|\nabla a|^2 + |\nabla b|^2)dx. \tag{1.13}$$

From the derivation in Appendix I and applying the boundary conditions, (11) can be simplified as

$$.\frac{\partial E}{\partial t} = \int_{\Pi}\langle -div(\nabla a),\ a_t\rangle\ dx + \int_{\Pi}\langle -div(\nabla b),\ b_t\rangle\ dx. \tag{1.14}$$

Using Cauchy-Schwartz inequality the field for which $E(t)$ decreases most rapidly is given by,

$$.\frac{\partial a}{\partial t} = div(\nabla a), \tag{1.15}$$

$$.\frac{\partial b}{\partial t} = div(\nabla b). \tag{1.16}$$

We can take either of the above fields along normal to the boundary of concentration as our required velocity field of level set evolution. So we write the curve evolution based pattern generation algorithm using reaction-diffusion model as follows:

Algorithm 1

Step 1: Initialize the embedding level set function $\varphi(x,y,0)$ at $t = 0$ by the distance function of any closed curve in the domain Π. So $\varphi(x,y,0) = 0$ on $\partial\Pi$, $\varphi(x,y,0) > 0$ inside $\partial\Pi$ and $\varphi(x,y,0) < 0$ outside $\partial\Pi$.

Step 2: Initiate minor (random) perturbation of a or b in the desired locations of Π.

Step 3: Calculate the speed function $\beta = div(\nabla b)$ following 1.16. This defines the speed of propagation of level set function $\varphi(x,y,t)$. Similarly, speed for chemical a can also be evaluated.

Step 4: Update the level set function $\varphi(x,y,t)$ following 1.3. Stop update $\varphi(x,y,t)$ when we get a stable pattern or there are *insignificant* changes in pattern in two consecutive iterations.

Next we show how shape optimization can be expressed as level set based curve evolution technique.

1.3.2. *Shape optimization based curve evolution*

In this case the challenge is to express the compliance minimization problem of Section 1.2.3 as curve boundary evolution problem. The velocity field for curve evolution is derived utilizing shape derivative technique.[6]

In linear elasticity setting (i.e. stress strain relation of the material is linear), let $\Omega \subset R^2$ be a bounded open set occupied by a linear isotropic elastic material (i.e. elastic properties are independent of the orientation of the axes of coordinates) with elasticity coefficient A. For simplicity we assume that there is no volume force but only surface loadings g. The boundary of Ω is made of three disjoint parts $\partial\Omega \equiv \Gamma \bigcup \Gamma_N \bigcup \Gamma_D$ with Dirichlet boundary conditions on Γ_D and Neumann boundary conditions on $\Gamma \bigcup \Gamma_N$ as shown in Fig. 1.4. The portion of the boundary where load is being applied is Γ_N whereas the portion of the boundary that is fixed is Γ_D. Remaining part of the boundary is Γ, which is allowed to vary in the optimization process.

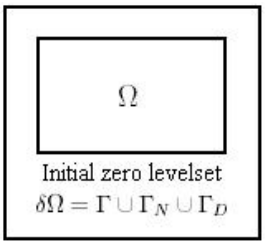


Fig. 1.4. Boundary defined on design domain for shape optimization.

The displacement field u of Ω is the unique solution of the linearized elasticity system $-div\,(A\,e\,(u)) = 0$ in Ω with boundary conditions $u = u_0$ on Γ_D and $(A\,e(u))\,n = g$ on $\Gamma \bigcup \Gamma_N$.[6] The solution involving displacement field u interprets that the variation in stress tensor is zero once the solution is reached. The solution of displacement field u is the desired pattern.

The strain tensor $e(u)$ is given as, $e(u) = 0.5(\nabla u + \nabla^t u)$ with t denotes the transpose operator. $A\,e(u)$ is the stress tensor. The prescribed initial value of u on Γ_D is u_0. The unit normal direction to boundary $\partial\Omega$ is n. The objective function for minimization is denoted by,

$$.J(\Omega) = \int_{\Gamma \bigcup \Gamma_N} gu\,ds = \int_{\Omega} A\,e(u)\,e(u)\,dx. \tag{1.17}$$

To take into account the weight of the structure, we rewrite 1.17 as minimization of $\inf\limits_{\Omega} J(\Omega) + l \int_{\Omega} dx$ where l is the positive Lagrange multiplier l. In general this minimization is well posed only if some geometrical and topological restrictions on the shape are enforced.[2] Using shape derivative method[6] and following the derivation in Appendix II, we find a gradient descent field, which minimizes the

objective function, as

$$.\theta = -v_0 n, \tag{1.18}$$

where $v_0 = 2[\frac{\partial(gu)}{\partial n} + H\,(gu)] - Ae(u)e(u)$ and n is the normal to Ω. Assuming a displacement from reference domain Ω_0 to $\Omega = (\mathrm{I}d + \theta)\,\Omega_0$, θ is the displacement field of Ω_0 and Id is the identity mapping in $W^{1,\infty}\left(R^2, R^2\right)$. To implement numerically, the design domain Ω_t is updated at every iteration with time step $\Delta t > 0$ as,

$$.\Omega_t = (Id + \Delta t\theta)\Omega_{t-1}. \tag{1.19}$$

From 1.18, we observe that the gradient descent field acts in the normal direction of the boundary as stipulated in level set based curve evolution. Controlling the boundary conditions, for example, nature and location of surface loading and support (g, e), various patterns are generated. The corresponding pattern generation algorithm is given as follows:

Algorithm 2

Step 1: Initialize the level set function similar to step 1 of Algorithm 1. Specify loading and support conditions for the shape.

Step 2: The boundary conditions are solved to find the displacement u.

Step 3: Calculate the speed function v_0 of 1.18 that defines the speed of propagation of $\varphi\,(x, y, t)$ and then update the level set function following 1.3.

Step 4: Stop update of $\varphi\,(x, y, t)$ when a stable pattern is obtained. The stability condition is also achieved when there are marginal changes in volume fraction of the shape (that is insignificant change of u) in two consecutive iterations.

In the next section, we show how these methods can be used to generate fascinating patterns.

1.4. Results

We first present the results obtained using Algorithm 1. The spot and stripes patterns after 8000 and 1000 iterations are shown in Figs. 1.5 respectively. For spot pattern, perturbation to the tune of 12 ± 0.5 is given in every alternate coordinate of 80x80 matrix. For the stripe pattern, the same perturbation is given at the centre of the 80x80 matrix.

The application of Algorithm 2 is shown in Fig. 1.6 Several patterns are shown and in each case the input is a rectangular design domain from which the patterns are carved out. The size of the rectangular design domain, the load applied to the shape including its coordinate and the support to the design domain are given in the Table 1.

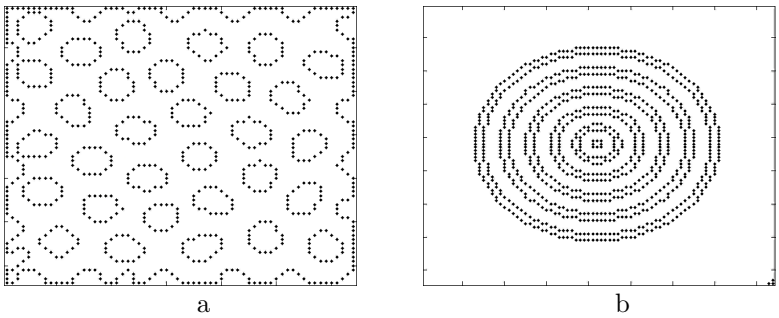
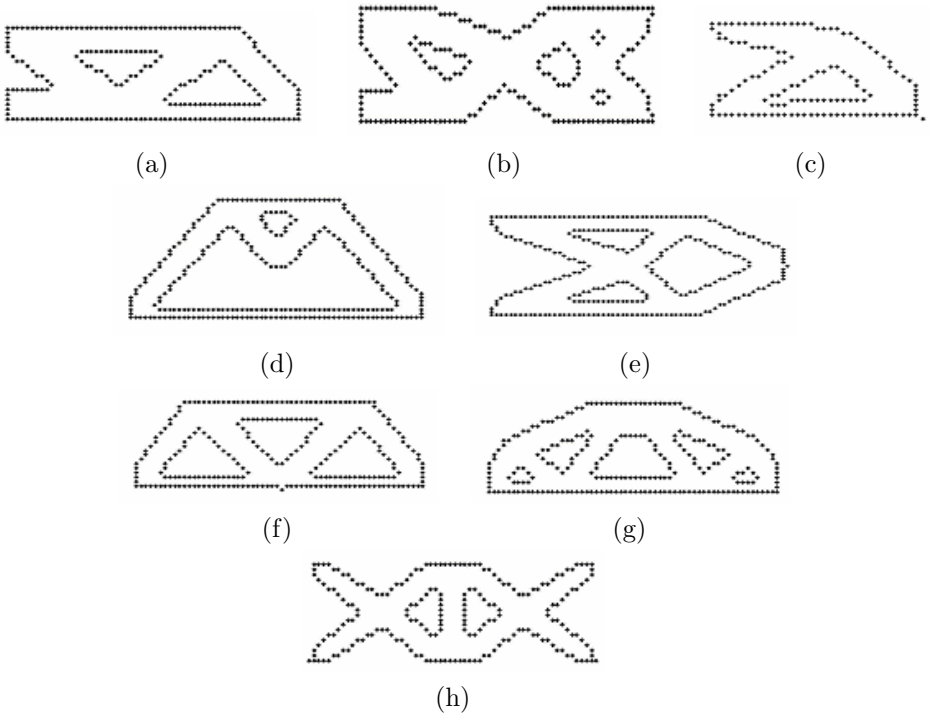


Fig. 1.5. (a)Spot pattern (8000 iterations). (b) Stripe pattern (1000 iterations)



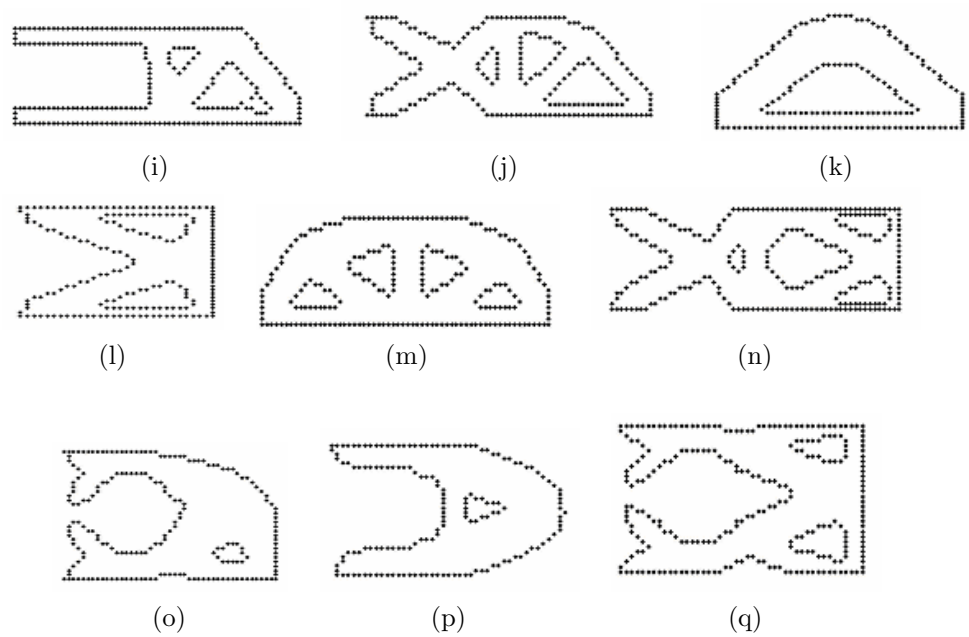


Fig. 1.6. Patterns generated using shape optimization technique (Algorithm 2). The original shape dimension, load distribution on the shape and fixed supports for the shape for these patterns are given in Table 1.1.

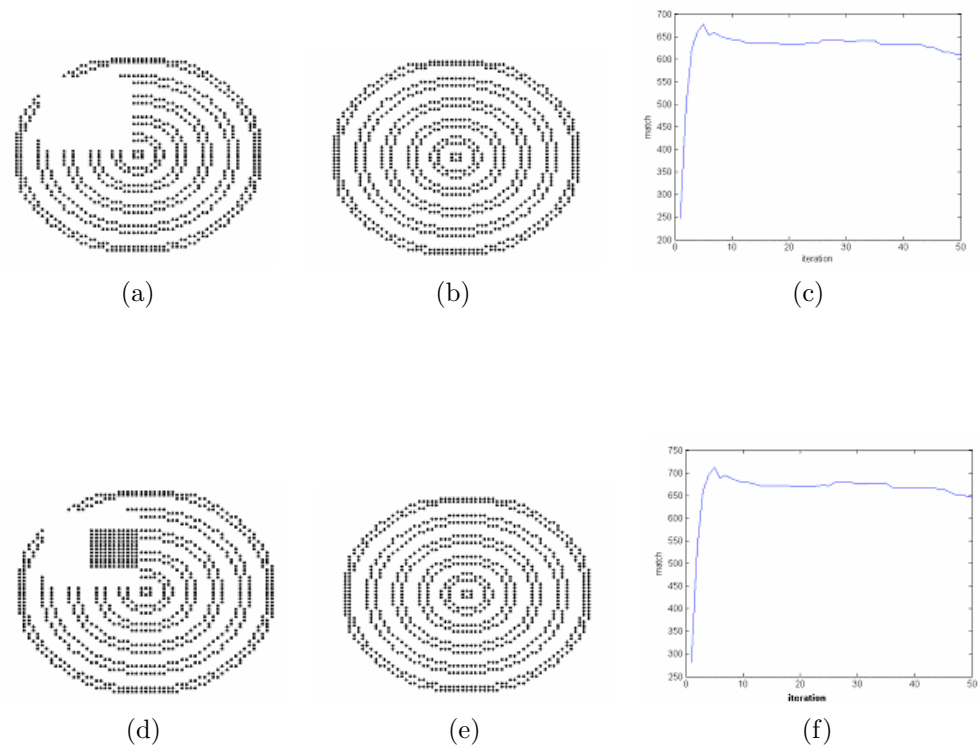
An important use of pattern generation scheme is to regenerate a part of the missing pattern or reconstruct a noise-corrupted pattern given the database of model parameters for pattern generation. This problem is often referred as pattern disocclusion problem as discussed next.

1.4.1. *Pattern disocclusion*

The problem of pattern disocclusion is addressed using a pattern database, which contains different pattern generation models, and the range of parameters required for the respective model (for example, perturbation amount and location, load and support for the models discussed in this paper). Note that there is no need of explicitly storing the patterns in the pattern database. Given a partially occluded pattern where part of the pattern is missing as shown in Fig. 1.7(a), patterns created from the pattern database are point-wise matched to the pattern of Fig. 1.7(a) to calculate the mean square error (MSE). MSE is calculated using point-wise multiplication of occluded pattern and the reconstructed pattern matrices followed by summation of non-zero elements of the product matrix. For the model and model parameters for which the generated pattern gives minimum MSE with respect to the occluded pattern is selected as the model for reconstructed pattern. The reconstructed pattern for Fig. 1.7(a) is shown in Fig. 1.7(b). The MSE plot against

iterations is shown in Fig. 1.7(c). Note that the MSE increases initially and then stabilizes approximately around 630. This stabilized value is minimum compared to all other stabilized MSEs using other reconstructed patterns derived from pattern database.

The same experiment is repeated where the occluded region of the pattern of Fig. 1.7(a) is filled with random dots. This is shown in Fig. 1.7(d). The corresponding reconstructed pattern is the same as that of Fig. 1.7(b) and is shown in Fig. 1.7(e). The MSE plot shown in Fig. 1.7(f) shows that the stabilized MSE is slightly increased, as expected due to noise in the occluded region, at around 645. In both cases the correct pattern could be identified from the occluded and noise-corrupted patterns.

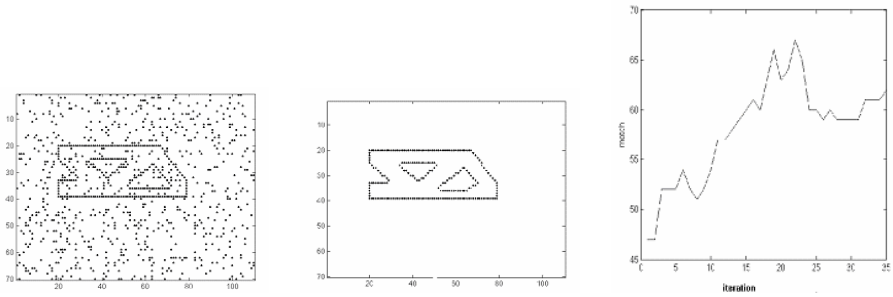


(a) (b) (c)

(d) (e) (f)

Fig. 1.7. Pattern disocclusion using reaction-diffusion model based curve evolution.

This is further extended for patterns developed using shape optimization model. The noise-corrupted pattern of Fig. 1.8(a) is successfully reconstructed as shown in Fig. 1.8(b) where MSE is stabilized at around 60 (Fig. 1.8(c)), which is minimum when the MSE of the noisy pattern is compared with the other reconstructed patterns of Fig. 1.6.



(a) (b) (c)

Fig. 1.8. Pattern disocclusion using shape optimization model.

1.5. Conclusions

In this paper a group of pattern generation methods is established as curve evolution based technique. The curve evolution is achieved through geometric and implicit function based level set method. We have also shown that given a pattern database, pattern disocclusion problem can be solved from the minimum error between occluded and derived pattern. Note that the pattern database can contain model parameters and there is no need to store the pattern itself. The extension of this technique for generating textured images and quasi-periodic patterns like human fingerprint etc. is what we are investigating now. At the same time suitable intensity interpolation scheme can be integrated with curve evolution to generate realistic rendering.

References

1. S. Acton, D.P. Mukherjee, J, Havelichek, A. Bovik, *IEEE Transactions on Image Processing*, 10(6), 2001, p. 885-896.
2. G. Aubert and P. Kornprobst, *Applied Mathematical Sciences Series*, Springer–Verlag, 147, 2002.
3. M. Bendsoe, *Structural Optimization*, 1, 1989, p. 193-202.
4. M. Meinhardt, *Models of Biological Pattern Formation*, Academic Press, NY, 1982.
5. D. P. Mukherjee, N. Ray and S. T. Acton, *IEEE Transactions on Image Processing*, 134, 2004, p. 562-572.
6. F. Murat and S. Simon, *Lecture notes in computer Science*, 41, 1976, p. 54-62.
7. J. Murray, *Mathematical Biology*, Springer-Verlag, Heidelberg, 2002.
8. G. Sapiro, *Geometrical Partial Differential Equations and Image Analysis*, Cambridge University Press, MA, 2001.
9. J. Sethian, *Fluid Mechanics, Computer Vision and Material Sciences*, Cambridge University Press, MA, 1999.
10. O. Sigmund, *Mechanics of structures and machines*, 254, 1997, p. 495-526.
11. A. Turing, *Phil. Trans. of Roy Soc. B*, 237, 1952, p. 37-72.
12. A. Witkin and K. Michael, *ACM SIGGRAPH Computer Graphics*, 4, 25 1991.
13. C. Zeinkiewicz, *Finite Element Method*, 3rd Edition., Tata McGraw Hill, 1979.

Appendix I Given $\|\nabla w\|^2 = |\nabla a|^2 + |\nabla b|^2$, the gradient descent direction for minimizing 1.12 is given by,

$\frac{\partial E}{\partial t} = \frac{1}{2}\frac{\partial}{\partial t}\int_\Pi (|\nabla a|^2 + |\nabla b|^2)dx$
$\Rightarrow \frac{\partial E}{\partial t} = \int_\Pi |\nabla a| \ \frac{\nabla a.\nabla a_t}{|\nabla a|}dx + \int_\Pi |\nabla b| \ \frac{\nabla b.\nabla b_t}{|\nabla b|}dx$
$\Rightarrow \frac{\partial E}{\partial t} = \int_{\partial\Pi} \nabla a.\, a_t\, \vec{n}\, ds - \int_\Pi div(\nabla a)\, a_t dx + \int_{\partial\Pi} \nabla b.\, b_t\, \vec{n}\, ds - \int_{\partial\Pi} div(\nabla b)\, b_t dx$
$\Rightarrow \frac{\partial E}{\partial t} = -\int_\Pi div(\nabla a)\, a_t dx - \int_\Pi div(\nabla b)\, b_t dx$
$\Rightarrow \frac{\partial E}{\partial t} = \int_\Pi \langle -div(\nabla a),\ a_t\rangle\, dx + \int_\Pi \langle -div(\nabla b),\ b_t\rangle\, dx$ (using boundary conditions specified in Section 1.3.1).

Appendix II

For the reference domain Ω_0 consider its variation $\Omega = (Id + \theta)\,\Omega_0$ with $\theta \in W^{1,\infty}(R^2; R^2)$. $W^{1,\infty}(R^2; R^2)$ is the space of all mappings from R^2 to R^2 which are differentiable infinitely many times and Id is the identity mapping in $W^{1,\infty}(R^2; R^2)$. The set $\Omega = (Id + \theta\,)\,\Omega_0$ is defined by $\Omega = \{\, x + \theta(x)\,|\, x \in \Omega_0\}$ where the vector field $\theta(x)$ is the displacement of Ω_0. We consider the following definition of shape derivative as the Frechet derivative.

Definition A: Let T be an operator on a normed space X into another normed space Y. Given $x \in X$, if a linear operator $dT(x) \in \chi[X, Y]$ exists such that $\lim\limits_{\|h\|\to 0} \frac{\|T(x+h)-T(x)-dT(x)h\|}{\|h\|} = 0$ then $dT(x)$ is said to be the Frechet derivative of T at x, and T is said to be Frechet differentiable at x. $\chi[X, Y]$ is the space of bounded linear operators on a normed space X into another normed space Y. The operator $dT : X \to \chi[X, Y]$, which assigns $dT(x)$ to x is called the Frechet derivative of T.

Definition B: The shape derivative of $J(\Omega)$ at Ω_0 is the Frechet derivative in $W^{1,\infty}(R^2; R^2)$ of $\theta \to J((Id + \theta)\Omega_0)$ at 0. Then, $\lim\limits_{\|\theta\|\to 0} \frac{\|J((Id+\theta)(\Omega_0))-J(\Omega_0)-J'(\Omega_0)\|}{\|\theta\|} = 0$. We apply the following results of shape derivative.[6]

Result (B.1): If $J_1(\Omega) = \int_\Omega f(x)\, dx$, then shape derivative of $J_1(\Omega)$ at Ω_0 is given by, $J_1'(\Omega_0)\,(\theta) = \int_{\Omega_0} div\,(\theta(x)\, f(x))dx = \int_{\partial\Omega_0} \theta(x).n(x)\, f(x)ds$, where $n(x)$ is the unit normal vector to $\partial\Omega_0$(boundary of Ω_0) and for any $\theta \in W^{1,\infty}(R^2; R^2)$.

Result (B.2): If $J_2(\Omega) = \int_{\partial\Omega} f(x)\, ds$, then shape derivative of $J_1(\Omega)$ at Ω_0 is given by,

$J_2'(\Omega_0)\,(\theta) = \int_{\partial\Omega_0} \theta(x).n(x)(\frac{\partial f}{\partial n} + H\, f\,)\, ds$, where H is the mean curvature of $\partial\Omega_0$ which is defined by, $H = div\,(n\,(x))$. Applying results (B.1) and (B.2), we get shape derivative of compliance as $J'(\Omega_0)\,(\theta) = \int_\Gamma \theta(x).n(x)(2[\frac{\partial(g.u)}{\partial n} + H\,(g.u)] - Ae(u)e(u))\, ds$, where Γ is the variable part of the boundary of the reference domain Ω_0, $n(x)$ is the normal unit vector to Γ, H is the curvature of Γ and u is the displacement field solution space of $-div\,(A\, e\,(u)) = 0$ in Ω_0. By Cauchy-Schwartz inequality we find a gradient descent field, which minimizes the objective function as, $\theta = -v_0\, n$ and then update the shape as $\Omega_t = (Id + \Delta t\theta)\Omega_{t-1}$ with $\Delta t > 0$ is the time step.

Table 1.1. Shape dimension and details of load and support to generate patterns of Fig. 1.6.

Fig.	Shape dimension	Unit load at nodes (load direction - v-vertical, h-horizontal, u-upwards, d-downwards)	Support nodes along x-directions (x:) and y-directions (y:)
1.6(a)	60x20	(1,1) (v)	x: (1,1) to (1,21) y: (61,21)
1.6(b)	60x20	(1,1) (v)	x: (1,1) to (1,21) y: (61,1) to (61,21)
1.6(c)	32x20	(33,21) (v)	x: (1,1) to (1,21) y: (1,1) to (1,21)
1.6(d)	32x20	(16,10) (v)	x: (1,21) y: (33,21)
1.6(e)	61x31	(62,16) (v)	x: (1,1) to (1,21) y: (1,1) to (1,21)
1.6(f)	60x20	(30,21) (v)	x: (1,21) y: (61,21)
1.6(g)	60x20	(31,11) (v)	x: (1,21) y: (61,21)
1.6(h)	60x20	(31,1) (v, u) (31,21) (v, d)	x: (1,1) to (1,21) y: (1,1) to (1,21) x: (61,1) to (61,21) y: (61,1) to (61,21)
1.6(i)	60x20	(31,1) (v, u) (31,21) (v, d)	x: (1,1) to (1,21) y: (61,21)
1.6(j)	60x20	(31,1) (v, u) (31,21) (v, d)	x: (1,1) to (1,21) y: (1,1) to (1,21) y: (61,21)
1.6(k)	45x45	(23,1) (v, u) (23,46) (v, d)	y: (1,46) y: (46,46)
1.6(l)	30x30	(31,1) (v, u) (31,31) (v, d)	x: (1,1) to (1,31) y: (1,1) to (1,31)
5(m)	60x20	(32,21) (v, u) (28,21) (v, d)	y: (1,21) y: (61,21)
1.6(n)	60x20	(15,1) (v, u) (15,21) (v, d)	x: (1,1) to (1,21) y: (1,1) to (1,21) y: (61,1) to (61,21)
1.6(o)	45x30 with a hole of radius 10, centered at (15,15).	(46,31) (v, u)	x: (1,1) to (1,31) y: (1,1) to (1,31) y: (46,1) y: (46,31)
1.6(p)	45x30 with a hole of radius 10, centered at (15,15).	(46,15) (h)	x: (1,1) to (1,31) y: (1,1) to (1,31)
1.6(q)	45x30 with a hole of radius 10, centered at (15,15).	(46,1) (v, d) (46,31) (v, u)	x: (1,1) to (1,31) y: (1,1) to (1,31)