ABSTRACT

A new method for generating artificial fingerprints is presented. Due to their uniqueness and durability, fingerprints are invaluable tools for identification for law enforcement and other purposes. Large databases of varied, realistic artificial fingerprints are needed to aid in the development and evaluation of automated systems for criminal or biometric identification. Further, an effective method for simulating fingerprints may provide insight into the biological processes underlying print formation. However, previous attempts at simulating prints have been unsatisfactory. We approach the problem of creating artificial prints through a pattern formation model. We demonstrate how it is possible to generate distinctive patterns that strongly resemble real fingerprints via a system of partial differential equations with a suitable domain and initial conditions.
ACKNOWLEDGEMENTS

Thanks to Dr. Dieter Armbruster, Dr. Rodrigo Platte, and Dr. Bruno Welfert, for serving on my thesis committee and for their time and assistance. Thanks in particular to Dr. Armbruster for chairing the committee and whose assistance made this project possible. Thanks to Eric Ray of the Arizona Department of Public Safety for his time and expert advice. Finally, thanks to Arizona State University and the School of Mathematical and Statistical Sciences.
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>2 Background</td>
<td>3</td>
</tr>
<tr>
<td>3 Methods</td>
<td>5</td>
</tr>
<tr>
<td>3.1 Creating a database</td>
<td>10</td>
</tr>
<tr>
<td>4 Pattern stability analysis</td>
<td>11</td>
</tr>
<tr>
<td>4.1 Inhomogeneous solutions</td>
<td>12</td>
</tr>
<tr>
<td>5 Generating fingerprint patterns</td>
<td>18</td>
</tr>
<tr>
<td>5.1 Type-independent parameters</td>
<td>19</td>
</tr>
<tr>
<td>5.2 Formation of level-one patterns</td>
<td>21</td>
</tr>
<tr>
<td>5.3 Generating arches</td>
<td>23</td>
</tr>
<tr>
<td>5.4 Generating whorls</td>
<td>24</td>
</tr>
<tr>
<td>5.5 Generating loops</td>
<td>25</td>
</tr>
<tr>
<td>5.6 Minutiae</td>
<td>27</td>
</tr>
<tr>
<td>6 Numerical method</td>
<td>29</td>
</tr>
<tr>
<td>7 Discussion and conclusion</td>
<td>33</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>35</td>
</tr>
<tr>
<td>APPENDIX A</td>
<td>37</td>
</tr>
</tbody>
</table>
Chapter 1

Fingerprints have been used since the third century B.C. for identification purposes (NIJ, 2011). Their uniqueness and their persistence throughout a person’s life make them ideal for definitively identifying individuals. Fingerprints are invaluable tools for criminal investigations, biometric identification systems, and the identification of missing persons and victims of natural disasters (FBI, 1986).

![Fingerprint Image]

Figure 1.1: A human fingerprint. Source: Federal Bureau of Investigation (1986).

Automated systems for the searching and comparison are central to the modern uses of fingerprints. For example, the Federal Bureau of Investigation’s Automated Fingerprint Identification System (AFIS) makes it possible for law enforcement officials to find matching fingerprints in databases containing millions of records, a task that would be prohibitively time consuming without automation (NIJ, 2011). In order to evaluate and improve the performance of such systems, it is necessary to test them with tens of thousands of sample prints (Cappelli, 2002, 2004). A database of synthetic prints would be useful for “improving testing of fingerprint systems, enhancing the security of a database of fingerprints, [and] training biometric systems” (Wang, 2005, p. iii). Databases of real fingerprints are insufficient for these purposes: aside from the enormous time and effort required to collect enough prints, privacy legislation imposes severe limits on the use of prints from real people (Cappelli, 2004; Ray, 2013). Beyond this need, an effective method for
simulating fingerprints may provide insight into the biological processes underlying print formation (Ray, 2013). The construction of a database of realistic artificial fingerprints would therefore be a worthwhile achievement.

In this thesis, a new model is presented for generating artificial fingerprints. Fingerprint formation is modeled as a process of biological pattern formation. Patterns on the finger are defined by a system of partial differential equations (PDEs) on a suitably constructed domain, with initial conditions that mirror the actual biological process.

We show that this model is capable of simulating a wide variety of fingerprints with a realistic appearance. It is not yet capable of producing fingerprints with enough realism to be nearly indistinguishable from real ones. This is primarily because the details of the simulated prints follow a different distribution than do real ones. However, it is likely that further research will make it possible to impose a realistic distribution on minutiae generated by the model. The fingerprint images produced so far serve as a proof of concept demonstrating the utility and validity of generating prints based on a pattern formation model. Our results show that this is a promising approach that merits deeper research and that, with some refinement, has the potential to be better suited for generating a reliable database than previous models.
Chapter 2

Background

The building blocks of fingerprints are friction ridges. These ridges form on the fingers while the fetus is developing. The biological mechanisms driving this process are still not fully understood (Kücken and Champod, 2013), but it is known that they result from compression forces on the developing skin (NIJ, 2011). Ridges form gradually as the fetus develops, between about 10 and 16 weeks of estimated gestational age (NIJ, 2011), and become the lines visible on any human finger. The broadest level on which fingerprints are classified is the large-scale pattern formed by the ridges near the center of the fingertip. Next, they can be classified by the number of ridges, as counted laterally across the ridges between two defined points (ridge count) or as measured by the number of ridges per square centimeter (ridge width). The details of fingerprints, which make it so that every human fingerprint is unique from every other (FBI, 1986), are known as minutiae, or defects. These defects occur where ridges fail to flow continuously, such as where they end or intersect another ridge. Typical minutiae are identified in Fig. 2.1.

![Figure 2.1: Two drawings of small sections of fingerprints with various types of minutiae identified. In the figure on left, gray circles denote minutiae; in the figure on right, white circles denote pores. Left figure is from Cappelli (2004); right figure is from Kücken and Newell (2005).](image)

Previous research has attempted to simulate fingerprints, with moderate success. Kücken and Newell (2004, 2005) modeled fingerprint formation as a physical buckling process. More recently, Kücken and Champod (2013) took a different approach by sim-
ulating prints with an agent based model. The agents used are Merkel cells, a type of epidermal cell that are hypothesized to be instrumental in friction ridge growth. Cappelli (2002, 2004) presented a method for starting with a set of real “master” prints, and added random disturbances in order to create several images that simulate different impressions of the same original print. Whereas the primary goal of Kücken and coauthors was to study and model the biological formation of fingerprints, Cappelli has no model of the formation process but rather works only with the final image. Wang (2005) uses a method similar to that of Cappelli, manipulating images of real fingerprints.

Cappelli’s images closely resemble real fingerprints to the naked eye (Cappelli, 2002), as do the others’ to a lesser extent. However, significant improvement is still needed. These previous attempts suffer from two principal shortcomings: first, the simulated fingerprints generated are not truly new, but rather are constructed by taking real fingerprints and varying them (Cappelli, 2002; Ray, 2013; Wang, 2005), and second, the minutiae are not distributed in the same way that they are distributed in real fingerprints (Ray, 2013; Wang, 2005). The new model presented in this thesis solves the first problem. The second problem is a challenging one that we do not claim to have solved either; however, we are hopeful that future research and improvements to the model may make it possible to make the minutiae distribution highly realistic.
Chapter 3

Methods

We model fingerprints as a pattern formation process. Patterns are universal in nature and obey similar mathematical descriptions even when the mechanisms driving them are different. Hoyle (2006, p. 1) writes:

Similar patterns are seen in wildly different natural contexts . . . It turns out to be common for a given pattern to show up in several different systems, and for many aspects of its behaviour to be independent of the small details of its environment. This has led to a symmetry-based approach to the description of pattern formation: from this point of view, patterns are universal, and we can find out nearly everything we need to know about them using only their symmetries and those of their surroundings.

For example, notice the similarities between patterns in nature arising from unrelated sources in Fig. 3.1.

This gives us reason to believe that modeling fingerprint formation as a pattern formation problem, with modifications to adapt to the particular biology of fingerprints, will give results that are just as accurate as those obtained by modeling the process on a lower, more biologically precise level, such as on the level of individual cells.

We make two major modifications to adapt a pattern formation model to the case of fingerprint formation. First, we design a domain with a suitable geometry and boundary conditions. According to Murray (2003, p. 154), “Patterns depend strongly on the geometry and scale of the reaction domain”; he suggests that this strong dependence on the domain may largely explain the different mammals exhibit different coat patterns. The second modification is the addition of volar pads to the model. Volar pads are bulges on the surface of the embryonic finger that are central to the print formation process.
Figure 3.1: Similar natural patterns arise in unrelated contexts, such as in leopard spots (a) and butterfly wings (b), zebra stripes (c) and ridges of desert sand (d), and giraffe spots (e) and heated cooking oil with spices (f). Images (a)–(c) are from Murray (2003); images (d)–(f) are from Hoyle (2006).
size and shape of the pad drives the large-scale patterns, and may partially drive the differences between fingerprints and other patterns found in nature (Kücklen and Newell, 2004). Fig. 3.2 displays an artist’s rendering of three differently shaped volar pads and shows how the shape of these pads affects the resultant pattern.

Figure 3.2: The shape of the volar pad in utero affects the shape of the fingerprint. Image from Ashbaugh (1999).

We model the formation of fingerprint patterns with the following reaction-diffusion system, which is presented in non-dimensionalized form by Murray (2003):

\[
\begin{align*}
    u_t &= \gamma f(u,v) + \nabla^2 u \\
    v_t &= \gamma g(u,v) + d\nabla^2 v.
\end{align*}
\] (3.1)

The quantities \( u \) and \( v \) represent the levels of “morphogens,” chemicals that control tissue growth. The extent to which this morphogen formulation reflects the underlying biology of the problem is unimportant. Again, the intent is to emulate the higher-level process of pattern formation, not the biology of the individual cells that comprise the finger. As Murray (2003, p. 75) writes, “it must be stressed again that mathematical descriptions . . . of patterning scenarios are not explanations.” The activator morphogen is given by \( u \) and the inhibitor morphogen by \( v \). The reaction part comes from \( f \) and \( g \), which are nonlinear.
The positive parameter $\gamma$ scales the dimensionless system to the size of the domain. On a two-dimensional domain, $\gamma$ scales linearly with the length of a side of the domain (Murray, 2003), so a doubling of $\gamma$ is equivalent to doubling the length or width of the domain or to halving the wavelength of a periodic solution $(u, v)$. The diffusion coefficient $d > 0$ determines the rate of diffusion of $v$ relative to diffusion of $u$. It will be proven in Sec. 4 that in order for the system to stabilize at a non-uniform state, the inhibitor must diffuse faster than the activator, so $d > 1$. Finally, note that if $d$ varies in space, as it will be made to in Sec. 5 for generating fingerprint patterns, the diffusion term in the equation for $v_t$ in (3.1) becomes instead

$$v_t = \gamma g(u, v) + \nabla \cdot (d \cdot \nabla v).$$

(3.1b)

It has been shown that with a suitable choice of the functions $f$ and $g$, the system (3.1) produces natural patterns, and several forms for $f$ and $g$ have been proposed (Murray, 1982). The system we use for modeling fingerprint formation is the Gierer-Meinhardt system (1972):

$$f(u, v) = a - bu + \frac{u^2}{v}$$
$$g(u, v) = u^2 - v,$$

(3.2) (3.3)

with positive parameters $a$ and $b$. Three other systems were considered: the Schnakenberg (1979), Thomas (1976), and modified Gierer-Meinhardt (1972) systems. Numerical experiments and the analysis in Sec. 4 confirm that the form of $f$ and $g$ is only of secondary importance for determining which patterns are formed: the size, shape, and boundary conditions of the domain and the value of $d$ are primarily responsible for defining patterns. The Gierer-Meinhardt system (3.2–3.3) has the advantages that it requires fixing fewer arbitrary parameters than the Thomas or modified Gierer-Meinhardt systems, and it can more easily produce patterns with some randomness than can the Schnakenberg or Thomas systems. The latter point is crucial for simulating the defects that make every human fingerprint unique. Fig. 3.3 shows patterns generated by the Gierer-Meinhardt and Schnakenberg
systems, both with random initial conditions and with all other parameters identical; the randomness in the Schnakenberg simulation is diffused into a perfectly geometrical final pattern.

Figure 3.3: Four simple prints: the two on the left were generated by the Gierer-Meinhardt system of functions $f$ and $g$, and the two on the right by the Schnakenberg system, all with random initial conditions. With the Gierer-Meinhardt system, this leads to random, asymmetric “defects” in the final print, whereas the Schnakenberg system produces an almost perfectly geometric pattern.

The reaction diffusion system (3.1) is solved on a domain constructed to resemble a fingertip. The activator morphogen $u$ is plotted to create an artificial fingerprint, with higher values darkened to represent ridges. To be precise, the domain and resulting simulated prints are intended to represent a fingerprint taken on a flat surface, as opposed to a live finger. Besides being computationally faster since it is a two- rather than three-dimensional surface, this is a more direct route since the objective is to create a database of fingerprints, not fingers. The finger domain used consists of a rectangle with part of an ellipse joined along the upper edge. (See for example Fig. 3.3.) This represents the section of the finger that is captured in a typical fingerprint: the segment from the fingertip to the first knuckle. The rectangle is oriented with the longer edge parallel to the $y$-axis. The origin is chosen such that the top edge of this rectangle lies on the $x$-axis, and the $y$-axis splits the rectangle in half lengthwise.

The top segment is not a full semiellipse, but rather an ellipse cut off above the horizontal axis. This allows ridges to run parallel to the top ellipse boundary and intersect
the sides of the rectangle without unrealistically sharp bends. Forces at the boundary play a central role in determining patterns (Kücken and Newell, 2005); Sec. 5.1 details how a mixture of Neumann, Dirichlet, and oblique boundary conditions are fixed on the domain to produce realistic prints.

3.1 Creating a database

The goal of this project is to lay the foundation for creating a database of realistic artificial fingerprints. The patterns in fingerprints are classified on three levels: level one details characterize the primary pattern, and levels two and three characterize the small-scale details or minutiae. Most importantly, a model for creating a database must be able to generate a wide variety of level one details. It should generate realistic level two details, preferably with a statistical distribution similar to that of real fingerprints.

We present here a program that generates artificial fingerprints. The user may input the level one details desired, and the program generates a fingerprint accordingly. Stochasticity is built in, which generates level two and three defects and ensures that no two prints are identical. The program is written in MATLAB and allows for easy user interaction and future extensibility. The heavy computational work of solving the PDEs is done in Fortran for efficiency.
Chapter 4

Pattern stability analysis

In order for the reaction-diffusion equations (3.1) to generate patterns, two conditions on the functions $f$ and $g$ must be satisfied. First, the system must be linearly stable to small, spatially homogeneous perturbations, so that any patterns generated are not transitory. Second, the steady-state solution must be unstable to spatial disturbances, so that inhomogeneous solutions are possible. Following Murray (2003), we will look for linearly stable, nontrivial solutions to (3.1), which reveals the patterns that this model can produce.

Recall that $f$ and $g$ are given by

$$f(u,v) = a - bu + \frac{u^2}{v} \quad (3.2)$$
$$g(u,v) = u^2 - v \quad (3.3)$$

Setting $u_t, v_t, \nabla^2 u,$ and $\nabla^2 v$ equal to zero gives the steady-state solution $u_0 = (a + 1)/b$ and $v_0 = (a + 1)^2/b^2$. Linearizing $u$ and $v$ about the steady state gives

$$f(u,v) = -1 - bu + \frac{[u + (a + 1)/b]^2}{v + (a + 1)^2/b^2} \quad (4.1)$$
$$g(u,v) = u^2 - v + \frac{2(a + 1)}{b} u. \quad (4.2)$$

The first-degree Taylor expansions about the steady state are

$$f(u,v) \approx -bu + \frac{2b}{a + 1} u - \left( \frac{b}{a + 1} \right)^2 v \quad (4.3)$$
$$g(u,v) \approx \frac{2(a + 1)}{b} u - v. \quad (4.4)$$

Thus the linearized system is

$$\mathbf{w}_t = \gamma L \mathbf{w} + D \nabla^2 \mathbf{w} \quad (4.5)$$

where $\mathbf{w} = (u,v)^T$, $D = \text{diag}(1,d)$, and

$$L = \begin{pmatrix} f_u & f_v \\ g_u & g_v \end{pmatrix} = \begin{pmatrix} -b + \frac{2b}{a+1} & -\left( \frac{b}{a+1} \right)^2 \\ \frac{2(a+1)}{b} & -1 \end{pmatrix}. \quad (4.6)$$
The Laplacian is zero at the steady state, so (4.5) reduces to

\[ w_t = \gamma L w. \]  

This means the steady state solution is stable if and only if both eigenvalues of \( L \) have negative real part. The eigenvalues are

\[ \lambda = \frac{\gamma}{2} \left( \text{tr}L \pm \sqrt{(\text{tr}L)^2 - 4b} \right), \]  

where \( \text{tr}L = -1 - b + 2b/(a + 1) \) is the trace.

Since by assumption \( a, b, \) and \( \gamma > 0 \), the real part of both values of \( \lambda \) is negative if and only if \( \text{tr}L < 0 \), or equivalently,

\[ a + 1 > \frac{2b}{b + 1}. \]  

Thus, we will restrict \( a \) and \( b \) such that (4.9) holds, so that patterns generated can be linearly stable to spatially homogeneous perturbations.

### 4.1 Inhomogeneous solutions

If we define \( w_k \) to be a solution, constant in time but inhomogeneous in space, to

\[ 0 = k^2 w_k + \nabla^2 w_k, \]  

then the solutions we seek will be superpositions of these solutions:

\[ w = \sum_k c_k e^{\lambda_k t} w_k. \]  

The \( \lambda_k \) are the eigenvalues of the following equation, which follows from (4.5), (4.10), and (4.11):

\[ \lambda_k w_k = \gamma L w_k - Dk^2 w_k. \]  

The values \( k \) are called the wave numbers, as they correspond to the wavelength of periodic solutions to (4.10). In particular, the reciprocal \( k^{-1} \) is proportional to the wavelength of the corresponding solution \( w_k \) (Murray, 2003).
Some algebra shows that these eigenvalues are
\[
\lambda_k = \frac{1}{2} \left( \gamma \text{tr} L - (d + 1)k^2 \pm \sqrt{[\gamma \text{tr} L - (d + 1)k^2]^2 - 4h(k^2)} \right),
\]  
(4.13)

where
\[
h(k^2) := \gamma^2 b + \gamma b d k^2 - \frac{2\gamma b}{a + 1} d k^2 + \gamma k^2 + d k^4.
\]  
(4.14)

The steady state is unstable to a spatial disturbance \( w_k \) corresponding to a given \( k^2 \) if and only if \( \text{Re} \lambda_k \) is positive for one of the two \( \lambda_k \) from (4.13). Condition (4.9) must hold so that linearly stability is assured, so \( \gamma \text{tr} L - (d + 1)k^2 \) must be negative. Thus, there is an eigenvalue \( \lambda_k \) with positive real part precisely when \( h(k^2) < 0 \).

Aside from the parameter \( \gamma \) which is used to scale \( f \) and \( g \) with the size of the domain (Murray, 2003), the set of unstable pattern functions \( w_k \) is determined by \( a, b, \) and \( d \). The Gierer-Meinhardt constants \( a \) and \( b \) are relatively unimportant: the range of values at which instability is possible is narrow, and numerical experiments show that adjusting their values within this narrow range does not substantially change the resultant patterns. Fig. 4.1 shows the Turing space for \( a \) and \( b \): that is, “[t]he domain in parameter space where diffusion can drive the system to a steady spatially inhomogeneous state” (Murray, 1982, p. 144). When \( d \) is near its critical value, which as explained later is where we will want to set it, \( a \) and \( b \) must be near 0.06 and 1, respectively. For values outside this range, there are no eigenvalues with positive real part for any \( k^2 \), and thus no linearly stable nontrivial solutions.

As such, \( d \) should be considered the bifurcation parameter. The value of \( d \) determines the curve \( h(k^2) \) as defined in (4.14): the spatially heterogeneous solutions are those \( w_k \) for which \( h(k^2) < 0 \), given \( d \). Fig. 4.2 shows the range of unstable values of \( k^2 \)—those values for which \( h(k^2) \) is negative—for various values of \( d \). Since \( h(k^2) \) is a parabola in \( k^2 \), these unstable values may be an interval, a single point, or an empty set. The value \( d = d_c \) where \( h(k^2) \) has a double root, leading to exactly one unstable value \( k^2 \), is the critical value of \( d \). When \( d < d_c \), there can be no spatially heterogeneous solutions \( w_k \), and
Figure 4.1: Turing space for the Gierer-Meinhardt functions $f$ and $g$, with diffusion coefficient $d = 7$. Diffusion-driven instability is possible only when the parameter values lie in the shaded region.

When $d$ is much larger than $d_c$, there may be multiple solutions with different wave numbers that interact in highly nonlinear ways due to the form of (4.11). Therefore, in order to control the patterns in a predictable way, we set $d$ to barely above the critical value $d_c$ so that the range of unstable $k^2$ is narrow.

Taking the derivative of (4.14) and setting it equal to zero gives that the minimum is attained at

$$k_{\text{min}}^2 = \frac{\gamma}{2d} \left(-bd + \frac{2bd}{a+1} - 1\right), \quad (4.15)$$

and this minimum value is

$$h_{\text{min}} = h(k_{\text{min}}^2) = \gamma^2 \left[b - \frac{(-bd + 2bd/(a+1) - 1)^2}{4d}\right]. \quad (4.16)$$

For (4.15) and (4.16) to be feasible, $k_{\text{min}}^2$ must be positive, so

$$-bd + \frac{2bd}{a+1} > 1. \quad (4.17)$$
Figure 4.2: Values of $h(k^2)$ for various values of $d$. Patterns can form when $h(k^2) < 0$.

Since $\text{tr}L = -1 - b + 2b/(a+1) < 0$ from (4.9) for linear stability, it must be that $d > 1$, and the inhibitor must diffuse faster than the activator. From (4.16), the minimum of $h(k^2)$ is negative iff

$$-bd + \frac{2bd}{a+1} - 1 > \sqrt{4bd}.$$  

(4.18)

(This implies (4.17) as well.) In that case, the two distinct real roots of $h(k^2)$ are

$$k^2 = \frac{1}{2d} \left( -bd + \frac{2bd}{a+1} - 1 \pm \sqrt{[-bd + 2bd/(a+1) - 1]^2 - 4bd} \right).$$  

(4.19)

Summing up, the reaction diffusion system studied here is capable of generating patterns if the steady state solution is linearly stable, an is unstable to the spatial disturbance corresponding to a positive wave number $k$. This is reflected in the following theorem:

**Theorem 1.** A stable, spatially inhomogeneous solution to the reaction-diffusion system (3.1) on $\mathbb{R}^n$ with functions (3.2) and (3.3) is possible if (4.9) and (4.18) hold. In that case, the allowable wave numbers are those values $k$ such that $k^2$ is in the interval between the two roots (4.19).
Although this tells us which wave numbers \( k \) are allowable on an unbounded domain, in practice the domain geometry limits which solutions may form. On a bounded domain, in order for an inhomogeneous solution \( w_k \) to be allowable, the wave number \( k \) must also have a corresponding eigenfunction that is a solution to (4.10) on the given domain and that satisfies the boundary conditions.

The solutions of (4.10) are eigenfunctions of the Laplacian. Consider a rectangular domain of dimension \( W \times L \) with homogeneous Neumann boundary conditions:

\[
(n \cdot \nabla) w = 0.
\]

The eigenfunction corresponding to eigenvalue \( k \) is \( \exp(ikx) \), where \( k = (k_x, k_y)^T \) is a directional vector with \( |k| = k \). This reduces to

\[
e^{ikx} = e^{i(xk_x + yk_y)}
\]

\[
= \cos(xk_x + yk_y) + i \sin(xk_x + yk_y)
\]

\[
= \cos(xk_x) \cos(yk_y) - \sin(xk_x) \sin(yk_y)
\]

\[
+ i (\sin(xk_x) \cos(yk_y) + \cos(xk_x) \sin(yk_y)).
\]

If we require a real solution satisfying the Neumann boundary conditions, we can then obtain the eigenfunction

\[
w_k = \cos(xk_x) \cos(yk_y);
\]  \hspace{1cm} (4.20)

in order for this to have a zero derivative on all boundaries, we must have \( k_x \) an integer multiple of \( \pi/(W/2) \), and \( k_y \) an integer multiple of \( \pi/L \). This means that (4.20) becomes

\[
w_k = \cos(2n\pi x/W) \cos(m\pi y/L), \ n, m \in \mathbb{Z}.
\]  \hspace{1cm} (4.20b)

Likewise, if we change the boundary conditions to be homogeneous Dirichlet \( (w = 0) \) on the boundaries in the \( y \) direction, while keeping the Neumann conditions on the boundaries in the \( x \) direction, it is easily verified that a solution is

\[
w_k = \cos(2n\pi x/W) \sin(m\pi y/L).
\]  \hspace{1cm} (4.21)
Allowable solutions are those defined by (4.21) with vectors

\[ \mathbf{k} = (k_x, k_y)^\top = (2n\pi/W, m\pi/L), \]  

(4.22)

where the square of the magnitude \( k^2 = |\mathbf{k}|^2 \) is such that \( h(k^2) < 0 \). For example, Fig. 4.2 shows that if \( d = 6.2 \), then \( h(k^2) < 0 \) for \( k^2 \) approximately between 13 and 20. Then, Fig. 4.3 reveals which patterns are possible by showing which combinations of \( k_x \) and \( k_y \) satisfy (4.22) and have the square of their magnitudes between 13 and 20.

![Figure 4.3: Allowable wave numbers \( \mathbf{k} = (k_x, k_y)^\top \) on a rectangular domain. Each circle represents a value of \( k^2 \), with the innermost being \( k^2 = 13 \) and the outermost \( k^2 = 20 \). Each line represents an allowable value of \( k_x \) or \( k_y \). Patterns may thus be formed with vectors \( \mathbf{k} \) corresponding to points that lie on the intersection of two lines and within the range of the circles.](image)

If the domain is circular instead of rectangular, analogous to the upper segment of a finger domain, the eigenfunctions of the Laplacian become Bessel functions. Analytically finding the eigenfunctions on an irregular domain, such as the domain used here for a finger, is intractable (Murray, 2003), but the analytical solutions on the rectangle and the circle strongly suggest the appearance of solutions on a finger-shaped domain consisting of a rectangle joined to a partial ellipse.
Chapter 5

Generating fingerprint patterns

Patterns in fingerprints are classified into three levels. Level one classification is the broadest classification and describes the overall shape of the pattern. Levels two and three describe the path of individual ridges and the details known as minutiae.

Fingerprint patterns come in three primary types. The types are characterized by the presence of singularities: these may be a delta, a location where two parallel ridges diverge, or a core, the location at the center of concentric curved ridges (FBI, 1986). The simplest type is the arch, in which ridges flow continuously from one side of the finger to the other, curving upwards in the center (FBI, 1986). Arches either have no singularities, or have a single delta and core both located approximately in the center of the finger (NIJ, 2011). See for example Fig. 5.1(a).

Figure 5.1: Examples of three real fingerprints. Red circles indicate cores, green circles indicated deltas, and red dots trace recurves. Source: Federal Bureau of Investigation (1986) (annotations added).

The second type is the whorl, an approximately symmetric pattern resembling concentric circles. The whorl is defined by the presence of a delta on each side of a core, each adjacent to a “recurve,” a ridge that curves back towards the direction from which it originated (FBI, 1986; NIJ, 2011). See Fig. 5.1(b). The third type, the loop, contains a single delta and a single recurve, which create an asymmetrical curve pattern (FBI, 1986). See
In the model presented here, the smaller level two and three details are determined by initial conditions, which may contain randomness. Level one patterns are generated by adjusting the primary model parameters. These parameters may be divided into type parameters, which control the primary pattern type (arch, loop, or whorl), and type-independent parameters.

5.1 Type-independent parameters

Certain parameters control the shape and dimensions of the fingerprint for prints of all three primary types. These parameters are $w$, $\ell$, $\ell_{\text{rect}}$, $\theta$, $\gamma$, $h$, $\Delta t$, and $T_{\text{max}}$.

The model domain consists of a rectangle attached to a part of an ellipse. The domain represents the section of the finger that is captured in a typical fingerprint: the segment from the fingertip to the first knuckle. The rectangle is oriented with the longer edge parallel to the $y$-axis. The origin is chosen such that the top edge of this rectangle lies on the $x$-axis, and the $y$-axis splits the rectangle in half lengthwise. A partial ellipse is attached to the top of the rectangle.

The width of the rectangle, $w$, is defined to be the width at the base of the fingerprint, that is, the width of the finger at the knuckle closest to the fingertip. The units used are unimportant, but for numerical stability it helps to choose a scale such that the width $w$ is of $\mathcal{O}(10)$. The length, $\ell$, is measured from the tip of the fingerprint to the base at the knuckle. The parameter $\ell_{\text{rect}}$ defines the length of the rectangular segment of the finger. This represents the distance between the knuckle and the approximate point at which the sides begin curving in towards the center. Thus, the corners of the rectangle lie at $(w/2, 0), (w/2, -\ell_{\text{rect}}), (-w/2, -\ell_{\text{rect}}),$ and $(-w/2, 0)$. The uppermost point of the ellipse lies at $(0, \ell - \ell_{\text{rect}})$.

The parameter $\gamma$, in the Gierer-Meinhardt system (3.2)–(3.3), represents the linear
size of the domain. Its function is therefore to determine the number of ridges appearing on the finger. The higher the value of $\gamma$, the higher the “ridge count” of the fingerprint.

The next type-independent parameter, $\theta$, relates to the boundary conditions of the domain. On most real fingerprints, the ridges run approximately parallel to the top and bottom edges of the finger. To force this to occur in simulated fingerprints, the boundary conditions are chosen to be homogeneous Dirichlet boundary conditions on the boundary of the ellipse and on the bottom edge of the finger:

$$ (u,v)^T = 0, x \text{ on } \partial\mathcal{E} \text{ or } y = -l_{\text{rect}}, $$

where $\partial\mathcal{E}$ is the boundary of the ellipse segment. On the sides of the rectangle part of the domain, “oblique” boundary conditions are used. These are defined by

$$ \mathbf{m} \cdot (\nabla(u,v)^T) = 0, x = \pm w/2, $$

where $\mathbf{m} = \mathbf{m}(x)$ is a unit vector pointing outward from the boundary. When $\mathbf{m}$ is normal to the boundary, this becomes a homogeneous Neumann boundary condition. The choice of $\mathbf{m}$ allows for more realistic simulated fingerprints, where ridges may intersect the edges at various angles as they do in real prints.

The direction of the unit vector $\mathbf{m}$ is parameterized by $\theta = \theta(x)$. At each point $x$ on the side boundaries, $\theta(x) \in [0, \pi/2)$ is defined to be the angle below the horizontal of the direction of $\mathbf{m}$. E.g., where $\theta(x) = 0$, $\mathbf{m}$ is the unit normal, and the boundary condition is Neumann.

Rather than requiring the direction of each individual ridge to be defined by the user, the model can accept the three parameters $\theta_{\text{min}}, \theta_{\text{max}},$ and $y_{\delta}$. Then, $\theta_{\text{min}}$ is the value of $\theta$ at the two points $(\pm w/2, y_{\delta})$, where $y_{\delta} \in (-l_{\text{rect}}, 0)$, $\theta_{\text{max}}$ is the value at $(\pm w/2, 0)$, $\theta(x)$ is linearly interpolated between $\theta_{\text{min}}$ and $\theta_{\text{max}}$ for $y_{\delta} < y < 0$, and $\theta(x) = 0$ for $y < y_{\delta}$. This scheme reflects the fact that on real fingerprints, ridges tend to intersect the edges at a steep angle near the fingertip and be nearly perpendicular to the edge closer to the base.
of the finger. The parameter $y_\delta$ represents the approximate $y$-coordinate of the delta(s) if present, a feature that will be explained below with the type parameters. A more complex formula for $\theta(x)$ may be used instead if the user wishes to generate more complicated or asymmetrical ridge patterns.

The numerical method used requires type-independent parameters for the grid step size $h$, the time step $\Delta t$, and the time of simulation $T_{\text{max}}$. These bear no relation to the physical fingerprint being simulated, but are crucial for numerical stability. The greater the value of $\gamma$, and thus the higher resolution needed in the image, the smaller the grid step $h$ must be. If $h$ is too large relative to $\gamma$, a “honeycombing” phenomenon occurs wherein parallel ridges appear to connect, producing an unrealistic image. To ensure stability, the time step $\Delta t$ must decrease as the square of $h$ to satisfy the CFL condition. The duration of the simulation, $T_{\text{max}}$, must be large enough to permit the system to settle at a stable solution.

The final type-independent parameter is the degree of randomness in the initial conditions. This is explained in Sec. 5.6.

5.2 Formation of level-one patterns
Two complementary factors drive the formation of different level-one patterns. The first is the diffusion coefficient $d$. As shown in Sec. 4.1, patterns form only when $d$ is greater than a certain critical value $d_c$. By controlling the value of $d$, which may be inhomogeneous in time and space, it is thus possible to control the location and onset of patterns. For example, by setting $d$ to a different value inside an ellipse near the center of the finger than in the rest of the finger, it is possible to force a whorl to form along the ellipse.

The second factor is the use of simulated volar pads, the bulges on the finger that arise in embryonic print growth. The volar pad in this model is an elevated surface in the initial conditions: the initial values of $u$ and $v$ are assigned by a Gaussian function centered near the center of the finger. The size and symmetry of the initial conditions may
be altered to produce different patterns, in a way that closely mirrors how real volar pads determine the resulting pattern (Ashbaugh, 1999; NIJ, 2011). We simulate low pads in order to generate arches, high pads to generate whorls, and asymmetric pads to generate loops, similar to what is shown in Fig. 3.2.

It appears that the latter approach of using volar pads to generate desired patterns is more reliable than the approach of manipulating $d$ and should be preferred, particularly for generating whorls and loops. It can produce recurves and concentric ridges more consistently, and it has an appealing biological interpretation. However, the ability to adjust $d$ is still of some use, particularly for generating arches, and may be useful for future improvements to the model.

One desirable feature of this model is that in addition to producing good end results, the simulated fingerprints appear to coalesce in a similar manner to real fingerprints. This can be seen in Fig. 5.2, which shows a simulated fingerprint at various points in its development alongside an artist’s rendition of a real finger in development.

Figure 5.2: The developmental process of prints in the model presented here mirrors that of real fingerprints. The top row shows drawings of a print in development in utero; prints form first at the top, bottom, and center as indicated. The bottom row shows stages in the process of simulating a fingerprint. Drawings are from Wertheim and Maceo (2002).
5.3 Generating arches

Arches are the simplest patterns to generate, and they are the pattern that forms naturally in the absence of any further modifications beyond the type-independent parameters listed above. Two notable features that characterize arches are the location where the ridges begin curving and whether or not the arch is “tented.” Tented arches are defined by the presence of a delta singularity at the center of the finger; in ordinary arches, the ridges progress continuously up the finger with the curvature increasing gradually in the direction of the fingertip.

The location where the ridges begin to curve upward, which for a tented arch is the location of the delta, is set by the boundary conditions as determined by the angle $\theta(x)$. The point $y_\delta$ above which the ridges reach the boundary at an angle determines the location of the delta, for a tented arch, or the center of the arch, for a plain arch.

The boundary conditions determine whether an arch is tented or plain. Specifically, this is determined by the angle $\theta(x)$, which gives the angle at which the vector $\hat{m}$ in the oblique boundary conditions intersects the side of the finger. Higher values of $\theta$ force steeper ridges which creates tented arches, whereas lower values create plain arches.
Inhomogeneity of the diffusion coefficient \( d \) is crucial for forming realistic arches. Rather than setting \( d \) to above a critical value from the beginning, \( d \) must be set critical only near the top and bottom of the finger, leaving it below critical in the center. Once the beginnings of an arch pattern have stabilized near the top and bottom, \( d \) is gradually raised to above criticality in the center, causing an arch pattern form there too. Under the volar pad approach, an arch is generated by an initial volar pad that is small and low relative to the surrounding initial conditions. This has the effect of creating a pattern that develops in largely the same way everywhere on the finger, without allowing for the formation of stable recurves necessary to generate whorls or loops.

5.4 Generating whorls

Whorl patterns are characterized by a core in the center of the finger, a number of ridges forming a spiral or concentric circles around the core, and a delta on each side of the central pattern. Whorls are roughly elliptical, and that is how they are implemented in the model. The level-one details of the whorl are explained by five elliptical parameters: the \( x \)- and \( y \)-coordinates of the core, which is considered the center of the ellipse; the length of the semi-major axis, defined as the maximum distance from the core to either of the two recurves surrounding the central whorl; the semi-minor axis, defined as the distance from core to recurve in a direction orthogonal to the major axis; and the orientation of the axes relative to the coordinate system of the finger domain, defined as an angle clockwise off the \( y \)-axis.

To generate a whorl pattern, the bifurcation parameter \( d \) is first set to be above the critical level \( d_c \) in the ellipse defined above, and below critical outside the ellipse. After enough time has passed for a spiral or concentric circles “target” to form, \( d \) is raised to the critical level on the entire domain, allowing the remaining ridges to coalesce around the central pattern. Including some stochasticity or heterogeneity in the initial conditions prevents the final loop patterns from being too close to a geometrically perfect, and biolog-
Fingerprint formation: Morphogen concentrations

Figure 5.4: A simulated fingerprint with a whorl patterns.

ically unrealistic, ellipse. Alternately, using the volar pad approach, the initial distributions of $u$ and $v$ should be a Gaussian curve centered at the core and with a high peak. The advantage of using a volar pad to force a whorl is that it produces more realistic patterns since it is better at avoiding the phenomenon of “honeycombing,” wherein parallel ridges connect to form a web of small circles as seen in Fig. 5.5. The drawback, though, is that controlling the value of $d$ makes it much easier to control the size of the whorl. When using a volar pad initial condition, the resulting whorl may expand to cover an area much larger than the base of the initial Gaussian curve.

In numerical simulations, whorl patterns where the ellipse is highly eccentric or close to the bottom of the finger tend to be unstable. This is realistic and perhaps lends some credibility to the patterns formed by the model, since in real life, whorls tend to be close to circular and form near the center of the fingertip. As of now, our model cannot be made to generate the more exotic “double loop” and “accidental” whorl varieties.

5.5 Generating loops

The parameters of the loop mirror those of the whorl. Loops are also characterized primarily by the location, shape, and orientation of the ellipse-like shape formed by the loop. As
Figure 5.5: Example of a simulated fingerprint that suffers from the phenomenon of “honeycombing,” wherein parallel ridges connect to form a web of small circles, causing the print to look unrealistic.

in the case of whorls, the center, axes, and orientation of the ellipse are again defined by the location of the core and the distances to the recurve.

Figure 5.6: Two simulated fingerprints with loop patterns.

The loop has two additional parameters unique to it. Since the loop is inherently asymmetrical in a way that arches and whorls are not, it takes a parameter for whether it is a left loop or right loop. A loop is characterized by a recurve that originates from one side of the finger, traces the outline of the ellipse-like figure that is the loop itself, and returns
to the boundary on the same side. A fingerprint has a left loop if this recurve abuts the boundary on the left side. Last are the $y$-coordinates of the two locations where the recurve abuts the boundary.

As with whorls, loops can be generated via either inhomogeneous values of $d$ or Gaussian initial conditions resembling a volar pad. The diffusion coefficient $d$ is first set above its critical value inside the ellipse, as with a whorl, and also in the region bordered by the edge of the ellipse, the two sides of the recurve, and the side of the finger. This allows the pattern to form here and connect with the ellipse, forming a distinct loop. When using a volar pad, the initial Gaussian curve must be highly peaked as for a whorl. The resulting pattern is forced to be a loop rather than a whorl by making the curve asymmetrical: it must be centered on either side of the $y$-axis and must be oriented at an angle rather than perpendicular to the axes. The advantages and disadvantages of these two approaches are the same as for whorls.

5.6 Minutiae

The previous three sections describe how to generate level one patterns. No such explicit formula exists in this model for generating the minutiae that comprise patterns of level two and, to an extent, level three. Such a formula would not even be practical, since the purpose of this model is not to mimic a particular fingerprint, but rather to generate a series of unique fingerprints that may be specified to fit certain broader, level one patterns.

Minutiae, or defects in the patterns, arise from two sources. First is the intersection of conflicting patterns. On a whorl print, for example, defects are liable to arise at the areas where the concentric circles of the whorl border the arch-like ridges emanating from the top and bottom of the finger. These defects are a natural consequence of the interaction of incompatible pattern areas. The second source of defects is randomness in the initial conditions. The initial conditions on the morphogen concentrations $u$ and $v$ affect the final pattern because they determine the coefficients $c_k$ in the constant in time, inhomogeneous
in space solutions in (4.11). These coefficients are the Fourier coefficients of the initial conditions. Randomness is added by adding independent, uniformly distributed random perturbations to the values of $u$ and $v$ on each grid cell. The more randomness in the initial conditions, as measured by the magnitude of the random perturbations added, the more defects will remain in the final print. Higher levels of randomness also force the ridges to be less smooth or geometrical in appearance. This is exactly to be expected, as real fingerprints with more defects tend to have less smooth ridge paths (E. Ray, personal communication, March 16, 2013).

Although this method is capable of creating adequate minutiae, it does not produce fingerprints with the same probability distribution of minutiae as real prints. This is an area in which this model is in need of improvement.
An alternating direction implicit (ADI) method is used to solve the reaction-diffusion PDEs. The ADI method, presented by Peaceman and Rachford (1955), is an operator splitting method. Each time step is split into two half-steps, and in each half-step the system is solved implicitly in one spatial direction and explicitly in the other.

The Peaceman-Rachford ADI scheme for the system 3.1 is

\[
\begin{align*}
    u_{i,j}^{n+1/2} &= u_{i,j}^n + \frac{\Delta t}{2} \gamma_f(u_{i,j}^n, v_{i,j}^n) + \frac{\Delta t}{2h_x^2} \left( \delta_x^2 u_{i,j}^n + \delta_y^2 u_{i,j}^{n+1/2} \right) \\
    v_{i,j}^{n+1/2} &= v_{i,j}^n + \frac{\Delta t}{2} \gamma_g(u_{i,j}^n, v_{i,j}^n) + \frac{\Delta t}{2h_y^2} \left( \delta_x^2 v_{i,j}^n + \delta_y^2 v_{i,j}^{n+1/2} \right) \\
    u_{i,j}^{n+1} &= u_{i,j}^{n+1/2} + \frac{\Delta t}{2} \gamma_f(u_{i,j}^{n+1/2}, v_{i,j}^{n+1/2}) + \frac{\Delta t}{2h_x^2} \left( \delta_x^2 u_{i,j}^{n+1/2} + \delta_y^2 u_{i,j}^{n+1/2} \right) \\
    v_{i,j}^{n+1} &= v_{i,j}^{n+1/2} + \frac{\Delta t}{2} \gamma_g(u_{i,j}^{n+1/2}, v_{i,j}^{n+1/2}) + \frac{\Delta t}{2h_y^2} \left( \delta_x^2 v_{i,j}^{n+1/2} + \delta_y^2 v_{i,j}^{n+1/2} \right),
\end{align*}
\]

(6.1)

where the superscript is time, the subscripts are \(x\) - and \(y\)-coordinates, and \(\delta_x^2\) and \(\delta_y^2\) are the second order central differences:

\[
\begin{align*}
    \delta_x^2 u_{i,j}^n &= u_{i-1,j}^n - 2u_{i,j}^n + u_{i+1,j}^n \\
    \delta_y^2 u_{i,j}^n &= u_{i,j-1}^n - 2u_{i,j}^n + u_{i,j+1}^n.
\end{align*}
\]

This is faster than an explicit scheme, since larger time steps may be used without loss of stability. It is faster than a fully explicit scheme because at each step it is only necessary to solve a tridiagonal system for the implicit part of the update, rather than a full or banded system. This can be done in linear time.

The ADI method is accurate on the order of \((\Delta t)^2\) (Britz, 2009). Theoretical results show that it is unconditionally stable (Hundsdorfer and Verwer, 1989, 2003). This is true when solving the homogeneous heat equation, that is, when \(f = g = 0\), as numerical tests confirm. However, in practice the choice of \(f\) and \(g\) may cause the method to be only conditionally stable.
In matrix form, the scheme (6.1) is approximately
\[
\begin{pmatrix}
  u^{n+1/2} \\
  v^{n+1/2}
\end{pmatrix}
= \begin{pmatrix}
  u^n \\
  v^n
\end{pmatrix}
+ \frac{\Delta t}{2h^2} D_x \begin{pmatrix}
  u^n \\
  v^n
\end{pmatrix}
+ D_x^2 \begin{pmatrix}
  u^{n+1/2} \\
  v^{n+1/2}
\end{pmatrix}
+ \frac{\Delta t}{2} \gamma L \begin{pmatrix}
  u^n \\
  v^n
\end{pmatrix}
+ \frac{\Delta t}{2h^2} D_y \begin{pmatrix}
  u^n \\
  v^n
\end{pmatrix}
+ D_y^2 \begin{pmatrix}
  u^{n+1/2} \\
  v^{n+1/2}
\end{pmatrix}
+ \frac{\Delta t}{2} \gamma L \begin{pmatrix}
  u^n \\
  v^n
\end{pmatrix},
\]
(6.2)

\[
\begin{pmatrix}
  u^{n+1} \\
  v^{n+1}
\end{pmatrix}
= \begin{pmatrix}
  u^{n+1/2} \\
  v^{n+1/2}
\end{pmatrix}
+ \frac{\Delta t}{2} \gamma L \begin{pmatrix}
  u^{n+1/2} \\
  v^{n+1/2}
\end{pmatrix}
+ \frac{\Delta t}{2h^2} D_x \begin{pmatrix}
  u^{n+1/2} \\
  v^{n+1/2}
\end{pmatrix}
+ \frac{\Delta t}{2} \gamma L \begin{pmatrix}
  u^{n+1/2} \\
  v^{n+1/2}
\end{pmatrix},
\]
(6.3)

with the only difference being that the nonlinear functions \(f\) and \(g\) are approximated by the matrix \(L\) which represents their linearizations about the steady state derived in Sec. 4. Here, \(D_x^2\) and \(D_y^2\) are the central second difference operators in the \(x\)- and \(y\)-directions, and \(D = \text{diag}(1, \ldots, 1, d, \ldots, d)\). This is equivalent to:
\[
\begin{pmatrix}
  u^{n+1} \\
  v^{n+1}
\end{pmatrix}
= A \begin{pmatrix}
  u^n \\
  v^n
\end{pmatrix},
\]
(6.4)

where
\[
A = \left(I - \frac{\Delta t}{2h^2} DD_x^2\right)^{-1} \left(I + \frac{\Delta t}{2h^2} DD_y^2 + \frac{\Delta t}{2} \gamma L\right) \left(I - \frac{\Delta t}{2h^2} DD_x^2\right)^{-1} \left(I + \frac{\Delta t}{2h^2} DD_y^2 + \frac{\Delta t}{2} \gamma L\right). \quad (6.5)
\]

Define \(\rho(A)\) as the spectral radius of \(A\). Then, \(\lim_{n \to \infty} A^n = 0\), and the method is stable, iff \(\rho(A) < 1\) (Quarteroni et al., 2007). Determining the eigenvalues analytically is intractable: even though the matrices \(D, D_x^2, D_y^2,\) and \(L\) are sparse, \(A\) is dense in general. We ran numerical tests by randomly selecting the parameters \(\Delta t, h, d, \gamma, a,\) and \(b\), creating the matrix \(A\) for thousands of randomly chosen combinations of these parameters, and computing the maximum eigenvalue.

The stability depends primarily on the product \(\gamma \Delta t\) and on the grid step size \(h\). Larger values of \(\gamma \Delta t\) and larger values of \(h\) lead to larger eigenvalues \(\rho(A)\) and thus less stability. This makes intuitive sense when we consider the limiting case where \(\gamma \Delta t \to 0\) and \(\Delta t/2h^2 \to \infty\); then, the functions \(f\) and \(g\) represented by the linearized matrix \(L\) disappear and the scheme is dominated by the second derivative terms. This approaches the case where \(f = g = 0\), when the scheme is unconditionally stable. The larger \(\gamma \Delta t\) grows and the
Figure 6.1: Values of the grid step size $h$ and domain scaling parameter $\gamma$ times time step $\Delta t$. Each point represents a randomly selected combination of the model parameters. The red points are those for which the ADI method is stable.

larger $h$ grows, the more $f$ and $g$ affect the scheme and drive it unstable. Figs. 6.1 and 6.2 show the magnitude of the eigenvalue as a function of $\gamma \Delta t$ and $h$. 
Figure 6.2: Spectrum of $A$ plotted against $\gamma \Delta t$. Each point represents a randomly selected combination of the model parameters. The ADI method is stable when $\rho(A) < 1$. The red points are those for which the method is stable. The value of $\gamma \Delta t$ is the most important determinant of stability.
Chapter 7

Discussion and conclusion

This thesis has presented a new model, based on theory of biological pattern formation, for generating artificial fingerprints. This model is capable of simulating prints that resemble real prints. It offers a significant advantage over previous methods in that the prints that it creates need not be directly based on a particular real fingerprint. This provides a strong reason for further work on synthetic fingerprints to proceed along the lines of models based on pattern formation and the Gierer-Meinhardt system. Although we do not have a finished product capable of generated a perfect database, we have presented a proof of concept demonstrating that this model is a viable approach.

![Frequency distribution of defects on fingerprints of whorl type.](image)

Figure 7.1: Frequency distribution of defects on fingerprints of whorl type. Although defects are concentrated near the center, the defects in artificial prints are even more frequent near the center and too rare near the edges. Source: Dutton et al. (2012).

The largest problem with the model is that the distribution of minutiae does not match that seen on real fingerprints. This is a difficulty that no model has yet been able to solve. Although real fingerprints tend to have more minutiae near the center than on the edges (Dutton et al., 2012) as illustrated in Fig. 7.1, our model, like the recent model
from Kücken and Champod (2013), still has too many minutiae near the center and especially near deltas. Extensive research exists regarding the statistical distribution of defects, including the distribution of locations of specific types of defects (Dutton et al., 2012). The priority for future research should focus on discovering the best way to translate this knowledge of frequency distributions into an algorithm for enforcing a certain minutiae distribution on the simulated fingerprints produced by the model. With further research and refinement, it may be possible to create databases of synthetic fingerprints that are effectively indistinguishable from a collected database of real prints.
REFERENCES


This appendix describes the parameter values used in simulations.

1. Domain size: This includes the width $w$ and length of the rectangular portion $\ell_{\text{rect}}$. The ratio $\ell_{\text{rect}}:w$ was set to 1.6, which is approximately the ratio on a typical finger when $\ell_{\text{rect}}$ is measured starting at the first knuckle. Given a fixed ratio, the value of each is arbitrary since the parameter $\gamma$ serves to scale the governing equations to the domain size. We used a value of $w = 30$ since it seemed to be more numerically stable than very large or small widths.

2. $\theta(x)$: This was parameterized by three values $\theta_{\text{max}}$, $\theta_{\text{min}}$, and $y_{\delta}$. We took $\theta$ to be a function of the $y$-coordinate only, so that the angle was the same on opposite sides of the finger. $\theta_{\text{max}}$ gives the value of $\theta$ at $y = 0$, i.e. at the intersection of the rectangular and elliptical sections. $y_{\delta}$ is the $y$-coordinate at which $\theta = 0$, so named because it determines the location of the delta(s) on a tented arch, whorl, or loop. Then, $\theta$ was linearly interpolated between $y = y_{\delta}$ and $y = 0$, with $\theta(y_{\delta}) = \theta_{\text{min}}$ and $\theta(0) = \theta_{\text{max}}$. $\theta(y) = 0$ for all $y < y_{\delta}$. Typical values used were $\theta_{\text{max}} = 0.35\pi$, $\theta_{\text{min}} = 0.4$, and $y_{\delta} = -25$. The two parameters $\ell_{\text{rect}}$ and $\theta(0)$ determine the shape of the partial ellipse, by determining the height and the angle formed by its intersection with the top of the rectangle. $\ell_{\text{rect}}$ was chosen to be $0.25\ell$, a ratio typical on real fingers.

3. $T_{\text{max}}$ was set to 10, since in practice this was sufficient time for the morphogen concentrations to converge to a steady state.

4. The grid step size $h$ must be sufficiently small to permit fine details and to avoid “honeycombing.” We used $h = 0.1$.

5. $\gamma$ determines the ridge count. Typical simulations such as those shown in this paper used $\gamma$ between 60 and 90.

6. The time step $\Delta t$ must be small enough to ensure stability. It usually suffices to set $\gamma\Delta t < 0.2$. We used $\Delta t = 0.00125$ for typical simulations.

7. $d$: Based on the stability analysis of the Gierer-Meinhardt system, the critical value $d_c = 6.2$ was chosen as it is barely above criticality. A below critical value $d_0$ was set to equal 2. In the example prints in this paper, $d$ was set as a function of the $y$-coordinate. It was first set to the critical value $d_c$ at the top and bottom, and set to $d_0$ in the center. The exact values used were:

For $0 \leq t < 2$:

\[
d(y) = d_0 \quad \text{at} \quad -\ell + 0.25(y_{\delta} + \ell) < y < 0.25(\ell_{\text{rect}} - y_{\delta}),
\]

\[
d(y) = d_c \quad \text{otherwise}.
\]

For $2 \leq t < 4$:

\[
d(y) = d_0 \quad \text{at} \quad -\ell + 0.50(y_{\delta} + \ell) < y < 0.50(\ell_{\text{rect}} - y_{\delta}),
\]

\[
d(y) = d_c \quad \text{otherwise}.
\]

For $4 \leq t < 6$:
\[ d(y) = d_0 \text{ at } -\ell + 0.75(y_{\delta} + \ell) < y < 0.75(\ell_{\text{rect}} - y_{\delta}), \]
\[ d(y) = d_c \text{ otherwise.} \]

For \( 6 \leq t < 10: \)
\[ d(y) = d_c \forall y. \]

8. \( a \) and \( b \) were set to \( a = 0.07 \) and \( b = 1. \)

9. The formula for the initial conditions was

\[
u_0(x,y) = v_0(x,y) = \max\{z_0, \exp(-((x \cos \alpha - y \sin \alpha - x_0 \cos \alpha + y_0 \sin \alpha)^2/2m_x^2 + (x \sin \alpha + y \cos \alpha - x_0 \sin \alpha - y_0 \cos \alpha)^2/2m_y^2)) + cU(x,y), \]

where:

- \( z_0 \) is a constant such that higher values of \( z_0 \) produce a relatively flatter volar pad. For arches, we use \( z_0 = 0.3 \); for loops and whorls, \( z_0 = 0.1. \)
- \( \alpha \) is the angle off the \( y \)-axis representing the orientation of the volar pad. For arches and whorls, \( \alpha \approx 0; \) for the example loops in this paper, we use \( \alpha = \pi/4. \)
- \( x_0, y_0 \) are the \( y \)-coordinates of the center of the volar pad. The arches and whorls shown use \( (x_0, y_0) = (0, -20); \) the loops use \( (x_0, y_0) = (5, -20). \)
- \( m_x \) and \( m_y \) were set to approximately 0.2w.
- \( U(x,y) \) is a uniform(0,1) random variable, whose values at the grid points are mutually independent. The constant \( c, \) representing the magnitude of the randomness included, was set between 0.1 and 0.3.

10. Lastly, post processing was applied to the resulting image to make it look more like a fingerprint. This was done by thresholding the top and bottom values. Specifically, the greatest five percent and lowest 40 percent of values of \( u \) were set to equal the fifth and twentieth percentiles, respectively.