GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

Characteristic and necessary minutiae in fingerprints

Dissertation

zur Erlangung des mathematisch-naturwissenschaftlichen Doktorgrades

“Doctor rerum naturalium”

der Georg-August-Universität Göttingen

im Promotionsprogramm “Mathematical Sciences” der Georg-August University School of Science (GAUSS)

vorgelegt von

Johannes Wieditz

aus Eschwege

Göttingen, 2021

Betreuungsausschuss:

Prof. Dr. Stephan F. Huckemann
Felix-Bernstein-Institut für Mathematische Statistik in den Biowissenschaften,
Georg-August-Universität Göttingen

Prof. Dr. Dominic Schuhmacher
Institut für Mathematische Stochastik, Georg-August-Universität Göttingen

Prof. Dr. Anja Sturm
Institut für Mathematische Stochastik, Georg-August-Universität Göttingen

Mitglieder der Prüfungskommission:

Referent:
Prof. Dr. Stephan F. Huckemann
Felix-Bernstein-Institut für Mathematische Statistik in den Biowissenschaften,
Georg-August-Universität Göttingen

Korreferent:
Prof. Dr. Dominic Schuhmacher
Institut für Mathematische Stochastik, Georg-August-Universität Göttingen

Weitere Mitglieder der Prüfungskommission:

PD Dr. Ulf-Rainer Fiebig
Institut für Mathematische Stochastik, Georg-August-Universität Göttingen

Prof. Dr. Preda Mihăilescu
Mathematisches Institut, Georg-August-Universität Göttingen

Dr. Yvo Pokern
Department of Statistical Science, University College London

Jun.-Prof. Dr. Daniel Rudolf
Institut für Mathematische Stochastik, Georg-August-Universität Göttingen

Tag der mündlichen Prüfung: 16. August 2021

Acknowledgements

First of all, I would like to express my sincere gratitude to my supervisors Prof. Dr. Stephan F. Huckemann and Prof. Dr. Dominic Schuhmacher who introduced me in the realms of fingerprint recognition and spatial statistics. They always had an open ear for all my questions however trivial they seemed to be and discussed them with me in countless meetings. I am moreover deeply indebted to Dr. Yvo Pokern who showed me how to properly use the tools of Bayesian statistics and make inference in this framework.

Furthermore, I would like to thank the committee members PD Dr. Ulf-Rainer Fiebig, Prof. Dr. Preda Mihăilescu, Dr. Yvo Pokern and Jun.-Prof. Dr. Daniel Rudolf for taking part in my defence.

Special thanks to Christian Böhm for his outstanding IT-support. He had always a helpful answer to even the most peculiar questions. I am also grateful to Corvin Grigutsch for writing large parts of the Java software and supporting me in the development of *MiSeal* and without whom this software project would probably never have been finished.

I gratefully acknowledge the financial support by the DFG Research Training Group 2088 "Discovering structure in complex data: Statistics meets Optimization and Inverse Problems", the Niedersachsen Vorab of the Volkswagen foundation and the Felix-Bernstein-Institute of Mathematical Statistics in the Biosciences.

I was very happy to work in the very pleasant atmosphere the Institute of Mathematical Stochastics provided. I would like to thank all my colleagues and friends I made through the last four years for creating such a comfortable working atmosphere even during the months of the COVID-19 pandemic. In particular thanks to Dr. Benjamin Eltzner, Marcel Klatt, Dr. Peter Kramlinger, Jonas Lueg, Raoul Müller, Viacheslav Natarovskii, Dr. Robin Richter, Marco Seiler, Dr. Kilian Stampfer, Dr. Carla Tameling and Christoph Weitkamp.

I would like to extend my gratitude to my friends and former student fellows from Ilmenau: Dario Dennstädt, Anna Marie Meisinger, Dr. Samuel Mohr, Jessica Reinholz, Hanne Wallraven and Sebastian Zeng. Especially in a time when it was not possible to visit each other, I enjoyed our bi-weekly *alumni talk* which always was a great opportunity to stay in contact and, indeed, have a beer (or two) together. In particular thanks to Dario for proofreading this entire thesis.

Moreover, I will always be thankful to my (former) flatmates Melina Dippel, Katharina Hinz, Tobias Kruse, Leonie Lehlbach, Anna-Sofie Meyer, Marie Neufeldt and Celine Storz who were always supporting me even when I was unmotivated to work and for getting my mind off when I was in dire need of it.

Last but definitely not least I like to thank my parents Angela Weidner-Wieditz and Werner Wieditz and my sister Katharina Wieditz for their emotional support who always made me feel like I was doing the right thing, even when I least believed it.

Preface

Identification and authentication using biometric information, especially fingerprints, has become increasingly popular during the last years. Typical applications are for instance smartphones and internet banking on the commercial side and in border control or forensics on the governmental side. A fingerprint is a pattern of interleaved ridge lines following an undirected orientation field (OF). Ridges vary in width, inducing a moderately varying ridge frequency (RF). Points in which ridge lines end or fork are called *minutiae*. These points are used in modern algorithms for identification and authentication.

Geometrically, minutiae can occur due to diverging ridge lines with a nearly constant RF or by widening of parallel ridges making space for new ridge lines originating at minutiae (and, indeed, combinations of both). We call these the *geometrically necessary* minutiae. In this thesis, we provide a mathematical framework based on vector fields in which orientation fields, ridge frequency as well as the *number of geometrically necessary minutiae* become tangible and easily computable. As a result, we obtain, given a fingerprint, its *intensity* μ of geometrically necessary minutiae whose value $\mu(z_0)$ in a point z_0 predicts the number of minutiae necessary within an small region around that point z_0. We provide algorithms and software to compute μ and point out corresponding notorious computational issues. It turns out that fingerprints feature additional minutiae which occur at rather arbitrary locations. We call these the *random* minutiae.

The presence of necessary and random minutiae gives rise to a point pattern separation problem:

> *Consider a point pattern* $\zeta = \{z_1, z_2, \dots, z_n\}$ *and suppose* $\zeta = \xi \,\dot\cup\, \eta$ *is a sample from the superposition* $\mathrm{Z} = \Xi \,\dot\cup\, \mathrm{H}$ *of two point processes* Ξ *and* H. *Determine an allocation vector* $\mathbf{W} \in \{0,1\}^n$ *with* $\mathbf{W}_i = \mathbf{1}\,(z_i \in \eta)$ *indicating the separation of* ζ *into* $\xi \sim \Xi$ *and* $\eta \sim \mathrm{H}$.

For our application, we assume the minutiae point pattern to be a realisation of the superposition $\Xi \,\dot\cup\, \mathrm{H}$ of two point processes Ξ and H modelling the random and necessary minutiae, respectively. A statistical model analysis suggests to model the process H as a Strauss process with an activity function given by the intensity field μ and an additional hard core to incorporate that ridges have a certain minimal distance between each other. The random minutiae are modelled by a homogeneous Poisson point process. Given a minutiae pattern we strive for a method allowing for separation of minutiae and inference for the model parameters. Repulsion between the points leads to intractabilities in the model likelihood (due to the Strauss density) prohibiting to apply classical approaches such as maximum likelihood estimation. Hence we have to apply more sophisticated methods and consider the problem from two view points.

From a frequentist point of view we first solely aim on estimating the model parameters (without separating the processes). To this end, we lay the foundations for parametric inference by deriving the density of the superimposed process and provide an identifiability result. We propose an approach for the computation of a maximum pseudo-likelihood estimator and highlight benefits and drawbacks of this estimator on real and simulated data. Having in mind only the estimation of the model parameters, naturally, this approach is unable to perform point process separation.

Following a Bayesian approach we are able to separate Ξ and H in a probabilistic sense. We propose an MCMC-based minutiae separating algorithm, *MiSeal*, allowing for exploring the distribution of necessary and random minutiae of a given fingerprint as well as the model parameters. The intractabilities in the Strauss density require the application of the auxiliary variable method to compute the transition probabilities of the Markov chain. We propose a specific choice of the distribution of the auxiliary variable which empirically leads to a good mixing of the resulting algorithm eliminating a major drawback of the method proposed by Redenbach et al. (2015). The algorithm is provided as a publicly available software package and ready-to-use also for application to separation problems beyond fingerprint recognition.

We analyse the performance of MiSeal for simulated and real data. In general, we observe good estimation of the underlying model parameters. Moreover, MiSeal provides for inference methods on the underlying assignment distribution of the minutiae to the two processes. This allows for the separation of minutiae patterns in a probabilistic sense. In an example of two different prints with similar OF and RF we can verify, using the posterior distribution of the minutiae being necessary, that random minutiae indeed convey fingerprint individuality giving rise to calling them also *characteristic minutiae*.

Contents

Chapter 1

Introduction

A *fingerprint* is the reproduction of the exterior appearance of the fingertip dermis. Fingerprints feature a *ridge line* pattern inducing an undirected *orientation field* usually exhibiting some singularities called *cores* (where neighbouring ridge lines go around an ending ridge) and *deltas* (where three ridge lines meet). These ridge line patterns and its induced features are the subject of investigation in fingerprint recognition and forensics.

The relevance of fingerprints has strongly increased since automatic authentication and identification by fingerprints has become increasingly popular in a wide variety of applications such as smartphones and internet banking on the commercial side and in border control on the governmental side. Also in the field of forensics fingerprints gain undiminished attention.

1.1 Motivation

For identification purposes, a fingerprint is usually reduced to a *minutiae* pattern consisting of points where the fingerprint's ridges end or fork. These point patterns have turned out to be sufficient to identify individuals with high accuracy and allowing for compression of an image to a 40–100 element point pattern, see e.g. Figure 1.1. Concerning the biological formation process of ridge lines and minutiae there is, to the best of our knowledge, yet only little known. Since the formation takes place during early gestational weeks, data acquisition is difficult and biological data is therefore scarce.

Kücken (2007) considered a model for the formation of the orientation field with its singularities and the ridge frequency, varying only within a small interval by expanding patterns satisfying suitable partial differential equations. The author notices, that in simulations minutiae *mostly occur in two circumstances*: when ridge lines diverge with new ridges inserted and when almost parallel ridge lines meet. He further observes that minutiae positions were quite sensitive to initial conditions and Kücken & Champod (2013) explained this biologically by *small differences in the Merkel cell distribution.* These (random) differences can be very subtle as Newman (1930) noted much earlier: *There are, however, numerous instances in which the prints of two of more homologous fingers are so nearly identical as to be indistinguishable to the naked eye. [...] it is possible only by using considerable magnification to discover differences in the branching of ridges and breaks in ridge continuity. Differences of this sort, however, are certain to be found, and afford an easy means of identification.*

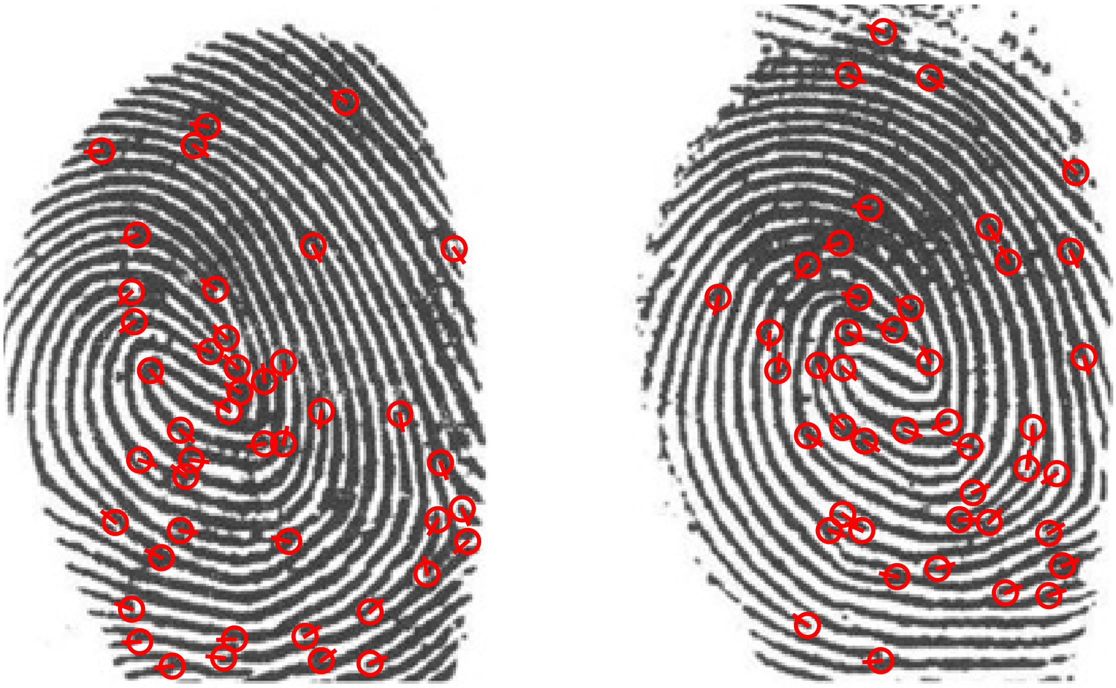


Figure 1.1: *Two fingerprints from monozygotic twins and marked minutiae patterns (red) containing 44 minutiae each. Images from Newman (1930), labelled 14_a and 14_b.*

Inspired by this, we focus theoretically and empirically on the interaction of minutiae, orientation field (OF) and ridge frequency (RF), and argue that orientation field divergence and ridge frequency changes geometrically necessitate minutiae which we hence call *necessary* minutiae. Statistical analysis endorses the above mentioned observations of additional *random* minutiae which are independent of the underlying smoothed OF and RF, cf. Figure 1.1 which shows prints of monozygotic twins from Newman (1930). For imprints with similar OF and RF leading to similar necessary minutiae point patterns, these random minutiae may (as it turns out) carry individuality information and for this reason, we also call them *characteristic* minutiae.

The theory of spatial point processes provides suitable tools to capture the complex interaction structure of minutiae patterns which is influenced by the OF and RF (leading to an inhomogeneous spatial distribution) and the discrete structure of the epidermis (leading to repulsion of points). Within this framework, minutiae patterns are assumed to consist of two sub-patterns $\eta \sim \mathrm{H}$ and $\xi \sim \Xi$ where η models the minutiae we can explain using OF and RF information and ξ captures the additional noise minutiae. Both are assumed to be random to include also external influences during the acquisition process such as manner of imprinting, dryness/ wetness of the finger, acquisition medium/ resolution or distortion of the finger.

The usage of this model comes along with certain challenges since classical approaches, both from frequentist and Bayesian statistical analysis, cannot be applied without further ado. Intractable normalising constants in the density and the lack of knowledge of the allocation of the points to the two processes require the application of Monte Carlo methods. We provide algorithms which solve for a proposed model the parameter estimation problem in a classical setting and in a Bayesian sense, the latter meaning that we estimate the posterior distribution of the model parameters given an observed minutiae pattern. This includes parameters of the distribution of Ξ and H. Moreover, estimating the distribution of a label vector encoding the assignment of the minutiae to the two sub-processes can be obtained following the Bayesian framework, however is not possible with the proposed frequentist method.

In application to manually (re-)marked fingerprints our minutiae separating algorithm (MiSeal) finds the presence of random minutiae. Furthermore, in an exemplary analysis of the two imprints of the twins from Figure 1.1 featuring similar OF and RF, we find that these minutiae are indeed characteristic: Without them the minutiae patterns become more similar.

1.2 Literature overview

The analysis of fingerprints has a long history, cf. (Scientific Working Group on Friction Ridge Analysis et al., 2011, Chapter 1). The interest in fingerprint data, especially for usage in modern authentication procedures, arose due to the consensus that fingerprints are distinct[1] and persistent for every individual, are easily collectible and can hence be used for identification purposes, cf. Galton (1892).

[1] so far there have never been observed fingerprints of different fingers exhibiting the same minutiae pattern and it is very unlikely to occur, cf. Pakanti et al. (2002); Page et al. (2011)

Fingerprint image processing and matching The procedure of comparing two imprints of fingers is called *matching* and is even nowadays still occasionally done by hand (e.g. when fingerprints are used as piece of evidence in a penal process). Automated matching algorithms launched in the late 1990s and 2000s. Most methods, however, are well kept proprietary secrets of commercial firms. A publicly available matching scheme based on the minutiae pattern's geometry was proposed by Cappelli et al. (2010a,b, 2011). Forbes et al. (2014) presented a Monte-Carlo based algorithm for computing the marginal likelihood ratio for the hypothesis that two observed prints originate from the same finger against the hypothesis that they originate from different fingers using marked point processes. Alberink et al. (2014); Neumann & Ausdemore (2019, 2020) critically discuss likelihood-ratio based matching methods and their applications in forensics.

Modern matching algorithms typically rely on minutiae templates marked by a minutiae extractor beforehand from digital fingerprint images, cf. Bansal et al. (2011). These images have to be of sufficiently good quality to produce reasonable results. Hence, preprocessing techniques are usually applied. The literature to this is rich; renowned approaches have been proposed by Hong et al. (1998); Chikkerur et al. (2007); Gottschlich (2011); Thai et al. (2016) to name just a few. However, hardly any of these papers provide corresponding software which was hence developed independently for this thesis. More recent algorithms often rely on machine learning based techniques, see e.g. Nixon & Aguado (2019); Zhao et al. (2020) for an overview. To produce good results, these algorithms, however, require to be trained on large data bases which are in the setting of fingerprints typically not available due to privacy constraints. Moreover, Tams et al. (2015) point out an application of minutiae templates in cryptography called *fuzzy vaults*.

Formation and individuality As already mentioned in Section 1.1, the biological formation process of minutiae, although already researched cf. e.g. Babler (1991); Kücken & Newell (2005); Kücken (2007); Kücken & Champod (2013); Burger et al. (2018); Düring et al. (2019), is still widely an open question. As a starting point, minutiae patterns of very similar fingerprints were studied in the literature to identify common features. Fingerprint similarity is expected to be largest amongst persons with close genetic information, hence twins' fingerprints (especially monozygotic) are often considered to this end, see e.g. Newman (1930); Cummins & Midlo (1976). Indeed, similarity has been established concerning fingerprint class/ type. Regarding the minutiae configurations, Jain et al. (2002) state that even fingerprints of identical twins can be distinguished using a minutiae-based matching algorithm with only slightly lower accuracy than non-twins, suggesting that also further factors play a role during formation of minutiae.

Relation of minutiae to fingerprint features The connection between minutiae locations and other fingerprint features, however, has gained only little attention in the literature. The observation that minutiae cause high local divergence of the ridge flow field has been used in Nikodémusz-Székely & Székely (1993) to locate them. To the best of our knowledge, including divergence information in minutiae matching has not been considered in the literature so far.

Point pattern separation Point pattern separation problems were previously considered in Cressie & Lawson (2000); Walsh & Raftery (2002) for separating two Poisson processes. Redenbach et al. (2015); Rajala et al. (2016) consider a separation problem for separating a homogeneous Strauss point process from a homogeneous Poisson point process using *Markov-Chain-Monte-Carlo* (MCMC) and variational Bayes methods, respectively. These algorithms served as a starting point for a more advanced method that enjoys better mixing and is able to deal with inhomogeneities in the Strauss process.

Fingerprint generation The development of algorithms to generate artificial, realistically looking fingerprints has experienced a sharp rise in popularity recently especially since machine learning based methods became very powerful. First steps have been made by Cappelli et al. (2002, 2004) which Gottschlich & Huckemann (2014) proved to be well separable from real fingerprints. Imdahl et al. (2015) propose a method based on Gabor filters, however makes the non-realistic assumption of a constant ridge frequency. More recent methods rely on variational autoencoders or generative adversarial networks, see e.g. Bontrager et al. (2018); Cao & Jain (2018); Minaee & Abdolrashidi (2018); Sadegh Riazi et al. (2020); Wyzykowski et al. (2020). The literature to this is evolving quickly. For a comparison of recent algorithms we refer to Mistry et al. (2019). A major bottleneck, however, seems to be the formalisation of realness and its implementation in the cost function of the networks. Moreover, Mistry et al. (2019) mentions that all models exhibit weak performance concerning realistic minutiae distributions. The analysis of the distribution of real minutiae point patterns is subject of this thesis.

1.3 Outline

This thesis is structured as follows. In Chapter 2 we introduce the terminology required for fingerprint recognition. We outline important features of a fingerprint and assess notorious issues with the considered data. Moreover, we have a look at minutiae patterns which are a central part of this work.

The main results of this thesis are discussed in Chapter 3. Here, we provide mathematical essentials on vector analysis and propose a mathematical model for the emergence of minutiae based on a fingerprint's orientation field and ridge frequency. A statistical analysis reveals that this model does not capture all the minutiae, but there is a certain amount of minutiae in a fingerprint not explained just by its orientation field and ridge frequency. We show later in Chapter 7 that the minutiae "excess" bear the potential of improving matching algorithms and give rise to a certain characteristicness of a fingerprint making them more typical for a certain individual. We aim on identifying these minutiae in a separation model, provide an algorithmic scheme to do so and analyse the fitted model on some high quality data.

Chapter 4 provides the mathematical framework for modelling the minutiae patterns as random point patterns. We introduce basic concepts for descriptive statistics of point processes and provide algorithms for simulation from fitted models. Moreover, we lay the mathematical foundation for parameter inference for superpositions of point processes. To be highlighted should be the algorithm discussed in Section 4.5 for computing a maximum pseudo-likelihood estimate of the model parameters. To the best of our knowledge, this is the first algorithm to solve parameter inference problems in the superimposed model from a frequentist point of view.

In Chapter 5 we introduce the reader to Bayesian inference and applications to point patterns. We point out issues arising when we consider Gibbs point processes whose density includes a normalising constant which is typically not available in practice. A central part of this chapter will be Markov Chain Monte Carlo (MCMC) algorithms which are employed to estimate a so-called posterior distribution of the model parameters, i.e. the distribution of these parameters after observing some data and including some prior knowledge which we have before observing the data.

The algorithms used for this work are presented in Chapter 6. Here we discuss all methods, issues and parameter choices for feature extraction of a fingerprint image. Moreover, we point out implementation details for the MCMC algorithm used for separation and link to our published software. The application of our algorithms to simulated and real fingerprint data is subject of Chapter 7.

To conclude, Chapter 8 summarises our achievements, still existing challenges and provides an outlook for further research.

Note, that the introduction of certain basics is outsourced to the Appendix A. There, we provide terminology of concepts which are not in the centre of our work, however are used and might be unfamiliar to the reader. We also like to remark that parts of Section 3.3, Section 3.4, Section 4.4, Section 6.4 and Chapter 7 have been submitted for publication in Wieditz et al. (2021) and are presented with minor changes.

Chapter 2

Fingerprint recognition and image processing

Biometric authentication and identification systems nowadays often rely on fingerprint information. A major reason for this is that fingerprints are regarded to be unique for every individual, they are easily collectable and provide a good level of security. In this chapter, we will introduce the reader to the thesis-relevant terminology and features considered in fingerprint recognition. For a more detailed overview see Maltoni et al. (2009).

2.1 Fingerprint images and global features

A *fingerprint* is the reproduction of the exterior appearance of the fingertip epidermis. The most evident structural characteristic of a fingerprint is a pattern of interleaved *ridges* and *valleys*. Fingerprints are usually available to us in the form of digital images. From a mathematical point of view one can see a *fingerprint image* as a function from a two dimensional grid to a pixel space or as a matrix with entries of the pixel space. We assume this pixel space to be either $\{0, 1, \ldots, 255\}$ (8-bit grey-value images) or $\{0, 1\}$ (binary images), cf. Figure 2.1a and 2.1b. In a fingerprint image ridges are usually dark whereas valleys are bright, cf. Maltoni et al. (2009).

Fingerprint images can be acquired in several ways. In law enforcement applications one often uses digitised *inked imprints* where the subject's finger is either pressed or rolled on a paper card which is then scanned using a general purpose scanner. The possibility of rolling the fingerprint allows to capture the fingerprint in its entirety. Nowadays, fingerprints images are usually obtained as *live* images using a digital scanner; no ink is required. Depending on the technique, the individual has to e.g. *touch* a surface (e.g. a glass prism), *sweep* its finger over a sensor or the finger is photographed by a digital camera from a short distance (*direct reading*). We refer to (Maltoni et al., 2009, Chapter 2) for a detailed discussion about fingerprint sensing technologies.

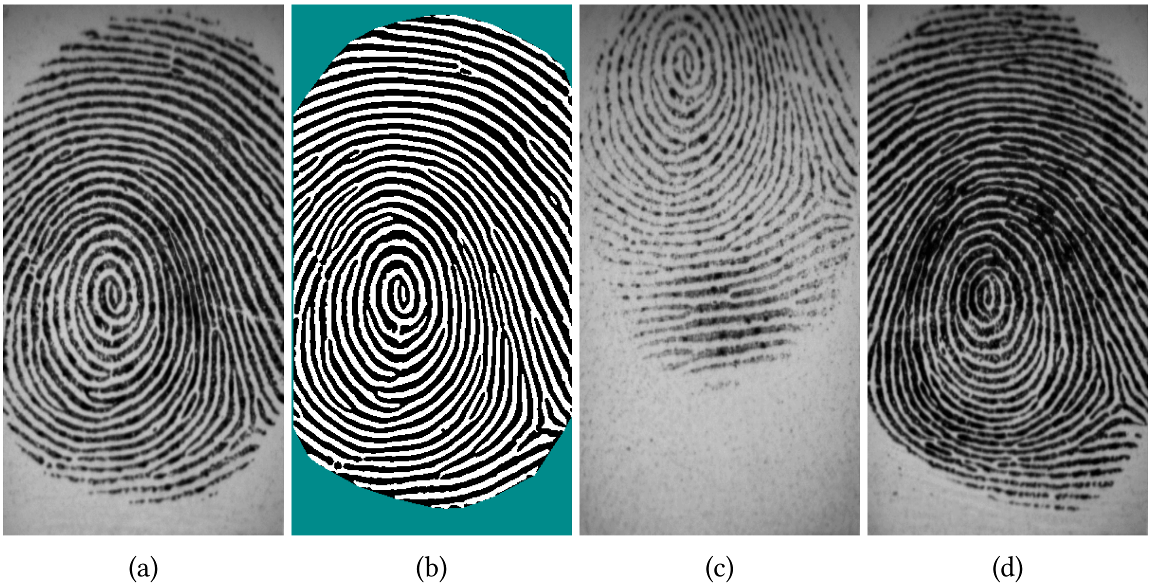
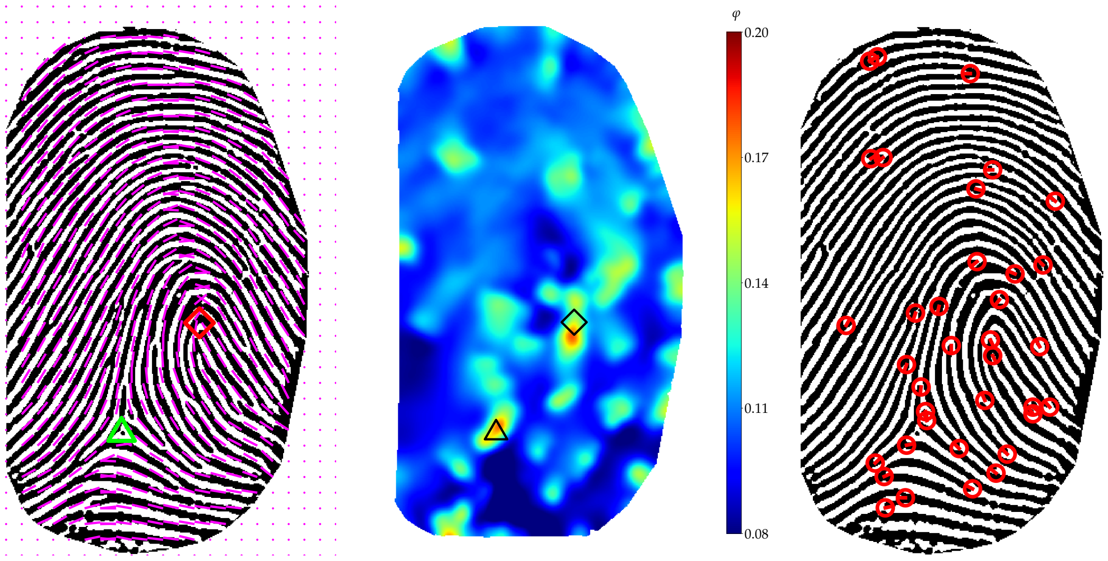


Figure 2.1: *From left to right: Example of an 8-bit grey-valued fingerprint image (a) (imprint 7_6 of DB2 in FVC2002 from Maio et al. (2002)) and its binarised version (b). The area not belonging to the region of interest (ROI) is indicated in turquoise. Although looking different, image (c) (imprint 7_5) corresponds to the same finger as (a) (intra-class variability) whereas (d) is an image from a different finger (imprint 101_5) looking similarly to (a) (inter-class similarity).*

The most typical application of fingerprint images is for the purpose of *verification* or *identification* of an individual. Here, verification means that we compare a fingerprint image of an individual with one (or several) previously captured reference images. In contrast, an identification system recognises an individual by searching an entire database for a *match* between the captured image and a stored one. The process of comparing two fingerprint images with the aim of computing a similarity score is referred to as *fingerprint matching*.

Images of fingerprints can exhibit the problem of small differences between impressions of different fingers (low inter-class variance, see Figure 2.1a and 2.1d) and large differences between images of the same finger (large intra-class variance, see Figure 2.1a and 2.1c). Hence, beneath the actual image additional features are used to make fingerprints more distinguishable. These features can be ordered hierarchically into three levels depending on the scale at which the fingerprint is considered. Hereby the robustness of the features decreases and their distinctiveness increases with the level. Level 1 (or global) features can be extracted even from noisy images and they are mostly used to align or classify fingerprints. Level 2 (or local) features can be reliably extracted as long as the quality of the image is not too bad. These features, especially minutiae point patterns (cf. Section 2.2), are mainly used for matching, since they have a good trade-off between robustness and distinctiveness. Level 3 features are hard to extract and are easily contaminated by noise, however, they are highly distinctive and can already be used for matching, if an image only contains a rather small part of the whole fingerprint.

Typical global features induced by the overall global ridge flow pattern are the orientation field and the ridge frequency. The *orientation field (OF)* assigns at each point (x, y) the angle $\theta(x, y)$ that the fingerprint ridges, crossing through an arbitrarily small neighbourhood around that point, form with the horizontal axis. Since this is only possible on a scale large enough to identify the ridges, this is clearly a global feature. Note, that since fingerprint ridges are not directed, this angle is only well-defined up to rotation by 180° and is hence is an element of, say, $[-\frac{\pi}{2}, \frac{\pi}{2})$. We store the local orientation of a fingerprint image on a lattice induced by the pixel grid in a matrix $\mathfrak{O} = (\theta(i, j))_{i,j}$ and refer to this matrix as the *orientation image*. An example is depicted in Figure 2.2a.



(a) (b) (c)

Figure 2.2: *Global and local features of a loop-type fingerprint from (Maio et al., 2002, DB2, image 105_2). Left: Orientation image (violet) and estimated singularities (green and red). ◇ denotes a core, Δ a delta. Middle: Ridge frequency image as heat map (red indicates high values whereas blue means low ridge frequency). Right: Minutiae pattern (red).*

The orientation field usually features discontinuities, i.e. points at which a direction cannot be assigned in the manner described above. These discontinuities, so-called *singularities*, are classified into cores and deltas. Penrose (1969), defines them as follows. A *core* (or *loop*) is formed when the parallel field turns through 180°, i.e. the field turns around and meets itself, cf. the diamond (◇) in Figure 2.2a. A *delta* (or *triradius*) is a point where three ridges meet, cf. the triangle (Δ) in Figure 2.2a. We will formalise this using the Poincaré index in Chapter 3. If not explicitly stated otherwise, we will adhere to the notation that cores and deltas are indicated with diamonds (◇) and triangles (Δ) for the rest of this thesis.

Using singularities, fingerprints are classified into classes, cf. Figure 2.3, such as loops (making up about 65.5% of the population's fingerprint patterns), whorls (about 27.9%) and arches (only about 6.6%); numbers from Wilson et al. (1994) which analysed the data base Watson & Wilson (1992). A typical application of this classification in fingerprint matching is to reduce the size of the database to be searched.

Orientation fields have been widely researched for a long time. Global mathematical models go back at least to Sherlock & Monro (1993). They introduced the *zero-pole model*, which uses just the position of the singularities to synthesise a fingerprint's orientation image. Improvements of this model were made by Vizcaya & Gerhardt (1996) by using piecewise linear approximations around singularities to adjust the zero and pole's behaviour. A non-linear model based on quadratic differentials was proposed by Huckemann et al. (2008), extending the zero-pole model and introducing control parameters with clear geometrical meaning such as e.g. singularity positions, cf. also Remark 3.5. Apart from these, there exist a variety of further approaches for modelling orientation fields. We refer to (Maltoni et al., 2009, Section 3.2) for a more detailed overview.

The *local ridge frequency (RF)* $\varphi(x, y)$ in a point (x, y) is the number of ridges per unit length (for this thesis we measure lengths in pixels) along a hypothetical line segment centred at (x, y) and orthogonal to the local ridge orientation $\theta(x, y)$ in (x, y), cf. (Maltoni et al., 2009, Section 3.3). It varies not only across different fingers but may also do so on the same finger, cf. Figure 2.2b or Gutiérrez-Redomero et al. (2008); Nayak et al. (2010) for an analysis on larger data bases. Closely related is the *inter-ridge distance* which is the distance from a given ridge to a neighbouring one. It is locally defined as the length of the segment connecting the centres of the two ridges along the line perpendicular to the first one, cf. Kovacs-Vajna et al. (2000), and is hence the reciprocal of the ridge frequency. As the local orientation is in practice only available on a digital lattice, so is the local ridge frequency. It is stored in a matrix, called the *frequency image* $\mathfrak{F} = (\varphi(i, j))_{i,j}$.

Whereas the extraction of global features of good quality images is already largely resolved, it is still an open issue for medium and low-quality fingerprint images. An overview about the literature of algorithms for the computation of orientation images, singularities, and ridge frequency images is provided in (Maltoni et al., 2009, Sections 3.2, 3.3 and 3.5). The

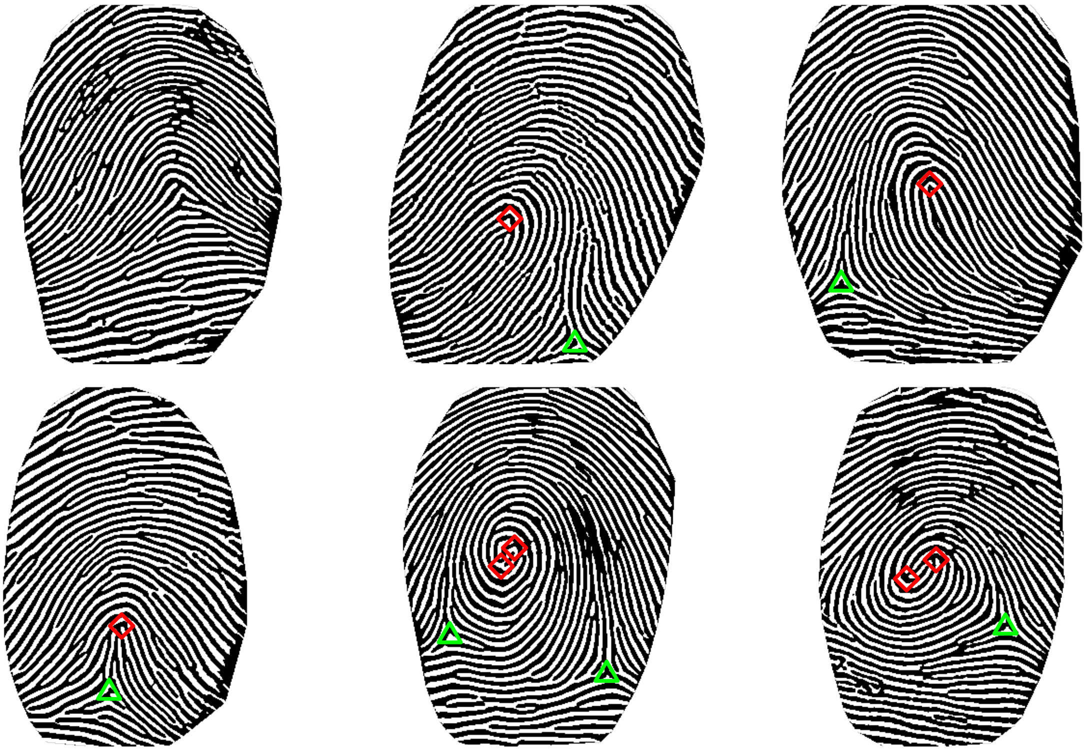


Figure 2.3: *Fingerprint classes (from left to right): arch, left loop, right loop (top); tented arch, whorl, twin-loop (bottom). Note that there is an invisible delta further to the left of the last image. Original images taken from Maio et al. (2002).*

algorithms used for our purposes require sufficiently good image quality and are presented in Chapter 6.

In practice, it is moreover often useful to separate the fingerprint image into the foreground or *region of interest (ROI)*, cf. Figure 2.1b, containing fingerprint information, and the *background* (not containing fingerprint information), to avoid feature extraction from noisy areas which often belong to the background. This procedure is called *segmentation*. Some authors use this term for the transformation of a greyscale image into a black and white image. We will refer to that procedure to as *binarisation* to distinguish between the two terms. For further details on segmentation and binarisation we refer to (Maltoni et al., 2009, Section 3.4), Thai et al. (2016) and the references therein.

2.2 Local features and fingerprint matching

For identification purposes global features are often not distinctive enough. Modern automatic systems for fingerprint comparison use hence also local features such as minutiae.

Minutiae (from the Latin word *minutus* meaning minute detail) are referred to discontinuities of the ridge lines such as ridge endings or bifurcations, cf. Figure 2.2c. The analysis of minutiae goes back at least to Galton (1892). In forensics, they are further distinguished but for our work we will adhere to the two types (endings and forks) mentioned above. A good quality fingerprint typically contains about 40–100 minutiae whereas in latent or partial prints, e.g. taken from crime scenes, the number is much less (approximately 20 to 30), cf. Zaeri (2011). However, the spatial and angular coincidence or correspondence of only a small number of minutiae (e.g. 12–15) is often sufficient to claim with high confidence that two imprints originate from the same finger.

To obtain good matching results a reliable *extraction* of fingerprint minutiae is vital. Common software for this purpose are for example Verifinger or Innovatrics on the commercial side or open source algorithms such as Mindtct (short for *minutiae detector*), cf. Garris et al. (2001), and FingerJetFX OSE[2]. Minutiae detection has already been widely researched, however, is still an open issue for medium and bad quality imprints. For an overview of the literature we refer to (Maltoni et al., 2009, Section 3.7) and Bansal et al. (2011). The minutiae patterns used for our work are hand-marked to extract all true minutiae and to avoid extraction of *spurious minutiae*, i.e. minutiae which have been automatically marked but erroneously.

The aim of fingerprint matching is, given two fingerprint images, to determine whether they belong to the same finger (binary decision) or to return a degree of similarity (e.g. a score between 0 and 1). Matching algorithms usually work with an extracted minutiae template, sometimes augmented with various features from the fingerprint such as the class or a local direction (pointing into the direction of the newly inserted ridge or newly inserted valley as per ISO/IEC 19794–2:2011), cf. Figure 1.1 or Figure 2.2c, where the direction of the minutiae is indicated with a bar. Hence a reliable preprocessing algorithm is required. Even for good quality images, matching has turned out to be quite challenging due to the high intra-class variance (see Figure 2.1) caused e.g. by displacement, rotation, distortion or partial observation of the finger. In the literature, a large number of algorithms have been proposed. For an overview, we refer to (Maltoni et al., 2009, Section 4.1). For our work we use the *minutiae cylinder code*, a matching algorithm proposed by Cappelli et al. (2010a) and enhanced by Cappelli et al. (2011) which turned out to perform reasonably well for our purposes.

It turns out that there is a relation between orientation image, ridge frequency image and minutiae. In Feng & Jain (2011) and Cao & Jain (2015) it was shown that the origi-

[2] https://github.com/FingerJetFXOSE/FingerJetFXOSE

nal greyscale image, and hence also orientation and ridge frequency image, can be reconstructed to large extent just from the minutiae template (i.e. minutiae locations and orientation). A central question of this thesis will be whether the minutiae pattern can be fully explained by the orientation and frequency image. This will be the subject of Section 3.3.

Chapter 3

Necessary and random minutiae

This chapter is dedicated to the derivation of an analytical relation between the estimated number of minutiae in a subregion of a fingerprint's region of interest, its orientation field and its ridge frequency. We like to outline this connection with a motivating example.

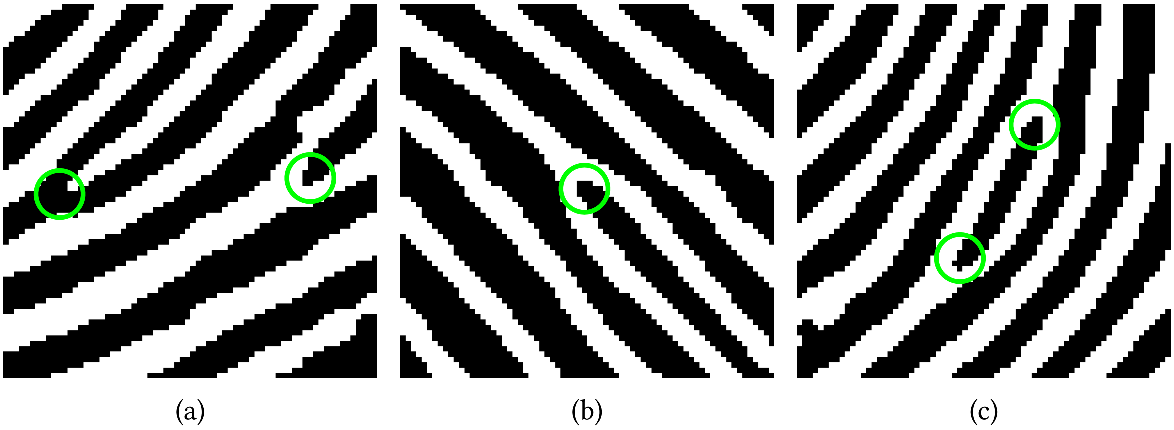


Figure 3.1: *Divergence of orientation field (a) and ridge frequency (b) resulting in minutiae (green circles). Panel (c) shows a case of minutiae not captured by divergence of ridges on the scale considered. Image sections obtained from Figure 2.1b.*

Example 3.1. Figure 3.1 shows three typical situations in which minutiae occur. On a small scale, one can determine the number of minutiae via ridge counts. In Figure 3.1a for instance, we have 7 ridges "entering" the patch on the lower and left side and 9 ridges "leaving" the patch on the upper and right side, resulting in $|9-7| = 2$ minutiae. Similarly, in Figure 3.1b we have 6 ridges "entering" the patch on the upper and left side and 7 ridges "leaving" the patch on the lower and right side, resulting in $|6 - 7| = 1$ minutia. While in Figure 3.1a the occurrence of new ridges can be mainly reduced to the divergence of the ridge lines, the reason in Figure 3.1b seems to be the thinning (or widening) of the ridges.

These phenomena can be mathematically described via the divergence of the orientation field and the ridge frequency, respectively, leading to the model proposed in this chapter.

On the given scale of observation, the minutiae in Figure 3.1c cannot be captured with the ridge counting approach; they carry information beyond the orientation field and ridge frequency. These minutiae often belong to one of the rare types (e.g. *lakes*, *islands*, *spurs*, *crossovers* or like here, an *independent ridge*), see also (Maltoni et al., 2009, p. 99), and are highly distinctive, making them very interesting from a forensic point of view.

Our aim is hence not only to estimate the number of minutiae in a certain part of the fingerprint explained by orientation field and ridge frequency information, but also to identify minutiae which exist beyond these two quantities. The latter will be part of Chapter 4 and Chapter 7.

In Section 3.2 we model the orientation field (OF) and ridge frequency (RF) in the context of two-dimensional vector fields; we highlight a link between the divergence of OF and RF and the occurrence of minutiae of a fingerprint induced by this model in Section 3.3. In Section 3.4 we validate the model for a number of high quality fingerprints. In the following Section 3.1 we recall some important preliminaries on vector analysis.

3.1 Preliminaries on vector analysis

In the following, let $\mathfrak{X} \subseteq \mathbb{R}^2$ be a compact set that models the *region of interest* (i.e. the part of the image containing the fingerprint), and assume that $A \subseteq \mathfrak{X}$ is a subset of $\mathfrak{X}$ with *piecewise smooth boundary*, i.e. ∂A is the image of a piecewise smooth curve $\gamma : [a, b] \to \mathbb{R}^2$ for $-\infty < a < b < \infty$. For this thesis, all vectors are assumed to be column vectors.

Definition 3.2 (Vector field, divergence)**.** A (vector-valued) mapping

$$\vec{F} : \mathbb{R}^2 \to \mathbb{R}^2, \qquad \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} F_1(x,y) \\ F_2(x,y) \end{pmatrix}$$

with $F_i(x, y) : \mathbb{R}^2 \to \mathbb{R}$, $i = 1, 2$ is called *vector field* . It is referred to as k-times *differentiable*, $\mathcal{C}^k$ or *smooth* on $A \subseteq \mathbb{R}^2$ if its components $F_i : A \to \mathbb{R}$ are k-times *differentiable*, $\mathcal{C}^k$ or *smooth*, respectively. For a differentiable vector field $\vec{F} : A \to \mathbb{R}^2$ we define via

$$\operatorname{div} \vec{F} : A \to \mathbb{R}, \qquad \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \frac{\partial F_1}{\partial x}(x, y) + \frac{\partial F_2}{\partial y}(x, y)$$

its *divergence*. Denoting by $J\vec{F}$ the Jacobian of $\vec{F}$, we have $\operatorname{div} \vec{F}(x, y) = \operatorname{tr} J\vec{F}(x, y)$.

Remark 3.3 (Product rule). The divergence operator

$$\operatorname{div} : \mathcal{C}^1(\mathbb{R}^2 \to \mathbb{R}^2) \to \mathcal{C}(\mathbb{R}^2 \to \mathbb{R}), \quad \vec{F} \mapsto \operatorname{div} \vec{F}$$

is linear and satisfies for differentiable functions $\vec{F} : A \to \mathbb{R}^2$ and $\varphi : A \to \mathbb{R}$ the following product rule:

$$\operatorname{div}\left(\varphi \vec{F}\right) = \varphi \operatorname{div} \vec{F} + \left\langle \nabla\varphi, \vec{F} \right\rangle \tag{3.1}$$

where $\nabla\varphi$ denotes the *gradient* of φ, $\langle\cdot,\cdot\rangle$ is the *Euclidean scalar product* and (3.1) is to be understood point-wise.

For a differentiable vector field, the total divergence within A can be computed via a line integral over the boundary ∂A. Recall, that for an integrable function $f : A \to \mathbb{R}$ the *line integral* (of first kind) of f along a positively oriented[3] boundary curve $\gamma : [a, b] \to \partial A$ is defined as

$$\int_{\partial A} f(z) \,\mathrm{d}z := \int_{\gamma} f(z) \,\mathrm{d}z := \int_a^b f(\gamma(t)) \|\dot{\gamma}(t)\| \,\mathrm{d}t.$$

The divergence theorem provides now the following relation between the integrated divergence of a vector field $\vec{F}$ within a set A and a line integral over the boundary ∂A.

Theorem 3.4 (Gauss' divergence theorem). *Let $A \subseteq \mathbb{R}^2$ be a bounded, closed region with piecewise smooth boundary ∂A and denote by $\vec{n}$ the (piecewise well-defined) outwards pointing[4] unit normal vector $\vec{n} : \partial A \to \mathbb{S}^1$. Then, for a differentiable vector field $\vec{F} : A \to \mathbb{R}^2$ holds*

$$\iint_A \operatorname{div} \vec{F}(z) \,\mathrm{d}z = \int_{\partial A} \left\langle \vec{F}(z), \vec{n}(z) \right\rangle \,\mathrm{d}z. \tag{3.2}$$

Physically speaking, the volume integral over all the sources and sinks within A is equal to the net flow through the boundary ∂A.

Proof. For a proof we refer to (Amann & Escher, 2009, Theorem XII.3.15 and Remark XII. 3.16 (c)) or (Adams & Essex, 2018, Theorem 16.7) for a more direct formulation. ■

[3] when travelling along γ, the interior of A is always to the left

[4] i.e. for a parametrisation γ of ∂A holds $\langle\dot{\gamma}(t), \vec{n}(\gamma(t))\rangle = 0$ for all t and $\vec{n}$ points towards the complement A^c; in points of discontinuity of γ, $\vec{n}$ can be arbitrarily chosen

3.2 A model for global features of fingerprints

3.2.1 Orientation fields

For a fingerprint, we model the *orientation field* induced by its ridge line pattern by a mapping $\theta : A \to \mathbb{R}P^1$ from A to the *real projective line*, i.e. the space of orientations of lines in $\mathbb{R}^2$. In fact, $\mathbb{R}P^1$ contains angles between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$. Topologically, $\mathbb{R}P^1$ is a circle. We assume that θ is well-defined, non-vanishing and $\mathcal{C}^2$ apart from a finite number of *isolated singularities*.

For every orientation $\theta_0 \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right)$ there are two well-defined *directions*, namely the original orientation $\theta_0 \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right)$ interpreted as direction and the direction $\theta_0 + \pi$. In simply connected regions $A \subseteq \mathfrak{X}$ not containing any of the singularities, we can select directions from θ such that the vector field $\vec{F} := \vec{F}_A$ with

$$\vec{F} : A \to \mathbb{S}^1, \qquad z \mapsto \begin{pmatrix} \cos\theta(z) \\ \sin\theta(z) \end{pmatrix}$$

is twice differentiable, see Section 6.3. In presence of singularities the assignment of a continuous direction $\vec{F} = \vec{F}_{\mathfrak{X}}$ globally, however, is not possible, cf. (Sherlock, 2004, Theorem 5.3.3.1).

Remark 3.5 (Quadratic differentials)**.** Huckemann et al. (2008) presents a low-dimensional parametric model for orientation fields based on *quadratic differentials* (QD), i.e. trajectories $z : \mathbb{R} \to \mathbb{C}$ of the initial value problem

$$Q(z(t))\dot{z}(t)^2 = \alpha(t), \qquad z(0) = z_0 \tag{3.3}$$

for a smooth, positive function $\alpha : \mathbb{R} \to \mathbb{R}_{++}$ and a rational function $Q : \mathbb{C} \to \mathbb{R}_+ \cup \{+\infty\}$. Here, the ridge pattern is formed by considering solution curves to different initial values $z_0 \in \mathbb{C}$. More precisely, for a given differential equation $Q(z)\dot{z}^2 = \alpha$, $Q \neq 0, +\infty$, the orientation field is given via the solution curves as

$$\left\{ \left(z(t), \frac{\dot{z}(t)^2}{|\dot{z}(t)^2|} \right) \in \mathbb{C} \times \mathbb{R}P^1 : Q(z(t))\dot{z}(t)^2 = \alpha(t), Q(z(t)) \neq 0, \infty,\ t \in \mathbb{R} \right\}.$$

The authors state, that α only changes the parametrisation of the solution curves, however not their qualitative behaviour. Remarkably, in the QD model the singularities of the orientation field have a clear meaning with respect to Q: zeroes of Q result in deltas in the ridge pattern, poles result in cores and whorls. This gives rise to an alternative proof of Penrose's formula, cf. Penrose (1969):

$$\text{no. of deltas} + 1 = \text{no. of loops} + \text{no. of fingers}. \tag{3.4}$$

In particular, for one considered fingerprint the number of deltas equals the number of cores, even though in applications some singularities may not be observed within the captured fingerprint region.

Fitting a QD to a given fingerprint is possible provided the singularities are given. However, QDs often deviate from the ground-truth ridge flow on a local scale due to the small number of model parameters. An extended version of the QD model introducing local correction points is presented in Gottschlich et al. (2016). Both approaches, however, rely on a reliable extraction of singular points. Without knowledge about the OF this approach can hence only be applied by letting an expert mark the singularities by hand; automation is thus only possible to a limited extent. We will discuss the computation of OF and its singularities in Chapter 6 and present a fully-automated method there.

For the rest of this section, assume the orientation field to be known. Then, the presence of singularities of an OF is indicated by the Poincaré index, cf. e.g. (Fulton, 2013, Chapter 7).

Definition 3.6 (Poincaré index). Let $\theta : A \to \left[-\frac{\pi}{2}, \frac{\pi}{2}\right)$ be an orientation field which is differentiable except possibly in $z_0 \in A$. Moreover, let $\gamma : [0, 2\pi] \to \partial A$ be a positively oriented, piecewise smooth, closed curve such that z_0 is always to the left of γ. Then, the *Poincaré index* of θ in z_0 is defined as

$$\operatorname{index}_\theta(z_0) := \frac{1}{2\pi} \int_0^{2\pi} \langle \nabla\theta(\gamma(t)), \dot{\gamma}(t) \rangle \, \mathrm{d}t.$$

Note, that the Poincaré index does not depend on the particular choice of the curve γ but only on the orientation field θ, cf. (Fulton, 2013, Lemma 7.1).

Henceforth, we define for a vector $z = (x, y)^\top \in \mathbb{R}^2$ its *argument* $\arg z := \angle(z, (1,0)^\top)$ as the angle z encloses with the x-axis which can be computed e.g. via the $\operatorname{arctan2}(y, x)$ function, see Section 6.1.

Example 3.7 (Poincaré index of orientation field singularities). We consider orientation fields $\theta : A \to \left[-\frac{\pi}{2}, \frac{\pi}{2}\right)$ on $A = [-1, 1]^2$ and are interested in the singularities of the induced vector field $\vec{F} : A \to \mathbb{S}^1$, $\vec{F}(z) = (\cos\theta(z), \sin\theta(z))^\top$.

(a) (Regular field) Let $\theta \equiv 0$ for $z \in A$. Then, $\vec{F} \equiv (1,0)^\top$ features a constant direction, cf. Figure 3.2a. Clearly, $\mathrm{index}_\theta(z) = 0$ for all $z \in A$, i.e. every point is a *regular point*.

(b) (Core) The field $\theta(z) = \frac{1}{2}\arg z$ models an orientation field with a *core* in the origin $z_0 = 0$. The corresponding vector field $\vec{F}(z) = \left(\cos\left(\frac{1}{2}\arg z\right), \sin\left(\frac{1}{2}\arg z\right)\right)^\top$ is depicted in Figure 3.2b. Note also the discontinuity of $\vec{F}$ on the negative real line. The derivative of θ is given as

$$\nabla\theta(z) = \begin{pmatrix} \frac{\partial}{\partial x}\frac{1}{2}\arctan\left(\frac{y}{x}\right) \\ \frac{\partial}{\partial y}\frac{1}{2}\arctan\left(\frac{y}{x}\right) \end{pmatrix} = \frac{1}{2}\frac{1}{x^2+y^2}\begin{pmatrix} -y \\ x \end{pmatrix} = \frac{1}{2\|z\|^2}\begin{pmatrix} -y \\ x \end{pmatrix}. \tag{3.5}$$

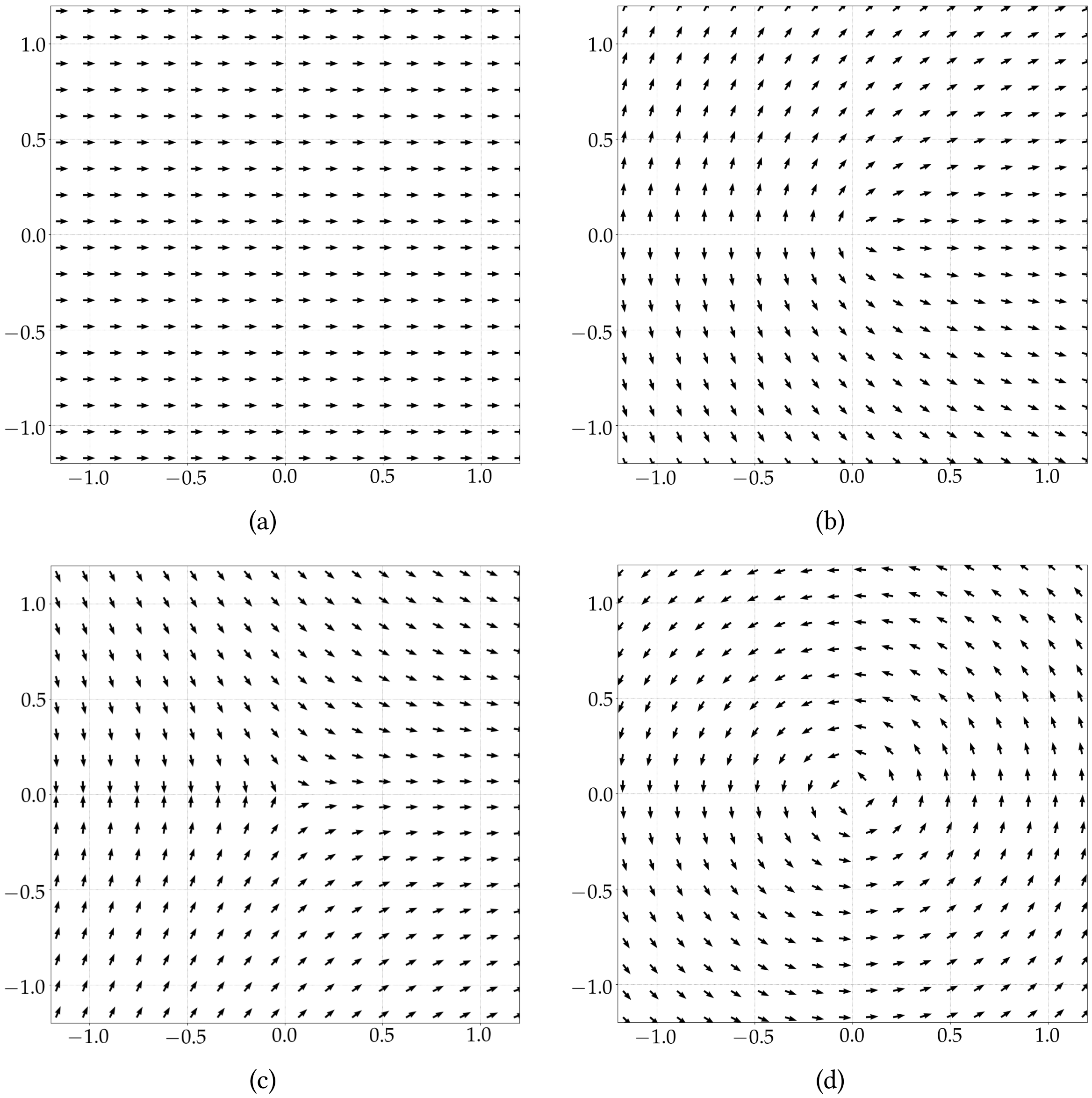


Figure 3.2: *Vector fields featuring no singularity (a) and singularities in the origin of core-type (b), delta-type (c) and whorl-type (d), respectively.*

Let $\gamma(t) = (\cos(t), \sin(t))^\top$ which is a positively oriented, smooth, closed curve enclosing $z_0 = 0$ and velocity $\dot{\gamma}(t) = (-\sin(t), \cos(t))^\top$. From (3.5) now follows with $\|\gamma(t)\| = 1$ that

$$\langle \nabla\theta(\gamma(t)), \dot{\gamma}(t) \rangle = \frac{1}{2}\left((-\sin(t))(-\sin(t)) + \cos(t)\cos(t)\right) = \frac{1}{2} \tag{3.6}$$

and hence, $\operatorname{index}_\theta(0) = \frac{1}{2\pi}\int_0^{2\pi} \frac{1}{2}\,\mathrm{d}t = \frac{1}{2}$.

(c) (Delta) The orientation field $\theta(z) = -\frac{1}{2}\arg z$ exhibits a *delta*-type singularity in $z_0 = 0$. Figure 3.2c shows the corresponding vector field

$$\vec{F}(z) = \begin{pmatrix} \cos\left(-\frac{1}{2}\arg z\right) \\ \sin\left(-\frac{1}{2}\arg z\right) \end{pmatrix} = \begin{pmatrix} \cos\left(\frac{1}{2}\arg z\right) \\ -\sin\left(\frac{1}{2}\arg z\right) \end{pmatrix}.$$

Note that θ is the negative of (b); $\vec{F}$, however, changes only its sign in the second component, leading to the different directional behaviour.

The linearity of $\nabla\theta$ and the scalar product implies that in the calculations (3.5) and (3.6) only the sign changes yielding $\operatorname{index}_\theta(0) = -\frac{1}{2}$.

(d) (Whorl) Let $\theta(z) = \arg z$. Then, θ has a *whorl*-type singularity in $z_0 = 0$. The corresponding vector field $\vec{F}(z) = (\cos(\arg z), \sin(\arg z))^\top$ circulates around the origin, see Figure 3.2d. Since $\nabla\theta(z) = \frac{1}{\|z\|^2}(-y, x)$ we obtain, analogously to (b) that $\operatorname{index}_\theta(0) = 1$.

In fact, the Poincaré index is always an integer multiple of $\frac{1}{2}$ and the following relation to singularities of orientation fields hold, cf. Sherlock & Monro (1993):

Theorem 3.8. *For an orientation field $\theta : A \to \left[-\frac{\pi}{2}, \frac{\pi}{2}\right)$ which is differentiable except possibly in $z_0 \in A$ holds:*

$$z_0 \textit{ is a } \left\{\begin{matrix} \textit{regular point} \\ \textit{delta} \\ \textit{core} \\ \textit{whorl} \end{matrix}\right\} \textit{ of } \theta \iff \begin{cases} \operatorname{index}_\theta(z_0) = 0, \\ \operatorname{index}_\theta(z_0) = -\frac{1}{2}, \\ \operatorname{index}_\theta(z_0) = \frac{1}{2}, \\ \operatorname{index}_\theta(z_0) = 1. \end{cases}$$

Moreover, Kawagoe & Tojo (1984) note that in a fingerprint Poincaré indices of higher order do not occur.

Bazen & Gerez (2002) pointed out, that, using Green's theorem, cf. Theorem A.1, the Poincaré index can be computed as a surface integral over the curl of the derivative of the orientation field. A mathematically rigorous proof is provided in the following Theorem 3.9. Recall, that the *curl* of a differentiable vector field $\vec{F} : A \to \mathbb{R}^2$ is defined as

$$\operatorname{curl} \vec{F} : A \to \mathbb{R}, \qquad \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \frac{\partial F_2}{\partial x}(x, y) - \frac{\partial F_1}{\partial y}(x, y).$$

Theorem 3.9. *Let $A \subseteq \mathbb{R}^2$ be a closed, compact set with piecewise smooth boundary and θ be twice differentiable except possibly in $z_0 \in A$. Then,*

$$\operatorname{index}_\theta(z_0) = \frac{1}{2\pi} \iint_A \operatorname{curl} \nabla\theta(z) \,\mathrm{d}z. \tag{3.7}$$

Proof. Let $\gamma : [0, 2\pi] \to A$ be a positively oriented, piecewise smooth, closed curve enclosing z_0 as in Definition 3.6. Under the given assumptions, Green's theorem, see Theorem A.1 on p. 151, applied to $\nabla\theta$ yields

$$\begin{aligned} \operatorname{index}_\theta(z_0) &= \frac{1}{2\pi} \int_0^{2\pi} \langle \nabla\theta(\gamma(t)), \dot{\gamma}(t) \rangle \,\mathrm{d}t \\ &= \frac{1}{2\pi} \iint_A \frac{\partial \nabla\theta_2}{\partial x}(x, y) - \frac{\partial \nabla\theta_1}{\partial y}(x, y) \,\mathrm{d}(x, y). \end{aligned} \tag{3.8}$$

Rewriting (3.8) we obtain

$$\operatorname{index}_\theta(z_0) = \frac{1}{2\pi} \iint_A \frac{\partial}{\partial x}\frac{\partial}{\partial y}\theta(x, y) - \frac{\partial}{\partial y}\frac{\partial}{\partial x}\theta(x, y) \,\mathrm{d}(x, y) \tag{3.9}$$

and hence the claim follows. Notice, that if A contains no singular point and θ is $\mathcal{C}^2$ on A, we can interchange the order of the derivatives in (3.9) yielding $\operatorname{index}_\theta(z_0) = 0$ immediately. However, if z_0 is a singularity, this is in general not true anymore. ■

3.2.2 Local ridge frequency

The fingerprint ridge pattern features moreover a ridge frequency which in a point $z \in \mathfrak{X}$ is the number of ridges per unit length along a hypothetical segment centred at z and orthogonal to $\theta(z)$, cf. Section 2.1. This frequency varies across different fingers and may also noticeably vary across different regions of the same fingerprint. Notably, women tend to have a slightly higher ridge frequency on average than men, see e.g. Gutiérrez-Redomero et al. (2008) for a study on 200 Caucasian men and women.

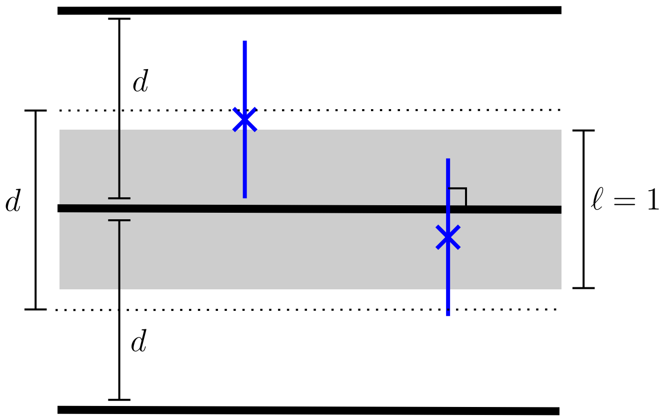


Figure 3.3: *Buffon's needle problem for a parallel grid of lines with distance d apart from each other. The centre $(\times)$ of the needles (blue) are thrown on the plane uniformly at random such that the needles land orthogonal to the grid. A needle crosses the line pattern if the centre lands in a band of width 1 centred around the grid lines (grey shaded for the middle grid line). The dotted lines indicate a strip of width d around the middle grid line.*

For our model, let $d : A \subseteq \mathfrak{X} \to (1, \infty)$ be a $\mathcal{C}^2$-function. Intuitively, d models the *inter-ridge distance* in a regular region. Note that d is bounded from below by one (unit) which is reasonable regarding the fact that two different ridge lines have at least a distance of one unit (e.g. pixels; otherwise they would be regarded as the same).

Consider now for $z \in A$ a lattice of parallel grid lines[5] with spacing $d(z)$ to both sides. Then, we define $\varphi(z)$ as the probability that a needle of unit length $\ell = 1$ thrown uniformly at random upon the grid such that the needle is orthogonal to the lattice cuts a grid line, see Figure 3.3. We call the function $\varphi : A \to [0, 1]$ induced in this way the *ridge frequency* (RF). This is a variation of the classical Buffon needle problem with preferred orientation of the needles. For an analysis of variations of the Buffon needle problem we refer to Watson (1978), Chung (1981) and the references therein.

As a matter of fact we have $\varphi(z) = \frac{1}{d(z)}$ for all $z \in \mathfrak{X}$ and hence φ is likewise $\mathcal{C}^2$ on $\mathfrak{X}$ apart from isolated singularities. To see this, we first note that, since the length of needle $\ell = 1 < d(z)$ is strictly smaller than the spacing of the grid, a needle can cut at most one line. Hence, we can restrict our analysis on a strip of width $d(z)$ around one grid line, cf. the dotted strip around the middle grid line in Figure 3.3. Looking on this illustration it becomes apparent that a needle crosses the middle line if and only if its centre is within $\frac{1}{2}$ units of either side of the strip. Adding $\frac{1}{2} + \frac{1}{2}$ from both sides and dividing it by the total width $d(z)$ of the strip yields $\varphi(z) = \frac{1}{d(z)}$.

[5] later on, these will be the fingerprint's ridge lines and the needle has to be thrown orthogonal to the local orientation

Note that for practical purposes, the grid is already given by the ridge lines. Thus, the orientation of the needles changes according to the local ridge orientation θ such that it is always orthogonal to θ. Moreover, if we assume a constant inter-ridge distance $d(z) \equiv d > 1$ on $\mathfrak{X}$ and thus $\varphi(z) \equiv \frac{1}{d}$, then for a curve γ of length $L(\gamma) \in (0, \infty)$ we have that

$$\frac{1}{L(\gamma)} \int_{\gamma} \varphi(z) \, \mathrm{d}z = \frac{1}{L(\gamma)} \left(\frac{L(\gamma)}{d} \right) = \frac{1}{d}$$

is the number of ridges crossing a curve orthogonal to the local ridge orientation at unit length, i.e. our model coincides with the intuitive definition from Chapter 2 in this case. A method for computing the ridge frequency in practice is presented in Section 6.2.

The mathematical concepts of orientation field θ and local ridge frequency φ motivate now the following definition. This is further illustrated in the Examples 3.11 and 3.12 further below.

3.3 A formula for necessary minutiae

Definition 3.10 (Necessary minutiae number)**.** Let $A \subseteq \mathfrak{X}$ be simply connected, compact with piecewise smooth boundary ∂A with (piecewise well-defined) outwards pointing normal $\vec{n} : \partial A \to \mathbb{S}^1$. Assume that A does not contain any of the singularities of $\vec{F}$ and let $\varphi : A \to (0, 1)$ twice differentiable. We call

$$m(A) := \left| \int_{\partial A} \varphi(z) \left\langle \vec{F}(z), \vec{n}(z) \right\rangle \mathrm{d}z \right| \tag{3.10}$$

the (usually non integer-valued) *number of geometrically necessary minutiae* in A, for short the *necessary minutiae number*.

As A contains no singularities, ridges near A carry a common directional flow induced by $\vec{F} = \vec{F}_A$, cf. Section 3.2.1. Then, the necessary minutiae number $m(A)$ counts the absolute difference of numbers of ridges entering A and leaving A, each weighted by the cosine of the angle between ridge and outwards pointing normal (counted fully if they intersect the boundary of A perpendicularly). Taking the absolute value of the difference provides independence of the particular flow direction $\vec{F}$. Thus $m(A)$ counts the number of minutiae *necessary* due to the geometry of the OF and the RF, see also Figure 3.4. Minutiae in A annihilating each other, e.g. due to a ridge beginning and ending in A, are not counted. Hence, Definition 3.10 is a mathematically tangible generalisation of the ridge counting approach presented in Example 3.1.

Example 3.11 (Orientation field divergence)**.** Let $\varepsilon > 0$ and consider $\mathfrak{X} = \{z = (x,y) \in \mathbb{R}^2 : \|z\| \geq \varepsilon\}$ and $\vec{F} : \mathfrak{X} \to \mathbb{S}^1, z \mapsto \frac{z}{\|z\|}$ pointing radially away from the origin. For $z_0 \in \mathfrak{X}$ and $\varepsilon < r < R$ consider an annular sector

$$A := \left\{z \in \mathbb{R}^2 : \left|\angle(z, z_0)\right| \leq \alpha, r \leq \|z\| \leq R\right\} \tag{3.11}$$

of opening angle $\alpha \in [0, \frac{\pi}{2}]$.

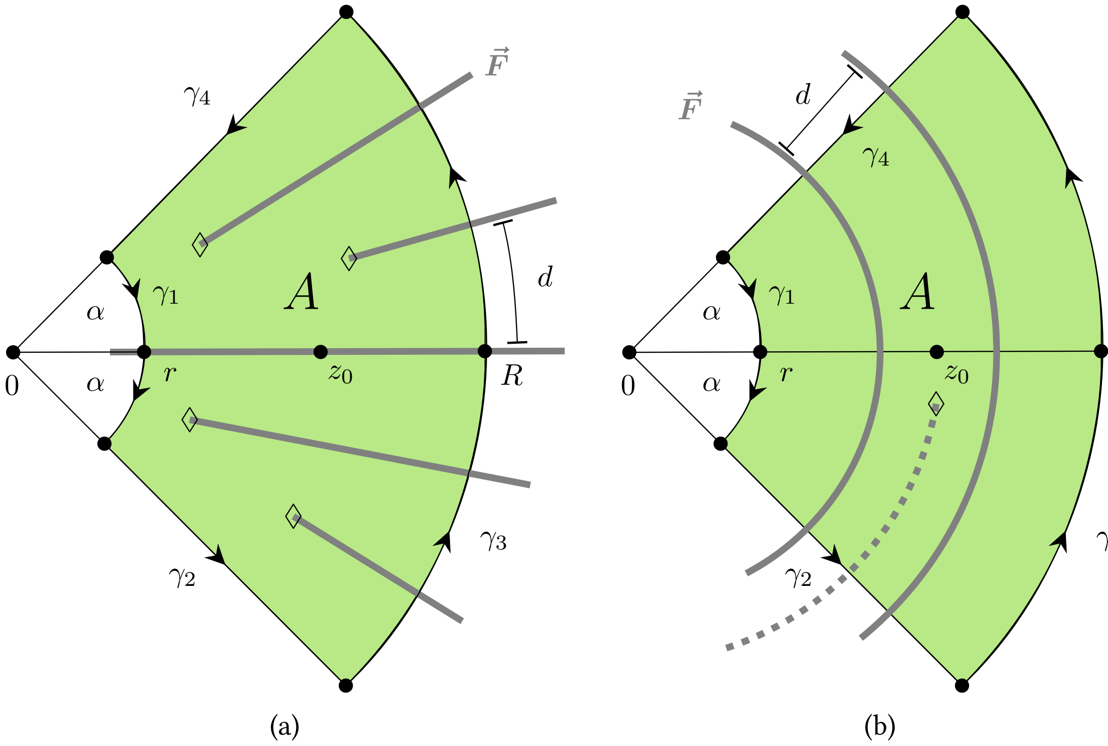


Figure 3.4: *Ridge pattern within an annular sector A (green) around z_0 (on the first axis between r and R) generated by the OF $\vec{F} : z = (x,y) \mapsto \frac{(x,y)}{\|z\|}$ (left) and the field $\vec{F} : z = (x,y) \mapsto \frac{(y,-x)}{\|z\|}$ (right), which is orthogonal to the field on the left. The number of minutiae in A is given by the number of new ridges emerging in A ($\Diamond$).*

Figure 3.4 shows the situation for a nearly constant inter-ridge distance $d(z) \approx d$, thus $\varphi(z) \approx \frac{1}{d}$, where one ridge line enters from the left and five ridge lines leave on the right, giving rise to four minutiae (marked with $\Diamond$) inside A. Along the circular arcs γ_1 (of length $2\alpha r$) and γ_3 (of length $2\alpha R$) the outwards pointing normal $\vec{n}$ of ∂A is first antiparallel and then parallel to the field, while on the radial arcs γ_2 and γ_4 the outwards pointing normal $\vec{n}$ of ∂A is orthogonal to the field. With $\varphi(z) = \frac{1}{d}$, this gives

$$m(A) = 2\alpha \frac{R - r}{d} .$$

Indeed, with

$$\frac{\partial F_1}{\partial x}(x,y) = \frac{\partial}{\partial x}\left(\frac{x}{\sqrt{x^2+y^2}}\right) = \frac{y^2}{(x^2+y^2)^{3/2}},$$
$$\frac{\partial F_2}{\partial y}(x,y) = \frac{\partial}{\partial y}\left(\frac{y}{\sqrt{x^2+y^2}}\right) = \frac{x^2}{(x^2+y^2)^{3/2}},$$

we have simply $\operatorname{div} \vec{F}(z) = \frac{1}{\|z\|}$, and in the presence of OF divergence only, introducing polar coordinates, we obtain

$$\begin{aligned}\iint_A \varphi(z)\, \operatorname{div} \vec{F}(z)\, \mathrm{d}z &= \frac{1}{d}\iint_A \frac{1}{\|z\|}\, \mathrm{d}z = \frac{1}{d}\int_r^R \int_{-\alpha}^{\alpha} \frac{1}{\rho}\left|\det\begin{pmatrix}\cos\psi & \sin\psi \\ -\sin\psi & \cos\psi\end{pmatrix}\right| \mathrm{d}\psi\, \mathrm{d}\rho \\ &= 2\alpha\,\frac{R-r}{d}\,.\end{aligned}$$

Assuming that there were fewer than four minutiae observed in Figure 3.4a, then fewer than five ridge lines would cross γ_3. This would necessitate a lower ridge frequency on γ_3 than on γ_1, yielding $\langle \nabla\varphi(z), \vec{F}(z)\rangle < 0 < \operatorname{div} \vec{F}(z)$, so that the OF divergence would be cancelled partially (or in total) by the RF divergence.

Example 3.12 (Ridge frequency divergence)**.** With $\mathfrak{X}$ and A from Example 3.11, consider now the field $\vec{F} : \mathfrak{X} \to \mathbb{S}^1, z = (x,y) \mapsto \frac{(y,-x)}{\|z\|}$, which is perpendicular to the field from Example 3.11, cf. Figure 3.4b. Since

$$\frac{\partial F_1}{\partial x}(x,y) = \frac{-xy}{(x^2+y^2)^{3/2}} = -\frac{\partial F_2}{\partial y}(x,y),$$

this field is divergence free and for constant ridge frequency $\varphi(z) = \frac{1}{d}$ we do not observe any minutiae in A, i.e.

$$m(A) = \left|\int_{\gamma_4} \varphi(z)\, \mathrm{d}z - \int_{\gamma_2} \varphi(z)\, \mathrm{d}z\right| = 0. \tag{3.12}$$

Indeed, now the field is orthogonal to the outwards pointing normals of ∂A along the circular arcs γ_1 and γ_3 while it is parallel and antiparallel, respectively, on the radial arcs γ_2 and γ_4, which are of equal lengths, so their contributions to $\varphi(z)\langle \vec{F}(z), \vec{n}(z)\rangle$ cancel.

If a minutia was observed within A, then it would be due to the RF divergence of the non-constant RF, namely of RF higher on γ_2 than on γ_4, as depicted with the dotted ridge in

Figure 3.4b. Assume for instance $\varphi|_{\gamma_4} \equiv \frac{1}{d}$ and $\varphi|_{\gamma_2} \equiv \frac{2}{d}$ i.e. the distance between ridges on γ_4 is d (cf. Figure 3.4b) but only $\frac{d}{2}$ on γ_2. Then, similarly to (3.12), we obtain

$$m(A) = \frac{2}{d}\int_{\gamma_4} \mathrm{d}z - \frac{1}{d}\int_{\gamma_2} \mathrm{d}z = \frac{R-r}{d}.$$

Hence, by doubling the ridge frequency, we obtain $\frac{1}{d}(R-r)$ new minutiae which corresponds to one minutiae for every new line inserted.

For a general case we can split the integral from (3.17) in the following way. Applying the divergence theorem, cf. Theorem 3.4 and the product rule (3.1), we obtain

$$\begin{aligned}\int_{\partial A} \left\langle \varphi(z)\vec{F}(z), \vec{n}(z)\right\rangle \,\mathrm{d}z &= \iint_A \operatorname{div}\left(\varphi\vec{F}\right)(z)\,\mathrm{d}z \\ &= \iint_A \varphi(z)\operatorname{div}\vec{F}(z)\,\mathrm{d}z + \iint_A \left\langle \nabla\varphi(z), \vec{F}(z)\right\rangle \,\mathrm{d}z, \end{aligned} \tag{3.13}$$

yielding

$$m(A) = \left| \iint_A \varphi(z)\operatorname{div}\vec{F}(z)\,\mathrm{d}z + \iint_A \left\langle \nabla\varphi(z), \vec{F}(z)\right\rangle \,\mathrm{d}z \right|. \tag{3.14}$$

The first term,

$$\iint_A \varphi(z)\operatorname{div}\vec{F}(z)\,\mathrm{d}z, \tag{3.15}$$

captures the effect of the *OF divergence*, whereas the second term,

$$\iint_A \left\langle \nabla\varphi(z), \vec{F}(z)\right\rangle \,\mathrm{d}z, \tag{3.16}$$

captures the *RF divergence*.

It may happen that, following the field in one direction, the inter-ridge distances decrease as the field lines converge (e.g. lines and the spaces between them get thinner). Then, RF divergence and OF divergence have different signs, nearly cancelling each other, yielding $m(A) \approx 0$. The minutiae number is always non-negative due to the absolute values taken in (3.10) and (3.14), in particular making the sum of divergences in (3.14) independent of the specific direction $\vec{F}$ of θ chosen, cf. Section 3.2.1.

In practice, the OF and RF are only available to us on a digital lattice. Hence, an approximation formula for the necessary minutiae number would be desirable. This will be derived in the following. To this end, recall that a set $A \subseteq \mathbb{R}^2$ is called *star-shaped* with respect to $z_0 \in \mathbb{R}^2$ if $tz_0 + (1-t)z \in A$ for all $z \in A$ and $t \in [0,1]$, cf. (Amann & Escher, 2008, p. 314 ff.). We write $r(A) := r_{z_0}(A) := \sup_{z \in A} \|z - z_0\|$ for the radius of such a set and $|A|$ for its area (if it is measurable).

Theorem 3.13 (Minutiae divergence formula)**.** *Let $z_0 \in \mathfrak{X}$ be fixed. Suppose that $A \subseteq \mathfrak{X}$ is a compact set that is star-shaped w.r.t. z_0, does not contain any of the singularities of $\vec{F}$ and has piecewise smooth boundary ∂A. Then,*

$$m(A) = \left|\varphi(z_0) \operatorname{div} \vec{F}(z_0) + \left\langle \nabla\varphi(z_0), \vec{F}(z_0) \right\rangle\right| \cdot |A| + o(|A|) \quad \text{as } r(A) \to 0. \tag{3.17}$$

Proof. We define $f : A \to \mathbb{R}, f(z) := \operatorname{div}(\varphi\vec{F})(z)$ which, cf. (3.13) and (3.14), satisfies $m(A) = \left|\iint_A f(z)\,\mathrm{d}z\right|$. The first term on right hand side of Equation (3.17) equals $|f(z_0)|$. Thus, we can reformulate the assertion to

$$\frac{\left|\left|\iint_A f(z)\,\mathrm{d}z\right| - |f(z_0)||A|\right|}{|A|} \longrightarrow 0$$

as $r(A) \to 0$. Since $\varphi \in \mathcal{C}^2(A, (0,1))$ and $\vec{F} \in \mathcal{C}^2(A, \mathbb{R}^2)$, we have $f \in \mathcal{C}^1(A, \mathbb{R})$ (where the derivatives of f in the boundary points have to interpreted appropriately). Since A is compact and does not contain any of the singularities of $\vec{F}$, we have $\sup_{\xi \in A} \|\nabla f(\xi)\| < \infty$. By the multivariate mean value theorem, we have for any $z \in A$ that

$$f(z) = f(z_0) + \nabla f(\xi_z)^\top (z_0 - z)$$

for some $\xi_z = z + t_z(z - z_0) \in A$ (since A is star-shaped w.r.t. z_0) and $t_z \in [0,1]$. Hence, we have

$$\begin{aligned}
\left|\left|\iint_A f(z)\,\mathrm{d}z\right| - |f(z_0)||A|\right| &\leq \left|\iint_A f(z)\,\mathrm{d}z - f(z_0)|A|\right| \\
&= \left|\iint_A f(z_0) + \nabla f(\xi_z)^\top (z - z_0) - f(z_0)\,\mathrm{d}z\right| \\
&= \left|\iint_A \nabla f(\xi_z)^\top (z - z_0)\,\mathrm{d}z\right| \\
&\leq \sup_{\xi \in A} \|\nabla f(\xi)\| \sup_{z \in A} \|z - z_0\| \; |A|,
\end{aligned}$$

where the inequalities follow from the reverse triangle inequality[6] and the triangle inequality, respectively. Above, ξ_z is a measurable selection from the measurable set $\{\xi \in A : 0 =$

[6] for $x, y \in \mathbb{R}$ holds $||x| - |y|| \leq |x - y|$

$f(z) - f(z_0) - \nabla f(\xi)^\top (z - z_0)\}$, for instance one with minimal first and, if necessary, also with minimal second component. Note, that the first term is also bounded for $r(A) \to 0$ due to the monotonicity of the supremum. Now, dividing by $|A|$ and letting $r(A) \to 0$ yields the assertion. ■

This theorem motivates the definition of the necessary minutiae intensity governed by the sum of local OF divergence and local RF divergence.

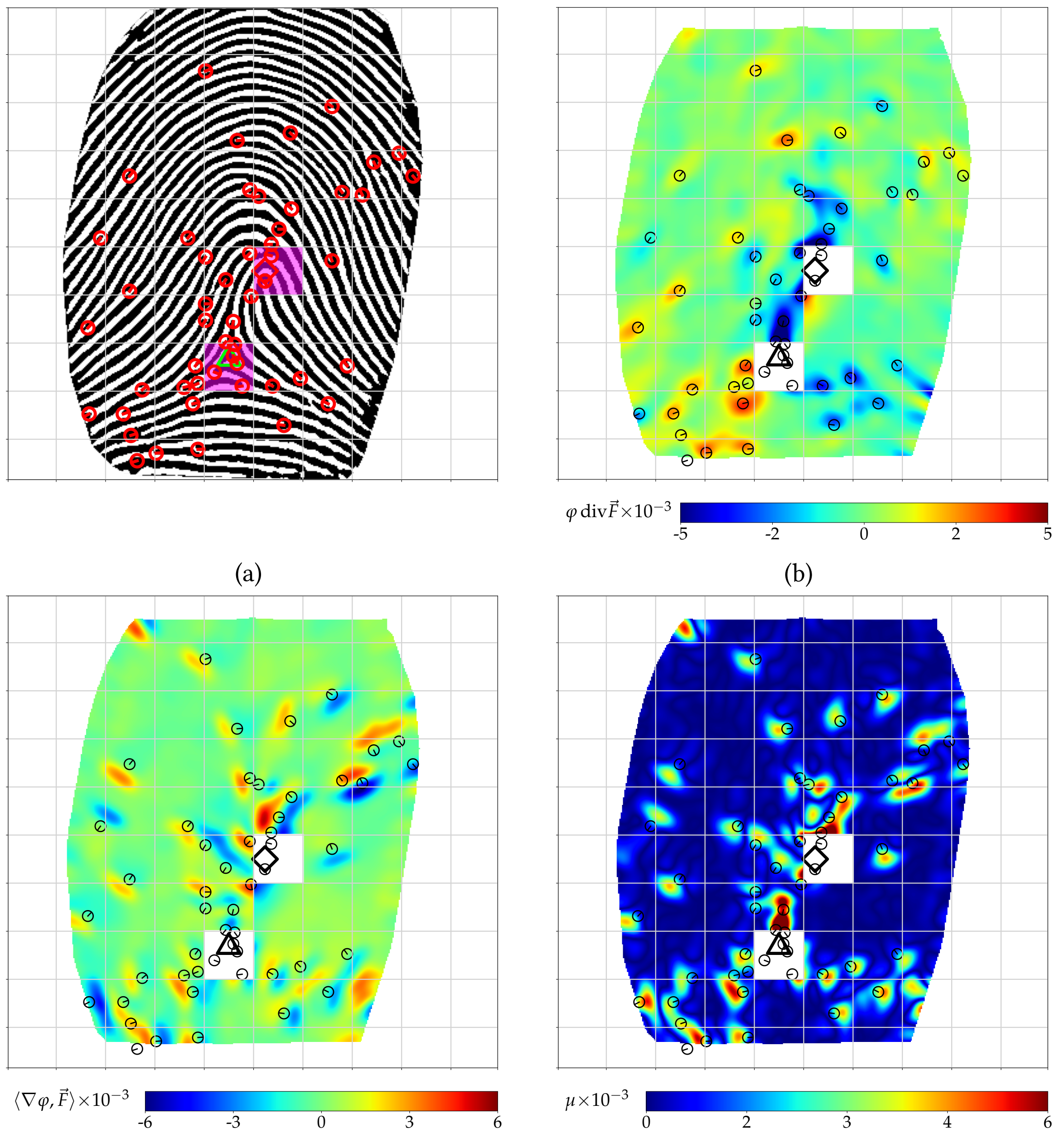
Definition 3.14 (Intensity of necessary minutiae). For $z_0 \in \mathfrak{X}$ outside the set of singularities of $\vec{F}$, call

$$\mu(z_0) = \left| \varphi(z_0) \operatorname{div} \vec{F}(z_0) + \left\langle \nabla\varphi(z_0), \vec{F}(z_0) \right\rangle \right| \tag{3.18}$$

the *intensity of necessary minutiae* at z_0.

Example 3.15. An example for the OF divergence and RF divergence as well as the resulting intensity of necessary minutiae of Fingerprint 1_1 from (Maio et al., 2002, DB1) is shown in Figure 3.5d. The computation is based on the approximation in Theorem 3.13 using $\varphi, \nabla\varphi, \vec{F}, \operatorname{div} \vec{F}$ which, in turn, are obtained by smoothing with a Gaussian kernel. Recall that we can assign a meaningful $\mathcal{C}^2$ direction field within patches not containing a singularity. Patches too close to a singularity (closer than 10 pixels, i.e. approx. one interridge distance; singularity detection according to the Poincaré index, cf. Theorem 3.8) are excluded from further consideration (marked violet in Figure 3.5a), because due to large derivatives of these quantities the approximations are typically bad; see the proof of Theorem 3.13. For details concerning algorithms to compute the intensity we refer to Chapter 6.

Notably, in regions close to minutiae locations, we usually observe a large intensity of necessary minutiae due to the OF divergence, see Figure 3.5b or RF divergence, see Figure 3.5c. This was already mentioned in Nikodémusz-Székely & Székely (1993) which used the divergence to locate minutiae. The question whether there are minutiae beyond the ones explained by OF and RF divergence as well as a corresponding quantitative analysis has, to the best of our knowledge, not been considered in the literature so far. This gives rise to the study in the following Section 3.4.



(a)

(b)

(c)

(d)

Figure 3.5: *OF divergence (b) and RF divergence (c) of Fingerprint 1_1 (a) from DB1 of Maio et al. (2002). The corresponding intensity image in (d). Minutiae have been marked as circles in red (a) and black (b–d), respectively. Note that regions too close to singularities (marked violet in (a)) have been excluded from the computations.*

3.4 The existence of random minutiae

Having found a formula predicting the number of necessary minutiae given the OF and the RF divergence, we investigate in this section whether there are additional minutiae in fingerprint patterns not explained by OF and RF divergence.

To this end, we preprocess 20 high quality fingerprints, cf. Figure 3.6, from the database (Maio et al., 2002, DB1) using the algorithm from Thai et al. (2016) in order to obtain enhanced and binarised versions of the images and the regions of interest. We then manually mark the minutiae, subdivide each image into approx. 100 rectangular patches A (aspect ratio taken from the images), cf. Figure 3.7, and count the number of minutiae in these patches.

The intensity of necessary minutiae is computed as described in Example 3.15, cf. also Figure 7.9 or Chapter 6 for a more detailed discussion. Patches too close to a singularity (closer than 10 pixels) are again discarded. An example for Fingerprint 1_1 is shown in Figure 3.7. Clearly, one can see that for most patches A the minutiae count is at least as large as our estimated minutiae intensity $m(A)$. Sometimes, however, the result is a little offset, see e.g. the orange patch on the upper right corner where the corresponding minutia is already assigned to the blue patch below. Overall, we observe many low intensity patches (blue), some of which still containing minutiae. Indeed, the sum $\sum_{i=1}^{100} m(A_i)$ over all 100 patches A_i (except the ones close to singularities) is 39, however 44 minutiae are observed in the considered patches, i.e. we actually observe 5 minutiae more than expected (or necessary) due OF and RF divergence. That this is not only an effect for this particular fingerprint is shown by a study over all fingerprints from Figure 3.6.

Figure 3.8 displays actual minutiae counts over all 20 imprints. At the height of every integer valued count observation k (on the y-axis) we have horizontally placed black dots at corresponding necessary minutiae numbers $m(A)$, i.e. at the not necessarily integer valued number of minutiae we expect in patches A (that actually feature k minutiae) if there were only necessary minutiae. We apply a Poisson regression with identity link, i.e. we fit the expected number of actual minutiae $\mu(A)$ in A as $\mu(A) = \beta_0 + \beta_1 m(A)$ and determine $\hat{\beta}_0, \hat{\beta}_1$ using maximum likelihood estimation (MLE). As a word of caution, the difference of the data to its regression line reflects a Poisson regression for count data, it does not reflect an ordinary least squares regression, see e.g. Cameron & Trivedi (2013).

The Poisson regression line in Figure 3.8 (red line), which fits surprisingly well the means of the massive histograms (based on bins with more than 100 elements which cover more than 87% of all patches). This confirms that on average the actual minutiae count increases with

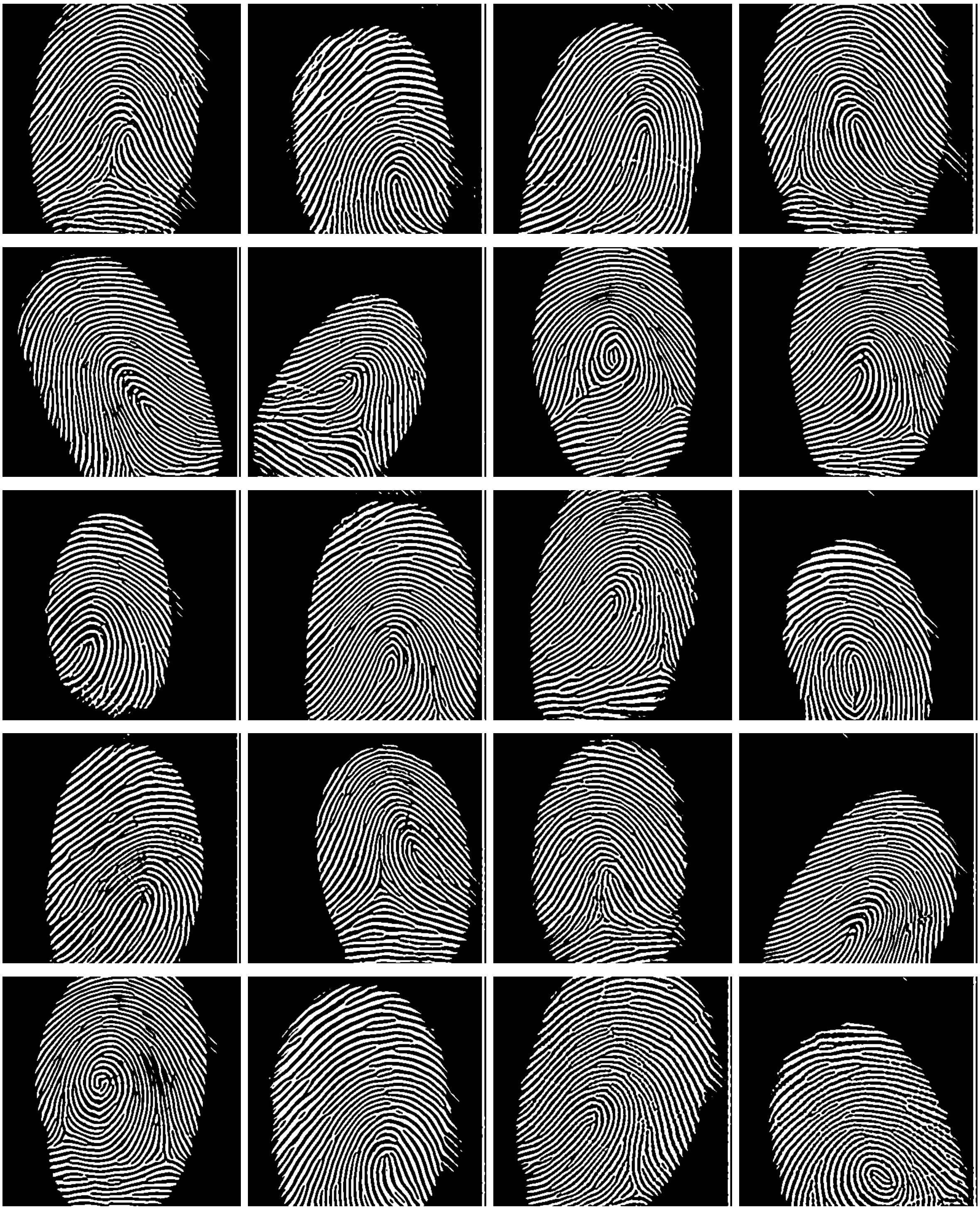


Figure 3.6: *Database of 20 high quality fingerprints from (Maio et al., 2002, DB1) labelled 1_1, 2_8, 7_1, 9_8, 13_4, 22_4, 25_2, 26_2, 28_4, 31_5, 34_1, 35_6, 53_6, 57_3, 59_2, 65_4, 66_2, 76_6, 89_5 and 100_6 (row-wise). These images form the basis of our analysis. The images depicted images are already preprocessed and binarised by the algorithm of Thai et al. (2016).*

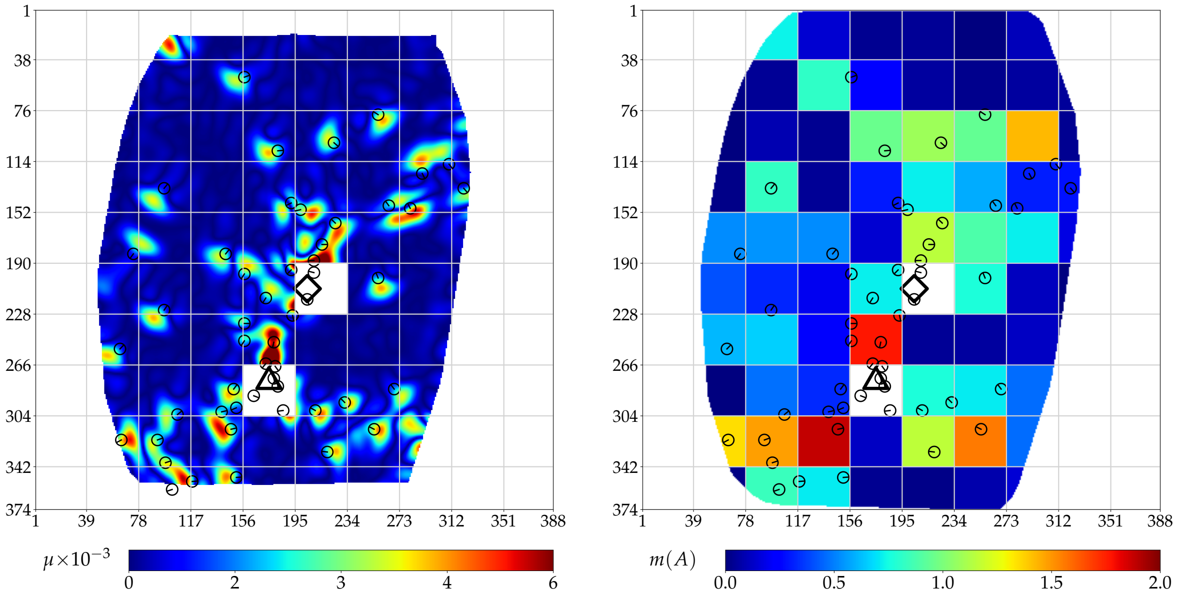


Figure 3.7: *Left: Intensity image μ of Fingerprint 1_1 from (Maio et al., 2002, DB1). Patches containing singularities are discarded (white). Pixel grid coordinates as axis labels. Actual minutiae positions are indicated with circles. Right: Integrated minutiae intensity $m(A)$ within patches A.*

a slope close to one (95% confidence interval $[0.903, 1.175]$) with a significant intercept of 0.14 (95% confidence interval $[0.106, 0.184]$; $p < 10^{-12}$) and suggests that the actual number of minutiae is larger than the number of minutiae necessary based on OF and RF divergence (dotted black line). We refer to the additional minutiae as *random minutiae*.

Remark 3.16 (Intensity of random minutiae)**.** Comparing the intercept of 0.14 random minutiae per patch to the average number of 0.50 total minutiae per patch, we obtain the rule of thumb that, out of 7 minutiae, 5 are necessary and 2 are random (i.e. approx. 28% of the total number of minutiae are random).

The images in FVC2002 DB1 have size 388 × 374 pixels (at a resolution of 500 dpi). Using the area of one pixel as our spatial unit (each pixel is a $0.0508 \times 0.0508\,\mathrm{mm}^2$ square) we conclude from the fact that the average patch size is approximately $\frac{388}{10} \times \frac{374}{10} \approx 1451$ pixels that we may use

$$\lambda_0 = 0.14\,/\,1451 \approx 10^{-4} \tag{3.19}$$

as an initial estimate for the random minutiae intensity.

Remark 3.17 (Discretisation effects)**.** At this point, the reader might wonder whether the proposed minutiae divergence formula would be exact (and hence the number of random minutiae being zero) if we just consider small enough patches A (i.e. for small $r(A)$). We

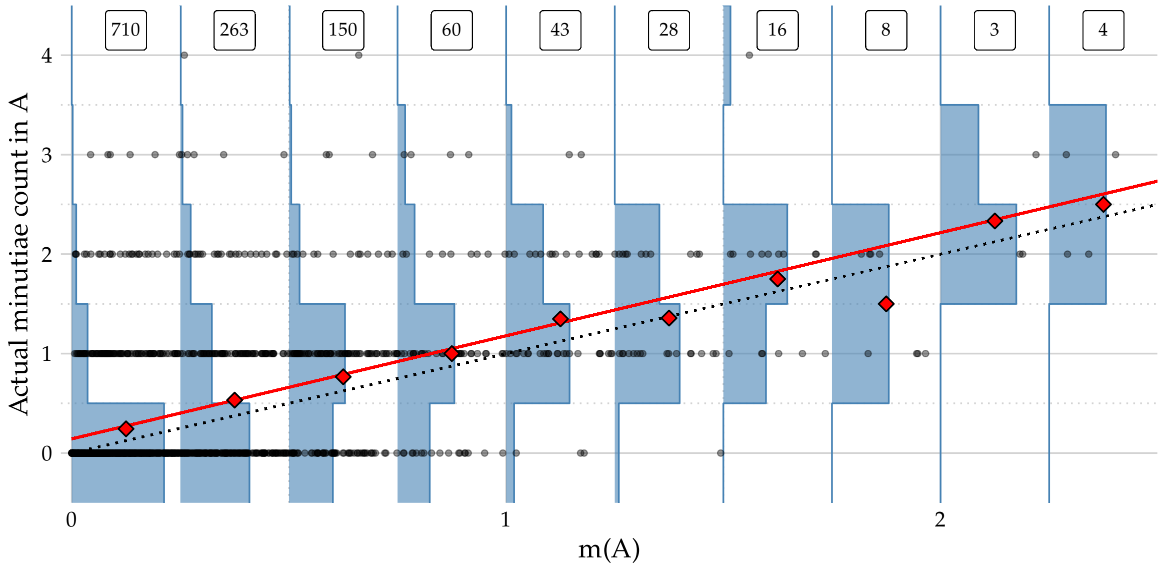


Figure 3.8: *Poisson regression (red line) for actual minutiae count (black dots) in patches A (of approx. 1451 pixels, cf. Figure 3.7 and Remark 3.16) versus necessary minutiae number $m(A)$ computed by Formula 3.14. The dotted black line (identity function) shows the relation we would expect if there were no random minutiae. The probability mass function of the counts within bins of width 0.25 is depicted in blue; their mean as red diamond (◇) in the centre of the bin. The number of observations within the bins is written on top. Based on 20 high quality fingerprints from FVC2002 DB1.*

are convinced of our model, however, cannot provide empirical evidence neither for nor against this theory. Since minutiae are only visible on a scale large enough to identify ridge lines we cannot zoom in arbitrarily far into the image, thus prohibiting us to make an analysis for arbitrarily small patches. However, we like to remark that the chosen patch size was just large enough to cover on average three neighbouring ridge lines which is the smallest size to discern divergence of ridge lines and thinning/ widening of spaces between ridges.

One possibility for going into this direction would be to consider the dependency of the amount of random minutiae depending on the patch size and check where there is a decrease of random minutiae for smaller patch size. To this end, the bandwidths of the smoothing filters and the unwrapping routine to compute the minutiae intensity and corresponding ingredients has to be adapted according to the considered scale, requiring a considerable additional effort. Thus, this approach is left for future investigations.

The study in this chapter, cf. Figure 3.8, shows that in general we observe more minutiae than are explained by OF and RF information. However, it does not give indication of *which* of the observed minutiae are (likely to be) random. In the following chapter, we investigate the separation of the minutiae pattern using a statistical model from the field of spatial point processes. In Section 7.3 we show that random minutiae can be *characteristic* in the sense that they can provide valuable information for distinguishing fingerprints with similar OFs and RFs.

Chapter 4

Spatial point processes

The aim of this chapter is to set up a mathematical model which allows for separation of a given minutiae pattern into necessary and random minutiae. To do so, we assume the minutiae pattern to be a realisation of the disjoint union of two independent point-pattern-valued random variables, so-called *spatial point processes* (or point processes for short).

Modern point process theory has been a major area of research in spatial statistics (i.e. statistics of spatial data) since the mid 20th century, its roots however have a long history at least up to Poisson (1837). In practical applications the considered point patterns often exhibit a complex interaction structure (i.e. the points attract or repel each other and are not independently distributed in space) and, moreover, we usually face small sample sizes (such as only a single point pattern of 40–100 points) introducing difficulties beyond the theory of classical statistics. Regarding the considered fingerprint setting we usually have only one minutiae pattern at hand. Moreover, the minutiae exhibit a certain inhibition between each other due to the discrete structure of the fingerprint ridges and the digital lattice of the image the fingerprint is captured on.

Modelling the interaction between points is conveniently possible using densities w.r.t. to a certain reference measure[7]. In the context of spatial point processes, this has to be a measure on the space of point patterns. Commonly used is the probability distribution of a stationary process with no interaction structure, the homogeneous *Poisson (point) process*. This process is, moreover, in practice a popular choice to model *noise* points because apart from the noise intensity it does not include any further information. In Sections 4.1–4.3, we provide the theoretical background for Poisson processes and point processes exhibiting interaction as well as some details about simulation. Assuming we know which minutiae are necessary and random, we analyse a model fit approach in Section 4.4. Dropping

[7] like a density on $\mathbb{R}^d$ is usually w.r.t. the Lebesgue measure on $\mathbb{R}^d$

this assumption we analyse a superposition model and a possible approach on parameter inference in Section 4.5. Proofs of this section have been outsourced in part to Section 4.6.

4.1 Poisson point processes

Let $\mathfrak{X} \subseteq \mathbb{R}^2$ be a compact subset of the Euclidean plane with positive Lebesgue measure (modelling the fingerprint's region of interest) and $\mathcal{B} = \sigma(\mathfrak{X})$ be the Borel σ-algebra of $\mathfrak{X}$.

In a nutshell, a finite point process on $\mathfrak{X}$ is a random finite point pattern on $\mathfrak{X}$ in which both the number of points as well as their locations are random. The mathematically rigorous definition is based on counting measures. Recall, that a measure ξ on $(\mathfrak{X}, \mathcal{B})$ is called *counting measure*, if $\xi : \mathcal{B} \to \mathbb{N}_0 \cup \{+\infty\}$, i.e. if it only assigns non-negative integers or infinity. It is *finite* if $\xi(\mathfrak{X}) < \infty$. We denote by $\mathfrak{N}$ the set of all finite counting measures and equip it with the smallest σ-algebra $\mathcal{N}$ such that the mappings $\xi \mapsto \xi(A)$ are measurable for all $A \in \mathcal{B}$.

Definition 4.1 (Point process, (Daley & Vere-Jones, 2008, Definition 9.1.VI.))**.** A (finite) *point process* on $(\mathfrak{X}, \mathcal{B})$ is a measurable mapping Ξ from a probability space $(\Omega, \mathcal{A}, \mathbb{P})$ to $(\mathfrak{N}, \mathcal{N})$. Indeed, for every $\omega \in \Omega$ the realisation $\Xi(\omega, \cdot)$ is a finite counting measure on $(\mathfrak{X}, \mathcal{B})$. Hence one could also call Ξ a random finite counting measure.

The following characterisation of point processes is often more convenient to work with.

Theorem 4.2. *Let Ξ be a point process on $(\mathfrak{X}, \mathcal{B})$. Then, there are random elements $X_1, X_2, \ldots$ on $(\mathfrak{X}, \mathcal{B})$ and an $\mathbb{N}_0$-valued random variable N such that*

$$\Xi = \sum_{i=1}^{N} \delta_{X_i} \tag{4.1}$$

Conversely, the right hand side of (4.1) for corresponding $(X_i)_{i \in \mathbb{N}}$ and N always defines a point process.

Proof. This result follows from (Daley & Vere-Jones, 2008, Proposition 9.1.III. & Proposition 9.1.X.). ■

Let us make a few remarks on point process terminology and notation before we proceed.

Remark 4.3. (a) If for a point process $\Xi = \sum_{i=1}^{N} \delta_{X_i}$, the X_i are almost surely distinct, the point process is also referred to as *simple*. In the context of this thesis, we will only consider simple point processes and therefore tacitly assume that every point process is simple.

(b) A *point pattern* is a realisation $\xi = \Xi(\omega, \cdot) \in \mathfrak{N}$ of Ξ for some $\omega \in \Omega$. For a simple process, we can interpret a point pattern ξ both as a finite counting measure $\xi = \sum_{i=1}^{n} \delta_{x_i}$ and as a finite set $\{x_1, x_2, \ldots, x_n\}$ coinciding with the support of ξ. Hence, using set-notation, we also write

$$\xi \cup \eta := \xi + \eta, \qquad \xi \setminus \eta := \xi - \eta, \qquad x \in \xi :\Longleftrightarrow x \in \operatorname{supp}(\xi)$$

etc. for $\xi, \eta \in \mathfrak{N}$ and $x \in \mathfrak{X}$. Note that for well-definedness of $\xi \setminus \eta$ we need $\eta \leq \xi$ (in measure-theoretical sense, i.e. $\eta(B) \leq \xi(B)$ for all $B \in \mathcal{B}$). We will use this notation whenever it is more convenient.

(c) A realisation $\Xi(\omega, \cdot)$ takes the value $\Xi(\omega, B)$ on any Borel set $B \in \mathcal{B}$, i.e. the number of points in B. Fixing $B \in \mathcal{B}$, the function $\Xi(\cdot, B) : \Omega \to \mathbb{R}_+, \omega \mapsto \Xi(\omega, B)$ is indeed a random variable, cf. (Daley & Vere-Jones, 2008, Proposition 9.1.VIII.). We write, suppressing the dependence of ω,

$$\Xi(B) := \Xi(\cdot, B) = \sum_{i=1}^{N} \mathbf{1}\left(X_i \in B\right)$$

for the random number of points falling in $B \in \mathcal{B}$.

(d) The *distribution* of a point process Ξ is the probability measure it induces on $(\mathfrak{N}, \mathcal{N})$, i.e. $\mathbb{P}_\Xi(A) := \mathbb{P}(\Xi \in A) := \mathbb{P}(\Xi^{-1}(A))$ for $A \in \mathcal{N}$. It is completely determined by its finite dimensional distributions, i.e. the joint distributions of the random variables $\Xi(A_1), \Xi(A_2), \ldots, \Xi(A_k)$ for any finite family of Borel sets $A_1, A_2, \ldots, A_k \in \mathcal{B}$, $k \in \mathbb{N}$, cf. (Daley & Vere-Jones, 2008, Corollary 9.2.IV).

(e) The measure

$$\mathbb{E}\Xi : \mathcal{B} \to \mathbb{R}_+, \quad B \mapsto \mathbb{E}\Xi(B) \tag{4.2}$$

is called the *intensity measure* or *expectation measure* of Ξ. For a given set $B \subseteq \mathfrak{X}$ we have that $\mathbb{E}\Xi(B)$ is the expected number of points of Ξ falling in B. If $\mathbb{E}\Xi$ has a density λ w.r.t. the Lebesgue measure, then λ is called the *intensity function*.

(f) For two point processes $\Xi = \sum_{i=1}^{N} \delta_{X_i}, \mathrm{H} = \sum_{j=1}^{M} \delta_{Y_j}$ we can define their *superposition* $\Xi \cup \mathrm{H}$ as

$$\Xi \cup \mathrm{H} := \sum_{i=1}^{N} \delta_{X_i} + \sum_{j=1}^{M} \delta_{Y_j}.$$

If Ξ and H are independent and $\Xi \cap \mathrm{H} = \emptyset$ almost surely (i.e. the X_i and Y_j are almost surely distinct), we also write $\Xi \dot{\cup} \mathrm{H}$ (in analogy to a disjoint union) to emphasise the independence and (a.s.) disjoint union.

One of the most prominent examples for a point process is given by the so-called *Poisson point process*.

Definition 4.4 (Poisson point process)**.** Let $\lambda : \mathfrak{X} \to \mathbb{R}_+$ be a Lebesgue-integrable function and denote for $B \in \mathcal{B}$ by $\Lambda(B) := \int_B \lambda(x)\,\mathrm{d}x$ the measure induced by λ. Let $N \sim \mathrm{Poi}(\Lambda(\mathfrak{X}))$ be Poisson distributed with parameter $\Lambda(\mathfrak{X})$ and $X_1, X_2, \ldots$ be independent and identically distributed with density $\frac{1}{\Lambda(\mathfrak{X})}\,\lambda$. Then, the point process

$$\Xi = \sum_{i=1}^{N} \delta_{X_i} \tag{4.3}$$

is called *Poisson (point) process* with *intensity* λ. We write $\Xi \sim \mathrm{Pop}(\mathfrak{X}, \lambda)$ or $\Xi \sim \mathrm{Pop}(\lambda)$ for short if there is no confusion of $\mathfrak{X}$ possible. If λ is constant, we call Ξ *homogeneous* (or *stationary*), otherwise *inhomogeneous* (or *non-stationary*).

Some realisations of Poisson processes can be seen in Figure 4.1. Whereas in the homogeneous case the points appear to be relatively evenly distributed over the region of interest, in the inhomogeneous case it is very apparent that points tend to lie in regions of larger intensity. Note that for a Poisson process the variance of the number of points is equally large as the intensity. Thus, for an inhomogeneous Poisson process with the minutiae intensity as underlying intensity function, the variance in the number of points is in regions close to the fingerprint's singularities very high. There, points can apparently get arbitrarily close to each other, cf. the bottom row Figure 4.1.

Remark 4.5. A Poisson process $\Xi \sim \mathrm{Pop}(\lambda)$ satisfies the following properties, cf. for instance (Daley & Vere-Jones, 2003, Chapter 2):

(a) $\Xi(B) \sim \mathrm{Poi}\left(\int_B \lambda(x)\,\mathrm{d}x\right)$, i.e. the one-dimensional distributions of $\Xi(B)$ are in fact Poisson distributions, hence the name.

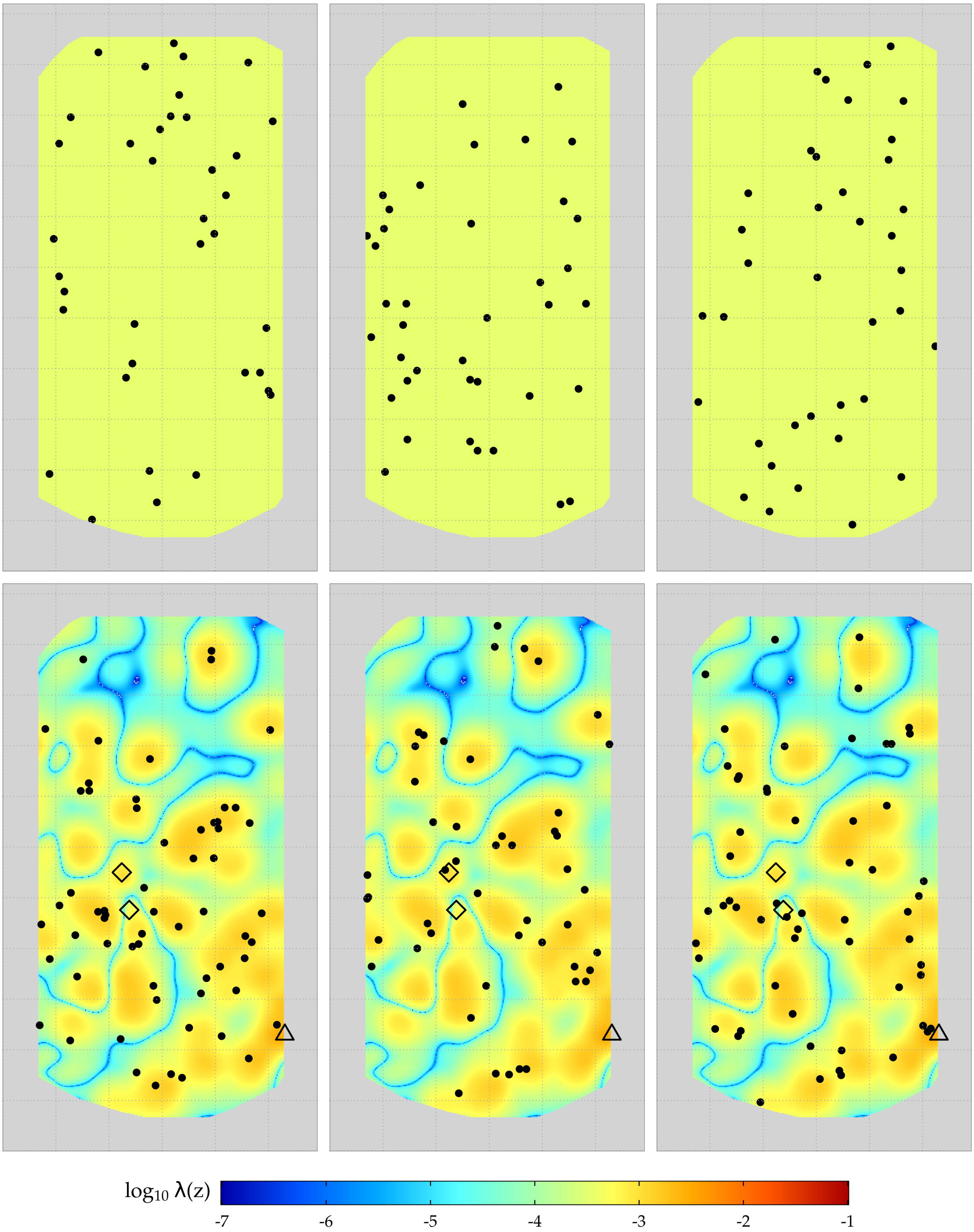


Figure 4.1: *Top: Different realisations of a homogeneous Poisson process with intensity* $\lambda = 3.5 \times 10^{-4}$ *(leading to an expected number of 48 points); bottom: an inhomogeneous Poisson process with intensity* λ *(scaled necessary minutiae intensity* μ *from Fingerprint 7_6 of (Maio et al., 2002, DB2), see Section 3.3) as heat map in the background (leading to the same number of expected points). Pixels outside the ROI are shaded in grey. The grid in the background indicates squares of side length 50 pixels.*

(b) In particular, we have $\mathbb{E}\Xi(B) = \Lambda(B)$, i.e. the expected number of points of Ξ in B is $\Lambda(B)$. Thus, Λ *is* the expectation measure and λ *is* the intensity (function) of Ξ.

(c) More generally, the finite distributions are given by

$$\mathbb{P}\left(\Xi(B_1) = k_1, \Xi(B_2) = k_2, \ldots, \Xi(B_r) = k_r\right) = \prod_{i=1}^{r} \frac{(\Lambda(B_i))^{k_i}}{k_i!} \mathrm{e}^{-\lambda(B_i)} \tag{4.4}$$

with $r \in \mathbb{N}$ disjoint sets $B_1, B_2, \ldots, B_r \in \mathcal{B}$ and $k_1, k_2, \ldots, k_r \in \mathbb{N}_0$. In particular, (4.4) implies that point counts on disjoints sets are independent.

(d) A homogeneous Poisson process is also referred to model *complete spatial randomness*, i.e. the points occur in a completely random fashion and exhibit no interaction between each other. Moreover, one can show that the Poisson process is the only simple process without fixed atoms[8] having this property, cf. (Daley & Vere-Jones, 2003, Section 2.3).

A very convenient way of dealing with distributions of random variables is via describing them with their density w.r.t. to a reference measure, for instance the Lebesgue measure on $\mathbb{R}^d$. For point processes a similar approach is possible but the reference measure is now not longer given by the Lebesgue measure but by the distribution of the *standard Poisson process*, i.e. the unit Poisson process $\Pi_1 := \mathrm{Pop}(\mathfrak{X}, 1)$. Here, by 1 we mean the function $x \mapsto 1$ which is constantly one on $\mathfrak{X}$.

Definition 4.6 (Density)**.** For a spatial point process Ξ we call a measurable function $f : \mathfrak{N} \to \mathbb{R}_+$ with

$$\mathbb{P}(\Xi \in A) = \int_A f(\xi)\, \Pi_1(\mathrm{d}\xi) \tag{4.5}$$

for any $A \in \mathcal{N}$, a *density* of Ξ (w.r.t. Π_1). Due to the Radon-Nikodym theorem, cf. (Klenke, 2020, Corollary 7.34), the density f exists if and only if the distribution of Ξ is absolutely continuous w.r.t. Π_1, i.e. for every Π_1-null set A we have $\mathbb{P}(\Xi \in A) = 0$. In this case, f is almost surely unique, cf. (Klenke, 2020, Theorem 7.29).

Remark 4.7. The integral in (4.5) can be computed as

$$\begin{aligned} &\int_A f(\xi)\, \Pi_1(\mathrm{d}\xi) \qquad (4.6)\\ &= \sum_{n=0}^{\infty} \frac{\exp\left(-|\mathfrak{X}|\right)}{n!} \int_{\mathfrak{X}} \cdots \int_{\mathfrak{X}} \mathbf{1}\left(\{x_1, x_2, \ldots, x_n\} \in A\right) f\left(\{x_1, \ldots, x_n\}\right) \mathrm{d}x_1 \ldots \mathrm{d}x_n, \end{aligned}$$

[8] a point $x \in \mathfrak{X}$ is called *fixed atom* of a point process Ξ if $\mathbb{P}(\Xi(\{x\}) > 0) > 0$

cf. (Møller & Waagepetersen, 2003, Section 6.1). However, the integral (4.6) has in general no explicit representation and its computation is often intractable.

For Poisson processes we can explicitly compute the density using the following result from (Møller & Waagepetersen, 2003, Proposition 3.8).

Lemma 4.8 (Density of Poisson processes). *Let* $\Xi \sim \operatorname{Pop}(\lambda)$ *for some integrable* $\lambda : \mathfrak{X} \to \mathbb{R}_+$. *Then* Ξ *is absolutely continuous w.r.t.* Π_1 *with density*

$$f : \mathfrak{N} \to \mathbb{R}_+, \qquad f(\xi) = \mathrm{e}^{\int_{\mathfrak{X}} 1-\lambda(u)\,\mathrm{d}u} \prod_{x \in \xi} \lambda(x). \tag{4.7}$$

Note, that in (4.7) we can explicitly compute not only the density, but also its normalising constant – a property which cannot be taken for granted as we will see in the next section.

4.2 Interaction processes

The Poisson processes considered in Section 4.1 have the property that there is no interaction between points. For Poisson processes using the minutiae intensity field as underlying intensity function, simulations show that in regions of high intensity (e.g. close to singularities, cf. Figure 4.1) points can lie arbitrarily close to each other. For minutiae patterns, however, we observe in practice a certain minimal distance between each other and, moreover, a bounded variance in the number of points also near singularities. We elaborate this in greater detail in Section 4.4. Henceforth, we consider a point process model which also takes interaction into account, more precisely repulsion, between pairs of points.

Definition 4.9 (Strauss process, Strauss (1975)). Let $\beta : \mathfrak{X} \to \mathbb{R}_+$ be a Lebesgue-integrable function. A *Strauss process* on $\mathfrak{X}$ with *trend* (or *activity*) function β, *hard core* $r \geq 0$, *interaction parameter* $\gamma \in [0, 1]$ and *interaction range* $R > r$ is a point process H with density

$$g : \mathfrak{N} \to \mathbb{R}_+, \quad \eta \mapsto \alpha \prod_{x \in \eta} \beta(x)\, \gamma^{s_R(\eta)}\, \mathbf{1}\left(d_{\min}(\eta) > r\right)$$

with respect to the unit Poisson process Π_1, where

$$s_R(\eta) := \sum_{\{x,y\} \subseteq \eta} \mathbf{1}(\|x - y\| \leq R) \tag{4.8}$$

is the number of R-*close* pairs in η (by convention $\{x, y\} \subseteq \eta$ shall always exclude the case $x = y$),

$$d_{\min}(\eta) := \min \{\|x - y\| : \{x, y\} \subseteq \eta\} \tag{4.9}$$

is the *minimum inter-point distance* (or *hard core distance*) in η and

$$\alpha = \left(\sum_{n=0}^{\infty} \frac{\exp(-|\mathfrak{X}|)}{n!} \int_{\mathfrak{X}} \cdots \int_{\mathfrak{X}} \prod_{i=1}^{n} \beta(x_i) \gamma^{s_R(\{x_1,\dots,x_n\})} \mathbf{1}\left(d_{\min}(\{x_1, \dots, x_n\}) > r\right) \mathrm{d}x_1 \dots \mathrm{d}x_n \right)^{-1}$$

is the normalising constant. If β is constant, the process is called *homogeneous* (or *stationary*), otherwise *inhomogeneous* (or *non-stationary*).

We write $\mathrm{H} \sim \mathrm{StraussHard}(\mathfrak{X}, \beta, \gamma, r, R)$ or $\mathrm{H} \sim \mathrm{StraussHard}(\beta, \gamma, r, R)$ for short. For processes with $r = 0$, i.e. without hard core we simply write $\mathrm{H} \sim \mathrm{Strauss}(\mathfrak{X}, \beta, \gamma, R)$ or $\mathrm{H} \sim \mathrm{Strauss}(\beta, \gamma, R)$ if there is no confusion about $\mathfrak{X}$ possible.

Strauss processes belong to the class of *Gibbs processes*. For a more detailed view into the literature we refer to (Møller & Waagepetersen, 2003, Chapter 6).

At this point, the reader might wonder why we propose a model with even two scales of repulsion (r and R). We will elaborate this in greater detail in Section 4.4, but like to give a brief answer also already here. Regarding the fingerprint setting we like to choose the hard core r depending on the inter-ridge distance (minutiae can only be observed on ridges). This distance, however, varies over the fingerprint. Since the model requires a constant hard core, we have to decide for a representative quantity, e.g. the minimal or average inter-ridge distance. By doing so, the above mentioned regularisation of the variance of point numbers in regions of high divergence could not be reached to a satisfying level, hence, an additional interaction scale was introduced.

Remark 4.10. To understand the influences of the model parameters, let us at first consider a Strauss process $\mathrm{H} \sim \mathrm{Strauss}(\beta, \gamma, R)$. Examples for simulated point patterns from homogeneous and inhomogeneous Strauss processes can be seen in Figure 4.2.

(a) For $\gamma = 1$, we obtain a Poisson process with intensity β, i.e. $\mathrm{Strauss}(\beta, 1, R) = \mathrm{Pop}(\beta)$ for any $R > 0$. Hence, the Strauss process is a generalisation of the Poisson process including interaction between points, where $\gamma = 1$ means no interaction at all and $\gamma = 0$ means total repulsion, see also (b).

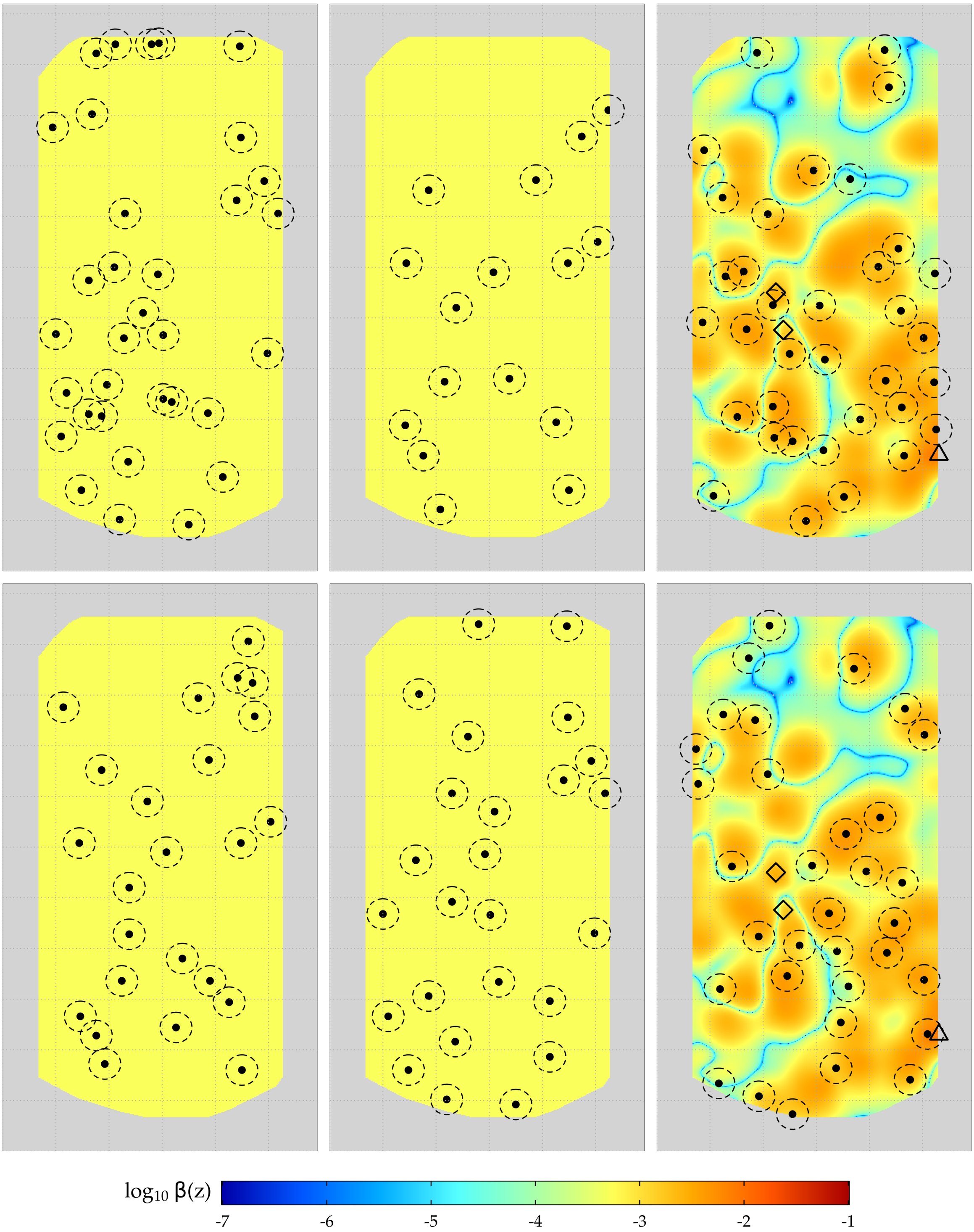


Figure 4.2: *Top and bottom show two different realisations of three Strauss processes (from left to right), respectively, in the region of interest of fingerprint 7_6 of FVC2002_DB2. Left: homogeneous Strauss with $\beta = 5 \times 10^{-4}$, $\gamma = 0.4$. Middle: homogeneous hard core process with $\beta = 5 \times 10^{-4}$. Right: inhomogeneous Strauss with $\beta(z) = 4 \times \mu(z)$, with μ from (3.18) as heat map for Fingerprint 7_6 from (Maio et al., 2002, DB2), $\gamma = 0.1$. The interaction distance is given as the diameter $R = 30$ of the dashed circles. Again, pixels outside the ROI are shaded in grey. The grid in the background indicates squares of side length 50 pixels.*

(b) For $\gamma = 0$, the density of the $\mathrm{Strauss}(\beta, 0, R)$ process w.r.t. Π_1 is

$$g(\eta) = \alpha \left(\prod_{x \in \eta} \beta(x) \right) \mathbf{1} \left(d_{\min}(\eta) > R \right). \tag{4.10}$$

This process is called *hard core process* with *hard core distance* R. A hard-core process does not allow R-close points at all. We write also $\mathrm{H} \sim \mathrm{Hardcore}(\beta, R)$.

(c) For every R-close pair in a point pattern the "likeliness" of the whole pattern decreases by a factor γ.

(d) In contrast to the Poisson process, the parameter β is for $\gamma < 1$ *not* the intensity of the process $\mathrm{H} \sim \mathrm{Strauss}(\beta, \gamma, R)$ but $\mathbb{E}\mathrm{H}(B) < \int_B \beta(x)\,\mathrm{d}x$, cf. Baddeley et al. (2012); Stucki & Schuhmacher (2014), i.e. the expected number of points expected in a set is smaller than the integrated trend.

(e) In general, the normalising constant $\alpha = \alpha(\beta, \gamma, r, R)$ is unknown and cannot be computed explicitely, see e.g. Kelly & Ripley (1976), (Møller & Waagepetersen, 2003, Section 6.2), Berthelsen & Møller (2006). Even in the homogeneous case and for known r, R it is not clear, how α, β and γ interact. Baddeley et al. (2012); Coeurjolly & Lavancier (2018) give some approximations.

To omit the intractable normalisation constant we also write

$$g(\eta) \propto \prod_{x \in \eta} \beta(x) \gamma^{s_R(\eta)},$$

where $\propto$ means *proportional* and the constant of proportionality does not depend on η. Gibbs processes originated in statistical physics where α^{-1} is also called the *partition function*.

Remark 4.11 (Edge effects)**.** The interaction between pairs of points becomes particularly apparent in regions of large activity β of inhomogeneous Strauss processes as in the right column of Figure 4.2. In regions near the boundary, this effect often results in points occurring close to an edge since there is no repulsion due to points outside the observation window (which are not observed), cf. also (Baddeley & Nair, 2012, Figure 1). This affects also statistical inference methods. Note that edge effects do not occur for Poisson processes since there is no interaction between points.

To deal with edge effects, two different scenarios have to be distinguished: either we assume that no points exist beyond the observation window or the point pattern exists on a larger window and the point pattern originates from clipping a larger point pattern to the observed area. For the application in fingerprint recognition, we face rather the latter case (restriction of the minutiae pattern of the *entire* fingerprint to the region captured by the fingerprint sensor) although it is not always justified to assume the existence of points beyond the observation window (e.g. in regions close to the fingernail on which no minutiae exist).

For point processes with a homogeneous trend edge correction procedures expand the activity function accordingly to the larger window. For inhomogeneous processes this is approach requires knowledge about the activity function outside the observation window which is typically not available in practice. Common workarounds are reflection of the activity at the boundary or periodic continuation, however neither of these seem to be very realistic concerning the fingerprint application. Since edge effects seem to have only little influence for the data considered in this thesis we do not use edge correction except automatically applied by statistical methods in R, R Core Team (2021), (we note this at the corresponding places). Edge correction procedures are discussed in more detail e.g. in (Baddeley et al., 2015, Section 6.5) or (Ripley, 1988, Chapter 3).

The interaction structure of Strauss processes finds expression in their second moments.

Definition 4.12 (Second moments, pair correlation)**.** Let H be a finite point process on $(\mathfrak{X}, \mathcal{B})$ with $\mathbb{E}\mathrm{H}(\mathfrak{X})^2 < \infty$.

(a) The *second factorial moment measure* is defined as

$$\alpha^{(2)} : \mathcal{B}(\mathfrak{X} \times \mathfrak{X}) \to \mathbb{R}_+, \qquad B \mapsto \mathbb{E}\left(\sum_{\substack{x,y \in \mathrm{H} \\ x \neq y}} \mathbf{1}\left((x,y) \in B\right) \right). \tag{4.11}$$

(b) If for a point process H, the intensity λ exists and $\alpha^{(2)}$ has a density a w.r.t. the Lebesgue measure, then

$$\rho^{(2)} : \mathfrak{X} \times \mathfrak{X} \to \mathbb{R}_+, \qquad (x,y) \mapsto \frac{a(x,y)}{\lambda(x)\,\lambda(y)} \tag{4.12}$$

is called the *pair correlation function* (PCF) of H where we take $\frac{a}{0} := 0$ for all $a \geq 0$.

(c) The point process is said to be *correlation isotropic*, if $\rho^{(2)}(x,y) = \rho(\|x-y\|)$ for some function $\rho : \mathbb{R}_+ \to \mathbb{R}_+$, i.e. the pair correlation is *isotropic*—its value in (x,y) depends only on the distance between the points $x, y \in \mathfrak{X}$.

For a Poisson process we obtain $a(x, y) = \lambda(x)\lambda(y)$ and hence $\rho^{(2)}(x, y) = 1$. If $\rho^{(2)}(x, y) < 1$ this indicates that pairs of points are less likely to occur (*inhibition*) at the locations x, y than for a Poisson process with the same intensity as H whereas $\rho^{(2)}(x, y) > 1$ suggests they are more likely to occur (*clustering*), see also (Illian et al., 2008, Section 4.3.4). Notably, Poisson processes and homogeneous Strauss processes are correlation isotropic, see also Baddeley & Nair (2012).

The interaction between pairs of points in Strauss processes is particularly easy to see by considering the pair correlation function, cf. (4.12), even though ρ is in general not explicitly available. Nevertheless, we can estimate the PCF for a given point pattern $\zeta = \{z_1, z_2, \dots, z_n\}$ where we will derive an estimator in the following.

Example 4.13 (Pair correlation function for stationary processes)**.** Assume for the moment that Z is a stationary process and let $r \geq 0$. Then,

$$\sum_{\substack{j=1 \\ j \neq i}}^{n} \mathbf{1}\left(\|z_i - z_j\| \leq r\right)$$

is the number of r-*neighbours* of z_i, $i \in \{1, 2, \dots, n\}$, i.e. the number of other data points in a ball of radius r around z_i, cf. Figure 4.3a (where we have 12 r-neighbours around $+$). Thus,

$$\frac{1}{n} \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} \mathbf{1}\left(\|z_i - z_j\| \leq r\right) \tag{4.13}$$

yields an estimate for the average number of r-neighbours per data point. This quantity will typically depend on the overall average density of points in the dataset. Thus, for comparability, this number is divided by the intensity λ of Z in the considered region $\mathfrak{X}$. As λ is typically not available explicitly in practice, it is substituted with an estimate $\hat{\lambda} = \frac{n-1}{|\mathfrak{X}|}$. Note, that we use here $n - 1$ instead of n, such that we obtain an unbiased estimator in the case of complete spatial randomness, i.e. for a homogeneous Poisson process. Moreover, in practice edge correction weights are introduced. The resulting estimator

$$\hat{K}(r) = \frac{|\mathfrak{X}|}{n(n-1)} \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} \mathbf{1}\left(\|z_i - z_j\| \leq r\right) e_{ij}(r), \tag{4.14}$$

where $e_{ij}(r)$ is an edge-correction weight, cf. (Baddeley et al., 2015, Section 7.6.2), now estimates the (cumulative) average number of points in a ball of radius r around a *typical*

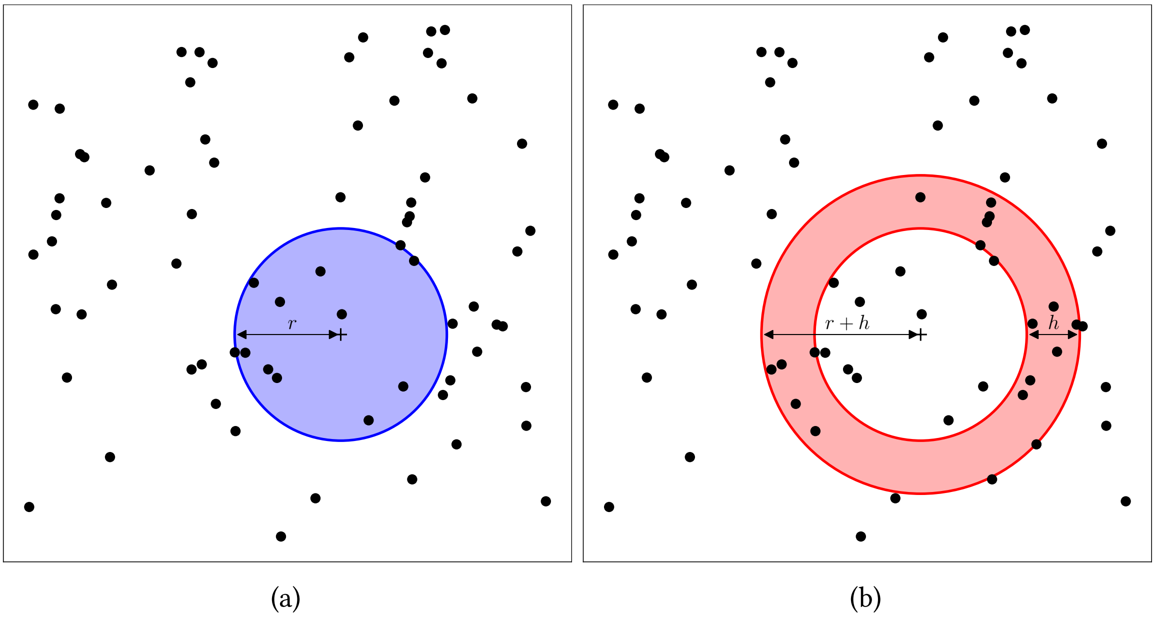


Figure 4.3: *Geometry of (a) Ripley's K-function and (b) the pair correlation function. Whereas for the K-function we count all points around a typical point (+) within a disk of radius r we count for the PCF only points in the annular ring around (+) between two concentric circles of radius r and $r+h$, respectively, for small $h > 0$.*

random point of Z, corrected by edge effects and in comparison to a homogeneous Poisson process with the same intensity. This standardisation allows for comparison between point patterns with different numbers of points and observed in different windows.

In the literature the estimator (4.14) is known as an estimator for *Ripley's K-function*, (Baddeley et al., 2015, Section 7.3):

$$K : \mathbb{R}_+ \to \mathbb{R}_+, \quad r \mapsto K(r) = \frac{1}{\lambda|\mathfrak{X}|}\mathbb{E}\left[\sum_{z_i \in \mathrm{Z}} \sum_{\substack{z_j \in \mathrm{Z} \\ j \neq i}} \mathbf{1}(\|z_i - z_j\| \leq r)\right]. \tag{4.15}$$

Geometrically, for the PCF $\rho(r)$ we draw two concentric circles of radius r and $r+h$, respectively, where $h > 0$ is a small increment, and count the number of points falling in the annulus between the two circles, see Figure 4.3b. This now captures the *interpoint distances* that lie in the range between r and $r+h$ (where there are 20 points in the red annulus around +). Their expected count is $\lambda K(r+h) - \lambda K(r)$. Standardising this quantity by dividing by the expected value for a homogeneous Poisson process leads to

$$\rho_h(r) = \frac{\lambda K(r+h) - \lambda K(r)}{\lambda\pi(r+h)^2 - \lambda\pi r^2} = \frac{K(r+h) - K(r)}{2\pi r h + \pi h^2} \longrightarrow \frac{K'(r)}{2\pi r}$$

as $h \to 0$. In fact, the PCF and the K-function (in the two-dimensional Euclidean plane) are related by

$$\rho(r) = \frac{K'(r)}{2\pi r},$$

cf. (Illian et al., 2008, Section 4.3). Hence, (4.14) leads to an estimator $\hat{\rho}$ for the pair correlation function as follows:

$$\hat{\rho} : \mathbb{R}_+ \to \mathbb{R}_+, \quad r \mapsto \hat{\rho}(r) = \frac{|\mathfrak{X}|}{2\pi n(n-1)r} \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} \kappa_b(r - \|z_i - z_j\|)\, e_{ij}(r), \tag{4.16}$$

where $\kappa_b(t) = \frac{1}{b}\kappa(\frac{t}{b})$ is a scaled version of a (e.g. Gaussian) smoothing kernel $\kappa : \mathbb{R} \to \mathbb{R}_+$ with smoothing bandwidth $b > 0$ chosen suitably in order to estimate the derivative $K'(r)$ as discussed in (Illian et al., 2008, Section 4.3.3).

The estimator from (4.16) often tends to overestimate the PCF for radii r close to zero due to the factor $\frac{1}{r}$ in (4.16). An alternative estimator which often exhibits better performance is

$$\hat{\rho} : \mathbb{R}_+ \to \mathbb{R}_+, \quad r \mapsto \hat{\rho}(r) = \frac{|\mathfrak{X}|}{2\pi n(n-1)} \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{\kappa_b(r - \|z_i - z_j\|)}{\|z_i - z_j\|}\, e_{ij}(r), \tag{4.17}$$

in which the contribution of an observed interpoint distance $\|z_i - z_j\|$ is $\frac{1}{\|z_i - z_j\|}$ rather than $\frac{1}{r}$, cf. (Baddeley et al., 2015, Section 7.6.2). We use the estimator (4.17) for our analyses in the following.

Let us now consider a $\mathrm{Strauss}(\beta, \gamma, R)$, a $\mathrm{Hardcore}(\beta, R)$ and a $\mathrm{StraussHard}(\beta, \gamma, r, R)$ process with $\beta = 5 \times 10^{-4}$, $\gamma = 0.4$, $r = 10$ and $R = 30$ from which we sampled 20 point patterns each. For each point pattern we then compute the estimate $\hat{\rho}$ for the corresponding pair correlation function ρ as in (4.17) using the `pcf` function of the `spatstat` package by Baddeley & Turner (2005) in R. Note that for the $e_{ij}(r)$ by default a translation edge correction[9] is implemented, see (Baddeley et al., 2015, Section 7.4.5). The resulting estimated PCFs are then *pooled* (taking the pointwise average, see (Baddeley et al., 2015, Section 16.8.2)) for each process to obtain a *typical* PCF for each process. The result can be seen in Figure 4.4.

[9] the edge correction weight e_{ij} is chosen as $|\mathfrak{X}|/|\mathfrak{X} \cap (|\mathfrak{X}| - z_i - z_j)|$, i.e. the reciprocal of the window area in which the first point z_i could be placed such that both z_i, z_j would be observable, assuming their relative positions were held fixed.

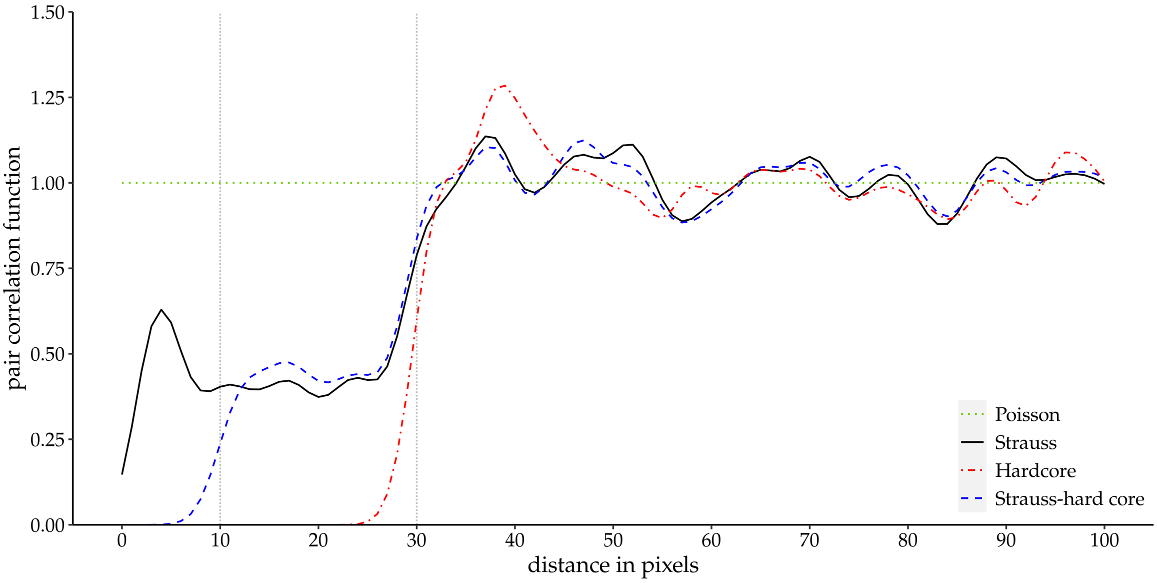


Figure 4.4: *Pooled pair correlation function for a* $\mathrm{Strauss}(\beta, \gamma, R)$*,* $\mathrm{Hardcore}(\beta, R)$ *and* $\mathrm{StraussHard}(\beta, \gamma, r, R)$ *process, respectively with* $\beta = 5 \times 10^{-4}$*,* $\gamma = 0.4$*,* $r = 10$ *and* $R = 30$*. The interaction radii* $r < R$ *as vertical dotted lines. For the computation a Gaussian smoothing kernel with bandwidth* $b = 2$ *was chosen.*

For the Strauss process, we see at a distance less than approx. $R = 30$ pixels less points of pairs compared to a Poisson process with intensity β (horizontal line at 1). This indicates the repulsion of points at a distance of less than R. Shortly after R, the curve overshoots the horizontal line at 1 before settling down afterwards around one, since there is no interaction at a distance larger than R. For the hard core process the inhibition is even stronger since we do not observe any points at a distance smaller than R. When considering a Strauss process with hard core, we can observe two regimes of inhibition: total repulsion up to a distance r and moderate repulsion at distance $(r, R]$.

Remark 4.14 (PCF for inhomogeneous processes)**.** In general, inhomogeneous Strauss processes are not correlation isotropic anymore, not even correlation stationary[10], i.e. the PCF being a function $\rho^{(2)}(x, y) = \rho(x - y)$ depending only on the relative position of the points x, y. Hence the PCF carries more information than provided by the estimator $\hat{\rho}$ in Example 4.13. To account for inhomogeneities of correlation stationary processes, usually one applies a first order correction where the factor $\frac{|\mathfrak{X}|}{n(n-1)}$ in (4.16) is substituted by the

[10] also called *second-order intensity reweighted stationary (soirs)* in the literature

product $\hat{\lambda}(z_i)\hat{\lambda}(z_j)$ of the estimated intensities in the considered data points such that we obtain the estimator

$$\hat{\rho}_{\text{inhom}} : \mathbb{R}_+ \to \mathbb{R}_+, \quad r \mapsto \hat{\rho}_{\text{inhom}}(r) = \frac{1}{2\pi r} \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{\kappa_b(r - \|z_i - z_j\|)}{\hat{\lambda}(z_i)\hat{\lambda}(z_j)} e_{ij}(r),$$

see (Baddeley et al., 2015, Section 7.10). The estimator (4.17) adjusts accordingly. Even though we are confident that this estimator does not provide the full information about the PCF (due to the lack of correlation stationarity of the inhomogeneous Strauss processes in general) we use this estimator as a heuristic to gain information about the true PCF and the interaction behaviour of a given point pattern. The `R` package `spatstat` provides a ready-to-use implementation called `pcfinhom` for the estimation of the PCF that accounts for inhomogeneities, cf. (Baddeley et al., 2015, Section 7.10.2) for details.

4.3 Simulation

At this point, let us say a few words about simulation of point patterns, i.e. how to draw samples from the two point process models proposed. Simulation of Poisson processes is usually easy.

Algorithm 4.15 (Simulation of Poisson processes)**.** We consider at first the case that we want to simulate a homogeneous Poisson process $\Xi \sim \text{Pop}(\mathfrak{X}, \lambda)$ for some $\lambda > 0$.

- Suppose $\mathfrak{X}$ is a box $B = [a_1, b_1] \times [a_2, b_2] \subseteq \mathbb{R}^2$. Then, cf. Ripley (1987), we first generate the number of points $\Xi(B) \sim \text{Poi}(\lambda(b_1 - a_1)(b_2 - a_2))$ and then the locations of the $\Xi(B)$ points independently and uniformly distributed on B.
- If $\mathfrak{X}$ is not a box but can be inscribed in a box B, we simulate Ξ on B and discard all the points in $B \setminus \mathfrak{X}$.

Simulation of inhomogeneous Poisson processes is based on independent thinning.

Definition 4.16 (Thinning)**.** Let $p : \mathfrak{X} \to [0, 1]$ be a function, Ξ a point process on $\mathfrak{X}$ and $U(x) \sim \mathcal{U}[0, 1]$ be mutually independent for $x \in \mathfrak{X}$ and independent of Ξ. Then, the process

$$\Xi_{\text{thin}} := \{x \in \Xi : U(x) \leq p(x)\}$$

is called *independent thinning* of Ξ with *retention probabilities* $p(x)$, $x \in \mathfrak{X}$. In words, we obtain Ξ_{thin} from Ξ by including $x \in \xi$ with probability $p(x)$ and the points are included/ excluded independently of each other.

Suppose now we want to sample from an inhomogeneous Poisson process $\Xi \sim \text{Pop}(\lambda)$ where $\lambda : \mathfrak{X} \to \mathbb{R}_+$ is supposed to be bounded, $\sup_{x \in \mathfrak{X}} \lambda(x) < \lambda_0$, say. Then we simulate at first $\Xi_0 \sim \text{Pop}(\lambda_0)$ and obtain Ξ via thinning of Ξ_0 with retention probabilities $p(x) = \frac{\lambda(x)}{\lambda_0}$, $x \in \mathfrak{X}$, cf. (Møller & Waagepetersen, 2003, Proposition 3.7).

In contrast, simulation of Strauss processes is in general non-trivial. Approximation based methods require running a Markov chain with corresponding invariant distribution. For a reminder on Markov chains, see Section A.2

Algorithm 4.17 (Simulation of Strauss processes)**.** A second method of simulating Strauss processes in addition to Markov chain-based simulations is based on construction by means of spatial birth-death processes (perfect sampling via dominated coupling from the past). The simulation using perfect sampling of homogeneous Strauss processes is possible in `R`, R Core Team (2021), using the procedure `rStrauss`, however, can be very time-consuming for processes exhibiting strong interaction of points. For a more detailed explanation of simulation of Gibbs processes we refer to (Chiu et al., 2013, Section 5.5.5) and the references therein.

A ready-to-use implementation for simulating inhomogeneous Strauss processes using perfect sampling does not yet exist. Instead, we use the `rmh` procedure of `R` (cf. (Baddeley et al., 2015, Section 13.9)) to simulate point patterns by running a Metropolis-Hastings algorithm (cf. Chapter 5). This method can also be applied to all other processes introduced in this chapter. In a nutshell, the idea is to run a Markov chain with the desired distribution as stationary distribution. After running the algorithm for a very long time, the state of the Markov chain may be regarded as a realisation of the desired distribution. However, it is not clear in practice how long is *long enough* and the choice of the number of iterations is usually more a matter of computational resources than of actual convergence. We elaborate details about such algorithms in Chapter 5 and refer to the literature for further discussions Clifford & Nicholls (1994); Møller & Waagepetersen (2003); Baddeley et al. (2015).

Using the provided theory we can now formulate and analyse a statistical model for minutiae patterns which allows for separation into necessary and random minutiae.

4.4 A superposition model for necessary and random minutiae

We consider a minutiae pattern $\zeta = \{z_1, z_2, \dots, z_n\} \subseteq \mathfrak{X}$ which we assume to be a sample of the superposition of two independent point processes Ξ and H modelling the random and the necessary minutiae, respectively. It is well known, see e.g. Stoney (1988); Chen & Moon (2006); Gottschlich & Huckemann (2014), that minutiae cannot be arbitrarily close to one another; they repel each other on a local scale. Indeed, due to the discrete nature of the ridge pattern, we cannot observe minutiae pairs at distance smaller than the inter-ridge distance. Although on good quality fingerprints, upon close inspection, occasionally closer minutiae pairs can be seen, e.g. bifurcations with one very short ridge, as these cannot be well discriminated from noise, they are usually removed as *false minutiae*, cf. (Maltoni et al., 2009, p. 157–158). This effect is well visible in Figure 4.5 showing the estimated (inhomogeneous) pair correlation function (similarly to Example 4.13) from the 20 hand-marked fingerprints considered in Figure 3.8 with approximate point-wise 95% confidence intervals (based on replications using the 20 fingerprints, see (Baddeley et al., 2015, Section 16.8.2) for details on this approach).

Figure 4.5 shows roughly two regimes of interaction. A regime of very strong inhibition in the range up to about 5–10 pixels and a regime of moderate inhibition up to 35–40 pixels. This suggests modelling the bulk of the minutiae by a Strauss process with hard core, see Section 4.2. For comparison, the pooled PCF estimate based on 20 simulated Strauss processes with hard core having activation functions $\beta\,\mu_k(z)$, $1 \leq k \leq 20$, is shown in Figure 4.5 (dash-dotted grey line), where μ_k is the necessary minutiae intensity (3.17) obtained from the k-th fingerprint image. The hard core distance $r = 8$ was chosen as the average inter-ridge distance (see Section 3.3). A pilot study on FVC2002 DB1 revealed that a Strauss interaction distance R of approximately three times the average inter-ridge distance seems to be a reasonable choice since only about 6% of all minutiae pairs of the considered fingerprints have a smaller distance. The parameters $\beta = 1.9$ and $\gamma = 0.37$ are reasonable choices in view of the simulations considered later in Chapter 7. Note that the real minutiae patterns also contain the random minutiae, which essentially explains the smaller PCF values of the grey curve (compared to the black one) for distances smaller than approx. 35 pixels. The local minimum of the grey curve at approx. 25 pixels indicates the distance from a typical point to regions with a small number of points beyond the nearest neighbours, cf. (Illian et al., 2008, Section 4.3.4). This could be due to the missing random minutiae or due to a random effect not included in the model. The red curve originates

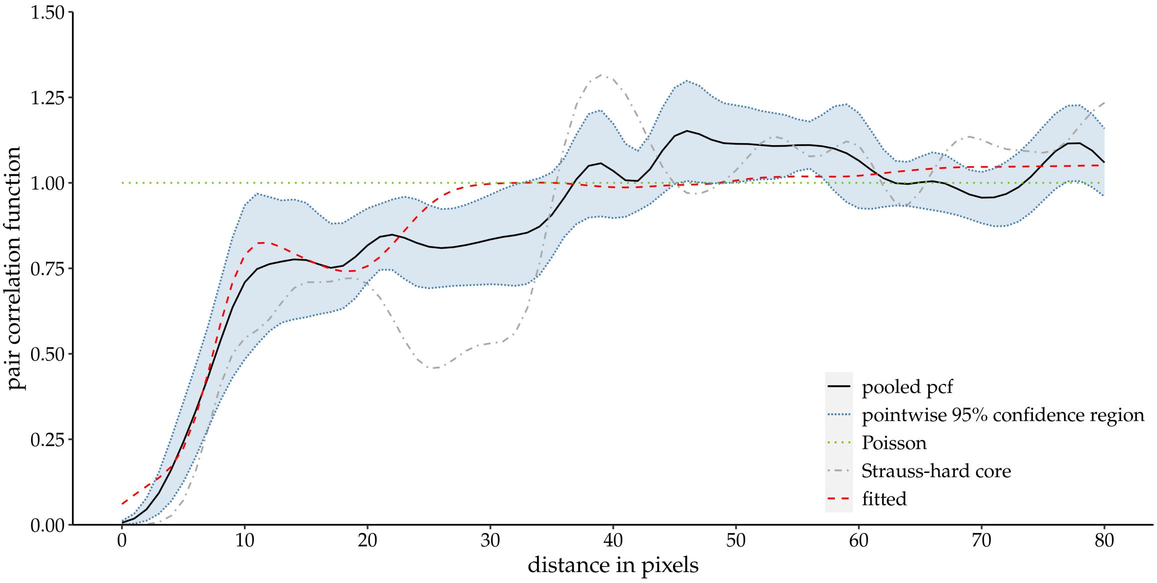


Figure 4.5: *Pooled pair correlation function (PCF) based on the 20 high quality fingerprint images from FVC2002 DB1 with hand marked minutiae (solid black). Dotted blue shaded: approximate pointwise 95% confidence intervals based on sample variances. Dotted green: theoretical PCF under the hypothesis of no interaction. Dash-dotted grey: pooled PCF based on simulated Strauss processes with hard core. Dashed red: pooled PCF for the 20 fitted models using the posterior mean from Section 7.2. The intensities are based on the necessary minutiae intensities of the same 20 prints; parameter choices are given in the main text.*

from a computed model fit from Chapter 7. The fact that this PCF is at distance approx. 25–30 pixels outside of the confidence band indicates a slightly too small choice of R. We will discuss this in further detail in Section 7.2.

Bearing the analysis of the PCF of the data in mind we come to the following modelling. As a model for the random minutiae process Ξ, we choose a homogeneous Poisson process, cf. Section 3.4 and Section 4.1, with unknown intensity $\lambda \geq 0$ and denote the density of Ξ w.r.t. Π_1 by f_λ. Ignoring that in practice random minutiae cannot be closer than the inter-ridge distance is harmless as their intensity is rather low, both in absolute terms and compared to the intensity of necessary minutiae (see Remark 3.16), but still visible in Figure 4.5.

If we also assumed that the necessary minutiae process H was a Poisson process, but inhomogeneous with intensity proportional to (3.10), then the independent superposition $\Xi \,\dot\cup\, \mathrm{H}$ would also be Poisson distributed, cf. Corollary 4.19. Under this assumption the theoretical pair correlation function would be one (cf. Definition 4.12) which, with regard to Figure 4.5, contrasts reality.

Last but not least, modelling the necessary minutiae as a point process with substantial inhibition of points is also advantageous from a conceptual point of view: up to certain errors arising from the discretisation of the OF and RF into minutiae information as well as from data acquisition and processing, the necessary minutiae counts should be *determined* by the underlying necessary minutiae intensity. Some inhibition between points is required to keep the variances of minutiae counts in regions with high necessary minutiae intensity small enough to be compatible with the data. Simulations we performed, cf. Figure 4.1 and the discussion in Definition 4.4, indicate that inhomogeneous Poisson processes based on the same intensity have too high variances.

As a result, we model the necessary minutiae by $\mathrm{H} \sim \mathrm{StraussHard}(\beta, \gamma, r, R)$ and denote the density of H w.r.t. Π_1 by $g_{\beta,\gamma}$. Based on formula (3.10), we assume

$$\beta(z) = \beta \cdot \mu(z) \text{ with } \mu(z) = \left|\varphi(z) \operatorname{div} \vec{F}(z) + \left\langle \nabla\varphi(z), \vec{F}(z) \right\rangle\right| \tag{4.18}$$

for some factor $\beta \approx 1$. We expect that β is in fact quite a bit larger than one because the presence of repulsion requires an activation larger than a Poisson intensity yielding comparable number of observed points, see Remark 4.10 (d) or Baddeley et al. (2012).

The hard core distance $r > 0$ and the interaction distance $R > r$ are assumed to be known in advance, since joint estimation of γ, r, R is notoriously difficult. This is on the one hand due to strong negative correlation of γ, r, R, cf. Redenbach et al. (2015). On the other hand and maybe more importantly, common procedures such as maximum (pseudo-) likelihood estimation often fail due to the lack of differentiability and non-concavity of the (pseudo-) likelihood (with respect to r, R) which may even be multimodal, see (Baddeley et al., 2015, Section 13.6.3). We choose the hard core distance r as the average inter-ridge distance of the finger – which seems to be be fairly realistic – and the interaction distance R to be three times as large, see above.

The observed minutiae pattern $\zeta = \{z_1, z_2, \ldots, z_n\}$ can then be written as $\zeta = \xi \mathbin{\dot\cup} \eta$ where ξ and η are samples from Ξ and H, respectively. We introduce a latent variable $\mathbf{W} \in \{0,1\}^n$ where $\mathbf{W}_i = \mathbf{1}(z_i \in \eta)$, so that $\mathbf{W}_i = 1$ means that minutia z_i is necessary. We combine the parameters into a vector $\boldsymbol{\theta} = (\lambda, \beta, \gamma) \in \Theta := \mathbb{R}_+ \times \mathbb{R}_+ \times [0,1]$. Then, due to the independence of Ξ and H, the density[11] of $\xi \mathbin{\dot\cup} \eta$ for known $\mathbf{W}$ is given as

$$f_\lambda(\xi)\, g_{\beta,\gamma}(\eta). \tag{4.19}$$

The question arises how to find suitable values of $\boldsymbol{\theta}$ and an assignment $\mathbf{W}$ of the minutiae to ξ and η. Although we primarily focus on Bayesian inference, we would like to briefly

[11] This can be interpreted as the density of the marked point pattern $\zeta = \sum_{i=1}^n \delta_{(z_i, \mathbf{W}_i)}$ given $\mathbf{W}$ w.r.t. a Poisson process with intensity measure $\mathrm{Leb}^2(\mathfrak{X}) \otimes (\delta_0 + \delta_1)$. For an overview about marked point processes see e.g. (Daley & Vere-Jones, 2003, Sections 6.4 and 7.3).

highlight a few aspects of frequentist inference for point patterns. From a frequentist point of view, a maximum likelihood estimator would be desirable. The computation of such an estimator for the parameters is notoriously difficult due to the intractable normalising constant. A common workaround for point processes is hence to consider maximum pseudolikelihood estimators. The mathematical fundamentals and computational details are presented in the following Section 4.5.

The approach presented there has turned out to be computationally considerably expensive and, additionally, does not provide information about $\mathbf{W}$[12]. Even more importantly, it is questionable whether there is only one single choice of $(\boldsymbol{\theta}, \mathbf{W})$ that fits best. Our view is that there likely are several choices which all fit reasonably well, particularly for $\mathbf{W}$. Bearing this in mind, we adopt a Bayesian approach. The mathematical essentials to this end are presented in Chapter 5. We explore the posterior distribution of $(\boldsymbol{\theta}, \mathbf{W})$ given the minutiae point pattern $\xi \dot{\cup} \eta$ in Chapter 7. Not only does this yield information about the parameter values, it also provides a quantification of the uncertainty of the assignment to the classes of necessary and random minutiae.

4.5 Maximum pseudolikelihood estimation for superpositions of point processes

The rest of this chapter is dedicated to a frequentist parameter estimation approach. Within the proposed model we are in the position to make inference on the model parameters $\boldsymbol{\theta}$, however not on the latent variable $\mathbf{W}$. This can be seen either as a disadvantage as we do not gain insights about the distribution of necessary and random minutiae or as an advantage as $\boldsymbol{\theta}$ estimation does not rely on the unknown quantity $\mathbf{W}$. We compare the results from this frequentist method with the ones from the Bayesian approach from Section 6.4 in Chapter 7.

We consider the superposition $\mathrm{Z} = \Xi \dot{\cup} \mathrm{H}$ of two a.s. disjoint independent point processes Ξ and H. Given that we know their density, the density of the superimposed process is as presented in Lemma 4.18. To the best of the author's knowledge, this has not been considered in the literature so far. Note, that auxiliary results for the proof have been moved to Section 4.6.

[12] Indeed, one could also include $\mathbf{W}$ as another parameter into the model. However, this would likely result in the (pseudo-) likelihood having several local maxima and thus leading to performance issues of algorithms in computing these maxima.

Lemma 4.18 (Density of the superposition of point processes)**.** *Let* Ξ, H *be two independent spatial point processes on* $\mathfrak{X}$ *with densities* f *and* g *w.r.t.* Π_1*, respectively. Then, the function*

$$h\colon \mathfrak{N} \to \mathbb{R}_+, \quad h(\zeta) = \mathrm{e}^{-|\mathfrak{X}|} \sum_{\xi \subseteq \zeta} f(\xi) g(\zeta \setminus \xi)$$

is a density of the superposition $\mathrm{Z} = \Xi \,\dot\cup\, \mathrm{H}$ *w.r.t.* Π_1.

Proof. Let

$$\Phi : \mathfrak{N} \times \mathfrak{N} \to \mathfrak{N} \times \mathfrak{N}, \quad \left(\sum_{n=1}^{N} \delta_{x_n}, \sum_{m=1}^{M} \delta_{y_m} \right) \mapsto \left(\sum_{n=1}^{N} \delta_{x_n}, \sum_{n=1}^{N} \delta_{x_n} + \sum_{m=1}^{M} \delta_{y_m} \right).$$

The mapping Φ is injective with inverse

$$\Phi^{-1} : \left\{ \left(\sum_{n=1}^{N} \delta_{x_n}, \sum_{\ell=1}^{L} \delta_{z_\ell} \right) \in \mathfrak{N} \times \mathfrak{N} : |\{n \mid x_n = x\}| \le |\{\ell \mid z_\ell = x\}|, x \in \mathfrak{X} \right\} \to \mathfrak{N} \times \mathfrak{N},$$
$$\left(\sum_{n=1}^{N} \delta_{x_n}, \sum_{\ell=1}^{L} \delta_{z_\ell} \right) \mapsto \left(\sum_{n=1}^{N} \delta_{x_n}, \sum_{\ell=1}^{L} \delta_{z_\ell} - \sum_{n=1}^{N} \delta_{x_n} \right).$$

and for any $(\mathcal{N} \otimes \mathcal{N}, \mathcal{B}(\mathbb{R}_+))$-measurable function $\tilde{h} : \mathfrak{N} \times \mathfrak{N} \to \mathbb{R}_+$ we have

$$\iint_{\mathfrak{N} \times \mathfrak{N}} \tilde{h}(\xi, \zeta)\, (\Pi_1 \otimes \Pi_1)\, \Phi^{-1}(\mathrm{d}\xi,\, \mathrm{d}\zeta) = \int_{\mathfrak{N}} \mathrm{e}^{-|\mathfrak{X}|} \sum_{\xi \subseteq \zeta} \tilde{h}(\xi, \zeta), \Pi_1(\mathrm{d}\zeta), \tag{4.20}$$

cf. Lemma 4.33 on p. 75. Let now $A \in \mathcal{N}$. Then,

$$\begin{aligned} \mathbb{P}(\mathrm{Z} \in A) &= \mathbb{P}(\Xi \,\dot\cup\, \mathrm{H} \in A) = \mathbb{P}\left((\Xi, \Xi \,\dot\cup\, \mathrm{H}) \in \mathfrak{N} \times A\right) \\ &= \mathbb{P}\left(\Phi(\Xi, \mathrm{H}) \in \mathfrak{N} \times A\right) = \mathbb{P}\left((\Xi, \mathrm{H}) \in \Phi^{-1}(\mathfrak{N} \times A)\right) \\ &= \int_{\Phi^{-1}(\mathfrak{N} \times A)} f(\xi) g(\eta)\, (\Pi_1 \otimes \Pi_1)\, (\mathrm{d}\xi, \mathrm{d}\eta)\,. \end{aligned}$$

Now, let $(f \cdot g)(\xi, \eta) := f(\xi) g(\eta)$. Then, with the transformation theorem (Bogachev, 2007, Theorem 3.6.1) follows

$$\begin{aligned} &\int_{\Phi^{-1}(\mathfrak{N} \times A)} f(\xi) g(\eta)\, (\Pi_1 \otimes \Pi_1)\, (\mathrm{d}\xi, \mathrm{d}\eta) \\ &= \int_{\Phi^{-1}(\mathfrak{N} \times A)} ((f \cdot g) \circ \Phi^{-1} \circ \Phi)(\xi, \eta)\, (\Pi_1 \otimes \Pi_1)\, (\mathrm{d}\xi, \mathrm{d}\eta) = \end{aligned}$$

$$
\begin{aligned}
&= \int_{\mathfrak{N}\times A} ((f \cdot g) \circ \Phi^{-1})(\xi, \eta)\, (\Pi_1 \otimes \Pi_1)\, \Phi^{-1}\, (\mathrm{d}\xi, \mathrm{d}\eta) \\
&= \int_{\mathfrak{N}\times A} f(\xi) g(\eta \setminus \xi)\, (\Pi_1 \otimes \Pi_1)\, \Phi^{-1}\, (\mathrm{d}\xi, \mathrm{d}\eta)\,.
\end{aligned}
$$

Now, we apply (4.20) for $\tilde{h}(\xi, \zeta) = f(\xi) g(\zeta \setminus \xi) \mathbf{1}_A(\zeta)$ and obtain

$$
\int_{\mathfrak{N}\times A} f(\xi) g(\zeta \setminus \xi)\, (\Pi_1 \otimes \Pi_1)\, \Phi^{-1}\, (\mathrm{d}\xi, \mathrm{d}\zeta) = \int_A \mathrm{e}^{-|\mathfrak{X}|} \sum_{\xi \subseteq \zeta} f(\xi) g(\zeta \setminus \xi) \Pi_1(\mathrm{d}\zeta),
$$

which concludes the proof. ■

The application of Lemma 4.18 to Poisson processes leads to a well-known result, see (Møller & Waagepetersen, 2003, Proposition 3.6).

Corollary 4.19. *The superposition of two independent Poisson processes* $\Xi \sim \mathrm{Pop}(\mathfrak{X}, \lambda)$ *and* $\mathrm{H} \sim \mathrm{Pop}(\mathfrak{X}, \mu)$ *is again a Poisson process, namely* $\Xi \,\dot{\cup}\, \mathrm{H} \sim \mathrm{Pop}(\mathfrak{X}, \lambda + \mu)$.

Proof. From Lemma 4.18 follows that

$$
h(\zeta) = \mathrm{e}^{-|\mathfrak{X}| - \int_{\mathfrak{X}} (\lambda+\mu)(x)\,\mathrm{d}x} \sum_{\xi \subseteq \zeta} \left(\prod_{x \in \xi} \lambda(x) \prod_{y \in \zeta \setminus \xi} \mu(y) \right) = \mathrm{e}^{-|\mathfrak{X}| - \int_{\mathfrak{X}} (\lambda+\mu)(x)\,\mathrm{d}x} \prod_{z \in \zeta} (\lambda(z) + \mu(z))\,.
$$

The latter equality follows by induction on $n = |\zeta|$ as follows. The claim is obviously true for $n = 0, 1$. Assume now $\zeta = \{z_1, \ldots, z_n\}$ for $n \geq 2$. Then,

$$
\begin{aligned}
\sum_{\xi \subseteq \zeta} \left(\prod_{x \in \xi} \lambda(x) \prod_{y \in \zeta \setminus \xi} \mu(y) \right) &= \sum_{\xi \subseteq (\zeta \setminus \{z_n\})} \lambda(z_n) \left(\prod_{x \in \xi} \lambda(x) \prod_{y \in (\zeta \setminus \{z_n\}) \setminus \xi} \mu(y) \right) + \\
&\quad + \sum_{\xi \subseteq (\zeta \setminus \{z_n\})} \mu(z_n) \left(\prod_{x \in \xi} \lambda(x) \prod_{y \in (\zeta \setminus \{z_n\}) \setminus \xi} \mu(y) \right) \\
&= \lambda(z_n) \prod_{z \in \zeta \setminus \{z_n\}} (\lambda(z) + \mu(z)) + \mu(z_n) \prod_{z \in \zeta \setminus \{z_n\}} (\lambda(z) + \mu(z)) \\
&= \prod_{z \in \zeta} (\lambda(z) + \mu(z))\,,
\end{aligned}
$$

which yields the claim. ■

Corollary 4.19 provides another argument that modelling both the necessary and the random minutiae by a Poisson process is, at least concerning parameter estimation, not a good idea—in this model only the sum of the parameters is identifiable, however the parameter themselves are not. Recall, that a family of probability distributions $(\mathbb{P}_{\boldsymbol{\theta}})_{\boldsymbol{\theta}\in\Theta}$ is said to be *identifiable* if the mapping $\boldsymbol{\theta} \mapsto \mathbb{P}_{\boldsymbol{\theta}}$ is injective. In other words, different parameters lead to different probability distributions.

Immediately the question arises whether inference using the superposition model from above is meaningful in terms of identifiability. We will consider this in the following. For the superposition of a homogeneous Poisson process $\Xi \sim \mathrm{Pop}(\lambda)$ and a Strauss process with hard core $\mathrm{H} \sim \mathrm{StraussHard}(\beta\mu, \gamma, r, R)$ with known interaction radii $r < R$, interaction parameter $\gamma < 1$ (non-Poisson case) and trend function $\mu : \mathfrak{X} \to \mathbb{R}_{++}$, we obtain an identifiability result as follows. Here we show only the case for $\mu = 1$. The proof for general μ can be found in Section 4.6. Note, that identifiability in the superposition model above has, to the best of the author's knowledge, not been considered in the literature so far.

Lemma 4.20 (Identifiability). *Let* $\Xi \sim \mathrm{Pop}(\lambda)$*,* $\mathrm{H} \sim \mathrm{StraussHard}(\beta, \gamma, r, R)$ *be independent with* $\lambda > 0$*,* $\beta > 0$ *and* $\gamma \in [0, 1)$ *and assume the parameters* $0 < r < R$ *to be known. Moreover, suppose that* $cR > r$ *for some* $c \in (\frac{1}{2}, 1)$ *and that* $\mathrm{diam}(\mathfrak{X}) > 2cR$*. Consider* $\mathrm{Z} = \Xi \,\dot{\cup}\, \mathrm{H}$ *with density* $h_{\boldsymbol{\theta}} : \mathfrak{N} \to \mathbb{R}_+$ *w.r.t. the unit Poisson process* Π_1 *where* $\boldsymbol{\theta} = (\lambda, \beta, \gamma)$*. Then, the parameters* (λ, β, γ) *are identifiable.*

Proof. (a) Let $A = \{\emptyset, \zeta_1, \zeta_2, \zeta_3\}$ with

$$\begin{aligned} \zeta_1 &= \delta_{x_1}, && \\ \zeta_2 &= \delta_{x_2} + \delta_{y_2}, && \text{with } r < \|x_2 - y_2\| < R \text{ and} \\ \zeta_3 &= \delta_{x_3} + \delta_{y_3} + \delta_{z_3}, && \text{with } r < \|x_3 - y_3\|, \|x_3 - z_3\| < R \text{ and } \|y_3 - z_3\| > R \end{aligned}$$

for $x_1, x_2, x_3, y_2, y_3, z_3 \in \mathfrak{X}$; existence of ζ_3 is ensured with $\mathrm{diam}(\mathfrak{X}) > R$. We show at first that $h_{\boldsymbol{\theta}}(\zeta) = h_{\tilde{\boldsymbol{\theta}}}(\zeta)$ for $\zeta \in A$ implies $\boldsymbol{\theta} = \tilde{\boldsymbol{\theta}}$.

(b) Suppose $h_{\boldsymbol{\theta}}(\zeta) = h_{\tilde{\boldsymbol{\theta}}}(\zeta)$ for $\zeta \in A$. Since $h_{\boldsymbol{\theta}}(\emptyset) = h_{\tilde{\boldsymbol{\theta}}}(\emptyset)$, we know that the normalising constants of $h_{\boldsymbol{\theta}}$ and $h_{\tilde{\boldsymbol{\theta}}}$ are the same. Hence, we may base our computations on equality on the unnormalised densities $\breve{h}_{\boldsymbol{\theta}}$, $\breve{h}_{\tilde{\boldsymbol{\theta}}}$.

$$\breve{h}_{\boldsymbol{\theta}}(\zeta_1) = \lambda + \beta, \tag{4.21}$$

$$\begin{aligned} \breve{h}_{\boldsymbol{\theta}}(\zeta_2) &= \lambda^2 + 2\lambda\beta + \beta^2\gamma \\ &= (\lambda + \beta)^2 + \beta^2(\gamma - 1), \end{aligned} \tag{4.22}$$

$$\begin{aligned}\breve{h}_{\boldsymbol{\theta}}(\zeta_3) &= \lambda^3 + 3\lambda^2\beta + (1+2\gamma)\lambda\beta^2 + \beta^3\gamma^2 \\ &= (\lambda+\beta)^3 + \beta^2(\gamma-1)(2\lambda+\beta(\gamma+1)). \end{aligned} \tag{4.23}$$

Hence, if $h_{\boldsymbol{\theta}}(\zeta) = h_{\tilde{\boldsymbol{\theta}}}(\zeta)$ for all $\zeta \in A$, the equations (4.21), (4.22) and (4.23) lead to

$$\lambda + \beta = \tilde{\lambda} + \tilde{\beta}, \tag{4.24}$$

$$\beta^2(\gamma-1) = \tilde{\beta}^2(\tilde{\gamma}-1), \tag{4.25}$$

$$2\lambda + \beta(\gamma+1) = 2\tilde{\lambda} + \tilde{\beta}(\tilde{\gamma}+1). \tag{4.26}$$

If we subtract two times (4.24) from (4.26) we obtain

$$\beta(\gamma-1) = \tilde{\beta}(\tilde{\gamma}-1),$$

yielding $\beta\tilde{\beta}(\gamma-1) = \tilde{\beta}^2(\tilde{\gamma}-1) = \beta^2(\gamma-1)$ where the latter equality originates from (4.25). Since $\beta > 0$ and $\gamma < 1$, we obtain $\beta = \tilde{\beta}$ and thus immediately also $\gamma = \tilde{\gamma}$ and $\lambda = \tilde{\lambda}$, which shows (a).

(c) Let $\boldsymbol{\theta} \neq \tilde{\boldsymbol{\theta}}$. Then we have to show that there is an A with $\Pi_1(A) > 0$ s. t. $h_{\boldsymbol{\theta}}(\zeta) \neq h_{\tilde{\boldsymbol{\theta}}}(\zeta)$ for all $\zeta \in A$. Assume the opposite, i.e. that for all A with $\Pi_1(A) > 0$ we have an $\zeta \in A$ such that $h_{\boldsymbol{\theta}}(\zeta) = h_{\tilde{\boldsymbol{\theta}}}(\zeta)$. Choose three pairwise different points x, y, z s. t. $\|x-y\| = \|x-z\| = cR$ and $\|y-z\| = 2cR$. For

$$\varepsilon = \frac{1}{4}\min\left\{R(1-c), 2(cR-r), R(2c-1)\right\} \tag{4.27}$$

let

$$\begin{aligned} A_1 &= \left\{\delta_{\tilde{x}} \,\middle|\, \tilde{x} \in \mathbb{B}_\varepsilon(x)\right\}, \\ A_2 &= \left\{\delta_{\tilde{x}} + \delta_{\tilde{y}} \,\middle|\, \tilde{x} \in \mathbb{B}_\varepsilon(x), \tilde{y} \in \mathbb{B}_\varepsilon(y)\right\}, \\ A_3 &= \left\{\delta_{\tilde{x}} + \delta_{\tilde{y}} + \delta_{\tilde{z}} \,\middle|\, \tilde{x} \in \mathbb{B}_\varepsilon(x), \tilde{y} \in \mathbb{B}_\varepsilon(y), \tilde{z} \in \mathbb{B}_\varepsilon(z)\right\}. \end{aligned}$$

Then $\Pi_1(\{\emptyset\})$, $\Pi_1(A_i) > 0$ for $i = 1, 2, 3$. Hence, according to our assumption, $h_{\boldsymbol{\theta}}(\emptyset) = h_{\tilde{\boldsymbol{\theta}}}(\emptyset)$ and, moreover, there are $x_1, x_2, x_3 \in \mathbb{B}_\varepsilon(x)$, $y_2, y_3 \in \mathbb{B}_\varepsilon(y)$, $z_3 \in \mathbb{B}_\varepsilon(z)$, such that for

$$\zeta_1 := \delta_{x_1}, \quad \zeta_2 := \delta_{x_2} + \delta_{y_2} \quad \zeta_3 := \delta_{x_3} + \delta_{y_3} + \delta_{z_3}$$

we have $h_{\boldsymbol{\theta}}(\zeta_i) = h_{\tilde{\boldsymbol{\theta}}}(\zeta_i)$ for $i = 1, 2, 3$. Furthermore, due to the triangle inequality and Equation (4.27) we have

$$\begin{aligned}\|x_2 - y_2\| \leq \|x_2 - x\| + \|x - y\| + \|y - y_2\| < \varepsilon + cR + \varepsilon \\ \leq \frac{R}{2}(1-c) + cR = \frac{R}{2}(1+c) < R\end{aligned}$$

as $c < 1$. Similarly it follows that $\|x_3 - y_3\| < R$, $\|x_3 - z_3\| < R$. Additionally, we have

$$\|x - y\| \leq \|x - x_2\| + \|x_2 - y_2\| + \|y_2 - y\| < 2\varepsilon + \|x_2 - y_2\|.$$

Thus, it follows with Equation (4.27) hat

$$\|x_2 - y_2\| > \|x - y\| - 2\varepsilon = cR - 2\varepsilon \geq cR - (cR - r) = r$$

and similarly we have $\|x_3 - y_3\| > r$, $\|x_3 - z_3\| > r$. Moreover,

$$\|y - z\| \leq \|y - y_3\| + \|y_3 - z_3\| + \|z_3 - z\| < 2\varepsilon + \|y_3 - z_3\|.$$

Using again Equation (4.27) we obtain

$$\begin{aligned}\|y_3 - z_3\| > \|y - z\| - 2\varepsilon = 2cR - 2\varepsilon \\ \geq 2cR - \frac{R}{2}(2c-1) = R\left(c + \frac{1}{2}\right) > R\end{aligned}$$

as $c > \frac{1}{2}$. Then, using (a), it follows that $\boldsymbol{\theta} = \tilde{\boldsymbol{\theta}}$, in contradiction to our assumption. Thus, the model is identifiable.

■

Knowing that, if $\mathfrak{X}$ is large enough, the model parameters in the superposition model are identifiable, we set out to estimate them for a given point pattern. For most point processes, such as the interaction processes considered in Section 4.2, maximum likelihood estimates (MLE) are analytically not feasible and computationally expensive due to the unknown normalising constant. Notably, even the evaluation of the unnormalised likelihood is already very time consuming (approximately of the same order as the conditional intensity, see the discussion below), making the computation of MLEs via Monte Carlo methods appear only little attractive.

Moreover, MLEs for spatial point processes in general do not exhibit the usual asymptotic properties, making them less popular than other methods. It is, nevertheless, worth pointing out that for point processes in a small neighbourhood around the Poisson process, i.e. if we observe only little interaction, there are results for consistency and asymptotic normality, see Dereudre et al. (2017) (where also Strauss processes with hard core are discussed) or Jensen (1993). For a more general discussion about MLE on point processes we refer to Møller & Waagepetersen (2003, 2007) and the references therein.

For processes with a likelihood containing an intractable normalising constant, pseudolikelihood methods are often employed since they remove the need of estimating the normalising constant. These methods use the conditional intensity defined as follows:

Definition 4.21 (Conditional intensity)**.** For a point process Z with density $h_{\boldsymbol{\theta}}$ w.r.t. Π_1 depending on some parameter $\boldsymbol{\theta} \in \Theta$ we define

$$c_{\boldsymbol{\theta}}(\cdot \mid \zeta) : \mathfrak{X} \to \mathbb{R}_+, \qquad z \mapsto \frac{h_{\boldsymbol{\theta}}(\zeta \cup \{z\})}{h_{\boldsymbol{\theta}}(\zeta \setminus \{z\})} \tag{4.28}$$

as the (Papangelou) *conditional intensity of* Z *given* ζ. Roughly speaking, $c_{\boldsymbol{\theta}}(z \mid \zeta)\,\mathrm{d}z$ is the conditional probability of finding a point in an infinitesimal neighbourhood around z given that the rest of the point process coincides with ζ. Notably, the normalising constants of $h_{\boldsymbol{\theta}}$ cancel out in (4.28).

Remark 4.22. It is sufficient to compute $c_{\boldsymbol{\theta}}(u \mid \zeta)$ for $u \notin \zeta$ because for $z \in \zeta$ we have

$$c_{\boldsymbol{\theta}}(z \mid \zeta) = \frac{h_{\boldsymbol{\theta}}(\zeta)}{h_{\boldsymbol{\theta}}(\zeta \setminus \{z\})} = \frac{h_{\boldsymbol{\theta}}((\zeta \setminus \{z\}) \,\dot\cup\, \{z\})}{h_{\boldsymbol{\theta}}(\zeta \setminus \{z\})} = c_{\boldsymbol{\theta}}(z \mid \zeta \setminus \{z\}), \tag{4.29}$$

and the right hand side (4.29) can then be computed using the formula for $u \notin \zeta$.

For densities which are log-linear in the parameters, the conditional intensity often has a very simple form.

Example 4.23. Let $\zeta \in \mathfrak{N}$ be a point pattern and let $u \in \mathfrak{X} \setminus \zeta$. Note that the results below also apply to $z \in \zeta$, cf. Remark 4.22.

(a) For $\Xi \sim \mathrm{Pop}(\lambda)$ with $\lambda : \mathfrak{X} \to \mathbb{R}_+$ we have

$$c_\lambda(u \mid \zeta) = \frac{\prod_{x \in \zeta \cup \{u\}} \lambda(x)}{\prod_{x \in \zeta} \lambda(x)} = \lambda(u),$$

i.e. the conditional intensity coincides with the intensity of a Poisson process and does only depend on the location considered but not on the surrounding point pattern ζ. This again reflects the spatial independence property of the Poisson process.

(b) Let $\boldsymbol{\theta} = (\beta, \gamma)$. For $\mathrm{H} \sim \mathrm{Strauss}(\beta, \gamma, R)$ holds

$$c_{\boldsymbol{\theta}}(u \mid \zeta) = \frac{\prod_{x \in \zeta \cup \{u\}} \beta(x) \gamma^{s_R(\zeta \cup \{u\})}}{\prod_{x \in \zeta} \beta(x) \gamma^{s_R(\zeta)}} = \beta(u) \gamma^{t_R(u|\zeta)},$$

where

$$t_R(u \mid \zeta) := s_R(\zeta \cup \{u\}) - s_R(\zeta) = \sum_{v \in \zeta} \mathbf{1}\left(\|u - v\| \leq R\right) \tag{4.30}$$

is the number of R-close points of u in ζ.

(c) In general, the superposition $\mathrm{Z} = \Xi \dot{\cup} \mathrm{H}$ of two independent point processes Ξ, H does not have a log-linear density anymore, even if Ξ and H do. Let $\Xi \sim \mathrm{Pop}(\lambda)$, $\mathrm{H} \sim \mathrm{StraussHard}(\beta, \gamma, r, R)$ and $\boldsymbol{\theta} = (\lambda, \beta, \gamma)$. Then, cf. Lemma 4.18, the conditional intensity of Z is

$$c_{\boldsymbol{\theta}}(u \mid \zeta) = \frac{\sum_{\eta \subseteq \zeta \cup \{u\}} \mathbf{1}(d_{\min}(\eta) > r) \prod_{x \in \eta} \beta(x) \gamma^{s_R(\eta)} \lambda^{|(\zeta \cup \{u\}) \setminus \eta|}}{\sum_{\eta \subseteq \zeta} \mathbf{1}(d_{\min}(\eta) > r) \prod_{x \in \eta} \beta(x) \gamma^{s_R(\eta)} \lambda^{|\zeta \setminus \eta|}}. \tag{4.31}$$

Definition 4.24 (Maximum pseudolikelihood). For a sample $\zeta = \{z_1, z_2, \ldots, z_n\}$ of a point process Z which has density $h_{\boldsymbol{\theta}}$ w.r.t. Π_1 depending on some parameter vector $\boldsymbol{\theta} \in \Theta$ we call the function

$$p\ell(\cdot \mid \zeta) : \Theta \to \mathbb{R}_+, \quad \boldsymbol{\theta} \mapsto p\ell(\boldsymbol{\theta} \mid \zeta) = \exp\left(-\int_{\mathfrak{X}} c_{\boldsymbol{\theta}}(u \mid \zeta)\, \mathrm{d}u\right) \prod_{i=1}^{n} c_{\boldsymbol{\theta}}(z_i \mid \zeta) \tag{4.32}$$

with $c_{\boldsymbol{\theta}}$ the conditional intensity (depending on $\boldsymbol{\theta}$), the *pseudolikelihood* of $\boldsymbol{\theta}$ given ζ, PL for short. A vector $\boldsymbol{\theta}^* \in \Theta$ with

$$\boldsymbol{\theta}^* \in \operatorname*{argmax}_{\boldsymbol{\theta} \in \Theta} p\ell(\boldsymbol{\theta} \mid \zeta) = \operatorname*{argmax}_{\boldsymbol{\theta} \in \Theta} \left(\prod_{i=1}^{n} c_{\boldsymbol{\theta}}(z_i \mid \zeta)\right) \exp\left(-\int_{\mathfrak{X}} c_{\boldsymbol{\theta}}(u \mid \zeta)\, \mathrm{d}u\right) \tag{4.33}$$

$$= \operatorname*{argmax}_{\boldsymbol{\theta} \in \Theta} \sum_{i=1}^{n} \log c_{\boldsymbol{\theta}}(z_i \mid \zeta) - \int_{\mathfrak{X}} c_{\boldsymbol{\theta}}(u \mid \zeta)\, \mathrm{d}u \tag{4.34}$$

is referred to as *maximum pseudolikelihood estimator* for $\boldsymbol{\theta}$, MPLE for short. As the conditional intensity does not contain the normalising constant of $h_{\boldsymbol{\theta}}$ anymore, neither does (4.33). For Gibbs processes with a log-linear density like in Example 4.23 (a) and (b), the point process model can then often be written as a generalised linear model and fit using a Berman-Turner device, cf. Berman & Turner (1992); Baddeley & Turner (2000). For models in which the density belongs to such a family MPLEs are consistent (converging to the true parameter) and asymptotically normal[13] under suitable conditions as noted in Baddeley & Turner (2000); Ba & Coeurjolly (2020). Generalisations beyond exponential families are still part of on-going research. Coeurjolly & Drouilhet (2010) considered consistency and asymptotic normality in models which do not-necessarily belong to an exponential family, however, they assume the process to be stationary which is concerning fingerprints no realistic assumption. Apart from that, in regard to fingerprint applications, the meaning of asymptotic results is in so far limited as the observed minutiae patterns are restricted to a region of interest which is at the most the size of a human finger.

Alternatively, one could think of extending the observation window by combining the ROIs of several fingerprints. By doing so, the data has to be assumed to exist in an unbounded window. In this case, (Baddeley & Turner, 2000, Section 9) mention that MPLEs feature edge effects. While for homogeneous processes, Baddeley & Turner (2000) propose some remedies for edge effects, an approach for inhomogeneous processes seems to be far more difficult, see also the discussion in Remark 4.11. Moreover, for this approach parameter inference methods would rather aim for parameter estimation of a "mean fingerprint" than of the individual fingerprints. The data analysis in Chapter 7 suggests that this approach yields potential for further research, however, a large variance in the posterior distribution of the model parameters might introduce major challenges.

Beyond that, concerning asymptotic inference for replicated point patterns, i.e. several independent point patterns from the same process (possibly depending on external factors or covariates such as the intensity field which might be different for each point pattern), to the best of our knowledge only little is known; especially if the point patterns are only assumed to be independent but not identically distributed. See e.g. (Baddeley et al., 2015, Chapter 16) for methods of explorative data analysis and descriptive statistics of replicated point patterns.

Remark 4.25 (Relation between MLE and MPLE)**.** Intuitively, as the pseudolikelihood only depends on the local dependence structure, global information may better be taken into account using the likelihood function. Optimality of the MLE and inefficiency (i.e. larger asymptotic variance) of the MPLE is established in Mase (1992) under very restrictive conditions (small neighbourhood around the Poisson process, see above). Huang & Ogata (1999) propose to compute the MPLE and then applying one Newton-Raphson step towards maximising the likelihood. Note, that (4.32) coincides with the likelihood of a Poisson process

[13] in the context of point processes, in order to "increase the sample size" we let the observation window expand to $\mathbb{R}^d$

when $c_{\boldsymbol{\theta}}(z \mid \zeta)$ does not depend on z. Hence, for point processes with weak interaction the MPLE and the MLE may be expected to be close to each other, cf. Baddeley & Turner (2000).

Remark 4.26 (Computation of MPLE)**.** For practical implementations of the MPLEs of Gibbs processes with a log-linear density, cf. the spatstat package of Baddeley & Turner (2005) in R Core Team (2021) and (Møller & Waagepetersen, 2003, Section 9.2.2). For the considered superposition model no previous software was available. The algorithm as presented below is implemented in R and can be found in Section 6.5 as well as in the repository of Wieditz (2021a).

Since the optimisation problem (4.33) is usually solved numerically, it is crucial for the pseudolikelihood to be efficiently computable. However, the numerator as well as the denominator of (4.39) contain of order $2^{|\zeta|}$ many summands, making computation infeasible without further ado. In the application of fingerprints, we can make use of the fact, that only a few of our points in fact do interact with each other. We propose a fairly efficient method to compute the MPLE in the superposition model in such cases.

We reduce the computation of the pseudolikelihood of the superimposed pattern $\zeta = \xi \,\dot{\cup}\, \eta$ with $\xi \sim \mathrm{Pop}(\lambda)$, $\eta \sim \mathrm{StraussHard}(\beta, \gamma, r, R)$, to the computation of $c_{\boldsymbol{\theta}}(z_i \mid \zeta \setminus \{z_i\})$ for $i = 1, 2, \ldots, n$. To this end, we follow a graph-theoretic approach which is based on graphs induced by sub-patterns $\eta \subseteq \zeta$. Some preliminaries on graph theory needed for this section are provided in Section A.3.

Definition 4.27 (Interaction graph, neighbourhood)**.** Let $\eta \subseteq \zeta$ and $\rho > 0$. We call the (undirected, simple) graph $G_\rho[\eta] = (V(\eta), E_\rho(\eta))$ with

$$V(\eta) = \eta, \quad E_\rho(\eta) = \{uv \in \mathfrak{P}_2(\eta) \mid \|u - v\| \leq \rho\}$$

the *interaction graph* induced by η with interaction range ρ. Here, $\mathfrak{P}_2(\eta)$ denotes the set of all two-element subsets of $\eta \subseteq \zeta$.

An example for the interaction graph of the fingerprint from Figure 2.1 is depicted in Figure 4.6. Clearly, the largest component contains only 4 out of 31 vertices in total.

For $\eta \subseteq \zeta$ and $u \in \mathbb{R}^2$ we define by

$$N_\rho(u \mid \eta) := \{w \in \eta : \|u - w\| \leq \rho\} \tag{4.35}$$

the (spatial) *neighbourhood* of radius ρ of u in η.

Then, cf. (4.8) and (4.30), we have that $s_\rho(\eta) = |E_\rho(\eta)|$ is the number of edges in the subgraph induced by η and $t_\rho(u \mid \eta) = |N_\rho(u \mid \eta)|$ is the number of ρ-close neighbours of u in η. Note, that for obtaining all the interaction graphs of $\eta \subseteq \zeta$ it is sufficient to compute $G_\rho[\zeta]$ since we can obtain $G_\rho[\eta]$ from $G_\rho[\zeta]$ by deleting the nodes in $\zeta \setminus \eta$ and all edges incident with these vertices.

If $u \in \eta$, then the spatial neighbourhood (4.35) of u in η coincides with the graph neighbourhood in $G_\rho[\eta]$, i.e.

$$N_\rho(u \mid \eta) = \{w \in \eta : uw \in E_\rho(\eta)\}. \tag{4.36}$$

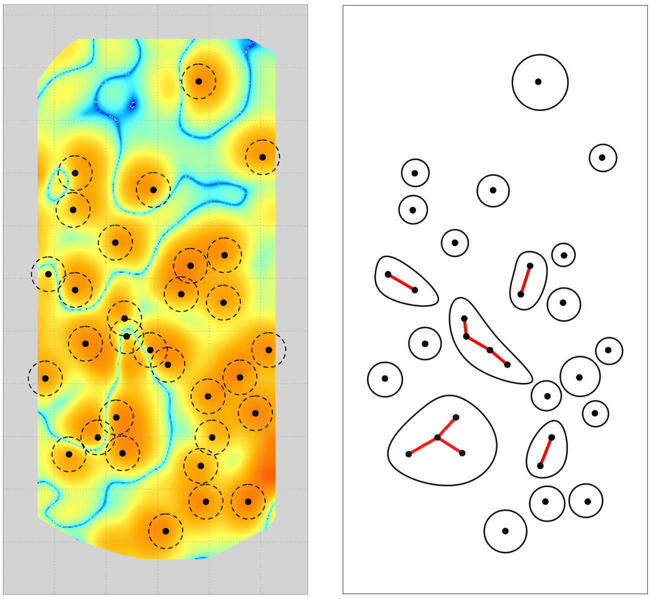


Figure 4.6: *Construction of the interaction graph (right) of the fingerprint from Figure 2.1. Left: The dashed circles have a radius $R/2$ where R is three times the average inter-ridge distance. Right: Two vertices (minutiae, black dots) are connected (red line) if and only if they are closer than R, i.e. the circles on the left overlap. The components of the graph are outlined in black (contour size is of no importance).*

For $u \in \mathbb{R}^2 \setminus \eta$ this is in general not true. However, we can compute the spatial neighbourhood (4.35) using a graph theoretic approach (as in (4.36)) by considering the graph neighbourhood in the interaction graph induced by the superimposed pattern $\eta \cup \{u\}$, i.e. for $u \in \mathbb{R}^2 \setminus \eta$ we have

$$N_\rho(u \mid \eta) = \{w \in \eta : uw \in E_\rho(\eta \cup \{u\})\}.$$

It turns out, that restricting the computation of the conditional intensity to components (outlined in black in the right panel in Figure 4.6) of the interaction graph has the potential to reduce the computational workload. We state here only the results and refer to Section 4.6 for the proofs.

Theorem 4.28. *Let* $\mathrm{Z} = \Xi \dot{\cup} \mathrm{H}$ *with* $\Xi \sim \mathrm{Pop}(\lambda)$, $\mathrm{H} \sim \mathrm{StraussHard}(\beta, \gamma, R, r)$ *and* $\zeta \sim \mathrm{Z}$ *be a point pattern. Moreover, let* $z \in \zeta$ *and denote by* $G_k = (\zeta_k, E_R(\zeta_k))$, $k = 1, 2, \dots, K$, *the components of* $G_R[\zeta \setminus \{z\}]$. *Then, the conditional intensity of* Z *satisfies*

$$
\begin{aligned}
&c_{\boldsymbol{\theta}}(z \mid \zeta) = c_{\boldsymbol{\theta}}(z \mid \zeta \setminus \{z\}) \\
&= \lambda + \beta(z) \prod_{k=1}^{K} \left(\frac{\sum_{\eta_k \subseteq \zeta_k} \left(\prod_{v \in \eta_k} \frac{\beta(v)}{\lambda} \right) \gamma^{|E_R(\eta_k)|} \gamma^{|N_R(z|\eta_k)|} \mathbf{1}\left(E_r(\eta_k \cup \{z\}) = \emptyset\right)}{\sum_{\eta_k \subseteq \zeta_k} \left(\prod_{v \in \eta_k} \frac{\beta(v)}{\lambda} \right) \gamma^{|E_R(\eta_k)|} \mathbf{1}\left(E_r(\eta_k) = \emptyset\right)} \right).
\end{aligned}
$$

Remark 4.29. Notice, that for the computation of the MPLE problem we have to compute the components of $\zeta \setminus \{z_i\}$ for every vertex $z_i \in \zeta$ individually, $i = 1, 2, \dots, n$. However, this has to be done only once in advance. Afterwards, the evaluation of the PL for one particular value of (β, γ, λ) requires

$$
\mathcal{O}\left(\sum_{i=1}^{n} K_i 2^{\max\limits_{k=1,\dots,K_i} |V_k^i|} \right)
$$

many operations. Here, K_i is the number of components $G_k^i = (V_k^i, E_{R,k}^i)$ of $G_k = G_R[\zeta \setminus \{z_i\}]$, $i \in \{1, 2, \dots, n\}$, $k \in \{1, 2, \dots, K_i\}$.

A similar argument as above holds true for the computation of the integral in (4.33) which is in practice approximated by the (multivariate) Riemann sum

$$
\int_{\mathfrak{X}} c_{\boldsymbol{\theta}}(u \mid \zeta) \, \mathrm{d}u \approx \frac{|\mathfrak{X}|}{L} \sum_{\ell=1}^{L} c_{\boldsymbol{\theta}}(u_\ell \mid \zeta), \tag{4.37}
$$

where $u_\ell \in \mathfrak{X}$, $\ell = 1, 2, \dots, L$ form some grid covering $\mathfrak{X}$ appropriately, e.g. a rectangular grid, see also Baddeley & Turner (2000). Again, since the structure of the graphs considered for each u_ℓ does not depend on $\boldsymbol{\theta}$, this can be precomputed in advance speeding up the computation of the MPLE.

To conclude this section, we like to outline some properties of the MPLE and issues coming along with the computation for a simulated example in a homogeneous setting. An example for applications in fingerprints is provided in Chapter 7.

Example 4.30 (MPLE for superpositions of homogeneous Strauss and Poisson processes)**.** We consider the sample in Figure 4.7 stemming from the superposition Z of $\Xi \sim \mathrm{Pop}(\lambda_0)$ and $\mathrm{H} \sim \mathrm{StraussHard}(\beta_0 \mu, \gamma_0, 0.03, 0.1)$ on the unit square $\mathfrak{X} = [0, 1]^2$. The true parameter $\boldsymbol{\theta}^{(0)} = (\lambda_0, \beta_0, \gamma_0) = (12, 1, 0.4)$, the activity $\mu = 42$ and the interaction radii were chosen to resemble the parameters in a real fingerprint application. Indeed, we have 17 / 45 Poisson points ($\approx 31\%$) which is slightly more than the expected amount of 28%, cf. Remark 3.16.

Three cuts through the parameter space, each at one of the true parameter values can be seen in Figure 4.8. The true parameter is marked with a black cross ($\times$). MPLEs for three initial points $\boldsymbol{\theta}^{(1)} = (20, 1, 0.75)$ (expected low interaction), $\boldsymbol{\theta}^{(2)} = \boldsymbol{\theta}^{(0)} = (12, 1, 0.4)$ (true value), $\boldsymbol{\theta}^{(3)} = (30, 2, 0.1)$ (expected strong repulsion) for the optimisation algorithm as yellow circle ($\circ$), blue triangle (Δ) and red square ($\Box$), respectively.

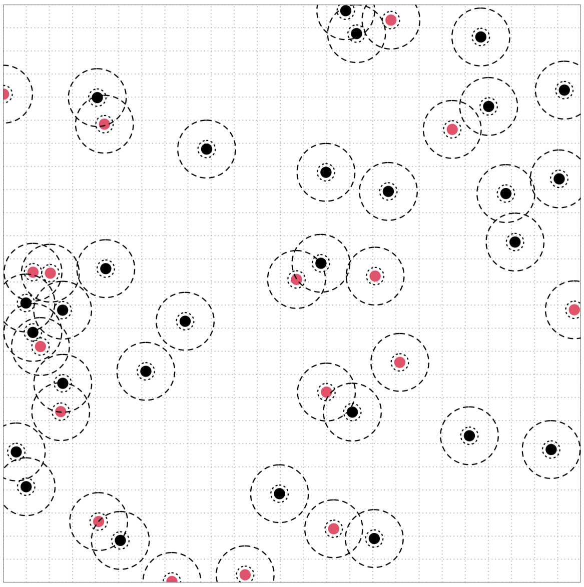
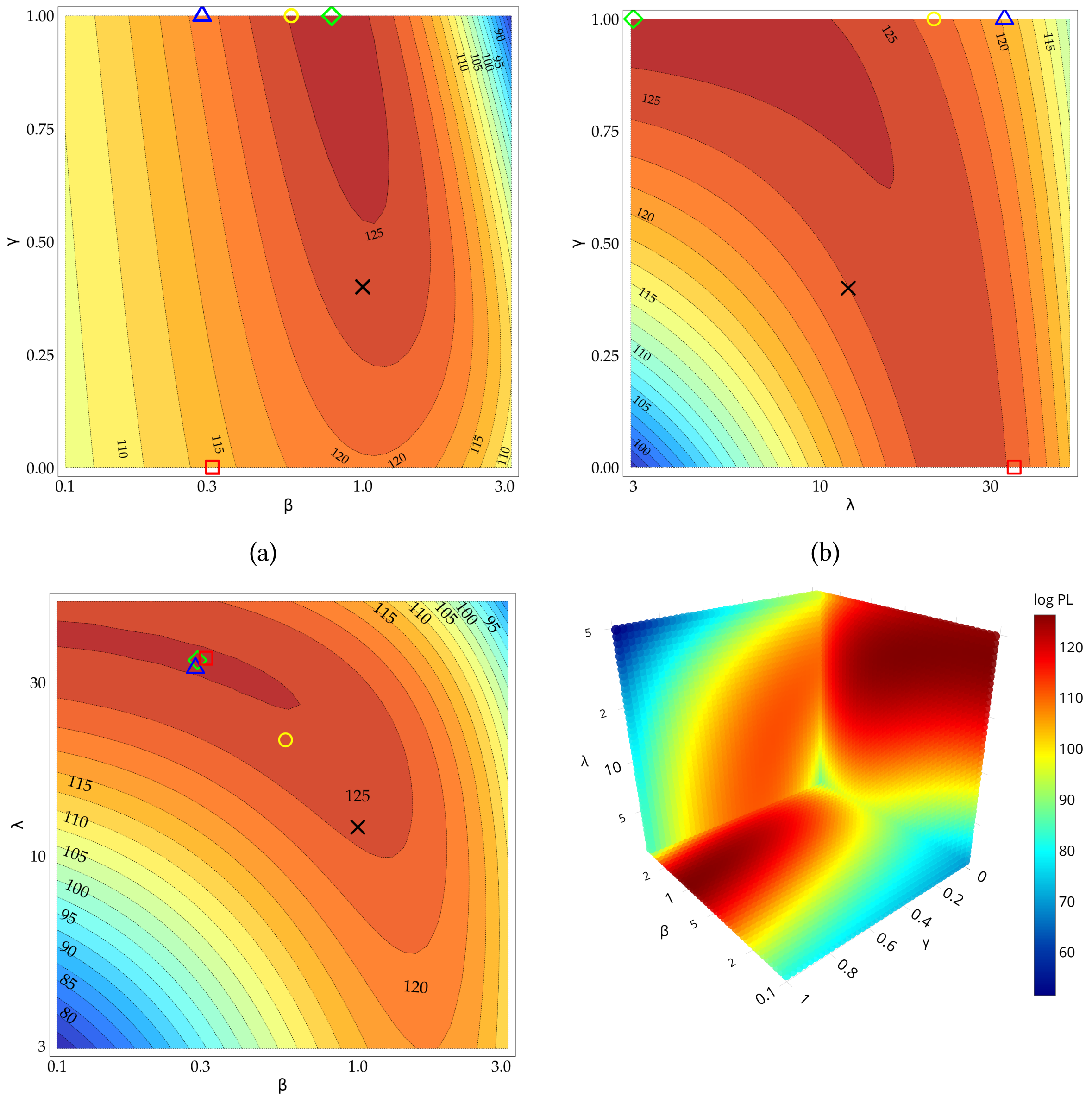


Figure 4.7: *Sample point pattern of* Z*; Strauss points in black, Poisson points in red. Interaction radii $r < R$ as dotted and dashed circles, respectively. The grid $\{u_\ell\}$ in the background of size 25×25 is used for the computation of the integration constant.*

The MPLE within each cut (i.e. conditional on knowing the respective third component) is indicated as green diamond ($\diamond$) each. The computed estimators with corresponding function values are listed in Table 4.1. For the optimisation we employed the `optim` function of `R` using the L-BFGS-B method, cf. Zhu et al. (1997), for solving box-constrained optimisation problems with standard precision 10^{-8}. This method relies on gradient descent steps and terminates when the norm of the gradient of the objective function is small. Hence, in general only local minima are to be found.

Indeed, as visible in Figure 4.8, the log-pseudolikelihood for the considered point pattern is not concave. The plot suggests that the PL has a very flat banana-shaped hill. We have reason to believe that there are probably at least two local maxima: one on the lower left and one in the middle on the right hand side of Figure 4.8d. Depending on where we initialise the optimisation, we can end up in one or the other region of the parameter space. Notably, the local maximum on the lower left is close to the case where the parameters of the model are not identifiable anymore ($\gamma = 1$) raising a further issue of this approach.

We like to remark, that the value of the PL for all computed MPLEs is approximately the same, see Table 4.1, even though the values for the initial value $\boldsymbol{\theta}^{(3)}$ and the optimisation conditional on $\gamma = 0.4$ is slightly larger. This suggests that all these parameter choice are about “equally good” in terms of PL. The MPLE approach, however, does not provide



(a)

(b)

(c)

(d)

Figure 4.8: *Top left to bottom right: Log-PL on three cuts through the parameter space for fixed $\lambda = \lambda_0$ (a), $\beta = \beta_0$ (b) and $\gamma = \gamma_0$ (c), respectively. The projection of the true parameter to the slices is marked as $\times$, the projected MPLEs within the shown slice (with the respective third component fixed at the true value) as green diamond and the projections of three MPLEs to the slices for different initial values from Example 4.30 for the optimisation procedure as yellow circle, blue triangle and red square, respectively. An aerial view on the log-PL over the 3D parameter space is depicted in (d) (visible are the three slices where $\lambda = 3$, $\gamma = 0$ or $\beta = 3$, respectively). An interactive graphic of the 3D log-PL can be found in Wieditz (2021b). Note that the numerical optimisation gives reason to believe that there are two local maxima for very small $\gamma \approx 0$ (approx. hard core process) and $\gamma = 1$ (unidentifiable case).*

	$\hat{\lambda}$	$\hat{\beta}$	$\hat{\gamma}$	$\log p\ell(\hat{\boldsymbol{\theta}} \mid \zeta)$
initial point $\boldsymbol{\theta}^{(1)}$	20.8513	0.5745	1.0000	126.2998
initial point $\boldsymbol{\theta}^{(2)}$	32.8837	0.2885	1.0000	126.2998
initial point $\boldsymbol{\theta}^{(3)}$	34.9765	0.3127	0.0000	126.3230
fixed $\boldsymbol{\theta}_1 = \lambda_0$	12.0000	0.7857	1.0000	126.2998
fixed $\boldsymbol{\theta}_2 = \beta_0$	3.0000	1.0000	1.0000	126.2998
fixed $\boldsymbol{\theta}_3 = \gamma_0$	34.5290	0.2930	0.4000	126.3023
true $\boldsymbol{\theta} = \boldsymbol{\theta}^{(0)}$	12.0000	1.0000	0.4000	124.1730

Table 4.1: *Computed MPLEs from different starting points (row 1–3) and within the two-dimensional subspaces when one component of $\boldsymbol{\theta}$ is set to the true value (row 4–6) with corresponding log-PL. The value of the log-PL for the true parameter $\boldsymbol{\theta}^{(0)}$ in the last row. All values are rounded to four decimal places.*

any information about the number of “equally good” parameter choices and associated uncertainties. This has to be kept in mind when using MPLE information in practice.

As a final remark, we like to note that the main computational workload comes from the computation of the integral $\int_{\mathfrak{X}} c_{\boldsymbol{\theta}}(u \mid \zeta)\,\mathrm{d}u$, see also Table 6.3 on p. 125. In view of the application of fingerprints, the used grid of 25×25 lattice points might be too coarse-meshed. Increasing the number of lattice points might improve the approximation of the integral, however, also leads to larger components and hence to an increase in computational time. The optimal choice of a mesh is beyond the scope of this thesis; we refer to the literature for improved versions, e.g. (Baddeley et al., 2015, Section 9.8).

The following Section 4.6 contains the remaining proofs of results stated in this chapter. The reader might skip this part and proceed with Chapter 5 in which we introduce an approach which takes uncertainties in the choice of parameters into account.

4.6 Proofs

In this section we present the proofs left out in the first part of Chapter 4. To start with, we provide three auxiliary results which are needed for the proof of Lemma 4.18. For the proof we need moreover the Poisson thinning theorem as stated below.

Theorem 4.31 (Poisson thinning theorem). *Suppose $\Xi \sim \operatorname{Po}(\mathfrak{X}, \lambda)$ is subject to independent thinning with retention probabilities $p(x)$ for $x \in \mathfrak{X}$ and let $\lambda_{\text{thin}}(x) = p(x)\lambda(x)$, $x \in \mathfrak{X}$. Then, Ξ_{thin} and $\Xi \setminus \Xi_{\text{thin}}$ are independent Poisson processes with intensities λ_{thin} and $\lambda - \lambda_{\text{thin}}$, respectively.*

Proof. For the proof, we refer to (Møller & Waagepetersen, 2003, Proposition 3.7). ■

Lemma 4.32. *Let*

$$\Phi : \mathfrak{N} \times \mathfrak{N} \to \mathfrak{N} \times \mathfrak{N}, \quad \left(\sum_{n=1}^{N} \delta_{x_n}, \sum_{m=1}^{M} \delta_{y_m} \right) \mapsto \left(\sum_{n=1}^{N} \delta_{x_n}, \sum_{n=1}^{N} \delta_{x_n} + \sum_{m=1}^{M} \delta_{y_m} \right).$$

Then, for $A, B \in \mathcal{N}$ we have

$$(\Pi_1 \otimes \Pi_1) \left(\Phi^{-1}(A \times B) \right) = \mathrm{e}^{-|\mathfrak{X}|} \int_B \sum_{\xi \subseteq \zeta} \delta_\xi(A) \Pi_1(\mathrm{d}\zeta).$$

Proof. The mapping Φ is bijective with inverse

$$\Phi^{-1} : \left\{ \left(\sum_{n=1}^{N} \delta_{x_n}, \sum_{\ell=1}^{L} \delta_{z_\ell} \right) \in \mathfrak{N} \times \mathfrak{N} : |\{n \mid x_n = x\}| \leq |\{\ell \mid z_\ell = x\}|, x \in \mathfrak{X} \right\} \to \mathfrak{N} \times \mathfrak{N},$$
$$\left(\sum_{n=1}^{N} \delta_{x_n}, \sum_{\ell=1}^{L} \delta_{z_\ell} \right) \mapsto \left(\sum_{n=1}^{N} \delta_{x_n}, \sum_{\ell=1}^{L} \delta_{z_\ell} - \sum_{n=1}^{N} \delta_{x_n} \right).$$

Denote by $\Pi_2 := \mathrm{Pop}(2)$ the Poisson process with intensity 2. Let $\mathrm{Z} \sim \Pi_2$ and Z_{thin} be a process obtained from Z via thinning with retention probability $\frac{1}{2}$. From Theorem 4.31 follows that for independent $\mathrm{H}_1, \mathrm{H}_2 \sim \Pi_1$ we have $(\mathrm{Z}_{\text{thin}}, \mathrm{Z} \setminus \mathrm{Z}_{\text{thin}}) \sim (\mathrm{H}_1, \mathrm{H}_2)$. Then, for $A, B \in \mathcal{N}$ we have:

$$\begin{aligned}(\Pi_1 \otimes \Pi_1) \left(\Phi^{-1}(A \times B) \right) &= \mathbb{P}((\mathrm{H}_1, \mathrm{H}_2) \in \Phi^{-1}(A \times B)) = \mathbb{P}\left(\Phi(\mathrm{H}_1, \mathrm{H}_2) \in A \times B\right) \\ &= \mathbb{P}\left(\Phi(\mathrm{Z}_{\text{thin}}, \mathrm{Z} \setminus \mathrm{Z}_{\text{thin}}) \in A \times B\right) = \mathbb{P}\left((\mathrm{Z}_{\text{thin}}, \mathrm{Z}) \in A \times B\right) \\ &= \int_B \mathbb{P}\left(\mathrm{Z}_{\text{thin}} \in A \mid \mathrm{Z} = \zeta\right) \mathbb{P}_{\mathrm{Z}}(\mathrm{d}\zeta)\end{aligned}$$

where the last equality follows from the definition of the conditional probability. Now, we have

$$\mathbb{P}\left(\mathrm{Z}_{\text{thin}} \in A \mid \mathrm{Z} = \zeta\right) = \frac{|\{\xi \subseteq \zeta : \xi \in A\}|}{|\{\xi \subseteq \zeta\}|} = \frac{\sum_{\xi \subseteq \zeta} \mathbf{1}(\xi \in A)}{2^{|\zeta|}}.$$

Moreover, due to $\mathrm{Z} \sim \Pi_2$ and using

$$\frac{\mathrm{d}\mathbb{P}_{\mathrm{Z}}}{\mathrm{d}\Pi_1}(\zeta) = \frac{\mathrm{d}\Pi_2}{\mathrm{d}\Pi_1}(\zeta) = \mathrm{e}^{-|\mathfrak{X}|} 2^{|\zeta|},$$

see Lemma 4.8, we obtain

$$\begin{aligned}\int_B \mathbb{P}\left(\mathrm{Z}_{\mathrm{thin}} \in A \mid \mathrm{Z} = \xi\right) \mathbb{P}_Z(\mathrm{d}\xi) &= \int_B \left(\frac{\sum_{\xi \subseteq \zeta} \mathbf{1}(\xi \in A)}{2^{|\zeta|}}\right) \mathbb{P}_{\mathrm{Z}}(\mathrm{d}\zeta) \\ &= \int_B \left(\frac{\sum_{\xi \subseteq \zeta} \mathbf{1}(\xi \in A)}{2^{|\zeta|}}\right) \mathrm{e}^{-|\mathfrak{x}|} 2^{|\zeta|} \Pi_1(\mathrm{d}\zeta) \\ &= \mathrm{e}^{-|\mathfrak{x}|} \int_B \sum_{\xi \subseteq \zeta} \delta_\xi(A) \Pi_1(\mathrm{d}\zeta),\end{aligned}$$

which yields the assertion. ■

Lemma 4.33. *Let Φ be the mapping from Lemma 4.32 and $h : \mathfrak{N} \times \mathfrak{N} \to \mathbb{R}_+$ be $(\mathcal{N} \otimes \mathcal{N}, \mathcal{B}(\mathbb{R}))$-measurable. Then,*

$$\iint_{\mathfrak{N} \times \mathfrak{N}} h(\xi, \zeta) \left(\Pi_1 \otimes \Pi_1\right) \Phi^{-1}(\mathrm{d}\xi,\, \mathrm{d}\zeta) = \int_{\mathfrak{N}} \mathrm{e}^{-|\mathfrak{x}|} \sum_{\xi \subseteq \zeta} h(\xi, \zeta) \Pi_1(\mathrm{d}\zeta). \tag{4.38}$$

Proof. We do the proof by employing measure theoretical induction.

(i) Let at first $A, B \in \mathfrak{N}$ and $h(\xi, \zeta) = \mathbf{1}_{A \times B}(\xi, \zeta) = \mathbf{1}_A(\xi)\mathbf{1}_B(\zeta)$. Then,

$$\iint_{\mathfrak{N} \times \mathfrak{N}} h(\xi, \zeta) \left(\Pi_1 \otimes \Pi_1\right) \Phi^{-1}(\mathrm{d}\xi,\, \mathrm{d}\zeta) = \left(\Pi_1 \otimes \Pi_1\right) \Phi^{-1}(A \times B)$$

and

$$\begin{aligned}\int_{\mathfrak{N}} \mathrm{e}^{-|\mathfrak{x}|} \sum_{\xi \subseteq \zeta} h(\xi, \zeta) \Pi_1(\mathrm{d}\zeta) &= \int_{\mathfrak{N}} \mathrm{e}^{-|\mathfrak{x}|} \sum_{\xi \subseteq \zeta} \mathbf{1}_A(\xi) \mathbf{1}_B(\zeta) \Pi_1(\mathrm{d}\zeta) \\ &= \int_B \mathrm{e}^{-|\mathfrak{x}|} \sum_{\xi \subseteq \zeta} \mathbf{1}_A(\xi) \Pi_1(\mathrm{d}\zeta)\end{aligned}$$

and Equation (4.38) follows from Lemma 4.32. Now, the system $\left\{A \times B \,\middle|\, A, B \in \mathfrak{N}\right\}$ is a λ-system and, moreover, an intersection-stable generator for $\mathcal{N} \otimes \mathcal{N}$. Hence, due to to Dynkin's π-λ-theorem (Klenke, 2020, Theorem 1.19) we obtain (4.38) even for arbitrary indicator functions $h = \mathbf{1}_D$ for $D \in \mathcal{N} \otimes \mathcal{N}$.

(ii) Since the integral is a linear operator, Equation (4.38) holds also true for linear combinations of indicator functions.

(iii) Moreover, note that the sum over all subsets of ζ on the right hand side of (4.38) is finite. Hence, monotone limits of linear combinations of indicator functions $\lim_{n \to \infty} \sum_{i=1}^{n} h_i(\xi, \zeta)$ can be pulled out of the limit and (4.38) follows by the monotonous convergence theorem.

■

Having found a formula for the density of superpositions of point proceses, we can derive the formula for the conditional intensity. We recall:

Theorem 4.28. *Let* $\mathrm{Z} = \Xi \,\dot\cup\, \mathrm{H}$ *with* $\Xi \sim \mathrm{Pop}(\lambda)$, $\mathrm{H} \sim \mathrm{StraussHard}(\beta, \gamma, R, r)$ *and* $\zeta \sim \mathrm{Z}$ *be a point pattern. Moreover, let* $z \in \zeta$ *and denote by* $G_k = (\zeta_k, E_R(\zeta_k))$, $k = 1, 2, \ldots, K$, *the components of* $G_R[\zeta \setminus \{z\}]$. *Then, the conditional intensity of* Z *satisfies*

$$\begin{aligned} c_{\boldsymbol{\theta}}(z \mid \zeta) &= c_{\boldsymbol{\theta}}(z \mid \zeta \setminus \{z\}) \\ &= \lambda + \beta(z) \prod_{k=1}^{K} \left(\frac{\sum_{\eta_k \subseteq \zeta_k} \left(\prod_{v \in \eta_k} \frac{\beta(v)}{\lambda}\right) \gamma^{|E_R(\eta_k)|} \gamma^{|N_R(z|\eta_k)|} \,\mathbf{1}\left(E_r(\eta_k \cup \{z\}) = \emptyset\right)}{\sum_{\eta_k \subseteq \zeta_k} \left(\prod_{v \in \eta_k} \frac{\beta(v)}{\lambda}\right) \gamma^{|E_R(\eta_k)|} \,\mathbf{1}\left(E_r(\eta_k) = \emptyset\right)} \right). \end{aligned}$$

Proof. We know

$$h_{\boldsymbol{\theta}}(\zeta) = \mathrm{e}^{-|\mathfrak{X}|} \sum_{\xi \subseteq \zeta} f_\lambda(\xi)\, g_{\beta,\gamma}(\zeta \setminus \xi),$$

where

$$f_\lambda(\xi) = \mathrm{e}^{(1-\lambda)|\mathfrak{X}|} \lambda^{|\xi|}, \qquad g_{\beta,\gamma}(\eta) = \alpha \left(\prod_{v \in \eta} \beta(v) \right) \gamma^{s_r(\eta)} \,\mathbf{1}\left(d_{\text{min}}(\eta) > r\right).$$

We show the claim at first for the superposition of a homogeneous Poisson and a homogeneous Strauss process. For $z \in \zeta$ we have

$$c_{\boldsymbol{\theta}}(z \mid \zeta) = \frac{h_{\boldsymbol{\theta}}(\zeta)}{h_{\boldsymbol{\theta}}(\zeta \setminus \{z\})} = \frac{\sum_{\eta \subseteq \zeta} \beta^{|\eta|} \gamma^{s_R(\eta)} \lambda^{|\zeta \setminus \eta|}}{\sum_{\eta \subseteq \zeta \setminus \{z\}} \beta^{|\eta|} \gamma^{s_R(\eta)} \lambda^{|(\zeta \setminus \{z\}) \setminus \eta|}} =$$

$$\begin{aligned} &= \frac{\sum_{\eta \subseteq \zeta \setminus \{z\}} \beta^{|\eta|} \gamma^{s_R(\eta)} \lambda^{|\zeta \setminus \eta|} + \sum_{\eta \subseteq \zeta \setminus \{z\}} \beta^{|\eta \,\dot\cup\, \{z\}|} \gamma^{s_R(\eta \,\dot\cup\, \{z\})} \lambda^{|(\zeta \setminus \{z\}) \setminus \eta|}}{\sum_{\eta \subseteq \zeta \setminus \{z\}} \beta^{|\eta|} \gamma^{s_R(\eta)} \lambda^{|(\zeta \setminus \{z\}) \setminus \eta|}} \\ &= \lambda + \beta \frac{\sum_{\eta \subseteq \zeta \setminus \{z\}} \left(\frac{\beta}{\lambda}\right)^{|\eta|} \gamma^{s_R(\eta) + t_R(z|\eta)}}{\sum_{\eta \subseteq \zeta \setminus \{z\}} \left(\frac{\beta}{\lambda}\right)^{|\eta|} \gamma^{s_R(\eta)}} = \lambda + \beta \frac{\sum_{\eta \subseteq \zeta \setminus \{z\}} \left(\frac{\beta}{\lambda}\right)^{|\eta|} \gamma^{|E_R(\eta)| + |N_R(z|\eta)|}}{\sum_{\eta \subseteq \zeta \setminus \{z\}} \left(\frac{\beta}{\lambda}\right)^{|\eta|} \gamma^{|E_R(\eta)|}}. \end{aligned} \tag{4.39}$$

It is now sufficient to show that the computation of the numerator and the denominator of (4.39) can both be decomposed into the computation of components of $G_R[\zeta \setminus \{z\}]$.

Let $K = 2$, i.e. $\zeta = \zeta_1 \,\dot\cup\, \zeta_2$. For the denominator define

$$\varphi : \mathfrak{P}_2(\zeta \setminus \{z\}) \to \mathbb{R}, \qquad \eta \mapsto \left(\frac{\beta}{\lambda}\right)^{|\eta|} \gamma^{|E_R(\eta)|}.$$

For $\eta_1 \subseteq \zeta_1, \eta_2 \subseteq \zeta_2$ we have $\varphi(\eta_1 \dot\cup \eta_2) = \varphi(\eta_1)\,\varphi(\eta_2)$. Then

$$\begin{aligned}\sum_{\eta\subseteq\zeta} \varphi(\eta) &= \sum_{\eta\subseteq\zeta} \varphi((\zeta_1 \cap \eta) \cup (\zeta_2 \cap \eta)) = \sum_{\eta\subseteq\zeta} \varphi(\zeta_1 \cap \eta)\,\varphi(\zeta_2 \cap \eta) \\ &= \sum_{\eta_1\subseteq\zeta_1} \sum_{\eta_2\subseteq\zeta_2} \varphi(\zeta_1 \cap (\eta_1 \cup \eta_2))\,\varphi(\zeta_2 \cap (\eta_1 \cup \eta_2)) \\ &= \sum_{\eta_1\subseteq\zeta_1} \sum_{\eta_2\subseteq\zeta_2} \varphi(\eta_1)\,\varphi(\eta_2) \\ &= \left(\sum_{\eta_1\subseteq\zeta_1} \varphi(\eta_1)\right)\left(\sum_{\eta_2\subseteq\zeta_2} \varphi(\eta_2)\right).\end{aligned}$$

For the numerator consider

$$\psi : \mathfrak{P}_2(\zeta \setminus \{z\}) \to \mathbb{R}, \qquad \eta \mapsto \left(\frac{\beta}{\lambda}\right)^{|\eta|} \gamma^{|E_R(\eta)|+|N_R(z|\eta)|}.$$

Since for $\eta_1 \subseteq \zeta_1, \eta_2 \subseteq \zeta_2$ we have

$$N_R(z \mid \eta_1 \dot\cup \eta_2) = \{v \in \eta_1 \dot\cup \eta_2 : \|v - z\| \leq R\} = N_R(z \mid \eta_1) \dot\cup N_R(z \mid \eta_2).$$

Analogously to the argumentation for φ we obtain hence $\psi(\eta_1 \dot\cup \eta_2) = \psi(\eta_1)\,\psi(\eta_2)$ and thus the same factorisation as above with φ replaced by ψ. Now, for $K \geq 3$ we obtain by induction

$$\begin{aligned}\sum_{\eta\subseteq\zeta} \psi(\eta) &= \prod_{k=1}^{K} \sum_{\eta_k\subseteq\zeta_k} \psi(\eta_k) = \prod_{k=1}^{K} \sum_{\eta_k\subseteq\zeta_k} \left(\frac{\beta}{\lambda}\right)^{|\eta_k|} \gamma^{|E_R(\eta_k)|+|N_R(z|\eta_k)|}, \\ \sum_{\eta\subseteq\zeta} \varphi(\eta) &= \prod_{k=1}^{K} \sum_{\eta_k\subseteq\zeta_k} \varphi(\eta_k) = \prod_{k=1}^{K} \sum_{\eta_k\subseteq\zeta_k} \left(\frac{\beta}{\lambda}\right)^{|\eta_k|} \gamma^{|E_R(\eta_k)|},\end{aligned}$$

and hence the claim for the homogeneous case.

For inhomogeneous β we substitute $\beta^{|\eta|}$ by $\prod_{v\in\eta} \beta(v)$. To include an additional hard core at distance $r \in (0, R)$ to the Strauss process, we have to check for every considered $\eta_k \subseteq \zeta_k$ that no two distinct points $u, v \in \eta_k$ are closer than r. Since $d_{\min}(\eta_k) > r \iff E_r(\eta_k) = \emptyset$, the assertion follows. ■

An identifiability result for the superposition model from Section 4.4 also holds in case of a non-constant activity function μ as follows:

Lemma 4.34 (Identifiability for inhomogeneous processes). *Consider two independent point processes* $\Xi \sim \mathrm{Pop}(\lambda)$, $\mathrm{H} \sim \mathrm{StraussHard}(\beta\mu, \gamma, R, r)$ *with* $\lambda > 0$, $\beta > 0$ *and* $\gamma \in [0, 1)$ *and assume the interaction radii* $0 < r < R$ *and* $\mu : \mathfrak{X} \to \mathbb{R}_{++}$ *to be known. Moreover, suppose that* $cR > r$ *for some* $c \in (\frac{1}{2}, 1)$ *and that* $\mathrm{diam}(\mathfrak{X}) > 2cR$. *Let* $\mathrm{Z} = \Xi \dot{\cup} \mathrm{H}$ *with density* $h_{\boldsymbol{\theta}} : \mathfrak{N} \to \mathbb{R}_+$ *w.r.t. the unit Poisson process* Π_1 *where* $\boldsymbol{\theta} = (\lambda, \beta, \gamma)$. *Then,* $\boldsymbol{\theta}$ *is identifiable.*

Proof. Let

$$
\begin{aligned}
&\zeta_i = \delta_{x_i}, && \text{for} \quad i = 1, 2, 3,\\
&\zeta_4 = \delta_{x_1} + \delta_{x_2}, && \text{with } r < \|x_1 - x_1\| < R \text{ and}\\
&\zeta_5 = \delta_{x_1} + \delta_{x_2} + \delta_{x_3}, && \text{with } r < \|x_1 - x_2\|, \|x_1 - x_3\| < R \text{ and } \|x_2 - x_3\| > R
\end{aligned}
$$

for $x_1, x_2, x_3 \in \mathfrak{X}$. Existence of ζ_5 is ensured by $\mathrm{diam}(\mathfrak{X}) > R$. To simplify notation we write $\beta_i := \beta\mu(x_i)$ and $\mu_i := \mu(x_i)$.

It suffices to show that $h_{\boldsymbol{\theta}}(\zeta) = h_{\tilde{\boldsymbol{\theta}}}(\zeta)$ for $\zeta \in A := \{\emptyset, \zeta_i \mid i = 1, 2, \ldots, 5\}$ implies $\boldsymbol{\theta} = \tilde{\boldsymbol{\theta}}$. Then, the assertion follows analogously to Lemma 4.20 (c).

As in Lemma 4.20 (a) it suffices to consider the unnormalised densities $\breve{h}_{\boldsymbol{\theta}}, \breve{h}_{\tilde{\boldsymbol{\theta}}}$ since $h_{\boldsymbol{\theta}}(\emptyset) = h_{\tilde{\boldsymbol{\theta}}}(\emptyset)$. The density $\breve{h}_{\boldsymbol{\theta}}$ for the provided point patterns ζ_i writes as

$$
\begin{aligned}
\breve{h}_{\boldsymbol{\theta}}(\zeta_i) &= \lambda + \beta_i, \qquad i = 1, 2, 3, && (4.40)\\
\breve{h}_{\boldsymbol{\theta}}(\zeta_4) &= \lambda^2 + \lambda(\beta_1 + \beta_2) + \beta_1\beta_2\gamma\\
&= (\lambda + \beta_1)(\lambda + \beta_2) + \beta_1\beta_2(\gamma - 1), && (4.41)\\
\breve{h}_{\boldsymbol{\theta}}(\zeta_5) &= \lambda^3 + \lambda^2(\beta_1 + \beta_2 + \beta_3) + \lambda(\gamma\beta_1(\beta_2 + \beta_3) + \beta_2\beta_3) + \gamma^2\beta_1\beta_2\beta_3\\
&= (\lambda + \beta_1)(\lambda + \beta_2)(\lambda + \beta_3) + \beta_1\beta_2(\gamma - 1)\left(\lambda\left(1 + \frac{\beta_3}{\beta_2}\right) + \beta_3(\gamma + 1)\right). && (4.42)
\end{aligned}
$$

Hence, if $h_{\boldsymbol{\theta}}(\zeta) = h_{\tilde{\boldsymbol{\theta}}}(\zeta)$ for all $\zeta \in A$, the equations (4.40), (4.41) and (4.42) lead to

$$
\begin{aligned}
\lambda + \beta_i &= \tilde{\lambda} + \tilde{\beta}_i,\\
\beta^2(\gamma - 1) &= \tilde{\beta}^2(\tilde{\gamma} - 1), && (4.43)\\
\lambda\left(1 + \frac{\mu_3}{\mu_2}\right) + \beta_3(\gamma + 1) &= \tilde{\lambda}\left(1 + \frac{\mu_3}{\mu_2}\right) + \tilde{\beta}_3(\tilde{\gamma} + 1).
\end{aligned}
$$

Note, that the μ_1, μ_2 cancel on both sides of Equation (4.43). If we subtract the first equation for $i = 3$ from the last equation we get

$$
\frac{\mu_3}{\mu_2}\lambda + \beta\mu_3\gamma = \frac{\mu_3}{\mu_2}\tilde{\lambda} + \tilde{\beta}\mu_3\tilde{\gamma}.
$$

Dividing by $\mu_3 > 0$ and plugging in $\lambda = \tilde{\lambda} + \tilde{\beta}_2 - \beta_2$ leads to $\beta\,(\gamma - 1) = \tilde{\beta}\,(\tilde{\gamma} - 1)$ or equivalently

$$\beta\tilde{\beta}(\gamma - 1) = \tilde{\beta}^2(\tilde{\gamma} - 1) = \beta^2(\gamma - 1), \tag{4.44}$$

where the latter equality originates from (4.43). Since $\beta > 0$ and $\gamma < 1$, we can divide by these quantities and obtain get $\beta = \tilde{\beta}$ and thus immediately also $\gamma = \tilde{\gamma}$ and $\lambda = \tilde{\lambda}$, which yields the claim. ■

Chapter 5

Bayesian inference and Markov Chain Monte Carlo algorithms

> *We balance probabilities and choose the most likely. It is the scientific use of the imagination, but we have always some material basis on which to start our speculation.*
>
> Sherlock Holmes, The Hound of the Baskervilles
> Arthur Conan Doyle, 1901

In Chapter 4 we outlined that for the superposition model proposed, a unique best choice of the underlying model parameters is unlikely. A similar behaviour is also expected for the separation of the minutiae into necessary and random minutiae. Thus, we seek for methods allowing separation in a probabilistic sense and incorporating uncertainties in the parameter choices. The tools to do so are provided by the theory of Bayesian statistics.

This chapter is intended to acquaint the reader with the world of Bayesian statistics. We provide the framework in which we like to practice statistical inference and show, how this can be applied for separating the minutiae into necessary and random ones. In the following section we present the Bayesian philosophy, point out differences to classical, frequentist statistics and provide common tools and algorithms. Section 5.2 considers applications of Bayesian methods to spatial point processes and the separation problem from Chapter 4.

5.1 Introduction to Bayesian statistics

The presented results below are obtained from Young & Smith (2005); Schervish (2012). A broader overview over the field is provided e.g. in Brooks et al. (2011) or Liu (2008). We start from a slightly more general point of view.

5.1.1 The Bayesian philosophy

Assume that we observe data $x \in \mathcal{X}$ in the *sample space* $\mathcal{X}$[14] sampled from a probability distribution $\mathbb{P}_{\boldsymbol{\theta}}$ which depends on the parameter $\boldsymbol{\theta} \in \Theta$, but the true value of $\boldsymbol{\theta}$ is unknown. The distribution $\mathbb{P}_{\boldsymbol{\theta}}$ is assumed to stem from a family of distributions $\{\mathbb{P}_{\boldsymbol{\theta}} \mid \boldsymbol{\theta} \in \Theta\}$ on $\mathcal{X}$. The *parameter space* Θ will in our context be a subset of $\mathbb{R}^d$ for some $d \in \mathbb{N}$. We like to make inference on $\boldsymbol{\theta}$ given the observed data x.

In classical statistics, sometimes also called *frequentist statistics*, the data X (before observing) is a random variable with distribution $\mathbb{P}_{\boldsymbol{\theta}}$ where $\boldsymbol{\theta}$ is unknown, but fixed. In contrast to this, in Bayesian statistics *both* the data and the parameter are regarded as random. Usually we have some information about the parameter even *before* knowing about the data, so-called *prior information*. The key idea is to update our prior beliefs in $\boldsymbol{\theta}$ after observing data x to take the new information into account to obtain the *posterior distribution*, i.e. the distribution of $\boldsymbol{\theta}$ *after* observing the data. Bayesian methods rely on Bayes' law of conditional probabilities (hence the name *Bayesian statistics*) which tells us exactly how to obtain the posterior distribution.

For a mathematical precise definition, suppose X and $\boldsymbol{\theta}$ are random variables with values in the measure spaces $(\mathcal{X}, \mathcal{B})$ and (Θ, τ), respectively, and let $\mathbb{P}_{\boldsymbol{\theta}}$ be the conditional distribution of X given $\boldsymbol{\theta}$. We assume that the parameter $\boldsymbol{\theta}$ has a probability density function π w.r.t. the Lebesgue measure μ on (Θ, τ). The density π is called *prior density* and represents our beliefs about the parameter before observing the data.

Moreover, assume that each $\mathbb{P}_{\boldsymbol{\theta}}$ has a density $h_{\boldsymbol{\theta}} = \frac{\mathrm{d}\mathbb{P}_{\boldsymbol{\theta}}}{\mathrm{d}\nu}$ w.r.t. to a measure ν on $(\mathcal{X}, \mathcal{B})$. Here, ν can e.g. be the counting measure on $\mathbb{N}$ (in a discrete setting), the Lebesgue measure (for the continuous case) or the measure induced by the unit Poisson process (for point process applications). We assume that $h_{\boldsymbol{\theta}}$ is measurable w.r.t. $\mathcal{B} \otimes \tau$ allowing us to integrate this function w.r.t. measures on both $\mathcal{X}$ and Θ.

[14] later on, this will be the space $\mathfrak{N}$ of point patterns on the region of interest $\mathfrak{X} \subseteq \mathbb{R}^2$

The function $\boldsymbol{\theta} \mapsto h_{\boldsymbol{\theta}}(x)$ as a function of $\boldsymbol{\theta}$ after $X = x$ is observed is called *likelihood* since it models how *likely* it is to observe the data x if the model parameter is $\boldsymbol{\theta}$.

For given $\boldsymbol{\theta}$, the function $x \mapsto h_{\boldsymbol{\theta}}(x)$ is the conditional density (w.r.t. ν) of X given $\boldsymbol{\theta}$. Assuming that the marginal distribution of X is absolutely continuous w.r.t. ν, we can write the density h_X of X w.r.t. ν as

$$h_X(x) = \int_{\Theta} h_{\boldsymbol{\vartheta}}(x)\pi(\boldsymbol{\vartheta})\,\mu(\mathrm{d}\boldsymbol{\vartheta}), \tag{5.1}$$

see e.g. (Schervish, 2012, Equation (1.23)). This density is called *marginal density* of X, *evidence* or *prior predictive density*.

The conditional distribution of $\boldsymbol{\theta}$ given $X = x$ is called *posterior distribution* of $\boldsymbol{\theta}$. Using Bayes' theorem, (Schervish, 2012, Theorem 1.31), we can compute its density $\pi(\cdot \mid x) : \Theta \to \mathbb{R}_+$ w.r.t. μ (the *posterior density*) as

$$\pi(\boldsymbol{\theta} \mid x) = \frac{\pi(\boldsymbol{\theta})h_{\boldsymbol{\theta}}(x)}{h_X(x)} = \frac{\pi(\boldsymbol{\theta})h_{\boldsymbol{\theta}}(x)}{\int_{\Theta} \pi(\boldsymbol{\vartheta})h_{\boldsymbol{\vartheta}}(x)\,\mu(\mathrm{d}\boldsymbol{\vartheta})}. \tag{5.2}$$

It is common to just write

$$\pi(\boldsymbol{\theta} \mid x) \propto \pi(\boldsymbol{\theta})h_{\boldsymbol{\theta}}(x) \tag{5.3}$$

or in words posterior $\propto$ prior × likelihood, where "$\propto$" means proportional and the constant of proportionality may depend on x but not on $\boldsymbol{\theta}$. This proportionality or normalising constant, and thus also the posterior distribution itself, is in general hard to compute. Thus, in practice usually sampling schemes are applied to approximate the posterior distribution which are designed in a way that sampling is possible without (explicitly) knowing the normalising constant. As a result, the right hand side of (5.3) is often sufficient to sample from the posterior distribution.

Moreover, in practice the choice of an appropriate prior is an important issue to think about, since inference for the posterior could heavily depend on how much we already know in advance. For a detailed discussion about the choice of prior distributions, see e.g. (Young & Smith, 2005, Section 3.6). Sometimes, a particular choice of the prior makes the computation of the posterior very easy and allows to write it even in closed form.

Example 5.1. (Conjugacy, (Young & Smith, 2005, Example 3.1)) Let $X \sim \operatorname{Poi}(\boldsymbol{\theta})$ be a Poisson distributed random variable but the true intensity $\boldsymbol{\theta} \in \Theta = \mathbb{R}_+$ is not known. Assume the prior density is a Gamma density with *shape* $a > 0$ and *rate* $b > 0$, i.e.

$$\pi(\boldsymbol{\theta}) = \frac{b^a}{\Gamma(a)} \boldsymbol{\theta}^{a-1} \mathrm{e}^{-b\boldsymbol{\theta}}$$

where $\Gamma(t) := \int_0^\infty u^{t-1} \mathrm{e}^{-u} \, \mathrm{d}u$ denotes the Gamma function. Since given $\boldsymbol{\theta}$, the data X is Poisson distributed, the likelihood (i.e. its probability mass function given $\boldsymbol{\theta}$) is of form

$$h_{\boldsymbol{\theta}} : \mathbb{N} \to \mathbb{R}_+, \qquad x \mapsto h_{\boldsymbol{\theta}}(x) = \frac{\boldsymbol{\theta}^x}{x!} \mathrm{e}^{-\boldsymbol{\theta}}.$$

Hence, for the posterior density holds

$$\pi(\boldsymbol{\theta} \mid x) \propto \pi(\boldsymbol{\theta}) h_{\boldsymbol{\theta}}(x) = \frac{b^a}{\Gamma(a) x!} \boldsymbol{\theta}^{a+x-1} \mathrm{e}^{-(b+1)\boldsymbol{\theta}},$$

which is, up to normalising constants, again the density of a Gamma distribution but now with shape $a + x$ and rate $b + 1$ instead of a and b. See (Gelman et al., 2013, Section 2.6) for a more detailed calculation and the computation of the normalising constant.

The fact that prior and posterior distribution belong to the same class of probability distributions is also referred to as *conjugacy* and the prior is called a *conjugate prior* for the likelihood function.

Example 5.1 gives rise to a useful application for Poisson point processes.

Example 5.2 (Conjugacy for Poisson processes)**.** Let $\Xi \sim \operatorname{Pop}(\lambda)$ be a homogeneous Poisson process on some compact subset $\mathfrak{X} \subseteq \mathbb{R}^2$ with positive Lebesgue measure where the intensity $\lambda > 0$ is unknown. We assume $\lambda \sim \Gamma(a, b)$ (as prior knowledge) for some $a, b > 0$ and are interested in the posterior distribution of λ after observing a point pattern $\xi = \{x_1, x_2, \ldots, x_n\} \sim \Xi$. To this end, recall the *scaling property* of the Gamma distribution: for $X \sim \Gamma(a, b)$ and $c > 0$ we have $cX \sim \Gamma\left(a, \frac{b}{c}\right)$.

From Remark 4.5 follows that $n = \Xi(\mathfrak{X}) \sim \operatorname{Poi}(\lambda |\mathfrak{X}|)$. Moreover, the scaling property of the Gamma distribution yields $\lambda |\mathfrak{X}| \sim \Gamma\left(a, \frac{b}{|\mathfrak{X}|}\right)$. Now, due to the conjugacy of Poisson and Gamma distribution, cf. Example 5.1, we obtain $\lambda |\mathfrak{X}| \mid n \sim \Gamma\left(a + n, \frac{b}{|\mathfrak{X}|} + 1\right)$ and hence, applying again the scaling property:

$$\lambda \mid \xi = \lambda \mid n \sim \Gamma\left(a + |\xi|, b + |\mathfrak{X}|\right). \tag{5.4}$$

In particular, the posterior distribution of λ only depends on the number n of observed points in ξ but not on their locations.

The use of a conjugate prior is not mandatory and there would be no reason to adhere with it if we had more prior information about the parameter λ. However, it is very convenient since it leads to a closed form for the posterior and we do not have to worry about the normalising constant $h_X(x)$. In non-conjugate cases, where everything has to be computed numerically, the computation of the normalising constant is usually the most laborious. For instance, for Strauss processes a similar approach as for Poisson processes is not possible without further ado because relation between the model parameters β, γ and the intensity is not tractable.

Remark 5.3 (Frequentist vs. Bayesian statistics)**.** In the setting of separating minutiae patterns, one advantage of Bayesian methods over frequentist methods becomes particularly apparent. As pointed out in Example 4.30, the likelihood of the parameters might be multimodal. In particular, when incorporating also the latent variable $\mathbf{W}$, cf. Equation (4.19), it can happen that for given parameters and a pair of points it might be equally likely to assign either of the two points to the Strauss and the Poisson process but not both to the same process, leading to two modes and negative correlation between the labels. While in classical approaches uncertainty estimation for e.g. the MPLE of the model parameters $\boldsymbol{\theta}$ can be obtained for instance by employing bootstrap or Monte Carlo methods, cf. Coeurjolly & Rubak (2013), a similar approach for the latent variables is, to the best of the author's knowledge, not known.

In contrast, Bayesian methods yield an uncertainty estimation of the posterior distribution of both the parameters and the labels given the data which captures the uncertainty and correlation effects mentioned above. This, however, comes at the price of dealing with the normalising constant of the posterior, cf. (5.1). Unfortunately, for the model considered in Section 4.4, we do not know of a corresponding conjugate prior (as in Example 5.2) for the model parameters (β, γ) and $\mathbf{W}$. Moreover, we cannot explicitly compute the integral (5.1), not even for known $\mathbf{W}$ due to the normalising constant of the Strauss process in (4.19).

To tackle this problem we proceed as follows: The rest of this section is dedicated to introducing the reader to general approaches for parameter estimation in a Bayesian sense if the marginal density (5.1) of the data is not explicitly available to us. With regard to estimating the parameters $\boldsymbol{\theta} = (\lambda, \beta, \gamma) \in \Theta$, cf. Section 4.4, in a Bayesian sense for an already separated point pattern (i.e. for known $\mathbf{W}$) we point out associated issues (particularly due to intractable normalising constants of Gibbs processes) in Section 5.2. We include the estimation of the posterior distribution of $\mathbf{W}$ in Section 6.4.

5.1.2 Metropolis-Hastings algorithms

In practice, one usually tries to avoid the computation of (5.1) as this integral can be very hard to compute especially if Θ is high-dimensional. Nevertheless, the prior predictive distribution or evidence (5.1) is sometimes used e.g. for the prediction of unseen data conditioned on the observed data or for model evaluation and validation, cf. (Gelman et al., 2013, Chapter 6 and 7). To this end, *Monte Carlo methods* are often employed to approximate (5.1) numerically. The idea is to sample from the parameter space Θ on random points, evaluate the integrand at these sample points and approximate the integral via an (possibly weighted) arithmetic mean of these values. Usually pseudo-random number generators are used to generate independent, identically distributed numbers on $[0, 1]$ and then transformation methods are applied to get any arbitrary desired distribution. However, this requires that we can at least evaluate the integrand, particularly the likelihood $h_{\boldsymbol{\theta}}$ making these analyses hard for models including Gibbs processes due to the intractable normalising constant. More recent algorithms rely on machine learning based techniques, see Kingma & Welling (2019) for introductory reading.

In the following we will focus on methods for parameter inference in which the computation of (5.1) plays only a minor role as algorithms are usually designed in a way to avoid the computation of normalisation constants. One of the most popular methods is the so-called *Metropolis-Hastings algorithm*, originally developed by Metropolis et al. (1953) and then extended by Hastings (1970), which we will introduce now.

The idea is to iteratively explore the parameter space Θ in a way such that the sampling distribution of the generated sequence $(\boldsymbol{\theta}^{(t)})_{t=1,2,\dots}$ converges to the posterior distribution, i.e. we obtain an approximation of the posterior distribution by drawing (independent) subsamples $(\boldsymbol{\theta}^{(t_i)})_{i=1}^{T} \subseteq (\boldsymbol{\theta}^{(t)})_{t=1,2,\dots}$ via

$$\pi(\boldsymbol{\theta} \mid x) \approx \frac{1}{T} \sum_{i=1}^{T} \delta_{\boldsymbol{\theta}^{(t_i)}}.$$

To this end, assume $\Theta \subseteq \mathbb{R}^d$ and let $\breve{\pi}(\boldsymbol{\theta} \mid x) = \pi(\boldsymbol{\theta}) h_{\boldsymbol{\theta}}(x)$ be the possibly unnormalised posterior density of the parameter of interest $\boldsymbol{\theta}$ given x. We introduce a trial density $q : \Theta \times \Theta \to \mathbb{R}_+$, $(\boldsymbol{\theta}, \boldsymbol{\theta}') \mapsto q(\boldsymbol{\theta}' \mid \boldsymbol{\theta})$, the *proposal*, which is some probability density satisfying $\int_\Theta q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta}' = 1$ for all $\boldsymbol{\theta} \in \Theta$. It models the likeliness to get from the state $\boldsymbol{\theta}$ to any other state $\boldsymbol{\theta}' \in \Theta$ or into a set of states $A \subseteq \Theta$ via $\int_A q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta}'$. We denote by $\boldsymbol{\theta}' \sim q(\cdot \mid \boldsymbol{\theta})$ that we draw $\boldsymbol{\theta}'$ as a sample from a random variable with density $q(\cdot \mid \boldsymbol{\theta})$. Then, a Metropolis-Hastings algorithm works as described in Algorithm 5.4.

Algorithm 5.4 (Metropolis-Hastings algorithm)**.**

```
Choose an arbitrary point $\boldsymbol{\theta}^{(0)}$ as starting point.
for $t = 0, 1, 2, \dots$ do
    Proposal: Given $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}$, generate a candidate $\boldsymbol{\theta}'$ for the next sample from the
      probability density $q(\cdot \mid \boldsymbol{\theta})$, i.e. $\boldsymbol{\theta}' \sim q(\cdot \mid \boldsymbol{\theta})$.
    Compute the Hastings ratio
        $H(\boldsymbol{\theta}' \mid \boldsymbol{\theta}) = \frac{\breve{\pi}(\boldsymbol{\theta}' \mid x)\, q(\boldsymbol{\theta} \mid \boldsymbol{\theta}')}{\breve{\pi}(\boldsymbol{\theta} \mid x)\, q(\boldsymbol{\theta}' \mid \boldsymbol{\theta})} = \frac{\pi(\boldsymbol{\theta}' \mid x)\, q(\boldsymbol{\theta} \mid \boldsymbol{\theta}')}{\pi(\boldsymbol{\theta} \mid x)\, q(\boldsymbol{\theta}' \mid \boldsymbol{\theta})}.$     (5.5)
    Calculate the acceptance ratio $\alpha = \min\{H(\boldsymbol{\theta}' \mid \boldsymbol{\theta}),\, 1\}$.
    Accept/ Reject: Generate a uniformly distributed random number $u \sim \mathcal{U}([0,1])$.
      Set
        $\boldsymbol{\theta}^{(t+1)} = \begin{cases} \boldsymbol{\theta}', & \text{if } \alpha \geq u \text{ (acceptance)}, \\ \boldsymbol{\theta}^{(t)}, & \text{if } \alpha < u \text{ (rejection)}. \end{cases}$
end for
If deemed reasonable, discard the first $t_0$ iterations (burn-in).
Output: Sample $(\boldsymbol{\theta}^{(t)})_{t=t_0,t_0+1,t_0+2,\dots} \approx \pi(\cdot \mid x)$.
```

Note, that the Algorithm 5.4 does not depend on the normalising constant $h_X(x)$ anymore since they cancel in the numerator and the denominator of the Hastings ratio (5.5). Since Algorithm 5.4 uses random steps, it belongs to the class of *Monte Carlo algorithms*. Its output is a sequence of random samples which, indeed, build a Markov chain, hence the name *Markov chain Monte Carlo*, short *MCMC*. This Markov chain converges under rather weak assumptions (Harris recurrence and irreducibility) to the desired stationary distribution with density $\pi(\boldsymbol{\theta} \mid x)$, see (Robert & Casella, 2013, Section 7.3). However, several major issues arise in practice.

(a) Theoretical results often guarantee the geometric convergence to the posterior (e.g. in the total variation distance, see Equation (5.8)). However, they do usually not provide error bounds which are applicable in an applied setting. To determine the distance from the sample distribution to the posterior, typically calculations are required which are only feasible in relatively simple settings and are hence not very useful in practice.

Thus, in practice one usually deletes the first few iterations of Algorithm 5.4 to allow for *burn-in*, i.e. the period during which the sample density is assumed to get close to the true posterior density. The choice of the burn-in length of the algorithm is, however, subjective and often more a matter of computational time than of actual convergence. There exist various diagnostics for convergence. See (Robert & Casella, 2013, Section 12) for an overview.

(b) Since $(\boldsymbol{\theta}^{(t)})_{t=1,2,\ldots}$ is a Markov chain the samples $\boldsymbol{\theta}^{(t)}$ are not independent and, even worse, often exhibit a dependence structure for many iterations. This impairs the quality of the estimated posterior distribution from $(\boldsymbol{\theta}^{(t)})_{t=1,2,\ldots}$ and requires a larger sample size compared to an i.i.d. sequence. In general, it is desirable that the dependence between $\boldsymbol{\theta}^{(t)}$ and $\boldsymbol{\theta}^{(t+s)}$ dies out quickly as s increases in which case the chain is said to be *well mixing*.

(c) As for the burn-in, also the overall run time is often a matter of computational time and rather a subjective choice. For example, the algorithm could discard the first 10,000 iterations which are treated as burn-in. Then it runs for another 100,000 iterations to generate a random sample from the desired distribution. From this, one often proceeds with a thinned sequence where we only consider every s-th sample for sufficiently large s to get approximately independent samples where s is some quantity measuring the correlation between samples, e.g. the *integrated autocorrelation time*

$$\tau_{\text{int}} := 1 + 2\sum_{t=1}^{\infty} \operatorname{cor}\left(\boldsymbol{\theta}^{(0)}, \boldsymbol{\theta}^{(t)}\right).$$

There are different opinions about whether thinning is worthwhile since the number of discarded samples can be quite large. For our purposes we decided to apply thinning with τ_{int} for our inference to obtain approx. independent samples. The reduced sample size is compensated for by a longer run of the Markov chain. For further discussions we refer to (Young & Smith, 2005, Section 3.7) and (Gelman et al., 2013, Section 11.6).

One method we will use afterwards is based on the following idea: Suppose $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_d) \in \Theta \subseteq \mathbb{R}^d$ to be d-dimensional. Then, instead of updating all components at once, we chose $i \in \{1, \ldots, d\}$, fix all components except the i-th and propose a new

$$\boldsymbol{\theta}' = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{i-1}, \boldsymbol{\theta}'_i, \boldsymbol{\theta}_{i+1}, \ldots, \boldsymbol{\theta}_d).$$

If we go through all components one after another, this is called a *Gibbs sampler*[15], if i is chosen randomly also *random (scan) Gibbs sampler*. The advantage of this procedure is, that in each step we only have to sample from a one-dimensional distribution instead of a d-dimensional. This would still require that we can sample from the conditional marginals but in practice this often turns out to be much easier, since we can for example use conjugate priors.

Example 5.5 (Proposal variance)**.** One of the most crucial ingredients to successfully implement a well-performing MCMC algorithm is parameter tuning, especially tuning the variance of the proposal distribution. We like to demonstrate the effect in a simple example.

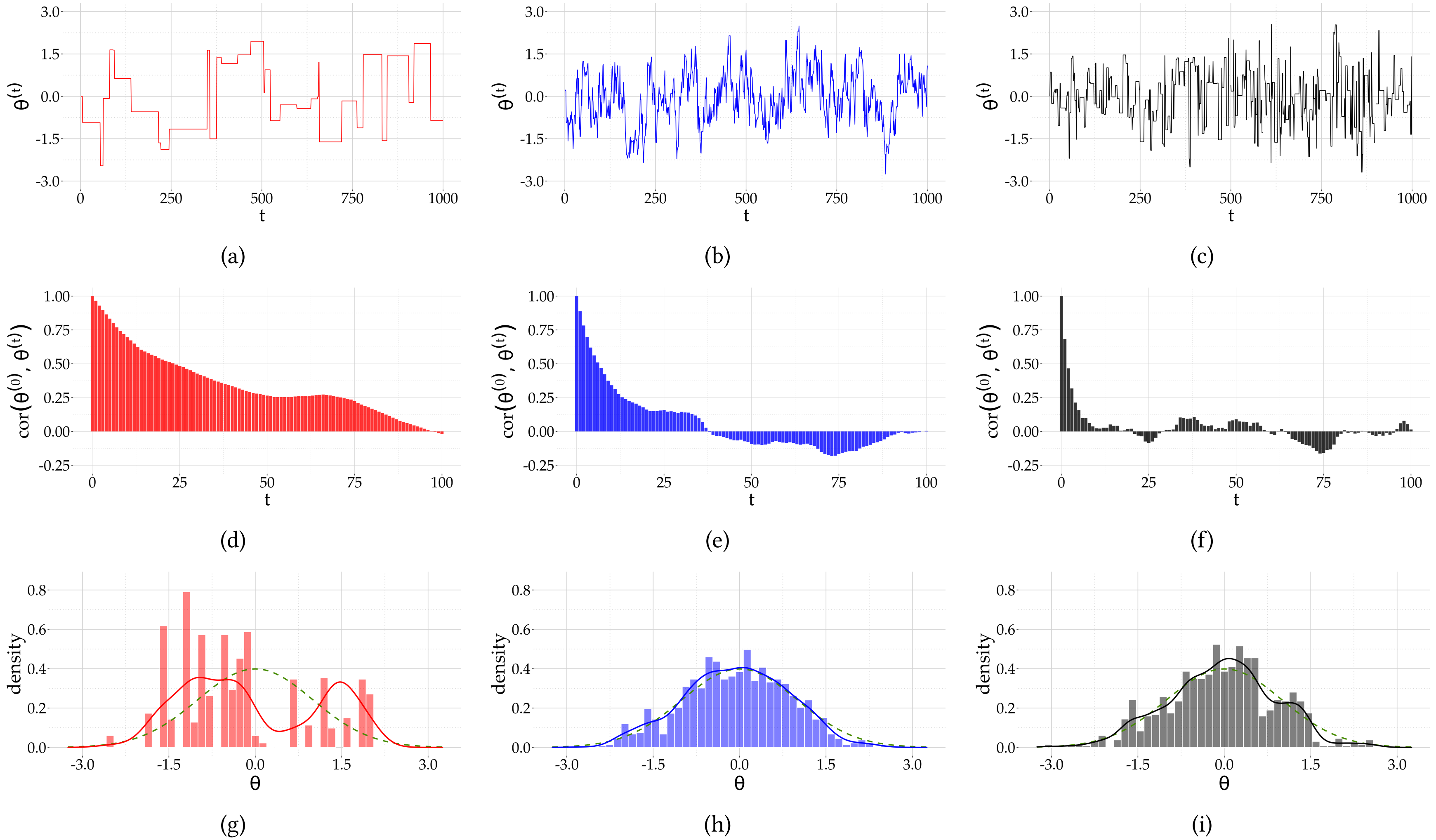
Assume we want to sample from a standard normal distribution, i.e. $\pi(\cdot \mid x)$ is a $\mathcal{N}(0,1)$ density (independent of x), using an MH-algorithm with an $\mathcal{N}(\boldsymbol{\theta},\sigma^2)$ proposal $q(\cdot \mid \boldsymbol{\theta})$ for some $\sigma > 0$. We execute the algorithm starting from $\boldsymbol{\theta} = 0$ for three different choices of σ, namely a large one $\sigma^{\text{large}} = 32$, a small one $\sigma^{\text{small}} = 0.5$ and an intermediate one $\sigma^{\text{intermediate}} = 4$. The results of a run of 1,000 iterations can be seen in Figure 5.1.

In the top row we plotted the *trace* of the Markov chain during the run time, i.e. the graph of $t \mapsto \boldsymbol{\theta}^{(t)}$, indicating the behaviour of the chain over the time. It is clearly visible, that for a large proposal variance we quite often discard the proposals and only rarely change the state: the *acceptance rate*, i.e. the percentage of accepted proposals in relation to the total proposals, is 0.3%. However, if this is the case, these jumps can be very large. In contrast to this, for the small proposal variance we almost always accept the proposals (acceptance rate 85%), however, the changes in the state are only minor. For the intermediate proposal variance we observe frequent state changes (acceptance rate 28%) of moderate size.

Let's now have a look how the proposal variance affects the autocorrelation of the Markov chain, shown in the second row of Figure 5.1. For the Markov chain with large proposal variance we observe only a slow decay in the autocorrelation, caused by the long dwell time in the states. The autocorrelation for the small proposal variance decays a little faster, however due to the minor state changes, consecutive $\boldsymbol{\theta}^{(t)}$ are highly correlated even for quite a few iterations. In fact, the (rounded) integrated autocorrelation times are

$$\tau_{\text{int}}^{\text{large}} = 54, \quad \tau_{\text{int}}^{\text{small}} = 18, \quad \tau_{\text{int}}^{\text{intermediate}} = 6.$$

[15] Substantially, this has nothing to do with the Gibbs processes from Chapter 4; merely the eponym is in both cases Josiah Willard Gibbs (1839–1903).



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 5.1: *Analysis of the influence for three different proposal variances from left to right: large $\sigma = 32$ (red), small $\sigma = 0.5$ (blue), intermediate $\sigma = 4$ (black). Top row: trace plots for 1,000 iterations. Middle row: Autocorrelation function. Bottom row: Histogram and estimated posterior density (solid) and true standard normal posterior (dashed green).*

Only the chain for $\sigma^{\text{intermediate}}$ shows a rapid decay of the autocorrelation, suggesting that already after short time, samples are approximately independent.

In the bottom row of Figure 5.1 the histograms and estimated densities for the sample obtained from the whole run can be seen. Ideally, this should resemble a standard normal density, indicated as dashed green line. Clearly, for σ^{large} the deviation is quite large – the estimated density is even bimodal. The estimated density of σ^{small} is quite good, maybe even better than the one for the intermediate proposal variance. However, it could happen e.g. for a bimodal posterior distribution, that due to the small step size we only explore one part of the parameter space and miss one of the two modes. Since the structure of the parameter space is not known in advance, a too small choice of the proposal variance can lead to convergence to the wrong posterior distribution. A very common diagnostic to find out about the structure of the parameter space is the *multistart heuristic*. The idea is to start the Markov chain at different points in order to examine the convergence behaviour of the chain. If the posterior distribution is multimodal or the state space of the chain has poorly connected regions we would find see this in different convergence behaviours depending on the chosen initial point.

In the literature the problem of choosing the right proposal variance is widely discussed and so far there is no universal answer on how to choose it *just right*. (Roberts, 1996, Section 3.5.2) proposed, as a rule of thumb, to try to achieve an acceptance rate between 15% and 50% which seems to be good enough for most applications. Hence, it is usually recommended to explore the state space using some pilot experiments, beginning with a large proposal variance and adjust the proposal variance accordingly.

Remark 5.6 (Further tuning)**.** In practice, the tuning of the proposal variance is not the only issue to consider to set up a well-performing MCMC algorithm. Jarner & Roberts (2007); Neal & Roberts (2011) mention that for fast convergence to the posterior compatibility in the tail behaviour of proposal and posterior is required (e.g. we might not converge geometrically when the proposal distribution is Gaussian while the posterior distribution of the parameter has heavy tails). Algorithms that take local information of the posterior into account in the proposal often require pointwise evaluations of derivatives of the posterior density, see e.g. Roberts & Rosenthal (1998). Parameter tuning for our application is discussed in Section 6.4. For further reading about tuning possibilities of MCMC algorithms and optimal proposals see e.g. Roberts & Rosenthal (2009) or (Brooks et al., 2011, Chapter 4).

Remark 5.7 (Posterior point estimators)**.** In practice, it is sometimes useful (or even necessary) to *summarise* the posterior distribution to a single value (e.g. for point estimation) as the whole posterior distribution is too much information for us to think about easily. Common methods to do so are, as in classical descriptive statistics, to describe the posterior distribution by e.g. the *posterior mean* (i.e. the expectation of the parameter w.r.t. the posterior distribution), the *posterior median* (i.e. the value cutting the probability mass of the posterior distribution in halves) or the *posterior mode* (sometimes called *maximum a posteriori estimator* (MAP), i.e. the maximum of the posterior distribution). The answer which one to use depends very much on the application, for a discussion we refer to literature about Bayesian decision theory, e.g. (Schervish, 2012, Chapter 3). In applications if the posterior distribution looks even vaguely like a normal distribution it is common to summarise it like

$$\boldsymbol{\theta} = \text{posterior mean} \pm \text{posterior standard deviation}.$$

If we are interested in the precision of the estimate, e.g. of the posterior mean, from a sample from the posterior distribution we state the estimate ± *standard error* (s.e.), i.e. the empirical posterior standard deviation divided by the square root of the sample size.

5.2 MCMC for point processes

Let us now consider how MCMC can be applied for exploring the posterior distribution of the model parameters from our superposition model from Section 4.4. To obtain information about both the model parameters of the Poisson and Strauss process and the labels $\mathbf{W} \in \{0,1\}^n$ as in Section 4.4, we have to include $\mathbf{W}$ as an additional variable to the parameter space. It has turned out to be difficult to sample from the joint distribution of parameters and labels at the same time. Hence, we decided to employ a Gibbs sampler and update the parameters and labels alternately. Implementation details are considered in Section 6.4. In this section we like to highlight some issues coming along the parameter updates for point processes. Hence, for the rest of this section, assume tacitly the allocation $\mathbf{W}$ of the points to the processes to be known. All probabilities are conditional on the number n of data points.

We consider a point pattern $\zeta = \xi \dot\cup \eta \sim \mathrm{Z} = \Xi \dot\cup \mathrm{H}$ from the superimposed process Z as in Section 4.4. The density of $\zeta = \xi \dot\cup \eta$ for known $\mathbf{W}$ w.r.t. Binomial process[16] is $h_{\boldsymbol{\theta}}(\zeta) \propto$

[16] i.e. a Poisson process conditioned on the total number of points being n

$f_\lambda(\{z_i : \mathbf{W}_i = 0\})\, g_{\beta,\gamma}(\{z_i : \mathbf{W}_i = 1\})$ for some $\boldsymbol{\theta} = (\lambda, \beta, \gamma) \in \Theta = \mathbb{R}_+ \times \mathbb{R}_+ \times (0,1)$. The posterior density

$$\pi(\boldsymbol{\theta} \mid \zeta) = \frac{\pi(\boldsymbol{\theta}) h_{\boldsymbol{\theta}}(\zeta)}{\int_\Theta \pi(\boldsymbol{\vartheta}) h_{\boldsymbol{\vartheta}}(\zeta)\,\mathrm{d}\boldsymbol{\vartheta}}$$

contains in fact two normalising constants, namely the one of the likelihood $h_{\boldsymbol{\theta}}$ in the numerator (from the Strauss density $g_{\beta,\gamma}$) and the one in the denominator, which are both in fact intractable. This problem is in the literature also referred to as *double intractability*. Whereas we can get rid of the normalising constant in the denominator by applying a Metropolis-Hastings algorithm, cf. Section 5.1.2, we cannot apply a similar argument for the normalising constant of the likelihood $h_{\boldsymbol{\theta}}$. To solve this problem, we apply an approach which goes back to Besag & Green (1993). For applications in point processes we refer to Berthelsen & Møller (2006); Redenbach et al. (2015); Rajala et al. (2016).

5.2.1 Auxiliary variable method

The idea is to extend the space and introduce an auxiliary point pattern $\tilde{\chi}$[17] living in the same space as the data ζ. We assume $\tilde{\chi}$ to be drawn w.r.t. a density $\varphi(\cdot \mid \zeta)$ w.r.t. the standard Poisson process and which does not depend on the current $\boldsymbol{\theta}$. Then, this point pattern is included into the model as an additional variable, i.e. the parameter space is now $\Theta \times \mathfrak{N}$. For this augmented model we have to define a new proposal density $\tilde{q}$ on the extended parameter space $\Theta \times \mathfrak{N}$ which we chose as

$$\tilde{q}(\boldsymbol{\theta}', \tilde{\chi}' \mid \boldsymbol{\theta}, \tilde{\chi}) = h_{\boldsymbol{\theta}'}(\tilde{\chi}') q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}), \tag{5.6}$$

i.e. we draw the new auxiliary point pattern $\tilde{\chi}'$ independently of the current auxiliary point pattern $\tilde{\chi}$ as a realisation of the Gibbs process of our model, but with parameter $\boldsymbol{\theta}'$. The proposal for the parameter $\boldsymbol{\theta}'$ remains as before. Now, the Hastings ratio for a parameter update from $(\boldsymbol{\theta}, \tilde{\chi})$ to $(\boldsymbol{\theta}', \tilde{\chi}')$ writes as

$$H(\boldsymbol{\theta}', \tilde{\chi}' \mid \boldsymbol{\theta}, \tilde{\chi}) = \frac{\pi(\boldsymbol{\theta}', \tilde{\chi}' \mid \zeta)}{\pi(\boldsymbol{\theta}, \tilde{\chi} \mid \zeta)} \frac{\tilde{q}(\boldsymbol{\theta}, \tilde{\chi} \mid \boldsymbol{\theta}', \tilde{\chi}')}{\tilde{q}(\boldsymbol{\theta}', \tilde{\chi}' \mid \boldsymbol{\theta}, \tilde{\chi})} = \frac{\varphi(\tilde{\chi}' \mid \zeta)}{\varphi(\tilde{\chi} \mid \zeta)} \frac{h_{\boldsymbol{\theta}'}(\zeta)}{h_{\boldsymbol{\theta}}(\zeta)} \frac{\pi(\boldsymbol{\theta}')}{\pi(\boldsymbol{\theta})} \frac{h_{\boldsymbol{\theta}}(\tilde{\chi})}{h_{\boldsymbol{\theta}'}(\tilde{\chi}')} \frac{q(\boldsymbol{\theta} \mid \boldsymbol{\theta}')}{q(\boldsymbol{\theta}' \mid \boldsymbol{\theta})}. \tag{5.7}$$

Since the normalising constants of the two $h_{\boldsymbol{\theta}}$-terms and the two $h_{\boldsymbol{\theta}'}$-terms cancel, this Hastings ratio can be computed explicitly. However, in every update we have to draw a

[17] the tilde indicates that this is not a parameter of the original model

new point pattern from a Gibbs process which can in practice be quite costly. The sampling scheme is summarised in Algorithm 5.8.

Algorithm 5.8 (Auxiliary variable method)**.**

```
Draw initial $\boldsymbol{\theta}^{(0)} \sim \pi(\cdot)$ from the prior and $\tilde{\chi}^{(0)} \sim h_{\boldsymbol{\theta}^{(0)}}(\cdot)$.
for $t = 0, 1, 2, \dots$ do
    Given $(\boldsymbol{\theta}, \tilde{\chi}) = (\boldsymbol{\theta}^{(t)}, \tilde{\chi}^{(t)})$ propose $\boldsymbol{\theta}' \sim q(\cdot \mid \boldsymbol{\theta})$.
    Generate an auxiliary point pattern $\tilde{\chi}' \sim h_{\boldsymbol{\theta}'}(\cdot)$ and compute
        $$H(\boldsymbol{\theta}', \tilde{\chi}' \mid \boldsymbol{\theta}, \tilde{\chi}) = \frac{\varphi(\tilde{\chi}' \mid \zeta)}{\varphi(\tilde{\chi} \mid \zeta)} \frac{h_{\boldsymbol{\theta}'}(\zeta)}{h_{\boldsymbol{\theta}}(\zeta)} \frac{\pi(\boldsymbol{\theta}')}{\pi(\boldsymbol{\theta})} \frac{h_{\boldsymbol{\theta}}(\tilde{\chi})}{h_{\boldsymbol{\theta}'}(\tilde{\chi}')} \frac{q(\boldsymbol{\theta} \mid \boldsymbol{\theta}')}{q(\boldsymbol{\theta}' \mid \boldsymbol{\theta})}.$$
    Accept $\boldsymbol{\theta}'$ with probability $\alpha = \min\{H(\boldsymbol{\theta}', \chi' \mid \boldsymbol{\theta}, \chi), 1\}$. In case of acceptance,
      set $(\boldsymbol{\theta}^{(t+1)}, \tilde{\chi}^{(t+1)}) = (\boldsymbol{\theta}', \tilde{\chi}')$; otherwise $(\boldsymbol{\theta}^{(t+1)}, \tilde{\chi}^{(t+1)}) = (\boldsymbol{\theta}, \tilde{\chi})$.
end for
```

A crucial influence on the algorithm's performance is the choice of φ which ideally should fit well to the proposal density $h_{\boldsymbol{\theta}}$. The best choice would of course be to choose $\varphi(\tilde{\chi} \mid \zeta) = h_{\boldsymbol{\theta}}(\tilde{\chi})$, which is not feasible since then (5.7) would contain the normalising constants again.

Another possibility is in choosing $\varphi(\tilde{\chi} \mid \zeta) = h_{\boldsymbol{\theta}_{\text{aux}}}(\tilde{\chi})$ with a fixed value $\boldsymbol{\theta}_{\text{aux}}$. This choice is expected to work well if the posterior distribution is concentrated around $\boldsymbol{\theta}_{\text{aux}}$. To follow this approach we have to be able to simulate auxiliary point patterns from the distribution induced by $h_{\boldsymbol{\theta}_{\text{aux}}}$. For homogeneous Strauss processes, we can draw auxiliary point patterns e.g. using the coupling from the past algorithms, cf. Algorithm 4.17. For the case of an inhomogeneous Strauss process which will be of interest to us later, unfortunately there does not yet exist appropriate software. Certainly, we can run another Markov chain with the appropriate invariant distribution. This approach, however, is not exact anymore, so convergence to the correct invariant distribution is not ensured. We will elaborate on this in more detail in Section 5.2.2. For further details regarding the auxiliary variable method for Bayesian analysis on Strauss processes, see Berthelsen & Møller (2006). The choice of φ for our application to fingerprints will be discussed in Section 6.4.

Remark 5.9 (Exchange algorithm)**.** Murray et al. (2012) proposed an approach which works similarly as the auxiliary variable method mentioned above, however, does not depend on an estimation of the parameter $\boldsymbol{\theta}_{\text{aux}}$. In the proposed method, referred to as *single variable exchange algorithm*, they do not associate an auxiliary point patterns $\tilde{\chi}, \tilde{\chi}'$ with the current and the proposed parameters $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ respectively, but generate only one $\tilde{\chi} \sim h_{\boldsymbol{\theta}'}$, see Algorithm 5.10 below. Notably, the Markov chain produced by Algorithm 5.8 is running on $\Theta \times \mathfrak{N}$ whereas Algorithm 5.10 projects down to Θ in every step.

Algorithm 5.10 (Exchange algorithm)**.**

```
1 Draw initial $\boldsymbol{\theta}^{(0)} \sim \pi(\cdot)$ from the prior.
2 for $t = 0, 1, 2, \dots$ do
3     Given $\boldsymbol{\theta} = \boldsymbol{\theta}^{(t)}$ propose $\boldsymbol{\theta}' \sim q(\cdot \mid \boldsymbol{\theta})$.
4     Generate an auxiliary point pattern $\tilde{\chi} \sim h_{\boldsymbol{\theta}'}(\cdot)$ and compute
          $H(\boldsymbol{\theta}' \mid \boldsymbol{\theta}) = \frac{h_{\boldsymbol{\theta}'}(\zeta)}{h_{\boldsymbol{\theta}}(\zeta)} \frac{\pi(\boldsymbol{\theta}')}{\pi(\boldsymbol{\theta})} \frac{h_{\boldsymbol{\theta}}(\tilde{\chi})}{h_{\boldsymbol{\theta}'}(\tilde{\chi})} \frac{q(\boldsymbol{\theta} \mid \boldsymbol{\theta}')}{q(\boldsymbol{\theta}' \mid \boldsymbol{\theta})}.$
5     Accept $\boldsymbol{\theta}'$ with probability $\alpha = \min\{H(\boldsymbol{\theta}' \mid \boldsymbol{\theta}), 1\}$. In case of acceptance, set
        $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}'$; otherwise $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}$.
6 end for
```

Especially, when the estimate of $\boldsymbol{\theta}_{\text{aux}}$ is poor, the exchange algorithm often outperforms the auxiliary variable method in terms of acceptance rate. This can help to improve the performance of the overall MCMC algorithm whenever low acceptance probabilities are an issue. As for the point patterns considered we do not observe a low acceptance rate in general, see Chapter 7, we refrained from implementing the exchange algorithm. Note, that Algorithm 5.10 also requires exact sampling of auxiliary point patterns from $h_{\boldsymbol{\theta}'}$. Hence, the problem that we do have to run a Markov chain to obtain these point patterns is not resolved by this approach either.

5.2.2 Double Metropolis-Hastings algorithm

As already mentioned, the algorithms from Section 5.2.1 require that sampling from a process with density $h_{\boldsymbol{\theta}}$ w.r.t. Π_1 is possible. As perfect sampling is in practice not always possible or very cumbersome, a common practice is to substitute the sampling step in Line 4 of Algorithm 5.8 or Algorithm 5.10, respectively, by running another Markov chain with the appropriate stationary distribution. Algorithms resulting from such a substitution are also referred to as *double Metropolis-Hastings algorithms*. Indeed, in every iteration of the (outer) Metropolis-Hastings algorithm we again run another Metropolis-Hastings algorithm.

Immediately the question arises whether we can still expect convergence of the overall Markov chain to the correct stationary distribution by letting the inner Markov chain (the one for generating a Gibbs point pattern) run just long enough. To this end, denote by

$\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}', \tilde{\chi})$ the acceptance probability of an update $\boldsymbol{\theta}'$ of the Metropolis-Hastings algorithm given the current value $\boldsymbol{\theta}$ using the auxiliary variable $\tilde{\chi}$ (either directly drawn or generated using Metropolis-Hastings). Moreover, assume that the via Metropolis-Hastings generated auxiliary variable has a density, m_t say, w.r.t. the standard Poisson process Π_1 where t is the number of iterations.

Let E and D denote the Markov kernels[18] for one step of the auxiliary variable method (or the exchange algorithm) and the double Metropolis-Hastings algorithm, respectively. Then, for $\boldsymbol{\theta} \in \Theta$ and $A \subseteq \Theta$ we have

$$\begin{aligned} E(\boldsymbol{\theta}, A) &= \int_{\mathfrak{N}} \int_A \alpha(\boldsymbol{\theta}, \boldsymbol{\theta}', \tilde{\chi})\, q(\boldsymbol{\theta}' \mid \boldsymbol{\theta})\, h_{\boldsymbol{\theta}}(\tilde{\chi}) \mathrm{d}\boldsymbol{\theta}' \Pi_1(\mathrm{d}\tilde{\chi}), \\ D(\boldsymbol{\theta}, A) &= \int_{\mathfrak{N}} \int_A \alpha(\boldsymbol{\theta}, \boldsymbol{\theta}', \tilde{\chi})\, q(\boldsymbol{\theta}' \mid \boldsymbol{\theta})\, m_t(\tilde{\chi}) \mathrm{d}\boldsymbol{\theta}' \Pi_1(\mathrm{d}\tilde{\chi}), \end{aligned}$$

where $q(\cdot \mid \boldsymbol{\theta})$ is the proposal density given $\boldsymbol{\theta} \in \Theta$. Given $\boldsymbol{\theta} \in \Theta$, we compare the measures induced by these Markov kernels using the total variation distance. Recall, that for two measures μ, ν on a measurable space $(\Omega, \mathcal{A})$, their *total variation distance* is defined as

$$\|\mu - \nu\|_{tv} := \sup_{A \in \mathcal{A}} |\mu(A) - \nu(A)| \,. \tag{5.8}$$

Then, we have

$$\begin{aligned} \|E(\boldsymbol{\theta}, \cdot) - D(\boldsymbol{\theta}, \cdot)\|_{tv} &\leq \sup_{A \in \sigma(\Theta)} \int_{\mathfrak{N}} \int_A \alpha(\boldsymbol{\theta}, \boldsymbol{\theta}', \tilde{\chi})\, q(\boldsymbol{\theta}' \mid \boldsymbol{\theta})\, |h_{\boldsymbol{\theta}}(\tilde{\chi}) - m_t(\tilde{\chi})| \, \mathrm{d}\boldsymbol{\theta}' \Pi_1(\mathrm{d}\tilde{\chi}) \\ &\leq \int_{\mathfrak{N}} \int_{\Theta} \alpha(\boldsymbol{\theta}, \boldsymbol{\theta}', \tilde{\chi})\, q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta}' \, |h_{\boldsymbol{\theta}}(\tilde{\chi}) - m_t(\tilde{\chi})| \, \Pi_1(\mathrm{d}\tilde{\chi}) \\ &= \int_{\mathfrak{N}} |h_{\boldsymbol{\theta}}(\tilde{\chi}) - m_t(\tilde{\chi})| \, \Pi_1(\mathrm{d}\tilde{\chi}) \longrightarrow 0 \end{aligned}$$

as $t \to \infty$. Here, the first inequality follows from the triangle inequality. The equality in the third row is due to $\int_{\Theta} \alpha(\boldsymbol{\theta}, \boldsymbol{\theta}', \tilde{\chi})\, q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta}' = 1$ and the convergence follows under quite general conditions, cf. Tsvetkov et al. (2013).

Hence, if we run the inner Metropolis-Hastings algorithm just long enough, the error between the auxiliary variable method (or exchange algorithm) and the double Metropolis-Hastings algorithm is small. However, this raises the question whether the errors from single iterations cumulate in a way that the convergence of the overall Metropolis-Hastings

[18] for some preliminaries on theory of Markov chains, see Section A.2.

algorithm is not longer ensured. This is a central topic of the perturbation theory of Markov chains. Approximate Metropolis-Hastings algorithms and criteria under which convergence is ensured are considered in Rudolf & Schweizer (2018).

Checking these conditions in an applied setting is often challenging and whenever resources are limited there is usually a trade-off between actual convergence and speed of the overall algorithm; especially, since the inner Markov chain has to run in every Hastings-step of the outer algorithm. We discuss this issue for our application to point processes in Section 6.4.2.

Having found a framework that allows for Bayesian inference, we are now in a position to compute all ingredients and elaborate on the choice of hyper-parameters needed for this inference. This will be discussed in the following Chapter 6.

Chapter 6

Algorithms and implementation details

This chapter considers all algorithms and algorithmic details used for this thesis. We focus at first on processing the fingerprint image itself in the Sections 6.1–6.3 whereas implementation details for the MCMC-based separation algorithm are discussed in Section 6.4. The provided software packages and an analysis on computational complexity are presented in Section 6.5.

For the model introduced in Chapter 3 we determine a fingerprint's divergence field using the algorithms presented below. For our experiments we use the databases of the fingerprint verification competition 2002, cf. Maio et al. (2002). We process an image in the following order:

(i) Segmentation (separation of fingerprint foreground and background)

(ii) Binarisation (transformation into a black-and-white image)

(iii) Orientation estimation and singularity detection

(iv) Ridge frequency estimation

(v) Orientation field divergence and ridge frequency divergence estimation

(vi) Estimation of intensity of necessary minutiae

General remarks All methods presented below depend on hyper-parameters such as smoothing bandwidths or window sizes of filters. The suitable choice of these parameters depends on many factors such as quality and resolution of the captured fingerprint or the used acquisition medium. To obtain optimal results of our analysis, these parameters have to be tuned individually for every considered fingerprint which is, of course, not suitable for application to large databases. Therefore, and to exclude the influence of hyper-parameters on our analysis, we restrict ourselves to a fixed set of these hyper-parameters which we determined experimentally to perform reasonably well on the considered fingerprints from Figure 3.6.

Segmentation & Binarisation We determine the region of interest using the segmentation algorithm of Thai et al. (2016). This algorithm also provides a binarised version of its input fingerprint. These binary images turned out to behave more robust than the non-binarised versions, especially regarding ridge frequency estimation, see Section 6.2. We hence use these binarised images for the entire following analysis. We like to stress that for the analysis of fingerprints the actual object of interest is the *actual fingerprint of the person* (i.e. the friction ridge pattern of the epidermis), however, not the fingerprint image itself (which is just a digital representation). Hence, image preprocessing and image enhancement can be seen as procedures improving the performance of feature extraction algorithms rather than manipulating the data of interest which is, indeed, not affected by it. However, there is no ground truth for neither the orientation field, singular points nor ridge frequency to compare with, since the actual fingerprint is not available to us.

6.1 Orientation field and singularity detection

The issue of extracting an orientation field has already been researched for a long time, however, is not yet resolved to a satisfying level for all kinds of images. Issues arise especially whenever the quality of the images deteriorates. We will henceforth focus on fingerprint images with high quality and follow a simple natural approach based on the computation of image gradients from Bazen & Gerez (2002). To this end, we will first make a brief excursion to image feature extraction procedures.

6.1.1 Image feature extraction and filtering

For this thesis, a *(digital) image* is considered to be a matrix $\mathfrak{I}$ with entries in a pixel space. Typically, this pixel space is $\{0, 1, \dots, 255\}$ (8-bit grey-value images) or $\{0, 1\}$ (binary images). To simplify notations and to omit rounding effects in computations, we allow for real-valued entries and assume $\mathfrak{I} \in \mathbb{R}^{h \times w}$ where $h, w \in \mathbb{N}$ are the *height* and *width* of the image, respectively.

In image processing, an *image feature* is a piece of information about the content of an image, typically whether a certain region of the image has certain properties. Features may be specific structures in the image such as points, edges or objects. A filter is a device for extracting image features or for removing unwanted components or features from an image. In the context of fingerprint recognition, interesting features of a fingerprint image are for instance the region of interest, ridge lines and their endings/ bifurcations (minutiae), ridge orientation, distances between ridges as well as singular points.

Faulty information in digital images is an intrinsic problem of image processing usually caused by the acquisition medium. Filtering is important and often necessary whenever the acquired fingerprint image is noisy or of bad quality to remove spurious features which do not actually belong to the depicted fingerprint. Whereas noise removal is often no issue for the human eye, it poses a challenge for computers. It is often difficult to find a suitable trade-off between removing spurious features on the one hand and remaining actual features of the image on the other hand. Hence, smoothing filters for noise removal are often combined with the corresponding feature extraction filters to obtain satisfying results. We briefly discuss the most important concepts to us as follows. For thorough insights into image processing and image feature extraction methods we refer to Bredies & Lorenz (2011); Nixon & Aguado (2019) or Zhao et al. (2020) for an overview of modern feature learning based algorithms.[19]

Definition 6.1 (Convolution)**.** . Let $\mathfrak{I} \in \mathbb{R}^{h\times w}$ be an image and $\mathcal{K} \in \mathbb{R}^{(2n+1)\times(2m+1)}$ be a matrix of dimension $(2n+1) \times (2m+1)$ for $n, m \in \mathbb{N}_0$, which is called *kernel*, also *filter* or *convolution matrix*. The *filtered image* $\mathfrak{I}^* := \mathcal{K} * \mathfrak{I} \in \mathbb{R}^{h\times w}$ of $\mathfrak{I}$ is given via

$$\mathfrak{I}^*(i,j) := \sum_{k=-n}^{n} \sum_{\ell=-m}^{m} \mathcal{K}(k,\ell)\, \mathfrak{I}(i+k, j+\ell)$$

for $1 \leq i \leq h$, $1 \leq j \leq w$. We set $\mathfrak{I}(i+k, j+\ell) = 0$ whenever $(i+k, j+\ell) \notin \{1,2,\dots,h\} \times \{1,2,\dots,w\}$ to capture boundary effects, see Remark 6.2 for the practical implementation.

Remark 6.2 (Boundary effects)**.** Note, that whenever the size of the filter extends beyond the edge of the image the filtered image depends on values outside of the original image. There are various ways to deal with this problem, e.g. zero-padding as in Definition 6.1, mirroring the image at the boundary or introducing periodic boundary conditions. For

[19] We like to remark that modern deep learning based feature extraction methods such as convolutional neural networks often produce good results, however need to be trained on large data bases to do so and are hence not applicable in our setting, see also the discussion in Section 1.2.

the practical implementation of our work we decided to not to consider pixels too close to the boundary (computationally setting these pixel values to `NA`; see also Remark 4.11 for boundary effects regarding point processes). Since the quality of the fingerprint close to the boundary is often bad, the loss of information due to this is negligible.

Example 6.3 (Smoothing kernels)**.** By application of a smoothing kernel an image is smoothed or blurred and noise is suppressed via building weighted averages of the colour values of pixels within a window which strides over the image, cf. Figure 6.1. Simple examples for smoothing filters are:

(a) The averaging kernel (3×3 box blur)

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} * \mathfrak{I}.$$

gives every pixel within a window around (i, j) (here: 3×3) the same weight. The result is the average of the considered pixel (i, j) and all its 8-neighbours, i.e. all pixels sharing a corner with (i, j), see Figure 6.1b. Indeed, we have

$$\mathfrak{I}^*(i,j) = \frac{1}{9}\sum_{k=-1}^{1}\sum_{\ell=-1}^{1}\mathfrak{I}(i+k, j+\ell).$$

(b) One of the most commonly applied smoothing kernels is the (isotropic) *Gaussian kernel* given by

$$\mathcal{K}(k,\ell) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{1}{2\sigma^2}\left(k^2+\ell^2\right)\right).$$

Note, that actually this kernel has an unbounded support. However, since the weights $\mathcal{K}(k, \ell)$ for large $\|(k, \ell)\|$ are very small and hence the contribution of pixels far apart from the centre of the kernel is negligible, it is common to restrict the kernel to a small window. For standard deviation $\sigma = 1$ for instance, see Figure 6.1c for an illustration, the filtered image using a Gaussian kernel $\mathcal{K}$ of size 5×5 has the form

$$\mathfrak{I}^* = \frac{1}{273}\begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix} * \mathfrak{I}.$$

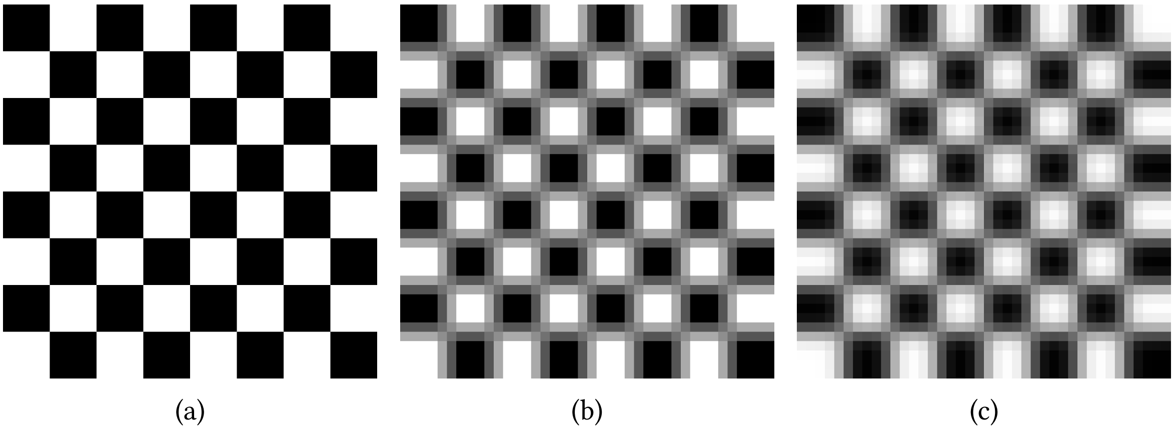


Figure 6.1: *Applications of a 3×3 averaging smoothing kernel (b) and a Gaussian smoothing kernel (b) with $\sigma = 1$ on the chequerboard pattern (a). Notice, that the edges of the original pattern are dilated by the averaging kernel whereas they are more blurry after application of the Gaussian kernel due to the differently assigned weights.*

The choice of the *filter variance* or *bandwidth* σ is often a balancing act between a sufficient smoothing of the image and preserving important image features at the same time, cf. the discussion at the beginning of the section. The particular value is often chosen according to empirical values and adjusted if deemed necessary.

To obtain the ridge lines orientation of an image, we compute the *image gradient* of the grey-valued image, indicating the change of the brightness (from light to dark) of an image. Its magnitude tells us how quickly the image colour is changing whereas its direction tells us the direction where the image changes most rapidly. As a result, image gradients are usually used for edge detection see e.g. Figure 6.2d. For a fingerprint image, the image gradient always points in the direction of the nearest ridge and is orthogonal to it. We can derive an orientation just by rotating the image gradient by $\frac{\pi}{2}$.

Remark 6.4 (Finite differences)**.** Since the image information is only given on a discrete grid, the image gradient rather corresponds to finite differences than to a derivative. From elementary calculus it is known that for a function $\mathfrak{I} : \mathbb{R}^2 \to \mathbb{R}$ on a continuous domain we have

$$\frac{\partial}{\partial x}\mathfrak{I}(x,y) = \lim_{h\to 0} \frac{\mathfrak{I}(x+h,y) - \mathfrak{I}(x-h,y)}{2h},$$
$$\frac{\partial}{\partial y}\mathfrak{I}(x,y) = \lim_{h\to 0} \frac{\mathfrak{I}(x,y+h) - \mathfrak{I}(x,y-h)}{2h}.$$

Hence, for sufficiently small h, the quotient on the right hand side is a good approximation for the derivative. Since the smallest unit in a digital image $\mathfrak{I}$ is one pixel, we use the following approximation of finite differences

$$\frac{\partial}{\partial x}\mathfrak{I}(i,j) \approx \frac{\mathfrak{I}(i+1,j) - \mathfrak{I}(i-1,j)}{2}, \tag{6.1}$$

$$\frac{\partial}{\partial y}\mathfrak{I}(i,j) \approx \frac{\mathfrak{I}(i,j+1) - \mathfrak{I}(i,j-1)}{2}. \tag{6.2}$$

We call (6.1) and (6.2) *image gradient* of $\mathfrak{I}$ in (i,j) in x- and y-direction, respectively.

For an image $\mathfrak{I}$ we can compute its image gradients efficiently by convolution.

Example 6.5 (Derivative filters). (a) The image gradient $\nabla\mathfrak{I} = (G_x\mathfrak{I}, G_y\mathfrak{I})^\top$ of an image $\mathfrak{I}$ can be computed by convolution as follows:

$$G_x\mathfrak{I} = \frac{1}{2}\begin{bmatrix}-1 & 0 & 1\end{bmatrix} * \mathfrak{I}, \quad G_y\mathfrak{I} = \frac{1}{2}\begin{bmatrix}-1\\0\\1\end{bmatrix} * \mathfrak{I}.$$

Indeed, we have for instance

$$\begin{aligned} G_x\mathfrak{I}(i,j) &= \frac{1}{2}\big((-1)\cdot\mathfrak{I}(i-1,j) + 0\cdot\mathfrak{I}(i,j) + (1)\cdot\mathfrak{I}(i+1,j)\big)\\ &= \frac{\mathfrak{I}(i+1,j) - \mathfrak{I}(i-1,j)}{2} \approx \frac{\partial}{\partial x}\mathfrak{I}(i,j), \end{aligned}$$

see Equation (6.1) for the finite differences approximation.

(b) Moreover, image gradients and smoothing kernels are often concatenated to obtain a smoother version of the gradient making the computation more robust to noise. A very popular choice for this is the Sobel filter

$$S_x\mathfrak{I} = \frac{1}{4}\begin{bmatrix}-1 & 0 & 1\\-2 & 0 & 2\\-1 & 0 & 1\end{bmatrix} * \mathfrak{I}, \qquad S_y\mathfrak{I} = \frac{1}{4}\begin{bmatrix}-1 & -2 & -1\\0 & 0 & 0\\1 & 2 & 1\end{bmatrix} * \mathfrak{I}.$$

For an illustration of the filtered images due to application of the Sobel filter, see Figures 6.2b and 6.2c. Figure 6.2d illustrates particularly well the edge detection property of the image gradient's magnitude.

There exists a variety of further derivative filters including information on directed or larger windows e.g. using Gabor filters or Gaussian filters, respectively.

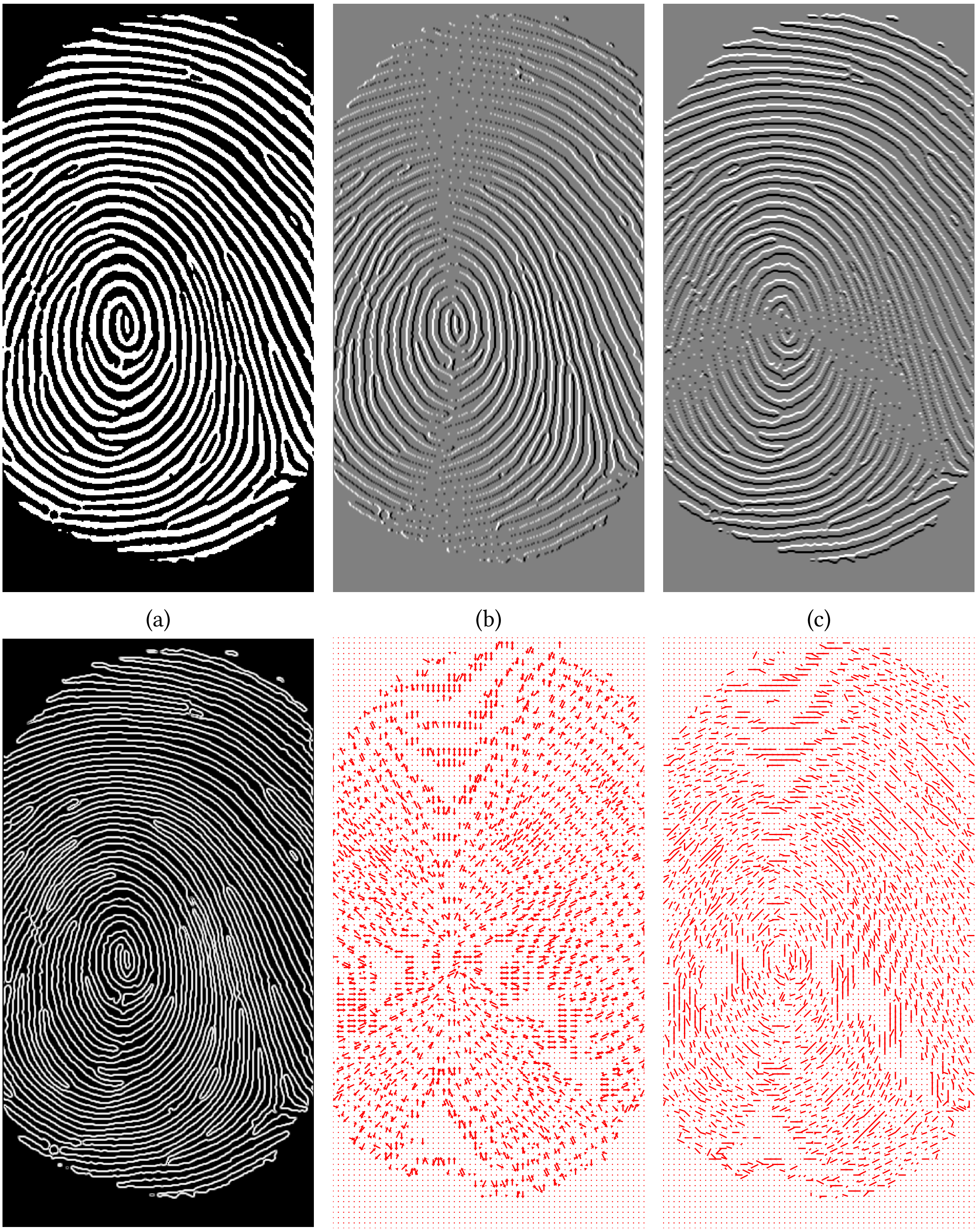


Figure 6.2: *Image gradients using the Sobel filter in x- and y-direction (b), (c), respectively, of the original image (a). The magnitude (d) of the image gradient (e) shows the edges in the image (changes from black to white). For (b) and (c), black denotes negative, white positive and grey zero values. (f) shows the estimated ridge orientation directly from (e) without smoothing.*

Gabor filters are directed according to a given direction and capture features at a given frequency. They are hence particularly suited to detect not only horizontal or vertical but also slanted edges (corresponding to directional derivatives), cf. Moreno et al. (2009); Gottschlich (2011). However, direction and frequency are parameters of the filters which we are trying to determine in the context of fingerprints making them less appropriate for orientation estimation but rather for image enhancement and post-processing.

Larger masks (i.e. filters of larger sizes) will in general give a better approximation of the derivatives since sometimes high frequency noise needs to be removed, however, the computation becomes more expensive. Hence, in practice we have to find a trade-off between performance and accuracy. For our purposes, the Sobel filter turned out to be a good compromise.

In image processing, smoothing filters are also referred to as *low-pass filters*. In contrast to this, *high-pass filters* such as the derivative filters above, have the property of sharpening edges. The terminology originates from signal processing, where images are considered as two-dimensional signals in the Fourier domain and filters correspond to "cutting off" high or low frequencies.

6.1.2 Computation of orientation images

Recall that we denote by $\mathfrak{O} = (\theta(i,j))$ the orientation image of the fingerprint where $\theta(i,j)$ is the local orientation of the ridge line pattern in pixel (i,j), cf. Section 2.1. Since we do not assume the line pattern to be directed, $\theta(i,j)$ only takes values in $[-\frac{\pi}{2}, \frac{\pi}{2})$ and we identify angles modulo π.

Using the results from Section 6.1.1 we can compute a fingerprint's orientation image via image gradients. To obtain more robust results we use the Sobel filters S_x, S_y from Example 6.5. Denote the obtained gradient images by $\mathfrak{G}_x = S_x\mathfrak{I}$, $\mathfrak{G}_y = S_y\mathfrak{I}$. From these gradients we compute the orientation image. The local orientation in $\theta(i,j)$ in (i,j) is the direction orthogonal to the image gradient in (i,j) modulo π, i.e.

$$\theta(i,j) = \angle(\mathfrak{G}_x(i,j), \mathfrak{G}_y(i,j)) + \frac{\pi}{2} \mod \pi \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right) \tag{6.3}$$

where the notation $a \mod \pi \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right)$ shall mean that

$$a \mod \pi = \begin{cases} a, & \text{if } |a| < \frac{\pi}{2}, \\ \pi + a & \text{if } a < -\frac{\pi}{2}, \\ \pi - a & \text{if } a \geq \frac{\pi}{2}. \end{cases}$$

Note, that (6.3) can be computed using the $\arctan2$ function, see also Section 3.2.1, via

$$\angle(x,y) = \arctan2(x,y) = \begin{cases} \arctan\left(\frac{y}{x}\right), & x > 0, \\ \arctan\left(\frac{y}{x}\right) + \operatorname{sgn}(y)\,\pi, & x < 0, \\ \operatorname{sgn}(y)\,\frac{\pi}{2}, & x = 0, y \neq 0. \end{cases}$$

where $\operatorname{sgn}(y)$ is the sign of $y \in \mathbb{R}$. Note that $\arctan2$ is not defined for $(x,y) = (0,0)$.

In applications it turned out that this approach is still prone to noise and the so obtained orientation image is often not smooth enough (see also Figure 6.2f) to induce a $\mathcal{C}^2$ vector field (apart from singularities) as required in Chapter 3. A subsequent smoothing routine is hence desirable or even necessary. Common smoothing procedures cannot be applied to the obtained orientation image directly since opposed orientations would cancel. For example, the mean orientation between $1°$ and $179°$ would rather be $0°$ than $\frac{1}{2}(1° + 179°) = 90°$. Moreover, averaging is not even always well-defined. For instance, the mean orientation between $0°$ and $90°$ can be either $45°$ or $135°$.

To solve this problem we follow an approach by Bazen & Gerez (2002). Computationally, every orientation is interpreted as direction in $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right)$. The idea is that each direction $\psi \in [-\pi, \pi)$ corresponds to a vector $\mathrm{e}^{\mathrm{i}\psi}$ in the complex plane $\mathbb{C}$ representing this direction. By squaring these vectors, opposite gradient vectors will point in the same direction (i.e. representing the same orientation) preventing that they cancel when computing an (extrinsic) mean in $\mathbb{C}$. After averaging, we convert the gradient vectors $\mathrm{e}^{\mathrm{i}\bar{\theta}}$ back via taking the root $\mathrm{e}^{\mathrm{i}\bar{\theta}/2}$.

Hence, we compute the squared gradients $(\mathfrak{S}_x, \mathfrak{S}_y)$ in a pixel (i,j) from the image gradients $\mathfrak{G}_x$, $\mathfrak{G}_y$ via

$$\mathfrak{S}_x(i,j) = \mathfrak{G}_x(i,j)^2 - \mathfrak{G}_y(i,j)^2, \qquad \mathfrak{S}_y(x,y) = 2\mathfrak{G}_x(i,j)\mathfrak{G}_y(i,j),$$

corresponding to real and imaginary part of the squared gradient. We now apply a Gaussian filter $\mathcal{K}$ to the squared gradient in order to obtain $\bar{\mathfrak{S}}_x$, $\bar{\mathfrak{S}}_y$ with

$$\bar{\mathfrak{S}}_x = \mathcal{K} * \mathfrak{S}_x, \qquad \bar{\mathfrak{S}}_y = \mathcal{K} * \mathfrak{S}_y.$$

To obtain the corresponding smoothed orientation image, we take the “square root” of this vector and rotate this again by $\frac{\pi}{2}$, i.e. we compute

$$\bar{\theta}(i,j) = \left(\frac{1}{2}\angle(\bar{\mathfrak{S}}_x, \bar{\mathfrak{S}}_y) + \frac{\pi}{2}\right) \mod \pi \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right).$$

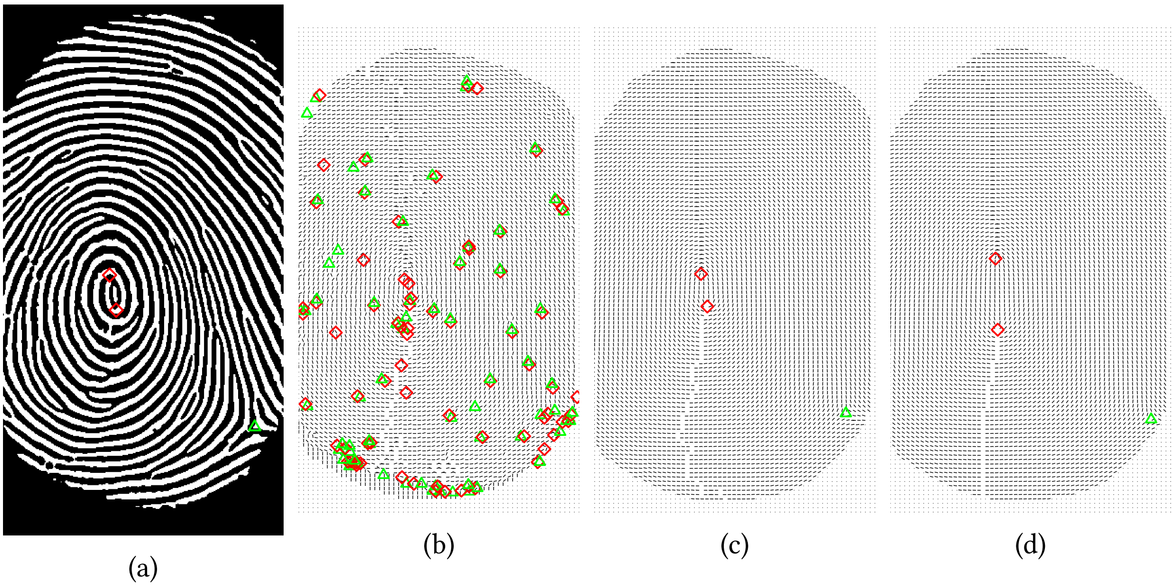


Figure 6.3: *Results of the smoothing procedure by Bazen & Gerez (2002) of the orientation image of the fingerprint (a) for different choices of* $\sigma = 1, 7, 20$ *((b)–(d), left to right). Clearly, the orientation images become smoother as* σ *increases. Detected cores using the Poincaré index are marked as red diamonds (*$\diamond$*), deltas as green triangles (*Δ*). As* σ *increases, the number of spuriously detected singularities decreases, however we observe displacement of the loop-type singularities towards the boundary. The drawn singularities for (a) were computed using* $\sigma = 7$*, cf. (c).*

Again, naturally the question arises which smoothing variance to choose for the smoothing kernel $\mathcal{K}$. To answer this, we considered several band widths, cf. Figure 6.3. A larger σ leads, as mentioned above, to a smoother orientation field, however also to a larger displacement of the loop-type singularities. In Figure 6.3 for instance, the upper core of the whorl moves upwards as σ increases, cf. also the discussion in (Maltoni et al., 2009, p. 122). Since for the analysis in Chapter 3 regions close to a singularity are excluded, we have to rely on a relatively precise estimation of the singularities. We hence choose $\sigma = 7$ for our further analysis since this seems to be a good compromise between a smooth orientation image and not too large displacement of the singularities on all considered images.

6.1.3 Singularity detection

Using the orientation image, we can now estimate the locations of the orientation field's singularities. An algorithm to do so is provided for instance by Bazen & Gerez (2002) or Awad & Baba (2012), and is based on the Poincaré index, cf. Definition 3.6.

Recall from Theorem 3.9 that the Poincaré index in $z_0 \in A$ can be computed as

$$\operatorname{index}_\theta(z_0) = \frac{1}{2\pi}\int_0^{2\pi} \langle \nabla\theta(\gamma(t)), \dot{\gamma}(t)\rangle \,\mathrm{d}t = \frac{1}{2\pi}\iint_A \operatorname{curl}\nabla\theta(z)\,\mathrm{d}z,$$

where $A \subseteq \mathbb{R}^2$ is compact and $\gamma : [0, 2\pi] \to \partial A$ is a positively oriented, piecewise smooth, closed curve enclosing z_0.

An implementation to compute the Poincaré index directly from the definition using cumulative changes in the orientation is described in Hong & Jain (1999). The algorithm presented there depends on the particular choice of the curve γ which the authors claim to be optimal for a square curve with a side length 25 pixels. A smaller curve results in spurious detections while a larger curve may ignore core-delta-pairs lying close to each other. This results in an image containing the Poincaré index for each pixel. Classification into the different types of singularities and regular points is finally applied based on connected components with similar Poincaré index. Hence, the localisation of singular points is relatively imprecise.

By contrast, the method by Bazen & Gerez (2002) relies on Theorem 3.9 and promises to extract all singular points, including the ones due to insufficient smoothing, see Figure 6.3b, at a precision of up to one pixel (i.e. maximal 2×2 pixel size singular points). The algorithm first takes the squared orientation image which is obtained from the orientation image by doubling the orientation angles. This eliminates the discontinuity of the orientation field between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$, cf. Section 6.1.2. By doing so, the Poincaré indices change to −1, 1, 2 and 0 for, respectively, a delta, a core, a whorl and a regular point.

Then we compute

$$\frac{1}{2\pi}\iint_A \operatorname{curl}\nabla 2\theta(z)\,\mathrm{d}z \approx \frac{1}{2\pi}\sum_{(i,j)\in A}\left(\frac{\partial}{\partial x}\frac{\partial}{\partial y}2\theta(i,j) - \frac{\partial}{\partial y}\frac{\partial}{\partial x}2\theta(i,j)\right)$$

whereas the derivatives are to be understood to be computed via finite differences as described in Section 6.1.1. In our application we choose A to be one pixel such that we can extract all singular points as precise as possible, cf. the discussion above. The classification is implemented by thresholding according to the above mentioned reference values, cf. also Theorem 3.8, with accuracy 10^{-3}.

The precision of the singularity point extraction depends severely on the extracted orientation field and the Poincaré index may lead to false singularities in noisy or low-quality regions of the fingerprint, see Figure 6.3b. A regularisation of the orientation field (e.g. via additional smoothing) is often quite effective in preventing the detection of false singularities, however, usually leads to displacement of the loop-type singularity positions towards the boundary of the image (see Figure 6.3d). For all images considered, $\sigma = 7$ seems to be a suitable choice. Since the detection of false singularities is often an indicator for bad quality of the extracted orientation image (given a predetermined amount of smoothing), we do not apply further regularisation on the orientation image. We rather see this as an advantage in information and exclude also regions around “false” singularities from further analysis to obtain results as precise as possible in the remaining region of interest.

6.2 Ridge frequency

Recall from Section 2.1 that the ridge frequency in a pixel (i, j) is the number of ridges per unit length (measured in pixels) along a hypothetical line segment centred at (i, j) and orthogonal to the local ridge orientation $\theta(i, j)$ in (i, j). In a region of the image where the ridge flow is continuous (i.e. no singularities or minutiae), the grey levels of the image form a (discrete) sinusoidal wave along a direction orthogonal of the local orientation, cf. Figure 6.4. The waves' frequency is the same as that of the ridges or valleys and hence the distance between two consecutive peaks correspond to the inter-ridge distance which is the reciprocal of the ridge frequency.

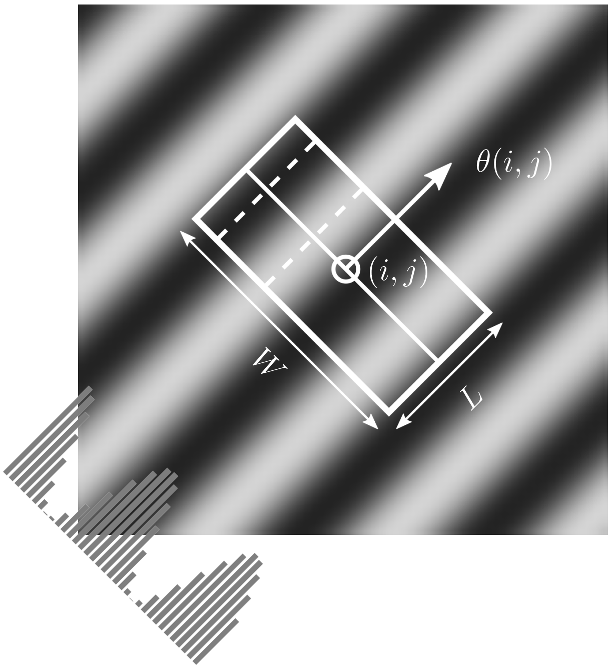


Figure 6.4: *An oriented window of size $L \times W$, centred at (i, j) for computing the ridge frequency. The dashed lines indicate the pixels whose colour values are accumulated for a given column of the x-signature, cf. Hong et al. (1998). The x-signature of (i, j) can be seen in the lower left as a bar plot.*

To compute the ridge frequency we follow an approach presented in Hong et al. (1998). For every pixel (i, j) we compute an oriented, rectangular window[20] centred at (i, j) that covers

[20] For our implementations we use $L = 33$, $W = 17$, cf. Algorithm 6.6.

W pixel rows along each containing L pixels orthogonal to the local ridge orientation, cf. Figure 6.4. Then, the grey-values are column-wisely cumulated; the resulting vector is called x-*signature*.

In Figure 6.4, the x-signature of the pixel (i, j) is indicated as a bar plot on the bottom left corner. Now, we compute the distance s_m between every two consecutive peaks in the x-signature. Averaging the s_m and taking the reciprocal results in the local ridge frequency $\varphi(i, j)$ in (i, j).

Since this approach is very prone to noise, which can cause additional local extrema in the grey-value profile, cf. Figure 6.5, Hong et al. (1998) provide an additional smoothing step. Experiments showed that this is often not sufficient and that considering the binarised images instead of grey-value images lead to more reliable results. Hence, we use for our computations the binarisation provided by the approach from Thai et al. (2016).

We summarise the procedure for computing the ridge frequency in the following algorithm:

Algorithm 6.6 (Ridge frequency).

```
Input  :  Binarised fingerprint image $\mathfrak{I}$, orientation image $\mathfrak{O} = (\theta(i, j))_{i,j}$.
for every pixel $(i, j)$ do
    Compute an oriented window of size $L \times W$ (33 × 17) that is defined in the
      ridge coordinate system centred in $(i, j)$, see Figure 6.4.
    Compute the $x$-signature $X[0], X[1], \dots, X[L-1]$ of the ridges and valleys
      within the oriented window, where
        $$X[\ell] := \frac{1}{W} \sum_{k=0}^{W-1} \mathfrak{I}(u_{\ell,k}, v_{k,\ell}), \quad \ell = 0, 1, \dots, L-1,$$
      and
        $$u_{\ell,k} = \left\lfloor i + \left(k - \frac{W}{2}\right) \cos\theta(i, j) + \left(\ell - \frac{L}{2}\right) \sin\theta(i, j) \right\rceil,$$
        $$v_{\ell,k} = \left\lfloor j + \left(k - \frac{W}{2}\right) \sin\theta(i, j) + \left(\frac{L}{2} - \ell\right) \cos\theta(i, j) \right\rceil,$$
      where $\lfloor x \rceil$ denotes rounding $x$ to the nearest integer, cf. p. 156.
    Compute the average $T(i, j)$ of the distances $s_m$ between two consecutive
      peaks in the $x$-signature, cf. Figure 6.5.
    Set $\varphi(i, j) = \frac{1}{T(i,j)}$.
end for
If deemed reasonable, apply a smoothing kernel to $\varphi$.
Output:  Ridge frequency image $\mathfrak{F} = (\varphi(i, j))$.
```

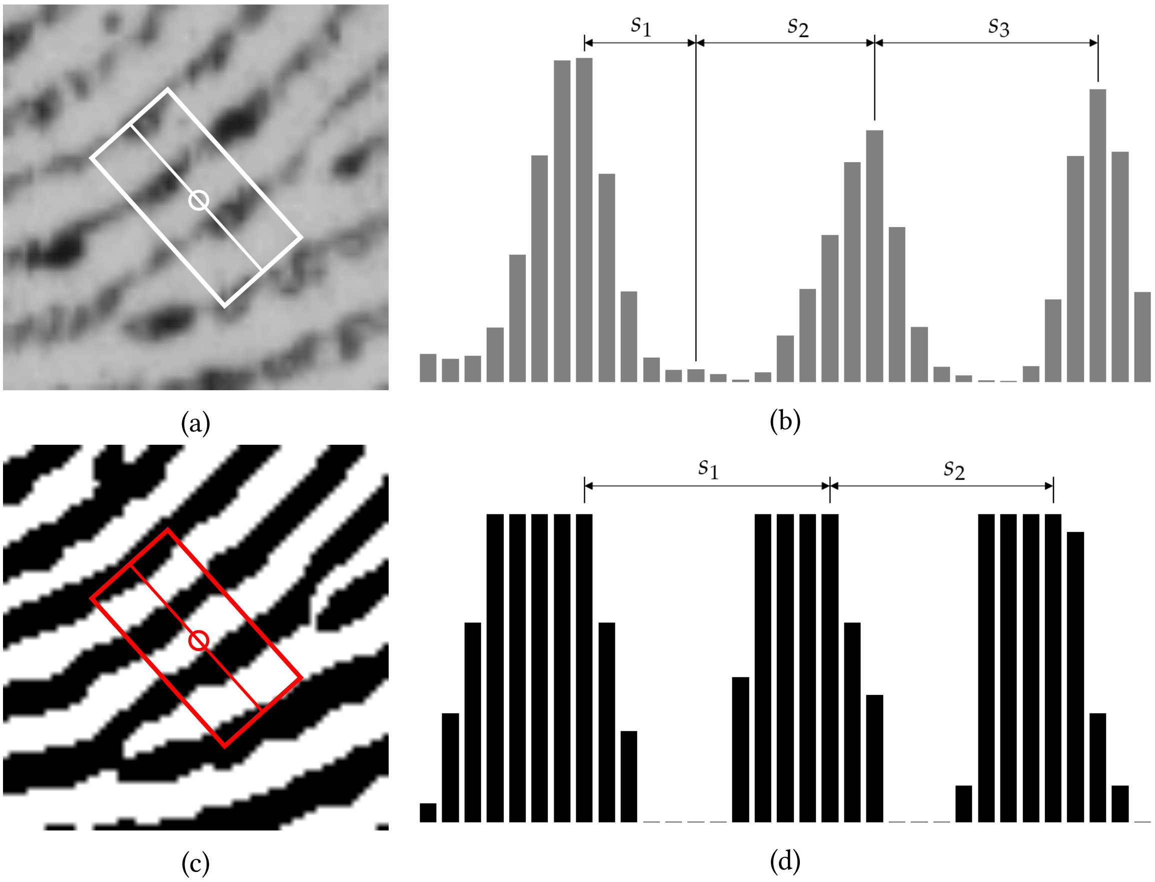


Figure 6.5: *An oriented window centred at the pixel ⊖ for a section of a grey-valued image (a) and the binarised counterpart (c). The local inter-ridge distance is computed as the average of the distances s_m between the peaks in the x-signature (right column). Notice, that the x-signature of the grey-value image (b) exhibits an additional small peak due to noise which is not present in the x-signature of the binarised image (d) and resulting in a much smaller ridge frequency for (d).*

Remark 6.7 (Ridge frequency via Fourier transform)**.** In the literature, there are apart from the presented approach for the computation of the ridge frequency, algorithms based on Fourier transform analysis, see e.g. Chikkerur et al. (2007); Patriciu & Spinu (2014). Zhan et al. (2006) compared ridge frequency estimation in the spatial domain and the Fourier domain and found that the former ones can be implemented more efficiently while the latter appear to be more robust to noise. Since the binarisation preprocessing seems to solve the problem of noise for the images considered to a large extent, we chose to adhere with the approach presented above.

6.3 Divergence and intensity

In this section we provide an algorithm for computing the divergence field of a fingerprint's ridge frequency weighted orientation field as in Theorem 3.13. We split the computation up into the computation of the fingerprint's orientation field divergence and the ridge frequency divergence, see (3.15) and (3.16).

Divergence of the orientation field A vector field's divergence, is the sum of the partial derivatives of its components. As a vector field has a direction, the divergence is a directed quantity. However, the orientation field has only an orientation, so in each point we can assign even two directions pointing in opposite directions, cf. Section 3.2.1. To get a meaningful divergence field we thus have to direct the orientation field (i.e. we have to decide for one the two possible directions). In general, it is not possible to globally direct the orientation field continuously as long as it contains singularities, cf. (Sherlock, 2004, Theorem 5.3.3.1). However, we can divide the image in small simply connected subregions (e.g. using a regular grid), exclude the regions containing the singularities and direct the orientation field within the remaining grid patches such that we obtain a continuous field within these subregions. The procedure of doing so is called *unwrapping* since we unwrap the angles from values in $[-\frac{\pi}{2}, \frac{\pi}{2}]$ to the real line such that the resulting function is continuous as a function from the grid patch to $\mathbb{R}$. Computationally, we obtain the unwrapped orientation $\psi(i, j)$ from the (wrapped) orientation $\theta(i, j)$ by adding an integer multiple of π, i.e.,

$$\psi(i, j) = \theta(i, j) + \pi k(i, j), \tag{6.4}$$

where $\theta(i, j) \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right)$ and $k(i, j) \in \mathbb{Z}$ is chosen such that ψ becomes continuous[21] for all pixels (i, j) within the considered subregion of the ROI.

Since the orientation field is only available via the orientation image, we compute image gradients (of the unwrapped orientation) instead of partial derivatives and compute the divergence from these image gradients. In general, if the orientation image exhibits singularities the unwrapping procedure is path dependent, i.e. the value of ψ depends on the order the pixels (i, j) are unwrapped. Hence, if we restrict ourselves only to patches not containing singularities this issue is theoretically not of concern. However, the determination of singularities is in practice often imprecise (depending on the degree of smoothness

[21] For discrete images we mean by continuity that the difference between the unwrapped orientation of adjacent pixels is small. We choose a tolerance of 10^{-3} for our implementations.

of the OF, cf. Section 6.1) and experiments showed that the unwrapping path indeed can affect the result of the unwrapping procedure even in patches which do not contain a singularity but are close to one.

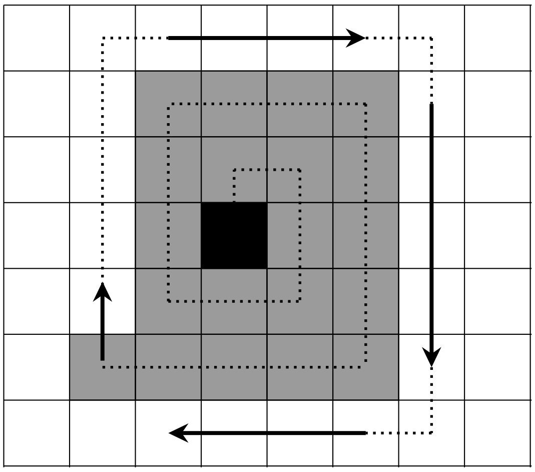


Figure 6.6: *Unwrapping scheme of the algorithm. The start pixel is indicated in black, already unwrapped pixels are grey and pixels yet to unwrap in white. The unwrapping path is indicated with dots ($\cdots$), the direction with arrows.*

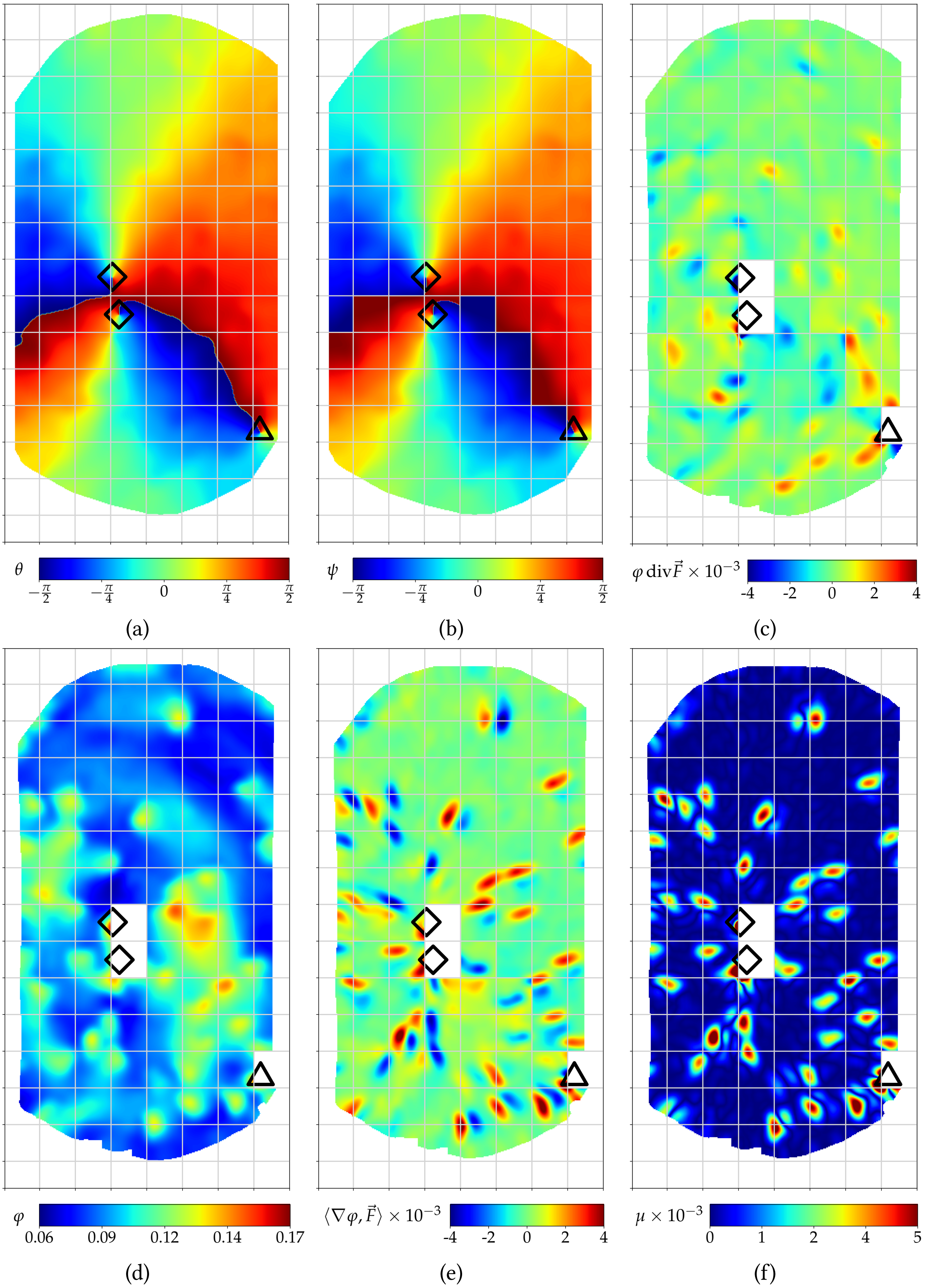
For our purposes, we use a pixel queue based unwrapping algorithm following a spiral path around the centre of a given patch, cf. Figure 6.6. We start the unwrapping procedure at the centre $(0,0)$ of a considered patch A and set $k(0,0) = 0$. The pixels are ordered into a queue Q according to the spiral path. Then, the integer $k(i,j)$ for the first pixel of the queue is determined such that $\psi(\ell,m)$ is continuous (in the above sense) for all already unwrapped pixels $(\ell,m) \in A \setminus Q$. We then remove the first element from the queue and iterate with the subsequent new front element. This turned out to perform reasonably well on all considered data. For a more detailed overview about robust phase unwrapping algorithms we refer to the literature, e.g. Ghiglia & Pritt (1998); Antonopoulos et al. (2015).

An example for the orientation image $\mathfrak{O} = (\theta(i,j))$ and its unwrapped version $\tilde{\mathfrak{O}} = (\psi(i,j))$ can be seen in Figure 6.7a and 6.7b, respectively. Note that within grid tiles the unwrapped orientation image is (sufficiently) smooth and that the lines of discontinuity after unwrapping are only on the grid lines but not within grid tiles anymore. Hence, it is reasonable to assume that the vector field $\vec{F}(i,j) = (\cos\psi(i,j), \sin\psi(i,j))^\top$ corresponding to the unwrapped orientation $\tilde{\mathfrak{O}}$ is $\mathcal{C}^2$ on the interior of the patches.

Using this sufficiently smooth unwrapped orientation image $\tilde{\mathfrak{O}}$ and the ridge frequency image $\mathfrak{F} = (\varphi(i,j))$, we can compute the OF divergence image $\mathfrak{D}\mathfrak{O}$ as a discrete approximation of Equation (3.15) via

$$\begin{aligned}\mathfrak{D}\mathfrak{O}(i,j) &= \varphi(i,j)\left(\frac{\cos\psi(i+1,j)-\cos\psi(i-1,j)}{2} + \frac{\sin\psi(i,j+1)-\sin\psi(i,j-1)}{2}\right)\\ &\approx \varphi(i,j)\left(\partial_x F_1(i,j) + \partial_y F_2(i,j)\right) = \varphi(i,j)\,\operatorname{div}\vec{F}(i,j).\end{aligned}$$

Note that, due to the finite difference approximation, we need to unwrap for each grid tile an additional padding of one pixel. An example for the divergence image can be seen in



(a) (b) (c)

(d) (e) (f)

Figure 6.7: *Top: The orientation image before (left) and after unwrapping (middle) using a* 8×15 *grid (side length 37×37 pixels). The resulting absolute OF divergence field is depicted in (c). Bottom: The ridge frequency image and its RF divergence image (d), (e) and the total minutiae intensity image (f). Patches containing singularities (◇ and △) were excluded from the computations.*

Figure 6.7c. Note also that the OF divergence image, as the unwrapped orientation, is only continuous within the patches but not necessarily globally, see e.g. the patches west and north-west of the delta at the bottom right of Figure 6.7c.

Divergence of ridge frequency The ridge frequency divergence in a pixel (i, j) is, cf. (3.16), given as

$$\left\langle \nabla\varphi(i,j), \vec{F}(i,j) \right\rangle .$$

Note, that this quantity also depends on the direction chosen for the orientation field. Since we have to combine the divergence of the orientation field and the ridge frequency to obtain the intensity of necessary minutiae, cf. Theorem 3.13 , we have to choose the direction consistently with the one chosen in the previous paragraph, i.e. we use the orientation field $\tilde{\mathfrak{O}}$ which is unwrapped on the same patches as above, i.e. $\vec{F}(i,j) = (\cos\psi(i,j), \sin\psi(i,j))^\top$. Let $(G_x\mathfrak{F}, G_y\mathfrak{F})$ be the finite differences approximation of $\nabla\varphi$, cf. Example 6.5. Then, the ridge frequency divergence image $\mathfrak{D}\mathfrak{F}$ is computed via

$$\mathfrak{D}\mathfrak{F}(i,j) = G_x\mathfrak{F}(i,j)\cos\psi(i,j) + G_y\mathfrak{F}(i,j)\sin\psi(i,j).$$

An example can be seen in Figure 6.7 where the ridge frequency image is depicted in panel 6.7d and the resulting RF divergence in panel 6.7e.

Intensity of the necessary minutiae The intensity of necessary minutiae in a certain pixel can, given the orientation field divergence image and the ridge frequency divergence image, according to Theorem 3.13 be estimated as

$$\mu(i,j) = \left| \varphi(i,j) \operatorname{div} \vec{F}(i,j) + \left\langle \nabla\varphi(i,j)\vec{F}(i,j) \right\rangle \right| .$$

Combining the divergence images from the two previous paragraphs, we obtain the necessary minutiae intensity image $\mathfrak{M}$ as $\mathfrak{M}(i,j) = |\mathfrak{F}(i,j)\mathfrak{D}\mathfrak{O}(i,j) + \mathfrak{D}\mathfrak{F}(i,j)|$. An example is depicted in panel 6.7f of Figure 6.7. We use this image $\mathfrak{M}$ as the activity/ trend image for the Strauss process for our computation.

Having computed the intensity images we can now formulate an algorithm to separate a given minutiae pattern into random and necessary minutiae. This will be subject of Section 6.4.

6.4 Bayesian inference using MCMC

This section is dedicated to the implementation details of an MCMC algorithm (as in Section 5.2.1 but we allow now also for changes in $\mathbf{W}$, see Section 6.4.3) for sampling from the posterior distribution of the parameters of the model from Section 4.4 given some minutiae pattern $\zeta = \{z_1, z_2, \ldots, z_n\} \subseteq \mathfrak{X}$. Recall that we denote the model parameters by $\boldsymbol{\theta} = (\lambda, \beta, \gamma) \in \Theta = \mathbb{R}_+ \times \mathbb{R}_+ \times (0,1)$ and the label vector by $\mathbf{W} \in \{0,1\}^n$. Moreover, $\mathfrak{X}$ denotes the region of interest. Note, that the tuning of hyper-parameters occurring in this section is based on the 20 high quality fingerprints from Figure 3.6 and all data-related statements made below relate to this data base. Bayes' theorem (5.2) yields for the posterior distribution

$$\pi(\boldsymbol{\theta}, \mathbf{W} \mid \zeta) = \frac{h_{\boldsymbol{\theta},\mathbf{W}}(\zeta)\, \pi(\boldsymbol{\theta}, \mathbf{W})}{\int h_{\boldsymbol{\vartheta},\mathcal{W}}(\zeta)\, \pi(\boldsymbol{\vartheta}, \mathcal{W})\, \mathrm{d}(\boldsymbol{\vartheta}, \mathcal{W})}. \tag{6.5}$$

The framework for the MCMC algorithm is described in the following Algorithm 6.8.

Algorithm 6.8 (Framework for the Minutiae Separation Algorithm (MiSeal)).

```
Input  : Minutiae pattern $\zeta = \{z_1, z_2, \ldots, z_n\} \subseteq \mathfrak{X}$.
Choose some initial $(\boldsymbol{\theta}^{(0)}, \mathbf{W}^{(0)}) \in \Theta \times \{0,1\}^n$. Draw $\tilde{\chi}^{(0)} \sim h_{\boldsymbol{\theta}^{(0)},\mathbf{W}^{(0)}}(\cdot)$.
for $t = 0, 1, 2, \ldots$ do
    Given $(\boldsymbol{\theta}^{(t)}, \mathbf{W}^{(t)}, \tilde{\chi}^{(t)}) = (\boldsymbol{\theta}, \mathbf{W}, \tilde{\chi})$, generate a candidate $(\boldsymbol{\theta}', \mathbf{W}', \tilde{\chi}')$ for the
      next sample from the probability density $\tilde{q}(\boldsymbol{\theta}', \mathbf{W}', \tilde{\chi}' \mid \boldsymbol{\theta}, \mathbf{W}, \tilde{\chi})$.
    Calculate the *Hastings ratio*
      $$H(\boldsymbol{\theta}', \mathbf{W}', \tilde{\chi}' \mid \boldsymbol{\theta}, \mathbf{W}, \tilde{\chi}) = \frac{\pi(\boldsymbol{\theta}', \mathbf{W}', \tilde{\chi}' \mid \zeta)}{\pi(\boldsymbol{\theta}, \mathbf{W}, \tilde{\chi} \mid \zeta)} \frac{\tilde{q}(\boldsymbol{\theta}, \mathbf{W}, \tilde{\chi} \mid \boldsymbol{\theta}', \mathbf{W}', \tilde{\chi}')}{\tilde{q}(\boldsymbol{\theta}', \mathbf{W}', \tilde{\chi}' \mid \boldsymbol{\theta}, \mathbf{W}, \tilde{\chi})}. \tag{6.6}$$
    Accept the candidate with probability $\min\{H(\boldsymbol{\theta}', \mathbf{W}', \tilde{\chi}' \mid \boldsymbol{\theta}, \mathbf{W}, \tilde{\chi}), 1\}$.
    In case of acceptance set $(\boldsymbol{\theta}^{(t+1)}, \mathbf{W}^{(t+1)}, \tilde{\chi}^{(t+1)}) = (\boldsymbol{\theta}', \mathbf{W}', \tilde{\chi}')$, otherwise
      $(\boldsymbol{\theta}^{(t+1)}, \mathbf{W}^{(t+1)}, \tilde{\chi}^{(t+1)}) = (\boldsymbol{\theta}, \mathbf{W}, \tilde{\chi})$.
end for
Project the samples to $\Theta \times \{0,1\}^n$ (i.e. drop $\tilde{\chi}$).
If deemed necessary, discard the first $t_0$ samples (burn-in).
Output:  A sample $(\boldsymbol{\theta}^{(t)}, \mathbf{W}^{(t)})_{t=t_0+1,t_0+2,\ldots}$ from the distribution induced by
         $\pi(\boldsymbol{\theta}, \mathbf{W} \mid \zeta)$.
```

For our model from Section 4.4 we have

$$h_{\boldsymbol{\theta},\mathbf{W}}(\zeta) = f_\lambda\left(\{z_i : \mathbf{W}_i = 0\}\right)\, g_{\beta,\gamma}\left(\{z_i : \mathbf{W}_i = 0\}\right)$$

is the model likelihood, cf. Equation (4.19), of the labelled data given parameters and labels. $\pi(\boldsymbol{\theta}, \mathbf{W})$ models our *prior* belief about the parameters and labels which we will define in Section 6.4.1.

Recall from Section 5.2 that we face a doubly intractable problem because the denominator in (6.5) is intractable, and $h_{\boldsymbol{\theta},\mathbf{W}}(\zeta)$ in the numerator contains another intractable normalising constant. Hence, we employ Algorithm 6.8 which is a variant of the auxiliary variable method (using auxiliary point patterns $\tilde{\chi}, \tilde{\chi}' \in \mathfrak{N}$), see Algorithm 5.8, to produce samples from the posterior distribution (6.5).

Remark 6.9. The performance of Algorithm 6.8 crucially depends on the computation of the Hastings ratio (6.6). Similarly to Equation (5.7) we can rewrite the Hastings ratio as

$$\begin{aligned} H(\boldsymbol{\theta}', \mathbf{W}', \tilde{\chi}' \mid \boldsymbol{\theta}, \mathbf{W}, \tilde{\chi}) &= \frac{\pi(\boldsymbol{\theta}', \mathbf{W}', \tilde{\chi}' \mid \zeta)}{\pi(\boldsymbol{\theta}, \mathbf{W}, \tilde{\chi} \mid \zeta)} \frac{\tilde{q}(\boldsymbol{\theta}, \mathbf{W}, \tilde{\chi} \mid \boldsymbol{\theta}', \mathbf{W}', \tilde{\chi}')}{\tilde{q}(\boldsymbol{\theta}', \mathbf{W}', \tilde{\chi}' \mid \boldsymbol{\theta}, \mathbf{W}, \tilde{\chi})} \\ &= \frac{h_{\boldsymbol{\theta}',\mathbf{W}'}(\zeta)}{h_{\boldsymbol{\theta},\mathbf{W}}(\zeta)} \frac{\varphi(\tilde{\chi}' \mid \mathbf{W}', \zeta)}{\varphi(\tilde{\chi} \mid \mathbf{W}, \zeta)} \frac{\pi(\boldsymbol{\theta}', \mathbf{W}')}{\pi(\boldsymbol{\theta}, \mathbf{W})} \frac{\tilde{q}(\boldsymbol{\theta}, \mathbf{W}, \tilde{\chi} \mid \boldsymbol{\theta}', \mathbf{W}', \tilde{\chi}')}{\tilde{q}(\boldsymbol{\theta}', \mathbf{W}', \tilde{\chi}' \mid \boldsymbol{\theta}, \mathbf{W}, \tilde{\chi})} \end{aligned}$$

where φ is the density of $\tilde{\chi}$ w.r.t. the standard Poisson process which *must not* depend on the value of $\boldsymbol{\theta}$, cf. Section 5.2, since otherwise the Hastings ratio would again contain intractable normalising constants.

In the following sections we elaborate further details on how to compute the individual quotients, the choice of our priors as well as the update procedures for $(\boldsymbol{\theta}, \tilde{\chi})$ and $\mathbf{W}$. Note that the auxiliary point pattern $\tilde{\chi}$ is only needed for the update of $\boldsymbol{\theta}$, cf. Section 5.2.1. Hence, we will always propose $(\boldsymbol{\theta}, \tilde{\chi})$ jointly, however do not propose a new $\tilde{\chi}$ in a $\mathbf{W}$ update step.

For the updates, we employ a random scan Gibbs sampler, cf. Section 5.1.2 or Liu (2008), with update probabilities $p_{\boldsymbol{\theta}}$ for $\boldsymbol{\theta}$ and $1-p_{\boldsymbol{\theta}}$ for $\mathbf{W}$ because the computation of the Hastings ratio for a joint update turned out to be difficult. A value of $p_{\boldsymbol{\theta}} = 0.05$ (reflecting on average 19 proposed flips out of the approx. 30–60 minutiae per finger for each $\boldsymbol{\theta}$ update) approximately yields the fastest mixing. The update for each component employs a Markov chain yielding a variant from the *Metropolis-within-Gibbs* class of algorithms, see Roberts & Rosenthal (2006).

6.4.1 Choice of priors

We write $\pi(\boldsymbol{\theta}, \mathbf{W}) = \pi(\boldsymbol{\theta})\pi(\mathbf{W})$ (assuming independence of model parameters $\boldsymbol{\theta}$ and labels $\mathbf{W}$) and suppose the prior of the parameters $\pi(\boldsymbol{\theta}) = \pi(\lambda)\pi(\beta)\pi(\gamma)$ is a product of the priors of the single parameters chosen as follows.

We choose $\beta \sim \Gamma(a_1, b_1)$ where the parameters $a_1 = b_1 = 5$ are chosen such that the expected value of β equals one (see the discussion below Equation (4.18)) and the variance is reasonably large. For flexibility we choose $\gamma \sim \mathrm{B}(p_1, q_1)$ to be beta distributed with $p_1 = 2$ and $q_1 = 5$ (hence, $\mathbb{E}\gamma = \frac{2}{7}$), since we expect the process to be rather inhibitive also outside the hard core distance.

For the intensity λ of the random minutiae we choose the prior to be a Gamma distribution $\Gamma(a_0, b_0)$. Being conjugate to the Poisson likelihood, cf. Example 5.2, this prior has the advantage that we do not have to perform Hastings steps when updating λ, but can draw directly from the posterior distribution conditional on $\mathbf{W}, \beta$ and γ, cf. (6.8) below. We choose the parameters $a_0 = 5$, $b_0 = \frac{5}{\lambda_0}$ such that the expected value of the prior is $\lambda_0 = 10^{-4}$ from (3.19) and its variance is $\lambda_0^2/5$ such that the ratio $1/\sqrt{5}$ of standard deviation over mean is reasonably sized.

Furthermore, for every imprint featuring n minutiae in its region of interest $\mathfrak{X}$, Remark 3.16 yields an expected number $\lambda_0|\mathfrak{X}|$ of random minutiae in $\mathfrak{X}$. Hence, for each finger individually, we choose the prior for the label vector as $\pi(\mathbf{W}) = \bigotimes_{i=1}^{k} \pi(\mathbf{W}_i)$ with $\mathbf{W}_i \overset{\text{i.i.d.}}{\sim} \mathrm{Ber}\,(p_{\mathbf{W}})$ and $p_{\mathbf{W}} = \max\left\{1 - \frac{\lambda_0|\mathfrak{X}|}{n}, 0\right\}$. We discard infeasible label vectors, i.e. label vectors for which some pairs of $\{z_i \mid \mathbf{W}_i = 1\}$ have distances smaller than r and thereby violate the hard core condition. The effect on the procedure is irrelevant since infeasible $\mathbf{W}$'s would be rejected by the MCMC algorithm in any case. The ratio of priors for different parameters $(\boldsymbol{\theta}, \mathbf{W})$, $(\boldsymbol{\theta}', \mathbf{W}')$ thus computes as

$$\begin{aligned}\frac{\pi(\boldsymbol{\theta}', \mathbf{W}')}{\pi(\boldsymbol{\theta}, \mathbf{W})} &= \frac{\pi(\boldsymbol{\theta}')}{\pi(\boldsymbol{\theta})}\frac{\pi(\mathbf{W}')}{\pi(\mathbf{W})} \\ &= \left(\frac{\lambda'}{\lambda}\right)^{a_0-1} \mathrm{e}^{-b_0(\lambda'-\lambda)} \left(\frac{\beta'}{\beta}\right)^{a_1-1} \mathrm{e}^{-b_1(\beta'-\beta)} \left(\frac{\gamma'}{\gamma}\right)^{p_1-1} \left(\frac{1-\gamma'}{1-\gamma}\right)^{q_1-1} \left(\frac{p_{\mathbf{W}}}{1-p_{\mathbf{W}}}\right)^{\ell'-\ell},\end{aligned} \tag{6.7}$$

where $1 \leq \ell, \ell' \leq n$ denote the number of ones in $\mathbf{W}, \mathbf{W}'$, respectively. We have adjusted the variances of the priors such that they concentrate on a domain we deem reasonable according to a some pilot experiments. We keep them rather uninformative, however, to avoid undesirable dependence of the posterior on our particular prior choices.

6.4.2 Update of $\boldsymbol{\theta}$

When updating $\boldsymbol{\theta}$, we randomly choose to update either λ or (β, γ). To this end, toss a coin with success probability $p_\lambda = 0.2$ (this is slightly smaller than $1/3$, taking into account that, due to the explicitly available posterior, see below, there are no rejections for the λ-updates). In case of success, update λ, which, since the Gamma prior is conjugate for the Poisson likelihood, we can draw directly from the posterior distribution, namely from

$$\lambda \mid (\zeta, \mathbf{W}) \sim \Gamma(a_0 + n_0, b_0 + |\mathfrak{X}|), \tag{6.8}$$

where n_0 is the number of minutiae currently labelled as random in $\mathbf{W}$, cf. also Example 5.2. In case of failure, we update the parameters of the Strauss process. To this end, as mentioned above, we have to apply the auxiliary variable method by proposing an auxiliary point pattern additionally to the new parameter.

For the update $(\boldsymbol{\theta}', \chi')$ we decompose the proposal $\tilde{q}$ as in Equation (5.6) as

$$\tilde{q}(\boldsymbol{\theta}', \mathbf{W}, \tilde{\chi}' \mid \boldsymbol{\theta}, \mathbf{W}, \tilde{\chi}) = \tilde{q}(\boldsymbol{\theta}', \mathbf{W}, \tilde{\chi}' \mid \boldsymbol{\theta}, \mathbf{W}) = g_{\beta', \gamma'}(\tilde{\chi}')\, q(\boldsymbol{\theta}', \mathbf{W} \mid \boldsymbol{\theta}, \mathbf{W}),$$

i.e. we propose at first a new parameter $\boldsymbol{\theta}' = (\lambda, \beta', \gamma') \sim q(\cdot, \mathbf{W} \mid \boldsymbol{\theta}, \mathbf{W})$ and then draw a new auxiliary point pattern $\tilde{\chi}'$ independently of the current auxiliary point pattern $\tilde{\chi}$ as a realisation of a Strauss process (with hard core) having parameter (β', γ').

To update $\boldsymbol{\theta}$ we propose a normally distributed update step for $\boldsymbol{\theta}$ in the natural parameter space (log-space), i.e. our proposal (β', γ') is log-normally distributed,

$$\begin{pmatrix} \beta' \\ \gamma' \end{pmatrix} \sim \mathcal{LN}\left(\begin{pmatrix} \log \beta \\ \log \gamma \end{pmatrix}, \begin{pmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 \\ \rho_{12}\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix} \right), \tag{6.9}$$

where the parameters σ_1, σ_2, ρ_{12} are fixed (see the end of this subsection). Denote the covariance matrix in (6.9) by Σ. Then, the proposal density q for a (β, γ)-update from $\boldsymbol{\theta} = (\lambda, \beta, \gamma)$ to $\boldsymbol{\theta}' = (\lambda, \beta', \gamma')$ is given as

$$q(\boldsymbol{\theta}', \mathbf{W} \mid \boldsymbol{\theta}, \mathbf{W}) = \frac{1}{2\pi\sqrt{\det \Sigma}} \frac{1}{\beta'\gamma'} \exp\left(-\frac{1}{2} \begin{pmatrix} \log \beta'/\beta \\ \log \gamma'/\gamma \end{pmatrix}^{\top} \Sigma^{-1} \begin{pmatrix} \log \beta'/\beta \\ \log \gamma'/\gamma \end{pmatrix} \right).$$

and hence

$$\frac{q(\boldsymbol{\theta}, \mathbf{W} \mid \boldsymbol{\theta}', \mathbf{W})}{q(\boldsymbol{\theta}', \mathbf{W} \mid \boldsymbol{\theta}, \mathbf{W})} = \frac{\beta'\gamma'}{\beta\gamma}. \tag{6.10}$$

Then, the Hastings ratio for a parameter update from $\boldsymbol{\theta} = (\lambda, \beta, \gamma)$ to $\boldsymbol{\theta}' = (\lambda, \beta', \gamma')$ is

$$\begin{aligned} H(\boldsymbol{\theta}', \mathbf{W}, \tilde{\chi}' \mid \boldsymbol{\theta}, \mathbf{W}, \tilde{\chi}) &= \frac{\pi(\boldsymbol{\theta}', \mathbf{W}, \tilde{\chi}' \mid \zeta)}{\pi(\boldsymbol{\theta}, \mathbf{W}, \tilde{\chi} \mid \zeta)} \frac{\tilde{q}(\boldsymbol{\theta}, \mathbf{W}, \tilde{\chi} \mid \boldsymbol{\theta}', \mathbf{W}, \tilde{\chi}')}{\tilde{q}(\boldsymbol{\theta}', \mathbf{W}, \tilde{\chi}' \mid \boldsymbol{\theta}, \mathbf{W}, \tilde{\chi})} \\ &= \frac{\varphi(\tilde{\chi}' \mid \mathbf{W}, \zeta)}{\varphi(\tilde{\chi} \mid \mathbf{W}, \zeta)} \frac{g_{\beta',\gamma'}(\eta)}{g_{\beta,\gamma}(\eta)} \frac{\pi(\boldsymbol{\theta}', \mathbf{W})}{\pi(\boldsymbol{\theta}, \mathbf{W})} \frac{g_{\beta,\gamma}(\tilde{\chi})}{g_{\beta',\gamma'}(\tilde{\chi}')} \frac{q(\boldsymbol{\theta}, \mathbf{W} \mid \boldsymbol{\theta}', \mathbf{W})}{q(\boldsymbol{\theta}', \mathbf{W} \mid \boldsymbol{\theta}, \mathbf{W})}, \end{aligned} \tag{6.11}$$

where the ratio of priors and proposals can be obtained from (6.7) and (6.10), respectively. Since the normalising constants of the two $g_{\beta,\gamma}$-terms and the two $g_{\beta',\gamma'}$-terms cancel, this Hastings ratio can be computed explicitly. However, in every update we have to draw a new point pattern from a Strauss process with hard core, for which we again have to run a Markov chain. This requires a considerable additional effort in each update step for (β, γ), cf. the discussion in Section 5.2.1 and the run time analysis in Section 6.5.

As mentioned in Section 5.2.1, a crucial influence on the algorithm's performance is the choice of φ which ideally should fit well to the proposal density $g_{\beta,\gamma}$. The best choice would of course be to choose $\varphi(\tilde{\chi} \mid \mathbf{W}, \zeta) = g_{\beta,\gamma}(\tilde{\chi})$, which is not feasible since then (6.11) would contain the normalising constants again, cf. Section 5.2.1. In Redenbach et al. (2015), the density of a Poisson process was used, which results in a rather poor mixing behaviour, cf. (Redenbach et al., 2015, Figure 6). We therefore choose for φ the density of another Strauss process with hard core, fixing its parameter at $\hat{\boldsymbol{\theta}}_{\text{aux}} = \hat{\boldsymbol{\theta}}_{\text{MPLE}}(\zeta, \mathbf{W})$, where $\mathbf{W}$ is the current label vector and $\hat{\boldsymbol{\theta}}_{\text{MPLE}}(\zeta, \mathbf{W})$ is the maximum pseudo-likelihood estimate (MPLE) based on the minutiae currently labelled as necessary; see e.g. (Baddeley et al., 2015, Section 13.13) and the references given there. If we knew the true $\mathbf{W}$ in advance, this would be a good initial guess for the parameters. However, in practice the true $\mathbf{W}$ is not known. We therefore adapt $\hat{\boldsymbol{\theta}}_{\text{aux}}$ iteratively during burn-in and keep it fixed for the rest of the run, so that we still obtain convergence to the desired posterior distribution, cf. also Section 7.1.

Note, that we cannot draw the auxiliary point patterns from the exact distribution, cf. Section 5.2.1. For the presented algorithm we used the MCMC-based `rmh`-procedure of the `spatstat` package (cf. also Section 4.3) to obtain the corresponding point patterns. A run time of 100,000 iterations for this inner Metropolis-Hastings algorithm turned out to be a good compromise between convergence to the right distribution and performance of the overall algorithm, cf. also Section 5.2.2 for some theoretical elaborations.

For the proposed MCMC algorithm the proposal variances σ_1, σ_2 and correlation ρ_{12} have to be determined. To this end, we considered for reasonable values of σ_1, σ_2 and $\rho_{12} = 0$[22] a pilot sample and estimate the correlation coefficient ρ_{12} as its sample correlation. The corresponding proposal variances are then adjusted such that the acceptance rate for a parameter proposal is between 20% and 30%, see (Brooks et al., 2011, Section 1.13) or the discussion in Example 5.5. For our computations we set $\sigma_1 = 0.07$, $\sigma_2 = 0.05$ and $\rho_{12} = -0.7$.

6.4.3 Update of W

When updating $\mathbf{W}$, we pick one component of $\mathbf{W}$ uniformly at random, $\mathbf{W}_i$, say, and propose to flip it to $\mathbf{W}'_i = 1 - \mathbf{W}_i$, while keeping the other components unchanged, $\mathbf{W}'_j = \mathbf{W}_j$ for $j \neq i$. Writing $\zeta = \xi \,\dot{\cup}\, \eta = \xi' \,\dot{\cup}\, \eta'$ for the partition in random and necessary minutiae before and after the proposed flip, respectively, we either have $\xi' = \xi \cup \{z_i\}$, $\eta' = \eta \setminus \{z_i\}$ if the flip of $\mathbf{W}_i$ is from 1 to 0 or $\xi' = \xi \setminus \{z_i\}$, $\eta' = \eta \cup \{z_i\}$ if the flip is from 0 to 1. Thus, the Hastings ratio is given as

$$\begin{aligned}
&H(\boldsymbol{\theta}, \mathbf{W}', \tilde{\chi} \mid \boldsymbol{\theta}, \mathbf{W}, \tilde{\chi}) \\
&\quad = \frac{\pi(\boldsymbol{\theta}, \mathbf{W}', \tilde{\chi} \mid \zeta)}{\pi(\boldsymbol{\theta}, \mathbf{W}, \tilde{\chi} \mid \zeta)} \, \frac{\tilde{q}(\boldsymbol{\theta}, \mathbf{W}, \tilde{\chi} \mid \boldsymbol{\theta}, \mathbf{W}', \tilde{\chi})}{\tilde{q}(\boldsymbol{\theta}, \mathbf{W}', \tilde{\chi} \mid \boldsymbol{\theta}, \mathbf{W}, \tilde{\chi})} \\
&\quad = \frac{\varphi(\tilde{\chi} \mid \mathbf{W}', \zeta)}{\varphi(\tilde{\chi} \mid \mathbf{W}, \zeta)} \, \frac{f_\lambda(\xi') g_{\beta,\gamma}(\eta')}{f_\lambda(\xi) g_{\beta,\gamma}(\eta)} \, \frac{\pi(\boldsymbol{\theta}, \mathbf{W}')}{\pi(\boldsymbol{\theta}, \mathbf{W})} \, \frac{q(\boldsymbol{\theta}, \mathbf{W} \mid \boldsymbol{\theta}, \mathbf{W}')}{q(\boldsymbol{\theta}, \mathbf{W}' \mid \boldsymbol{\theta}, \mathbf{W})} \qquad (6.12) \\
&\quad = \begin{cases} \dfrac{\lambda}{\beta(z_i)\gamma^{t_R(z_i \mid \eta \setminus \{z_i\})}} \, \dfrac{1 - p_{\mathbf{W}}}{p_{\mathbf{W}}} & \text{if } \mathbf{W}_i = 1 \text{ and } \mathbf{W}'_i = 0, \\ \dfrac{\beta(z_i)\gamma^{t_R(z_i \mid \eta)}}{\lambda} \, \mathbf{1}\left(d_{\min}(\eta \cup \{z_i\}) > r\right) \, \dfrac{p_{\mathbf{W}}}{1 - p_{\mathbf{W}}} & \text{if } \mathbf{W}_i = 0 \text{ and } \mathbf{W}'_i = 1. \end{cases}
\end{aligned}$$

Here, $t_R(z_i \mid \eta)$ denotes the number of R-close neighbours of z_i in η, cf. (4.30).

Remark 6.10. Note that in the last equality we used that the first factor of (6.12) is equal to 1, which is valid *after* the burn-in phase, when we do not update the parameters of the auxiliary target density φ anymore. By contrast, it seems that the same factor was erroneously omitted in (Redenbach et al., 2015, Section 3.2.2). It is not equal to 1 there, because the auxiliary target density is homogeneous Poisson with intensity depending on

[22] We are aware that the parameters β, γ are likely to be negatively correlated, so a prior with a negative value for ρ_{12} would perhaps be more reasonable. However, as we do not wish to bias the resulting estimation we choose $\rho_{12} = 0$ at this point.

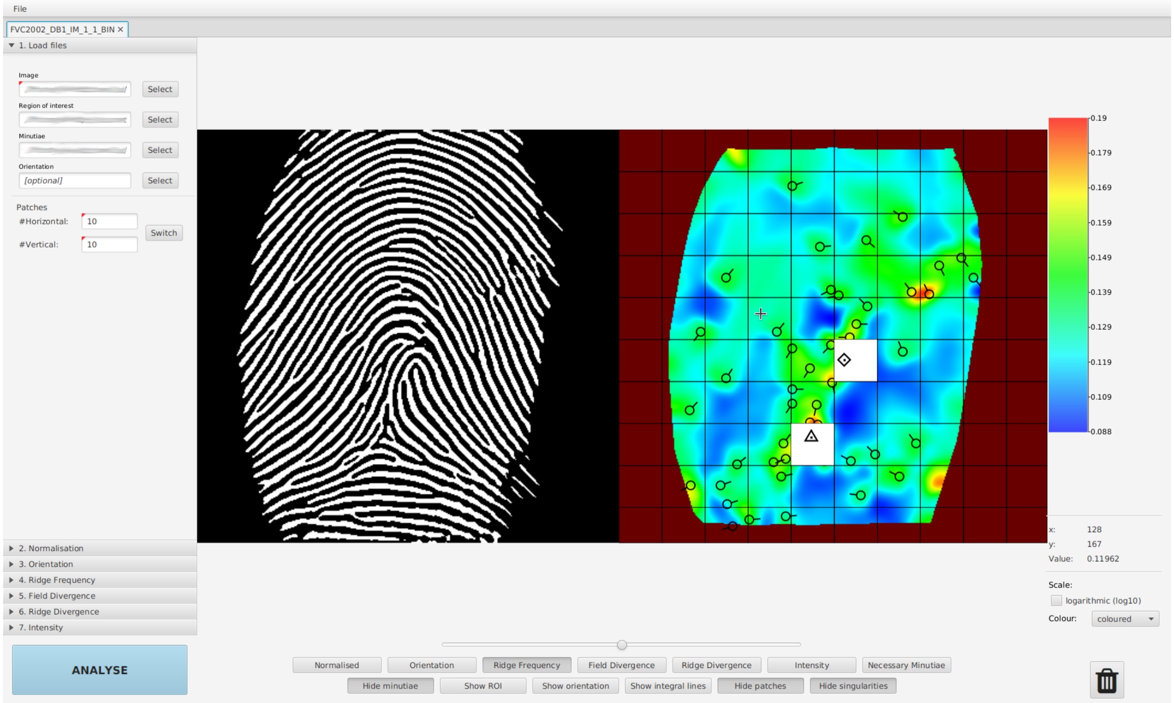
η (in our notation). This may be another part of the reason for the unfavourable mixing behaviour in Redenbach et al. (2015).

In Chapter 7 we apply the MiSeal from this section to simulated data and real minutiae patterns. In the following Section we present our implemented software and analyse the required computing time on a database of real fingerprints.

6.5 Software packages and computational costs

The algorithms presented above in Sections 6.1–6.3 are implemented in Java whereas the separation algorithm of Section 6.4 and the MPLE algorithm from Section 4.5 are implemented in R. All the software can be found on GitHub following the links https://github.com/jwieditz/MiSeal and https://github.com/jwieditz/SuperMPLE, respectively, including instructions for use, see Wieditz (2020, 2021a).

The tool to compute a fingerprint's intensity field for necessary minutiae is provided as command line interface (CLI) along with an extending graphical user interface (GUI), see Figure 6.8.



Orientation: [optional]

Patches
#Horizontal: 10
#Vertical: 10

x: 128
y: 167
Value: 0.11962

□ logarithmic (log10)
Colour: coloured

Figure 6.8: *Screenshot of the graphical user interface of the MiSeal tool from Wieditz (2020), applied to image 1_1 of database FVC2002 DB1 from Maio et al. (2002).*

As an input, it is necessary to pass an image (in .png format). In addition to this, you have the possibility to supply the region of interest, a minutiae template (just for visualisation in the GUI) or a pre-computed orientation field which is then used and the computation step from Section 6.1 is skipped.

To get an impression about the computational complexity of our algorithms, we tested them on the 20 fingerprints from Figure 3.6. We executed all algorithms on a server based on 2 × Intel® Xeon® X5660 2.80GHz CPU and 96GB RAM. For our computations we used exclusively 1 CPU 6-cores (12 threads) and 1 memory only node (NUMA) 6 × 8GB memory modules via isolated cgroup environment. The execution run times of the the Java CLI for processing an image, cf. Table 6.1, include saving times for the corresponding images. Note, that for the image processing some preprocessing steps take place at the beginning of the execution. Smoothing steps are included in all procedures. Unwrapping of the OF is included in the corresponding divergence computations. Certainly, the computational effort for processing a fingerprint image and to compute the necessary intensity numbers is with approximately 42 seconds per image too large to be included in real-time-algorithms such as minutiae matchers. Since for every considered fingerprint this has to be computed only once, this performance was sufficient for our purposes, however, we believe that a speed-up of our algorithms is certainly possible.

	run time (in s ± s.d.)
orientation field	4.4 ± 0.5
ridge frequency	13.9 ± 3.0
OF divergence	11.8 ± 2.3
RF divergence	11.6 ± 1.8
total	41.7 ± 6.0

Table 6.1: *Computational costs of computation of global features for the 20 fingerprints from Figure 3.6.*

Moreover, we computed experimentally 100,000 MCMC iterations (plus 10,000 iterations burn-in) for each fingerprint from Figure 3.6 to analyse the computational workload of MiSeal given the input parameters. The results can be seen in Table 6.2.

Note that we broke down the computational cost into the time consumed for a proposal for the considered parameters (β, γ), λ and $\mathbf{W}$, respectively, the average time for one MCMC step as well as the total time needed for a 100,000 iterations run with a 10,000 iterations burn-in. Clearly, a proposal of the Strauss parameter takes the longest since we have to run another Markov chain in order to draw an auxiliary point pattern, cf. Section 6.4.2. The

	run time (in ms $\pm$ s.d.)
(β, γ)-proposal	164,523 $\pm$ 20,984
λ-proposal	248 $\pm$ 66
$\mathbf{W}$-proposal	64 $\pm$ 54
average proposal	6,613 $\pm$ 32,425
total for 10^5 steps (in s)	727.46 $\pm$ 36.25

Table 6.2: *Computational costs of the update steps in the MiSeal.*

proposal of a label flip or a new λ proposal on the other hand only takes a few milliseconds making the procedure overall quite efficient.

For the computation of the MPLE approach from Section 4.5 we excluded three of the fingerprints due to too large minutiae components (> 10) leading to unreasonably large run times. For approximating the integral as in (4.37) we used 50×50 lattice points. The times required for the computations of the interaction graphs for point pattern and integration lattice as well as the run time of the used optimisation procedure are listed in Table 6.3.

	run time (in s $\pm$ s.d.)
graphs for point pattern	4.9 $\pm$ 6.1
graphs for integration grid	99.4 $\pm$ 87.6
optimisation	336.9 $\pm$ 117.5
total	441.2 $\pm$ 119.9

Table 6.3: *Computational costs of the MPLE approach from Section 4.5 for the 20 fingerprints from Figure 3.6.*

Evidently, the computation of an MPLE is quite costly especially with regard to the provided information of only one point estimator for the model parameters. Notable is also the high variance in the computation times, indicating that the computational cost of this approach heavily depends on the spatial configuration of the minutia pattern considered. In view of the run times for the Bayesian approach, this is yet another argument for the usage of MiSeal instead. For a comparison we can argue that the computation of the MPLE corresponds approximately to running the MiSeal for 60,000 steps after a 10,000 iterations burn-in.

Let us finally consider how MiSeal performs for the computation of the posterior distribution of $(\boldsymbol{\theta}, \mathbf{W})$ on simulated and real data. This is subject of Chapter 7.

Chapter 7

Applications

This chapter is dedicated to the application of the MiSeal from Section 6.4 for the separation of random and necessary minutiae in simulated and real minutiae patterns. Moreover, we show in a proof of concept that random minutiae indeed may carry individual information beyond the orientation field and ridge frequency, giving rise to calling them also *characteristic minutiae*.

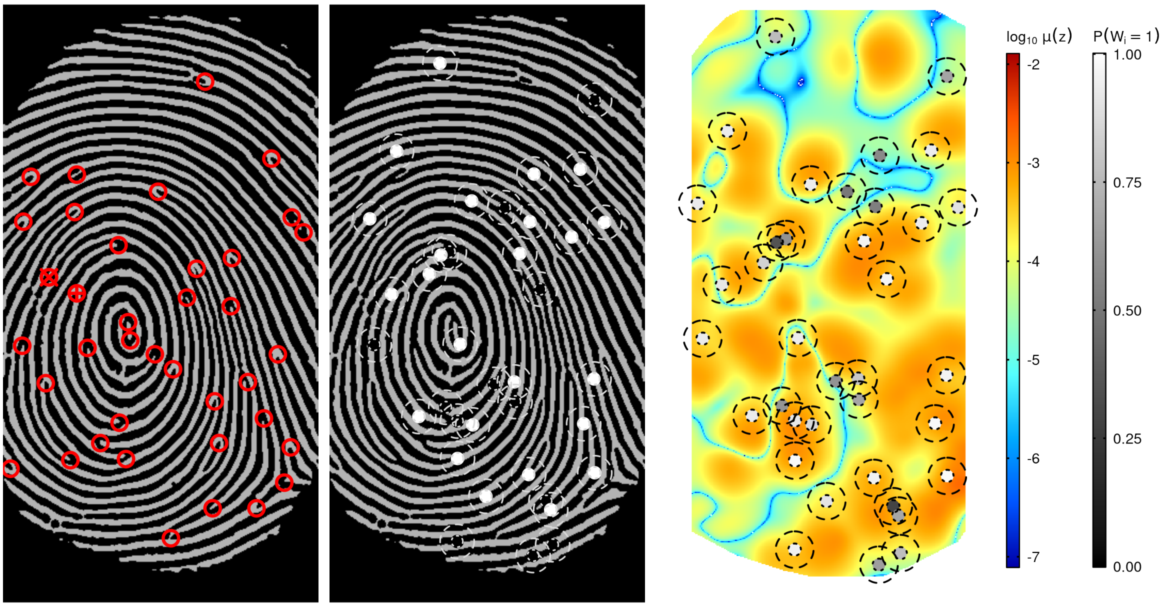
7.1 Performance of the minutiae separating algorithm

In order to benchmark MiSeal for separating random from necessary minutiae, we first simulate a test scenario close to real fingerprints with true parameters known. To this end, for each of the manually marked 20 fingerprints and the one from Figure 2.1a (cf. also Figure 7.1a), we compute a smoothed necessary minutiae intensity image $\mathfrak{M} = (\mu(i,j))$ from (3.18) as described in Sections 6.1–6.3. Then we draw "true" parameters (λ, β, γ) from the priors specified in Section 6.4.1 and simulate a sample of superimposed random and necessary minutiae following the model (4.19). Such a simulated minutiae pattern is depicted in Figure 7.1b where the true parameters (rounded to two decimals) are

$$\lambda = 1.04 \times 10^{-4}, \quad \beta = 1.17, \quad \gamma = 0.24, \quad r = 10.63, \quad R = 31.89, \tag{7.1}$$

see also Table 7.1. The original minutiae pattern of the same print is seen in Figure 7.1a and the heat map of the necessary minutiae intensity on which the simulation is based in Figure 7.1c.

We determine the parameter $\hat{\boldsymbol{\theta}}_{\text{aux}}$ for the auxiliary point pattern density (cf. Section 6.4.2) as follows: During the burn-in phase of 10,000 iterations, we compute after every 1,000



(a) (b) (c)

Figure 7.1: *(a) Minutiae pattern of the fingerprint from Figure 2.1 (imprint 7_6 from (Maio et al., 2002, DB2)). (b) Simulated random minutiae (black) and necessary minutiae (white) using the necessary minutiae intensity computed by (3.18) from the fingerprint image; the outer circles (dashed) have radius $R/2 = 15.95$, cf. (7.1). (c) Simulated minutiae with posterior probabilities in grey values from random (black) to necessary (white). The heat map gives the computed necessary minutiae intensity ($\log_{10}$-scale).*

iterations the MPLE $\hat{\boldsymbol{\theta}}^j_{\text{aux}} = (\hat{\beta}^j, \hat{\gamma}^j)$ $(j = 1, \dots, 10)$ for (β, γ) given the labels $\mathbf{W}^{(t)}$ from the respective current iteration $t = 1000j$ and use $\hat{\boldsymbol{\theta}}^j_{\text{aux}}$ for the next 1,000 iterations. We then use the component-wise mean $\hat{\boldsymbol{\theta}}_{\text{aux}} := \frac{1}{10}\sum_{j=1}^{10} \hat{\boldsymbol{\theta}}^j_{\text{aux}}$ for the rest of the entire run (alternatively, one could take the mean in the natural parameter space). This MPLE can be efficiently computed using the `ppm` function of the `R` package `spatstat` by Baddeley et al. (2015).

We consider the samples from an MCMC run of 1,000,000 iterations. Trace plots and the estimated posterior densities for the example in Figure 7.1b are depicted in Figure 7.2. The results for the Strauss parameters (β, γ) are shown in the panels 7.2b, 7.2c and 7.2e, respectively. Overall, we observe a good mixing behaviour of the Markov chain even though the parameter $\hat{\boldsymbol{\theta}}_{\text{aux}}$ for the auxiliary variable method was only determined heuristically. Especially for γ we see that the MPLE-estimate $\hat{\gamma}$ is considerably differs from the true value, cf. Table 7.1. In all 20 cases the posterior distributions of (β, γ) concentrate around the true parameters, exemplarily shown for the simulated point pattern from Figure 7.1b in Figure 7.2e. In 11 out of 20 cases β has been overestimated by the posterior mean as in

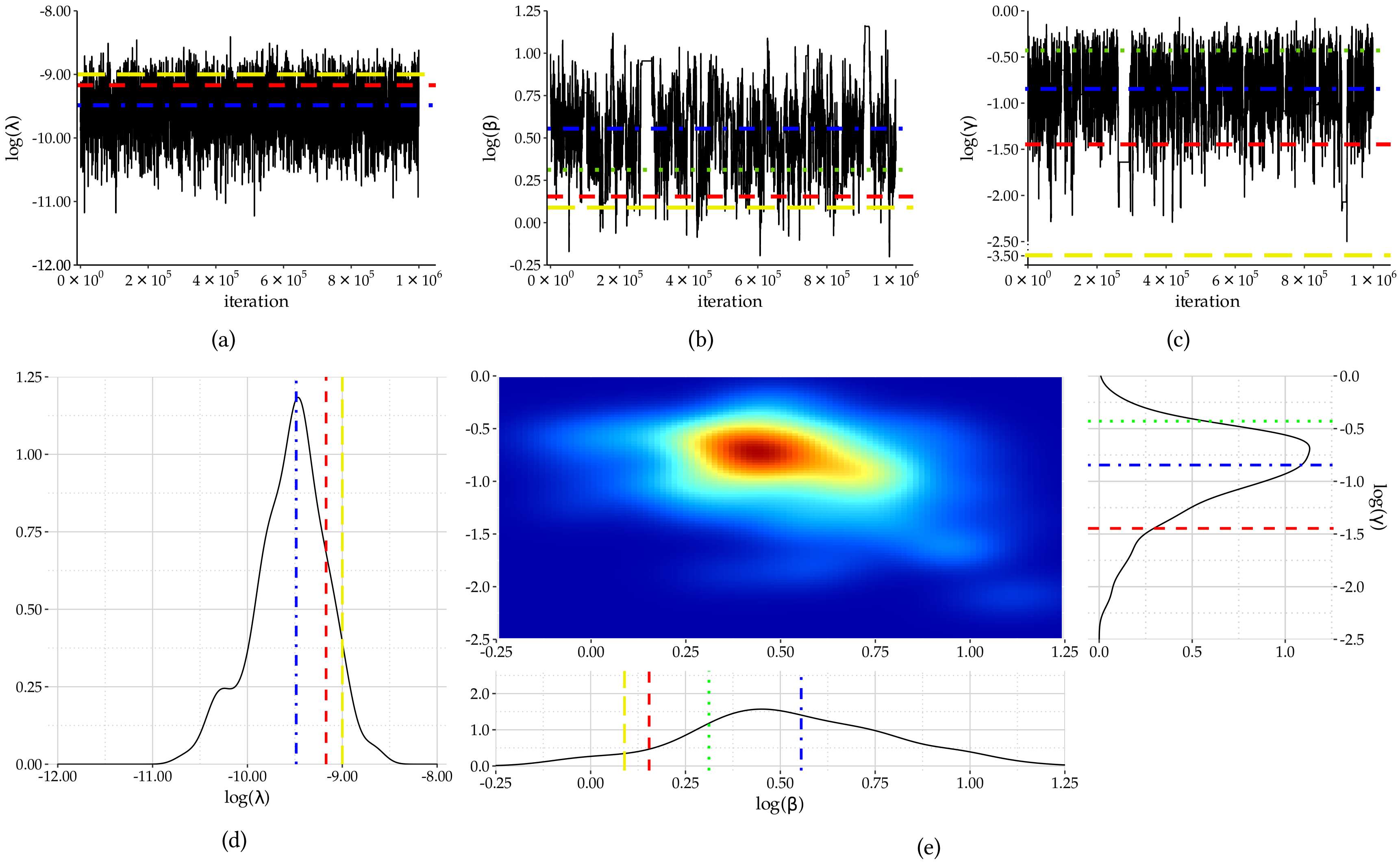


Figure 7.2: *Top: Trace plots of the parameters λ, β, γ for the simulated pattern from Figure 7.1b on the log-scale with true value (dashed red), $\hat{\boldsymbol{\theta}}_{aux}$ (dotted green), unconditional MPLE (long-dashed yellow) and posterior mean (dash-dotted blue). Bottom: Posterior distribution of λ (left) and (β, γ) and its marginals (right) on the log-scale.*

Figure 7.2b and in 12 out 20 cases γ has been overestimated as in Figure 7.2c. This suggests that our method is not substantially biased in one or the other direction.

A trace plot for λ can be seen in Figure 7.2a with corresponding estimated posterior density in Figure 7.2d. Note that the samples for λ, due to conjugacy, cf. Equation (6.8), are drawn directly from the posterior distribution (hence we have no rejections of λ proposals) and concentrate also well in the vicinity of the true value.

We use the samples from the posterior of the parameter $\boldsymbol{\theta}$ as well as the labels $\mathbf{W}$ to compute posterior means of the marginals of both $\boldsymbol{\theta}$ and $\mathbf{W}$. Moreover, we compute the MPLE $\hat{\boldsymbol{\theta}}_{\text{MPLE}}$ from Section 4.5 for the superimposed pattern unconditional on $\mathbf{W}$. The values for the posterior mean $\bar{\boldsymbol{\theta}}_{\text{post}}$, the parameter $\hat{\boldsymbol{\theta}}_{\text{aux}}$ for the auxiliary point pattern density, the unconditional MPLE $\hat{\boldsymbol{\theta}}_{\text{MPLE}}$ as well as the true value are listed in Table 7.1.

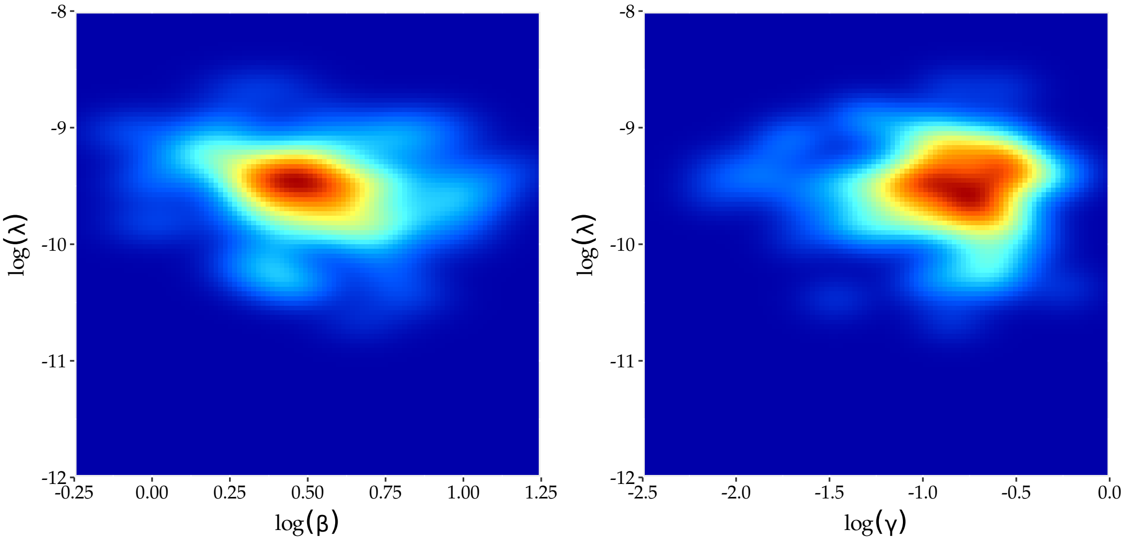
	$\hat{\lambda}$	$\hat{\beta}$	$\hat{\gamma}$
$\hat{\boldsymbol{\theta}}_{\text{aux}}$	—	1.365992	0.650895
uncond. MPLE $\hat{\boldsymbol{\theta}}_{\text{MPLE}}$	0.000123	1.093297	0.031331
posterior mean $\bar{\boldsymbol{\theta}}_{\text{post}}$	0.000076	1.741961	0.429357
± standard error	0.000002	0.039575	0.012402
true value	0.000104	1.166708	0.235489

Table 7.1: *Estimated and true parameter values for the simulated point pattern in Figure 7.1b. The interaction radii were chosen as $r = 10.63$ and $R = 31.89$.*

For the considered example from Figure 7.1b, the unconditional MPLE (indicated as yellow in Figure 7.2) estimates the parameters β and λ (possibly surprisingly) quite precisely. For the computation of the unconditional MPLE we started the optimisation procedure from different points in various regions within the parameter space we deemed reasonable. In contrast to Example 4.30 the optimisation converged to the same point in all cases considered, suggesting a more benign, possibly even globally concave shape of the pseudolikelihood in the inhomogeneous case as well as the existence of a unique global maximum.

The above mentioned precision of the MPLE for the true parameter could in general not be achieved for the database from Figure 3.6. The β-component of $\hat{\boldsymbol{\theta}}_{\text{MPLE}}$ is underestimated in 17 out of 20 cases by about 1–100% with an average of 40%, cf. Equation (7.1) for the true parameters. For the λ-component we observe an overestimation in 16 out of 20 cases (of which 15 coincide with an underestimated β-parameter) by about 7–250% with an average of 90%.

For the simulated minutiae pattern from Figure 7.1b the parameter γ is strongly underestimated and does not even lie within the estimated posterior distribution, cf. Figure 7.2c;



(a) (b)

Figure 7.3: *Heat map of the posterior density of (a) (β, λ) and (b) (γ, λ), respectively for the simulated minutiae pattern from Figure 7.1b ranging from blue (low values) to red (high values).*

note the discontinuity at the lower end of the axis. In contrast, for the simulated minutiae patterns from the fingerprints from Figure 3.6 we observe an overestimation of γ in all cases. In 19 of 20 cases the unconditional MPLE for γ is even equal to 1 (unidentifiable case). Issues in the estimation of γ have already been observed in Redenbach et al. (2015); Rajala et al. (2016) for other estimation methods. We believe that this is on the one hand due to the issue that interaction in the superimposed process can be caused both due to Strauss and Poisson points. On the other hand, the observed point patterns are relatively small in comparison to the region of interest, making it difficult to observe interaction at all (in fact, only 41% of the observed points interact with at least one other point), cf. also the discussion in Example 4.30.

Notably, the estimation of the posterior mean $\bar{\boldsymbol{\theta}}_{\text{post}}$ of $\boldsymbol{\theta}$ is quite precise (see the small standard errors in row four of Table 7.1), however, differs considerably from the true value. This seems plausible due to the low sample size of only one point pattern considered, thus considering the whole posterior distribution is worthwhile.

The univariate marginals of the posterior distribution of the label vector $\mathbf{W}$ are depicted in the top right panel of Figure 7.1c in grey values ranging from certainly random (black) to certainly necessary (white). While overall the minutiae separation is not too far from the truth, minutiae in regions of large necessary minutiae intensity not violating the Strauss hard core condition are more likely classified as necessary. In contrast, minutiae in regions of low intensity or those lying very close to one another are more likely classified as random. Notably, such candidates tend to cluster which indicates their high correlation. In a realisation, however, within such a cluster in expectation only as many minutiae will be random as the sum of marginal probabilities indicates, so that most clusters disappear (compare also Figure 7.1b with Figure 7.1c), making the random minutiae pattern Poisson-like. This corresponds exactly to the model design from Section 4.4.

We also considered the bivariate marginals of the posterior distribution of $\boldsymbol{\theta}$, see Figure 7.2e and Figure 7.3 for the simulated point pattern from Figure 7.1b. For the parameters (γ, λ) we observe a weak correlation of $\operatorname{cor}(\gamma, \lambda) = -0.0804$ suggesting that considering the univariate marginals of these two parameters might be sufficient. For the parameter pairs (β, γ) and (β, λ), however, we observe moderate negative correlations of $\operatorname{cor}(\beta, \gamma) = -0.3876$ and $\operatorname{cor}(\beta, \lambda) = -0.1462$ (more minutiae classified as random results in less minutiae classified as necessary and vice versa provided a fixed number of minutiae). Hence, the respective joint distribution seems to carry more information than just the marginals. This indicates that considering the whole distribution of the parameters is more informative than just point estimates from their marginals such as the posterior mean.

7.2 Model analysis

We finally apply MiSeal to real minutiae patterns of real fingerprints, including the one depicted in Figure 7.1a. The intensity images and minutiae patterns of the fingerprints from the database in Figure 3.6 are depicted in Figure 7.9 on p. 139 at the end of this section. We show here exemplarily the results for the minutiae pattern from Figure 7.1a. Again we obtain a good mixing behaviour for the parameter estimation as can be seen in Figure 7.5 (except for a somewhat longer period of rejections near iteration 6×10^5). Note that the β-component of the MPLE-estimate $\hat{\boldsymbol{\theta}}_{\text{aux}}$ for the auxiliary point patterns is quite large, cf. Table 7.2. This could be an indicator that during burn-in a lot of minutiae are classified as necessary due to the small number of interacting points, see Figure 7.4.

The posterior distribution of the parameters overall resembles the results from the simulation study. In fact, the values for the posterior means of both, the simulated and real minutiae pattern from Figure 7.1b and 7.1a, respectively, are very similar (notably, their estimation is again very precise, see the small standard error). This suggests that both real and simulated point patterns carry the same information in terms of the posterior parameter distribution and could be understood as an argument in favour of our model.

	$\hat{\lambda}$	$\hat{\beta}$	$\hat{\gamma}$
$\hat{\boldsymbol{\theta}}_{\text{aux}}$	—	3.484271	0.286867
uncond. MPLE $\hat{\boldsymbol{\theta}}_{\text{MPLE}}$	0.000036	2.049738	0.108702
posterior mean $\bar{\boldsymbol{\theta}}_{\text{post}}$	0.000048	1.744231	0.373191
± standard error	0.000002	0.030755	0.010469

Table 7.2: *Estimated parameter values for the real minutiae pattern from Figure 7.1a. As for the simulated point pattern from Section 7.2 interaction radii were chosen as* $r = 10.63$ *and* $R = 31.89$.

Whereas the estimated values for both λ and β for the posterior mean and the unconditional MPLE are relatively close to each other, we have a large discrepancy between the estimated values of γ. The estimated unconditional MPLE for γ is again very small and lies in a region where the posterior distribution has almost no probability mass. This problem could probably be reduced to the small number of interacting points, see also the discussion in Section 7.1. For the fingerprints in Figure 7.9 we do not observe this behaviour. For these examples, the interaction parameter γ is estimated in all cases as to be 1 by the MPLE. However, in almost all considered cases the number of interacting pairs of points is also considerably larger, cf. Figure 7.9.

Again, we considered the bivariate posterior distributions, cf. Figure 7.5d and Figure 7.6 for the results from fingerprint 7.1a. As for the simulated patterns we observe almost no (empirical) correlation between the parameters (γ, λ), namely $\operatorname{cor}(\gamma, \lambda) = 0.0024$ whereas there is a considerable negative correlation between the parameter pairs (β, γ) and (β, λ), i.e. $\operatorname{cor}(\beta, \gamma) = -0.3816$ and $\operatorname{cor}(\beta, \lambda) = -0.1355$. This suggests that considering the joint parameter distribution is particularly worthwhile in order to capture dependencies between the parameters.

The marginal posterior probabilities for the real minutiae pattern from Figure 7.1a are depicted in Figure 7.4. The discrimination

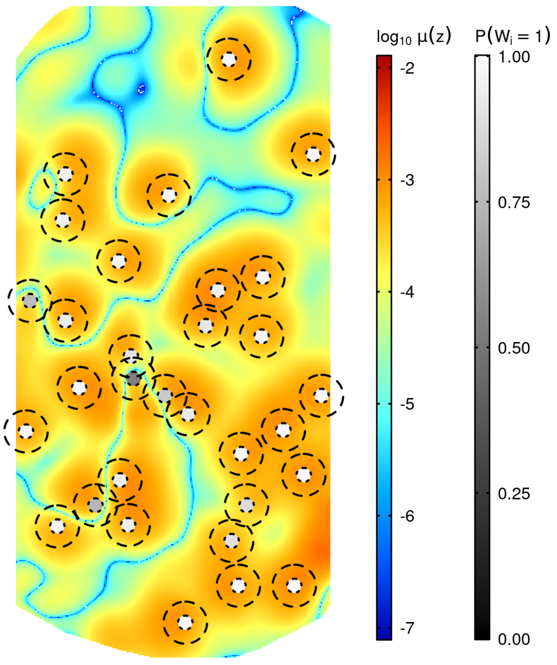


Figure 7.4: *Real minutiae pattern from Figure 7.1a with posterior probabilities in grey values from certainly random (black) to certainly necessary (white). The heat map gives the computed necessary minutiae intensity.*

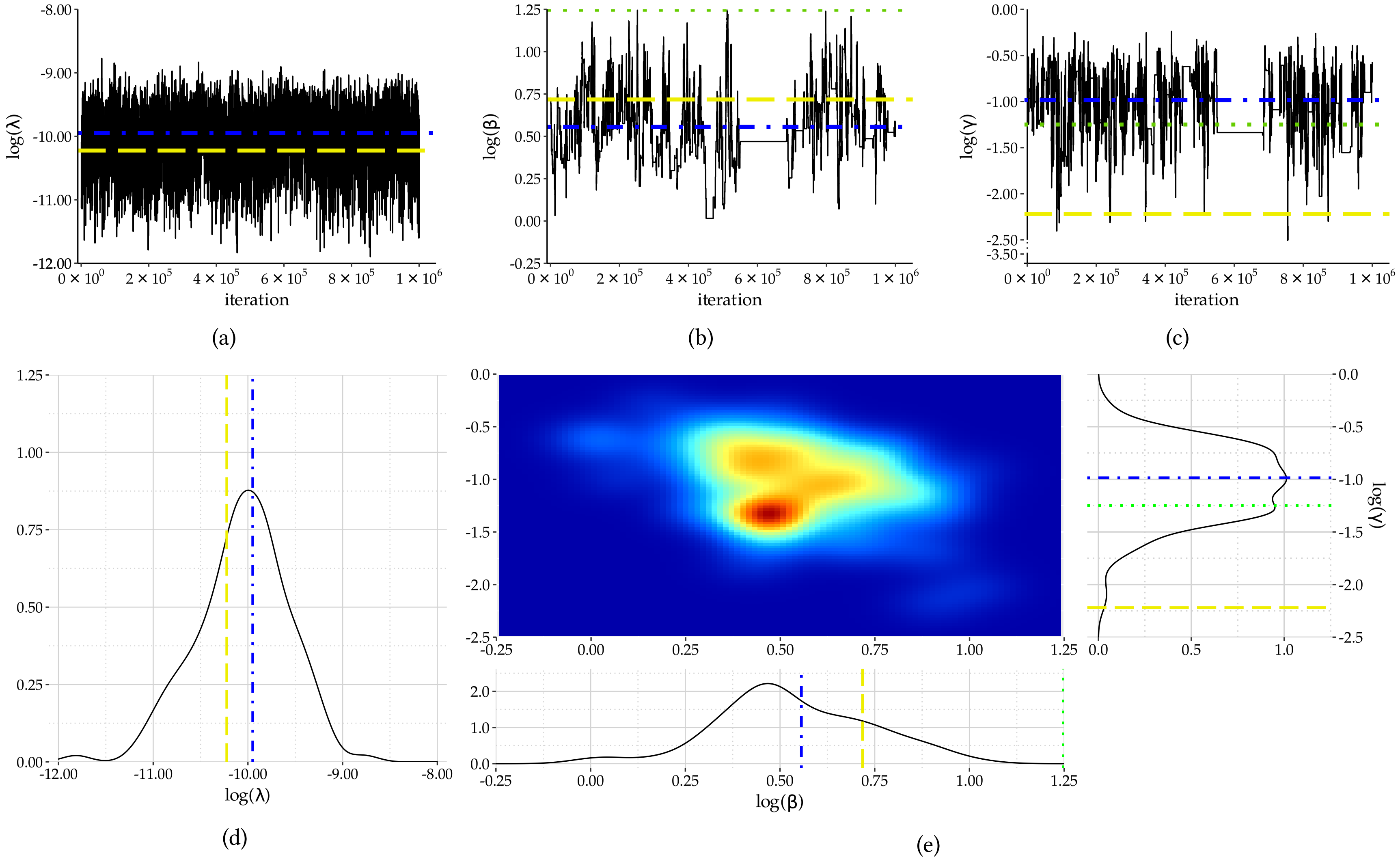


Figure 7.5: *Top: Trace plots of the parameters λ, β, γ for the real minutiae pattern from Figure 7.1a on the log-scale with $\hat{\boldsymbol{\theta}}_{aux}$ (dotted green), unconditional MPLE (long-dashed yellow) and posterior mean (dash-dotted blue). Bottom: Posterior distribution of λ (left) and (left) (β, γ) and its marginals (right) on the log-scale.*

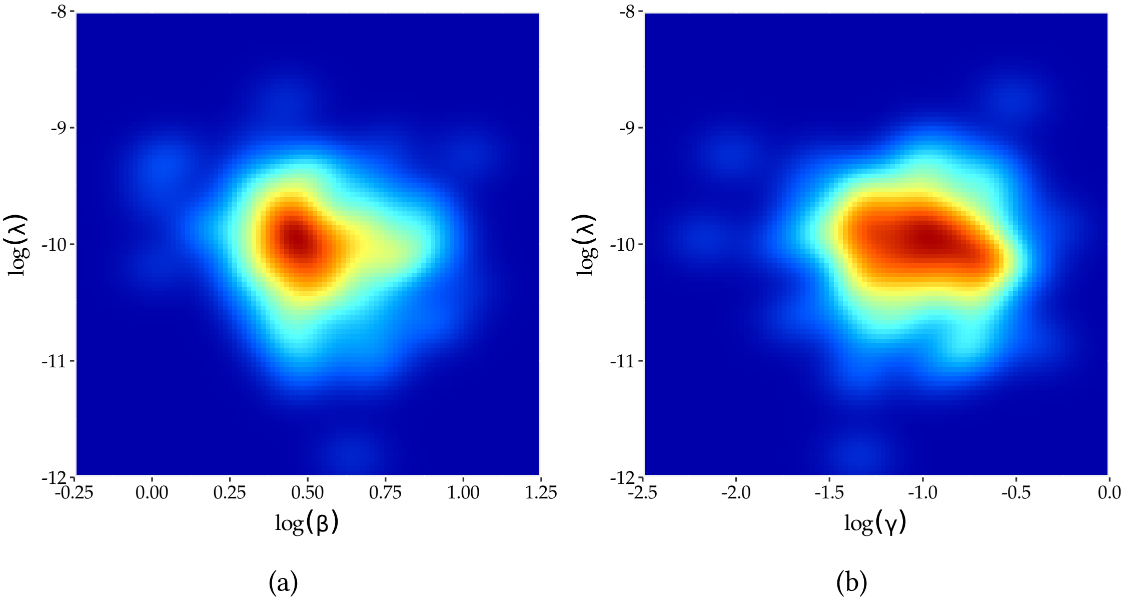


Figure 7.6: *Heat map of the posterior density of (a) (β, λ) and (b) (γ, λ), respectively for the real minutiae pattern from Figure 7.1a ranging from blue (low values) to red (high values).*

between random and necessary minutiae in this image is not as obvious as in Figure 7.1c. However, we can observe the same behaviour as mentioned at the end of Section 7.1, namely minutiae which do interact with other minutiae or which are located in regions of low intensity tend to have a lower marginal posterior probability of being necessary.

Considering only the marginal posterior probabilities might in general, however, not be sufficient as we observe that the components of $\mathbf{W}$ are not independent. Considering minutiae pairs within interaction distance R from each other, Fisher's exact test always rejects the hypothesis of independence at the 1% level.

As an example, Table 7.3 gives the sampled posterior frequencies of the different label pairs for the two minutiae marked on Figure 7.1a with $\oplus$ and $\otimes$ (halfway north-west from the whorl). In parentheses are the expected frequencies under independence. Notably, we thinned the run by 100 (a little more than the integrated auto-correlation time), resulting in approximately independent subsamples. We partitioned these subsamples in 100 batches and computed batch-wise Matthews (1998) correlation coefficients to obtain a Monte Carlo estimate of the dependence between $\mathbf{W}_\oplus$ and $\mathbf{W}_\otimes$. This yields a correlation of −0.093 on average (with standard error of 0.007), suggesting negative correlation. Moreover, we compute the Kullback-Leibler divergence of the joint distribution of $(\mathbf{W}_\oplus, \mathbf{W}_\otimes)$ to the closest independent distribution resulting in a Kullback-Leibler divergence of 0.0069 (using base-2 logarithms).

$\mathbf{W}_\otimes$ $\mathbf{W}_\oplus$	0	1	total
0	73 (172)	636 (537)	709
1	2,349 (2,250)	6,942 (7,041)	9,291
total	2,422	7,578	10,000

Table 7.3: *Contingency table of two selected components of* $\mathbf{W}$ *(labelled* $\oplus$ *and* $\otimes$ *in Figure 7.1a, halfway north-west from the whorl) whose minutiae lie within interaction distance* R*, with frequencies under hypothesis of independence in parentheses.*

Judging from this analysis, it seems important to consider the whole distribution of $\mathbf{W}$ provided by Algorithm 6.8, rather than only the marginals obtained by the method in Rajala et al. (2016).

Finally, to obtain an impression of the overall model fit we applied MiSeal on the data base from Figure 3.6. Figure 7.7 depicts the estimated posterior means for $\boldsymbol{\theta}$. Assuming that the estimated posterior mean is approximately normally distributed around the true posterior mean, we obtain a quantification of uncertainty using the shown Gaussian 95% confidence ellipses (in the colour of the corresponding posterior mean for λ) for the parameters (β, γ) (top), (β, λ) (middle) and (γ, λ) (bottom), respectively, for the 20 considered fingerprints. Note that for reasons of clarity and comprehensibility we did not include an indicator for the variance of λ in the upper plot of Figure 7.7.

Obviously, the posterior means are rather spread out in the parameter space, however the corresponding confidence ellipses largely overlap and suggest a considerable variability on the estimated parameters. Indeed, due to the individuality of the minutiae patterns in terms of size and interaction structure, one would expect varying values of β and γ for the corresponding fingerprints whose fitted values fit these quantities. Moreover, the parameters (λ, β, γ) could also depend on unobserved quantities, such as manner of imprinting on the acquisition medium, quality or resolution of the image. Overall, Figure 7.7 suggests that it might be worthwhile to consider a model for the distribution of $\boldsymbol{\theta}$ using universal parameters for whole data base of fingerprints yielding potential for future research.

Furthermore, we estimated the PCF, cf. Example 4.13, of the fitted process using the posterior mean as parameter for each of the data considered based on 100 draws from the model, cf. Figure 7.7. The resulting PCFs are pooled and depicted as a red dashed curve in Figure 4.5, see also Figure 7.8. We observe that the PCFs of the fitted processes run mostly within the pointwise confidence band, however it seems clear that the R chosen based on a pilot study is somewhat too small. The preselection of the interaction radii h and R leaves

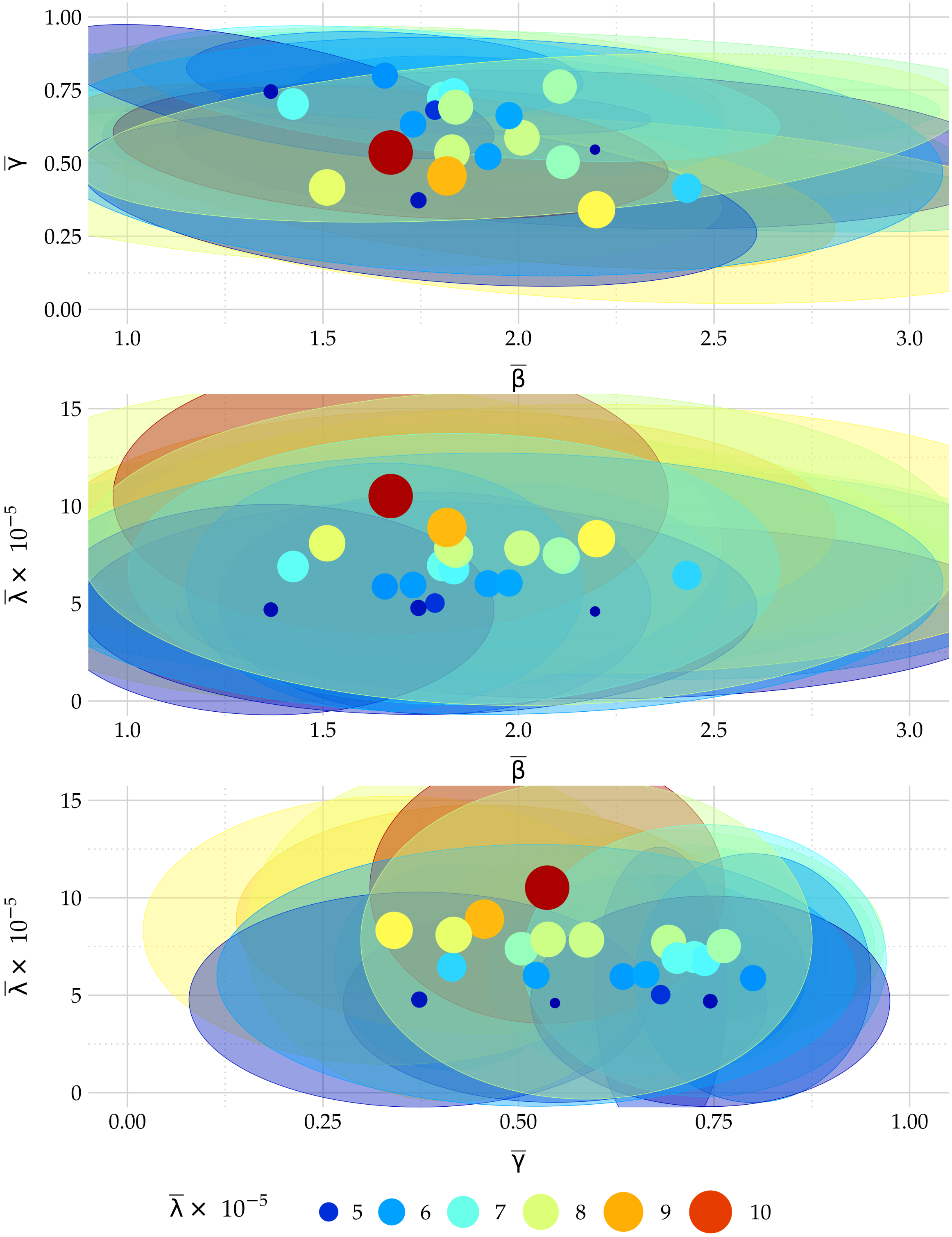


Figure 7.7: *Posterior means $\bar{\boldsymbol{\theta}} = (\bar{\lambda}, \bar{\beta}, \bar{\gamma})$ of $\boldsymbol{\theta}$ of the 20 example fingerprints from Section 4.4 and the one from Figure 7.1a in the (β, γ) (top) (β, λ) (middle) and (γ, λ) plane (bottom). The value of $\bar{\lambda}$ is indicated as the size and colour of the bubble. Around each bubble a Gaussian 95% confidence ellipse in the corresponding two dimensional slice of the parameter space is shown.*

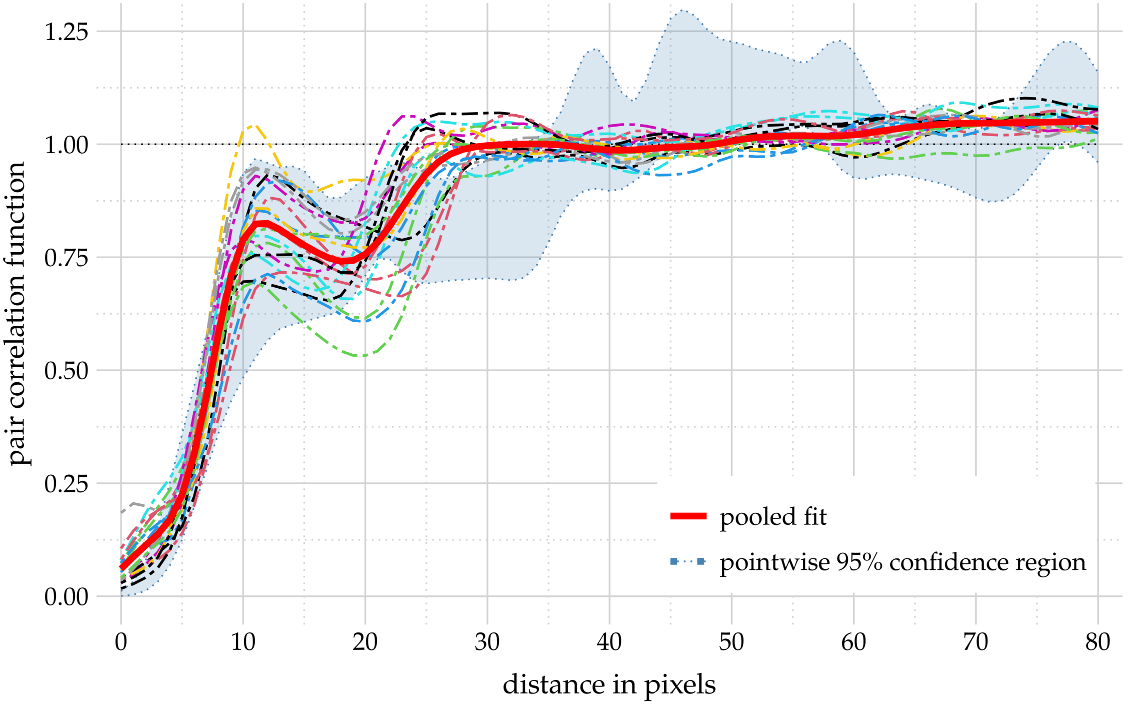


Figure 7.8: *PCFs of the fitted models (dashed) using the posterior mean and their pooled PCF (red). The confidence region of the data from Figure 4.5 is shaded in blue.*

potential for future research, cf. also the discussion in Section 4.4. It may be desirable to adjust R using more sophisticated characteristics of the individual fingerprint than just the average inter-ridge distance. At small distances, the PCFs tend to be slightly above the confidence band due to Poisson minutiae forming close pairs with other minutiae. This might not be very relevant in practice, and in any case it is hard to avoid this behaviour without making the model mathematically much more difficult. As a final remark, we note that a global way of assessing the model fit might be more desirable. One approach in this direction is given by the global envelope-based goodness-of-fit tests in Myllymäki et al. (2017) and Myllymäki & Mrkvička (2019), which have been developed for a single observation of a point pattern. An extension of the approach provided there to replicated point patterns is beyond the scope of this thesis.

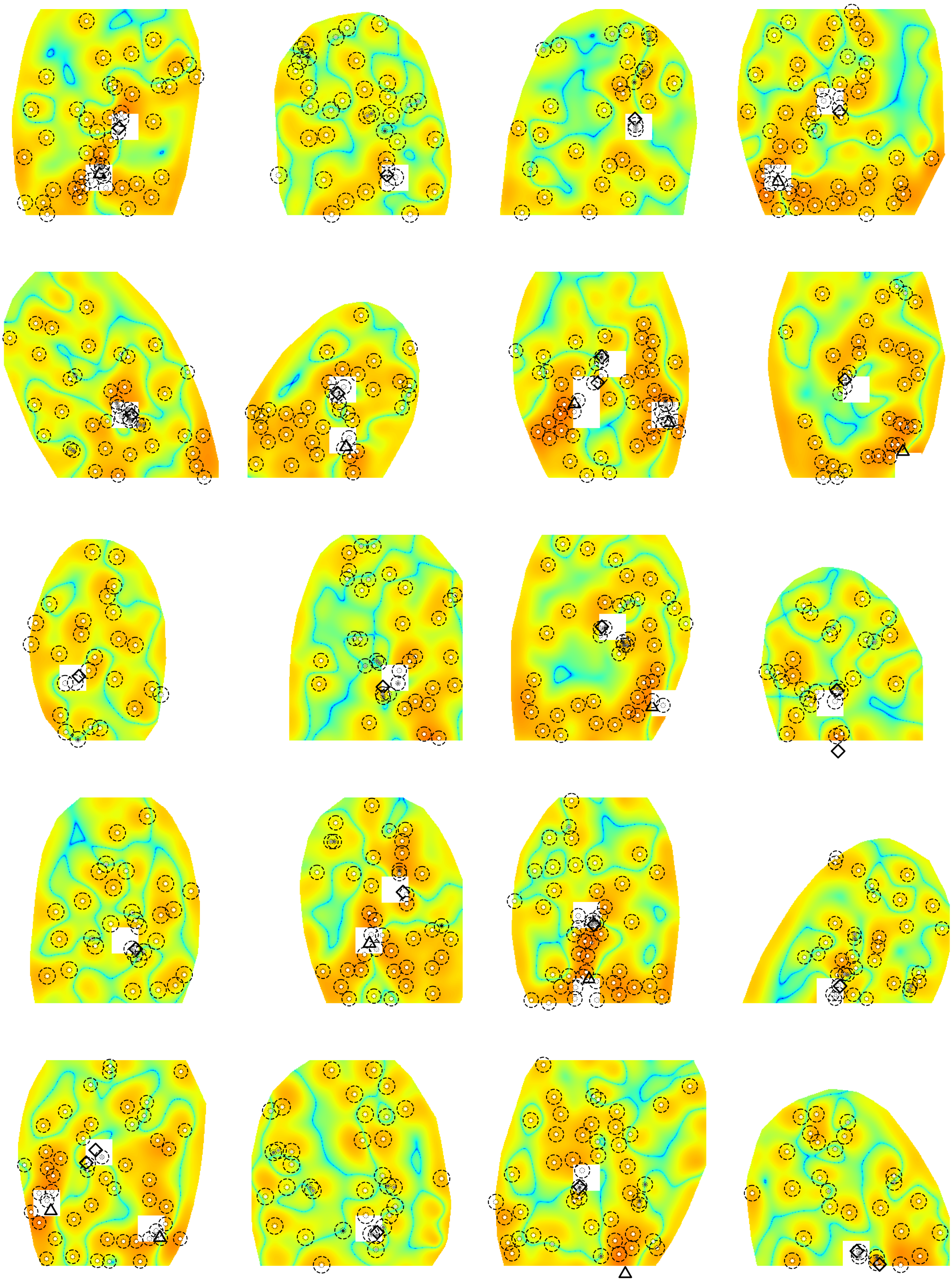


Figure 7.9: *Computed minutiae intensity images $\mathfrak{M}$ as heat maps ($\log_{10}$-scale) ranging from blue (low values) to red (high values). Manually marked minutiae as dots with corresponding hard-core (dotted) and interaction radius (dashed) as in Section 4.4. Corresponding marginal posterior probability of being necessary in grey values from certainly random (black) to certainly necessary (white). Data from (Maio et al., 2002, DB1) (row-wise), see also Figure 3.6 on p. 34. Estimated singularity positions are marked with ◇ (core) and △ (delta), respectively.*

7.3 Investigation for characteristicness

To conclude this chapter, we consider an application of the information resulting from the posterior distribution of the labels $\mathbf{W}$ as computed in Section 7.2 in fingerprint matching. To this end, we consider the two fingerprints from Newman (1930), cf. Figure 1.1 on p. 2. At first glance, they appear very similar based on their OFs and RFs, but actually stem from two different persons. We claim that excluding the *random minutiae* from the minutiae patterns makes them look more alike (w.r.t. some similarity criterion) in comparison to excluding the same amount of (any) minutiae *at random*.

As before, we enhance the images using the algorithm by Thai et al. (2016) and extract the minutiae manually. Then, we approximate the posterior distributions π_1, π_2 of the label vectors $\mathbf{W}$ using our MiSeal (Section 6.4). Their marginal probabilities are depicted in Figure 7.10 on a grey-scale ranging from white (certainly necessary) to black (certainly random). In particular the northeast part of the right print in Figure 7.10 contains candidates for random minutiae (from grey to black), that are not found on the left print. Note that identifying random minutiae just from the marginal posterior distributions, however, is not always possible due to the correlation between the components, cf. the discussion at the end of Section 7.1 and the analysis around Table 7.3.

For comparison of two minutiae patterns we use the Minutiae Cylinder Code (MCC) matching algorithm obtained from Cappelli et al. (2010b), which is publicly available. The MCC compares two given minutiae templates $\zeta^{(1)}, \zeta^{(2)}$ exploiting local information, i.e. spatial and directional similarity of minutiae and their neighbourhood (cylinders) and combines the most similar cylinders to a global score $S(\zeta^{(1)}, \zeta^{(2)}) \in [0, 1]$ where 1 means very similar and 0 means very different.

In order to assess *characteristicness* of random minutiae, as mentioned above, we investigate whether deleting random minutiae leads to more similar fingerprints than deleting the same number of arbitrary minutiae. For this, we repeat the following procedure 1,000 times:

(1) Draw a sample $\mathbf{W}^{(i)} \sim \pi_i$ from the posterior π_i of the labels and let $r^{(i)}$ be the number of random minutiae in the minutiae template $\zeta^{(i)}$, $i = 1, 2$.

(2) Delete from the minutiae template $\zeta^{(i)}$ the minutiae labelled as random under $\mathbf{W}^{(i)}$ to obtain a new template $\zeta^{(i,n)}$ containing only the necessary minutiae, $i = 1, 2$.

(3) Draw uniformly at random $r^{(i)}$ minutiae from $\zeta^{(i)}$ and delete them from $\zeta^{(i)}$ to obtain a new template $\zeta^{(i,r)}$ having the same number of minutiae as $\zeta^{(i,n)}$, $i = 1, 2$.

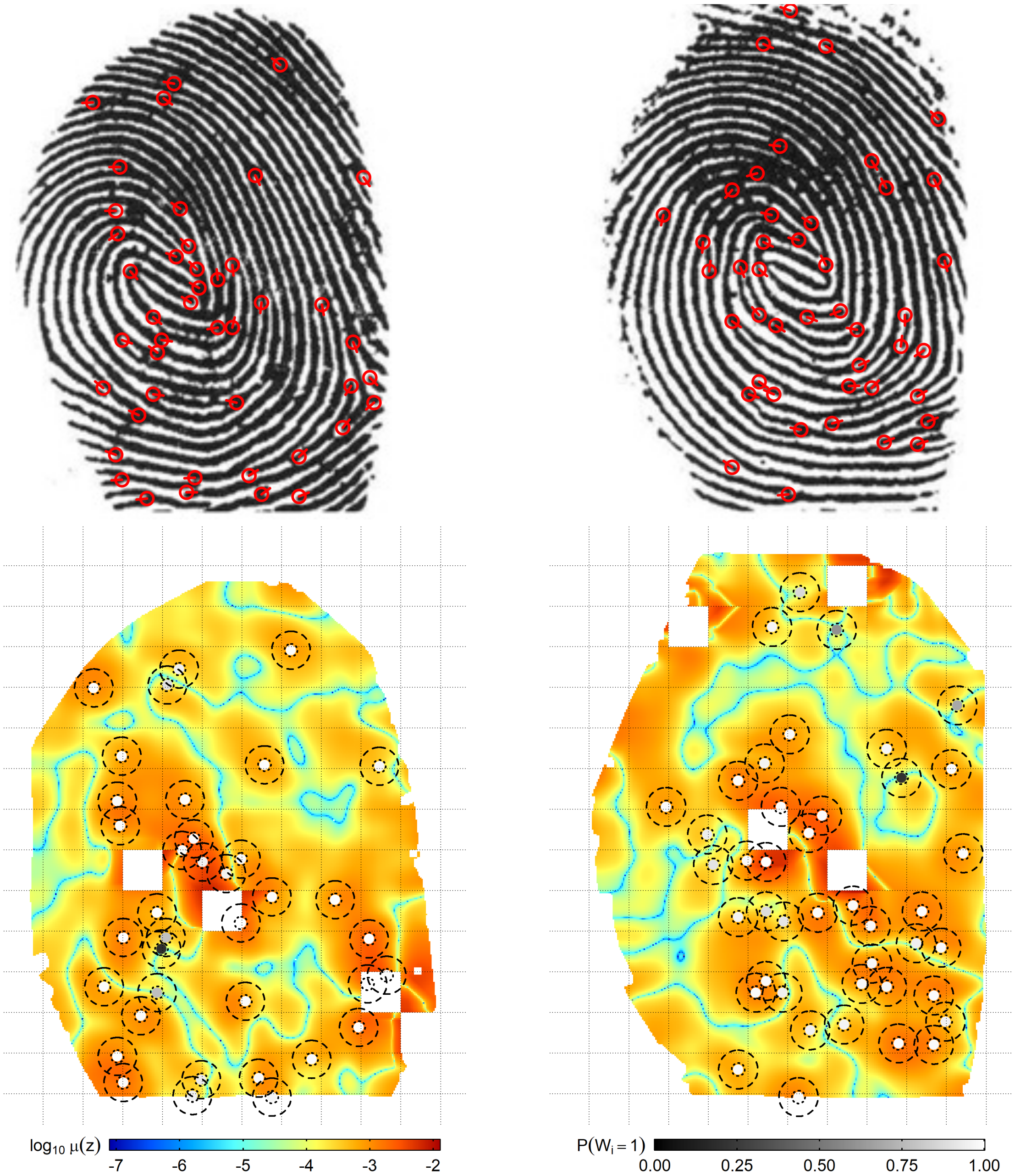


Figure 7.10: *Top row: Fingerprints from a pair of monozygotic twins (Newman, 1930, 14a and 14b) with almost identical orientation and ridge frequency fields and manually marked minutiae, see also Figure 1.1. Bottom row: The necessary minutiae intensities as heat maps on the $\log_{10}$-scale from blue (low) to red (high). Marginal posterior probabilities are indicated on a greyscale from black (certainly random) to white (certainly necessary). Hard core distance r as dotted circle. The dashed circles have radius $R/2$, i.e. intersecting circles indicate minutiae within interaction distance R. Note that the intensity is computed patch-wise (lattice indicated as dotted lines) and patches containing detected singularities were excluded.*

(4) Compute the matching scores $S^{(n)} := S(\zeta^{(1,n)}, \zeta^{(2,n)})$ and $S^{(r)} := S(\zeta^{(1,r)}, \zeta^{(2,r)})$ using the MCC.

We then compute the differences between these 1,000 pairs of matching scores. Note that local clusters of minutiae, which make a major contribution to the MCC matching score, are often dissolved by our deletion scheme, leading to scores on a very small scale. We therefore consider the relative score differences in Figure 7.11. We obtain a Monte Carlo estimate of 93.6% (with standard error of 0.8%) of the cases in which the matching score after deletion of random minutiae is larger than the score after deleting the same number of minutiae at random. The Monte Carlo estimate for the relative difference of scores yields a 23.7% improvement (with standard error of 0.6%).

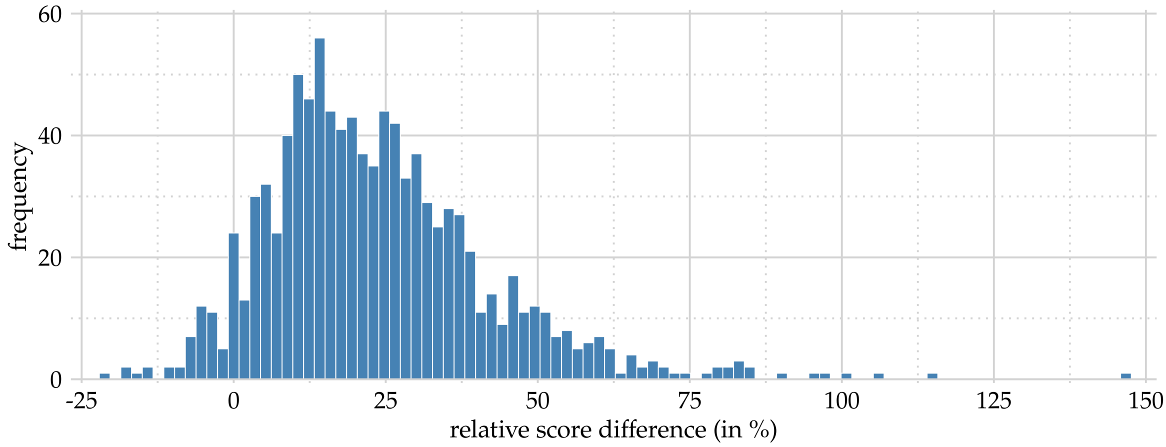


Figure 7.11: *Histogram of relative score differences* $\left(S^{(n)} - S^{(r)}\right)/S^{(r)}$ *(in %). Overall 93.6% of these distances are positive and the average difference is 23.7%.*

Hence, we may conclude that the two different fingerprints become more similar to one another after deleting the random minutiae in comparison to just randomly deleting minutiae. This preliminary proof of concept suggests that for fingerprints with similar OFs and RFs the information encoded in random minutiae is *characteristic* to distinguish them from one another.

An interesting application of the posterior distribution of $\mathbf{W}$ would be to include this information into matching algorithms to find out whether we can increase matching rates also on larger data bases. Unfortunately, most matching algorithms are proprietary software making it impossible to include data beyond the minutiae templates. Even though the above mentioned MCC is publicly available it contains closed-source elements. The computation of the MCC similarity score is based on a similarity matrix which can be read and modified (as described above). The processing of this similarity matrix, however, is a black box, making it difficult to analyse the influence of changes in the similarity matrix to the final score. An independent implementation of a matching algorithm is very difficult (cf. (Maltoni et al., 2009, Chapter 4)) and was out of scope of this thesis. Possible applications of the posterior distributions of parameters and labels in fingerprint recognition software are hence left for future research.

Chapter 8

Discussion and outlook

It is now time to pull over to the side nearly at the end of our journey, to re-examine our goals and to look how far we have come: We provide parameter inference methods in a superposition model of point processes. Moreover, we allow for point pattern separation in a mathematical model for the occurrence of minutiae based on orientation field (OF) and ridge frequency (RF). We point out benefits and limitations of our work as well as perspectives in the following.

Minutiae extraction using intensity images In Chapter 3 we introduced a model which provides a formula for predicting locally the number of necessary minutiae determined by OF and RF of a fingerprint. To the best of the author's knowledge, this is the first mathematical model including OF and RF information to do so. A statistical analysis, based on 20 high quality images (cf. Figure 3.6), revealed that fingerprints feature additional *random* minutiae. To this end, our study requires a reliable extraction of minutiae and fingerprint features. Minutiae detection algorithms experience problems in finding *all* minutiae within a fingerprint's region of interest, i.e. on the one hand not extracting spurious minutiae and, on the other hand detecting also rare-type minutiae (e.g. short ridges or lakes) which are hard to distinguish from noise. Hence, an investigation of large data bases is still left for future research. Moreover, it would be certainly interesting to investigate the influence of the scale (i.e. the degree of smoothing) on which we consider a fingerprint's minutiae intensity field to the amount of random minutiae. Here, especially the question arises whether there are no random minutiae (and hence the minutiae-divergence formula being exact) on an arbitrarily small scale. Since minutiae are only visible on a scale large enough to observe the ridge lines, this is in practice hard to verify. One possible approach in this direction could be to consider fingerprint *skeletons* (i.e. one-pixel ridge structures) instead of the raw images to increase the resolution. A corresponding result could be applied to improve minutiae detection methods using the intensity image similarly to Nikodémusz-Székely & Székely (1993). Moreover, the dependence of the amount of random minutiae depending on the considered scale (i.e. different patch sizes) could be worth investigating in this direction. To this end, depending on the considered scale we have to adjust the used bandwidth parameters accordingly, which requires a considerable additional effort and was hence left for future research.

By considering the pair correlation function for the 20 fingerprints from Figure 3.6, we have concluded that it is reasonable to model the necessary minutiae by an inhomogeneous Strauss process with hard core whose activation is given by the necessary minutiae intensity, while the additional random minutiae can be modelled by a homogeneous Poisson point process. The aim of identifying necessary and random minutiae for a given minutiae pattern leads to a separation problem for the superposition of point processes which is considered from two view points.

MPLE for superpositions with multi-scale interaction processes Following a frequentist approach we lay the foundation for parametric inference in independent superpositions of point processes in stating the corresponding density. To the best of the author's knowledge, point process separation problems have not been considered from this theoretical side in the literature so far. For the model considered in Section 4.4 we state identifiability results and an inference method using maximum pseudolikelihood estimation for the model parameters and, moreover, provide corresponding software. The MPLE method performed reasonably well on simulated and real data for the model parameters β and λ but exhibited weaker performance on the interaction parameter γ. Moreover, we discussed theoretical issues in a homogeneous setting and the drawback that we do not obtain information about the allocation of the points to the two process in the frequentist setting. An extension of the MPLE method to a more general superposition model, e.g. for superpositions of multi-scale Strauss processes and Poisson processes, is (also computationally) easily conceivable by adapting the interaction graphs presented in Section 4.5 accordingly. Note, that for each step in the interaction function computationally a corresponding interaction graph is required.

More general, the incorporation of a continuous interaction potential, i.e. the interaction parameter γ depending continuously on the distance R between pairs of points, e.g. for a Lennard-Jones process, is conceivable. As for the Lennard-Jones process the interaction between pairs of points vanishes quickly with increasing distance. In practice the maximal interaction distance is cut at a certain distance R. Using this distance as interaction distance R in the MPLE approach from Section 4.5 and considering a weighted graph[23] where the edge weights $w(e)$ of an edge $e = uv$ consist of the value $w(e) = \exp(V(\|u - v\|))$ where V is the Lennard-Jones potential (instead of $w(e) = \gamma$ for a Strauss process), lead to a possible extension of the results of Section 4.5 to models of infinite interaction range. Computationally, we substitute $\gamma^{|E_R(\eta)|}$ in Theorem 4.28 by $\prod_{e \in E_R(\eta)} w(e)$. Note, however, that depending on the application and the interaction range R the corresponding graph (and especially its components) can be very large and hence the computation of the conditional intensity can be extremely time-consuming.

Separation beyond fingerprint data Following a Bayesian mindset, we applied an MCMC algorithm for exploring the distribution of necessary and random minutiae of a given fingerprint as well as, simultaneously, the model parameters. The proposed Mi-Seal (Chapter 5) is based on the work of Redenbach et al. (2015) but provides significant improvements in terms of mixing times and does not need assumptions on the independence of the components of the label vector as in Rajala et al. (2016) which turned out to be unrealistic in our applications. We provide corresponding software containing a tool to compute all ingredients of MiSeal as the necessary minutiae intensity image as well as the MCMC algorithm whose parameters are tuned for the considered data. The software package is ready-to-use also for applications beyond fingerprint recognition allowing for an easy means for separation and parameter estimation, e.g. in noise identification in patterns of air bubbles in ice cores, cf. Redenbach et al. (2015) or mine field detection, cf. Walsh & Raftery (2002). Data analysis and application of the software to other data sets, however, was beyond the scope of this thesis. Nevertheless a comparison to Redenbach et al. (2015), cf. Section 6.4.2, promises an improved performance also on other datasets.

Perfect simulation A crucial ingredient for good mixing seems to be a good choice of the marginal distribution for the auxiliary point pattern. The associated parameter $\hat{\boldsymbol{\theta}}_{\text{aux}}$ has thus to be chosen appropriately which we achieve by repeated estimation during the burn-in phase. As a future improvement, updating $\hat{\boldsymbol{\theta}}_{\text{aux}}$ can also be performed during the entire run if we let the adjustments diminish as the iteration number increases and adapt the Hastings ratio for the $\mathbf{W}$-update appropriately, see e.g. Roberts & Rosenthal (2007). As we can only draw auxiliary point pattern samples from an inhomogeneous Strauss process via MCMC, convergence of MiSeal to the true posterior distribution is not ensured. The implementation of an efficient perfect simulation scheme for inhomogeneous Strauss processes

[23] i.e. every edge $e \in E(G)$ in the graph G gets assigned a weight $w(e) \in \mathbb{R}$, cf. Section A.3.

yields potential for future research filling a gap in an area where otherwise double MCMC algorithms would have been employed (cf. Section 5.2.2). For perfect simulation, dominated coupling from the past (CFTP) algorithms can be applied using an inhomogeneous Poisson process with the desired trend μ as intensity as dominating process, see Kendall & Møller (2000). However, in regions of intense interaction, i.e. when the interaction radius relatively to the underlying intensity becomes too large (e.g. close to singularities) these CFTP algorithms are expected to converge very slowly, cf. Kendall & Møller (1999), making them in practice often very inefficient to use.

Improving matching rates In an application to two fingerprints from two monozygotic twins exhibiting similar OF and RF we found out that excluding random minutiae yields a highly significant improvement of the similarity score as compared to excluding arbitrary minutiae. This suggests that the random minutiae carry characteristic information of fingerprint individuality going beyond OFs and RFs, which is why we refer to them as *characteristic minutiae*. The extent to which this information can effectively be used to distinguish different fingerprints with similar OFs is the subject of current and future research. A possible application could be in robust automatic minutiae detection by applying MiSeal to an automatically extracted minutiae point pattern (possibly containing spurious minutiae). As false minutiae are more likely to be classified as random, a filtering scheme focussing only on the random minutiae could facilitate the detection of spurious minutiae and hence improve the performance and robustness of minutiae detection algorithms. Moreover, the inclusion of information about characteristicness in matching algorithms (improving matching rates by also comparing the degree of characteristicness) yields an interesting perspective.

Additionally, various parameters of our MiSeal, for instance the smoothing of the necessary minutiae intensity, can be more finely tuned based on larger data sets. Since smoothing can be thought of as the scale on which fingerprints are considered, the number of random and necessary minutiae may vary accordingly; the investigation of the influence however requires further research.

Generation of synthetic minutiae patterns Finally, we believe that the mathematical model for minutiae patterns from Section 4.4 can be applied to produce training data for machine-learning based matching algorithms. In a field where the availability of large databases of realistic fingerprints is still an issue due to privacy constraints this offers great potential for improving the existing algorithms. To produce training data we can draw point patterns from a superimposed process as proposed in Section 4.4 provided the OF and RF from a real fingerprint. Beyond that, to obtain more variability to existing data, one focus could be on generating artificial, realistically looking OF and RF which then are used for the generation of minutiae patterns using our model. First results from a master thesis project by Corvin Grigutsch using a convolutional variational autoencoder (VAE) based approach already promise great potential. However, the formalisation of realness and its usage as objective function in the VAE to produce realistically looking OFs and RFs seems to be a major bottleneck.

The analysis in Section 7.2 suggests that on first order the posterior distributions of different minutiae patterns might resemble one another, opening up new perspectives for sampling of minutiae patterns, namely, sampling from the proposed model for a *typical* or *mean* set of parameters. Whether the minutiae patterns drawn from our model are competitive with minutiae patterns produced by recent algorithms for generating synthetic fingerprints, see e.g. Cappelli et al. (2004); Mistry et al. (2019) for a comparison of recent algorithms, is part of current and future research. However, since all algorithms considered in Mistry et al. (2019) leave, according to them, great potential for improvement regarding a realistic minutiae distribution, this seems to be worthwhile to consider.

A Appendix

A.1 Vector analysis

Theorem A.1 (Green's theorem). *Let $A \subseteq \mathbb{R}^2$ be a bounded domain with piecewise smooth boundary ∂A which is parametrised by the positively oriented curve $\gamma : [a, b] \to \partial A$. Consider for an open neighbourhood $B \supseteq \bar{A}$ of the closure $\bar{A}$ of A the function $f = (f_1, f_2) : B \to \mathbb{R}^2$ which is assumed to be differentiable except in finitely many points. Then,*

$$\int_a^b \langle f(\gamma(t)), \dot{\gamma}(t)\rangle \,\mathrm{d}t = \iint_A \frac{\partial f_2}{\partial x}(x, y) - \frac{\partial f_1}{\partial y}(x, y)\,\mathrm{d}(x, y). \tag{A.1}$$

Proof. Denote by $\vec{n}$ the unit outer normal vector on γ which is given by

$$\vec{n}(\gamma(t)) = \frac{1}{\|\dot{\gamma}(t)\|}\begin{pmatrix}\gamma_2(t)\\ -\gamma_1(t)\end{pmatrix}$$

and let $g = (f_2, -f_1)$. Then,

$$\begin{aligned}\int_a^b \langle f(\gamma(t)), \dot{\gamma}(t)\rangle \,\mathrm{d}t &= \int_a^b \left\langle \begin{pmatrix} f_2(\gamma(t))\\ -f_1(\gamma(t))\end{pmatrix}, \frac{1}{\|\dot{\gamma}(t)\|}\begin{pmatrix}\gamma_2(t)\\ -\gamma_1(t)\end{pmatrix}\right\rangle \|\dot{\gamma}(t)\|\,\mathrm{d}t\\ &= \int_a^b \langle g(\gamma(t)), \vec{n}\,(\gamma(t))\rangle\,\|\dot{\gamma}(t)\|\,\mathrm{d}t.\end{aligned}$$

Applying the divergence theorem (Pfeffer, 1990, Theorem 4.14) to g yields

$$\begin{aligned}\int_a^b \langle g(\gamma(t)), \vec{n}\,(\gamma(t))\rangle\,\|\dot{\gamma}(t)\|\,\mathrm{d}t &= \iint_A \operatorname{div} g(x, y)\,\mathrm{d}(x, y)\\ &= \iint_A \frac{\partial f_2}{\partial x}(x, y) - \frac{\partial f_1}{\partial y}(x, y)\,\mathrm{d}(x, y)\end{aligned}$$

which concludes the proof. ■

A.2 Markov chains

A *Markov chain* is a collection $(X_n)_{n\in\mathbb{N}}$ of random variables X_n on a probability space $(\Omega, \mathcal{A}, \mathbb{P})$ which satisfies the *Markov property*, i.e.

$$\mathbb{P}\left(X_n \in A \mid X_1, \dots X_{n-1}\right) = \mathbb{P}\left(X_n \in A \mid X_{n-1}\right)$$

holds almost surely for all $n \in \mathbb{N}$ and $A \in \mathcal{A}$. Its behaviour can completely be described by its *transition probabilities* $P(x, A) := \mathbb{P}(X_n \in A \mid X_{n-1} = x)$ for $x \in \mathbb{R}^d$ and $A \in \mathcal{A}$. The mapping P is referred to as *Markov kernel.* A probability measure μ is called *stationary* or *invariant measure* of a Markov chain with Markov kernel P if

$$\int_A \mu(\mathrm{d}x) = \int_{\mathbb{R}^d} P(x, A)\mu(\mathrm{d}x).$$

If μ is the stationary measure of a Markov chain $(X_n)_{n\in\mathbb{N}}$ and $X_0 \sim \mu$, then $X_n \sim \mu$ for all $n \in \mathbb{N}$ and $\mathbb{P}(X_n \in A) = \mu(A)$ for all $A \in \mathcal{A}$. In this case, μ is also called *stationary* or *invariant distribution* of (X_n). For a more detailed overview on Markov chains we refer to Meyn & Tweedie (2012).

A.3 Graphs

An (undirected, simple) *graph* $G = (V(G), E(G))$ is a pair of a finite set $V(G)$, the *vertex set*, and a subset $E(G) \subseteq \mathfrak{P}_2(V)$ the set of *edges*. Here, $\mathfrak{P}_2(V)$ denotes the set of all two-element subsets of $V(G)$. A *subgraph* $H = (V(H), E(H))$ of G is a graph with $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$; we write $H \subseteq G$ for short. A subgraph is *induced* , if $E(H) = \mathfrak{P}_2(V(H)) \cap E(G)$. For a set of vertices $W \subseteq V(G)$ we denote by $G[W] = (W, \mathfrak{P}_2(W) \cap E(G))$ the subgraph of G induced by W.

For an edge $e \in E(G)$ connecting vertices $u, v \in V(G)$, we write also $e = uv$ (or $e = vu$) and call u and v *adjacent*. To avoid ambiguities, we always assume $V(G) \cap E(G) = \emptyset$.

For a vertex $v \in V(G)$ the *neighbourhood* of v is given as

$$N_v(G) = \{u \in V(G) \mid uv \in E(G)\}.$$

A graph $P = (V(P), E(P))$ with $V(P) = \{v_1, v_2, \dots, v_n\}$ and $E(P) = \{v_i v_{i+1} \mid i = 1, 2, \dots, n-1\}$ is called *path* from v_1 to v_n if the vertices v_i are pairwise distinct. A graph G is *connected* if for every two vertices $u \neq v \in V(G)$ there is a path P from u to v in G, i.e. P is a subgraph of G. A *component* C of G is a maximally connected subgraph of G, i.e. for each pair $x, y \in V(C)$ there is a path from x to y in C and there is no connected graph $\tilde{C} \subseteq G$ with $V(C) \subsetneq V(\tilde{C})$.

For a graph G and a set $W \subseteq V(G)$ we define $E(W) = \{uv \in E(G) \mid u, v \in W\}$ as the set of edges induced by W. A set $W \subseteq V(G)$ is called *independent* if $E(W) = \emptyset$, i.e. no two vertices $u, v \in W$ are adjacent in G.

Let G be a graph and $w : E(G) \to \mathbb{R}$ be a function, the *weight-function*. Then, (G, w) is called *weighted* graph and $w(e)$ is called *weight* of an edge $e \in E(G)$.

For a more detailed introduction to graph theory see e.g. Diestel (2010).

Notation index

$\mathbb{N}$	the natural numbers $\{1, 2, 3, \dots\}$
$\mathbb{N}_0$	the non-negative integers $\{0, 1, 2, 3, \dots\}$
$\mathbb{Z}$	the integers $\{0, \pm 1, \pm 2, \pm 3, \dots\}$
$\mathbb{R}$	the real numbers
$\mathbb{C}$	the complex numbers
$\mathbb{R}^d$	$\{(v_1, v_2, \dots, v_d) \mid v_i \in \mathbb{R}, i = 1, 2, \dots, d\}$ the vector space of all real-valued d–tuples, $d \in \mathbb{N}$
e_i	the i-th canonical unit vector of $\mathbb{R}^d$ for $i \in \{1, 2, \dots, d\}$
$\mathbb{R}^d_+$	$\{(v_1, v_2, \dots, v_d) \mid v_i \geq 0, i = 1, 2, \dots, d\}$ the non-negative orthant of $\mathbb{R}^d$, $d \in \mathbb{N}$
$\mathbb{R}^d_{++}$	$\{(v_1, v_2, \dots, v_d) \mid v_i > 0, i = 1, 2, \dots, d\}$ the positive orthant of $\mathbb{R}^d$, $d \in \mathbb{N}$
$\mathbb{R}^{n \times m}$	$\{(a(i,j)) : a(i,j) \in \mathbb{R}, 1 \leq i \leq n, 1 \leq j \leq m\}$ the space of all $n \times m$–matrices with real entries, $n, m \in \mathbb{N}$
(a, b) / $[a, b]$	the open/ closed interval from a to b for $a, b \in \mathbb{R} \cup \{\pm\infty\}$ with $a \leq b$
$\mathbb{S}^d$	$\{v \in \mathbb{R}^{d+1} : \lVert v \rVert = 1\}$ the d-dimensional unit sphere, $d \in \mathbb{N}_0$
$\mathbb{R}P^1$	$\mathbb{S}^1/\mathbb{S}^0$ the real projective line
$\operatorname*{argmin}_{m \in M} f(m)$	the set of minimisers of a function $f : M \to \mathbb{R}$
$\operatorname*{argmax}_{m \in M} f(m)$	the set of maximisers of a function $f : M \to \mathbb{R}$

$\mathbb{B}_\varepsilon(x)$	the open ball around x with radius $\varepsilon > 0$
∂A	the boundary of a set A
$\mathfrak{P}(A)$	the power set of a set A
$\mathfrak{P}_k(A)$	the set of all sets $B \subseteq A$ with $\lvert B \rvert = k$
$\langle \cdot, \cdot \rangle$	the canonical scalar product in $\mathbb{R}^d$
$\angle(u, v)$	the angle between two vectors $u, v \in \mathbb{R}^d$
$\arg v$	the argument of $v \in \mathbb{R}^2$, i.e. the angle $\angle(v, e_1)$ between v and the e_1-axis
$\lVert v \rVert$	the Euclidean norm of a vector $v \in \mathbb{R}^d$
$v^\top$	the transposed vector of $v \in \mathbb{R}^d$
$A^\top$	the transposed matrix of $A \in \mathbb{R}^{n \times m}$
$\operatorname{tr} A$	the trace of $A \in \mathbb{R}^{n \times n}$
$\det A$	the determinant of $A \in \mathbb{R}^{n \times n}$
$\operatorname{sgn} x$	the sign of a real number $x \in \mathbb{R}$ where $\operatorname{sgn}(x) = \begin{cases} \frac{x}{\lvert x \rvert}, & x \neq 0, \\ 1, & x = 0. \end{cases}$
$\lfloor x \rceil$	rounding x to the nearest integer with tie-breaking rule round half up
$\mathcal{B}(\mathfrak{X})$	the Borel σ-algebra of the set $\mathfrak{X}$
$\mathfrak{N}$	the space of all finite counting measures on $(\mathfrak{X}, \mathcal{B}(\mathfrak{X}))$, see p. 40
$\mathbb{P}$	a probability measure
$\mathbb{P}_X$	the distribution of a random variable X under a probability measure $\mathbb{P}$
$\operatorname{supp} X$	the support of a random variable X
$\mathbb{E}X$	the expectation of a random variable X
$\operatorname{var} X$	the variance of a random variable X

$\mathrm{cov}(X, Y)$	the covariance of two random variables X, Y
$\mathrm{cor}(X, Y)$	the correlation of two random variables X, Y
δ_x	the Dirac measure measure in $x \in \mathfrak{X}$
$\mathrm{Leb}^d(A)$	the d-dimensional Lebesgue measure on $A \subseteq \mathbb{R}^d$
$\mathrm{Ber}(p)$	the Bernoulli distribution with parameter $p \in [0, 1]$
$\mathrm{Poi}(\lambda)$	the Poisson distribution with intensity $\lambda \geq 0$
$\mathcal{U}([a, b])$	the uniform distribution on $[a, b] \subsetneq \mathbb{R}$ with $a < b$
$\mathcal{N}(\mu, \Sigma)$	the (multivariate) normal distribution with mean $\mu \in \mathbb{R}^d$ and covariance matrix $\Sigma \in \mathbb{R}^{d \times d}$
$\mathcal{LN}(\mu, \Sigma)$	the (multivariate) log-normal distribution with parameters $\mu \in \mathbb{R}^d$ and $\Sigma \in \mathbb{R}^{d \times d}$
$\Gamma(\alpha, \beta)$	the Gamma distribution with shape $\alpha > 0$ and rate $\beta > 0$
$\mathrm{B}(\alpha, \beta)$	the Beta distribution with shape parameters $\alpha > 0$ and $\beta > 0$
$\mathrm{Pop}(\mathfrak{X}, \lambda)$	the Poisson process on $\mathfrak{X}$ with intensity $\lambda : \mathfrak{X} \to \mathbb{R}_+$, see p. 42
$\mathrm{Strauss}(\mathfrak{X}, \beta, \gamma, R)$	the Strauss process on $\mathfrak{X}$ with activity function $\beta : \mathfrak{X} \to \mathbb{R}_+$, interaction parameter $\gamma \in [0, 1]$ and interaction range $R > 0$, see p. 46.
$\mathrm{Hardcore}(\mathfrak{X}, \beta, r)$	the hard core process on $\mathfrak{X}$ with activity function $\beta : \mathfrak{X} \to \mathbb{R}_+$, hard core $r > 0$, see p. 48
$\mathrm{StraussHard}(\mathfrak{X}, \beta, \gamma, r, R)$	the Strauss-hard core process on $\mathfrak{X}$ with activity function $\beta : \mathfrak{X} \to \mathbb{R}_+$, hard core $r > 0$ interaction parameter $\gamma \in [0, 1]$ and interaction range $R > r$, see p. 46
$d_{\min}(\zeta)$	the minimal distance $\min_{u \neq v \in \zeta} \lVert u - v \rVert$ between two distinct points in $\zeta \in \mathfrak{N}$
$s_\rho(\zeta)$	the number of ρ-close pairs of points within $\zeta \in \mathfrak{N}$, $\rho > 0$
$t_\rho(z \mid \zeta)$	the number of ρ-close neighbours of z in $\zeta \in \mathfrak{N}$, $\rho > 0$

$\mathop{\Large\times}\limits_{i=1}^{n} A_i$ . . . . the n-fold cartesian cross product $A_1 \times A_2 \times \cdots \times A_n$

$\mathbf{1}_A$ . . . . the indicator function of a set A where $\mathbf{1}_A(x) = \begin{cases} 1, & x \in A, \\ 0, & \text{else.} \end{cases}$

$g \circ f$ . . . . the concatenation of functions $g \circ f(x) = g(f(x))$

$g * f$ . . . . the (discrete) convolution of functions f and g

$\mathcal{C}(A \to B)$ . . . . the space of continuous functions $f : A \to B$

$\mathcal{C}^k(A \to B)$ . . . . the space of k-times continuously differentiable functions $f : A \to B$, $k \in \mathbb{N}$

$\dot{\gamma}(t)$ . . . . the velocity $\frac{\mathrm{d}}{\mathrm{d}t}\gamma(t)$ of a curve $\gamma \in \mathcal{C}^1\left([a, b] \to \mathbb{R}^d\right)$ in $t \in [a, b]$

∇f . . . . the gradient of a scalar function $f \in \mathcal{C}^1\left(\mathbb{R}^d \to \mathbb{R}\right)$

$\operatorname{div} \vec{F}$ . . . . the divergence of a vector field $\vec{F} \in \mathcal{C}^1\left(\mathbb{R}^2 \to \mathbb{R}^2\right)$

$\operatorname{curl} \vec{F}$ . . . . the curl of a vector field $\vec{F} \in \mathcal{C}^1\left(\mathbb{R}^2 \to \mathbb{R}^2\right)$

$\operatorname{index}_\theta(z)$ . . . . the Poincaré index of a differentiable orientation field $\theta : \mathbb{R}^2 \to \left[-\frac{\pi}{2}, \frac{\pi}{2}\right)$ in z, see p. 21

$G_\rho(\zeta)$ . . . . the interaction graph induced by $\zeta \in \mathfrak{N}$ with interaction range $\rho > 0$, see p. 68

$V(G)$ . . . . the vertex set of a graph G

$E(G)$ . . . . the edge set of a graph G

$N_\rho(u \mid \zeta)$ . . . . the spatial neighbourhood of radius ρ of u in $\zeta \in \mathfrak{N}$, $\rho > 0$

Bibliography

Adams, R., & Essex, C. (2018). *Calculus: A complete course.* Pearson Canada Incorporated.

Alberink, I., de Jongh, A., & Rodriguez, C. (2014). Fingermark evidence evaluation based on automated fingerprint identification system matching scores: the effect of different types of conditioning on likelihood ratios. *Journal of forensic sciences*, *59*(1), 70–81.

Amann, H., & Escher, J. (2008). *Analysis II*. Basel: Birkhäuser, 1st English ed.

Amann, H., & Escher, J. (2009). *Analysis III*. Basel: Birkhäuser, 1st English ed.

Antonopoulos, G. C., Steltner, B., Heisterkamp, A., Ripken, T., & Meyer, H. (2015). Tile-based two-dimensional phase unwrapping for digital holography using a modular framework. *PloS one*, *10*(11), e0143186.

Awad, A. I., & Baba, K. (2012). Singular point detection for efficient fingerprint classification. *International Journal on New Computer Architectures and Their Applications (IJNCAA)*, *2*(1), 1–7.

Ba, I., & Coeurjolly, J.-F. (2020). High-dimensional inference for inhomogeneous Gibbs point processes. *arXiv preprint arXiv:2003.09830.*

Babler, W. (1991). Embryologic development of epidermal ridges and their configurations. *Birth defects original article series*, *27*(2), 95–112.

Baddeley, A., & Nair, G. (2012). Approximating the moments of a spatial point process. *Stat*, *1*(1), 18–30.

Baddeley, A., Nair, G., et al. (2012). Fast approximation of the intensity of Gibbs point processes. *Electronic Journal of Statistics*, *6*, 1155–1169.

Baddeley, A., Rubak, E., & Turner, R. (2015). *Spatial point patterns: methodology and applications with R.* Chapman and Hall/CRC.

Baddeley, A., & Turner, R. (2000). Practical maximum pseudolikelihood for spatial point patterns: (with discussion). *Australian & New Zealand Journal of Statistics*, *42*(3), 283–322.

Baddeley, A., & Turner, R. (2005). spatstat: An R package for analyzing spatial point patterns. *Journal of Statistical Software*, *12*(6), 1–42.

Bansal, R., Sehgal, P., & Bedi, P. (2011). Minutiae extraction from fingerprint images-a review. *arXiv preprint arXiv:1201.1422*.

Bazen, A. M., & Gerez, S. H. (2002). Systematic Methods for the Computation of the Directional Fields and Singular Points of Fingerprints. *IEEE transactions on pattern analysis and machine intelligence*, *24*(7), 905–919.

Berman, M., & Turner, T. R. (1992). Approximating point process likelihoods with GLIM. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *41*(1), 31–38.

Berthelsen, K. K., & Møller, J. (2006). Bayesian analysis of Markov point processes. In *Case studies in spatial point process modeling*. Springer.

Besag, J., & Green, P. J. (1993). Spatial statistics and Bayesian computation. *Journal of the Royal Statistical Society: Series B (Methodological)*, *55*(1), 25–37.

Bogachev, V. I. (2007). *Measure theory*, vol. 1. Springer Science & Business Media.

Bontrager, P., Roy, A., Togelius, J., Memon, N., & Ross, A. (2018). Deepmasterprints: Generating masterprints for dictionary attacks via latent variable evolution. In *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, (pp. 1–9). IEEE.

Bredies, K., & Lorenz, D. (2011). *Mathematische Bildverarbeitung*, vol. 1. Springer.

Brooks, S., Gelman, A., Jones, G., & Meng, X.-L. (2011). *Handbook of Markov chain Monte Carlo*. CRC press.

Burger, M., Düring, B., Kreusser, L. M., Markowich, P. A., & Schönlieb, C.-B. (2018). Pattern formation of a nonlocal, anisotropic interaction model. *Mathematical Models and Methods in Applied Sciences*, *28*(03), 409–451.

Cameron, A. C., & Trivedi, P. K. (2013). *Regression analysis of count data*, vol. 53. Cambridge university press.

Cao, K., & Jain, A. K. (2015). Learning fingerprint reconstruction: From minutiae to image. *IEEE Transactions on Information Forensics and Security*, *10*(1), 104–117.

Cao, K., & Jain, A. K. (2018). Fingerprint synthesis: Evaluating fingerprint search at scale. In *2018 International Conference on Biometrics (ICB)*, (pp. 31–38).

Cappelli, R., Ferrara, M., & Maltoni, D. (2010a). Minutia Cylinder-Code: A New Representation and Matching Technique for Fingerprint Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(12), 2128–2141.

Cappelli, R., Ferrara, M., & Maltoni, D. (2011). Fingerprint Indexing Based on Minutia Cylinder-Code. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(5), 1051–1057.

Cappelli, R., Ferrara, M., Maltoni, D., & Tistarelli, M. (2010b). MCC: A baseline algorithm for fingerprint verification in FVC-onGoing. In *2010 11th International Conference on Control Automation Robotics Vision*, (pp. 19–23).

Cappelli, R., Maio, D., & Maltoni, D. (2002). Synthetic fingerprint-database generation. In *Object recognition supported by user interaction for service robots*, vol. 3, (pp. 744–747). IEEE.

Cappelli, R., Maio, D., & Maltoni, D. (2004). SFinGe: An approach to synthetic fingerprint generation. In *International Workshop on Biometric Technologies (BT2004)*, (pp. 147–154).

Chen, J., & Moon, Y.-S. (2006). A statistical study on the fingerprint minutiae distribution. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 2, (pp. II–II). IEEE.

Chikkerur, S., Cartwright, A. N., & Govindaraju, V. (2007). Fingerprint enhancement using STFT analysis. *Pattern recognition*, *40*(1), 198–211.

Chiu, S. N., Stoyan, D., Kendall, W. S., & Mecke, J. (2013). *Stochastic geometry and its applications*. John Wiley & Sons.

Chung, C. (1981). Application of the Buffon needle problem and its extensions to parallel-line search sampling scheme. *Journal of the International Association for Mathematical Geology*, *13*(5), 371–390.

Clifford, P., & Nicholls, G. (1994). Comparison of Birth-and-Death and Metropolis-Hastings Markov chain Monte Carlo for the Strauss process. Tech. rep.

Coeurjolly, J.-F., & Drouilhet, R. (2010). Asymptotic properties of the maximum pseudo-likelihood estimator for stationary Gibbs point processes including the Lennard-Jones model. *Electronic journal of statistics*, *4*, 677–706.

Coeurjolly, J.-F., & Lavancier, F. (2018). Intensity approximation for pairwise interaction Gibbs point processes using determinantal point processes. *Electronic journal of statistics*, *12*(2), 3181–3203.

Coeurjolly, J.-F., & Rubak, E. (2013). Fast covariance estimation for innovations computed from a spatial Gibbs point process. *Scandinavian Journal of Statistics*, *40*(4), 669–684.

Cressie, N., & Lawson, A. B. (2000). Hierarchical probability models and Bayesian analysis of mine locations. *Advances in Applied Probability*, (pp. 315–330).

Cummins, H., & Midlo, C. (1976). Fingerprints, Palms, and Soles. South Berlin.

Daley, D. J., & Vere-Jones, D. (2003). *An introduction to the theory of point processes. Volume I: Elementary Theory and its Methods*. Springer.

Daley, D. J., & Vere-Jones, D. (2008). *An introduction to the theory of point processes. Volume II: General Theory and Structure*. Springer.

Dereudre, D., Lavancier, F., et al. (2017). Consistency of likelihood estimation for Gibbs point processes. *The Annals of Statistics*, *45*(2), 744–770.

Diestel, R. (2010). *Graph theory*. Graduate Texts in Mathematics ; 173. Berlin: Springer, 4th ed.

Düring, B., Gottschlich, C., Huckemann, S., Kreusser, L. M., & Schönlieb, C.-B. (2019). An anisotropic interaction model for simulating fingerprints. *Journal of mathematical biology*, *78*(7), 2171–2206.

Feng, J., & Jain, A. K. (2011). Fingerprint reconstruction: From minutiae to phase. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(2), 209–223.

Forbes, P. G., Lauritzen, S., & Møller, J. (2014). Fingerprint analysis with marked point processes. *arXiv preprint arXiv:1407.5809*.

Fulton, W. (2013). *Algebraic topology: a first course*, vol. 153. Springer Science & Business Media.

Galton, S. F. (1892). *Hereditary genius: An inquiry into laws and consequences*. MacMillan.

Garris, M. D., Watson, C. I., McCabe, R., & Wilson, C. L. (2001). User's guide to NIST fingerprint image software (NFIS). *No. NIST Interagency/Internal Report (NISTIR)-6813*.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). *Bayesian data analysis*. CRC press.

Ghiglia, D. C., & Pritt, M. D. (1998). Two-dimensional phase unwrapping: theory, algorithms, and software. *Wiely-Interscience, first ed.(April 1998)*.

Gottschlich, C. (2011). Curved-region-based ridge frequency estimation and curved Gabor filters for fingerprint image enhancement. *IEEE Transactions on Image Processing*, *21*(4), 2220–2227.

Gottschlich, C., & Huckemann, S. (2014). Separating the real from the synthetic: minutiae histograms as fingerprints of fingerprints. *IET Biometrics*, *3*(4), 291–301.

Gottschlich, C., Tams, B., & Huckemann, S. (2016). Perfect fingerprint orientation fields by locally adaptive global models. *IET Biometrics*, *6*(3), 183–190.

Gutiérrez-Redomero, E., Alonso, C., Romero, E., & Galera, V. (2008). Variability of fingerprint ridge density in a sample of Spanish Caucasians and its application to sex determination. *Forensic Science International*, *180*(1), 17–22.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, *57*(1), 97–109.

Hong, L., & Jain, A. (1999). Classification of fingerprint images. In *Proceedings of the scandinavian conference on image analysis*, vol. 2, (pp. 665–672).

Hong, L., Wan, Y., & Jain, A. (1998). Fingerprint image enhancement: algorithm and performance evaluation. *IEEE transactions on pattern analysis and machine intelligence*, *20*(8), 777–789.

Huang, F., & Ogata, Y. (1999). Improvements of the maximum pseudo-likelihood estimators in various spatial statistical models. *Journal of Computational and Graphical Statistics*, *8*(3), 510–530.

Huckemann, S., Hotz, T., & Munk, A. (2008). Global models for the orientation field of fingerprints: an approach based on quadratic differentials. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *30*(9), 1507–1519.

Illian, J., Penttinen, A., Stoyan, H., & Stoyan, D. (2008). *Statistical analysis and modelling of spatial point patterns*, vol. 70. John Wiley & Sons.

Imdahl, C., Huckemann, S., & Gottschlich, C. (2015). Towards generating realistic synthetic fingerprint images. In *2015 9th International Symposium on Image and Signal Processing and Analysis (ISPA)*, (pp. 78–82). IEEE.

ISO/IEC 19794–2:2011 (2011). Information technology — Biometric data interchange formats — Part 2: Finger minutiae data. Standard, International Organization for Standardization, Geneva, CH.

Jain, A. K., Prabhakar, S., & Pankanti, S. (2002). On the similarity of identical twin fingerprints. *Pattern Recognition*, *35*(11), 2653–2663.

Jarner, S. F., & Roberts, G. O. (2007). Convergence of heavy-tailed monte carlo markov chain algorithms. *Scandinavian Journal of Statistics*, *34*(4), 781–815.

Jensen, J. L. (1993). Asymptotic normality of estimates in spatial point processes. *Scandinavian Journal of Statistics*, (pp. 97–109).

Kawagoe, M., & Tojo, A. (1984). Fingerprint pattern classification. *Pattern recognition*, *17*(3), 295–303.

Kelly, F. P., & Ripley, B. D. (1976). A note on Strauss's model for clustering. *Biometrika*, (pp. 357–360).

Kendall, W. S., & Møller, J. (1999). *Perfect Metropolis-Hastings simulation of locally stable point processes.* University of Aarhus. Centre for Mathematical Physics and Stochastics (MaPhySto).

Kendall, W. S., & Møller, J. (2000). Perfect simulation using dominating processes on ordered spaces, with application to locally stable point processes. *Advances in Applied Probability*, (pp. 844–865).

Kingma, D. P., & Welling, M. (2019). An introduction to variational autoencoders. *arXiv preprint arXiv:1906.02691*.

Klenke, A. (2020). *Probability theory: a comprehensive course*. Springer Science & Business Media, 3rd ed.

Kovacs-Vajna, Z. M., Rovatti, R., & Frazzoni, M. (2000). Fingerprint ridge distance computation methodologies. *Pattern Recognition*, *33*(1), 69–80.

Kücken, M. (2007). Models for fingerprint pattern formation. *Forensic science international*, *171*(2-3), 85–96.

Kücken, M., & Newell, A. C. (2005). Fingerprint formation. *Journal of theoretical biology*, *235*(1), 71–83.

Kücken, M., & Champod, C. (2013). Merkel cells and the individuality of friction ridge skin. *Journal of Theoretical Biology*, *317*, 229–237.

Liu, J. S. (2008). *Monte Carlo strategies in scientific computing*. Springer Science & Business Media.

Maio, D., Maltoni, D., Cappelli, R., Wayman, J. L., & Jain, A. K. (2002). FVC2002: Second fingerprint verification competition. In *Object recognition supported by user interaction for service robots*, vol. 3, (pp. 811–814). IEEE.

Maltoni, D., Maio, D., Jain, A. K., & Prabhakar, S. (2009). *Handbook of fingerprint recognition*. Springer Science & Business Media.

Mase, S. (1992). Uniform LAN condition of planar Gibbsian point processes and optimality of maximum likelihood estimators of soft-core potential functions. *Probability theory and related fields*, *92*(1), 51–67.

Matthews, P. C. (1998). *Vector calculus*. Springer undergraduate mathematics series. London [u.a.]: Springer.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics*, *21*(6), 1087–1092.

Meyn, S. P., & Tweedie, R. L. (2012). *Markov chains and stochastic stability*. Springer Science & Business Media.

Minaee, S., & Abdolrashidi, A. (2018). Finger-GAN: Generating realistic fingerprint images using connectivity imposed GAN. *arXiv preprint arXiv:1812.10482*.

Mistry, V., Engelsma, J. J., & Jain, A. K. (2019). Fingerprint Synthesis: Search with 100 million Prints. *arXiv preprint arXiv:1912.07195*.

Møller, J., & Waagepetersen, R. P. (2003). *Statistical inference and simulation for spatial point processes*. Chapman and Hall/CRC.

Møller, J., & Waagepetersen, R. P. (2007). Modern statistics for spatial point processes. *Scandinavian Journal of Statistics*, *34*(4), 643–684.

Moreno, P., Bernardino, A., & Santos-Victor, J. (2009). Improving the SIFT descriptor with smooth derivative filters. *Pattern Recognition Letters*, *30*(1), 18–26.

Murray, I., Ghahramani, Z., & MacKay, D. (2012). MCMC for doubly-intractable distributions. *arXiv preprint arXiv:1206.6848*.

Myllymäki, M., & Mrkvička, T. (2019). GET: Global envelopes in R. *arXiv preprint arXiv:1911.06583*.

Myllymäki, M., Mrkvicka, T., Grabarnik, P., Seijo, H., & Hahn, U. (2017). Global envelope tests for spatial processes. *Journal of the Royal Statistical Society*.

Nayak, V. C., Rastogi, P., Kanchan, T., Yoganarasimha, K., Kumar, G. P., & Menezes, R. G. (2010). Sex differences from fingerprint ridge density in Chinese and Malaysian population. *Forensic science international*, *197*(1-3), 67–69.

Neal, P., & Roberts, G. (2011). Optimal scaling of random walk metropolis algorithms with non-gaussian proposals. *Methodology and Computing in Applied Probability*, *13*(3), 583–601.

Neumann, C., & Ausdemore, M. (2020). Defence against the modern arts: the curse of statistics—part II: ”score-based likelihood ratios”. *Law, Probability and Risk*, *19*(1), 21–42.

Neumann, C., & Ausdemore, M. A. (2019). Defence against the modern arts: the curse of statistics ”score-based likelihood ratios”. *arXiv preprint arXiv:1910.05240*.

Newman, H. H. (1930). The finger prints of twins. *Journal of Genetics*, *23*(3), 415–446.

Nikodémusz-Székely, É., & Székely, V. (1993). Image recognition problems of fingerprint identification. *Microprocessors and Microsystems*, *17*(4), 215 – 218.

Nixon, M., & Aguado, A. (2019). *Feature extraction and image processing for computer vision*. Academic press.

Page, M., Taylor, J., & Blenkin, M. (2011). Uniqueness in the forensic identification sciences—fact or fiction? *Forensic science international*, *206*(1-3), 12–18.

Pakanti, S., Prabhakar, S., & Jain, A. (2002). On the individuality of fingerprint. *IEEE Trans. Pattern Anal. Machine Intell*, *24*(8), 1010–1025.

Patriciu, V.-V., & Spinu, S. (2014). Fingerprint ridge frequency estimation in the Fourier domain. *Advances in Electrical and Computer Engineering*, *14*(4), 95–98.

Penrose, L. S. (1969). Dermatoglyphics. *Scientific American*, *221*(6), 72–87.

Pfeffer, W. F. (1990). Divergence theorem for vector fields with singularities. In *New Integrals*, (pp. 150–166). Springer.

Poisson, S. D. (1837). *Recherches sur la probabilité des jugements en matière criminelle et en matière civile*. Bachelier.

R Core Team (2021). *R: A Language and Environment for Statistical Computing*. Vienna, Austria. https://www.R-project.org.

Rajala, T., Redenbach, C., Särkkä, A., & Sormani, M. (2016). Variational Bayes approach for classification of points in superpositions of point processes. *Spatial statistics*, *15*, 85–99.

Redenbach, C., Särkkä, A., & Sormani, M. (2015). Classification of points in superpositions of Strauss and Poisson processes. *Spatial Statistics*, *12*, 81 – 95.

Ripley, B. (1987). *Stochastic simulation*. Wiley series in probability and statistics. Wiley.

Ripley, B. D. (1988). *Statistical inference for spatial processes*. Cambridge university press.

Robert, C., & Casella, G. (2013). *Monte Carlo statistical methods*. Springer Science & Business Media.

Roberts, G. O. (1996). Markov chain concepts related to sampling algorithms. *Markov chain Monte Carlo in practice*, *57*, 45–58.

Roberts, G. O., & Rosenthal, J. S. (1998). Optimal scaling of discrete approximations to langevin diffusions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *60*(1), 255–268.

Roberts, G. O., & Rosenthal, J. S. (2006). Harris recurrence of Metropolis-within-Gibbs and trans-dimensional Markov chains. *The Annals of Applied Probability*, (pp. 2123–2139).

Roberts, G. O., & Rosenthal, J. S. (2007). Coupling and ergodicity of adaptive markov chain monte carlo algorithms. *Journal of applied probability*, *44*(2), 458–475.

Roberts, G. O., & Rosenthal, J. S. (2009). Examples of adaptive mcmc. *Journal of computational and graphical statistics*, *18*(2), 349–367.

Rudolf, D., & Schweizer, N. (2018). Perturbation theory for Markov chains via Wasserstein distance. *Bernoulli*, *24*(4A), 2610–2639.

Sadegh Riazi, M., Chavoshian, S. M., & Koushanfar, F. (2020). SynFi: Automatic Synthetic Fingerprint Generation. *arXiv*, (pp. arXiv–2002).

Schervish, M. J. (2012). *Theory of statistics*. Springer Science & Business Media.

Scientific Working Group on Friction Ridge Analysis, S., Technology, & of America, U. S. (2011). Fingerprint sourcebook.

Sherlock, B. G. (2004). Computer Enhancement and Modeling of Fingerprint Images. In N. Ratha, & R. Bolle (Eds.) *Automatic Fingerprint Recognition Systems*, (pp. 87–112). New York, NY: Springer New York.

Sherlock, B. G., & Monro, D. M. (1993). A model for interpreting fingerprint topology. *Pattern recognition*, *26*(7), 1047–1055.

Stoney, D. A. (1988). Distribution of epidermal ridge minutiae. *American Journal of Physical Anthropology*, *77*(3), 367–376.

Strauss, D. J. (1975). A model for clustering. *Biometrika*, *62*(2), 467–475.

Stucki, K., & Schuhmacher, D. (2014). Bounds for the probability generating functional of a Gibbs point process. *Advances in Applied Probability*, *46*(1), 21–34.

Tams, B., Mihăilescu, P., & Munk, A. (2015). Security considerations in minutiae-based fuzzy vaults. *IEEE Transactions on Information Forensics and Security*, *10*(5), 985–998.

Thai, D. H., Huckemann, S., & Gottschlich, C. (2016). Filter design and performance evaluation for fingerprint image segmentation. *PloS one*, *11*(5).

Tsvetkov, D., Hristov, L., & Angelova-Slavova, R. (2013). On the convergence of the Metropolis-Hastings Markov chains. *arXiv preprint arXiv:1302.0654*.

Vizcaya, P. R., & Gerhardt, L. A. (1996). A nonlinear orientation model for global description of fingerprints. *Pattern Recognition*, *29*(7), 1221–1231.

Walsh, D. C. I., & Raftery, A. E. (2002). Detecting mines in minefields with linear characteristics. *Technometrics*, *44*(1), 34–44.

Watson, C. I., & Wilson, C. L. (1992). NIST special database 4. *Fingerprint Database, National Institute of Standards and Technology*, *17*(77), 5.

Watson, G. (1978). Characteristic statistical problems of stochastic geometry. In *Geometrical Probability and Biological Structures: Buffon's 200th Anniversary*, (pp. 215–234). Springer.

Wieditz, J. (2020). Minutiae Separating Algorithm (MiSeal). https://github.com/jwieditz/MiSeal.

Wieditz, J. (2021a). Maximum pseudo-likelihood estimation for superpositions of Strauss-hard core and Poisson processes. https://github.com/jwieditz/SuperMPLE.

Wieditz, J. (2021b). Personal website – Software. https://stochastik.math.uni-goettingen.de/supermple. Accessed: 8th July 2021.

Wieditz, J., Pokern, Y., Schuhmacher, D., & Huckemann, S. (2021). Characteristic and necessary minutiae in fingerprints. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*. Accepted.

Wilson, C. L., Candela, G. T., & Watson, C. I. (1994). Neural network fingerprint classification. *Journal of Artificial Neural Networks*, *1*(2), 203–228.

Wyzykowski, A. B. V., Segundo, M. P., & Lemes, R. d. P. (2020). Level three synthetic fingerprint generation. *arXiv preprint arXiv:2002.03809*.

Young, G. A., & Smith, R. L. (2005). *Essentials of Statistical Inference*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.

Zaeri, N. (2011). Minutiae-based fingerprint extraction and recognition. In J. Yang (Ed.) *Biometrics*. Rijeka: IntechOpen.

Zhan, X., Sun, Z., Yin, Y., & Chu, Y. (2006). Fingerprint ridge distance estimation: algorithms and the performance. In *International Conference on Biometrics*, (pp. 294–301). Springer.

Zhao, H., Lai, Z., Leung, H., & Zhang, X. (2020). *Feature Learning and Understanding: Algorithms and Applications*. Springer Nature.

Zhu, C., Byrd, R. H., Lu, P., & Nocedal, J. (1997). Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on mathematical software (TOMS)*, *23*(4), 550–560.

Index