Robust Parametric and Semi-parametric Spot Fitting for Spot Array Images

Norbert Brändle[1], Horng-Yang Chen[1], Horst Bischof [1], Hilmar Lapp[2]

[1]Vienna University of Technology
Pattern Recognition and Image Processing Group
Treitlstr. 3/1832, A-1040 Vienna
Tel. +43 1 58801-18360, Fax +43 1 58801-18392
nob@prip.tuwien.ac.at

[2]Novartis Research Institute Vienna
Genetics
Brunner Str. 59, A-1230 Vienna
Tel. +43 1 86634-631
hilmar.lapp@pharma.novartis.com

Abstract

In this paper we address the problem of reliably fitting parametric and semi-parametric models to spots in high density spot array images obtained in gene expression experiments. The goal is to measure the amount of label bound to an array element. A lot of spots can be modelled accurately by a Gaussian shape. In order to deal with highly overlapping spots we use robust M-estimators. When the parametric method fails (which can be detected automatically) we use a novel, robust semi-parametric method which can handle spots of different shapes accurately. The introduced techniques are evaluated experimentally.

Keywords: spot arrays, image analysis, robust methods, parametric fitting

Introduction

Genetic spot array images have to be analyzed in the course of high-throughput hybridization experiments, where a spot in the array can identify specific expressed gene products. Common to all array-based approaches is the necessity to analyze digital images of the array. The ultimate image analysis goal is to automatically assign a quantity to every array element giving information about the hybridization signal (*spot fitting* or *quantification*). Figures 1 and 2 show a typical array image generated in the course of a oligonucleotide fingerprint (ONF) experiment: The high density medium is a filter (nylon membrane) comprising a total of 57600 cDNA spots which were spotted in different spotting cycles by a robot arm carrying a matrix of needles. Detailed information about the spotting procedure can be found in (Meier-Ewert *et al.* 1993). The intensity of every spot corresponds to the amount of label remaining after hybridizing a liquid containing the labelled probes and subsequently washing off probe not bound to the genetic material. A digital image of the filter is generated with the help of a scanner. In the framework of a fully automated image analysis, the *grid fitting* procedure should automatically provide coarse initial locations of the spots. An approach for spot array images containing a safety grid of *guide spots* is described in (Brändle, Lapp, & Bischof 1999). There, the grid fitting procedure consists of the following main steps: Initially, the guide spot locations are detected by a maximum search in a response image comprising amplified guide spot locations. The guide spot locations are amplified by simple digital filters. Additionally, a prior guide spot grid is defined with the help of theoretical spot distances given by the image size and the scanner resolution. The main idea is to transform (rotate and translate) the prior guide spot grid with the help of global estimates of the grid rotation and the grid translation. The transformed prior guide spot grid can then be used to uniquely assign the detected guide spot locations to the grid nodes. The locations of non-guide spots can be initialized with the help of the grid rotation estimation and the theoretical spot size. The initialized spot

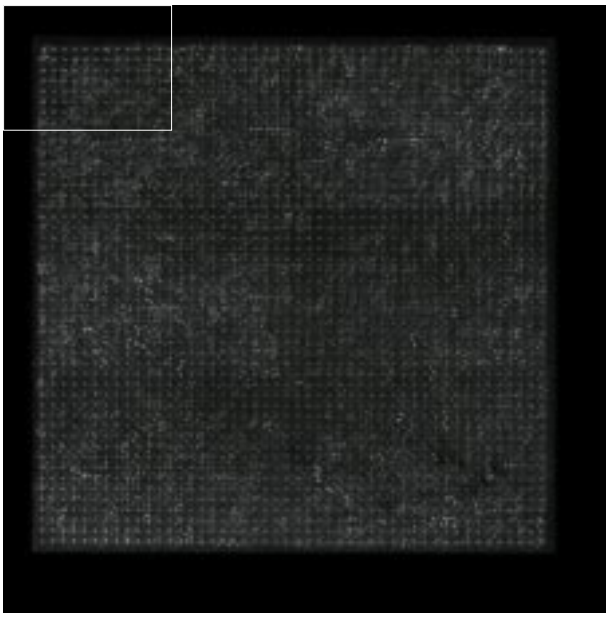


Figure 1: Genetic Spot Array Image. The white rectangle indicates the region belonging to the zoomed right image shown in Fig. 2

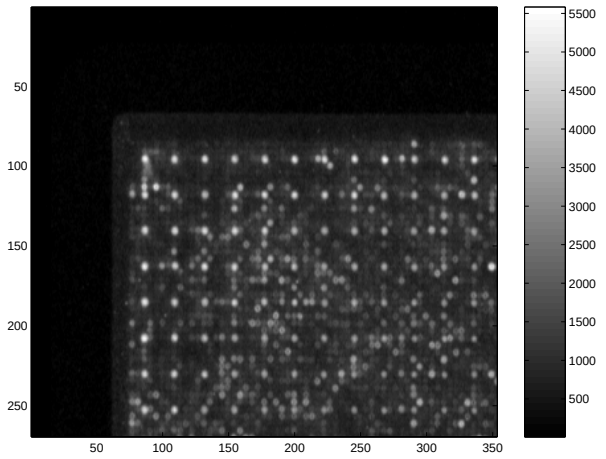
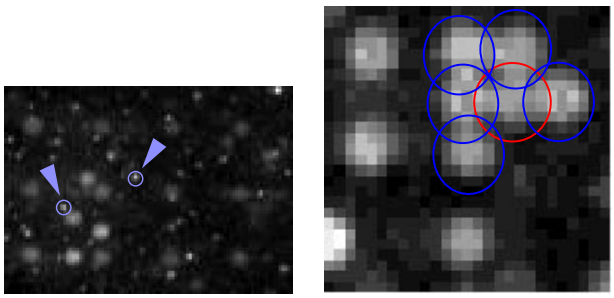


Figure 2: Genetic Spot Array Image. Zoomed image corresponding to the white rectangle of Fig. 1

locations are refined by the methods described in this paper. We first describe the main problems of quantification and the problem of background estimation. The next section deals with non-robust and robust parametric spot fitting. Then we introduce a semi-parametric spot fitting method. Finally, we present experimental results on real data.

Quantification

The goal of the spot fitting procedure is to provide an accurate estimate of the volume, i.e. the amount of genetic material of every spot. It must cope with the following three major problems:



(a) Artifacts (b) Overlapping spots

Figure 3: Spot fitting problems

1. *Noise and outliers*: Sometimes gross errors like artifacts which do not comprise gene expression information can occur (Fig. 3a).
2. *Overlapping spots*: Spots with high intensity or spots in low-resolution images may interfere with neighboring spots (Fig. 3b).
3. *Various spot shapes*: Depending on the type of the experiment different spot shapes are possible (Fig. 4).

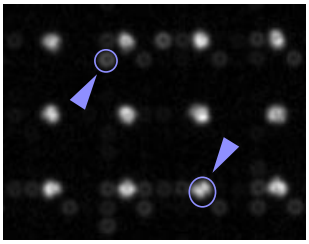


Figure 4: Spots with uncommon shapes

We decided to use parametric spot models, i.e. two-dimensional analytic models the parameters of which are fitted to the given spot data. This allows us to deal adequately with the phenomenon of overlapping spots. However, parametric spot models rely on a priori assumptions about the spot shape which cannot always be guaranteed. We therefore also introduce a more flexible semi-parametric spot fitting procedure. Noordmans and Smeulders (Noordmans & Smeulders 1998) provided a general approach for the detection and characterization of overlapping spots. However, their approach is restricted to parametric models and is non-robust. Robustness is required in order to deal with artifacts. In the following two sections we describe both a parametric and a non-parametric approach for spot fitting.

Estimating the Background

A problem independent of the spot fitting strategy is the fact that the background values (areas with no hybridization signal) need not be uniform across the spot image. The background values are an important additive constant for spot models. In order to obtain continuously varying background values for the entire image we estimate the background in a global manner. In principle, the *background image* is the spot image subtracted by the quantified spots. However, the information about the hybridization signal is available *after* the spot fitting. To tackle this chicken-egg problem, we estimate the background in two passes. The method we show is applicable to spot arrays with a small percentage of hybridized spots including a guide spot safety grid, like the ONF image in Fig. 1. An adaption to other kinds of spot array images is straightforward. The grid fitting procedure provides us with the locations of the guide spots. We furthermore know that the guide spots always have a (strong) hybridization signal. As a first approximation we could therefore subtract the pixels belonging to the guide spots from the spot image. In order to get smooth results, we use a hierarchical interpolation method based on Gaussian image pyramids (Jolion & Rosenfeld 1994). An image pyramid combines the advantages of high and low resolution. It is a collection of images $\mathbf{S}^{[l]}$ of a single scene at exponentially

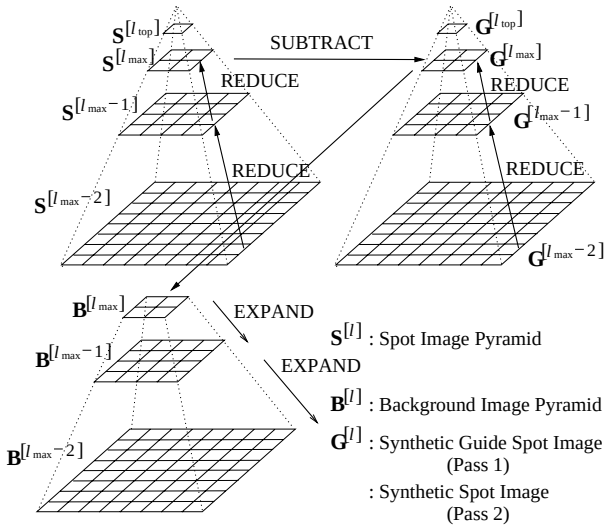


Figure 5: Principle of Background Estimation

decreasing resolutions $l \in \{0 \dots l_{\text{top}}\}$. The bottom level of the pyramid is the original image. In the simplest case, each successive level of the pyramid is obtained from the previous level by a filtering operation followed by a sampling operator (Haralick, , & Shapiro 1991). Figure 5 plots the principle of the background estimation with image pyramids.

Pass 1 This pass is performed after grid fitting and before the spot fitting. A pyramid $\mathbf{S}^{[l]}$ of the spot image and a pyramid $\mathbf{G}^{[l]}$ of the synthetic guide spot image are built. At the level $l_{\max}$ the resolution of the images is so low such that the guide spot grid structure is no longer present in the images, meaning that the guide spots are merged. At the merging level $l_{\max}$ we subtract $\mathbf{G}^{[l_{\max}]}$ from $\mathbf{S}^{[l_{\max}]}$ resulting in a low-resolution background image $\mathbf{B}^{[l_{\max}]}$. In order to get a background image at the original resolution of the spot image, the levels of the background pyramid are computed by the EXPAND function, which consists of bicubic interpolation of the grey values (Press *et al.* 1992).

Pass 2 The background is estimated a second time after the spot fitting procedure has finished. With the knowledge about the model parameters for every spot we are able to reconstruct a complete synthetic spot image (see also Fig. 15). The pyramid $\mathbf{G}^{[l_{\max}]}$ in Fig. 5 is now the reconstructed synthetic spot image. The subtraction and the expansion are performed the same way as in pass 1.

Parametric Spot Fitting

A parametric fit on a set of intensities (pixels) belonging to a spot assumes a given analytic model the unknown parameters of which have to be determined. The approximate initial locations of the spots are given by the grid fitting procedure. The extension of the pixel set belonging to a spot is determined by the prior knowledge about the theoretical spot size, which in return is given by the image size and the scanner resolution. In this section we first introduce a Gaussian spot model. We then describe a non-robust fit for the Gaussian parameters. In the last part of this section we introduce a robust spot fitting method.

The Gaussian Spot Model

Let $S = \{(\mathbf{p}_i, z_i),\ \mathbf{p} \in \mathbb{R}^2,\ z_i \in \mathbb{R}\}$ be a set of n points corresponding to a spot, where z_i denotes the intensity at location $\mathbf{p}_i$. An initial analysis has shown that most of the spots can be characterized fairly accurately by a Gaussian shape. The (non-normalized) Gaussian function is denoted as

$$\mathrm{G}(\mathbf{p}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \exp\left[-\frac{1}{2}(\mathbf{p}-\boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\mathbf{p}-\boldsymbol{\mu})\right] \qquad (1)$$

with $\boldsymbol{\mu} \in \mathbb{R}^2$ and $\boldsymbol{\Sigma} \in \mathbb{R}^{2\times 2}$ (2×2 matrix). Let the Gaussian spot model be defined as

$$\mathrm{Z}(\mathbf{p}) = \mathrm{Z}(a, \mathbf{p}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, b) := a\ \mathrm{G}(\mathbf{p}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) + b \qquad (2)$$

with the following parameters:

1. a is the amplitude of the Gaussian model corresponding to the "height" of the spot.
2. $\boldsymbol{\mu}$ is the mean of the Gaussian model corresponding to the "center" (location) of the spot.
3. $\boldsymbol{\Sigma}$ is the 2×2 dispersion matrix:

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{pmatrix} \qquad (3)$$

   describing the extension of the spot.
4. b is the background value. Note that the background $b(\mathbf{p})$ is slowly varying over the image. For the characterization of the background of a spot it is sufficient to take one background sample $b(\boldsymbol{\mu})$ at the spot center $\boldsymbol{\mu}$. Hence we can denote the background simply as b.

Non-robust Parameter Estimation

The parameters of the Gaussian spot model (2) can be computed by maximum likelihood (ML) estimators and minimization of the sum of square errors (Bishop 1995).

Estimating the Mean and Dispersion Matrix The shape of a spot can be interpreted as a distribution of the x- and y-coordinates. A spot patch S contains n data points $\mathbf{p}_i$ and has the intensities $\mathrm{I}(\mathbf{p}_i)$. In order to take into account the estimated background value $\widehat{b}$ we denote the corrected intensities as $z_i := \max(\mathrm{I}(\mathbf{p}_i) - \widehat{b}, 0)$ [1]. The ML estimate $\widehat{\boldsymbol{\mu}}$ of the

[1] since the background can be overestimated – especially in the first background estimation – we correct negative values to zero

center $\boldsymbol{\mu}$ is then computed as follows:

$$\widehat{\boldsymbol{\mu}} = \frac{1}{T}\sum_{i=1}^{n} z_i \mathbf{p}_i \tag{4}$$

where T is the total sum of the corrected intensities of the patch:

$$T = \sum_{i=1}^{n} z_i. \tag{5}$$

Similarly, the ML estimate $\widehat{\boldsymbol{\Sigma}}$ of the dispersion matrix $\boldsymbol{\Sigma}$ is given by

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{T}\sum_{i=1}^{n} z_i (\mathbf{p}_i - \widehat{\boldsymbol{\mu}})(\mathbf{p}_i - \widehat{\boldsymbol{\mu}})^T \tag{6}$$

Hence, the estimate $\widehat{\boldsymbol{\mu}}$ of the center $\boldsymbol{\mu}$ is given by the sample average (i.e. the average with respect to the given data set) of the coordinates weighted by the corrected pixel intensities. Similarly, the ML estimate $\widehat{\boldsymbol{\Sigma}}$ of the dispersion matrix $\boldsymbol{\Sigma}$ is given by the sample average of the outer product $(\mathbf{p}_i - \widehat{\boldsymbol{\mu}})(\mathbf{p}_i - \widehat{\boldsymbol{\mu}})^T$ weighted by the pixel intensities.

Estimating the amplitude The estimate $\widehat{a}$ of the amplitude a can be computed by a minimization of the sum of square errors. Let us define the error function between the data point and the Gaussian function as

$$\mathrm{E} = -\frac{1}{2}\sum_{i=1}^{n} \Big\{ z_i - a\ \mathrm{G}(\mathbf{p}_i, \widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\Sigma}}) \Big\}^2. \tag{7}$$

The estimate for a is computed by setting the partial derivative of E with respect to the parameter a to zero:

$$\frac{\partial \mathrm{E}}{\partial a} = \sum_{i=1}^{n} \Big\{ z_i - a\ \mathrm{G}(\mathbf{p}_i, \widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\Sigma}}) \Big\}\, \mathrm{G}(\mathbf{p}_i, \widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\Sigma}}) = 0. \tag{8}$$

The solution to (8) yields the estimator $\widehat{a}$:

$$\widehat{a} = \sum_{i=1}^{n} z_i \mathrm{G}(\mathbf{p}_i, \widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\Sigma}}) \Bigg/ \sum_{i=1}^{n} \mathrm{G}(\mathbf{p}_i, \widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\Sigma}})^2. \tag{9}$$

Quantification The brightness V of the spot is estimated as the volume under the fitted Gaussian function:

$$\widehat{V} = \widehat{a}\ (2\pi)\ \sqrt{\det(\widehat{\boldsymbol{\Sigma}})}. \tag{10}$$

A derivation of the Gaussian integral (10) can be found in (Bishop 1995). Sometimes the scanner is square rooting the intensities during the scanning process. We therefore provide an estimator for the brightness W of the spot with squared intensities. Using the fact that $\mathrm{G}(\mathbf{p}_i, \boldsymbol{\mu}, \boldsymbol{\Sigma})^2 = \mathrm{G}(\mathbf{p}_i, \boldsymbol{\mu}, 2\cdot\boldsymbol{\Sigma})$, one can easily verify that

$$\widehat{W} = (\widehat{a})^2\ \pi\ \sqrt{\det(\widehat{\boldsymbol{\Sigma}})}. \tag{11}$$

The estimators presented so far are non-robust, meaning that they are sensitive to outliers, like the artifacts in (Fig. 3a).

Robust Parameter Estimation

We show how parametric models can be fit in a robust manner. Information about robust statistics can be found in (Huber 1981) and (Rousseeuw & Leroy 1987). One quality measure of a robust estimator is the *breakdown point*. The breakdown point ϵ^* gives the limit to which the percentage of outliers can increase which the estimator still can tolerate. For instance the breakdown point of the mean is $\epsilon^* = 0.0$ and the breakdown point of the median is $\epsilon^* = 0.5$ We have chosen M-Estimators for the spot fitting problem. We first introduce the theory of M-estimators for univariate distributions.

M-Estimators M-Estimators (ML type estimators) of location are based on the idea of replacing the squared error between the data and the model by another function ρ of the error. Let $x_1, x_2, \ldots, x_n$ be a sequence of identically independently distributed observations. The M-estimator of location $\tilde{\mu} = \tilde{\mu}(x_1, x_2, \ldots, x_n)$ is defined as the solution of the minimizing problem

$$\sum_{i=1}^{n} \rho(x_i - \mu)\ \rightarrow\ \min \tag{12}$$

with respect to μ, where ρ is a function $\mathbb{R} \mapsto \mathbb{R}$. If ψ denotes the first derivative of ρ with respect to μ, the estimator $\tilde{\mu}$ is the solution to the equation

$$\sum_{i=1}^{n} \psi(x_i - \mu) = 0. \tag{13}$$

Equation (13) can be written equivalently as

$$\sum_{i=1}^{n} w_i (x_i - \mu) = 0 \tag{14}$$

with

$$w_i = \frac{\psi(x_i - \mu)}{x_i - \mu}. \tag{15}$$

This gives a formal representation of μ as a weighted mean

$$\mu = \sum_{i=1}^{n} w_i x_i \Bigg/ \sum_{i=1}^{n} w_i \tag{16}$$

with weights depending on the sample. We used the following ψ-function, known as Tukey's biweight:

$$\psi(x) = \begin{cases} x\left(1 - \left(\frac{x}{a}\right)^2\right)^2 & , \quad |x| \le a \\ 0 & , \quad |x| > a \end{cases} \tag{17}$$

e.g. with $a = 4$.

Scale invariant M-Estimators The solution to (13) is not scale-invariant, since in general $\psi(cx) \neq c\psi(x)$. In practice it means that an M-estimator should be supplemented by an estimator of scale. The scale invariant

version of the M-estimator is defined by the solution to the equation:

$$\sum_{i=1}^{n} \psi\left(\frac{x_i - \mu}{\sigma}\right) = 0. \tag{18}$$

This procedure is also called studentizing. Since σ is usually unknown it is replaced by an estimator of the scale. A widely used estimator for the scale is the standard deviation, however it is not robust. So a robust substitute for the scale must be found. Usually one takes the MAD (median absolute deviation) divided by 0.6745:

$$\tilde{\sigma} = \frac{\text{median}|x_i - \text{median}(x_i)|}{0.6745}. \tag{19}$$

The MAD is a robust estimator for $u_{0.75} \cdot \sigma = 0.6745\sigma$ ($u_{0.75}$ is the 0.75 quantile of the standard normal distribution). So MAD/0.6745 is a robust estimator for σ.) The breakdown point of this estimator is $\epsilon^* = 0.5$.

The theory of robust M-estimators for multivariate distributions with elliptically symmetric density function is studied by Maronna (Maronna 1976). We adapt the approach for location estimates to our needs (overlap and outlier handling) and therefore use a weighting scheme based on the deviation from the Gaussian model.

Estimating the Mean and Dispersion Matrix The M-estimate $\widehat{\boldsymbol{\mu}}$ for the location $\boldsymbol{\mu}$ is computed as

$$\widehat{\boldsymbol{\mu}} = \sum_{i=1}^{n} w_1(e_i) z_i \mathbf{p}_i \Big/ \sum_{i=1}^{n} w_1(e_i) z_i \tag{20}$$

where the weights are defined as

$$w_1(x) = \frac{\psi(x)}{x} \tag{21}$$

with Tukey's biweight (17) as the ψ - function and $a = 5$. The studentized error e_i between the data and the model for each point is

$$e_i := e_i(\widehat{a}, \widehat{b}, \widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\Sigma}}) := \frac{(z_i - \widehat{a}\ \mathrm{G}(\mathbf{p}_i, \widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\Sigma}}))}{\sigma} \tag{22}$$

with unknown spread σ. The estimate $\widehat{\boldsymbol{\Sigma}}$ of the dispersion matrix $\boldsymbol{\Sigma}$ is given by the weighted outer product as

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{T} \sum_{i=1}^{n} w_1(e_i)^2 z_i (\mathbf{p}_i - \widehat{\boldsymbol{\mu}})(\mathbf{p}_i - \widehat{\boldsymbol{\mu}})^T \tag{23}$$

with T as the total sum of the intensities of the patch (Eq. 5).

Estimating the Amplitude The M-estimate $\widehat{a}$ of the amplitude a is given by

$$\widehat{a} = \frac{\sum_{i=1}^{n} w_1(e_i) z_i \mathrm{G}(\mathbf{p}_i, \widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\Sigma}})}{\sum_{i=1}^{n} w_1(e_i) \mathrm{G}(\mathbf{p}_i, \widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\Sigma}})^2}. \tag{24}$$

with w_1 and e_i as defined in (21) and (22) respectively.

Computation of the Parameter Estimators In general, an M-estimator cannot be computed directly. An iteration scheme has to be used instead. The equations for the mean $\widehat{\boldsymbol{\mu}}$ (20), the dispersion matrix $\widehat{\boldsymbol{\Sigma}}$ (23) and the amplitude $\widehat{a}$ (24) can be solved by the weighted least square iteration:

$$\boldsymbol{\mu}_{j+1} = \sum_{i=1}^{n} w_1(e_{ij}) z_i \mathbf{p}_i \Big/ \sum_{i=1}^{n} w_1(e_{ij}) z_i \tag{25}$$

$$\widehat{\boldsymbol{\Sigma}}_{j+1} = \frac{1}{T} \sum_{i=1}^{n} w_1(e_{ij})^2 z_i (\mathbf{p}_i - \boldsymbol{\mu}_j)(\mathbf{p}_i - \boldsymbol{\mu}_j)^T \tag{26}$$

$$a_{j+1} = \frac{\sum_{i=1}^{n} w_1(e_{ij}) z_i \mathrm{G}(\mathbf{p}_i, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{\sum_{i=1}^{n} w_1(e_{ij}) \mathrm{G}(\mathbf{p}_i, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)^2} \tag{27}$$

with w_1 as defined in (21) and

$$e_{ij} = \frac{(z_i - a_j\ \mathrm{G}(\mathbf{p}_i, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j))}{\sigma_j} \tag{28}$$

and the MAD (19) adapted to the deviation from the Gaussian model as the estimate for the scale

$$\sigma_j = \text{median}_{i \in I^*} \frac{|a_j\ \mathrm{G}(\mathbf{p}_i, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) - z_i|}{0.6745}, \tag{29}$$

where $I^* = \{i \mid \mathrm{G}(\mathbf{p}_i, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) < c\}$, e.g. $c = 1.6 * u_{0.95}$ where $u_{0.95}$ denotes the 0.95 quantile of the standard normal distribution.

Managing Overlapping Spots

The problem when fitting models to overlapping spots is that they will be biased towards the overlapping neighbor. This will result in dislocated fitted models and a too high quantification. One possible method to tackle this problem is to correct the input intensities for a spot by subtracting overlapping neighboring models. However, usually too much is subtracted due to the overlapping situation, such that an iteration process between subtracting neighbor models and re-fit is needed. Another possible approach the usage of robust estimators. Intensities which are too high due the overlap are regarded as outliers and are subsequently down-weighted. In our paper we use a combination of the two schemes: In a first step a robust fit is performed, then the background estimation is improved and finally a robust re-fit is performed on the data with subtracted neighboring models.

Subtracting neighboring models Let $\mathcal{G}$ be the grid of spots defined as $\mathcal{G} = \{g_{ij} \mid i \in \{1, \ldots, I_\mathrm{G}\}, j \in \{1, \ldots, J_\mathrm{G}\}\}$. For each spot g_{ij} let us assume we have computed a spot model $\mathrm{Z}_{ij}(\mathbf{p}, \mathbf{q})$ with the parameter vector $\mathbf{q} \in \mathbb{R}^k$ and spot location $\mathbf{p} \in \mathbb{R}^2$. Consider the

image patch $S_{ij} = S_{ij}(\mathbf{p})$ for g_{ij}. In order to take into account overlapping spots we can recompute the model Z_{ij} by using the modified spot patch

$$S^*_{ij} = S_{ij} - \sum_{k,l\in\{-1,0,1\},(k,l)\neq(0,0)} \mathrm{Z}_{i+k\ j+l} \qquad (30)$$

i.e. subtracting neighboring spot models. We furthermore set the models $\mathrm{Z}_{ij} := 0$ for $i \in \{0, I_\mathrm{G}+1\} \vee j \in \{0, J_\mathrm{G}+1\}$ in oder to deal with the special cases of border points. One could iterate this procedure for every spot g_{ij} over the whole image. One then gradually obtains better models for every spot, stopping when the parameters of the model for each spot stabilize.

Semi-parametric Spot Fitting

A semi-parametric approach can describe the spot shape more accurately in the case of deviations from the model assumptions, which is the case in Fig. 4. However, overlap handling will be difficult, because a semi parametric fit will lack an intrinsic declension of the tails of a parametric model.

Algorithm

The basic idea of this method is to reduce dimensionality of given data using prior knowledge. Assuming that the spot has elliptically symmetric shape the fit is computed in the following steps:

A. Find the spot center We first perform a Gaussian fit computing M-estimators for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ as described in (20) and (23). The estimate $\widehat{\boldsymbol{\mu}}$ is the spot center. Since the M-estimator of the location is robust it will also deal with spots with uncommon shapes. Passing a line perpendicular to the x, y-plane through $\widehat{\boldsymbol{\mu}}$ gives us the axis $\mathbf{a}$.

B. Transform the points The estimated dispersion matrix $\widehat{\boldsymbol{\Sigma}}$ gives us an ellipse in the x, y-plane. Let e_1 and e_2 be the two eigenvalues of $\widehat{\boldsymbol{\Sigma}}$, (without loss of generality $e_1 \geq e_2$), $\mathbf{v}_1$ and $\mathbf{v}_2$ the corresponding eigenvectors and ϵ be the half-plane spanned by $\lambda_1\mathbf{a} + \lambda_2\mathbf{v}_1$; $\lambda_1 \in \mathbb{R}$, $\lambda_2 \in \mathbb{R}^+_0$. Consider the one parametric family of ellipses with the principle axis directions $\mathbf{v}_1$ and $\mathbf{v}_2$, and diameters λe_1 and λe_2, $\lambda \in \mathbb{R}^+_0$ and center $\boldsymbol{\mu}$. The family covers the x, y-plane without intersection, each point in the x, y-plane lies exactly on one ellipse. We "rotate" the given intensity points $(\mathbf{p}_i, z_i)$ following the path corresponding to $\mathbf{p}_i$ into the half-plane ϵ yielding a point cloud $ParamVector_i$ in 2-space (see Fig. 13f. The first coordinate can be easily computed by:

$$e_1 \cdot |\mathbf{p}|^2 / \sqrt{e_1^2(\mathbf{p}\cdot\mathbf{v}_1)^2 + e_2^2(1-(\mathbf{p}\cdot\mathbf{v}_1)^2)} \qquad (31)$$

and the second coordinate is the unchanged z-coordinate.

C. Compute a profile We introduce a simplified, efficient and robust version of curve approximation for scattered points suited to our purpose. First we compute m points $\mathbf{c}_i = (x_i, y_i)$, $i = 1, \ldots, m$ well describing the shape of curve to be computed. Consider the vertical parallel strip with y-axis and $x \equiv \max x_i$ as borders. We then segment the strip into m commensurate parallel strips and compute $\mathbf{c}_i = \mathrm{median}_k \mathbf{r}_{ki}$, where $\mathbf{r}_{ki}$ are those points $\mathbf{q}_k$ lying in the i^{th} strip, see Fig. 13f. We further cut off tails of the profile by gradually lowering the profile points down to zero in the last quarter, because 1) especially at the tail there may be some overlapping situation and 2) generally there are fewer points at the tail. For our purpose it is enough to interpolate the points $\mathbf{c}_i$ by a polygon and to perform a smoothing scheme on the profile points, e.g. by replacing each point with a weighted sum of its neighbors. Alternatively one can compute a spline interpolating the points $\mathbf{c}_i$ for the profile curve.

D. Compute Volume The profile curve is rotated following the elliptical paths as in step B. The brightness V of the spot is then estimated by taking

$$\widehat{V} = \frac{e_2}{e_1} \cdot \frac{1}{3} \sum_{i=2}^{m} (x_{i-1}^2 + x_{i-1}x_i + x_i^2)\pi(y_i - y_{i-1}). \quad (32)$$

In order to yield good results one should use known numerical integration schemes as (composite) Simpson's rule.

Relative Error and Goodness-of-Fit

In order to quantitatively assess how well the (Gaussian) model assumption holds for a given set of n intensities z_i belonging to a spot we introduce a measure for the error. We apply an approach also used in linear regression analysis as in (Hartung 1989) by comparing the model to a "standard model Z_0":

$$T_1 := \frac{1}{n}\sum_{i=1}^{n}(z_i - \mathrm{Z}(\mathbf{p}_i))^2 \Big/ \frac{1}{n}\sum_{i=1}^{n}(z_i - z_0)^2 \qquad (33)$$

with $z_0 := \frac{1}{n}\sum z_i$ as the mean of the given intensity values. The standard model Z_0 in this case is a plane parallel to the image plane at the height z_0, i.e. $\mathrm{Z}_0 \equiv z_0$. T_1 relates the mean squared error between the Gaussian model and the data to the mean squared error between a constant model and the data. T_1 can also be regarded as the mean squared error between the Gaussian model and data normalized by the variance of the error. We will call T_1 the *relative squared error* or for short *relative error*. In the literature $1 - T_1$ is called the *goodness-of-fit*.

Spot Detection Limit

A spot fitting algorithm should decide whether a location contains a spot before performing a fit. Imagine having input intensities with perfect zero values, computing the mean would lead to a division by zero or

leading to a singular dispersion matrix. This can happen rather often since the first background estimation is overestimating the background.

One could use our test for goodness-of-fit as spot detection by testing the "Zeromodel" $Z_{Zero} \equiv 0$ being 'd-appropriate' or not, using the test statistic:

$$T_1 := d^2 \cdot \frac{\sum_{i=1}^{n} z_i^2}{\sum_{i=1}^{n} (z_i - z_0)^2} \tag{34}$$

e.g. $d^2 = 2$. However, this measure is non-robust. We use instead

$$T_2 := \text{median}(z_i) > d \tag{35}$$

for spot detection where $d = \log(2) \cdot V^*/n$ and V^* is the minimum volume a location carries where a spot still can be expected. The interpretation is that if the volume of a location V is greater than V^*, we expect that there is a spot. The easiest way to estimate the volume is $n \cdot \sum z_i$, leading to

$$T_{2a} := \sum z_i > V^*/n. \tag{36}$$

In order to overcome noise and outliers for example due to overlaps we use the median. Assuming that z_i is exponentially distributed which comes close to our situation, the $\log(2) \approx 0.6931$ times the median estimates the mean. Replacing the mean by the median yields T_2.

Spot Fitting Algorithm Overview

Fig. 6 shows an overview of the spot fitting algorithm. After the grid fitting the first pass of the background estimation takes place. After the background estimation a spot test based on (36) is performed for every spot location . In case of a hybridized spot the parameters of a Gaussian spot model (2) are fit to the intensity data with M-estimators. After the second pass of the background estimation the neighborhood models are subtracted in order to cope with overlap. If the subsequent robust Gaussian fit has a high relative error (33) a semi-parametric fit is performed. Finally, the volume of the spot model is computed.

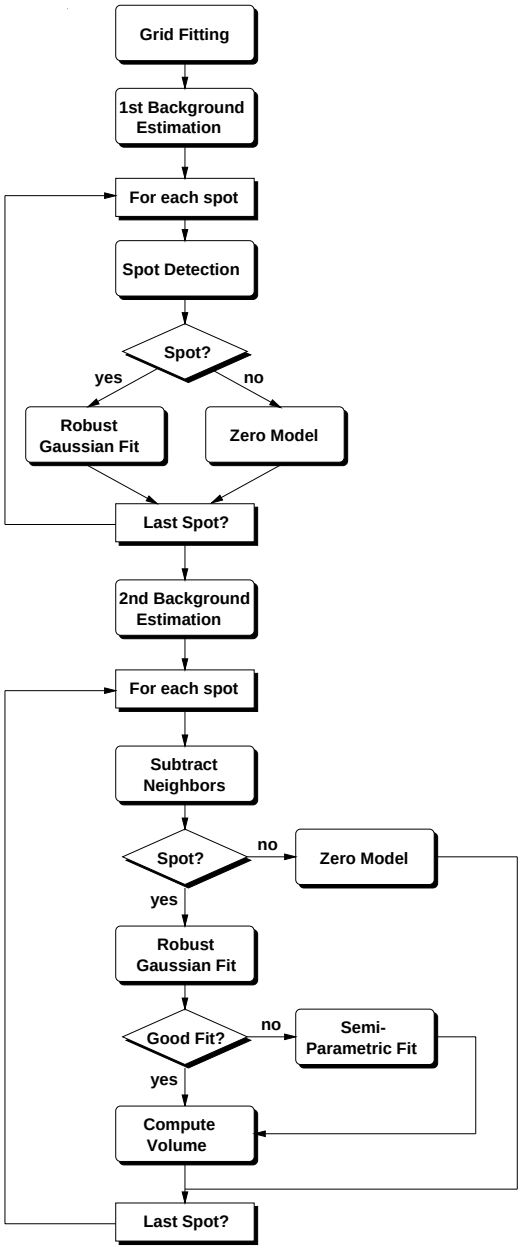


Figure 6: Spot Fitting Overview

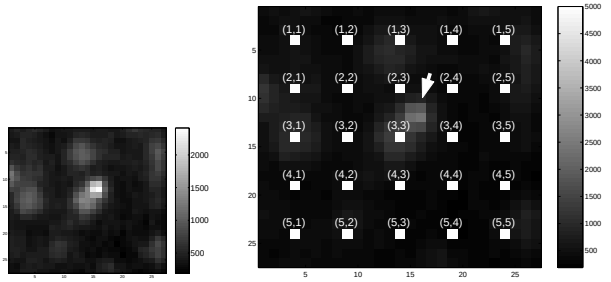
Experimental Results

Artifacts

Consider the patch in Fig. 7a. The prior spot locations after the grid fitting are shown in Fig. 7b. The spot $(3, 3)$ in the center is distorted by an artifact. As can be seen in Fig. 8a, a simple Gaussian fit will fail, because the location is biased towards the location of the artifact. The robust Gaussian fit can overcome the outlier. Figure 8b shows the result after 6 weighted least squares iterations. The label "vol." denotes the volume of the spot and "qvol." denotes the volume with squared intensities.

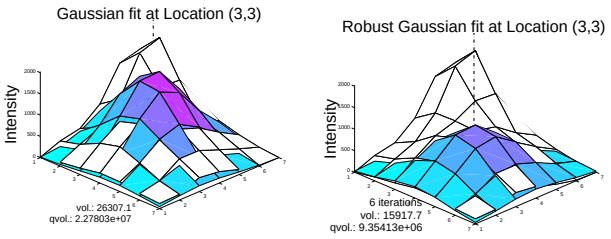
Overlapping Spots

We demonstrate how the robust Gaussian fit works on image data with overlapping spots. Figure 9a shows a 5×5 block originating from an ONF image with low resolution. Figure 9b shows the prior spot locations after the grid fitting. Before a fit is performed a spot detection limit as introduced in (35) is computed with limit $V^* = 30000$ corresponding to $d = 400$. In Fig. 10a the white marks indicate the detected spots. Fig. 10b shows a 3D plot of the block. Almost all locations are classified correctly including location $(5, 1)$, where a neighboring spot is interfering from the left. Location $(2, 1)$ is falsly



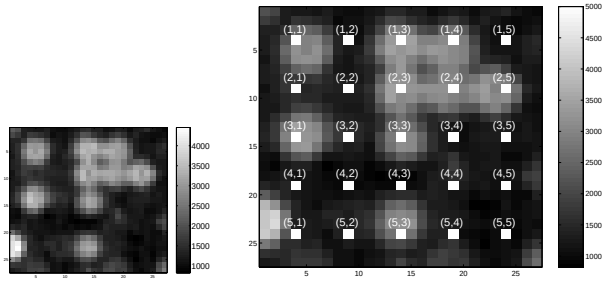
(a) Image data (b) Prior spot locations

Figure 7: Given patch with artifact



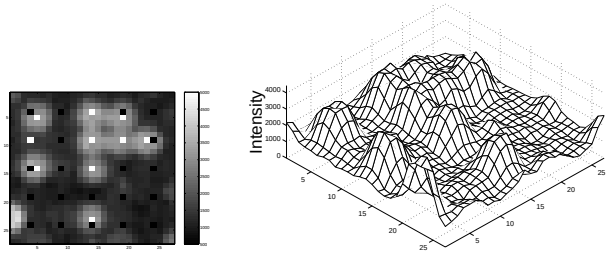
(a) (b)

Figure 8: Dealing with artifacts in Fig.7



(a) Intensities (b) Location Initializations

Figure 9: Block with overlapping spots



(a) (b)

Figure 10: Detected spots and 3D plot of image data

detected as a spot, because two neighboring spots are overlapping. Spot (3,1) is an ordinary spot with no interfering neighbors, the robust estimator stops after 3 iterations without any big changes.

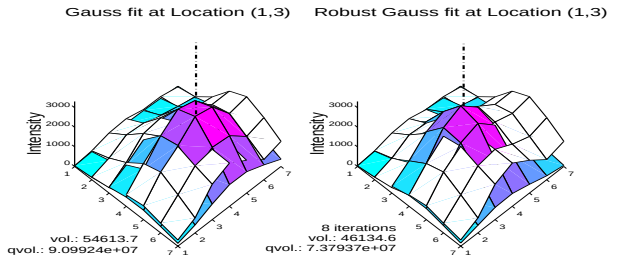
Spots $(1,3)$, $(2,5)$ and $(3,3)$ have up to three overlapping neighbors, here the robust estimator can recover the original spot location quite well, especially for $(1,3)$ and $(3,3)$. Spot $(1,3)$ is plotted in Fig. 11a. The non-robust Gaussian fit is biased towards the neighboring spots, whereas the location of robustly fitted Gaussian spot is more plausible. Spots $(1,4)$, $(2,3)$ and $(2,4)$ have over four overlapping neighbors and are therefore difficult cases, but still some improvements can be done. The non-robust and robust Gaussian fits are plotted in Fig. 11b.

After the first robust Gauss fit we refit on every location with subtracted neighborhood models. The centers computed during the first fit are taken as the a priori centers for the second fit. When taking a look at the new patches with subtracted neighbors (see Fig. 12a) one will notice that the patches are now less distorted than the previous patch and are more "spot like" – an indication that the situation has improved.

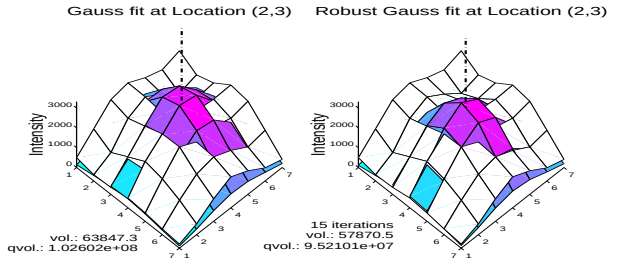
When investigating the goodness of fit and the patch shapes, the first robust fitting resolved the overlaps at spos $(1,3)$ (see Fig. 12a) and $(3,3)$ very well. The results for the spots (1,4) and (2,5) are good, the results for (2,3) (see Fig. 12b) are acceptable, and the results for (2,4) are not good enough. Generally, on can say that the robust estimation will perform well up to four overlapping neighbors while more than four will make problems. This is can be explained by the fact that highest possible breakdown point of a robust estimator is $\epsilon^* = 0.5$. If more than 50% of the input datas are false the situation cannot be recovered directly by a robust estimator. An overview of the fitted models can be seen in Fig. 12c.

Uncommon Shapes

Figures 13a and b show a volcano spot with an overlap from the right hand side. An ordinary Gaussian fit would be biased to the right neighbor, but a robust estimator recovers the location easily (Fig. 13c). Performing a robust Gauss fit on both sides we subtract the neighborhood spot model from the patch receiving the corrected data (see Fig. 13d). After a Gaussian refit the initial volume estimation can be observed in Fig. 13e but the estimated volume is not very reliable due to the high relative error rate. Using the center and dispersion we performed a semi parametric fit (see Fig. 13f). We smoothed the profile points by replac-

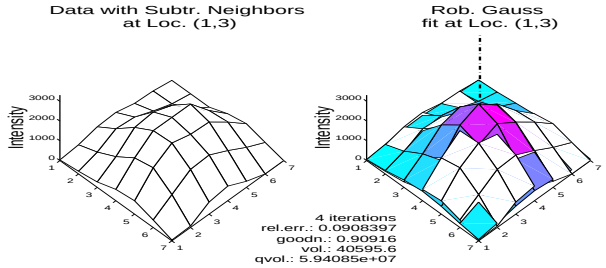


(a) Initial spot fitting for spot (1,3)

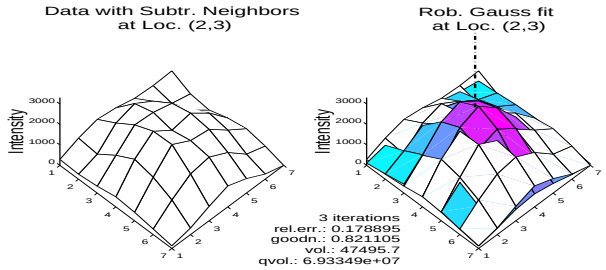


(b) Initial spot fitting for spot (2,3)

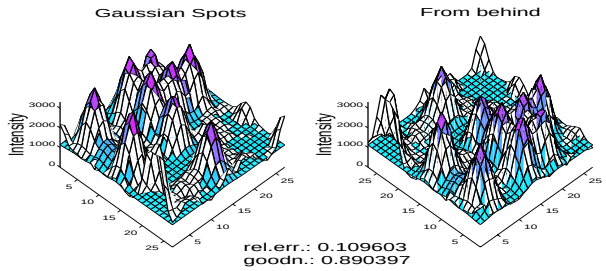
Figure 11: Initial non-robust and robust Gaussian spot fitting



(a) After subtraction of neighborhood models of spot (1,3)



(b) After subtraction of neighborhood models of spot (2,3)



(c) Models of all detected spots

Figure 12: Detected spots and 3D plot of image data

ing each point (except at the border) with the weighted sum over the left, the point itself and right neighbor with the weights 3,6, and 2. The left neighbor received higher weights, because the points on the left hand side are more reliable since they are closer to the center. The goodness of fit improved and a more reliable quantification is done. We also compared the algorithms to each other by plotting the percentage of data covered by the strip with the two offset profile curves as borders yielding a performance curve, see Fig. 14. A quick ascending curve indicates that the method is performing well, because the data points are covered early. As one can see the semi-parametric fit is better than the Gaussian fit.

Entire Image

We demonstrate the result of the spot fitting for the image in Fig. 2 containg a total of $40 \times 60 = 240$ spots. After the grid fitting we have the prior locations of every spot and apply the first background estimation routine to be ready for the first run. A major issue is the detection limit. It determines whether the location possibly contains a spot of interest. Pretending we do not know much about the volume of a spot we set $V^* = 0$ using the detection limit Eqn. 35. The algorithm will then fit at every location. After a second backgound estimation and a second run the fits from the first run are refined. The reconstructed image can be seen in Fig. 15. An analysis showed that most of the spots have a volume between 10000 and 20000 – locations with a volume lower than 10000 possibly contain no spots. With this knowledge we set V^* to 10000 and rerun our program. This time no singular locations are detected. In Fig. 16 we plotted the relative error for each spot location. Some locations have a significant error over 1.0. This is due to a bad fit as the result of fitting a model to location containing no spot. A location without a spot can still pass the detection limit when neighboring spots are interfering. We therefore perform a post processing procedure by simply rejecting a model with too high error, e.g. relative error greater that 0.4. In general, a relative error smaller than 0.1 indicates a very good, smaller than 0.2 a good fit, while at values bigger than

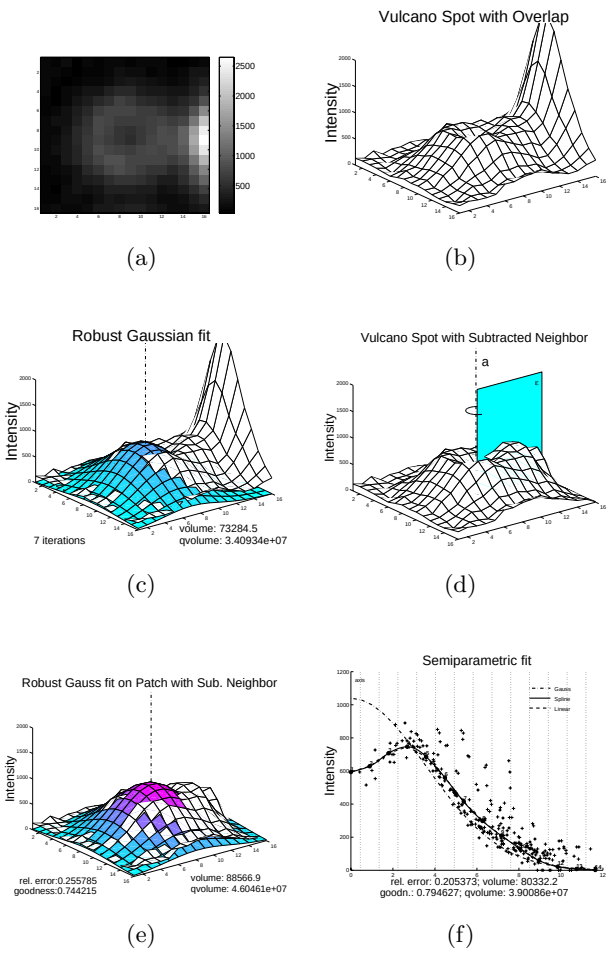


Figure 13: Volcano Spot with Overlapping Neighbor

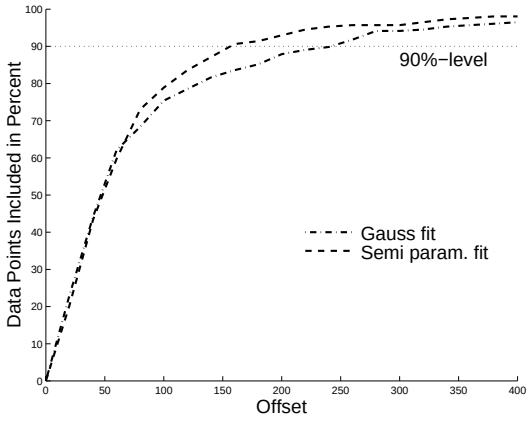


Figure 14: Semi-parametric fit

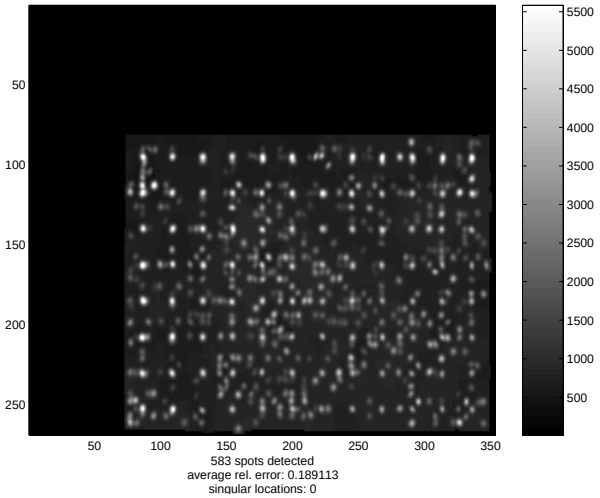


Figure 15: Reconstruction of the image in Fig.2

0.4 or 0.5 the fit should be rejected.

Complexity

Table 1 shows the CPU-time costs for each method per fit in flops (the methods have been implemented in Matlab™). The values should be interpreted as follows:

Resolution →	Low Res. 7x7	High Res. 16x16
Method ↓	flops/per fit	flops/per fit
Gaussian Fit	10.000	47.000
Semi-param. Fit	2.000	15.000

Table 1: CPU-time in Flops

1. A (non robust) Gaussian fit in low resolution requires approximately 10.000 flops.
2. A robust Gaussian fit with k iterations requires approximately $(k+1) \times 10.000$ flops (1 fit for the initial guess and k remaining fits for each iteration).
3. A semi-parametric fit with 5 "profile points" costs 2.000 flops in low resolution, while in high resolution 14 "profile points" are computed requiring 15.000 flops.
4. A single semi-parametric fit is approximately four times faster than a Gaussian fit in low and high resolution. However, one should keep in mind that a semi-parametric fit in general can not be performed directly without any preceding center search by a M-estimator of location.
5. Let $n \times n$ be the dimension of the input patch, i.e. $n = 7/n = 16$ for low/high resolution. While the computing time for the Gaussian fit will increase with $O(n^2)$, the computing time for a semi-parametric fit will increase with $O(n^2 \cdot \log(n))$. The reason is that a Gaussian fit basically sums over all data points while sorting algorithms are needed for a semi-parametric fit.
6. An already implemented C-version of the non-robust

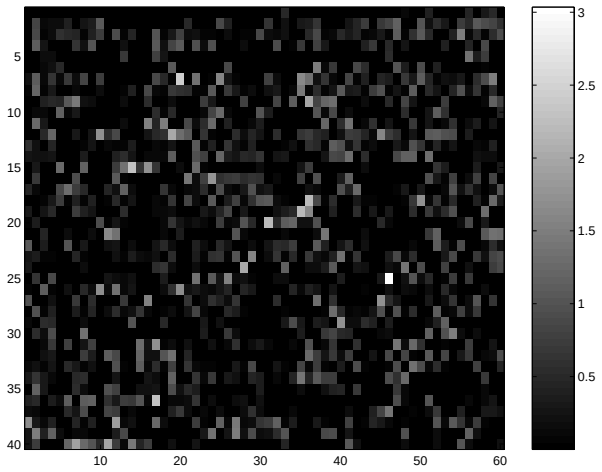


Figure 16: Relative Error of the 40×60 Spots

spot fitting with subtracting neighbors for all spots needs about four minutes on the same machine (including the grid fitting).

Conclusion

The basic problems in spot fitting are overlapping and non Gaussian spots. Overlaps with up to three or four neighbors can be reliably solved by robust fitting and subtracting neighboring models with a subsequent refit. For overlapping situations with more than four neighbors too few consistent data is available for robust estimators. One should remember that the highest possible breakdown point of a robust estimator is $\epsilon^* = 0.5$. In such a case one should avoid any fitting and assign a "standard spot model" to the spot location and do the first fit after subtracting the neighbors. Such a "standard spot model" can be computed by first estimating the center by M-estimation of location, second taking a 'standard' dispersion matrix (the spots are approximately of equal size) and estimating the amplitude by least square.

Outlook

We provide the following suggestions for the future work:

Finding alternative measures for goodness of fit and detection limit We are rather satisfied with the introduced measures for goodness of fit and detection limit. However, one may find an alternative approach by constructing other statistics or using a (maximum) entropy method for detection.

Constructing confidence intervals for parameters In statistics it is common to give confidence intervals, ellipses, etc. for the parameters or even for the model. For our problem it would be useful not only to have confidence intervals for the parameters but also for the volume. Assuming a given center or a perfectly determined center a confidence interval for the volume can be directly constructed from a confidence interval for the amplitude and dispersion matrix.

Developing machine learning algorithms When analyzing a ONF-library the computer computes over 4600 million fits. However, the computer does not learn what is a good fit or what is a spot. It does not learn that a certain volume estimation cannot be possible. The computer should adapt to new conditions and be more fault-tolerant. Robust estimation, detection limit, fit acceptance depend on parameters the prior choice of which may not stay optimal from image to image, from library to library or even from experiment to experiment. Furthermore, the computer could develop some heuristics like: the Gaussian fit always overestimates the volume by 10%.

References

Bishop, C. M. 1995. *Neural Networks for Pattern Recognition*. Clarendon Press.

Brändle, N.; Lapp, H.; and Bischof, H. 1999. Automatic Grid Fitting for Genetic Spot Array Images Containing Guide Spots. In *8th Intl. Conf. on Computer Analysis of Images and Patterns, Ljubljana, Slovenia, September 1–3*, 357–366.

Haralick, R. M.; ; and Shapiro, L. G. 1991. Glossary of computer vision terms. *Pattern Recognition* 24:69–93.

Hartung, J. 1989. *Multivariate Statistik*. R. Oldenburg Verlag München Wien.

Huber, P. J. 1981. *Robust Statistics*. John Wiley and Sons.

Jolion, J.-M., and Rosenfeld, A. 1994. *A Pyramid Framework for Early Vision*. Kluwer.

Maronna, R. A. 1976. Robust M-estimators of multivariate location and scatter. *The Annals of Statistics* 4(1):51–67.

Meier-Ewert, S.; Maier, E.; Ahmadi, A. R.; Curtis, J.; and Lehrach, H. 1993. An automated approach to generating expressed sequence catalogues. *Nature* 361:375–376.

Noordmans, H. J., and Smeulders, A. W. M. 1998. Detection and Characterization of Isolated and Overlapping Spots. *Computer Vision and Image Understanding* 70(1):23–35.

Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; and Flannery, B. P. 1992. *Numerical Recipes in C*. Cambridge University Press.

Rousseeuw, P. J., and Leroy, A. M. 1987. *Robust Regression & Outlier Detection*. John Wiley & Sons.