BACHELOR

Objective noise characterization

Richardson, O.M.

Award date:
2014

Link to publication
Abstract

The noise in an electron microscope (EM) image is determined by the dose of electrons, the detection efficiency of the scattered electrons and the properties of the object under the microscope. At present, the amount of noise in EM images is often determined manually. Unfortunately, this method has proven to be inadequate for certain types of images and it does not suffice when the noise should be detected automatically. Another method is based on taking multiple images and obtaining for each pixel both the average brightness and its variance, but this method is not able to deal with images having different points of origin, caused by objects slightly drifting while the images are created. In this bachelor thesis, the standard method of dealing with noise is analysed, and a different method is developed using repeated linear interpolation and by analysing proper bin handling. This method especially yields positive results for images with a few but major brightness transitions. The developed method is illustrated on two image sets with different complexities. In addition, this thesis also supplies a mathematical handle for describing images of drifting objects.
## Contents

1 Introduction \hfill \hfill 3
2 Modelling \hfill \hfill 3
3 Assumptions \hfill \hfill 5
4 Test Image Sets \hfill \hfill 6
5 Scaling the Registered Brightness \hfill \hfill 8
6 Estimating the Amount of Noise \hfill \hfill 10
   6.1 Estimating Poisson Distribution Parameter \hfill \hfill 10
   6.2 Filling the Bins \hfill \hfill 11
   6.3 Numerical Results of Set M1 \hfill \hfill 12
   6.4 Numerical Results of Set L1 \hfill \hfill 13
7 Surface Drift \hfill \hfill 14
   7.1 Drift-corrected Average \hfill \hfill 15
   7.2 Drift-correction Algorithm \hfill \hfill 17
   7.3 Validation of Drift-correction algorithm \hfill \hfill 17
   7.4 Numerical Results on Set L2 \hfill \hfill 18
   7.5 Numerical results on Set M2 \hfill \hfill 19
8 Discussion/Future Research \hfill \hfill 20
A Shot Noise \hfill \hfill 21
B MATLAB commands used \hfill \hfill 21
Acknowledgements

I would like to thank Eric Bosch and Seyno Sluyterman for providing me with relevant data and background knowledge for my research, as well as taking the time to discuss my ideas. I would also like to thank Martijn Anthonissen for the feedback on the final version of my bachelor thesis.

I would like to express my special gratitude to my bachelor thesis supervisor Dr. Jos Maubach, for all the patience and time it took to convert my wild ideas into (more) comprehensive theories.
1 Introduction

It is important that images from an electron microscope are as accurate as possible. However, EM images suffer from noise, which is the difference between the perfect image and the actually registered image. The noise in an electron microscope image (EM image) is determined by the dose of electrons, the detection efficiency of the sensor and sample properties. In this thesis we focus on sensor-related noise only. At present, noise in EM images is often characterized using manual inspection of areas with supposedly uniform material composition. However, not all images contain such areas of sufficiently large sizes and in general one cannot a priori distinguish between noise and sample information. For instance, for biological samples there is often no correlation between the brightness in one pixel and a neighboring pixel, because of the many small irregular structures present. An important part of the noise can be classified as shot noise. This type of noise is caused by a nonuniform distribution of electrons. As a result, surfaces with the same brightness will look slightly different and grainy in pictures. Noise in images is a ‘fact of life’, it is impossible to prevent it. However, it is possible to analyse the level of noise by comparing multiple images of the same sample. It is important to know the approximate level of noise so the results of the image can be better interpreted.

The basic method to determine the level of present noise is based on taking multiple images of the same specimen under the EM, binning all pixels with equal average brightness values, and calculating the variance of all those pixels as a function of their average brightness. This allows obtaining for each pixel not only the average brightness, but also its variance between images. The slope of the variance can then be used as a noise measure. This approach usually works, albeit that objects tend to drift while under a microscope. This causes problems in binning the correct pixels. The consequence for binning the wrong pixels is an unrepresentatively high variance, which toughens the estimation and the interpretation of the variance slope. This thesis will analyse both the basic method and develop an additional method which is better suited in case of drifting objects.

2 Modelling

In this section we describe the modelling of perfect images, registered images, noise and drift. The modelling choices are primarily made to facilitate the calculations on images, in order to compare the computed variance with the predicted variance.

All images are defined on area $S^* \subset \mathbb{R}^2$. Let $S_k$ be the registered part of $S^*$ at moment $t_k$ ($k = 1, ..., K$). At any given time $t_k$, an image has a two-dimensional representation called observed brightness, expressed in numerical values between 0 and $2^p - 1$, which is the function $\Phi_k$ over its area $S_k \subset S^* \subset \mathbb{R}^2$:

$$\Phi_k : S_k \rightarrow [0, 2^p - 1].$$

(1)

We will model every image to be a $m \times n$ matrix, where every entry assumes the average observed brightness of the corresponding area (computed by integrating $\Phi_k$ over the area). This way each image is defined as a grid of pixels. We define $S_k$ as

$$S_k = [1, m] \times [1, n] + (x_k, y_k) \subset S^*,$$

(2)

---

1 The choice for $p$ depends on the number of bits in the image format, further explained in Section 5.
where \( m \times n \) represents the resolution of all the images \( P_k \) and \((x_k, y_k) \in \mathbb{R}^2\) represents the offset of \( S_k \) with respect to \( S^* \) at time \( t_k \). Defining \( S_k \) this way will justify the use of matrices to represent an image and facilitates the mathematical definition of a surface drift. The part of the area corresponding with the point in row \( i \) and column \( j \) of (the matrix of) image \( P_k \) is denoted by \( s^k_{i,j} \) and will be referred to as a pixel area. The coordinates \((i, j)\) will be referred to as a pixel as pictured in Figure 1. So by definition,

\[
S_k = \bigcup_{i=1,j=1}^{i=m,j=n} s^k_{i,j},
\]

where the square unit-areas

\[
s^k_{i,j} = [j - 1, j] \times [m - i + 1, m - i] + (x_k, y_k)
\]

are called pixel areas. The reason for this model approach: In practice the specimen under the EM can drift with time, requiring the offset \((x_k, y_k)\). Furthermore, the images are registered with a digital sensor, which outputs pixel values \( P_k(i, j) \) based on the amount of collisions of electrons into the small square surfaces \( s^k_{i,j} \).

A perfect image is a very high resolution matrix of brightness values. A registered image is a lower resolution image approximating the perfect image. The definition for an \( m^* \times n^* \) perfect image with \( m^* \gg m \) and \( n^* \gg n \):

\[
P^*_k(i, j) = \int_{s^*_k} \Phi_k(x, y) dxdy \quad \forall i \in \{1, \ldots, m^*\}, \quad j \in \{1, \ldots, n^*\}.
\]

(3)

In this definition, \( s^*_k \) is a sufficiently small area. In an attempt to capture the observed brightness and produce the perfect image, a microscope (with finite precision) produces a sequence of \( K \) images \( P_1, \ldots, P_K \) with timestamps \( t_1 < t_2 < \ldots < t_K \). Now we are able to give a definition for a registered image matrix \( P_k \):

\[
P_k(i, j) = \int_{s^k_{i,j}} \Phi_k(x, y) dxdy \quad \forall i \in \{1, \ldots, m\}, \quad j \in \{1, \ldots, n\}.
\]

(4)

Figure 1: \( P_k \) with pixels \((i, j)\) corresponding with area \( s^k_{i,j} \).
In addition, to have a definition of $P_k$ as a function, we define $\phi_k$ to be the piecewise constant image function corresponding to the image matrix $P_k$:

$$\phi_k(x, y) := P_k(i, j) \iff (x, y) \in s^k_{i,j}.$$  

(5)

The function representation of an image will turn out to be more convenient when analysing surface drifts.

To determine the brightness value of pixel $(i, j)$, the sensor "counts" the number of electrons that hit $s^k_{i,j}$ during a small period of time called the exposure time. These electrons arrive according to a Poisson process, with the consequence that for different images with the same exposure time, the electron count usually differs. The resulting difference in brightness between images is called shot noise. A formal definition follows. Define $\bar{\Phi}$ as the average of the functions $\phi_k$:

$$\bar{\phi}(x, y) = \frac{1}{K} \sum_{k=1}^{K} \phi_k(x, y).$$  

(6)

Then shot noise is defined to be the variance of the observed brightness of all images $(P_k(i, j))_{k=1}^{K}$. Because of the nature of EM imaging, the registered brightness in pixel $(i, j)$ is stochastic rather than deterministic. Statistically speaking, $P_k(i, j)$ is a scaled observed value from a Poisson distribution\footnote{The source of notation and definitions regarding probability can be found in the bibliography [1]. For more information see Section A.} with equally scaled parameter $\bar{P}(i, j)$, so for a scale factor $c$:

$$\frac{P_k(i, j)}{c} \sim \text{POI} \left( \frac{\bar{P}(i, j)}{c} \right).$$  

(7)

Because the noise is of a stochastic nature, its level cannot be exactly determined based on one image. Using multiple images, it is possible to compare the difference in observed brightness and estimate its distribution. By analysing the distribution, we can measure the level of noise. The noise analysis will be performed on the area which all the images have in common: the intersection

$$S = \bigcap_{k} S_k$$  

(8)

as illustrated in Figure 2.

3 Assumptions

To estimate the noise, it is assumed that

1. All noise in the registered images $P_k$ is shot noise.
2. The exposure time is equal for all images.
3. The amount of noise can still be accurately determined after cropping the image by a few pixels, i.e., we assume that the boundaries of the images do not contain critical information.
4. Every registered image in the set has the same $m \times n$ dimension.
5. The surface drift is a translation. Rotations and stigmatisms are excluded.

6. For the sake of convenience there is no larger than pixel-size drift, since any drift greater than a pixel can be reduced to a sub-pixel drift simply by moving the image. This implies that \(x_k, y_k \in (-1, 1)\) for all \(k = 1, ..., K\).

7. All images are grayscale images.

4 Test Image Sets

The current cases of interest are those where the brightness in the images only attain a discrete small amount of values, i.e., \(\{P_k(i, j) k = 1, ..., K, i = 1, ..., m, j = 1, ..., n\}\) is discrete and small. To illustrate the algorithms, we will distinguish between the amount of transitions. We use Set L1 (Figure 3) to analyse binning for images with a low number of transitions. This image set was actually registered by an EM and illustrates the need of a different noise analysis method in Section 7. This set has three distinct brightness values and only a few transitions. The set consists of 5 images with a resolution of 1024 \(\times\) 1024 with a 16-bit format. The dimension parameters are \(m = n = 1024\) and bit size \(p = 16\).

Set L2 is a set of images made by the researcher with the goal to imitate Set L1. The benefit of this imitation is the full control and knowledge regarding the drift and the noise of the images. This set consists of 6 images with an 8-bit format and a resolution of 815 \(\times\) 611. Note that this means the dimension parameters are \(m = 611\) and \(n = 815\) (and bit size \(p = 8\)). This image set also contains three different brightness values and a few transitions.

Set M1 (Figure 5) is an image set with more brightness values and more transitions. This set is used to validate binning Algorithm 1 and 2. This set has 5 images with a resolution of 800 \(\times\) 600, corresponding with dimension parameters \(m = 600, n = 800\) and bit size \(p = 8\).
Set M2 (Figure 6) is an image set containing less distinct brightness values than Set M1, but more transitions than the other sets. This set is used to validate the drift-correcting Algorithm 3 for an image of higher complexity. This set has 6 images with dimension parameters \( m = 815 \times 611 \) and bit size \( p = 8 \). Reason not to use Set M1 for validating Algorithm 3 is the lack of a (high resolution) perfect image.

Figure 3: An image from Set L1, Deviations: surface drift, shot noise.

Figure 4: An image from Set L2. Deviations: surface drift, shot noise.
5 Scaling the Registered Brightness

Note that an image is only an approximation of an area, and two important aspects of that fact are scaling and clipping.

The brightness level of area $s_{i,j}^k$ is registered as an integer value. The brightness of an image is set such that without an electron beam, the brightness is zero. The maximum brightness depends on the sample, and the intensity of the electron beam aimed at the sample. The latter is not specified for this thesis. We assume that the minimum attainable value is 0. The maximum brightness value is set to $2^p - 1$, where $p$ represents the amount of bits used in the image format.

We will distinguish between the case in which $p$ is known and in which $p$ is unknown. If $p$ is known, we can compute the maximum brightness value without any issues. If $p$ is unknown,
we will have to estimate the maximum integer value attainable (see Section 6.2) and assume that it is of the form \(2^p - 1\). That is, we assume that all brightness values are registered with \(p\)-bit accuracy. This digitalization also causes a small level of noise, but the effects are negligible in comparison to the sensor noise.

For example the images in Set L1 are in 16-bit format. This means that all brightness values of the images should be integer values between 0 and \(2^{16} - 1 = 65535\). Upon analysing the brightness values in one of the images in Set L1, we see the image attains a minimum brightness of 1605 and a maximum brightness of 53516.

When \(P_k\) is being examined, it is important that its brightness values fall between the brightness minimum and maximum of the perfect image \(P^*\). If not, clipping can occur, meaning that brightness values below the minimum (or above the maximum) are registered as the minimum (or maximum) brightness. Because a whole range of clipped brightness values are clipped to the same value, shot noise in the registered image \(P_k\) at the maximum attained brightness value of \(P^*\) will not exceed the maximum of \(2^p - 1\). This is why clipping must be prevented when registering and analysing an image. Using the same image from Set L1 as an example, we see clipping most likely does not occur since the minimum brightness attained is higher than 0, and the maximum brightness attained is lower than 65535.

Prior to any calculations, the brightness is rescaled from its integer \(p\)-bit range to a \([0,1)\) scale. If \(p\) is known, this is done by dividing all brightness values \(P_k(i,j)\) for all \(k = 1, ..., K, i = 1, ..., m, j = 1, ..., n\) by \(2^p\). If \(p\) is unknown, first an educated guess is made by finding the highest brightness value in the image, and by choosing \(p\) to be the smallest integer such that \(\max_{i,j,k} P_k(i,j) \leq 2^p - 1\). More precise:

\[
p = \lceil \log_2 \max_{i,j,k} P_k(i,j) \rceil,
\]

which defines the maximum brightness value to be

\[
L_{\text{max}} = 2^p - 1.
\]

This estimate for \(p\) will only be correct if the highest brightness value in the image is at least half of the brightness maximum.

The images from Set M1 are chosen in such a way that a lot of the brightness spectrum is covered. This implies that we cannot guarantee clipping will not occur, but because the shot noise is artificially added, the impact of clipping is only visible for the higher brightness values.

When being processed in MATLAB, we choose to scale the brightness values to fit between 0 and 1, making the image continuous again. This enables operations like averaging and interpolation. Below is given the definition of an average image.

Say an image set contains \(K\) images, \(P_1, ..., P_K\). Then the average image \(\bar{P}\) is defined to be

\[
\bar{P} = \frac{1}{K} \sum_{k=1}^{K} P_k.
\]

9
6 Estimating the Amount of Noise

6.1 Estimating Poisson Distribution Parameter

Recall the number of electrons that hit a pixel area \( s_{i,j}^k \) is Poisson distributed. This makes sense according to the requirements for a Poisson process, as illustrated in Section A. If we regard the pixel brightness values of the average image \( \bar{P}(i,j) \) as an estimate of the number of electron collisions, recall from (7) that the pixel brightness values of the images \( P_k(i,j) \) for some \( c \in \mathbb{R} \) have the following distribution:

\[
\frac{P_k(i,j)}{c} \sim \text{POI} \left( \frac{\bar{P}(i,j)}{c} \right).
\]

The shot noise is the variance in the registered images, so one needs to be able to estimate the number of electrons that hit \( s_{i,j}^k \) during the exposure time. Recall that the brightness values are scaled to \( p \)-bit values. This means that instead of the number of electrons, we are given the pixel brightness \( P_k(i,j) \), which we assume to be linearly proportional to the number of electrons that hit pixel \( s_{i,j}^k \).

Let \( X \) be the amount of electrons that hit \( s_{i,j}^k \). Then we assume \( cX = P_k(i,j) \). This knowledge can be used to determine the rescaling. \( X \) is Poisson distributed with parameter \( \lambda = \bar{P}(i,j)/c \), so we know

\[
\mathbb{E}(X) = \text{Var}(X) = \lambda.
\]

Our sample is scaled, but otherwise Poisson distributed:

\[
\mathbb{E} \left( \frac{P_k(i,j)}{c} \right) = \text{Var} \left( \frac{P_k(i,j)}{c} \right),
\]

which is equivalent to

\[
\mathbb{E}(P_k(i,j)) \cdot \frac{1}{c^2} = \text{Var}(P_k(i,j)) \cdot \frac{1}{c^2},
\]

so \( c \) is given by

\[
(12) \quad c = \frac{\text{Var}(P_k(i,j))}{\mathbb{E}(P_k(i,j))}.
\]

With the scale factor, we can calculate distribution parameter \( \lambda \) for every observed sample \( (P_k(i,j))_{k=1}^K \) by simply dividing the average brightness of a pixel by the scale factor:

\[
(13) \quad \lambda = \mathbb{E}(X) = \frac{\bar{P}(i,j)}{c}.
\]

This could be done (but is not) for every brightness value \( \bar{P}(i,j) \) and for all \( i,j \). This is not desirable because (1) the amount of distinct average brightness values \( \bar{P}(i,j) \) can be large and (2) because there are too few approximations \( P_k(i,j) \) to \( \bar{P}(i,j) \) to be statistically relevant. Therefore we will instead work with disjoint interval of brightness values, bins. The easiest way to do this is to establish an initial bin partition, which constitutes an iterative procedure; the proper bin size is not known beforehand, but it is possible to estimate the uniformly partitioned bin size and then calculate the variance related to each bin. From there the scale factor can be calculated for a more ideal number of bins. The size of each bin has a big impact on the obtained variance and consequently, on the estimated amount of shot noise. That is
why it is important to compute a proper bin size. In the end, the ideal bin size has a bin for every integer average number of electron collisions. The procedure of creating the bins will now be explained.

6.2 Filling the Bins

The basic way to compose the bins is to register the multiple EM images $\phi_k$ on area $S_k$, and to compare them to the average image $\bar{P}$.

Binning is the process of creating a partition of the interval of attained brightness values, into a finite amount of disjoint intervals. Say that from our image set, we want to group the average brightness values in $B$ bins, each with a size of $\frac{1}{B}$. Thus every $I_b$ (with $b = 1, ..., B$) will contain average brightness values in

$$I_b = \left( \frac{b - 1}{B}, \frac{b}{B} \right).$$

Then we create the distribution $D_b$ as follows. $D_b$ is a sequence of values which for every $k \in \{1, .., K\}$ contains all the pixel brightness values $P_k(i, j)$ for which $\bar{P}(i, j) \in I_b$:

$$D_b = (P_k(i, j)|\bar{P}(i, j) \in I_b)_{k=1}^{K}.$$

Then the variance of bin $I_b$ is said to be the variance of distribution $D_b$ with size $N_b$ and is calculated as follows:

$$\text{Var}(D_b) = \frac{N_b}{N_b - 1} \sum_{n=1}^{N_b} (D_b(n) - \bar{E}(D_b))^2.$$

The variance of $D_b$ is calculated and plotted to the expected average value of $I_b$, which we assume to be

$$\bar{E}(D_b) = \frac{1}{2} \left( b - \frac{1}{B} + \frac{b}{B} \right),$$

by applying Algorithm 1. We choose to estimate the average value of $D_b$ instead of computing it is because it is an accurate estimate (as validated in Section 6.3) and it significantly increases the efficiency of the algorithm.

**Data**: A sequence of $K$ registered images $P_k$, average image $\bar{P}$, number of bins $B$

**Result**: A sequence of $B$ distributions $D_b$

```
for b ← 1,..., B do
    I_b = [b - 1/B, b/B];
    for (i, j) ← {1, ..., m} × {1, ..., n} do
        if $\bar{P}(i, j) \in I_b$ then
            for k ← 1,..., K do
                $D_b \leftarrow (D_b, P_k(i, j))$;
            end
        end
    end
end
```

**Algorithm 1**: Filling the distributions with brightness values.
Data: A sequence of $B$ distributions $D_b$

Result: The regression slope $\beta_1$ of the variance
Create $b \times 2$ matrix $E$ and column vector $V$;
for $b \leftarrow 1, \ldots, B$ do
\begin{align*}
E_{b,1} & \leftarrow 1; \\
E_{b,2} & \leftarrow E(D_b); \\
V_b & \leftarrow \text{Var}(D_b);
\end{align*}
end

$$(\beta_0, \beta_1) = \arg\min_{\beta_0, \beta_1 \in \mathbb{R}} \| E \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} - V \|^2;$$

Algorithm 2: Computing the slope of regression (linear least square method).

![Figure 7: Histogram with 15 bars of the distribution $D_6$ for $B = 10$ in set M1.](image)

6.3 Numerical Results of Set M1

In order to illustrate Algorithm 1, we focus on Set M1. Recall set M1 has 5 images. Our initial number of bins $B$ is 10. We investigate the average brightness values of $I_6$, i.e., $D_6$ (see the histogram in Figure 7).

$$\left( P_k(i,j) | \bar{P}(i,j) \in [0.5, 0.6] \right)^5_{k=1}$$

The variance of $D_6$ is 0.0019 and its mean is 0.549. Note that in this instance, we calculated the mean by averaging the values in $D_6$. The difference with the estimate in (17) is insignificantly small. The scale factor $c$ (according to (7)) equals $0.0019/0.549 \approx 3.472 \cdot 10^{-3}$ meaning the average number of electron collisions in distribution $D_6$ equals $0.55/(3.472 \cdot 10^{-3}) \approx 158$.

With a proper bin size, where we use a bin for every integer number of electron collisions, our number of bins would be $B = 1/(3.472 \cdot 10^{-3}) \approx 288$.

It is important to note that the distributions $D_b$ are determined according to the brightness values of the average image $\bar{P}$. Because this is an estimate based on a relatively low number of registered images, the variance of $D_b$ is very depending on $K$. For that reason, we apply linear least square regression on the variance points as visible in Figure 8). Using regression, $c$ can be estimated better as the slope $\beta_1$ (regression has only been applied to the relevant sample

---

3In graphs, a histogram will be represented by $H(n)$, where $n$ is the number of bars.
Figure 8: Plot of variance of $D_b$ as a function of $E(I_b)$ for Set M1 ($B = 10, 100, 1000$).

Figure 9: Plot of variance of $D_b$ as a function of $E(I_b)$ for Set L1 ($B = 10, 100, 1000$). Note the lack of linear regularity.

data, so from $D_2$ to $D_8$. This way we avoid the negative effects of clipping). In our case regression yield a slope of $\beta_1 = 3.44 \cdot 10^{-3}$, corresponding with an average number of electron collisions in $D_6$ of $0.55/(3.44 \cdot 10^{-3}) \approx 160$. In this case, a proper bin size is $c \approx 3.44 \cdot 10^{-3}$, resulting in $B = 290$ bins.

6.4 Numerical Results of Set L1

When this method is applied to Set L1, a set of images actually registered by an EM, the results are not as clear as with Set M1. In Figure 9 the variance of the distributions is plotted to the average brightness of the bins. The lack of is caused by several factors, the most relevant factor being the fact that the sample in the EM drifted over time. The drift has a negative impact on the noise estimate, and it is not possible to provide a uniform scale factor $c$, not even with the regression slope $\beta_1$. 
7 Surface Drift

While averaging and binning is a technically involved but straightforward way to determine the level of noise, it leads to poor results when applied to certain EM images, especially images with a discrete spectrum of brightness values (i.e., where $\phi_k$ attains only two or three distinct values). Drift is caused by the fact the specimen in the EM tends to drift while the images are being registered. This way, every image still captures the relevant part of the area, intersection $S$, but deviates slightly from other images. The small deviations cause problems (especially for $\phi_k$ which only attains few distinct values) as will be explained now.

The issue lies with the determination of the average image $\bar{P}$. When $\bar{P}$ is determined by computing the average as done in Section 5, it is assumed that for every image $\phi_k$, pixel areas $s^k(i,j)$ are identical, meaning $s^{k_1}(i,j) = s^{k_2}(i,j)$ for all $k_1, k_2 = 1, ..., K$. For future reference, this way of averaging will be called uncorrected averaging. When surface drift occurs, this is no longer the case.

If we have an image set with only a few brightness levels present and we compute the simple average, the effect of surface drifts is best visible in the region of brightness level transition. Recall Set L2, and the two transitions between the darker and lighter regions. Say we examine two images $P_1, P_2$ of Set L2 where $P_1$ has a drift relative to $P_2$ and focus on a transition area. The surface drift can cause a dark pixel in $P_1$ to be averaged with a neighbouring light pixel in $P_2$. Now the average of the two pixels is gray, but the dark and the light pixel are contained in the same distribution. This results in a variance much higher than expected based on noise only. An example of this phenomenon occurring in Set L2 is given in Figure 10. In this figure, we see the distribution of brightness levels in one of the registered images (left)\(^4\), a variance plot of Set L2 without drift (center) and a variance plot of Set L2 with drift (right). In order

![Figure 10: The impact of surface drift on the pixel variance, using the simple average $\bar{P}$.](image)

...to improve noise characterization, we will try to find another way to calculate $\bar{P}$.

---

\(^4\)Note that the pixels transition regions in the image are not visible in the histogram nor the image. They do exist, for they have a variance as seen in the center and the right plot, but the number of pixels is too small to be seen. That makes the spike in variance all the more distracting from the noise estimate.
7.1 Drift-corrected Average

The following algorithm is based on taking the drifted images and computing a non-drifted approximation for each image called drift-corrected image $P_k$. Those images are used to calculate an average image per registered image. This drift-corrected average image can then be used for noise estimation.

The method is described below. All mathematical notation is introduced in Section 2. Pick any one image as a reference ($P_1$): Define this image to have a drift of $(x_1, y_1) = (0, 0)$ and recalculate the drifts of every other image relative to this image. Say the next image $P_2$ has a relative drift\(^5\) of $(x_0, y_0)$. Recall $s_{i,j}^k$ is the pixel area of pixel $(i,j)$ of image $P_k$. Then for any row $i$ and column $j$ the pixel area of $s_{i,j}^1$ differs from that of $s_{i,j}^2$, so their brightness values should not be compared. But $s_{i,j}^1$ is contained in the union of $s_{i,j}^2$ and its neighbours; more precisely (since the drift is smaller than one pixel), by at most 3 neighbours: $s_{i,j-1}^2$, $s_{i+1,j}^2$ and $s_{i+1,j-1}^2$ (see Figure 11).

---

\(^5\)It is important to note the relative drift is given in Cartesian coordinates ($(0,0)$ is lower left corner), but the rows and columns of the image matrix have a different orientation ($(0,0)$ is upper left corner).

---
We can not obtain the exact brightness of the parts of image $P_2$ pixel areas that construct $s_{i,j}^1$ in the sense that $P_2$ is piecewise constant, but we can determine an estimate by calculating the weighted transaffine average of these areas, where the weight of every area $s_{i,j}^2$ for all $i, j$ is defined by the size of the intersection with $s_{i,j}^1$. In other words, using pixel areas from $P_2$ with known brightness we integrate the pixel brightness values over $s_{i,j}^1$ to obtain $P_1^2(i,j)$. If we ignore the border pixels of image $P_1$ (which are assumed to be not specifically required for noise analysis), we can calculate $P_1^k$ for all $k = 1, \ldots, K$ and $i = 2, \ldots, m - 1$, $j = 2, \ldots, n - 1$. For a more natural definition, recall that every image matrix $P_k$ has an equivalent image function $\phi_k$. Then

$$P_1^1(i,j) = \int_{s_{i,j}^1} \phi_2(x,y) \, dx \, dy = (1 - x_2)(1 - y_2)P_2(i,j) + x_2(1 - y_2)P_2(i, j - 1) + (1 - x_2)y_2P_2(i + 1, j) + x_2y_2P_2(i + 1, j - 1),$$

where the right-hand side applies in case of drifts $x_0, y_0 \geq 0$. In our example, where $P_2$ has a drift of $(x_2, y_2) = (0.3, 0.6)$, the integration weights are presented in Figure 12.

$$P_2^1(i,j) = 0.28P_2(i,j) + 0.12P_2(i, j - 1) + 0.42P_2(i + 1, j) + 0.18P_2(i + 1, j - 1).$$

We can subsequently compute an estimate from every other image $P_k$ of the set, and use these estimated images to compute a more useful average image, the drift-corrected average, all defined on the $s_{i,j}^1$. For $P_1$, the drift corrected average $\bar{P}_1$ is defined as

$$\bar{P}_1 = \frac{1}{K} \sum_{k=1}^{K} P^1_k.$$

In Section 7.4 this method will be applied to Set L2. First an implementation of the algorithm is presented.
Data: A sequence of $K$ $m \times n$ images with drifts.

\[(x_k, y_k)_{k=1}^{K}\] with respect to a certain reference point.

Result: An average image $\bar{P}_{k0}$ existing of corrected images with respect to $P_{k0}$

Pick $k_0 \in \{1, ..., K\}$ (without loss of generality assume $k_0 = 1$);

for $k \leftarrow 2, ..., K$ do

\[(\Delta_1 x_k, \Delta_1 y_k) \leftarrow (x_k, y_k) - (x_1, y_1);\]

end

for $(i, j) \leftarrow \{2, ..., m - 1\} \times \{2, ..., n - 1\}$ do

\[P^1_k(i, j) \leftarrow P_k(i, j);\]

end

for $k \leftarrow 2, ..., K$ do

calculate integration stencil based on $(\Delta_1 x_k, \Delta_1 y_k)$ see Section 7.1 for an example;

for $(i, j) \leftarrow \{2, ..., m - 1\} \times \{2, ..., n - 1\}$ do

compute $P^1_k(i, j)$ by applying the stencil on:

\[P_k(i, j), P_k(i - \text{sgn}(\Delta_1 y_k), j), P_k(i, j + \text{sgn}(\Delta_1 x_k))\] and

\[P_k(i - \text{sgn}(\Delta_1 y_k), j + \text{sgn}(\Delta_1 x_k));\]

end

end

Algorithm 3: Correcting the drifted images.

7.2 Drift-correction Algorithm

After drift-correcting all images with respect to $P^{k0}$ one calculates the drift-corrected average $\bar{P}^{k0}$ (relative to $P_{k0}$) as

\[\bar{P}^{k0} = \frac{1}{K} \sum_{k=1}^{K} P^{k0}_k.\] (21)

7.3 Validation of Drift-correction algorithm

To validate Algorithm 3, we create a linear image (i.e., an image matrix corresponding with a linear function) $L^*$ with dimension $m = n = 1024$. This images attains a minimum brightness in pixel $(1, 1)$ and gradually becomes brighter in such a way that pixel $(n, n)$ attains the brightness maximum (visible in Figure 13). The definition for $L^*$:

\[L^*(i, j) = (i + j - 2)/(2 \cdot n) \quad \forall i, j = 1, ..., n.\] (22)

This image is created and manipulated in MATLAB, so the image brightness values are directly on a $[0, 1)$-scale. Next, we need drifted images which we will construct as follows: (perfect) image $L^*$ is duplicated 4 times and each duplicate (registered) image is given a sub-pixel drift as was done with Set L2 (resulting in images $L_1, ..., L_4$). Note that for duplication images $L_1, ..., L_4$ were saved in an 8-bit format.

Because the image function is linear, using a transaffine average in the drift-correction algorithm is errorless in approximation. The MATLAB implementation of the drift-correction algorithm has been applied on image $L_1, ..., L_4$. Result: $\bar{L}^1 \approx L_1$, where the error was due to digitalization noise caused by the 8-bit format. This validates Algorithm 3.
7.4 Numerical Results on Set L2

The 6 images in Set L2 (see Figure 4) were each given a unique sub-pixel drift and noise manifestation as to mimic Set L1, accomplished by using a large high resolution image (3246 × 2448) as a perfect image, shifting the image 1 to 3 pixels in the \(x\) and \(y\) direction and lower the resolution by combining 4 by 4 pixels into 1. Afterwards, artificial shot noise was added to every image.

The first step confirms that the spikes in variance are caused by the surface drift. In Figure 10 is a plot of the variance of the binned pixels per brightness bin. Next to it is a plot where the same images are used, only without a surface drift. The histogram on the left shows the brightness distribution in the image. It is clear that even though the amount of pixels in the midsection (i.e. pixel brightness \((0.2, 0.6))\) is insignificantly small, the surface drift causes the corresponding variance to be significantly larger than could be expected based on the noise.

Next, we applied the drift-correction method to correct the images. The resulting images were averaged once for every image and compared, and to estimate the level of noise the variance was once more plotted to the average brightness (Figure 14).

It is clear that the drift-correcting method has a positive result on eliminating the surface drift and estimating the level of noise. To measure the effectiveness of this method, we need to compare it to the same images, only without a surface drift. After all, there is no way of correcting for surface drifts more flawless than having no surface drifts at all. This is the same set of images used in previous Figure 10 and in this example, using a simple average will suffice.

The result is depicted in Figure 15. The drift-correction method has a higher variance where the major brightness levels are present \((0.1,0.8)\) which may be caused by the clipping of the registered images, but performs very well on the transition regions (brightness level 0.2-0.6).
7.5 Numerical results on Set M2

The images in Set M2 were constructed in the same way as the images in Set L2 (see Section 7.4). If we compare the uncorrected average with the drift-corrected average we see the result in Figure 16. It is clear the decrease in variance is much less strong than it was in Set L2, but that is caused by the fact that the transitions in the image are not as extreme as in Set L2. It is visible that after drift-correcting the set, the application of regression yields more representative results. In Figure 17 the result of the drift-correction algorithm is compared to the same image set without drift. The result is less spectacular than in Set L2, but still a significant improvement over uncorrected averaging.

Figure 14: Comparison of the two averaging methods in set L2.

Figure 15: Comparison of drift-corrected average to the set L2 without drift.
Figure 16: Comparison of the interpolating averaging method to a simple average of the set M2 without drift.

Figure 17: Comparison of drift-corrected average to the set M2 without drift.

8 Discussion/Future Research

Following these observations there remain several questions. The most important ones are listed:

- Are there other ways to correct for surface drifts, requiring less assumptions than made in this thesis?
- Formulate requirements in terms of brightness values and transitions for the success of the drift-correction algorithm.
- Find the error estimate of using transaffine average to determine the drift-corrected average.
- Determine the complexity of the algorithms.
A Shot Noise

When analysing the shot noise, it is assumed the magnitude of the noise (i.e. the amount of collisions of the electrons onto the surface) is Poisson distributed. The related process satisfies the requirements typical for Poisson distributions: (1) the electron collisions are mutually independent, (2) for any region and time interval small enough, the probability of 2 or more collisions is negligible, and (3) the number of electron collisions is approximately uniform for any part of the surface. Given a random variable $X$ (representing the observed brightness in this thesis), its Poisson distribution is denoted as

$$X \sim \text{POI}(\lambda),$$

where $\lambda$ denotes the parameter of the distribution, which reflects the rate of occurrence of (in this case) the electron collisions. The expected value or mean of a random sample $X$ is denoted with $E(X)$. The variance of the sample is denoted with $\text{Var}(X)$. Distinctive for the Poisson distribution are the value of the mean and the variance of a random sample:

$$E(X) = \text{Var}(X) = \lambda.$$  \hfill (24)

B MATLAB commands used

All of the images and results have been generated in MATLAB. The important commands are included below.

- `imread('filename.ext')` - converts the image on the specified path into a matrix or a set of matrices, depending on the presence of color in the image.
- `imwrite(IMG,'filename.ext')` - creates an image based on matrix `IMG`.
- `rgb2gray(IMG)` - converts image `IMG` in RGB to an image in grayscale.
- `imnoise(IMG,'poisson')` - adds shot noise to given image `IMG`.

In addition, to facilitate computations with images several new MATLAB scripts were created (with given code length):

- `readImage('filename')` - reads the image on the given path, converts it to a grayscale image (if necessary) and scales the values to $[0-1)$ by estimating the number of bits (18 lines).
- `createNoiseSamples('filename','fmt',n)` - opens image on specified path with specified format, converts the image to grayscale (if necessary), duplicates the image $n$ times and adds a unique layer of shot noise to each image (15 lines).
- `getBin(IMset,AVIM,intensity,binwidth)` - fills the distribution around brightness value $\text{intensity} \pm 0.5 \times \text{binwidth}$ with images `IMset` by binning average image `AVIM`. Implemented in Algorithm 1 (26 lines).
- `getIMsetBrightnessVars(IMsubset,AVIM,binwidth)` - compares the images in `IMsubset` to average image `AVIM`. Calculates the variances and the intensities of the distributions with `binwidth`. Corresponds with Algorithm 1 (51 lines).
createSubGrid(drift, sps, 'filename') - drifts the specified image with drift drift and creates a lower resolution version (by combining sps$^2$ pixels into one pixel) of the specified image (45 lines).

getDriftCorrectedImage(IMset, driftset, im_ids) - calculates the drift-corrected version of the images in IMset for every image in im_ids, where the drifts are specified in driftset (42 lines).

compareSimpToAvg - script that for a given drifted image set calculates the uncorrected average image, computes the variances of the corresponding distributions, calculates the drift-corrected average for all images, computes the variances of the corresponding distributions again (for the drift-corrected averages), and plots both results (38 lines).
Index

average image, 9
bins, 10
clipping, 8
digitalization, 9
drift-corrected average, 16
linear image, 17
noise, 3
perfect image, 4
pixel area, 4
pixel, 4
registered image, 4
scaling, 8
shot noise, 3
surface drift, 4
uncorrected averaging, 14
References
