Technical Report
Bias Reduction of Centroid Detection
June 09
Jo Helmuth and Janick Cardinale

Contents
1 Abstract 1
2 Introduction 1
3 Bias of the Original Algorithm 2
4 Improved Multi-Step Algorithm 3
5 Alternatives: Gaussian Fit Algorithm 5
6 Conclusions 7
7 Appendix 7

1 Abstract
In Sbalzarini et. al. [3] a feature point tracking algorithm is developed and benchmarked. Based on tracking tests with images of moving point sources the accuracy and precision of the estimated displacements are shown to be reasonably high on a wide range of SNRs and point source velocities. The presented algorithm outperforms all competitors. Besides the SNR, the true point location could possibly influence the localization accuracy and precision, an aspect that has so far not been addressed systematically. There is experimental evidence [1] that true locations on boundaries between pixels are harder to estimate which leads to a systematic bias towards pixels centers. In this study, we demonstrate this bias towards pixel centers on synthetic test images. We speculate about the source of the bias. The most likely source is removed to yield an improved point detection algorithm, which is then empirically validated.

2 Introduction
The point detection algorithm in [3] (the original algorithm) is the one presented in [2]. It consists of two steps: In the first step, local intensity maxima serve as tentative point locations. The input to the algorithm is a matrix of gray-scale intensities, a digital image. The individual values essentially correspond to the true physical light intensities impinging onto a small region of the detector integrated over a small period of time. These regularly arranged regions on the chip and the corresponding regions in the image are referred to as a pixel. The images of individual points are assumed to be of a peak-like shape that is a few pixels wide; nevertheless, this does not exclude the applicability of the algorithm to other shapes. The first algorithmic step yields positions up to at most single pixel accuracy.

The tentative point locations are refined in the second algorithmic step. This step is based on the assumption that the point’s images (intensity peaks) have an axial symmetry with the axes’ origin centered on the true point location. Most microscopic imaging systems approximately produce radial symmetry, so the assumption is fulfilled by definition. This allows to estimate the point location by estimating the
intensity-weighted pixel centroid (intensity centroid). This task involves only rather simple computations that can be implemented efficiently, which is a desirable property of the algorithm.

The intensity centroid is only defined for finite domains. Therefore only pixels whose center is within a symmetric mask centered on the tentative point location are considered. In the original algorithm this mask is circular and its radius is a free parameter. It should be chosen small enough such that at most one peak is contained in it. If it is too small, however, parts of the peak might be outside, which renders the detection inaccurate. In case of isolated peaks, it seems that larger regions should therefore be favored. It turns out, however, that this can not be considered a general guideline.

The influence of the image quality on the detection accuracy has so far not been addressed theoretically. Besides the level of noise present in the images, the non-specific background signal plays an important role. Clearly, the intensity centroid within a mask can only coincide with the true point location if the image is background-free. Any uniform background signal biases the detection towards the mask’s center [2, 4]. With increasing mask sizes and higher background intensities this effect gets worse. In the original algorithm, the background is therefore removed by a box-car average filter. What should be kept in mind, however, is that the background needs to be determined prior to removal. This amounts to solving an estimation problem for the parameters (e.g. the mean) of the intensity distribution in the background. Such a statistical procedure is never perfect and hence a small background will always remain, increasing the variance and possibly the bias of the detection. One has to expect that this residual background is larger for smaller signal-to-noise-ratios (SNR). We summarize that in case of imperfect images and even for isolated peaks there are contradicting requirements to the mask size.

As already stated, small masks risk that significant parts of the peak are not used for the centroid computation. In this case, the symmetry of the image within the mask with respect to the true point location is no longer guaranteed. If the mask is far off the true location, this will clearly result in a large bias. If, however, the mask is already centered on the true location, the bias will be small, if not vanishing. If larger masks are used, the bias due to the background depends on the current mask center position as well. Again, the closer the mask center to the true point location, the smaller the bias. This consideration motivates a multi-step algorithm: Iteratively re-centering the mask onto the last estimated centroid position should yield an estimator that converges to the correct position.

The original algorithm is, up to a very special case, a single step procedure. Due to the issues outlined above, the mask size is usually chosen large enough to contain most of the significant part of the peak. Since the mask is centered on the tentative point location (which is in the center of a pixel) one would expect to see a bias towards the center of this pixel. The further the true point location is off the pixel center the larger the observed bias should be. Reversely, one would expect to see estimated locations on pixel boundaries less frequently than those on pixel centers. In section 3, we try to quantify this effects on synthetic data and show in which practical situations it is relevant.

The multi-step algorithm should reduce the biasing effect of too small masks or not ideally removed backgrounds. Such an algorithm requires to replace the discrete binary masks used in the original algorithm by a continuous equivalent. These masks take into account that only parts of the pixels may be contained within a mask. They assign weights to the pixels proportional to the amount of overlap. In section 4 the continuous masks are introduced and a full specification of the iterative algorithm is given. Using the same synthetic data as before, the bias reduction effect is demonstrated.

3 Bias of the Original Algorithm

We hypothesize that the original point tracking algorithm exhibits a position determination bias that is more complex then the bias demonstrated in [3]. Three factors should influence this bias: The SNR, the mask size parameter \( w \) and the true point location \((x_p, y_p)\). In order to resolve these dependencies in a semi-quantitative but illustrative manner a new benchmark is constructed: In an image domain of 21 pixels a single point source is placed at locations \((x_p, y_p)\) in \([10.0, 10.03, 10.06, \ldots, 10.99] \times [10.0, 10.03, 10.06, \ldots, 10.99]\), one at a time. The resulting image is generated by integrating a Gaussian with standard deviation \( \sigma = 1 \) over the square regions of the pixels as described in [4] and a background of \( b = 10 \) is added. As described in [3] the maximum intensity above \( b \) is tuned to yield the desired SNR once the images are corrupted by Poisson-distributed noise. We generated images with SNRs 3.5, 6.5 and \( \infty \) (i.e. is perfect noise-free images). For each true point location and corresponding set of images, the original detection algorithm is applied with a fixed mask size parameter \( w \in \{1, 2, 3, 6\} \) and the estimated location is compared with the true location.

In figure 1, the resulting pattern of offsets (lines) between estimated (red dots) and true (green dots)
locations is shown. As expected, the errors get smaller with increased SNRs. However, depending on the mask size used, a residual bias can be observed, that, for small sizes, is severe. The tendency is clearly towards centers of pixels (that lie on locations mask size used, a residual bias can be observed, that, for small sizes, is severe. The tendency is clearly towards locations is shown. As expected, the errors get smaller with increased SNRs. However, depending on the mask size used, a residual bias can be attributed to the finite discontinuous masks used. If the mask is too small to cover most of the brighter parts of the peak in the image, the symmetry of the image within the mask is lost. In addition, within the mask the truncated image of the peak appears to have a non-zero background, which causes a bias towards pixel centers due to the reasons already discussed.

For larger mask sizes, this bias is reduced until it can no longer be observed given the apparent magnitude of indeterminate errors. This is clearly in contradiction to the initially formulated expectations: We hypothesized that the background-removal strategy applied could never perfectly eliminate the background, which causes a bias. The inability to remove the background is related to the stochastic nature of the background, i.e. the higher the background noise, the harder it is to estimate the background and hence the larger the residual background will be. However, even if the background could be removed ideally, the background noise would remain, which would cause an increase of the variance of the location estimation. The bias and variance due to the noisy background can of course not be changed independently, which could be confirmed by additional experiments with higher background, but otherwise identical SNRs (data not shown). In summary, the bias due to the residual background is much smaller in magnitude than the variance due to the residual background noise, which makes it insignificant and only detectable by a much larger number of repetitions than performed in this study.

4 Improved Multi-Step Algorithm

In the original algorithm an offset from the initial point locations is computed based on the intensity-weighted centroid of pixels within a circular mask. Since nothing is known about the intensity distribution within a pixel, the intensity-weighted centroid within the pixel is put into its geometric center. In the following, we use the same notation as in \[3\]. The mask is defined implicitly by the summation rule in equations (7) and (8), which only considers pixels that are at positions \((i, j)\) relative to the initial location \((\hat{x}_p, \hat{y}_p)\) for which \(i^2 + j^2 \leq w^2\). This implies that the initial locations have to be on pixels centers. Relaxing this restriction allows to construct an algorithm in which the initial location is iteratively updated with the last computed offset until convergence is achieved. To facilitate convergence continuous weights \(W\) are assigned to the pixels proportional to their overlap with a circular mask centered on the current location estimate \((\tilde{x}_p, \tilde{y}_p)\). Equation (7) and (8) are replaced by:

\[
\begin{bmatrix}
\varepsilon_x(p) \\
\varepsilon_y(p)
\end{bmatrix} = \frac{1}{m_0(p)} \sum_{|i| \leq w, |j| \leq w} s_{ij} A_i^j(\tilde{x}_p + i, \tilde{y}_p + j) W(\tilde{x}_p + i, \tilde{y}_p + j), \tag{1}
\]

and

\[
m_0(p) = \sum_{|i| \leq w, |j| \leq w} A_i^j(\tilde{x}_p + i, \tilde{y}_p + j) W(\tilde{x}_p + i, \tilde{y}_p + j). \tag{2}
\]

The vector \(s_{ij}\) points from the current location estimate \((\tilde{x}_p, \tilde{y}_p)\) to the centroid of the masked area of the pixel at position \((\tilde{x}_p + i, \tilde{y}_p + j)\) (see Appendix for details).

Prior to the first iteration, \((\tilde{x}_p, \tilde{y}_p)\) is initialized with \((\tilde{x}_p, \tilde{y}_p)\). In each iteration of the algorithm \((\varepsilon_x(p), \varepsilon_y(p))\) is computed according to Eq. 1 and the updates

\[
\begin{bmatrix}
\tilde{x}_p \\
\tilde{y}_p
\end{bmatrix} \leftarrow \begin{bmatrix}
\hat{x}_p + \varepsilon_x(p) \\
\hat{y}_p + \varepsilon_y(p)
\end{bmatrix}, \tag{3}
\]

and

\[
\begin{bmatrix}
\tilde{x}_p \\
\tilde{y}_p
\end{bmatrix} \leftarrow \text{pixel center closest to} \begin{bmatrix}
\hat{x}_p \\
\hat{y}_p
\end{bmatrix} \tag{4}
\]

are carried out.

The pixel weights \(W\) have to be recomputed in each iteration, which adds to the computational cost of the algorithm. The weights should vary smoothly from 0 to 1 with the pixel position, where pixels totally inside or outside should be assigned a weight of 1 or 0, respectively. Furthermore, for pixels on the boundary of the mask, the weight should be approximately proportional to the area fraction overlapping with the mask. Fig. 2A shows the overlap of some example square pixels with a circular mask of radius \(R\). Since squares have non-smooth boundaries, different cases have to be distinguished for the computation.
Figure 1: Position dependent detection accuracy and precision of original algorithm. Columns are fixed SNRs (3.5, 6.5, $\infty$ from left to right), rows are fixed mask widths $w$ (1, 2, 3, 6 from top to bottom).
Figure 2: Continuous pixel weights. Pixel weights $W$ are assigned proportional to the overlap between mask and pixel (gray areas). A) Square pixels. $r_{ij}$ is the offset of the pixel $(i,j)$ relative to the mask’s center. Three cases for the calculation of the overlap have to be distinguished. B) Circular pixels. Knowledge of $d_{ij} = ||r_{ij}||$ and $R$ suffices to calculate the fraction of overlap.

of the overlap. Which case is present depends on the location of the pixel’s center position $r_{ij}$ relative to the mask’s center $(\tilde{x}_p, \tilde{y}_p)$ (Fig. 2A).

To reduce complexity and the computational overhead due to the case differentiation, we approximate the pixel areas by the inner circle of the square. This reduces all different cases to a single one (Fig. 2B). The area fraction of the overlap then only depends on two scalar variables, namely the radius $R$ of the mask and the distance between mask and pixel center $d_{ij} = ||r_{ij}||$. Given the current estimated point location $(\tilde{x}_p, \tilde{y}_p)$ and expressing all lengths in units of the pixel width the weights become:

$$W(\hat{x}_p + i, \hat{y}_p + j) = \frac{A_{\text{overlap}}(\hat{x}_p + i, \hat{y}_p + j)}{A_{\text{circle}}} = \frac{1}{\pi} \cos^{-1} \left( \frac{d_{ij}^2 + 0.25 - R^2}{d_{ij}} \right) + \frac{4R^2}{\pi} \cos^{-1} \left( \frac{d_{ij}^2 - 0.25 + R^2}{2d_{ij}R} \right) - \frac{2}{\pi} \sqrt{(-d_{ij} + 0.5 + R)(d_{ij} + 0.5 - R)(d_{ij} - 0.5 + R)(d_{ij} + 0.5 + R)}.$$  \hspace{1cm} (5)

These weights are invariant to rotations of the pixel grid around the mask center.

The improved algorithm is benchmarked on the same images of single point sources as the original algorithm (see section 3 and Fig. 1). For the improved algorithm, the resulting pattern of offsets (lines) between estimated (red dots) and true (green dots) locations is shown in Fig. 3. As before, the errors get smaller with increased SNRs. For mask size $w = 1$ a residual bias is observed. Unlike for the original algorithm, the bias is not clearly towards the pixel center, but rather away form pixel corners. The magnitude of the bias is much smaller, however, not negligible. The reason for this small residual bias is most probably the approximation of the overlapped areas with the intersection of two circles. Especially at the corners of the true pixel areas, the overlap is underestimated, which makes the centroid estimation less responsive to the pixels farthest away from the current position estimate and hence hampers convergence in diagonal direction.

In summary, the improved algorithm is always at least as good, but in most settings better than the original algorithm. Importantly, the dependence on the mask width $w$ (everything greater than $w = 1$ works fine) is reduced, which makes it more user-friendly.

5 Alternatives: Gaussian Fit Algorithm

In [4] a Gaussian fit algorithm is derived. The starting point is a least squares fit of the observed image with a Gaussian intensity profile, which is then transformed into an implicit equation for the point location, which is then iteratively solved. Mathematically, the equation is very similar to Eq. 1 together with Eq. 3 since an intensity-weighted pixel centroid is computed. The mask used, however, is different from the mask used for the original or improved algorithm. It weights the pixels depending on the distance to the current centroid, and the weighting function is a Gaussian centered on the current location estimate. A cutoff for the mask is not specified. For the tests shown below, we used the Gaussian weights for pixels inside the mask (i.e. a circle of radius $w$ around the current location estimate), but set them zero outside (i.e. we effectively used a truncated Gaussian).
Figure 3: Position dependent detection accuracy and precision of improved algorithm with pixel areas approximated by spheres. Columns are fixed SNRs (3.5, 6.5, \(\infty\) from left to right), rows are fixed mask widths \(w\) (1, 2, 3, 6 from top to bottom).
The Gaussian fit algorithm also performs fairly well, since, as the improved algorithm, profits from the possibility of iteratively improving the location estimate. Benchmark results based on the same images as before are shown in Fig. 4. However, for small mask widths \( w \) and low SNRs (e.g. SNR = 3.5, \( w = 2 \)) the bias is larger, and as for the original algorithm towards the pixel center. This is possibly due to the discontinuity in the truncated Gaussian used. For larger mask widths, the Gaussian fit performs as well as the improved algorithm, since the effect of the truncation get negligible.

6 Conclusions

7 Appendix

In the following the derivation of the formula for the weights \( W \) and the computation of the vector \( s_{ij} \) is summarized. Fig. 5 shows the situation of a pixel \((i,j)\) that is partly contained in a mask of radius \( R \). As before, the circle is approximated as being circular. Except for the \( x \) and \( y \) components of \( s \), the situation only depends on the two radii \( R \) and \( r \) of the circles and the distance \( d \) between their centers. Geometric considerations yield:

\[
\Delta x_p = \hat{x}_p - \tilde{x}_p + i, \quad \Delta y_p = \hat{y}_p - \tilde{y}_p + j, \\
d = \sqrt{\Delta x_p^2 + \Delta y_p^2}, \quad d_1 = \frac{d^2 - r^2 + R^2}{2d}, \quad d_2 = \frac{d^2 + r^2 - R^2}{2d}, \\
\alpha = \cos^{-1}\left(\frac{d_1}{R}\right), \quad \text{and} \quad \beta = \cos^{-1}\left(\frac{d_2}{r}\right). \quad (6)
\]

\( A_1 \) and \( A_2 \) are calculated using the formula for a circle segment:

\[
A_1 = R^2 \alpha - d_1 \sqrt{R^2 - d_1^2}, \quad \text{and} \quad A_2 = r^2 \beta - d_1 \sqrt{r^2 - d_2^2}, \quad (7)
\]

from which Eq. 5 then follows directly. The vector \( s_{ij} = (\Delta s_x, \Delta s_y)^T \) is computed using the centroids of the two circle segments:

\[
s_1 = \frac{2R(\sin \alpha)^3}{3(\alpha - \sin \alpha \cos \alpha)}, \quad s_2 = \frac{2r(\sin \beta)^3}{3(\beta - \sin \beta \cos \beta)}, \\
s = \frac{A_1 s_1 + A_2 s_2}{A_1 + A_2}, \\
\Delta s_x = \frac{\Delta x_p s}{d}, \quad \text{and} \quad \Delta s_y = \frac{\Delta y_p s}{d}. \quad (8)
\]

References


Figure 4: Position dependent detection accuracy and precision of gaussian fit algorithm. Columns are fixed SNRs (3.5, 6.5, $\infty$ from left to right), rows are fixed mask widths $w$ (1, 2, 3, 6 from top to bottom).
Figure 5: Quantities used for the computation of the masked area of a pixel and its centroid.