A COMPARISON OF METHODS FOR ESTIMATION OF INTENSITY NONUNIFORMITIES IN 2D AND 3D MICROSCOPE IMAGES OF FLUORESCENCE STAINED CELLS.

J. Lindblad, E. Bengtsson

Centre for Image Analysis, Uppsala University
Lägerhyddvägen 17, SE-752 37 Uppsala
e-mail: joakim@cb.uu.se

ABSTRACT

A comparison of the accuracy and robustness of different data driven methods for intensity nonuniformity field estimation (background shading estimation) on simulated and real images of fluorescence stained cells is presented. A novel attempt to reduce the parameter space of a B-spline based algorithm for shading estimation using automatic thresholding with a kernel density estimator is tested and compared with exhaustive testing to find the optimal parameter value.

1. INTRODUCTION

Intensity nonuniformity (INU), also commonly referred to as shading artifact, refers to smooth local changes in the image intensity introduced by the data acquisition technique. The impact of INU on visual interpretation of images is limited. It does however affect image segmentation methods because most of them rely, at least partly, on absolute intensity values. Moreover INU strongly affects the possibility to use intensity values of the segmented images as quantitative measures in further data processing.

Correction of intensity nonuniformity is typically based on either a multiplicative model of the artifact:

$$o(\boldsymbol{x}) = s(\boldsymbol{x})t(\boldsymbol{x}) \tag{1}$$

or an additive model:

$$o(\boldsymbol{x}) = t(\boldsymbol{x}) + s(\boldsymbol{x}) \tag{2}$$

where o and t is the observed respectively the true intensity at the spatial location $\boldsymbol{x}$, and s is the distortion caused by the INU. If the model and the INU field is known, the true image is easily obtained from the observed. One can smoothly move from one model to the other. By taking the logarithm of the image intensities: $\tilde{o}(\boldsymbol{x}) = log[o(\boldsymbol{x})]$ Eq. 1 becomes $\tilde{o}(\boldsymbol{x}) = \tilde{s}(\boldsymbol{x}) + \tilde{t}(\boldsymbol{x})$, i.e. equivalent to the additive model. By taking the exponent of the additive model we can similarly move back to the multiplicative model.

Estimation of the INU field (often called bias field) can either be based on specialized acquisition protocols such as imaging of homogeneous phantoms or black-level images in relation to the ordinary imaging step, or be based on analysis of the image data itself. Whether it is feasible to acquire a good background image directly in conjunction with the imaging procedure or not is very much dependent on the imaging environment. For example in MRI, most of the INU is caused by properties of the subject in the scanner [7], which implies that finding a good correction image may be very hard.

Data-driven approaches are usually simpler from a practical viewpoint, but may however introduce artifacts if the algorithm used cannot reliably solve the estimation problem.

Intensity nonuniformity is typically present in MRI data and this seems to be the area of imaging and image analysis with the richest plethora of methods for estimating the INU field. Intensity nonuniformity is however present also in relation to other areas of imaging. This paper aims explicitly at the applicability of algorithms on microscope images of fluorescence stained cells. Most methods described in relation to MRI data use the assumption that the intensity inside the different object classes (tissue classes) are fairly constant (i.e. when the INU has been removed) [1]. In fluorescence images of cells, this is clearly not the case, as the intensity varies a lot inside the cells. It is therefore not clear that a well functioning INU estimator taken from the MR field will also produce good results on fluorescence images of cells.

The aim of this paper is to look closer at a set of data-driven methods for INU estimation, trying to estimate their robustness and accuracy in relation to microscope images of fluorescence stained cells.

3D is emerging more and more in the field of medical image analysis, so also in fluorescence microscopy imaging, primary using confocal techniques. All algorithms in this paper are easily (and have also been) extended to three or more dimensions but most examples will be relating to 2D data. 2D images are easier to visualize, and require less computing power for the validation testings.

2. THE IMAGES OF INTEREST

A source of information about the cell which is gaining more and more in importance is the detection and analysis of fluorescence emitted when the cell has been exposed to fluorescence dyes attached to some biologically active molecule and is exposed to excitation light. The applications are ranging from pure research of the cell functionality, to cancer research and to drug discovery. In all cases it is desirable to acquire a pure and noise free image, to be able to segment and analyze the cell features correctly. One important step in this process is the correction for intensity nonuniformity caused by the image acquisition process.

3. THE METHODS

A number of authors have proposed spatial filtering (often homomorphic filtering [4]) as a means to estimate the INU field [2]. These techniques include the straightforward use of a large mean filtering of the image, or using some percentile filtering (including the 0% percentile to get a minimum filter). The disadvantage of the spatial filtering methods is that they in general rely on the assumption that the frequency spectrum of the INU field $s(\boldsymbol{x})$ and that of the true image $t(\boldsymbol{x})$ are separable, which is typically not the case, neither in MR nor in fluorescence images. As such, spatial filtering tend to introduce severe, undesirable filtering artifacts.

Rolling ball algorithms may perform better than just a flat minimum filter but they have the same inherent properties and corresponding problems as a minimum filter (dark background case). If the filter is too small, it will include the foreground objects. If the filter is too large, it will smooth the background. Also, using the extreme values as the rolling ball and minimum filter approach does is noise sensitive. A better approach is to use a small percentile, e.g. the 5% percentile. Unfortunately the choice of an optimal percentile is dependent on the amount of noise present in the image.

Another approach is to try to estimate the background field as a smoothly varying function [1]. Many different functions have been proposed in the literature, but the most popular is probably the class of uniform cubic B-spline functions (see e.g. [5]). Cubic B-spline functions have many nice properties, they guarantee $\mathcal{C}^2$ continuity, i.e. they are always smooth, and different levels of flexibility can be allowed by varying the number of control points used for the function. The task of fitting a B-spline function to data may be formulated as an ordinary least squares solution, and is thus analytically solvable in an efficient way, which is of course also a very nice property.

When discussing this approach of approximating the INU field with a function we will typically refer to the background as a surface, even though all concepts in this article are easily (and have also been) extended to 3D, where the surface becomes a volume. Alternatively one can think of the volume case as fitting a hyper-surface in a 4D space, this way the similarity with the 2D case will be very direct.

When fitting a surface to the background field, the big problem is to decide what is the background. We need to find the set of pixels Φ which belongs to the background, so that we have something to fit the surface to. I.e. some sort of foreground / background segmentation is needed. This segmentation is not trivial, as the INU field we wish to estimate is present and will sabotage thresholding attempts. It is also very difficult to apply some edge detection scheme when regarding the images at hand, as no distinct edges are present. On the other hand, if the INU field was not there, a simple thresholding would suffice. This inevitably leads to the following iterative algorithm for the INU field estimation

0. k=0. Make a poor first estimate of the true image; $\hat{t}_0(\boldsymbol{x}) = o(\boldsymbol{x})$.

1. Estimate the background by thresholding the estimated image data $\hat{t}_k(\boldsymbol{x})$ to get the set Φ_k. See below for discussions about the threshold.

2. Make a least squares fit of a uniform B-spline surface (or volume) to the segmented background pixels (voxels) Φ_k in the observed image $o(\boldsymbol{x})$ to get $\hat{s}_k(\boldsymbol{x})$.

3. Correct the image according to the fitted background (we have used the additive model) $\hat{t}_{k+1}(\boldsymbol{x}) = o(\boldsymbol{x}) - \hat{s}_k(\boldsymbol{x})$

4. If $\text{rms}(\hat{s}_k - \hat{s}_{k-1}) < \delta$ we are finished, otherwise set $k \leftarrow k + 1$ and repeat from step one.

which closely resembles the one designed by Gilles et al. [3].

The convergence criterium that has been used in this study is that the estimated background does not change more than that the root mean square (rms) of the difference between two consecutive background estimates $\hat{s}$ is less than $\delta = 1/1000$ of the original image variation.

Figure 1 shows an images cross section at various stages of the algorithm.

3.1. How to do the background segmentation

As mentioned before, a single threshold is supposed to be sufficient for the background segmentation, as the corrected image will more and more approach the case of a flat background.

Gilles et al. [3] suggest the use of a fixed threshold ϵ (two actually) decided experimentally. Unfortunately, if we wish to have optimal performance, this threshold must vary with the noise level of the image, as we can see in figure 2.

By assuming that the internal variation in the background roughly approximates a normal distribution, we can

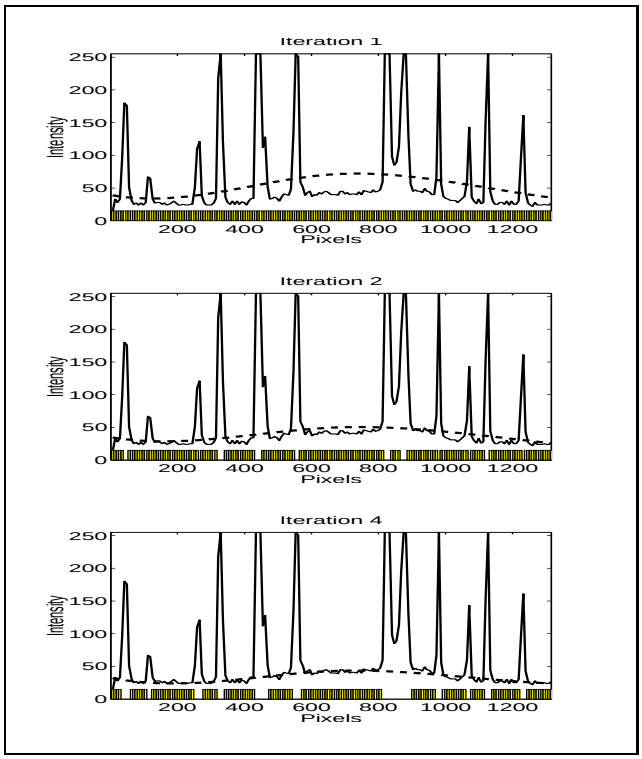

Fig. 1: The intensity profile across one image with fitted background (dashed) after the first, second and fourth (final) iteration of the background approximation algorithm. At the bottom of each graph is indicated which pixels from the original image that have been used for the background fitting in the current step, i.e. the set Φ_k.

get a fairly noise independent threshold by cutting this background distribution at some fixed percentile to get the background set Φ. The question of which percentile (or equivalent; how many standard deviations, σ) one should set the threshold to, still remains to be solved. This is extensively tested in section 4.

A way to get rid of the problem of subjectively choosing an appropriate threshold is to use some histogram thresholding scheme to objectively find a suitable segmentation threshold. The most popular histogram thresholding technique is to look for a minimum in the distribution. In general, no local minimum in the histograms of the images studied does however exist and this method can therefore not be used as a threshold selector. Another choice of thresholds is local maxima in the second derivative of the intensity distribution of the image. This seems to be a manageable approach to the problem, and is therefore tested in this paper. The intensity distribution is estimated with a kernel density estimator (KDE) with the so called 'rule of thumb' value for the kernel bandwidth. This is supposed to be a fairly good choice of bandwidth, as the distribution rarely deviates much from the normal distribution which the 'rule of thumb' value is based on. [6].

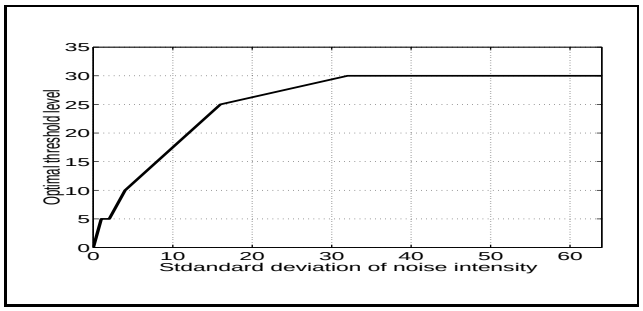

Fig. 2: Empirically determined optimal threshold for different noise levels. The optimal threshold is dependent on the overall noise level of the image.

3.2. B-spline approximation

The INU field is assumed to be $\mathcal{C}^2$-differentiable (i.e. it is smooth and without sharp edges), which leads us to the use of *cubic* splines. The control points of the B-spline are distributed on a regular mesh. In this paper mesh sizes of 4x4 and 5x5 have been used for the 2D case, and 4x4x4 for the 3D case. The B-spline surface (here in two dimensions) is written as:

$$\hat{s}(x,y) = \sum_{i=0}^{n_x-1} \sum_{j=0}^{n_y-1} p_{ij} B_i^x(\boldsymbol{x}) B_j^y(y) \qquad (3)$$

where n_x and n_y is the number of control points in the x and y directions and p_{ij} are the control points. The control points are chosen to minimize the least-squares equation:

$$Err = \sum_{x \in \Phi} (\hat{s}(\boldsymbol{x}) - s(\boldsymbol{x}))^2 \qquad (4)$$

where Φ is the set of background pixels (voxels) selected by the thresholding.

4. TESTING THE METHODS

To validate the methods, a dataset with known INU field must exist. The dataset used in this study consists of computer generated synthetic test images. The synthetic images have been designed to imitate the looks of the real images that we are aiming to correct, but with some simplifications which should not affect the estimation results with anything remotely near the inter image variations. Finally, the methods have also been applied to real test images.

4.1. Synthetic images

4.1.1. The true image $t(\boldsymbol{x})$

The true image $t(\boldsymbol{x})$ is supposed to mimic more or less randomly distributed fluorescence labelled cells on a flat background. The cells are simulated as local cosine functions of varying sizes.

In one dimension the generating equation becomes

$$t_{i+1}(x) = t_i(x) + c((x-m)/w) \tag{5}$$

$$c(x) = \begin{cases} (cos(x)+1)/2 & -\pi < x < \pi \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

to add one cell of width w centered at the position m. The 2D and 3D versions are simply tensor products of the 1D case.

Note that there does not exist any sharp edge where the cell ends and the background starts. This is true in both the real situation as well as in the synthetic images.

4.1.2. The shading image $s(\boldsymbol{x})$

The shading image, representing the INU to be estimated, may be of varying nature. Assuming that the shading originates from illumination artifacts in the microscope, using a cosine function similar to the one used to generate the cells but of larger scale, seems to be a good model. Additionally, various sloped fields may make the shading image more complex.

4.1.3. Noise image $n(\boldsymbol{x})$

To produce at least fairly realistic images, noise has to be added in some sense. Random Gaussian noise of varying amplitude has been the source used in this study.

4.2. Putting the image sources together

How are the three sources $t(\boldsymbol{x})$, $s(\boldsymbol{x})$ and $n(\boldsymbol{x})$ mixed to create the observed image $o(\boldsymbol{x})$? This surely depends on the imaging device and the data acquisition technique. Which of the additive or the multiplicative models that applies best on the fluorescence images studied may be debated. One can argue that there is a shading effect arising from leak through in the filters from the excitation light source. This is clearly an additive contribution, which is present also when there is no specimen to observe. But if the excitation light is uneven, this will also show up as a multiplicative effect, as the fluorescence is proportional to the excitation illumination. So the answer is that neither of the multiplicative nor the additive model applies.

As no perfect solution to this problem is in sight, we have simply resigned to use the additive model, equation 2. Knowing that this is not the correct model, we must think of what implications this has for our study. If we design the testing so that the main goal is to estimate the background field $s(\boldsymbol{x})$ and not to apply the correction to get the true image $t(\boldsymbol{x})$, the choice of model will not be crucial. Yes, it does affect the estimation of $s(\boldsymbol{x})$ as well, but to a lesser extent. The real correction is dependent on the specific imaging setup to such extent that it is very difficult to draw general conclusion about it. Therefore we will not go further than to the estimation and evaluation of the INU field in this paper.

Noise is of course also neither additive nor multiplicative, but rather a strange unknown function. A function which we here insult gravely by modeling it with a simple mixture of additive and multiplicative Gaussian noise.

Summarizing this gives the final image model

$$o(\boldsymbol{x}) = (1 + \alpha n_1(\boldsymbol{x}))(t(\boldsymbol{x}) + s(\boldsymbol{x})) + \beta n_2(\boldsymbol{x}) \tag{7}$$

where the constants α and β are varied to create different noise situations.

5. TESTING THE ALGORITHMS

5.1. Error measure

To validate the different methods, and to relate them to each other, an error measure is needed. We have chosen to both look at the root mean square value of the difference between the estimated shading $\hat{s}(\boldsymbol{x})$ and the true $s(\boldsymbol{x})$ as well as the maximum difference between the two.

$$rms = \sqrt{\sum_{x \in \Omega} (\hat{s}(\boldsymbol{x}) - s(\boldsymbol{x}))^2} \tag{8}$$

$$maxerr = \max_{x \in \Omega} |\hat{s}(\boldsymbol{x}) - s(\boldsymbol{x})| \tag{9}$$

Ω is the whole image.

5.2. The different test cases

The different methods tested are

1. Homomorphic filtering using mean, 5% percentile, 10% percentile, and median filters of sizes 15,31,63 and 127 pixels.

2. Iterative cubic B-spline fitting with 4x4 resp. 5x5 control points. The threshold is set to 1.0, 1.2, 1.4, ... 4.0 times the standard deviation of the estimated background in each step

3. Iterative cubic B-spline fitting with 4x4 resp. 5x5 control points. The threshold is set to the first local maximum in the second derivative of the kernel density estimate of the corrected image $\hat{t}_k$ in each step.

The images used to create the test images can be seen in figures 3 and 4. In all the synthetic test cases, Gaussian additive noise of standard deviations 0, 1, 2, 4, 8, 16, 32 and 64 have been applied. Two examples of synthetic images can be seen in figure 5.

The iterative B-spline fitting algorithms has also been tested on real image data of two types. These are the type of

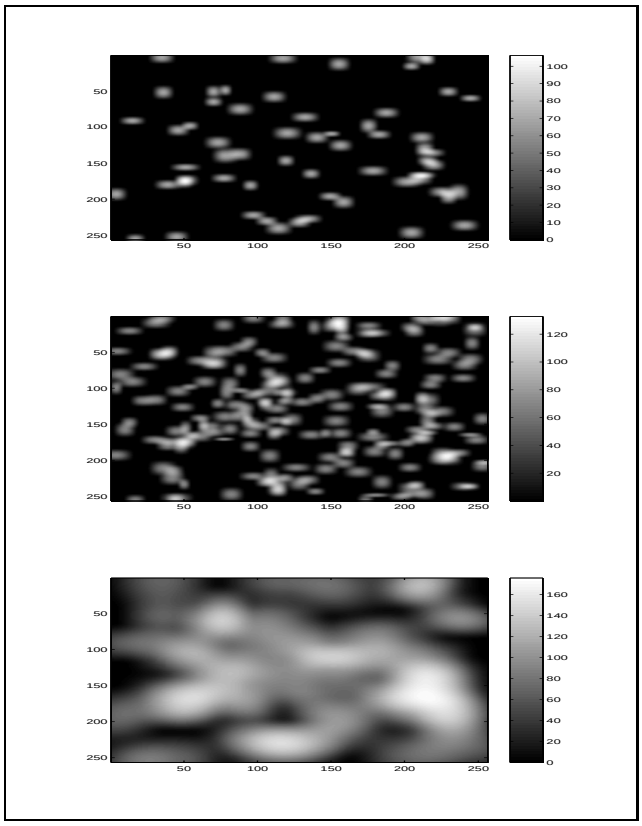

Fig. 3: The three different 256x256 pixels synthetic 'true' images ($t(\boldsymbol{x})$) tested are 60 small cosine objects of intensity 100 (top), 200 small cosine objects of intensity 100 (middle) and 60 larger cosine objects of intensity 100 (bottom).

images that we aim to mimic with the synthetic test images. As they are from real situations, no known ground truth exists for them and therefore no objective results can be plotted in relation to them. Subjective analysis is of course always possible.

The first real data set consists of 2D fluorescence images of cells cultivated on microscope slides. The background is very controlled in this situation and the iterative B-spline fitting method performs excellent, both when using KDE estimation of the background, and when using a preset factor of the standard deviation as a threshold. An example image from this data set can be seen in figure 10.

The second real data set consists of a volume image created by optical sectioning through a fluorescence stained slice of a cervix tumor. Autofluorescence, i.e. natural tissue fluorescence, gives rise to background variation. Thus, there are two kinds of background variation present in the image; the background variation caused by the imaging system and the background caused by the autofluorescence. This gives the algorithms a more problematic situation, and using a two step approximation is probably a good solution. Work on this case is in progress. A slice through one example image from this data set can be seen in fig. 6.

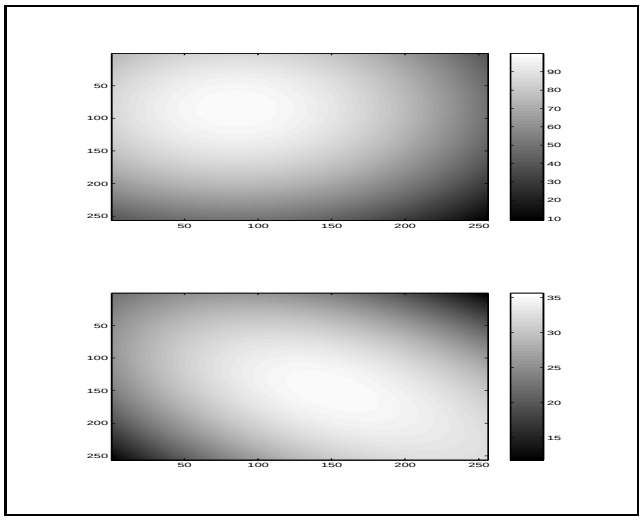

Fig. 4: The different shading images ($s(\boldsymbol{x})$) added are strong additive cosine background of intensity 100 (top), less prominent cosine background of intensity 30 added to a bright corner image of intensity 30 (bottom).

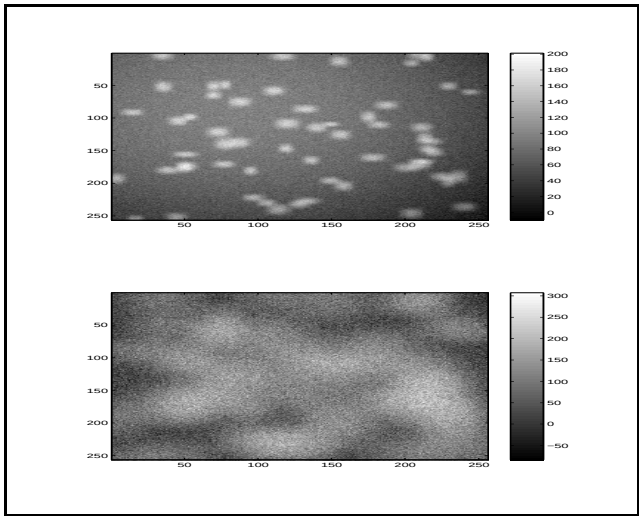

Fig. 5: Two examples of observed ($o(\boldsymbol{x})$) images. 60 small objects on strong cosine background with N(0,8) noise (top) and 60 large objects on mixed background with N(0,32) noise (bottom).

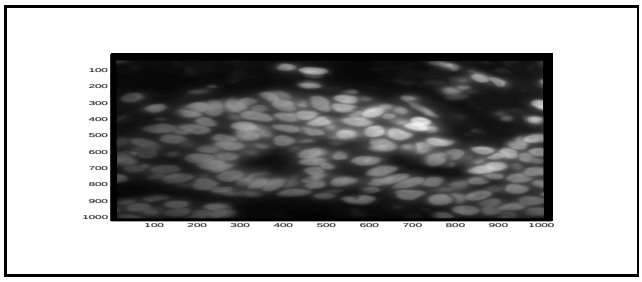

Fig. 6: One slice of a 3D image volume of which the algorithms have been tested on.

5.3. Results

Figure 7 plots the performance for homomorphic filtering using the optimal filter size for each of the three test images with the strong cosine background. The optimal filter size has in each case been selected as the best performing filter size when we perform exhaustive testing and comparing the result with the here known true solution. The mean and median filters are not well suited for the asymmetric problem and performs very poorly in the low noise case. They are fairly insensitive to noise though, as opposed to the 5% and the 10% percentile filters, which fails severely when the noise level increases. The optimal filter size varies with the size of objects, and the small percentile filters usually have the peak in their optimum at a smaller size. Table 1 summarizes the optimal filter sizes. A small tendency towards larger filter sizes for higher noise levels is also present.

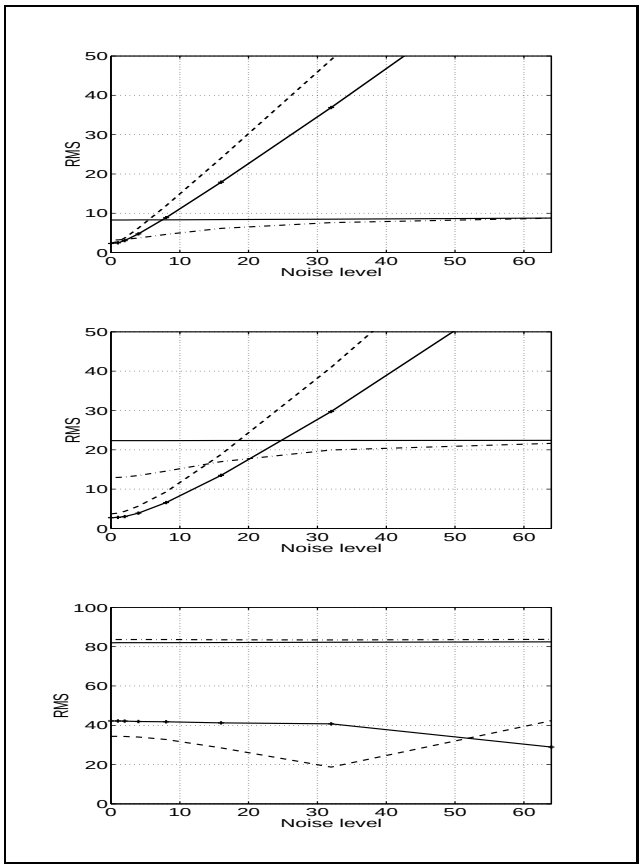

Fig. 7: The rms error values for the homomorphic filtering at the optimal filter size for different noise situations on image with; 60 small objects (top), 200 small objects (middle) and 60 larger objects (bottom) on a strong cosine background for; mean filter (solid line), 5% percentile (dashed), 10% percentile (crossed) and median filter (dash-dotted).

Figure 8 plots the performance at different threshold levels for the three test images with the strong cosine background, using the iterative B-spline fitting algorithm with a 4x4 control point mesh.

Table 1: Optimal filter sizes selected from the set 3, 7, 15, 31, 63, 127 and 255 pixels.

Image	mean	5%	10%	median
60 small objects (diam. ca 20 pixels)	63	15	15	31
200 small objects	63	15	31	63
60 larger objects (diam. ca 80 pixels)	63	63	63	31

The method works well in the case of the smaller objects, but not at all so well on the much more difficult large object image. The optimal value for the threshold varies between $1.2/\sigma$ and $1.8/\sigma$. The number of iterations required for convergence of the estimation varies between 2 and 40, but usually stays below 15.

Comparing the kernel density estimate thresholding with the optimal threshold value (figure 9) reveals the estimate to be quite close to the optimum, and even performing better than any possible choice of a fixed threshold in the 200 object case. In the large object case it does not really find a good solution in the moderately noisy situations, and is there outperformed by the optimal threshold settings.

Similar results are acquired for the mixed background situation (not shown). Using a spline mesh of 5x5 control points instead of 4x4 changes the estimations with very small numbers. Using even finer meshes is not recommended as they will tend to model the foreground variations also.

We have chosen only to plot rms values in the evaluation graphs. It is of course also interesting to look at the maximum error values of the estimations. As paper space is limited, and the maximum errors are well correlated with the rms values and do not show any significant additional information, no graphs of these values are shown. The maximum error values are usually about twice as large as the corresponding rms values for the B-spline approach and somewhere between two and four times the corresponding rms values for the filtering approach.

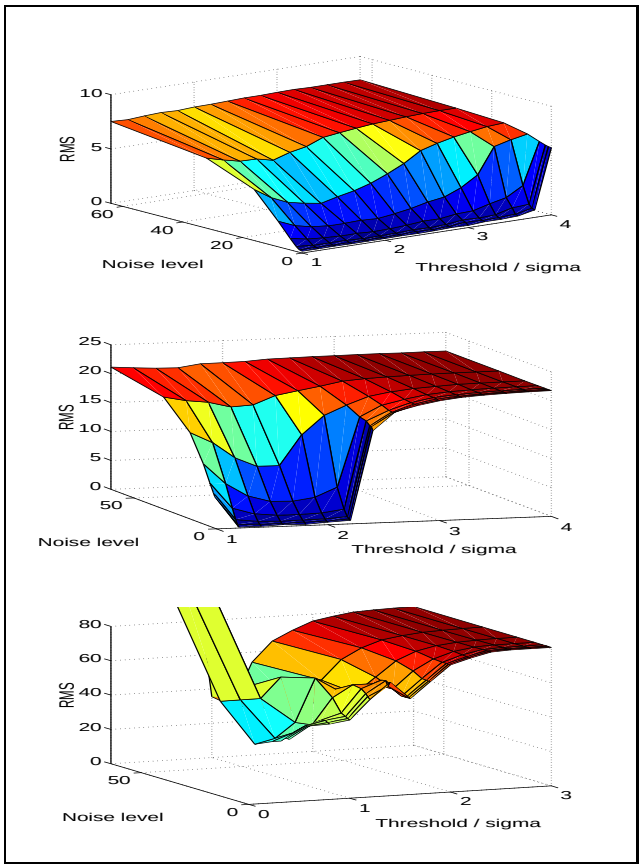

Fig. 8: The rms error values for the iterative B-spline fitting algorithm, plotted against different thresholds and noise situations for; 60 small objects (top), 200 small objects (middle) and 60 larger objects (bottom) on a strong cosine background.

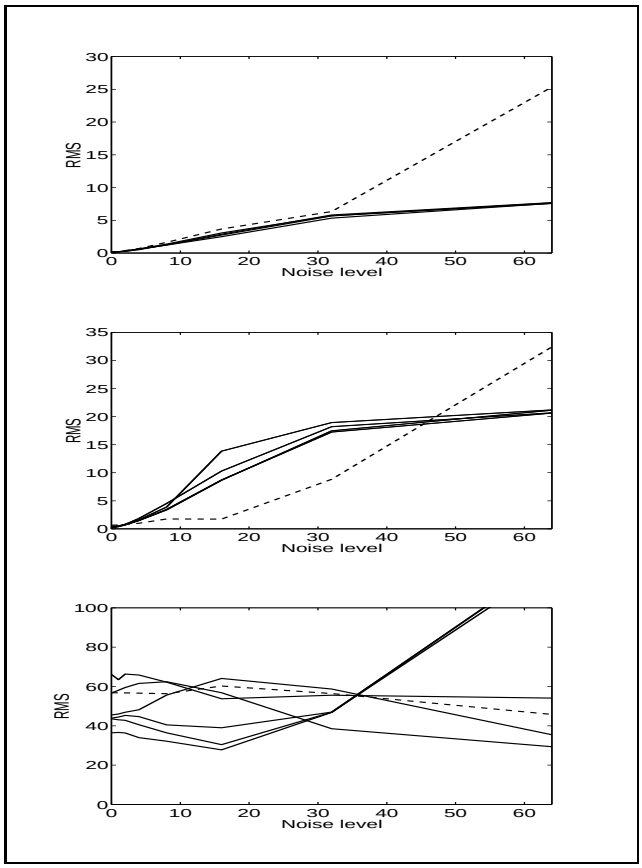

Fig. 9: The rms error values for the best performing threshold values (solid lines) together with the result using the kernel density estimate thresholding scheme (dashed) for; 60 small objects (top), 200 small objects (middle) and 60 larger objects (bottom) on a strong cosine background. Note the super optimal behavior for the KDE estimate in the 200 object image, i.e. the KDE thresholding scheme performs better than any fixed threshold value. This is possible as the KDE version may select different thresholds in different iterations of the INU estimation algorithm.

6. CONCLUSIONS

In comparing homomorphic filtering with iterative B-spline fitting for estimation of shading artifacts in fluorescence images, the B-spline estimate outperformed the spatial filtering in all but one case (noise level 32 in the large object image). This can most easily be seen when comparing figures 7 and 9. Using a kernel density estimate technique to find the threshold needed in the B-spline estimation algorithm seems to be a fairly good choice. It is not always optimal, but in general not far behind the optimal threshold choice and sometimes even outperforming it. Note that the optimal choice of threshold is found by exhaustive search of all thresholds, and the best solution is selected via comparison with the ground truth image. This is of course not possible to do in a real situation, as it requires the solution to be known.

ACKNOWLEDGMENTS

Many thanks to Carolina Wählby who has been very supportive and helpful. Thanks to Amersham Pharmacia Biotech and Fredrik Erlandsson for material and inspiration.

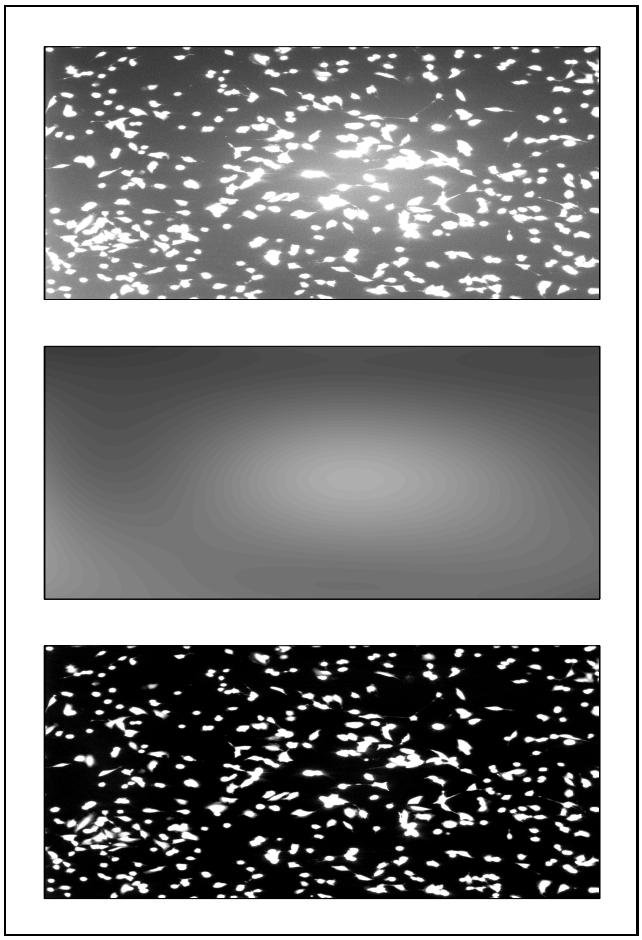

Fig. 10: Real image example with fitted background. Original image before background correction (top), the fitted spline surface of the background (middle) and the image after background subtraction (bottom).

7. REFERENCES

[1] C. Brechbühler, G. Gerig, and G. Székely. Compensation of spatial inhomogeneity in MRI based on a parametric bias estimate. In *Proceedings of the Fourth International Conference on Visualization in Biomedical Computing (VBC)*, pages 141–146, Hamburg, Germany, Sept. 1996. Springer.

[2] B. H. Brinkmann, A. Manduca, and R. A. Robb. Optimized homomorphic unsharp masking for MR grayscale inhomogeneity correction. *IEEE Trans. on Medical Imaging*, 17:161–171, April 1998.

[3] S. Gilles, M. Brady, J. Declerck, J. Thirion, and N. Ayache. Bias field correction of breast MR images. In *Proceedings of the Fourth International Conference on Visualization in Biomedical Computing (VBC)*, pages 153–158, Hamburg, Germany, Sept. 1996. Springer.

[4] R. C. Gonzales and R. E. Woods. *Digital Image Processing*. Addison-Wesley, Reading, 1993.

[5] P. Lancaster and K. Šalkauskas. *Curve and Surface Fitting, an Introduction*. Academic Press, London, 1986.

[6] J. Lindblad. Histogram thresholding using kernel density estimates. In *Proceedings of the SSAB Symposium on Image Analysis*, pages 41–44, Halmstad, Sweden, March 2000.

[7] J. G. Sled and G. B. Pike. Standing-wave and RF penetration artifacts caused by elliptic geometry: an electrodynamic analysis of MRI. *IEEE Trans. on Medical Imaging*, 17:653–662, Aug. 1998.