Numerical optimisation in spot detector design

F. van der Heijden *, W. Apperloo, L.J. Spreeuwers

Department of Electrical Engineering, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands

Abstract

Spots are image details resulting from objects, the projections of which are so small that the inner structure of these objects cannot be resolved from their image. Spot detectors are image operators aiming at the detection and localisation of spots in the image. Most spot detectors can be tuned with parameters. This paper addresses the problem of how to select the parameters. We propose to use carefully designed test images, a performance measure, and numerical optimisation techniques to solve this problem. Several optimisation methods are compared, and their adequacy for spot detector design is tested. © 1997 Elsevier Science B.V.

Keywords: Image analysis; Spot detection; Performance measure; Numerical optimisation

1. Introduction

Spots in images are phenomena which correspond to certain objects in the scene. The projections of these objects are so small (relative to the image resolution) that the internal structure of the objects cannot be observed in the image. An example of spots is the image of a starry night where the projection of a star appears as the point spread function of the imaging system. Spot detectors are image processing algorithms for the detection and localisation of spots in the image plane. Applications are numerous: detection of hot spots in infrared imagery for medical diagnosis and fire detection, particle detection in microscopy images and X-ray imagery for material analysis and medical diagnosis, surface defect inspection, marker detection in navigation systems, and so on.

Spot detection is difficult in the sense that the data representing a spot is local. Therefore, increasing the spatial extent of an operator above a certain limit does not further improve the signal-to-noise ratio (and with that the detection quality). Furthermore, in most applications neighbouring spots and other image structures (edges and lines resulting from other objects in the scene) may interfere with a spot.

There are various approaches to the design of spot detectors, e.g. matched filtering (van der Heijden, 1994), wavelet based filtering (Antoine et al., 1993), and approaches based on a statistical model of the image data, e.g. the cvm operator (van der Heijden, 1995). All of these detectors have parameters to tune the operator so as to match the image characteristics of the application. This paper addresses the problem of how to select these operator parameters so as to arrive at spot detectors that perform well in the circumstances imposed by the application at hand. Our approach to tackle this problem is illustrated in Fig. 1. A test image generator produces an artificial
a test image containing spots. It also generates an accompanying reference map indicating the positions of those spots. The result of a spot detector under design, together with the reference map, is evaluated based on a well chosen performance criterion. A numerical optimisation process tries to optimise this criterion by varying the spot detector parameters.

The success of this design method depends on a number of important aspects: the particular choice of the test images (Section 2), the choice of the performance criterion (Section 4), and the optimisation algorithms with its parameters (Section 5). The functioning of the design method is experimentally tested and analysed (Section 6). The “covariance model based” (cvm) spot detector serves as an example. A number of spot detectors are designed for two different image characteristics with several different numerical optimisation techniques. A resulting spot detector is also tested with a “real” image (Section 6.3). A discussion and conclusions finalise the paper (Section 7).

2. Test images

The test images must be chosen such that they reflect the dominant image characteristics of the application. Furthermore, a reference map must be present that shows the true positions of all spots. An accurate reference map cannot be generated for real images. Therefore, we use synthetic images.

If the size of the projection of an object is below the resolution of the imaging system, it appears in the image as the point spread function \( psf \) of the imaging system. We assume a space invariant psf. The height of the spot depends on the radiometry and the geometry of the object, and on the properties of the imaging device. With that, the \( i \)th spot is modelled with \( \alpha_i \cdot psf(x - x_i, y - y_i) \) where \((x_i, y_i)\) is its position, and \( \alpha_i \) is its height. The background of the image may be described as \( b(x, y) \) and may be constant or contain edge or line like structures caused by interfering objects in the scene. In the continuous domain, the test images can be described as

\[
f(x, y) = \sum_i \alpha_i \cdot psf(x - x_i, y - y_i) + b(x, y) + n(x, y),
\]

where \( f(x, y) \) is the intensity at position \((x, y)\) in the image, and \( n(x, y) \) is 2-D noise. The summation is over all spots \( i \) in the image. The positions of spots is modelled as a “random points in space” process with uniform density \( \lambda \) (= expected number of spots per unit area). Furthermore, in some applications, there is a minimum distance \( R_{\text{min}} \) between any pair of spots.

In our experiments, digital test images \( f_{n,m} = f(n \Delta, m \Delta) \) were generated with sampling period \( \Delta = 1 \). We assumed a constant background \( b(x, y) = b \), a Gaussian distributed spot height with zero mean and standard deviation \( \sigma_a \), and Gaussian white noise with standard deviation \( \sigma_n \). For the point spread function of the imaging device we chose a Gaussian with standard deviation \( \sigma_{psf} \). The overlap between spots is controlled by the ratio \( R_{\text{min}} / \sigma_{psf} \). The probability that overlap occurs depends also on the spot density \( \lambda \).

3. The spot detector

The most straightforward method to detect spots in an image is matched filtering. First, the image \( f_{n,m} \) is convolved with a kernel \( h_{n,m} \) that resembles the shape of the spot. Then, all positions corresponding to a (local) maximum of the convolved image are marked as candidate spots (i.e. non-local maximum suppression). Finally, all candidate spots for which the convolved image exceeds a certain threshold are accepted as detected spots.

The matched filter approach is sensitive to interfering image structures and overlapping spots. Fur-
thermore, it cannot detect spots with negative height. These shortcomings are partly averted with the cvm operator (van der Heijden, 1995, 1992). This operator can be regarded as an extension of matched filtering:

\[ g_{n,m} = \sum_{k=1}^{K} \gamma_k (h(k)_{n,m} * f_{n,m})^2. \]  

(2)

It corresponds to a parallel bank of K filters with kernels \( h(k)_{n,m} \), the squared outputs of which are summed with weights \( \gamma_k \). Non-local maximum suppression and thresholding applied to \( g_{n,m} \) completes the process. Note that with \( K = 1 \) the cvm operator is fully equivalent with the matched filter.

The design of the cvm operator is based on a statistical model of spots. The model roughly resembles the one used for the test images. It has the same parameters. To distinguish between the two sets of parameters, those of the cvm operator are written with a hat: \( \hat{\lambda}, \hat{R}_{\text{min}}, \hat{\sigma}_s \) and \( \hat{\sigma}_{\text{psf}} \). Since for the cvm operator \( \hat{\sigma}_s \) and \( \hat{\sigma}_n \) are dependent, \( \hat{\sigma}_n \) may be set to 1.

One would expect the cvm operator to be optimal if its parameters are chosen equal to the parameters used to generate the test images. This, however, is not true since a limited number of kernels is used and the kernels are truncated.

4. The performance criterion

The performance criterion used to evaluate the performance of a spot detector is a modification of the average risk measure (AVR) used to quantify the quality of edge detectors (van der Heijden, 1992; Spreeuwers and van der Heijden, 1992). Four types of errors are distinguished: (1) missed spots, (2) spuriously detected spots, (3) multiple responses, and (4) localisation errors over a distance \( r \). The AVR performance criterion is defined as a weighted sum over the densities \( \lambda_i \) with which these error types occur:

\[ \text{AVR} = c_1 \lambda_1 + c_2 \lambda_2 + c_3 \lambda_3 + \sum_r c_4(r) \lambda(r). \]  

(3)

The weights \( c_i \) should be chosen to match the application dependent cost of the error types. The AVR resembles the Bayesian risk of a classifier. Lower AVR means a better classifier. For perfect detection the AVR = 0. If all weights are 1 (as in our experiments), the AVR is equal to the density \( \lambda_{\text{err}} \) of errors \( \lambda_{\text{err}} \cdot \text{the expected number of errors per unit area} \). The number of errors in an image with size \( N \times N \) has a Poisson distribution with parameter \( \lambda_{\text{err}} N^2 \). With that, the variance of an estimated AVR becomes

\[ \text{Var} \{ \text{AVR} \} = \lambda_{\text{err}} / N^2. \]  

(4)

5. The optimisation algorithms

The purpose of the optimisation is to find the parameter vector \( \vec{p} = [\lambda, \sigma_{\text{psf}}, \sigma_s, \hat{R}_{\text{min}}]^T \) that minimises the AVR. Recently a number of articles and books, e.g. (Bhanu and Lee, 1994; Harvey and Marshall, 1995), were published in which the use of genetic algorithms is advocated in this kind of problems. Other algorithms might also be attractive:

- Coordinate strategy (CS).
- Powell’s method based on conjugate directions.
- Nelder and Mead’s simplex strategy.
- Gradient strategy (GS).

The genetic algorithm that was used here is described in (Michalewicz, 1994). The implementation is based on Matthew’s Galib C++ library (http://lancet.mit.edu/ga/). The coordinate strategy iteratively performs simple line searches. Powell’s method and Nelder and Mead’s simplex method are described in (Press et al., 1992). The gradient strategy that was used in our experiments is a short step method. In the \( i \)th iteration it updates an estimate \( \vec{p}_i \) with a correction term \(- \beta \nabla \text{AVR}(\vec{p}_i)\).

All optimisation algorithms have variables with which the functioning can be adjusted. In the experiments, all these variables are tuned to values that yield good results for one test image. From that point on, all experiments were done with the variables kept constant.

6. Experiments

The first goal of the experiments is to establish which optimisation algorithm is most suitable. The second goal is to verify that our design method
results in good spot detectors. First, the characteristics of the objective function will be examined. Next, for two types of test images, and for two realisations of these types, various optimisation algorithms will be applied. The search efficiency, the found minima, and the corresponding parameters of these minima are the criteria with which the suitability of the various methods are tested. Finally, we will check whether the spot detectors found also perform well in case of real images.

The two types of test images differ only in the density $\lambda$ of spots and in the minimum distance $R_{\text{min}}$. See Table 1 and Fig. 2 (for the sake of visibility only subimages with $64 \times 64$ pixels are shown). The parameters are chosen such that the two types represent two different settings: one in which no overlap between spots occurs (type $a$), and another in which overlaps occur frequently (type $b$). The image size is $N = 512$. With that, the uncertainty of the estimated AVR is not too large. At the same time the memory requirement is just not impractical.

### 6.1. Function characterisation

To make a good choice between the optimisation algorithms we must have some insight in the behaviour of the objective function. The first step to obtain that is to find out a setting of the parameters where the AVR is close to its minimum, and then to plot the AVR as a function of one of its parameters. The second step is to measure the uncertainties in the AVR and to analyse the propagation to the found optimal parameters.

Fig. 3 shows the four cross sections of the AVR of two types of images. In each graph one parameter is varied while the others are kept constant near the values for which the AVR is minimal. The two types are generated with the parameters shown in Table 1. For each type the plots of two different realisations are given.

According to Eq. (4), the standard deviation in the estimated AVR is 0.0001 (type $a$) and 0.0003 (type $b$). The plots in Fig. 3 are in accordance with that. Another observation in Fig. 3 is that the objective function looks smooth with a clear global minimum. Some small local minima occur due to the uncertainty in the evaluation of the AVR. Here and there, some step-like transitions can be seen. Upon closer examination it appeared that the transitions are caused by truncations in the number $K$ of kernels in the cvm-operator.

The width $\hat{\sigma}_{\text{psf}}$ of the modelled spot and the minimum distance $\hat{R}_{\text{min}}$ have plateaus where the objective function hardly changes. The plateaus adjoin steep slopes. With respect to the modelled density $\hat{\lambda}$ and the noise $\hat{\sigma}_n$ the shape of the AVR around the minima is rather skew. (Note that in Fig. 3 the scale of $\hat{\lambda}$ is logarithmic.)

It is important to know the uncertainty of the estimated optimal parameters since that knowledge can prevent a lot of useless AVR evaluations. Due to the finiteness of the test image the calculated AVR has an uncertainty induced by the randomness of the

<table>
<thead>
<tr>
<th>Type</th>
<th>Width $\sigma_{\text{psf}}$</th>
<th>Height $\sigma_n$</th>
<th>Noise $\sigma_e$</th>
<th>Offset $\lambda [1/\hat{\lambda}]$</th>
<th>Density $\hat{\lambda}$</th>
<th>Inh. distance $R_{\text{min}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td>1</td>
<td>1</td>
<td>0.1</td>
<td>100</td>
<td>0.008</td>
<td>5</td>
</tr>
<tr>
<td>$b$</td>
<td>1</td>
<td>1</td>
<td>0.1</td>
<td>100</td>
<td>0.05</td>
<td>2</td>
</tr>
</tbody>
</table>

Fig. 2. Test images with accompanying reference map.
Fig. 3. AVR of two realisations of two types of images.

Test image: $\text{AVR}_{\text{calculated}}(\hat{p}) = \text{AVR}(\hat{p}) + m(\hat{p})$ where $m(\cdot)$ is the random part. Suppose that the true optimal parameter vector is $\hat{p}_{\text{opt}}$. If we assume that in a small neighbourhood of $\hat{p}_{\text{opt}}$, the AVR can be approximated with a second order Taylor series expansion, then the difference between the calculated optimal parameters and the true optimal parameters $\hat{e} = A^{-1} \hat{m}$. Here, $A$ is the Hessian matrix of

Fig. 4. Search efficiency.
AVR(\(\vec{p}_{\text{opt}} + \vec{e}\)), and \(\vec{v}_{\text{m}}\) is the gradient vector of \(m(\vec{p}_{\text{opt}} + \vec{e})\), see (Press et al., 1992). The uncertainty itself is quantified by the covariance matrix of \(\vec{e}\):

\[ C_{e} = E\{\vec{e}\vec{e}^{T}\} = A^{-1}C_{v_{m}}A^{-1}. \]  

(5)

\(C_{v_{m}}\) is the covariance matrix of the gradient vector \(\vec{v}_{m}\). Both \(A\) and \(C_{v_{m}}\) are unknown, as yet. However, they can be estimated from a number of realisations of the test images using finite differences. Application of this procedure to 20 realisations of the two types of images revealed that both \(C_{v_{m}}\) and \(C_{e}\) are almost completely uncorrelated. The standard deviations of the optimal operator parameters (found as the square root of the diagonal elements of \(C_{e}\)) are given in Table 2. The results are in accordance with Fig. 3. Note, for instance, that the uncertainty in \(\hat{R}_{\text{min}}\) for type \(a\) is rather high. This corresponds well to the fact that the optimum of AVR is found on a horizontal plateau of \(\hat{R}_{\text{min}}\).

### 6.2. Results of numerical optimisation

The various optimisation algorithms have been applied to the test images. The algorithms were all (except GA) initiated with operator parameters set to the corresponding image parameters. The maximum number of AVR evaluations was about 750. If during 25 iterations an algorithm did not improve its AVR, it was presumed that the process had converged. Fig. 4 shows the AVR versus the number of AVR evaluations for one test image. A step in a plot corresponds to the end of an iteration. The example is typical for a type \(a\) image. With a change of scale of the vertical axis it is also typical for a type \(b\) image.

Table 3 gives the results of applying the five algorithms to two realisations of each type of test image. The calculated optimal parameters and corresponding AVRs are averaged over the two realisations and over the five algorithms. The sample standard deviations are also given. Since there were no outliers, we conclude that in all cases all algorithms converged to the same AVR. However, Fig. 4 shows that the search efficiency of the algorithms differs a lot.

#### 6.3. Application to a real image

The cvm spot detector obtained with the type \(b\) image in the previous section is depicted in Fig. 5. The operator consists of 12 convolutions, the kernels of which are shown as bitmaps in Fig. 5(a). Here, a 50% grey level corresponds to zero. Black is a negative kernel element; white is a positive element.

<table>
<thead>
<tr>
<th>Table 2</th>
<th>Standard deviation of calculated optimal operator parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Width</td>
</tr>
<tr>
<td>(a)</td>
<td>0.15</td>
</tr>
<tr>
<td>(b)</td>
<td>0.09</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 3</th>
<th>Average and sample standard deviation of calculated optimal operator parameters (av. is average; ssd is sample standard deviation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>AVR</td>
</tr>
<tr>
<td>(a)</td>
<td>(\hat{a}_{\text{avg}})</td>
</tr>
<tr>
<td>(b)</td>
<td>(\hat{a}_{\text{avg}})</td>
</tr>
</tbody>
</table>

Fig. 5. (a) cvm spot detector. (b) Input image. (c) Detected spots.
7. Discussion and conclusion

Our experiments indicate that all algorithms involved can find the optimal parameters of the spot detector within the predicted accuracy. The numbers of function evaluations, however, are much different. The simplex method and the gradient strategy appeared to be efficient. The genetic algorithm advocated in (Bhanu and Lee, 1994; Harvey and Marshall, 1995) is less efficient in this application. The reason is that our objective function is smooth, and that there are no local, deep minima. With that, methods that explicitly or implicitly use gradient information are applicable. The smoothness is due to two factors. The first one is our choice of the parameter space of the operator. This choice is model driven. Therefore, the parameters match the characteristics of the test image. This is quite opposite to, for instance, the case in (Harvey and Marshall, 1995) where the parameter space is spawned by the structuring elements of a number of morphological operators and the ordering of these operators. The second factor is the large image size. This induces a good signal-to-noise ratio in the objective function.

Discussion

Sklansky: I would like to suggest a very nice and important problem that this could be applied to: detection of microcalcifications in mammograms. This is a quite challenging and important problem and I would be delighted to work with you if you are interested.

Van der Heijden: I am very much interested, thank you.

Mardia: These spots seem to have dimensions: height, width. What you show are small hot spots, but they could probably be much bigger.

Van der Heijden: That is true. I have defined spots as being represented by the point spread function. I don’t think that is strictly necessary. What we have done also is for instance detectional deletes of electronic components on a PCB. In the image, the dimension of deletes is much larger than the point spread function. In that case a spot detector also worked well.

Mardia: Then this average risk function, you took it to be linear. Is that what you believe in?

Van der Heijden: I think that’s depending on the application. Also the constants, which are inside this expression are application dependent.

Mardia: That is what I thought.

References