Detection of Spots in 2-D Electrophoresis Gels by Symmetry Features

Martin Persson and Josef Bigun

School of Information Science, Computer and Electrical Engineering,
Halmstad University, P.O. Box 823
SE-301 18 Halmstad, Sweden
Martin.Persson@ide.hh.se

Abstract. We have implemented an algorithm for detection and segmentation of protein spots in 2-D gel electrophoresis images using symmetry derivative features computed using low level image processing operations. The implementation was compared with a previously published Watershed segmentation and a commercial software. Our algorithm was found to yield segmentation results that were either better than or comparable to the other solutions while having fewer free parameters and a low computational cost.

1 Summary

Two-dimensional gel electrophoresis (2-DE) is a major workhorse in proteomics. 2-DE data comes as spot maps containing a vast number of proteins, requiring automatic image processing for efficient analysis. Quantification of individual proteins and tracing changes in expression between gels require accurate spot detection. Existing spot detection algorithms often require user intervention for setting free parameters and time consuming morphological post processing. We approach the problem by using a set of computationally cheap and robust symmetry derivative features and minimal post processing. A feed forward neural network is used to find decision boundaries in the feature space. The neural network is trained with features extracted from manually segmented 2-DE images. Classification performance is compared with the published non-commercial algorithm of Bettens [1] and one commercial 2DE image analysis program, ImageMaster™ 2D Platinum v5.0 (GE Healthcare, formerly Amersham Biosciences). The result, presented as ROC curves, show that we perform at least as well as both Bettens and Imagemaster in terms of spot detection and segmentation, while using fewer free parameters, and a limited amount of computational resources.

1.1 Originality and Contribution

We propose a set of symmetry derivative features [2, 3] to be used in automatic segmentation of 2DE gel images with minimal post-processing. Symmetry derivatives give immediate information on local shape that otherwise requires time consuming regional processing. In addition to achieving better segmentation performance, this moves the focus of the problem from post processing to basic signal processing.

S. Singh et al. (Eds.): ICAPR 2005, LNCS 3686, pp. 436 – 445, 2005.
© Springer-Verlag Berlin Heidelberg 2005

1.2 Introduction to the Problem

2-DE [4] is able to separate thousands of proteins in a sample, presented as image data. Spot detection and segmentation into spot regions is of central importance for the quantitative and differential analysis of proteomics experiments. The segmentation is complicated by the large range of protein concentration affecting spot geometry, overlapping spots, irregular spot shapes, and random noise.

1.3 Alternative Spot Detection Techniques

Most spot detection solutions are closed source, making fair comparison difficult. However many older image analysis packages [5, 6] applied model fitting for direct segmentation of the protein spots. The alternative approach is to use a crude initial segmentation followed by computation of morphological and grey level features of the initial regions for a final decision. The Laplacian of Gaussian (LoG) filter response is a weak feature that has been widely used in segmentation [7, 8, 9]. In recent years the unsupervised Watershed algorithm has also become a popular choice [1, 9] for initial 2DE image segmentation. The second step often consists of iterative model fitting within the regions [1, 8, 10] to closer determine spot properties. Fitting each segmented area to a model is often a computationally expensive step.

1.3.1 Gaussian Fitting

The 2-D Gaussian function is used for smoothing and noise removal in image processing, and is also the most common protein spot model:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad . \tag{1}$$

As the function is separable smoothing can be done either with a 2-d convolution with the Gaussian kernel above or by convolution with two orthogonal 1d kernels. The reason the Gaussian is a popular in spot modeling is that the Gaussian function corresponds to diffusion from a single point. Most spots are formed under conditions similar to diffusion in the MR/PI directions, making the model appropriate for many spots. The main weaknesses are poor modeling of irregularly (ie non ellipsoid) shaped spots, and in fitting with saturated “flat top” spots. Bettens as well as Rogers [8] address the flat top spot problem by assuming diffusion from a central area rather than a point, representing the spot as a central disc or irregularly shaped region convolved with a Gaussian kernel. In spot detection the Gaussian is used either by iteratively optimizing the parameters around a peak [5] with respect to a residual, or by similar fitting to a segmented region that is expected to contain one spot only.

1.3.2 Second Derivative

The second derivative gives information on the curvature of the local surface. As protein spots appear as dark blobs on a white background they have convex curvature. This weak criterion is widely used in initial spot segmentation [7, 8, 9]. Since the second derivative amplifies noise a smoothing operation is often applied before the derivative is computed. In practice the computation of the second derivative of a gray scale image is often implemented by means of 2-d convolution with the LoG filter

$$\mathrm{LoG}(x, y) = \frac{-1}{\pi\sigma^4}\left[1 - \frac{x^2 + y^2}{2\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}} \quad . \tag{2}$$

which in the case of local concave curvature gives a positive value on a spot.

1.3.3 The Watershed Transformation

The watershed transformation (WST) is a powerful segmentation algorithm commonly used for segmentation of locally homogenous grey value images that has been implemented in linear time [11]. The transform finds the reliefs that separates catchment basins around local minima. In 2-DE the WST has been applied to a smoothed grey scale image [1] and to the gradient strength image [9]. The WST tends to over-segment spots when applied to the gradient strength image. Usually heuristics are applied to merge or discard regions in a computationally costly post processing step.

2 Method

2.1 Features and Feature Extraction

For each pixel we generate a feature vector containing the local LoG transform of the original image and the local symmetry derivative response. The LoG captures the concavity of spot surfaces while symmetry derivatives capture spot shape. Symmetry derivatives are powerful textural features that have a wide range of applications in image processing [3, 12, 13]. The main strength of symmetry derivatives is an ability to represent local shape without initial segmentation of the raw image, a costly step in terms of computation. Symmetry Derivatives are differential operators that are based on

$$D_x + iD_y = \partial/\partial x + i\partial/\partial y \; . \tag{3}$$

which yields a complex vector field when applied to images. Higher order symmetry derivatives of the n'th order, and their conjugates, are defined as

$$(D_x + iD_y)^n \; . \tag{4}$$

$$(D_x - iD_y)^n \; . \tag{5}$$

respectively. Applying a differential operator to an image corresponds to convolution with a derivative kernel in the direction of the differential combined with a window function. We choose the discrete Gaussian function and its derivative as window function and derivative kernel respectively. In our application we first use derivative operators to compute the local orientation map of an image

$$Z=(dxf+idyf)^2 \; . \tag{6}$$

Further smoothing of Z (convolution with a 2-d window function such as the Gaussian kernel) of the local orientation map directly gives the local Complex moments I_{20} and I_{11}. Complex moments of order m+n are defined as:

$$I_{mn} = \iint (u+iv)^m (u-iv)^n \left|F(u,v)\right|^2 dudv \ . \tag{7}$$

I_{20} can be computed as the local weighted sum of Z, while I_{11} is computed as the weighted sum of the absolute values of Z. The relationship between the magnitude of I_{20} and I_{11} represents how well the local power spectrum of the image f fits to a line, and is widely used for edge and corner detection. Working on the local orientation map in this manner corresponds to using the 0'th order symmetry derivative operator, or linear symmetry. If we instead choose further application of the Symmetry derivative operators before smoothing, then that corresponds to a coordinate transform followed by fitting to a line in the power spectrum. The coordinate transform and the following line fitting also corresponds to fitting certain symmetric shapes to the image in the Cartesian (x,y) coordinate system. We are using the first and second order conjugate symmetry derivatives which corresponds to parabolic and spiral (Circular Symmetry, CS) structures in Cartesian coordinates as shown in *Fig 1*. The choice is motivated by spot geometry. The Local orientation map of spot centers show strong similarity to the Local orientation map of ideal circular patterns, and the spot boundaries analogously have a parabolic structure. As D_x and D_y are implemented with separable 1-D Gaussian filters the computation is very fast.

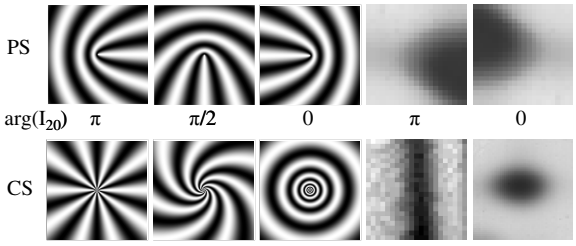

Fig. 1. Phantom and 2-DE patterns in the Cartesian coordinate system and the corresponding argument of the moment I_{20}

We compute two types of moments for the circular and parabolic symmetry derivative responses. For each type we have I_{20} as the local sum of the complex response weighted by a complex filter, and I_{11} as the sum of the magnitudes of the complex responses weighted with the magnitudes of the complex filter. Parabolic symmetry requires 8 1-dimensional convolutions, circular symmetry another 4 (circular symmetry can be computed as applying symmetry derivative operators to parabolic symmetry). Computation of the moments I_{20} and I_{11} requires another 4 convolutions for each type, for a total of 20 1-D convolution operations.

2.2 Classification of Pixels and Final Segmentation of the Gel

We use a Feed forward Artificial Neural Network (ANN) for initial classification of the pixels based upon our feature set. A two layer ANN with a non-linear transfer function is chosen as such a network can find an arbitrarily close approximation of any function or decision boundary [14]. We choose the hyperbolic tangent function as

transfer function in the hidden layer and a logistic sigmoid function as transfer function in the output layer. We use two output nodes representing the posterior probabilities of a pixel belonging to the background or a spot core, and let the experiments described in **3.2.1** determine the number of nodes in the hidden layer. Training is done by Resilient Backpropagation [16]. As the neural network ignores spatial continuity in the data the initial result can be expected to be undersegmented. We first split the segmented regions along the local minima of real(I_{20}) for CS. The split segments are then kept if their size is above a certain threshold.

3 Data Set and Experimental Design

3.1 Data Set

Training and validation data were obtained from 8-bit tif images of eight silver stained gels (four human and four E-coli) with varying signal to noise ratio and background intensity. We keep two E-coli gel images as test data. In addition we used an artificially generated gel [16] with known spot positions (available for download at http://www.isbe.man.ac.uk/~mdr/content.php?f=electrophoresis). The continuous regions of the smoothed real gel images whose second derivative is above zero were split along the watershed lines of the Laplacian strength image. The resulting regions were manually inspected and if necessary merged with their neighbours. Finally the resulting regions were eroded by two pixels to eliminate the smallest spots and to separate spot cores from boundaries.

The following features were computed from the 8 bit grey value images:

$\mathbf{x}_1$= real(I_{20}) for CS $\mathbf{x}_2$= imag(I_{20}) for CS $\mathbf{x}_3$=I_{11}-|I_{20}| for CS $\mathbf{x}_4$=|I_{20}|/I_{11} for CS
$\mathbf{x}_5$= real(I_{20}) for PS $\mathbf{x}_6$= imag(I_{20}) for PS $\mathbf{x}_7$= |I_{20}|/I_{11} for PS $\mathbf{x}_8$= LoG

Pixels for training of the neural network were selected as follows: For the spot core class all pixels belonging to the identified spot regions in the training set were chosen. For the background class we chose all the pixels within a cityblock distance of two pixels from the spot core, and a number of pixels equal to that of the spot core class were randomly chosen from the remaining pixels.

3.2 Experiments

3.2.1 Selection of Features and Parameters for ANN

Parameter selection is carried out using the Backward Elimination technique. The method consists of two steps. First we set the number of nodes by changing the number of nodes in the hidden layer between 1 to 15 and performing five-fold cross-validation for each configuration. In the second step we use the number of nodes that gave the lowest number of misclassified pixels in a leave one feature out experiments. The feature with the smallest effect on the total error was then removed, and the two steps repeated with the reduced feature set. The result is presented as error bars for the best number of nodes for each number of features.

Table 1. Test parameters

Algorithm	Parameter	Values
Symmetry derivative	Minimum Core size	0 to 19
ImageMaster™ 2D Platinum v5.0[1]	Saliency	1, 2, 5, 10, 20
Bettens Watershed[2]	Minimum watershed size	10 to 160 in increments of 10

[1]Imagemaster also has the parameters *Smooth* and *Min Area* which were held constant at 2 and 5 respectively as experiments showed that they have little effect on the performance on our data set.
[2]Bettens algorithm has one more parameter, *Maximum Grey value in Watershed*, that we after experiments decided to hold constant at 238 for the real images and 250 for the artificial images.

3.2.2 Evaluation of Segmentation Performance

In the final segmentation experiments we feed the neural network with a reduced feature set for an initial classification. Post processing is done by splitting the regions at local minima of the feature $x_{1,}$, followed by discarding regions of a size smaller than a certain threshold.Using a segment of a real silver stained image containing 215 manually identified spots and an artificial silver stained image with 924 spots we compute precision and recall with respect to spots found. A spot is considered valid if the centre of the segmented region is within the identified core of a real gel, or within 20% of the standard deviation plus three pixels distance of the centre of the artificial spots. As the watershed segmentation only gives target regions for later gaussian fitting we provide an alternative segmentation performance measure as well. For these we consider all segmented regions that overlap precisely one spot center to be valid, giving a measure of the potential improvement from post processing. Precision and recall is computed for our algorithm as well as the Watershed of Bettens and Imagemaster 2D Platinum. The result is presented as ROC curves with respect to different parameters of the algorithms (*Table 1*).

3.2.3 Comparison of Execution Time

As Image Master is a closed source package we choose measure execution time in order to estimate the comparative computational complexity. Cropped versions of a real image with sides 128, 181, 256, 362, 512, 724, and 1024 pixels were used for the experiment. The Symmetry derivative and Watershed based approaches were implemented in Matlab 7.0. For the testing of Image Master we used a downloadable trial version (available at http://www1.amershambiosciences.com/). Execution times for the Matlab implementations were measured using Matlab's internal timer functions, while the execution times for Image Master were measured manually with a stopwatch. All experiments were performed under Windows XP Professional on a AMD Athlon XP2400+ with 1Gb RAM. The results are presented as a log-log plot of execution time vs the number of pixels.

4 Results and Interpretation

4.1 Feature Selection and ANN Parameter Selection

Figure 2 shows the crossvalidation error and standard deviation after removing features. Four features can be removed without hurting performance. The redundant

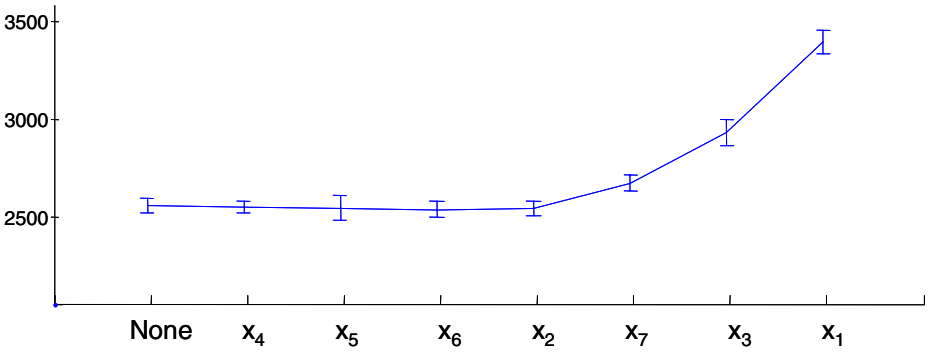

Fig. 2. Crossvalidation error and standard deviation of the average error after successive removal of features. The features were removed in the order listed from left to right.

features represent direction of parabolic symmetry, lack of CS, and swirly spiral patterns. With the exception of the lack of CS this is unsurprising. However one of the remaining features represents relative strength of CS, a property that is related to lack of circular symmetry accordingly. We choose to keep features 1, 3, 7, and 8.

4.2 Segmentation Performance

Fig. 3 shows ROC curves for the experiments. The Symmetry based approach consistently yields a higher recall compared to the most sensitive setting of Image Master, but also finds a higher number of false positives. A large subset of the false positives found by our algorithm are caused by splitting of actual spots. Such false positives are less of a problem than false negatives when the output of a spot detection algorithm is used to find targets for MS analysis, as the goal often is to find low abundance novel proteins. Image Master achieves the highest precision, but never manages to achieve an equal error rate even on the artificial gel. The comparison with the Watershed algorithm shows that we achieve comparable initial segmentation performance except for in the case of the artificial gel. The artificial gel used in the test has an unrealistically low noise level and thus lack the non-spot catchment basins that otherwise cause the Watershed algorithm to oversegment an image. The difference between the segmentation and detection curves shows the potential improvement of the algorithms by further post processing. One low cost improvement could be to compute the spot center using a method that takes pixel values into account rather than computing the center of gravity of the regions only, as done in this paper. Our segmentation approach returns smaller regions, corresponding to spot cores, compared to Watershed and Imagemaster, which would result in computationally cheaper post processing. The results closest to an equal error rate for each algorithm are show in *Table 2*. The results for Image Master are from detection at the most sensitive settings.

4.3 Computational Complexity and Execution Time

Fig 4 shows a log-log plot of execution time vs the number of pixels in the image. A realistic image of size 1024*1024 pixels is segmented in 10s by Image Master, 31s by

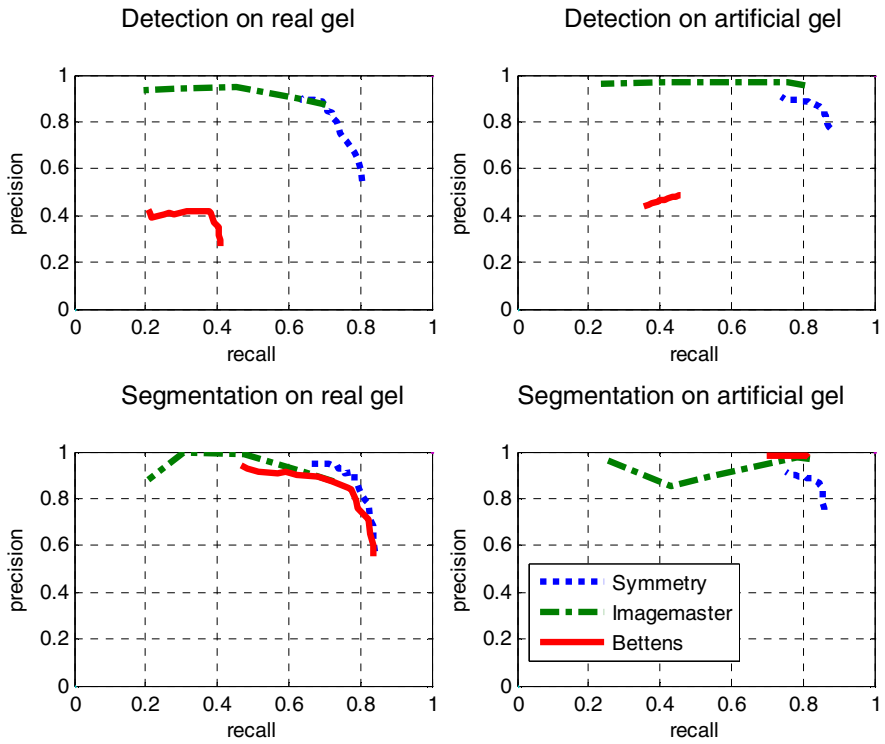

Fig. 3. ROC curves for spot segmentation and detection on real and artificial gels

Table 2. Best performance

	Symmetry Derivatives		Watershed		Imagemaster	
Detection	Precision	Recall	Precision	Recall	Precision	Recall
Real gels	73%	74%	39%	39%	87%	70%
Virtual gel	85%	85%	51%	43%	96%	80%
Segmentation	Precision	Recall	Precision	Recall	Precision	Recall
Real gels	79%	81%	79%	79%	88%	73%
Virtual gel	84%	85%	93%	88%	97%	82%

the Symmetry Derivative based method, and 560s by the Watershed based method. The execution time for the Symmetry derivative approach and Image Master is linear, while our implementation of the Watershed segmentation has much worse performance. The poor performance of the Watershed based segmentation stems from the post processing that is dependent on the number of watersheds as well as their size. Our implementation also uses loops which are very inefficiently implemented in Matlab, further adding to the execution time. Execution times for the Symmetry based approach were on average 3.2 times longer than for Image Master. This difference is remarkably small as Image Master is a compiled, optimised software, while the

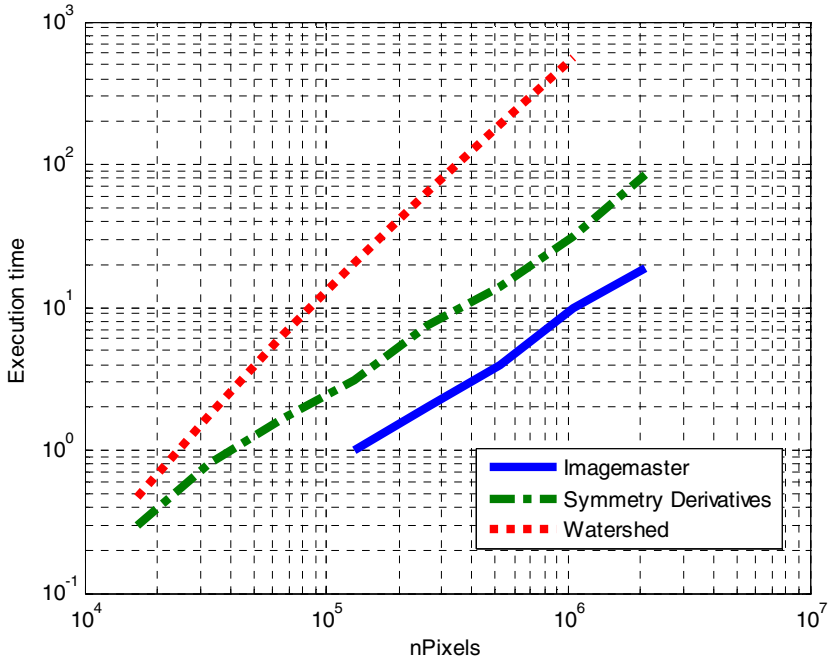

Fig. 4. Execution time vs number of pixels in the image. Image Master had execution times lower than 1 second for images with fewer than 131044 pixels.

Symmetry Derivative based segmentation was implemented in an interpreting language using an inefficient double precision representation of all data.

4 Conclusions

In this paper we have presented a novel 2-DE image segmentation algorithm based upon Symmetry Derivatives and compared it with the widely used Watershed segmentation technique and a commercial software package. We achieve a better equal error rate and a better performance on the recall measure compared to Imagemaster, and comparable or results to a Watershed segmentation approach that relies on significant post processing. The comparison shows that we achieve equivalent or better spot detection compared to the other approaches, using only the basic signal processing operation of one-dimensional convolution and a size criterion.

References

[1] Bettens E, Scheunders P, Van Dyck D, Moens L, Van Osta P, Electrophoresis, 18, pp 792-798, 1997

[2] Bigün J., Bigün T., Nilsson K., *IEEE-PAMI* 26, pp 1590-1605, 2004

[3] Persson M, Bigün J, In J. Bigun and T. Gustavsson, editors, *Scandinavian Conference on Image Analysis*, volume LNCS-2749, pp 520-525. Springer, 2003
[4] Görg A, Weiss W, Dunn M.J., Proteomics 2004, no 4, pp 3665-3685
[5] Garrels J.I., J. Biol. Chem. 1989, 264, pp 5269-5282
[6] Appel R. D., Vargas J. R, Palagi P. M, Walther D. et al Electrophoresis 1997, 18, pp 2724-2734
[7] Baker M, Busse H, Vogt M, Medical Imaging 2000: Image Processing, San Diego, SPIE, Bellingham 2000, 2979, pp 426-436
[8] Rogers M, Graham J,. Tonge R.P., Proteomics 2003, no. 6, pp 887-896
[9] Pleissner, K.-P., Hoffman F, Kriegel K, Wenk C, Wegner S, Sahlström S, Oswald H, Alt H, Fleck E, Electrophoresis 1999, 20, pp 755-765
[10] Mo X, Wilson R, In J. Bigun and T. Gustavsson, editors, *Scandinavian Conference on Image Analysis*, volume LNCS-2749, pp 430-437. Springer, 2003
[11] Vincent L, Soille P, IEEE PAMI, 1991, 13, pp 583-598
[12] Nilsson K, Bigün J, Pattern Recognition Letters, 24, pp 2135-2144, 2003
[13] Premaratne H. L., Bigün J, Pattern Recognition, 37, pp 2081-2089, 2004
[14] Funahashi, K, Neural Networks 2 (3), pp 183-192
[15] Riedmiller M, Braun H, In Proc. of the IEEE Intl. Conf. on Neural Networks, San Francisco 1993
[16] Rogers M, Graham J,. Tonge R.P., Proteomics 2003, no. 6, pp 879-886