ACTA CHROMATOGRAPHICA, NO. 13, 2003

COMPARISON OF IMAGE-TRANSFORMATION METHODS USED IN MATCHING 2D GEL ELECTROPHORESIS IMAGES

K. Kaczmarek[1], *B. Walczak*[1], *S. de Jong*[2], and *B. G. M. Vandeginste*[2]

[1]Institute of Chemistry, Silesian University, 9 Szkolna Street, 40-006 Katowice, Poland
[2]Unilever Research and Development Vlaardingen, Olivier van Noortlaan 120, 3133 AT, Vlaardingen, The Netherlands

SUMMARY

Automatic alignment (matching) of two-dimensional gel electrophoresis images is of primary interest in the field of proteomics. The method of 2D gel image matching proposed in a previous paper [1] is based on fuzzy alignment of features extracted from gels' images and enables both global and local interpolation of the image grid, followed by brightness interpolation. There are two types of pixel coordinate mapping (forward and inverse) and the choice of mapping method might affect the quality of the results obtained. In this paper the two types of grid mapping method are compared.

INTRODUCTION

In the last few years one of the most promising areas in life sciences was genomics, which was supposed to give answers about all human diseases. After some time, however, all the enthusiasm vanished because it was revealed that most diseases are caused by variations in the protein composition of cells. For this reason there was a shift of research interest from genomics to proteomics. The term proteome describes all the proteins present in the cell and thus proteomics entails identification of proteins and determination of their physiological relevance. The impossibility of predicting the proteome on the basis of gene sequencing is because any single gene can correspond to as many as 50 different proteins, because of mRNA splicing and because of post-translation interactions and modifications. For these reasons the protein composition of a cell might be substantially modified even when the genome remains unchanged. Proteomics has a major disadvantage over genomics, however – protein analysis is much more difficult than gene analysis, mainly because of the secondary and tertiary structure of proteins, the inability to perform amplification in the same way as for DNA, the possibility of denaturation, and the poor solubility of some proteins.

The first step in proteomics analysis is protein separation, which should be followed by identification of specific proteins and determination of their effects. Many techniques are used to separate proteins extracted from a cell or tissue, e.g. chromatography or capillary electrophoresis; the most powerful is two-dimensional gel electrophoresis, introduced in 1975 by O'Farrell [2]. This method of separation combines two modes of electrophoretic separation which exploit completely independent property of the molecules – their molecular masses and charges [3]. Utilization of these two properties enables separation of as many as 5000 different proteins in a single analysis. After separation the proteins are visualized by staining with different dyes, silver compounds, or radiolabeling. Although there is currently no substitute for this technique, there are several areas which can be still improved [4]. Major issues concerning 2D gels are discussed in detail elsewhere [5]. One of the main problems of proteomics is the lack of automation of the whole analysis, especially automatic processing of the results obtained, e.g. 2D gel image analysis in comparative studies of biological materials. The possibility of comparison of gel images might give important information about the effect of the presence or absence of some of the proteins on the state of the tissue investigated. For example, to investigate the difference between healthy and diseased patients it is necessary to compare many samples from both groups to discover differences caused by the individual variety of the patients and to identify proteins related to the disease under investigation. By comparing healthy and diseased tissues at the protein level it is possible to identify potential protein targets as diagnostic markers, to evaluate drug efficiency and toxicity, or to design the appropriate drugs for treatment of the disease [6,7]. The possibility of simultaneous comparison of many gel images is also of great importance to chemists for optimization of protein separation.

Although many software packages are available for analysis of 2D gel images [5,8–15], most require intensive user interactions; in high-throughput analysis this is one of the major bottlenecks of entire analysis. 2D gel images can be compared in two different ways; one is based on extracted features (e.g. spot coordinates) [8–13,16–18] whereas the other compares whole images [14,15,17,19]. In the first approach the features must be extracted before the comparison, which requires detection and quantification of protein spots in each gel image. This approach is often associated with manual steps such as choice of threshold value for estimation of the background and selection of landmark spots (control points). When the features are extracted automatically the lists of unlabeled features obtained require efficient algorithms capable of dealing simultaneously with feature correspondence and feature mapping. In the second approach, the whole gel images are used. Gel images are first aligned with each other according to their intensity distribution, and then compared. The effectiveness of this approach is highly dependent on the similarity measure, the transformation function, and the optimization procedure used.

The primary aim of our study was to compare two different methods of pixel coordinate mapping, to map the coordinates of pixels in an image, on the basis of a method developed for comparison of the 2D gel images [20]. The secondary goal was to determine whether the use of one of these methods had a substantial impact on the quality of the results obtained.

THEORY

Notation

The standard image is denoted ‘A’ whereas the image being matched is denoted ‘B’. The transformed image B is denoted ‘t(B)’. The coordinates of a pixel from image A are denoted ‘(x_A, y_A)’ Features extracted from images are organized in matrices and denoted ‘**A**’ and ‘**B**’, respectively. ‘$I(x, y)$’ denotes the brightness of the pixel with coordinate (x, y). Coordinates of non-pixel points are given as (x^*, y^*).

Matching of the Images of 2D Gel Plates

To match two (or more) digital images it is necessary to calculate a geometrical transformation (mapping function) which enables transformation of all points (pixels) from the matched image (B) to the standard image (A). In the next step the brightness of each point of the transformed image t(B) must be interpolated. Complete matching consists of two steps – transformation of the geometrical coordinates of all the pixels and determination of their intensity. To perform the first step, the parameters of transformation must be known.

In our previous paper [1] the feature-based strategy of image matching was developed. This uses features from images to calculate the geometric transform parameters. The mapping function can be determined on the basis of a few corresponding points from images, called control points or landmarks, chosen from the set of extracted features. The fifty most intense spots were chosen as the features, and described in terms of the coordinates of their centers, areas, and intensities. Spots with very irregular shapes and areas exceeding a certain value were discarded and spots for which there were no spots in the second image within a predefined neighborhood were also removed [1].

Determination of Correspondence and Transformation Parameters

Features of the image can be chosen manually, which enables correspondence between selected features to be specified simultaneously. Manual selection of features is, however, very burdensome and time-consuming. For this reason automatic extraction of features is preferred, even though automatic extraction results in lack of information about correspondence between the features. Because of this aspect of automatic extraction there is a need for determination of the correspondence between features which enables selection of control points. Without a knowledge of the transformation which will enable features to be mapped it is, however, impossible to determine the correspondence between these features. In turn, correspondence between features must be known before the transformation could be calculated. As a workaround of the problem of simultaneous calculation of the correspondence and transformation parameters the fuzzy-matching procedure was proposed [1]. This entails iterative repetition of two steps in the procedure:

0. initialize correspondence among features;
1. calculate actual transformation parameters for given correspondence among features;
2. calculate actual correspondence for transformed features; and
3. repeat steps 1 and 2 until convergence is attained.

Correspondence among features is determined by fuzzy matching of all pairs of features with different degrees of fuzziness, i.e. at the beginning of the algorithm the degree of fuzziness has a high value and is decreased in consecutive steps. The use of fuzzy correspondence for calculation of the transformation parameters ensures that only the corresponding features effect the transformation parameters.

Mapping of Pixels

The transformation parameters are information about the differences between matched images and can be used to transform the image B in such a way that it will match the standard image A as closely as possible. This is done in two steps:

1. geometrical grid transformation
2. pixels' brightness determination

Grid transformation is used to determine the coordinates of pixels from image t(B) and can be done in two different ways. The first method aligns coordinates of pixels from image B to coordinates of pixels from image A, whereas the second method aligns coordinates of pixels from image A to coordinates of pixels from image B. The first method is called forward transform; the second is the inverse transform (Fig. 1) [21]. After

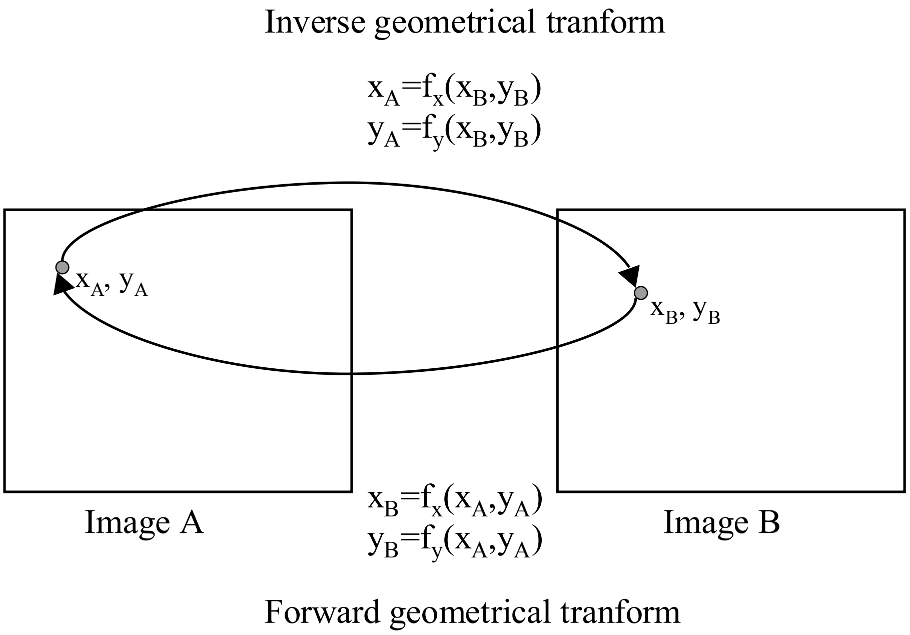


Fig. 1

Forward and inverse transform of geometrical coordinates

the geometric transformation of the grid (of pixel coordinates) it is necessary to perform interpolation of the intensity of the pixels. Depending on the method of grid mapping chosen the appropriate method of determination of pixel intensities is performed. For forward mapping of the grid the intensity of each pixel from image B, on the basis of the grid transform, is mapped on to the coordinates of image A (Fig. 2). In this case, however,

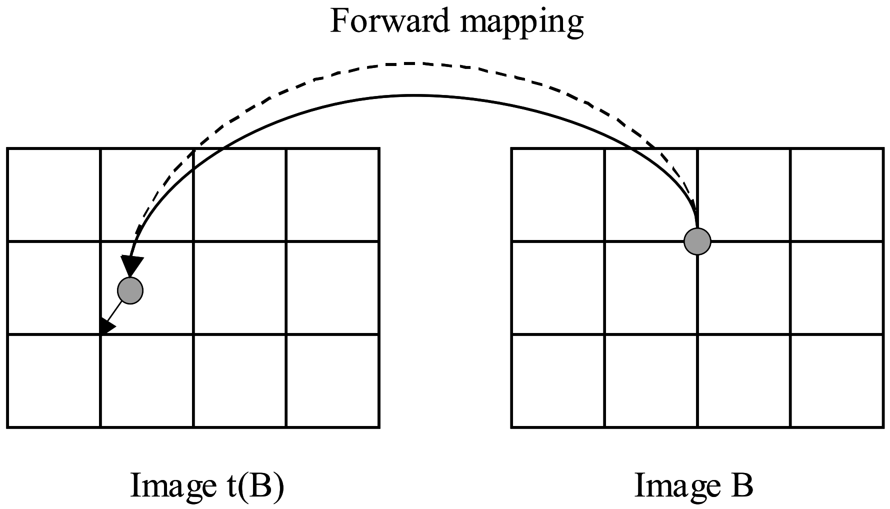
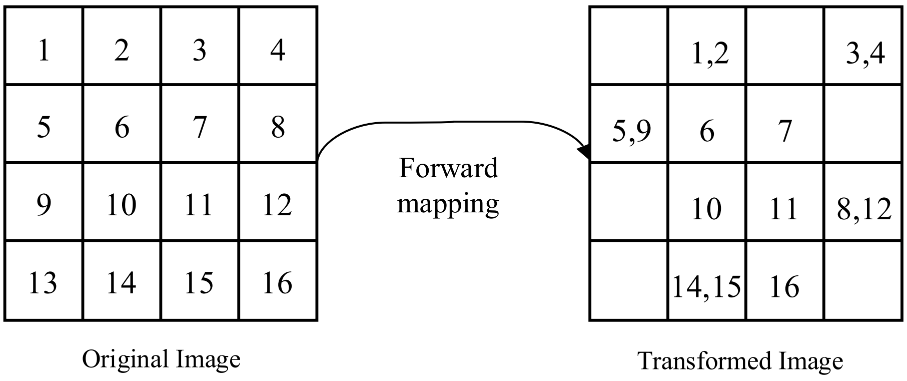


Fig. 2

Forward mapping of pixels. The dashed arrow depicts transform of coordinates and the solid arrow depicts alignment of the brightness values of pixels

after mapping the integer coordinates of pixels from image B usually become real numbers which indicate points lying out of real pixels (non-pixel points). Because of this it might happen that some of the pixels in image t(B) will not be determined and, in effect, black 'holes' will appear in the image t(B) (Fig. 3). The inverse transformation of the grid is followed by determination of pixels' intensities in another way – the brightnesses



1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

	1,2		3,4
5,9	6	7	
	10	11	8,12
	14,15	16	

Fig 3

Illustration of the appearance of 'holes' in forward grid mapping of pixels from image t(B) are determined as values of points from image B that have been indicated as corresponding to pixels from image A by grid transform (Fig. 4). This ensures that for every pixel in transformed image t(B) the brightness value will be determined (Fig. 5). For this reason the quality of matching between images A and t(B) should depend to a large extent on the method of mapping chosen. In inverse mapping, even when the transformed coordinates correspond to the non-pixel points, it is possible to determine the intensity at these points by interpolating the intensities of surrounding real pixels. In forward mapping when this happens, however, the most commonly performed operation is to use nearest-neighbor interpolation, i.e. to assign the brightness of a non-pixel point to the nearest real pixel (Fig. 3). That feature of forward mapping can lead to worsening of the results of matching. If there is an opportunity of using interpolation in forward mapping there should, nevertheless, be no difference between images t(B) constructed by forward and inverse grid mapping. Such a method, enabling interpolation of scattered data points to a regular grid, is offered by Matlab software and is based on Delaunay triangulation [22–24].

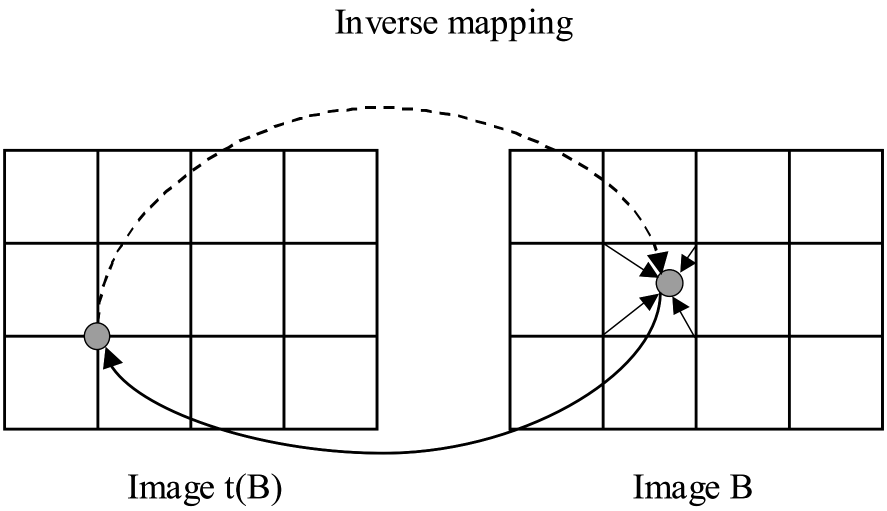
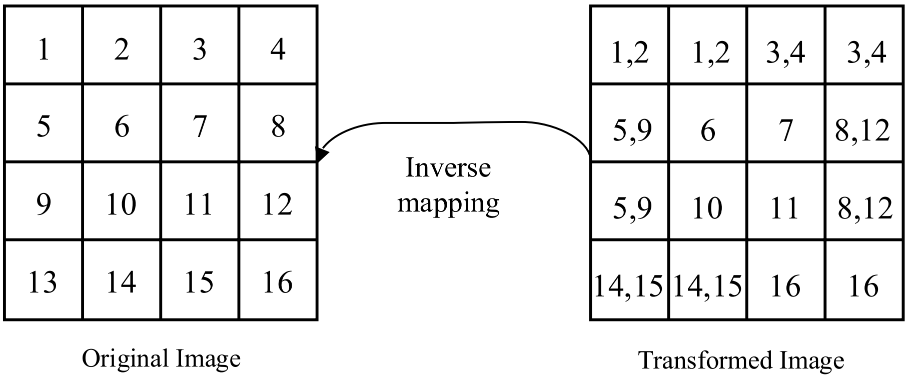


Fig. 4

Inverse mapping of pixels. The dashed arrow depicts transform of coordinates and the solid arrow depicts alignment of the brightness values of pixels



1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

1,2	1,2	3,4	3,4
5,9	6	7	8,12
5,9	10	11	8,12
14,15	14,15	16	16

Fig. 5

Illustration of alignment of new pixels' values in inverse grid mapping

Interpolation of Brightness

In inverse grid mapping, when all the neighbors of a point indicated by transformed coordinates are known it is possible to use a different method of interpolation [25]. The simplest is the nearest-neighbor interpolation, which assigns to the non-pixel point the value of the nearest real pixel:

$$I(x^*,y^*) = I(\mathrm{round}(x^*), \mathrm{round}(y^*))$$

Nearest-neighbor interpolation assigns a pixel value with an error equal at most to half a pixel. Such inadequacy is visually perceptible, especially on boundaries of objects. A more complicated method which also gives better results is linear interpolation, which assumes that the brightness of pixels changes linearly in a close neighborhood. The brightness of a non-pixel point is determined as function of the surrounding four pixels:

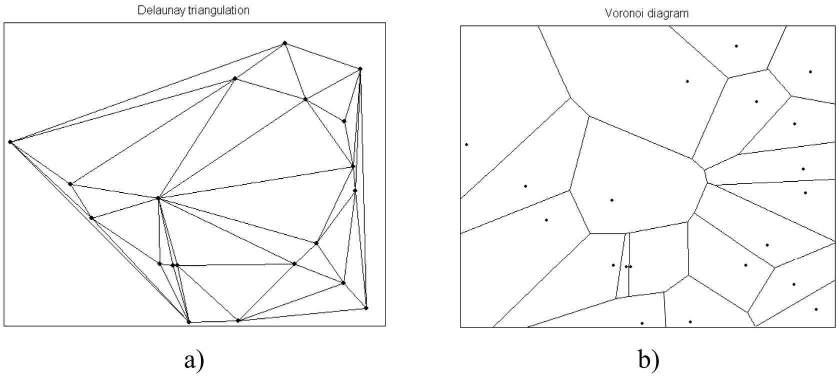
$$I(x^*,y^*) = (1 - a)(1 - b)I(l,k) + \mathrm{a}(1 - b)I(\mathrm{l} + 1,k) + \\ + \mathrm{b}(1 - a)I(l,k + 1) + abI(l + 1,k + 1)$$

where $l = \mathrm{round}(x^*)$, $k = \mathrm{round}(y^*)$, $a = x - l$, and $b = y - k$

Linear interpolation usually causes only a small decrease in resolution and blurring, because of its averaging nature. Linear interpolation can, however, be used only when the values of the surrounding pixels are known. In the forward mapping of a grid the brightness values of the neighboring pixels are unknown and, as already mentioned, one of the most used interpolations is the nearest-neighbor method, which introduces significant errors into the transformed image. It is, nevertheless, possible to use the Delaunay triangulation to tessellate the transformed grid to obtain neighbors of all non-pixel points, and, by exploiting this information, to interpolate values of brightness of real pixels lying on uniform grid [26,27].

Delaunay Triangulation

An important problem in computational geometry is mesh generation and its triangulation. Mesh generation consists of choosing points from a domain (space) whereas triangulation is partition of a mesh into triangles (two dimensional space) or tetrahedra (three dimensional space) that intersect only at shared edges and vertices (Fig. 6a). The edges of triangles connect each of the points to its natural neighbors. In image matching the points of a transformed grid produce a mesh.



a) b)

Fig. 6

The Delaunay triangulation (a) based on the Voronoi diagram (b) of twenty random points

Several methods are used for triangulation of a plane [23]. The simplest consists in constructing polygons and recursively splitting them into triangles by finding their diagonals. By use of such approach it is, however, possible to obtain many different triangulations of a plane of different appearance. More complicated approaches to data tessellation can also be used. These approaches optimize quality criteria, because for a particular number of points it is possible to construct a certain number of triangles so the quality criterion depend only on the shape of the triangles constructed. Typical measures of quality include angles, height, edge lengths, and triangle area. One such optimum triangulations is Delaunay triangulation, which simultaneously optimizes several of the quality measures – minimum and maximum value of angle, and minimum and maximum area of circumcircle.

Delaunay triangulation is based on a method which enables tessellation of a plane into specific polygonal areas called a Voronoi diagram [28]. Polygons of the Voronoi diagram are constructed as non-overlapping areas that contain only one point from the mesh. Every point of this polygon is also closer to a given mesh point than to any other point (Fig. 6b). Delaunay triangulation is a planar representation of the Voronoi diagram, i.e. the edges of triangulation connect the points that share a common edge in the Voronoi diagram (Fig. 6a). More formal description of the Delaunay triangulation might be formulated as follows. If a and b are two points from the transformed grid (mesh) they will be connected by an edge if and only if there is a circle passing through a and b that intersects no other points from the mesh. Also, each circumscribing circle of a triangle does not contain any of the mesh points in its interior (Fig. 7).

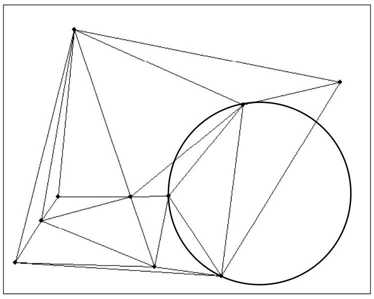


Fig. 7

The circle circumscribed on the Delaunay triangle

Edges of triangles constructed in this manner connect each of the pixels with its natural neighbors. Thus Delaunay triangulation provides ordering information in an initially unordered set of points from a transformed grid. Having obtained the neighbors of all points in the irregular, transformed grid it is now possible to interpolate the values of the brightness of real pixels lying on a uniform grid (Fig. 8). The brightness value of every real pixel is obtained by interpolating the values of neighboring triangle's vertices. By analogy with inverse mapping different methods of interpolation, i.e. nearest neighbor, linear and cubic, can be used in this case also. As mentioned above cubic interpolation yields the best results for image transformation.

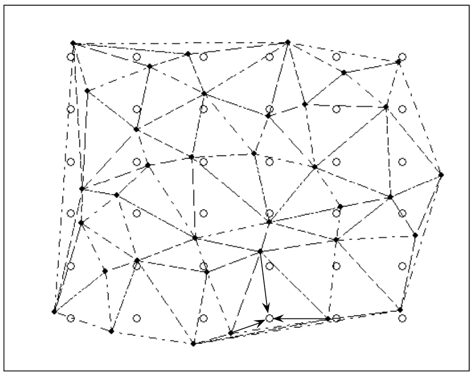


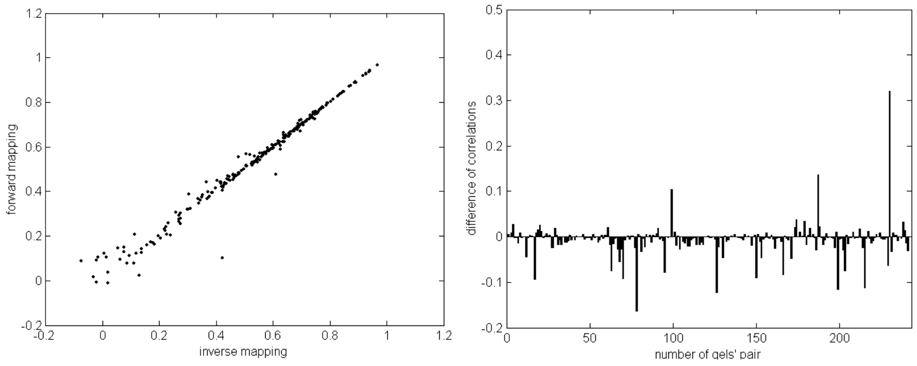
Fig. 8

Illustration of interpolation of uniform grid brightness values on the basis of Delaunay triangulation

RESULTS

The performance of both methods of mapping when applied to matching gel images has been evaluated. Real images of gel plates publicly available in internet databases [29–31] were used as the test images. The performance of mapping was tested on a total of almost 250 pairs of gel images, the whole image-matching procedure being applied to every pair of images. In other words, for every image the most intense spots were detected and described by their intensity, area, and coordinates. This step was followed by use of the iterative fuzzy-matching algorithm, which enabled determination of the correspondence between features and, simultaneously, the transformation parameters. Matching of the images, on the basis of the transformed grids, was performed after feature matching. Grids were transformed by use of the parameters of the global transform determined by means of the fuzzy-matching algorithm. The global transform was fine-tuned by use of local approximations of the transform function obtained by interpolating local distortions in the gel images on the basis of control points. To compare the quality of matching achieved by use of the two different mapping methods the correlation between standard image A and both transformed images t(B) (obtained by forward and inverse mapping) was calculated.

The mean values of correlations between original image A and transformed images t(B) obtained by forward and inverse mapping were equal to 0.5492 and 0.5438, respectively; this small difference is statistically insignificant. Fig. 9a shows the scatter plots of both correlations. It is apparent that the correlation values obtained are almost identical. The significant differences are observable only for images with very low correlations, i.e. images for which the wrong matching results were obtained. The bar plot of differences between the correlations for inverse and forward mapping is presented in Fig. 9b. This figure also shows that



a) b)

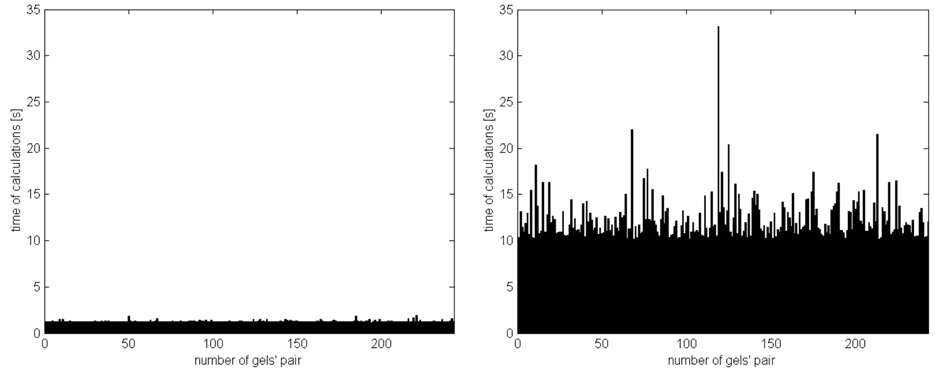
Fig. 9

(a) Scatter plot of correlation between standard image A and transformed images t(B) constructed by inverse (*x* axis) and forward (*y* axis) mapping. (b) Bar plot of differences between these correlations

the differences, with the exception of a few pairs of images, are rather small. On the basis of these results it can be concluded that both mapping methods enable matching of equal quality. The only significant difference between the mapping methods is the computation time. Bar plots of calculation times are presented in Fig. 10. The mean computing times for forward and inverse mapping are 12.3470 and 1.2845 s, respectively. It is apparent the difference is very large (almost a factor of ten) to the disadvantage of forward mapping.

CONCLUSIONS

Inverse mapping of a grid is much preferred to forward mapping, because of the great disadvantage of forward grid mapping followed by nearest-neighbor interpolation, i.e. serious errors in alignment of brightness



a) b)

Fig. 10

Bar plots of calculation times of inverse (a) and forward (b) mapping

values and the possible occurrence of 'holes'. Testing of the performance of forward and inverse grid mapping has, however, clearly shown that both methods yield comparable results on condition that forward mapping is associated with interpolation of the brightness values of mapped pixels. Unfortunately, the need for triangulation of the scattered data points increases computing time. For this reason and the insignificant difference in the quality of matching obtained use of the inverse mapping method is still recommended.

ACKNOWLEDGMENT

K. Kaczmarek thanks Unilever Research (Vlaardingen, Holland) for financial support of his PhD study.

REFERENCES

[1] K. Kaczmarek, B. Walczak, S. de Jong, and B.G.M. Vandeginste, J. Chem. Inf. Comput. Sci., **42**, 1431 (2002)

[2] P.H. O'Farrell, J. Biol. Chem., **250**, 4007 (1975)

[3] R. Westermeier, Electrophoresis in Practice: A Guide to Theory and Practice, VCH, Weinheim, 1993

[4] J.E. Celis and P. Gromov, Curr. Opin. Biotechnol., **10**, 16 (1999)

[5] J.L. Harry, M.R. Wilkins, B.R. Herbert, N.H. Packer, A.A. Gooley, and K.L. Williams, Electrophoresis, **21**, 1071 (2000)

[6] D.S. Young and R.P. Tracy, J. Chromatogr. A, **698**, 163 (1995)

[7] L.R. Bandara and S. Kennedy, Drug Discovery Today, **7**, 411 (2002)

[8] P.F. Lemkin, C. Merril, L. Lipkin, M. Van Keuren, W. Oertel, B. Shapiro, M. Wade, M. Schultz, and E. Smith, Comput. Biomed. Res., **12**, 517–544 (1979)

[9] P.F. Lemkin, L.E. Lipkin, and E.P. Lester, Clin. Chem., **28**, 840 (1982)

[10] J.I. Garrels, J. Biol. Comput., **264**, 5269 (1989)

[11] P.F. Lemkin, Electrophoresis, **18**, 461 (1997)

[12] R.D. Appel, J.R. Vargas, P.M. Palagi, D. Walther, and D.F. Hochstrasser, Electrophoresis, **18**, 735 (1997)

[13] K.P. Pleißner, F. Hoffmann, K. Kriegel, C. Wenk, S. Wegner, A. Sahlström, H. Oswald, H. Alt, and E. Fleck, Electrophoresis, **20**, 755 (1999)

[14] S. Veeser, M.J. Dunn, and G.Z. Yang, Proteomics, **1**, 856 (2001)

[15] Z. Smilansky, Electrophoresis, **22**, 1616 (2001)

[16] F. Hoffmann, K. Kriegel, and C. Wenk, Discrete Appl. Math., **93**, 75 (1999)

[17] K. Takahashi, M. Nakazawa, Y. Watanabe, and A. Konagaya, Genome Informatics, **9**, 173 (1998)

[18] J. Pánek and J. Vohradský, Electrophoresis, **20**, 3484 (1999)

[19] K. Conradsen and J. Pedersen, Biometrics, **48**, 1273 (1992)

[20] K. Kaczmarek, B. Walczak, S. de Jong, and B.G.M. Vandeginste, J. Chem. Inf. Comput. Sci., in press

[21] G. Wolberg, H.M. Sueyllam, M.A. Ismail, and K.M. Ahmed, J. Graphics Tools, **5**, 11 (2000)

[22] Matlab Function Reference, The MathWorks, Inc. 1999

[23] M. Bern and D. Eppstein, Mesh Generation and Optimal Triangulation, Computing in Euclidean Geometry. In: Ding-Zhu Du and Frank Hwang (Eds), Lecture Notes Series on Computing, Vol. 1, World Scientific, Singapore, 1992

[24] A. Goshtasby, Pattern Recogn., **19**, 459 (1986)

[25] M. Sonka, V. Hlavac, and R. Boyle, Image Processing, Analysis, and Machine Vision, Brooks and Cole Publishing, 1999

[26] J.-D. Boissonnat and F. Cazals, Smooth Surface Reconstruction via Natural Neighbour Interpolation of Distance Functions. Proc. 16th Annu. ACM Symp. Comput. Geom. (2000) pp. 223–232; http://citeseer.nj.nec.com/boissonnat00smooth.html

[27] G. Barequet and M. Sharir, Piecewise-Linear Interpolation Between Polygonal Slices, Proc. 10th Annu. ACM Symp. Comput. Geom. (1994) pp. 93–102; http://citeseer.nj.nec.com/188829.html

[28] N. Mayya and V.T. Rajan, J. Math. Imaging Vision, **6**, 355 (1996)

[29] World-2DPage – 2D Page databases and services at http://www.expasy.ch/ch2d/2d-index.html

[30] NCI Flicker Database at http://www-lecb.ncifcrf.gov/2dwgDB/

[31] GelBank at http://gelbank.anl.gov