Optimization Methods and Software
Vol. 22, No. 1, February 2007, 155–168

Taylor & Francis
Taylor & Francis Group

Protein image alignment via tensor product cubic splines

F. A. POTRA* and X. LIU

Department of Mathematics and Statistics, University of Maryland, Baltimore County, USA

(Received 29 April 2005; in final form 26 February 2006)

The class of tensor product cubic splines is considered in an optimization process for two-dimensional polyacrylamide gel electrophoresis (2D-PAGE) image alignment problems arising in proteomics studies. Numerical results are presented along with comparisons with previous results based on piecewise affine and bilinear transformations.

Keywords: Proteomics 2D-PAGE; Image alignment; Tensor product spline; Multi-resolution

1. Introduction

Protein image alignment problem is a very important issue in proteomics studies. Proteomics is the large-scale analysis of complex protein mixtures focusing on the qualitative and quantitative variations of protein expression levels [1,2].

One of the core technologies for obtaining protein mixtures is provided by two-dimensional polyacrylamide gel electrophoresis (2D-PAGE), which wa first introduced by O'Farrell [3] and Klose [4] in 1975. Along the first dimension, proteins are sorted electrophoretically according to their pH gradient. Each stabilizes at a point where its net charge is zero. Along the other dimension, proteins, soaked in sodium dodecyl sulphate, separate according to their molecular weight. Hence, on a 2D gel, two coordinates uniquely characterize each protein spot: its isoelectric point and its molecular weight. The separated proteins are then stained with fluorescent dyes [5] so that they are detectable for imaging. The resulting images are scanned and stored in a database. In recent years, 2D-PAGE analysis has became again one of the most active fields in proteomics studies [2,6–8].

Experimental conditions may cause the gels to be distorted, which results in deformed images with new spot coordinates, shapes, sizes and intensity values. Some possible factors contributing to such distortions include the structure of the polyacrylamide net, the characteristics of transporting solution, the solvent conditions, and the nature of the electric field [8].

*Corresponding author. Email: potra@math.edu

Optimization Methods and Software
ISSN 1055-6788 print/ISSN 1029-4937 online
http://www.tandf.co.uk/journals
DOI: 10.1080/10556780600882116

Unfortunately, these distortions cannot be avoided. Even under strictly controlled laboratory conditions, repeated analysis of the same sample may produce non-identical gels [7,9]. In order to analyse variations in the protein gels obtained from different groups that account for biological variations attributable, for example, to a treatment we must first eliminate distortions to properly align images. The image alignment is recognized as a major bottleneck in proteomics.

Alignment of a family of gel images can be done by means of geometrical transformations applied to the coordinate domain of the images [10,11]. This technique is also known as image warping [9,12,13], or image registration [14–17]. The transformed images should have no (or very little) geometrical variations, so that only statistical and biological variation are observed. Most image warping methods rely on some preassigned control points or landmarks, which are a relatively small group of spots present in all the gels being compared. The landmarks should be selected so that they are evenly distributed over the gel image. The number of the landmarks should be large enough, so that they can carry enough information. However, too many landmarks will increase computational complexity and may result in over-fitting. The shape and size of the protein spots chosen as landmarks may vary significantly, but for the purpose of image alignment, only the x–y coordinates of their centres are considered. In the remainder of the current paper, we refer to landmarks simply as the x–y coordinates of their centres.

In our recent paper [18], we have proposed a combined forward–inverse transformation approach to align families of gel images without any pre-selected reference gel. The information contained in the landmarks of the whole family is used to create an ideal gel. Then we determine proper transformations that optimally align all the gels of the family to this ideal gel. Using the approach in [18], we have managed to align very large families of gel images. The transformations used in [18] are piecewise affine and bilinear transformations based on a sequence of hierarchical grids. A similar hierarchical approach has been used in [9,17] for pairwise alignment. In the present paper, we adopt the approach in [18], with the transformations replaced by the family of tensor product cubic splines. We will impose proper constraints so that the transformation is globally C^2, in the sense that the transformation itself, the first derivatives and the second derivatives are continuous.

Tensor product splines have been used for many years [19–22] for interpolation and surface fitting. However, they have never been used in connection to optimization for image processing. In contrast with traditional interpolation, where the values at grid points (vertices) are to match some given values, we consider the values at grid points as part of the unknowns in a large-scale optimization problem.

In section 2 we present the construction of piecewise polynomial transformations based on segmentations of the gel images. We then discuss the tensor product cubic spline in section 3. The combined forward–inverse transformation approach is presented in section 4. In section 5 we evaluate the performance of the method on two protein image data sets which were also used in [18]. Some final conclusions are presented in section 6.

2. Segmentation of the gel images

Given a collection of M gel images $\mathcal{I}^{(1)}, \mathcal{I}^{(2)}, \ldots, \mathcal{I}^{(M)}$, with N preassigned landmarks on each image, we denote by $L_{il} = (x_{il}, y_{il})^T$ the lth landmark on $\mathcal{I}^{(i)}$, $l = 1, \ldots, N$. All the gel images in our applications are rectangles. We assume in general that $\mathcal{I}^{(i)}$ is included in the rectangle $\Omega^{(i)} = [\check{\tau}_i^x, \hat{\tau}_i^x] \times [\check{\tau}_i^y, \hat{\tau}_i^y]$. In contrast with the traditional pairwise alignment approach, we want to construct a set of ideal landmarks, L_l, $l = 1, \ldots, N$ an ideal gel image $\mathcal{I}$, along with a set of geometric transformations $T^{(i)}$: $\Omega^{(i)} \to \mathbb{R}^2$, as the solutions obtained

from the following optimization problem

$$\min_{T^{(i)},L_l} \varphi(T^{(i)}, L_l) \tag{1}$$

$$\text{subject to } \|T^{(i)}(L_{il}) - L_l\|_\infty \leqslant \varepsilon, \quad i = 1, \ldots, M, \;\; l = 1, \ldots, N,$$

$$\left\| L_l - \frac{1}{M} \sum_{i=1}^{M} L_{il} \right\|_\infty \leqslant \delta, \quad l = 1, \ldots, N,$$

$$\text{other smoothness constraints}$$

where the objective function φ is constructed in such a way as to maximize the smoothness of the transformations, and ε and δ are some constant parameters. The first constraint in equation (1) is used to ensure that the transformed landmarks $T^{(i)}(L_{il})$ are close enough (within ε pixels) to the ideal landmark L_l, and the second constraint is used to set the ideal landmark L_l close (within δ pixels) to the centre of the corresponding group of landmarks L_{il}, $i = 1, \ldots, M$.

As suggested in [18,23], in order to avoid the drawbacks and limitations of global transformations, we consider piecewise low-order polynomial transformations based on segmentations of the gel images as follows. Rectangle $\Omega^{(i)}$ is divided into $p_i \times q_i$ equal sub-rectangles

$$\Omega_{jk}^{(i)} = \left\{ v = \begin{pmatrix} x \\ y \end{pmatrix} \in R^2, s_j^{(i)} \leqslant x \leqslant s_{j+1}^{(i)}, t_k^{(i)} \leqslant y \leqslant t_{k+1}^{(i)} \right\} \tag{2}$$

$$j = 1, \ldots, p_i, \quad k = 1, \ldots, q_i$$

with vertices $v_{jk}^i = (s_j^{(i)}, t_k^{(i)})^T$, where

$$s_j^{(i)} = \breve{\tau}_i^x + (j-1)\Delta x^{(i)}, \quad j = 1, \ldots, p_i + 1 \tag{3}$$

$$t_k^{(i)} = \breve{\tau}_i^y + (k-1)\Delta y^{(i)}, \quad k = 1, \ldots, q_i + 1 \tag{4}$$

and

$$\Delta x^{(i)} = \frac{\hat{\tau}_i^x - \breve{\tau}_i^x}{p_i}, \quad \Delta y^{(i)} = \frac{\hat{\tau}_i^y - \breve{\tau}_i^y}{q_i} \tag{5}$$

On each sub-rectangle $\Omega_{jk}^{(i)}$, we define a local transformation

$$T_{jk}^{(i)}(x, y) = \begin{pmatrix} f_{jk}^{(i)}(x, y) \\ g_{jk}^{(i)}(x, y) \end{pmatrix} : \Omega_{jk}^{(i)} \longrightarrow R^2$$

We then define the global transformation by combining all the local transformations

$$T^{(i)}(x, y) = \begin{pmatrix} f^{(i)}(x, y) \\ g^{(i)}(x, y) \end{pmatrix} = \begin{pmatrix} \sum_{j=1}^{p_i} \sum_{k=1}^{q_i} \omega_{jk}^{(i)}(x, y) f_{jk}^{(i)}(x, y) \\ \sum_{j=1}^{p_i} \sum_{k=1}^{q_i} \omega_{jk}^{(i)}(x, y) g_{jk}^{(i)}(x, y) \end{pmatrix} \tag{6}$$

where

$$\omega_{jk}^{(i)}(x, y) = \begin{cases} 1 & \textit{if } (x, y) \in \Omega_{jk}^{(i)} \\ 0 & \textit{otherwise} \end{cases} \tag{7}$$

In [18,23], we have managed to use piecewise affine transformations and piecewise bilinear transformations to align families of 2D-gels images. The transformations were constructed in

such a way that they are globally continuous. A question arose naturally: how to obtain classes of globally C^1 or even C^2 transformations? In the following sections, tensor product cubic splines will be constructed in such a way that the global transformation is C^2 with optimized curvature.

3. Tensor product cubic splines

A general form of tensor product splines is given by

$$\sum_{i=0}^{m}\sum_{j=0}^{n} V_{ij} B_i(x) C_j(y) \tag{8}$$

where V_{ij} are parameters and $B_i(x)$ and $C_j(y)$ are base functions. A typical representative of tensor product splines is the class of tensor product B-splines where the base function is a B-spline function [20–22]. The simplest form of tensor product splines is with linear base functions [20]. This is in fact the piecewise bilinear transformation used in [9,17,18,23]. By choosing cubic polynomial base functions in equation (8), we obtain tensor product cubic splines. Although tensor product cubic splines have been used before [19,22] for interpolation and surface fitting, they have not been used in an optimization process for image alignment. In the present section, we aim to construct globally C^2 tensor product cubic splines by choosing proper cubic base functions.

A tensor product cubic spline is decided by a set of tensor product cubic transformations through equation (6). On each sub-rectangle, a tensor product cubic transformation can be defined as

$$T_{jk}^{(i)}(x, y) = \begin{pmatrix} f_{jk}^{(i)}(x, y) \\ g_{jk}^{(i)}(x, y) \end{pmatrix} = \begin{pmatrix} \sum_{\lambda=0}^{3}\sum_{\gamma=0}^{3} a_{\lambda\gamma}^{ijk} x^{\lambda} y^{\gamma} \\ \sum_{\lambda=0}^{3}\sum_{\gamma=0}^{3} b_{\lambda\gamma}^{ijk} x^{\lambda} y^{\gamma} \end{pmatrix} \tag{9}$$

We note there are 16 parameters in the transformation on each dimension. As suggested in [18], using these coefficients to define the transformation is not efficient. We thus reformulate the transformation using a unique set of base functions h_{00}, h_{01}, h_{10} and h_{11} that satisfies the following conditions

$$\begin{aligned}
&h_{00}(0) = 1, \quad h_{00}(1) = 0, \quad h'_{00}(0) = 0, \quad h'_{00}(1) = 0 \\
&h_{10}(0) = 0, \quad h_{10}(1) = 1, \quad h'_{10}(0) = 0, \quad h'_{10}(1) = 0 \\
&h_{01}(0) = 0, \quad h_{01}(1) = 0, \quad h'_{01}(0) = 1, \quad h'_{01}(1) = 0 \\
&h_{11}(0) = 0, \quad h_{11}(1) = 0, \quad h'_{11}(0) = 0, \quad h'_{11}(1) = 1
\end{aligned}$$

That is, for $\mu, \nu, \iota, \sigma \in \{0, 1\}$, we have

$$h_{\mu\nu}^{(\iota)}(\sigma) = \delta_{\mu\sigma}\delta_{\nu\iota} \tag{10}$$

where

$$\delta_{ij} = \begin{cases} 1 & \textit{if} \quad i = j \\ 0 & \textit{otherwise} \end{cases}$$

We could use standard polynomial interpolation techniques such as divided difference, to obtain the base functions:

$$
\begin{aligned}
h_{00}(x) &= 2x^3 - 3x^2 + 1,\\
h_{10}(x) &= -2x^3 + 3x^2,\\
h_{01}(x) &= x^3 - 2x^2 + x,\\
h_{11}(x) &= x^3 - x^2.
\end{aligned}
$$

This is in fact the special case of the cubic Hermite basis [20].

We are now ready to re-formulate the tensor product cubic transformation using the above base functions. We omit the index i for simplicity, and obtain

$$
\begin{aligned}
f(x, y) &= \sum_{j=1}^{p}\sum_{k=1}^{q}\omega_{jk}(x, y)\sum_{\iota,\sigma,\mu\nu\in\{0,1\}} f^{\mu\nu}_{j+\iota,k+\sigma}h_{\iota\mu}\left(\frac{x - x_j}{\Delta x}\right)h_{\sigma\nu}\left(\frac{y - y_k}{\Delta y}\right)\\
g(x, y) &= \sum_{j=1}^{p}\sum_{k=1}^{q}\omega_{jk}(x, y)\sum_{\iota,\sigma,\mu\nu\in\{0,1\}} g^{\mu\nu}_{j+\iota,k+\sigma}h_{\iota\mu}\left(\frac{x - x_j}{\Delta x}\right)h_{\sigma\nu}\left(\frac{y - y_k}{\Delta y}\right)
\end{aligned}
$$

where $f^{\mu\nu}_{j+\iota,k+\sigma}$, $\iota, \sigma, \mu, \nu \in \{0, 1\}$ are 16 free parameters defining the corresponding tensor product cubic transformation, and likewise for $g^{\mu\nu}_{j+\iota,k+\sigma}$. Moreover, we note that because the base functions satisfy condition (10), the following relations hold for any value of the parameters

$$
\begin{aligned}
\frac{\partial^{\mu+\nu} f}{\partial x^{\mu}\partial y^{\nu}}(x_{j+\iota}, y_{k+\sigma}) &= \frac{1}{(\Delta x)^{\mu}(\Delta y)^{\nu}} f^{\mu\nu}_{j+\iota,k+\sigma}\\
\frac{\partial^{\mu+\nu} g}{\partial x^{\mu}\partial y^{\nu}}(x_{j+\iota}, y_{k+\sigma}) &= \frac{1}{(\Delta x)^{\mu}(\Delta y)^{\nu}} g^{\mu\nu}_{j+\iota,k+\sigma},
\end{aligned}
\tag{11}
$$

where $\iota, \sigma, \mu, \nu \in \{0, 1\}$. We also note that for any values of (x, y), $f(x, y)$, $g(x, y)$ and their arbitrary order partial derivatives are linear functions with respect to the coefficients $f^{\mu\nu}_{j+\iota,k+\sigma}$ and $g^{\mu\nu}_{j+\iota,k+\sigma}$, respectively.

Let us now analyse the conditions we need to impose on the parameters to ensure global C^2 property for the tensor product cubic spline. Since each tensor product cubic transformation is C^2 on the corresponding local rectangular area, it is enough to check the C^2 property on the common edges and vertices of adjacent rectangles. For simplicity we consider a typical interior vertex $v = (\xi, \tau)$. Let Ω_1, Ω_2, Ω_3, and Ω_4 be the four rectangles adjacent to v. Let us denote by f_1, f_2, f_3, f_4 the horizontal component of the local transformations on defined on the rectangles Ω_1, Ω_2, Ω_3, and Ω_4, respectively. That is, $f_i = f|_{\Omega_i}$, $i = 1, 2, 3, 4$. We also denote by Δ_{ij} the common edge of rectangles Ω_i and Ω_j, as shown in figure 1.

Let us consider as an example the horizontal common edge Δ_{14} in figure 1. We dnote by $v' = (\zeta, \tau)$ the other end point on Δ_{14}.

We first check the C^0 property on Δ_{14}. Since $y = \tau$ is fixed on Δ_{14}, $f(x, y)$ is a polynomial of order 3 with respect to x. We note that a polynomial of order 3 is completely determined by two functions values and two derivative values. At the two end points of Δ_{14}, the following

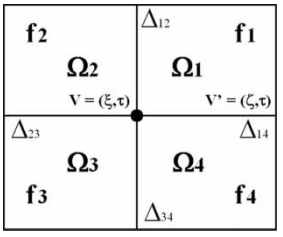


Figure 1. A typical interior vertex.

equalities always hold

$$f_1(\xi,\tau) = f_4(\xi,\tau), \qquad \frac{\partial}{\partial x} f_1(\xi,\tau) = \frac{\partial}{\partial x} f_4(\xi,\tau)$$
$$f_1(\zeta,\tau) = f_4(\zeta,\tau), \qquad \frac{\partial}{\partial x} f_1(\zeta,\tau) = \frac{\partial}{\partial x} f_4(\zeta,\tau)$$

This implies that

$$f_1(x,\tau) \equiv f_4(x,\tau) \qquad \forall x \in [\xi,\tau] \tag{12}$$

Therefore, continuity holds on Δ_{14}, for arbitrary choices of the parameters $f^{\mu\nu}_{j+\iota,k+\sigma}$.
Let us now consider the C^1 property. It follows immediately from equation (12) that

$$\frac{\partial}{\partial x} f_1(x,\tau) \equiv \frac{\partial}{\partial x} f_4(x,\tau), \qquad \forall x \in [\xi,\tau].$$

Let us now consider $\partial f/\partial y$. We note that on Δ_{14}, $\partial/\partial y f(x,y)$ is a polynomial of order 3 with respect to x. At the two end points of Δ_{14}, we have

$$\frac{\partial}{\partial y} f_1(\xi,\tau) = \frac{\partial}{\partial y} f_4(\xi,\tau), \qquad \frac{\partial^2}{\partial x \partial y} f_1(\xi,\tau) = \frac{\partial^2}{\partial x \partial y} f_4(\xi,\tau)$$
$$\frac{\partial}{\partial y} f_1(\zeta,\tau) = \frac{\partial}{\partial y} f_4(\zeta,\tau), \qquad \frac{\partial^2}{\partial x \partial y} f_1(\zeta,\tau) = \frac{\partial^2}{\partial x \partial y} f_4(\zeta,\tau)$$

This implies

$$\frac{\partial}{\partial y} f_1(x,\tau) \equiv \frac{\partial}{\partial y} f_4(x,\tau), \qquad \forall x \in [\xi,\tau] \tag{13}$$

Therefore, the C^1 property holds on Δ_{14}, for arbitrary choices of the parameters $f^{\mu\nu}_{j+\iota,k+\sigma}$.

At last we want to obtain C^2 for the tensor product cubic spline. Again it follows from equation (12) that

$$\frac{\partial^2}{\partial x^2} f_1(x,\tau) \equiv \frac{\partial^2}{\partial x^2} f_4(x,\tau), \qquad \forall x \in [\xi,\tau].$$

And it follows from equation (13) that

$$\frac{\partial^2}{\partial x \partial y} f_1(x,\tau) \equiv \frac{\partial^2}{\partial x \partial y} f_4(x,\tau), \qquad \forall x \in [\xi,\tau]$$

Moreover, we note that since f_1 and f_4 are completely smooth so that we can change the order of differentiation within their domains. Therefore, the equation

$$\frac{\partial^2}{\partial y \partial x} f_1(x,\tau) \equiv \frac{\partial^2}{\partial y \partial x} f_4(x,\tau), \qquad \forall x \in [\xi,\tau]$$

holds. In fact, we can apply this argument throughout the present paper. We claim that we can change the order of differentiation in any equation involving partial derivatives and the resulting equation will hold as well.

Therefore it remains to analyze the continuity of $(\partial^2/\partial y^2) f(x, y)$. Once $y = \tau$ is fixed, $(\partial^2/\partial y^2) f(x, y)$ is a polynomial of order 3 with respect to x. We thus impose the following constraints

$$\frac{\partial^2}{\partial y^2} f_1(\xi, \tau) = \frac{\partial^2}{\partial y^2} f_4(\xi, \tau), \qquad \frac{\partial^2}{\partial y^2} f_1(\zeta, \tau) = \frac{\partial^2}{\partial y^2} f_4(\zeta, \tau) \tag{14}$$

and

$$\frac{\partial^3}{\partial y^2 \partial x} f_1(\xi, \tau) = \frac{\partial^3}{\partial y^2 \partial x} f_4(\xi, \tau), \qquad \frac{\partial^3}{\partial y^2 \partial x} f_1(\zeta, \tau) = \frac{\partial^3}{\partial y^2 \partial x} f_4(\zeta, \tau) \tag{15}$$

The C^2 property is therefore enforced on Δ_{14}.

A similar analysis can be done to retrieve the conditions needed to ensure the C^2 property on a vertical common edge between two adjacent horizontal rectangles.

Let us summarize the necessary constraints to enforce the C^2 property globally. According to equation (14), constraints

$$\frac{\partial^2}{\partial y^2} f_1(v) = \frac{\partial^2}{\partial y^2} f_4(v), \qquad \frac{\partial^2}{\partial y^2} f_2(v) = \frac{\partial^2}{\partial y^2} f_3(v) \tag{16}$$

are to be enforced on Δ_{14} and Δ_{23}, respectively. We note that the corresponding equalities hold automatically on Δ_{12} and Δ_{34}, i.e.

$$\frac{\partial^2}{\partial y^2} f_1(v) = \frac{\partial^2}{\partial y^2} f_2(v), \qquad \frac{\partial^2}{\partial y^2} f_3(v) = \frac{\partial^2}{\partial y^2} f_4(v)$$

hold for any arbitrary choices of parameters. Therefore, instead of equation (16), it is enough to impose only

$$\frac{\partial^2}{\partial y^2} f_1(v) = \frac{\partial^2}{\partial y^2} f_4(v). \tag{17}$$

It follows from equation (15) that we have to impose also the following constraints to ensure the continuity of $(\partial^2/\partial y^2) f(v)$

$$\frac{\partial^3}{\partial y^2 \partial x} f_1(v) = \frac{\partial^3}{\partial y^2 \partial x} f_4(v), \qquad \frac{\partial^3}{\partial y^2 \partial x} f_2(v) = \frac{\partial^3}{\partial y^2 \partial x} f_3(v) \tag{18}$$

However, the corresponding equalities hold automatically on Δ_{12} and Δ_{34}. This is because

$$\frac{\partial^2}{\partial x \partial y} f_1(z) \equiv \frac{\partial^2}{\partial x \partial y} f_2(z), \qquad \forall z \in \Delta_{12}$$

and

$$\frac{\partial^2}{\partial x \partial y} f_3(z) \equiv \frac{\partial^2}{\partial x \partial y} f_4(z), \qquad \forall z \in \Delta_{34}$$

We thus have

$$\frac{\partial^3}{\partial y^2 \partial x} f_1(v) = \frac{\partial^3}{\partial y^2 \partial x} f_2(v), \qquad \frac{\partial^3}{\partial y^2 \partial x} f_3(v) = \frac{\partial^3}{\partial y^2 \partial x} f_4(v)$$

Therefore, instead of equation (18), it is enough to impose only

$$\frac{\partial^3}{\partial y^2 \partial x} f_1(v) = \frac{\partial^3}{\partial y^2 \partial x} f_4(v) \tag{19}$$

A similar analysis can be done for $(\partial^2/\partial x^2) f(v)$. We need to impose the following constraints to ensure the continuity of $(\partial^2/\partial x^2) f(v)$ on all four edges

$$\frac{\partial^2}{\partial x^2} f_1(v) = \frac{\partial^2}{\partial x^2} f_2(v) \tag{20}$$

$$\frac{\partial^3}{\partial x^2 \partial y} f_1(v) = \frac{\partial^3}{\partial x^2 \partial y} f_2(v) \tag{21}$$

From the above argument, it follows that

$$f, \frac{\partial f}{\partial x}, \qquad \frac{\partial f}{\partial y}, \qquad \frac{\partial^2 f}{\partial x^2}, \qquad \frac{\partial^2 f}{\partial y^2}, \qquad \frac{\partial^2 f}{\partial x \partial y} \tag{22}$$

are well defined and continuous on the four edges if we impose equality constraints (17), (19), (20), (21). Moreover they are also continuous at each interior vertex. Therefore, we claim that we only need to impose constraints (17), (19), (20), (21), at any interior vertex v.

We next consider the constraints needed to be imposed on a vertex on the boundary. By using a similar analysis, it turns out that we need to impose the constraints (17), (19) for each vertex on a vertical boundary, where in this case, f_1 and f_4 in (17) and (19) stand for the transformations from the two (vertically) adjacent rectangles, and the constraints (20), (21) for each vertex on a horizontal boundary, where in this case, f_1 and f_2 in equations (20) and (21) stand for the transformations from the two (horizontally) adjacent rectangles.

The exact analysis can be performed on $g(x, y)$. In summary, the tensor product cubic spline is C^1 automatically owing to our selection of the base functions. In order to achieve C^2, we need to impose constraints the (17), (19), (20) and (21), if applicable (that is, depending on the location of the vertex). Altogether we need to impose

$$2(4(p_i - 1)(q_i - 1) + 2(2(p_i - 1) + 2(q_i - 1))) = 8(p_i q_i - 1)$$

equality constraints in order to have a C^2 tensor product cubic spline $T^{(i)}$.

If we consider $T^{(i)}$ in equation (1) defined by tensor product cubic splines we obtain

$$\begin{aligned} \min_{L_l, f_{j,k}^{(i)\mu\nu}, g_{j,k}^{(i)\mu\nu}} \sum_{i=1}^{M} \sum_{j=1}^{p_i+1} \sum_{k=1}^{q_i+1} \Bigg(& \left(\frac{\partial^2}{\partial x^2} f(v_{jk}^{(i)}) \right)^2 + \left(\frac{\partial^2}{\partial y^2} f(v_{jk}^{(i)}) \right)^2 + \left(\frac{\partial^2}{\partial x \partial y} f(v_{jk}^{(i)}) \right)^2 \\ & + \left(\frac{\partial^2}{\partial x^2} g(v_{jk}^{(i)}) \right)^2 + \left(\frac{\partial^2}{\partial y^2} g(v_{jk}^{(i)}) \right)^2 + \left(\frac{\partial^2}{\partial x \partial y} g(v_{jk}^{(i)}) \right)^2 \\ & + w_f \left(f(v_{jk}^{(i)}) - v_{jk}^{(i)} \right)^2 + w_g \left(g(v_{jk}^{(i)}) - v_{jk}^{(i)} \right)^2 \Bigg) \end{aligned} \tag{23}$$

s.t. constraints (17), (19), (20), (21)

$$\| T^{(i)}_{\alpha(i,l),\beta(i,l)}(L_{il}) - L_l \|_\infty \leqslant \varepsilon, \quad i = 1, \ldots, M, \;\; l = 1, \ldots, N$$

$$\left\| L_l - \frac{1}{M} \sum_{i=1}^{M} L_{il} \right\|_\infty \leqslant \delta, \quad l = 1, \ldots, N$$

where $(\alpha(i,l), \beta(i,l))$ is the index of the rectangle that L_{il} lies in

$$L_{il} \in \Omega^{(i)}_{\alpha(i,l),\beta(i,l)}$$

and w_f and w_g are non-negative constant parameters. The weighted terms in the objective function measure the difference between $T^{(i)}$ and an identical transformation. The rest of the terms in the objective function are squared second-order derivative information of the transformations at the vertices, which is a good measure of the curvature of the transformation. We note that these second-order derivatives, along with the constraints, can be written as linear function of the unknowns $f^{(i)\mu\nu}_{j+\iota,k+\sigma}$, $g^{(i)\mu\nu}_{j+\iota,k+\sigma}$ and L_l. Therefore, equation (23) is a valid quadratic program (QP). Let us show next that it is in fact a convex QP.

It follows directly from the formulation that $((\partial^2/\partial x\partial y)f(v_{jk}))^2$, $((\partial^2/\partial x\partial y)g(v_{jk}))^2$, $(f(v^{(i)}_{jk}) - v^{(i)}_{jk})^2$ and $(f(v^{(i)}_{jk}) - v^{(i)}_{jk})^2$ are convex. Since the sum of convex functions is still convex, it is enough to show that $((\partial^2/\partial x^2)f(v_{jk}))^2$, $((\partial^2/\partial x^2)g(v_{jk}))^2$ $((\partial^2/\partial y^2)f(v_{jk}))^2$ and $((\partial^2/\partial y^2)g(v_{jk}))^2$ are convex. It can be derived directly from the definition of the cubic Hermite basis that

$$h''_{00}(x) = 12x - 6, \qquad h''_{10}(x) = -12x + 6, \qquad h''_{01}(x) = 6x - 4, \qquad h''_{11}(x) = 6x - 2.$$

We thus have $(\partial^2/\partial x^2)f(v_{jk}) = -6f^{00}_{jk} + 6f^{00}_{j+1,k} - 4f^{10}_{jk} - 2f^{10}_{j+1,k}$. It can be easily shown that the Hessian of $((\partial^2/\partial x^2)f(v_{jk}))^2$ is positive semi-definite, which in turn implies convexity. The same arguments also apply to the rest of the second-order derivatives. Thus we claim that equation (23) is a convex QP, which can be solved efficiently by interior-point methods.

Let us now analyse the feasibility of the above QP. It our applications we take $p_i = q_i = p$, so that we have $8M(p+1)^2$ unknowns. There are $8M(p^2-1)$ linear equality constraints to ensure the C^2 property, and $2MN + 2N$ equality constraints when $\varepsilon = \delta = 0$. A solution in this case is clearly a feasible point for the QP with $\varepsilon \geqslant 0, \delta \geqslant 0$. In order to ensure feasibility, it suffices to have $16Mp + 16M \geqslant 2MN + 2N$. In all our applications $16M \geqslant 2N$, so that QP (23) is always feasible if $p \geqslant N/8$. For instance, for a set of gel images with 20 landmarks on each, a 3×3 grid is enough to ensure feasibility.

4. Algorithm

We now describe the algorithm. We use the combined forward–inverse transformation approach from [18]. A forward tensor product cubic spline is constructed based on a relatively sparse segmentation of the image. As a result of the forward phase, a set of ideal landmarks L_l, $l = 1, \ldots, N$, along with the approximate size of the ideal gel $\Omega = [\check{\tau}^x, \hat{\tau}^x] \times [\check{\tau}^y, \hat{\tau}^y]$, are obtained from QP (23). In the second phase of our algorithm, we construct an inverse tensor product cubic spline for every pair of $\mathcal{I}$ and $\mathcal{I}^{(i)}$. That is, for each source image $\mathcal{I}^{(i)}$, we construct a tensor product cubic spline $T^{(i)}\colon \mathcal{I} \to \mathcal{I}^{(i)}$ as the solution of an optimization problem similar to equation (23). For simplicity, we omit the index i and describe this process below.

The rectangle Ω is divided into $p \times q$ equal sub-rectangles

$$\Omega_{jk} = \left\{ z = \begin{pmatrix} x \\ y \end{pmatrix} \in R^2, s_j \leqslant x \leqslant s_{j+1}, t_k \leqslant y \leqslant t_{k+1} \right\}$$
$$j = 1, \ldots, p, \quad k = 1, \ldots, q$$

with vertices $v_{jk} = (s_j, t_k)^T$, where

$$s_j = \check{\tau}^x + (j-1)\Delta x, \quad j = 1, \ldots, p+1 \tag{24}$$

$$t_k = \check{\tau}^y + (k-1)\Delta y, \quad k = 1, \ldots, q+1 \tag{25}$$

and

$$\Delta x = \frac{\hat{\tau}^x - \check{\tau}^x}{p}, \quad \Delta y = \frac{\hat{\tau}^y - \check{\tau}^y}{q} \tag{26}$$

We construct a tensor product cubic spline $T: \mathcal{I} \to \mathcal{I}^{(i)}$ based on this segmentation. Following the ideas from section 3, we impose the following inequalities

$$\|T_{m(l),n(l)}(L_l) - L_{il}\|_\infty \leqslant \epsilon, \quad l = 1, \ldots, N \tag{27}$$

where $(m(l), n(l))$ are the indices of the rectangle containing L_l

$$L_l \in \Omega_{m(l),n(l)}$$

The objective function and the constraints are kept as before and we obtain the following QP for pairwise alignment between $\mathcal{I}$ and $\mathcal{I}^{(i)}$.

$$\begin{aligned} \min_{f_{j,k}^{\mu\nu}, g_{j,k}^{\mu\nu}} \sum_{j=1}^{p+1} \sum_{k=1}^{q+1} \Bigg(& \left(\frac{\partial^2}{\partial x^2} f(v_{jk}) \right)^2 + \left(\frac{\partial^2}{\partial y^2} f(v_{jk}) \right)^2 + \left(\frac{\partial^2}{\partial x \partial y} f(v_{jk}) \right)^2 \\ & + \left(\frac{\partial^2}{\partial x^2} g(v_{jk}) \right)^2 + \left(\frac{\partial^2}{\partial y^2} g(v_{jk}) \right)^2 + \left(\frac{\partial^2}{\partial x \partial y} g(v_{jk}) \right)^2 \\ & + w_f (f(v_{jk}) - v_{jk})^2 + w_g (g(v_{jk}) - v_{jk})^2 \Bigg) \end{aligned} \tag{28}$$

s.t. constraints (17), (19), (20), (21),

$$\|T_{m(l),n(l)}(L_l) - L_{il}\|_\infty \leqslant \varepsilon, \quad l = 1, \ldots, N,$$

We adopt the multi-resolution approach [18,23] in the inverse transformation phase: we construct a sequence of increasingly finer partitions $\mathcal{P}_1^{(i)}, \mathcal{P}_2^{(i)}, \ldots$ on the ideal gel $\mathcal{I}$. Partition $\mathcal{P}_\kappa^{(i)}$ consists in dividing $\mathcal{I}$ into $p_{i,\kappa} \times q_{i,\kappa}$ rectangles where $p_{i,1}, p_{i,2}, \ldots, p_{i,\kappa}, \ldots$ and $q_{i,1}, q_{i,2}, \ldots, q_{i,\kappa}, \ldots$ are two increasing sequences of positive integers. At iteration κ, we construct an inverse tensor product cubic spline based on partition $\mathcal{P}_\kappa^{(i)}$. If the stopping criteria, such as the accuracy or smoothness requirements, are not met, we proceed to the next iteration. For instance, in our application we take $p_{i,\kappa+1} = 2p_{i,\kappa}$, $q_{i,\kappa+1} = 2q_{i,\kappa}$, so that if the stopping criteria are not met, we divide each rectangle into four equal sub-rectangles and repeat the construction.

Using this combined forward–inverse transformations with the multi-resolution approach in the inverse transformation phase has certain advantages in assigning the intensity values for the transformed images and decreasing the computational cost. For a complete analysis of this approach, we refer to [18]. Our algorithm can be described as follows.

Algorithm.

Given source gels $\mathcal{I}^{(i)}$ with landmarks L_{il},
(*forward transformations*)
compute ideal landmarks L_l and

estimate the size Ω of ideal reference gel $\mathcal{I}$ from equation (23);
(*inverse transformations*)
for $i = 1, \ldots, M$,
start with partition $\mathcal{P}_0 : p \times q$,
repeat
Compute s_j and t_k from equations (24) and (25);
Obtain $T^{(i)}$ from the solution of equation (28);
if some stopping criterion is satisfied or
the maximum number of iterations is exceeded
Return;
else
Refine the grid by setting $p = 2p, q = 2q$;
compute $\hat{\mathcal{I}}^{(i)} = T^{(i)}(\mathcal{I})$.

5. Numerical experiments

Our data sets were taken from a gel image database made available to public by Peter Lemkin from National Cancer Institute (http://binkley.ncifcrf.gov/users/lemkin).

Data set 1 is the Molt-4 data (http://www.lecb.ncifcrf.gov/2DgelDataSets/index.html#MOLT-4).

It contains four gel images (512×512 pixels, 8-bit, 250 microns/pixel). 22 landmarks are picked on each gel.

Data set 2 is the human leukemias data (http://www.lecb.ncifcrf.gov/2DgelDataSets/index.html#HEME-MALIG).

It contains 170 gel images (512×512 pixels, 8-bit, 250 microns/pixel). For more details on this data see [24,25]. We choose only the gel images that contain 22 landmarks on each. There are 123 such gel images in this data set. The above two data sets have been used also in [18].

We adopt the similarity measure used in [18] to measure the quality of the alignment. The normalized l_2-norm difference between two images I and J is given by

$$SM = \sqrt{\frac{\sum_{i,j}(I_{ij} - J_{ij})^2}{MN}} \tag{29}$$

where I_{ij} stands for the intensity value of the pixel (i, j) on image $\mathcal{I}$, and likewise for J_{ij}.

The main computational tools we used is Matlab on a personal computer $P4$ 2.4 Hz with 768 MB of RAM. The QP problems are solved by the state-of-the-art software Mosek using interior-point methods. Experimental results for data set 1 was shown in table 1, along with the alignment results by using piecewise affine and bilinear transformations [18]. In order to have comparable results, the gray level of each image is scaled to the interval [0, 255] before computing the similarity measure. The forward transformations are constructed based on a 16×16 grid for the piecewise affine and bilinear transformations, and a 12×12 grid for the tensor product cubic spline. Then the inverse transformations are constructed based on a refined grid (32×32 grid and 24×24 grid, respectively).

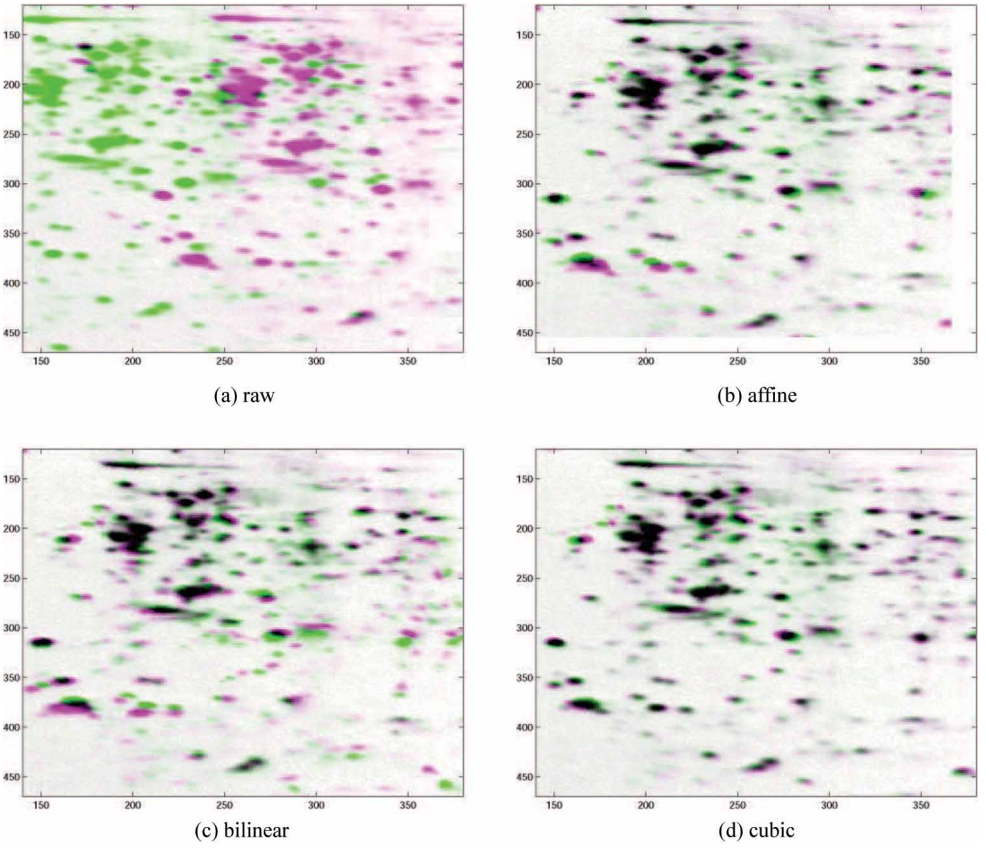
The results presented in table 1 show that the similarity measure is improved by all three methods. It is remarkable that the tensor product cubic spline method performs the best, although it uses a coarser grid. The fact that the tensor product cubic spline method performs better can be also seen, by visual inspection, in figure 2. We believe that the similarity between images 1 and 4 is worse than the similarity between images 1 and 2 because images 1 and 4

Table 1. A sample comparison of the similarity measurse – the molt-4 data.

	Comparison made for	Images		
		1 versus 2	1 versus 3	1 versus 4
	Source images	41.17	49.72	34.30
Affine	32×32 grid	16.77	33.27	20.88
Bilinear	32×32 grid	19.06	34.36	21.55
Cubic	24×24 grid	12.64	30.78	19.22

differ in their protein content, as suggested in figure 3. It is exactly such differences that proteomics studies are looking for.

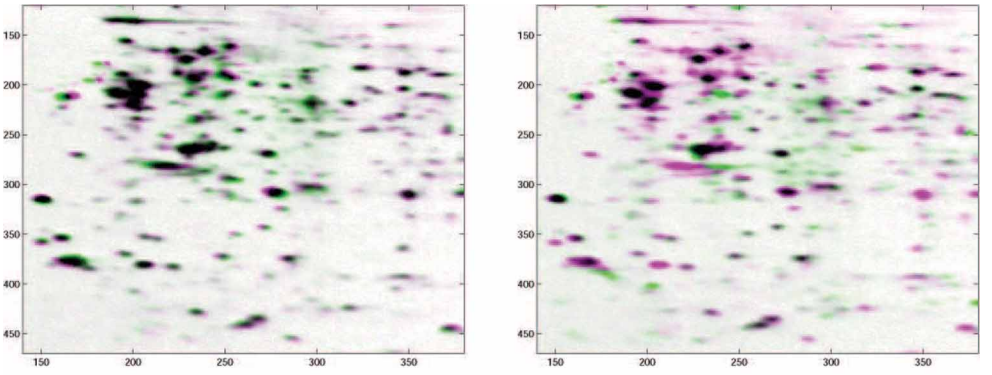
There are 123 gel images in data set 2. We have managed to align this large collection of gel images by using a combined forward-inverse tensor product cubic spline approach. Because of lack of space, we do not show any images from this data set. As a simple example, we show in table 2 a sample result of aligning image 1 and image 3, along with the alignment results by using piecewise affine and bilinear transformations. Here all three methods obtain about the same improvement of the similarity measure. We note however that the tensor product cubic spline method uses a coarser grid.



(a) raw (b) affine

(c) bilinear (d) cubic

Figure 2. The superimposition of images 1 and 2 in the molt data: (a) shows the superimposition of the original images; (b) shows the superimposition of the images after affine transformations; (c) shows the superimposition of the images after bilinear transformations; (d) shows the superimposition of the images after cubic transformation.



(a) images 1 and 2 (b) images 1 and 4

Figure 3. Alignment result of a pair of images in the MOLT-4 data, using tensor product cubic spline: (a) shows the superimposition of transformed images 1 and 2. (b) shows the superimposition of transformed images 1 and 4.

Table 2. A sample comparison of the similarity measure of images 1 and 3 in the human leukemia data.

	Original images	32×32 grid affine	32×32 grid bilinear	24×24 grid cubic
SM	23.04	13.42	13.65	13.59

6. Conclusion

In the present paper, we have presented a methodology for aligning families of 2D gels that does not rely on choosing one of the gels as reference, but constructs an ideal gel and the corresponding warping transformations from the solution of a quadratic programming problem. We have used a combined forward–inverse tensor product cubic splines approach to align very large collections of gel images. By choosing proper base functions and constraints, the resulting tensor product cubic splines are globally C^2 with optimized curvature.

To our knowledge, this is the first multi-resolution method that yields a C^2 warping map for aligning 2D gels. It gives the same resolution as the previous method on a coarser grid, thus reducing the computational time. It also provides a smooth, minimal curvature transformation of the image. Theoretical considerations as well as initial numerical experiments show the potential of our approach. Further experimental work, supported by the National Institute of Health, Grant No. 1R01GM075298-01, will be done to compare the performance of our method of state-of-the-art commercial software packages and to improve its robustness and efficiency.

This application shows the importance of optimization techniques in the field of proteomics. The development of interior-point software in recent years makes it possible to have this type of large-scale proteomics studies.

Acknowledgements

This work was supported in part by the National Institute of Health, Grant No. 1R01GM075298-01.

References

[1] Page, M.J., Amess, B., Rohlff, C., Stubberfield, C. and Parekh, R., 1999, Proteomics: a major new technology for the drug discovery process. *Drug Discovery Today*, **4**, 55–62.

[2] Westermeier, R. and Naven, T., 2002, *Proteomics in Practice* (1st edn) (Weinheim: Wiley-VCH).

[3] O'Farrell, P.H., 1975, High resolution two-dimensional electrophoresis of proteins. *Journal of Computational Biology*, **250**, 4007–4021.

[4] Klose, J., 1975, Protein mapping by combined isoelectric focusing and electrophoresis: a two-dimensional technique. *Humangenetik*, **26**, 231–234.

[5] Blum, H., Beier, H. and Gross, H.J., 1987, Silver stain protocol. *Electrophoresis*, **8**, 93–99.

[6] Conradsen, K. and Pedersen, J., 1992, Analysis of two-dimensional electrophoretic gels. *Biometrics*, **48**, 1273–1287.

[7] Voss, T. and Haber, P., 2000, Observations on the reproducibility and matching efficiency of two-dimensional electrophoresis gels: Consequences for comprehensive data analysis. *Electrophoresis*, **21**, 3345–3350.

[8] Dowsey, A.W., Dunn, M.J. and Yang, G.-Z., 2003, The role of bioinformatics in two-dimensional gel electrophoresis. *Proteomics*, **3**, 1567–1596.

[9] Salmi, J., Aittokallio, T., Westerholm, J., Griese, M., Rosengren, A., Numan, T.A., Lahesmaa, R. and Nevalainen, O., 2002, Hierarchical grid transformation for image warping in the analysis of two-dimensional electrophoresis gels. *Proteomics*, **2**, 1504–1515.

[10] Kaczmarek, K., Walczak, B., de Jong, S. and Vandeginste, B.G.M., 2003, Comparison of image-transformation methods used in matching 2d gel electrophoresis images. *Acta Chromatographica*, **13**, 7–21.

[11] Kaczmarek, K., Walczak, B., de Jong, S. and Vandeginste, B.G.M., 2003, Matching 2d gel electrophoresis images. *Journal of Chemical Information and Computer Sciences*, **43**, 978–986.

[12] Glasbey, C.A. and Mardia, K.V., 1998, A review of image warping methods. *Journal of Applied Statistics*, **25**, 155–171.

[13] Gustafsson, J.S., Blomberg, A. and Rudemo, M., 2002, Warping two-dimensional electrophoresis gel images to correct for geometric distortions of the spot pattern. *Electrophoresis*, **23**, 1731–1744.

[14] Goshtasby, A., 1986, Piecewise linear mapping functions for image registration. *Pattern Recognition*, **19**, 459–466.

[15] Goshtasby, A., 1987, Piecewise cubic mapping functions for image registration. *Pattern Recognition*, **20**, 525–533.

[16] Smilansky, Z., 2001, Automatic registration for images of two-dimensional protein gels. *Electrophoresis*, **22**, 1616–1626.

[17] Veeser, S., Dunn, M.J. and Yang, G.-Z., Multiresolution image registration for two-dimensional gel electrophoresis. *Proteomics*, **1**, 856–870.

[18] Potra, F.A. and Liu, X., 2004, Aligning families of 2d-gels by a combined multi-resolution forward-inverse transformation approach. Preprint, UMBC. *Journal of Computational Biology*. Accepted for publication.

[19] Berger, S.A., Webster, W.C., Tapia, R.A. and Atkins, D.A., 1966, Mathematical ship lofting. *Journal of Ship Research*, **10**, 203–222.

[20] de Boor, C., 1978, *A practical guide to splines, Applied Mathematical Sciences*, Vol. 27 (New York: Springer Verlag).

[21] Forsey, D. and Bartels, R.H., 1995, Surface fitting with hierarchical splines. *ACM Transactions on Graphics*, **14**, 134–161.

[22] O'Dwyer, J. and Vonhof, E., 1994, Saturable variable reluctance motor simulation using spline functions. Preprint, University College, Dublin, Ireland.

[23] Potra, F.A., Liu, X., Seillier-Moiseiwitsch, F., Roy, A., Hang, Y., Marten, M.R., Raman, B. and Whisnant, C., 2004, Protein image alignment via global and segmented local forward affine transformations. Preprint, UMBC, *Journal of Computational Biology*. Accepted for publication.

[24] Lester, E.P., Lemkin, P. and Lipkin, L., 1982, A two-dimensional gel analysis of autologous t and b lymphoblastoid cell lines. *Clinical Chemistry*, **28**, 828–839.

[25] Lester, E.P., Lemkin, P. and Lipkin, L., 1984, Protein indexing in leukemias and lymphomas. *Annals of New York Academy of Science*, **428**, 158–172.