Numerical Algorithms (2006) **41:** 203–217
DOI 10.1007/s11075-005-9004-4

An algorithm for matching point sets using the l_1 norm

I. Al-Subaihi and G. A. Watson

Department of Mathematics, University of Dundee, Dundee, DD1 4HN, UK
E-mail: gawatson@maths.dundee.ac.uk

Received 24 February 2005; accepted 14 August 2005
Communicated by C. Brezinski

The problem is considered of matching two sets of points in R^n, by translation and rotation. There are many applications, for example in geodesy, computer vision and in the assessment of manufactured parts. When the matching criterion is least squares, there is a well known solution process based on the singular value decomposition of an $n \times n$ matrix. Here we consider the use of the l_1 norm, which may be more appropriate than least squares in the context of wild points in the data. An algorithm is developed, and is illustrated by some examples for the case $n = 3$.

Keywords: point sets in R^n, matching, l_1 norm

1. Introduction

Consider 2 sets of points in R^n,

$$\mathbf{p}_i, \mathbf{q}_i, i = 1, \ldots, m.$$

We are interested in the problem of trying to transform one set into the other by applying translation and rotation. This is the problem of point set matching, a problem which arises in many applications, for example in geodesy, computer vision, or in the analysis of measurements of a manufactured part, where a part defined by a set of measured points is to be matched to another set of points defined by a model for the purpose of quality control. Let $\mathbf{t}$ denote the translation, and let rotation be defined by an orthogonal matrix R. Then if

$$r_i = \|\mathbf{v}_i\|, i = 1, \ldots, m,$$

where

$$\mathbf{v}_i = \mathbf{p}_i - R\mathbf{q}_i - \mathbf{t}, i = 1, \ldots, m,$$

and unadorned norms are l_2 norms, the matching problem can be posed as

$$\text{minimize } \|\mathbf{r}\|_A,$$

where the norm is a norm on R^m.

When $\|.\|_A = \|.\|$, there is a well known method for this problem due to Hanson and Norris [3], which uses the singular value decomposition (SVD) of an $n \times n$ matrix.

When the norm is the Chebyshev norm, a method is given in [11] which applies when $n = 2$. Methods for the same problem based on Lawson's algorithm are considered in [9]. The Chebyshev norm has important applications to accept/reject decisions for manufactured parts.

The orthogonal matrix R has $N = \frac{n(n-1)}{2}$ degrees of freedom (excluding a sign, which can be fixed), and a representation in terms of individual rotation angles is given by writing R as $R(\phi)$ where $\phi \in R^N$, where

$$R(\phi) = R_1(\phi_1)R_2(\phi_2)\dots R_N(\phi_N), \tag{1}$$

and where each $R_j(\phi_j)$ is a Givens rotation matrix $G(j_1, j_2, \phi_j)$ representing a counter-clockwise rotation through an angle ϕ_j in the (j_1, j_2) coordinate plane [2]. Clearly there are ${}^nC_2 = \frac{n(n-1)}{2} = N$ selections of coordinate planes.

Späth [5] develops an iterative descent method for the (weighted) least squares problem when $n = 3$ which explicitly uses (1), and so does not involve the SVD: see also [6]. This is extended to more general n in [7].

Both the Chebyshev and l_1 norms are considered in [5] and in particular an algorithm for the l_1 problem is proposed based on solving a sequence of l_2 problems, although it is pointed out that convergence results are poor. The l_1 norm is of value in the context of wild points or outliers in the data, which may be a feature of such problems (for example [1]). The purpose of this paper is to deal with the case when the l_1 norm is the right one to use and to look at alternative possibilities, based on equation (1), which have a chance of good performance.

In the next section the problem is identified, and the following two sections show how algorithms of Newton type may be applied. Starting points may be obtained from the l_2 solution, and in section 5 we summarise the Hanson–Norris method. Finally in section 6, numerical results are presented for some examples in the case $n = 3$.

2. The l_1 problem

We wish to minimize

$$F(\phi, \mathbf{t}) = \|\mathbf{r}\|_1 = \sum_{i=1}^{m} \|\mathbf{v}_i\|. \tag{2}$$

It is possible to include weights, but this does not add much, so we do not consider this. Difficulties in minimizing this function are mainly caused by a lack of differentiability when any component of $\mathbf{r}$ is zero. Let $R^{(j)}$ denote $\frac{\partial R(\phi)}{\partial \phi_j}$ so that

$$R^{(j)} = R_1(\phi_1)\dots R_{j-1}(\phi_{j-1})R'_j(\phi_j)R_{j+1}(\phi_{j+1})\dots R_N(\phi_N), j = 1, \dots, N,$$

where the dash denotes differentiation with respect to the single variable on which R_j depends. Then if no component of $\mathbf{r}$ is zero, we can write

$$\nabla_{\phi,\mathbf{t}} F = -\sum_{i=1}^{m} \frac{\mathbf{v}_i^T}{\|\mathbf{v}_i\|}[R^{(1)}\mathbf{q}_i : \dots : R^{(N)}\mathbf{q}_i : I].$$

In other words

$$\nabla_{\phi,\mathbf{t}} F = -\sum_{i=1}^{m} \frac{1}{\|\mathbf{v}_i\|} [\mathbf{v}_i^T R^{(1)} \mathbf{q}_i : \ldots : \mathbf{v}_i^T R^{(N)} \mathbf{q}_i : \mathbf{v}_i^T]. \tag{3}$$

If no r_i (or $\mathbf{v}_i$) is zero, then $\nabla_{\phi,\mathbf{t}} F = 0$ is a necessary condition for a minimum. Otherwise the necessary (or stationary point) condition becomes

$$\sum_{i \in Z} \mathbf{w}_i^T [R^{(1)} \mathbf{q}_i : \ldots : R^{(N)} \mathbf{q}_i : I]$$

$$+ \sum_{i \notin Z} \frac{1}{\|\mathbf{v}_i\|} [\mathbf{v}_i^T R^{(1)} \mathbf{q}_i : \ldots : \mathbf{v}_i^T R^{(N)} \mathbf{q}_i : \mathbf{v}_i^T] = 0, \tag{4}$$

with

$$\|\mathbf{w}_i\| \le 0, i \in Z,$$

where

$$Z = \{i : \|\mathbf{v}_i\| = 0\}.$$

Each zero component of $\mathbf{r}$ of course corresponds to n conditions. Assume there are two zero components at a solution, corresponding to indices j, k. Then

$$\mathbf{p}_j - R(\phi)\mathbf{q}_j - \mathbf{t} = 0,$$

$$\mathbf{p}_k - R(\phi)\mathbf{q}_k - \mathbf{t} = 0$$

so that

$$\mathbf{p}_j - \mathbf{p}_k = R(\phi)(\mathbf{q}_j - \mathbf{q}_k).$$

But this is only possible if

$$\|\mathbf{p}_j - \mathbf{p}_k\| = \|\mathbf{q}_j - \mathbf{q}_k\|.$$

This suggests that if the data are genuinely contaminated by error and not ‘fixed’ in a particular way, then at a solution to the problem there is **at most one zero component of r**. We will interpret this as a nondegenerate situation.

Note that the terms in (3) can be expressed in a slightly different way. For example

$$\begin{aligned} \mathbf{v}_i^T R^{(1)} \mathbf{q}_i &= (\mathbf{p}_i - \mathbf{t} - R(\phi)\mathbf{q}_i)^T R^{(1)} \mathbf{q}_i \\ &= (\mathbf{p}_i - \mathbf{t})^T R^{(1)} \mathbf{q}_i \end{aligned}$$

because

$$\mathbf{q}_i^T R_N(\phi_N)^T \ldots R_2(\phi_2)^T R_1'(\phi_1)^T R_1(\phi_1) R_2(\phi_2) \ldots R_N(\phi_N)\mathbf{q}_i = 0$$

since $R_1'(\phi_1)^T R_1(\phi_1)$ is skew-symmetric. Similarly for the other derivatives. Thus equation (3) becomes

$$\nabla_{\phi,\mathbf{t}} F = -\sum_{i=1}^{m} \frac{1}{\|\mathbf{v}_i\|}[(\mathbf{p}_i - \mathbf{t})^T R^{(1)}\mathbf{q}_i : \ldots : (\mathbf{p}_i - \mathbf{t})^T R^{(N)}\mathbf{q}_i : \mathbf{v}_i^T]. \tag{5}$$

Perhaps the most commonly used method for solving nonlinear l_1 minimization problems is the Gauss–Newton method or one of its variants. However there are two potentially serious drawbacks here. The first is the lack of differentiability of any component of **r** which is zero (unlike the l_1 orthogonal distance regression problem [10], which has a similar feature, this does seem a serious problem). The second is the fact that second order convergence would require $N + n$ zero components of **r** [8] and we have seen that at most one is expected. So the Gauss–Newton method in its usual form does not seem appropriate for this problem. Further, the lack of differentiability of the components of **r** complicates the application of other methods (for example [4]) which can normally be used to restore a fast convergence rate.

The above discussion of zeros (and the fact that in general there is at most one) suggests that applying a smooth optimization method in a controlled way might be a good way to tackle this problem. In some ways this is in the spirit of other smoothing methods which have been given for (linear) l_1 problems. If we apply an optimization method for smooth problems, then if there are no zeros, this should work well. If there is a zero, and we can identify the index, then we can eliminate the non-differentiability, and again apply a method for smooth problems to a reduced problem. We will investigate these ideas in the next two sections.

3. The case when the problem is differentiable

Let us assume that the $F(\phi, \mathbf{t})$ is differentiable, and we want to generate a descent process. We have the expression for $\nabla_{\phi,\mathbf{t}} F$ given by (5), and in this form it is easy to calculate second derivatives. First define for every $i = 1, \ldots, m$,

$$\mathbf{c}_i^T = [(\mathbf{p}_i - \mathbf{t})^T R^{(1)}\mathbf{q}_i : \ldots : (\mathbf{p}_i - \mathbf{t})^T R^{(N)}\mathbf{q}_i : \mathbf{v}_i^T],$$

so

$$\nabla_{\phi,\mathbf{t}} \|\mathbf{v}_i\| = -\frac{1}{\|\mathbf{v}_i\|}\mathbf{c}_i^T.$$

Then

$$\nabla_{\phi,\mathbf{t}} F = -\sum_{i=1}^{m} \frac{1}{\|\mathbf{v}_i\|}\mathbf{c}_i^T.$$

Also

$$\nabla^2_{\phi,\mathbf{t}}F = \nabla_{\phi,\mathbf{t}}(\nabla_{\phi,\mathbf{t}}F^T) = -\sum_{i=1}^{m}\nabla_{\phi,\mathbf{t}}(\frac{1}{\|\mathbf{v}_i\|}\mathbf{c}_i)$$

$$= -\sum_{i=1}^{m}\left\{\frac{1}{\|\mathbf{v}_i\|}\nabla_{\phi,\mathbf{t}}\mathbf{c}_i + \mathbf{c}_i\nabla_{\phi,\mathbf{t}}\frac{1}{\|\mathbf{v}_i\|}\right\}.$$

Now $\nabla_{\phi,\mathbf{t}}\mathbf{c}_i$ can be partitioned naturally into the block 2×2 matrix

$$\nabla_{\phi,\mathbf{t}}\mathbf{c}_i = \begin{bmatrix} G_i & -H_i \\ -H_i^T & -I \end{bmatrix},$$

where

$$G_i = \begin{bmatrix} (\mathbf{p}_i - \mathbf{t})^T R_1'' R_2 \dots R_N \mathbf{q}_i : \dots : (\mathbf{p}_i - \mathbf{t})^T R_1' R_2 \dots R_N' \mathbf{q}_i \\ (\mathbf{p}_i - \mathbf{t})^T R_1' R_2' \dots R_N \mathbf{q}_i : \dots : (\mathbf{p}_i - \mathbf{t})^T R_1 R_2' \dots R_N' \mathbf{q}_i \\ \dots \\ (\mathbf{p}_i - \mathbf{t})^T R_1' R_2 \dots R_N' \mathbf{q}_i : \dots : (\mathbf{p}_i - \mathbf{t})^T R_1 R_2 \dots R_N'' \mathbf{q}_i \end{bmatrix}, \tag{6}$$

$$H_i^T = [R^{(1)}\mathbf{q}_i : \dots : R^{(N)}\mathbf{q}_i],$$

with

$$R_j'' = \frac{\partial^2 R_j}{\partial \phi_j^2}, j = 1, \dots, m.$$

Further

$$\nabla_{\phi,\mathbf{t}}\frac{1}{\|\mathbf{v}_i\|} = -\frac{1}{\|\mathbf{v}_i\|^2}\nabla_{\phi,\mathbf{t}}\|\mathbf{v}_i\| = \frac{1}{\|\mathbf{v}_i\|^3}\mathbf{c}_i^T.$$

Therefore we can write

$$\nabla^2_{\phi,\mathbf{t}}F = \sum_{i=1}^{m}\left\{\frac{1}{\|\mathbf{v}_i\|}\begin{bmatrix} -G_i & H_i^T \\ H_i^T & I \end{bmatrix} - \frac{\mathbf{c}_i\mathbf{c}_i^T}{\|\mathbf{v}_i\|^3}\right\}. \tag{7}$$

The Newton step is given by

$$(\nabla_{\phi,\mathbf{t}}F)^T + \nabla^2_{\phi,\mathbf{t}}F\mathbf{d} = 0.$$

Far from a solution the Hessian matrix may not be positive definite, and we would prefer to use a positive definite approximation in case we need a line search. But it is easy to identify a positive definite part of the Hessian. To see this, it helps to consider an alternative way of looking at the derivatives. We can write

$$\nabla_{\phi,\mathbf{t}}F = \sum_{i=1}^{m} \frac{1}{\|\mathbf{v}_i\|} \mathbf{v}_i^T \nabla_{\phi,\mathbf{t}} \mathbf{v}_i,$$

so that

$$\nabla^2_{\phi,\mathbf{t}}F = \sum_{i=1}^{m} \left\{ \frac{1}{\|\mathbf{v}_i\|} \nabla_{\phi,\mathbf{t}} \mathbf{v}_i^T \nabla_{\phi,\mathbf{t}} \mathbf{v}_i + \frac{1}{\|\mathbf{v}_i\|} (\nabla^2_{\phi,\mathbf{t}} \mathbf{v}_i) \mathbf{v}_i - \frac{1}{\|\mathbf{v}_i\|^3} (\nabla_{\phi,\mathbf{t}} \mathbf{v}_i^T \mathbf{v}_i)(\mathbf{v}_i^T \nabla_{\phi,\mathbf{t}} \mathbf{v}_i) \right\}$$

or

$$\nabla^2_{\phi,\mathbf{t}}F = \sum_{i=1}^{m} \left\{ \frac{1}{\|\mathbf{v}_i\|} \begin{bmatrix} H_i H_i^T & H_i \\ H_i^T & I \end{bmatrix} + \frac{1}{\|\mathbf{v}_i\|} (\nabla^2_{\phi,\mathbf{t}} \mathbf{v}_i) \mathbf{v}_i - \frac{\mathbf{c}_i \mathbf{c}_i^T}{\|\mathbf{v}_i\|^3} \right\}, \tag{8}$$

since clearly

$$\nabla_{\phi,\mathbf{t}} \mathbf{v}_i = -[H_i^T : I],$$

and

$$\mathbf{c}_i^T = -\mathbf{v}_i^T \nabla_{\phi,\mathbf{t}} \mathbf{v}_i = [\mathbf{v}_i^T H_i^T : \mathbf{v}_i^T] = [(\mathbf{p}_i - \mathbf{t})^T H_i^T : \mathbf{v}_i^T],$$

as before. Note also that if we think of the matrices G_i given by (6) as being

$$G_i(\mathbf{p}_i - \mathbf{t}),$$

showing the dependence on $(\mathbf{p}_i - \mathbf{t})$, then

$$(\nabla^2_{\phi,\mathbf{t}} \mathbf{v}_i) \mathbf{v}_i = \begin{bmatrix} -G_i(\mathbf{v}_i) & 0 \\ 0 & 0 \end{bmatrix}.$$

Further

$$\begin{aligned} G_i(\mathbf{v}_i) &= G_i(\mathbf{p}_i - \mathbf{t} - D(\phi)\mathbf{q}_i) \\ &= G_i(\mathbf{p}_i - \mathbf{t}) - G_i(D(\phi)\mathbf{q}_i) \\ &= G_i(\mathbf{p}_i - \mathbf{t}) + H_i H_i^T, \end{aligned}$$

as may readily be shown. This demonstrates the connection between the two different forms of $\nabla^2 F$ given by equations (7) and (5) above.

Substituting for $\mathbf{c}_i$ into equation (8) gives

$$\begin{aligned} \nabla^2_{\phi,\mathbf{t}}F &= \sum_{i=1}^{m} \left\{ \frac{1}{\|\mathbf{v}_i\|} \begin{bmatrix} H_i \\ I \end{bmatrix} [H_i^T : I] - \begin{bmatrix} H_i \\ I \end{bmatrix} \frac{\mathbf{v}_i \mathbf{v}_i^T}{\|\mathbf{v}_i\|^3} [H_i^T : I] + \frac{1}{\|\mathbf{v}_i\|} (\nabla^2_{\phi,\mathbf{t}} \mathbf{v}_i) \mathbf{v}_i \right\} \\ &= \sum_{i=1}^{m} \{ R_i + \frac{1}{\|\mathbf{v}_i\|} (\nabla^2_{\phi,\mathbf{t}} \mathbf{v}_i) \mathbf{v}_i \} \end{aligned}$$

where

$$R_i = \frac{1}{\|\mathbf{v}_i\|}\begin{bmatrix} H_i \\ I \end{bmatrix}\left(I - \frac{\mathbf{v}_i\mathbf{v}_i^T}{\|\mathbf{v}_i\|^2}\right)[H_i^T \,:\, I].$$

Note that for any $\mathbf{x} \in R^n$,

$$\mathbf{x}^T\left(I - \frac{\mathbf{v}_i\mathbf{v}_i^T}{\|\mathbf{v}_i\|^2}\right)\mathbf{x} = \mathbf{x}^T\mathbf{x} - \frac{(\mathbf{x}^T\mathbf{v}_i)^2}{\|\mathbf{v}_i\|^2}$$
$$\geq 0,$$

using the Cauchy-Schwartz inequality, and so R_i is a positive semi-definite matrix. Also $\sum_{i=1}^m R_i$ is positive definite unless there exists a vector $\mathbf{y} \in R^{n+N}$, non-zero, such that

$$\mathbf{y}^T R_i \mathbf{y} = 0, i = 1, \ldots, m.$$

Thus we can normally choose as a positive definite approximation to $\nabla^2_{\phi,\mathbf{t}}F$

$$\nabla^2_{\phi,\mathbf{t}}F \approx \sum_{i=1}^m R_i = \sum_{i=1}^m \frac{1}{\|\mathbf{v}_i\|}\begin{bmatrix} H_i \\ I \end{bmatrix}\left(I - \frac{\mathbf{v}_i\mathbf{v}_i^T}{\|\mathbf{v}_i\|^2}\right)[H_i^T \,:\, I]. \tag{9}$$

This enables us to define a method which is in the spirit of a Gauss–Newton method, since we have dropped second derivative terms to get a positive definite approximation to the Hessian matrix of F.

A compromise might be to take

$$\nabla^2_{\phi,\mathbf{t}}F \approx \sum_{i=1}^m R_i - \theta\sum_{i=1}^m \frac{1}{\|\mathbf{v}_i\|}\begin{bmatrix} G_i(\mathbf{v}_i) & 0 \\ 0 & 0 \end{bmatrix},$$

where $0 \leq \theta \leq 1$. The choice $\theta = 0$ can be interpreted as giving the ‘Gauss–Newton’ method, the choice $\theta = 1$ is the Newton method and it may be possible to adaptively adjust θ as the computation proceeds.

4. The case when there is one known zero component of r

Suppose it is known that there is one zero component at a solution, corresponding to $r_k = \|\mathbf{v}_k\| = 0$ so that

$$\mathbf{t} = \mathbf{p}_k - R(\phi)\mathbf{q}_k.$$

Then $\mathbf{t}$ can be eliminated and the problem reduced to one in ϕ only, for which a zero derivative can be sought. In other words we want to minimize

$$\sum_{i\neq k} \|\mathbf{v}_i\|$$

where

$$\mathbf{v}_i = (\mathbf{p}_i - \mathbf{p}_k) - R(\phi)(\mathbf{q}_i - \mathbf{q}_k), i = 1, \ldots, m,$$

and so a zero derivative gives, as for (3),

$$\sum_{i\neq k} \frac{1}{\|\mathbf{v}_i\|} [\mathbf{v}_i^T R^{(1)}(\mathbf{q}_i - \mathbf{q}_k) : \ldots : \mathbf{v}_i^T R^{(N)}(\mathbf{q}_i - \mathbf{q}_k)] = 0.$$

Note that in this case, from equation (4), the stationary point condition implies that

$$\mathbf{w}_k + \sum_{i\neq k} \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|} = 0.$$

Thus a check for a stationary point is that

$$\| \sum_{i\neq k} \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|} \| \leq 1. \tag{10}$$

Newton's method can also be implemented as before for the reduced problem, with derivatives calculated in a similar way. If we define for each i,

$$\hat{\mathbf{p}}_i = \mathbf{p}_i - \mathbf{p}_k, \hat{\mathbf{q}}_i = \mathbf{q}_i - \mathbf{q}_k,$$

$$\hat{\mathbf{c}}_i^T = [\hat{\mathbf{p}}_i^T R^{(1)} \hat{\mathbf{q}}_i : \ldots : \hat{\mathbf{p}}_i^T R^{(N)} \hat{\mathbf{q}}_i],$$

$$\hat{G}_i = \begin{bmatrix} \hat{\mathbf{p}}_i^T R_1'' R_2 \ldots R_N \mathbf{q}_i : \ldots : \hat{\mathbf{p}}_i^T R_1' R_2 \ldots R_N' \mathbf{q}_i \\ \hat{\mathbf{p}}_i^T R_1' R_2' \ldots R_N \mathbf{q}_i : \ldots :; \hat{\mathbf{p}}_i^T R_1 R_2' \ldots R_N' \mathbf{q}_i \\ \ldots \\ \hat{\mathbf{p}}_i^T R_1' R_2 \ldots R_N' \mathbf{q}_i : \ldots : \hat{\mathbf{p}}_i^T R_1 R_2 \ldots R_N'' \mathbf{q}_i \end{bmatrix},$$

then

$$\nabla_\phi F = -\sum_{i\neq k} \frac{1}{\|\mathbf{v}_i\|} \hat{\mathbf{c}}_i^T$$

$$\nabla_\phi^2 F = -\sum_{i\neq k} \left\{ \frac{1}{\|\mathbf{v}_i\|} \hat{G}_i + \frac{1}{\|\mathbf{v}_i\|^3} \hat{\mathbf{c}}_i \hat{\mathbf{c}}_i^T \right\}. \tag{11}$$

Again while this gives a convenient form for the Hessian, it is not so helpful far from a solution when it might not be positive definite. So again we try to identify a positive definite 'part'. As before we can write

$$\nabla_\phi F = \sum_{i\neq k} \frac{1}{\|\mathbf{v}_i\|} \mathbf{v}_i^T \nabla_\phi \mathbf{v}_i,$$

so that

$$\nabla_\phi^2 F = \sum_{i \neq k} \left\{ \frac{1}{\|\mathbf{v}_i\|} (\nabla_\phi^2 \mathbf{v}_i) \mathbf{v}_i + \frac{1}{\|\mathbf{v}_i\|} (\nabla_\phi \mathbf{v}_i^T)(\nabla_\phi \mathbf{v}_i) - \frac{1}{\|\mathbf{v}_i\|^3} (\nabla_\phi \mathbf{v}_i^T \mathbf{v}_i)(\mathbf{v}_i^T \nabla_\phi \mathbf{v}_i) \right\}$$

or

$$\nabla_\phi^2 F = \sum_{i \neq k} \left\{ S_i + \frac{1}{\|\mathbf{v}_i\|} (\nabla_\phi^2 \mathbf{v}_i) \mathbf{v}_i \right\} \tag{12}$$

where

$$S_i = \frac{1}{\|\mathbf{v}_i\|} \hat{H}_i \left(I - \frac{\mathbf{v}_i \mathbf{v}_i^T}{\|\mathbf{v}_i\|^2} \right) \hat{H}_i^T,$$

with

$$\hat{H}_i^T = [R^{(1)} \hat{\mathbf{q}}_i : \ldots : R^{(N)} \hat{\mathbf{q}}_i],$$

and

$$\hat{\mathbf{c}}_i = \hat{H}_i \mathbf{v}_i,$$

for all i, $i = 1, \ldots, m, i \neq k$.

Arguing as before we can normally take the positive definite approximation to $\nabla_\phi^2 F$

$$\nabla_\phi^2 F \approx \sum_{i \neq k} S_i = \sum_{i \neq k} \left\{ \frac{1}{\|\mathbf{v}_i\|} \hat{H}_i \left(I - \frac{\mathbf{v}_i \mathbf{v}_i^T}{\|\mathbf{v}_i\|^2} \right) \hat{H}_i^T \right\}. \tag{13}$$

Note that if we again think of $\hat{G}_i$ as $\hat{G}_i(\hat{\mathbf{p}}_i)$, showing the dependence on $\hat{\mathbf{p}}_i$, then

$$(\nabla_\phi^2 \mathbf{v}_i) \mathbf{v}_i = -\hat{G}(\mathbf{v}_i),$$

and we have for $i \neq k$,

$$\begin{aligned} G_i(\mathbf{v}_i) &= \hat{G}_i(\hat{\mathbf{p}}_i - D(\phi)\hat{\mathbf{q}}_i) \\ &= \hat{G}_i(\hat{\mathbf{p}}_i) - \hat{G}_i(D(\phi)\hat{\mathbf{q}}_i) \\ &= \hat{G}_i(\hat{\mathbf{p}}_i) + \hat{H}_i \hat{H}_i^T, \end{aligned}$$

again showing the connection between the two forms of Hessian (11) and (12).

Again, as before, another possibility is to take

$$\nabla_\phi^2 F \approx \sum_{i \neq k} S_i - \theta \sum_{i \neq k} \frac{1}{\|\mathbf{v}_i\|} \hat{G}_i(\mathbf{v}_i),$$

where $\theta = 0$ again can be interpreted as giving a matrix in the spirit of that used in the Gauss–Newton method, $\theta = 1$ is Newton's method, and it may be possible to adaptively adjust θ as the computation proceeds.

An alternative to the above approximation is as follows.

- Work out LDL^T factors of the $N \times N$ matrix $A = \nabla^2_\phi F$. This is easily done, although may not exist for degenerate cases, which we exclude.
- Replace A by a positive definite approximation if necessary, e.g., $L\hat{D}L^T$ where $\hat{D}$ contains elements which are the modulus of those of D.
- Use a line search.

When we get close enough to a solution, A should be positive definite, full steps will be possible and second order convergence will occur.

5. Starting values

It is necessary to provide starting values for any iterative algorithm. One possibility is to solve the l_2 problem, using the Hanson–Norris method [3], although it should be borne in mind that in the context of data containing outliers, this may not be a particularly good approximation. The work here is in the SVD of an $n \times n$ matrix. For convenience and completeness we summarise the procedure. The l_2 problem is to minimize

$$\sum_{i=1}^{m} \|\mathbf{v}_i\|^2 = \sum_{i=1}^{m} \|\mathbf{p}_i - Q\mathbf{q}_i - \mathbf{t}\|^2$$

with respect to $\mathbf{t}, Q$ where $Q^TQ = I$. Differentiating with respect to $\mathbf{t}$ gives

$$\mathbf{t} = \bar{\mathbf{p}} - Q\bar{\mathbf{q}},$$

where

$$\bar{\mathbf{p}} = \frac{1}{m}\sum_{i=1}^{m} \mathbf{p}_i, \;\; \bar{\mathbf{q}} = \frac{1}{m}\sum_{i=1}^{m} \mathbf{q}_i.$$

So the problem reduces to that of minimizing over all $Q, Q^TQ = I$

$$\sum_{i=1}^{m} \|\mathbf{p}_i - \bar{\mathbf{p}} - Q(\mathbf{q}_i - \bar{\mathbf{q}})\|^2.$$

Define

$$\mathbf{a}_i = \mathbf{p}_i - \bar{\mathbf{p}}, \;\; i = 1, \ldots, m.$$

$$\mathbf{b}_i = \mathbf{q}_i - \bar{\mathbf{q}}, \;\; i = 1, \ldots, m.$$

and let

$$A = [\mathbf{a}_1 : \mathbf{a}_2 : \ldots : \mathbf{a}_m] \in R^{n \times m},$$

$$B = [\mathbf{b}_1 : \mathbf{b}_2 : \ldots : \mathbf{b}_m] \in R^{n \times m}.$$

Then

$$\begin{aligned} \sum_{i=1}^{m} \|\mathbf{p}_i - \bar{\mathbf{p}} - Q(\mathbf{q}_i - \bar{\mathbf{q}})\|^2 &= \sum_{i=1}^{m} \|\mathbf{a}_i - Q\mathbf{b}_i\|^2 \\ &= \sum_{i=1}^{m} \|Q^T\mathbf{a}_i - \mathbf{b}_i\|^2 \\ &= \|Q^T A - B\|_F^2 \\ &= \|B^T - A^T Q\|_F^2. \end{aligned}$$

The problem

$$\min_{Q^TQ=I} \|B^T - A^T Q\|_F^2$$

is a Procrustes problem and is easily solved (see for example Golub and Van Loan [2], page 582). We have

$$\begin{aligned} \|B^T - A^T Q\|_F^2 &= \text{trace}((B - Q^T A)(B^T - A^T Q)) \\ &= \text{trace}(BB^T) + \text{trace}(AA^T) - 2\text{trace}(Q^T AB^T). \end{aligned}$$

So $\|B^T - A^T Q\|_F^2$ is minimized when $\text{trace}(Q^T AB^T)$ is maximized. Let AB^T have the singular value decomposition $AB^T = U\Sigma V^T$, where

$$\Sigma = \text{diag}\{\sigma_1, \sigma_2, \ldots, \sigma_n\},$$

and let $Z = V^T Q^T U$. Then

$$\begin{aligned} \text{trace}(Q^T AB^T) &= \text{trace}(Q^T U\Sigma V^T) \\ &= \text{trace}(V^T Q^T U\Sigma) \\ &= \text{trace}(Z\Sigma) \\ &= \sum_{i=1}^{n} Z_{ii}\sigma_i \\ &\leq \sum_{i=1}^{n} \sigma_i, \end{aligned}$$

since Z is an orthogonal matrix. The upper bound is attained when $Z = I$, or $Q = UV^T$. To summarise, the solution is obtained as follows:

1. Form $C = AB^T$.
2. Form the SVD of C: $C = U\Sigma V^T$.
3. Set $Q = UV^T$.

We have to recover the explicit angles from Q and this is easily done, following the strategy of [3], where zeros are systematically introduced into Q by premultiplying by N plane rotation matrices to reduce it to diagonal form. For example, for the special case $n = 3$, when $N = 3$ we can take

$$\phi_1 = \alpha, \text{rotation in the } (1,2) \text{ plane},$$

$$\phi_2 = \beta, \text{ rotation in the } (1,3) \text{ plane},$$

$$\phi_3 = \gamma, \text{ rotation in the } (2,3) \text{ plane}.$$

Then we have α given by

$$\tan(\alpha) = \frac{Q(2,1)}{Q(1,1)}, \tag{14}$$

and can define

$$Q_1 = D_1^T(\alpha)Q.$$

Then Q_1 has $(2,1)$ element zero. Now let β be defined by

$$\tan(\beta) = \frac{Q_1(3,1)}{Q_1(1,1)},$$

and define

$$Q_2 = D_2^T(\beta)Q_1.$$

Then Q_2 has $(2,1)$ and $(3,1)$ elements zero (and so also $(1,2)$ and $(1,3)$ elements zero). Finally define γ by

$$\tan(\gamma) = \frac{Q_2(3,2)}{Q_2(2,2)}.$$

Then clearly

$$D_3^T(\gamma)Q_2 = I.$$

In other words

$$D_3^T(\gamma)D_2^T(\beta)D_1^T(\alpha)Q = I,$$

or

$$Q = D_1(\alpha)D_2(\beta)D_3(\gamma).$$

In fact we can give explicit expressions for the angles, namely equation (14) and:

$$\tan(\beta) = \frac{Q(3,1)}{\cos(\alpha)Q(1,1) + \sin(\alpha)Q(2,1)}, \tag{15}$$

$$\tan(\gamma) = \frac{\cos(\beta)Q(3,2) - \sin(\beta)(\cos(\alpha)Q(1,2) + \sin(\alpha)Q(2,2))}{\cos(\alpha)Q(2,2) - \sin(\alpha)Q(1,2)}. \tag{16}$$

6. Numerical results

The contents of the last three sections give the ingredients of algorithms which can be used to solve the l_1 problem. For example, we could consider an algorithm as follows.

1. Solve the l_2 problem by the Hanson and Norris method.
2. Calculate the N rotation angles as indicated in the last section.
3. Assuming no zeros, apply a method as described in section 3, monitoring progress and in particular the closest value of r_i to zero.
4. If a particular value r_k falls below a threshold value, stop and taking $\|\mathbf{v}_k\| = 0$, use a method as described in section 4.
5. On convergence to a certain tolerance, check for a stationary point using equation (10).
6. If a stationary point is obtained, stop, otherwise re-enter Step 3 with a reduced threshold.
7. Repeat until convergence to a given tolerance.

We illustrate for the special (and important) case of $n = 3$.

Example 1. Consider the example from Späth [5]. In particular take data points $\mathbf{p}$ and $\mathbf{q}$ from Examples 3 and 1, respectively. Here $m = 13$.

The l_2 solution gives

$$\mathbf{a} = (-0.0946, 0.3746, 0.3693)^T, \ \mathbf{t} = (1.5303, 4.3571, -2.6011)^T.$$

Using the method of section 3 (Gauss–Newton version), after eight iterations we have $\|\mathbf{v}_2\| < 0.001$. Here

$$\mathbf{a} = (-0.0886, 0.3677, 0.3699)^T, \ \mathbf{t} = (1.3897, 4.3390, -2.5991)^T.$$

Using a modified Newton method from section 4, based on performing an LDL^T factorization of the Hessian and replacing the elements of D by their modulus, followed by a line search, we get after two iterations $\|\mathbf{d}\| < 0.00001$ and the check (10) is satisfied with

$$\|\mathbf{w}_2\| = 0.935.$$

The solution is

$$\mathbf{a} = (-0.088479, 0.369532, 0.370690)^T, \quad \mathbf{t} = (1.388927, 4.338263, -2.601475)^T.$$

Example 2. Consider $m = 50$ with randomly generated $\mathbf{p}$ and $\mathbf{q}$ by first using 'rand('state',1)' followed by random $\mathbf{p}$ then random $\mathbf{q}$. Starting with the l_2 solution, after 48 iterations of a 'Gauss–Newton' method from section 3, $\|\mathbf{v}_{18}\| < 0.001$. This becomes 33 iteration if the Gauss–Newton method switches to Newton when $\|\mathbf{d}\| < 0.01$).

Now use the method of section 4 fixing $\|\mathbf{v}_{18}\| = 0$. There is convergence ($\|\mathbf{d}\| < 0.00001$) after three iterations but the check (10) fails.

Now return to method of section 3, lowering the tolerance. There is convergence ($\|\mathbf{d}\| < 0.000001$) after eight iterations. At the solution $\|\mathbf{v}_{18}\| = 0.0009012..$ but this seems correct, and so there are in fact no zero values of the components. So this is quite a difficult problem, with a "close to zero but not zero" distance. This is the worst scenario: the problem is actually smooth but has some components of $\mathbf{r}$ 'close' to zero at a solution.

Other examples with randomly generated data usually had no zeros and no components of $\mathbf{r}$ close to zero, and so convergence was obtained without difficulty the first time the method of section 3 was applied. However such problems are easy to generate but have little basis in reality.

Example 3. This is really a set of examples constructed as follows. Starting from point sets $\mathbf{q}_i, i = 1, \ldots, m$, (with m having values up to 100) sets $\mathbf{p}_i, i = 1, \ldots, m$ were generated by satisfying

$$\mathbf{p}_i = R(\phi)\mathbf{q}_i - \mathbf{t}, i = 1, \ldots, m,$$

for a particular ϕ and $\mathbf{t}$, perturbing the points $\mathbf{p}_i$ and introducing wild points. In particular, for larger values of m, no zeros tend to be identified and they are solved as differentiable problems, often in around five or so iterations from the l_2 solution.

7. Concluding remarks

The l_1 problem of minimizing (2) is one which in a sense can be very easily solved: (1) if there are no zero components of $\mathbf{r}$ at a solution, then a second order convergence rate is normally achievable; (2) if there is one component of **r** which is known to be zero at a solution, again a second order convergence rate is normally achievable; (3) more than one zero component of **r** at a solution may be regarded as a form of degeneracy.

What makes the problem difficult is that of course it is not normally known in advance if there are zero components of **r** at a solution or not, and if there is a zero component of **r** at a solution, then it is not known in advance which one it is. So a main task in algorithmic development is addressing these issues, and if they can be dealt with in a satisfactory way, then the solution process is relatively straightforward.

The purpose of this paper has been to develop machinery and put forward proposals which can lead to satisfactory and efficient methods of solution.

References

[1] H. Chui and A. Rangarajan, A new algorithm for non-rigid point matching, IEEE Conference on Computer Vision and Pattern Recognition (CVPR) II (2000) 44–51.

[2] G.H. Golub and C.F. Van Loan, *Matrix Computations* (Johns Hopkins University Press, 1989).

[3] R.J. Hanson and M.J. Norris, Analysis of measurements based on the Singular Value Decomposition, SIAM J Sci. Stat. Comp. 2 (1981) 363–373.

[4] W. Murray and M.L. Overton, A projected Lagrangian algorithm for nonlinear l_1 optimization, SIAMJ. Sci. Stat. Comp. 2 (1981) 207–214.

[5] H. Späth, Identifying spatial point sets, Math. Comm. 8 (2003) 69–75.

[6] H. Späth, A numerical method for determining the spatial HELMERT transformation in the case of different scale factors, Z. Vermess.wes. 129 (2004) 255–257.

[7] H. Späth, Fitting affine and orthogonal transformations between two sets of points, Math. Comm. 9 (2004) 27–34.

[8] G.A. Watson, *Approximation Theory and Numerical Methods* (Wiley, Chichester 1980).

[9] G.A. Watson, An approximation problem in the analysis of measurements, in *Approximation Theory VIII, Vol. 1: Approximation and Interpolation*, eds. C.K. Chui and L.L. Schumaker (World Scientific, Singapore, 1995).

[10] G.A. Watson, On the Gauss–Newton method for l_1 orthogonal distance regression, IMA J. Num. Anal. 22 (2002) 345–357.

[11] D. Zwick, A planar minimax algorithm for analysis of co-ordinate measurements, Adv. Comput. Math. 2 (1994) 375–391.