# Determination of transformation parameters between two reference systems without common points. Three application examples from digital terrain models, laser scanning and GNSS seismology

**Athanasios Dermanis**

*Department of Geodesy and Surveying, Aristotle University of Thessaloniki*

**Abstract:** Algorithms for the determination of the transformation parameters from one reference system to the other are presented for the case where the data points lie on the same surface or curve, while there exist no common data points. Iterative solutions are proposed based on localized surface or curve interpolation and least squares best fitting. Three application examples are presented. The first is the matching of two overlapping digital terrain models where horizontally gridded height data are referring to two different horizontal and vertical reference systems. The second is the matching of two point clouds with coordinates obtained from two independent scannings of overlapping surfaces, which refer to different three-dimensional reference systems. The last example is the matching of velocities derived from GNSS and accelerometers, with different reference and time systems, where the data are velocity component time series, i.e. points on a three-dimensional time-dependent curve.

## 1. Introduction

A very common problem in geodesy and surveying is the determination of transformation parameters relating two reference systems, each one established independently for a set of points. The relation is determined thanks to the existence of points common to both point sets, e.g. through the common subnetwork of two overlapping control networks, in the framework of network densification. There are applications however within the broader field of geomatics where no such common points exist, where the available observational data refer to different point sets on the same surface or curve. The common shape of the surface or curve provides the information for relating the established reference systems when as usual the geometric form of a set of points is conveniently described using coordinates. We will give here three examples from quite diverse application fields and we will sketch a possible way of treating this problem in a practically feasible way. From a theoretical point of view, it may be more attractive, or at least more elegant, to construct an analytical form for the curve or surface by interpolation and then consider the problem of the best fitting of two curves or two surfaces. There are two difficulties in such an approach. The first is related to the “global” interpolation from points to curves/surfaces, which requires the more-or-less arbitrary adoption of an interpolating method and the choice of an analytical (parameterized) description of the curve/surface complex enough to adequately describe its physical counterpart. The last requirement may be difficult to meet especially when rather complicated curves/surfaces are pointwise observed. The second difficulty relates to the choice of an adequate measure of the degree of fitting (or equivalently non-fitting) of two curves/surfaces and the a priori determination of the overlapping part which will enter in the fitting measure. One idea would be to minimize the area/volume between the two curves/surfaces under comparison; another one would be to minimize the integral of the square of the distances of each curve/surface point from its closest point on the corresponding curve/surface. In view of such problems, we will pursue here more feasible (though somewhat ad hoc) methods in two steps: an approximate alignment using principal component analysis and an iterative further improvement using localized interpolation and least squares best fitting methods.

In our three following examples two deal with surface matching (digital terrain models and laser scanning of surfaces) and one with curve matching (merging of co-seismic GNSS with accelerometer data).

There is a considerable literature in the field of pattern recognition and computer graphics for matching curves and surfaces in the presence of considerable deformation either real or due to different perspective views. These are too complicated for our case where the only difference between the two objects is due to observational noise. A similar problem is that of photogrammetric correlation where homologous points and lines are identified by detecting characteristic points and linear or nonlinear features. In our cases however, such characteristic points either do not exist, or they are so few and vague that they can be used only for pre-processing purposes to achieve a first-step approximate matching.

## 2. Approximate matching two point clouds on the same curve or surface using principal component analysis

Let us assume that we have two sets of coordinates $\mathbf{x}_i$, $i=1,2,...,n$, and $\mathbf{x}'_i$, $i=1,2,...,n'$, densely covering a curve or a surface, which refer to different reference systems. We seek to determine the transformation parameters $\boldsymbol{\theta}$ for rotation and $\mathbf{d}$ for translation, according to the coordinate transformation

(2.1) $$\mathbf{x}' = \mathbf{R}(\boldsymbol{\theta}) + \mathbf{d} ,$$

where $\mathbf{R}(\boldsymbol{\theta}) = \mathbf{R}_3(\theta_3)\mathbf{R}_2(\theta_2)\mathbf{R}_1(\theta_1)$ is the orthogonal matrix of rotations around the three axes with angles $\theta_1$, $\theta_2$, $\theta_3$, and $\mathbf{d} = [d_1\, d_2\, d_3]^T$ is the translation vector. Related to the shape of the curve or surface there are some intrinsic form-related axes (figure axes), which, when the observed point cloud is homogenously and sufficiently dense, they can be identified with the help of the dispersion matrix of the point coordinates. Identifying the coordinates of the point cloud barycentre as

$$\mathbf{m} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i \,, \tag{2.2}$$

the dispersion matrix is computed by

$$\mathbf{S} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T \,. \tag{2.3}$$

The (unit length) eigenvectors of $\mathbf{S}$ determine the three aforementioned figure axes. Thus if $\mathbf{S}\mathbf{u}_k = \lambda_k \mathbf{u}_k$, $k = 1,2,3$, where $\mathbf{u}_k$ are the eigenvectors and $\lambda_k$ the corresponding eigenvalues, it holds that $\mathbf{S}[\mathbf{u}_1\,\mathbf{u}_2\,\mathbf{u}_3] = \; = [\lambda_1\mathbf{u}_1\;\lambda_2\mathbf{u}_2\;\lambda_3\mathbf{u}_3]$ and we have the diagonalization

$$\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T, \quad \mathbf{U} = [\mathbf{u}_1\,\mathbf{u}_2\,\mathbf{u}_3], \quad \mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}. \tag{2.4}$$

It is possible then to apply a coordinate transformation $\tilde{\mathbf{x}}_i = \mathbf{U}^T(\mathbf{x}_i - \mathbf{m})$, which brings the origin of the reference system at the cloud barycentre and its three axes along the directions of the three figure axes. In a similar way, the second point cloud has barycentre $\mathbf{m}' = \frac{1}{n'}\sum_i \mathbf{x}'_i$ and dispersion matrix $\mathbf{S}' = \frac{1}{n'}\sum_i (\mathbf{x}'_i - \mathbf{m}')(\mathbf{x}'_i - \mathbf{m}')^T$ and its diagonaligation $\mathbf{S}' = \mathbf{U}'\mathbf{\Lambda}'\mathbf{U}'^T$ determines a transformation $\tilde{\mathbf{x}}'_i = \mathbf{U}'^T(\mathbf{x}'_i - \mathbf{m}')$, which brings the origin of the reference system at the barycentre and its three axes along the three figure axes. These barycenters and dispersion matrices have been found as discrete approximations of the surface barycentre $\boldsymbol{\mu} = \frac{1}{A}\int_S \mathbf{x}\, d\sigma$ and of the surface dispersion matrix $\mathbf{\Sigma} = \frac{1}{A}\int_S (\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T d\sigma$, where $A$ is the surface area. Therefore, we may assume that these approximate barycenters and the derived approximate directions of the surface figure axes coincide to a high degree of approximation with their true counterparts. (For a curve we only need to replace the surface element $d\sigma$ with the length element $ds$ and the area $A$ with the curve length $L$.)

Thus combining the transformation $\tilde{\mathbf{x}}_i = \mathbf{U}^T(\mathbf{x}_i - \mathbf{m})$ of the first cloud with the inverse transformation $\mathbf{x}'_i = \mathbf{U}'\tilde{\mathbf{x}}'_i + \mathbf{m}'$ of the second cloud we have a good approximation of the transformation from the first cloud to the second

(2.5) $$\mathbf{x}' = \mathbf{U}'\tilde{\mathbf{x}}' + \mathbf{m}' \approx \mathbf{U}'\tilde{\mathbf{x}} + \mathbf{m}' = \mathbf{U}'\mathbf{U}^T(\mathbf{x} - \mathbf{m}) + \mathbf{m}' = \mathbf{R}\mathbf{x} + \mathbf{d},$$

where

(2.6) $$\mathbf{R} = \mathbf{U}'\mathbf{U}^T, \quad \mathbf{d} = \mathbf{m}' - \mathbf{U}'\mathbf{U}^T\mathbf{m} = \mathbf{m}' - \mathbf{R}\mathbf{m}.$$

In the above we have assumed that the ordered eigenvalues $\lambda_1 > \lambda_2 > \lambda_3$ are all distinct and non-zero. We need therefore to examine the non-regular cases of zero eigenvalues and of equal eigenvalues (spectral multiplicity case). When only $\lambda_3 = 0$, then the cloud points lie on the same plane and the surface or curve are planar ones. When only $\lambda_2 = \lambda_3 = 0$, then the cloud points lie on the same line. This cannot happen for reasonable surfaces and in the case of a curve it means that it is in fact a straight line. Since $\lambda_2 = \lambda_3 (= 0)$ we have a case of spectral multiplicity, the consequences of which will be examined in the following. The case $\lambda_1 = \lambda_2 = \lambda_3 = 0$ should be excluded since it means that all cloud points are identical.

In the case of spectral multiplicity, e.g. $\lambda_1 = \lambda_2 \neq \lambda_3$ only the $\mathbf{u}_3$ axis is determined. Any pair of mutually perpendicular $\mathbf{u}_1$, $\mathbf{u}_2$ within the plane perpendicular to $\mathbf{u}_3$ is a set of eigenvectors. Thus, the principal components can only identify a common figure axis for the two coordinate sets but the direction of coincidence under all possible rotations around the determined axis must be found by other means, such as the use of characteristic points. The same holds for the case $\lambda_1 \neq \lambda_2 = \lambda_3$ where only the $\mathbf{u}_1$ direction can be identified. Finally, in the case $\lambda_1 = \lambda_2 = \lambda_3$ none of the three axes can be identified.

## 3. Matching Digital Terrain Models

An interesting problem is the merging of digital terrain models (DTMs, also called digital elevation models DEMs) referring to different horizontal and vertical reference systems into a unified one (see e.g. Biagi et al., 2011, 2012, 2014, Carcano, 2014). In the case of two digital terrain models, one of the axes, the vertical one, is already common, while the points are located on a rectangular grid as far as their horizontal coordinates are concerned. Letting the third Cartesian coordinate to be the vertical one the relation between the coordinates in the two DEM reference systems is of the form

(3.1) $$\tilde{\mathbf{x}} = \mathbf{R}_3(\Theta)\mathbf{x} + \mathbf{d}.$$

Once the two reference systems have been brought close to each other, e.g. by the principal components analysis described in the previous session, the coordinate transformation for small $\Theta$ and $\mathbf{d}$ can be replaced by the linear approximation

$$(3.2)\qquad \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{h} \end{bmatrix} = \begin{bmatrix} x + y\Theta + d_x \\ y - x\Theta + d_y \\ h + d_h \end{bmatrix},$$

where $x$, $y$ are the horizontal (east, north) and $h$ the vertical coordinate. We further assume that an interpolation scheme provides the heights $h'(x,y)$ in the second DTM as well as the height derivatives $a(x,y) = \frac{\partial h'}{\partial x}(x,y)$ and $b(x,y) = \frac{\partial h'}{\partial y}(x,y)$. Each grid point $(i,k)$ of the first DTM has coordinates $x_{ik}$, $y_{ik}$, $h_{ik}$ and transformed coordinates $\tilde{x}_{ik}$, $\tilde{y}_{ik}$, $\tilde{h}_{ik}$. In order to find the transformation parameters $\Theta$, $d_x$, $d_y$, $d_h$ of the two surfaces we will minimize the sum of the squares between the heights in the first DTM and the interpolated heights in the second DTM

$$(3.3)\qquad \phi = \sum_{ik} e_{ik}^2 = \sum_{ik} \left[ h'(\tilde{x}_{ik}, \tilde{y}_{ik}) - \tilde{h}_{ik} \right]^2 = \min .$$

In first order linear approximation

$$(3.4)\qquad \begin{aligned} e_{ik} &= h'(\tilde{x}_{ik}, \tilde{y}_{ik}) - \tilde{h}_{ik} \approx \\ &\approx h'(x_{ik}, y_{ik}) + \left( \frac{\partial h}{\partial x} \right)_{x_{ik}, y_{ik}} (\tilde{x}_{ik} - x_{ik}) + \left( \frac{\partial h}{\partial y} \right)_{x_{ik}, y_{ik}} (\tilde{y}_{ik} - y_{ik}) - h_{ik} - d_h \equiv \\ &\equiv h'_{ik} + a_{ik}(\Theta y_{ik} + d_x) + b_{ik}(-\Theta x_{ik} + d_y) - h_{ik} - d_h = \\ &= h'_{ik} - h_{ik} + \left[ a_{ik} y_{ik} - b_{ik} x_{ik} \quad a_{ik} \quad b_{ik} \quad -1 \right] \begin{bmatrix} \Theta \\ d_x \\ d_y \\ d_h \end{bmatrix} \equiv \Delta h'_{ik} - \mathbf{q}_{ik}^T \mathbf{z}, \end{aligned}$$

where $\mathbf{z} = [\Theta\, d_x\, d_y\, d_h]^T$, $h'_{ik} = h'(x_{ik}, y_{ik})$, $a_{ik} = \frac{\partial h'}{\partial x}(x_{ik}, y_{ik})$, $b_{ik} = \frac{\partial h'}{\partial y}(x_{ik}, y_{ik})$, and $\Delta h'_{ik} = h'(x_{ik}, y_{ik}) - h_{ik}$. The target function $\phi = \sum_{i,k} (\Delta h'_{ik} - \mathbf{q}_{ik}^T \mathbf{z})^2$ is minimized by setting its derivative with respect to $\mathbf{z}$ equal to zero

$$(3.5)\qquad \frac{\partial \phi}{\partial \mathbf{z}} = \sum_{i,k} 2(\Delta h'_{ik} - \mathbf{q}_{ik}^T \mathbf{z}) \frac{\partial(\Delta h'_{ik} - \mathbf{q}_{ik}^T \mathbf{z})}{\partial \mathbf{z}} = 2\sum_{i,k} (\Delta h'_{ik} - \mathbf{q}_{ik}^T \mathbf{z})(-\mathbf{q}_{ik}^T) = \mathbf{0},$$

which has solution $\mathbf{z} = (\sum_{i,k} \mathbf{q}_{ik} \mathbf{q}_{ik}^T)^{-1} (\sum_{i,k} \Delta h'_{ik} \mathbf{q}_{ik})$ or explicitly

$$(3.6)\qquad \begin{bmatrix} \Theta \\ d_x \\ d_y \\ d_h \end{bmatrix} =$$

$$= \left( \sum_{i,k} \begin{bmatrix} (a_{ik} y_{ik} - b_{ik} x_{ik})^2 & a_{ik}(a_{ik} y_{ik} - b_{ik} x_{ik}) & b_{ik}(a_{ik} y_{ik} - b_{ik} x_{ik}) & -(a_{ik} y_{ik} - b_{ik} x_{ik}) \\ a_{ik}(a_{ik} y_{ik} - b_{ik} x_{ik}) & a_{ik}^2 & a_{ik} b_{ik} & -a_{ik} \\ b_{ik}(a_{ik} y_{ik} - b_{ik} x_{ik}) & a_{ik} b_{ik} & b_{ik}^2 & -b_{ik} \\ -(a_{ik} y_{ik} - b_{ik} x_{ik}) & -a_{ik} & -b_{ik} & 1 \end{bmatrix} \right)^{-1} \times$$

$$\times \left( \sum_{i,k} \begin{bmatrix} (a_{ik} y_{ik} - b_{ik} x_{ik})[h'(x_{ik}, y_{ik}) - h_{ik}] \\ a_{ik}[h'(x_{ik}, y_{ik}) - h_{ik}] \\ b_{ik}[h'(x_{ik}, y_{ik}) - h_{ik}] \\ -[h'(x_{ik}, y_{ik}) - h_{ik}] \end{bmatrix} \right).$$

The obtained values are used to transform the coordinates according to (3.2). Then the transformed coordinates can be used as input coordinates for the next iteration step until convergence (zero transformation parameters) is achieved.

As a convenient localized interpolation scheme, we propose the fitting to the $n^2$ closest point heights $h_i = h(x_i, y_i)$, i.e. those on the closest $n \times n$ sub-grid, the polynomial model with $n^2$ coefficients

$$(3.7)\qquad h(x,y) = \boldsymbol{\varphi}(x)^T \mathbf{A} \boldsymbol{\varphi}(y) = [\boldsymbol{\varphi}(y)^T \otimes \boldsymbol{\varphi}(x)^T] vec\mathbf{A},$$

where

$$(3.8)\qquad \boldsymbol{\varphi}(x) = \begin{bmatrix} 1 & x & x^2 & \cdots & x^{n-1} \end{bmatrix}^T, \quad \boldsymbol{\varphi}(y) = \begin{bmatrix} 1 & y & y^2 & \cdots & y^{n-1} \end{bmatrix}^T.$$

From the known heights at the sub-grid points $h_i = h(x_i, y_i)$, $i = 1,2,...,n^2$, the coefficient matrix is determined according to

$$(3.9)\qquad vec\mathbf{A} = \begin{bmatrix} \boldsymbol{\varphi}(y_1)^T \otimes \boldsymbol{\varphi}(x_1)^T \\ \vdots \\ \boldsymbol{\varphi}(y_{n^2})^T \otimes \boldsymbol{\varphi}(x_{n^2})^T \end{bmatrix}^{-1} \begin{bmatrix} h_1 \\ \vdots \\ h_{n^2} \end{bmatrix}.$$

Using as origin the center of the $n \times n$ closest point grid and the grid spacings $\Delta_x$, $\Delta_y$ as length units, it is more convenient to interpolate instead the function $\chi(\xi,\eta) = h(\Delta_x x + x_0, \Delta_y y + y_0)$. In this case, the coefficient matrix to be inverted in (3.9) has elements with integer values. The desired interpolated values mat be computed from the inverse relation $h(x,y)=\chi\left(\frac{\xi-x_0}{\Delta_x},\frac{\eta-y_0}{\Delta_y}\right)=\boldsymbol{\varphi}(\frac{\xi-x_0}{\Delta_x})^T\mathbf{A}\boldsymbol{\varphi}(\frac{\eta-y_0}{\Delta_y})$ .

## 4. Matching two point clouds obtained by laser scanning

Repeated laser scanning produce coordinates $\mathbf{x}_i$, $i=1,2,...,n$, and $\mathbf{x}'_i$, $i=1,2,...,n'$, related to two different point clouds on the same surface. The two coordinate sets refer to different reference systems, which are related by the general transformation relation (3.1). Apart from the fact that all rotation angles $\boldsymbol{\theta}=[\theta_1\,\theta_2\,\theta_3]^T$ are now present, there are two more differences with respect to the previous case of DTM matching. The first is that the cloud points are not located on some grid and the second that there is no preferable matching direction as is the vertical direction in the DTM case. The lack of gridded data calls for a different localized interpolation technique. The lack of a single reference direction calls for a three-dimensional best fitting principle. We will sketch here a possible way of attacking the problem, assuming that the two reference systems have already been aligned to a good approximation either by principal component analysis or with the use of characteristic points. For small values of the transformation parameters the coordinate transformation can be described with sufficient accuracy by

(4.1) $$\tilde{\mathbf{x}}_i=\mathbf{R}(\boldsymbol{\theta})\mathbf{x}_i+\mathbf{d}\approx(\mathbf{I}-[\boldsymbol{\theta}\times])\mathbf{x}_i+\mathbf{d}=\mathbf{x}_i+[\mathbf{x}_i\times]\boldsymbol{\theta}+\mathbf{d},$$

or explicitly

(4.2) $$\begin{bmatrix}\tilde{x}_i\\ \tilde{y}_i\\ \tilde{z}_i\end{bmatrix}=\begin{bmatrix}x_i\\ y_i\\ z_i\end{bmatrix}+\begin{bmatrix}0 & -z_i & y_i\\ z_i & 0 & -x_i\\ -y_i & x_i & 0\end{bmatrix}\begin{bmatrix}\theta_x\\ \theta_y\\ \theta_z\end{bmatrix}+\begin{bmatrix}d_x\\ d_y\\ d_z\end{bmatrix}=\begin{bmatrix}x_i-z_i\theta_y+y_i\theta_z+d_x\\ y_i+z_i\theta_x-x_i\theta_z+d_y\\ z_i-y_i\theta_x+x_i\theta_y+d_z\end{bmatrix}.$$

We denote by $[\mathbf{a}\times]=\mathbf{A}$, the $3\times3$ antisymmetric matrix with axial vector $\mathbf{a}$, uniquely defined by $A_{32}=-A_{23}=a_1$, $A_{13}=-A_{31}=a_2$, $A_{21}=-A_{12}=a_3$ and $A_{11}=A_{22}=A_{33}=0$.

Thus to each point $\mathbf{x}_i$ corresponds a transformed point $\tilde{\mathbf{x}}_i$, which will be matched with the surface which is created by the interpolation of the points $\mathbf{x}'_k$ of the second cloud. Let us assume that a localized interpolation has been achieved based on a set of points $\{\mathbf{x}'_k\,|\,k\in K_i\}$ where $K_i$ is the set of the indices of say the $m$ closest points to $\mathbf{x}_i$ among all points $\mathbf{x}'_k$ of the second cloud. One convenient form of interpolation is $z=z(x,y)$ where $z$ is one of the coordinates and $x$, $y$ the other two. We have taken here $z$ to be the third coordinate, without loss of generality, but this may require the use of a new appropriate working reference system. The coordinates $x$, $y$ serve in this case as intrinsic surface coordinates and the surface is described by $\mathbf{x} = \mathbf{x}(x, y) = \begin{bmatrix} x & y & z(x, y) \end{bmatrix}^T$. For the matching criterion, we need to find the surface point

(4.3) $$\bar{\mathbf{x}}_i = \mathbf{x}(\bar{x}_i, \bar{y}_i) = \begin{bmatrix} \bar{x}_i & \bar{y}_i & z(\bar{x}_i, \bar{y}_i) \end{bmatrix}^T,$$

which is closest to the transformed point $\tilde{\mathbf{x}}_i$. Then as target function for optimal matching we may use

(4.4) $$\phi = \sum_i (\bar{\mathbf{x}}_i - \tilde{\mathbf{x}}_i)^T (\bar{\mathbf{x}}_i - \tilde{\mathbf{x}}_i) = \sum_i \mathbf{e}_i^T \mathbf{e}_i = \min .$$

where $\mathbf{e}_i = \bar{\mathbf{x}}_i - \mathbf{x}_i + [\mathbf{x}_i \times]\boldsymbol{\theta} - \mathbf{d}$. Therefore, we have a typical least-squares problem with observation equations

(4.5) $$\mathbf{x}_i = \bar{\mathbf{x}}_i + [\mathbf{x}_i \times]\boldsymbol{\theta} - \mathbf{d} - \mathbf{e}_i = \begin{bmatrix} \bar{x}_i \\ \bar{y}_i \\ z(\bar{x}_i, \bar{y}_i) \end{bmatrix} + [\mathbf{x}_i \times]\boldsymbol{\theta} - \mathbf{d} - \mathbf{e}_i, \quad i = 1, 2, \ldots, n .$$

The point $\bar{\mathbf{x}}_i$ is the closest to $\tilde{\mathbf{x}}_i$ when the vector $\bar{\mathbf{x}}_i - \tilde{\mathbf{x}}_i$ is perpendicular to the surface at the point $\tilde{\mathbf{x}}_i'$. To express this mathematically we will use the two tangent vectors at $\bar{\mathbf{x}}_i$, which are tangent to the coordinate lines, namely

(4.6) $$\mathbf{t}_x = \frac{\partial \mathbf{x}}{\partial x}(\bar{x}_i, \bar{y}_i) = \begin{bmatrix} 1 \\ 0 \\ \frac{\partial z}{\partial x}(\bar{x}_i, \bar{y}_i) \end{bmatrix}, \quad \mathbf{t}_y = \frac{\partial \mathbf{x}}{\partial y}(\bar{x}_i, \bar{y}_i) = \begin{bmatrix} 0 \\ 1 \\ \frac{\partial z}{\partial y}(\bar{x}_i, \bar{y}_i) \end{bmatrix}.$$

The orthogonality conditions $\mathbf{t}_x^T(\bar{\mathbf{x}}_i - \tilde{\mathbf{x}}_i) = 0$ and $\mathbf{t}_y^T(\bar{\mathbf{x}}_i - \tilde{\mathbf{x}}_i) = 0$ give rise to the constraints

(4.7) $$(\bar{x}_i - \tilde{x}_i) + \frac{\partial z}{\partial x}(\bar{x}_i, \bar{y}_i)[z(\bar{x}_i, \bar{y}_i) - \tilde{z}_i] = 0,$$

$$(\bar{y}_i - \tilde{y}_i) + \frac{\partial z}{\partial y}(\bar{x}_i, \bar{y}_i)[z(\bar{x}_i, \bar{y}_i) - \tilde{z}_i] = 0, \quad i = 1, 2, \ldots, n,$$

which counterbalance the introduction of the additional unknowns $\bar{x}_i$, $\bar{y}_i$. Before we proceed, we need to linearize both the observation equations (4.5) and the constraints (4.7) with respect to the unknown parameters $\bar{x}_i$, $\bar{y}_i$. In linear approximation we have by Taylor expansions

$$(4.8a)\qquad z(\bar{x}_i,\bar{y}_i)=z(x_i,y_i)+\left(\frac{\partial z}{\partial x}\right)_{(x_i,y_i)}(\bar{x}_i-x_i)+\left(\frac{\partial z}{\partial y}\right)_{(x_i,y_i)}(\bar{y}_i-y_i)=$$

$$z(x_i,y_i)+a_{xi}(\bar{x}_i-x_i)+a_{yi}(\bar{y}_i-y_i),$$

$$(4.8b)\qquad \frac{\partial z}{\partial x}(\bar{x}_i,\bar{y}_i)=\frac{\partial z}{\partial x}(x_i,y_i)+\left(\frac{\partial^2 z}{\partial x^2}\right)_{(x_i,y_i)}(\bar{x}_i-x_i)+\left(\frac{\partial^2 z}{\partial y\partial x}\right)_{(x_i,y_i)}(\bar{y}_i-y_i)=$$

$$a_{xi}+b_{xxi}(\bar{x}_i-x_i)+b_{xyi}(\bar{y}_i-y_i),$$

$$(4.8c)\qquad \frac{\partial z}{\partial y}(\bar{x}_i,\bar{y}_i)=\frac{\partial z}{\partial y}(x_i,y_i)+\left(\frac{\partial^2 z}{\partial x\partial y}\right)_{(x_i,y_i)}(\bar{x}_i-x_i)+\left(\frac{\partial^2 z}{\partial y^2}\right)_{(x_i,y_i)}(\bar{y}_i-y_i)=$$

$$a_{yi}+b_{xyi}(\bar{x}_i-x_i)+b_{yyi}(\bar{y}_i-y_i),$$

where

$$(4.9)\qquad a_{xi}=\frac{\partial z}{\partial x}(x_i,y_i),\qquad a_{yi}=\frac{\partial z}{\partial y}(x_i,y_i)a_{yi},$$

while the second order derivatives $b_{xxi}=\dfrac{\partial^2 z}{\partial x^2}(x_i,y_i)$, $b_{xyi}=\dfrac{\partial^2 z}{\partial y\partial x}(x_i,y_i)$, $b_{yyi}=\dfrac{\partial^2 z}{\partial y^2}(x_i,y_i)$, will cancel out in the linear approximation. Replacing the above values, as well as $\tilde{x}_i=x_i-z_i\theta_y+y_i\theta_z+d_x$, $\tilde{y}_i=y_i+z_i\theta_x-x_i\theta_z+d_y$ from (4.2) and deleting second order terms we obtain the linearized observations equations

$$(4.10)\qquad \mathbf{x}_i=\mathbf{h}_i+\mathbf{G}_i\mathbf{p}_i-\begin{bmatrix}[\mathbf{x}_i\times] & \mathbf{I}\end{bmatrix}\begin{bmatrix}\boldsymbol{\theta}\\ \mathbf{d}\end{bmatrix}-\mathbf{e}_i,\quad i=1,2,\ldots,n,$$

and the linearized constraints

$$(4.11)\qquad \mathbf{g}_i+\mathbf{Q}_i\mathbf{p}_i+\mathbf{B}_i\begin{bmatrix}\boldsymbol{\theta}\\ \mathbf{d}\end{bmatrix}=\mathbf{0},\quad i=1,2,\ldots,n,$$

where

$$(4.12a)\quad \mathbf{h}_i=\begin{bmatrix}0\\0\\z(x_i,y_i)-a_{xi}x_i-a_{yi}y_i\end{bmatrix},\quad \mathbf{p}_i=\begin{bmatrix}\bar{x}_i\\ \bar{y}_i\end{bmatrix},\quad \mathbf{g}_i=\begin{bmatrix}a_{xi}\Delta z_i-(1+a_{xi}^2)x_i-a_{xi}a_{yi}y_i\\ a_{yi}\Delta z_i-a_{xi}a_{yi}x_i-(1+a_{yi}^2)y_i\end{bmatrix},$$

(4.12b) $$\mathbf{G}_i = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ a_{xi} & a_{yi} \end{bmatrix}, \quad \mathbf{Q}_i = \begin{bmatrix} 1+a_{xi}^2 & a_{xi}a_{yi} \\ a_{xi}a_{yi} & 1+a_{yi}^2 \end{bmatrix},$$

(4.12c) $$\mathbf{B}_i = \begin{bmatrix} a_{xi}y_i & z_i - a_{xi}x_i & -y_i & -1 & 0 & -a_{xi} \\ a_{yi}y_i - z_i & -a_{yi}x_i & x_i & 0 & -1 & -a_{yi} \end{bmatrix}.$$

Noting that

(4.13) $$D_i = \det \mathbf{Q}_i = 1 + a_{xi}^2 + a_{yi}^2 \neq 0\,,$$

we may utilize the inverse

(4.14) $$\mathbf{Q}_i^{-1} = \frac{1}{D_i} \begin{bmatrix} 1+a_{yi}^2 & -a_{xi}a_{yi} \\ -a_{xi}a_{yi} & 1+a_{xi}^2 \end{bmatrix},$$

in order to eliminate the unknowns $\mathbf{p}_i = [\bar{x}_i \ \bar{y}_i]^T$. Indeed, from (4.11) $\mathbf{p}_i = -\mathbf{Q}_i^{-1}\mathbf{g}_i - \mathbf{Q}_i^{-1}\mathbf{B}_i \begin{bmatrix} \boldsymbol{\theta} \\ \mathbf{d} \end{bmatrix}$, which replaced in (4.10) provides the reduced observation equations

(4.15) $$\mathbf{b}_i \equiv \mathbf{x}_i - \mathbf{h}_i + \mathbf{G}_i\mathbf{Q}_i^{-1}\mathbf{g}_i = -\left(\mathbf{G}_i\mathbf{Q}_i^{-1}\mathbf{B}_i - \big[[\mathbf{x}_i\times] \quad \mathbf{I}\big]\right)\begin{bmatrix} \boldsymbol{\theta} \\ \mathbf{d} \end{bmatrix} - \mathbf{e}_i \equiv \mathbf{A}_i \begin{bmatrix} \boldsymbol{\theta} \\ \mathbf{d} \end{bmatrix} - \mathbf{e}_i\,.$$

After carrying out all the necessary tedious calculations and after some astonishing term eliminations we find that

(4.16) $$\mathbf{b}_i \equiv \mathbf{x}_i - \mathbf{h}_i + \mathbf{G}_i\mathbf{Q}_i^{-1}\mathbf{g}_i = \frac{z(x_i, y_i) - z_i}{1 + a_{xi}^2 + a_{yi}^2} \begin{bmatrix} a_{xi} \\ a_{yi} \\ -1 \end{bmatrix},$$

(4.17) $$\mathbf{A}_i = -\left(\mathbf{G}_i\mathbf{Q}_i^{-1}\mathbf{B}_i - \big[[\mathbf{x}_i\times] \quad \mathbf{I}\big]\right) =$$

$$= -\frac{1}{1 + a_{xi}^2 + a_{yi}^2} \begin{bmatrix} a_{xi} \\ a_{yi} \\ -1 \end{bmatrix} \begin{bmatrix} (y_i + a_{yi}z_i) & -(x_i + a_{xi}z_i) & -(a_{yi}x_i - a_{xi}y_i) & a_{xi} & a_{yi} & -1 \end{bmatrix}.$$

The estimates of the unknown transformation parameters are found from the solution of the normal equations

(4.18) $$\begin{bmatrix} \hat{\boldsymbol{\theta}} \\ \hat{\mathbf{d}} \end{bmatrix} = \left(\sum_{i=1}^{n} \mathbf{A}_i^T \mathbf{A}_i\right)^{-1} \left(\sum_{i=1}^{n} \mathbf{A}_i^T \mathbf{b}_i\right) = \left(\sum_{i=1}^{n} \frac{1}{1 + a_{xi}^2 + a_{yi}^2} \mathbf{d}_i \mathbf{d}_i^T\right)^{-1} \left(\sum_{i=1}^{n} \frac{z_i - z(x_i, y_i)}{1 + a_{xi}^2 + a_{yi}^2} \mathbf{d}_i\right),$$

where

(4.19) $$\mathbf{d}_i = \left[ (y_i + a_{yi} z_i) \quad -(x_i + a_{xi} z_i) \quad -(a_{yi} x_i - a_{xi} y_i) \quad a_{xi} \quad a_{yi} \quad -1 \right]^T .$$

The obtained values are used to transform the coordinates according to (4.2). Then the transformed coordinates can be used as input coordinates for the next iteration step until convergence (zero transformation parameters) is achieved.

A similar localized interpolation scheme as in the DTM case can be implemented, utilizing the $m$ closest points in the $x, y$ sense. In this case, the simplifications arising from the fact that the DTM data are gridded do not apply any more.

## 5. Integrating co-seismic GNSS velocities with accelerometer data

Continuous registration of GNSS data during a seismic event allows for baseline-like solutions between two consecutive epochs, thus providing displacements (Colosimo, 2012, Colosimo et al, 2011). For the standard case of observations at equal time intervals, these displacements are equivalent to velocities and the GNSS receiver acts as a velocimeter. Despite the cancelation of tropospheric and ionospheric errors, the computed velocities are rather noisy, due to the direct propagation of remaining noise in single epoch measurements instead of the usual compound ones based on averaging over a small time interval (usually 30 seconds) in static GNSS where the receiver is not subject to motion. The noise level is much larger than those in velocities computed by integrating accelerometer data, even those from very low cost accelerometers (Benedetti, 2015), which observe at much higher frequencies. On the other hand, low cost accelerometers, which are easy to collocate with GNSS receivers, are suffering from instabilities and must be calibrated for nonlinear trend removal. The two velocity records, visualized as two time dependent curves, refer to different reference and time systems. The problem of time synchronization can be solved a priori using characteristic points, such as zero velocity crossings. Application of the principal components method described in section 2, showed errors in reference system orientation of the order of 1-5 degrees, which have insignificant effects on the transformed velocity component time series of the velocity curve (Benedetti, 2015). Here we will briefly outline how to achieve an even better accuracy by localized interpolation and least squares matching techniques. The input data are two velocity component time series $\mathbf{v}_i^G = \mathbf{v}^G(t_i^G)$, $i = 1, 2, ..., n_G$, from the GNSS receiver and $\mathbf{v}_i^A = \mathbf{v}^A(t_i^A)$, $i = 1, 2, ..., n_A$, from the accelerometer. We need to transform one set, say the accelerometer one, in a way that it assumes the GNSS reference system according to

(5.1) $$\tilde{\mathbf{v}}_i^A = (1+\lambda)\mathbf{R}(\boldsymbol{\theta})\mathbf{v}_i^A + \mathbf{d} \approx \mathbf{v}_i^A + \lambda \mathbf{v}_i^A + [\mathbf{v}_i^A \times]\boldsymbol{\theta} + \mathbf{d} .$$

A scale factor of $1+\lambda$ has been included to account for the possible use of different velocity units in GNSS receivers and accelerometers. Since in theory velocities are affected by only a rotation of the reference system, $\mathbf{d}$ and $\lambda$ should be viewed as additional calibration parameters. For the GNSS data, we assume that a localized interpolation scheme provides the function $\mathbf{v}^G(t)$ at each neighbourhood of the $t_i^A$ observation epochs. An optimal least squares fitting is obtained by minimizing

$$(5.2)\qquad \sum_{i=1}^{n_A}\mathbf{e}_i^T\mathbf{P}_i\mathbf{e}_i=\sum_{i=1}^{n_A}[\mathbf{v}^G(t_i^A)-\tilde{\mathbf{v}}_i^A]^T\mathbf{P}_i[\mathbf{v}^G(t_i^A)-\tilde{\mathbf{v}}_i^A]=$$

$$=\sum_{i=1}^{n_A}[\mathbf{v}^G(t_i^A)-\mathbf{v}_i^A-\lambda\mathbf{v}_i^A-[\mathbf{v}_i^A\times]\boldsymbol{\theta}-\mathbf{d}]^T\mathbf{P}_i[\mathbf{v}^G(t_i^A)-\mathbf{v}_i^A-\lambda\mathbf{v}_i^A-[\mathbf{v}_i^A\times]\boldsymbol{\theta}-\mathbf{d}]=\min .$$

This is a typical least squares problem with observation equations

$$(5.3)\quad \mathbf{b}_i\equiv\mathbf{v}^G(t_i^A)-\mathbf{v}_i^A=\lambda\mathbf{v}_i^A+[\mathbf{v}_i^A\times]\boldsymbol{\theta}+\mathbf{d}+\mathbf{e}_i=\begin{bmatrix}[\mathbf{v}_i^A\times] & \mathbf{I} & \mathbf{v}_i^A\end{bmatrix}\begin{bmatrix}\boldsymbol{\theta}\\ \mathbf{d}\\ \lambda\end{bmatrix}+\mathbf{e}_i\equiv\mathbf{E}_i\mathbf{x}+\mathbf{e}_i .$$

The parameter estimates $\hat{\boldsymbol{\theta}}$, $\hat{\mathbf{d}}$, $\hat{\lambda}$ are provided by the solution of the normal equations

$$(5.4)\qquad \hat{\mathbf{x}}=\begin{bmatrix}\hat{\boldsymbol{\theta}}\\ \hat{\mathbf{d}}\\ \hat{\lambda}\end{bmatrix}=\left(\sum_{i=1}^{n_A}\mathbf{E}_i^T\mathbf{P}_i\mathbf{E}_i\right)^{-1}\left(\sum_{i=1}^{n_A}\mathbf{E}_i^T\mathbf{P}_i\mathbf{b}_i\right).$$

For simple least squares ($\mathbf{P}_i=\mathbf{I}$) the solution simplifies to

$$(5.4)\qquad \hat{\mathbf{x}}=\begin{bmatrix}\hat{\boldsymbol{\theta}}\\ \hat{\mathbf{d}}\\ \hat{\lambda}\end{bmatrix}=\begin{bmatrix}\mathbf{C}_{0A} & -[\mathbf{m}_A\times] & \mathbf{0}\\ [\mathbf{m}_A\times] & \mathbf{I} & \mathbf{m}_A\\ \mathbf{0} & \mathbf{m}_A^T & s_{0A}^2\end{bmatrix}^{-1}\begin{bmatrix}-\mathbf{h}_0\\ \mathbf{m}_G-\mathbf{m}_A\\ s_{0AG}-s_{0A}^2\end{bmatrix},$$

where

$$(5.5a)\quad \mathbf{C}_{0A}=-\frac{1}{n_A}\sum_{i=1}^{n_A}[\mathbf{v}_i^A\times]^2,\qquad \mathbf{m}_A=\frac{1}{n_A}\sum_{i=1}^{n_A}\mathbf{v}_i^A,\qquad s_{0A}^2=\frac{1}{n_A}\sum_{i=1}^{n_A}(\mathbf{v}_i^A)^T\mathbf{v}_i^A,$$

$$(5.5b)\quad \mathbf{h}_0=\frac{1}{n_A}\sum_{i=1}^{n_A}[\mathbf{v}_i^A\times]\mathbf{v}^G(t_i^A),\quad \mathbf{m}_G=\frac{1}{n_A}\sum_{i=1}^{n_A}\mathbf{v}^G(t_i^A),\quad s_{0AG}=\frac{1}{n_A}\sum_{i=1}^{n_A}(\mathbf{v}_i^A)^T\mathbf{v}^G(t_i^A).$$

Analytical inversion leads to the explicit solution

(5.6) $$\hat{\boldsymbol{\theta}} = -\mathbf{C}_A^{-1}\mathbf{h}\,, \qquad \hat{\lambda} = \frac{s_{AG}}{s_A^2} - 1\,, \qquad \hat{\mathbf{d}} = -[\mathbf{m}_A \times]\hat{\boldsymbol{\theta}} + \mathbf{m}_G - (1+\hat{\lambda})\mathbf{m}_A\,,$$

where

(5.7a) $$\mathbf{C}_A = -\frac{1}{n_A}\sum_{i=1}^{n_A}[(\mathbf{v}_i^A - \mathbf{m}_A)\times]^2\,, \qquad \mathbf{h} = \frac{1}{n_A}\sum_{i=1}^{n_A}[(\mathbf{v}_i^A - \mathbf{m}_A)\times][\mathbf{v}^G(t_i^A) - \mathbf{m}_G]\,,$$

(5.7b) $$s_A^2 = \frac{1}{n_A}\sum_{i=1}^{n_A}(\mathbf{v}_i^A - \mathbf{m}_A)^T(\mathbf{v}_i^A - \mathbf{m}_A)\,, \qquad s_{AG} = \frac{1}{n_A}\sum_{i=1}^{n_A}(\mathbf{v}_i^A - \mathbf{m}_A)^T[\mathbf{v}^G(t_i^A) - \mathbf{m}_G]\,.$$

The obtained values are used to transform the velocities according to (5.1). Then the transformed velocities can be used as input velocities for the next iteration step until convergence (zero transformation parameters) is achieved.

Localized interpolation is rather simple in this case, utilizing the $2n+1$ closer epoch data $\mathbf{v}^G(t_{k-n}^G)$ … $\mathbf{v}^G(t_k^G)$ … $\mathbf{v}^G(t_{k+n}^G)$, where $t_k^G$ is the epoch closest to the relevant accelerometer epoch $t_i^A$. The polynomial interpolation for each velocity component $v_k^G$, $k = 1,2,3$, has the form

(5.8) $$v_k^G(t) = a_{k,0} + a_{k,1}t + a_{k,2}t^2 + \ldots + a_{k,2n+1}t^{2n}\,.$$

The coefficients are computed from

(5.9) $$\begin{bmatrix} a_{k,0} \\ a_{k,1} \\ a_{k,2} \\ \vdots \\ a_{k,2n+1} \end{bmatrix} = \begin{bmatrix} 1 & t_{k-n}^G & (t_{k-n}^G)^2 & \cdots & (t_{k-n}^G)^{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & t_k^G & (t_k^G)^2 & \cdots & (t_k^G)^{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & t_{k+n}^G & (t_{k+n}^G)^2 & \cdots & (t_{k+n}^G)^{2n} \end{bmatrix}^{-1} \begin{bmatrix} v_k^G(t_{k-n}^G) \\ \vdots \\ v_k^G(t_k^G) \\ \vdots \\ v_k^G(t_{k+n}^G) \end{bmatrix}.$$

## 6. Conclusions

We have presented algorithms for relating the different reference systems in two point clouds for three completely different applications, which however share a common characteristic, the fact that both points sets lie on the same geometric figure, a surface or a curve. The common shape of this figure provides the means for realizing the required matching. All three algorithms rely on the same strategy: initial approximate matching and iterative improvement based on linearization, localized interpolation and application of the least squares matching principle. The differences in the algorithms emerge from the particularities of each application. In the DTM application, the vertical direction is already common and this restricts the transformation parameters from one reference system to the other. In both the

DTM and the laser scanning applications no scale parameter is used in the coordinate transformation, because it is reasonable to assume that the same unit of length is used in both instances, or that any very small difference does not affect the result in view of the relatively low accuracy of DTMs or the small distances in laser scanning. On the contrary, a scale parameter has been introduced for the GNSS – accelerometer application to account for differences in the unit of length, especially in the case of low cost accelerometers.

The point correspondence required for the application of the least squares matching principle is realized by matching each point of one set to a point of the locally interpolated surface or curve of the second point set. The correspondence is established along the common vertical direction in the DTM case and through time synchronization in the GNSS – accelerometer case, where we have assumed that the problem of different time system (in origin and scale) has been a priori resolved in an independent manner. Of course, it is rather easy to generalize the algorithm to a simultaneous determination of both reference and time system transformation parameters. In the laser scanning case, the correspondence is realized by assigning to each point of one set the closest point of the locally interpolated surface for the second set. This complicates matters slightly by introducing as additional unknowns two surface coordinates for each interpolated closest point. Fortunately, in the linearized case these unknowns can be readily eliminated without need to resort to least squares with constraints.

As possible area for further research we suggest the modification of the algorithms using nonlinear least squares without resorting to linearization. These will lead to a set of nonlinear normal equations that must be iteratively solved. The difference with the usual nonlinear least squares is the fact that the localized interpolation needs to be also updated in each iteration step.

## References


Benedetti, E. (2015): New strategies for structures and ground monitoring in real-time based on GNSS and MEMS accelerometers integration. PhD Thesis, Faculty of Engineering, Department of Civil, Building and Environmental Engineering, Geodesy and Geomatics Area, University of Rome "Sapienza".

Biagi L., Brovelli M.A., Campi A., Cannata M., Carcano L., Credali M., De Agostino M., Manzino A., Sansò F., Siletto G. (2011). Il progetto HELI-DEM (Helvetia-Italy Digital Elevation Model): scopi e stato di attuazione. Bollettino della Società Italiana di Fotogrammetria e Topografia, no1/2011, pp. 35-51.

Biagi L., L. Carcano, M. De Agostin (2012: DTM cross validation and merging: Problems and solutions for a case study within the HELI-DEM project. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XXXIX-B4, 2012, XXII ISPRS Congress, 25 August – 01 September 2012, Melbourne, Australia.

Biagi L., Caldera S., Carcano L., Lucchese A., Negretti M., Sansò F., Triglione D., Visconti M.G. (2014): Il progetto HELI-DEM (Helvetia-Italy Digital Elevation Model). Bollettino SIFET, Numero Speziale 13, Societa Italiana di Fotofgrammetria e Topografia.

Carcano L. (2014): Merging local DTMS: HELI-DEM project, problems and solutions. PhD Thesis, Politecnico di Milano.

Colosimo G. (2012): VADASE: Variometric Approach for Displacement Analysis Stand-alone Engine. PhD Thesis, Faculty of Engineering, Department of Civil, Building and Environmental Engineering, Geodesy and Geomatics Area, University of Rome "Sapienza".

Colosimo G., Crespi M., Manzzoni A. (2011): Real-time GPS seismology with a stand-alone receiver: A preliminary feasibility demonstration. Journal of Geophysical Research: Solid Earth, 116, B11302.