UNITED STATES DEPARTMENT OF THE INTERIOR

GEOLOGICAL SURVEY

Notes on a Method to Transform Digitized Coordinates to Geographic Coordinates

by

Gerald I. Evenden[1]

Open-File Report 91–17

This report is preliminary and has not been reviewed for conformity with U.S. Geological Survey editorial standards. Use of tradenames is for purposes of identification only and does not constitute endorsement by the U.S. Geological Survey.

January 7, 1991

[1]Woods Hole, MA

Contents

[illegible]

Notes on a Method to Transform Digitized Coordinates to Geographic Coordinates

Gerald I. Evenden

January 7, 1991

Abstract

The process of conversion of coordinates (typically cartesian) obtained from digitizing devices into application defined coordinate systems is discussed. Application of a secondary function representing the transformation of the application coordinate system to and from cartesian space will greatly simplify conversion of highly non-linear systems. Although emphasis is given to cartographic conversions the method will work with a wide variety of digitizing problems.

Introduction

In order to build digital data bases from archival geographic information in the form of chart or map graphics it is necessary to perform a process termed digitization. Although this process can be manually performed by scaling and recording coordinates directly from the map, the volume of data generally requires the use of automated equipment such as tablet or scanning digitizers. In most cases, such equipment only provides the cartesian coordinates of the map data and not the required geographic coordinates. It is the purpose of this paper to discuss a very simple but effective mechanism to perform the final conversion of the digitizer cartesian coordinates to geographic coordinates.

The principles discussed here have universal application to conversion of digitizer data. Consequently, the method is first developed in terms of abstract coordinate spaces and the problems associated with specific cartographic applications are deferred until later examples.

Operating System and Application Software

All of the software discussed is coded in the C programming language and example process executions are coded as Bourne shell scripts designed for use on the UNIX operating system. Programs **proj** and **toxy** referred to in this paper are available from from the author. Although the author has developed procedures to capture data from digitizing devices, these procedures are too specialized in terms of specific hardware requirements to be of general interest.

Basic transformation process

In most cases the mechanical devices used for data digitizing produce cartesian coordinates, (x,y), which must be converted to an application defined coordinate system, (u,v), with the only constraint that there is a single, unique functional relationship between the (u,v) and (x,y) system. Consequently, one of the principle functions of digitizing software is to provide a bivariate *inverse* transformation function to perform this conversion:

$$T^{-1}(x,y) \rightarrow u,v \tag{1}$$

Note that the inverse of T^{-1} is the *forward* transformation function

$$T(u,v) \rightarrow x,y \tag{2}$$

that must be able to operate with T^{-1} and produce the identity:

$$u,v \equiv T^{-1}(T(u,v)) \tag{3}$$

that will hold for all (u,v).

A general method of creating a function is by determining a pair of bivariate polynomial functions:

$$u = \sum_{i=0}^{N}\sum_{j=i}^{N} a_{ij}P_i(x)P_j(y) \tag{4}$$

$$v = \sum_{i=0}^{N}\sum_{j=i}^{N} b_{ij}P_i(x)P_j(y) \tag{5}$$

where the function P is typically the Chebychev polynomial and the coefficients a_{ij} and b_{ij} are determined by standard least squares methods based upon data acquired during a calibration phase at the beginning of the digitizing operation. The degree, N, would probably be determined by analysis of the calibration data but will always be limited by the number of calibration points used which consist of a set of (u,v) coordinates and their corresponding (x,y) digitizer values. The principle problem with general application of this method is that the required value for N for non-linear transforms may be sufficiently large so as to require a large number ($\geq (N+1)(N+2)/2$) of calibration points. For example, logarithmic data would require a fairly high order polynomial to transform the data accurately. In addition, higher order polynomials have a tendency to become unstable and prone to develop unwanted undulations in the transformation surface. However, where the (u,v) system is cartesian this is the method of choice and only requires the input of three or more calibration points for a first degree ($N = 1$) polynomial.

Equation 1 can be rewritten for the polynomial method as:

$$T_N^{-1}([u_c, v_c]_k, [x_c, y_c]_k, x, y) \rightarrow u, v \qquad (6)$$

where the bracketed coordinate pairs represent arrays of k calibration points and where $k \geq (N+1)(N+2)/2$. The function **toxy** (Evenden, 1987) implements this method in the UNIX environment by operating on an input stream in a manner similar to the argument list of the function T_N^{-1}. The basic internals of the procedure are:

- input the first two sections of calibration data into arrays (each section is terminated by a # character),
- check for equal number of entries for each section and that the number of entries (k) is sufficient for the selected polynomial degree N,
- determine the two sets of bivariate polynomial coefficients using commonly available least squares routines,
- check success of previous step and, if successful,
- process the remaining data with the determined polynomial transformation equations.

Digitizing complex coordinate systems

For (u,v) coordinate systems that have a high degree of curvature the aforemention problems of polynomial modeling surfaces complicate the application of this technique. However, a simple modification that is applicable to a wide variety of problems will simplify and stabilize the transformation.

If there is a forward function F and inverse function F^{-1} with the same properties of equations 1, 2 and 3 that will transform the (u,v) coordinates to and from any intermediate cartesian system then equation 6 can be modified to:

$$F^{-1}(T_1^{-1}([F(u_c, v_c)]_k, [x_c, y_c]_k, x, y)) \rightarrow u, v \qquad (7)$$

where $k \geq 3$. T_N^{-1} now only has to function in the first degree mode since it is only concerned with a cartesian to cartesian transformation to remove axis offset, scaling and rotation in the intermediate coordinate system. The F^{-1} function performs the final conversion to the (u,v) coordinate system.

F and F^{-1} express the operation of the transformation with a mathematical expression rather than with a polynomial approximation that may be difficult to determine. T_1^{-1} is still required because the digitizer's coordinate system will not normally match that required for input into F^{-1}.

A Simple Example

Digitization of a simple semi-log graph serves as a demonstration of this technique. In this case the x-axis is linear and the y-axis is a display of $\log_e(y)$ and the resultant data required is simply x,y. The first process is to create the F and F^{-1} transformation filter functions (file `tfor.c` contains a C language procedure in this example):

```
main(argc, argv) char **argv; {
   char s[250];
   double x, y, log(), exp();
   int forward;

   /* set mode */
   forward = strcmp(*argv, "tinv");
   while (gets(s))
     if (sscanf(s,"%le %le", &x, &y) == 2)
        printf("%e\et%e\n", x,
           forward ? log(y) : exp(y));
     else
        puts(s);
}
```

For convenience, only one routine is written and the forward-inverse mode is selected based upon the program's name so that the following UNIX shell script should be followed:

```
cc -o tfor tfor.c -lm
ln tfor tinv
cat <<EOF | tee $$temp | tfor | tinv | \
     paste $$temp -
  1         0.03
```

```
 2          1.3
 0.5       5000.
10          101.
EOF
rm $$temp
```

The resultant side-by-side listing of the input and output of the forward-inverse check sequence:

```
 1       0.03     1.000000e+00     3.000000e-02
 2        1.3     2.000000e+00     1.300000e+00
.5     5000.      5.000000e-01     4.999999e+03
10      101.      1.000000e+01     1.010000e+02
```

provides a quick visual verification as to the proper operation of the transformation function(s) on the four sample data values.

During the digitizing calibration phase, four points will be determined at the corners of the original plot that have the data values x=0 and 10 and y=.01 and 1000 entered in a clockwise manner. Consequently, the (u,v) calibration file, **calib**, will consist of:

```
0       .01
0       100
10      100
10      .01
```

The digitizer's output at these four corners plus some sample data yield the corresponding coordinates in the file `raw`:

```
#
1000    2000
1000    3000
2000    3000
2000    2000
#
1000    2250
1100    2500
1500    2750
1800    2250
     ...
```

The delimiter flag expected by **toxy** was inserted at the beginning and end of the digitizing calibration sequence. The shell script to transform the information will be:

```
tfor <calib | toxy - raw | tinv >reduced
```

The file `reduced` contains the results:

```
0.000000e+00    1.000000e-01
1.000000e+00    1.000000e+00
5.000000e+00    9.999999e+00
8.000000e+00    1.000000e-01
                ...
```

Although the data and their results were actually processed, the example's digitizer output is obviously artificial so the reader can visually check the conversion. Other compilers such as FORTRAN or interpreters such as **awk**(1) could have been used for the transformation functions.

Cartographic digitizing

Digitization of cartographic information is the logical extension of the semi-log example with the replacement of the `tfor.c` filter program with appropriate cartographic transformation procedure. Cartographic transformations (more commonly called projections) are often complex, but a program **proj** (Evenden, 1990) may be used that performs forward and inverse projections for many common projection types. All that remains is demonstration of how the pieces fit together.

In this example a few geographic positions will be obtained from the New York $2^\circ \times 1^\circ$, 1:250,000 scale map published by the U.S. Geologic Survey. The Universal Transverse Mercator (UTM) projection is used and **proj**'s default ellipsoidal parameters are applicable to this map. Consequently, the cartographic parameters for **proj** will appear as: `+proj=utm` and `+lon_0=73w`. Note that the central meridian value of 73° is based upon the New York sheet's central meridian and will be adjusted automatically by **proj** to the appropriate UTM zone 18 central meridian of 75°. The calibration process will use the four corners of the map so that the following list will apply for (u,v):

```
40n 74w
41n 74w
41n 72w
40n 72w
```

A Summagraphic's Microgrid digitizing table with cross-hair cursor is used as the digitizing device with the output collected a placed into a file named `NYin`. Cursor positioning was done with the unaided eye and the paper map sheet was in reasonable physical condition (it had not been folded but showed some signs of wear). The following listing of file `NYin` shows the results of the digitization:

```
# -c
+12444  +12986
+12796  +30448
+39186  +30250
+39262  +12797
# -e
+15873  +17287
+22594  +21568
+35888  +25879
+35904  +17149
```

Using the Bourne shell script:

```
a="+proj=utm +lon_0=73w"
proj $a -r2,1 NYcal | toxy - NYin | \
    proj $a +inv >NYdata
```

After execution the contents of file NYdata read as:

```
73d44'59.983"w   40d14'58.033"n
73d15' 0.153"w   40d29'58.922"n
72d15' 2.098"w   40d44'59.764"n
72d14'59.138"w   40d15' 0.073"n
```

The original points digitized were graticule fiducial marks at:

73°45′ W	40°15′ N
73°15′ W	40°30′ N
72°15′ W	40°45′ N
72°15′ W	40°15′ N

The above results should be considered acceptable for a map of this scale when it is considered that a 0.1 mm digitizing error at this scale (1:250,000) will result in an error of about 1″ in longitude and 0.8″ in latitude. Determination of digitizer coordinates to within 0.2 mm, as the above example demonstrates, is about the best that can be expected for manual digitizing. Registration errors apply to the calibration determinations as well as the data so that the entire process for one map can be compromised by poor calibration.

Factors involved in digitizing accuracy include the resolution and accuracy of the equipment, accuracy of the source document, and the skill level, technique and fatigue of the operator in manual methods. A rigorous treatment of these error factors requires statistical analysis beyond the scope of this paper. Recent use of scanning digitizers significantly reduces accuracy factors related to operator skills.

Digitizing unknown cartographic systems

An unfortunate situation often arises when the cartographic characteristics (projection type, ellipsoidal figure, etc.) of the map are unknown. Experienced technical personel can often estimate the projection used based upon the nature of the map's coordinate graticule and select appropriate projection parameters. But if this fails, **toxy**'s ability to use higher degree transformation polynomials must be used along with the necessity of additional calibration information. This process requires some degree of trial and error effort in order to determine the largest value of N necessary for desired accuracy of the transformation and should be used with a great deal of caution.

It cannot be over emphasized that lack of legend information describing the cartographic characteristics of a published map is a serious omission. In addition, a map without an adequate graticule destroys the graphic's ability to convey quantitative information. The graticule is normally the only information on a map that can be used as a calibration device in digitizing data.

As an example, the previously described New York map's cartographic characteristics are assumed to be missing and a guess of the projection used cannot be made because of the lack of any definitive characteristics. Examination of the map indicates that the graticule is nearly rectangular so that hopefully only a second degree transformation is required.

To maintain a balanced distribution of calibration points in both axis a calibration set of 9 points (6 or more are required for a second degree surface) is determined at the graticule fiducials as entered in file NYcal2:

```
40      -74
40      -73
40      -72
40.5    -74
40.5    -73
40.5    -72
41      -74
41      -73
41      -72
```

Note that the coordinates must be in decimal degrees because **toxy** only deals with simple, real number input. Using the same test points as in the previous example the resultant output from the digitizer stored in file NYin2:

```
# -c
+13763  +17331
+27165  +17447
+40566  +17718
+13752  +26056
+27038  +26159
+40333  +26430
+13733  +34785
+26919  +34893
+40111  +35153
# -e
+17086  +21701
+23709  +26118
+36911  +30716
+37108  +21986
```

Execution of the Bourne shell script (the procedure **dtodms** converts the output to degree-minute-second format for consistency with previous **proj** operations):

```
for a in 1 2
```

```
do
   toxy -d $a NYcal2 NYin2 | dtodms >NYtest$a
done
```

produces the following results for the first degree transformation:

```
40d14'56.919"N  73d45'12.634"W
40d29'48.875"N  73d15' 2.821"W
40d44'58.958"N  72d15'11.121"W
40d14'57.241"N  72d14'50.789"W
```

Unless the accuracy requirements are low these results are not acceptable. The following results for a second degree transformation, however, are usable and commensurate in accuracy with the method using program **proj**:

```
40d14'59.365"N  73d45' 1.608"W
40d29'59.570"N  73d15' 1.919"W
40d45' 1.454"N  72d14'59.894"W
40d14'59.216"N  72d15' 1.18"W
```

In general, digitizing of large scale maps of limited extent and in the equatorial to upper-mid-latitude range can often be performed with a second degree transformation function. Even first degree can be used for small spans of the calibrated area. But for maps where the extent is larger and curvature is more of a factor, then the use of **proj** is recommended. If in doubt, both methods should be used and an analysis of the results performed before investing a great deal of time in production digitizing.

Conclusions

Introduction of a forward-inverse transformation function into the process of converting digitized coordinates to the application coordinate system greatly facilitate the digitizing process. This minimizes the overhead of calibration operations and allows large, non-linear regions to be processed.

This technique has been successfully employed for several years by the U.S. Geological Survey in creating marine digital data bases from archival information and manually drafted data on work maps. It is simple to implement and is readily adaptable to a wide variety of digitizing problems.

References

Evenden, G.I., 1987, toxy—convert digitizer counts to user coordinates: Documentation file **toxy.1**, 1 p.

————, 1990, Cartographic Projection Procedures for the UNIX Environment—A User's Manual: U.S. Geological Survey Open-File Report 90–284, 63 p.