N82 28727

7.6 COMPUTATIONAL ASPECTS OF REMAPPING DIGITAL IMAGERY

Albert L. Zobrist
Jet Propulsion Laboratory
Pasadena, CA 91103

1. Introduction

One of the advantages of automated cartography is that map data stored in the digital computer can be plotted or displayed at any scale or projection by recomputing the coordinates of the data. This is especially easy in the case of vector (graphics) data but in the case of digital image (raster) data, remapping is a more difficult operation. Examples of the remapping of digital imagery would include rectification of a Landsat MSS to an orthographic or Mercator projection, warping of one image to register with another, or rotation, scale, or aspect changes of a digital image. Inputs consist of the digital image and geometric control information. Control information can include scanner location and pointing, ground truth, and the map transformation. Digital remapping consists of two major steps. First, a distortion model is computed from the control information. Second, the image is warped according to the distortion model.

The first step involves traditional mathematical techniques of estimating a surface from sample points. Several approaches persist because of varying needs of different applications. The second step involves highly specialized computational methods for efficient warping of large images according to a geometric distortion model. Use of general purpose computers and array processors for this task will be covered. Data processing error will be discussed for each modelling/warping approach.

2. Determination of geometric distortion model

Mathematically, a geometric distortion is a mapping from the plane to the plane. The mapping is usually one-to-one and continuous but there may be discontinuities. Orthogonal components of the mapping (the x and y coordinates) are independent and each can be viewed as a surface over the plane. For a point p, the value of the x-distortion surface at p specifies how far the data at p must move in the x direction in the remapping. Some simple geometric distortions are used to rotate images or change pixel size. In this case the general transformation is called affine or linear and the x and y components are planes. Map coordinate conversions (for example UTM to Transferse Mercator) are given by formulas which can also be viewed as distortion surfaces over a plane. Singularities and zone boundaries are not a problem here but are dealt with in sectioning images for a data base.

A more complex geometric distortion problem is the "rubber sheet" case where a set of control points relating the input to the output is known. A number of techniques are known for generating surfaces to fit the control points and give a distortion model over the entire surface. Some are general in nature: polynomial fit, nearest-neighbor interpolation, finite element method, and the method of potential functions. These are used in cases where there is no need or desire to use a priori knowledge of the nature of the geometric distortion. If one knows (from physical considerations) the general functional form of a distortion, then there are methods (least squares, Kalman filtering, etc.) of fitting the functional form to the observations. Table 1 compares some basic properties of these methods.

The most complex distortion models arise from sensor geometry correction. Taking the Landsat MSS to be a basic example, the raw data are perturbed by earth rotation, mirror scan nonlinearity, spherical earth, variations in platform altitude, roll, pitch, and yaw. Most of these components can be modelled by continuous functions, but one component, the line-to-line skew induced by earth rotation is discontinuous at every sixth line. Furthermore, the distortion model is no longer a simple function but a composite of several functions that are applied in order. The first correction function calculates uniform sample spacing in orthographic or Mercator projection along single scan lines. Then a second correction function moves entire lines according to the sensor and earth rotation skew for each line. A third correction for map projection could now be performed if desired. Two basic techniques for model fitting are in common use today. The first is to use nominal values for spacecraft location, etc., and produce a corrected product which has slight deviations from a perfectly mapped product. Note that the largest deviation is a simple lateral translation, which can be fixed later by a single point observation. The second technique is to fit the model according to control points determined by external means. These methods are covered in other reports in this workshop.

3. Representation of geometric distortion model

Digital computation requires that the geometric distortion be represented in an efficient manner. Three methods are covered here. The first method is to leave the model in its functional or natural form. Model fitting provides coefficients or data for a subroutine F which can be invoked at a point p to give the distortion F(p). The second method is to convert the model to a gridded approximation. A rectangular grid $a_{11}, a_{12}, \ldots, a_{21}, \ldots, a_{mn}$ is set up and the subroutine F is calculated at these points. The values a_{ij} and $F(a_{ij})$ are stored in a data structure so that the value F(p) can be generated by interpolation in an efficient manner. The third method is a highly specialized one for scanner type data such as MSS. In cases where some components of F are functions of one variable (separable components) a "dope vector" can be set up to represent the shift. As an example, the mirror scan nonlinearity is a function of position along a scan line only. A dope vector of the same length as a scan line can represent this correction on a per pixel basis. Earth skew offset and sensor readout delay can be represented by a dope vector with a per line lookup. Both of these corrections are "along track", that is, in the direction of scanning. It is possible to have dope vectors for across track corrections as well, if needed.

Direct computation of F for each pixel is a slow method, although it may be helped by array processor techniques. Gridded representation offers great speedup simply because a 50 x 50 grid, for example, requires a factor of 3072 times fewer evaluations of F than a 2400 by 3200 image would by direct computation. Gridding introduces a data processing error, however (see later section on error). In general, discontinuous functions and functions which are not approximated well by interpolation on a grid are poor candidates for gridding. Dope vectors can only be used on separable functions, of course.

A special strategy for MSS involves a combination of dope vectors for the discontinuous and highly nonlinear elements of F and a gridded approximation for the remainder. Evaluation at F(p) would involve several table lookup operations in the dope vectors followed by the grid interpolation.

4. Large image warping computation

Regardless of method, the remapping of a digital image can be an enormous computation. For example, an MSS input contains over seven megabytes of data per spectral band and yields over ten megabytes of data for 57-meter square pixels. Executing ninety machine instructions per output pixel at one microsecond per instruction would occupy about 16 minutes of processor time. Yet this slim number of cycles must accomplish the following:

(1) For each output pixel, calculate the location of the input point that maps to it (the inverse of the mapping).

(2) For each output pixel, calculate the pixel value based on interpolation of input pixel values neighboring the input point.

(3) Buffer the input and output so that a reasonable main memory region can accommodate the calculation without excessive disk head motion or file rereading.

Computational aspects of these steps for an ordinary digital computer will be covered in order.

The first step requires that for an output pixel location p and an inverse mapping F that F(p) be calculated. If F is a composite function, then each component must be calculated in order. Functions represented by formula are evaluated by their subroutine. There are some opportunities for speeding up function evaluation, for example, by use of table lookup for parts of a function (such as a cosine). Another example is incremental evaluation where

$$F(x + dx) = F(x) + G(x,dx)$$

and G(x,dx) is faster to compute than F(x). A concrete example of this is

$$\cos(x + dx) = \cos x \cos dx - \sin x \sin dx$$

so for uniform dx a cosine can be calculated with two multiplies and an add, assuming that sin x is maintained in a similar fashion. The incremental evaluation can even be an approximation if care is taken to restart with an exact evaluation frequently enough to limit the error to an acceptable range. Functions represented by a grid are amenable to a much faster treatment. Within each grid cell an incrementing scheme can be set up consisting of

$F(x_o, y_o)$	an initial point
Δx	
Δy	increments
Δxy	

which allow for recalculation of F for a series of increments in the x direction

$$F(x + dx, y) = F(x,y) + \Delta x$$

and for a move to the next line of output

$$F(x,y + dy) = F(x,y) + \Delta y$$

$$\Delta x = \Delta x + \Delta xy$$

This corresponds to bilinear interpolation on the grid. If non-uniform increments are needed because of function composition, additional multiplications b dx and dy will be required. When dope vectors are used it is usually accurate enough to use the correction value from the nearest pixel. As an example, the along-track correction for an across track dope vector is

$$F(x,y) = x + D(\text{round } (y))$$

One special problem arises from discontinuities in the mapping function. For purposes of pixel value interpolation, it is necessary to know about local discontinuities in the neighborhood of F(x,y). Hence for cubic spline interpolation for MSS a point (x,y) maps into four locations for four lines which are offset from each other. Fortunately in this case, the samples are uniformly spaced (Figure 1).

The second step of warping computation is the actual interpolation for the output pixel value from the neighboring input. Methods for this are discussed elsewhere in the workshop.

The third problem involves the allocation of limited main storage to storage of a part of the raster input so that the raster output can be computed efficiently. Two previous methods did not work well for large or highly rotated input rasters (for example 3000 x 3000 rotated 11°). Method 1 stores a band of raster lines internally as shown in Figure 2. Because of rotation an output line will only have a short intersection with this band. Therefore, it is only possible to calculate an extremely large number of short output segments which must be written to disk and later reconstructed into the output raster. The later reconstruction involves excessive disk head motion for large cases. If the raster is n by n and available storage is fixed, then the length of calculated segments is $O(1/n)$, the number of cells is $O(n^2)$, hence the number of short segments is $O(n^3)$ and disk head motion will increase by this factor under a simple reconstruction scheme. Method 2 avoids the reconstruction of short segments by storing all of the input raster in the neighborhood of an output line. But because the stored input area is not a band, the input file must be reread as many times as there are plateaus in the lower part of the stored area. The thickness of the stored area is $O(1/n)$ and the length of the line is $O(n)$ hence the number of rereads of the input data set is $O(n^2)$. Since the amount of data is $O(n^2)$ the disk head motion will increase by $O(n^4)$. The new method developed computes a uniform vertical band of optimal width in the output by storing a corresponding swath of input. The output segment width is independent of n hence disk motion depends on the number of times the input has to be read which is $O(n)$ times amount of data yielding $O(n^3)$. The reconstruction stage is $O(n^2)$. Thus method 2 is unsuitable for large cases and methods 1 and 3 appear the same. A closer analysis reveals that method 1 forces a head motion for sequential passes over the data which minimizes head motion and rotation latency. Methods 1 and 3 are implemented in the VICAR routines LGEOM and MGEOM respectively, and method 2 was reported by H. K. Ramapriyan (1977).

Unusual approaches to this problem have been proposed. One is to resample horizontally, rotate the image 90° and then resample vertically (which is horizontal after rotation). Good methods for 90° rotation are available

(Twogood and Ekstrom, 1976). Resampling techniques and experimentation are reported by Friedmann (1981).

5. Data processing error

Image registration and rectification error analysis is the subject of another report in this workshop. Therefore, model errors will not be considered here. Data processing error includes only the error introduced in the following ways:

(1) errors in calculation of the inverse mapping F(x,y)

(2) errors introduced by interpolation on a grid or dope vector representation of F(x,y)

(3) errors in the location of neighboring pixels of F(x,y) for input to the interpolation scheme.

There may also be error in the interpolation scheme, but this is not a location error. With regard to the three errors, note that a 1/10 pixel error on a 3000 x 3000 image requires an accuracy of one part in 30,000. Gridding methods are the most difficult to hold within such an error budget. One component of grid error is the deviation of the bilinear surface from the model surface. A second component is accumulative error in the incrementing scheme described in the last sections. Both of these errors are controlled by keeping the grid size small. The accumulative error necessitates the use of computer arithmetic with greater precision than 32- or 36-bit floating point.

6. References

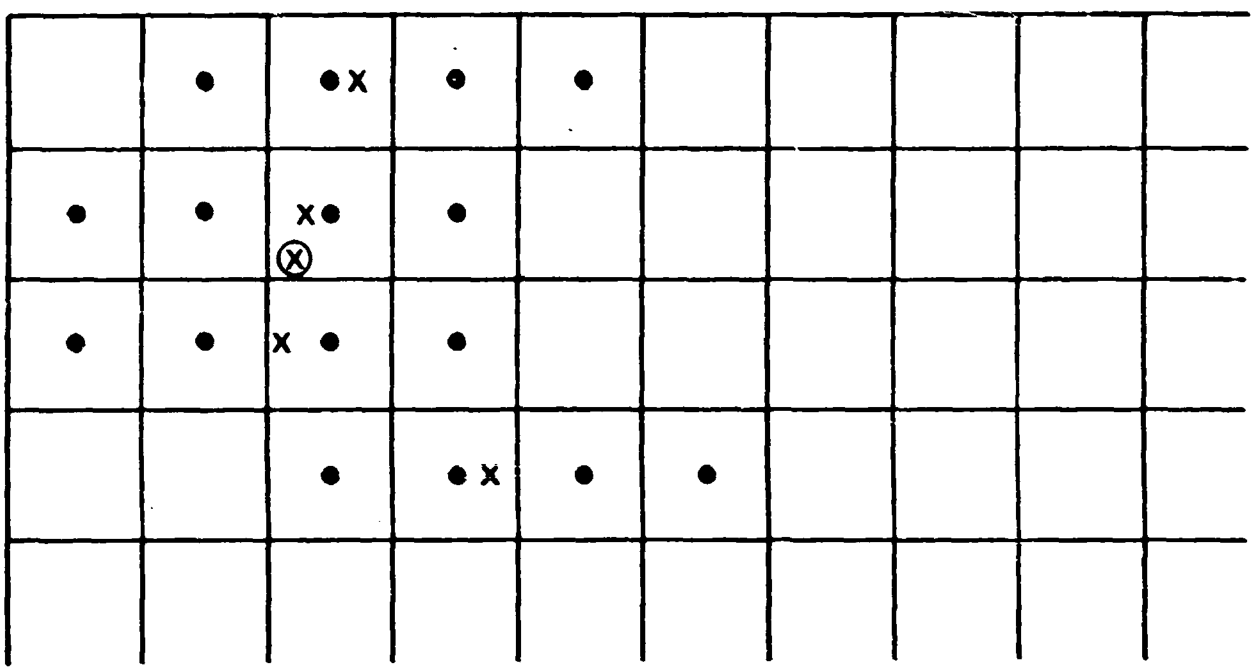
Ramapriyan, H. K. (1977), "Data handling for the geometric correction of large images," IEEE Transactions on Computers, vol. C-26, no. 11, pp. 1163-1167.

Twogood, R. E., and M. P. Ekstrom (1976), "An extension of Eklundh's matrix transposition alogirthm and its application in digital image processing," IEEE Transactions on Computers, vol. C-25, no. 9, pp. 950-952.

Friedmann, D. E. (1981), "Two-dimensional resampling of line scan imagery by one-dimensional processing," Photogrammetric Engineering and Remote Sensing, vol. 47, no. 10, pp. 1459-1467.

Table 1. Characteristics of Some Common Surface Fitting Methods

METHOD \ PROPERTIES OF SURFACE	CONTINUOUS	DIFFERENTIABLE	EVALUATES AT INPUT POINT	WELL BEHAVED NO MESAS
TRIANGULATION	YES	NO	YES	YES
INTERPOLATION r^{-1}	NO	NO	YES	YES
INTERPOLATION r^{-p}	NO	NO	YES	NO
POLYNOMIAL FIT	YES	YES	NO	NO
POTENTIAL FUNCTION	YES	YES	YES	?
SPACECRAFT MODEL	YES	YES	NO	YES



⊗ INVERSE MAPPED LOCATION OF OUTPUT PIXEL

x CENTERS OF INTERPOLATION ALONG A LINE

• INPUT PIXELS USED IN BI-CUBIC SPLINE INTERPOLATION

Figure 1. Interpolation in the Presence of Discontinuities in Input Data Along a Line

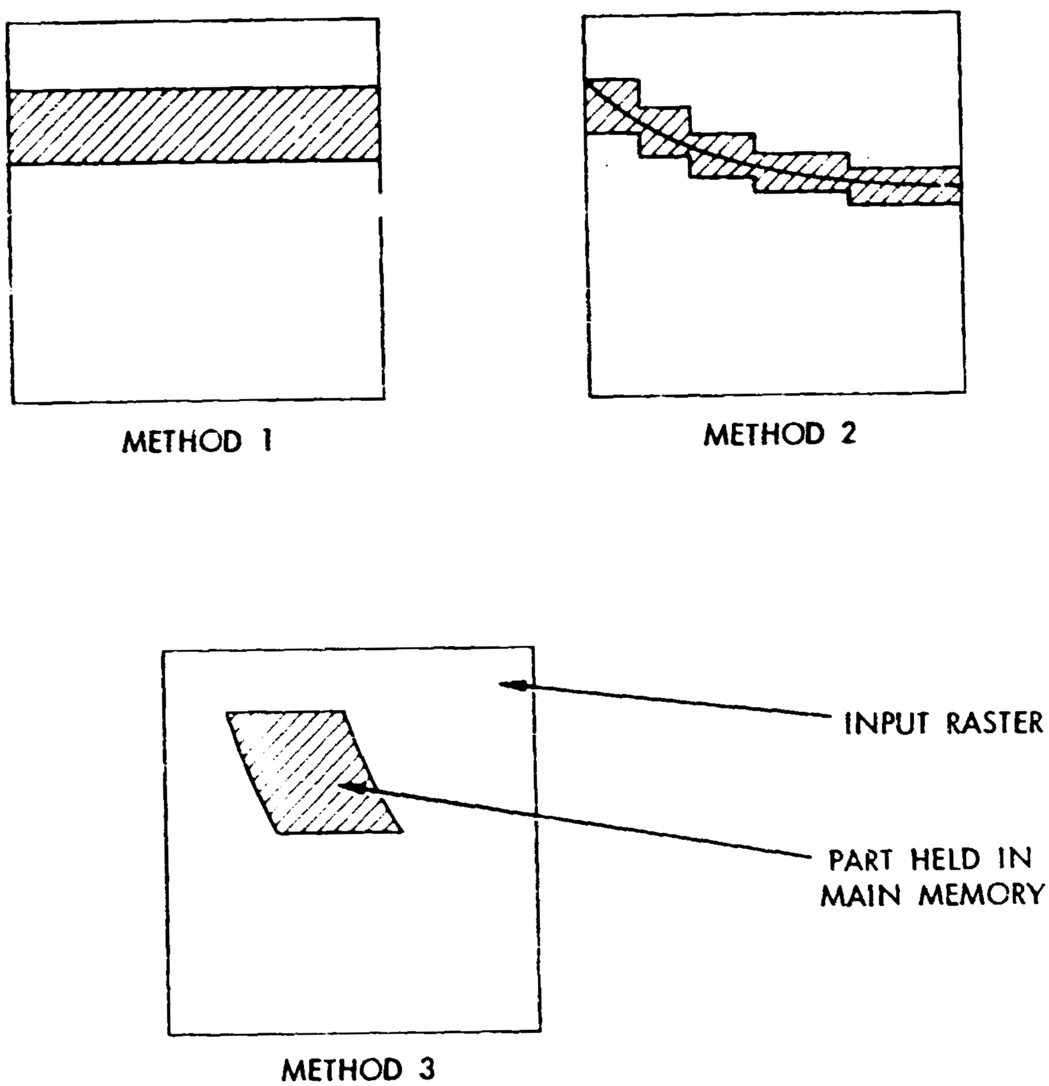


Figure 2. Diagram of Allocation Schemes for Main Memory for Large Image Warping Computation

ORIGINAL PAGE IS
OF POOR QUALITY