ON-DEMAND PROJECTION OF LARGE RASTER DATASETS IN GIS

Hong Xu, Peng Gao and Jason Willison

ESRI, Inc
Redlands, California 92373
USA
hxu@esri.com pgao@esri.com jwillison@esri.com

ABSTRACT

On-demand projection of large raster datasets is becoming increasingly important as the growing interest in utilizing regional and global raster datasets as well as the use of terabyte-sized high-resolution images in GIS, and these raster datasets are normally not stored in the same projection/coordinate system. Current available algorithms do not meet the fast and accurate requirements of the on-demand raster projection. This paper presents a new algorithm based on a bilinear approximation mesh for transforming images from one projection to another. Implementation and tests show that the algorithm is accurate and fast and has built a solid foundation for GIS applications utilizing large image holdings of local or global raster data.

BACKGROUND

Transforming geographic data from one projection to another is fundamental to GIS analysis and visualization. Data that represent the same geographic location may come from various data sources, which could be produced and delivered in different projection/coordinate systems. Performing GIS analysis with data in different projections requires re-projecting those data into a common coordinate system.

As the development of remote sensing technology, image data is becoming widely used in GIS applications because very large image archives are available from which to access, and they provide more recent information for decision-making. Since the recent availability of high-resolution image data as well as the growing interests in global environmental research and applications, the images used in GIS applications and visualization are becoming very large, and gigabyte or even terabyte images are increasingly common. To transform large images accurately and efficiently becomes crucial and on-demand raster projection, which transforms raster data to a common coordinate system on-the-fly without physically writing to a file on disk, becomes more favorable. This however, requires higher performance in the raster projection algorithm.

The methods of projecting raster data that are used in existing commercial software are suitable for small images. However, it is either slow or inaccurate for handling large images, especially for images of regional and global scales (Ueser, 2003). This paper introduces a new and practical algorithm based on an approximation mesh and a bilinear interpolation method. The algorithm has been implemented in ESRI's upcoming ArcGIS 9.2 product for on-demand raster projection. The test shows that this algorithm is faster, accurate, and practical for projecting large raster datasets.

INTRODUCTION

A projection algorithm is one or more mathematical functions that projects points (x,y) from one projection system to another; this process is called point-to-point projection. The point-to-point projection algorithms apply directly to feature datasets that consist of points, lines and polygons. Raster data consists of pixels, whereby each pixel has a location (x, y) and an area that the pixel represents in map space. A direct solution for transforming raster data is to project each pixel using the point-to-point projection algorithm. This method, which has been used in commercial software including the PROJECT command in ESRI's Workstation ArcInfo, can achieve high accuracy, however it usually is very slow and thereby not practical for projecting large images, needless to say on-demand raster projection.

The common approach to increase the raster projection performance in most existing commercial software, including ESRI's ArcGIS 8.3 or older, was based on a polynomial approximation. This algorithm first projects a fixed number of control points such as 16 x 16 (chosen evenly throughout the extent of the raster dataset); then builds a single, third order polynomial transformation using a LSF (Least Square Fitting) algorithm on the two sets of control points (from and to); lastly, it transforms the pixels using the calculated polynomial (Figure 1). This algorithm is fast and has an acceptable accuracy for raster data that covers a small extent. However, since control points used in this algorithm are fixed for any raster data regardless the extent, this approach results in a big error when the image covers a large extent, or when it is located at high latitudes. The problem is due to that the single polynomial does not approximate the transformations between various projection types very well, especially for projections that have big distortions or interrupted projections such as the Cube or Fuller projections.

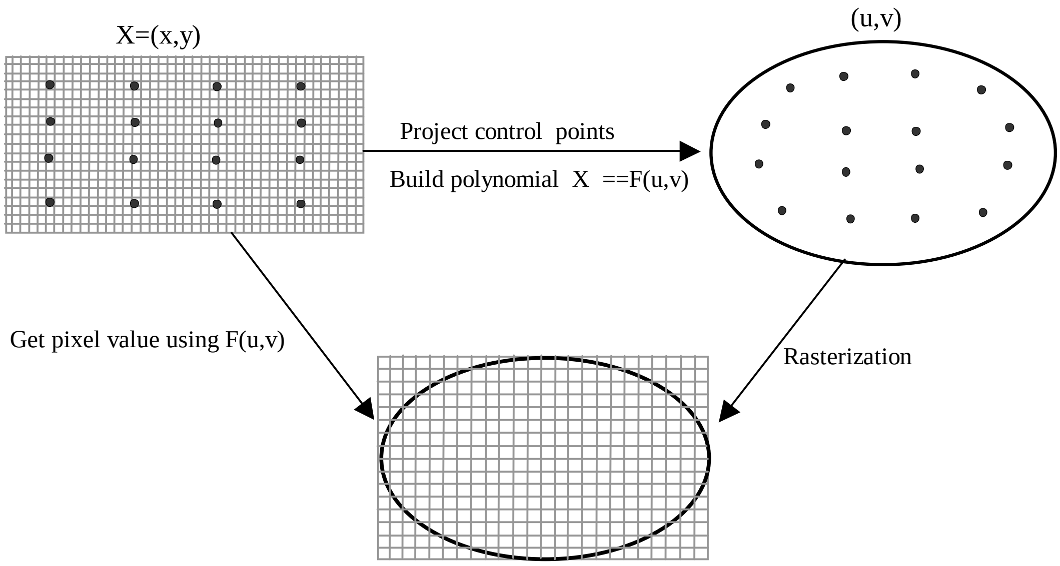

Figure 1: Flowchart of Projecting Raster Using Single Polynomial Transformation Algorithm.

An improved polynomial algorithm was developed and implemented in ArcGIS 9.0 products. This algorithm, called piecewise polynomial transformation, divides the whole raster dataset into small blocks and then transformed each block using one polynomial. Since polynomial approximation does not transform input control points exactly to the output control points, to control the seamlessness between transformed blocks required a small block size, which increases the total number of blocks, whereby increasing the number of computations to calculate the polynomials (using LSF). This led to the computational complexity of that raster projection implementation.

BILINEAR APPROXIMATION MESH ALGORITHM

To enhance the accuracy and performance for projecting large images, a new algorithm, based on a bilinear approximation mesh, has been developed. This raster projection algorithm has three characteristics: 1. It approximates the point-to-point projection using a point grid and bilinear interpolation between points; 2. It handles interrupted projection types such as Fuller and Cube projections by dividing the output space into continuous domains and maintaining the continuity between domains; 3. It practically guarantees that the approximation error does not exceed a given tolerance.

Projection Domains

Every projection has a horizon that defines the geographic limits of a map projection. The horizons of most projection types can be represented as one polygon. For example, WGS84 has a rectangular horizon and Sinusoidal has an ellipsoidal horizon. Those projections have one domain. However some projections have more than one polygon in the horizon such as the Fuller projection, Cube projection, or projections with a non-zero central meridian. For example, Cube projection has 9 domains (Figure 2). For a given input and output projection types, the projection domains are computed by getting the horizon in GCS (geographic coordinate system) from both the input and output projections, overlaying them in GCS, and then projecting them into the output space to create a set of domain polygons that are continuous in both input and output space. The following figures (2 through 6) explain the concept of the projection domains from a Mercator projection into the Cube projection.

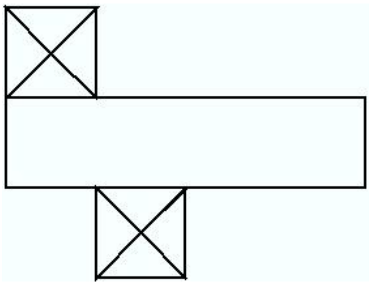

Figure 2: Cube Projection Domains.

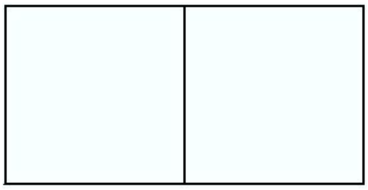

Figure 3: Input is Mercator Projection with a Non-Zero Central Meridian. It Has Two Domains After Being Projected to GCS.

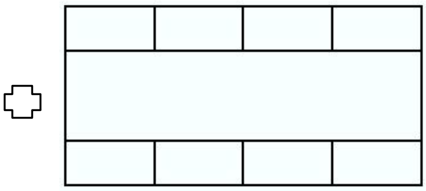

Figure 4: Domains of Cube Projection After Being Projected to GCS.

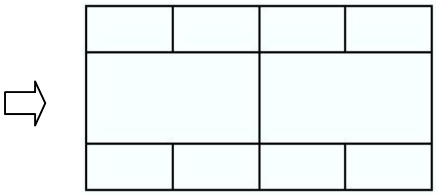

Figure 5: Projection Domains After Overlying in GCS.

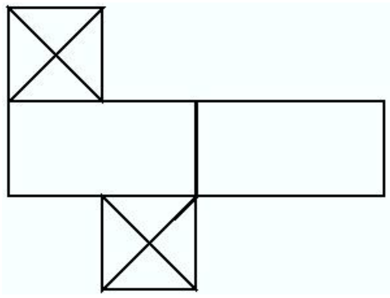

Figure 6: Final Projection Domains in Output Space

Projection Approximation Using an Approximation Mesh

Projecting raster includes both forward and reverse projection of raster data. This process entails first projecting the horizon of the input raster dataset to the output space, resampling the horizon in output space into a desired pixel size, reversely projecting the pixels in the output space to the input space, and finally performs resampling in the input space to get pixel values to fill in the output space.

An approximation mesh for raster projection is a grid that is formed by regular points in output space (Figure 7). Each cell of the grid consists of many pixels (16 x 16, or 256 pixels total, were used in this study). Given a pair of input and output projections, the projection approximation mesh is independent of the extent and cell size of the input raster dataset. The origin of the projection grid is the upper-left corner of the extent of the projection domains.

A cell in the point grid has four corner points and contains four cells of its sub grid (Figure 8 and Figure 9). For each cell in point grid, the algorithm first projects the four corner points using point-to-point projection, then tests if the four corner points are all within the output domains; next it interpolates the five check points on the sub grid using the four corner points (Figure 8), and also uses point-to-point projection of the five points on the sub grid (Figure 9); lastly it calculates the interpolation error as the distance between point-to-point projected check points and interpolated check points in the input space. A cell in a point grid is considered to be valid if all of its corner points are inside the output domains AND the bilinear interpolation results in an error at all five check points, which does not exceed the chosen tolerance.

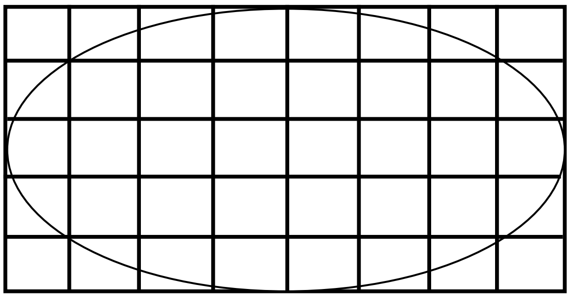

Figure 7: A Representation of Approximation Mesh of Sinusoidal Projection at Certain Level.

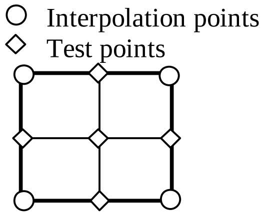

Figure 8: A Sub-Grid of a Cell. Bilinear Interpolation Is Used to Compute the Values on the Five Sub-Grid Points.

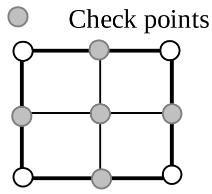

Figure 9: A Sub Grid of a Cell Point-to-Point Is Used to Project the Five Points of the Sub-Grid.

Once a projection grid is tested and verified, to project a point or a pixel, the pixel is first snapped to a cell of the point grid and then the validation of the cell is performed. If it is valid, the point is projected using a simple bilinear interpolation; otherwise, the point is projected using a point-to-point projection.

Since the grid testing phase ensures that errors at all five check points are less than the tolerance, and the interpolation is done in one of four sub cells within the test cell, the final interpolation error will practically be less than the tolerance. Although this is not guaranteed experimentation and testing shows that the raster projection error is always smaller than the tolerance.

The Raster Projection Algorithm

For a given pair of source and destination projections, the algorithm can be described as follows:

1) Compute the output domain as describe above.
2) Choose an approximation point grid such as 16 x 16 or 32 x 32 depending on the application requirement and experiment (Note: the point grid used in this study is 16 x 16).
3) Project the extent of input raster to output space.
4) Rasterize the projected extent in the output space at a given pixel size, requested by users or by the upper level application such as a display device.
5) The pixels in the output extent are sub-divided into small pixel blocks of a predefined size, e.g. 256 x 256 to fetch one at a time. For each pixel block, and for each domain, the domain is intersected with the extent of the pixel block.
6) All the pixel center points in the output space are projected reversely either using bilinear interpolation or point-to-point projection to input space, the bilinear approximation mesh algorithm described above.
7) The pixel values are calculated in the input space through various resampling methods for the output pixel block.
8) Since pyramids, which are a stack of reduced resolution resampled layers of pixels for fast raster access, will have been built for the large input raster dataset, only the pyramid layer necessary to satisfy the display requirements will be used in calculating the pixel values for the output pixel blocks.

TESTING AND RESULTS

Error Analysis and Performance Testing

The performance and accuracy of raster projection are related to many factors: 1. The input and output projection types, some projection types have big geometric distortions; 2. The extent of the input raster data, images that are located on the equator have less geometric distortions comparing to images close to the polar area; 3. The pixel size of the raster data; and 4. Choice of the projection error tolerance. Extensive testing has been performed to understand the accuracy and performance of the algorithm with regard to the above factors.

The experiment selects WGS84 and projections of most representative projection types: Conic, Cylindrical, and Azimuthal as the input and output projections, and performs two types of testing, which can be summarized with the following two questions:

1). At a given input raster data pixel size and dimension (extent), how is the accuracy and performance of this algorithm related to projection types and tolerance values?

The input raster data used in this test is a global image (1km pixel size with dimension of 24,000 x 48,000) in WGS84 and other projection types. The output projections are Mercator, Conic, Hotine, Robinson, Sterographic, Bonne, etc. For each projection pair and a given set of tolerances, the test randomly selects 10,000 random points, then projects those points using both this new raster projection algorithm as well as a point-to-point program, and then compares the performance and accuracy of raster projection with regard to point-to-point projection. The test iterates 100 times to get an average set of statistics. Below are the selected results of projecting from WGS84 to Conic projection.

The testing shows (Figure 10) that the algorithm can guarantee the raster projection error is always within the given tolerance for all projection types. Since this algorithm is actually interpolated on a sub point grid, the error is much smaller than the tolerance specified. Figure 11 shows that projecting same amount of points, using the raster projection algorithm is much faster than using point-to-point projection. The extent used in this study is the whole horizon of WGS84. For the points that are close to the horizon, point-to-point projection is used mostly. When using a raster of a smaller extent, which is a common use case, there should be an even higher performance boost since bilinear interpolation will be used primarily.

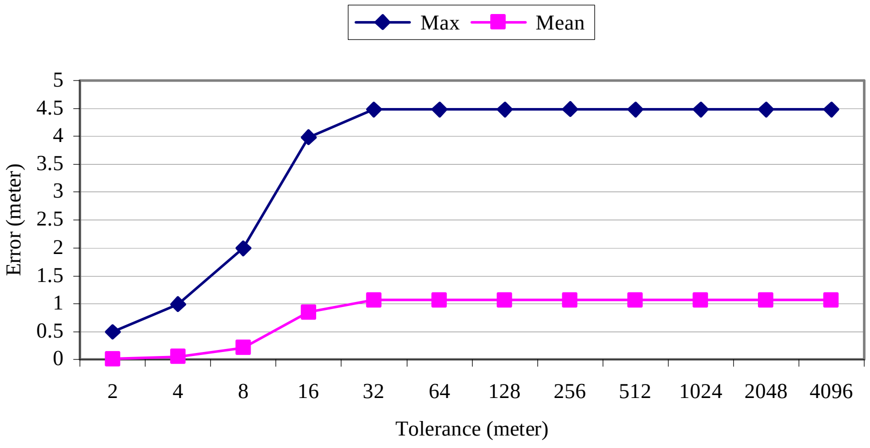

Figure 10: Project Sample Points using Raster Projection Algorithm and Point-to-Point Projection Algorithm. Input projection type is WGS84 and Output projection Type is Conic Projection. The Mean and Max Error of the New Raster Projection Algorithm is Always Under Control.

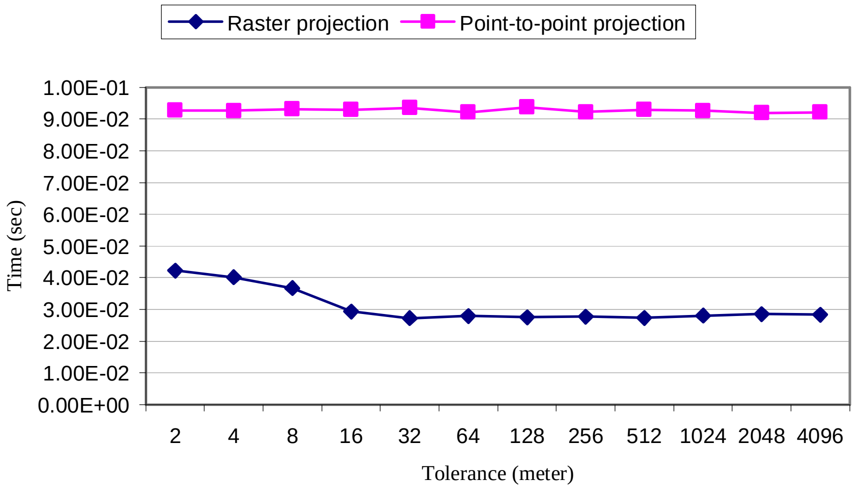

Figure 11: Project Sample Points using Raster Projection Algorithm and Point-to-Point Projection Algorithm Input projection type is WGS84 and Output projection Type is Conic Projection. The Raster Projection Algorithm is Much Faster than Point-to-Point Projection for Projecting Same Amount of Points.

2). At a given projection tolerance and extent of the input raster, how does the input pixel size affect the performance and the accuracy of the algorithm?

The input and output projection used in this test are WGS84 and Mercator projections, respectively. The extent is fixed (-180, 180, -90, 90) and the error tolerance is set to 100 meters. The test performs on a set of pixel sizes starting from 30 meters up to 16,000 meters (in output space). For each cell size, the test selects 10,000 points and projects them using the raster projection algorithm, and then iterates 100 times to get an average set of statistics (Figure 12 and Figure 13). The biggest error is 30, which is about 1/300 of the pixel size and far below the tolerance 100, and this happens at the 8,000 meter pixel resolution. The error curve explains the characteristics of the raster projection algorithm. At small pixel sizes, such as 30 or 60, the interpolation technique can better approximate the point-to-point projection, and also be fast and accurate. As the pixel size increases, interpolation technique is still being used mostly in the raster projection algorithm and reaches the highest performance advantage of the algorithm. As the pixel size increases, point-to-point projection becomes used more readily in the raster projection algorithm but not always, so the error is very small but the performance is still faster than point-to-point projection.

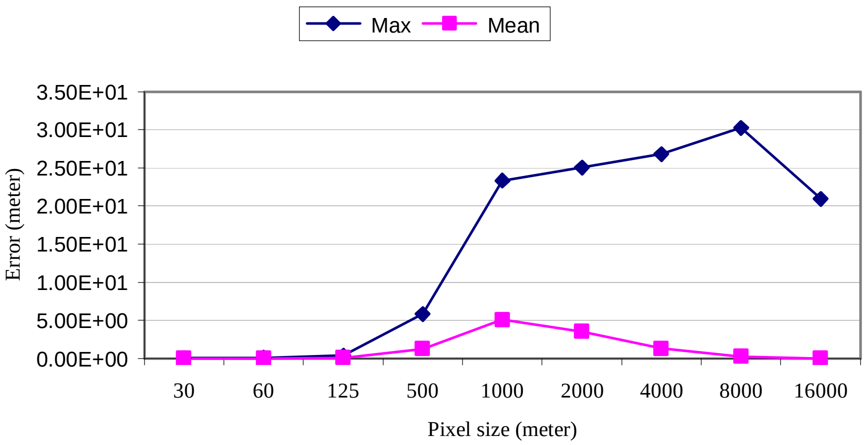

Figure 12: Project Sample Points from WGS84 to Mercator using Raster Projection Algorithm at Each Input Resolution. The Error is Calculated Comparing to Point-to-Point Projection.

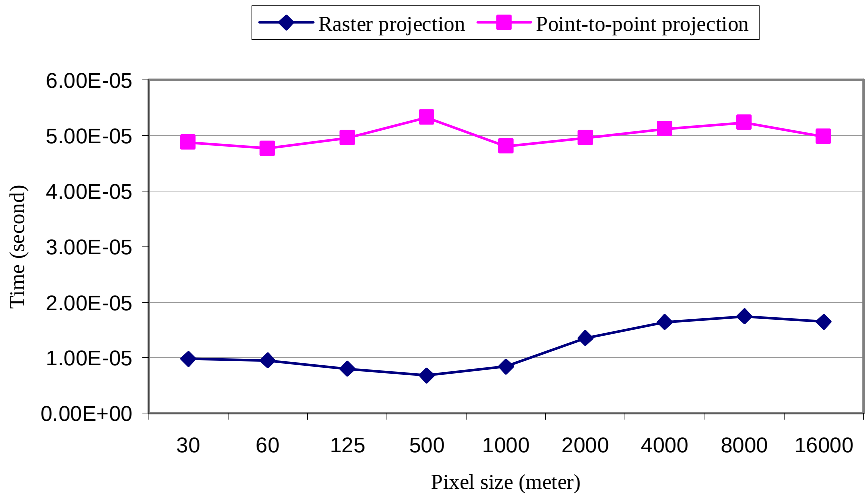

Figure 13: Project Sample Points from WGS84 to Mercator using Raster Projection Algorithm at Each Input Resolution. The Time is Calculated Comparing to Point-to-Point Projection.

Raster Projection Results

The bilinear approximation mesh algorithm has been implemented in ArcGIS software and will be available in the coming release (ArcGIS 9.2). The following figures show some examples of raster projection from WGS84 to various other projections.

Figure 14: To Bonne Projection

Figure15: To Conic Projection

Figure 16: To Cube Projection

Figure17: To Fuller Projection

CONCLUSIONS

The bilinear approximation mesh algorithm proposed in this paper is a new and practical algorithm. The experiment results show that it is fast and accurate for projecting raster data. Since this algorithm has been implemented in ArcGIS and will be available in the next release, this research and implementation on raster projection will certainly benefit GIS applications and researchers that use regional and global raster data.

The implementation and testing is based on point grid of 16 x 16. Future experimentation and research will look at how the grid size affects the performance and accuracy of raster projection.

REFERENCES

Ueser, E.L., Seong, J.C., Steinwayand, D.R. and Finn, M.P., 2003, Accurate Projection of Small-Scale Raster Datasets, *Proceedings of the 21st International Cartographic Conference (ICC).*

Steinwand, D.R., J.A Hutchinson, and J.P. Snyder, 1995. Map Projections for Global and Continental Data Sets and an Analysis of Pixel Distortion Caused by Reprojecion, *Photogrammetric Engineering and Remote Sensing*, 61(12), pp. 1487-1499.

ON-DEMAND PROJECTION OF LARGE RASTER DATASETS IN GIS

Hong Xu, Peng Gao and Jason Willison

ESRI, Inc
Redlands, California 92373
USA
hxu@esri.com **pgao@esri.com** **jwillison@esri.com**

BIBLIOGRAPHY

Hong Xu holds a MS degree in Computational Mathematics from Jilin University, China and a MA degree in Geography from The Ohio State University, USA, specializing in GIS.

She is a Product Specialist at ESRI, Inc, working on raster transformations within ArcGIS products.