Journal of Terrestrial Observation

Volume 2, Issue 1 *Winter 2010* *Article 6*

Automated Georeferencing of Historic Aerial Photography

Jae Sung Kim, Christopher C. Miller, and James Bethel

 ISSN 1946-1143.

Automated Georeferencing of Historic Aerial Photography

Jae Sung Kim, Christopher C. Miller, and James Bethel

Purdue University

ABSTRACT

Historic aerial photos are typically very popular, sought-after collections in library archives. Their ability to be used in GIS is limited, however, by the laborious and painstaking georeferencing process. A method is presented for automating the georeferencing of historic aerial photography. In this method, the Harris Corner Detector is used to detect corner points in already-georeferenced and unreferenced aerial photos, which are then matched by cross correlation. Falsely-matched corner points are removed with the application of the Random Sample Consensus with a six parameter transformation. The pixel arrays of inlier corner point pairs are input to a Java program developed with ArcObjects, which converts their pixel coordinates to map coordinates. The unreferenced aerial photos are then rectified and georeferenced by a Warp() function with a first order polynomial option. The results of tests thus far indicate the method is satisfactory and promises to be an important component in an aerial photo library digitization workflow.

KEYWORDS

Georeferencing; Harris Corner Detector; Matching; Cross Correlation; RANSAC; ArcObjects

1. INTRODUCTION

The increasing importance of GIS across disciplines is placing additional emphasis on the need to georeference useful analog materials that are often left to atrophy in map collections and libraries. Typically, georeferencing is a time- and labor-intensive process whereby users manually determine in unreferenced data a series of control points that can also be identified in referenced data. This is usually performed in a desktop GIS package and therefore requires significant process overhead. This paper presents an automated georeferencing methodology that automates the approximate georeferencing of vertical analog aerial photo exposures in areas of low relief to digital orthophotoquads. A consecutive application of the Harris Corner Detector (Harris and Stephens, 1988), matching by cross correlation, and Random Sample Consensus (RANSAC) identifies inlier control point matches from like imagery and the resulting coordinate array informs a Java *Warp()* function, specifically a first-order polynomial or six-parameter transformation in ArcObjects to

permanently transform the unreferenced data in real coordinate space.

2. METHODOLOGY

2.1. Harris corner detector

To georeference images, a geometric relationship between images must be established by identifying corresponding points from both files. One method of defining a set of interest points is to assume all corner points are candidates and simply extract the most conspicuous corners based on their higher cornerness values. The Harris corner detector algorithm (Harris and Stephens, 1988) was developed based on the earlier Moravec low-level corner detector (Moravec, 1980), improving upon the Moravec detector's anisotropic response, noisy response, and sensitivity to edges. The Harris corner detector analyzes gradients in a patch to provide cornerness measures for image data (Stottinger, 2008) and is described by Derpanis (2007) as equation (1),

$$c(x,y) = \sum_{W} [I(x_i, y_i) - I(x_i + \Delta x, y_i + \Delta y)]^2 \tag{1}$$

where I is the image function and $c(x,y)$ is the auto-correlation function at a point (x, y) given a shift ($\Delta x, \Delta y$).

$$\begin{aligned} c(x,y) &= \sum_{W} [I(x_i, y_i) - I(x_i + \Delta x, y_i + \Delta y)]^2 \\ &= \sum_{W} (I(x_i, y_i) - I(x_i, y_i) - \begin{bmatrix} I_x(x_i, y_i) & I_y(x_i, y_i) \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix})^2 \\ &= \sum_{W} (\begin{bmatrix} I_x(x_i, y_i) & I_y(x_i, y_i) \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix})^2 \\ &= \begin{bmatrix} \Delta x & \Delta y \end{bmatrix} \begin{bmatrix} \sum_W (I_x(x_i, y_i))^2 & \sum_W I_x(x_i, y_i) I_y(x_i, y_i) \\ \sum_W I_x(x_i, y_i) I_y(x_i, y_i) & \sum_W (I_y(x_i, y_i))^2 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \\ &= \begin{bmatrix} \Delta x & \Delta y \end{bmatrix} C(x,y) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \end{aligned} \tag{2}$$

If λ_1, λ_2 are the eigenvalues of matrix C in equation (2), there are three cases to be considered. If both eigenvalues are small, it signifies a flat local auto correlation and a constant intensity image window. A high value in one of the eigenvalues and a low value in the other means an edge was detected. Two high eigenvalues identify a corner, a requirement for this approach. To locate corners, an image function (Harris and Stephens, 1988) was used, as in equation (3):

$$R = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2 = \det(A) - k \times trace^2(A) \quad (3)$$

R is positive in the corner regions, negative in the edge regions, and small in the flat region. Therefore, we created image function *R* using a threshold from the maximum *R* so that an appropriate number of corners are detected. The constant *k* in equation (3) indicates a tunable parameter. In the Harris function, 0 is the border between corner and edge (Stottinger, 2008) and the values between 0.04 and 0.15 are known as feasible values for *k*. Once the corners are detected, non-maximum suppression was applied in order to reduce each corner in N-neighborhood to a single pixel. To do this, an *n* by *n* maximum filter was created and compared to the corner image and only maximum values are extracted.

2.2. Match by cross correlation

According to Gonzalez et al. (2004), the best match of *w(x,y)* in *f(x,y)* is the location of the maximum value in the resulting correlation image when we treat *w(x,y)* as a spatial filter and compute the sum of products (or a normalized version of it) for each location of *w* in *f*. Therefore, we created 101 by 101 pixel windows for each corner in the referenced image, computed the correlation at each point in the unreferenced image, and extracted maximum values in both directions--from and to the unreferenced image. The correlation at a point can be computed as equation (4),

$$c(u,v) = \frac{\sum (u_i - \overline{u}) \times (v_i - \overline{v})}{\left[\sum (u_i - \overline{u})^2 \times \sum (v_i - \overline{v})^2\right]^{\frac{1}{2}}} \quad (4)$$

2.3. RANSAC for six-parameter transformation

RANSAC (Random Sample Consensus, Fischler and Bolles, 1981) extracts only inliers from samples by fitting data to a model with the most inliers of all models generated randomly *N* times (as in equation 6). A six-parameter transformation (Mikhail et al., 2001) such as equation (5) was used as a model.

$$\begin{aligned} X &= a_0 + a_1 x + a_2 y \\ Y &= b_0 + b_1 x + b_2 y \end{aligned} \quad (5)$$

where X, Y is the coordinate in the referenced image and x, y is the coordinate in the unreferenced image.

RANSAC with a six-parameter transformation is executed in six steps:

1. Since there are six unknown parameters (a_0, a_1, a_2, b_0, b_1, b_2), we need six equations to solve for the unknowns. Because one point pair gives two equations, we randomly select three points from matched point samples.

2. Calculate six parameters (a_0, a_1, a_2, b_0, b_1, b_2) from randomly chosen point pairs.

3. Compute every pair of points in the second image using a six-parameter transformation of the points in the first image.

4. Determine the sum of squared error between the estimated points and original points in second image.

5. If the error for each point is less than the tunable threshold (we used 30 square pixels), the points are inliers. Otherwise, they are outliers.

6. Repeat steps (1-5) N times as in equation (6)

$$N = \frac{\log(1-p)}{\log(1-(1-e)^s)} \tag{6}$$

where e is the probability that a point is an outlier, S is the number of points in a sample, N is the number of iterations, and p is the desired probability in a good sample.

The transformation that produces the most inliers is the best model and the inliers are the corresponding points in both images.

2.4. ArcObjects with JAVA

ArcObjects is a suite of libraries released by Environmental Systems Research Institute (ESRI) for software and tool development for their ArcGIS environment. After point pairs are matched in both the referenced and unreferenced images, the unreferenced image was rectified and referenced by ArcObjects' *Warp()* geoprocessor function (ESRI, 2009). Because the coordinates of the matched points in each image are nothing more than a pair of pixel index values relative to the edge of the image, they must be converted to some geographic coordinate system in order to be transferred from the referenced image to the unreferenced. ArcObjects' *Raster* class includes two methods *.toMapX()* and *.toMapY(),* which convert raster pixel coordinates to map coordinates within some geographic coordinate reference system. These map coordinates are then called by the *Warp()* function to transform the image into a fully georeferenced dataset. For transformation, it was found out that the first order polynomial like equation (7) sufficed:

$$\begin{aligned} X &= a_0 + a_1 x + a_2 y \\ Y &= b_0 + b_1 x + b_2 y \end{aligned} \tag{7}$$

where source (x, y) and target coordinates (X, Y) share the map's linear unit.

3. APPLICATION AND RESULT

Two early tests indicate the algorithm performs very well. In both, scanned, unreferenced aerial photo exposures were referenced against georeferenced photos of the same area in some other (near) year. Photos chosen for the tests included 1963 and 1971 images of the Purdue West Lafayette campus and 1939 and 1950 images of the Washington Street bridge area in Crawfordsville, Indiana. All images were in tagged image file format (tiff). These frame images were near vertical with terrain relief a small fraction of the flying height above ground.

Table 1. Test datasets for automatic georeferencing

Area	Referenced	Unreferenced	Time Gaps (yrs)
Purdue Campus	1971.tiff	1963.tiff	8
Crawfordsville	1939.tiff	1950.tiff	11

To start the test, the 1971 Purdue campus and 1939 Crawfordsville photos were manually georeferenced against 2005 IndianaMap Orthophotography to be used as reference data for the 1963 and 1950 unreferenced images, respectively. No attempt was made to reference images 1963 and 1950 directly against the 2005 IndianaMap Orthophotography because many features on the earth surface are suspected to have been changed. The corner points of each image were extracted and matched by the procedure explained above. The next step is to relate 1963 and 1971 campus images, and 1939 and 1950 Crawfordsville images. Figures 1 and 2 show, for each image pair, the matching points identified in *both* images. These matched pairs were then piped into a Java program built with ArcObjects and the unreferenced images were thereby rectified and referenced. The left image of Figure 3 shows the automatically referenced 1963.tiff under 1971.tiff. The right image of Figure 3 shows a detailed view, the right half of which is output from automated georeferencing (dark area) the 1963 image contrasted with the 1971 reference image on the left half. The roads are connected quite well at the border of both images. From the Crawfordsville tests (Figure 4), the left image shows the manually referenced 1939 data and the right image shows the automatically georeferenced 1950 image laid over the 1939 image. Figure 5 shows the detailed view of the comparison between the manually referenced 1939 image and the 1950 automated result. The river and forest features are well connected at the border in both images.

4. CONCLUSION AND FUTURE RESEARCH

Similar aerial photos can be georeferenced by the proposed automated methodology and the results are useful for applications that do not require high accuracy or as an initial approximation to a more rigorous procedure. There are several limitations that must be overcome before the process can be fully automated and operationalized.

Figure 1. Matched points from 1963 and 1971 aerial photo on Purdue Campus.

Figure 2. Matched points from 1939 and 1950 aerial photos of Crawfordsville, Indiana.

Figure 3. Left: Automatically referenced image (1963) overlaid by reference image (1971), Right: Detailed view of referencing (left light: 1971, right dark: 1963).

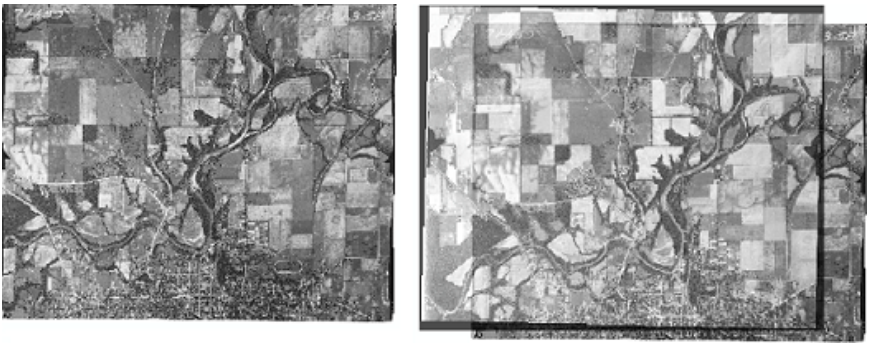

Figure 4. Manually referenced 1939 aerial photo (left) and 1950 automatically georeferenced photo (right, 30% transparent) overlaid on the 1939 photo.

Figure 5. Automated result from 1950, Crawfordsville, Indiana (left light), and manually referenced 1939 photo (right dark).

1. The extent and resolution of the photos must be similar.

2. There must be an adequate number of common, persistently identifiable features in both images.

3. The software must have access to an ArcGIS license (proprietary, expensive).

4. Imagery must be near vertical.

5. Terrain relief must be a small fraction of the flying height.

Because this procedure relies on identifying image objects that are not only unique but persist across different years' imagery sets, it is important that the source images contain man-made infrastructure. The angles, lines, and shapes of the built environment typically match well from year to year, but thus far it has proved important to find them in largely the same place on each image (e.g., in the upper left corner in 1963 *and* in 1971). We found that the correlation between window size and the number of corner points plays a very important role in the outcome of this procedure. Images with significantly dissimilar spatial footprints tend to cause false matching or failure of matching and negatively affect the ability to identify shared locations.

As this research progresses, attention will be paid to reducing the amount of similarity required between pairs of images. Additionally, interest point detectors other than the Harris corner detector will be tested in an effort to either choose a single best corner detector or to develop detector profiles that pair certain data circumstances with best-fit corner detectors. Hierarchical methods also seem to be well suited to this problem. We will likewise test non-proprietary transformation utilities in order to sever the dependence on costly, proprietary software.

REFERENCES

Derpanis, K. 2007. *The Harris Corner Detector*. Technical report, Department of Computer Science and Engineering, York University, Toronto, Ontario, Canada.

Environmental Systems Research Institute, Inc. (ESRI). 2009. Building ArcGIS Engine applications using Java-Concepts and Samples [online]. ESRI. Available from: http://resources.esri.com/help/9.3/arcgisengine/java/concepts_start.htm [accessed 24 February 2009].

Fischler, M., and Bolles, M.R. 1981. Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography. *Communications of the Association for Computing Machinery* 24:381-395.

Gonzalez, R.C., Woods, R.E., and Eddins, S.L. 2004. *Digital Image Processing using Matlab.* Upper Saddle River, NJ: Prentice Hall.

Harris, C., and Stephens, M. 1988. A combined Corner and Edge Detector. In *Proceedings of the 4th Alvey Vision Conference,* 189-192.

Indiana Geographic Information Council. 2009. IndianaMap. IGIC. Available from: http://www.igic.org/projects/indianamap/index.html [accessed 14 February2009]

Mikhail, E.M., Bethel, J.S., and McGlone, J.C. 2001. *Introduction to Modern Photogrammetry.* New York: John Wiley & Sons.

Moravec, H. 1980. *Obstacle avoidance and navigation in the real world by a seeing robot rover.* Technical report CMU-RI-TR-80-03, Robotics Institute, Carnegie Mellon University, doctoral dissertation, Stanford University.

Stottinger, J. 2008. *Local Colour Features for Image Retrieval. A More Distinct Coloured Scale-invariant Interest Point Detector.* Saarbrucken, Germany: VDM.