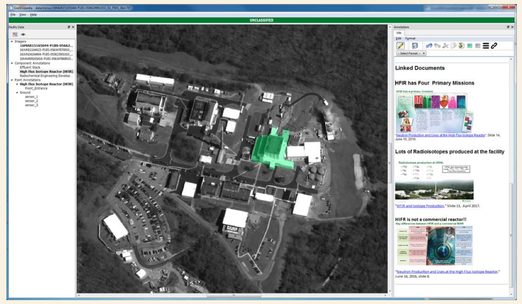
Automated Annotation of Satellite Imagery using Model-based Projections

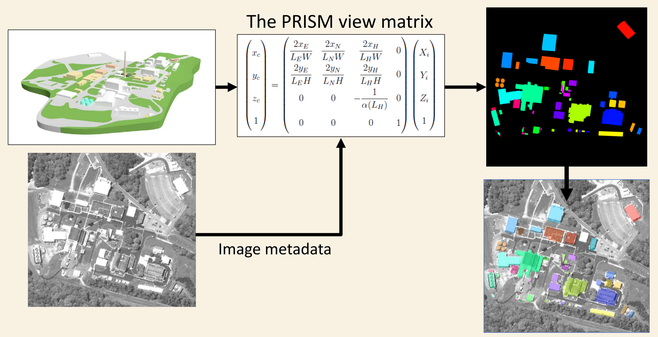
Randy S. Roberts*[1], John W. Goforth[1], George F. Weinert[1], Charles W. Grant[1], Will R. Ray[2], Brad Stinson[2] and Andrew Duncan[2]

[1]Lawrence Livermore National Laboratory, [2]Oak Ridge National Laboratory



Graphics in right pane courtesy of Oak Ridge National Laboratory, US Dept. of Energy. Facility imagery

- **GeoVisipedia automatically adds content to satellite imagery, regardless of viewing angle.**
- **SMEs add their insight. Knowledge integrates as we go along.**
- **Annotate once, annotated forever.**
- ***LLNL has patents and copyrights in the GeoVisipedia software. It is available to licensees and U.S. Government end users by request.***



The PRISM algorithm uses satellite metadata and a 3D model to create mask that is projected onto the satellite image

$$y_n = \frac{\sum_{i=1}^{20} a_i \rho_i(P,L,H)}{\sum_{i=1}^{20} b_i \rho_i(P,L,H)}$$
$$x_n = \frac{\sum_{i=1}^{20} c_i \rho_i(P,L,H)}{\sum_{i=1}^{20} d_i \rho_i(P,L,H)}$$

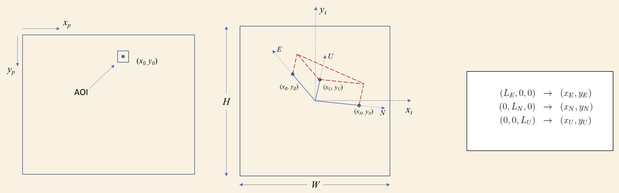
- One set of RPCs per image, 80 values in all
- Provide mapping between (latitude, longitude, height) and (column, row) in image.
- Bundle together a bunch of collection effects including orbital variations, curvature of earth, image distortions from imaging system, and many other collection artifacts.
- Very complex, lots of cross-coupling of terms
- Relatively accurate, but unsuitable for graphics pipelines

$$\sum_{i=1}^{20} a_i \rho_i(P,L,H) =$$
$$a_1 + a_2 L + a_3 P + a_4 H + a_5 LP + a_6 LH + a_7 PH + a_8 L^2 + a_9 P^2 + a_{10} H^2 + a_{11} PLH +$$
$$a_{12} L^3 + a_{13} LP^2 + a_{14} LH^2 + a_{15} L^2 P + a_{16} P^3 + a_{17} PH^2 + a_{18} L^2 H + a_{19} P^2 H + a_{20} H^3$$

Rational Polynomial Coefficients (RPCs) are a camera model for the satellite imaging system

The PRISM Algorithm is used to project model components onto an image

The first step is to create a right prism coordinate system in the center of the AOI image



$$(L_E,0,0) \rightarrow (x_E,y_E)$$
$$(0,L_N,0) \rightarrow (x_N,y_N)$$
$$(0,0,L_U) \rightarrow (x_U,y_U)$$

The next step is to use coordinates from the PRISM to calibrate an affine camera model

$$\begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = \begin{pmatrix} p_{11} & p_{12} & p_{13} & 0 \\ p_{21} & p_{22} & p_{23} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{pmatrix}$$

(a) Start with affine camera model

$$x_i = p_{11}X_i + p_{12}Y_i + p_{13}Z_i$$
$$y_i = p_{21}X_i + p_{22}Y_i + p_{23}Z_i$$

(b) Expand equations

$$(L_E,0,0) \rightarrow (x_E,y_E)$$
$$(0,L_N,0) \rightarrow (x_N,y_N)$$
$$(0,0,L_U) \rightarrow (x_U,y_U)$$

$$x_E = p_{11}L_E$$
$$y_E = p_{21}L_E$$
$$x_N = p_{12}L_N$$
$$y_N = p_{22}L_N$$
$$x_U = p_{13}L_U$$
$$y_U = p_{23}L_U$$

(c) Use correspondences to solve equations

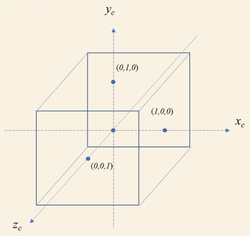
$$\mathbf{P}_a = \begin{pmatrix} \frac{x_E}{L_E} & \frac{x_N}{L_N} & \frac{x_U}{L_U} & 0 \\ \frac{y_E}{L_E} & \frac{y_N}{L_N} & \frac{y_U}{L_U} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(d) Enter results into affine projection matrix

Augment projection matrix for homogeneous coordinates

$$\begin{pmatrix} \frac{x_E}{L_E} & \frac{x_N}{L_N} & \frac{x_U}{L_U} & 0 \\ \frac{y_E}{L_E} & \frac{y_N}{L_N} & \frac{y_U}{L_U} & 0 \\ \frac{y_E}{L_E} & \frac{y_N}{L_N} & \frac{y_U}{L_U} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \rightarrow \left(\begin{array}{cccc} \frac{x_E}{L_E} & \frac{x_N}{L_N} & \frac{x_U}{L_U} & 0 \\ \frac{y_E}{L_E} & \frac{y_N}{L_N} & \frac{y_U}{L_U} & 0 \\ \hline 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right)$$

Another matrix scales the model to normalized device coordinates, allowing use of real-time graphics pipelines



Clipping cube used by graphics pipelines. Coordinates must be scaled to fit in this cube or they will be clipped when rendered.

$$s_x = \frac{2}{W}, \quad s_y = \frac{2}{H}, \quad s_z = \frac{-1}{\alpha L_U} \rightarrow \begin{pmatrix} \frac{2}{W} & 0 & 0 & 0 \\ 0 & \frac{2}{H} & 0 & 0 \\ 0 & 0 & \frac{-1}{\alpha L_U} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

These scaling factors scale the AOI image ($W \times H$) pixels into the clipping cube. The scaling factor α prevents tall objects from being clipped.

The view matrix is the product of the scaling and projection matrices

$$\begin{pmatrix} \frac{2x_E}{L_E W} & \frac{2x_N}{L_N W} & \frac{2x_H}{L_H W} & 0 \\ \frac{2y_E}{L_E H} & \frac{2y_N}{L_N H} & \frac{2y_H}{L_H H} & 0 \\ 0 & 0 & -\frac{1}{\alpha(L_H)} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \frac{2}{W} & 0 & 0 & 0 \\ 0 & \frac{2}{H} & 0 & 0 \\ 0 & 0 & \frac{-1}{\alpha L_H} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{x_E}{L_E} & \frac{x_N}{L_N} & \frac{x_U}{L_U} & 0 \\ \frac{y_E}{L_E} & \frac{y_N}{L_N} & \frac{y_U}{L_U} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Summary of PRISM Algorithm

- **It uses Rational Polynomial coefficients to create a view matrix that projects model components into a component mask**
  - A right prism is constructed in the AOI image, and the image coordinates of the right prism axes *m* meters away from the origin are used to calibrate an affine camera model.
  - Scaling is performed to scale model coordinates into normalized device coordinates.
- **PRISM exploits the fact that the camera (i.e., satellite) is very far from the scene, and that perspective changes are minimal. Structures in the AOI image can be treated as parallel projections.**
- **The view matrix created by PRISM renders models very quickly- tens of frames per second. Which means:**
  - GeoVisipedia can process changes in 3D facility models very quickly
  - GeoVisipedia can process large collections of imagery very fast

LLNL-POST-773539
This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

Lawrence Livermore National Laboratory