PNNL-22851

Pacific Northwest
NATIONAL LABORATORY
Proudly Operated by Battelle Since 1965

Methodology for Defining Gap Areas between Course-over-ground Locations

Visual Sample Plan

September 2013

JE Wilson

U.S. DEPARTMENT OF ENERGY

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

DISCLAIMER

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062;
ph: (865) 576-8401
fax: (865) 576-5728
email: reports@adonis.osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
email: orders@ntis.gov <http://www.ntis.gov/about/form.aspx>
Online ordering: http://www.ntis.gov

This document was printed on recycled paper.
(8/2010)

PNNL-22851

Methodology for Defining Gap Areas between Course-over-ground Locations

JE Wilson

September 2013

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99352

Summary

Finding all areas that lie outside some distance d from a polyline is a problem with many potential applications. This application of the Visual Sample Plan (VSP) software required finding all areas that were more than distance ***d*** from a set of existing paths (roads and trails) represented by polylines. An outer container polygon (known in VSP as a “sample area”) defines the extents of the area of interest. The term “gap area” was adopted for this project, but another useful term might be “negative coverage area.” The project required a polygon solution rather than a raster solution. The search for a general solution provided no results, so this methodology was developed.

Acronyms and Abbreviations

COG	course-over-ground
VSP	Visual Sample Plan

Contents

Figures

1.0 Background

Typically, a site being sampled to detect unexploded ordnance target areas is evaluated by having a sensor pass over the site in a series of parallel transects. Transects are spaced at a distance necessary to traverse and detect a target area of a certain size and shape and containing a minimum density of metallic anomalies spatially distributed in a predictable pattern.

However, some areas are difficult to access with complete coverage of such parallel transects. Additionally, the sites may have existing roads and paths that could be used as survey transects. Therefore, is the spacing and pattern of these existing paths sufficient to provide the necessary coverage to traverse and detect the target area, or are additional transects needed? If additional transects are needed, where are they needed?

A new augmentation method was developed in VSP for these sites. This document describes the process used to find the areas ("gap areas") where additional transects are needed.

2.0 Methodology

The general methodology is to:

- expand the existing paths (also known as course-over-ground or COG) into polygons
- find the sections of those polygons that are not contained inside of another polygon
- combine those sections into complete polygons.

The following is an overview of the methodology. The actual computer code is more complicated, but this overview presents enough detail to understand the process. Computer source code can be obtained from the author.

2.1 Step-by-step Overview

The first step is to remove duplicate or overlapping COG segments, as they will consume processing resources. COG segments are represented by rectangles. The width of the rectangle represents the width (or footprint) of the detector as it is moved over the ground. The length represents the forward progress of the detector during a certain interval of the survey (such as a one or two second period of time). The COG segment is defined by its end points (the points at the center of its ends) and its width. Figure 1 represents a series of COG segments.

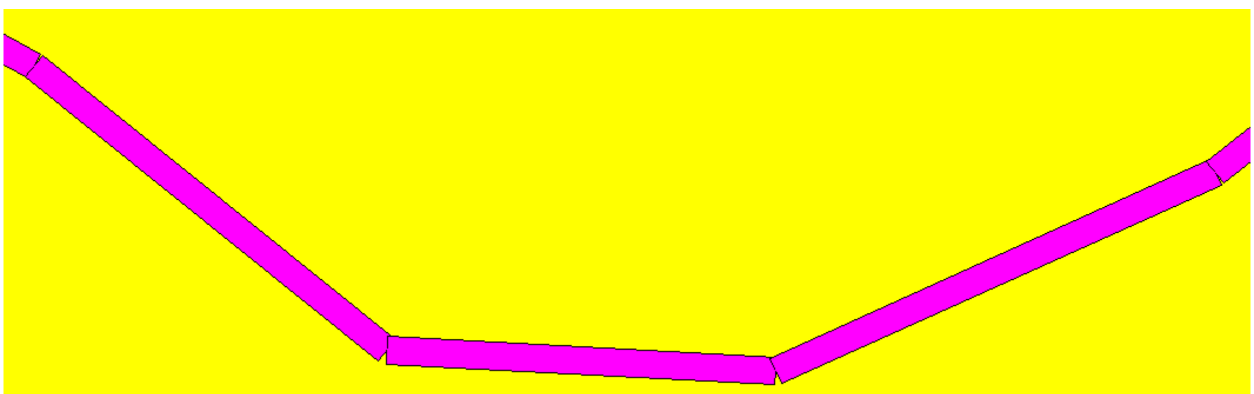

Figure 1. Series of COG Segments

Duplicate COG segments have the same beginning and ending points. A COG segment is considered to be overlapping when both end points of a segment are contained within or on the edge of another segment.

The next step is to string together the centerlines of connected COG segments into long polylines. It is more efficient to process a smaller number of polylines than a larger number of line segments. When no more segments remain that can be attached to the ends of the polyline, the polyline is added to a list and another polyline is started. When there are no remaining COG segments, this step is complete. Figure 2 shows a completed polyline that has been strung together from 25 COG segments.

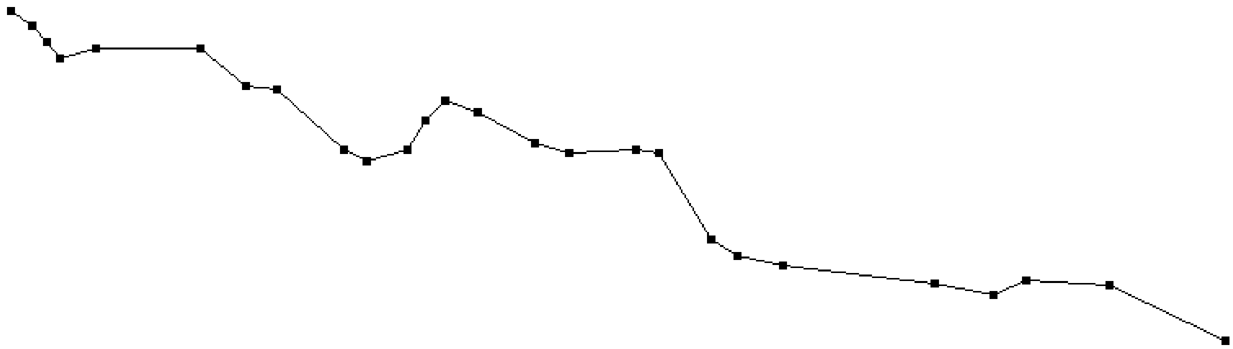

Figure 2. Polyline Assembled from 25 COG Segments

The next step is to expand each of the polylines in the list into a polygon. The resulting polygon represents the area that is within some distance ***d*** of the polyline. ***d*** is known in VSP as the ***buffer distance*** and represents the area within which a target area of radius ***d*** would be traversed by the polyline. During the polygon expansion, each segment is expanded according to the pattern shown in Figure 3. All of these segment polygons are joined together with a "union" operation to create the complete polygon. Figure 4 shows the complete polygon expanded from all of the polyline segments from Figure 2.

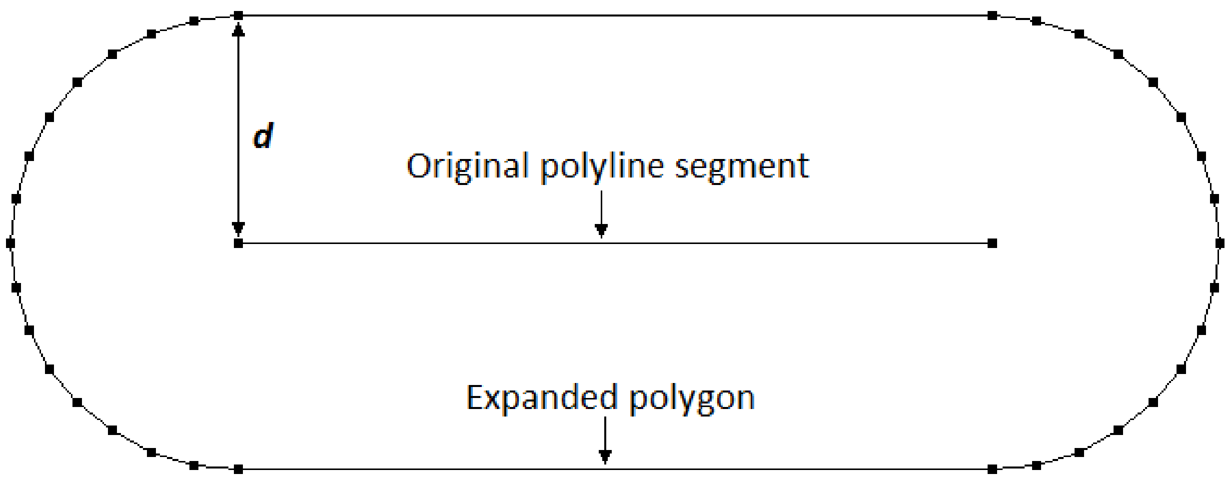

Figure 3. Expansion of a Polyline Segment into a Polygon

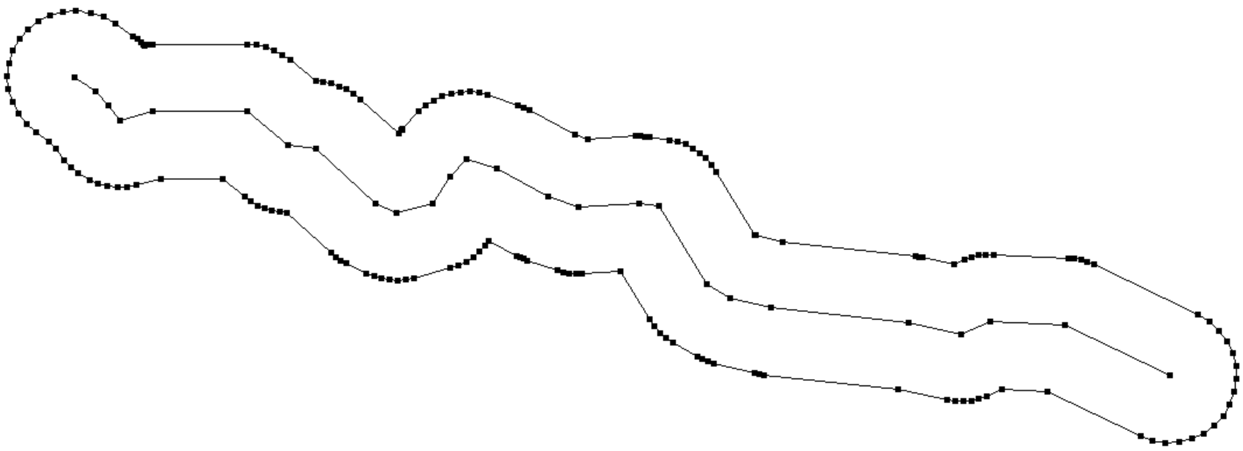

Figure 4. Complete Polygon Expanded from All of the Polyline Segments

Once complete, the polygon is added to a list. During expansion of a line, it is possible for one or more holes to be formed (Figure 5) where if the target area of radius ***d*** is centered in a hole, it will not be traversed. Each hole is added to the polygon list but marked as being a hole.

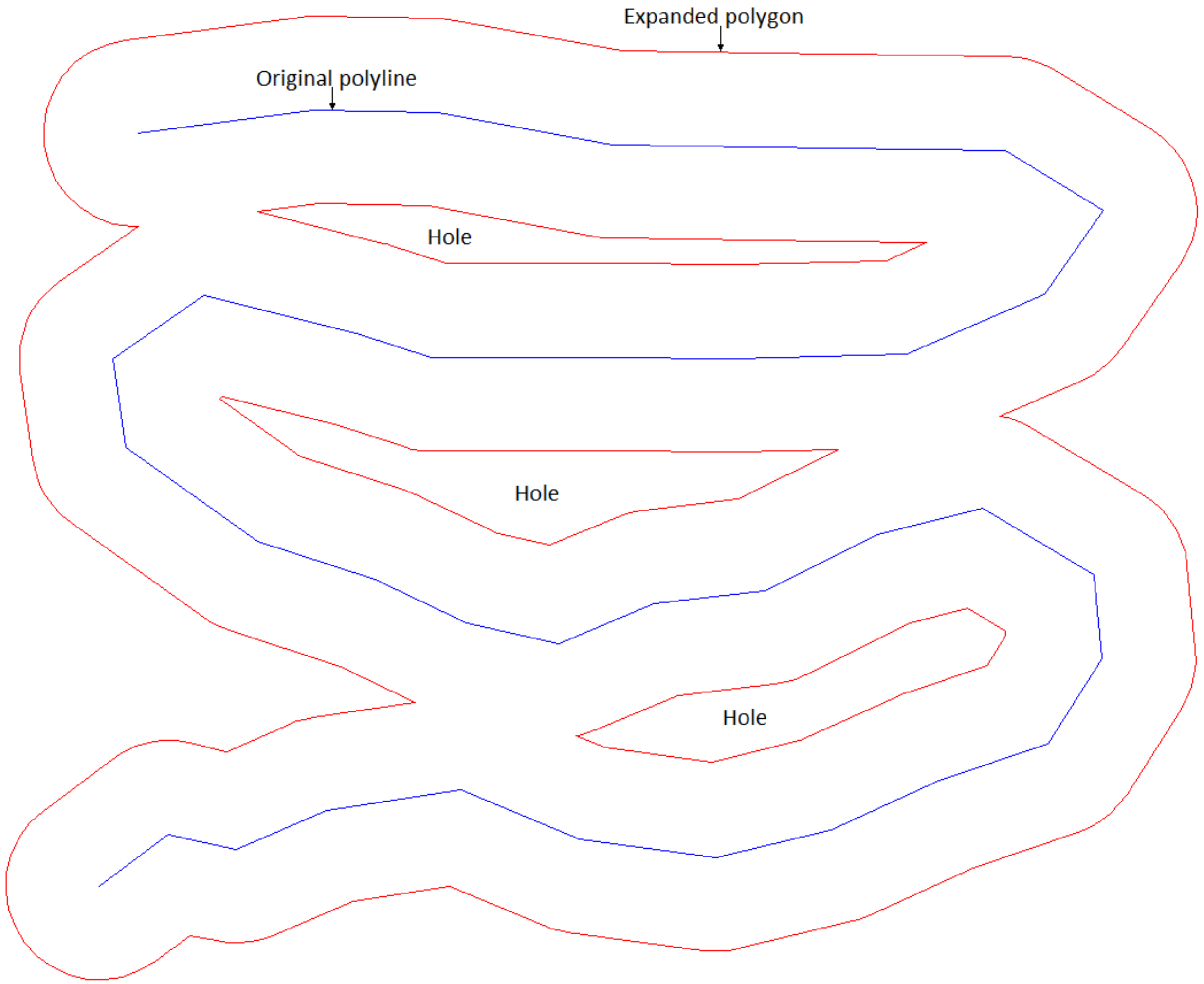

Figure 5. Holes Resulting from Polyline Expansion

Polygons have an orientation—the vertices are ordered in clockwise or counter-clockwise order when viewed from above. All of the polygons need to be oriented in the correct direction. Any polygon that is not oriented in the correct direction must have its vertices reversed. The sample area (the polygon that is the container for all of the COG data) must be oriented clockwise. Holes are also oriented clockwise. The other polygons are oriented counter-clockwise. Figure 6 shows a sample area (in yellow) along with the COG segments (in purple) and the resulting 27 expanded polygons (in blue).

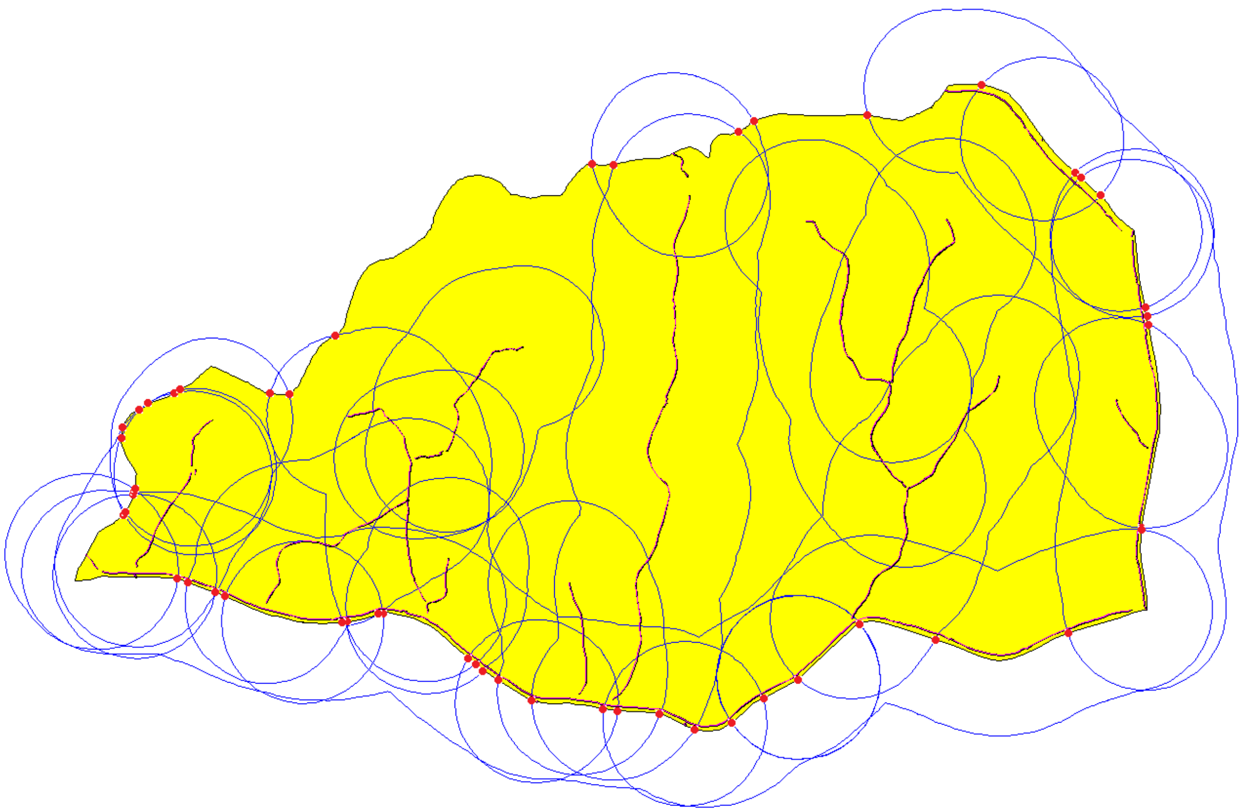

Figure 6. All Expanded Polygons

The next step is to intersect the edge of the sample area with all of the polygons. The points of intersection are added to the sample area and the polygons. The intersection points are also stored in a sorted list to speed up processing later. The red dots in Figure 6 indicate the intersections between the sample area edge and the polygons. Next, intersections between every polygon and every other polygon are found. As before, these intersection points are added to the polygons themselves and to the sorted list. It is also necessary to find intersections between exclusion areas and polygons. Exclusion areas are polygon holes in the sample area where access is not permitted. Exclusion areas cannot be part of the final gap areas. Exclusion areas are illustrated in Figure 7 by the white areas. Intersection points are added to the exclusion areas and polygons and to the sorted list.

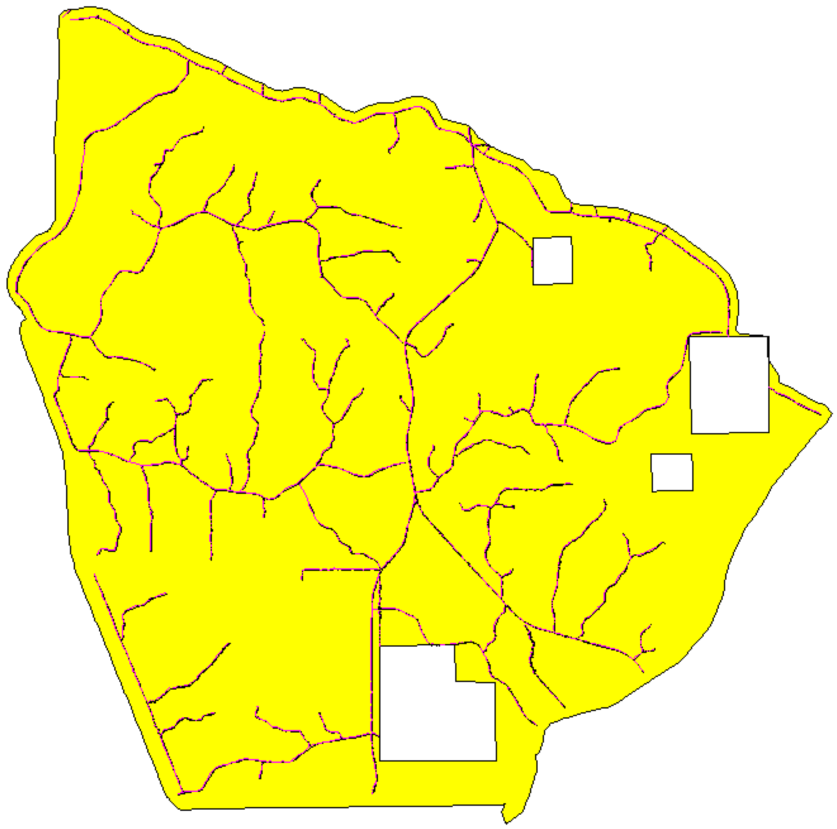

Figure 7. Exclusion Areas

The next step is to use the sorted list of intersection points to construct polylines that are sections of the sample area, expanded polygons and exclusion areas, and represent edges of the final gap areas. Each section (set of vertices between two intersection points) is examined to determine whether it is a potential gap section. The sample area, expanded polygons, and exclusion areas each have slightly different rules to determine whether they are a *gap section* (a list of vertices that define a section of a gap area):

- sample area sections must not be inside of any expanded polygon
- exclusion sections must not be inside of any expanded polygon; if it is inside an expanded polygon that is marked as a hole, it is saved in a special in-hole list (described below)
- expanded polygon sections must be inside of the sample area and must not be inside of any other expanded polygon or exclusion area; if it is inside an expanded polygon that is marked as a hole, it is saved in the special in-hole list.

Sections in the special in-hole list are tested next. Depending on whether the section is part of (the edge of) a hole or not part of (the edge of) a hole, the sections have different rules to determine whether they are a gap section:

- sections that are part of a hole must not be inside of any expanded polygon except its own parent (the expanded polygon that originally contained the hole)
- sections that are not part of a hole must be inside of a hole and must not be inside any other expanded polygon other than holes and parents of holes.

Sections of polygons that conform to these rules are saved in a section list. Figure 8 shows the vertices (black dots) of the sections from Figure 6 that conform to these rules and are therefore considered to be edges of gap areas.

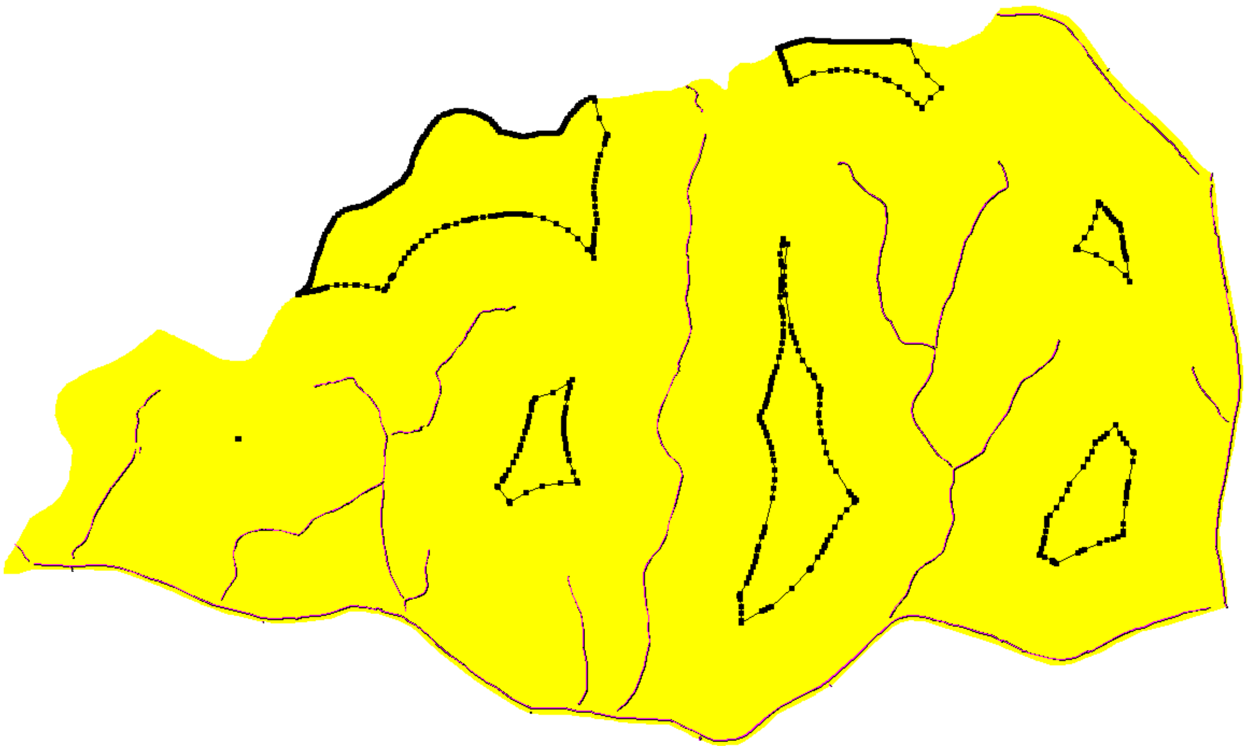

Figure 8. Vertices of Polygon Sections that are Edges of “Gap” Areas

Creating gap areas is simply a matter of joining the gap sections together to make complete polygons. The first section is removed from the list and becomes the basis for a gap area. Another section is found that connects to the end of the gap area. It is removed from the list and added to the gap area. This is where the orientation of the sample area, expanded polylines and holes (completed above) becomes important so that points are added in the correct order. If the added section closes the polygon (connects back to the first point), the gap area is complete and it is added to the list of gap areas. If the section list is not empty, the process is repeated for another gap area.

Finally, any hole that does not intersect any other expanded polyline or exclusion area is added in its entirety to the list of gap areas. Figure 9 shows the final gap areas in green.

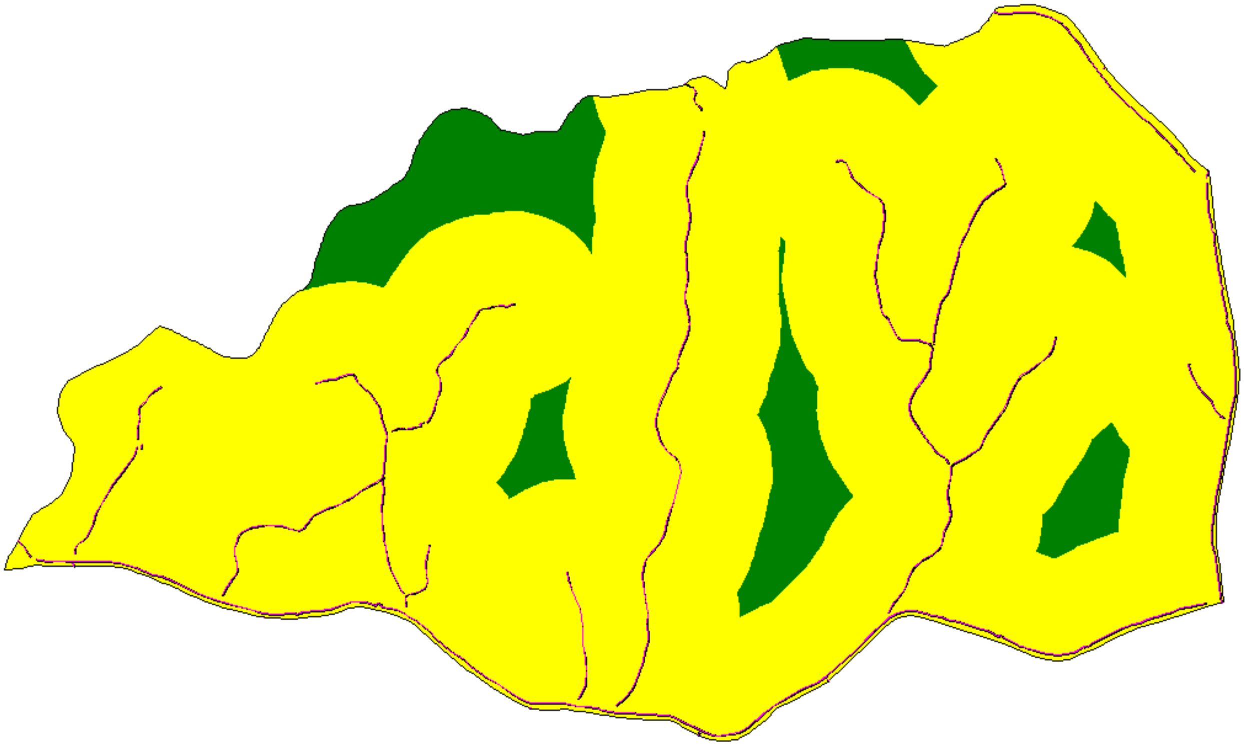

Figure 9. Final Gap Areas

3.0 Conclusions

This methodology is straight-forward and can be applied to any set of polylines, regardless of complexity. Output is a list of fully-formed polygons.

For sections of code where individual polyline segments are compared to each other, computer run time can increase with the square of *n* (where *n* is the number of polyline vertices). However, using sorted lists of coordinates can greatly reduce run time.

Line simplification of the input polyline paths can reduce the overall number of vertices and greatly reduce the run time needed for polygon expansion and results in simpler polygons (polygons with fewer vertices). Simpler polygons require less time to check for intersections with other polygons. Using sorted lists of coordinates to avoid unnecessary intersection checks also greatly reduces run time.

Pacific Northwest
NATIONAL LABORATORY

Proudly Operated by Battelle *Since 1965*

U.S. DEPARTMENT OF
ENERGY

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99352
1-888-375-PNNL (7665)
www.pnnl.gov