Using Dynamic Segmentation to Create Address Ranges for Street Centerlines

Abstract: The ability to use geocoding services greatly enhances any GIS, but this often requires a street centerline dataset with address range attributes. The ideal address ranges would reflect the addresses as they actually exist and not based on an address model. The ranges on connecting segments would not have any range gaps from one segment to the next. Creating these ideal ranges manually would be very cumbersome. This paper discussed in detail how the use of dynamic segmentation can be used to take a layer of existing address points and assign address ranges to a street centerline layer.

The Address Ranges

GIS is a technology that is continually growing, and the use of GIS is becoming more and more widespread. As GIS use increases, other aspects of GIS are being utilized. One enhancement that an organization can find useful is geocoding. The ability to use geocoding services in any GIS is dependent upon the available data. One commonly used data source for geocoding is street centerlines. These street centerlines must have some kind of address range attributes assigned to them in order for them to serve as a geocoding data source. Assigning address ranges to street centerlines is a hurdle that must be faced before the benefits of geocoding can be seen.

Addressing plans can be used to assign address ranges to street centerlines. ArcGIS comes with developer sample that assigns these ranges based on a few different plans. This is a definite good start. It is quite likely, however, that addresses in a town or county do not follow any given plan. Taking it one step further to assign ranges that reflect reality is a lot more difficult. In the case of the Town of Hilton Head Island, a coverage of address points exists. This address point layer contains a point at every location in the town that has an address. Creating a relationship between those address points and the street centerline arcs would provide a great foundation to build address ranges from.

The first step is to determine which arc each point should be associated with. ArcINFO's NEAR command is useful to determine which arc each point is closest to. Using the LOCATION option, the output of this command is a new coverage with all the same address points with additional attribute information indicating the COVER# of the street centerline arc that the point was closest to and the X and Y coordinate along that arc where it is closest.

Of course it is possible that an address point may be closer to a road that it is not addressed off of. An ArcINFO AML script can be used to loop through each street name. For each name it creates a subset of the streets and address points with that street name. It performs a NEAR command on those subset coverages and transfers the unique street coverage COVER-ID to the address points. Then it appends the resulting point coverages into one. The resulting coverage is basically the original address coverage with a few additional attributes, including the COVER-ID of the street centerline and the X and Y coordinate along the closest arc.

Now that I know which arc each point is closest to, I have a start towards assigning address ranges to the street centerlines. It would be very easy from here to write a script to loop through all the street centerline arcs, grab the related address points with the matching street COVER-ID, and put the highest and lowest address number into address range fields for that arc. This would produce a street centerline coverage with useable address ranges for geocoding. The problem is that these ranges only represent the addresses that exist now. If an arc perhaps does not

have a point that is closest to it, then that arc has no address range on it. For example, in a string of 5 arcs, suppose the second and fourth arc have no address points closest to it and, therefore, have no address range on them. Figure 1 illustrates this idea. It would be nice to have the ranges filled in for those arcs to have more of a conceptual

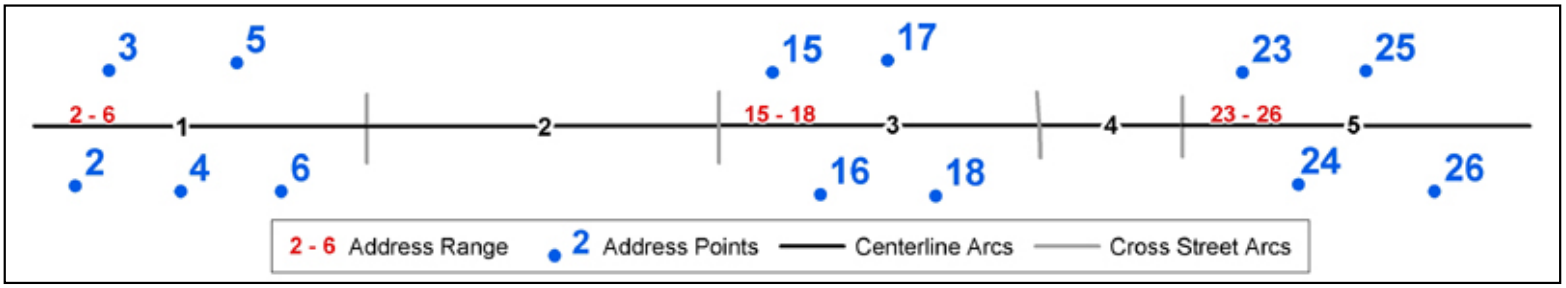

Figure 1

address range. These conceptual address ranges would provide the location of future addresses and therefore would not have to be modified as addresses are created.

There are few things to consider when designing the conceptual address ranges. If the first arc has a range of 2 – 6, and the third arc has a range of 15 – 18, what range should the second arc have? A range of 7 – 14 would make sense. Consider the scenario shown in figure 2. The distance from address number 6 to the end of the first arc is fairly significant. There is enough room to the end of the road segment to build a few more houses thus creating

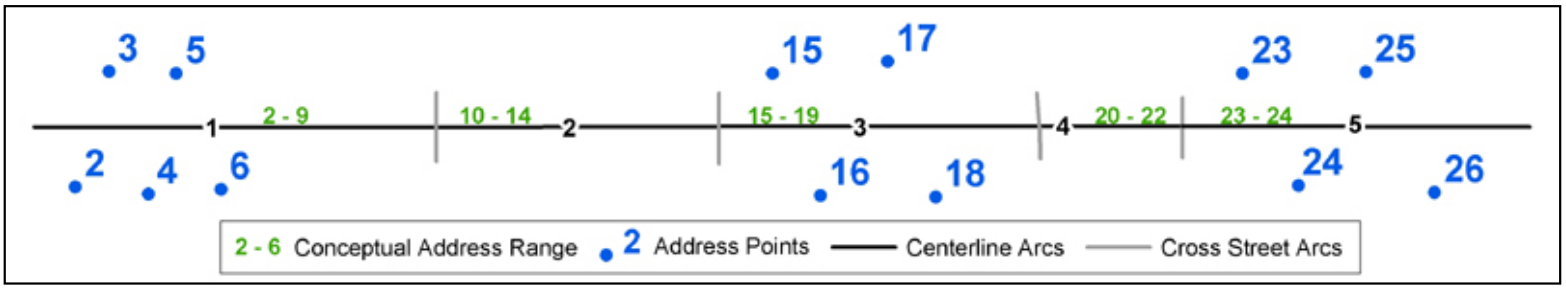

Figure 2

more addresses on that segment. Perhaps on that first arc it would be better to increase the range from 2 – 6 to 2 – 9. That would put the range of the second arc starting at 10. So now the idea of these conceptual address ranges is to not only assign ranges to the untouched arcs, but also modify the ranges of touched arcs to better reflect the future.

The solution we decided on is to distribute the ranges between the known addresses based on the distance between them. In Figure 3, if the length of segment A is 10 and segment B is 25 and segment C is 5, the total distance between address number 6 and 15 is 40 units. Segment A is 25% of the total distance, so I want the range to be increased 25% of the difference in address numbers. 15 – 6 = 9 and 25% of 9 is 2.25. So I want to increase

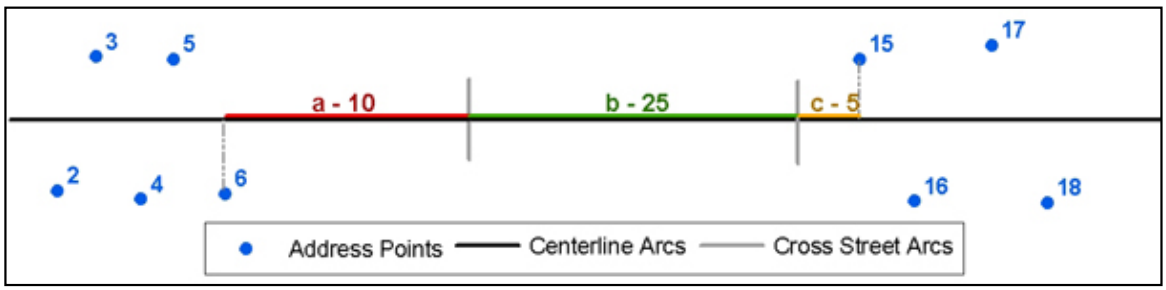

Figure 3

the range of the first arc by 2. The resulting range for the first arc is 2 – 8. The second arc starts with 9. The problem is now an issue of where along the road segment a given address occurs. The length of each arc is known in the attribute table, but is it possible to determine a distance from the endpoint along the arc to a given location? Dynamic Segmentation is the key.

Dynamic Segmentation and Linear Referencing

Dynamic Segmentation or Linear Referencing allows for locating spatial information relative to a linear feature. Whereas a point coverage is a method to locate spatial information that utilizes XY coordinates. Suppose there is an accident on a highway. The location of the accident can be represented by an XY coordinate, or it can be represented as occurring at mile marker 25 along Interstate Highway 75. This relative location includes the linear feature (or route) that it occurs on and the measure along that route. These relative locations are called events. With dynamic segmentation, events can be created to represent attributes that occur at given locations along routes.

The next step in the process is to create events out of the address points. Using the X and Y coordinate attributes added to the address point coverage during the NEAR command, I create a new coverage. The points in this new coverage are located at these X and Y coordinates, and I transfer the attributes from the points in the original address coverage to the new coverage. The red points in Figure 4 illustrate the resulting coverage. I use this arc address coverage to create events. In order to create events, there must be routes for the events to occur on.

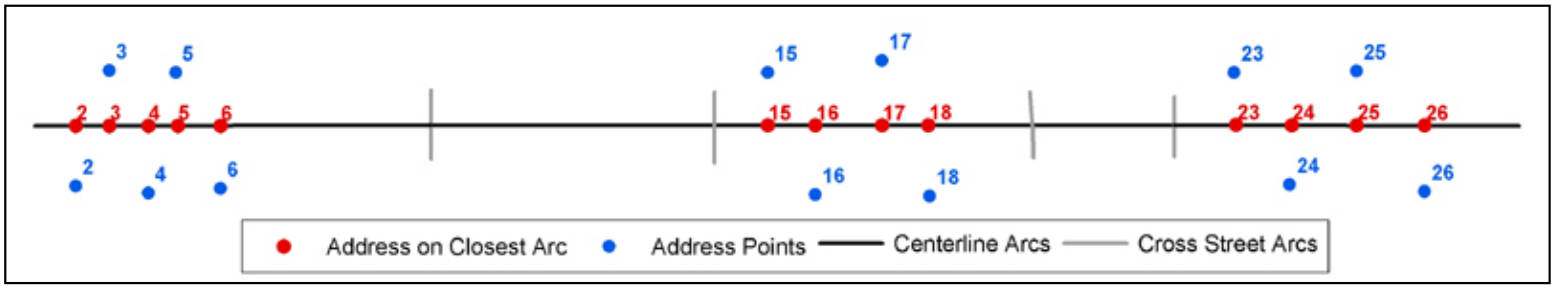

Figure 4

I have decided to create two route feature classes. The first has one route for each arc, and the second has one route for each street name. There are a few different ways to create routes, but in the interest eliminating as much human interaction, I choose to use the MEASUREROUTE command. This command takes an entire coverage of arcs and groups them together into routes based on an item value in the attribute table. To create the arc routes I use the unique COVER-ID as the grouping item, therefore creating one route for each arc. The street name routes are created using the street name item to group them into one route for each street name.

To create events the ADDROUTEMEASURE command will take points from a point coverage and determine which route it occurs on and the measure value along the route where it occurs. Using this command with the arc address coverage and routes, the resulting event table provides the information needed to locate the different addresses along the different arcs. ADDROUTEMEASURE should be done twice to create two different event tables, one for arc routes and one for the street name routes. Then the addresses are linearly referenced along the roads in two different ways.

When using MEASUREROUTE, the measure values are calculated for each route feature, and the starting node for the measures is determined by the software. The desired measures will increase in the direction of increasing

address numbers. In order to ensure that this in fact occurs we can compare the address event measures to the address numbers. A script is necessary to loop through the data and determine the highest and lowest address for each street name. If the highest address number has a measure value that is lower than the measure of the lowest address number, the route measures need to be flipped. The script can identify the routes that need to be flipped and remeasure the routes. Once this is complete the event table needs to be recreated to reflect the new corrected measure values for the address events.

Now the route directions correspond to the direction of increasing address numbers. The same idea should be considered for the arc direction. The address range of an arc will be interpreted in a geocoding service using the arc direction. The addresses near the beginning of the range will be close to the start node of the arc and those near the end of the range will be close to the end node. It only makes sense that the arc direction be in the direction of increasing address numbers as well. We can utilize the route directions to modify the arc directions.

An ArcINFO coverage route system is comprised of two feature classes. Route and Section feature classes are created each having attribute tables. The sections reference the arcs that compose the routes. There are two attributes in the section attribute table, F-POS and T-POS, that define the starting and ending position of the section in terms of percentage along the arc that comprises it. If the section is comprised of an entire arc the F-POS and T-POS values will be 0 and 100. The section comprises from 0 to 100 percent of the arc. If the arc direction is opposite the direction of the route the F-POS will be 100 and the T-POS will be 0 indicating that the section of the route goes from the end node to the start node of the arc. In the case of the street name routes, the goal is to get the arcs comprising the routes to go in the same direction as the route itself. The section attribute table also has an attribute ARCLINK#, which is the value of the COVER# of the arc that comprises the section. I can set up a relate then between the arc attribute table and the section table based on the COVER# and the ARCLINK#. Using the relate, I can select the arcs that have sections with an F-POS of 100 and flip them. Once the arcs are flipped, I must use REMEASURE to recalculate the information in the route and section attribute tables.

Using the Events

Now that the events have been created, the measure values are correct, and the arcs are going in the right direction, we have everything ready to use to properly assign the address ranges. In my script, I have a complex series of loops and queries to traverse every street and populate the address ranges. A few things are done in the beginning of the script to prepare the loops. I want to ensure that as it loops through the streets it is going in order from lowest to highest measure. The script exports the arc attribute table into a new street table. The measures of the arcs are stored in the section table, so the script transfers the FMEAS and TMEAS values from the section table to the street table. Then the street table is sorted on FMEAS. I also want to ensure that the address events are accessed in order of the measure values, so the street name route address event table is sorted on measure. The last thing is that the script only cares about the highest and lowest address number that exist on an arc. Figure 5 illustrates these needed events. This is where the arc route address events become useful. The script loops through individual arc

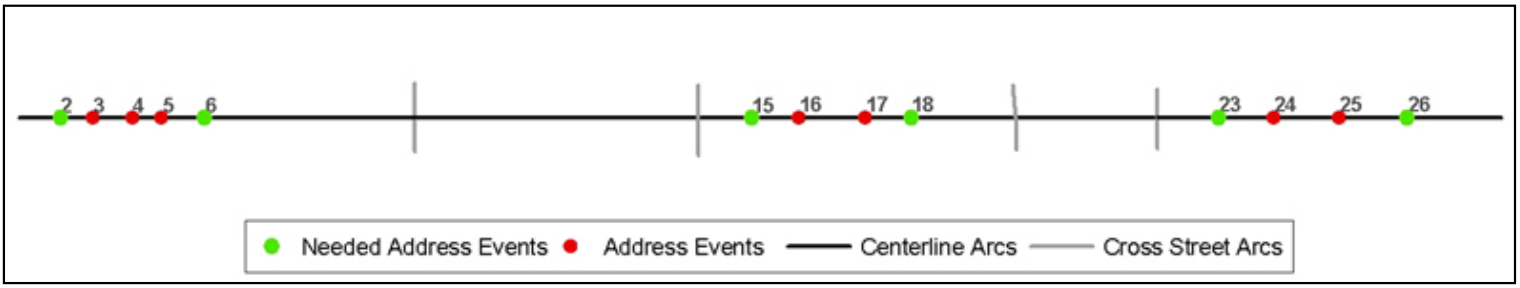

Figure 5

routes and finds the events in the arc route event table that occur on that route. For each route the script determines which event is the highest and which is the lowest. Those events are flagged in a MINMAX attribute. Once that is complete the MINMAX attribute is transferred to the events in the street name route event table. Lastly, a table is generated containing one record for each street name, to enable looping through each street name.

The looping begins by going through each street name. For each street name, it selects the min and max events that occur on that street name. For each event (in order from lowest to highest measure), it keeps track of the measure value and the address number for both the current event and the previous event. It selects the arc segments whose FMEAS and TMEAS encompass the previous measure value through to the current measure value. It then loops through each of those selected arcs (in order from lowest to highest measure). It calculates the difference in measure values between the current and previous event and determines the percentage of that distance that occurs on that arc. That percentage is multiplied by the difference in address numbers. The result is added to the address number and put into the range field. The script keeps track of this new range number to add onto it for the next arc until it reaches the last arc for those events. The loop increments to the next event until there are no more events on that street name. The resulting address ranges represent more than the existing addresses and allow for very useful geocoding solutions.

Dynamic segmentation and linear referencing provides a solution to this problem. It presents the needed information to determine where a given address occurs along a road. The information can then be used to manipulate the data to create these conceptual address ranges.

Kelly Bigley
GIS Analyst
Town of Hilton Head Island
1 Town Center Court
Hilton Head Island, SC 29928
843-341-4797
843-842-2264 fax
kellyb@ci.hilton-head-island.sc.us