Exploiting Online Sources to Accurately Geocode Addresses

Rahul Bakshi
University of Southern California
Information Science Institute
4676 Admiralty Way
Marina del Rey, CA 90292
rbakshi@isi.edu

Craig A. Knoblock
University of Southern California
Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292
knoblock@isi.edu

Snehal Thakkar
University of Southern California
Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292
thakkar@isi.edu

ABSTRACT

Many Geographic Information System (GIS) applications require the conversion of an address to geographic coordinates. This process is called geocoding. The traditional geocoding method uses a street vector data source, such as, Tigerlines, to obtain address range and coordinates of the street segment on which the given address is located. Next, an approximation technique is used to estimate the location of the given address using the address range of the selected street segment. However, this provides inaccurate results since the approximation assumes that properties exist at all possible addresses and all properties are of equal size. To address the inaccuracy of the traditional geocoding approach, we propose two new methods for geocoding using additional online data sources. The first method, the uniform-lot-size method, uses the number of addresses/lots present on the street segment to approximate the location of an address. The second method, the actual-lot-size method, takes into consideration the lot sizes on the street segment and the orientation of the lots as well. Moreover, we describe an implementation of these methods using an information mediator to obtain information about actual number of lots and sizes of the lots on the streets from various property tax web sites. We geocoded an area covering 13 blocks (267 addresses) using all three methods. Our evaluation shows that the traditional method results in an average error of 36.85 meters, while the uniform-lot-size and the actual-lot-size methods result in the average error of 7.87 meters and 1.63 meters, respectively.

Categories and Subject Descriptors

H.2.8 [**Information Systems**]: Database Management – Database Applications – *Spatial databases and GIS.*

General Terms

Algorithms, Performance, Experimentation

GIS'04, November 12–13, 2004, Washington, DC, USA.

Keywords

Geospatial data integration, Geocoder, Mediator, Information integration

1. INTRODUCTION

As we move to the next generation of the Internet, the World Wide Web is turning into a set of data sources that can be queried. The challenge lies in using these data sources to solve existing problems. One such challenge is to accurately geocode street addresses. Geocoding is the process of obtaining the geographic coordinates (latitude/longitude) of a given address. The software which does this is called a geocoder. Accurate geocoding is important for a variety of applications, such as environmental health studies to demarcate areas with potential hazardous exposure in relation to where people live [3]. Accurate geocoding is also important in applications that align vector data with imagery [5] and for urban rescue and recovery operations According to a report by the US Federal Geographic Data Committee (FGDC), the geographic location is a key feature of 80-90% of all government data [11]. Therefore it is important to have geocoding methods that provide results with maximum accuracy. The existing approaches to geocoding provide values which have a significant error in them as they rely on approximation techniques based on the assumption that for a street segment all the addresses within a given address range exist for the street segment. This error in the values can be appreciably reduced if property-related information from various online data sources is integrated with the existing geocoding techniques. In this paper we describe two approaches to utilize various online data sources to obtain more accurate geographic coordinates for a given address.

The remainder of the paper is organized as follows. Section 2 describes the traditional geocoding method and shows why the traditional method of geocoding results in inaccurate geographic coordinates. Section 3 describes our approaches to perform more accurate geocoding by utilizing property information from various property tax web sites. In Section 4 we describe an implementation of our approaches for more accurate geocoding using an information mediator. Section 5 describes the evaluation of our approach. Section 6 discusses the relevant related work and Section 7 concludes the paper by recapping the key ideas and describing some directions for future work.

```
Step 1: currentaddress ← parse the given address to get street address
Step 2: Query street data source:
            fromlatitude, fromlongitude, tolatitude, tolongitude ← coordinates of end points
            fromaddrleft, toaddrleft, fromaddrright, toaddrright ← address ranges on either side of the street
Step 3: If currentaddress % 2 == fromaddrleft % 2
            toaddress ← toaddrleft
            fromaddress ← fromaddrleft
        Else
            toaddress ← toaddrright
            fromaddress ← fromaddrright
Step 4: rel_loc ← ABS((toaddress - currentaddress)/(toaddress - fromaddress))
Step 5: Calculate the latitude and longitude based on the ratio
            currentlatitude ← tolatitude - (rel_loc * (tolatitude - fromlatitde))
            currentlongitude ← tolongitude - (rel_loc * (tolongitude - fromlongitude))
```

Figure 1. Algorithm for address range method

2. TRADITIONAL APPROACH TO GEOCODING

The traditional geocoding method uses a street vector data source to obtain address range and coordinates of the street segment on which the given address is located. Next, it uses an approximation technique to estimate the location of the given address using the address range of the selected street segment. The main sources of street data that the existing services use are commercially available products such as the TIGER/Line data from the Bureau of Census[1], Navtech data from Navigation Technologies[2], GDT data from Geographic Data Technology[3], etc. These data sources provide geographic coordinates (latitude and longitude) of street segments. They also provide possible address ranges on each side of the street between the two sets of coordinates for the given street segment. These data sources provide a good estimate, but do not give exact information about the number of addresses actually present on the street segment. For example, if the address “625 Sierra St, El Segundo, CA, 90245” is queried in the TIGER/Line data source, it returns a tuple which has the end-points of the street segment on which the address is located and the possible addresses. For this address, the range on the left side of the street is 601 – 699 and on the right side of the street is 600 – 698[4]. This information suggests that there are 50 address lots present on either side of the street segment. However there are only 7 addresses present on either side of this particular street segment. Furthermore, there is no information about the size of each address/lot in these data sources.

2.1 Existing Method

The existing method uses information present in a typical street data source to interpolate an address in relation to the end points of the street segment to which it belongs. Figure 1 gives the algorithm for this traditional approach, which we call the address-range method.

As the first step in the algorithm, we parse the given address into individual tokens representing the street address, street name, city, state and zip. Based on this information, at the second step, we query the street data source and obtain the street segment to which the current address belongs. We get the end point coordinates of this segment (fromlatitude, tolatitude, fromlongitude, tolongitude) and also the address range present on either side of the street (fromaddrleft, toaddrleft, fromaddrright, toaddrright). Next, we find which side of the street the given address belongs to. This is done by checking to see if the given address is even or odd. If the given address is odd then, we select the side of the street that contains the odd addresses. Once the side of the street to which the current address belongs is decided we find the relative location of the given address (the address to be geocoded) on the street segment by taking ratio of number of addresses before the current address with the total number of addresses on the street segment on the selected side, assuming that all possible addresses exist on the segment[5]. For example, if the street data source returns addresses 601 – 699 present on the left side, which is also the side where the current address exists, this method would assume that 50 addresses are present on the left side of the street. It then calculates the relative location of the current address in the range of 50 addresses. The relative location calculated is then interpolated between the street end points to get the geographic coordinates of the current address (step 5).

2.2 Limitations of This Method

This method has some limitations. First, it assumes that all the lots/addresses specified by the street data source in the address range actually exist. Second, it assumes that all these lots are of equal size. And lastly, it does not take into account the dimension occupied by the corner lots which actually may be a part of the other intersecting street segments. Figure 2 shows the geocoded locations for the addresses on a block.

[1] http://www.census.gov/geo/www/maps

[2] htttp://www.navteq.com

[3] http://www.geographic.com

[4] The left and right are the directions taken in the sense when one travels from the ‘from’ coordinates to the ‘to’ coordinates in the street data sources.

[5] For simplicity we do not consider addresses ending with fractional number such as 1225 ½. Those are typically handled by ignoring the fractional component.

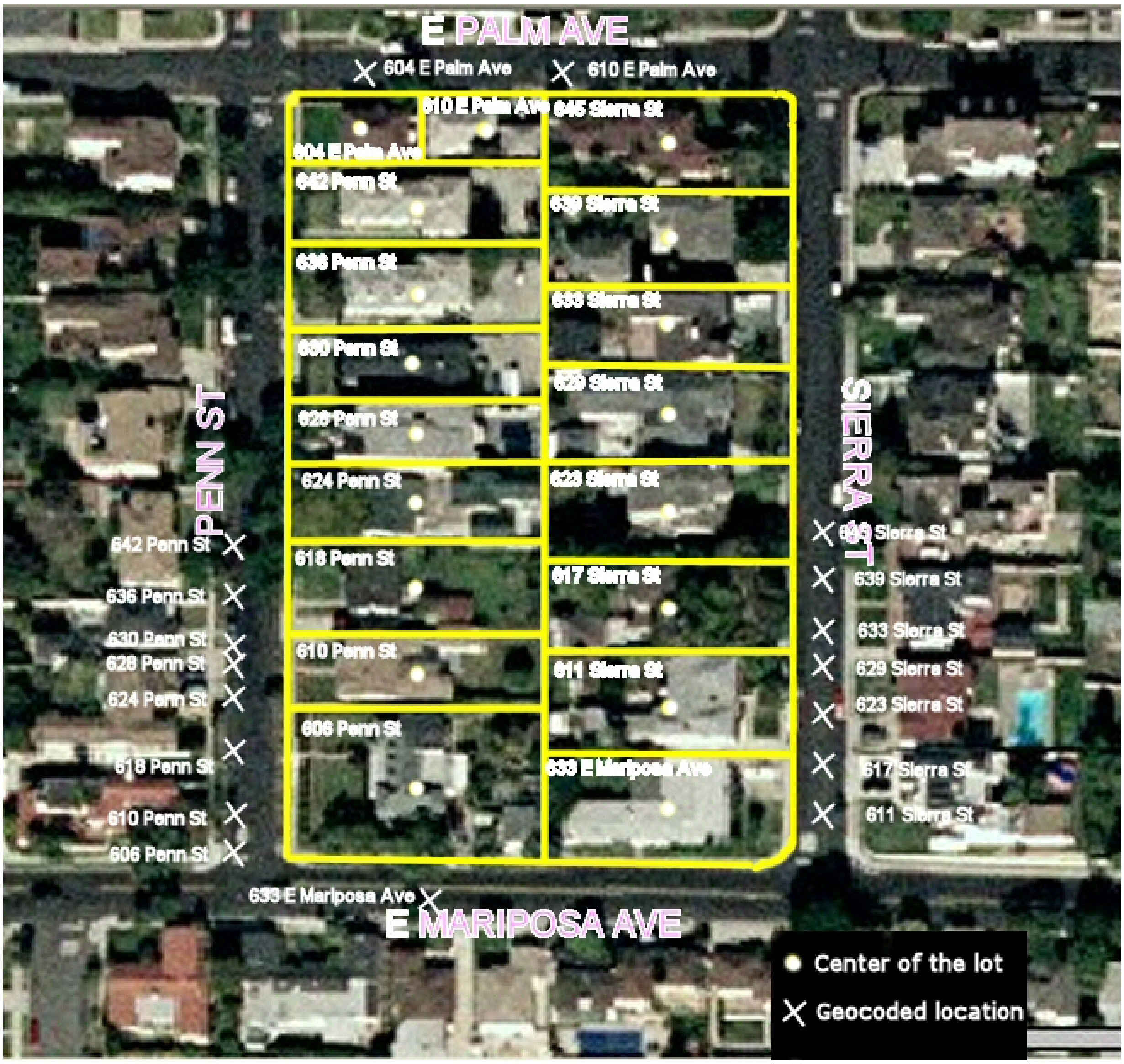

Figure 2. Geocoded locations using the traditional method

Consider the example of finding the location of a nonexistent address in Los Angeles County: “625 Sierra St, El Segundo, CA, 90245”. We used this address to query a number of the popular mapping services on the Internet. All of these services returned the location of this nonexistent address. The mapping services we used were Yahoo! Map[6], Geocode[7], MapQuest[8] and MapPoint[9]. Thus the present method can be misleading at times, as in this case when it gives the location of a nonexistent address. Consider another example. The address “645 Sierra St., El Segundo, CA, 90245” is present on the intersection of Sierra St. and E. Palm Ave. However, all of these mapping services display this address somewhere on the middle of the Sierra St segment to which this address belongs. The apparent reason is that the data source that they use returns a result which has addresses 601 to 699 present on the side of the street where 645 Sierra St is located. This range implies that there are 50 lots present on the selected side of the street. In reality, there are seven lots present on this street segment. So when the interpolation is done by taking 50 addresses, it leads to results with a large error.

These observations validate our claim that the existing services for geocoding do not check for validity of addresses and approximate the given address based on the information about the end-point of the street and an approximation of the address range present on the street. The observations also imply that the existing services do not consider the size of the lots on the street.

[6] http://maps.yahoo.com

[7] http://www.geocode.com

[8] http://www.mapquest.com

[9] http://www.mappoint.com

```
Step 1:  currentaddress ← parse the given address to get street address
Step 2:  Query street data source:
             fromlatitude, fromlongitude, tolatitude, tolongitude ← coordinates of end points
             fromaddrleft, toaddrleft, fromaddrright, toaddrright ← address ranges on either side of the street
Step 3: If currentaddress % 2 == fromaddrleft % 2
             toaddress ← toaddrleft
             fromaddress ← fromaddrleft
        Else
             toaddress ← toaddrright
             fromaddress ← fromaddrright
Step 4: Query the property tax data source for the selected side:
             nb ← number of lots between fromaddress and currentaddress
             na ← number of lots between currentaddress and toaddress
Step 5: Calculate the length of the street segment obtained in step 2 using the distance formula
             street_len ← SQRT($(fromlatitude - tolatitude)^2$ + $(fromlongitude - tolongitude)^2$)
Step 6: Assume uniform size for all lots and divide 'street_len' by the number of lots
        present on the street + 1: The additional lot is added to account for the corner lot that may be on an intersecting
        street
             lotsize ← street_len / (nb + 1 + na + 1)
Step 7: Divide the lot size obtained in Step 6 by two, to get the increment factor 'offset'
             offset ← lotsize / 2
Step 8: Calculate the slope θ (theta) for the street segment
             θ ← $\text{Tan}^{-1}$ ((tolongitude - fromlongitude) / (tolatitude - fromlatatitude))
Step 9: Calculate the latitude of the currentaddress
             currentlatitude ← fromlatitude + (offset + nb * lotsize + offset) * Cos (θ)
             currentlongitude ← fromlongitude + (offset + nb * lotsize + offset) * Sin(θ)
```

Figure 3. Uniform lot-size method

3. EXPLOITING ONLINE SOURCES TO IMPROVE ACCURACY

More accurate geocoding can be performed by utilizing the number of properties on a given street and their dimensions. Our approach for increasing the accuracy of geocoding takes into account these facts and shows a remarkable improvement in the geocoded values. We call the new geocoder Columbus[10]. This section discusses our methods to perform accurate geocoding. Section 3.1 describes the uniform lot-size method, which takes into account the number of lots on the street. Section 3.2 describes the actual lot-size method which also takes into account the lot dimensions and orientations in addition to the number of lots on the street.

The main reason why the address-range method produced results with significant error is because it infers the numbers of houses/lots present on the street segment from the street address range. It is seldom the case that all the addresses specified in the street data source actually exist. If the exact number of addresses existing on a street segment is known, it can be used to significantly improve the accuracy of geocoding. Furthermore, if the orientation and sizes of the lots on the corner of the street are known, it would result in further improvement in accuracy.

3.1 Uniform Lot-size Method

The idea behind the uniform lot size method is to use the actual number of houses/lots existing on the street to calculate the latitude and longitude of the current address. This information can be obtained from the property tax websites of different regions. The property tax websites provide the number of address/parcel lots present on the street. Some property tax websites also provide the dimensions of each of the lots present in their region. Figure 3 shows the algorithm for the uniform lot size method.

The first three steps of this algorithm are similar to the previous algorithm described in section 2. At the fourth step, we query the property tax data source to get the number of houses before (nb) and after (na) the current address on the street segment. The fifth step calculates the length of the street segment. To do this, we use the Euclidian distance formula. This formula is valid for planar surfaces. Since the segments on the street data source are very small compared to the size of the earth, we can use this formula without significantly affecting our results. In the next step, we calculate the size of each lot.

At this stage, we face a challenge of deciding on which street the lots on the corners of the street segment belong. A given street segment can have at most two corner lots. To generalize, we assume that out of the two corner lots one belongs to the given street and the other is a part of an intersecting street. The corner lot which belongs to the intersecting street however does occupy a dimension on the given street segment. It needs to be accounted for when we estimate the average lot size on the street. Thus at the sixth step, we divide the street length by the number of houses

[10] The geocoder is named Columbus after the famous traveler Christopher Columbus.

```
Step  1:  currentaddress ← parse the given address to get street address
Step  2: Query street data source:
                    fromlatitude, fromlongitude, tolatitude, tolongitude ← coordinates of end points
                    fromaddrleft, toaddrleft, fromaddrright, toaddrright ← address ranges on either side of the street
Step  3: If currentaddress % 2 == fromaddrleft % 2
                    toaddress ← toaddrleft
                    fromaddress ← fromaddrleft
          Else
                    toaddress ← toaddrright
                    fromaddress ← fromaddrright
Step  4: Query street data source:
                    fromlatitudeP, fromlongitudeP, tolatitudeP, tolongitudeP ←
                              end points of the street segments that form a block
                    relYcoord, relXcoord, relBlocklen_meters, relBlockwid_meters ←
                              coordinates and size of the block
Step  5: If block not rectangular, perform Uniform lot-size geocoding
Step  6: Query the property tax data source and get the dimensions of  each of the lots
          present on the block
Step  7: Calculate the actual dimensions of the streets in the block based on the data from the
          source used in Step 2 and Step 4 using the Great Circle Distance Formula:

          EarthRadius = 6378137.0
          street_len_meters  ← EarthRadius * (Cos$^{-1}$(Sin(tolatitude) * Sin(fromlatitude) + Cos(tolatitude)
                              * Cos(fromlatitude) * Cos(tolongitude - fromlongitude)))
Step  8: There are 2 possible assignments for each corner lot and there are 4 corner lots. So,
          there are 16 possible combinations of assignments of corner lots in a given rectangular
          block.
                    orientations[1..16] //array with all 16 possible orientations
                    error[1..16] //error in street length for each orientation
                    For i ← 1 to 16 do:  //for all 16 orientations
                              estimated_len_meters = Σ length of all lots on the street in orientations[i]  +
                                        Σ depth of corner lots (if present in orientation[i])
                              For k ← 1 to 4
                                        errorstreet[k] = ABS(street_len_meters of street[k] –
                                                  estimated_len_meters of street[k])

                              error[i] ← Σ errorstreet[1..4]
Step 9:   Select the orientation with minimum error in step 9
                    j = indexOf(min(error), error)   // find element in error with minimum error
Step 10: Based on the assignement selected, obtain the center point of the lot to be geocoded
                    relXcoord, relYcoord ← orientation[j]
Step 11:Convert the relative position in Step 11 to absolute latitude and longitude
                    latitude  = toplat – ((relYcoord)*(toplat – bottomlat) / (relBlocklen))
                    longitude = leftlon + ((relXcoord)*(rightlon – leftlon) / (relBlockwid))
```

Figure 4. Algorithm for actual-lot-size method

present on the street plus the extra corner lot. Since at this stage, it is not known to which end of the street the corner lot exists, we start with an offset which is half the average calculated lot size on the street segment. The slope of the street segment (θ) is then calculated in the eighth step. Once the slope is known, the projection of latitude and longitude are obtained from the trigonometric functions sine and cosine respectively. We add another offset value so that we get to the center of the lot.

3.2 Actual Lot-size Method

There are two main reasons to improve further from the uniform lot size method. First, it assumes that all the lots on a street segment are equal in size (widths). Second, the problem of locating the corner lot is not solved. In the actual lot size method we find out the exact orientation of the corner lots. However, this method currently assumes that the addresses to be geocoded are part of a rectangular block.

Figure 4 gives the algorithm for the actual-lot-size method. Similar to the previous two approaches, steps 1 through 3 obtain the segment of street to which the address belongs and all the relevant attributes of that street segment. The fourth step gets the coordinates of the end points of the other streets that form the block. After obtaining the coordinates of all the four corners of the block, in the fifth step we determine if the block is rectangular. If it is, the algorithm proceeds to the next step, else it reverts to uniform lot size geocoding method. Next, we query the property tax source and get the dimensions of all the lots on the block. The seventh step calculates the actual lengths of street segments that form the block. We use the great circle distance formula to calculate the length.

For a rectangular block, there are four corner lots and each of these could belong to either of the two streets which intersect on the corner. This leads to sixteen possible combinations for the orientation of the corner lots for the given block. In step eight, we calculate an error value which is the difference between the sum of the actual lengths of the street segments and the calculated length of the street for a particular orientation. This error is calculated for all possible sixteen orientations for the block. The orientation which gives the least error value is selected as the one for the current block. Thus at the end of step nine, the exact layout of the block and the orientations of all the four corner lots for the block are known. Once the layout of the block is known, we obtain the center point for the lot to be geocoded in terms of relative coordinates for the block. The relative coordinates are with respect to the top left corner of the block being the origin (0,0). These relative coordinates are converted into latitude and longitude values by a simple mapping function. Step eleven shows a sample mapping function which assumes that the latitude of the block increases as we move from south to north and the longitude increases as we move from west to east. A trivial change is needed for blocks which do not have this type of layout. Thus we obtain the latitude and longitude for the lot.

4. AUTOMATICALLY SELECTING ONLINE SOURCES USING A MEDIATOR

The algorithms discussed in Section 3 assume that there exists a single source for obtaining property data. However, there are over two thousand property tax assessment districts in the US and each of these regions organizes the data in different manner. Different property tax sites may provide different types of data, e.g. some sites may provide dimensions of the property while the others may not. The coverage of different property tax sites may be limited to a city, county, state or some other aggregate region. The challenge is to determine the appropriate property tax sources for geocoding a given address. Similarly, street information for different regions may be available from different data sources as well. In Columbus, we utilize the Prometheus mediator [22, 23] to provide a unified query interface to different property tax data sources as well as different street data sources.

The Prometheus mediator is a data integration system that builds on previous work data integration [8, 9, 12, 13, 15, 16]. Traditionally, data integration systems have a set of domain relations on which the users can specify queries. The task of the data integration system is to translate a query into a set of queries on the source relations using a domain model that relates source relations to domain relations. In order to utilize Prometheus mediator for geocoding we have to perform three tasks: (1) model web services as source relations, (2) determine a set of domain relations, and (3) define relationships between different source relations and domain relations.

The first step of defining a domain model is to describe all available web services as source relations. The available web services for Columbus are a set of property tax web services generated from various property tax web pages, a set of street information web services such as, Tigerlines street information web service, and a set of services to approximate the location of the given address on the given street segment. Each web service is modeled as a source relation with binding restrictions, i.e. in order to obtain information from the source relation, the values of all attributes with binding restrictions must be provided. The input attributes of the web services are modeled as attributes in the corresponding source relations with binding restrictions. For example, the Tigerlines service that accepts the streetaddress, city, state, and zip attributes and returns streetname, streettype, frlat, frlon, tolat, tolon, zipl, zipr, fraddr, fraddl, toaddr, toaddl attributes is modeled as the following source relation. The '$' symbol before an attribute denotes attribute with a binding restriction.

LAProperty($sa, $ci, $st, $zi, frlat, frlon, tolat, tolon, fename,
fetype, zipl, zipr, fraddr, fraddl, toaddr, toaddl)

Once we have modeled all available web services as source relations, we need to determine a set of domain relations for Columbus. We define PropertyTax and Street domain relations in Columbus as virtual relations representing all available property tax and street information web services respectively. The three different methods to geocode given addresses are modeled as the following three domain relations that user's can query: (1) AddressRangeGeocoder, (2) UniformLotSizeGeocoder, and (3) ActualLotSizeGeocoder.

Now that we have modeled all available web services as data sources and determined domain relations, we need to define a set of rules to relate the source relations with the domain relations. Traditionally, data integration systems have utilized three approaches to relate domain relations to available source relations. In a Global-As-View (GAV) approach, a domain expert defines the domain relations as views over the available source relations. In the Local-As-View (LAV) approach, available source relations are defined as views over the domain relations. In the GAV model query reformulation is straight-forward. However, adding additional data sources in the GAV model may require modifying definitions of all domain relations. In LAV one only needs to add the view definition for the new source to add additional source. Duschka [6] and Levy et.al. [17] have described algorithms to translate user queries into set of source queries using the LAV approach. More recently, there has been another approach termed GLAV [7] that allows user to combine the advantages of both the GAV and LAV approaches. The Prometheus mediator supports all three approaches. In Columbus we use the GLAV approach as it would be complicated to encode complex geocoding algorithms in the domain model using the Local-As-View model and adding new web services may require changing entire domain model if we use the Global-As-View model.

As shown in Figure 5 we define some example property tax web services and street web services as views over the PropertyTax and Street domain relations, respectively. When the mediator receives a user query, the mediator inverts these definitions to compute PropertyTax and Street domain relations. By modeling these web services as views over the domain relations we simplify the process of adding new property tax web service or street information web service. We discuss more about adding new property tax web services or street information web services in Section 4.1. Moreover, we can clearly define the coverage provided by different web services as order constraints in the

R1: **LAProperty**(street, city, county, state, zip, before, after, fraddr, fraddl, toaddr, toaddl):-
PropertyTax(street, city, county, state, zip, fraddr, fraddl, toaddr, toaddl,
before, after, lotwidth, lotdepth) **^**
(state = "CA") **^** (county = "Los Angeles")

R2: **NYProperty**(street, city, county, state, zip, before, after, fraddr, fraddl, toaddr, toaddl):-
PropertyTax(streetaddress, city, county, state, zip, fraddr, fraddl, toaddr, toaddl,
before, after, lotwidth, lotdepth) **^**
(state = "NY")

R3: **TigerLinesCA**(streetaddress, city, state, zip, frlat, frlon, tolat, tolon, fename, fetype, zipl,
zipr, fraddr, fraddl, toaddr, toaddl):-
Street(streetaddress, city, state, zip, frlat, frlon, tolat, tolon, fename, fetype,
zipl, zipr, fraddr, fraddl, toaddr, toaddl) **^**
(state = "CA")

R4: **NavTechLinesNY**(streetaddress, city, state, zip, frlat, frlon, tolat, tolon, fename, fetype, zipl, zipr, fraddr, fraddl,
toaddr, toaddl):-
Street(streetaddress, city, state, zip, frlat, frlon, tolat, tolon, fename, fetype,
zipl, zipr, fraddr, fraddl, toaddr, toaddl) **^**
(state = "NY")

Figure 5 Example Source Descriptions for Columbus

rules. For example, consider the rule R1 that defines LAProperty web service as a view over PropertyTax domain relation. The rule R1 states that LAProperty web service provides property information for only properties located in "Los Angeles" county in the state of "California". The mediator can utilize the provided order constraints to reduce the number of requests sent to each web service.

As shown in Figure 6, the three domain predicates representing different geocoding methods are defined as views on the available source relations or other domain relations. For example, the UniformLotSizeGeocoder domain relation is defined as a join over Street and PropertyTax domain relations and the UniformLotApproximation source relation.

ActualLotSizeGeocoder and AddressRangeGeocoder implement the actual-lot-size method and the address-range method for geocoding, respectively. Once we have defined the domain model, the Prometheus mediator can accept requests to geocode different addresses using different methods. For example, to geocode the address “123 Main St, Los Angeles, CA 90007” using the uniform-lot-size method, we would specify the following query to the mediator.

Q1(lat, lon) :- UniformLotSizeGeocoder(strtaddr, city, county,
state, zip, lat, lon)^
(strtaddr = “123 Main St”)^
(city = “Los Angeles”)^
(state = “CA”)^
(zip = “90007”)

4.1 Adding New Property Tax Web Services

New property tax web sites and street information web sites are becoming available everyday. As more and more property data sources become available online, their descriptions can be incrementally added to the mediator’s domain model to expand the coverage of Columbus. Therefore, one of the key design considerations in Columbus is to make it easy to add new web services to the domain model. Adding new property tax or street information web services to Columbus’ domain model is a easy task as it uses GLAV approach. For example, if new county data (say Fresno) is available online, it is defined by the following source relation:

Fresno($streetaddress, $city, $county, $state, $zip, before,
after, fraddr, fraddl, toaddr, toaddr)

After modeling the new web service as a source relation, we define the new source relation as a view over the PropertyTax domain relation.

Fresno(streetaddress, city, county, state, zip, before, after,
fraddr, fraddl, toaddr, toaddl):-
PropertyTax(streetaddress, city, county, state, zip,
fraddr, fraddl, toaddr, toaddl, before, after) ^
(state = "CA") ^
(county = "Fresno")

Once we add this source description to the domain model, Columbus can utilize the Fresno county property tax web service when we query Columbus to geocode an address located in Fresno county.

5. RESULTS

We present empirical results for the new geocoding methods. For the evaluation, we selected a region in the City of El Segundo consisting of 13 blocks (267 addresses). We selected this region due to the availability of the conflated [20] TIGER/Line data source and the satellite imagery. We utilized techniques developed in our previous work [4] to obtain the conflated TIGER/Lines . The actual geographic coordinates of the lots were calculated from the assessors map for the area as the center points of the lots. The online property related data for the region is converted into XML web services using the Fetch agent

```
R1: AddressRangeGeocoder(strtaddr, city, state, zip, lat, lon):-
        Street(strtaddr, city, state, zip, frlat, frlon, tolat, tolon, strtname, strttype, zipl, zipr, fraddr, fraddl,
              toaddr, toaddl)^
        BlockApproximation(sa, fraddr, fraddl, toaddr, toaddl, frlat, frlon, tolat, tolon, lat, lon)

R2: UniformLotSizeGeocoder(strtaddr, city, county, state, zip, lat, lon):-
        Street(strtaddr, city, county, state, zip, frlat, frlon, tolat, tolon, strtname, strttype, zipl, zipr, fraddr,
             fraddl, toaddr, toaddl)^
        PropertyTax(strtaddr, city, county, state, zip, fraddr, fraddl, toaddr, toaddl, before, after, lotwidth,
            lotdepth)^
        UniformLotApproximation(frlat, frlon, tolat, tolon, before, after, lat, lon)

R3: ActualLotSizeGeocoder(strtaddr, city, county, state, zip, lat, lon):-
        Street(strtaddr, city, county, state, zip, frlat, frlon, tolat, tolon, strtname, strttype, zipl, zipr, fraddr,
              fraddl, toaddr, toaddl)^
        PropertyTax (strtaddr, city, county, state, zip, fraddr, fraddl, toaddr, toaddl, before, after, lotwidth,
             lotdepth)^
        LotApproximation(strtaddr, strtname, strttype frlat, frlon, tolat, tolon, before, after, lat, lon)
```

Figure 6 Domain Rules for Columbus

platform[11]. To measure the error in meters, we use the Sinnott's Formula [21] assuming the average radius of Earth as 6,378,137 m. The error is measured as the driving distance on the street from the geocoded location to the actual location. Figure 7 shows the actual locations for addresses on these streets and their geocoded locations from both methods.

Out of the 267 addresses, all three methods of geocoding were applicable on 208 addresses and we present our results based on these addresses. The remaining addresses were excluded, since the actual-lot-size method requires that the block formed by the intersection of the streets is rectangular. Thus it does not handle certain irregular layouts of the blocks. The Actual lot-size method could be extended to handle these new geometric shapes and is a part of the future work for this research. A more detailed description of the results is available in [2].

Table 1 gives a comparison of the error values of the new methods over the traditional method. The uniform-lot-size method had an average improvement of 79 percent over the traditional method while the Actual lot-size method showed an average improvement of 91 percent.

The large error in the address range method is due to the reasons already discussed in section 2.2. The Uniform lot-size method reduces the error significantly. The Actual lot-size method performs even better than the uniform lot-size approach. The error in this last method can be attributed to the fact that the Tiger/LINES do not perfectly align with the street. The actual lot-size method gives us the center points of the lots. To measure the error, we projected these points on the street.

The average response time for the query was 410 ms for the Address-range method, 511 ms for the Uniform lot-size method and 3415 ms for the Actual lot-size method. The property related data was cached locally for these experiments and was not retrieved in real-time. The Actual lot-size method is more expensive because of the computation of the corner lots and there is a considerable room for optimizing this further and is a part of future work for this research.

Table 1. Comparison of error values

	Address-range	Uniform lot-size	Actual lot-size
Average Error	36.85	7.87	1.63
Standard Devia	20.49	9.92	1.47
Minimum Error	0.87	0.07	0.03

6. RELATED WORK

The research work related to this paper can be broadly classified into two categories. The first category of research work is in the area of measuring the inaccuracy of available street data and existing geocoding web sites [3, 14, 19], while the second category of research is in the area of geo-spatial data integration using data integration systems .

All geocoding algorithms rely on some street vector data to identify the location of the given address. A study by Ratcliffe [19] about accuracy of address-range files (similar to the Tigerline files) in Australia showed that out of 20,000 addresses geocoded using address-range data, less than 5% of geocoded points were on the correct lot. The two key factors behind the error are inaccuracy of the address-range data and inaccuracy introduced by the approximation performed by the geocoding algorithm. In this paper, we reduce the inaccuracy introduced by the geocoding algorithm by 89% by utilizing online data sources. In past work our group has introduced automated conflation techniques to align street vector data with satellite imagery [5] . For the experiments with Columbus, we used the conflated TIGER/Line data obtained from these techniques. Cayo and Talbot [3] and Krieger et. al. [14] have studied the accuracy of commercial geocoding sites for addresses in the U.S.A. Both studies support our claims that the traditional geocoding methods may provide geographic coordinates for inaccurate addresses and the geographic coordinates provided for existing addresses by the traditional methods are often very inaccurate.

[11] http://www.fetch.com

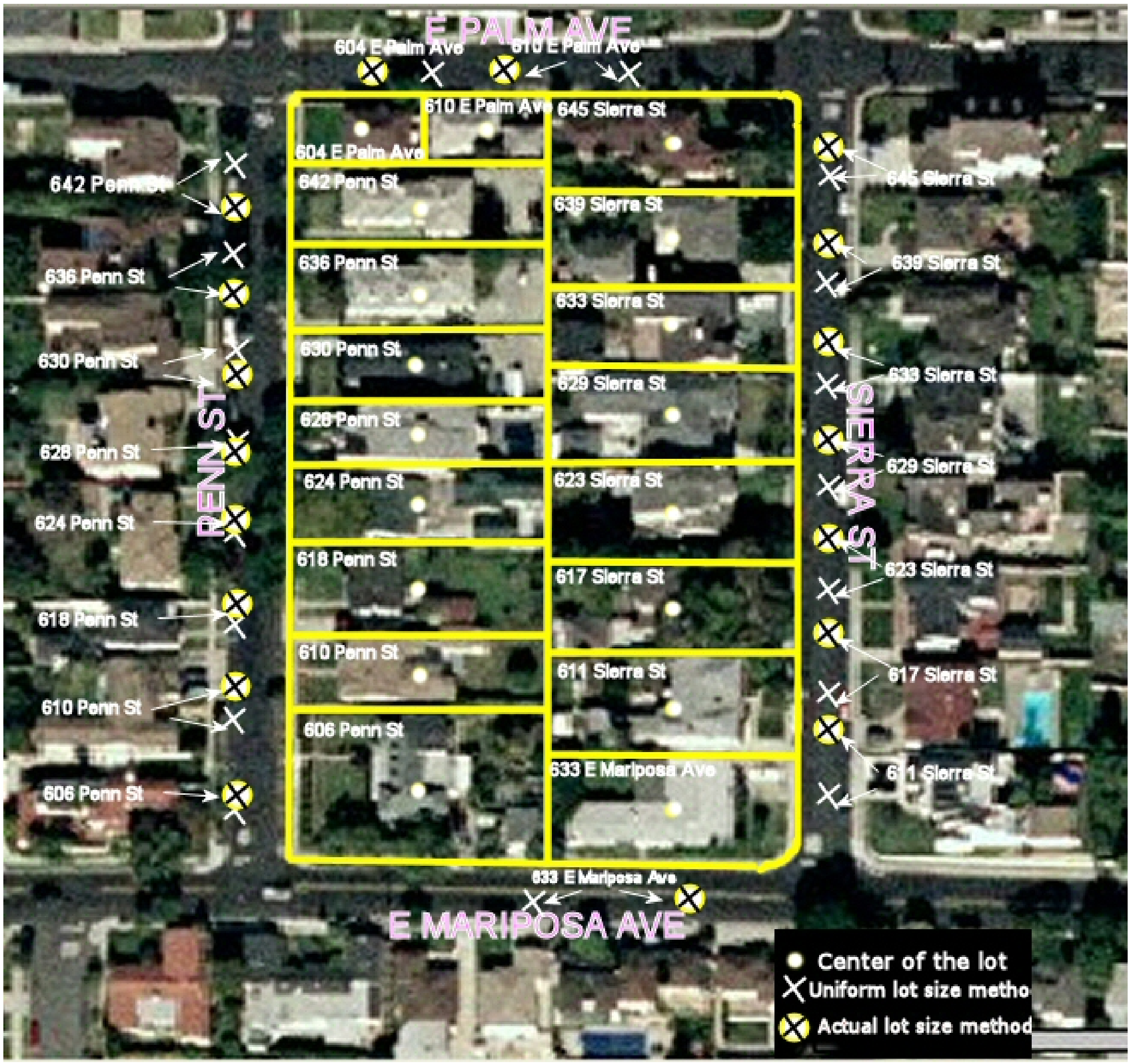

Figure 7. Geocoded locations from the new methods

In the data integration community there has been some work on integrating geo-spatial datasets. The goal of the MIX mediator [10] and the Hermes mediator [1] is to provide unified access to a wide variety of data sources. Both mediator systems utilize Global-As-View approach to integrate data. Adding new sources to Global-As-View model may require changing all the rules in the domain model. In the case of Columbus, new property tax web sites become available everyday, therefore, the GLAV approach is more suitable. In general, as geo-spatial data sources often vary in coverage and new data sources with different coverage regularly become available, the GLAV approach is more suitable.

7. CONCLUSION

This research shows that information integration techniques can be used to achieve a remarkable improvement in the geocoding process. The two new methods proposed achieved much better results over the existing methods. The uniform lot size method showed an improvement of 74.32% over the existing methods while the actual lot size method provided an improvement of over 91.33%.

Thus we have successfully realized a geocoder which provides more accurate values of latitude and longitude for a given address. When a request for geocoding is given to Columbus, depending on the availability of the property data sources, the best approach of the three described here is performed. We have also solved the problem of validating addresses before geocoding, depending on the availability of appropriate data sources.

In future work, we plan to add a normalization step to the geocoder to handle the wide variety of address formats. Instead of building a specialized normalization routine, we plan to do this by exploit record linkage techniques [18] for linking records across related sources. In this case we can link the submitted address to the standardized address in an address database obtained from the US Postal Service.

8. ACKNOWLEDGMENTS

This material is based upon work supported in part by the National Science Foundation under Award No. IIS-0324955, in part by the Defense Advanced Research Projects Agency (DARPA), through the Department of the Interior, NBC, Acquisition Services Division, under Contract No. NBCHD030010, in part by the Air Force Office of Scientific Research under grant number FA9550-04-1-0105, in part by the United States Air Force under contract number F49620-02-C-0103, and in part by a gift from the Microsoft Corporation.

The U.S. Government is authorized to reproduce and distribute reports for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of any of the above organizations or any person connected with them.

9. REFERENCES

[1] Adali, S., Candan, K.S., Papakonstantinou, Y., and Subrahmanian, V.S. Query Caching and Optimization in Distributed Mediator Systems, *In Proceedings of the ACM SIGMOD*, 1996,137-148.

[2] Bakshi, R. *Exploiting online sources to accurately geocode addresses.* Masters Thesis, University of Southern California, Los Angeles, CA, 2004.

[3] Cayo, M.R. and Talbot, T.O. Positional error in automated geocoding of residential addresses, *International Journal of Health Geographics*, 2, 10 (2003).

[4] Chen, C.-C., Knoblock, C.A., Shahabi, C., and Thakkar, S. Automatically and Accurately Conflating Satellite Imagery and Maps, *International Workshop on Next Generation Geospatial Information*, Cambridge, MA, 2003.

[5] Chen, C.-C., Thakkar, S., Knoblock, C.A., and Shahabi, C. Automatically Annotating and Integrating Spatial Datasets, *In the Proceedings of International Symposium on Spatial and Temporal Databases*, Santorini Island, Greece, 2003.

[6] Duschka, O.M. *Query Planning and Optimization in Information Integration.* Ph. D. Thesis, Stanford University, 1997.

[7] Friedman, M., Levy, A., and Millstein, T. Navigational Plans for Data Integration, *In Proceedings of the 16th National Conference on Artificial Intelligence*, 1999,67-73.

[8] Garcia-Molina, H., Hammer, J., Ireland, K., Papakonstantinou, Y., Ullman, J., and Widom, J. Integrating and Accessing Heterogeneous Information Sources in TSIMMIS, *Proceedings of the AAAI Symposium on Information Gathering*, Stanford, CA, 1995.

[9] Genesereth, M.R., Keller, A.M., and Duschka, O.M. InfoMaster: An information integration system, *In Proceedings of ACM SIGMOD-97*, 1997.

[10] Gupta, A., Marciano, R., Zaslavsky, I., and Baru, C. Integrating GIS and Imagery through XML-Based Information Mediation, *In Proceedings of the NSF International Workshop on Integrated Spatial Databases: Digital Images and GIS*, 1999.

[11] Kiernan, A. Homeland Security and Geographic Information Systems – How GIS and mapping technology can save lives and protect property in post-September 11th America, *Public Health GIS News and Information*.US Federal Geographic Data Committee, 2003,20-23.

[12] Knoblock, C., Minton, S., Ambite, J.L., Ashish, N., Muslea, I., Philpot, A., and Tejada, S. The ARIADNE Approach to Web-Based Information Integration, *International Journal on Intelligent Cooperative Information Systems (IJCIS)*, 10, 1-2 (2001), 145-169.

[13] Knoblock, C.A., Minton, S., Ambite, J.L., Ashish, N., Modi, P.J., Muslea, I., Philpot, A.G., and Tejada, S. Modeling web sources for information integration, *In Preceedings of the Fifteenth National Conference on Artificial Intelligence*, Madison, WI, 1998.

[14] Krieger, A., Waterman, P., Lemieux, K., Zieler, S., and Hogan, J.W. On the wrong side of the tracts? Evaluating the accuracy of geocoding in public health research, *American Journal of Public Health*, 91, 7 (2001), 1114-1116.

[15] Lenzerini, M. Data integration: A theoretical perspective, *In Proceedings of ACM Symposium on Principles of Database Systems*, Madison, Winsconsin, USA, 2002.

[16] Levy, A., *Logic-Based Techniques in Data Integration*, in *Logic Based Artificial Intelligence*, J. Minker, Editor. 2000, Kluwer Publishers.

[17] Levy, A.Y., Rajaraman, A., and Ordille, J.J. Query-answering algorithms for information agents, *In Proceedings of AAAI-96*, 1996.

[18] Michalowski, M., Knoblock, C.A., and Thakkar, S. Exploiting Secondary Sources for Unsupervised Record Linkage, *Proceeding of 2003 KDD Workshop on Data Cleaning, Record Linkage, and Object Consolidation*, 2003.

[19] Ratcliffe, J.H. On the accuracy of TIGER-type geocoded address data in relation to cadastral and census areal units, *International Journal of Geographical Information Science*, 15, 5 (2001), 473-485.

[20] Saalfeld, A. *Conflation: Automated Map Compilation.* Ph.D. Thesis, University of Maryland, 1993.

[21] Sinnott, R.W. Virtues of the Haversine, *Sky and Telescope*, 68, 2 (1984), 159.

[22] Thakkar, S., Ambite, J.L., and Knoblock, C.A. A Data Integration Approach to Automatically Composing and Optimizing Web Services, *In Proceedings of the ICAPS Workshop on Planning and Scheduling for Web and Grid Services*, 2004.

[23] Thakkar, S. and Knoblock, C.A. Efficient Execution of Recursive Integration Plans, *In Proceeding of 2003 IJCAI Workshop on Information Integration on the Web*, Acapulco, Mexico, 2003.