

## INTRODUCING AN AGENT-BASED OBJECT RECOGNITION OPERATOR FOR PROXIMITY ANALYSIS

S. Behzadi<sup>a,\*</sup>, A. Ali. Alesheikh<sup>b</sup>

<sup>a</sup> PhD student in GIS Dept., Faculty of Geodesy and Geomatics Eng., K.N. Toosi University of Technology, Tehran, Iran, behzadi@gmail.com

<sup>b</sup> Associate Professor of the GIS Dept., Faculty of Geodesy and Geomatics Eng., K.N. Toosi University of Technology, Tehran, Iran, alesheikh@kntu.ac.ir

**KEY WORDS:** Geographic Information System (GIS), Agent based model, Geometric buffer, Proximity analysis

### ABSTRACT:

Object selection is a basic procedure in a Geographic Information System (GIS). Most current methods for doing so, select objects in two phases: create a simple distance-bounded geometric buffer; and intersect it with available features. This paper introduces a novel and intelligent selection operator based on the autonomy of the agent-based approach. The proposed operator recognizes objects around one object only in one step. In the proposed approach, each point object acts as an agent-automata object. It then senses its vicinity and identifies the surrounding objects. To assess the proposed model, the operator is designed, implemented, and evaluated in a case study. Finally, the results are evaluated and presented in details in the paper.

### 1. INTRODUCTION

The analytical capabilities of GIS made the system more practical. Proximity analysis is one of the most basic GIS capabilities that is used to discover proximity relation (deBy et al., 2001). The analysis recognizes objects with buffer features (Basnyat et al., 2000). The buffer operation generates polygon feature regardless of geographic features and delineates spatial proximity (Shekar and Xiong, 2008). Although this method works well for many applications, it has limitations when dealing with complex queries such as finding the nearest object in the buffered area, counting the number of objects in the buffered one, and so on.

Producing a buffer or service area around a feature is a basic command of a GIS (Basnyat et al., 2000, Sieber et al., 2009), which has been exercised by numerous researchers. Rebolj and Sturm (1999) generated a buffer to identify the population residing in a vicinity of a road that is affected by noise pollution above the recommended levels. Žalik et al. (2003) presented a four steps algorithm for constructing the buffer of a given line segments, using a sweep-line approach. The buffer introduced by Žalik is called geometric outlines. It presented a specific area around line segments. The area was asymmetric; therefore, it deviated from the other practiced buffer operators. The buffer proposed by Basnyat (Basnyat et al., 2000) was called forested buffer to determine the affect of various attributes in water quality. In this paper, the buffer is generated based on a mathematical model. The governing factors of the model were land use/land cover, slope, and soil types. The buffer was asymmetric; it expanded differently in various directions.

Although buffer generations are mostly based on the combination of Thiessen polygons and maximum-distance Euclidean (Haggett et al., 1977), Upchurch (Upchurch et al., 2004), proposed a new type of buffer for defining service areas that supplement the previous studies by introducing network-distance buffer.

The majority of researchers determined the objects of interest in two phases: generate a polygon around the specific object; intersecting the buffer area and specific object to select the interested objects in that area. Therefore, object selection was practiced in two steps.

This paper introduces a new agent-based operator that selects objects only in one step. Moreover, the proposed operator is object-based and works for vector data.

This paper is constructed in 4 sections. The first section introduces the concept of agents. The second part illustrates the method used to implement the operator. The third section briefly evaluates the efficiency of the operator. The paper ends up with some conclusions and recommendations in section 4.

### 2. AGENT

An agent is a computer system that is situated in environment. Agent is capable of autonomous action in this environment to meet its design objectives (Wooldridge, 2002, Weiss, 1999, Schumacher, 2001, Dunin-Keplicz and Verbrugge, 2010, Vidal, 2010). Agent-based solution is a new view for solving problems. If the conditions of the problem are structured based on the above definition, the solution follows an agent-based approach. In general, four basic types of agent programs are available: simple reflex agents, model-based reflex agents, goal-based agents, and utility-based agents. These four types of agents are shown in Figure 1 (Russell and Norvig, 2003).

In the simple reflex agent, the agent selects actions on the basis of the current percept, ignoring the rest of the percept history. In the Model-based reflex agent, the agent maintains some sort of internal state that depends on the percept history; therefore, the agent simulate at least some of the unobserved aspects of the current state (Uhrmacher and Weyns, 2009). In the Goal-based agent, beside the current state description, the agent needs some sort of goal information that describes desirable situations (Vidal, 2010). In the utility-based agent, a utility function is

---

\* Corresponding author

used to describe the associated degree of success (North and Macal, 2007), one of which might be used for solving the problem. The reason of using utility function depends on the structure of the problem. In this paper, the utility-based agent is used. At first, the goal is defined for agent; then the agent tries to reach the defined goal optimally.

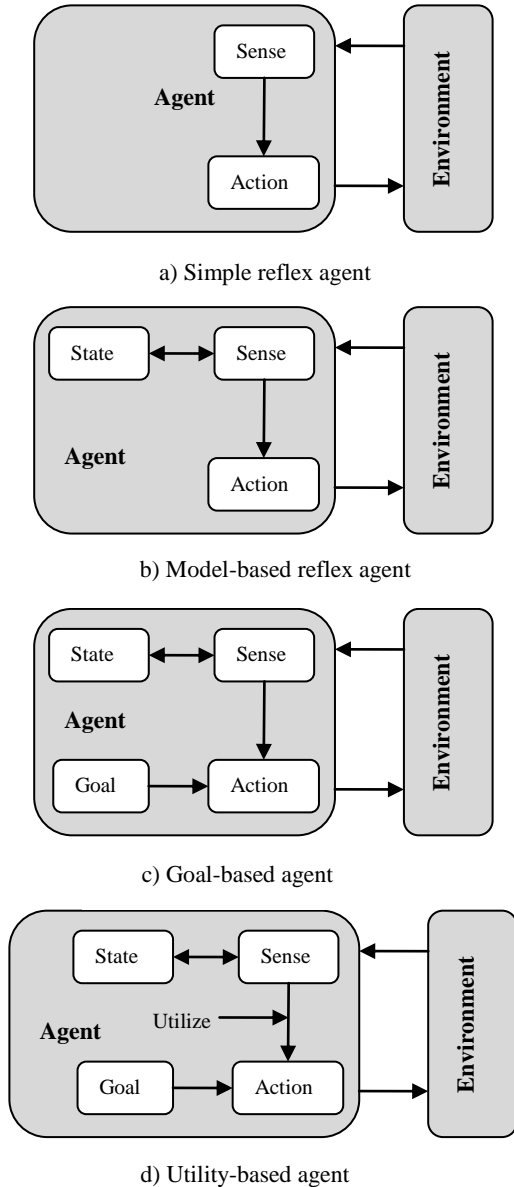


Figure 1. Four basic types of agent programs

### 3. THE PROPOSED METHODOLOGY

In this model, four steps are defined for the process; at first, the agent observes the environment; then it gathers some data. Next, the perception of agent is obtained based on some analysis done on the observation. In the next step, agent generates its knowledge base. Finally, it selects objects based on its knowledge. Figure 2 shows the flowchart of the proposed agent for extracting linear objects.

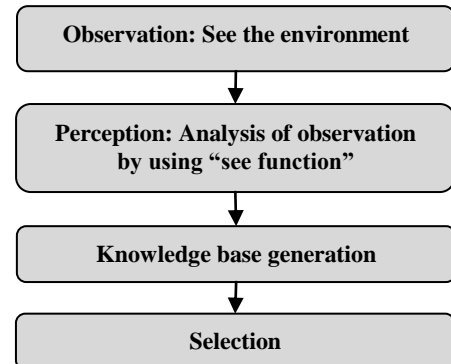


Figure 2. The process of decision making in agent

#### 3.1 Observation of the environment

When a point object is selected to identify its nearby linear objects, an agent is formed. Then the functionalities of such an agent must be defined. The agent observes and senses its vicinity. In this step, the spatial coordinates (x, y) of all objects are stored in a database. The x and y coordinates of endpoints are considered as agent's observation. In this case, the agent is a point with its x, y coordinates. The objects around the agent are categorized into three types: points, lines, and polygons. The distance between the agent and point object is calculated based on Euclidian distance. Numerous researchers estimated the distance between polygon objects based on the distance of their centroid (Uno and Kashiyama, 2008). This study constrains its agents to sense nearby linear objects using the combination of some parameters, all of which are described in the following section.

#### 3.2 Sensing function

The see angle (Equation 1) of the agent toward the line is calculated based on the gathered data.

$$See\ Angle = \frac{P_{Agent} \vec{P}_A \cdot P_{Agent} \vec{P}_B}{\left| P_{Agent} \vec{P}_A \right| \left| P_{Agent} \vec{P}_B \right|} \quad (1)$$

Where:  $P_{Agent}$ ,  $P_A$  and  $P_B$  are the positions (x and y) of the agent and both ends of the line respectively. This angle is shown in Figure 3.

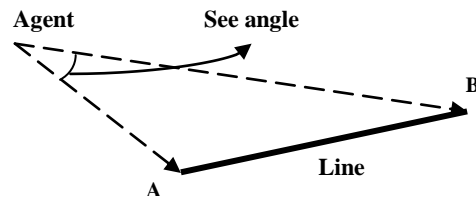


Figure 3. See angle of an agent

The locus of all points in the area having the same see angle to the line is a part of a circle as shown in Figure 4. Therefore, another parameter is required to identify the specific object uniquely. The direction of point related to the line is the parameter extracted from the observed data. The direction is calculated by Equation 2.

$$D = (P_B \vec{P}_A \cdot P_{Agent} \vec{P}_A) (P_A \vec{P}_B \cdot P_{Agent} \vec{P}_B) \quad (2)$$

Where:  $P_{Agent}$ ,  $P_A$  and  $P_B$  are the positions (x, y) of the agent and both ends of the line respectively.

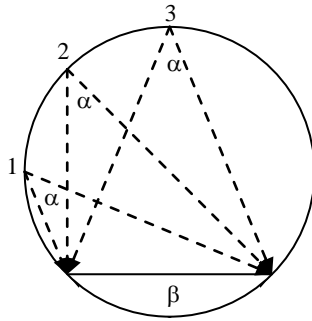


Figure 4. The locus of all points in the area having the same see angle to the line

For many proximity analysis experimented by authors, a “see angle” is enough for identifying the closest object. Based on the see angle, two cases might occur:

1. “see angle” is less than  $90^\circ$ : In this situation, two sub-cases happen:
  - a. The agent is near the line, but it is located out of the space between line endpoints (e.g. point number 1 in Figure 4).
  - b. The agent is near to the line; such as point 2 and 3 in Figure 4.
2. “see angle” is more than  $90^\circ$ : The point is located near the line; as a result, the “see angle” is close to  $180^\circ$  (e.g. the see angle of point  $\beta$  shown in Figure 4).

The last parameter is the length of the line observed by the agent. If the line is long and far away from the agent, it has a negative effect on the “see angle. So the length of the lines must be considered as well.

### 3.3 Agent's Knowledge Base

Agent can make a decision of which object is in its proximity area based on the rules in agent's knowledge base. These rules are based on the see angle (Eq.1) and the sign of the direction of point (Eq.2). This knowledge is considered for all objects around the agent. These rules are as:

**Rule (1):** The direction of point to the line (D) is evaluated. If the sign D is negative, it shows that the agent is not in the area between two ends of the line, so it must be deleted; otherwise, if the sign is positive, it shows that the agent is in the area between the two ends of the line.

**Rule (2):** If the “see angle” is more than  $90^\circ$ , the objects are selected. For these objects another factor is calculated. The factor shows the approximate distance of the agent to the object. The factor is estimated as:

$$S = \frac{L}{2 \tan(\alpha/2)} \quad (3)$$

Where:  $\alpha$  is the “see angle” of agent to the object and L is the length of the line.

Figure 5 shows the flowchart of the proposed agent for extracting linear objects. In reality, It is rare that more than one line with “see angle” greater than  $90^\circ$ ; if so, the algorithm needs to use an approximate distance (S) to filter the reminding points.

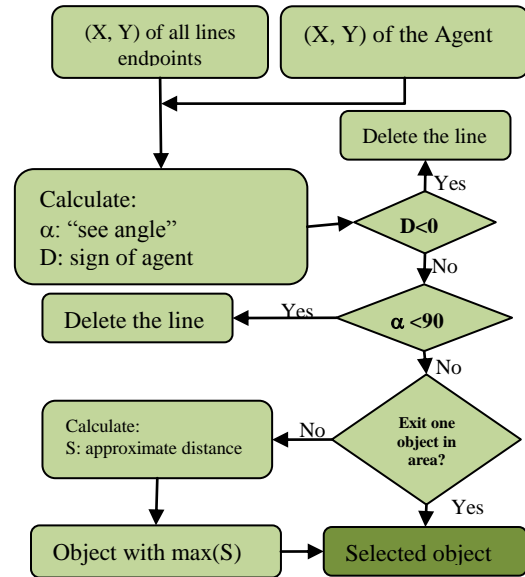


Figure5. The methodology used for object selection

## 4. IMPLEMENTATION AND DISCUSSIONS

The proposed agent is implemented on the sample data as shown in Figure 6. The test area contains 11 lines with various directions and lengths around the point object (agent). The question is to identify and select the closest line to the agent. At first, the see angle and the D parameters is calculated for each line. Next, the nearest object to the agent is selected based on the rules of the agent.

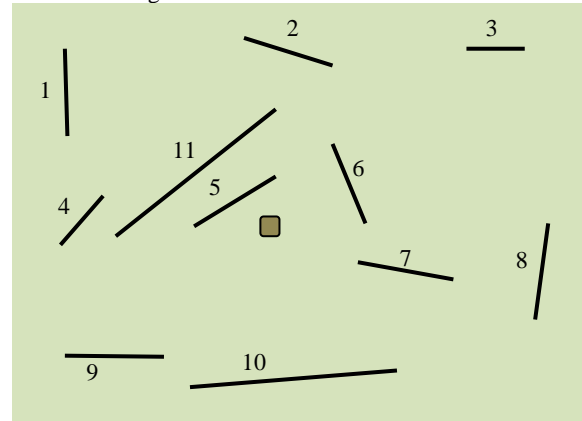


Figure 6. The experimental data contain 11 lines and one point

As seen the result of the algorithm, the nearest line is line number 5. Similarly, Table 1 shows that line number 5 and 11 have the highest see angles. Once the approximate distances are calculated, it shows that line number 5 is the one which has all conditions. It must be mentioned that the selected line is not mathematically the nearest one to the point; as seen in Table 1, the nearest line to the point is the line number 7, but it is not selected by the agent.

To assess the efficiency of the proposed extraction operator, the common and the agent-based selection operator are implemented on a real urban map of Tehran (Iran). The process is done to update the traffic geospatial data base. In this problem, the connection between house and road is determined. In this case, it must be defined which road is related to which house. As a result, the ID of the road is stored in the house layer. The problem is to select the nearest line (road) for the identified points (houses). At first, the problem is solved by

using Arc GIS. In this system, for a selected polygon, a buffer area is generated, and then the nearest line to the polygon is specified by intersection of buffered area and all lines. In the proposed method, the selected polygon is converted to agent, after that the agent will directly find the nearest line based on the proposed solution. Figure 7 shows the results of the two methods. Line selection operation used in common GIS software is done in two phases; however, in the proposed method, not only is the process done in one step, but also it is independent of the user. The proposed method is also time-saving; it solves the problem 23% faster than the common method for the region of 792 parcels.

Table1. The Results of agent computation

Point	See angle	Sign of "D"	Line length	$S = \frac{L}{2 \tan(\alpha/2)}$	Distance of the object to line
1	15°	-	1.5541		4.0008
2	33°	+	1.9044	3.214573327	3.1257
3	7°	-	1.2050		3.1778
4	13°	-	1.2206		2.7610
5	106°	+	1.8836	0.709697204	0.6314
6	48°	+	1.5834	1.778187314	1.7875
7	5°	-	1.9769		0.3266
8	18°	+	1.7899	5.650491918	5.5507
9	19°	-	2.0746		2.3215
10	74°	+	4.1962	2.78427274	2.7617
11	109°	+	4.1777	1.489962225	1.4276

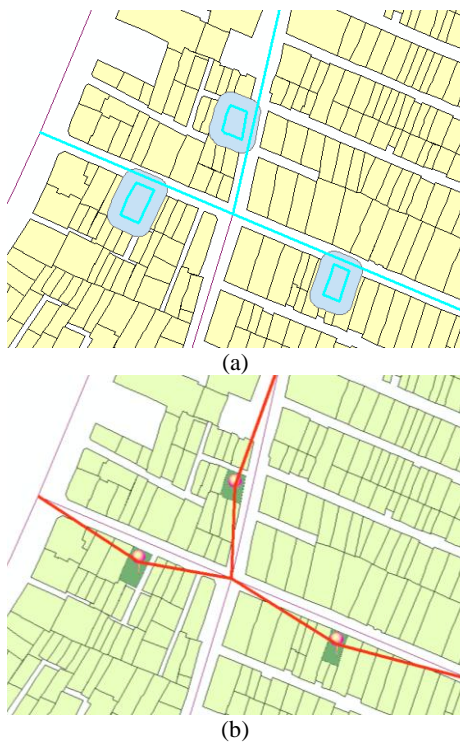


Figure7. The comparison of (a) common buffer operator and (b) Agent-based line selection

### 5. CONCLUSION

Object selection is one of the most common operators used in GIS. In all researches, for object selection, at first a buffer area is defined; then in the buffer area, specific objects are selected. Moreover, the two-folded procedure is field-based. This paper proposed an object-based approach for selecting objects directly. In the proposed method, the agent-based operator chooses directly these specific objects without defining the buffer area. The results of implementation showed that in a

complex area the agent-based buffer works correctly as shown in Figure 6 and Table 1. The proposed operator is also compared with commonly used buffer for selecting a certain type of objects. The proposed buffer found the specific object in one step while the common method works in two steps. In this paper, the simplest type of object selection is offered in agent's view. In some research, finding object between two bounds is needed. For example, in GIS, there is a need to find objects with the distance between D1 and D2 from the river. In this case, a specific angle (see angle) must be selected for agent. Besides, additional rules are needed, which the authors now work on such the buffer. Further investigation is required to make linear objects as agent.

### REFERENCES

BASNYAT, P., TEETER, L., LOCKABY, B. & FLYNN, K. 2000. The use of remote sensing and GIS in watershed level analyses of non-point source pollution problems. *Forest Ecology and Management*, 128, 65-73.

DEBY, R. A., KNIPPERS, R. A., SUN, Y., ELLIS, M. C., KRAAK, M.-J., C.WEIR, M. J., GEORGIADOU, Y., RADWAN, M. M., VANWESTEN, C. J., KAINZ, W. & SIDES, E. J. 2001. *Principles of Geographic Information System*, The Netherlands, ITC.

DUNIN-KEPLICZ, B. & VERBRUGGE, R. 2010. *TEAMWORK IN MULTI-AGENT SYSTEMS (A Formal Approach)*, Chichester, Wiley

HAGGETT, P., CLIFF, A. D. & FREY, A. E. 1977. *Locational Analysis in Human Geography*, London, Edward Arnold.

NORTH, M. J. & MACAL, C. M. 2007. *MANAGING BUSINESS COMPLEXITY*, New York, Oxford University Press, Inc.

REBOLJ, D. & STURM, P. 1999. A GIS based component-oriented integrated system for estimation, visualization and analysis of road traffic air pollution. *Environmental Modelling and Software*, 14, 531-539.

RUSSELL, S. J. & NORVIG, P. 2003. *Artificial Intelligence*.

SCHUMACHER, M. 2001. *Objective Coordination in Multi-Agent System Engineering*, Zürich, Springer.

SHEKAR, S. & XIONG, H. 2008. *Encyclopedia of GIS*. Springer, 1370.

SIEBER, S., PANNELL, D., MÜLLER, K., HOLM-MÜLLER, K., KREINS, P. & GUTSCHE, V. 2009. Modelling pesticide risk: A marginal cost-benefit analysis of an environmental buffer-zone programme. *Land Use Policy*.

UHRMACHER, A. M. & WEYNS, D. 2009. *Multi-Agent Systems Simulation and Applications*, Taylor & Francis Group.

UNO, K. & KASHIYAMA, K. 2008. Development of simulation system for the disaster evacuation based on multi-agent model using gis. *Tsinghua Science & Technology*, 13, 348-353.

UPCHURCH, C., KUBY, M., ZOLDAK, M. & BARRANDA, A. 2004. Using GIS to generate mutually exclusive service areas linking travel on and off a network. *Journal of Transport Geography*, 12, 23-33.

VIDAL, J. M. 2010. Fundamentals of Multiagent Systems.

WEISS, G. 1999. Multiagent Systems.

WOOLDRIDGE, M. 2002. An Introduction to Multiagent Systems.

ŽALIK, B., ZADRAVEC, M. & CLAPWORTHY, G. 2003.  
Construction of a non-symmetric geometric buffer from a set of  
line segments. Computers and Geosciences, 29, 53-63.