

Intelligent Tourist Information System

Tirth Shah
B. Tech.
Dept. of IT, Charusat

Sonal Rami
Assistant Professor
Dept. of IT, Charusat

Ayesha Shaikh
Assistant Professor
Dept. of IT, Charusat

ABSTRACT

This paper proposes a model for implementation on intelligent tourist information system. It uses the concept of knowledge-base. The model will be based on the study of human behavior as tourism guide. It builds the relationships between the knowledge based system and the guide, so that it provides service for any visitor which meets their needs and the objective of gaining information of places. There are different modules, different path finding systems and shortest path finding algorithms of artificial intelligence in this thesis. The proposed system should be designed in such a way that it runs on most of devices i.e. palmtop and mobiles. Thus it can be useful while visiting new places. This system would find a route using user criteria. The Shortest path finding algorithm should work efficiently and optimally in most of the cases. The system should find a path that fulfills user criteria, show name of objects, related photos and short description about the place. It should also be able to find distance, time and cost to travel particular destination.

Keywords

Recommendation technique, Shortest path algorithm, Intelligent tourist system, A* algorithm.

1. INTRODUCTION

People love to travel. For travelling, a person must select the destination city. After selecting he/she must decide the places which they want to visit such as resorts, amusement parks, art gallery, restaurants etc. Sometimes people do not have time to plan out these things before, so they ask their relatives, friends who have visited the place earlier and organize it accordingly. Generally we do not have any valuable information except phone number and address, which can make us decide that we want to visit such place. This work basically concentrates on designing systems that provide us the short description about the places. So that we can decide which place we want to visit. This system determines the aspects of knowledge base system which helps to provide all features in the application. The system should be designed in such a way that it should be able to run on all platforms and devices.^[2]

2. OBJECTIVES

- The main objective of the research paper is to design a knowledge base system which will be built on the study of human's behavior and adopting the human intelligence in the machine.
- To help in designing the efficient, fast path searching algorithms with optimization.
- To provide the structured database, each place with its short description, timings, ratings etc with GPS and GPRS. The geographical database of the places should also be built.
- To find shortest path between two points (location) in map i.e. navigation systems.

3. EXISTING SYSTEM

In the existing system, people who want to travel searches lots of tourism websites. They need to know the information about the place of interest. They visit certain travel websites or hire a tourist guide which may consume a lot of time. The existing system gives the details like phone number, address, ratings. If a person wants to decide which place he/she wants to visit first, they need to know the information about the place, which the existing system is unable to provide.^[1]

4. RECOMMENDER SYSTEM

A recommender system is a subclass of information filtering system that seeks to predict the preference that a user would give to an item. It is used in variety of areas including the research articles. It typically produces a list of results in one of the two ways- collaborative and context based filtering.^[1]

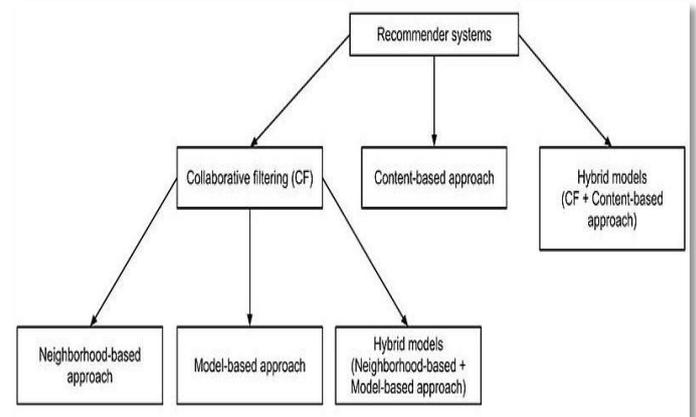


Figure 1 Recommender System Classification

5. ARCHITECTURE

The term artificial intelligence means to embody the human intelligence into machine. The machine should have the capabilities that a human possess. The capabilities such as smart, solving unstructured and complex problems like human does. The machine should understand the language spoken by humans, able to analyze data and information. For example when some expert delivers lecture to student, student learns from it, does hard work and performs activities after learning it. In the same manner machine is made to embody the expert knowledge into some computer program for carrying out some task. The keyboard will be replaced by speech through natural language processing. Human can solve algorithmic and non algorithmic problems. There must be some methods to solve non algorithmic problems, in the form that machine can solve. This capability is the most important thing in artificial intelligence and we are concentrating on them. It includes various functional modules such as user interface, knowledge base, dynamic database, Inference engine^[8]

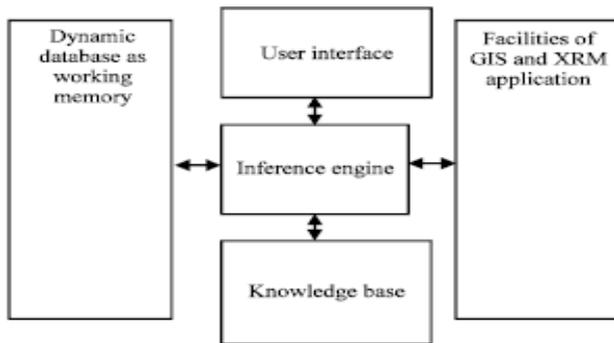


Fig 2 Architecture of Intelligent tourist information system

5.1 User interface

Through the user interface, the traveler or user can communicate with the system. The information is processed from user to computerized system in one form to another form. The specific protocol is used to connect the user with the machine. The chosen method for interaction will be based on the format used for representation of knowledge base.

5.2 Knowledge base

The design of the knowledge base depends on the type of representation used for knowledge. There are many forms of knowledge representation such as rule base, semantic nets, case base, logic forms. The most important part in building knowledge based system is building its knowledge base. Humans have common sense, analogical reasoning facilities. These things are not included in knowledge representation, because the logical deduction is rule base, analogical reasoning is case base, common sense is using black board. The system which is defined, possess the combination of the rule base and the case base using blackboard. This will add three features in a single scheme. Various search techniques and problem solving methods exists to design the inference for the new system. Human knowledge is not limited to any one form. It can be represented in more than one form for efficient solving of problem. Let us take an example in which the rule is represented in table form, where the first column represents the rule and the action, while the other columns represent the conditions to be performed. This is called procedural representation.

5.3 GIS

GIS stands for geographic information systems. It is used to display and analyze the data that are linked to database. The link between data and database acts as the force behind the working of GIS. The database helps to draw a map and the data can be enhanced from the map whenever the database is updated. The association map gets updated. The GIS database consists of large variety of information such a political, geographic, demographic data. It helps to determine the important places for tourism. It helps to determine the plan which is optimum to sightseeing place, attractive, eye catching places. It helps to determine the shortest distance between selected places. It maps the digital files to internet.

5.4 XRM

XRM stands for extended relationship management. It is a strategy which understands what information is needed, by whom is it needed, and how it needs to be displayed for better decision making. It offers end to end line of application development platforms that includes the best of user as well as the systems. It makes application rapidly and cost effectively.

5.5 Dynamic database

Initially the database is empty, but during the run time execution of the system, it will be comprised of data and assertions. The data provided by GIS technology is used for problem solving. The dynamic database acts as a working memory. The processes between knowledge base and interaction between user and system lead to generation of assertions. This GIS technology consists of computer based data, storage and analyzes previously unrelated information into easily understood maps as per user requirements. The inference engine can see all the types of media such as maps, tables or graphs and select the best choice as per user requirements

5.6 Inference engine

Inference engine is responsible for building functional model of machine as human. Although implementation of it depends on the type of representation used from knowledge base. Inference engine is simulation for solving the problems with activities like deduction, reasoning and common sense. The inference engine implementation includes the combination of problem solving, search technique and reasoning. Heuristic function is used to retrieve knowledge from its knowledge base. The heuristic search technique uses dynamic database in the inference engine. For example given a set of tasks or rules, the conclusion is inferred from it. The problem reduction method is used in project.

6. EXISTING PATH FINDING SYSTEMS

This includes the comparisons of various existing path finding systems. There have been systems which allow the traveler to find a path using navigation and maps. The traveler must know the destination place they want to visit, in order to find the path. The system was not able to recognize the most interesting places of the city by itself. The existing path finding systems were too slow as compared to the path finding systems using artificial algorithms. There are certain advantages and disadvantages of existing path finding system.^[11]

6.1 Automapa

Automapa is basically a software program which covers the map of entire Poland. The main advantage is that it has the most accurate map of the island consisting each and every road on map. The point of interest database is very huge and complex source of information. It can add new POI's into map, which is other advantage. The major drawback is that it is very slow. Drawing map with many POI's and adding new POI's takes a lot of time. Another drawback is that the program faces difficulty in finding path if the traveler is moving in opposite side. It only allows to search POI's name and type.

6.2 IGO

IGO is a system which covers map of entire Europe. It also has huge base of POI's database, as similar to automapa it can only store name and address of POL. Thus the short description of POI is not provided by IGO also. Hence the traveler is unable to decide which POI or place he must choose to visit. The main advantage is that it works fast and fine. It is also speed control i.e. it will make some sound if the vehicle is moving above the speed limit.

6.3 Tomtom

TOMTOM covers the map of entire Europe. It is the best software in this path finding system category. It is very popular navigation application consisting of huge base of POI's. It has nice user interface with 3D view on road. It is the fastest software for navigation. It doesn't provide POI description which allows selecting the objects to be visited.

6.4 Flaws in existing systems

- Major drawback is POI database. It is huge and consists of lot of objects from different categories. When a tourist arrives, they want to know specific place's cost of entrance, timings etc, so that they can take decision effectively and decide which place should be visited. Another drawback is traveler must know where they want to go.
- The new system would ask about information and using these criteria, they should build some path. The response should be quick and path obtained should be optimal for user. The existing systems allow the traveler to estimate their actual position, find path to reach to their selected target. They do not allow estimating path using human intelligence.
- For example a path to the best sightseeing of particular city, they would give all the paths to that place but will not be able to give the shortest optimal path for every instance.
- The new system is believed to remove all such flaws and give user information about POI's like short description, photos, entrance costs and timings.

Table 1 comparison between various existing path finding systems.

Application	Interface	Performance	Covering
AutoMapa	natural	good	only Poland
IGO	natural and limpud	good	depend from
TomTom	natural	very good	version (Europe)

7. A* ALGORITHM

In computer science, A* is an efficient algorithm generally used for path-finding and graph traversal by efficiently plotting directed path between multiple nodes. Peter Hart, Nils Nilsson and Bertram Raphael of Stanford Research Institute invented the A* algorithm. A* uses dynamic programming to reach goal. A* algorithm has better performance and accuracy than Dijkstra. A* is complete as well as optimal search algorithm.^[3] A* algorithm uses graph as data structure and is an informed search algorithm. A* searches for all the immediate neighbors from the current node to find shortest path. A* introduces the most significant characteristic of the algorithm which makes it efficient than others that is heuristics.^[6] Heuristic function is used to estimate the cheapest path from current node to goal node and they should be admissible. A* minimizes:

$$f(n) = g(n) + h(n)$$

where n is the last node on path, g(n) is cost of the path from start node to n and h(n) is heuristic function. Thus, f(n) is cheapest cost to goal through n. A* uses priority queue managing an opened and closed set. As A* searches in tree manner the explanation is in the same form. Given is the simple explanation of how A* search works with an example.

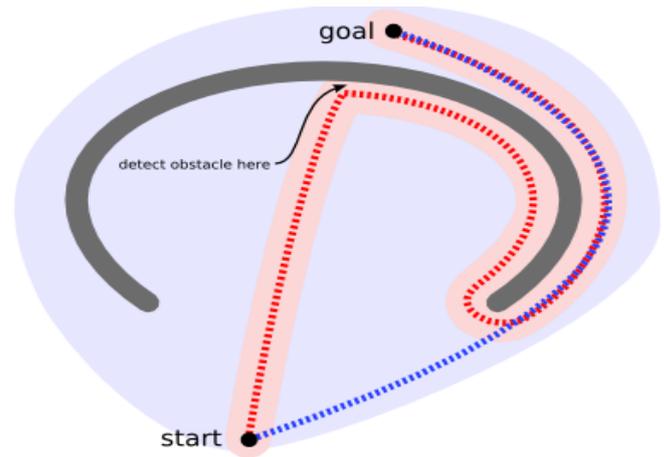


Fig 3

The unit is initially at the bottom of the map and wants to reach to goal. Initially it scans (shown in pink) the area, there is no obstacle at first when it starts scanning, then obstacle it discovered which indicates that the unit should not move up, so it changes direction. It then finds its way around the "U"-shaped obstacle, following the red path. In contrast, a pathfinder would have scanned a larger area (shown in light blue), but found a shorter path (blue), never sending the unit into the concave shaped obstacle.

We can however extend a algorithm to work around traps like the one shown above. Either avoid creating concave obstacles, or mark their convex hulls as dangerous (to be entered only if the goal is inside)

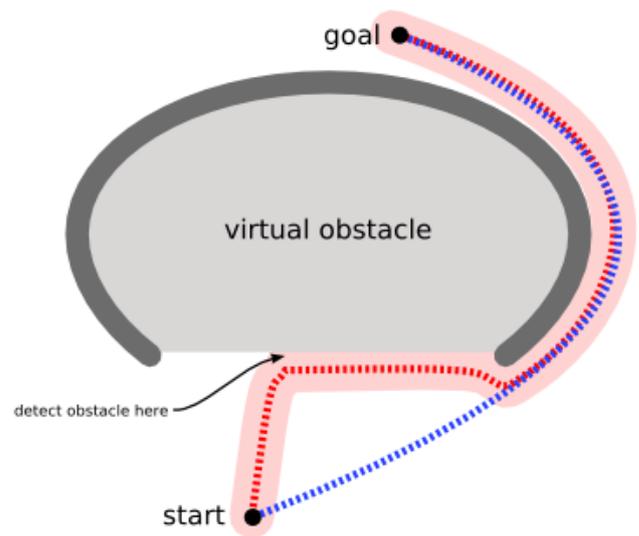


Fig 4

Pathfinding navigation system let you plan ahead rather than waiting until the last moment to discover there's a problem. There's a tradeoff between planning with pathfinders and reacting with movement algorithms. Planning generally is slower but gives better results; movement is generally faster but can get stuck. If the game world is changing often, planning ahead is less valuable. I recommend using both: pathfinding for big picture, slow changing obstacles, and long paths; and movement for local area, fast changing, and short paths.^[5]

7.1 Pseudo code

```
initialize the open list
initialize the closed list
put the starting node on the open list
while the open list is not empty
    find the node with the least f on the open list, call it "q"
    pop q off the open list
    generate q's successors and set their parents to q
    for each successor
        if successor is the goal, stop the search
        successor.g = q.g + distance between successor and q
        successor.h = distance from goal to successor
        successor.f = successor.g + successor.h
        if a node with the same position as successor is in the
        OPEN list \
            which has a lower f than successor, skip this successor
        if a node with the same position as successor is in the
        CLOSED list \
            which has a lower f than successor, skip this successor
        otherwise, add the node to the open list
    end
push q on the closed list
```

7.2 Time complexity

The time complexity of A* depends on the heuristic, if the function is admissible and better then we get optimal solution with optimal time complexity. In the worst case of an unbounded search space, the number of nodes expanded is exponential in the depth of solution, $d = O(b^d)$ where b is branching factor.^[9]

7.3 Applications

A* is commonly used for path-finding in games. It is also used in solving problem of parsing using stochastic grammars in NLP.^[10]

7.4 Improvements

Bidirectional A* is used an improvement to A* where the main concept is to implement A* search from source as well as destination. In which, it depends on how search at both the end meets.

7.5 Limitations

A* search will not give optimal result if heuristics are not admissible. The main limitation of A* is that as it has to save

entire open list so it occupies more space in some cases for such as eight queen problem. As it needs more space it still seems quite infeasible to use in practical navigation systems. However if algorithm is improvised it can provide you the essential functionalities of navigation.

8. CONCLUSION

The conclusion of this work is that we can now design a knowledge base system which will be built on the study of human's behavior and adopting the human intelligence in the machine. It will help in designing the efficient, fast searching artificial intelligence algorithms with optimization. It will provide the structured database, each place with its short description, timings, ratings etc with GPS and GPRS. The geographical database of the places should also be built in order to provide proper navigation system. The system will provide the information such as description about POI's, timings, cost evaluation and shortest path finding.

9. REFERENCES

- [1] Intelligent Tourist Information System Project [online] <http://projectsgeek.com/2016/08/intelligent-tourist-information-system.html>
- [2] Intelligent Tourist Information System. [online] <http://www.projectsparadise.com/intelligent-tourist-information-system/#>
- [3] Korf, Richard E. "Iterative-deepening-A: an optimal admissible tree search." Proceedings of the 9th international joint conference on Artificial intelligence-volume 2. Morgan Kaufmann Publishers Inc., 1985.
- [4] Russell, Stuart, Peter Norvig, and Artificial Intelligence. "A modern approach." Artificial Intelligence. Prentice-Hall, Englewood Cliffs 25 (1995): 27.
- [5] Introduction to the A* Algorithm [online] <http://mnemstudio.org/path-finding-a-star.htm>
- [6] Introduction to A - Stanford CS Theory [online] <http://theory.stanford.edu/~amitp/GameProgramming/AStarComparison.html>
- [7] Intelligent tourist system project | NevonProjects [online] nevonprojects.com/intelligent-tourist-system-project/
- [8] Intelligent Tourism Management System [online] <https://www.researchgate.net/publication/305426547>
- [9] Russell, Stuart, Peter Norvig, and Artificial Intelligence. "A modern approach." Artificial Intelligence. Prentice-Hall, Englewood Cliffs 25 (1995): 27.
- [10] Rich, Elaine, and Kevin Knight. "Artificial intelligence." McGraw-Hill, New (1991).