ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

Methodical Path Planning of Road by Caching

Fathima Farseen PK

P.G Scholar, Dept. of Computer Science and Engineering, Cochin College of Engineering, Kerala, India

ABSTRACT: Path planning is a basic operation of route navigation services. It finds out route between our required starting place and ending place. Now there are number of applications is present like GPS and digital mapping. But due to unexpected variation in driving direction,losing of GPS signal like many issues we need to propose this path planning method. We propose a system, cash based path planning method. It respond to us with respect our query, also it return result that previously queried that stored in our database. Our system technique decreases almost fifty percentage of computation latency on average.

KEYWORDS: Cache, Path planning

I. INTRODUCTION

In advance of GPS and other path finding methods we propose another system technique is path planning by cache supported.Due to unexpected variation in driving direction,losing of GPS signal like many issues we need to propose this path planning method. It find rout network between specified starting and ending places. Also we can store our queries in database and can use in later. In our cache supported technique, cached query will result when it completely matches to latest query. So system only needs to compute in case of unmatched query. Therefore here we can reduce the overall system latency.

Benefaction of our work includes:

- Path planning by caching (PPC) that return answer for path planning queries, here we save approximately 32 % of time by comparing with other path finding systems.
- Ppattern, path that cached will share segments with other path
- In the basis of user priority of roads on different types, we also introduced cache replacement operation

II. RELATED WORK

In this section we discuss about the related work in this domain. Here mention three related work based on this research area. That is:

- **Shared execution of path queries on road networks:** In this paper, Path queries that find the shortest path between a source and a destination of the user[1]. In particular, they address the problem of finding the shortest paths for a large number of simultaneous path queries in road networks. Traditional systems that consider one query at a time are not suitable for many applications due to high computational and service costs. These systems cannot guarantee required response time in high load conditions. Here propose an efficient group based approach that provides a practical solution with reduced cost. The key concept of this approach is to group queries that share a common travel path and then compute the shortest path for the group. Experimental results show that thias pproach is on an average ten times faster than the traditional approach in return of sacrificing the accuracy by 0.5% in the worst case, which is acceptable for most ofthe users.
- **An efficient path computation model for hierarchically structured topographical road maps:** In this paper, They developed a HiTi (Hierarchical MulTi) [4] graph model for structuring large topographical road maps to speed up the minimum cost route computation. They propose a shortest path algorithm named SPAH, which utilizes HiTi graph model of a topographical road map for its computation. Our performance analysis of SPAH on grid graphs showed that it significantly reduces the search space over existing methods. We also present an in-depth experimental analysis of HiTi graph method by comparing it with other similar works on grid graphs. Within the HiTi graph framework, they also propose parallel shortest path algorithm named

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

ISPAH. Results shows that inter query shortest path problem provide more opportunity for scalable parallelism than the intra query shortest path problem.

- **Exact and approximate distances in graphs – a survey:** They considerconsider many different settings and models and try to identify some remaining open problems.
- **Implementations of Dijkstra's algorithm based on multi-level buckets:** A 2-level bucket data structure has been shown to perform well in a Dijkstra's algorithm implementation [2], [3]. In this paper they study how the implementation performance depends on the number of bucket levels used. In particular we are interested in the best number of levels to use in practice.
- **heuristic determination of minimum cost paths:** The problem of determining the minimum cost path through a graph arises naturally in a number of interesting applications, there has been no underlying theory to guide the development of efficient search procedures[5]. This paper describes how heuristic information from the problem domain can be incorporated into a formal mathematical theory of graph searching and demonstrates an optimality property of a class of search strategies.

III. PROPOSED SYSTEM

A. Advantage of proposed system

In previous section we saw many disadvantages. Summary of that are, A cached query is returned only when it matches completely with a new query.The time complexity is high.The cache content may not be up to date to respond to recent trends in issued queries.The cost of constructing a cache is high, since the system must calculate the benefit values for all sub-paths in a full-path of query results.To solve these we implement our system. That solves most of the disadvantages of existing systems. Let's see the advantage of our system:

- PPC leverages partially matched queried-paths in cache to answer part(s) of the new query. As a result, the server only needs to compute the unmatched path segments, thus significantly reducing the overall system workload.
- It efficientlyanswers a new path planning query by using cached paths to avoid undergoing a time-consuming shortest path computation.
- On average, we save up to 32 percent of time in comparison with a conventional path planning system (without using cache).
- We introduce the notion of PPattern, i.e., a cached path which shares segments with other paths. PPC supports partial hits between PPatterns and a new query.
- A novel probabilistic model is proposed to detect the cached paths that are of high probability to be a PPattern for the new query based on the coherency property of the road networks.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

B. System architecture

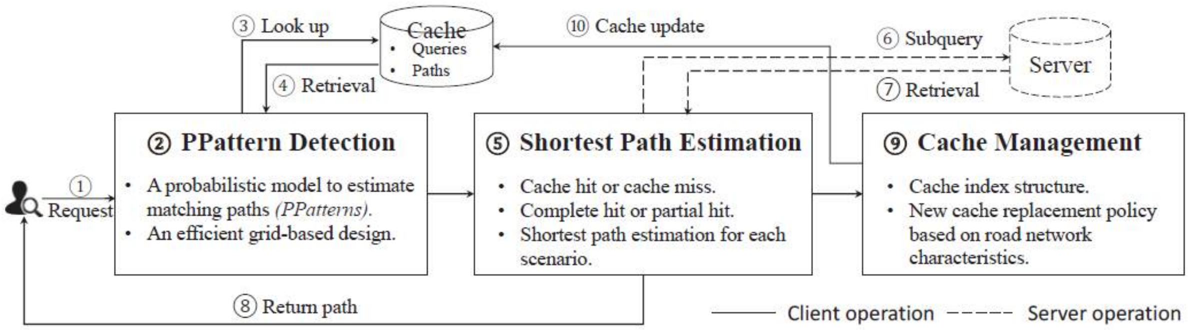

Fig. 1: overview of the ppc system

Fig.1 represents the overall functioning of our system. It mainly consist of three components. That are (i) **PPattern Detection (path pattern), (ii) Shortest Path Estimation, and (iii) Cache Management.**Here firstly find out and return number of historical path in cache, That is called PPattern. Fig.2 shows an example of illustration of PPattern(PPatternis another path which shares at leasttwo consecutive nodes).

Fig.2: Example of illustration of PPattern

In figure the paths $p_{s,t}$and $p_{s',t'}$ are2-PPattern to each other as they share a common sub-path $p_{a,b}$. Node *a*is the pattern head and node *b* is the pattern tail.

In shortest path estimation part, that build candidate path for new query and select shortest one. Here if PPattern is same as query, system immediately returns to user. In other case system compute unmatched path segment between PPattern and query. And if cache is full cache management part determine which queried-paths in cache should be deleted.

IV. IMPLEMENTATION

When implementing our paper, itincludes mainly four modules. That is**system construction Module, Probabilistic model for PPattern detection,an efficient grid-based solution and cache construction and update**.

A. System Construction Module

In system constructionmodule, we develop the system with the required entities to implement our proposed model and evaluate the effectiveness of the system. The main goal in this work is to reduce the server workload by leveraging the queried-paths in cache to answer a new path planning query. An intuitive solution is to check whether there exists a cached queried-path perfectly matching the new query. Here, a perfect match means that the source and destination nodes of the new query are the same as that of a queried-path in cache.

DOI: 10.15680/IJIRCCE.2017. 0503049

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

B. Probabilistic model for PPattern detection

To detect the best PPatterns, an idea is to calculate the estimation distance based on each cached path, and select the cached path with the shortest distance. It faces many challenges. Firstly, the distance estimation requires the server to compute the unshared segments. Therefore, it incurs significant computation to exhaustively examine all cached paths.Secondly, such an exhaustive operation implicitly assumes that each cached path is a PPattern candidate to the query.

Algorithm of PPattern detection:

STEP 1: If distance between nodes s(source) and t(destination) less than threshold(D_l) value, return the result
PPattern = NULL
STEP 2: Divide the target space by grid cell size
STEP 3: Find out source(g_s) and destination(g_t) grid
STEP 4: Assign Q_s = Logged queries whose paths pass start grid(g_s)
AndQ_t = Logged queries whose paths pass destination grid(g_t)
STEP 5: Store the intersect value of Q_s and Q_t into Q
STEP 6: Assign path from source grid to destination grid for each query into PT
STEP 7: Return cached path(PT).

C. Efficient Grid Based Solution

To retrieve these patterns, we invent a grid-based solution to further improve the system efficiency. Here divide the whole space into equally sized grid cells, here endpoints of all paths are mapped to the grid cells. By counting the total number of covered grids we will get the distance measure.

D. Cache Construction and Update

It is an important part of cache management. Here we invent a cache replacement method by taking unique characteristics of road pattern. By observing many users, we found that certain routes are selected by most of the users. Most of them select main roads than branch roads. Because of theefficiency, popularity and capacity of these major roads. In a road network $G = (V,E)$, each edge from node $v_m$to $v_n$is associated with a weight $W_{m,n}$it is a system computed value curresponding to the road type. If $W_{m,n}$ is highindicates that the corresponding road type is high.

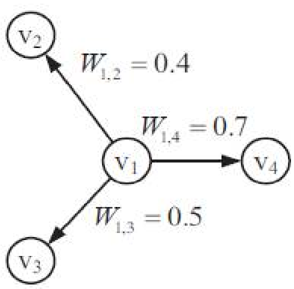

Fig.3: Example for edge and node weight

Fig. 3 shows an example for node and edge weight to road types. Here edges are connected with node v_1. The weights of three edges are $W_{1,2}$= 0.4, $W_{1,3}$= 0.5, and $W_{1,4}$= 0.7, respectively. The weight of $W_1$is set to $W_{1,4}$= 0.7.Nodeweight represents howpath planning querywith it as the source node to be issued later. Information like this can be used to propose the cache replacement policy.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

Algorithm of PPattern detection:

STEP 1: Assign PPatterns Detection into variable PT.
STEP 2: Assign Shortest Path Estimation from PTinto variable p.
STEP 3: **if** cache C is not full then insert p into cache and return C
STEP 4:Otherwise calculate usability for cached path and stored into {μ} and path with minimum usability and stores It into p*
STEP 5: Check whether the usability value of the current path *p* is larger than the minimum usability value in the current cache. If so, we place the current query into C. and return C.

V.CONCLUSION

In this paper we implemented a cached path planning method. Also it solves all disadvantage of existing system. That is, our implemented system reduces the time complexity, Cached query is returned when it partially matches with a new query also, Cache content is up to date and also cost of constructing cache is low. Here server only needs to calculate unmatched pathsegments. So here workload of system is very law. That is, oursystem reducesthe system latency almost 32%.

REFERENCES

1. H. Mahmud, A. M. Amin, M. E. Ali, and T. Hashem, "Shared Executionof Path Queries on Road Networks", Clinical Orthopaedicsand Related Research, vol. abs/1210.6746, 2012.
2. W. Dijkstra, "A Note on Two Problems in Connexion with Graphs", Numerische Mathematik, vol. 1, no. 1, pp. 269–271, 1959.
3. U. Zwick, "Exact and approximate distances in graphs – a survey", in Algorithms – ESA 2001, 2001, vol. 2161, pp. 33–48.
4. S. Jung and S. Pramanik, "An Efficient Path Computation Modelfor Hierarchically Structured Topographical Road Maps", IEEETransactions on Knowledge and Data Engineering, vol. 14, no. 5, pp.1029–1046, 2002.
5. P. Hart, N. Nilsson, and B. Raphael, "A Formal Basis for theHeuristic Determination of Minimum Cost Paths",IEEE Transactionson Systems Science and Cybernetics, vol. 4, no. 2, pp. 100–107, 1967.
6. L. Zammit, M. Attard, and K. Scerri, "Bayesian Hierarchical Modellingof Traffic Flow - With Application to Malta's Road Network", in International IEEE Conference on Intelligent TransportationSystems, 2013, pp. 1376–1381.
7. S. Jung and S. Pramanik, "An Efficient Path Computation Modelfor Hierarchically Structured Topographical Road Maps", IEEE Transactions on Knowledge and Data Engineering, vol. 14, no. 5, pp. 1029–1046, 2002.