**Efficient Cache-Supported Path Planning on Roads**

**ABSTRACT:**

Owing to the wide availability of the global positioning system (GPS) and digital mapping of roads, road network navigation services have become a basic application on many mobile devices. Path planning, a fundamental function of road network navigation services, finds a route between the specified start location and destination. The efficiency of this path planning function is critical for mobile users on roads due to various dynamic scenarios, such as a sudden change in driving direction, unexpected traffic conditions, lost or unstable GPS signals, and so on. In these scenarios, the path planning service needs to be delivered in a timely fashion. In this paper, we propose a system, namely, Path Planning by Caching (PPC), to answer a new path planning query in real time by efficiently caching and reusing historical queried-paths. Unlike the conventional cache-based path planning systems, where a queried-path in cache is used only when it matches perfectly with the new query, PPC leverages the partially matched queries 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. Comprehensive experimentation on a real road network database shows that our system outperforms the state-of-the-art path planning techniques by reducing 32 percent of the computation latency on average.

**EXISTING SYSTEM:**

* Path planning needs to be delivered in a timely fashion. The requirement of timeliness is even more challenging when an overwhelming number of path planning queries is submitted to the server, e.g., during peak hours. As the response time is critical to user satisfaction with personal navigation services, it is a mandate for the server to efficiently handle the heavy workload of path planning requests.
* Jung and Pramanik propose the HiTi graph model to structure a large road network model. HiTi aims to reduce the search space for the shortest path computation. While HiTi achieves high performance on road weight updates and reduces storage overheads, it incurs higher computation costs when computing the shortest paths than the HEPV and the Hub Indexing methods.
* To compute time-dependent fast paths, Demiryurek et al. propose the B-TDFP algorithm by leveraging backward searches to reduce the search space. It adopts an area-level partition scheme which utilizes a road hierarchy to balance each area.

**DISADVANTAGES OF EXISTING SYSTEM:**

* 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.

**PROPOSED SYSTEM:**

* To meet existing need, we propose a system, namely, Path Planning by Caching (PPC), that aims to answer a new path planning query efficiently by caching and reusing historically queried paths (queried-paths in short).
* The proposed system consists of three main components: (i) PPattern Detection, (ii) Shortest Path Estimation, and (iii) Cache Management.
* Given a path planning query, which contains a source location and a destination location, PPC firstly determines and retrieves a number of historical paths in cache, called PPatterns, that may match this new query with high probability.
* The idea of PPatterns is based on an observation that similar starting and destination nodes of two queries may result in similar shortest paths (known as the path coherence property).
* In the component PPatern Detection, we propose a novel probabilistic model to estimate the likelihood for a cached queried-path to be useful for answering the new query by exploring their geospatial characteristics.
* To facilitate quick detection of PPatterns, instead of exhaustively scanning all the queried paths in cache, we design a grid-based index for the PPattern Detection module. Based on these detected PPatterns, the Shortest Path Estimation module (see Steps (5)-(8)) constructs candidate paths for the new query and chooses the best (shortest) one.
* In this component, if a PPattern perfectly matches the query, we immediately return it to the user; otherwise, the server is asked to compute the unmatched path segments between the PPattern and the query (see Steps (6)-(7)). Because the unmatched segments are usually only a smaller part of the original query, the server only processes a “smaller subquery”, with a reduced workload.
* Once we return the estimated path to the user, the Cache Management module is triggered to determine which queried-paths in cache should be evicted if the cache is full. An important part of this module is a new cache replacement policy which takes into account the unique characteristics of road networks.
* In this paper, we provide a new framework for reusing the previously cached query results as well as an effective algorithm for improving the query evaluation on the server.

**ADVANTAGES OF PROPOSED 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.
* We propose an innovative system, namely, path planning by caching, to efficiently answer 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. Our experiments indicate that partial hits constitute up to 92.14 percent of all cache hits on average.
* 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. Our experiments indicate that these PPatterns save retrieval of path nodes by 31.69 percent on average, representing a 10-fold improvement over the 3.04 percent saving achieved by a complete hit.
* We have developed a new cache replacement mechanism by considering the user preference among roads of various types. A usability measure is assigned for each query by addressing both the road type and query popularity. The experimental results show that our new cache replacement policy increases the overall cache hit ratio by 25.02 percent over the state-of-the-art cache replacement policies.

**SYSTEM SPECIFICATION**

**Hardware Requirements:**

* System : Pentium IV 3.4 GHz (Min)or Later versions.
* Hard Disk : 40 GB.
* Monitor : 14’ Colour Monitor.
* Mouse : Optical Mouse.
* Ram : 1 GB.(Min)

**Software Requirements:**

* Operating system : Windows Family.
* Coding Language : J2EE (JSP,Servlet,Java Bean)
* Data Base : MY Sql Server.
* IDE : Eclipse Juno
* Web Server : Tomcat 6.0