

Propuesta de diseño de un índice espacio-temporal vinculado a redes móviles

Daniela C. Giraudi, Gabriela S. Segura Guzmán, Edilma O. Gagliardi

Universidad Nacional de San Luis, Facultad de Ciencias Físico,

Matemáticas y Naturales, Departamento de Informática

San Luis, Argentina, D5700HHW

{dcgiraud, ggsegura, oli}@unsl.edu.ar

Gregorio Hernández Peñalver

Universidad Politécnica de Madrid, Facultad de Informática

Madrid, España

gregorio@dma.fi.upm.es

Resumen

Nuestra investigación consistió en proveer la manera en que dada una red, total o parcialmente conocida, modelada como un grafo geométrico, sobre la que se ubican objetos en movimiento, sea posible encontrar una ruta desde una posición origen a una posición destino mediante diversas heurísticas, basadas en conocimientos parciales de la red subyacente, con la característica principal de considerar la definición de la red según el índice espacio-temporal I+MON-Tree. Proponemos un posible diseño de extensión a la estructura I+MON-Tree en relación a las redes móviles. De este modo, se pretende mantener registro de los objetos en movimiento que exploran su ambiente de trabajo a medida que van descubriendo la ruta hacia su destino, basándose en diversas heurísticas de búsqueda que están delineadas en función de las poligonales (rutas) definidas sobre la red. Presentamos una primera versión de diseño que permitiría considerar dichas redes. A esta propuesta la hemos llamado *I+MON-Tree-RM*.

En esta parte, nos abocaremos a una propuesta de diseño, y dejamos de lado el desarrollo su correspondiente refinamiento, implementación y evaluación experimental del buen desempeño de la estructura.

Palabras claves: Estructuras de datos, índices espacio-temporales, redes móviles, Geometría Computacional.

1. INTRODUCCIÓN

El índice espacio-temporal I+MON-Tree [COG06] permite almacenar y recuperar información histórica y actual de objetos en movimiento sobre redes fijas. Este método de acceso indexa objetos que se mueven sobre redes conocidas, asumiendo que los mismos tienen la capacidad de informar las coordenadas y el tiempo en el que arriban a una nueva posición. Esta estructura, además de mantener información histórica de los objetos, tiene la capacidad de almacenar y consultar información de la posición actual de los mismos. Este método surgió por la necesidad de que ciertas aplicaciones requieren conocimiento acerca de la última posición en la que el objeto se encuentra y/o por la necesidad de conocer la trayectoria del mismo, basado en el MON-Tree [AG04] [AG04a].

En nuestra línea de investigación, vimos que al estudiar ruteo, se podía proveer la manera en que dada una red, total o parcialmente conocida, modelada como un grafo geométrico, sobre la que se ubican objetos en movimiento, sea posible encontrar una ruta desde una posición origen a una posición destino mediante diversas heurísticas, basadas en conocimientos parciales de la red subyacente, con la característica principal de considerar la definición de la red según el índice espacio-temporal I+MON-Tree.

Así fue necesario plantear un posible diseño de extensión a la estructura I+MON-Tree en relación a las redes móviles, de modo que se pueda mantener registro de los objetos en movimiento que exploran su ambiente de trabajo a medida que van descubriendo la ruta hacia su destino, basándose en diversas heurísticas de búsqueda que están delineadas en función de las poligonales (rutas) definidas sobre la red y también de la red misma, dado que es dinámica en su estructura.

Por ello, presentamos una primera versión de diseño que permitiría considerar dichas redes. A esta propuesta la hemos llamado *I+MON-Tree-RM*, cuyos desarrollos y evaluaciones experimentales para medir su desempeño corresponden a etapas posteriores a la presentada en este artículo.

2. ANTECEDENTES

Antes de presentar el diseño propuesto, brindamos algunos detalles del índice tomado como punto de partida, el cual incluye la definición del MON-Tree [AG04a]. También, agregamos una presentación acerca de un índice espacio-temporal, I+3 R-TREE [CGGG06], estructura basada en el 2+3R-Tree.

2.1 I+MON-TREE

I+MON-Tree es un índice espacio-temporal que tiene capacidad para almacenar y recuperar información histórica y actual de objetos en movimiento. Este método de acceso indexa objetos que se mueven sobre redes conocidas, asumiendo que los mismos tienen la capacidad de informar las coordenadas y el tiempo en el que arriban a una nueva posición. Esta estructura, además de mantener información histórica de los objetos, tiene la capacidad de almacenar y consultar información actual de los mismos. Este método surgió por la necesidad de que ciertas aplicaciones requieren conocimiento acerca de la última posición en la que el objeto se encuentra y/o por la necesidad de conocer la trayectoria del mismo.

Las características de I+MON-Tree son las siguientes:

- 1 Utiliza un conjunto de estructuras, que almacenan el movimiento de los objetos, en donde tanto los intervalos de posición y tiempo están cerrados. Es decir, mantienen la trayectoria o estados pasados de los objetos en movimiento.

- 2 Utiliza una estructura adicional que tiene por finalidad almacenar la información actual de los objetos, donde tanto los intervalos de posición y tiempo están abiertos, es decir, se desconoce el tiempo final de permanencia en una posición. Cuando un objeto se mueve a una nueva posición p_1 en el tiempo t_1 , su intervalo de tiempo es $[t_1, *)$, ésta información se almacena en este índice. Cuando el objeto se desplaza a una nueva posición p_2 en el tiempo t_2 , el intervalo de tiempo en que el objeto estuvo en p_1 es $[t_1, t_2)$, y los intervalos $[p_1, p_2]$ y $[t_1, t_2)$ se almacenan en las estructuras que mantienen información sobre estados pasados de los objetos y el intervalo de duración en p_2 que es $[t_2, *)$, es la nueva información correspondiente al objeto que se registra en el índice de información actual.

La estructura I+MON-Tree, asume que los objetos se mueven a lo largo de poligonales.

Estos pueden estar asociados a arcos, en el caso de que se esté trabajando con el primer modelo de red, o pueden estar asociados a rutas, si el modelo de red con el que se está trabajando es el segundo.

El índice I+MON-Tree consiste de:

- 1 Un conjunto de estructuras utilizado para almacenar la información histórica de los objetos, denominado *Índice de Información Histórica*, de ahora en más *IIH*; es decir, los estados pasados de los mismos. Se distinguen dos niveles: en el nivel superior se encuentran dos estructuras, un 2DR-Tree y una estructura de hashing; en el nivel inferior se encuentra un conjunto de 2DR-Trees.

A continuación, detallamos cada nivel.

- Nivel Superior:

- 1 *2DR-Tree*, denominado R_S : indexa la mínima caja que recubre cada poligonal. La información almacenada en los nodos de la estructura es la siguiente:

Nodos Internos $\langle mbb, childpt \rangle$, donde mbb es el MBB que contiene a todas las MBBs de las entradas en el vértice hijo, y $childpt$ es un puntero al vértice hijo.

Nodos Hojas $\langle mbb, polypt, treept \rangle$, almacena la información de la poligonal, donde mbb es la MBB que la contiene, $polypt$ apunta a la representación real de la misma y $treetpt$ es un puntero al respectivo R-Tree de aquella.

- 2 *Hashing*, denominado H : con los campos $\langle polyid, ptrrt \rangle$, donde $polyid$ almacena el identificador de la poligonal y $ptrrt$ el puntero al 2DR-Tree del nivel inferior que le corresponde a la misma.

- Nivel Inferior:

- 3 Un conjunto de *2DR-Tree*, denominados R_n , donde n es el número de R-Trees: éstos indexan el movimiento de los objetos a lo largo de la poligonal. La información almacenada en los vértices de la estructura es la siguiente:

Nodos Internos $\langle mbb, childpt \rangle$, donde el mbb es el MBB que cubre los MBBs de los vértices hoja, $childpt$ es un

puntero a los vértices hijos.

Nodos Hojas $\langle moid, mbb, polyid, [p_1, p_2], [t_1, t_2], p_tray_back, p_tray_next \rangle$, donde *moid* es el identificador del objeto móvil, *mbb* es el MBB que cubre el movimiento, *polyid* es el identificador de la poligonal sobre la cual se produjo el movimiento, los intervalos $[p_1, p_2]$ y $[t_1, t_2]$ corresponden a la posición y tiempo respectivamente, en el cual se produjo el movimiento, *p_tray_back* almacena la información relacionada al lugar dentro de la estructura donde se guarda la información correspondiente al movimiento anterior del objeto y finalmente *p_tray_next* almacena la información relacionada al lugar, dentro de la estructura, donde se guarda la información correspondiente al movimiento posterior del objeto.

2 *Índice de Información actual*, denominado *IIA*: estructura que almacena cubos abiertos, que hacen referencias a objetos cuyo instante final de permanencia en una posición aún no está definido. También se mantiene información que hace referencia a los cubos anteriores que permite recuperar la trayectoria del objeto. La información almacenada es $\langle moid, mbb, polyid, p_i, t_i, p_next, p_back, p_tray \rangle$, en donde: *moid* es el identificador del objeto, *mbb* es una caja construida con la información de las coordenadas correspondientes a la posición informada por el objeto, *polyid* es el identificador de la poligonal en la que se encuentra el objeto, p_i es la posición dentro de la poligonal donde se haya el objeto, t_i es el tiempo de arribo a la posición, *p_next* es el puntero al objeto con tiempo posterior inmediato de arribo a una posición, que se encuentra en la estructura, *p_back* es el puntero al objeto con tiempo anterior inmediato de arribo a una posición, que se encuentra en la estructura, y *p_tray* puntero al cubo anterior, almacenado en *III*, utilizado para mantener la trayectoria del objeto.

La estructura *I+MON-Tree* es capaz de responder cuatro tipos de consultas: consulta por ventana, consulta por rango, consulta por instante de tiempo o *TimeSlice* y consulta por trayectoria.

CONSULTA POR VENTANA O WINDOW QUERY

Este tipo de consulta consiste en recuperar todos los objetos que estuvieron o están en un área denominada ventana o sub-espacio, en un intervalo de tiempo dado.

El algoritmo recibe una ventana espacial y un intervalo de tiempo y procede de la siguiente manera. Lo primero es verificar si es posible que en el *III* se encuentre información que deba ser devuelta en esta consulta. En caso de que sea así, la existencia de tal información en las estructuras destinadas a almacenar estados pasados, se realiza una consulta por ventana en las mismas, recuperando los datos de los objetos que pasaron por ese espacio en el intervalo de tiempo dado. Es decir, se recuperan datos históricos.

Tanto en el caso de que no haya posibilidades de que se encuentre información de interés para esta consulta, como en el que ya se haya recuperado la información histórica, el siguiente paso es recuperar del *IIA* los objetos cuya posición interseca el espacio consultado en el intervalo de tiempo dado, verificando previamente de que sea posible que en dicho índice se haya registrado información útil para la consulta.

CONSULTA POR RANGO O RANGE QUERY

Este tipo de consulta consiste en recuperar los identificadores de aquellos objetos que estuvieron o están en un área denominada ventana o sub-espacio, en un intervalo de tiempo dado.

El algoritmo para este tipo de consulta recibe los mismos argumentos y procede de la misma manera que el algoritmo de consulta por ventana antes descrito, con un paso adicional que es la eliminación de los objetos duplicados.

CONSULTA POR INSTANTE DE TIEMPO O TIMESLICE QUERY

Este tipo de consulta consiste en recuperar todos los objetos que estuvieron o están en un área denominada ventana o sub-espacio, en un instante de tiempo dado.

Este algoritmo es muy similar al de la consulta por ventana, pero a diferencia de ésta, el argumento recibido no es un intervalo de tiempo sino un instante de tiempo particular y procede de la siguiente manera. Primero, el algoritmo verifica si es posible que haya objetos dentro de la estructura de estados pasados que se hayan movido en ese tiempo. Si esto es así, será útil la búsqueda en ese lugar y, por ende, se realiza una consulta de manera similar a la consulta por ventana, pero consultando sólo por el instante de tiempo dado.

Ya sea que se realice una búsqueda en *III* o no, el siguiente paso es realizar la misma verificación en la estructura de los estados actuales para determinar la existencia de objetos que intersequen el espacio consultado en el instante de tiempo dado.

CONSULTA POR TRAYECTORIA O TRAJECTORY QUERY

Este tipo de consulta consiste en recuperar las posiciones que un objeto ha ocupado.

La recuperación de la trayectoria puede ser *parcial*, *completa* y por *intervalo de tiempo*. En la trayectoria parcial se informan las rutas o aristas que atravesó el objeto. La trayectoria completa, consiste en devolver todas las posiciones que ocupó el objeto desde el primer al último movimiento informado por el mismo. Por su parte, la trayectoria por intervalo de tiempo recibe un intervalo temporal, y devuelve la trayectoria recorrida por ese objeto durante el tiempo estipulado.

El algoritmo para esta consulta recibe como argumento el identificador del objeto del que se desea conocer la trayectoria y procede de la siguiente manera. Primero, recupera de la estructura de estados actuales, el último movimiento realizado por el objeto, accede al campo que almacena la trayectoria y, siguiendo las referencias, accede a los movimientos almacenados en la estructura de estados pasados conformando de esa manera la trayectoria del objeto.

2.2 I+3 R-TREE

Este método de acceso [CGGG06] es un índice espacio-temporal que tiene la capacidad de almacenar y recuperar información histórica y actual de objetos que se mueven en espacios libres de restricciones.

El I+3 R-Tree se basa en el método expuesto en [Nst99] que al igual que la idea original, mantiene los datos históricos en un 3D R-Tree, y reemplaza el R-Tree por una estructura denominada *índice*, denotada con *I*, la cual se utiliza para guardar datos de la misma índole.

El I+3 R-Tree está enfocado para aquellas aplicaciones en las cuales la cantidad de objetos en movimiento es fija y el espacio donde éstos se mueven es acotado y conocido. Se asume que los objetos son capaces de informar, en forma discreta las coordenadas y el tiempo en que el objeto alcanzó una nueva posición espacial.

Una de las principales desventajas del 3D R-Tree, en donde se requiere conocer los intervalos de tiempo de manera anticipada, con el I+3 R-Tree queda salvada. Esto es posible debido a que ésta estructura permite almacenar y realizar consultas sobre datos actuales.

La estructura del I+3 R-Tree, recupera información histórica y actual de objetos que se mueven en espacios libres.

El I+3 R-Tree está formado por un 3D R-Tree el cual almacena la información pasada de los objetos y una estructura denominada *Índice* que almacena la información actual de los objetos.

El *Índice I* almacena los *cubos abiertos*. Esto es, aquellos cubos para los cuales su techo aún no está definido. También se guardan las referencias necesarias a los cubos anteriores que describen la trayectoria del objeto.

El índice *I* es una lista secuencial de *N* elementos, donde *N* es la cantidad de objetos considerados, que almacena nuplas de la forma $\langle oid, mbr, t, p3D, pa, ps \rangle$ donde:

- *oid*: código identificador del objeto;
- *mbr*: región aproximada que ocupa actualmente el objeto;
- *t*: tiempo de llegada del objeto a su ubicación actual;
- *p3D*: puntero al cubo anterior correspondiente al mismo *oid*, utilizado para mantener un historial de trayectoria;
- *pa*: puntero al objeto insertado en el instante de tiempo inmediatamente anterior;
- *ps*: puntero al objeto insertado en el instante de tiempo siguiente.

Un ejemplo de *Índice* se muestra en la siguiente figura.

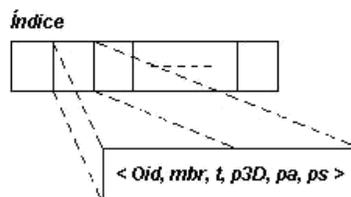


Figura 1. Estructura del Índice.

La lista secuencial se encuentra ordenada por el *oid*; como la cantidad de objetos que se almacenan es fija, esta lista puede ser accedida como un direccionamiento directo. Esta lista se mantiene en memoria principal.

Los punteros *pa* y *ps* son utilizados para enlazar los tiempos en orden creciente. Este ordenamiento por tiempo tiene como objetivo acceder de manera directa a un instante de tiempo determinado en el *índice*.

El 3D R-Tree almacena la información histórica de los objetos, esto permite recuperar la trayectoria de los mismos. Esto consiste en mantener una lista para cada objeto enlazando en el 3D R-Tree los cubos correspondientes a las diferentes posiciones por las cuales ha pasado el objeto. El acceso a estas listas se realiza a través del *Índice*, desde la posición actual de cada objeto, mediante un direccionamiento directo, ya que la cantidad de objetos es fija.

El 3D R-Tree mantiene una estructura similar a la del R-Tree con la diferencia que mantiene una dimensión más y se almacenan los *cubos definidos*, cada uno de los cuales representa la estadía de un objeto en una posición durante un intervalo de tiempo definido. Es decir, toda la información espacio – temporal histórica se mantiene en el 3D R-Tree. Las nuplas almacenadas en el 3D R-Tree son del tipo $\langle oid, mbr_3D, p_tray \rangle$ donde:

- *oid*: código identificador del objeto;
- *mbr_3D*: región tridimensional cuya altura representa el intervalo temporal durante el cual el objeto se mantuvo en la posición espacial definida por su base;
- *p_tray*: puntero al cubo anterior correspondiente al mismo *oid*, utilizado para mantener un historial de la trayectoria.

Un ejemplo de la estructura del I+3 R-Tree se muestra en la siguiente figura.

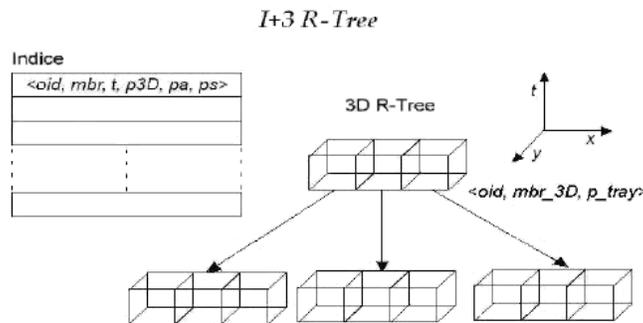


Figura 2. Estructura general del I+3 R-Tree.

Como se puede observar en la figura anterior, el I+3 R-Tree está conformado por un *Índice*, para almacenar la información actual, y por un *3D R-Tree*, para almacenar la información histórica.

3. PROPUESTA DEL I+MON-TREE-RM

Recordando la definición del I+MON-Tree, vimos que existen dos tipos diferentes de modelos de redes que indexa este índice. El primer modelo es orientado a arcos, es decir, la red se compone de aristas y vértices, donde cada arista tiene asociada una poligonal. El segundo modelo está orientado a rutas, es decir, la red se compone de un conjunto de rutas y un conjunto de intersecciones entre esas rutas. De todo esto, rescatamos fundamentalmente la posibilidad de determinar poligonales constituidas sobre la red. Entonces, en lo subsiguiente nos abocaremos a considerar dichas poligonales, y no haremos discriminación por modelo.

Para comprender un poco más, veamos el siguiente ejemplo. Supongamos que el objeto *o* transita sobre la red con destino al vértice *d*. Es claro que, en el camino hacia su destino éste puede atravesar una o varias poligonales. Cuando el objeto arriba a un vértice, éste informa su llegada de manera que se registra el instante de tiempo en que arribó a ese punto. De acuerdo a este registro y conociendo cuál es su próximo destino inmediato, podemos conocer cuál es la posición relativa del mismo. Veamos la siguiente figura:

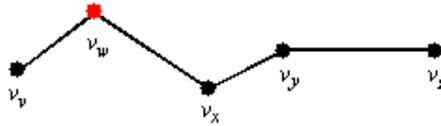


Figura 3. El objeto o arriba al vértice v_w en un instante de tiempo

En la figura vemos que el objeto se registra en el vértice v_w y transita sobre esta poligonal en dirección al vértice inmediato v_x .

Como vimos anteriormente, los R-Tree's inferiores indexan el movimiento de los objetos que transitan sobre las poligonales que constituyen la red. Este movimiento se representa por medio de un intervalo de posición ($p_1; p_2$) y un intervalo de tiempo ($t_1; t_2$), donde 0 representa el instante de tiempo en que el objeto estuvo en la posición p_1 y 1 representa el instante de tiempo en que el objeto estuvo en la posición p_2 . Así, p_1 y p_2 almacenan la posición relativa del objeto dentro de la poligonal en el instante t_1 y t_2 , respectivamente.

Es decir, el objeto o registró su arribo en el vértice v_w en el instante t_i . En un instante de tiempo t_j , donde $i < j$, siendo que el objeto se dirige hacia el vértice v_x podemos conocer su posición relativa.

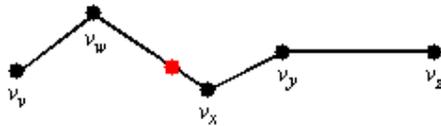


Figura 4. El objeto transita sobre la poligonal

Como vemos en la figura anterior, el objeto transita sobre la poligonal. En ese instante de tiempo no se conoce cuál es la posición física del objeto, sino que podemos conocer cuál es la posición relativa respecto del último arribo. Esto significa, que conocemos el porcentaje de movimiento que efectuó el objeto sobre una parte de la poligonal, a partir del último vértice al que arribó. Esta información se almacena en el R-Tree correspondiente a dicha poligonal, donde se almacena el registro de los movimientos de cada uno de los objetos que transitan sobre ella.

Por otro lado, como ya vimos, el índice I+MON-Tree está formado por un Índice de Información Histórica, para almacenar los estados pasados de los objetos; y, de un Índice de Información Actual, para almacenar los estados actuales de los objetos cuyo instante final de permanencia en una posición aún no ha sido definido.

También, debemos tener en cuenta que la red está expuesta a posibles cambios, por lo que esta nueva información debe ser registrada en el índice. En este sentido, nosotros consideramos los cambios desde un punto de vista más general, dirigido a una poligonal, más que a un vértice en particular. Por lo tanto, debemos emplear alguna política de almacenamiento. En la siguiente figura, mostramos un posible diseño de extensión a la estructura I+MON-Tree en relación a las redes móviles.

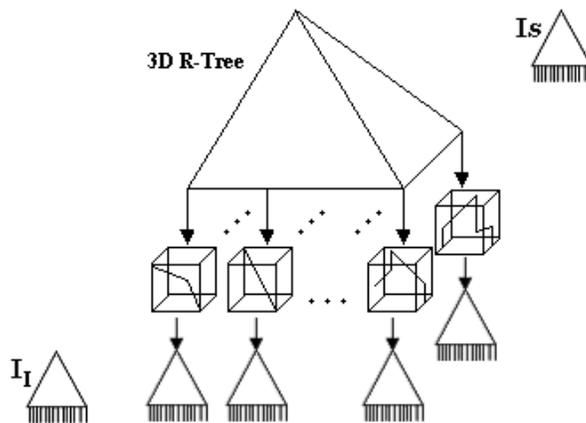


Figura 5. I+MON-Tree-RM

En este sentido, nosotros cambiamos el 2D R-Tree superior de la estructura por un I+3 R-Tree de manera tal de poder mantener los cambios producidos en las poligonales a través del tiempo.

En cada R-Tree inferior del índice I+MON-Tree se almacena información de los objetos que atraviesan la red en un momento dado. En este índice, la red es fija, de manera que, la posición de cada uno de todos los puntos que constituyen las poligonales de la red subyacente se mantienen de manera constante a través del tiempo. En el caso de una red móvil, la misma está sujeta a cambios. Por lo que, el índice debe ser capaz de registrar las modificaciones efectuadas en la red. Para ello proponemos el siguiente diseño:

- El *3D R-Tree* almacena los *cubos definidos*, cada uno de los cuales representa la estadía de la poligonal en una posición durante un intervalo definido de tiempo. Es decir, toda la información espacio – temporal histórica se mantiene en el 3D R-Tree. Las nuplas almacenadas en el 3D R-Tree son del tipo $\langle pid, mbr_3D, p_cubos \rangle$ donde:
 - *pid*: código identificador de la poligonal;
 - *mbr_3D*: región tridimensional cuya altura representa el intervalo temporal durante el cual la poligonal se mantuvo en la posición espacial definida por su base;
 - *p_cubos*: puntero al cubo anterior correspondiente al mismo pid, utilizado para mantener un historial de trayectoria de la poligonal.
- En el índice I_S se almacenan los *cubos abiertos*. Esto es aquellos cubos para los cuales su techo aún no está definido. También se guardan las referencias necesarias a los cubos anteriores que describen los cambios de posiciones de las poligonales.

El índice I_S es una lista secuencial de N elementos, donde N es la cantidad de poligonales consideradas, que almacena nuplas de la forma $\langle pid, mbr, t, p3D, pa, ps \rangle$ donde:

- *pid*: código identificador de la poligonal;
- *mbr*: región aproximada que ocupa actualmente la poligonal;
- *t*: tiempo en que la poligonal cambio sus posiciones a la ubicación actual;
- *p3D*: puntero al cubo anterior correspondiente al mismo pid, utilizado para mantener los cambios físicos de las poligonales;
- *pa*: puntero a la poligonal insertada en el instante de tiempo inmediatamente anterior; y
- *ps*: puntero a la poligonal insertada en el instante de tiempo siguiente. Los punteros *pa* y *ps* son utilizados para enlazar los tiempos en orden creciente.

La idea subyacente es mantener en la estructura I+3 R-Tree todas las modificaciones que se producen en la red, a nivel de poligonales. De esta forma, resulta posible consultar las distintas posiciones de las mismas definidas sobre la red.

En la siguiente figura observamos los cambios físicos de una poligonal en diferentes instantes de tiempo, así es que,

cada cubo que representa la poligonal en el instante de tiempo t_i , tiene asociado un R-Tree inferior con todos aquellos objetos que transitaron por esa poligonal en el tiempo t_i .

Cuando consultamos en esta estructura, primeramente accedemos a la estructura I_s , recordemos que esta mantiene la ubicación actual de la poligonal con un puntero que permite acceder a las ubicaciones anteriores de la misma.

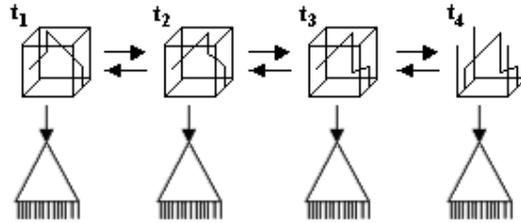


Figura 6. Ejemplo de almacenamiento de una poligonal en diferentes instantes de tiempo

El soporte de consultas es una de las funciones más importantes de un sistema de información. La respuesta a una consulta depende del tipo de consulta, del sistema de información que administra la consulta y de los índices disponibles para dicho sistema de información.

El índice espacio-temporal I+MON-Tree responde a los principales tipos de consultas, estas son: Instante de tiempo, Trayectoria, Ventana y Rango. Al plantear una extensión del índice I+MON-Tree vimos que era necesario realizar un análisis de las modificaciones que deberían efectuarse sobre las consultas que mencionamos anteriormente.

A continuación presentamos cada una de las consultas y señalamos las posibles modificaciones para su adaptación en redes móviles.

Cabe aclarar que, en los diferentes tipos de consultas se mantienen las variables $TMAX3D$ y $TMININD$ con el objetivo de mejorar la eficiencia en el procesamiento de las consultas evitando en algunos casos el acceso innecesario a alguna de las estructuras.

La variable $TMAX3D$ mantiene el tiempo mas reciente registrado en el 3D R-Tree y la variable $TMININD$ mantiene el tiempo más antiguo registrado en el Índice.

CONSULTA POR VENTANA (WINDOW QUERY)

Este tipo de consulta consiste en recuperar todos los objetos que estuvieron o están en un área denominada ventana o sub-espacio, en un intervalo de tiempo dado.

El algoritmo recibe una ventana espacial y un intervalo de tiempo, y busca en el índice, los datos que responden a la consulta.

La ventana de consulta está asociada a un espacio y a un intervalo de tiempo dado, y contiene partes de poligonales, que atraviesan el mismo espacio en esos instantes, incluidos en el intervalo de tiempo. Por tanto, resulta necesario realizar una búsqueda en la estructura I+3 R-Tree para recuperar todos los cubos, los cuáles contienen poligonales, que intersecan con la ventana de consulta.

La manera en que procede la consulta Window Query es la siguiente: identifica las poligonales cuyas posiciones espaciales intersecan el área denominada ventana o sub-espacio en el intervalo de tiempo dado.

Para ello, realiza la búsqueda dentro de la estructura I+3 R-Tree todos los cubos que intersecan con la ventana de consulta. Así, recupera todas las poligonales que están en dicho intervalo de tiempo.

Luego, accede a los R-Tree's inferiores asociados a cada uno de los cubos que contienen poligonales recuperados anteriormente y recupera aquellos objetos cuyo tiempo de permanencia coincida con alguno de los instantes de tiempo del intervalo. En caso de no haber ocurrido cambios en las posiciones espaciales que conforman las poligonales se realiza la búsqueda dentro de los R-Tree's inferiores correspondientes. Ver algoritmos [GSGH07].

CONSULTA POR RANGO O INTERVALO (RANGE QUERY)

El procesamiento de este tipo de consulta es muy sencillo. Se siguen los mismos pasos para procesar la consulta

Window Query, sólo que se incorpora un paso que permite la eliminación de objetos duplicados, en el conjunto de respuesta.

El algoritmo procede de la siguiente manera: aplica Window Query, así obtiene un conjunto de objetos candidatos que responden a la consulta. Inicializa como vacío el conjunto de respuesta de Range Query. Y para cada uno de los objetos candidatos, si no pertenece al conjunto de respuesta de Range Query, entonces lo incluye al conjunto de respuesta final. Ver algoritmos [GSGH07].

CONSULTA POR INSTANTE DE TIEMPO (TIME SLICE QUERY)

Este tipo de consulta intenta determinar la existencia de objetos que intersequen el espacio consultado en el instante de tiempo dado.

El procesamiento de esta consulta es similar al de Window Query, pero a diferencia de ésta, el argumento recibido no es un intervalo de tiempo sino un instante de tiempo particular.

El espacio de consulta está asociado a un instante de tiempo dado. Éste contiene partes de poligonales, que atraviesan el mismo espacio en ese instante de tiempo.

La manera en que procede la consulta Time Slice Query es la siguiente: identifica las poligonales cuyas posiciones espaciales intersecan el área denominada ventana o sub-espacio en un instante de tiempo dado.

Para ello, realiza la búsqueda dentro de la estructura I+3 R-Tree todos los cubos que intersecan con la ventana de consulta. Así, recupera todas las poligonales que están en dicho instante de tiempo.

Luego, accede a los R-Tree's inferiores asociados a cada uno de los cubos que contienen poligonales recuperados anteriormente y recupera aquellos objetos cuyo tiempo de permanencia coincida con instante de tiempo correspondiente. En caso de no haber ocurrido cambios en las posiciones espaciales que conforman las poligonales se realiza la búsqueda dentro de los R-Tree's inferiores correspondientes. Ver algoritmos [GSGH07].

CONSULTA POR TRAYECTORIA (TRAJECTORY QUERY)

Este tipo de consulta consiste en recuperar las posiciones espaciales que un objeto fue atravesando en su trayectoria.

Debemos recordar que la recuperación de la trayectoria puede ser de distintos tipos: *parcial*, *completa* y por *intervalo de tiempo*. En la trayectoria parcial se informan las rutas o aristas que atravesó el objeto. En la trayectoria completa se informa todas las posiciones que atravesó el objeto desde el primer al último movimiento informado por el mismo. Y en la trayectoria por intervalo de tiempo recibe como argumento un intervalo temporal, e informa la trayectoria recorrida por ese objeto durante ese intervalo de tiempo.

La manera en que procede la consulta Trajectory Query es la siguiente: recibe el identificador del objeto cuya trayectoria se desea conocer. De acuerdo al tipo de trayectoria, se siguen los siguientes pasos.

En caso de tratarse de una trayectoria parcial, el algoritmo accede a los R-Tree's inferiores. Allí, la búsqueda se basa en el tiempo y en las posiciones en que el objeto informó su arribo. Con esta información, reconstruye la trayectoria, identificando aquellas aristas que forman parte de las poligonales por las cuales atravesó el objeto.

En caso de tratarse de una trayectoria completa, el algoritmo accede a los R-Tree's inferiores y recupera la siguiente información: identificador de la poligonal, identificador del objeto, intervalo de tiempo y posición.

Finalmente, en caso de tratarse de una trayectoria por intervalo de tiempo, el algoritmo realiza los mismos pasos que en la trayectoria parcial, con la restricción que la trayectoria del objeto se reconstruye considerando el intervalo de tiempo, es decir, obtiene las posiciones por las cuales transitó el objeto en movimiento en los instantes de tiempo pertenecientes al intervalo. Ver algoritmos [GSGH07].

4. CONCLUSIONES Y VISIÓN DE FUTURO

Hemos presentado una posible adaptación del índice I+MON-Tree en redes móviles, como así también, hemos planteado las posibles modificaciones que se deberían realizar sobre las consultas que soporta dicho índice.

Debemos recordar que, en este trabajo sólo damos una primera aproximación del diseño de adaptación del índice I+MON-Tree, cuando la red subyacente se trata de una red móvil. Esto implica que, no necesariamente este diseño es el

mejor modelo, sino que sólo es una presentación inicial, la cual puede ser modificada y/o reemplazada por otros diseños más adecuados.

Claramente mostramos que, no consideramos costos de almacenamiento y tiempo, entre otros. Por lo que creemos haber dejado un camino abierto para futuras investigaciones sobre el tema, referidos a la investigación y desarrollo su correspondiente refinamiento, implementación y evaluación experimental del buen desempeño de la estructura.

5. REFERENCIAS

[AG04] Almeida, V. y Güting, R. Indexing the trajectories of moving objects in networks. *Technical Report 309, Fernuniversität Hagen, Fachbereich Informatik*, 2004.

[AG04a] Almeida, V. y Güting, R. Indexing the trajectories of moving objects in networks. In Proc. of the 16th Intl. Conf. on Scientific and Statistical Database Management (SSDBM). 2004.

[BRINK00], Brinkhoff, T. Generating Network-Based Moving Objects. In Proc. of the 12th Intl. Conference on Scientific and Statistical Database Management Berlin, IEEE Computer Society Press. 2000.

[COG06]. Correa, L; Ortiz, N; Gagliardi, E.; I+Mon-Tree: Índice Espacio-Temporal para Objetos en Movimiento. CACIC 2006. Congreso Argentino de Ciencias de la Computación. 2006.

[CGGG06] Carrasco, Fernando D., Gagliardi, Edilma O., García Sosa, Juan C.; Gutierrez, G. Una propuesta de un método de acceso espacio-temporal: I+3 R-Tree CACIC 2006. Congreso Argentino de Ciencias de la Computación. 2006.

[GSGH07] Giraudi, D.; Segura Guzman, G.; Gagliardi, E.; Hernández Peñalver, G. Ruteo e Indexación Espacio-Temporal en redes móviles. Trabajo Final de la Licenciatura en Ciencias de la Computación, Universidad Nacional de San Luis, Argentina, 2007.

[Nst99] Nascimento, M., Silva, J. y Theodoridis, Y. Evaluation of Access Structures for Discretely Moving Points. In Proc. of the Intl. Workshop on Spatio-Temporal Database Management, STDBM, pages 171–188. 1999