



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 6

Título: Técnicas para indexar información espacio-temporal para el análisis de trayectorias de objetos móviles.

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS
INFORMÁTICAS

Autor: *Lenier Pérez Ahmed.*

Tutor: *M.SC. Yuniel Eliades Proenza Arias.*



“ Los estudiantes son en su mayoría revolucionarios. Revolucionarios por naturaleza, porque pertenecen a ese estrato de jóvenes que se abren a la vida y que adquieren todos los días conocimientos nuevos ”.

Ernesto Guevara de la Serna (El Che)

Dedico el presente trabajo:

A mis madres por estar siempre a mi lado apoyándome, por brindarme todo su cariño, su comprensión y porque gracias a ellos soy todo lo que soy.

A mi novia Yanisleydis por aparecer en mi vida justo cuando más la necesitaba, por su dedicación y cariño, por comprenderme y por ser muy especial para mí.

A mi hermana Lenia, a todos mis demás familiares y amigos. Y sobre todo a mí por todo el esfuerzo puesto en la realización de este trabajo que ha sido una gran meta en la carrera de mi vida.

Quiero darles las gracias a todas las personas que de una forma u otra contribuyeron en el desarrollo de este trabajo, como también quiero agradecerles a mi madre, mi padre, mi hermana, mi novia y demás familiares.

También quiero agradecerle a mi tutor por todo el apoyo y la paciencia que ha tenido conmigo, por orientarme y ganarse mi confianza. Por confiar en mí y no dudar nunca de mis capacidades.

A todas mis amistades en general.

Declaro que soy el único autor de este trabajo y autorizo a la Facultad 6 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Firma del Autor

Firma del Tutor

Síntesis del Tutor

M.Sc. Yuniel Eliades Proenza Arias

Graduado de Ingeniería Informática en la Universidad de Holguín en el año 2006. Máster en Ciencias en Ingeniería de Software e Inteligencia Artificial por la Universidad de Málaga en el año 2011. Se ha desempeñado como profesor en la Universidad de las Ciencias Informáticas, además de otras tareas asociadas a la producción. Tiene experiencia laboral en la línea del desarrollo de sistemas y la arquitectura de software.

Correo: yproenza@uci.cu

Resumen

Con el rápido avance tecnológico en computación móvil, realizar el seguimiento de los cambios de la posición o formas de los objetos se hace cada vez más necesario. Actualmente, un gran número de aplicaciones utilizan Bases de Datos espacio-temporales, que representan información de objetos en movimiento. Un ejemplo de ello son las aplicaciones para el control de flotas, las cuales se encargan de supervisar la trayectoria de objetos, tales como autos, trenes, aviones y barcos. Estos son algunos de los objetos con los cuales deben tratar estos Sistemas de Información Geográfica (SIG), los cuales se dedican al estudio de la información geoespacial recogida.

Una de las características más destacables de los SIG es su capacidad para gestionar enormes cantidades de información. Esto hace que uno de los temas de investigación más importantes en el área sea el estudio de estructuras de indexación que permitan un acceso eficiente a la información. Es por ello que el principal objetivo de esta investigación es analizar el trabajo realizado en el área, partiendo del análisis de las propuestas existentes para la indexación de objetos móviles, así como identificar líneas abiertas para trabajo futuro.

Palabras Claves: consultas espacio-temporales, índices, objetos en movimiento.

Índice

Introducción1

Capítulo 1: Fundamentación Teórica5

 1.1 Introducción5

 1.2 Conceptos asociados al dominio del problema5

Índice5

Trayectoria5

Trayectoria Espacio-Temporal.....6

Datos Espaciales.....6

Flota7

Objetos móviles.....7

Control de Flota.....7

Sistema de Información Geográfica.....8

 1.3 Trabajo Relacionado.....8

 1.4 Técnicas de indexación sobre información de entidades móviles9

Búsqueda en un R-Tree13

Inserción en un R-Tree.....14

Eliminación en un R-Tree15

Conclusiones Parciales15

Capítulo 2: Métodos de Acceso y Procesamiento de Consultas Espacio-Temporales16

 1.1 Introducción16

 1.2 Marco de Referencia para el análisis de las técnicas.....16

 1.3 Análisis de la trayectoria de los objetos en movimiento17

 1.3.1 Análisis del comportamiento histórico.....22

1.3.2	Recuperar posiciones futuras de objetos en movimiento.....	30
	Conclusiones Parciales	31
Capítulo 3: Presentación de la Solución Propuesta.....		32
1.4	Introducción.....	32
1.5	Criterios de Comparación	32
1.5.1	Comparación entre el SEST-Index y el HR-Tree	35
1.5.2	Comparación entre el SEST-Index y el SESTL	40
1.5.3	Comparación entre el STR-Tree y el TB-Tree	43
1.5.4	Solución Propuesta	47
	Conclusiones Parciales	48
Conclusiones Generales.....		50
Recomendaciones		52
Referencias Bibliográficas.....		53
Bibliografía.....		55
Glosario de Términos.....		57

Índice de Figuras

Figura 1: Agrupaciones de MBRs generadas por un R-tree. 13

Figura 2: El R-tree de los rectángulos de la Figura 1. 13

Figura 3: Distintas posibilidades de división de un nodo. (a) Mala división, (b) Buena división..... 14

Figura 4: Diferentes escenarios de partición. 19

Figura 5: Diferentes etapas de la inserción. 21

Figura 6: Subárboles para varios objetos en dos instantes de tiempo. 23

Figura 7: Indexación en el HR-Tree. 23

Figura 8 : Objetos moviéndose sobre un mapa particionados 28

Figura 9 : Partición uniforme del espacio representado por celdas 29

Figura 10 : Almacenamiento utilizado por el SEST-Index y el HR-Tree..... 37

Figura 11: Número de bloques accesados por una consulta de tipo instante de tiempo utilizando el SEST-Index (rango espacial formado por el q% de la longitud de cada dimensión). 38

Figura 12 Número de bloques accesados por una consulta de tipo intervalo de tiempo utilizando el SEST-Index..... 39

Figura 13 : Espacio utilizado por los métodos de indexación TB-Tree y STR-Tree. 44

Figura 14 : Número de bloques accesados por una consulta de tipo intervalo de tiempo utilizando TB-Tree y STR-Tree. 45

Figura 15 : Número de bloques accesados por una consulta de tipo instante de tiempo utilizando el, TB-Tree y STR-Tree..... 46

Índice de Tablas

Tabla 1: Características más destacadas del método de acceso SEST-Index.....33

Tabla 2: Características más destacadas del método de acceso HR-Tree33

Tabla 3 : Características más destacadas del método de acceso SESTL.....34

Tabla 4: Características más destacadas del método de acceso TB-Tree.....35

Tabla 5 Características más destacadas del método de acceso STR-Tree.....35

Tabla 6 : Porcentajes de ahorro de almacenamiento del SEST-Index conseguido por medio de la estrategia de reutilización del espacio del HR-tree.....37

Tabla 7 : Resultados destacados de la evaluación del SEST-Index y el HR-Tree.....40

Tabla 8: Resultados destacados de la evaluación del SEST-Index y el SESTL.43

Tabla 9: Resultados destacados de la evaluación del STR-Tree y TB-Tree.....47

Índice de Algoritmos

Algoritmo 1: Algoritmo para insertar en el STR-tree.....18
Algoritmo 2: Algoritmo para insertar en el TB-Tree.21

Introducción

Los Sistemas de Gestión de Bases de Datos son por excelencia la tecnología informática que ofrece las herramientas adecuadas para manejar grandes volúmenes de datos interrelacionados. Inicialmente esta tecnología sólo permitía representar tipos de datos básicos o primitivos, tales como: enteros o cadenas de caracteres. Los sistemas existentes tienen la capacidad de poder representar solamente la información más reciente de los objetos, de esta forma, si se cambia la información de un objeto, se pierde la información histórica. Las Bases de Datos Espacio-Temporales surgen para representar la evolución de las entidades con el tiempo o el cambio de su información. En este tipo de Bases de Datos no sólo es posible modelar el estado reciente de los objetos, sino también su comportamiento histórico.

En las Bases de Datos espacio-temporales se consideran dos dimensiones de tiempo: *tiempo válido*, que corresponde al tiempo en el cual un hecho se considera verdadero para la realidad modelada y *tiempo de transacción*, que hace referencia al tiempo en el cual el hecho se registró en la Base de Datos (BD). Existen de esta manera aplicaciones que consideran mantener en una misma relación las dos dimensiones de tiempo, este tipo de relación es denominada *relación bitemporal*. Un objeto que viaja equipado con un Sistema de Posicionamiento Global (GPS), por sus siglas en inglés que envía su localización actual hacia una BD central con un retardo mínimo, el tiempo válido y el tiempo de transacción son similares, por lo tanto la Base de Datos se puede considerar de tiempo válido, de tiempo de transacción y bitemporal a la vez.

Actualmente están apareciendo muchas aplicaciones del dominio espacio-temporal en el ámbito del transporte. Ejemplo de ello son los sistemas de supervisión de tráfico, que monitorean las posiciones de los vehículos y patrones de desplazamiento, con el propósito de poder predecir en un futuro congestiones de tráfico, rutas óptimas, límites de velocidad y predicciones de movimientos futuros, todo ello con el objetivo de evitar posibles accidentes. Se hace necesaria la utilización de técnicas que permitan tener un acceso rápido a los datos y la preparación de los mismos para que puedan ser procesadas consultas que tributen a resolver los problemas existentes en esta esfera del transporte.

En Cuba, durante la última década se ha trabajado en función de desarrollar aplicaciones que recojan estos datos para mejorar algunas áreas de la sociedad, por ejemplo el sector del transporte. La Universidad de las Ciencias Informáticas (UCI) se destaca como principal impulsora del desarrollo de esta

generación o tipología de aplicaciones, debido a que es un centro con un gran potencial en el desarrollo de soluciones informáticas, donde se desarrollan software que apoyan en gran medida al desarrollo de la economía del país. La facultad número 6 cuenta con el Centro de Desarrollo de Geoinformática y Señales Digitales (GEySED), el cual posee como misión desarrollar productos y soluciones informáticas en el campo del procesamiento de Señales Digitales y la Geoinformática. En el departamento de Geoinformática se encuentran varios proyectos entre los cuales se puede apreciar el proyecto Control de Flotas. Este proyecto posee como misión el desarrollo de Sistemas de Información Geográfica, basados en tecnologías de software para el control de flotas. Dentro de las líneas del proyecto se encuentran, el análisis de trayectorias y flujos GPS y algoritmos inteligentes de razonamiento sobre datos espacio-temporales.

El proyecto presenta la necesidad de mecanismos para acelerar el acceso a los datos recogidos por los GPS y la preparación de estos para la realización de consultas para el análisis de las trayectorias. Se ha hecho un estudio de los mecanismos que permiten mejorar con rapidez las búsquedas de datos dentro de grandes almacenes de datos. En este campo, para acceder eficientemente a la información, se emplean técnicas para indexar información espacio-temporal de objetos móviles. Se hace necesario el estudio y profundización sobre estas técnicas, así como establecer una relación entre los puntos móviles que estas procesan, con las flotas que se deben estar supervisando. Todo lo anteriormente expresado posibilitará el desarrollo del Sistema de Información Geográfica para la supervisión y control de las flotas.

Cuando se analiza la estructura de la información asociada a flotas se hace evidente que presenta un carácter espacio-temporal. Dentro de la Inteligencia Artificial existe un área poco explorada asociada al tratamiento de los denominados objetos móviles (puntos, líneas y regiones móviles). Analizando características afines, se puede hacer una analogía entre lo que se denomina flota y punto móvil. En la literatura, se ha denominado flotas a los aviones pertenecientes a una aerolínea, empresas de transportación (ómnibus, taxis, trenes) y la red de personas interconectadas con sus teléfonos móviles a diferentes servicios. Existen numerosos trabajos enmarcados en la indexación de puntos móviles, en los cuales no se ha establecido una relación de si estas técnicas podrán ser utilizadas en el control de flotas. Debido a la presente denominación de flotas es necesario el estudio de las técnicas de indexación de puntos móviles con el objetivo de poder darle respuesta a la problemática existente. El presente trabajo

tiene el objetivo de poder decantar de las técnicas existentes de indexación de puntos móviles, cuáles podrán darle, con eficiencia, soporte a las especificaciones del proyecto.

Surge entonces la siguiente interrogante: ¿Cómo representar los datos de objetos en movimiento de forma que se pueda tener un rápido acceso a la información y la estructura adecuada para darle respuesta a consultas espacio temporales para el control de flotas?

El **objetivo general** de este trabajo es proponer las técnicas de indexación para el análisis de información sobre trayectoria de objetos móviles para el control de flotas.

El **objeto de estudio** son las técnicas de indexación sobre información de entidades móviles.

El **campo de acción** que enmarca este trabajo son las técnicas de indexación sobre información de puntos móviles para un control de flotas.

La **idea a defender** es la siguiente: Con la selección de un conjunto de técnicas para la indexación de información sobre trayectorias de objetos móviles para el control de flotas, se permitirá analizar de manera rápida y eficiente grandes almacenes de datos así como la realización de diferentes tipos de análisis sobre la información como estudio de la trayectoria y detección de patrones de movimiento.

Los **Métodos teóricos** utilizados en el desarrollo de este trabajo son:

- ✚ El Análisis y Síntesis: Para conocer, reflexionar y aumentar los conocimientos acerca de las técnicas existentes de indexación de información espacio-temporal para el análisis de trayectorias de objetos móviles, basándose en la bibliografía consultada y luego, la síntesis para arribar a conclusiones de la investigación.
- ✚ La Inducción y Deducción: Con el objetivo de reflejar eventos en común del objeto de estudio a través de la profundización de conocimientos generales y de esta forma arribar a conclusiones basadas en la información consultada.
- ✚ El Histórico-Lógico: Este método se utiliza a la hora de realizar un estudio de antecedentes que existen respecto a los modelos realizados hasta el momento del análisis de técnicas para indexar información espacio-temporal para el análisis de trayectorias de objetos móviles.

Los **Métodos Empíricos** utilizados en el desarrollo de este trabajo son:

- ✚ El Experimento: Para realizar pruebas que permitan poseer más argumentos que tributen a la clasificación de las técnicas más idóneas para el control de flotas.

Este trabajo está estructurado de la siguiente forma:

- ✚ Luego de esta introducción, el Capítulo 1 aborda todo lo concerniente al objeto de estudio, así como los términos utilizados y los antecedentes que dieron paso a la presente investigación donde se hace el análisis de la documentación existente asociada a las técnicas de indexación sobre información de entidades móviles para un control de flotas.
- ✚ El Capítulo 2 se centra en describir los métodos de acceso espacio temporal desde el punto de vista de los datos y las consultas que procesan, para una posterior caracterización y comparación determinando los más eficientes para resolver los problemas existentes en el proyecto Control de Flota.
- ✚ En el Capítulo 3 finalmente se definen cuáles de las propuestas existentes, son las más idóneas para cumplir con las necesidades actuales del proyecto.

Capítulo 1: Fundamentación Teórica

1.1 Introducción

Este capítulo contiene las definiciones y conceptos que serán de utilidad para el entendimiento de la investigación. Se da una visión de cuál ha sido el enfoque que se le ha dado a los trabajos relacionados a la panorámica espacio-temporal. También se presenta la estructura de indexación R-Tree, con el objetivo de que sirva de base a la hora de poder comprender las restantes técnicas de indexación.

1.2 Conceptos asociados al dominio del problema

Sobre la base de que el lector pueda tener un mayor entendimiento y comprensión de los temas que serán tratados durante la investigación y que están directamente relacionados con el objeto de estudio se describen a continuación un grupo de conceptos asociados al dominio del problema.

Índice

Según *La Real Academia* en (Aguayo González), un índice es el identificador usado para acceder a la información almacenada, o una estructuración especial interna de los datos almacenados en tablas de una base de datos que aceleran la búsqueda de los registros seleccionados.

Trayectoria

La definición de movimiento según (Gámez Sosa, 2009) se refiere a la noción de cambiar de posición física de una entidad con respecto a un sistema de referencia, en el cual cada uno asume posiciones conocidas. Frecuentemente el sistema referenciado es un espacio geográfico. Una trayectoria es el trayecto o camino creado por la entidad en movimiento a través del espacio donde se desplaza.

El trayecto nunca se crea instantáneamente, dado que requiere una cierta cantidad de tiempo, por lo tanto, el tiempo es un aspecto inseparable de las trayectorias, ejemplo si se tiene la posición inicial y final, que es el momento en que termina, para cualquier momento ti entre la posición inicial y final hay una posición en el espacio que está siendo ocupada por una entidad en ese momento. Por lo tanto una trayectoria puede ser vista como una función que empareja momentos de tiempo en el espacio; o también puede ser visto como una consistencia de pares (*tiempo y ubicación*).

Trayectoria Espacio-Temporal

En (Spaccapietra, y otros, 2007) se define a una trayectoria espacio-temporal como la posición cambiante de un objeto en el espacio geográfico, sea el espacio en 3D (por ejemplo la trayectoria de un avión), o en 2D (la trayectoria descrita por una pelota en un juego, la trayectoria descrita por una bala). Mayormente estos objetos en movimiento son expresados geoméricamente como un punto, ejemplo de ellos son una persona, un animal, un automóvil, un camión, un avión, una nave, un tren.

En (Spaccapietra, y otros, 2007) se destaca que una trayectoria tiene dos facetas:

La faceta geométrica

Este es el registro espacio temporal de la posición del punto en movimiento. Es un segmento delimitado por la posición del objeto durante toda la vida útil del objeto, siendo representada esta faceta geométrica como la función $[t_{inicial}, t_{final}]$.

La faceta semántica

Esta es la información que se transmite sobre la orientación de la trayectoria y sus características correspondientes.

Datos Espaciales

Según (Guillaumet, y otros, 2000) un dato espacial es una variable asociada a una localización del espacio, las formas de un objeto geográfico y las relaciones entre objetos geográficos comúnmente con coordenadas y topología.

Los datos espaciales son el componente fundamental de cada proyecto o aplicación SIG. Presentan la información en representaciones subjetivas a través de mapas y símbolos, que representan la geografía como formas geométricas, a las cuales se les asignan sus respectivos atributos que las definen y describen.

Los datos espaciales cumplen con los siguientes principios básicos.

-  Tienen posición absoluta: sobre un sistema de coordenadas (x, y, z) .
-  Tienen una posición relativa: frente a otros elementos del paisaje (topología: incluido, adyacente, cruzado, etc.).

- ✚ Tienen una figura geométrica que los representa (punto, línea, polígono).
- ✚ Tienen atributos que lo describen (características del elemento o fenómeno).

Flota

Una flota es un grupo de vehículos, que pueden ser terrestres, marítimos o aéreos. Cuando los vehículos comparten las mismas características se dice que la flota es homogénea y si son diferentes son clasificados como flota heterogénea.

Objetos móviles

Existen tres tipos de objetos móviles: *el punto móvil*, utilizado para representar la información cuando se rastrean celulares, aviones, vehículos o personas con un alto nivel de abstracción; *la región móvil*, para representar huracanes y su movimiento, lagos, la dispersión de una epidemia, que pueden cambiar la extensión y un caso poco común, *la línea móvil*, para representar entidades que ocupan una extensión lineal, que puede aumentar o disminuir o desplazarse con el tiempo, como por ejemplo una flota de barcos o un convoy. En general un objeto móvil no es más que la abstracción de una entidad que va a poseer características espacio-temporales.

También se clasifican los objetos móviles teniendo en cuenta “*las restricciones en el movimiento de los objetos*”, las cuales deben tenerse en cuenta al realizar cualquier análisis. Se tienen:

- ✚ Objetos con movimiento en ambientes no restringidos.
- ✚ Objetos con movimiento en ambientes restringidos.
- ✚ Objetos con movimiento en redes.

Control de Flota

El control de flota está dirigido básicamente al seguimiento de vehículos asociados a una organización determinada. Es un mecanismo implementado para detectar cualquier desvío de los patrones de movimiento establecidos previamente, haciendo posible la debida regulación de las entidades con las cuales tratan.

El control de flota es posible realizarlo obteniendo datos como la ubicación de los vehículos, velocidad y trayectoria. Los datos pueden ser recogidos a través de métodos de transmisión que incluyen tanto los

sistemas terrestres como las vías satelitales. Estas últimas garantizan el seguimiento de la flota en entornos remotos. Después de la captura, estos datos pasan a ser examinados, donde se analiza si existe alguna incongruencia entre los objetivos planificados y las actividades cumplidas; por ejemplo entre las rutas concebidas para un determinado vehículo y las trayectorias seguidas por este en la realidad.

Sistema de Información Geográfica

Los SIG forman parte del ámbito más extenso de los Sistemas de Información, siendo capaces de integrar, almacenar, editar, analizar, compartir y mostrar la información geográficamente referenciada. Una definición más completa y aceptada podría estar dirigida a pensar en un SIG como un sistema de hardware, software y un conjunto de procedimientos diseñados con el objetivo de facilitar la obtención, manipulación, gestión, modelado, análisis, representación y salida de información georeferenciada, para resolver problemas complejos de planificación y gestión y contribuir a la toma de decisiones, dando además una perspectiva totalmente nueva y dinámica de la información. (Miñano, 2000)

1.3 Trabajo Relacionado

Los SIG han ido ganando terreno en numerosas áreas de la sociedad. La idea de meros sistemas para estudiar información geoespacial, localización geográfica y características del terreno se ha ido sustituyendo por apoyo a la toma de decisiones, previsión de desastres, seguimiento y predicción del movimiento de huracanes y control de movimiento de vehículos. De este último proceso se encargan las aplicaciones que se denominan: aplicaciones para el control de flotas. Todo esto ha estado estrechamente relacionado con el creciente desarrollo y uso de la tecnología Sistema de Posicionamiento Global, GPS¹ por sus siglas en inglés.

La idea de lo que se denomina flota se ha ido generalizando con el transcurso de los años, en sus inicios estuvo asociada a los vehículos con los que contaba determinada empresa para la transportación de personal o mercancías. Hoy se puede catalogar como flotas a:

- ✚ Los aviones pertenecientes a una aerolínea comercial o militar.
- ✚ Una empresa de transportación (ómnibus, taxis, trenes).
- ✚ La red de personas interconectadas con sus teléfonos móviles a diferentes servicios.

¹ GPS acrónimo de Sistema de Posicionamiento Global por sus siglas en inglés

Los SIG generalmente están enfocados en recoger la evolución o la alteración sufrida por determinados objetos. En este caso se contemplan conceptos relacionados con objetos en movimiento, permitiendo consultas de los datos espacio-temporales, capturando esa evolución con respecto al tiempo tanto de la posición como de la extensión. Fue sólo en los 90's que las bases de datos espacio-temporales se volvieron un área de investigación activa.

Los trabajos relacionados con las BD espacio-temporales se han centrado en modelar el tiempo como un atributo asociado con objetos espaciales, ejemplo visible es el cambio del valor de los atributos de un objeto o cambios en la geometría o topología. Estos SIG representan el cambio en los objetos con respecto al tiempo como una colección de objetos temporalmente indexados o como una colección de snapshots². Estos representan imágenes de un estado del mundo en instantes de tiempos particulares.

1.4 Técnicas de indexación sobre información de entidades móviles

El desarrollo de estructuras de indexación que permitan recuperar objetos espaciales de manera eficiente ha sido un tema de interés en las últimas décadas. El objetivo principal ha sido mejorar la eficiencia a la hora de recuperar los datos que se ajustan a una consulta de búsqueda determinada.

“Actualmente los índices más extendidos en las bases de datos espaciales existentes son el R-Tree y en segundo lugar los quadtree o grid multinivel” (Manso). Sin embargo estos métodos no son los más robustos para representar modelos del mundo real con todo el dinamismo que implican y administrar objetos que cambian su ubicación y/o forma a través del tiempo.

Se han desarrollado diferentes propuestas para soportar simultáneamente tiempo y espacio. Estas apuntan a indexar objetos que se mueven en un espacio. La mayoría de estas técnicas extienden los algoritmos de acceso espacial para incluir componentes temporales. Algunos métodos se enfocan en la recuperación de datos históricos, un ejemplo de ello es la estructura HR-tree. De igual manera otros están enfocados en la trayectoria de los objetos en movimiento, tales como TB-tree y STR-tree. Finalmente, algunos son usados para recuperar posiciones futuras de objetos en movimiento, basadas en la posición actual y patrones de movimiento, ejemplo el TPR-tree.

² Snapshots corresponde a un estado de la base de datos en un instante determinado

Según (Gutiérrez Retamal, 2007) existen varios métodos de acceso espacio-temporal que son adecuados para aplicaciones que consideran cambios espaciales de manera discreta. Una clasificación de ellos es la siguiente:

- ✚ Métodos que tratan el tiempo como otra dimensión.
- ✚ Métodos que incorporan la información temporal dentro de los nodos de la estructura sin considerar el tiempo como otra dimensión.
- ✚ Métodos basados en versiones de la estructura.
- ✚ Métodos orientados a la trayectoria.

Al momento de utilizar un índice es importante determinar qué tipo de consulta se tendrá en cuenta, ya que esto establece la forma en que se administra y almacena la información, así como también la manera de recuperar la información necesaria. El mayor objetivo de los sistemas de administración de BD espacio-temporales es la eficiencia para manipular consultas de la forma “buscar todos los objetos que están ocultos dentro de un área determinada (o en algún punto específico), durante un intervalo de tiempo específico (o en un instante de tiempo concreto) “, se espera que sean los más comunes. Este tipo de consulta define una región en el espacio y obtiene como resultado todos los objetos geográficos indexados que se solapan con dicha región.

“Las consultas espacio-temporales utilizadas por las BD que han recibido una mayor atención, son las consultas de tipo instante de tiempo e intervalo de tiempo” (Gutiérrez Retamal, 2007). Una consulta de tipo instante de tiempo recupera todos los objetos que interceptan un rango espacial en un instante de tiempo específico. Una consulta de tipo intervalo de tiempo extiende la misma idea, pero a un intervalo de tiempo. Aunque estas son las que han recibido una mayor atención, existen otras consultas que resultan ser las más típicas, ejemplo de ellas son:

- ✚ Consulta sobre **eventos**, en la cual se recuperan eventos que sucedieron en una región en un instante dado. Estos eventos pueden ser objetos que han aparecido o desaparecido en una región en un cierto instante.
- ✚ La consulta **trayectoria**, es la que recupera el conjunto de posiciones espaciales en las que un objeto ha permanecido en un intervalo de tiempo dado.

Un aspecto importante a revisar es el escenario donde se mueven los objetos espacio-temporales en cuestión. Para los distintos tipos de indexación existentes se han tenido en cuenta los siguientes escenarios:

- ✚ **Movimiento sin restricciones** es aquel en donde los objetos se mueven libremente dentro de la región en la que se encuentran, un ejemplo en este contexto es el monitoreo de animales. Dentro de los movimientos sin restricciones, existen distintos métodos de indexación que se diferencian de acuerdo a la información que indexan: métodos que indexan el pasado; métodos que indexan el presente y métodos que indexan el presente y futuro.
- ✚ Un segundo escenario es el **movimiento restringido** que es aquel en que el escenario se presenta limitado para los objetos que se mueven. Esta limitación tiene que ver con obstáculos (infraestructuras) que obligan al objeto a tomar caminos alternativos, ya que éstos no pueden ser evitados, un ejemplo de este contexto es el de un robot autónomo al cual se le indica qué hacer, pero no cómo hacerlo, es decir, planifica sus movimientos en base al conocimiento que tiene de un determinado ambiente y sus obstáculos.
- ✚ Y como último, el **movimiento restringido a redes**, donde se modelan los movimientos basándose en una red más que en un espacio bidimensional. Resulta interesante pensar que un movimiento restringido a una red, es aquel donde no interesa el espacio subyacente a ella para moverse, es decir, al ver una carretera y un automóvil moviéndose en ella, no interesa el espacio o el tamaño, tampoco la posición del objeto en términos de coordenadas x e y , sino que interesa la posición relativa respecto de la red, es decir, en el kilómetro 24 de la ruta 3, por ejemplo a menudo, una red dirige el tráfico de los objetos que se mueven en ella. Es decir, al tener una red predefinida, los objetos que se mueven en ella tienden a moverse siguiendo los diferentes caminos que la misma presenta. La red puede ser tanto artificial, creada basándose en un criterio para el movimiento de los objetos, como real, basada en una del mundo real.

Los métodos existentes de indexación utilizan “con algunas modificaciones” el método de indexación R-tree. Se debe realizar un estudio de esta estructura para tener una mejor comprensión de los métodos expuestos a continuación. El R-tree, propuesto en (Antonin, 1984), es uno de los métodos de acceso

multidimensional o espacial y que ha sido adoptado por productos de Sistemas de Bases de Datos, tales como Oracle y PostgreSQL. Un R-tree es una estructura que permite indexar objetos espaciales, puntos y regiones. Cada nodo corresponde a una página o bloque de disco. Los nodos hojas de un R-tree contienen entradas de la forma $\langle MBR, oid \rangle$ donde oid es el identificador del objeto espacial en la base de datos y MBR^3 (Minimum Bounding Rectangle) es un rectángulo multidimensional que corresponde al mínimo rectángulo que encierra al objeto espacial. Los nodos internos (nodos no-hojas) contienen entradas de la forma $\langle MBR, ref \rangle$, donde ref es la dirección del correspondiente nodo hijo en el R-tree y MBR es el rectángulo mínimo que contiene a todos los rectángulos definidos en las entradas del nodo hijo. La cantidad máxima de entradas en un nodo está dada por el parámetro M y $m \leq M/2$ el cual contiene el número mínimo de entradas a un nodo.

Según la definición propuesta por el autor en (Antonin, 1984), el R-tree cumple con las siguientes características:

- ✚ Los nodos hojas contienen entre m y M entradas a menos que sea la raíz.
- ✚ Cada entrada en un nodo hoja es de la forma $(I, Identificador)$, I es el rectángulo más pequeño que espacialmente contiene al objeto n-dimensional referenciado por el identificador como se muestra en la **Figura: 1**.
- ✚ Cada nodo rama tiene entre m y M hijos a menos que sea la raíz.
- ✚ Cada entrada en un nodo rama (nodos no hojas) es de la forma $(I, punteroAlHijo)$, I es el rectángulo más pequeño que espacialmente contiene los rectángulos en el nodo hijo.
- ✚ La raíz tiene como mínimo dos hijos a no ser que sea hoja.
- ✚ Todos los nodos aparecen en el mismo nivel.

La **Figura: 1** muestra las particiones del espacio a diferentes niveles, por ejemplo los MBR R1 y R2 forman una partición en un primer nivel, a su vez los MBR R6 y R7 conforman una partición del sub espacio R2 en un segundo nivel y así sucesivamente.

³ MBR acrónimo de mínimo rectángulo delimitador por sus siglas en inglés

verificando si el MBR de la correspondiente entrada del nodo y Q^4 se intersectan. Producto de que las particiones en los diferentes niveles no son disjuntas, puede ser necesario recorrer varios caminos desde un nodo interno a las hojas en una consulta. Por ejemplo, en el primer nivel Q solamente se intersecta con el rectángulo $R1$ (**ver Figura: 1**). Sin embargo, en el segundo nivel se intersecta con $R3$ y $R4$ lo que implica que los punteros asociadas a las particiones $R3$ y $R4$ se deben explorar hasta llegar a las hojas. Cuando se alcanza un nodo hoja, el algoritmo verifica si el MBR del objeto se intersecta con Q y de intersectarse el objeto pasa a ser un candidato del resultado de la consulta.

Inserción en un R-Tree

La estructura R-tree inserta sus elementos en las hojas, produciendo eventualmente de esta forma una división del nodo hoja por tener un máximo de elementos y propagando un elemento hacia el padre, el que a su vez también se puede dividir y propagar a su correspondiente padre. El proceso puede continuar recursivamente hasta llegar a la raíz, la cual también se puede dividir y generar una nueva.

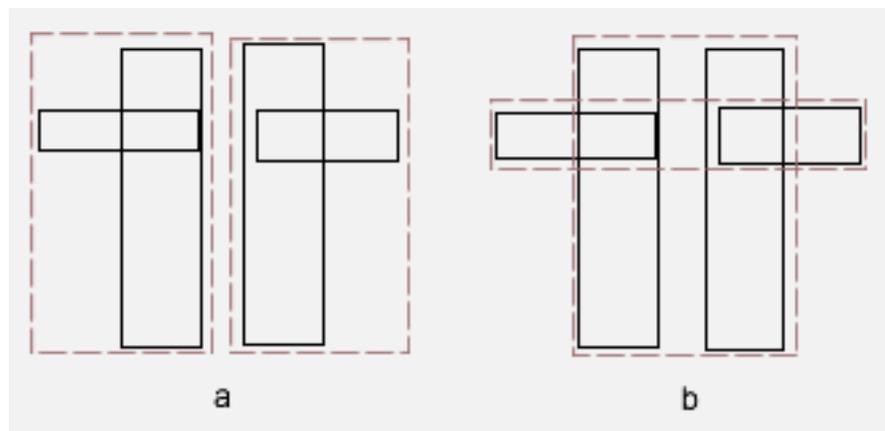


Figura 3: Distintas posibilidades de división de un nodo. (a) Mala división, (b) Buena división.

Cuando se agrega una nueva entrada en un nodo que se encuentra lleno, es decir, contiene M entradas, es necesario repartir las $M + 1$ entradas entre dos nodos. La división debería realizarse de tal forma que la posibilidad de acceder ambos nodos en búsquedas futuras sea lo más pequeña posible. Ya que la decisión de visitar un nodo depende de si el área cubierta por su rectángulo se sobrepone con el área de

⁴ Q consulta especificada en un espacio particionado

búsqueda, el área total cubierta por los dos rectángulos después de dividir un nodo debería ser minimizada. En la **Figura: 3 (a)** se considera una mala división pues el área cubierta por los rectángulos es mucho mayor que en aquella considerada una buena división (**Figura: 3 (b)**).

Eliminación en un R-Tree

El proceso de eliminación de una entrada E desde un R-tree comienza buscando el nodo hoja en el cual se encuentra dicha entrada E. Luego se elimina del nodo hoja y si dicho nodo queda con menos entradas que las permitidas, estas se reubican. La eliminación se propaga hacia la raíz del árbol, si es necesario ajustando las áreas cubiertas por los rectángulos a partir de la raíz hacia la hoja.

Conclusiones Parciales

En el presente capítulo se han expuesto diversos conceptos y elementos teóricos que sustentan el problema de la investigación a tratar, asociados a las disímiles técnicas de indexación de información espacio-temporal para el uso en un control de flotas. Se describió el objeto de estudio en general y todo lo asociado al dominio del problema donde se describe una de las principales técnicas a tener en cuenta, la estructura R-tree propuesta en (Antonin, 1984), la que posibilita indexar de manera eficiente la información espacial contenida en las BD espaciales, siendo de vital importancia el análisis, facilitando algunos argumentos que permitan comprender mejor los términos a manejar en la investigación. La necesidad de poder tener un equilibrio entre la información espacial junto a la temporal surge luego de la implementación de R-tree, ya que no solo se deseaba modelar los cambios ocurridos en los objetos sino también sus cambios en el tiempo. De esta forma se espera dar una visión de los aspectos teóricos necesarios manejados en este trabajo y asegurar los elementos para el posterior desarrollo de la investigación.

Capítulo 2: Métodos de Acceso y Procesamiento de Consultas Espacio-Temporales

1.1 Introducción

Un Sistema de Gestión de BD espacio-temporal debe proveer, entre otros servicios, métodos de acceso que permitan procesar de forma eficiente consultas cuyos predicados contemplan relaciones espacio-temporales sobre los objetos. En esta sección se describen los principales métodos de acceso espacio-temporales estudiados hasta ahora, considerando principalmente las consultas que estos pueden procesar sobre la información así como la estructura de cada uno. Los cuales pueden ser catalogados como:

- ✚ Métodos que indexan la trayectoria de objetos en movimiento.
- ✚ Métodos que Indexan los datos históricos de objetos en movimiento.
- ✚ Métodos que indexan los datos de objetos en movimiento, para predecir posibles cambios en su trayectoria.

La indexación se refiere a registrar ordenadamente la información elaborando índices de los datos espacio-temporales. Esta es un área bastante extensa a tratar, debido que los SIG necesitan de índices para poder obtener resultados más rápidos por la gran cantidad de datos que registran.

1.2 Marco de Referencia para el análisis de las técnicas

Las técnicas de acceso espacio-temporal deben proveer estructuras robustas a fin de poder darle respuestas a las exigencias planteadas, como tener un rápido acceso a los datos y la estructura adecuada para darle respuesta a las consultas especializadas. En el siguiente epígrafe se estará contando con los aspectos que se consideran más significativos a la hora de analizar las técnicas. Se estarán tratando en específico de las tres categorías antes mencionadas, solamente: “indexación de trayectorias de objetos en movimiento” y “indexación a través de datos históricos de objetos en movimiento”.

De cada una de las técnicas se describe brevemente su estructura de datos así como las consultas que estas pueden procesar. Se estarán contando con terminologías a la hora de ver las consultas como:

- ✚ Consultas de tipo **instante de tiempo** (time-slice), no son más que aquellas que se encargan de consultar solo los objetos que estuvieron en una zona para un instante de tiempo específico.
- ✚ Consultas de tipo **intervalo de tiempo** (time-interval), siendo casi muy parecida a la de instante de tiempo pero esta vez para un intervalo de tiempo.
- ✚ A su vez las consultas sobre **eventos**, que permiten recuperar los hechos espaciales que en un determinado momento han ocurrido en un área fijada, por ejemplo, “cuántos autos ingresaron al área del centro de la ciudad de Habana a las 12 horas de ayer”.

1.3 Análisis de la trayectoria de los objetos en movimiento

Estos métodos de acceso se basan en la trayectoria que siguen los objetos. Tales métodos de acceso tienden a privilegiar las consultas sobre las trayectorias de los objetos más que las consultas de tipo instante de tiempo o intervalo de tiempo.

STR-Tree

El STR-tree propuesto en (Pfooser, y otros, 2000), es una extensión de R-tree para soportar el procesamiento de trayectorias de puntos móviles. A pesar de ser una extensión del R-tree este posee una estrategia diferente a la hora de insertar y particionar los nodos.

Estructura de datos

Los nodos hojas tienen la siguiente estructura id, tyi, MBR, o donde tyi es un identificador de la trayectoria y o es la orientación de la trayectoria dentro del MBR . La idea fundamental del STR-tree es tratar de almacenar juntos los segmentos pertenecientes a la trayectoria de un mismo objeto, manteniendo en lo posible la cercanía espacial de tal manera de no deteriorar el rendimiento del R-tree (Pfooser, y otros, 2000).

Inserción

El algoritmo de inserción no selecciona el MBR donde insertar un objeto de acuerdo a un criterio de minimizar el crecimiento del área definido en el R-tree, sino que la elección se privilegia con los segmentos que corresponden a la trayectoria de un mismo objeto almacenando estos lo más cerca posible. De esta forma, cuando se inserta un nuevo segmento de línea, el objetivo es hacerlo lo más cerca

posible del segmento predecesor en la trayectoria del objeto. El algoritmo primero busca el nodo hoja que contiene el segmento predecesor en la trayectoria del objeto. De esta forma se pueden dar dos casos: que el nodo se encuentre lleno o no.

- ✚ Si el nodo contiene suficiente espacio, entonces el segmento se inserta, se ajustan los *MBR* de los nodos anteriores y el algoritmo de inserción termina.
- ✚ Si el nodo hoja seleccionado está lleno, se hace necesario hacer una partición del mismo. El algoritmo de partición debe considerar varios escenarios entre los segmentos del nodo hoja al momento de particionar. En el **Algoritmo: 1** se describe tanto la inserción como este nuevo algoritmo FindNode que propone STR-Tree.

```

Algoritmo Insert (N,E)
INS1 Invoca FindNode(N,E)
INS2 IF nodo N' es encontrado,
    IF N' hay espacio,
        insertamos E
    ELSE
        IF El p-1 nodo padre esta lleno,
            Invocamos ChooseLeaf(N'',E) dentro de un árbol
            indicado por N'' que excluye la rama actual,
        ELSE invocamos Split (N').
    ELSE ChooseLeaf(N,E).

Algoritmo FindNode(N,E)
FN1 IF N no es hoja,
    Por cada entrada E' de N cuyo MBB intersecta con el MBB de E,

        invocamos FindNode (N',E), donde N' es el nodo hijo N
        apuntado por E'..

    ELSE
        IF N contiene una entrada que se conecte a E,
            RETURN N

```

Algoritmo 1: Algoritmo para insertar en el STR-tree.

Algoritmo de Partición

En (Pfooser, y otros, 2000) se plantea como objetivo principal preservar las trayectorias en el índice, la división de un nodo de la hoja requiere de un análisis para determinar qué tipos de segmentos están contenidos en un nodo. Los segmentos de un nodo hoja pueden pertenecer a la misma trayectoria o no. Si

pertenecen a la misma trayectoria, pueden tener criterios de valoración comunes o no. “De esta forma se pueden determinar tres escenarios de partición de un nodo” (Pfooser, y otros, 2000) los cuales se muestran a continuación en la **Figura: 4**.

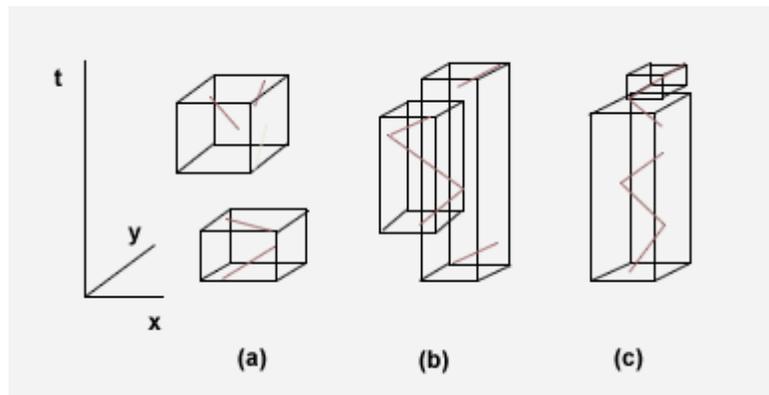


Figura 4: Diferentes escenarios de partición.

En la **Figura: 4 (a)** cuando los segmentos se encuentran desconectados se invoca al método de partición cuadrática para determinar la partición del nodo, en caso de la **Figura: 4 (b)** donde no todos los nodos pero al menos uno se desconecta, los segmentos desconectados se colocan en el segmento recién creado y en el caso de la **Figura: 4 (c)** donde no existen segmentos desconectados, el más reciente con respecto al tiempo, conectado hacia atrás, se coloca en el nodo recién creado.

Combinación de la búsqueda del STR-Tree con el R-Tree

El primer paso según los autores en (Pfooser, y otros, 2000) para poder procesar este tipo de búsquedas combinada es recuperar un conjunto inicial de segmentos basados en un rango. Se aplica la idea de la búsqueda por rango usada por el R-Tree, la idea es descender por el árbol según las condiciones de la intersección hasta llegar a los nodos hojas buscando las entradas para recuperar las trayectorias de los objetos, de esta forma se completa la primera fase de la búsqueda combinada.

En un segundo paso se extraen estas trayectorias, siguiendo la idea de coger cada uno de estos segmentos encontrados y tratar de conectarlos con sus predecesores, primeramente con los que se encuentran en ese nodo hoja y luego en los restantes nodos hojas, tratando de conectar ambos extremos de la trayectoria. Un gran problema que pudiese ocurrir en este algoritmo a la hora de recuperar las

trayectorias, es que se recupere en dos ocasiones la misma trayectoria. Esto se evita guardando el número de la trayectoria y chequeando cada vez que se recupere una trayectoria para ver si esta ya fue recuperada.

TB-Tree

El método de acceso TB-tree propuesto en (Pfoser, y otros, 2000), es totalmente diferente al STR-tree. El STR-tree intenta que los segmentos de una misma trayectoria queden almacenados juntos, en cambio el TB-tree considera una estructura de datos que permite acceder de manera directa a la trayectoria completa de un objeto. Este introduce una nueva metodología a la hora de insertar y dividir un nodo con la idea de lograr una mejor orientación a la trayectoria guardada. Con el TB-tree se aspira a un método de acceso que conserve estrictamente todas las trayectorias de un mismo objeto, de manera que en un nodo hoja solo queden los segmentos que pertenezcan a la misma trayectoria, por lo que este índice se puede entender como un conjunto de trayectorias. *“La forma de poder unir cada uno de los segmentos de una misma trayectoria en el método de acceso TB-Tree es por la incorporación de una lista simplemente enlazada”* (Pfoser, y otros, 2000).

Estructura de datos

Los nodos hojas de un TB-tree contienen segmentos que pertenecen a la trayectoria de un mismo objeto. Una de las desventajas del TB-tree es que los segmentos de trayectorias de distintos objetos quedan almacenados en bloques diferentes a pesar de estar espacialmente cercanos, lo que tiene como consecuencia una baja capacidad de discriminación espacial y esto a su vez tiene un efecto negativo sobre el rendimiento de las consultas de tipo instante de tiempo e intervalo de tiempo. En el TB-tree se propone disminuir esta restricción particionando la trayectoria total en tramos y manteniendo la trayectoria total por medio de una lista doblemente enlazada de los tramos. La estructura de un nodo hoja queda de la forma $\langle id, MBR, o \rangle$ donde id es el identificador del nodo hoja o es la orientación de la trayectoria dentro del MBR . La estructura solo resta de los datos que posee el STR-Tree el dato tyi que es el identificador de la trayectoria, ya que la estructura solo permite almacenar segmentos de una misma trayectoria, este dato solo se almacena en la cabecera de cada nodo.

Inserción

El procedimiento para insertar propuesto en (Pfoser, y otros, 2000), plantea cortar la trayectoria entera del movimiento de un objeto en piezas, donde cada pieza contenga M segmentos de líneas, por ejemplo un nodo hoja contiene M segmentos de la trayectoria. La **Figura: 5** ilustra el procedimiento de inserción. Las etapas más importantes son marcadas con negro y se encuentran circuladas con los números del 1 al 6.

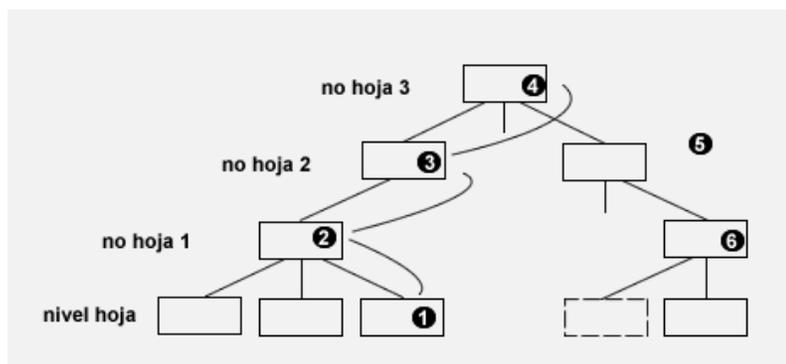


Figura 5: Diferentes etapas de la inserción.

Para poder insertar una nueva entrada simplemente se busca el nodo hoja que contenga su predecesor en la trayectoria. Se inicia atravesando el árbol, desde la raíz y pasando por cada nodo hijo que se solape con el MBR de la nueva línea de segmento.

```

Algoritmo de Insert(N,E)
INS1 Se invoca a FindNode(N,E)
INS2 IF N' se encuentra,
    IF N' tiene espacio,
        Se inserta en nuevo segmento
    ELSE
        Se crea un nuevo nodo para el nuevo segmento
ELSE
    Se crea un nuevo nodo para el nuevo segmento
    
```

Algoritmo 2: Algoritmo para insertar en el TB-Tree.

Se debe elegir el nodo hoja que contenga algún segmento conectado hacia la nueva entrada (etapa 1 en la **Figura: 5**). La búsqueda de un segmento es descrita en el algoritmo FindNode, que es idéntica a la

implementada por el STR-tree. En caso de que el nodo hoja esté lleno, se aplica una estrategia de partición. Particionando un nodo hoja se puede violar el principio de la preservación de la trayectoria total. Debido a esto se procede a crear un nuevo nodo hoja. En la **Figura: 5** se avanza en el árbol hasta que se encuentre un nodo padre que no está lleno (etapa 2 a la 4). Se elige la ruta que se encuentra más a la derecha (etapa 5) para insertar un nuevo nodo. Si existe espacio en el nodo padre (etapa 6) se inserta el nuevo nodo hoja como se muestra en la figura. En caso de que este se encuentre lleno, se hace una partición del nodo mediante la creación de un nuevo nodo (no hoja) en nivel 1 que tiene el nuevo nodo hoja como su único descendiente. Si es necesario la partición se seguirá realizando.

Búsqueda

El algoritmo de búsqueda es muy similar al descrito en el STR-tree, la diferencia que existe es en cómo se recuperan las trayectorias. Se tienen dos posibilidades que los segmentos se encuentren en el mismo nodo o no. Si se encontrase en el mismo nodo es trivial encontrar estos segmentos conectores, de ser en el caso contrario se tendría que aplicar la lista simplemente enlazada para poder conectar las trayectorias de un mismo objeto (Pfooser, y otros, 2000).

1.3.1 Análisis del comportamiento histórico

Las técnicas propuestas en este epígrafe permiten recuperar información histórica sobre objetos en movimiento. Las cuales apuntan a responder consultas de tipo instante de tiempo, intervalo de tiempo y consultas sobre eventos. Se presenta una clasificación de las técnicas más representativas a fin de poder comprender su sintaxis.

HR-Tree

El método HR-tree propuesto en (Nascimento, y otros, 1998) se basa en el concepto de sobre posición y trata de impedir el exceso de almacenamiento producido al contar con un R-tree para cada uno de los instantes de tiempo. En la **Figura: 6** se muestran los cambios ocurridos para varios objetos para dos instantes de tiempo donde se sombrea solo los objetos que tuvieron alguna modificación sea espacial o temporal.

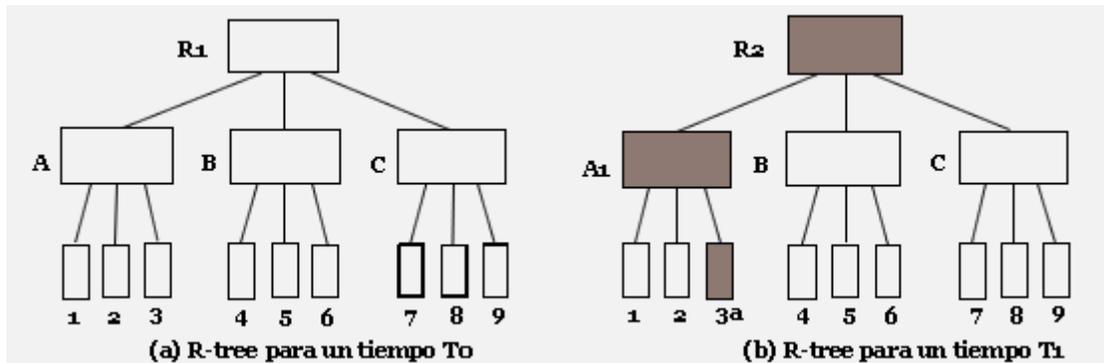


Figura 6: Subárboles para varios objetos en dos instantes de tiempo.

En la **Figura: 7** se aprecia el ahorro que alcanza esta técnica al evitar almacenar objetos comunes entre R-tree para los instantes de tiempo consecutivos, por lo que los sub-árboles pueden reutilizar los instantes de tiempo.

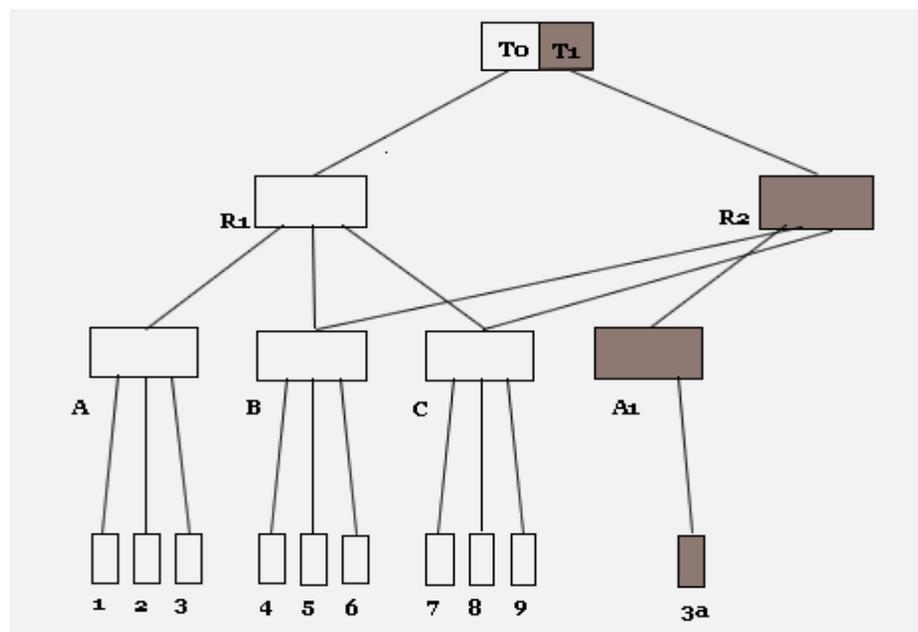


Figura 7: Indexación en el HR-Tree.

Consulta de tipo instante de tiempo

Para procesar una consulta de tipo **instante de tiempo**, el método HR-tree primero se ubica en el R-tree adecuado de acuerdo al tiempo especificado en la consulta. Posteriormente usando el árbol seleccionado, se recuperan todos los objetos cuyos atributos espaciales se interceptan con el rectángulo especificado en la consulta.

Consulta de tipo intervalo de tiempo

Una consulta del tipo **intervalo de tiempo** se procesa de una manera muy similar a la utilizada para procesar consultas de tipo **instante de tiempo**, sólo que es necesario recorrer varios R-tree. Para poder procesar la consulta especificada, en primer lugar se selecciona el R-tree del primer instante de tiempo. Luego, usando el rectángulo Q , se recuperan todos los rectángulos que se interceptan con Q . Seguido de este proceso se verifica si los rectángulos contenidos dentro de la región encontrada, se interceptan con Q , si resulta verdadero se exploran estos rectángulos verificando si los *MBRs* contenidos en ellos se interceptan con Q , recuperando así los objetos que satisfacen la consulta espacio-temporal.

SEST-Index

Según Retamal en (Gutiérrez Retamal, 2007) el método de acceso SEST-Index considera la creación de snapshots (considerado el estado de la base de datos en un instante de tiempo) teniendo en cuenta la posición de todos los objetos en el instante de tiempo en que se decide crear los snapshots; y las bitácoras mantienen la información de los eventos que se van produciendo en el espacio completo. Las entradas en las bitácoras se mantienen ordenadas por el tiempo y permiten reconstruir el estado de la base de datos entre dos snapshots consecutivos. El SEST-Index utiliza un R-tree para indexar los objetos considerando sus posiciones espaciales en el instante de tiempo en que se decide crear un nuevo snapshots. Por ejemplo, los objetos considerados en el snapshots creado en un instante de tiempo t_0 se almacenan en el R-tree R_0 y los eventos de los objetos producidos en el intervalo temporal t_0, t_1 se almacenan en la bitácora L_0 . De tal forma, que para recuperar el estado en la base de datos en un instante t con $t_0 < t < t_1$, se empieza desde el R-tree en el instante t_0 y se actualiza con la información contenida en la bitácora L_0 .

Estructura de datos

La estructura de datos para el R-tree es básicamente la misma que la propuesta por Guttman en (Antonin, 1984). La bitácora se implementa por medio de una lista encadenada simple de nodos. Las entradas en los nodos de una bitácora tienen la estructura $t, geom, oid, op$, donde t corresponde al tiempo en el cual la modificación sucedió por ejemplo, la inserción de un nuevo elemento, oid es el identificador del objeto, $geom$ corresponde a los valores de la componente espacial y que depende del tipo de objeto espacial que se requiere almacenar, e op indica el tipo de operación o evento es decir, inserción o eliminación. Si los objetos son puntos en un espacio de dos dimensiones, entonces $geom$ será un par de coordenadas x, y . Si los objetos son polígonos, entonces $Geom$ será un conjunto de puntos que lo definen o su aproximación espacial, por ejemplo un *MBR*.

En el SEST-Index se consideran dos tipos de operaciones o eventos: *move_in* (un objeto se mueve a una nueva posición) y *move_out* (un objeto deja de permanecer en su última posición). (Gutiérrez Retamal, 2007) De esta forma, la creación de un objeto se modela como un *move_in*⁵, una eliminación como un *move_out*⁶ y un objeto que cambió su posición o forma se modela como un *move_out* seguido de una operación *move_in*. Los snapshots se encuentran indexados por el tiempo. Para ello se utiliza un arreglo A , tal que $A_i.t$ indica el tiempo en que el i -ésimo snapshots fue creado, $A_i.R$ es el R-tree del snapshots y $A_i.L$ es la bitácora de eventos producidos entre $A_i.t + 1$ y $A_{i+1}.t$. En el SEST-Index también se considera un parámetro d ⁷ que define el largo máximo que pueden tener las bitácoras.

Consultas de tipo instante de tiempo

Para procesar una consulta de tipo Instante de tiempo Q, t el primer paso es encontrar el correspondiente snapshots de acuerdo al instante de tiempo t especificado en la consulta. Este snapshots corresponde al creado en el último instante de tiempo dentro del intervalo $0, t$. Luego, se realiza una búsqueda espacial usando el mismo método definido para el R-tree descrito en el capítulo 1. El conjunto

⁵ *Move_in* un objeto se mueve a una nueva posición

⁶ *Move_out* un objetos deja de permanecer en su última posición

⁷ Parámetro d (medido en cantidad de bloques de disco) que define el largo máximo de las bitácoras.

de objetos obtenidos por la búsqueda anterior se actualiza con las entradas de la correspondiente bitácora.

Consultas de tipo intervalo de tiempo

El procedimiento para procesar este tipo de consultas es similar al seguido para las consultas de tipo instante de tiempo. Así, para evaluar o procesar una consulta de tipo intervalo de tiempo, (Q, t_i, t_f) primero se ubica el último snapshot creado en el intervalo $[0, t_i]$, sea $t \leq t_i$ que este instante de tiempo. Luego, haciendo una búsqueda en el R-tree seleccionado, se recupera el conjunto de los objetos que se interceptan espacialmente con Q . Posteriormente este conjunto se actualiza con los eventos ocurridos entre t, t_i , produciendo la respuesta para el instante t_i . Seguidamente, el conjunto se actualiza con los eventos ocurridos en el intervalo $[t_i, t_f]$, obteniendo los objetos en cada instante de tiempo del intervalo.

Consultas sobre eventos

Una de las novedades presentes en el método del SEST-Index, es la capacidad para procesar, no solamente consultas de tipo instante de tiempo e intervalo de tiempo, sino que además consultas sobre los eventos almacenados en las bitácoras. Dada una región Q y un instante de tiempo t , una consulta de este tipo permite encontrar el total de objetos que salió del rectángulo Q o ingresó al mismo en el instante t . Consultas de este tipo son posibles y útiles en aplicaciones que requieren analizar patrones de movimiento de los objetos. Una implementación simple de este tipo de consultas con el SEST-Index consiste en usar un arreglo B para ubicar el primer bloque de la bitácora que almacena el primer evento ocurrido en t sin tener que acceder al R-Tree para este tipo de consulta.

SESTL

La idea del SESTL descrita en (Gutiérrez Retamal, 2007) consiste en dividir, mediante un único R-tree, el espacio original en regiones al nivel de las hojas y asignarle una bitácora a cada una de estas regiones. Aquí una bitácora es una estructura que almacena en las hojas snapshots y eventos.

Estructura de datos

El SESTL también considera un R-tree con la misma estructura para las entradas de los nodos. Sin embargo, las bitácoras son diferentes ya que consideran dos tipos de entradas: eventos y entradas de

snapshots. El primer tipo de entradas (eventos) tiene la misma estructura y significado que las definidas en el SEST-Index. El segundo tipo de entrada almacena el snapshots de un nodo hoja por cada entrada de los objeto en el instante en que este fue creado. Las entradas en la bitácora se almacenan ordenadas por el tiempo. El SESTL también considera un parámetro d , igual al definido en SEST-Index para todas las bitácoras en la estructura, que define el número máximo de eventos que es posible almacenar entre snapshots consecutivos.

Consultas de tipo instante de tiempo

Para procesar una consulta de tipo instante de tiempo Q, t el primer paso consiste en encontrar todas las hojas que interceptan el rectángulo Q . Luego, para cada bitácora de cada hoja, se obtiene el snapshot adecuado de acuerdo al tiempo t . Este snapshots corresponde al creado en el último instante de tiempo tr tal que $tr < t$. Los objetos espaciales que están almacenados en el snapshot seleccionado y que se interceptan espacialmente con Q , forman el conjunto inicial de la respuesta.

Este conjunto se actualiza con los eventos almacenados como entradas de tipo evento de la bitácora dentro del intervalo tr, t . El proceso anterior se repite para cada una de las bitácoras cuya sub-región asignada se intercepta con Q . La respuesta de la consulta se consigue por medio de la unión de los conjuntos obtenidos a partir de cada una de las bitácoras.

Consultas de tipo intervalo de tiempo

Para procesar una consulta de tipo intervalo de tiempo $Q, (ti, tf)$ se requiere encontrar el conjunto de objetos que intercepta espacialmente con Q en el instante de tiempo ti . Esto es equivalente a ejecutar una consulta de tipo instante de tiempo en el instante ti . Luego, el conjunto de objetos se actualiza con los eventos ocurridos en el intervalo ti, tf .

Consultas sobre los eventos

Una de las novedades presentes en el método del SESTL, es la capacidad para procesar, no solamente consultas de tipo instante de tiempo e intervalo de tiempo, sino que además consultas sobre los eventos almacenados en las bitácoras, por ejemplo, dada una región Q y un instante de tiempo t , una consulta de este tipo permite encontrar el total de objetos que salió del rectángulo Q o ingresó al mismo en el

instante t . Consultas de este tipo son posibles y útiles en aplicaciones que requieren analizar patrones de movimiento de los objetos.

Consultas sobre patrones de movimiento

En (Chairman, y otros, 2001) se propone un modelo discreto (en el espacio y el tiempo) para procesar consultas sobre los patrones de desplazamiento que siguen los objetos dentro de un conjunto de regiones previamente definidas y fijas, tal como se puede ver en la **Figura: 8**.

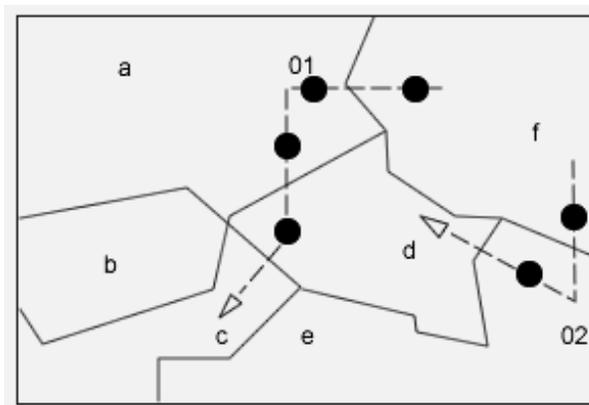


Figura 8 : Objetos moviéndose sobre un mapa particionados

En esta propuesta las trayectorias de los objetos se consideran como un string formado por los identificadores de las regiones. Una trayectoria de un objeto se representa como una secuencia que tiene la siguiente forma: $l1\{T1\}.l2\{T2\}...ln\{Tn\}$, caracterizando las sucesivas regiones por donde un objeto pasó y el tiempo en que se mantuvo en cada una de ellas. Por ejemplo, la secuencia $a\{2\}.b\{4\}.c\{8\}.d\{2\}$, indica que el objeto primero pasó por la región a donde permaneció 2 unidades de tiempo, luego se trasladó a la región b donde se mantuvo por 4 unidades de tiempo. Por otra parte, las consultas sólo se pueden ejecutar sobre un tramo de la trayectoria de un objeto y no sobre su historia completa.

De igual forma una consulta STP (Spatio-Temporal Pattern) se define como una secuencia m de predicados espacio-temporales de la forma: $\{(Q1, T1), (Q2, T2), \dots, (Qm, Tm)\}$ donde en cada par (Q, T) , Q representa un predicado espacial y T una restricción temporal. Una consulta STP puede estar formada por predicados, donde cada predicado puede estar asociado con un instante de tiempo o intervalo de tiempo

(STPWith Time - STPWT) o, más generalmente, con un orden relativo con los otros predicados (STPWith Order - STPWO). Un ejemplo de una consulta STPWT es la siguiente: “encontrar los objetos que cruzaron la región A en t_1 , llegaron a estar lo más cerca posible del objeto B en t_2 y luego se detuvieron dentro del círculo C en algún instante del intervalo $[t_3, t_4]$ ”. Un ejemplo de una consulta STPWO es la siguiente: “encontrar los objetos que primero cruzaron la región A, luego pasaron lo más cerca posible del objeto B y finalmente se detuvieron dentro del círculo C”. En este último tipo de consultas sólo interesa el orden relativo de los predicados espaciales independientemente de cuándo ocurrieron exactamente. Una forma de evaluar las consultas de tipo STPWT es evaluando cada predicado individualmente y luego combinar cada una de estas respuestas parciales para obtener una respuesta definitiva.

Para consultas STPWO en (Hadjieleftheriou, y otros, 2005), se propone una estructura de datos llamada CellList que corresponde a una partición uniforme del espacio en celdas. En esta estructura, cada celda se representa por un único identificador. A cada celda se le asocia una lista ordenada de entradas del tipo (P_l, t_k) , donde cada entrada indica el instante de tiempo en que un objeto alcanzó la celda, por ejemplo, el objeto P_3 cruzó la celda B en el instante de tiempo 3 y en el instante 4 el objeto P_1 entró a la celda B y P_2 entró en el instante 9 a B y en el instante 13 lo hizo de nuevo. Por lo tanto, la lista para la celda B es $\{(P_1,4), (P_2,9), (P_2,13), (P_3,3)\}$, la celda A $\{(P_1,3), (P_1,5), (P_2,8)\}$ y la celda C $\{(P_2,4), (P_2,10), (P_3,2)\}$. La lista se mantiene ordenada por objeto e instante de tiempo **Figura: 9**.

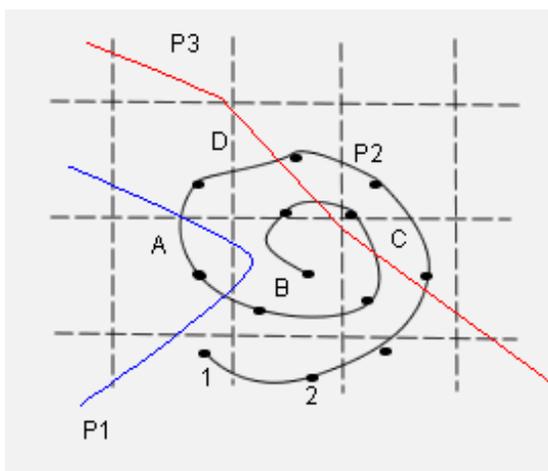


Figura 9 : Partición uniforme del espacio representado por celdas

Con esta estructura es posible realizar consultas de la forma: “*determinar los objetos que cruzaron A en $t = [5,8]$ y luego pasaron por B en $t = [9,13]$ ” lo cual conlleva procesar las listas de las celdas A y B.*

Tal como se comentó anteriormente, la estructura del CellList considera una partición del espacio. Esta estrategia también es utilizada por el SESTL, ya que también divide el espacio pero, en este caso, las celdas o subregiones se obtienen a partir del R-tree y por lo tanto no son necesariamente del mismo tamaño. Cada entrada de tipo evento en una bitácora tiene la estructura $\langle Oid, t, Geom, Opi \rangle$ que es un superconjunto de los atributos considerados por el CellList $\langle Oid, ti \rangle$ y en consecuencia, se pueden asimilar a las bitácoras del SESTL con las listas de las celdas del CellList (tienen exactamente la misma función). Con estas consideraciones es posible aplicar, casi directamente, los algoritmos definidos para evaluar consultas STPWO utilizando el CellList, considerando como estructura de datos subyacente la del SESTL. Las consultas definidas para patrones de desplazamiento, descritos anteriormente son factibles evaluarlas con el CellList y por lo tanto, también es posible hacerlo con el SESTL. El CellList, sin embargo, es una estructura para consultas STPWO y su utilidad para evaluar otros tipos de consultas espacio-temporales no es tan clara.

1.3.2 Recuperar posiciones futuras de objetos en movimiento

La posibilidad de predecir el movimiento ha revolucionado totalmente la ciencia, fundamentalmente la que rodea el análisis de objetos en movimiento. Esto ayuda a resolver problemáticas asociadas a la predicción de posibles desastres, navegación inteligente y otras aéreas. Los algoritmos o estructuras tradicionales para almacenar y devolver la ubicación de los objetos estáticos no es suficiente para los nuevos entornos móviles. Las nuevas soluciones espacio-temporales son llamadas consultas posicionales con parámetros temporales. Preguntas tales como “*Reportar los carros que se encontrarán en la región R en los siguientes 5 minutos*”, son usualmente implementadas por estos algoritmos. Determinar la posible ubicación de un objeto en un determinado tiempo es un poco más complicado debido a que los actuales métodos con los que cuentan las BD espacio-temporales suponen que los objetos se mueven en un plano lineal, siendo esta situación en la práctica un poco más compleja debido a que los objetos pueden seguir diferentes patrones de movimiento.

En autor Yufei Tao en el artículo titulado (Prediction and Indexing of Moving Objects with Unknown Motion Patterns, June 13-18 2004) establece que para poder superar todos estos problemas se establece un

marco general para indexar los objetos en movimiento, donde cada objeto calcula la función que describe su movimiento con exactitud. A fin de evitar las continuas actualizaciones de la ubicación de los objetos esta solo guarda la función de movimiento descrita $o(t)$ por cada objeto o que devuelva su ubicación en sus estados futuros t . Los trabajos existentes en este tema se centran en objetos que describen movimientos lineales, determinándose por la siguiente función $o(t) = o(t_0) + v_0(t - t_0)$ donde t_0 es la marca de tiempo usada por el objeto y $o(t_0)$ es la localización de o en un tiempo t_0 y v_0 es su velocidad correspondiente. En la práctica la mayoría de los movimientos no son lineales y la adopción del modelo lineal no resuelve los posibles cambios arbitrarios que pueden seguir los objetos. Aunque las funciones lineales pueden aproximarse a las curvas estas no presentan una gran efectividad frente a la diversidad de funciones descritas por los objetos en movimiento. Aún con la importancia que tiene, existen muy pocos trabajos que abordan la predicción en general.

Conclusiones Parciales

En el presente capítulo se realizó un análisis general de la problemática existente y los elementos asociados a la representación de objetos móviles y razonamiento espacio-temporal sobre la semántica del movimiento. Se realizó un estudio de las principales propuestas para la representación de objetos móviles junto a una clasificación de las mismas a fin de poder entender su semántica. Se pudo identificar que existen pocas propuestas que se enfocan directamente en el dominio de objetos móviles y que existen algunas áreas abiertas a la investigación como son: el estudio profundo de la predicción, debido a los pocos trabajos que se enfocan en este dominio de objetos móviles.

Capítulo 3: Presentación de la Solución Propuesta

1.4 Introducción

En este capítulo se realiza una comparación de las técnicas con el objetivo de sustentar la solución propuesta. Se estarán analizando las técnicas en términos de almacenamiento y procesamiento de las consultas. Con esta propuesta se aportan beneficios y mejoras que se obtienen con el uso de estos algoritmos para el control de flotas.

1.5 Criterios de Comparación

Con el objetivo de poder evaluar solo el rendimiento de las estructuras de acceso, se han definido dos criterios de comparación. Las comparaciones estarán enfocadas en el almacenamiento y eficiencia en las consultas. Una consulta para obtener “cuántos objetos entraron a un área determinada en un instante t ”, se resuelve realizando una consulta de tipo instante de tiempo para un instante t y otra en el instante inmediatamente anterior, para luego verificar qué objetos se encuentran en el conjunto respuesta de la primera consulta, pero no en el conjunto respuesta de la segunda. De esta forma puede ser evaluada la eficiencia de las consultas y además se pueden extender las conclusiones a las que se ha arribado en otros métodos de acceso propuestos en la literatura, con el objetivo de comparar los rendimientos de las estructuras.

En las siguientes tablas se muestran un resumen de las características más significativas de cada una de las estructuras descritas en este capítulo, con el objetivo de poder establecer una comparación entre ellas considerando los criterios definidos.

SEST-Index	Características
Almacenamiento	El SEST-Index utiliza menos espacio en la medida que el número de objetos disminuye.
Tipos de Consultas	
 Intervalo de tiempo	La estructura requiere accesar menos cantidad de bloques cuando el porcentaje de movilidad es bajo. Aumentando su rendimiento en la

	medida que la longitud del intervalo temporal aumenta.
✚ Instante de tiempo	El rendimiento de esta estructura mejora en la medida que el parámetro d disminuye.
✚ Sobre eventos	La estructura mantiene un rendimiento más o menos constante frente a las variaciones del parámetro d .

Tabla 1: Características más destacadas del método de acceso SEST-Index.

HR-Tree	Características
Almacenamiento	En porcentajes de movilidad bajos o cuando hay menos cantidad de objetos el método utiliza mucho espacio de almacenamiento.
Tipos de Consultas	
✚ Intervalo de tiempo	En intervalos temporales altos el rendimiento de esta estructura se degenera. Esto se debe a que la estructura necesita leer cada uno de R-Tree asociados a estos instantes de tiempo.
✚ Instante de tiempo	El rendimiento se mantiene constante frente a las variaciones de movilidad de los objetos. Esto se debe a que este método solo necesita leer un solo R-tree dentro de los R-tree virtuales.
✚ Sobre eventos	La estructura necesita acceder muchos bloques para este tipo de consultas debido a que requiere leer dos R-tree consecutivos para descubrir los eventos.

Tabla 2: Características más destacadas del método de acceso HR-Tree

SESTL	Características
Almacenamiento	Utiliza menos espacio en la medida que número de objetos aumenta. Se puede destacar que esta estructura para un valor de $d(d = 8)$ es cuando requiere de la menor cantidad de espacio para el almacenamiento.
Tipos de Consultas	
 Instante de tiempo	En un rango espacial formado por un porcentaje mayor a un 60% , en cada dimensión disminuye la eficiencia de esta estructura. Esto es debido a que en la estructura los eventos producidos se encuentran repartidos en varias bitácoras y en la medida que el rango espacial aumenta. Es necesario procesar gran cantidad de bitácoras para darle respuesta a este tipo de consulta.
 Sobre eventos	Para un rango espacial que sea mayor o igual a un 11%, con cada dimensión que se procesa la eficiencia de esta estructura disminuye debido a la gran cantidad de bitácoras en las que reparte los eventos según van sucediendo.

Tabla 3 : Características más destacadas del método de acceso SESTL.

TB-Tree	Características
Almacenamiento	El tamaño de utilización de la estructura es bastante pequeño. Para un tamaño de un índice de 51kb por objetos utiliza un 100% del espacio.
Tipos de Consultas	
 Intervalo de tiempo	La estructura presenta un buen rendimiento.
 Instante de tiempo	Presenta un buen rendimiento en la medida que el número de

	objetos que se estén analizando sean pocos.
--	---

Tabla 4: Características más destacadas del método de acceso TB-Tree.

STR-Tree	Características
Almacenamiento	El tamaño de utilización de la estructura es grande. Para un tamaño de un índice de $57kb$ por objetos utiliza un 100% del espacio.
Tipos de Consultas	
 Intervalo de tiempo	No presenta un buen rendimiento por la cantidad de bloques de disco que debe acceder.
 Instante de tiempo	Si existen grandes concentraciones de objetos a analizar la estructura presenta un buen rendimiento.

Tabla 5 Características más destacadas del método de acceso STR-Tree.

1.5.1 Comparación entre el SEST-Index y el HR-Tree

Evaluación Experimental

Con el objetivo de evaluar el rendimiento del SEST-Index, éste fue comparado con el HR-tree en términos de almacenamiento y eficiencia de las consultas (instante de tiempo, intervalo de tiempo y sobre eventos). Se eligió comparar con HR-tree, ya que éste es el método más conocido según la bibliografía consultada que resuelve de manera más eficiente las consultas de tipo instante de tiempo. Con HR-tree es posible procesar consultas sobre eventos utilizando las respuestas de las consultas instante de tiempo. Por ejemplo, para obtener cuántos objetos entraron a un área determinada en un instante t , se resuelve realizando una consulta de tipo instante de tiempo en el instante t y otra en el instante inmediato anterior para luego verificar qué objetos se encuentran en el conjunto respuesta de la primera consulta pero no en el conjunto respuesta de la segunda. De esta forma se puede comparar la eficiencia de las consultas sobre eventos al procesarlas con SEST-Index y con HR-tree y extender las conclusiones a otros métodos de acceso propuestos en la literatura que evalúan dichas consultas utilizando los resultados de dos o más consultas de tipo instante de tiempo. Para comparar el almacenamiento necesario o utilizado por el SEST-

Index e HR-tree se usó el número de bloques utilizados por la estructura después de insertar todos los objetos y los eventos. Por otra parte, la eficiencia de las consultas fue medida como el número de bloques accedidos (Gutiérrez Retamal, 2007).

Almacenamiento utilizado por el SEST-Index y el HR-Tree

El SEST-Index usa menos espacio que el HR-tree. Esta diferencia aumenta en la medida que el número de objetos disminuye o el porcentaje de movilidad aumenta. También es posible, que conforme el tamaño del parámetro d disminuye, el espacio requerido por el SEST-Index aumenta, llegando al caso extremo en que necesita casi el mismo almacenamiento que el HR-Tree para $d = 4$ véase **Figura: 11**. Esta debilidad del SEST-Index se explica por el hecho de que con el parámetro d pequeño es muy probable que quepan en cada bitácora los eventos correspondientes a un solo instante de tiempo. En tal situación es necesario crear un snapshot por cada instante de tiempo y, por lo tanto, la estructura degenera en muchos R-tree individuales (uno por cada instante de tiempo almacenado en la base de datos). Una idea simple para reducir el espacio usado por el SEST-Index es considerar la estrategia básica del HR-tree al momento de crear un nuevo R-tree en un snapshot, reutilizar parte del R-tree previo. Esta idea fue evaluada por Retamal resultando en los experimentos obtenidos por el autor, que el ahorro es muy pequeño dependiendo del número de objetos, del porcentaje de movilidad y del parámetro d , véase la **Tabla: 6**. El escaso ahorro se explica por el hecho que los eventos producidos entre snapshots provocan que la reutilización de los R-tree sea baja y de esta forma mientras mayor sea la cantidad de eventos que pueda almacenar la bitácora (parámetro d), menor será el ahorro de espacio (Gutiérrez Retamal, 2007).

Número de Objetos	Porcentajes de ahorro		
	$d = 4$	$d = 8$	$d = 12$
1500	7.3	8.6	9.9
2500	8.6	7.8	8.1
3500	10.0	8.3	8.1
4500	9.6	9.0	7.0
7500	10.8	9.4	8.5

10500	7.2	8.6	6.3
13500	10.0	7.0	7.5
12500	7.7	4.5	5.7
17500	8.1	5.2	5.2
22500	5.6	5.5	5.5

Tabla 6 : Porcentajes de ahorro de almacenamiento del SEST-Index conseguido por medio de la estrategia de reutilización del espacio del HR-tree.

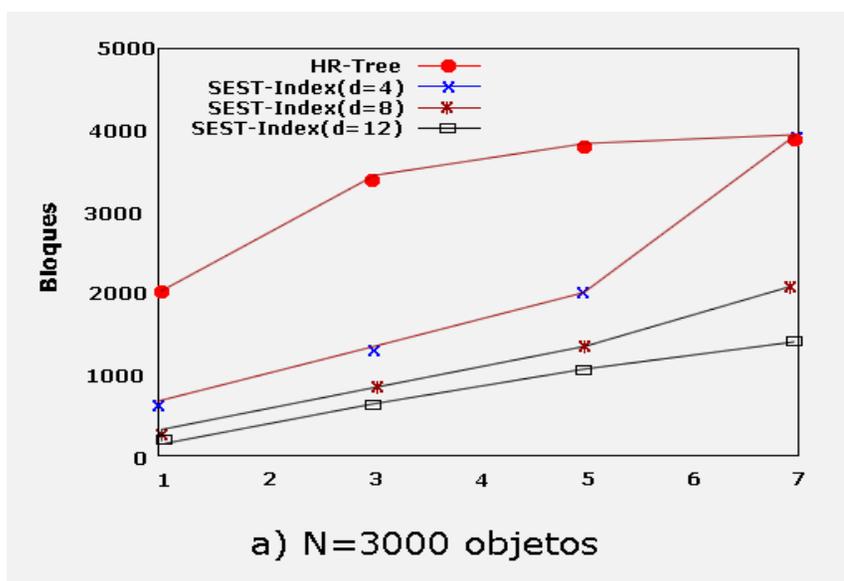


Figura 10 : Almacenamiento utilizado por el SEST-Index y el HR-Tree.

Consultas utilizadas por el SEST-Index y el HR-Tree

📌 Consulta de tipo instante de tiempo

En (Gutiérrez Retamal, 2007) según los experimentos presentados el SEST-Index necesita leer más bloques que el HR-tree para este tipo de consultas. Para la mayoría de los casos el SEST-Index, en promedio, necesitará realizar d_2 accesos más que el HR-tree. También es posible observar en la **Figura: 12** tomada de (Gutiérrez Retamal, 2007), que el rendimiento del HR-tree se mantiene constante frente a

las variaciones de la movilidad de los objetos. Es posible observar que el rendimiento del SEST-Index mejora en la medida que el parámetro d disminuye.

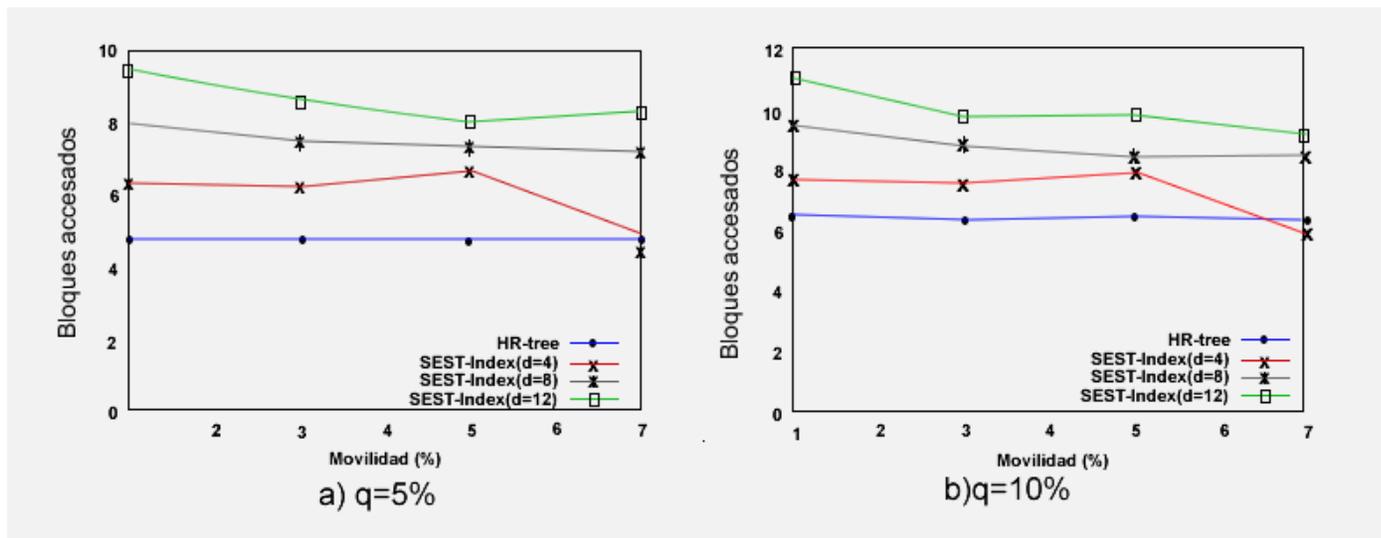


Figura 11: Número de bloques accedidos por una consulta de tipo instante de tiempo utilizando el SEST-Index (rango espacial formado por el $q\%$ de la longitud de cada dimensión).

🚦 Consulta de tipo intervalo de tiempo

Según los experimentos realizados en (Gutiérrez Retamal, 2007), fue posible observar que el SEST-Index necesita acceder muchos menos bloques que el HR-tree. En la medida que el intervalo temporal aumenta, la ventaja del SEST-Index aumenta con respecto al HR-Tree. Esta ventaja se debe al hecho de que el HR-tree requiere leer dos R-tree consecutivos para descubrir los eventos, mientras que con el SEST-Index es posible recuperar los eventos de manera directa desde las bitácoras, sin necesidad de recorrer ningún R-tree, véase la **Figura: 13**.

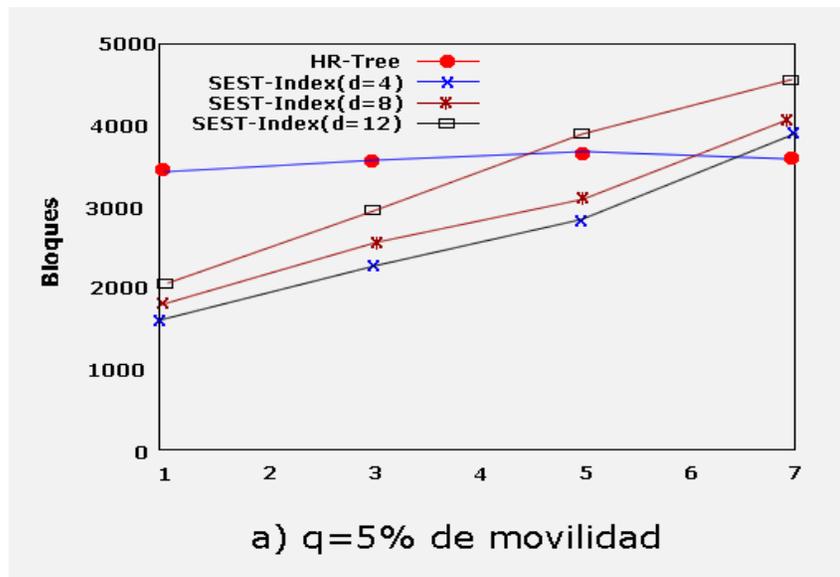


Figura 12 Número de bloques accedidos por una consulta de tipo intervalo de tiempo utilizando el SEST-Index.

✚ Consulta de tipo evento

Según los experimentos realizados por el autor es posible apreciar que el SEST-index mantiene un rendimiento más o menos constante frente a las variaciones del parámetro d . En los resultados descritos en (Gutiérrez Retamal, 2007) se observó que el SEST-Index necesita acceder muchos menos bloques que el HR-tree. Esto se debe al hecho de que el HR-tree requiere leer dos R-tree consecutivos para descubrir los eventos, mientras que con el SEST-Indexes posible recuperar los eventos de manera directa desde las bitácoras, sin necesidad de recorrer ningún R-tree.

Conclusión de la evaluación

El SEST-Index presenta deficiencias en cuanto al almacenamiento siendo superado en este punto por el HR-Tree. No obstante para las consultas de tipo instante de intervalo y sobre eventos se puede apreciar un buen rendimiento.

La desventaja presentada en cuanto al almacenamiento se debe al rápido crecimiento de la estructura, lo cual se explica por:

- ✚ El mayor número de snapshots (R-Trees) que es necesario crear en la medida que el porcentaje de movilidad o la cantidad de objetos aumenta.
- ✚ Cada vez que se crea un nuevo R-tree (nuevo snapshot), todos los objetos se duplican incluyendo aquellos que no han sufrido modificaciones entre snapshots consecutivos.

En la **Tabla: 7** se muestra un resumen con los resultados más destacados de las evaluaciones vistas del SEST-Index y el HR-Tree.

SEST-Index y el HR-Tree	Evaluación
Almacenamiento	SEST-Index requiere de menos espacio que HR-tree para porcentajes de movilidad bajos.
Tipos de Consultas	
✚ Intervalo de tiempo	SEST-Index supera al HR-tree en la medida que el área o el largo del intervalo aumenta.
✚ Instante de tiempo	HR-tree supera al SEST-Index.
✚ Sobre eventos	SEST-Index resultó ser más eficiente que el HR-tree.

Tabla 7 : Resultados destacados de la evaluación del SEST-Index y el HR-Tree.

1.5.2 Comparación entre el SEST-Index y el SESTL

Evaluación Experimental

En (Gutiérrez Retamal, 2007) hicieron evaluaciones en las que se comparó el SESTL contra el SEST-Index. Se compararon ambos métodos en términos de almacenamiento y tiempo necesario para procesar las consultas. Se utilizaron datos de 23.268 objetos (puntos) distribuidos de manera uniforme dentro del espacio en el instante 0 cuyas posiciones cambian a lo largo de 200 instantes de tiempo, estos también de manera uniforme.

Almacenamiento utilizado por el SEST-Index y el SESTL

En los experimentos realizados se puede apreciar como el SESTL supera el método de acceso SEST-Index. Por ejemplo, en los experimentos se estuvo evaluando el almacenamiento en megabytes el cual fue fijado a 40 megabytes, en el cual el SESTL acceso alrededor de 300 bloques. En cambio SEST-Index requirió acceder más de 3.000 bloques para poder darle respuesta a la consulta efectuada.

El rendimiento del SEST-Index es desfavorable con respecto al SESTL, teniendo en cuenta dos factores:

- a) El SEST-Index duplica en cada snapshot todos los objetos, incluyendo aquellos que no sufren modificaciones en sus atributos espaciales. El problema de la duplicación, en principio, se podría resolver utilizando la estrategia de sobre posición utilizada por el HR-tree, aunque los resultados obtenidos muestran que el ahorro en espacio es bajo, alrededor de un 7% y 8%. Por el contrario, el SESTL construye un nuevo snapshot sólo para las hojas que han sufrido cambios.
- b) El SEST-Index agrupa los eventos sólo por tiempo y no por tiempo y espacio. Todos los eventos que ocurren entre snapshot consecutivos se almacenan en una sola bitácora, los cuales pueden requerir de varios bloques de disco. A la hora de procesar una consultad de tipo instante de tiempo el SEST-Index obtiene un conjunto inicial de objetos y este no filtra los bloques de las bitácoras duramente el procesamiento. Según el autor, este problema puede ser resuelto disminuyendo el parámetro d , aunque esto lo que hace es agudizar el problema expuesto en a).

Consultas utilizadas por el SEST-Index y el SESTL

Consulta de tipo instante de tiempo.

En (Gutiérrez Retamal, 2007) fue comparado el rendimiento de ambas estructuras para evaluar consultas de tipo instante de tiempo. El autor consideró para evaluar en ambas estructuras una movilidad de los objetos de un 10%. El experimento demostró que ambas estructuras usan aproximadamente la misma cantidad de espacio en el almacenamiento de los objetos. En (Gutiérrez Retamal, 2007), consideró para los experimentos un $d = 816$ bloques para el SEST-Index y un $d=4$ para el SESTL. El autor formula como conclusión, después de haber evaluado estas dos estructuras frente a estas condiciones, que el SEST-Index supera a SESTL para consultas de tipo instante de tiempo siendo esto posible cuando el rango

espacial está formado por un porcentaje superior a 60% en cada dimensión. El rendimiento favorable del SEST-Index, con respecto a SESTL, se explica por el hecho de que todos los eventos producidos en un mismo instante de tiempo se encuentran en bloques contiguos en la bitácora del SEST-Index. En cambio, en el SESTL se encuentran repartidos en varias bitácoras y en la medida que el rango espacial aumenta como se formula para un porcentaje por encima de un 60%, es necesario procesar una mayor cantidad de bitácoras para obtener la respuesta para este tipo de consulta.

Consulta de tipo evento

El SEST-Index supera a SESTL para este tipo de consulta en el caso de que el rango espacial esté formado por un porcentaje en cada dimensión mayor o igual a 11%. El SEST-Index supera en medida a la cantidad de bloques accedidos por el SESTL, lo cual se debe a que este método puede acceder de manera directa al primer bloque de la bitácora donde son almacenados los eventos de cada uno de los instantes de tiempo.

Conclusión de la evaluación

El SEST-Index no presenta un buen rendimiento en cuanto al almacenamiento frente a la estructura SESTL. Este rendimiento es desfavorable teniendo en cuenta dos factores:

-  El SEST-Index duplica en cada snapshot todos los objetos.
-  El SEST-Index agrupa los eventos sólo por tiempo y no por tiempo y espacio.

En cuanto a las consultas de tipo instante de tiempo en los experimentos se demostró que el SEST-Index utiliza el mismo espacio de almacenamiento que el SESTL. Y el SEST-Index supera al SESTL para consultas de tipo evento en la medida que la cantidad de bloques accedidos por el SESTL, lo cual se debe a que este método puede acceder de manera directa al primer bloque de la bitácora donde son almacenados los eventos de cada uno de los instantes de tiempo.

En la **Tabla: 8** se muestra un resumen con los resultados más destacados de las evaluaciones vistas del SEST-Index y el SESTL.

SEST-Index y el SESTL	Evaluación
Almacenamiento	El SEST-Index no presenta un buen rendimiento en cuanto al almacenamiento frente a la estructura SESTL
Tipos de Consultas	
⚡ Instante de tiempo	El SEST-Index supera al SESTL.
⚡ Sobre eventos	El SEST-Index supera al SESTL.

Tabla 8: Resultados destacados de la evaluación del SEST-Index y el SESTL.

1.5.3 Comparación entre el STR-Tree y el TB-Tree

En (Pfooser, y otros, 2000) se hicieron evaluaciones en las que se comparó el STR-Tree contra el TB-Tree. Se compararon ambos métodos en términos de almacenamiento y tiempo necesario para procesar las consultas.

Almacenamiento utilizado por el STR-Tree y el TB-Tree

En los experimentos realizados por Pfooser en (Pfooser, y otros, 2000) muestra el tamaño de dos índices diferentes (STR-Tree, TB-Tree) y el tamaño utilizado por los mismos, donde el espacio utilizado por las estructuras TB-Tree y STR-Tree es de un 100%. Para un juego de datos utilizados de 1000 objetos el tamaño alcanzado por estos índices es de alrededor 57 MB según el autor. Esta diferencia se debe principalmente a que la estructura de indexación. En los experimentos vistos el índice espacio-temporal TB-Tree es mucho más pequeño que el STR-Tree, aunque los dos índices utilizan casi el mismo espacio de almacenamiento. Para una mejor comprensión de lo antes expuesto ver la **Figura: 14**.

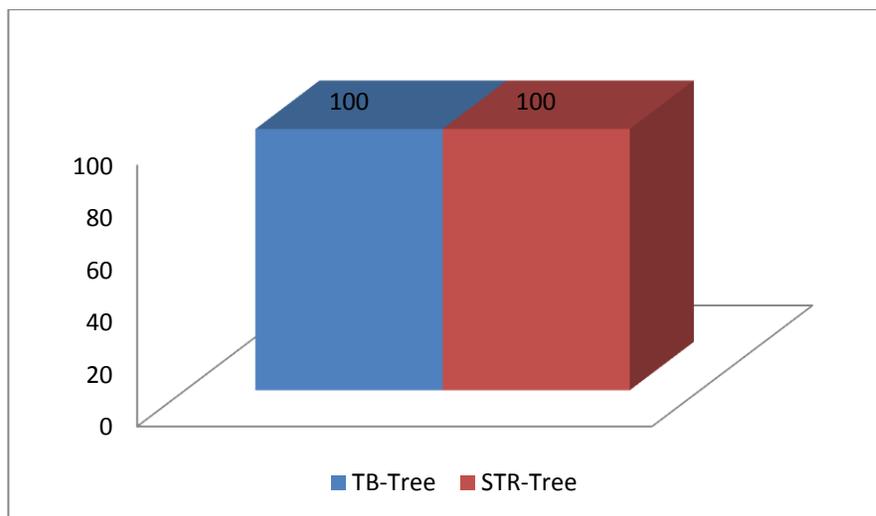


Figura 13 : Espacio utilizado por los métodos de indexación TB-Tree y STR-Tree.

Consultas utilizadas por el STR-Tree y el TB-Tree

✚ Consulta de tipo intervalo de tiempo

En el experimento visto se utilizó un conjunto de datos generados desde 10 hasta 1000 objetos en movimiento. Donde se aprecia como el rendimiento del STR-Tree con respecto al TB-Tree es inferior debido que la estructura STR-Tree debe acceder a más bloques del disco para procesar la misma cantidad de objetos en movimientos **Figura: 15**.

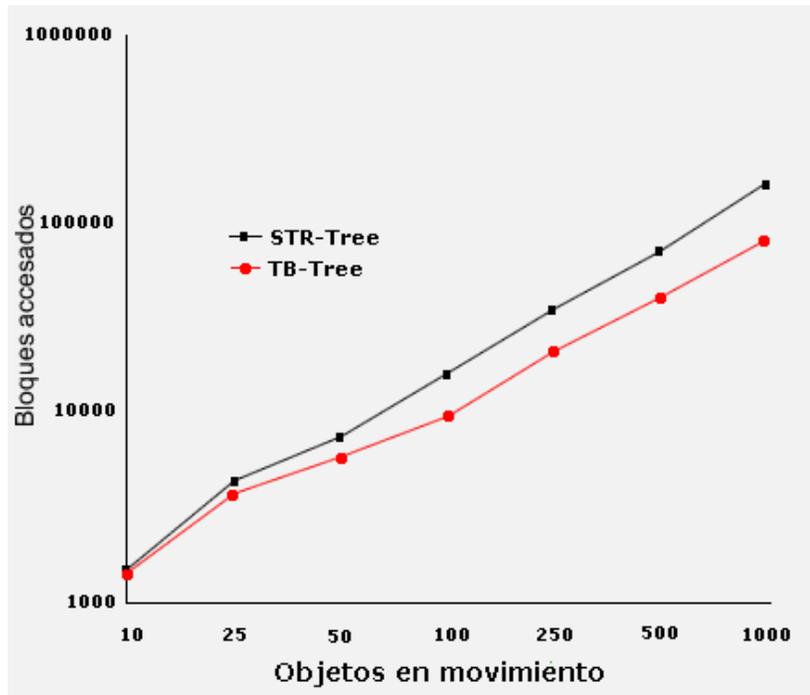


Figura 14 : Número de bloques accedidos por una consulta de tipo intervalo de tiempo utilizando TB-Tree y STR-Tree.

✚ Consulta de tipo instante de tiempo

De igual manera se analizó un juego de datos de 10 a 1000 objetos en movimiento, donde se aprecia como el rendimiento del TB-Tree es superior al STR-Tree en la medida que el número de objetos que se estén analizando sean pocos así como la cantidad de bloques a acceder. En caso de una mayor concentración de objetos a analizar la diferencia entre la cantidad de bloques del disco a acceder cambia superando STR-Tree a la estructura TB-Tree en cuanto a su rendimiento **Figura: 16**.

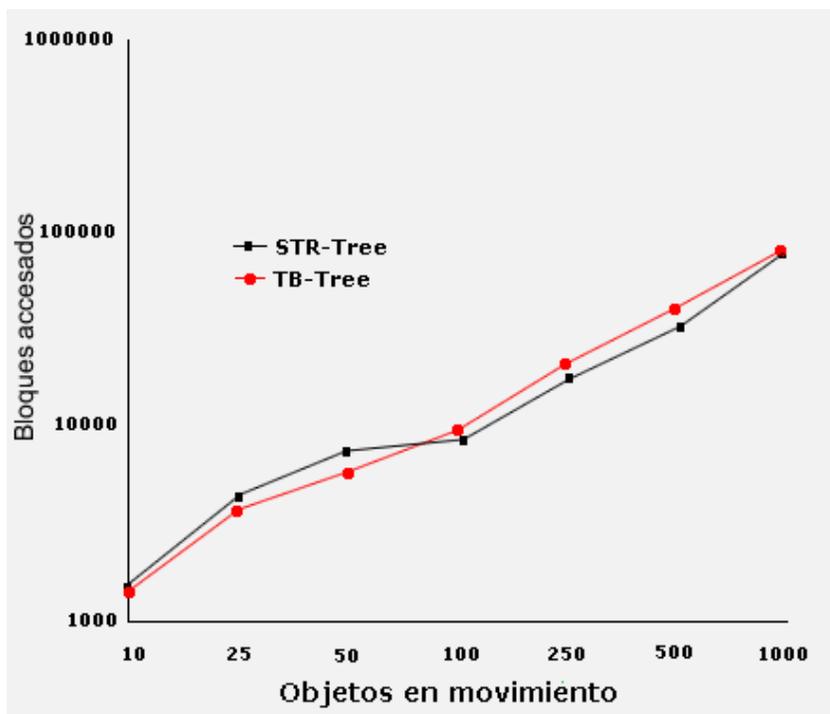


Figura 15 : Número de bloques accedidos por una consulta de tipo instante de tiempo utilizando el, TB-Tree y STR-Tree.

Conclusión de la evaluación

Ambas estructura presentan el mismo espacio de almacenamiento, a pesar de que la estructura TB-Tree es considerada más pequeña que STR-Tree. En cuanto a las consultas de tipo instante de tiempo en los experimentos se demostró que el método TB-Tree presenta un buen rendimiento en la medida que el número de objeto que se estén analizando sean pocos no siendo así en el caso contrario donde la estructura STR-Tree muestra un mejor rendimiento. En el caso de las consultas de tipo intervalo de tiempo la estructura STR-Tree debe acceder a más bloques de disco para poder darle respuesta.

En la **Tabla: 9** se muestra un resumen con los resultados más destacados de las evaluaciones vistas del STR-Tree y TB-Tree.

STR-Tree y TB-Tree	Evaluación
Almacenamiento	El índice espacio-temporal TB-Tree es mucho más pequeño que el STR-Tree, aunque a pesar de esta diferencia utilizan casi el mismo espacio de almacenamiento.
Tipos de Consultas	
 Instante de tiempo	El rendimiento del TB-Tree es superior al STR-Tree en la medida que el número de objetos que se estén analizando sean pocos así como la cantidad de bloques a acceder.
 Intervalo de tiempo	El rendimiento del STR-Tree con respecto al TB-Tree es inferior debido que la estructura STR-Tree debe acceder a más bloques del disco.

Tabla 9: Resultados destacados de la evaluación del STR-Tree y TB-Tree.

1.5.4 Solución Propuesta

Con el objetivo de proporcionarle un trabajo más factible a los desarrolladores de la UCI, especialmente a aquellos que trabajan dentro del proyecto Control de Flotas, se realiza una propuesta de algoritmos. Esta propuesta agilizará el trabajo referente a la implementación del Sistema para el Control de Flotas aportándole rapidez y la preparación de los datos espacio-temporales frente a las diferentes consultas realizadas. Como parte de este proceso de selección de propuesta se realizaron diferentes comparaciones, las cuales fueron reflejadas en el documento anteriormente. De las comparaciones antes descritas, se agrupan y analizan las principales técnicas de indexación para representar objetos móviles, analizar posición y trayectorias, predecir movimientos futuros siendo considerando esta última novedosa.

Análisis de la trayectoria de objetos en movimiento

Realizados los estudios necesarios y las pruebas pertinentes, se decidió seleccionar para las técnicas de análisis de trayectorias de objetos el método de acceso espacio-temporal TB-Tree. En dicho método se considera una estructura de datos que permite acceder de manera directa a la trayectoria completa de un objeto, con el objetivo de lograr una mejor orientación de la trayectoria guardada. Con esta estructura se tiene como principal objetivo la preservación de las trayectorias aportando una estabilidad a la hora de

guardar las trayectorias de las flotas monitoreadas. Otra de las ventajas es que este utiliza muy poco espacio de almacenamiento de los objetos.

El método de acceso TB-Tree a la hora de procesar consultas de tipo intervalo de tiempo, en la medida que va aumentando el rango espacial en el cual se están consultando los objetos mejora su rendimiento siendo esto factible a la hora de procesar grandes intervalos de tiempo aunque no sea así a la hora de poder procesar consultas de tipo instante de tiempo. Aun cuando existen estas fallas a la hora de realizar consultas de tipo instante de tiempo el rendimiento de la estructura es estable.

Análisis del comportamiento histórico

Realizados los estudios necesarios y las pruebas pertinentes, se decidió seleccionar para las técnicas de análisis del comportamiento histórico de los objetos móviles los métodos de acceso espacio-temporal SEST-Index y SESTL. La estructura de acceso SEST-Index presenta deficiencias en cuanto al almacenamiento, por la duplicación que realiza de los objetos aunque para procesar consultas de tipo instante de tiempo y sobre eventos la estructura presenta un buen rendimiento. El rendimiento favorable de la estructura, se explica por el hecho de que todos los eventos producidos en un mismo instante de tiempo se encuentran en bloques contiguos en las bitácoras.

Por otra parte la estructura SESTL presenta una gran ventaja por su similitud con la estructura CellList definida para procesar consultas STPWO. Esta similitud es posible debido a que cada entrada de tipo evento en una bitácora del SESTL es un gran conjunto de los atributos considerados por el CellList y en consecuencia, se pueden asimilar a las bitácoras del SESTL. Con estas consideraciones es posible aplicar, casi directamente, los algoritmos definidos para evaluar consultas STPWO utilizando el CellList, considerando como estructura de datos subyacente la del SESTL. Brindando una gran ventaja a la hora de procesar consultas sobre patrones de desplazamiento.

Conclusiones Parciales

El presente capítulo durante su desarrollo tuvo como principal objetivo formular la solución propuesta. Se estuvieron viendo experimentos realizados por algunos autores con el objetivo de determinar las técnicas más idóneas a ser utilizadas por los Sistemas de Gestión de BD. Se describe una breve caracterización de cada una de las técnicas teniendo en cuenta el almacenamiento y las consultas especializadas que

estas procesan. Se compararon primeramente las técnicas de indexación de trayectorias de objetos móviles y sucesivamente las de indexación de datos histórico de los objetos. Luego de cada una de estas comparaciones es presentada una tabla que refleja de manera muy breve las ventajas y desventajas de cada comparación.

Conclusiones Generales

Existe mucho interés en incorporar a los Sistemas de Administración de Bases de Datos tipos de datos espacio-temporales con el propósito de apoyar la implementación de tales aplicaciones. Dicho interés ha abierto un campo amplio de investigación. En el presente documento se hace un análisis de los métodos que ayudan a la recuperación de datos históricos y análisis de trayectorias de objetos móviles. De esta forma se resume que:

- ✚ La estructura SEST-Index necesita de menos espacio que el HR-Tree para porcentajes de movilidad bajos, siendo el HR-Tree una de las estructuras que optimiza el rendimiento, evitando el almacenamiento de objetos comunes para instantes de tiempo consecutivos. Las ventajas presentadas por el SEST-Index sobre el HR-Tree para procesar consultas de tipo intervalo de tiempo y sobre eventos aumenta en la medida que aumenta el área que se está consultando así también con el aumento de la longitud del intervalo. Aunque en el caso de las consultas para un instante de tiempo determinado se puede apreciar como el SEST-Index necesita de más accesos a bloques de disco que el HR-Tree para darle respuesta a la consulta efectuada.
- ✚ De igual forma fue analizada la estructura SESTL con respecto al SEST-Index siendo está considerada mejor que el HR-Tree y se pudo apreciar en los experimentos vistos como la estructura SESTL superaba a la estructura SEST-Index en espacio utilizado no siendo así frente al rendimiento de las diferentes consultas.
- ✚ Fueron analizados los métodos de acceso espacio-temporal TB-Tree y STR-Tree en términos de almacenamiento siendo considerado que las dos estructuras presentan el mismo espacio de almacenamiento, aunque se consideró que la estructura TB-Tree utiliza poco espacio a la hora de guardar cada una de las páginas en disco.
- ✚ También se demostró que con la estructura SESTL es posible procesar consultas sobre patrones espacio-temporales y patrones de movimiento. Además fue encontrada la variante CellList que es utilizada para procesar consultas STPWO la cual se ajusta a la estructura de las bitácoras definidas para el SESTL.

Luego de haber realizado estas comparaciones entre los métodos de acceso descritos con anterioridad, se propone para las técnicas de análisis de trayectorias de objetos móviles la estructura TB-Tree por las características antes expuestas que propician una mejor preservación de las trayectorias de las flotas. Y para los métodos de análisis del comportamiento histórico se determinó como las más idóneas las estructuras SEST-Index y SESTL por las características antes expuestas.

Recomendaciones

Se recomienda en este trabajo, dar cumplimiento a los siguientes puntos:

- ✚ Poner en práctica la solución propuesta en este trabajo, la cual facilitará y ahorrará tiempo con la implementación de estas herramientas de indexación de datos espacio-temporal.

- ✚ Realizar un estudio más abarcador sobre las técnicas de predicción de trayectorias, debido a la falta de bibliografía que las caracterizara de una mejor forma para su posterior comprensión.

- ✚ Utilizar la documentación generada sobre técnicas de indexación espacio-temporal para el análisis de trayectorias de objetos móviles como material de consulta en la realización de trabajos u otros documentos.

Referencias Bibliográficas

1. **Aguayo González, Rafael.** *Scribd. Scribd.* [En línea] [Citado el: 6 de octubre de 2010.] <http://www.scribd.com/doc/22030503/Diccionario-Gis-Indexado>.
2. **Antonin Guttman.** 1984. *R-Trees. A Dynamic Index Structure.* California : s.n., 1984.
3. **Aref, Mohamed F. Mokbel Thanaa M. Ghanem Walid G.** [En línea] [Citado el: 22 de 02 de 2011.]
4. **Chairman, Dr. Amitabh Mishra, Chairman, Dr. Srinidhi Varadarajan y Committee Member, Dr. Luiz Dasilva.** 2001. *Mobility Pattern Aware Routing in Mobile Ad Hoc.* 2001.
5. **Gámez Sosa, Efraín Ricardo.** 2009. *Estudio de métodos para proteger y explotar datos.* 2009.
6. **Guillaumet, y otros.** 2000. *Minería de Datos Espaciales.* Facultad Regional de Rosario. Rosario : s.n., 2000.
7. **Gutiérrez Retamal, Gilberto Antonio.** 2007. *Métodos de Acceso y Procesamiento de Consultas Espacio-Temporales.* Santiago - Chile : s.n., 2007.
8. **Hadjieleftheriou, Marios y Kollios, George .** 2005. *Complex Spatio-Temporal Pattern Queries.* Computer Science Department. 2005.
9. **Manso, Miguel Ángel.** Escuela Técnica Superior de Ingenieros. [En línea] [Citado el: 05 de octubre de 2010.] <http://www.topografia.upm.es/pdi/m.manso/docencia/bbddee2010/0.-presentacion.pdf>.
10. **Nascimento, Mario A., Silva, Jefferson R. O. y Theodoridis, Yannis.** 1998. *Access Structures for Moving Points.* 1998.
11. **Pfoser, Dieter, Jensen, Christian S y Theodoridis, Yannis.** 2000. *Novel Approaches to the Indexing of Moving Object Trajectories.* Cairo, Egypt : s.n., 2000.
12. *Prediction and Indexing of Moving Objects with Unknown Motion Patterns.* **Yufei Tao, Christos Faloutsos, Dimitris Papadias, Bin Liu.** June 13-18 2004. Paris, France. : s.n., June 13-18 2004.

13. **Spaccapietra, Stefano, y otros. 2007.** [En línea] 29 de mayo de 2007. [Citado el: 8 de noviembre de 2010.] <http://www.google.com.cu/url?sa=t&source=web&cd=1&ved=0CBkQFjAA&url=http%3A%2F%2Fciteerx.ist.psu.edu%2Fviewdoc%2Fdownload%3Fdoi%3D10.1.1.172.3105%26rep%3Drep1%26type%3Dpdf&rct=j&q=A%20Conceptual%20View%20on%20Trajectories&ei=roELTcbiDIbWnAebza35DQ&usg=A>.

Bibliografía

1. **Agarwal, Pankaj K, Lars Arge y Jeff Erickson. 2002.** *Indexing Moving Points*. 2002.
2. **Aguayo González, Rafael.** *Scribd. Scribd.* [En línea] [Citado el: 6 de octubre de 2010.] <http://www.scribd.com/doc/22030503/Diccionario-Gis-Indexado>.
3. **Antonin Guttman. 1984.** *R-Trees. A Dynamic Index Structure*. California : s.n., 1984.
4. **Aref, Mohamed F. Mokbel Thanaa M. Ghanem Walid G.** [En línea] [Citado el: 22 de 02 de 2011.]
5. **Chairman, Dr. Amitabh Mishra, Chairman, Dr. Srinidhi Varadarajan y Committee Member, Dr. Luiz Dasilva. 2001.** *Mobility Pattern Aware Routing in Mobile Ad Hoc*. 2001.
6. **Gámez Sosa, Efraín Ricardo. 2009.** *Estudio de métodos para proteger y explotar datos*. 2009.
7. **Giraudi, Daniela C, Segura Guzmán, Gabriela S y Gagliardi, Edilma O.** *Propuesta de diseño de un índice espacio-temporal vinculado a redes móviles*. Matemáticas y Naturales, Departamento de Informática, Universidad Nacional de San Luis, Facultad de Ciencias Físico. San Luis, Argentina : s.n. pág. 10.
8. **Guillaumet, y otros. 2000.** *Minería de Datos Espaciales*. Facultad Regional de Rosario. Rosario : s.n., 2000.
9. **Gutiérrez Retamal, Gilberto Antonio. 2007.** *Métodos de Acceso y Procesamiento de Consultas Espacio-Temporales*. Santiago - Chile : s.n., 2007.
10. **Hadjieleftheriou, Marios y Kollios, George . 2005.** *Complex Spatio-Temporal Pattern Queries*. Computer Science Department. 2005.
11. **Manso, Miguel Ángel.** Escuela Técnica Superior de Ingenieros. [En línea] [Citado el: 05 de octubre de 2010.] <http://www.topografia.upm.es/pdi/m.manso/docencia/bbddee2010/0.-presentacion.pdf>.

12. **Nascimento, Mario A., Silva, Jefferson R. O. y Theodoridis, Yannis. 1998.** *Access Structures for Moving Points.* 1998.
13. **Ning Hu, Minglong Shao. 2003.** *Kalman-Tree: An Index Structure on Spatio-Temporal Data.* Computer Science Department, Carnegie Mellon University. 2003.
14. **Pfoser, Dieter, Jensen, Christian S y Theodoridis, Yannis. 2000.** Novel Approaches to the Indexing of Moving Object Trajectories. Cairo, Egypt : s.n., 2000.
15. *Prediction and Indexing of Moving Objects with Unknown Motion Patterns.* **Yufei Tao, Christos Faloutsos, Dimitris Papadias, Bin Liu.** June 13-18 2004. Paris, France. : s.n., June 13-18 2004.
16. **Rodriguez, Andrea. 2005.** *Una revision de Base de Datos de objetos en movimiento.* 2005. pág. 10.
17. **Simonas Saltenis, Christian S. Jensen, Scott T. Leutenegger, Mario A. Lopez. 2000.** *Indexing the Positions of Continuously Moving Objects.* Department of Mathematics and Computer Science, University of Denver, Colorado, USA. 2000.
18. **Spaccapietra, Stefano, y otros. 2007.** [En línea] 29 de mayo de 2007. [Citado el: 8 de noviembre de 2010.] <http://www.google.com/cu/url?sa=t&source=web&cd=1&ved=0CBkQFjAA&url=http%3A%2F%2Fciteseerx.ist.psu.edu%2Fviewdoc%2Fdownload%3Fdoi%3D10.1.1.172.3105%26rep%3Drep1%26type%3Dpdf&rct=j&q=A%20Conceptual%20View%20on%20Trajectories&ei=roELTcbiDIbWnAebza35DQ&usg=A>.

Glosario de Términos

Algoritmo: Proveniente del latín dicitur *algorithmus*. Un algoritmo constituye una lista bien definida, ordenada y finita de operaciones que se ejecutan en un tiempo determinado empleando cierta cantidad de recursos computacionales para hallar la solución a un problema.

Árbol: Se llama árbol al grafo que no tiene ciclos y que enlaza todos sus puntos. Su importancia radica en que los arboles constituyen grafos que conectan todos los vértices de forma tal que se utilice el menor número de aristas posibles.

Eventos (Consultas sobre eventos): Este tipo de consultas permite verificar si ciertos eventos se han producido en un instante o periodo de tiempo. Los eventos pueden ser, por ejemplo el ingreso de un objeto a una región, la salida de un objeto desde una región o sencillamente la permanencia de un objeto en una región en un determinado instante o periodo de tiempo.

Inteligencia Artificial: Ciencia de la computación encargada de la creación de máquinas dedicadas a la implementación de tareas relacionadas con el comportamiento humano, logrando gran similitud al pensamiento del hombre.

Move_out: Un objeto deja de permanecer en su última posición.

Move_in: Un objeto se mueve a una nueva posición.

Snapshots: Un snapshots corresponde a un estado de la base de datos en un instante determinado, es decir, las posiciones espaciales de todos los objetos en dicho instante.

Vértice o Nodo: Los vértices constituyen uno de los dos elementos que conforman un grafo. Es considerada la unidad fundamental, ya que son tratados como objetos indivisibles y sin propiedades; ejemplo de ello es la representación en un grafo de una red carretera en la que las ciudades o construcciones representan los vértices.