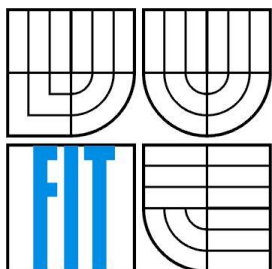


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

INDEXOVÁNÍ ČASOPROSTOROVÝCH DAT INDEXES FOR SPATIO-TEMPORAL DATA

VYBRANÉ PROBLÉMY INFORMAČNÍCH SYSTÉMŮ (VPD)
SEMESTRÁLNÍ PRÁCE PŘEDMĚTU

AUTOR PRÁCE

TOMÁŠ VOLF

BRNO 2012

Obsah

| | | |
|-----|--|----|
| 1 | Úvod..... | 2 |
| 2 | Zajímavé aspekty indexování časoprostorových dat..... | 3 |
| 4 | 2-3TR strom..... | 5 |
| 4.1 | 2-3TR strom na principu 2+3R stromu..... | 5 |
| 4.2 | 2-3TR strom na principu TB stromu..... | 5 |
| 4.3 | Přínosy 2-3TR stromu..... | 6 |
| 5 | PPFI index..... | 7 |
| 6 | ST ² B strom..... | 9 |
| 6.1 | Klíč pro indexování času | 9 |
| 6.2 | Klíč pro indexování prostorových dat | 10 |
| 6.3 | Klíč indexu..... | 11 |
| 7 | HR strom..... | 12 |
| 7.1 | Hashovací tabulky HR stromu..... | 12 |
| 7.2 | Sestavování HR stromu | 13 |
| 8 | Recursively Partitioned Trajectory Index (RPTI) | 15 |
| 8.1 | Indexování prostorových dat | 15 |
| 8.2 | Indexování času | 16 |
| 9 | TMN strom..... | 17 |
| 9.1 | Indexování prostorových dat | 17 |
| 9.2 | Indexování času | 18 |
| 9.3 | Indexování identifikátorů pohybujících se objektů..... | 19 |
| 9.4 | Tabulka přilehlých hran..... | 19 |
| 9.5 | Soubor se záznamy trajektorií..... | 19 |
| 10 | Závěr | 21 |

1 Úvod

Mnoho aplikací v dnešní době potřebuje nějakým způsobem ukládat prostorová data vyšších dimenzí nebo dokonce časoprostorová data pohybujících se objektů, respektive jejich trajektorií. Typickým příkladem mohou být například různé sledovací a dohledové systémy, aplikace pro letový provoz nebo monitorování dopravní situace ve městě či různé simulační programy.

Ukládat taková data do běžných relačních databází je velice neefektivní, protože pro tato data potřebujeme speciální podporu na úrovni ukládání dat tak, aby data byla efektivně uložena, a podporu na úrovni indexování dat pro rychlejší a efektivnější dotazy. Z tohoto důvodu je indexování trajektorií pohybujících se objektů je v posledních letech tématem výzvy pro výzkum.

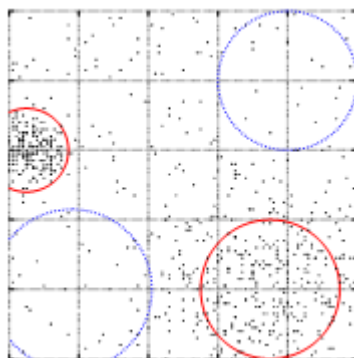
Tato práce si klade za cíl hlouběji zmapovat dostupné techniky pro indexování pohybujících se objektů, respektive jejich trajektorií. Mnoho indexačních technik vychází ze základních, jako jsou například R stromy či jejich odvozeniny, případně jsou založeny na kombinaci různých přístupů. Těmito základními principy se tato práce nezabývá, práce je věnována novějším poznatkům na poli indexačních technik.

Práce je dále členěna následovně: v kapitole 2 jsou zmíněny zajímavé aspekty indexování časoprostorových dat, v kapitolách 3 – 9 jsou představeny jednotlivé indexy (2-3TR strom, PPFÍ index, ST²B strom, HR strom, Recursively Partitioned Trajectory Index (RPTI) a TMN strom) a v kapitole 10 je stručně shrnuta tato práce.

2 Zajímavé aspekty indexování časoprostorových dat

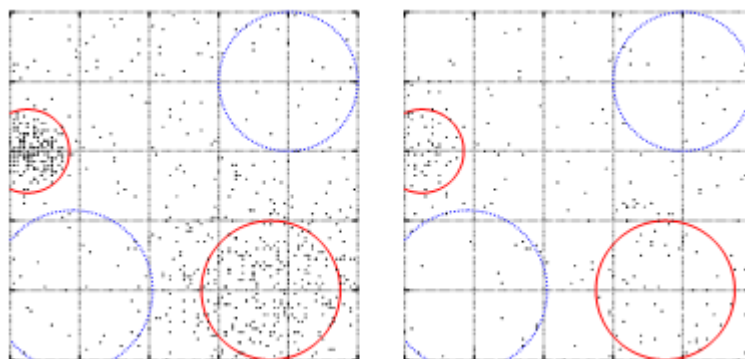
Z hlediska indexování trajektorií jsou zajímavé různé typy dotazů. Jedním z dotazů je rozsahový dotaz na časoprostorová data, kdy zadáme nějaké dotazové okno na dvoudimenzionální prostor a čas, například „Najdi všechny objekty, které se nacházely v oblasti $r = (x_1, y_1, x_2, y_2)$ a v čase $t = (t_1, t_2)$ “. V jiném případě může být užitečným dotazem například zjištění celé trajektorie, například „Zjisti, kudy všude prošel objekt o “. Takový dotaz může být samozřejmě omezen například na nějakou oblast či časový interval, důležité je však to, že chceme získat celou trajektorii v dané oblasti či v daném čase. Dále nás může zajímat dotaz na trajektorie na základě topologického vztahu, například „Zjisti, zda se potkal člověk p_1 s člověkem p_2 “. Dalším zajímavým dotazem z hlediska trajektorií může být dotaz na pozici objektu v budoucnosti, například „Zjisti pozici meteoritu m_1 za 10 let.“. Jak již bylo v příkladu naznačeno, takovýto dotaz může posloužit například v astronomii pro výpočty budoucí dráhy vesmírných těles na základě charakteru jejich pohybu v minulosti.

Z pohledu pohybujících se objektů je také zajímavá různorodost dat, a to konkrétně různorodost v prostoru, různorodost v čase a různorodost jak v čase, tak i prostoru. V případě různorodosti v prostoru objekty v daném prostoru nemusí být vůbec rovnoměrně rozložené, jak je zobrazeno na obrázku 1. Červeně jsou označena místa s vysokou koncentrací objektů, červeně naopak místa s nízkou koncentrací objektů.



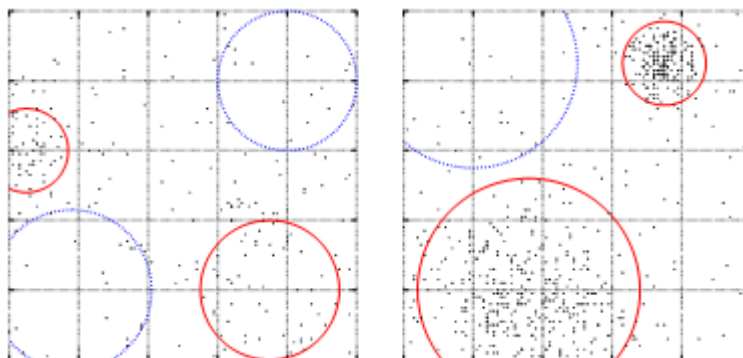
Obrázek 1: Různorodost dat v prostoru [4]

V případě různorodosti časové se po nějaké době může změnit koncentrace objektů v daných oblastech. Situace je znázorněna na obrázku 2, kdy oblasti zůstaly stejné, avšak koncentrace objektů v těchto oblastech se po nějakém čase změnila.



Obrázek 2: Různorodost dat v čase [4]

Různorodost jak v prostoru, tak i čase je kombinací prostorové různorodosti a časové různorodosti dohromady. Kromě změny koncentrace objektů v oblastech se společně s tím, jak se objekty pohybují, může pohybovat také koncentrace objektů jak v prostoru, tak i čase. Situace je ukázána na obrázku 3, kde se za nějaký čas τ přesunuly oblasti s vysokou koncentrací objektů na jinou pozici, stejně tak i oblasti s nízkou koncentrací objektů. Typickým příkladem jsou například pohyby lidí mezi domovem a prací, kdy ráno se velká koncentrace objektů pohybuje od domovů směrem k průmyslovým a kancelářským oblastem, a večer je tomu naopak.



Obrázek 3: Různorodost dat jak v prostoru, tak i v čase

4 2-3TR strom

Indexování pomocí 2-3TR stromu [1] je postaveno na principu 2+3R stromu a TB stromu.

4.1 2-3TR strom na principu 2+3R stromu

Samotný 2-3TR strom se, podobně jako 2+3R strom, skládá z dvou R stromů, přičemž není nutné používat přímo R stromy. Místo R stromů mohou být použity jeho odvozeniny, jako například R^* strom nebo Hilbertovy R stromy.

První R strom se používá pro indexování dvourozměrných bodů, které určují aktuální prostorovou informaci dat, jako je například aktuální pozice objektu v nějakém prostoru. Současně s prostorovou informací je uchovávána také časová informace, odkdy tato data drží danou prostorovou informaci.

Druhý R strom se používá pro indexování třídímenzionálních bodů a čar, které k daným datům uchovávají historii informací. Třídímenzionální bod uchovává pozici, na které se daný objekt nacházel v jediném časovém bodě. Oproti tomu třídímenzionální čára uchovává pozici, na které se daný objekt nacházel v průběhu času.

4.2 2-3TR strom na principu TB stromu

Reprezentace 2-3TR stromu založeného na TB stromu se skládá z objektů `FeatureSnapshot` a `FeatureTrajectory`.

`FeatureSnapshot` obsahuje informace o pozici pomocí prostorových souřadnic a také časovou informaci. Časová informace může být udána jako časové razítko nebo jako časový interval složený z počátečního a koncového času. Dále objekt `FeatureSnapshot` obsahuje pole, které propojuje daná časová razítka s odpovídajícími množinami atributů. Takto mohou být různé množiny dat uloženy do pole v případě, kdy objekt setrval na stejné pozici v průběhu času; naopak pokud byl objekt na dané pozici v průběhu času pouze jednou, bude v poli pouze jediný záznam.

Každý objekt `FeatureSnapshot` ukazuje na objekt `FeatureTrajectory`. Objekt `FeatureTrajectory` uchovává všechny pozice, na kterých se daný objekt v čase nacházel, pomocí pole, které k jednotlivým časovým razítkům přidružuje ukazatele na odpovídající objekty `FeatureSnapshot`. V případě, že se objekt na dané pozici nacházel déle v průběhu času, je odkazován pouze, je odkazován pouze počátečním časem, kdy se na této pozici objevil. Jednotlivé záznamy pole jsou seřazeny podle času.

4.3 Přínosy 2-3TR stromu

V 2-3TR stromu je možné využít dotazování pomocí rozsahu hodnot. Jelikož každý objekt `FeatureSnapshot` obsahuje ukazatel na svůj záznam v rámci objektu `FeatureTrajectory`, v němž je zaznamenána celá trajektorie sledovaného objektu, a zároveň každý objekt `FeatureTrajectory` obsahuje ukazatele na dílčí pozice trajektorie reprezentované objektem `FeatureSnapshot`, pak je možné z jediného konkrétního objektu `FeatureSnapshot` postupně získat všechny následující pozice sledovaného objektu pomocí pole v objektu `FeatureTrajectory`.

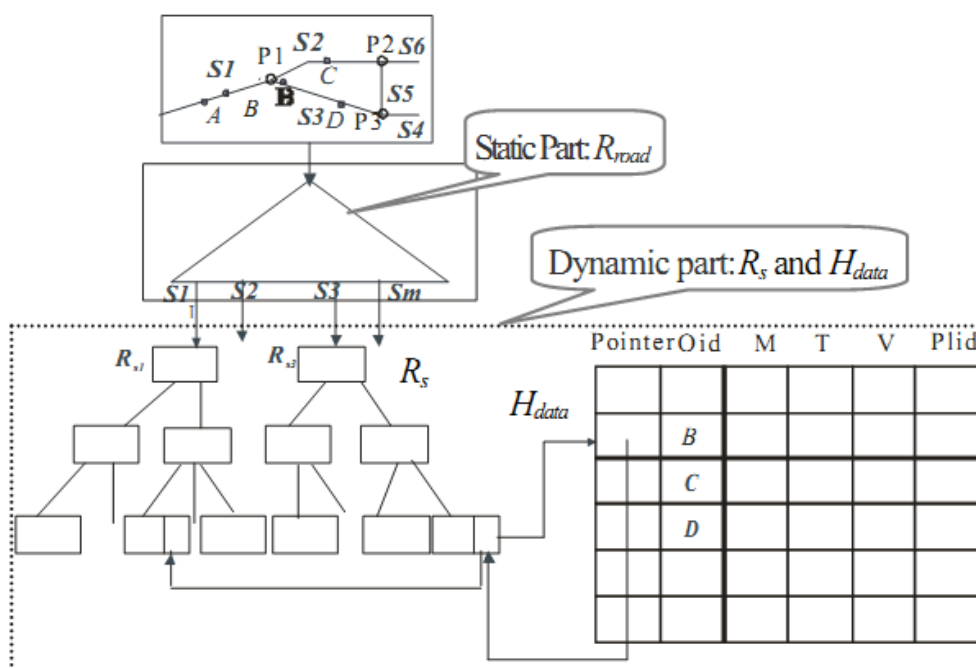
Stejně tak je možné získat pomocí pole ve `FeatureTrajectory` určitý úsek trajektorie, například pozice sledovaného objektu v průběhu daného časového intervalu.

5 PFFI index

PPFI index [5] (Past-Present-Future Index) je indexační technikou umožňující ukládání trajektorií pohybujících se objektů, jak se pohybovaly v minulosti, aktuální pozice pohybujících se objektů, ale také pro předpovídání pozice, na které se budou v blízké budoucnosti nacházet pohybující se objekty v pevné síti.

Struktura indexu PFFI je hybridní a skládá se ze statické a dynamické části. Statickou část tvoří 2D R* strom, který je založen na křivkách popisujících úseky silnic pro správu pevných sítí. Dynamická část obsahuje množinu 1D R* stromů, které indexují časové intervaly trajektorií pohybujících se objektů, které se po těchto křivkách pohybovaly. Dále dynamická část obsahuje hashovací tabulku, která umožňuje zaznamenávat aktuální stavy pohybujících se objektů a také předpovídání budoucí pozice pohybujících se objektů.

Na obrázku 4 je znázorněna celková struktura PFFI indexu. Statickou část tvoří 2D R* strom, v němž jsou indexovány silnice úseky silnic R_{road} . Každý listový uzel tohoto stromu ukazuje na časový 1D R* strom R_s .



Obrázek 4: Celková struktura PFFI indexu [5]

Tyto časové 1D R* stromy R_s indexují pohyb objektu po dané křivce. Nelistové uzly tohoto stromu obsahují záznamy ve tvaru $\langle childptr, T_1, T_2 \rangle$, kde $childptr$ je ukazatelem na příslušný synovský uzel a T_1 a T_2 definují časový rozsah, který je tímto synovským uzlem spravován.

Listové uzly obsahují strukturu ve tvaru $\langle Oid, Plid, T_1, T_2, M_s, M_e, P_1, P_d \rangle$, v níž Oid reprezentuje identifikátor daného pohybujícího se objektu, $Plid$ je identifikátorem křivky,

na které se objekt nachází. Hodnoty T_1 , T_2 , M_s a M_e společně popisují pohyb daného objektu v rámci dané křivky, konkrétně hodnoty T_1 a T_2 vyjadřují časový interval pohybu tohoto objektu, který je uchován v tomto listovém uzlu, a hodnoty M_s a M_e udávají počáteční a koncovou vzdálenost objektu v daném časovém intervalu. Vzdálenost M_x nějakého bodu x je udávána jako vzdálenost podél křivky od počátečního bodu do bodu x .

Hashovací tabulka H_{data} zaznamenává čas T úplně poslední změny, vzdálenost M a rychlost V pohybujícího se objektu v síti, identifikátor křivky $Plid$, na které se pohybující se objekt momentálně nachází a identifikátor daného pohybujícího se objektu Oid . Navíc, aby bylo možné dostat se i z hashovací tabulky k historickým datům, je v hashovací tabulce ukazatel $Pointer$ ukazující zpátky do časového R_s stromu na stejný záznam, pomocí něžž je možné procházet strom hlouběji do historie pohybu daného objektu.

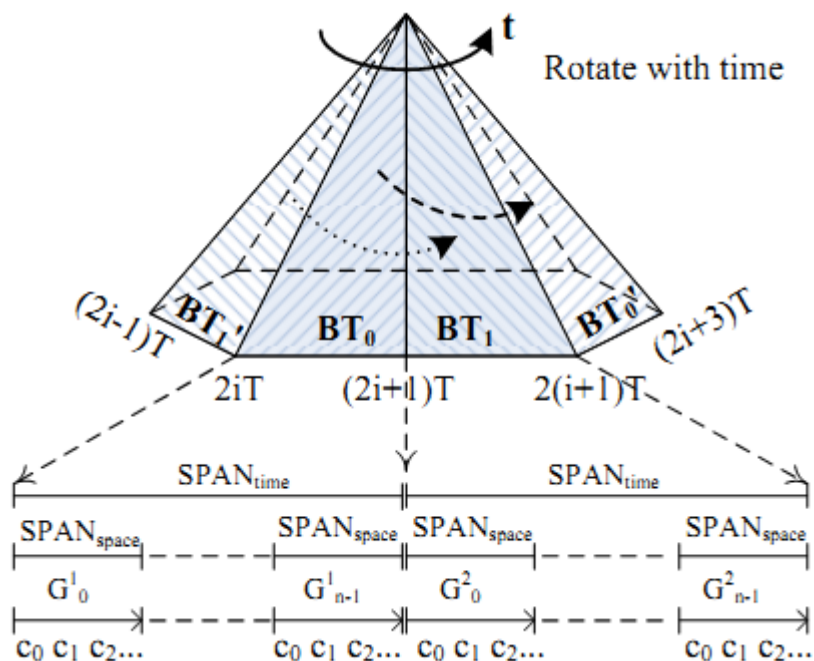
6 ST²B strom

ST²B strom [4] je založen na B⁺ stromu, do něhož nepřináší žádné změny jak z hlediska struktury, tak z hlediska algoritmů vkládání a mazání dat. Každý pohybující se objekt indexuje jako 1D bod, který se skládá ze dvou složek KEY_{time} a KEY_{space}.

6.1 Klíč pro indexování času

Vzhledem k pojmu pohybující se objekt, předpokládejme, že každý takový objekt změní alespoň jednu svojí polohu v čase T.

ST²B strom rozděluje B⁺ strom do dvou podstromů BT₀ a BT₁. Čas je rozdělován na určité časové rozsahy, které jsou těmto podstromům přiřazovány. První časový rozsah $[2iT, (2i+1)T)$ pokrývá první podstrom BT₀ a druhý časový rozsah $[(2i+1)T, (2i+2)T)$ náleží druhému podstromu BT₁, v nichž $i = 0, 1, 2, \dots$ V průběhu času se časové rozsahy rotují střídavě mezi těmito dvěma stromy. Index se v čase otáčí a tak se sám přizpůsobuje. Princip indexování pomocí ST²B stromu je znázorněn na obrázku 5.



Obrázek 5: Princip ST²B stromu

Samotné indexování do jednotlivých podstromů probíhá tak, že aktualizace uskutečněné v čase $[0, T)$ jsou indexovány pomocí prvního podstromu BT₀. Aktualizace uskutečněné v čase $[T, 2T)$ jsou indexovány naopak pomocí druhého podstromu BT₁. Následující aktualizace v čase $[2T, 3T)$ se vrací opět do podstromu BT₀, časový rozsah $[3T, 4T)$ indexuje opět podstrom BT₁ a tak dále.

Jako příklad předpokládejme objekt o , který v čase t_{up} aktualizuje (\vec{x}, \vec{v}) u daného objektu o , kde \vec{x} představuje umístění objektu, \vec{v} značí rychlost pohybu daného objektu v čase t_{up} . Objekt je umístěn do podstromu podle toho, do kterého časového rozsahu patří daný čas t_{up} .

Každý podstrom má svůj referenční čas T_{ref} , ve kterém je indexován objekt o s jeho umístěním. Umístění v čase T_{ref} lze vypočítat podle vzorce $\vec{x} = \vec{x} + \vec{v} \cdot (T_{ref} - t_{up})$. Referenční čas T_{ref} je nastaven na horní mez časové rozsahu, tedy:

$$T_{ref} = \begin{cases} (2i + 1)T, & \text{kdy } t_{up} \in [2iT, (2i + 1)T) \\ (2i + 2)T, & \text{kdy } t_{up} \in [(2i + 1)T, (2i + 2)T) \end{cases}$$

Pro identifikaci podstromu, ve kterém se indexovaný objekt o nachází, se používá proměnná KEY_{time} , jejíž hodnota se určuje následovně:

$$KEY_{time} = \begin{cases} 0, & \text{kdy } t_{up} \in [2iT, (2i + 1)T) \\ 1, & \text{kdy } t_{up} \in [(2i + 1)T, (2i + 2)T) \end{cases}$$

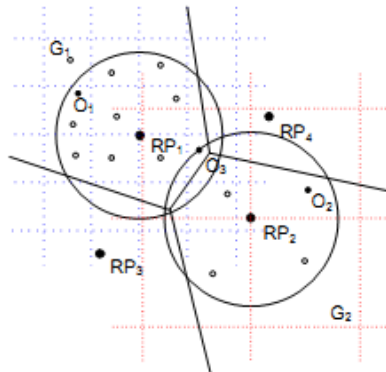
6.2 Klíč pro indexování prostorových dat

Prostorová data jsou indexována pomocí n referenčních bodů $\{RP_0, RP_1, \dots, RP_{n-1}\}$, kdy jsou data indexována do n disjunktních oblastí $\{VC_0, VC_1, \dots, VC_{n-1}\}$ na základě vzdálenosti od těchto referenčních bodů. Tyto oblasti spolu s jejich referenčními body tvoří síť voronoiů, jak je znázorněno na obrázku 6.



Obrázek 6: Síť voronoiů při dělení indexovaného prostoru [4]

Každou oblast VC_i plně pokrývá vlastní mřížka G_i , jejíž střed je v referenčním bodu RP_i oblasti VC_i . Na obrázku 7 jsou znázorněny mřížky G_1 a G_2 se středem v referenčních bodech RP_1 a RP_2 pokrývající oblasti VC_1 a VC_2 .



Obrázek 7: Indexování objektů pomocí mřížek [4]

Podobně jako u indexování času předpokládejme objekt $o(\vec{x}, \vec{v})$ v indexovaném prostoru. Dále předpokládejme nějaký referenční bod RP_i , kde i je identifikátorem tohoto referenčního bodu a k němuž je objekt o vzdálenostně nejbliže, a jemu odpovídající mřížku G_i . Pro každou mřížku je v B^+ stromu rezervováno určité množství souvislých klíčů, které je určeno hodnotou $SPAN_{space}$. Pro určení identifikátoru buňky, do které patří hodnota \vec{x}' , slouží funkce $cid(\vec{x}', G_i)$. Prostorová složka KEY_{space} je určena následovně:

$$KEY_{space}(o) = i \times SPAN_{space} + cid(\vec{x}', G_i)$$

Buňkám jsou identifikátory přiřazeny pomocí křivek v prostoru. ST^2B strom pro tento účel využívá Hilbertovy křivky pro zajištění dobré lokality.

Mějme objekty o_1 , o_2 a o_3 , jak je znázorněno na obrázku 7. Je zřejmé, že objekt o_1 je pokryt pouze mřížkou G_1 , navíc jeho nejbližším referenčním bodem je bod RP_1 , a proto je indexován mřížkou G_1 . Podobně i objekt o_2 je pokryt pro změnu pouze mřížkou G_2 a jeho nejbližším referenčním bodem je bod RP_2 , a proto je indexován mřížkou G_2 . Problém však nastává u objektu o_3 , který je pokryt jak mřížkou G_1 , tak i mřížkou G_2 . Objekt o_3 je indexován mřížkou G_1 , protože jeho nejbližším referenčním bodem je bod RP_1 . Jak je i z obrázku patrné, i když se mřížky překrývají, buňky voronoiů s referenčními body jsou disjunktní, a proto je u každého objektu možné určit, do které mřížky patří.

6.3 Klíč indexu

Ve výsledku je objekt o v ST^2B stromu indexován pomocí klíče KEY_{ST^2} , který se určí pomocí následujícího vzorce:

$$KEY_{ST^2} = KEY_{time} \times SPAN_{time} + KEY_{space}$$

kde KEY_{time} je klíč času popsáný v kapitole 6.1, podobně také KEY_{space} je klíč k prostorovým datům, jak je popsán v kapitole 6.2. $SPAN_{time}$ je velikost rezervovaného rozsahu klíčů pro každý podstrom.

Jak je vidět na obrázku Obrázek 5: Princip ST^2B stromu5, prostor prostorových klíčů je v daný čas poloviční díky využití dvou podstromů a , který z těchto dvou podstromů je vybrán, záleží na čase, ve kterém změna probíhá. Prostor prostorových klíčů je dále rozdělen na n mřížek. Na nejnižší úrovni jsou objekty v každé mřížce seřazeny vzestupně dle hodnoty identifikátoru buňky, do které patří.

7 HR strom

HR strom [6] (Hash R strom) je založen na principu R stromu, ale na rozdíl od R stromu jsou v jednotlivých uzlech každé úrovně tohoto stromu minimální ohraničující obdélníky, do nichž se minimální ohraničující obdélníky z nižších úrovní přiřazují na základě hashované adresy (pomocí algoritmu Hash address sorting).

7.1 Hashovací tabulky HR stromu

U hashovacích algoritmů je hash jednorozměrným obrazem daného záznamu a zároveň jeho klíčem. U prostorových a časoprostorových databází je problém ve větším počtu jejich dimenzí, což by vedlo na vícerozměrné adresování. Vícerozměrnou adresu však lze transformovat na adresu jedné dimenze.

Vstupem hashovací funkce jsou souřadnice levého horního a pravého spodního rohu minimálního ohraničujícího obdélníka, obrazem tohoto vstupu je pak střed minimálního ohraničujícího obdélníka. Nejdříve se počítá střed vnějšího ohraničujícího obdélníku, poté se počítá střed minimálního ohraničujícího obdélníku, který se nachází uvnitř. Například v dvourozměrném prostoru bude postup následující:

1. Začíná se u primárních minimálních ohraničujících obdélníků nejvyšší úrovně, které neleží uvnitř jiného většího ohraničujícího obdélníku. Samotné primární ohraničující obdélníky obsahují další dílčí minimální ohraničující obdélníky. U všech těchto primárních ohraničujících obdélníků se vypočítají souřadnice jejich středů. Tyto souřadnice středů se průběžně řadí podle x-ové hodnoty souřadnice středů a ukládají se do hashovací tabulky první úrovně. Každý záznam v hashovací tabulce má strukturu `<mbr_id, coordinates, ptr>`, kde `mbr_id` je identifikátorem daného minimálního ohraničujícího obdélníku, `coordinates` definují souřadnice středu tohoto obdélníku a `ptr` je ukazatel na další úroveň hashovací tabulky, jež obsahuje dílčí minimální ohraničující obdélníky.
2. Vypočítají se souřadnice středů pro všechny minimální ohraničující obdélníky druhé úrovně a postupně se seřadí podle x-ové hodnoty souřadnice středů, podobně jako u ohraničujících obdélníků nejvyšší úrovně. Souřadnice středů se uloží do hashovací tabulky druhé úrovně a adresa této tabulky druhé úrovně je zapsána do hodnoty pole `ptr` u odpovídajícího ohraničujícího obdélníku v hashovací tabulce první úrovně. Záznamy v hashovací tabulce druhé úrovně mají stejnou strukturu jako záznamy v hashovací tabulce první úrovně.

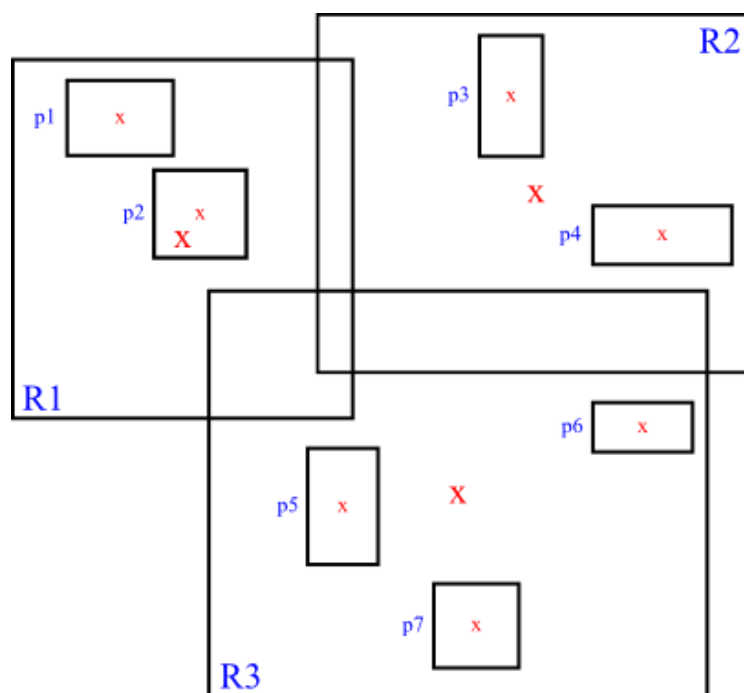
3. Minimální ohraničující obdélníky samotných datových objektů jsou uloženy v hashovací tabulce nejnižší úrovně a mají strukturu $\langle id, coordinates \rangle$, kde je význam jednotlivých položek shodný s položkami uvedenými v bodě 1.

7.2 Sestavování HR stromu

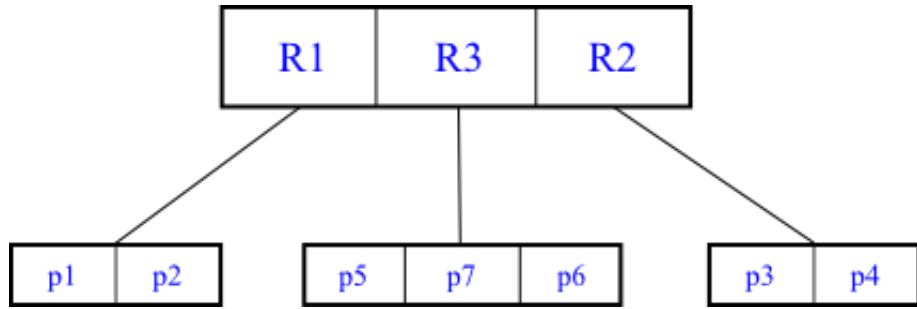
Při sestavování HR stromu se postupuje obdobně jako u R stromu, pouze s tím rozdílem, že u HR stromu jsou minimální ohraničující obdélníky v každém uzlu seřazeny podle x-ových hodnot souřadnic středů. Jednotlivé uzly stromu jsou vytvořeny dle těchto pravidel:

1. Kořenový uzel je vytvořen ze všech minimálních ohraničujících obdélníků, jejichž souřadnice středů jsou zaznamenány v hashovací tabulce první úrovně. Potomci kořenového uzlu jsou seřazeni dle jejich x-ové hodnoty souřadnic středů.
2. Nelistové uzly dalších úrovní jsou vytvořeny z minimálních ohraničujících obdélníků, jejichž souřadnice středů jsou zaznamenány v hashovací tabulce druhé úrovně. V každém nelistovém uzlu je minimální ohraničující obdélník, který ohraničuje všechny obdélníky potomků. Potomci nelistových uzlů jsou rovněž seřazeni dle jejich x-ové hodnoty souřadnic středů.
3. Listové uzly obsahují strukturu $\langle oid, mbr \rangle$, kde oid značí daný prostorový objekt a mbr je jeho nejmenší obdélníková obálka.

Hashovací tabulky propojují rodičovské uzly s uzly jejich potomků. Na obrázku 9 je ukázán princip indexování minimálních ohraničujících obdélníků z obrázku 8, kdy jsou data v jednotlivých uzlech seřazena podle x-ové hodnoty středů jejich minimálních ohraničujících obdélníků.



Obrázek 8: Minimální ohraničující obdélníky objektů a jejich středy (podle [6])



Obrázek 9: HR strom pro minimální ohraničující obdélníky z obrázku 8 (podle [6])

8 Recursively Partitioned Trajectory Index (RPTI)

Index (RPTI)

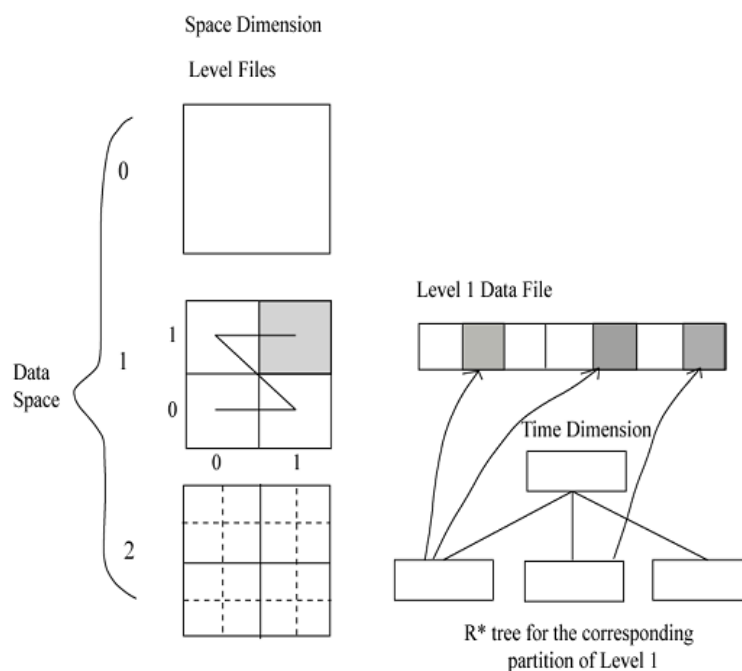
Rekurzivně rozdělovaný index pro trajektorie (RPTI) [2] využívá myšlenky, kdy po celou dobu existence pohybujícího se objektu se mění rozměr prostorových dat velmi pomalu na rozdíl od času, který plyne konstantně. RPTI zvláště indexuje prostorovou doménu a zvláště doménu časovou. Pro indexování prostorových dat využívá metodu rekurzivního rozdělování prostoru, časová data indexuje pomocí R^* stromu.

8.1 Indexování prostorových dat

Indexování prostorových dat v RPTI je založeno na statickém prostoru. Prostor není rozdělován do překrývajících se či nepřekrývajících se oblastí s různou velikostí jako v případě R stromů a jejich modifikací, ale je rozdělován hierarchicky do oblastí se stejnou velikostí. Pokud označíme počet úrovní stromu jako K a úrovně budou značeny k ($k = 0$ pro 1. úroveň, ..., $k = K - 1$ pro úroveň poslední), pak v každé úrovni stromu bude 4^k oblastí. Oblasti pro úroveň k jsou vytvořeny pomocí $2^k - 1$ rovnoměrně rozložených čar v každé dimenzi prostoru.

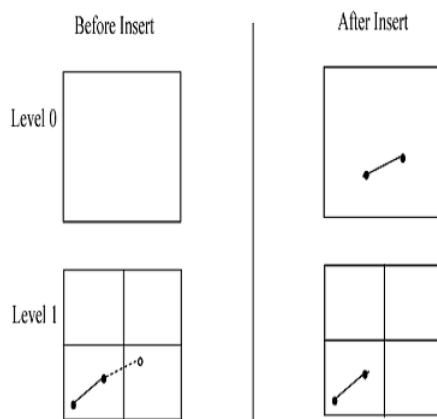
Každá prostorová oblast má svůj vlastní R^* strom pro indexování časové domény. Podobně jako u počtu prostorových oblastí má každá úroveň 4^k R^* stromů v každé úrovni.

Struktura rekurzivně rozdělovaného indexu pro trajektorie je zobrazena na obrázku 10.



Obrázek 10: Struktura rekurzivně rozdělovaného indexu pro trajektorie (RPTI) [2]

U prostorových dat je každý úsek trajektorie uložen jako n -tice do úrovněvých souborů. Do které úrovně úsek patří lze zjistit jednoduchým pravidlem: V případě, že daný úsek trajektorie prochází dělicí čarou k -té úrovně, pak tento úsek trajektorie patří do úrovně $k - 1$ a bude uložen do příslušného úrovněvého souboru. Vždy je snahou umístit daný úsek trajektorie do co nejmenší oblasti, jak jen to jde. Umístění úseků trajektorií do oblastí v různých úrovních je zobrazeno na obrázku 11. Každý úrovněvý soubor tak obsahuje pouze úseky trajektorií patřící do jedné jediné oblasti.



Obrázek 11: Umístění úseků trajektorií do oblastí v různých úrovních [2]

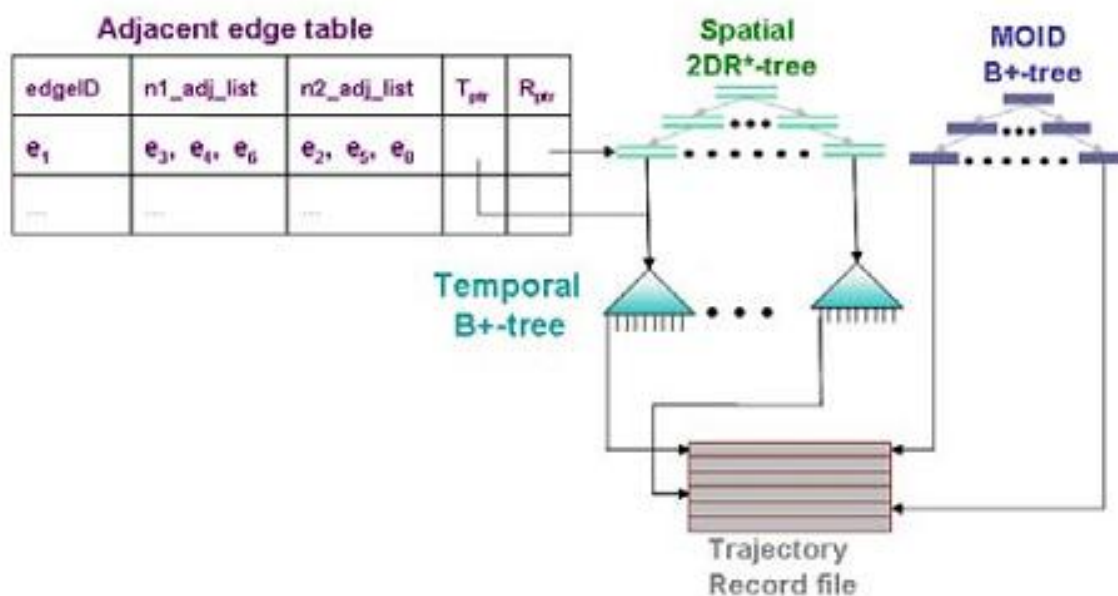
8.2 Indexování času

Z hlediska časové dimenze trajektorií je v každé oblasti použit R^* strom pro indexování datové stránky, která pokrývá určitý časový interval všech úseků uložených v této stránce. Ukázka indexu časové dimenze trajektorie pro jednu konkrétní oblast je znázorněn na obrázku 10. Není problém indexovat časovou dimenzi místo R^* stromu jinou indexační strukturou. Pokud je v každé datové stránce uložen pouze jediný záznam s úsekem trajektorie, je vyhledávání v R^* stromu efektivnější. Naopak při vkládání nebo mazání záznamů z datové stránky se může změnit časový interval, pro který tato datová stránka záznamy uchovává, a v tomto případě je pak nutné upravit i odpovídající R^* strom.

9 TMN strom

TMN strom [3] (*Trajectory of Moving objects on Network Tree*) je indexační struktura pro trajektorie pohybujících se objektů v prostorových sítích. Tento indexační algoritmus poskytuje efektivní algoritmy zpracovávající dotazy zaměřené na trajektorie. Velkými výhodami tohoto přístupu je schopnost dotazovat rozsahy hodnot a také schopnost dotazovat se na podobnost trajektorií. Tento přístup indexuje zvlášť prostorovou doménu a zvlášť časovou doménu pohybujícího se objektu a zachovává jeho kompletní trajektorii.

Celková koncepce TMN-stromu je znázorněna na obrázku 12. TMN-strom se sestává z pěti součástí, v nichž figurují tři stromy. První strom indexuje prostorové sítě, které uchovává v prostorovém $2DR^*$ stromu. Druhý strom slouží pro indexování času pohybujících se objektů a je reprezentován B^+ stromem. Třetí strom má také podobu B^+ stromu a slouží pro indexování identifikátorů pohybujících se objektů (*MOID*). Zbývající části TMN-stromu tvoří tabulka přilehlých hran a soubor se záznamy trajektorií.



Obrázek 12: Architektura TMN stromu [3]

9.1 Indexování prostorových dat

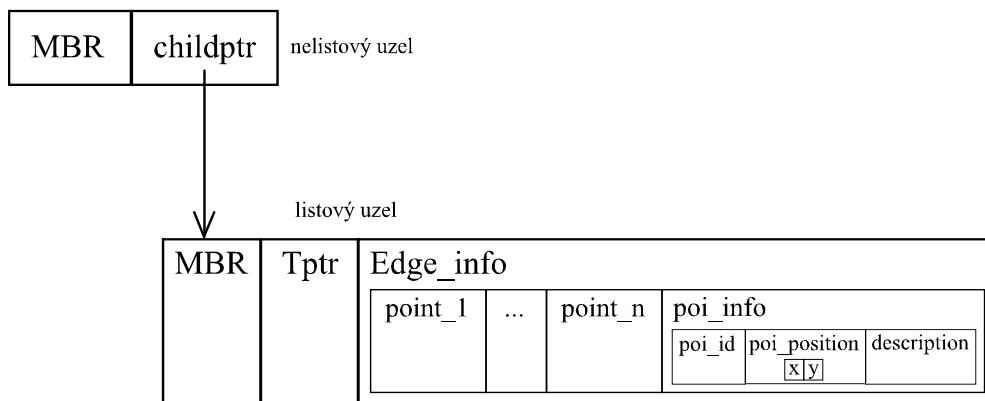
Prostorová síť v podobě hran je ukládána do $2DR^*$ stromu. Nelistové uzly tohoto stromu odkazují na své potomky pomocí struktury ve tvaru $\langle MBR, \text{childptr} \rangle$, kde *MBR* je minimální ohraničující obdélník (*minimum bounding rectangle*) obsahující prostor všech potomků synovského uzlu a *childptr* je ukazatel na tento synovský uzal. Listové uzly prostorového $2DR^*$ stromu jsou navázány na časové B^+ stromy, jak je znázorněno na obrázku 12.

Samotné listové uzly obsahují strukturu ve formátu $\langle \text{MBR}, \text{Tptr}, \text{Edge_info} \rangle$, kde MBR je obdélník ohraničující konkrétní hranu, Tptr je ukazatel na časový B⁺ strom a Edge_info je objekt, ve kterém jsou uchovávány informace jak o hraně, tak i o důležitých bodech (*POIs = Points Of Interests*) této hrany.

Objekt Edge_info má formát $\langle \{ \text{point}_1, \text{point}_2, \dots, \text{point}_n \}, \text{poi_info} \rangle$, který se skládá z interpolačních bodů $\{ \text{point}_1, \dots, \text{point}_n \}$ a informací o hraně na úrovni zajímavých bodů poi_info.

Samotný objekt poi_info má strukturu $\langle \text{poi_id}, \text{poi_position}, \text{description} \rangle$, v níž poi_id jednoznačně identifikuje daný zajímavý bod, poi_position udává pozici zajímavého bodu v prostorové síti pomocí souřadnic (x, y) a description uchovává informace k danému zajímavému bodu.

Celková struktura tohoto 2DR* stromu je zobrazena na obrázku 13.



Obrázek 13: Struktura 2DR* stromu

9.2 Indexování času

Časový B⁺ strom indexuje časovou dimenzi pohybujících se objektů, časová dimenze je reprezentována časovými razítky. Pomocí listového uzlu časového B⁺ stromu se dostaneme ke konkrétnímu segmentu trajektorie do souboru se záznamy trajektorií, jak je naznačeno na obrázku 12.

Listový uzel stromu obsahuje strukturu ve formátu $\langle \text{time}, \text{offset}, \text{trajectory_ptr} \rangle$, kde time je časové razítko nějakého okamžiku pro pohybující se objekt, offset určuje polohu jeho segmentu v trajektorii pohybu tohoto objektu a trajectory_ptr je zmíněný ukazatel do souboru se záznamy trajektorií.

Výhodou umístění času do B⁺ stromu je snadný přístup k požadovanému intervalu času, kdy ve stromu vyhledáme počáteční čas intervalu a následně sekvenčně procházíme listové uzly stromu, dokud nenarazíme na požadovaný koncový čas intervalu.

9.3 Indexování identifikátorů pohybujících se objektů

MOID (*Moving Object ID*) B^+ strom je strom indexující identifikátory pohybujících se objektů, pomocí nichž jsou odkazovány jednotlivé trajektorie v souboru se záznamy trajektorií.

Listový uzel MOID B^+ stromu obsahuje strukturu $\langle \text{MOID}, \text{trajectory_ptr} \rangle$, v níž MOID je identifikátorem dané trajektorie a trajectory_ptr je vlastní ukazatel na tuto trajektorii do souboru se záznamy trajektorií.

9.4 Tabulka přilehlých hran

Tabulka přilehlých hran slouží k lepšímu sledování trajektorií pohybujících se objektů. V tabulce se vytváří seznam uzlů udržující identifikátory přilehlých hran v prostorové síti. Každý záznam také uchovává ukazatel do prostorového R^* stromu a na časový B^+ strom.

Výhodou tohoto přístupu je, že pro vyhledání přilehlých hran není nutné prohledávat prostorové R^* stromy, ale postačí pouze jediný přístup na disk do tabulky přilehlých hran. Další výhodou je, že tabulka přilehlých hran může být uchovávána v hlavní paměti, což přináší další zlepšení z hlediska výkonnosti, protože se při přístupu do této tabulky eliminuje přístup k disku a přístup k paměti je rychlejší.

Záznamy v tabulce přilehlých tvar jsou uchovávány ve formě $\langle \text{edge_id}, \text{n1_adjlist_ptr}, \text{n2_adjlist_ptr}, \text{Tptr}, \text{Rptr} \rangle$. Položka edge_id identifikuje hranu, n1_adjlist_ptr a n2_adjlist_ptr jsou ukazatele na seznamy počátečních (n1_adjlist_ptr) a koncových (n2_adjlist_ptr) uzlů, Tptr je ukazatel na příslušný časový B^+ strom a Rptr ukazuje na listový uzel prostorového R^* stromu.

9.5 Soubor se záznamy trajektorií

Soubor se záznamy trajektorií slouží k uchovávání kompletní trajektorie pohybujících se objektů a dalším informacím, které s těmito trajektoriemi souvisí. Každý záznam trajektorie umožňuje uchovat pouze určitý počet úseků trajektorie.

Pokud se tedy stane, že nějaká trajektorie svým počtem úseků překročí tento maximální počet, pak se musí alokovat nový záznam a do něj se vloží zbývající úseky trajektorie. Původní zaplněný záznam s úseky trajektorie se musí propojit s tímto nově vytvořeným záznamem.

Struktura záznamu má tvar $\langle \text{MOID}, \text{traj_num}, \text{next_ptr}, \text{prev_ptr}, \{\text{segment}_1, \dots, \text{segment}_n\} \rangle$, v níž je MOID identifikátorem pohybujícího se objektu, traj_num udává počet úseků, next_ptr ukazuje na následující záznam a prev_ptr naopak

na předchozí záznam, {segment_1, ..., segment_n} je množina úseků trajektorie či její části, v případě většího počtu úseků.

Každý úsek je uložen ve formě <t, edge_id, x, y, direction>, v němž t je časové razítko, kdy se pohybující se objekt pohyboval na daném úseku, edge_id je identifikátor hrany, kudy procházel, x a y jsou souřadnice pozice pohybujícího se objektu a direction určuje směr pohybu po daném úseku, pomocí něhož se dá předpovídat příští pozice na trajektorii pohybujícího se objektu.

Celková struktura záznamů v souboru se záznamy trajektorií je zobrazena na obrázku 14.

| | | | | | | | | | | |
|-------------------------|----------|-------------------------|----------|--|-----------|-----|-----------|-------------------------|--|-------------------------|
| MOID | traj_num | next_ptr | prev_ptr | <table border="1"> <tr> <td>segment_1</td> <td>...</td> <td>segment_n</td> </tr> <tr> <td>t edge_id x y direction</td> <td></td> <td>t edge_id x y direction</td> </tr> </table> | segment_1 | ... | segment_n | t edge_id x y direction | | t edge_id x y direction |
| segment_1 | ... | segment_n | | | | | | | | |
| t edge_id x y direction | | t edge_id x y direction | | | | | | | | |

Obrázek 14: Struktura záznamů v souboru se záznamy trajektorií

10 Závěr

Tato práce si kladla za cíl hlouběji se zaměřit na různé principy, jakými se indexují časoprostorová data. V práci byly zmíněny novější přístupy k indexování časoprostorových dat, avšak indexy zmiňované v této práci nejsou vyčerpávajícím přehledem existujících řešení, protože jistě existují také další.

Oblast indexování časoprostorových dat je relativně mladou oblastí, a tak se dá očekávat, že se v této oblasti objeví další nové přístupy k efektivnímu indexování pohybujících se objektů, respektive jejich trajektorií.

Literatura

- [1] Mahdi Abdelguerfi, Julie Givaudan, Kevin Shaw, and Roy Ladner. The 2-3TR-tree, A Trajectory-Oriented Index Structure for Fully Evolving Valid-Time Spatio-Temporal Datasets. In *Proceedings of the 10th ACM international symposium on Advances in geographic information systems, GIS '02*, pages 29–34, New York, NY, USA, 2002. ACM.
- [2] Elizabeth Antoine, Kotagiri Ramamohanarao, Jie Shao, and Rui Zhang. Recursive Partitioning Method for Trajectory Indexing. In *Proceedings of the Twenty-First Australasian Conference on Database Technologies - Volume 104, ADC '10*, pages 37–46, Darlinghurst, Australia, Australia, 2010. Australian Computer Society, Inc.
- [3] Jae-Woo Chang, Myoung-Seon Song, and Jung-Ho Um. TMN-tree: New Trajectory Index Structure for Moving Objects in Spatial Networks. In *Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on*, pages 1633 –1638, 29 2010-july 1 2010.
- [4] Su Chen, Beng Chin Ooi, Kian-Lee Tan, and Mario A. Nascimento. ST²B-tree: A Self-Tunable Spatio-Temporal B⁺-tree Index for Moving Objects. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data, SIGMOD '08*, pages 29–42, New York, NY, USA, 2008. ACM.
- [5] Ying Fang, Jiaheng Cao, Yuwei Peng, and Liwei Wang. Indexing the Past, Present and Future Positions of Moving Objects on Fixed Networks. In *Computer Science and Software Engineering, 2008 International Conference on*, volume 4, pages 524 –527, december 2008.
- [6] Guobin Li and Jine Tang. A new HR-tree index based on hash address. In *Signal Processing Systems (ICSPS), 2010 2nd International Conference on*, volume 3, pages V3–35 –V3–38, july 2010.