

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERINIŲ TINKLŲ KATEDRA

Andrius Pukėnas

Vietos duomenų istorijos atkūrimo sistema

Magistro darbas

Darbo vadovas

prof. dr. R. Plėštys

Kaunas, 2008

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERINIŲ TINKLŲ KATEDRA

Andrius Pukėnas

Vietos duomenų istorijos atkūrimo sistema

Magistro darbas

Recenzentas

doc. dr. P. Kanapeckas

2008-01-10

Vadovas

prof. dr. R. Plėštys

2008-01-14

Atliko

IFM-2/4 gr. stud.
Andrius Pukėnas

2008-01-14

Kaunas, 2008

Location Data History Restoration System

Summary

The usage of moving objects location information becomes very popular nowadays. This location information can be used in many places: for emergency purposes, tracking cars with cargos, animal tracking, tourism, etc. The moving object's positions can be detected using location techniques, such as GPS, GSM, etc. When detected moving object's position is kept in Moving Object Databases (MODB). When object's position is detected constantly we get a lot of positions, which represent the movement of particular object. We can call this movement a trajectory. The main problem is that these trajectories are too big to keep them in database, to update them or to make a search among them. That's why there is a need for effective trajectory compression methods in order to make trajectories smaller without losing the most important object's movement information. Different location techniques generate different types of trajectories, so each type of trajectory needs a different approach how to apply compression technique to it. The trajectory compression can be applied while generating trajectory or it can be done when the generation of trajectory is finished. There are presented different trajectory compression methods for different location techniques (GPS, GSM) in this work. These methods are incorporated into location data history restoration system in order to apply them for location data compression. In this work statistical methods for evaluation of location data compression methods effectiveness are presented. Statistical results for these methods prove that methods are reliable and anyone can get the same results when using these methods.

Turinys

1.	Įvadas.....	5
2.	Vietos duomenų sistema ir jos analizė	7
2.1.	Sistemos vartotojai	8
2.1.1.	Veiklos proceso modelis	8
2.1.2.	Veiklos panaudojimo atvejų modelis.....	9
2.2.	Vietos nustatymo technologijos	10
2.3.	Judančių objektų duomenų bazė	11
2.3.1.	Trajektorijų vaizdavimas.....	11
2.3.2.	Trajektorijų suspaudimo metodų analizė	12
2.3.3.	Esamų metodų trūkumai	22
2.4.	Kompiuterizuojami panaudojimo atvejai.....	22
3.	Beveilių mobiliųjų technologijų vietos duomenų istorijos atkūrimas ir jam skirta sistema.....	24
3.1.	GSM vietos duomenų suspaudimo metodas	24
3.2.	GPS vietos duomenų suspaudimo metodai.....	31
3.2.1.	Dabarties didelio tikslumo trajektorijų (GPS) suspaudimo metodas.....	32
3.2.2.	Praeities didelio tikslumo trajektorijų (GPS) suspaudimo metodas	36
3.3.	Vietos duomenų istorijos atkūrimo sistemos modelis	38
3.3.1.	Duomenų bazės koncepcinis modelis	39
3.3.2.	Sistemos loginė architektūra	39
3.3.3.	Objektų būsenų ir veiklos diagramos.....	40
3.3.4.	Vietos duomenų suspaudimo metodų algoritmų realizacijos principai	41
4.	Vietos duomenų istorijos atkūrimo efektyvumo tyrimas	43
4.1.	Duomenų su pastoviu laiko intervalu suspaudimo efektyvumo tyrimas	43
4.2.	Duomenų su kintančiu laiko intervalu suspaudimo efektyvumo tyrimas.....	45
4.3.	Dabarties didelio tikslumo trajektorijų (GPS) suspaudimo metodo algoritmo eksperimentinis įvertinimas.....	47
4.4.	Praeities didelio tikslumo trajektorijų (GPS) suspaudimo metodo algoritmo eksperimentinis įvertinimas.....	52
4.5.	Dabarties ir praeities didelio tikslumo trajektorijų (GPS) metodų algoritmų efektyvumo palyginimas.....	54
4.6.	Praeities mažo tikslumo trajektorijų (GSM) suspaudimo metodo įvertinimas	55
5.	Išvados	58
6.	Literatūros sąrašas.....	59
7.	Priedai.....	61

1. Įvadas

Su abonentų pozicija geografinėje erdvėje susijusi informacija naudojama labai įvairiose srityse: greitajai pagalbai teikti (112 paslauga), teisėsaugoje, krovinių transportavime (stebima kur šiuo metu yra transportuojami kroviniai), turizme, eismo sąlygų stebėjimui, artimiausių ligoninių, restoranų, lankytinų vietų paieškai, pramogoms, mobiliojo tinklo struktūros planavimui (nustačius kur yra didesnis abonentų susibūrimas, pastatoma daugiau bazinių stočių).

Tokia su objektų pozicijomis susijusi informacija kaupiama Judančių Objektų duomenų bazėse (JODB), kuri yra judančių objektų sekimo sistemos dalis. Judančiu objektu laikomas toks objektas, kuris bėgant laikui keičia savo geografinę ir laikinę poziciją trimatėje x,y,t erdvėje. Anksčiau vietos informacijai rinkti buvo naudojama tik GPS technologija. GPS technologijos pagrindinis trūkumas tas, kad ji sunkiai veikia miesto tipo vietovėse ir praktiškai neveikia požeminiuose ir antžeminiuose statiniuose. Tačiau vystantis įvairiomis mobiliosiomis ir bevielėms technologijoms, daugėjant jų naudojimo vietų, atsiranda galimybė tokius duomenis rinkti pasinaudojant ne tik GPS, bet ir GSM, WLAN, Bluetooth ir kitomis bevielėmis technologijomis. Todėl atsiranda galimybė GPS pagalba gautus vietos duomenis patikslinti, papildyti arba netgi, jei nereikalaujama didelio tikslumo, judančių objektų vietą nustatyti be GPS panaudojimo.

Stebint judančius objektus „visur ir visada“ kiekvienam objektui susidaro baigtinis pozicijų skaičius per tam tikrą laiko tarpą, kuris vadinamas to objekto judėjimo trajektorija. Trajektorijos duomenų kiekis priklauso nuo vietos nustatymo dažnumo, taip pat ir nuo naudojamos technologijos. Tačiau kai vienu metu stebima labai daug judančių objektų, susidaro labai dideli duomenų kiekiai, kurių apdorojimas (paieška, atnaujinimas, trynimas) yra neefektyvus. Trajektorijų informacija turi savo savybes, kurios dar priklauso ir nuo mobiliosios technologijos, kuria jos yra gautos. Todėl reikalingi trajektorijų suspaudimo metodai, įvertinantys esminius mobiliųjų technologijų vietos nustatymo aspektus: tikslumą, neapibrėžtumą, vietos duomenų nuskaitymo intervalų kitimą ir kt.

Duomenų suspaudimas gali būti atliekamas realiu arba nerealiu (praeities) režimais. Vietos duomenis spausti galima 2 būdais: prarandant arba neprarandant vietos informacijos tikslumo ją atkūrus į pradinę būseną. Atstatomų vietos duomenų tikslumas kartais gali būti specialiai „gadinamas“, tokiu būdu reguliuojant, kokio tikslumo informaciją vartotojui pateikti. Tokiu atveju naudojamas trajektorijos duomenų suspaudimo koeficientas, kuris apsprendžia atstatomų duomenų tikslumo lygį. Duomenų suspaudimo ir atstatymo procesai būtinai turi eiti vienas po kito

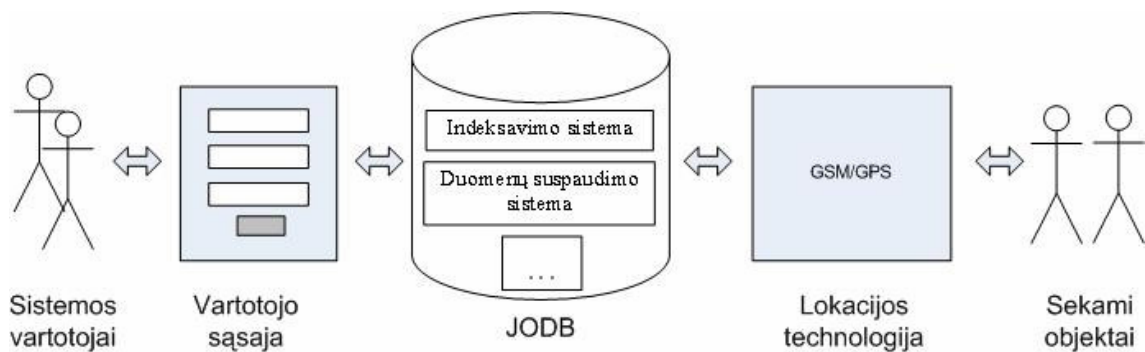
ir negali būti vykdomi atvirkščia tvarka, todėl atstačius suspaustus vietos duomenis jie visada bus praeities, t.y. istorijos duomenys. Tokia sistema vadinama Vietos duomenų istorijos atkūrimo sistema, kurios pagrindas yra nuo mobiliosios technologijos priklausomi vietos duomenų suspaudimo metodai. Ši sistema integruojama į judančių objektų sekimo sistemą kaip sudėtinė judančių objektų duomenų bazės (JODB) dalis.

Kuriant sistemą buvo išanalizuoti žinomi vietos duomenų suspaudimo metodai, nustatyti jų trūkumai, sukurti nauji vietos duomenų suspaudimo metodai, tinkami naujoms bevielėms vietos nustatymo technologijoms. Išanalizavus jau esančius duomenų suspaudimo metodus, šie metodai buvo pagerinti ir pritaikyti tam, kad jie galėtų būti realiai naudojami kuriamoje sistemoje. Sukurta prototipinė duomenų suspaudimo metodus naudojanti sistema, metodai įvertinti eksperimentiškai šios sistemos pagalba. Vartotojas į sistemą paduoda įvairius duomenų suspaudimo metodus reikalingus parametrus (dažniausiai tai yra leistinos paklaidos). Vartotojas gali pasirinkti kokio tipo duomenų suspaudimo metodais sistemoje apdoroti savo duomenis. Šiuos duomenis į sistemą galima importuoti, o apdorojus juos duomenų suspaudimo metodais – eksportuoti. Tačiau eksportuoti suspausti vietos duomenys gali būti ne vienintelis sistemos gražinamas rezultatas, kadangi sistemoje yra įdiegti duomenų suspaudimo metodų efektyvumą įvertinantys algoritmai, leidžiantys analizuoti sistemoje integruotų duomenų suspaudimo metodų efektyvumą.

2. Vietos duomenų sistema ir jos analizė

Pagrindinis darbo tikslas yra sukurti efektyvius vietos duomenų suspaudimo metodus. Tačiau šis tikslas yra ne vienintelis – kitas tikslas yra šių sukurtų duomenų suspaudimo metodų apjungimas į vieną bendrą visumą. Tai gali būti padaryta realizuotos sistemos pagalba. Šiame darbe kuriama sistema neturi būti programinis produktas, tačiau jos pagalba turi būti vykdomas duomenų suspaudimo problemos sprendimas. Ši sistema (programa) įgyvendina realybėje sukurtus duomenų suspaudimo modelius ir taip sudaro galimybes juos patikrinti ir įvertinti. Tik tokio pobūdžio sistemos pagalba gali būti atliktas pats tyrimas (duomenų suspaudimo metodų efektyvumo), nes rankiniu būdu to padaryti neįmanoma arba per daug sudėtinga, gauti rezultatai nėra tikslūs. Kitaip tariant sistema turi būti sukurta tuo tikslu, kad daromos prielaidos apie duomenų suspaudimo metodų efektyvumą būtų įrodytos. Į sistemą yra paduodami įvairūs parametrai (skiriamos atminties dydis, laiko periodas) generatoriui, kuris generuoja tariamas judančių objektų pozicijas konkrečiais laiko momentais (naudodamas užduotą laiko periodą). Naudojantis sistema yra nurodomos trajektorijų suspaudimo paklaidos (iš vartotojo pusės), sistemos grąžinamas rezultatas ir nauda yra susiję ne su tiesioginiu jos panaudojimu, tačiau gaunami statistiniai metodų įverčiai turi parodyti metodų efektyvumą skirtingoms vietos nustatymo technologijoms (GSM, GPS).

Darbe nagrinėjama vietos duomenų sistema su joje vykstančiais procesais susideda iš 5 pagrindinių komponentų, kurios suprastintas bendras modelis pateiktas 1 pav.



1 pav. Vietos duomenų sistema

Modelyje yra tokie komponentai:

- Sistemos vartotojai – aktoriai, kurie gali naudotis sistemos teikiama informacija, ją apdoroti;

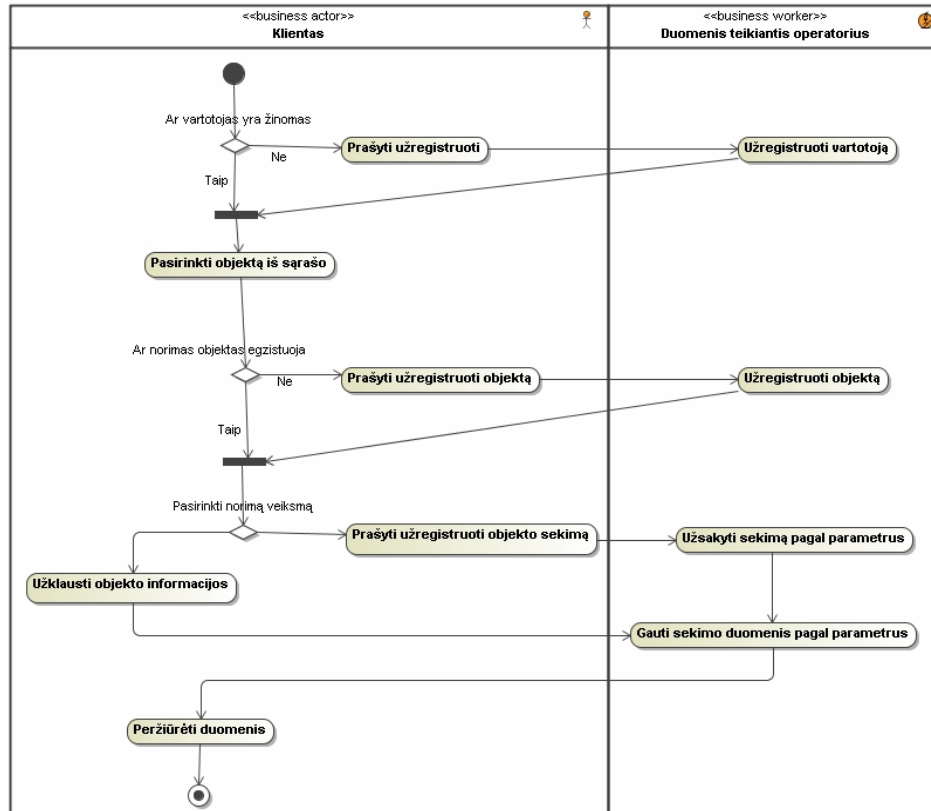
- Vartotojo sąsaja – komponentas, leidžiantis vartotojo naudojamąsi JODB (judančių objektų duomenų bazės) resursais;
- JODB – vietos duomenims saugoti naudojama duomenų bazė, besinaudojanti specialiai erdvėje ir laike judantiems objektams pritaikytais indeksavimo ir duomenų suspaudimo metodais. Kadangi šių metodų yra daug jie skiriami į atskiras indeksavimo ir duomenų suspaudimo ir atstatymo sistemas, kurios naudojamos darbui su duomenimis JODB.
- Lokacijos technologija – naudojama gauti objektų vietos duomenims realiu laiku. Kaip lokacijos technologijos dažniausiai naudojamos plačiai paplitusios specializuotos ir bendros bevielės technologijos: GPS, GSM, WLAN ir kt.
- Sekami objektai – sistemos objektai, apie kuriuos yra kaupiama su jų judėjimu susijusi informacija. Tai gali būti ne tik žmonės, bet ir transporto priemonės, gyvūnai.

2.1. Sistemos vartotojai

Sistemos vartotojai yra analizuojami veiklos proceso ir veiklos panaudojimo atveju modeliuose.

2.1.1. Veiklos proceso modelis

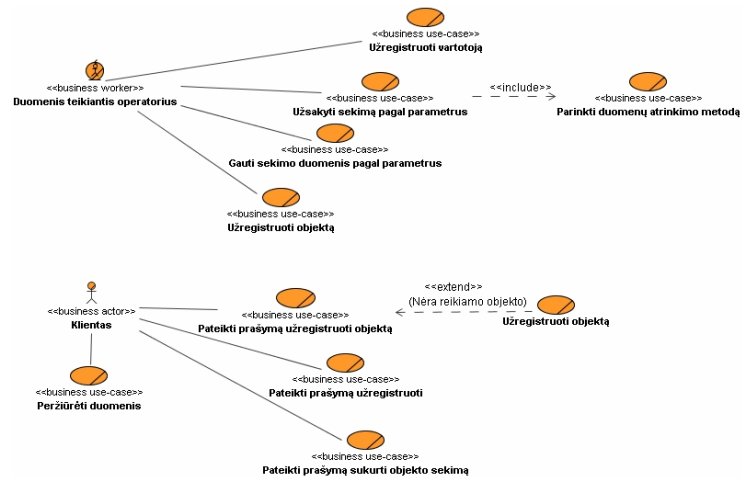
Pagrindinio veiklos proceso (2 pav.) metu yra atliekamas duomenų apie judančius objektus generavimas ir jų peržiūra. Tam, kad vartotojas galėtų bendrauti su „Duomenis teikiančiu operatoriumi“, jis turi būti jam žinomas ir pas jį užregistruotas. Vėliau vartotojas gali pasirinkti sekamą objektą iš sąrašo arba paprašyti sukurti naują, jeigu ten nėra jam reikiamo objekto. Vartotojas dažniausiai atlieka du veiksmus – „Prašyti užregistruoti objekto sekimą“ arba „Užklausti objekto informacijos“ (kai jau duomenys apie judantį objektą jau yra surinkti DB ir reikia tik juos išgauti pagal norimus parametrus). Galiausiai klientas gali peržiūrėti jam pateiktus duomenis apie jį dominusį judantį objektą.



2 pav. Veiklos proceso modelis

2.1.2. Veiklos panaudojimo atvejų modelis

Daugelis veiklos panaudojimo atvejų (3 pav.) atitinka veiklos proceso modelyje esančius procesus. Tačiau šiuo atveju „Klientas“, prašydamas užregistruoti naują objektą („Pateikti prašymą užregistruoti objektą“), gali pasirinkti objekto tipą jeigu reikiamo tipo nėra. „Duomenis teikiantis administratorius“ gali jį sukurti („Užregistruoti objektą“). Taip pat „Duomenis teikiantis administratorius“ turi galimybę parinkti duomenų suspaudimo metodą panaudojimo atvejui „Užsakyti sekimą pagal parametrus“. Taigi sugeneruoti duomenys apie objekto judėjimą bus suspaudžiami pasirinktu metodu. Vėliau bus analizuojama koks duomenų suspaudimo metodas veikia geriausiai ir efektyviausiai tenkina „Kliento“ poreikius.



3 pav. Veiklos panaudojimo atvejų modelis

2.2. Vietos nustatymo technologijos

Skirtingos mobiliosios technologijos pasižymi skirtingomis vietos duomenų gavimo ypatybėmis. GPS - vietos nustatymo sistema, leidžia nustatyti objekto koordinates bet kurioje pasaulio vietoje (tik atviroje vietovėje). Sistemos pagrindas – IT technologijų sąveika su planeta gaubiančiu GPS palydovų tinklu. Duomenys apie objekto poziciją gaunami iš kelių palydovų (oid, x, y) pavidalu, kur oid - judančio objekto identifikatorius, x, y – geografinės objekto koordinatės. Tai yra tiksliausia pozicionavimo sistema. GSM – radijo celėmis paremtas tinklas, kuriomis pilnai padengiama geografinė teritorija. Tokio tinklo privalumai: išaugęs aptarnaujamų vartotojų skaičius, sumažėjęs energijos sunaudojimas, efektyviau išnaudojamas plotas priklausomai nuo jame vidutiniškai esančių abonentų skaičiaus. Būtent dėl celių GSM tinkle atsiranda abonto vietos nustatymo galimybė, remiantis technologijomis: Cell-ID, Cell-ID + Timing Advance, E-OTD, A-GPS ir pan. [1,2]. GSM tinkle dažniausiai yra naudojama Cell-ID metodika, kadangi yra paprasta ir ekonomiška, be to, nereikalauja kliento naudojamų ryšio priemonių ar tinklo infrastruktūros atnaujinimo, tačiau tikslumas priklauso nuo celės, kurioje yra abonentas, dydžio. Tačiau GSM vietos nustatymas ne toks tikslus kaip kad GPS, nes vietos duomenys turi pavidalą: (oid, x, y, r), kur r – celės, kurioje gali būti objektas, spindulys. WLAN (Wireless Local Area Network) reiškia bevielį LAN (vietinis tinklas), kuris apjungia kompiuterius, nenaudojant laidų. Jis gali būti panaudotas ne tik prieigai prie interneto, tačiau ir atlikti kliento vietos nustatymą. Iš pradžių įvairiuose patalpos taškuose išmatuojamas signalo stiprumas, o gauti duomenys surašomi į signalų stiprumo lentelę. Remiantis šia lentele galima rasti esamą objekto poziciją, kadangi prie

WLAN tinklo prisijungęs objektas skleidžia tam tikro stiprumo signalą. Toks vietos nustatymo būdas nėra toks tikslus, kaip kad GPS, kadangi gaunama zona, kurioje gali būti objektas [3], nėra taisyklinga, tačiau ši zona paprastai nėra labai didelė ir yra žinomoje vietoje, pvz., pastate, degalinėje ir pan. Skirtingai nuo kitų pozicionavimo sistemų, WLAN paskirtis nėra susijusi su kliento judėjimo trajektorijos generavimu. Jis gali tik papildyti arba patikslinti kitomis technologijomis gautą trajektorijos informaciją, kurios pagalba galima keisti objektui pateikiamų nuo buvimo vietos priklausomų paslaugų turinį. Čia koncentruojamasi į tam tikrus įvykius (pavyzdžiui, tam tikros zonos kirtimas, prisijungimas), kuriais remiantis klientui pateikiamos skirtingos nuo jo pozicijos priklausančios paslaugos [3], pvz., QoS, saugumas (resursais leidžiama naudotis tik apibrėžtoje zonoje), artimiausio įrenginio (pavyzdžiui, spausdintuvo) radimas.

2.3. Judančių objektų duomenų bazė

Tai duomenų bazė, skirta judančių objektų vietos duomenims saugoti. Svarbiausi jos komponentai – duomenų saugykla, indeksavimo ir duomenų suspaudimo sistemos. Šiame darbe analizuosime tik duomenų suspaudimą ir atstatymą. Darbe yra analizuojami ne tik duomenų suspaudimo metodai, tačiau juos išanalizavus atsiras galimybė sukurti jų gerąsias savybes įrodančią sistemą. Sistema turi šiuos metodus integruoti ir parodyti, kad juos galima naudoti realiomis sąlygomis, gaunant iš to naudą. Todėl šios sistemos kūrimo pagrindas yra duomenų suspaudimo metodų realaus panaudojimo tyrimas.

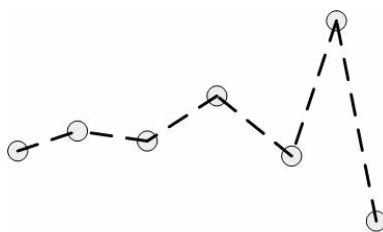
2.3.1. Trajektorijų vaizdavimas

Literatūroje judantys objektai analizuojami labai įvairiose srityse. Tai judančių objektų vietos informacijos talpinimas, indeksavimas, įvairių užklausų algoritmų ir duomenų suspaudimo metodų kūrimas. Vienas iš svarbiausių dalykų kalbant apie vietos duomenų suspaudimą - trajektorijų vaizdavimo modelio pasirinkimas. Daugelis įprastinių modelių, skirtų duomenų vaizdavimui, yra statiniai, tačiau judančio objekto trajektorija pastoviai kinta [4].

Jeigu vietos nustatymo technologija yra labai tiksli (GPS), tuomet judančio objekto trajektorija laike atvaizduojama kaip atkarpomis sujungtų taškų, kurie gaunami vietos nustatymo technologija apibrėžtais arba atsitiktiniais laiko intervalais, eilutė. Toks metodas vadinamas tiesiniu interpoliavimu [5]. Tikslu objekto pozicija tarp trajektorijos atkarpos galų yra neapibrėžta ir gali būti nusakoma su paklaida, kuri priklauso nuo objekto judėjimo krypties, greičio ir nuo viena po kitos nustatomų pozicijų dažnumo ir vietos nustatymo technologijos. Nuo technologijos

taip pat priklauso ir atkarpos galų vietos informacijos tikslumas. Judančių objektų duomenų bazėje objekto trajektorija gali būti atvaizduojama kaip vietos įrašų seka laike [6, 7], pvz., (id, t_i, x_i, y_i) , kur id – unikalus trajektorijos identifikatorius, t_i - laiko momentas, kuriuo objektas buvo pozicijoje (x_i, y_i) , $i \in (1, \infty)$. Informacijos atnaujinimo intervalai gali būti parinkti priklausomai nuo reikalavimų rezultatų tikslumui, o bėgant laikui atnaujinimo intervalo dydis tam pačiam objektui gali kisti.

Dažniausiai judančių objektų trajektorijos yra vaizduojamos tiesinės interpoliacijos būdu (4 pav.). Užfiksuotos objektų pozicijos yra apjungiamos tiesėmis, tokiu būdu yra atspindimas tikrasis objekto judėjimas.



4 pav. Trajektorijos vaizdavimas tiesinės interpoliacijos būdu

2.3.2. Trajektorijų suspaudimo metodų analizė

Trajektorijų suspaudimo metodai yra reikalingi tam, kad sumažintų trajektorijos taškų skaičių, tokiu būdu atlaisvinant disko vietas, pagreitinant duomenų išrinkimą ir kt. Trajektorija, kaip jau minėta, susideda iš vietų nusakančių įrašų sekos laike. Trajektorija savo prasme yra praeities informacija. Santykis su realiu laiku yra tik toks: trajektorija realiu laiku vis dar formuojama ar jau pabaigta formuoti. Todėl trajektorijos informacija dažnai vadinama *istorijos* informacija. Istorijos informacija betarpiškai susijusi su suspaudimo metodu, nes leidžia įvairiais tikslumais atvaizduoti objektų praeities judėjimo istoriją. Pritaikius skirtingus trajektorijų duomenų suspaudimo metodus, vietos duomenų istorija bus atvaizduojama skirtingai. Metodai pasirenkami pagal reikiamą tikslumą, istorijai saugoti reikalingą saugyklos talpą ir kt.

Trajektorijos suspaudimas – šių įrašų kiekio mažinimas. Įrašas gali būti suprantamas kaip objekto pozicija dvimatėje erdvėje (x,y) laiko momentu t . Siekiant sumažinti trajektorijos taškų skaičių naudojami įvairūs trajektorijos duomenų suspaudimo algoritmai. Egzistuoja du trajektorijų suspaudimo metodų tipai, kurie gali būti taikomi kartu arba atskirai:

- 1) Suspaudimas, skirtas praeities trajektorijoms. Šiuo atveju žinoma visa objekto praeities judėjimo trajektorija. Spaudžiant galima remtis ir ankstesniais ir vėlesniais duomenimis lyginant su spaudžiamuoju intervalu.
- 2) Suspaudimas atliekamas realiu laiku. Čia remiamasi tik ankstesniais trajektorijų taškais lyginant su spaudžiamuoju laiko intervalu.

Šie būdai gali būti taikomi abu: iš pradžių realiu laiku formuojant trajektoriją atliekamas realaus laiko suspaudimas (atsižvelgiama tik į kaimynines pozicijas). Po to, laikui bėgant, trajektorijas galima sendinti palaiptams (tuomet, kai to nori vartotojas ir su jo norimu tikslumu) naudojant praeities trajektorijoms skirtus metodus (atsižvelgiama į visą judėjimą).

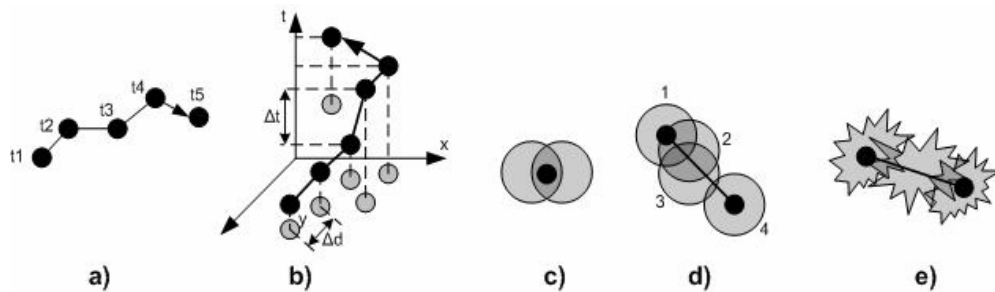
Praeities trajektorijų suspaudimo pavyzdžiu galima laikyti DP algoritmą [8]. Pats DP algoritmas ir jo pagrindu sukurti modifikuoti algoritmai [9] yra skirti praeities trajektorijų suspaudimui, kadangi algoritmas naudoja visą duomenų bazėje išsaugotą trajektoriją. Realaus laiko trajektorijų apdorojimu pagrįsti algoritmai (statinio duomenų suspaudimo [10], atstumu pagrįstas duomenų suspaudimo [10], grupavimu pagrįstas duomenų suspaudimo [10], dinaminio duomenų suspaudimo [10], mišrus slenksčių modelis [11]) apdoroja trajektorijos duomenis dalimis vos tik juos gavę. Sprendimas atliekamas, remiantis tik keliais praeities taškais, dėl to yra mažai atsižvelgiama į bendrą judančio objekto judėjimo tendenciją ir kryptingumą. Tokio pobūdžio algoritmai naudoja sistemos resursus realiu laiku dėl to jų naudojimas gali daugiau apkrauti sistemą negu naudojimas algoritmu, apdorojančiu praeities trajektorijas tokiu laiku, kai sistema yra visai neapkrauta. Be to, praeities trajektorijas apdorojantiems metodams yra svarbiau tai padaryti ne kuo greičiau, tačiau su vartotojo užduotu trajektorijos spaudimo efektyvumo lygiu. Todėl naudojant praeities trajektorijas apdorojančius algoritmus duomenų suspaudimo lygį galima keisti pagal vartotojų poreikius, vienu metu galima sekti žymiai daugiau objektų, nes tolesnis trajektorijos duomenų apdorojimas vyksta tuomet, kai sistema yra neapkrauta. Be to, tokio pobūdžio algoritmais atskirų judančių objektų trajektorijos gali būti suspaudžiamos atskirai (vienu metu vienu, kitu metu - kitų objektų trajektorijos), kai tuo tarpu duomenis apdorojant realiu laiku tai reikia daryti visiems judantiems objektams lygiagrečiai vienu metu.

Trajektorijų suspaudimo realiu laiku metodai. Pagrindinis trajektorijos informacijos suspaudimo metodų tikslas – sumažinti į judančių objektų duomenų bazę patenkančių duomenų kiekį, tuo pačiu kuo mažiau prarandant jų tikslumą. Trajektorijos duomenų suspaudimui naudojami laiku [10,11,12,13], atstumu [10,11,12,14,15,16] pagrįsti vietos duomenų atnaujinimo metodai. Šių metodų esmė – tam tikrais laiko intervalais atnaujinti judančio objekto vietos informaciją, kurios kitimas priklauso nuo objekto judėjimo greičio, krypties ar įveikto atstumo.

Kai sistema labai apkrauta, išimtiniais atvejais gali būti taikomas spėjimu [10] pagrįstas metodas, kuris spėja judančio objekto sekančią poziciją pagal greitį ir kryptį. Tačiau trajektorijos gali būti formuojamos ne tik remiantis įvairiomis laiko ar judančio objekto judėjimo charakteristikomis. Objektų pozicijos į trajektoriją gali būti įtraukiamos priklausomai nuo to, ar objekto judėjimas yra panašus į tiesę - tada trajektorija turės mažiau taškų. Jei judėjimas panašus į kreivę, tada trajektorija turės daugiau taškų priklausomai nuo judėjimo kreivumo lygio [11].

Trajektorijų informacijai suspausti dažniausiai naudojami modeliai: statinio ar dinaminio duomenų suspaudimo modeliai, atstumu ar grupavimu pagrįsti duomenų suspaudimo modeliai bei mišrus slenksčių modelis.

Statinio duomenų suspaudimo modelyje (5 pav. a) judančio objekto pozicijos yra fiksuojamos naudojant nekintantį duomenų nuskaitymo periodą. Šis modelis universalus ir tinka visoms vietos nustatymo technologijoms, tačiau tikslumas ir saugomų duomenų kiekis priklauso nuo duomenų nuskaitymo periodo. Kuo mažesnis periodas, tuo daugiau vietos užims vietos informacija ir tuo objekto judėjimo trajektorija bus tikslesnė. Tikslumas gaunamas didelio duomenų kiekio sąskaita ir dideliame objektų skaičiui, kad ir kokia vietos nustatymo technologija jie būtų stebimi, statinis duomenų suspaudimo modelis netinka. Todėl šis modelis dažniausiai naudojamas tik palyginimui su kitais modeliais.



5 pav. Statinis modelis (a), atstumu pagrįstas modelis (b), atstumu pagrįsto modelio situacijos (c,d,e)

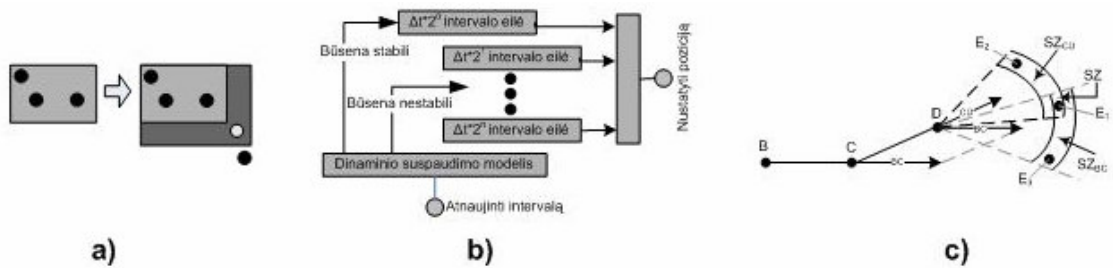
Atstumu pagrįstame duomenų suspaudimo modelyje (5 pav. b) naudojamas kintamas vietos duomenų nuskaitymo intervalas. Objektui judant greitai nuskaitymo intervalai yra maži, kai tuo tarpu judant lėtai - duomenys nuskaitinėjami rečiau. Čia nustatomas nueito atstumo kitimo dydžio slenkstis ir kiekvieną kartą atnaujinant objekto vietos duomenis, nueitas atstumas yra lyginimas su slenksčiu, tokiu būdu kontroliuojant nuskaitymų skaičių. Šis modelis pilnai tinka GPS technologijai, kadangi ja pakankamai tiksliai nustatoma objekto pozicija. GSM technologija galima nustatyti poziciją tik su celės dydžio paklaida, todėl atsiranda vietos duomenų neapibrėžtumas. Tarkime, kad buvo nustatytos dvi objekto pozicijos, kurios yra skirtingose celėse.

Jei celės persikerta tampa nebeaišku ar objektas pajudėjo, nes objektas gali būti bet kurioje celės vietoje (5 pav. c). Tačiau gretimų celių neregistruojant, galima sumažinti neapibrėžtų duomenų kiekį, t.y. duomenys yra suspaudžiami išmetant iš jų perteklinę informaciją. Tada metodas pasidaro tinkamas GSM technologijai. Tarkime, kad 1 celė yra paskutinė DB įrašyta pozicija, tada 2 ir 3 celės nėra įrašomos, nes persikerta su 1, 4 įrašoma, nes nesikerta su 1, o persikirtimus galima apskaičiuoti, kadangi yra žinomas technologinis netikslumas, susijęs su spinduliu r (5 pav. d). WLAN technologijai metodą taikyti sunkiau. Reikia paskaičiuoti WLAN prieigos taško maksimalią veikimo zoną ir neregistruoti persikertančių zonų. Objekto nuotolis registruojamas tik nepersikertančių zonų atveju (5 pav. e).

Grupavimu pagrįstame duomenų suspaudimo modelyje (6 pav. a) naudojamas minimalus trajektoriją apimantis stačiakampis, kuris objektui judant didėja. Jei nagrinėjamas judantis objektas juda stačiakampio ribose, vietos duomenų nuskaitymai yra reti, tačiau jeigu objektas juda greičiau negu didėja stačiakampis, nuskaitymas dažnėja. GPS technologijai šis modelis tinka, GSM galima būtų taikyti, tikrinant kiek procentų trajektorijos apskritimo yra už stačiakampio ribų. Šie procentai atitinka per didelį į JODB patalpintų įrašų skaičių. WLAN technologijai modelį būtų galima pritaikyti padarius analogiškus pakeitimus, t.y. skaičiuojama kiek procentų geometrinės figūros išlenda iš stačiakampio ribų.

Dinaminio duomenų suspaudimo modelis (6 pav. b) taikomas kai sistemos resursai (atmintis, procesorius) yra apkrauti. Jeigu sistema yra stabilioje būsenoje, judančio objekto vietos duomenys nuskaitymi kaip ir statiniame modelyje. Kai sistema nėra stabili, judantys objektai pagal prioritetą paskirstomi į keletą eilių su skirtingais nuskaitymo intervalais $\{\Delta t * 2^0, \Delta t * 2^1, \dots, \Delta t * 2^n\}$. Tinka visoms technologijoms, tik GSM ir WLAN reikia padaryti analogiškus tikrinimus kaip ir atstumu pagrįstame modelyje. Šį modelį galima taikyti kuriamai sistemai, kai judančių objektų skaičius viršija tam tikrą kiekį.

Mišriame slenksčių modelyje (6 pav. c) naudojamos saugios zonos. Saugi zona yra tokia zona, į kurią patekę taškai nėra įtraukiami į trajektoriją (galime objekto judėjimą laikyti saugiu, prognozuojamu, todėl objekto pozicijų fiksuoti nereikia). Tarkime, kad B ir C yra paskutiniai į JODB įrašyti trajektorijos taškai [11], o D - paskutinė nustatyta, bet dar neįrašyta trajektorijos pozicija. Saugi zona SZ_{CD} sudaroma remiantis C ir D taškais, o SZ_{BC} - remiantis B ir C. Modelyje saugi zona SZ (6 pav. c) yra apskaičiuojama perdengiant saugias zonas SZ_{CD} ir SZ_{BC} .

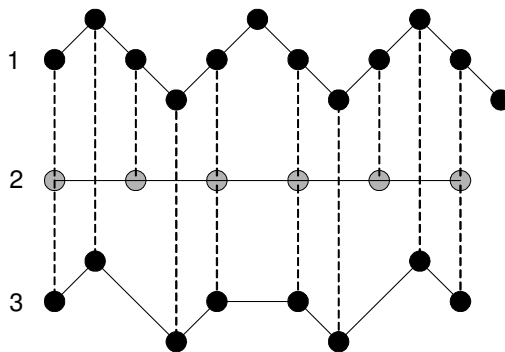


6 pav. Grupavimu pagrįstas modelis (a), dinaminio duomenų suspaudimo modelis (b), mišrus slenksčių modelis (c)

Jei trajektorijos taškas patenka į saugumo zoną, jis ignoruojamas (E1). Priešingu atveju fiksuojama, kad įvyko reikšmingas pasikeitimas objekto judėjime (E2, E3 atvejai). Tuomet nepatenkantis į saugumo zoną taškai įrašomi į trajektoriją. Modelis lengvai taikomas GSM, nes esant celėms saugios zonos būtų daug didesnės. Modelis galėtų būti pritaikytas ir WLAN, nors objekto buvimo vieta ir nėra tiksliai aprašoma. Čia reikėtų įvertinti maksimaliai galima paklaidą tarp realios objekto pozicijos ir nustatytos. Įvertinus paklaidą, galima objekto pozicija nebus taisyklinga geometrinė figūra. Saugių zonų sudarymas šiuo atveju taptų sudėtingesnis dėl reikalaujamų didesnių resursų, o pačios saugios zonos nebūtų tokios taisyklingos, todėl daugeliu atveju reikėtų spėti ar objekto judesio pasikeitimas yra svarbus ar ne.

Duomenų atrinkimui taip pat galima naudoti ir kitas strategijas, pavyzdžiui, egzistuojančius metodus naudoti kartu. Arba norint sužinoti ar reikia įterpti sugeneruotą objekto poziciją į trajektoriją, reikia iš duomenų bazės parinkti du paskutinius taškus (1 ir 2) ir įvertinti skirtumą tarp jų koordinačių (tarp 1 ir 2 taško). Tuomet yra atliekamas patikrinimas ar 3 taškas (naujai atėjęs) skiriasi nuo 2 lygiai per tiek pat kiek 2 skiriasi nuo 1. Jeigu skirtumas tarp 2 ir 3 taškų yra didesnis negu tarp 1 ir 2 taškų, vadinasi 3 tašką reikia įterpti, kadangi objektas juda ne tiesia trajektorija.

Trajektorijos taškų skaičius gali būti sumažintas imant tik kas kelintą tašką (-us) arba trinant kai kuriuos taškus. Trajektorijos kokybė priklausys nuo priimto sprendimo, kuriuos taškus reikia naikinti. 7 pav. yra pavaizduota „tobula“ trajektorija (1), trajektorija, gauta naikinant kas antrą tašką (2) ir trajektorija naikinant kas trečią tašką (3). Dėl pasirinktų skirtingų sprendimų (2) ir (3) trajektorijos skiriasi. Tokiu būdu galima lengvai sumažinti trajektorijos tikslumą (ne visada norima pateikti geriausios kokybės duomenis (saugumo tikslais)). Taip pat galima naikinti visus iš eilės einančius taškus, jeigu pastebimas pastovumas, bet tik tuo atveju, jeigu nėra nukrypimo nuo tiesios trajektorijos. Tokiu būdu būtų galima apjungti metodus, atrenkančius tik kai kuriuos taškus, ir metodus, neleidžiančius praleidinėti trajektorijos taškų svarbiais judėjimo momentais (kai objektas juda kreivai).



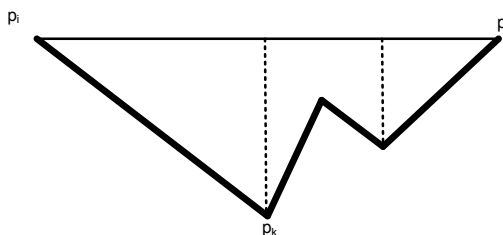
7 pav. Originali (1) ir 2 iš jos gautos trajektorijos (2) ir (3)

Egzistuoja du vietos duomenų suspaudimo metodų tipai, realiu laiku apdorojant objektų judėjimo trajektorijas dar prieš įrašant jas į duomenų bazę. Pirmajame priklausomai nuo informacijos svarbos keičiami judančių objektų vietos duomenų gavimo intervalai – kuo objektas greičiau ir daugiau nenuspėjamai juda, tuo objekto vietos informacijos nuskaitymai turi būti dažnesni. Šia idėja remiasi atstumu ir gretimų pozicijų grupavimu pagrįsti, o taip pat dinaminio duomenų surinkimo modeliai. Antrajame remiamasi ne laiko intervalų keitimu, tačiau judančio objekto pozicijų fiksavimu, priklausomai nuo informacijos svarbumo. Šią idėją geriausiai išnaudoja aukščiau aprašytas mišrus slenksčių modelis. Pirmas tipas taikant mobiliosioms ir bevielėms technologijoms yra geresnis, kadangi gaunamas didesnis trajektorijos tikslumas, kai tuo tarpu antrojo tipo atveju tikslumo maksimumas priklauso nuo sistemoje įrašyto (nekeičiamo) nuskaitymų intervalo. Tačiau abu tipai turi mobiliosioms ir bevielėms technologijoms tinkamų savybių, kurias apjungus galima sukurti visiškai naują modelį, apimančią abiejų tipų geriausias savybes. Pirmojo tipo modeliuose objektui judant greičiau vietos informacijos nuskaitymų skaičius didinamas, o judant lėčiau – mažinamas, tačiau visai yra neatsižvelgiama ar jis juda tiesiai ar kreivai. Jeigu objektas juda prognozuojamai, faktas ar objektas juda greitai ar lėtai tampa nesvarbus – į duomenų bazėje saugomą trajektoriją neturi būti įtraukiamos jo judėjimo pozicijos, sistema turi išlikti neapkrauta, kadangi tai nėra svarbi informacija. Tam, kad įvertinti objekto judėjimo kreivumą turi būti papildomai pritaikomos antrojo tipo modelių savybės. Tokiu būdu pirmojo tipo modelių savybių dėka galima didinti tikslumą, o antrojo tipo modelių - sutaupyti disko vietos. Tuo atveju, kai disko vietos kiekis yra ribotas galima laikytis strategijos, kad senas trajektorijas dar geriau suspausti, kad liktų vietos naujai atėjusiems trajektorijos taškams arba naujai atėjusius trajektorijos taškus atmesti, priklausomai nuo to, kuri informacija yra svarbesnė – praeities ar dabarties. Taip pat galima kontroliuoti eiliškumą, kuriuo bus trajektorijos sistemoje apdorojamos, suteikiant trajektorijoms prioritetus. Vienas iš šio darbo tikslų yra šio naujo metodo realizavimas ir darbo rezultatų įvertinimas. Šis metodas yra darbe kuriamos sistemos, kuri leidžia

simuliuoti sekamų objektų judėjimą, trajektorijų spaudimą realiu laiku, praeities trajektorijų suspaudimą ir kt., dalis.

Praeities trajektorijų suspaudimo metodai. Kreivių taškų mažinimas buvo studijuojamas iš daugelio perspektyvų: geografinėse informacinėse sistemose [17,18], skaitmeninių vaizdų apdorojime [19], kompiuterinėje geometrijoje [20,21]. Tačiau dabar toks kreivių prastinimas gali būti pilnai panaudotas judančių objektų sekimo sistemų gautiems duomenims apdoroti. Stengiantis išsiaiškinti, kuris metodas labiausiai tinkamas praeities trajektorijų suspaudimui, dažnai vienu iš pagrindinių tikslų tampa trajektorijos suspaudimo paklaidos valdymas, kas leidžia metodui nustatyti šią paklaidą kaip parametą (apdorojant trajektoriją realiu laiku nėra įvertinamos ilgesnio laiko trajektorijos krypties tendencijos, dėl to paklaidos parametras gali būti nustatytas tik mažai trajektorijos daliai, o ne visai trajektorijai). Parametras yra laisvai parenkamas ir užtikrina, kad paklaida bet kuriuo metu bus lygi arba bus mažesnė negu nustatytas parametras. Šis parametras priklauso nuo dalykinės srities. Jeigu dalykinė sritis yra susijusi su krovinių transportavimu arba plukdymu, tai paklaida gali būti didelė, tačiau, jeigu yra sekamas bankinis automobilis, tuomet paklaida turi išlikti maža, kadangi reikia labai tiksliai žinoti, kur sekamas objektas buvo tam tikru metu.

Vienas iš populiariausių metodų, kuriuo galima nustatyti trajektorijos suspaudimo paklaidą yra Douglas-Poiker (DP) algoritmas [17]. Jis buvo sukurtas kreivių suspaudimui ir stengiasi išsaugoti objekto judėjimo kryptį aproksimuotoje kreivėje, naudodamas iš pradžių nustatytą nukrypimo nuo originalios trajektorijos slenkstį, kuris gali kisti priklausomai nuo to, kokio suspaudimo lygio yra reikalaujama. DP algoritmas rekursiškai aproksimuoja duotą kreivę. Labiausiai nutolusi viršūnė nuo tiesės, kuri jungia pradžią ir pabaigą, parenkama kaip trajektorijos padalinimo taškas. Tarkime *pradžios_viršūnė* yra p_i ir *pabaigos_viršūnė* yra p_j . Jeigu didžiausias atstumas nuo kokios nors viršūnės p_k iki linijos segmento $p_i p_j$ yra didesnis negu pasirinkta leidžiama paklaida ϵ , trajektorija yra padalinama į dvi dalis ties p_k ir rekursiškai kviečia procedūrą kiekvienai iš trajektorijos dalių – $p_i p_k$ ir $p_k p_j$. Jeigu didžiausias atstumas nuo p_k iki linijos segmento yra mažesnis negu pasirinkta leidžiama paklaida ϵ , tuomet visos viršūnės tarp p_i ir p_j yra pašalinamos iš trajektorijos, o šis segmentas yra aproksimuojamas į tiesią liniją $p_i p_j$ (8 pav.).



8 pav. DP algoritmu apdorojama kreivė: p_k – toliausiai nuo $p_i p_j$ nutolęs taškas, ties kuriuo padalinama kreivė.

Tačiau tokio pobūdžio algoritmas gali būti pritaikytas ir judančių objektų trajektorijoms, įsivaizduojamoms, kaip kreivės dvimatėje erdvėje. Tačiau realiai trajektorijos turi dar vieną labai svarbų parametą - laiką. Į laiką turi būti atsižvelgta, nes nėra tikslo naikinti trajektorijos taškus, tarp kurių yra dideli laiko tarpai. Taigi paklaida gali būti skaičiuojama ne kaip atstumas tarp trajektorijos segmento ir labiausiai nuo jo nutolusio taško, tačiau kaip atstumai tarp laiko atžvilgiu sinchronizuotų pozicijų (vienos originalioje trajektorijoje, kitos aproksimuotoje trajektorijoje).

Greičiu pagrįstas DP algoritmas. Dažniausiai praeities trajektorijų suspaudimo algoritmai taikomi žemėlapiuose mastelio keitimui, tačiau tokio pobūdžio algoritmai gali būti pritaikyti ir bevielėms technologijoms. Egzistuoja keletas tokio pobūdžio metodų, tačiau jie turi trūkumų. Egzistuojantį trajektorijų suspaudimo algoritmą (DP) galima pagerinti išnaudojant laiko ir erdvės informaciją [16]. Tai gali būti padaryta analizuojant apskaičiuotus judančių objektų greičius vienas po kito einančiuose trajektorijos segmentuose. Didelis skirtumas tarp judėjimo greičių dviejuose vienas po kito einančiuose trajektorijos segmentuose gali būti kriterijumi, kuris panaudojamas tam, kad išlaikyti tašką, esantį šių segmentų susikirtimo vietoje. Tuo tikslu gali būti nustatytas *greičio_slenkstis*. Toliau yra surandamas trajektorijos taškas, kuriame greičio skirtumas yra didžiausias. Jeigu toks greičio skirtumas yra didesnis negu iš pradžių nustatytas *greičio_slenkstis*, toks trajektorijos taškas visada bus išsaugomas.

Algoritmo pseudokodas:

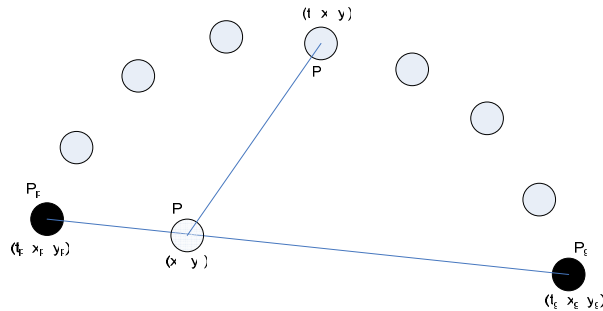
```

Procedure TD_SP(trajektorija, greičio_slenkstis)
IF ilgis(trajektorija) <= 2 THEN
RETURN trajektorija
ELSE RASTAS_GREITIS=0
RASTO_TASKO_INDEKSAS=0
FOR i=2 UNTIL ilgis(trajektorija)-1 DO
 $V_{i-1} = |trajektorija[i]_{erdvė} - trajektorija[i-1]_{erdvė}| /$ 
 $(trajektorija[i]_{laikas} - trajektorija[i-1]_{laikas})$ 
 $V_i = |trajektorija[i+1]_{erdvė} - trajektorija[i]_{erdvė}| /$ 
 $(trajektorija[i+1]_{laikas} - trajektorija[i]_{laikas})$ 
IF  $|V_i - V_{i-1}| > RASTAS_GREITIS$  THEN
RASTAS_GREITIS= $|V_i - V_{i-1}|$ 
RASTO_TASKO_INDEKSAS=i
END IF
END FOR
IF RASTAS_GREITIS > greičio_slenkstis THEN
RETURN TD_SP(trajektorija[1, RASTO_TASKO_INDEKSAS], greičio_slenkstis) +
TD_SP(trajektorija[RASTO_TASKO_INDEKSAS, ilgis(trajektorija)], greičio_slenkstis)
ELSE
RETURN [trajektorija[1], trajektorija[ilgis(trajektorija)]]
END IF
END ELSE

```

Laiku pagrįstas DP algoritmas. Tam, kad sprendimas pašalinti trajektorijos tašką būtų korektiškas, jis turi remtis jo laiko ir erdvės dedamosiomis ir aproksimuotomis judančio objekto

pozicijomis naujoje trajektorijoje [16]. Todėl atstumas tarp originalios trajektorijos taško ir aproksimuotos trajektorijos nebūtinai turi būti statmuo (9 pav.).



9 pav. Originalios trajektorijos taškai, įskaitant ir P_i , pradžios ir galo taškai P_p, P_g naujoje trajektorijoje, P_i taško aproksimuota pozicija P_i' .

Čia pavaizduotas originalus trajektorijos taškas P_i ir jo aproksimacija P_i' naujoje trajektorijoje $P_p - P_g$. P_i' taško koordinatės yra apskaičiuojamos, remiantis dviem laiko intervalais Δg (1) ir Δi (2), reiškiančiais judėjimo laiką iš P_p į P_g ir iš P_p į P_i . Šie judančio objekto judėjimo laikai yra nustatomi iš originalios trajektorijos duomenų, kaip laiko momentų skirtumai. Po to, kai aproksimuota pozicija P_i' yra nustatyta, remiantis (3) ir (4), sekantis žingsnis apskaičiuoti atstumą tarp jos ir originalios P_i , ir naudoti tokį atstumą kaip kriterijų, kuriam būtų galima iš pradžių nustatyti leidžiamą slenkstį

$$\Delta g = t_g - t_p; \quad (1)$$

$$\Delta i = t_i - t_p; \quad (2)$$

$$x_i' = x_p + \Delta i / \Delta g (x_g - x_p); \quad (3)$$

$$y_i' = y_p + \Delta i / \Delta g (y_g - y_p); \quad (4)$$

Tai gali būti svarbus DP algoritmo pagerinimas ne tik dėl to, kad yra naudojamas tikslesnis atstumo skaičiavimas, bet ir dėl to, kad įtraukiama ir laiko dedamoji. Tokius pakeitimus įvertinančio algoritmo pseudokodas:

```

Procedure TD_TR(trajektorija, atstumo_slenkstis)
IF ilgis(trajektorija) <= 2 THEN
RETURN trajektorija
ELSE RASTAS_ATSTUMAS=0
RASTO_TASKO_INDEKSAS=0
 $\Delta g = trajektorija[ilgis(trajektorija)]_{laikas} - trajektorija[1]_{laikas}$ 
FOR i=2 UNTIL ilgis(trajektorija)-1 DO
 $\Delta i = trajektorija[i]_{laikas} - trajektorija[1]_{laikas}$ 
 $(x_i', y_i') = trajektorija[1]_{erdvė} + (trajektorija[e]_{erdvė} - trajektorija[1]_{erdvė}) * \Delta i / \Delta g$ 

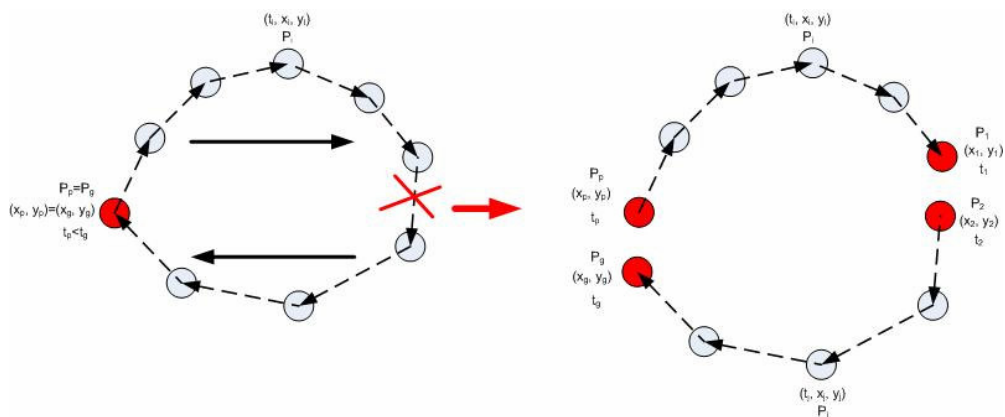
```

```

IF |trajektorija[i]_erdvė - (xi, yi)| > RASTAS_ATSTUMAS THEN
RASTAS_ATSTUMAS =|trajektorija[i]_erdvė - (xi, yi)|
RASTO_TASKO_INDEKSAS=i
END IF
END FOR
IF RASTAS_ATSTUMAS > atstumo_slenkstis THEN
RETURN TD_TR(trajektorija[1, RASTO_TASKO_INDEKSAS], atstumo_slenkstis) +
TD_TR(trajektorija[RASTO_TASKO_INDEKSAS, ilgis(trajektorija)], atstumo_slenkstis)
ELSE
RETURN [trajektorija[1], trajektorija[ilgis(trajektorija)]]
END IF
END ELSE

```

Tačiau šis nagrinėjamas laiku pagrįstas DP algoritmas pasidaro netinkamas (10 pav.), kai realus objekto judėjimas yra panašus į kilpą arba grįžtant atgal. Šiuo atveju, jeigu objektas grįžta į savo pradinę poziciją ($P_p=P_g$), trajektorijos taškui P_i tampa nebeįmanoma paskaičiuoti projekcijos, algoritmas veikia korektiškai tik tada, kai objektas juda tik į priekį. Siūlomas sprendimas problemai spręsti yra toks, kad reikia trajektoriją nukirsti ten, kur keičiasi objekto judėjimo kryptis (10 pav.). Tuomet ir taškui P_i ir taškui P_j projekcijos bus paskaičiuojamos korektiškai.



10 pav. Laiku pagrįsto DP algoritmo trūkumai ir jų pašalinimas

Šio patobulinto metodo pagalba gali būti atlikti keli iš eilės einantys praeities trajektorijos suspaudimo etapai. Kiekvienam sekančiam suspaudimo etapui leidžiama paklaida būtų vis didesnė, todėl kuo daugiau etapų atlikta, tuo mažesnę tikslumą turės gauta trajektorija, tačiau tuo pačiu metu ji užims žymiai mažiau disko vietos. Iš pirmo žvilgsnio gali pasirodyti, kad nebegalima panaudoti mažo tikslumo informacijos, tačiau toks požiūris yra klaidingas. Net labai suspaudus trajektoriją vis dar galima gauti kartais labai naudingos informacijos, pvz., kad tuo metu objektas buvo tam tikroje šalyje, nesigilinant į judėjimo ypatumus.

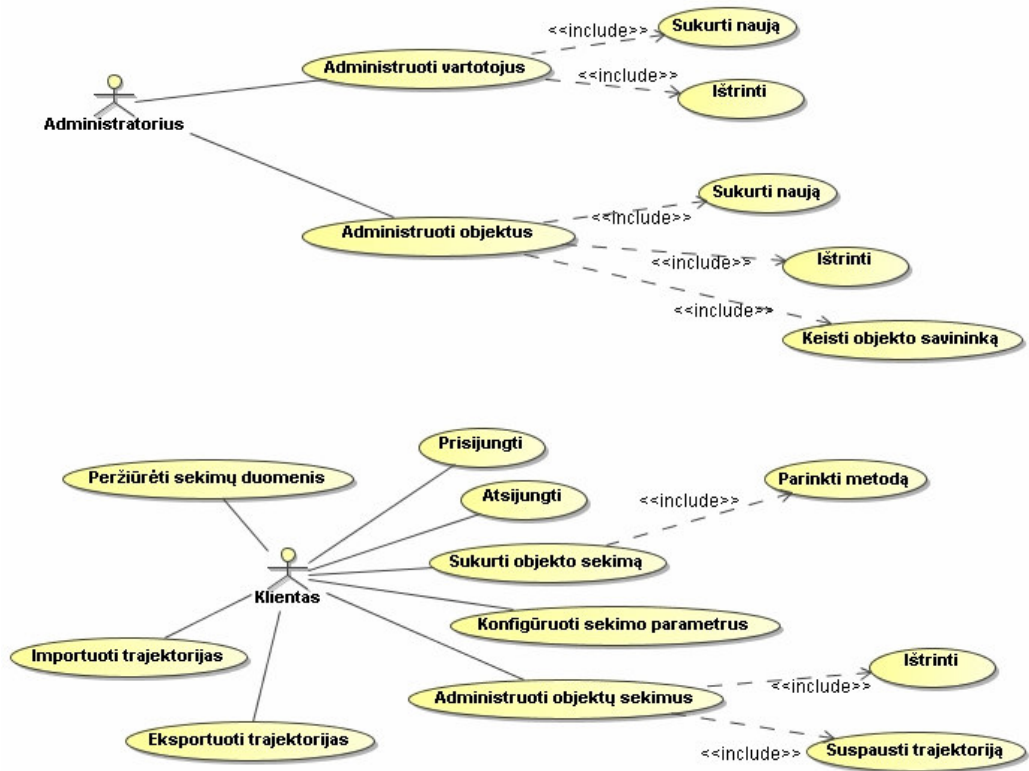
Tokio pobūdžio algoritmai gali būti naudojami atskirai, tačiau gali būti ir naudojami kartu metodus grupuojant (greičiu pagrįstas – laiku pagrįstas arba laiku pagrįstas – greičiu pagrįstas) siekiant žymiai pagerinti trajektorijų suspaudimo rezultatus (kokybę).

2.3.3. Esamų metodų trūkumai

- Yra metodologija (teoriniame lygmenyje), tačiau nėra pateikta trajektorijų suspaudimo metodų realizacijų, trūksta statistikos apie realų jų veikimą ir palyginimą.
- Nenagrinėjamas trajektorijų suspaudimo metodų tinkamumas skirtingoms technologijoms.
- Kai kurie minėti algoritmai (laiku pagrįstas DP algoritmas) yra pritaikyti tik objektų judėjimui į priekį.
- Nenagrinėtas disko vietos sutaupymas dėl trajektorijų suspaudimo metodų taikymo.
- Analizuojami metodai yra skirti labai tikslioms pozicionavimo technologijoms (GPS), todėl GSM technologijai jie nėra tinkami.
- Nėra atliktų tyrimų, kurie parodytų, kaip sumažėja trajektorijos kokybė, suspaudus ją 2, 4, 8, 16, ... n kartų.
- Esamuose darbuose net neužsimenama, kaip vienus trajektorijų suspaudimo metodus būtų galima derinti su kitais.

2.4. Kompiuterizuojami panaudojimo atvejai

Kompiuterizuojami panaudojimo atvejai (11 pav.) šiek tiek skiriasi nuo veiklos panaudojimo atvejų. Čia yra įvedamas vartotojų administravimas, objektų administravimas, ir pan. Vartotojui „Klientas“ atsiranda panaudojimo atvejai „Administruoti objektų sekimus“, „Eksportuoti trajektorijas“, „Importuoti trajektorijas“. Administravimo veiksmus atlieka naujas aktorius „Administratorius“. „Duomenis teikiančio operatoriaus“ nelieka, užsiregistravęs vartotojas pats gali sukurti sekamus objektus, sukurti pačius objektų sekimus su galimybe keisti jiems parametrus (pvz., parinkti laiko intervalą, metodą kuriuo bus duomenys kaupiami).



11 pav. Kompiuterizuojami panaudojimo atvejai

3. Bevielių mobiliųjų technologijų vietos duomenų istorijos atkūrimas ir jam skirta sistema

Skirtingomis bevielėmis technologijomis (GPS, GSM, WLAN) yra gaunami skirtingo formato ir tikslumo vietos duomenys, todėl šių duomenų suspaudimui turi būti naudojami skirtingi metodai, kurie įvertintų šių technologijų ypatybes.

Kadangi tikslo bandyti sukurti universalų vietos duomenų suspaudimo metodą visoms technologijoms, siūloma kiekvienos technologijos duomenis apdoroti skirtingais metodais. Iš esmės būtų galima tirti tik GPS technologija gaunamus duomenis, kadangi jie yra tiksliausi, o GPS technologija vis dažniau naudojama įvairiose srityse, tačiau GPS technologija veikia ne visur, ne visi mobilūs įrenginiai turi integruotus GPS modulius, todėl kaip alternatyva jai yra taikoma GSM technologija, leidžianti rasti judančio objekto poziciją ir uždaroje patalpose (GPS technologija pastatuose dažniausiai neveikia arba veikia nepakankamai gerai), arba WLAN, kuri leidžia gana tiksliai nustatyti objekto vietą pastato viduje. Be to, GSM technologija žymiai labiau paplitusi, negu GPS.

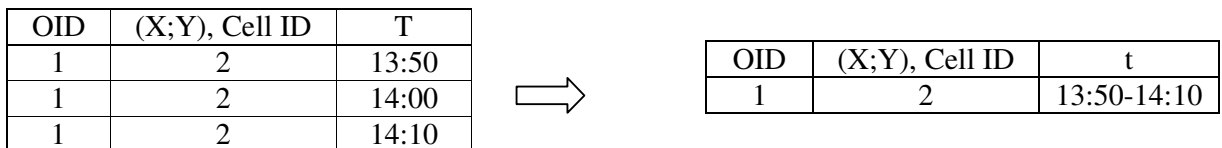
Darbe kuriama eksperimentinė sistema suteikia galimybę atlikti istorijos atkūrimo įvairiomis technologijomis gautiems vietos duomenims.

3.1. GSM vietos duomenų suspaudimo metodas

GSM technologija gauti vietos duomenys yra mažesnio tikslumo už GPS technologija gautus duomenis. Nors šiai technologijai teoriškai galima taikyti jau žinomus duomenų suspaudimo modelius, tačiau tiesmukas jų taikymas būtų neefektyvus nei greičio, nei tikslumo nei naudojamos disko vietos prasmėmis. Jeigu GPS technologija gautiems duomenims grubiai galime taikyti formatą (OID, x_i, y_i, t_i) , kur OID – objekto identifikacinis numeris, x_i, y_i – pozicija, kurioje objektas buvo t_i momentu, tuo tarpu GSM technologija gauti duomenys atitinka šabloną $(OID, x_i, y_i, t_i, R_i)$, kur OID – objekto identifikacinis numeris, x_i, y_i – pozicija (celės arba apskritimo centras), kurioje objektas buvo t_i momentu, R_i – objekto buvimo vietos (celės arba apskritimo) spindulys.

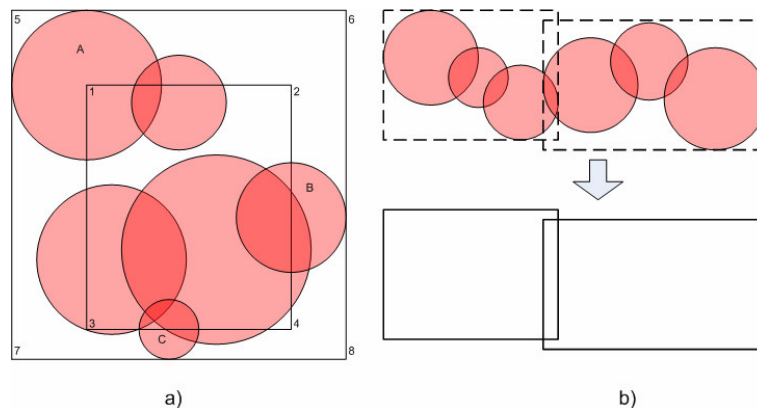
Jeigu GSM technologija gaunami vietos duomenys turi pastovų laiko intervalą, tokių duomenų suspaudimas tampa paprastesnis. Šiuo atveju neatsižvelgiama į erdvės (plokštumos ir laiko) paklaidą, dėl to nėra jokių garantijų, kad duomenų suspaudimas bus atliekamas iki iš anksčiau nustatytos ribos. GSM celės dydžio ir celės judėjimo nevertinimas gali sąlygoti nekontroliuojamą duomenų suspaudimo paklaidą. Priimant sąlygą, kad celės nejuda ir daugelis

objektų pozicijų sutampa su celės koordinatėmis, tenkinančiomis formatą (OID, x_i , y_i , t_i , R_i) galima taikyti duomenų bazės reliacinėse lentelėse esančių duomenų prastinimą (12 pav.). Tačiau intervalai tarp tokių įrašų turi būti vienodi, todėl geriausia taikyti statinį realaus laiko duomenų suspaudimo modelį trajektorijoms gauti. Toks duomenų suspaudimas galimas tik GSM technologijai, priėmus sąlygą, kad GSM tinklo celės nejuda, o judantis objektas yra aprašomas tik celės, kurioje yra, identifikaciniu numeriu (Cell ID). Kadangi vienai celei yra priskiriama daug judančio objekto nustatytų pozicijų, objekto pozicijos pasikartoja visuose įrašuose (skiriasi tik laikas) dėlto gali būti apjungtos. Siūlomame GSM vietos duomenų suspaudimo metode atsižvelgiame į tai kad intervalai tarp gretimų pozicijų vienoje celėje gali būti nevienodi.



12 pav. Mažesnio tikslumo duomenų prastinimo modelis

Pagrindinė siūlomo GSM vietos duomenų suspaudimo metodo idėja – aproksimuoti GSM technologijos pagalba gautą vietos informaciją (celių koordinates ir su jomis susietus laiko momentus) į stačiakampius (13 pav.).

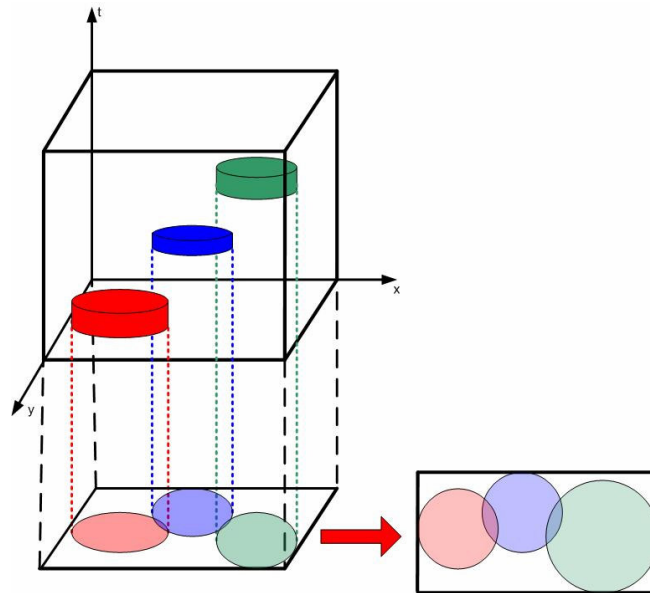


13 pav. GSM celių aproksimavimas į stačiakampius

Iš pradžių yra atrenkama įrašų (celių ar apskritimų) grupė (įrašų grupės atrinkimas yra detalai aprašomas žemiau pateiktame algoritme). Tuomet šie įrašai yra aproksimuojami. Tai gali būti atlikta 2 būdais: stačiakampis brėžiamas per suspaudžiamų celių centrus C_A , C_B , C_C (13 pav.) arba taip kad visos spaudžiamos celės pilnai į jį sutilptų, t.y. nubrėžiamas minimalus ribojantis stačiakampis [23]. Pirmu atveju paklaida būtų žymiai didesnė, kadangi negalima būti tikriems,

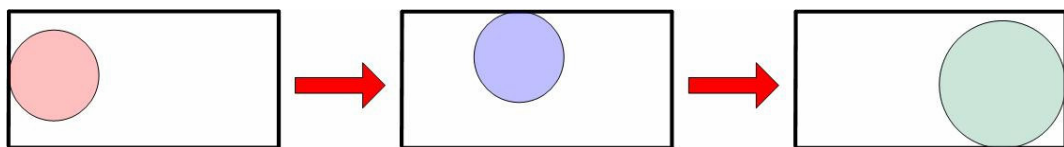
kad judantis objektas neperžengė minimaliai ribojančio stačiakampio ribų, nes objektas tikimybiškai galėjo būti bet kurioje celės vietoje.

Kadangi judančių objektų trajektorijos turi ne tik erdvės, tačiau ir laiko dedamąją, jos vaizduojamos trimatėje erdvėje. Jeigu laiko dedamoji nebūtų vaizduojama, trimatė trajektorija (judantis objektas eina per GSM technologijos celes) galėtų būti atvaizduota plokštumoje, kaip kad pateikta 14 pav.



14 pav. GSM trajektorija, vaizduojama trimatėje erdvėje ir jos projekcija dvimatėje erdvėje

15 pav. pavaizduotas algoritmo darbas grafiškai. Priimama sąlyga, kad tam tikru laiko momentu judantis objektas yra tik vienoje celėje, nors celės ir persikerta.



15 pav. Judantis objektas visada yra priskirtas vienai celei

Tarkime, kad turime tokius pradinius parametrus, kuriuos naudoja naujai kuriamas algoritmas:

m – įrašų apie judančius objektus kiekis;

i – esama pozicija sąrašė;

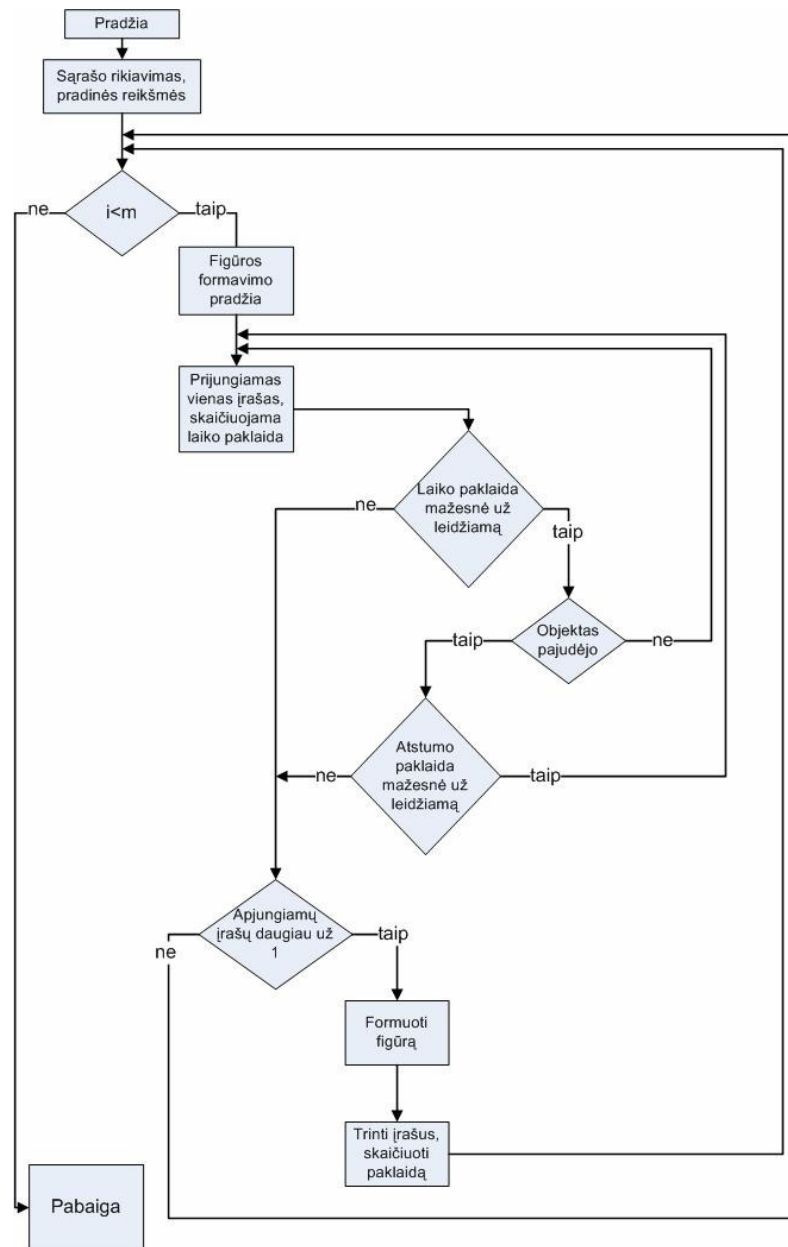
n_{ap} – apjungiamų į figūrą įrašų kiekis;

$(OID, x_i, y_i, R_i, t_i), (OID, x_{z1}, y_{z1}, x_{z2}, y_{z2}, t_1, t_2)$ – pradinis ir kelių apjungtų įrašų formatai, kur:

OID – judančio objekto identifikacinis numeris.

x_i, y_i, R_i – celės, kurioje yra objektas laiko momentu t_i , centro koordinatės ir spindulys;
 $x_{z1}, y_{z1}, x_{z2}, y_{z2}, t_1, t_2$ – apjungus gauto stačiakampio celės (14 pav.) dvi kampinės koordinatės ir laiko intervalas, kuriuo judantis objektas buvo toje zonoje.
 ϵ_t – galima laiko tarp įrašų paklaida figūroje, apjungiančioje celės;
 ϵ_d – galima atstumo tarp celių paklaida figūroje, apjungiančioje celės;
 E_t – esama laiko tarp įrašų paklaida figūroje, apjungiančioje celės;
 $E_{t,momentinė}$ – leistina laiko tarp 2 įrašų paklaida;
 $KL_{įrašo}$ – prarasta įrašo kokybė;
 $KL_{apjungimo}$ – prarasta bendra apjungtų įrašų į figūrą (stačiakampį) kokybė;
 $KL_{trajektorijos}$ – prarasta visos trajektorijos kokybė;

GSM vietos duomenų suspaudimo metodo algoritmas pateiktas 16 pav.



16 pav. GSM vietos duomenų suspaudimo metodo algoritmas

Algoritmo pradžioje įrašai išrūšiuojami pagal laiką didėjančia tvarka. Tikrinama ar tenkinama algoritmo pabaigos sąlyga: pasibaigę visų judančių objektų įrašai (esama pozicija (i) sąrašė negali būti didesnė už įrašų apie judančius objektus kiekį (m)). Tuomet yra fiksuojama naujos figūros pradžia. Nuskaicius įrašą, į atmintį yra įrašomos pirmos celės naujoje figūroje koordinatės ir laikas $(x_{\text{pirmas}}, y_{\text{pirmas}}, R_{\text{pirmas}}, t_{\text{pirmas}}) = (x_i, y_i, R_i, t_i)$. Pažymima vieta, nuo kurios formuojama figūra. Kintamajam E_t priskiriama reikšmė „0“, kadangi pradėjus naują figūrą, skaitliukas paklaidą pradeda skaičiuoti iš naujo. Kol požymis = „true“ yra nuskaitomi sekantys įrašai, formuojantys stačiakampį (kadangi požymis niekada nesikeičia, iš ciklo išeinama, kai yra viršytos laiko arba atstumo paklaidos. Nuosekliai nuskaitomas sekantis įrašas (OID, x_i, y_i, R_i, t_i). E_t momentinė yra atimama, nes laiko intervalas tarp įrašų visada bus, todėl nors ir tarp visų įrašų bus maži ir lygūs laiko intervalai, paklaida E_t vis tiek didės su kiekvienu įrašu, o ji turėtų didėti tik didesnių nukrypimų negu E_t momentinė paklaida atveju. Jeigu ($E_t > \epsilon_t$), tuomet tokių įrašų negalima apjungti, kadangi esama laiko tarp įrašų paklaida (E_t) viršys galimą (ϵ_t) paklaidą, o tai reikštų, kad tokia aproksimacija įgaus didesnę paklaidą negu nustatytoji. Fiziškai tai reiškia, kad objekto buvimo laikas (po aproksimacijos) tam tikroje vietoje gali labai skirtis nuo realaus laiko, kuriuo jis buvo toje pozicijoje. Jeigu $(x_{i-1}, y_{i-1}, R_{i-1}) < > (x_i, y_i, R_i)$ vadinasi judantis objektas perėjo į kitą celę, todėl yra galimybė, kad padidėjo atstumo paklaida. Jeigu $(x_{i-1}, y_{i-1}, R_{i-1}) = (x_i, y_i, R_i)$, tuomet judantis objektas liko toje pačioje celėje. Jeigu $|(x_{\text{pirmas}}, y_{\text{pirmas}}, R_{\text{pirmas}}) - (x_i, y_i, R_i)| > \epsilon_d$, tuomet tokių įrašų negalima apjungti, kadangi atstumo tarp celių paklaida viršys galimą paklaidą. Šiuo atveju paklaida skaičiuojama kitaip negu laiko dedamajai. Yra matuojami nuotoliai ne tarp gretimų celių, tačiau yra tikrinama, kad visos tolimesnės celės (2,3,4,...) nebūtų per daug nutolusios nuo 1 celės. Jeigu ($n_{\text{ap}} > 1$), tuomet yra apjungiamas daugiau nei vienas įrašas ir galima formuoti stačiakampį. Stačiakampis generuojamas, pagal tokias taisykles:

- 1) Pagal esamų celių centrų koordinates (x, y) ir spindulius (r) yra atrenkamos tokios celės, kurios formuoja stačiakampio rėmelį (13 pav., A, B ir C celės);
- 2) įvertinus atrinktų kraštinių celių centrus ir spindulius brėžiamas stačiakampio rėmelis (13 pav., 5678 stačiakampis);
- 3) Gautas stačiakampis (OID, $x_{z1}, y_{z1}, x_{z2}, y_{z2}, t_1, t_2$) pakeičia apjungtas celes (13 pav.). Tokiu būdu įrašai, susieti su celėmis yra ištrinami, o vienas įrašas, susietas su stačiakampiu, yra įrašomas į duomenų bazę.

Skaičiuojama paklaida laiko intervale (tarp pirmo ir paskutinio įrašų, apjungtų stačiakampiu, laiko žymių) tarp anksčiau buvusios situacijos ir dabartinės (kai celės yra apjungtos

stačiakampiu). Kiekvienam t_i (apjungtam gautu stačiakampiu) yra paskaičiuojama: $KL_{\text{įrašo}} = S_{\text{stačiakampio}} - S(x_i, y_i, R_i)$, kur $S_{\text{stačiakampio}}$ yra stačiakampio, apjungiančio celes, plotas, $S(x_i, y_i, R_i)$ yra celės, kurioje objektas buvo laiko momentu t_i , plotas; $KL_{\text{apjungimo}}$ kintamasis padidinamas per $KL_{\text{įrašo}}$. Stačiakampiu pakeistos celės ištrinamos. $KL_{\text{trajektorijos}}$ kintamasis padidinamas per $KL_{\text{apjungimo}}$. Gauta paklaida $KL_{\text{trajektorijos}}$ reiškia (sąlygoja) bendrą objekto buvimo neapibrėžtumą erdvės ir laiko atžvilgiu. Algoritmo pseudokodas:

```

1  Rikiuoti_sąrašą(t); i=0; KLtrajektorijos=0;
2  while i <= m
3      (xpirmas, ypirmas, Rpirmas, tpirmas) = (xi, yi, Ri, ti); pr_indeksas=i;
4      nap=1; Et=0; požymis=true;
5      while požymis=true
6          nap = nap+1; i=i+1;
7          Et = Et + (ti - ti-1 - Et, momentinė);
8          if (Et > εt) then GOTO 12; else GOTO 9;
9          if (xi-1, yi-1, Ri-1) <> (xi, yi, Ri) then GOTO 10; else GOTO 5;
10         if |(xpirmas, ypirmas, Rpirmas)-(xi, yi, Ri)| > εd then GOTO 12; else GOTO 5;
11         end while;
12         if (nap > 1) then GOTO 13; else GOTO 2;
13         (OID, xz1, yz1, xz2, yz2, t1, t2)=Formuoti_stačiakampį(nap, pr_indeksas);
14         KLapjungimo = Skaiciuoti_paklaida(nap, pr_indeksas);
15         Trinti_celes(nap, pradinis_indeksas);
16         KLtrajektorijos= KLtrajektorijos + KLapjungimo;
17         GOTO 2;
18     end while
19 end

```

Bendra trajektorijos paklaida $\sigma_{k_{\text{int amam_int_bendr}}}$ yra paskaičiuojama atsižvelgiant kiek procentų padidėjo gauto stačiakampio plotas, palyginti su buvusios celės plotu (gauta paklaida įvertinama procentais). Kiek iš viso „sutaupyta“ įrašų algoritmo darbo metu apskaičiuojama:

$$n_{\text{sutaupyta}} = \frac{n_{\text{ištrinta}} - n_{\text{stačiakampiu}}}{n_{\text{originalios}}} \cdot 100\% ; \quad (5)$$

Trajektorijos suspaudimo kokybės koeficientas (6) parodo santykį tarp sutaupytų įrašų kiekio ir dėl suspaudimo atsiradusios objekto buvimo vietos paklaidos. Didelis sutaupytų įrašų kiekis didina koeficiento reikšmę, tačiau didelė atsiradusi paklaidą jį mažina, kadangi ši paklaida išreiškia neigiamą duomenų suspaudimo metodo aspektą. Apibendrinus galima pasakyti, kad šis koeficientas parodo algoritmo darbo efektyvumą tam tikrais momentais ir yra apskaičiuojamas taip:

$$k_{\text{kokyb}} = \frac{n_{\text{sutaupyta}}}{\sigma_{k_{\text{int amam_int_bendr}}}}, \quad 0 \leq k_{\text{kokyb}} < \infty ; \quad (6)$$

Kokybės koeficientas gali įgyti reikšmes nuo nulio (įrašų nei kiek nesumažėjo, bet išsaugo objekto buvimo vietos paklaidą) iki begalybės (reiškia, kad nebuvo rastas nei vienas suspaudimui tinkamas įrašas, dėl to vietos paklaidą liko 0). Be to, tokio koeficiento skaičiavimas turi vieną apribojimą. Koeficientas gali būti skaičiuojamas tik tuo atveju, jeigu duomenų suspaudimas trajektorijoje vyko ir tuomet kai $\sigma_{k_{int} amam_int_bendr} < 0$. Tik tada koeficiento skaičiavimas bus korektiškas.

Algoritme numatyta galimybė naudoti skirtingų dydžių celes. Tačiau vykdant algoritmą, neapibrėžtumas didėja bet kokių atveju, nepriklausomai naikinam didelę ar mažą celę. Jeigu naikinama celė yra didelė (ir nedaug persidengia su jau esančiomis), bendra prastinimo paklaidą išauga žymiai daugiau, kadangi tai smarkiai įtakoja stačiakampio ribas. Prijungiant mažą celę, bendra paklaidą auga mažesniu greičiu. Sukurtas algoritmas taip pat įvertina ne tik buvimą ilgą laiką toje pačioje celėje, tačiau ir grįžimą celėmis atgal, pastoviai augant laikui.

Algoritmui baigus darbą įrašų laiko dedamoji yra maksimaliai suspausta (nebelieka įrašų, susietų su vienodomis pozicijomis, bet turinčių skirtingus laikus), kadangi apjungiami visi įrašai, susiję su tomis pačiomis celėmis, tačiau erdvės dedamąją galima prastinti ir toliau. Turint omenyje, kad algoritmo grąžinamas rezultatas yra stačiakampiai, apjungiantys celes, po tokio suspaudimo gautus stačiakampius galima apjunginėti į kitus stačiakampius, kol tai yra naudinga ir kol yra tenkinama naujai užduota paklaidą. Gauti stačiakampiai gali būti toliau perduoti į indeksavimo posistemę, kuri pagreitintų tokio pobūdžio duomenų išrinkimą ir pagerintų sistemos darbo rezultatus.

Tokiu būdu būtų sutaupoma daug disko vietos, kas sąlygotų pagreitintą duomenų suradimą. Metodas yra tuo efektyvesnis, kuo toje pačioje celėje praleidžiama daugiau laiko arba kai celės yra didelės. Be to, naujai gautą struktūrą, susidedančią iš stačiakampių, yra lengviau indeksuoti (rezultate yra gaunama mažiau figūrų, stačiakampiai lengviau apdorojami negu apskritimai) ir pritaikyti tam tikrų tipų užklausoms (pavyzdžiui, objektų, esančių tam tikroje zonoje paieška (*range query*) arba trajektorijos radimas pagal judančio objekto identifikacinį numerį (*OID*)).

GSM duomenų suspaudimo metodo algoritmą galima būtų patobulinti dinamiškai keičiant leidžiamas vietas ir laiko paklaidas. Turint omenyje, kad mieste ir užmiestyje judančio objekto vietos informacija įgauna skirtingą svarbą (miesto ribose gaunama informacija dažniausiai yra svarbesnė ir tikslesnė negu užmiestyje), užmiestyje vietos ir laiko paklaidos galėtų būti leidžiamos didesnės, kai tuo tarpu miesto zonoje, jas reikėtų mažinti. Faktas ar objektas yra užmiestyje, ar mieste gali būti atpažintas pagal celių koordinatas (paprastai yra žinoma ar GSM tinklo celė yra

miesto ribose ar ne). Tokiu būdu būtų randama reikiamo tikslumo informacija be reikalo neapkraunant sistemas. Patobulinimai gali būti atlikti, įvertinus, kad objektas gali judėti tik tam tikrame kelių tinkle [22]. Tokiu atveju gautos judančio objekto trajektorijos taškus būtų galima priskirti tam tikriems realaus kelio segmentams. Tačiau tokiu atveju objektų judėjimas būtų apribotas kelių tinklu ir ne visai atitiktų kitokio tipo objektų (ne autotransporto priemonių) judėjimą. Pavyzdžiui, jeigu sekamas objektas yra žmogus, galbūt tik mažą dalį jo nueito atstumo galima apdoroti pagal iš pradžių paruoštą žemėlapi.

Sukurtas algoritmas gali būti nesunkiai pritaikytas ir kitoms technologijoms, pvz., WLAN. WLAN pozicionavimo technologija neįmanoma tiksliai nustatyti objekto buvimo vietos, todėl dažniausiai tikslas yra rasti kuriame kambaryje, kabinete ar patalpoje yra sekamas objektas. Dėl tokios objekto pozicijos vaizdavimo specifikos vietos duomenis spausti neapsimoka, o tiesiog suindeksuoti tam tikrų struktūrų pagalba grafų arba medžių [24]. Tačiau jeigu norima objekto buvimo pozicija vaizduoti ne simboliu būdu, o tam tikra figūra, reikia, kad į duomenų bazę įrašoma objekto buvimo vieta, būtų aprašoma taškais. Tuomet realiai gaunama pozicija primena „debesį“, kuriame galėjo būti objektas, tačiau DB nėra praktiška saugoti visas tokio debesio viršūnes ir objekto buvimo vietą aprašyti debesio koordinatėmis, todėl duomenų bazėje objekto pozicija gali būti laikoma apskritimu su tam tikru spinduliu, kurio dydis atitinka technologijos apimamą zoną. Tokiu atveju duomenų bazėje saugomos WLAN technologija gautos objekto pozicijos yra panašios į GSM celes, kuriose yra fiksuojama judančio objekto pozicija. WLAN technologija gautos pozicijos taip pat, kaip ir GSM celės gali persidengti, kadangi tai nėra tokia tiksli informacija, kaip gaunama iš GPS (taškai). Dėl šių priežasčių algoritmas gali būti taikomas WLAN technologija gautų objektų pozicijų duomenų suspaudimui.

3.2. GPS vietos duomenų suspaudimo metodai

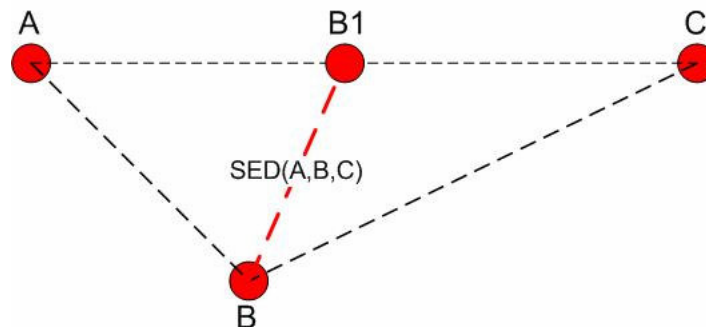
GPS technologija gauti duomenys dažniausiai iš karto įrašomi į duomenų bazę, kai tikslas yra sekti judantį objektą. Tačiau jeigu į duomenų bazę įterpiami visi be išimties duomenys, laikui bėgant krenta sistemos darbo našumas, užklaustos trajektorijų duomenų užkrovimui yra vykdomos lėčiau (sulėtėja duomenų išrinkimas), kadangi jos naudoja didelius duomenų kiekius. Dėl šios priežasties yra siūloma mažinti duomenų kiekį dar prieš juos įterpiant. Tokiu būdu būtų padaromos išvados, lyginant kiekvieno naujai atėjusio taško svarbumą su trajektorijoje jau esančių taškų svarbumais. Turint omenyje, kad didžiausią svarbumą turintys taškai turi būti išsaugoti, o mažiausią svarbumą – pašalinti, naujai atėję trajektorijų taškai yra kartais (priklausomai nuo svarbumo) yra sukeičiami su jau esančiais trajektorijoje taškais. Tokiu būdu

yra įvertinamas ir bendras trajektorijos vaizdas (randami ne momentiniai nukrypimai), kadangi naujai atėjusio taško svarbumas yra palyginimas ne tik su kaimyninių taškų svarbumu, tačiau su kiekvieno trajektorijos taško svarbumu. Tam, kad sistemai nereikėtų pereiti visos trajektorijos taškų kiekvieną kartą, kai ateina naujas taškas, taškų nuotoliai nuo trajektorijos yra saugojami atmintyje, kol vyksta trajektorijos formavimo procesas.

Darbe siekiama patobulinti duomenų suspaudimo procesą. Iš pradžių galėtų būti taikomas dabarties (realaus laiko) didelio tikslumo trajektorijų (GPS) suspaudimo algoritmas. Vėliau tiems patiems duomenims galėtų būti daug kartų (užduodant skirtingas paklaidas) pritaikytas praeities didelio tikslumo trajektorijų (GPS) suspaudimo algoritmas.

3.2.1. Dabarties didelio tikslumo trajektorijų (GPS) suspaudimo metodas

Judančio objekto vietos duomenims pasiekus duomenų atrinkimo procesą turi būti padarytas sprendimas realiu laiku – reikia įterpti atėjusius duomenis į duomenų bazę, ar naujai atėję duomenys yra svarbesni už senus? Egzistuoja keletas sukurtų algoritmų realaus laiko trajektorijų apdorojimui. Kiekvienas tokio pobūdžio algoritmų turi savo stipriąsias puses, todėl būtų naudinga apjungti kelių tipų trajektorijos suspaudimo realiu laiku algoritmus. Tokiu būdu skirtingų strategijų apjungimas būtų realizuotas taip, kad kiekvienas įterptas įrašas būtų naudingas trajektorijos atvaizdavime, o ne tik užimtų vis daugiau disko vietos. Tai reiškia, kad žymiai svarbiau turėti mažiau trajektorijų, bet tikslių, negu turėti daugiau, bet tokių, pagal kurias dėl didelio duomenų kiekio nebūtų galima operatyviai nustatyti objekto judėjimo trajektoriją. Dažniausiai trajektorijos taško įtaka trajektorijos tikslumui yra nustatoma pagal jo nuotolį nuo trajektorijos segmento. Korektiškiausia nuokrypį rasti pagal taško atstumą iki pačios trajektorijos segmento (17 pav.). Taško B atstumas SED (*Synchronous Euclidean Distance*) iki trajektorijos segmento AC yra randamas ne brėžiant statmenį iki jos (statmens brėžimas būtų korektiškas, jeigu duomenų apdorojimas vyktų tik plokštumoje, tačiau šiuo atveju dar yra įvertinama (naudojantis (8) ir (9)) ir laiko dedamoji).



17 pav. Atstumas nuo taško B iki trajektorijos segmento AC.

Toliau atstumas yra skaičiuojamas taip:

$$SED(A, B, C) = \sqrt{(x_{B1} - x_B)^2 + (y_{B1} - y_B)^2}, \text{ čia:} \quad (7)$$

$$x_{B1} = x_A + v_{AC}^x \cdot (t_B - t_A), \text{ čia: } v_{AC}^x = \frac{x_C - x_A}{t_C - t_A} \quad (8)$$

$$y_{B1} = y_A + v_{AC}^y \cdot (t_B - t_A), \text{ čia: } v_{AC}^y = \frac{y_C - y_A}{t_C - t_A} \quad (9)$$

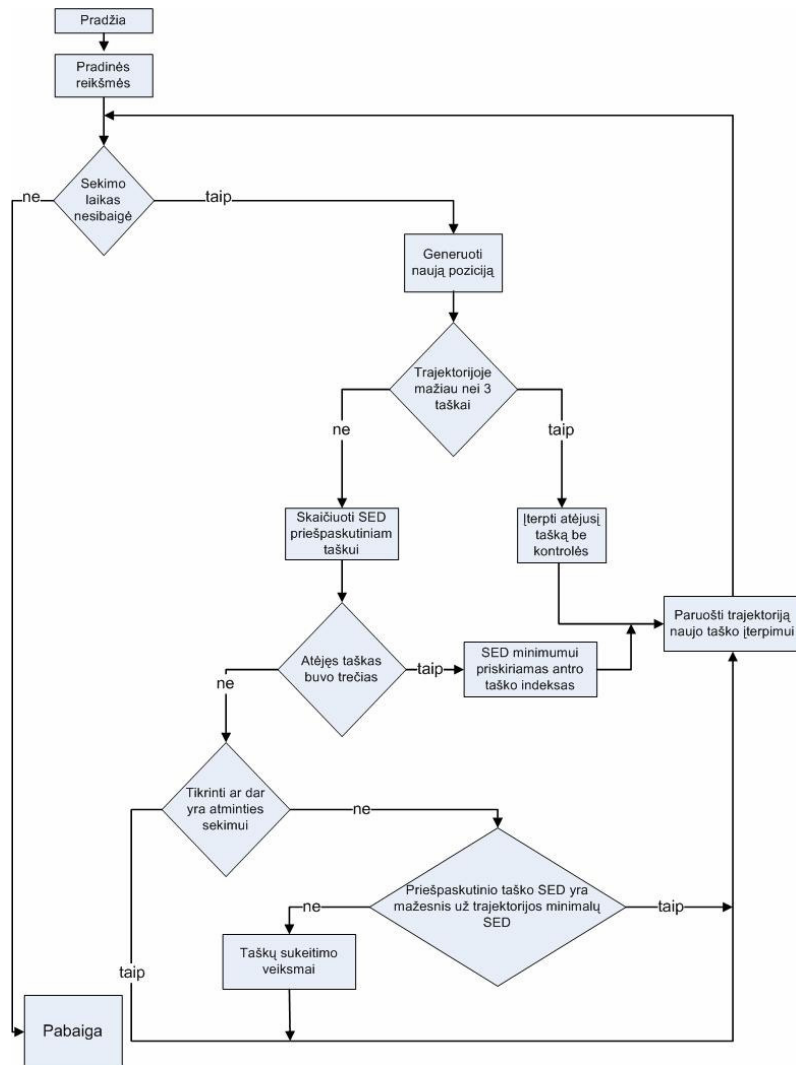
Šis metodas įvertina judančio objekto daromus posūkius, judėjimo kreivumą ir mažiausiai iš esamų dabarties trajektorijų apdorojimo būdų apkrauna sistemą, kadangi turi paprastesnę realizaciją, trajektorijos kreivumo testas nereikalauja didelių kompleksinių skaičiavimų (kaip mišriajame slenksčių modelyje). Priešingai negu kituose metoduose, kur judėjimo kreivumas yra vertinamas pagal kaimyninius taškus, šis metodas tai daro, atsižvelgdamas į bendrą objekto judėjimo vaizdą, lyginant atėjusius taškus su visais trajektorijoje esančiais taškais. Tam tikros strategijos pagalba galima valdyti metodo maksimaliai naudojamos atminties kiekį. Kuo didesnis vartotojo prioritetas, tuo daugiau atminties gali būti išskirta jo sekamų objektų vietos informacijai saugoti (vienam sekimui daugiau atminties ir didesnis sekimų skaičius, tačiau bendras naudojamos atminties kiekis neturi viršyti leistino limitu). Sekant objektą iš pradžių į atmintį yra įtraukiami visi be išimties taškai. Kai atmintis yra pilnai užpildyta, tuomet yra tikrinama, kurie taškai yra svarbesni: naujai atėję ar jau trajektorijoje esantys. Jeigu naujai atėję taškai yra svarbesni negu jau trajektorijoje esantys, tuomet nauji taškai yra įterpiami senų taškų sąskaita (seni, ne tokie svarbūs taškai yra ištrinami, o trajektorija yra paslenkama, kad nebūtų tuščių vietų). Tokiu būdu trajektorija yra tobulinama, paliekant tik pačius svarbiausius ir labiausiai atspindinčius objekto judėjimo ypatumus įrašus.

Tačiau šis duomenų suspaudimo metodas pilnai neišnaudoja visų objekto judėjimo ypatumų tam, kad dar efektyviau būtų panaudota disko vieta. Pagrindinis šio metodo trūkumas yra tas, kad metodas niekada nekeičia savo elgesio, priklausomai nuo susiklosčiusios situacijos. Metodas visą laiką naudoja vienodą duomenų nuskaitymo iš judančių objektų intervalą, kuris galėtų būti keičiamas priklausomai nuo judėjimo kreivumo. Jeigu objektas pastoviai juda tiesiai ir nedaro staigių posūkių, nėra reikalo apkrauti telekomunikacijų tinklo, pastoviai užklausinėjant objektų apie esamas jų pozicijas, kadangi jų judėjimas yra prognozuojamas. Tokiu atveju būtų keičiamas duomenų nuskaitymo intervalas (tam tikra prasme yra atliekama užklauskos optimizacija), priklausomai nuo objekto judėjimo kreivumo. Jeigu objektas juda kreivai, nuskaitytinėjama dažniau, nuskaitytinėjimo intervalas yra mažinamas, jeigu objektas juda tiesiai arba

beveik tiesiai (ar juda tiesiai apsprendžia užduota paklaida), nuskaitinėjimo intervalas didinamas, nuskaitinėjama rečiau. Toks metodas galėtų įvertinti ir užklaustų duomenų vėlinimą. Jeigu užklausti duomenys apie objektą nėra gražinami, tuomet yra atliekamas spėjimas, kur objektas tuo metu galėjo būti.

Šiuo atveju yra taupoma ne tik disko vieta, atrinkinėjant tik svarbiausius posūkius atspindinčius taškus, tačiau ir tinklo apkrautumas, įvertinant poreikį nuskaityti duomenis, remiantis judėjimo prognozavimu.

Dabarties didelio tikslumo trajektorijų (GPS) suspaudimo metodo algoritmas pateiktas 18 pav.



18 pav. Dabarties didelio tikslumo trajektorijų (GPS) suspaudimo algoritmas

Kintamasis i rodo dabartinę trajektorijos poziciją. Kol vykdomas sekimas (t_{dabar} reiškia dabartinį laiką, t_{iki} reiškia laiką iki kurio objektas turi būti sekamas) atsitiktinai generuojamos judančio objekto pozicijos. Patys pirmi taškai įterpiami be apribojimų, kadangi norint apskaičiuoti SED yra reikalingi bent trys trajektorijos taškai (pagal pirmą ir trečią taškus yra skaičiuojamas SED atstumas antram taškui). Skaičiuojamas SED atstumas priešpaskutiniam taškui, remiantis naujai atėjusiu iš trajektorijų generatoriaus tašku. Priskiriamas pradinis minimalų SED turinčio taško indeksas. Tikrinama ar paskirta naudojimui atmintis dar yra laisva. Jeigu priešpaskutinio taško SED atstumas yra mažesnis už visos trajektorijos minimalų SED atstumą, toks taškas yra ignoruojamas, priešingu atveju yra atliekami taškų sukeitimo veiksmai. Trinamas trajektorijos taškas, kuris turi mažesnę svarbumą už naujai atėjusio taško. Trajektorijai priskiriama nauja pozicija, randamas trajektorijos minimalus SED atstumas. Paruošiama vieta naujai atėjusiam trajektorijos taškui.

Algoritmo pseudokodas:

```

1. i=1;
2. while  $t_{dabar} < t_{iki}$  do (OID,  $x_i$ ,  $y_i$ ,  $t_i$ )=Generuoti_nauja();
3. if (i<3) then traj[i]=(OID,  $x_i$ ,  $y_i$ ,  $t_i$ ); GOTO 10;
4. SED=SED(i-2;i-1;i);
5. if (i=3) then traj[i]=(OID,  $x_i$ ,  $y_i$ ,  $t_i$ ); min_ind=i-1; GOTO 10;
6. if (i>i_max) then GOTO 7; else GOTO 10;
7. if SED<traj[min_ind] then Mazinti_int(); GOTO 10; else Didinti_int();
8. trinti(traj[min_ind]);
9. traj[i]=(OID,  $x_i$ ,  $y_i$ ,  $t_i$ ); min_ind=rasti_min();
10. i=i+1;
11.end while

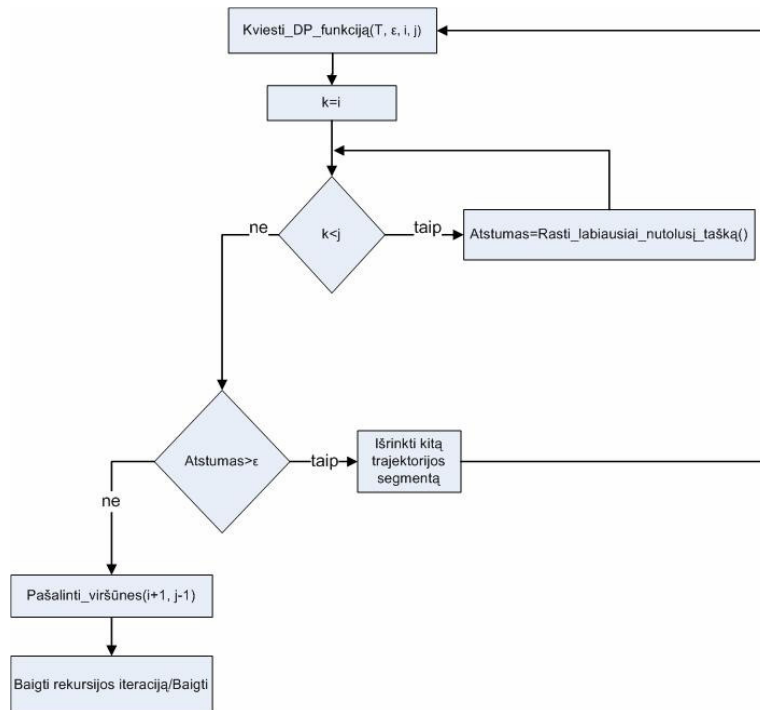
```

Judančių objektų vietos informacija gali būti apdorojama tam tikru eiliškumu pagal prioritetą, jeigu vienu metu gaunami labai dideli kiekiai informacijos. Tai gali būti realizuota sistemoje objektams pridėdant tam tikrus prioritetus, kai sistemos resursai yra labai apkrauti.

Judančių objektų trajektorijų suspaudimui galėtų būti naudojamas tik šis algoritmas, tačiau dabarties didelio tikslumo trajektorijų (GPS) suspaudimo metodas užtikrina, kad nebūtų netinkamų duomenų ir jie be reikalo neužimtų kitoms judančių objektų trajektorijoms reikalingos vietos. Todėl laikomasi strategijos šiek tiek sumažinti įeinančių duomenų kiekį, pasinaudojant dabarties didelio tikslumo trajektorijų (GPS) suspaudimo metodo algoritmu, o vėliau bėgant laikui keičiant trajektorijos prastinimo koeficientą nuolat atlaisvinti disko vietą, kad būtų galima saugoti šiuo metu naujausią ir aktualiausią informaciją.

3.2.2. Praeities didelio tikslumo trajektorijų (GPS) suspaudimo metodas

Dabarties didelio tikslumo trajektorijų (GPS) suspaudimo metodas gali apdoroti ateinančius įrašus apie judančio objekto poziciją prieš pat įterpiant juos į duomenų bazę, be to užtikrinti, kad vieno objekto sekimo metu nebus viršytas leidžiamas įrašų limitas, kuris nustatomas prieš generuojant trajektoriją. Tačiau jeigu norima trajektorijas nuosekliai spausti bėgant laikui, reikia sistemoje realizuoti algoritmą, kuris galėtų pagal įvestą tam tikrą paklaidos koeficientą suspausti tam tikrą trajektoriją ir atlaisvinti disko vietas. Šiuo tikslu gali būti panaudotas DP algoritmas, tačiau su tam tikromis modifikacijomis. Paprastas DP algoritmas įvertina tik objekto pozicijos nukrypimus, jam paduodama paklaida reiškia, kad reikia prastinti pozicijas, turinčias mažesnę nuokrypį nuo trajektorijos negu nurodo paklaidos dydis. Tačiau algoritmas vietoje to, kad įvertintų tik nuotolį, gali įvertinti ir laiką, naudodamas SED atstumus, naudojamus ir dabarties didelio tikslumo trajektorijų (GPS) suspaudimo metodo algoritme. Tačiau dabarties didelio tikslumo trajektorijų (GPS) suspaudimo metodo algoritmas įvertina trajektorijos taškų nuokrypius pagal kaimyninius taškus, o praeities didelio tikslumo trajektorijų (GPS) suspaudimo metodo algoritmas vertina trajektorijos taškų nuokrypius, atsižvelgdamas į bendrą trajektorijos vaizdą. Praeities didelio tikslumo trajektorijų (GPS) suspaudimo metodo algoritmas pateiktas 19 pav.



19 pav. Praeities didelio tikslumo trajektorijų (GPS) suspaudimo algoritmas

Algoritmas turi tokius parametrus: T – trajektorija (taškų masyvas), ϵ – trajektorijos sendinimo koeficientas, i – trajektorijos pradžios viršūnė, j – trajektorijos pabaigos viršūnė. Einama nuosekliai per įrašus trajektorijos segmente. Randama labiausiai nutolusi viršūnė nuo trajektorijos segmento. Šis nuotolis yra skaičiuojamas pagal SED radimo formules, aprašytas dabarties didelio tikslumo trajektorijų (GPS) suspaudimo metodo algoritme. Tikrinama ar rastas labiausiai nutolęs nuo trajektorijos taškas yra nutolęs per didesnę atstumą negu leidžia iš anksčiau nustatyta riba. Rekursiškai kviečiamas algoritmas, kuris apdoros pirmą trajektorijos segmentą. Rekursiškai kviečiamas algoritmas, kuris apdoros antrą trajektorijos segmentą (pirmą segmentą nuo antro atskiria viršūnė, kurioje buvo rastas maksimaliai nuo trajektorijos segmento nutolęs taškas. Trinamos visos viršūnės trajektorijos segmente. Vienos algoritmo iteracijos darbo pabaiga.

Algoritmo pseudokodas:

```
1. DP(T,  $\epsilon$ , i, j)
2. for k=i to k=j do
3.   atstumas=Rasti_labiausiai_nutolusia();
4. end for
5. if atstumas> $\epsilon$  then
6.   DP(T,  $\epsilon$ , i, k);
7.   DP(T,  $\epsilon$ , k, j);
8. else
9.   Pasalinti_virsunes(i+1, j-1);
10. return
11. end if
12. end DP
```

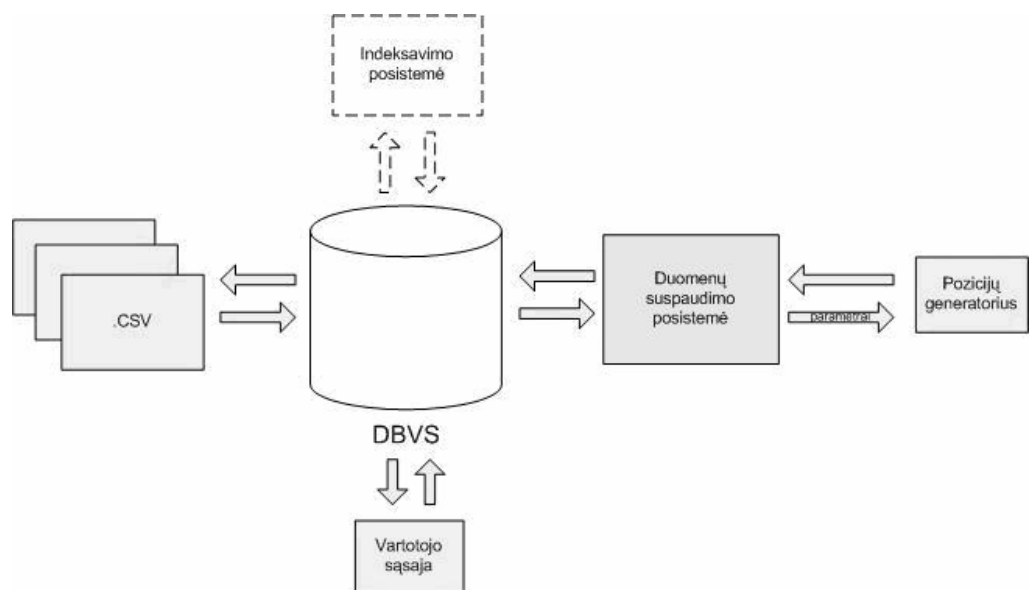
19 pav. pateiktame algoritme vaizduojamos kelios galimos baigtys. Veiksmas „Baigti“ bus vykdomas tuo atveju, jeigu rekursijos gylis bus lygus 0 (kai buvo įvykdyta paskutinė rekursijos iteracija arba rekursija iš viso nebuvo vykdoma). Veiksmas „Baigti“ bus vykdomas tuo atveju, jeigu rekursijos gylis $\neq 0$, tuomet yra baigiama viena rekursijos iteracija, o jos rezultatai naudojami tolesnėse rekursijos iteracijose.

Kuo trajektorijų suspaudimo koeficientas didesnis, tuo mažesnė bus suspaustos trajektorijos kokybė. Pačiu blogiausiu atveju (kai atminties sutaupoma maksimaliai) trajektorija turi tik du taškus (pradinį ir galinį) ir yra laikoma, kad objektas judėjo tiesia linija. Tačiau net ir toks mažas informacijos kiekis gali būti labai naudingas. Pavyzdžiui, turint trajektoriją iš kelių taškų, galima sužinoti kokioje šalyje ar kokiam mieste konkrečiu metu buvo judantis objektas.

3.3. Vietos duomenų istorijos atkūrimo sistemos modelis

Nagrinėjamas supaprastintas vietos duomenų sistemos modelis (1 pav.) yra pagrindas istorijos atkūrimo sistemai sukurti. Sistema turi galimybę apdoroti skirtingų tikslumų duomenis. Kuriamą sistemą (20 pav.) sudaro tokie komponentai:

- Duomenų bazė (DB), kurioje laikomi judančių objektų vietos duomenys reliacinėse lentelėse.
- Failai (CSV formatu), kuriais galima importuoti judančių objektų trajektorijas iš išorės (kitų sistemų), taip pat naudojami duomenų eksporto metu. Šie failai kartu su trajektorijų generatoriumi gali pakeisti Lokacijos technologiją 1 pav. modelyje.
- Duomenų suspaudimo sistema, kurioje atliekama trajektorijų suspaudimas pagal tam tikrus įdiegtus metodus.
- Vietos duomenų generatorius automatiškai ir atsitiktinai generuoja judančių objektų pozicijas taip, kad būtų galima imituoti realų sistemos veikimą, kai sistemą pasiekia iš pozicionavimo sistemų gauti duomenys.
- Vartotojo sąsajos pagalba galima tirti naudojamų duomenų suspaudimo metodų efektyvumą įvairiais aspektais, peržiūrėti originalias ir suspaustas trajektorijas trimačių grafikų pagalba.
- Indeksavimo posistemė galėtų būti naudojama, kaip sistemos išplėtimas, suteikiantis greičio užklausų vykdymui, tačiau šiame darbe neanalizuojamas.

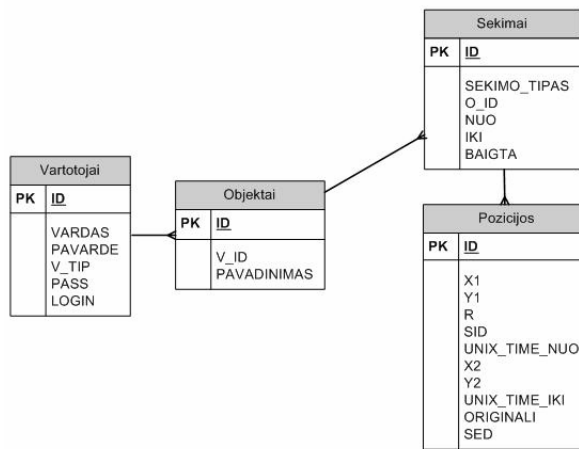


20 pav. Vietos duomenų istorijos atkūrimo sistema

3.3.1. Duomenų bazės koncepcinis modelis

Sistemoje naudojama duomenų bazė, kurioje saugoma visa su judančių objektų vieta susijusi informacija. Koncepcinis duomenų bazės modelis (21 pav.) turi keturias lenteles:

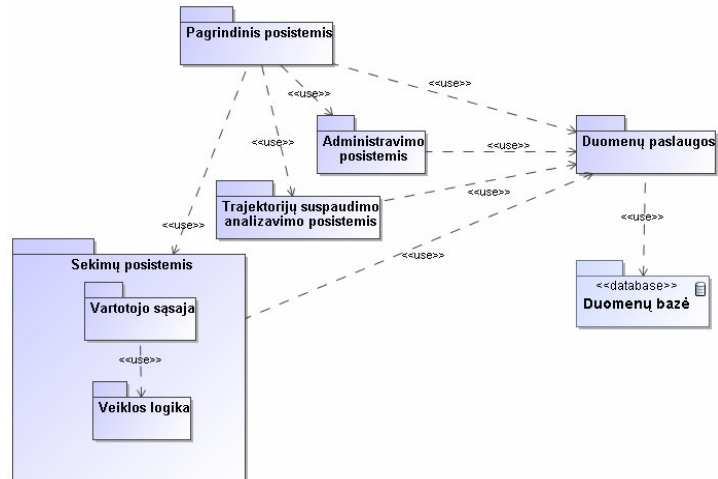
- Vartotojai – šioje lentelėje saugoma informacija, susijusi su sistemos vartotojais, kurie gali inicijuoti judančių objektų vietos stebėjimą ir peržiūrėti trajektorijų suspaudimo rezultatus.
- Objektai – vartotojų pasirinkti sekami objektai.
- Sekimai – vartotojų sukurti judančių objektų stebėjimai tam tikriems parinktiems objektams.
- Pozicijos – šioje lentelėje saugomos judančių objektų trajektorijos. Eksperimento labai yra saugomos ne tik suspaustos, tačiau ir originalios judančių objektų trajektorijos su tikslu, kad būtų galima jas palyginti ir iširti naudojamų naujai sukurtų duomenų suspaudimo metodų algoritmų efektyvumą.



21 pav. Duomenų bazės koncepcinis modelis

3.3.2. Sistemos loginė architektūra

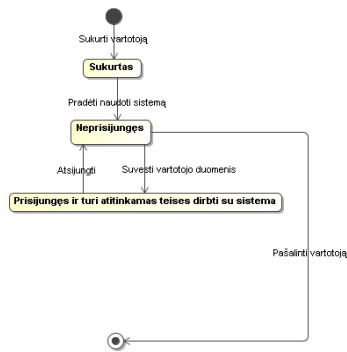
Sistemos loginė architektūra pavaizduota 22 pav. „Pagrindinis“ posistemis tik apibendrina sistemos dalis, tačiau jokie tiesioginio funkcionalumo neteikia. „Administravimo posistemis“ atsakingas už įvairių klasifikatorių turinį, t.y. vartotojų teisių priskyrimo, naujų modelių prijungimo ir kt. „Trajektorijų suspaudimo analizavimo posistemis“ atsakingas už objektų suspaudimo duomenų metodų veikimo efektyvumą, juo gali naudotis administratorius. Visi sistemos elementai susiję su duomenų baze, todėl naudoja bendrą prieigos tašką „Duomenų paslaugos“, kuris nukreipia į „Duomenų bazę“.



22 pav. Sistemos loginė architektūra

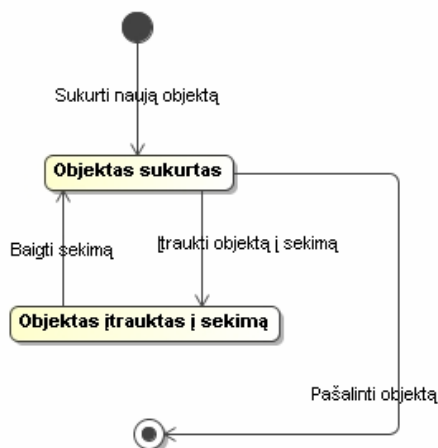
3.3.3. Objektų būsenų ir veiklos diagramos

Būsenų diagrama pateikta 23 pav. vaizduoja „Vartotojo“ gyvavimo ciklą. Kai tik „Vartotojas“ yra sukurtas, jis gali prisijungti ir naudotis sistema pagal jam suteiktas teises. Tam, kad vartotoją pašalinti iš sistemos, jis turi būti atsijungęs nuo sistemos.



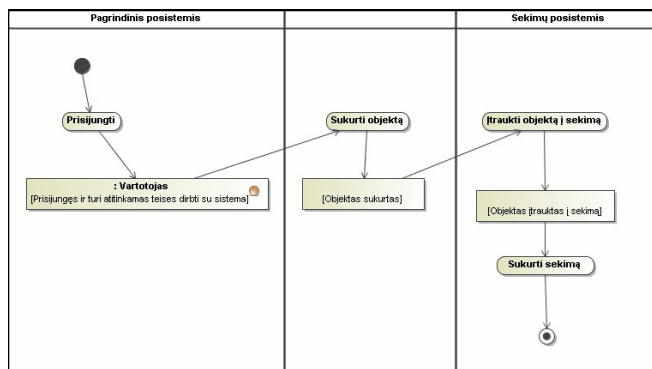
23 pav. „Vartotojas“ gyvavimo ciklas

Būsenų diagrama pateikta 24 pav. vaizduoja „Sekamas objektas“ gyvavimo ciklą. Kai „Sekamas objektas“ yra sukurtas jis gali būti įtrauktas į stebėjimą, o kai stebėjimas baigtas, objektas gali būti pašalintas.



24 pav. „Sekamas objektas“ gyvavimo ciklas

Veiklos diagrama pateikta 25 pav. rodo objekto sekimo sukūrimo procesą. Iš pat pradžių vartotojas turi prisijungti prie sistemos ir įgauti tam tikras teises, tuomet jis turi sukurti arba pasirinkti iš sąrašo stebimą objektą, jį įtraukti į stebėjimą ir sukurti patį objekto stebėjimo procesą.



25 pav. Objekto sekimo sukūrimo veiklos diagrama

3.3.4. Vietos duomenų suspaudimo metodų algoritmų realizacijos principai

Dabarties didelio tikslumo trajektorijų (GPS) suspaudimo metodo algoritmo realiam veikimo įvertinimui duomenys yra generuojami generatoriaus pagalba. Generatoriaus veikimo principas pagrįstas atsitiktinių pozicijų generavimu, naudojant pastovų iš pradžių nustatytą duomenų užklausimo intervalą. Šis intervalas gali būti keičiamas automatiškai priklausomai nuo to ar objektas juda tiesiai ar kreivai. Jeigu objektas juda tiesiai, reiškiąs intervalą galima didinti, nes objekto judėjimas tampa prognozuojamas, jeigu objektas juda kreive, tuomet duomenų generavimo intervalas turi būti mažinamas, kad nebūtų prarasta informacija. Tokia intervalo

keitimo galimybę sistemoje galima išjungti ir palikti duomenų nuskaitymo intervalą pastovų. Jeigu duomenų nuskaitymo intervalas pastovus, o atsitiktinai keičiamas nueito atstumo dydis, vadinasi objekto judėjimo greitis yra atsitiktinis ir objektas judėdamas nueina vis skirtingus nuotolius. Laikomasi principo, kad objektas pastoviai juda, toliau generatorius atsitiktinai sugeneruoja objekto poziciją iš tam tikro reikšmių intervalo. Generatorius naudojamas, kad būtų kuo realiau (tikroviškiau) atkartotas vietos duomenų gavimo iš judančio objekto procesas. Duomenys iš generatoriaus toliau paduodami apdorojimui dabarties didelio tikslumo trajektorijų (GPS) suspaudimo metodo algoritmui. Algoritmo pagalba šiek tiek suspaustą trajektoriją toliau galima apdoroti praeities didelio tikslumo trajektorijų (GPS) suspaudimo metodo algoritmu daug kartų, kiekvieną kartą vis keičiant paklaidos dydį.

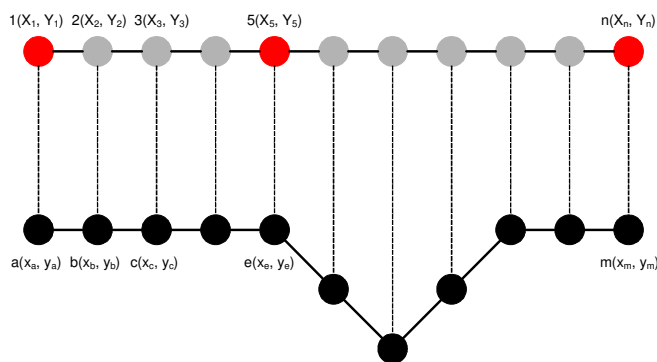
Generatorius yra naudojamas ne tik dabarties didelio tikslumo trajektorijų (GPS) suspaudimo metodo algoritmo apdorojamoms trajektorijoms gauti, tačiau ir praeities didelio tikslumo trajektorijų (GPS) suspaudimo metodo algoritmo apdorojamoms trajektorijoms gauti. Tačiau šiuo atveju generatoriaus veikimo principas yra šiek tiek kitoks. Skirtumas yra tame, kad iš pradžių atsitiktinai yra sugeneruojamas įvykis ar objektas pakeitė savo poziciją. Jeigu objektas pozicijos nepakeitė, tolesnis generavimas yra praleidžiamas ir laukiama tiek laiko, koks yra dabartinis duomenų nuskaitymo intervalas. Tačiau, jeigu sugeneruojamas įvykis, kad objektas pajudėjo yra atsitiktinai generuojamos sekančio trajektorijos taško koordinatės ir celės spindulys. Tai yra daroma todėl, kad GSM technologijos pagalba gaunamos pozicijos gali kartotis, o skirtis tik laiko momentai, kada jos gautos, tai yra todėl, kad GSM yra ne taip tiksliai nustatanti technologija kaip GPS, GSM technologijos pagalba nustatyta pozicija yra panaši į celę (apskritimą), todėl kol objektas pereis į kitą celę, gali būti padaryti keli įrašai toje pačioje celėje.

Tam, kad išbandyt praeities didelio tikslumo trajektorijų (GPS) suspaudimo metodo algoritmo veikimą, galima paimti trajektoriją, gautą iš generatoriaus ir apdorotą dabarties didelio tikslumo trajektorijų (GPS) suspaudimo metodo algoritmu arba galima importuoti iš CSV formato failo. Po apdorojimo šiuo algoritmu, trajektoriją galima eksportuoti atgal į CSV formato failą, kad būtų užtikrintas duomenų apsikeitimas su kitomis sistemomis. Analogiškas duomenų importavimo ir eksportavimo mechanizmas yra realizuotas ir praeities didelio tikslumo trajektorijų (GPS) suspaudimo metodo algoritmo apdorojamoms trajektorijoms.

4. Vietos duomenų istorijos atkūrimo efektyvumo tyrimas

4.1. Duomenų su pastoviu laiko intervalu suspaudimo efektyvumo tyrimas

Jeigu vietos duomenų gavimo intervalai vienodi ir nereikia atsižvelgti į laiko dedamąją, duomenų suspaudimo efektyvumas gali būti iširtas užpildant trūkstamas koordinatas suspaustoje trajektorijoje pagal originalią trajektoriją ir lyginant skirtumus tarp jų. Šioje dalyje yra atliekama analizė, kurios rezultatai priklauso nuo sekimo metu naudoto duomenų suspaudimo modelio. Tam, kad būtų galima atlikti tokio pobūdžio analizę yra saugomi originalūs trajektorijos duomenys ir trajektorijos duomenys po duomenų atrinkimo modelio taikymo. Kadangi taikant duomenų atrinkimo modelį siekiama sumažinti esamų duomenų kiekį, po tokio modelių pritaikymo trajektorija turi mažiau taškų. Tam, kad būtų galima sulygtinti gautą trajektoriją su originalia trajektorija, gauta trajektorija turi būti dirbtinai užpildoma taškais, kuriuos galima būtų lyginti su originalios trajektorijos taškais (kiekvienas taškas originalioje trajektorijoje yra lyginimas su kiekvienu tašku gautoje trajektorijoje). 26 pav. yra vaizduojamos dvi trajektorijos: originali trajektorija, sudaryta iš raidėmis pažymėtų taškų, suspausta trajektorija sudaryta iš taškų, kurie yra pažymėti skaičiais. Gautoje trajektorijoje yra trys taškai: $1(X_1, Y_1)$, $5(X_5, Y_5)$ ir $n(X_n, Y_n)$. Kadangi originalioje trajektorijoje taškų yra daugiau, reikia gautą trajektoriją papildyti dirbtinai sugeneruotais taškais.



26 pav. Originali ir gauta trajektorijos

Kiekvienam intervalui (pvz., nuo $1(X_1, Y_1)$ iki $5(X_5, Y_5)$) yra paskaičiuojamas koordinatės pokytis, kuris yra vėliau padalinamas iš koordinatės pokyčių kiekio:

$$\Delta X = \frac{X_5 - X_1}{4}; \Delta Y = \frac{Y_5 - Y_1}{4}; \quad (10)$$

Tuomet dirbtinai sugeneruojami taškai prie pradinio taško pridant atitinkamą kiekį koordinatinių pokyčių:

$$X_2 = X_1 + \Delta X * 1; Y_2 = Y_1 + \Delta Y * 1; X_3 = X_1 + \Delta X * 2; Y_3 = Y_1 + \Delta Y * 2; \quad (11)$$

Po trajektorijos užpildymo dirbtiniais taškais yra skaičiuojamas vidutinis kvadratinis nuokrypis, kuris rodo kiek gauta trajektorija skiriasi nuo originalios trajektorijos:

$$\sigma_{pastoviam_int} = \sqrt{\sum_{i=1}^n (X_i - x_i)^2 (Y_i - y_i)^2} \text{ (RMSE);} \quad (12)$$

Kuo šis nuokrypis yra didesnis, tuo duomenų atrinkimo metodas turi mažesnę efektyvumą, kadangi metodo apdorota trajektorija tampa išdaryta ir nebegalima atkurti originalios trajektorijos. Kadangi vienas duomenų atrinkimo metodas gali būti panaudotas daugelyje sekimų reikia atsižvelgti į sekimų ilgius:

$$\sigma_{bendras} = \frac{\sigma_i * n_i}{N}, N = \sum n_i; \quad (13)$$

kur $\sigma_{bendras}$ - bendras duomenų atrinkimo metodo efektyvumas, σ_i - vieno sekimo efektyvumas, n_i - vieno sekimo trajektorijos taškų skaičius, N - visų sekimų bendras trajektorijų taškų skaičius. Sutaupytas vietos kiekis yra apskaičiuojamas:

$$E = \frac{M - m}{M} * 100\%; \quad (14)$$

kur m – gautos trajektorijos taškų skaičius, M – originalios trajektorijos taškų skaičius (kiekis). Jeigu netaikomas joks duomenų atrinkimo metodas, tačiau tiesiog imamas kas kažkelintas taškas, susidariusi paklaida apskaičiuojama taip:

$$P = \frac{e}{10} * 100\% , \text{ kur } e - \text{ praleistų taškų } 10 - \text{ čiai taškų skaičius.} \quad (15)$$

Toks metodo efektyvumo skaičiavimas tinka tik tuo atveju, jeigu trajektorija susideda iš taškų (labai tiksliai nustatytos pozicijos, kurią vaizduoja taškas plokštumoje). Taigi jis tinka tik GPS technologijos atveju ir jeigu laikysime, kad trajektorija susideda iš taškų, o ne iš apskritimų. Turint omenyje, kad kitos technologijos grąžina ne tašką, bet tam tikrą geometrinę figūrą, kaip abonento buvimo vietą, iš pradžių turi būti atlikti tam tikri pakeitimai aprašomi vėlesniuose poskyriuose ir tik tuomet taikomas toks efektyvumo nustatymo metodas.

4.2. Duomenų su kintančiu laiko intervalu suspaudimo efektyvumo tyrimas

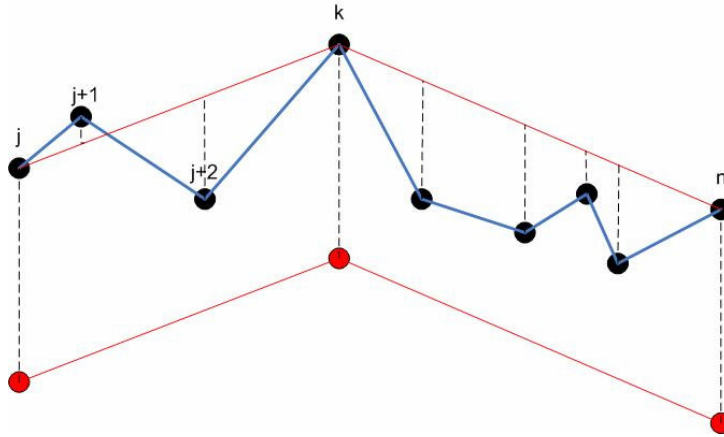
Aukščiau aprašytas duomenų suspaudimo su pastoviu intervalu tyrimas yra gana paprastas ir trivialus, tačiau jis netinka tuo atveju, jeigu duomenys yra nuskaityti skirtingais intervalais ir reikia įvertinti laiko dedamąją. Šiuo atveju reikia galvoti apie atstumo erdvėje radimą tarp originalios ir po suspaudimo gautos trajektorijos taškų. Turint omenyje, kad kiekvienas gautos trajektorijos taškas buvo ir originalioje trajektorijoje, o taškai jau esančios trajektorijos viduje neatsiranda, o tik ištrynus senus, pridėdami prie galo, galime rasti paklaidas kiekvienam trajektorijos segmentui, paskui bendrą visos trajektorijos paklaidą. Tarkime, kad:

$1 \leq i \leq n$ (originalios trajektorijos taškų skaičius n , einamasis taškas i);

$j \leq k$ (k - indeksas to taško, kuris originalioje ir gautoje trajektorijoje sutampa, j – to segmento (kuriame buvo atliekamas prastinimas) einamasis taškas (pradžioje $j=1$));

m – intervalų, kuriuose yra atliekamas prastinimas, skaičius.

Nuosekliai einama per originalios trajektorijos taškus, kol randamas taškas, kuris yra ir gautoje trajektorijoje (indeksas k) (27 pav.).



27 pav. Duomenų suspaudimo su kintančiu intervalu efektyvumo tyrimas

Tuomet nuo j iki k intervale atliekame skaičiavimus, gaunama paklaida vienam intervalui:

$$SED(j, j+1, k) = \sqrt{(x_{j+1} - x'_{j+1})^2 + (y_{j+1} - y'_{j+1})^2}, \text{ čia:} \quad (16)$$

$$x'_{j+1} = x_j + v_{j,k}^x \cdot (t_{j+1} - t_j), \text{ čia: } v_{j,k}^x = \frac{x_k - x_j}{t_k - t_j}; \quad (17)$$

$$y'_{j+1} = y_j + v_{j,k}^y \cdot (t_{j+1} - t_j), \text{ čia: } v_{j,k}^y = \frac{y_k - y_j}{t_k - t_j}; \quad (18)$$

$$\sigma_{k \text{ int } amam_int} = \frac{1}{n-2} \cdot \sum_{j=1}^{k-2} SED(j, j+1, k), \text{ čia:} \quad (19)$$

n yra sumažintas per 2, kadangi SED'ai skaičiuojami tik viduriniams taškams.

Kai pereinama prie kito intervalo $j=k_{senas}$; $k=k_{naujas}$ (šiuo atveju n). Kai visų intervalų paklaidos jau yra įvertintos, randama bendra trajektorijos paklaida. Kadangi bendra trajektorijos paklaida turi būti skaičiuojama ne tik įvertinant taškus, kurie buvo intervaluose, tačiau ir sutampančius taškus (sutampantys taškai mažins bendrą trajektorijos paklaidą), ji turi būti skaičiuojama taip:

$$\sigma_{k \text{ int } amam_int_bendr} = \frac{1}{n} \sum_1^m \sigma_{k \text{ int } amam_int}. \quad (20)$$

Jeigu būtų įvertinti tik intervaluose esantys taškai tuomet būtų surastas vidutinis momentinis nuokrypis nuo trajektorijos. Ši paklaida turi būti perskaičiuojama po kiekvieno trajektorijos prastinimo, lyginant originalią trajektoriją su gauta trajektorija. Kadangi praeities didelio tikslumo trajektorijų (GPS) suspausimo metodo algoritmo darbo metu skaičiuojama paklaida gali būti paskaičiuota tik naikinamiems taškams, ji negali būti paskaičiuota pirmame etape sunaikintiems taškams, todėl ji turi būti pilnai perskaičiuojama po kiekvieno duomenų suspaudimo, kad būtų įvertinti ir anksčiau pašalinti taškai.

Kiek iš viso yra sutaupyta įrašų (procentais) algoritmo darbo metu yra apskaičiuojama taip:

$$n_{sutaupyta} = \frac{n_{originalios} - n_{suprastintos}}{n_{originalios}} \cdot 100\% ; \quad (21)$$

(21) formulė turi apribojimą, kad trajektorija negali būti sudaryta iš nulio taškų ($n_{originalios} \neq 0$). Trajektorijos suspaudimo kokybės koeficientas parodo algoritmo darbo efektyvumą tam tikrais momentais ir yra apskaičiuojamas taip:

$$k_{kb} = \frac{n_{sutaupyta}}{\sigma_{k_{int} amam_int_bendr}} \cdot 100\% ; \quad (22)$$

(22) formulė turi apribojimą, kad bendra trajektorijos paklaida negali būti lygi 0 ($\sigma_{k_{int} amam_int_bendr} \neq 0$). Taip gali atsitikti tuomet, kai trajektorija yra nei kiek nesuspausta, tuomet kokybės koeficientas (k_{kb}) artės į begalybę.

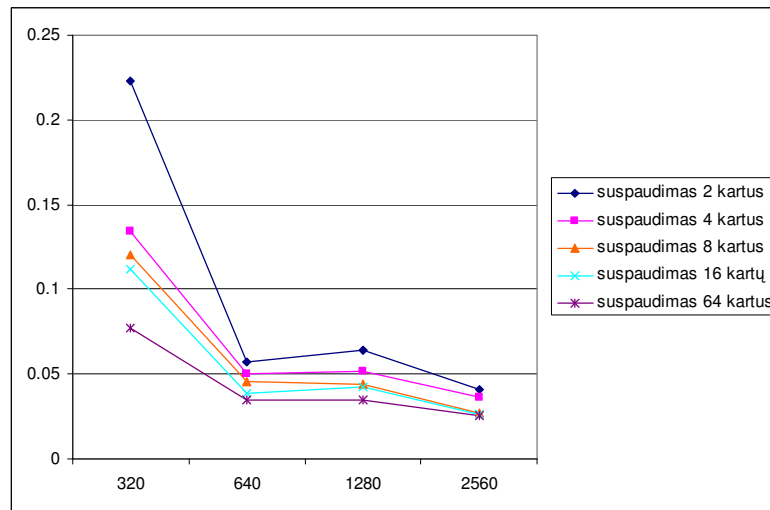
4.3. Dabarties didelio tikslumo trajektorijų (GPS) suspaudimo metodo algoritmo eksperimentinis įvertinimas

Tam, kad būtų galima įvertinti algoritmo darbą, sistemoje turi būti realizuotas trajektorijų generavimas. Trajektorijų generavimas yra integruota sistemos dalis. Trajektorijas galima generuoti, imituojant realius vietos duomenis, ateinančius iš vietos nustatymo technologijos. Čia yra nagrinėjamas atvejis, kai objektas juda visą laiką į priekį. Iš pradžių yra laikoma, kad objektas yra koordinatinių sistemos pradžioje. Generatoriaus laiko periodas yra nustatomas vartotojo (pvz., kas 10 sekundžių bus sugeneruotas naujas trajektorijos taškas). Prie kiekvienos pradinės koordinatės (x, y) yra pridedamos atsitiktinės reikšmės iš intervalų (pvz., [0-100]). Šis veiksmas

yra kartojamas daug kartų, kol yra gaunama norimo ilgio trajektorija. Jeigu ir x , ir y koordinatėms atsitiktinai bus parinktos intervalo reikšmės, kurios lygios 0, tai reikš, kad objektas nepajudėjo. Kadangi sistema yra realizuota, naudojant *PHP* programavimo kalbą, atsitiktiniam reikšmės parinkimui buvo parinkta funkcija *rand()*, grąžinanti reikšmes, turinčias vieningą pasiskirstymą (*uniform distribution*). Tai reiškia, kad išrinkimo iš intervalo tikimybė visoms reikšmėms yra vienoda, todėl galima teigti, kad yra generuojama visiškai atsitiktinė judančio objekto trajektorija, paremta vieningu pasiskirstymu. Tokiu būdu buvo sugeneruota 20 trajektorijų. Kadangi trajektorijos generuojamos naudojant tuos pačius generatoriaus parametrus (tokį patį duomenų generavimo intervalą ir tokį patį intervalą, iš kurio atsitiktinai parenkama objekto nueito atstumo reikšmė), jos savo ilgiu ir suspaudimo kokybe turi būti panašios. Jeigu generatoriaus sugeneruotos trajektorijos taškai būtų labai išsibarstę, tuomet nebūtų įmanoma tiksliai apskaičiuoti trajektorijos kokybės, kadangi rezultatai įtakotų ne taškų skirtumas tarp originalios ir suspaustos trajektorijos, tačiau atsitiktiniai nutolę taškai. Kadangi judantis objektas dažniausiai juda įvairiais greičiais yra daroma prielaida, kad trajektorijas reikia generuoti atsitiktinai parenkant judančio objekto nueitus atstumus. Kiekvieno objekto sekimo (trajektorijos generavimo) metu yra naudojamas tam tikras trajektoriją sudarančių taškų limitas (šis limitas reiškia maksimalų įrašų kiekį, kuris yra skirtas vienam sekimui). Kadangi sistemos darbo metu gali būti parinkti įvairūs trajektoriją sudarančių taškų limitai, trajektorijos yra generuojamos, naudojant skirtingus taškų limitus (5 įrašai, 10 įrašų, 20 įrašų vienai trajektorijai). Šis leidžiamas įrašų kiekis iš pradžių yra užpildomas be jokių apribojimų. Tačiau kai šis trajektoriją sudarančių taškų limitas yra viršytas, prasideda duomenų atrinkinėjimo procesas, kuris reikalingas tam, kad nebūtų išnaudota kitam vartotojo sekimui reikalinga atmintis. Duomenų atrinkimo procese naudojamas dabarties didelio tikslumo trajektorių (GPS) suspaudimo metodo algoritmas, kuris nusprendžia ar naujai atėję taškai yra svarbesni už jau trajektorijoje esančius taškus. Dabarties didelio tikslumo trajektorių (GPS) suspaudimo metodo algoritmas yra taikomas todėl, kad prieš terpiant judančių objektų pozicijas į judančių objektų duomenų bazę, būtų duomenys filtruojami, kadangi ne visi atėję taškai turi tokį patį informacijos kiekį. Šio algoritmo panaudojimas leidžia apriboti trajektorijos taškų kiekį. Tai yra daroma todėl, kad esant disko vietos trūkumui nebūtų iš viso draudžiama sekti objektus, tačiau būtų galima juos sekti su mažesniu tikslumu. Algoritmo darbas baigiamas tuo metu, kai baigiamas pats objekto sekimas. Tuomet yra apskaičiuojami sekimą įvertinantys rezultatai. Bendra trajektorijos paklaida $\sigma_{kintanam_int_bendr}$ parodo kiek suspausta objekto judėjimo trajektorija vidutiniškai skiriasi nuo originalios trajektorijos ir yra paskaičiuojama *duomenų suspaudimo su kintančiu intervalu efektyvumo tyrimo* metu.

Trajektorijos kokybės koeficiento priklausomybės nuo trajektorijos ilgio įvairiems suspaudimo laipsniams grafikai yra pateikti 28 pav. Suspaudimo laipsnis yra apskaičiuojamas pagal formulę:

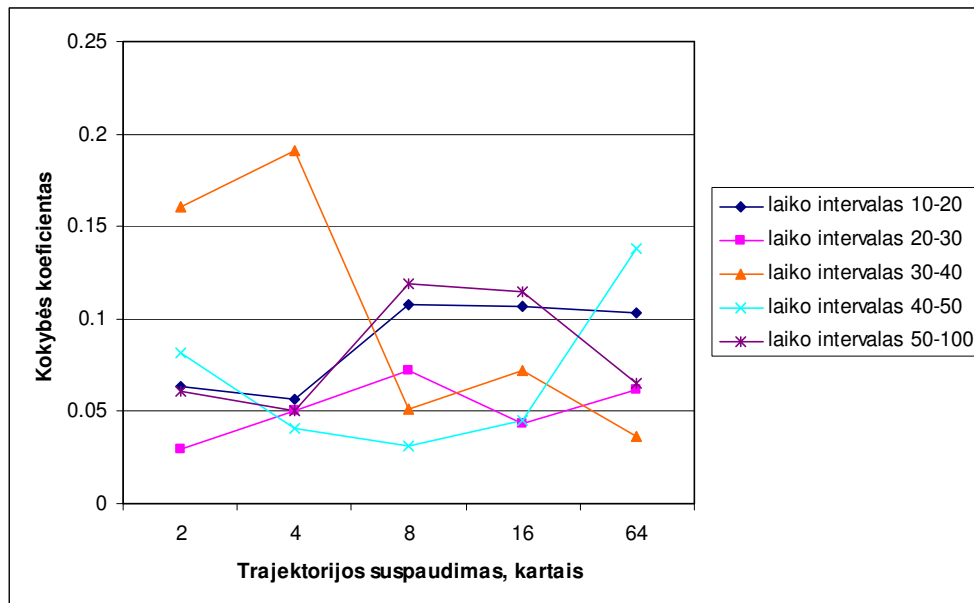
$$r = \frac{\text{trajektorijos_taškų_kiekis}}{\text{leistinas_įrašų_kiekis}}; \quad (23)$$



28 pav. Kokybės koeficiento priklausomybė nuo trajektorijos ilgio

Nuosekliai didinant suspaudimo laipsnį, kokybės koeficientas mažėja, kad ir kiek trajektorija turėtų taškų. Jeigu iš viso buvo sugeneruota 320 taškų (originalios trajektorijos dydis), o atminties buvo skirta tik 160 įrašų limitas, tuomet trajektorija buvo suspausta 2 kartus ir yra tik 160 sutampančių taškų originalioje trajektorijoje ir atmintyje. Likę 160 taškų (kurių nėra atmintyje) didina bendrą trajektorijos paklaidą ir tokiu būdu mažina kokybės koeficiento reikšmę. Jeigu įrašų limitas yra lygus originalios trajektorijos dydžiui, tuomet kokybės koeficientą skaičiuoti nėra korektiška, kadangi visi atėję taškai be jokio atrinkimo buvo patalpinti į duomenų bazę ir trajektorija iš viso nebuvo spaudžiama. Iš grafiko galima padaryti išvadą, kad didėjant trajektorijos ilgiui, kokybės koeficiento skirtumai, naudojant įvairius suspaudimo laipsnius, tampa mažesni. Ateinančius trajektorijos taškai yra paskirstomi dviejų tipų intervaluose. Dalis taškų yra priskiriami pirmajam intervalo tipui (leistinam įrašų skaičiui), kita dalis taškų yra priskiriama antrajam intervalo tipui (reikšmės, kurios nepatenka į leistinų įrašų skaičių). Tarkime, jeigu suspaudimas yra 2 kartai, o taškų yra 320, tuomet pirmojo tipo intervalo ilgis yra 160, antrojo irgi 160. Jeigu suspaudimas yra 4 kartai, tuomet pirmojo tipo intervalo ilgis yra 80, antrojo ilgis 240. Ties mažu taškų skaičiumi (320) intervalų, į kuriuos nepatenka reikšmės, ilgiai yra 160, 240, 280,

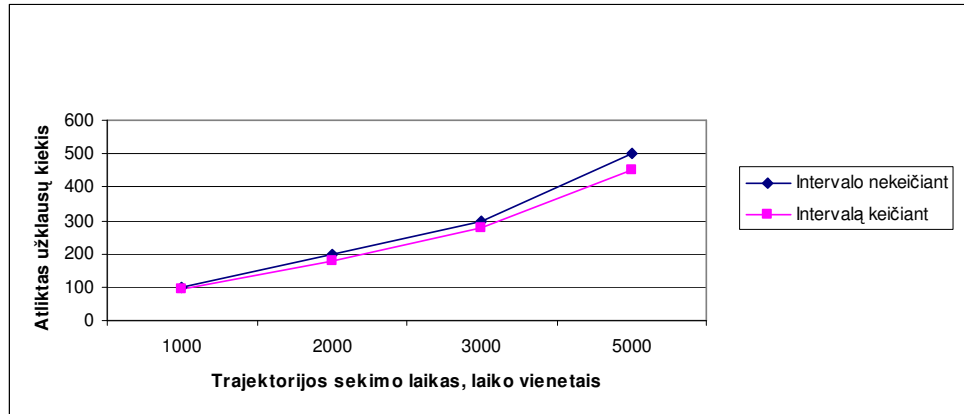
300, 315. Kai tuo tarpu prie didesnio taškų skaičiaus (640), antrojo tipo intervalų dydžiai yra 320, 480, 560, 600, 630. Kuo trumpesni intervalai, tuo jame esančios reikšmės daugiau skiriasi, vėliau intervalams didėjant, reikšmės tampa mažiau išsibarsčiusios, todėl koeficientai tampa vienodesni.



29 pav. Kokybės koeficiento priklausomybė nuo trajektorijos ilgio

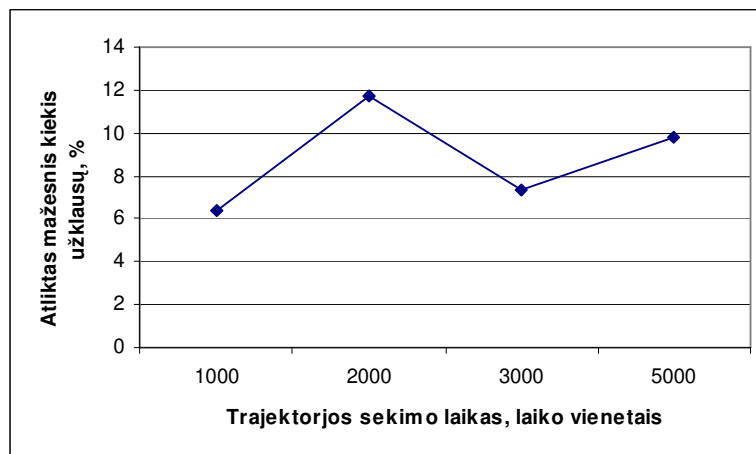
Kitas tyrimas (29 pav.) buvo atliktas, atsitiktinai keičiant duomenų nuskaitymo intervalą (intervalo reikšmė buvo parenkama atsitiktinai iš tam tikros reikšmių aibės (nuo 10 iki 20, nuo 20 iki 30 ir pan.). Remiantis 29 pav. grafiko duomenimis galima padaryti išvadą, kad koordinatės pokytis turi didesnę įtaką algoritmo darbui negu intervalo pokytis, todėl kokybės koeficientas mažiau priklauso nuo laiko intervalo negu nuo koordinatės pokyčio.

Algoritmas gali būti panaudotas net tik trajektorijos taškų suspaudimui, tačiau dėl galimybės įvertinti objekto judėjimo kreivumą, gali būti panaudotas ir šiek tiek kitokiems tikslams – užklausos optimizavimui ir užklausų apie objekto pozicija kiekio mažinimui. Turint omenyje, kad dabarties didelio tikslumo trajektorijų (GPS) suspaudimo metodo algoritmas priklausomai nuo to, kaip juda objektas, tiesiai (prognozuojamai) ar kreivai gali keisti intervalą tarp objekto vietos užklausų, buvo atliktas tyrimas kiek vidutiniškai galima sumažinti užklausų kiekį. Grafike (30 pav.) yra pavaizduotos dvi funkcijos. Aukščiau esanti funkcija rodo užklausų skaičių tuo atveju jeigu algoritmas naudotų pastovų duomenų nuskaitymo intervalą ir jo nekeistų. Žemiau esanti funkcija rodo mažesnę atliktų užklausų kiekį, jeigu algoritmas naudoja duomenų nuskaitymo iš judančio objekto intervalo keitimą, atsižvelgiant į objekto judėjimo prognozes.



30 pav. Kokybės koeficiento priklausomybė nuo trajektorijos ilgio

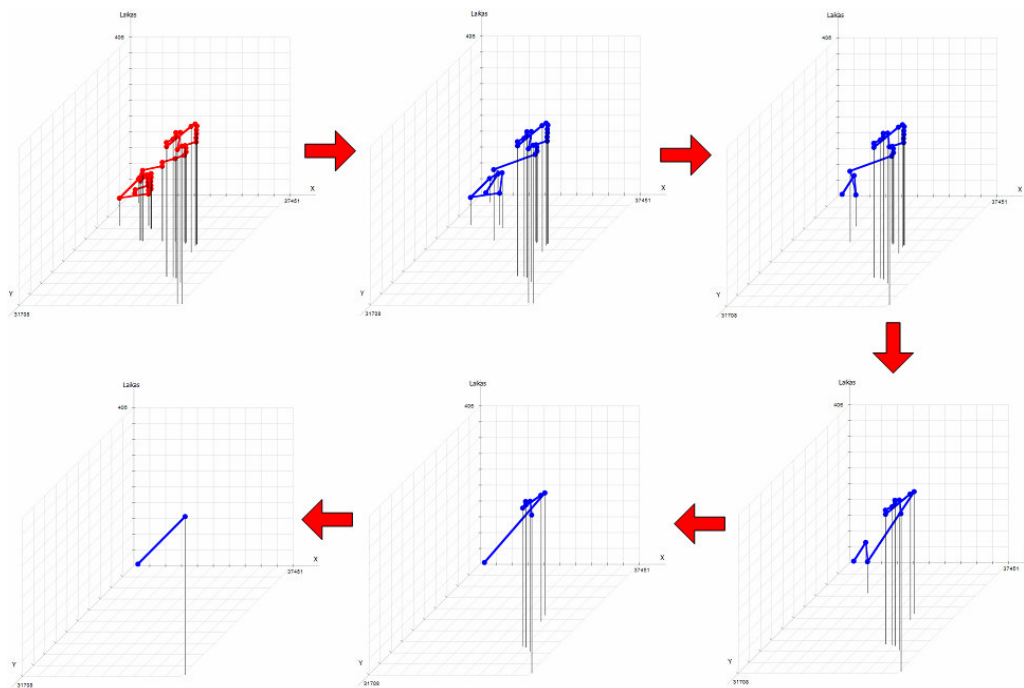
31 pav. pateiktame grafike yra matoma, kiek galima sumažinti užklausų kiekį (procentais), jeigu algoritme yra naudojamas judančio objekto duomenų nuskaitymo intervalo, keitimas. Funkcijos reikšmės yra pasiskirstę netolygiai, kadangi trajektorijos yra generuojamos atsitiktinai, tačiau, kad ir koks būtų trajektorijos sekimo laikotarpis, galima sutaupyti daromų užklausų kiekį. Įvykęs didesnis objekto judėjimo nukrypimas (keičiamas intervalas) gali smarkiai sąlygoti ateities rezultatus. Taip yra todėl, kad generatoriaus intervalo keitimas yra inertiškas ir ne iš karto atstato reikiamą intervalo reikšmę. Todėl toks intervalo keitimas yra naudingesnis, jeigu objektas daro ne staigius judesius, o palengva daro posūkį (nukrypsta nuo tiesaus kurso). Atsižvelgiant į gautus rezultatus gali būti padaryta išvada, kad ilgesnių judėjimo intervalų buvo daugiau negu trumpesnių, o kai intervalai yra ilgesni, tuomet ir mažiau taškų yra tokio paties ilgio trajektorijoje.



31 pav. Kokybės koeficiento priklausomybė nuo trajektorijos ilgio

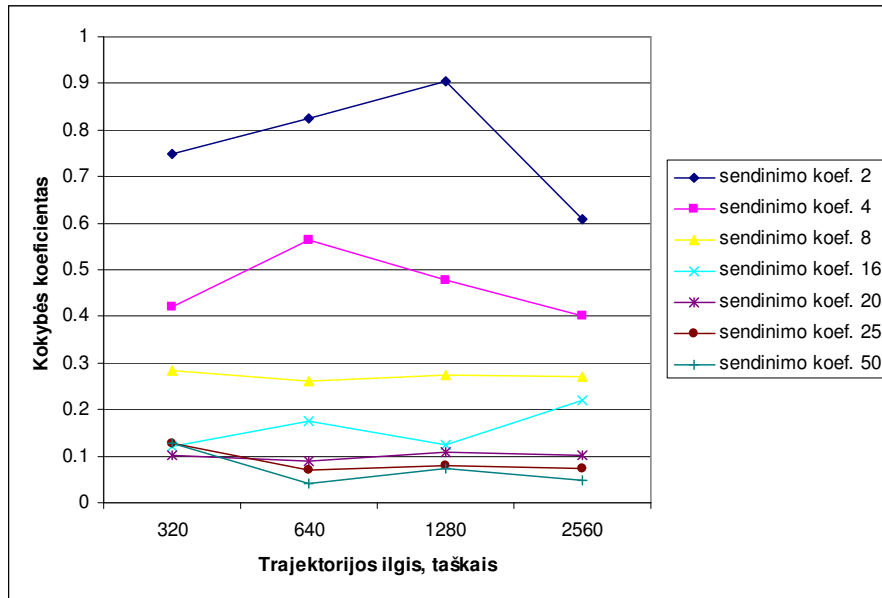
4.4. Praeities didelio tikslumo trajektorijų (GPS) suspaudimo metodo algoritmo eksperimentinis įvertinimas

Kai judančio objekto sekimas yra baigtas, duomenų bazėje išsaugomos dvi jo judėjimo trajektorijos. Pirmoji yra originali, apimanti visus be išimties atėjusius taškus, antroji - apdorota dabarties didelio tikslumo trajektorijų (GPS) suspaudimo metodo algoritmu. Originalioji trajektorija yra saugoma todėl, kad būtų galima patikrinti algoritmo veikimo efektyvumą. Kai judančio objekto trajektorija jau yra duomenų bazėje, o sekimas yra baigtas, tuomet yra naudinga taikyti *trajektorijų suspaudimo strategiją*. Iš pradžių duomenų bazėje yra saugoma tik dabarties didelio tikslumo trajektorijų (GPS) suspaudimo metodo algoritmo apdorota trajektorija, tačiau ją suspausti galima nuosekliai panaudojant praeities didelio tikslumo trajektorijų (GPS) suspaudimo metodo algoritmą. Tokiu būdu atmintis atlaisvinama trajektorijos kokybės sąskaita. Trajektorijos kokybė reguliuojama naudojant *suspaudimo koeficientą*. Spaudžiamos trajektorijos grafinis vaizdas yra pateikiamas 32 pav.



32 pav. Trajektorijos suspaudimo nuosekliai didinant paklaidos koeficientą

Trajektorija spaudžiama nuosekliai didinant prastinimo koeficientą, tokiu būdu gaunami vis skirtingi algoritmo darbo efektyvumo įvertinimai (33 pav.).

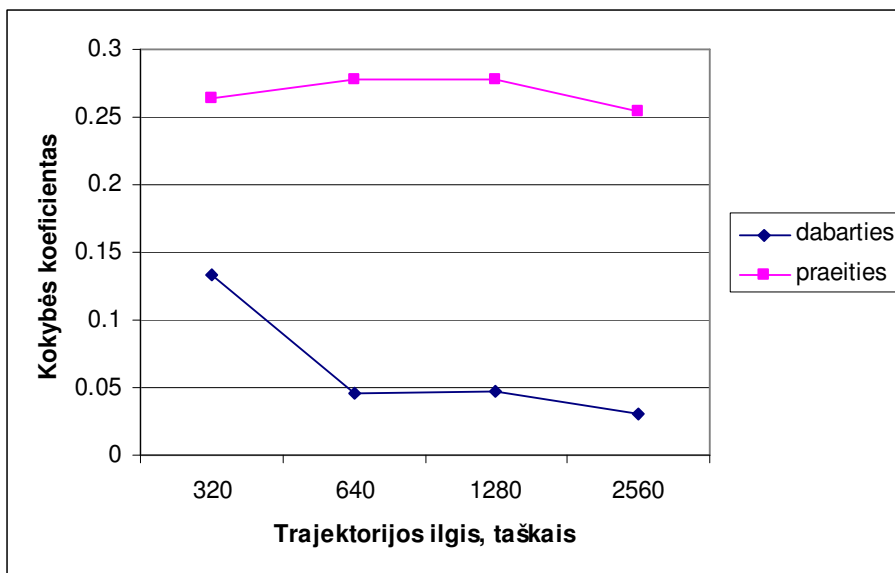


33 pav. Kokybės koeficiento priklausomybė nuo trajektorijos ilgio

Iš grafiko galima spręsti, kad geriausias kokybės koeficientas yra tuomet, kai taikomas mažiausias suspaudimo koeficientas. Nuolat didinant suspaudimo koeficientą iš trajektorijos pašalinama vis daugiau taškų, o tai sąlygoja pablogėjusią trajektorijos kokybę. Iš kitos pusės, didinant koeficientą yra sutaupoma vis daugiau įrašų, tačiau vieta taupoma yra ne taip greitai, kaip prastėja suspaustos trajektorijos kokybė. Egzistuoja galimybė, kad esant tam tikriems suspaudimo koeficientams, kai kuriais laiko momentais kokybės koeficiento negalima paskaičiuoti (jis artėja į begalybę). Tai gali atsitikti jeigu su tam tikru parinktu suspaudimo koeficientu nėra pašalinamas nei vienas trajektorijos taškas. Tokios situacijos dažniausiai gali atsitikti trajektorijoms, kai jose visi taškai yra nutolę per didesnius atstumus (SED) negu užduotas suspaudimo koeficientas. Kuo mažesnis trajektorijos ilgis, tuo mažiau išsibarstę jos taškai (tuo daugiau yra įvairesnių objekto judėjimo atkarpų, kadangi generatorius atsitiktinai keičia objekto judėjimo greitį), todėl ties kai kuriais paklaidos koeficientais tiesiog nėra ką suspausti. Visi taškai yra susispietę aplink panašias reikšmes yra pašalinami pritaikius kokį nors vieną paklaidos koeficientą. Kitaip tariant, jeigu trajektorija yra trumpa, objektas daug nespėja nukrypti nuo tiesaus kurso, todėl jo trajektorijai didelės prastinimo paklaidos nėra aktualios. Šiuo atveju galima padaryti išvadą, kad algoritmo darbas tiesiogiai nuo trajektorijos ilgio nepriklauso, žymiai didesnė priklausomybė yra nuo suspaudimo koeficiento keitimo.

4.5. Dabarties ir praeities didelio tikslumo trajektorijų (GPS) metodų algoritmų efektyvumo palyginimas

Sekančio tyrimo objektas yra dabarties ir praeities didelio tikslumo trajektorijų (GPS) algoritmų darbo palyginimas. Grafikams brėžti yra panaudotų anksčiau gautų reikšmių vidurkiai, ties vienodomis padalomis (trajektorijos ilgiu, taškais). Iš gautų statistinių rezultatų (34 pav.) yra matoma, kad praeities didelio tikslumo trajektorijų (GPS) suspaudimo metodo algoritmo darbas yra efektyvesnis bet kokiam trajektorijos ilgiui. Tačiau čia grafikai, patvirtindami praeities didelio tikslumo trajektorijų (GPS) suspaudimo metodo algoritmo geresnę veikimą, nereiškia, kad reikia atsisakyti dabarties didelio tikslumo trajektorijų (GPS) suspaudimo metodo algoritmo, kadangi jų tikslai skiriasi iš esmės. Dabarties didelio tikslumo trajektorijų (GPS) suspaudimo metodo algoritmas geriau tinka tuo atveju, kai ilgu laikotarpiu reikia garantuoto likusio disko vietos kiekio, kai tuo tarpu praeities didelio tikslumo trajektorijų (GPS) suspaudimo metodo algoritmas tinka tuomet, kai reikia garantuotos trajektorijos kokybės, kadangi ji yra valdoma užduodamo suspaudimo koeficiento pagalba.

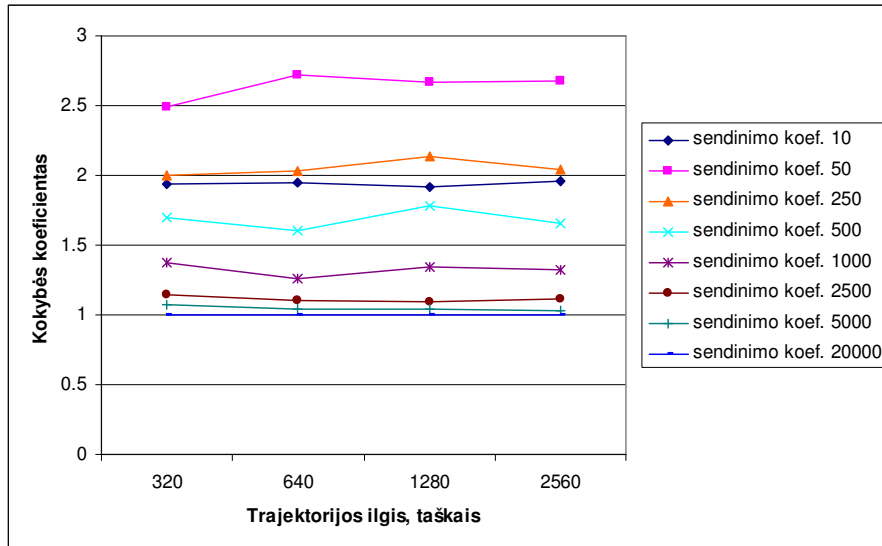


34 pav. Dabarties ir praeities didelio tikslumo trajektorijų (GPS) metodų algoritmų darbo efektyvumo palyginimas

4.6. Praeities mažo tikslumo trajektorijų (GSM) suspaudimo metodo įvertinimas

Šio metodo efektyvumui patikrinti buvo taip pat generuojamos trajektorijos. Generavimas yra beveik analogiškas trajektorijų, skirtų dabarties didelio tikslumo trajektorijų (GPS) suspaudimo metodo algoritmui. Kadangi GPS technologija gautus vietos duomenis galima laikyti taškais, o GSM technologija gautus duomenis galima laikyti apskritimais, todėl šiuo atveju dar papildomai yra sugeneruojamas apskritimo spindulys. Kuo didesnis spindulys, tuo didesnis objekto buvimo vietos neapibrėžtumas. Apskritimo spindulys taip pat yra atsitiktinai generuojamas pagal vieningą pasiskirstymą.

Metodo, atliekančio GSM technologija gautų trajektorijų suspaudimą, pagrindinė idėja yra pakeisti GSM celes (suprastintu variantu jos yra laikomos apskritimais) į kitokias figūras. Toks pakeitimas padidintų objekto buvimo pozicijos paklaidą, tačiau galėtų apjungti įrašus, priklausančius tom pačiom celėm tokiu būdu eliminuojant laiko dedamąją. Panašaus algoritmo negalima sukurti GPS technologijos pagalba gautoms trajektorijoms, kadangi tose trajektorijose nėra dažnai pasikartojančių pozicijų dėl judančio objekto vietos nustatymo tikslumo. Algoritmas turi dvi pagrindines paklaidas, kurios leidžia reguliuoti algoritmo darbo pabaigoje gautos trajektorijos kokybę. Laiko paklaida būtų aktualesnė, jeigu trajektorija būtų generuojama (duomenys būtų nuskaitomi) ne vienodais intervalais. Tuomet pradėtų veikti algoritmo laiko įvertinimo mechanizmas, tačiau šiame darbe koordinačių pokyčiai yra aktualesni, todėl koncentruojamasi būtent į atstumo paklaidos keitimą ir algoritmo rezultatų stebėjimą. Reikia pabrėžti, kad skirtingai nuo praeities didelio tikslumo trajektorijų (GPS) suspaudimo metodo algoritmo paklaidos skaičiavimo (skaičiuojami nuotoliai tarp originalios ir suspaustos trajektorijos), čia ši paklaida yra skaičiuojama atimant buvusios celės plotą iš gauto stačiakampio ploto.



35 pav. Mažesnio tikslumo trajektorijos prastinimo algoritmo darbo vertinimas

Algoritmo darbo pabaigoje yra apskaičiuojami sekimą įvertinantys rezultatai. Iš grafiko galima spręsti, kad kokybės koeficientas neturi tiesioginės priklausomybės nuo trajektorijos ilgio, tačiau yra priklausomybė nuo suspaudimo koeficiento keitimo. Yra viena išimtis, kai esant didesniam suspaudimo koeficientui (50, 250), gautas kokybės koeficientas didesnis negu koeficientas, gautas, naudojant suspaudimo koeficientą lygų 10. Taip yra todėl, kad naudojant šį suspaudimo koeficientą yra labai nedaug sumažinama įrašų judančių objektų duomenų bazėje. Gautas kokybės koeficientas yra skirtingos eilės skaičius negu kokybės koeficientas, apskaičiuotas pagal GPS technologija gautas trajektorijas. 35 pav. pateiktas grafikas, vertinantis algoritmo darbą. Gauta statistika rodo, kad algoritmo darbo efektyvumas krenta (mažėja kokybės koeficientas), kai didinamas galimos atstumo paklaidos koeficientas. Šiam algoritmui didinant paklaidos koeficientą, trajektorijos kokybė krenta žymiai greičiau negu sutaupomi duomenys, todėl algoritmas gali būti efektyviau taikomas tik tam tikromis sąlygomis (kai vienoje celėje judantis objektas yra praleidęs kuo daugiau laiko, tuo efektyvesnis algoritmo veikimas). Jeigu ši užduota prastinimo paklaida artėja į nulį, tai reiškia, kad stačiakampis bus padaromas kiekvieną kartą, kai tik bus pereinama į naują celę (informacijos mažiausiai prarandama, jeigu paklaida nustatoma maža, tuomet gauti daug stačiakampių duoda žymiai didesnę tikslumą negu vienas didelis stačiakampis), todėl algoritmo darbas su mažomis paklaidomis bus efektyvus tik tuo atveju, jeigu kiekvienoje celėje objektas buvo ilgesnį laiką ir yra daugiau negu vienas įrašas, žymintis objekto buvimą toje celėje. Tokiu atveju verta prastinti duomenis. Tačiau jeigu apjungiamų įrašų į stačiakampį yra vienas, tuomet algoritmas praneš, kad prastinti šios vietos neverta, nes atsiras tik nuostoliai dėl objekto buvimo pozicijos tikslumo (apskritimas virstų į

stačiakampį ir sąlygotų išaugusią paklaidą). Jeigu algoritmui perduota trajektorija sudaryta iš didelių celių (didelių palyginti su judančio objekto greičiu ir jis užgaišta kiekvienoje celėje daug laiko), algoritmo darbas taps efektyvesnis. Turint omeny, kad mieste ir užmiestyje celių dydžiai skiriasi, o užmiestyje jos yra didesnės algoritmas būtų žymiai efektyvesnis užmiestyje negu mieste.

5. Išvados

- Atlikta vietos nustatymo technologijų analizė, išskirtos kiekvienai šių technologijų būdingos savybės ir jomis gaunamų vietos duomenų formatai. Taip pat atlikta vietos duomenų trajektorijų suspaudimo metodų lyginamoji analizė, pateiktos jų taikymo sritys ir trūkumai. Pasiūlytas nauji trajektorijos suspaudimo metodai, panaudojantys geriausias žinomų trajektorijos suspaudimo metodų savybes.
- Nustatyta kad esami vietos duomenų suspaudimo metodai netinka tiesioginiam taikymui mobilios ryšio sistemų pagalba gautiems vietos duomenims, nes visiškai nevertinamas trajektorijos netikslumas atsirandantis dėl pozicionavimo technologijos savybių.
- GSM vietos duomenų suspaudimo metodas yra tuo efektyvesnis, kuo toje pačioje celėje praleidžiama daugiau laiko arba kai celės yra didelės. Todėl metodas puikiai tinka neypatingai judrių objektų (žmonių, kurie kiekviena diena dirba toje pačioje darbo vietoje, grįžta namo ir pan.) trajektorijų suspaudimui. Tokių žmonių yra dauguma.
- Vietos duomenys gali būti spaudžiami iki vartotojui priimtino vietos duomenų išsikraipymo lygio. Seniems ir todėl ne tokiems svarbiems trajektorijos duomenims gali būti leidžiamas didesnis išsikraipymas, o patys naujausi duomenys, kuri laiką saugojami netgi be jokio suspaudimo.
- Dabarties didelio tikslumo trajektorijų (GPS) suspaudimo metodas įvertina judančio objekto daromus posūkius, judėjimo kreivumą ir mažiausiai iš esamų dabarties trajektorijų apdorojimo būdų apkrauna sistemą, tačiau įvertina tik paskutinius gautus trajektorijos taškus, kai tuo tarpu praeities didelio tikslumo trajektorijų (GPS) suspaudimo metodas įvertina visą turimą trajektoriją, todėl jis yra efektyvesnis. Esant reikalui abu metodai gali būti naudojami kartu.
- Dabarties didelio tikslumo trajektorijų (GPS) suspaudimo metodo algoritme didėjant trajektorijos ilgiui, kokybės koeficiento skirtumai, naudojant įvairius suspaudimo laipsnius, mažėja, nes kuo mažiau trajektorijoje taškų, tuo daugiau skiriasi esančios reikšmės, vėliau trajektorijoje esančių taškų kiekiui didėjant, reikšmės tampa mažiau išsibarsčiusios, todėl koeficientai tampa vienodesni, o algoritme koordinatės pokytis turi didesnę įtaką algoritmo darbui negu intervalo pokytis, todėl kokybės koeficientas mažiau priklauso nuo laiko intervalo negu nuo koordinatės pokyčio.
- Praeities didelio tikslumo trajektorijų (GPS) suspaudimo metodo algoritme geriausias kokybės koeficientas yra tuomet, kai taikomas mažiausias suspaudimo koeficientas.

6. Literatūros sąrašas

- [1] **Schreiner, M. Tangemann, M. Nikolai, D.** A new network-based positioning method for location services in 2G and 3G mobile communications. Res. & Innovation, Alcatel SEL AG, Stuttgart, Germany, ISSN: 0537-9989, ISBN: 0-85296-753-5.
- [2] **Trevisani, E.; Vitaletti, A.** Cell-ID location technique, limits and benefits: an experimental study. Mobile Computing Systems and Applications, 2004
- [3] **Santosh Pandey, B. Kim, F. Anjum, and P. Agrawal.** Client assisted location data acquisition scheme for secure enterprise wireless networks. IEEE WCNC 2005, 2005
- [4] **Zhao-Hong Liu, Xiao-Li Liu, Jun-Wei Ge, and Hae-Young Bae.** Indexing Large Moving Objects from Past to Future with PCFI+-Index. 11th International Conference on Management of Data (COMAD 2005) (COMAD 2005), Goa, India, January 6-8, 2005
- [5] **Guting, R. H., B'OHLEN, M. H., Erwig, M., Jensen, C. S., Lorentzos, N. A., Schneider, M., and Vazirgiannis, M.** A Foundation for Representing and Querying Moving Objects. ACM Transactions on Database Systems 25, 1, pp1–42, Mar, 2000
- [6] **L. Florizzi, R. H. Guting, E. Nardelli, and M. Schneider.** A data model and data structures for moving objects databases. Technical Report 260-10, Fern-Universit'at Hagen, 1999
- [7] **D. Pfoser and C. S. Jensen.** Indexing of network constrained moving objects. ACM GIS, pages 25–32. ACM Press, 2003
- [8] **H. Cao, O. Wolfson, and G. Trajcevski.** Spatio-temporal data reduction with deterministic error bounds. Journal of Very Large Databases (VLDBJ), 2006
- [9] **A. Abutaleb and V. O. K. Li.** Location update optimization in personal communication systems. Wireless Networks, vol. 3, no. 3, pp. 205-216, 1997
- [10] **Kyoung-Wook MIN and Jong-Hyun PARK.** Techniques for Acquisition of Moving Object Location in LBS. Paris, France, April 13-17, 2003
- [11] **Michalis Potamias, Kostas Patroumpas, Timos K. Sellis.** Sampling Trajectory Streams with Spatiotemporal Criteria. SSDBM 2006: 275-284
- [12] **Pukėnas A., Matulis D.** Mobilijų ir bevielų tinklų abonentų vietos duomenų suspaudimo metodai. 12-oji tarpuniversitetinė doktorantų ir magistrantų konferencija "Informacinė visuomenė ir universitetinės studijos IVUS'07", 2007
- [13] **A. Bar-Noy, I. Kessler, and M. Sidi.** Mobile users : To update or not to update? Wireless Networks, vol. 1, no. 2, pp. 187-196, 1995
- [14] **I. F. Akyildiz and J. S. M. Ho.** Dynamic mobile user location update for wireless PCS networks. Wireless Networks, vol. 1 no. 2 pp. 187-196, 1995
- [15] **J. S. M. Ho and I. F. Akyildiz.** Mobile user location update and paging under delay constraints. Wireless Networks, vol. 1. no. 4, pp. 413-425, 1995
- [16] **N. Meratnia, R.A. de By.** A new perspective on trajectory compression techniques. Quebec city, Canada, October 2-3, 2003
- [17] **Douglas, D., Peucker, T.** Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Can. Cartographer 10(2), 112–122, 1973
- [18] **McMaster, R.** Automated line generalization. Cartographica 24(2), 74–111, 1987

- [19] **Hobby, J.D.** Polygonal approximations that minimize the number of inflections. Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 93–102, 1993
- [20] **Agarwal, P.K., Varadarajan, K.R.** Efficient algorithms for approximating polygonal chains. Discrete Comput. Geom. 23, 273–291, 2000
- [21] **Chan, W., Chin, F.** Approximation of polygonal curves with minimum number of line segments or minimum error. Int. J. Comput. Geom. Appl. 6, 50–77, 1996
- [22] **Hu Cao, Ori Wolfson.** Nonmaterialized Motion Information in Transport Networks. International Conference on Database Theory, 2005
- [23] **Guttman, A.** R-trees: A Dynamic Index Structure for Spatial Searching. In Proceedings of the ACM SIGMOD Conference (Boston, MA), pp. 47–57, 1984
- [24] **H. J. Ohlbach, M. Rosner, B. Lorenz, E. P. Stoffel.** NL Navigation Commands from Indoor WLAN fingerprinting position data. Institute for Informatics, Ludwig-Maximilians-Universität München, 2006

7. Priedai

1 priedas

MOBILIŲJŲ IR BEVIELIŲ TINKLŲ ABONENTŲ VIETOS DUOMENŲ SUSPAUDIMO METODAI

Andrius Pukėnas, Darius Matulis

Kauno Technologijos Universitetas, Studentų g. 50

Straipsnyje išanalizuotos naujos vietos nustatymo technologijos, vietos duomenų suspaudimo metodai, pateiktas šių metodų taikymo tinkamumo mobiliuose ir bevielėse technologijose gaunamiems vietos duomenims įvertinimas, pateiktos vietos duomenų suspaudimo rekomendacijos taikant bevielio ir mobilaus ryšio technologijas

1. Įvadas

Su abonentų pozicija geografinėje erdvėje susijusi informacija naudojama labai įvairiose srityse: greitajai pagalbai teikti (911 paslauga), teisėsaugoje, krovinių transportavime (stebima kur šiuo metu yra transportuojami kroviniai), turizme, eismo sąlygų stebėjimui, artimiausių ligoninių, restoranų, lankytinų vietų paieškai, pramogoms, mobiliojo tinklo struktūros planavimui (nustačius kur yra didesnis abonentų susibūrimas, pastatoma daugiau bazinių stočių) ir t.t.

Judančiu objektu laikomas toks objektas, kuris bėgant laikui keičia savo pozicija erdvėje. Tokia su objektų pozicijomis susijusi informacija kaupiama Judančių Objektų duomenų bazėse (JODB). Anksčiau vietos informacijai rinkti buvo naudojama tik GPS technologija. GPS technologijos pagrindinis trūkumas tas, kad ji sunkiai veikia miesto tipo vietovėse ir praktiškai neveikia požeminiuose ir antžeminiuose statiniuose. Tačiau vystantis įvairiomis mobiliosiomis ir bevielėms technologijoms, daugėjant jų naudojimo vietų, atsiranda galimybė tokius duomenis rinkti pasinaudojant ne tik GPS, bet ir GSM, WLAN, Bluetooth. Tokiu būdu atsiranda galimybė GPS pagalba gautus vietos duomenis patikslinti, papildyti arba netgi, jei nereikalaujama didelio tikslumo, judančių objektų vietą nustatyti nenaudojant GPS iš viso.

Stebint judančius objektus „visur ir visada“ bėgant laikui kiekvienam objektui susidaro baigtinis pozicijų skaičius, kuris vadinamas trajektorija. Trajektorijos duomenų kiekis priklauso nuo vietos nustatymo dažnumo, taip pat ir nuo naudojamos technologijos. Tačiau kai stebima labai daug judančių objektų vienu metu susidaro dideli duomenų kiekiai, kuriems saugoti reikia labai daug disko vietos. Todėl reikalingi tokių duomenų suspaudimo metodai, įvertinantys esminius mobiliųjų technologijų vietos nustatymo aspektus: tikslumą, neapibrėžtumą, vietos duomenų nuskaitymo intervalų kitimą.

Toliau įvertinami vietos duomenų gavimo aspektai skirtingomis bevielėmis ir mobiliosiomis technologijomis, pateikiami jau žinomi vietos duomenų suspaudimo metodai ir įvertinamos jų taikymo galimybės naujoms vietos nustatymui naudojamoms GSM, WLAN ir Bluetooth technologijoms.

2. Mobiliosios ir bevielės vietos nustatymo technologijos

GPS - vietos nustatymo sistema, leidžia nustatyti objekto koordinatas bet kurioje pasaulio vietoje. Sistemos pagrindas – IT technologijų sąveika su planetą gaubiančiu GPS palydovų tinklu. Duomenys apie objekto poziciją gaunami iš kelių palydovų (oid,x,y) pavidalu, kur oid - judančio objekto identifikatorius, x,y – geografinės objekto koordinatės. Tai yra tiksliausia pozicionavimo sistema. GSM – radijo celėmis paremtas tinklas, kuriomis pilnai padengiama geografinė teritorija. Tokio tinklo privalumai: padidėjęs aptarnaujamų vartotojų skaičius, sumažėjęs energijos sunaudojimas, efektyviau padengiamas plotas priklausomai nuo jame vidutiniškai esančių

abonentų skaičiaus. Būtent dėl celių GSM tinkle atsiranda abonento vietos nustatymo galimybė, remiantis technologijomis: Cell-ID, Cell-ID + Timing Advance, E-OTD, A-GPS ir pan. Tačiau GSM vietos nustatymas ne toks tikslus kaip kad GPS, nes vietos duomenys turi pavidalą: (oid, x, y, r) , kur r – celės, kurioje gali būti objektas, spindulys. WLAN yra beveik LAN (vietinis tinklas), kuris apjungia kompiuterius, nenaudojant laidų. Jis gali būti panaudotas ne tik prieigai prie interneto, tačiau ir atlikti kliento vietos nustatymą. Iš pradžių įvairiuose patalpos taškuose išmatuojamas signalo stiprumas, o gauti duomenys surašomi į signalų stiprumo lentelę. Remiantis šia lentele galima rasti esamą objekto poziciją, kadangi prie WLAN tinklo prisijungęs objektas skleidžia tam tikro stiprumo signalą. Toks vietos nustatymo būdas nėra toks tikslus, kaip kad GPS ar GSM, kadangi gaunama netaisyklinga zona, kurioje gali būti objektas [7], tačiau ši zona paprastai nėra labai didelė ir yra žinomoje vietoje, pvz., pastate, degalinėje ir pan. Skirtingai nuo kitų pozicionavimo sistemų, WLAN tikslas nėra generuoti kliento judėjimo trajektoriją. Jis gali tik papildyti arba patikslinti kitomis technologijomis gautą trajektorijos informaciją, kurios pagalba galima keisti objektui pateikiamų nuo buvimo vietos priklausomų paslaugų turinį. Čia koncentruojamasi į tam tikrus įvykius (pavyzdžiui, tam tikros zonos kirtimas, prisijungimas), kuriais remiantis klientui pateikiamos skirtingos nuo jo pozicijos priklausančios paslaugos [10], pvz., QoS, saugumas (resursais leidžiama naudotis tik apibrėžtoje zonoje), artimiausio įrenginio (pavyzdžiui, spausdintuvo) radimas. Bluetooth standartas suteikia galimybę apsiukeisti informacija tarp įrenginių, palengvina paslaugų aptikimą ir parametrų jiems nustatymą. Šios technologijos pagalba galima atlikti ribotą judančių objektų pozicionavimą, kadangi klientui prisijungus prie Bluetooth įrenginio yra žinomas tik to įrenginio identifikatorius. Tokio vietos nustatymo paklaida proporcinga įrenginio apimamai ryšio zonai, kadangi vartotojas gali būti ant paslaugų teikimo zonos ribos, nebūtinai prie pat terminalo.

3. Trajektorijų vaizdavimas

Literatūroje judantys objektai analizuojami labai įvairiose srityse. Tai judančių objektų vietos informacijos talpinimas, indeksavimas, įvairių užklausų algoritmų ir duomenų suspaudimo metodų kūrimas. Vienas iš svarbiausių dalykų kalbant apie vietos duomenų suspaudimą yra trajektorijų vaizdavimo modelio pasirinkimas. Daugelis įprastinių modelių, skirtų duomenų vaizdavimui, yra statiniai, tačiau judančio objekto trajektorija pastoviai kinta [5].

Judančio objekto trajektorija laike atvaizduojama kaip atkarpomis sujungtų taškų, kurie gaunami vietos nustatymo technologija tam tikrais apibrėžtais arba atsitiktiniais laiko intervalais, eilutė. Toks metodas vadinamas tiesiniu interpoliavimu [6]. Tiksliai objekto pozicija tarp trajektorijos atkarpos galų yra neapibrėžta ir gali būti nusakoma su paklaida, kuri priklauso nuo objekto judėjimo krypties, greičio ir nuo viena po kitos nustatomų pozicijų dažnumo ir vietos nustatymo technologijos. Nuo technologijos taip pat priklauso ir atkarpos galų vietos informacijos tikslumas.

Judančių objektų duomenų bazėje objekto trajektorija gali būti atvaizduojama kaip vietos įrašų seka laike, pvz., (id, t_i, x_i, y_i) , kur id – unikalus trajektorijos identifikatorius, t_i - laiko momentas, kuriuo objektas buvo pozicijoje (x_i, y_i) , $i \in (1, \infty)$. Informacijos atnaujinimo intervalai gali būti parinkti priklausomai nuo reikalavimų rezultatų tikslumui, o bėgant laikui atnaujinimo intervalo dydis tam pačiam objektui gali kisti.

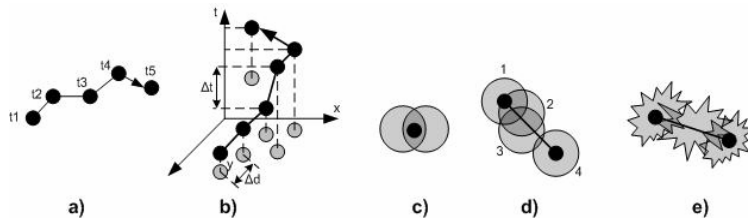
4. Vietos nustatymo technologijų taikymas trajektorijų informacijos suspaudimui

Pagrindinis trajektorijos informacijos suspaudimo modelių tikslas – sumažinti į judančių objektų duomenų bazę patenkančių duomenų kiekį, tuo pačiu kuo mažiau sumažinant jų tikslumą.

Trajektorijos duomenų suspaudimui naudojami laiku [1,8], atstumu [2,3,4,8] pagrįsti vietos duomenų atnaujinimo metodai. Šių metodų tikslas - tikrais laiko intervalais atnaujinti judančio objekto vietos informaciją, kurios kitimas priklauso nuo objekto judėjimo greičio, krypties ar įveikto atstumo. Kai sistema yra labai apkrauta išimtiniais atvejais gali būti taikomas spėjimu [8] pagrįstas metodas, kuris spėja judančio objekto sekančią poziciją pagal greitį ir kryptį. Tačiau trajektorijos gali būti formuojamos ne tik remiantis įvairiomis laiko ar judančio objekto judėjimo charakteristikomis. Taškai į trajektoriją gali būti įtraukiami priklausomai nuo to, ar objekto judėjimas yra panašus į tiesę - tada trajektorija turės mažiau taškų. Jei judėjimas panašus į kreivę, tada trajektorija turės daugiau taškų priklausomai nuo judėjimo kreivumo lygio [9].

Trajektorijų informacijai suspausti dažniausiai naudojami tokie modeliai: statinio ar dinaminio duomenų suspaudimo modeliai, atstumu ar grupavimu pagrįsti duomenų suspaudimo modeliai bei mišrus slenksčių modelis.

Statinio duomenų suspaudimo modelyje (1 pav. a) judančiam objektui stebėti naudojamas pastovus duomenų nuskaitymo periodas. Šis modelis universalus ir tinka visoms vietos nustatymo technologijoms, tačiau tikslumas ir saugomų duomenų kiekis priklauso nuo duomenų nuskaitymo periodo. Kuo mažesnis periodas, tuo daugiau vietos užims vietos informacija ir tuo objekto judėjimo trajektorija bus tikslesnė. Tikslumas gaunamas didelio duomenų kiekio sąskaita ir dideliu objektų skaičiui, kad ir kokia vietos nustatymo technologija jie būtų stebimi, jis netinka. Todėl šis modelis dažniausiai naudojamas tik palyginimui su kitais modeliais.



1 pav. Statinis modelis (a), atstumu pagrįstas modelis (b), situacijos taikant atstumu pagrįstą modelį (c,d,e)

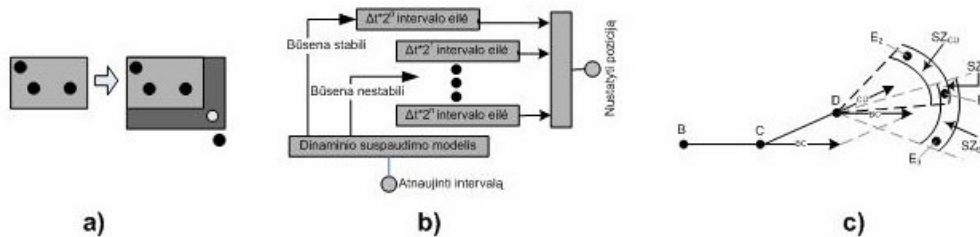
Atstumu pagrįstame duomenų suspaudimo modelyje (1 pav. b) naudojamas kintamas vietos duomenų nuskaitymo intervalas. Objektui judant greitai nuskaitymo intervalai yra maži, kai tuo tarpu judant lėtai - duomenys nuskaityjami rečiau. Čia nustatomas nuėito atstumo kitimo dydžio slenkstis ir kiekvieną kartą atnaujinant objekto vietos duomenis, nuėitas atstumas yra lyginamas su slenksčiu, tokiu būdu kontroliuojant nuskaitymų skaičių. Šis modelis pilnai tinka GPS technologijai, kadangi ja pakankamai tiksliai nustatoma objekto pozicija. GSM technologija galima nustatyti poziciją tik su kelės dydžio paklaida, todėl atsiranda vietos duomenų neapibrėžtumas. Tarkime, kad buvo nustatytos dvi objekto pozicijos, kurios yra skirtingose kelėse. Jei kelės persikerta tampa neaišku ar objektas pajudėjo, nes objektas gali būti bet kurioje kelės vietoje (1 pav. c). Tačiau gretimų celių neregistruojant, galima sumažinti neapibrėžtų duomenų kiekį, t.y. duomenys yra suspaudžiami išmetant iš jų neapibrėžtą informaciją. Tada metodas pasidaro tinkamas GSM technologijai. Tarkime, kad 1 kelė yra paskutinė DB įrašyta pozicija, tada 2 ir 3 kelės nėra įrašomos, nes persikerta su 1, 4 įrašoma, nes nesikerta su 1, o persikirtimus galima apskaičiuoti, kadangi yra žinomas technologinis netikslumas, susijęs su spinduliu r (1 pav. d). WLAN technologijai metodą taikyti sunkiau, paskaičiavus WLAN prieigos taško maksimaliai apimamą zoną ir neregistruojant persikertančių zonų. Objekto nuotolis registruojamas tik nepersikertančių zonų atveju (1 pav. e).

Grupavimu pagrįstame duomenų suspaudimo modelyje (2 pav. a) naudojamas minimalus trajektoriją apimantis stačiakampis, kuris objektui judant didėja. Jei nagrinėjamas judantis objektas juda stačiakampio ribose, vietos duomenų nuskaitymai yra reti, tačiau jeigu objektas juda

greičiau negu didėja stačiakampis, nuskaitymas dažnėja. GPS technologijai šis modelis tinka, GSM galima būtų taikyti, tikrinant kiek procentų trajektorijos apskritimas išlenda iš stačiakampio. Šie procentai atitinka per didelį į JODB patalpintų įrašų skaičių. WLAN technologijai iš esmės būtų galima pritaikyti, modelyje padarius analogiškus pakeitimus, t.y. skaičiuojama kiek procentų netaisyklingos geometrinės figūros išlenda iš stačiakampio ribų.

Dinaminio duomenų suspaudimo modelis (2 pav. b) taikomas kai sistemos resursai (atmintis, procesorius) yra apkrauti. Jeigu sistema yra stabilioje būsenoje, judančio objekto vietos duomenys nuskaitymi kaip ir statiniame modelyje. Kai sistema nestabili, judantys objektai pagal prioritetą paskirstomi į keletą eilių su skirtingais nuskaitymo intervalais $\{\Delta t * 2^0, \Delta t * 2^1, \dots, \Delta t * 2^n\}$. Tinka visoms technologijoms, tik GSM ir WLAN reikia padaryti analogiškus tikrinimus kaip ir atstumu pagrįstame modelyje.

Mišriame slenksčių modelyje (2 pav. c) naudojamos saugios zonos. Saugi zona yra tokia zona, į kurią patekę taškai nėra įtraukiami į trajektoriją. Tarkime, kad B ir C yra paskutiniai į JODB įrašyti trajektorijos taškai [9], o D - paskutinė nustatyta, bet dar neįrašyta trajektorijos pozicija. Saugi zona SZ_{CD} sudaroma remiantis C ir D taškais, o SZ_{BC} - remiantis B ir C. Modelyje saugi zona (SZ) yra apskaičiuojama perdengiant saugias zonas SZ_{CD} ir SZ_{BC} .



2 pav. Grupavimu pagrįstas modelis (a), dinaminio duomenų suspaudimo modelis (b), mišrus slenksčių modelis (c)

Jei trajektorijos taškas patenka į saugumo zoną, jis ignoruojamas (E1). Priešingu atveju fiksuojama, kad įvyko reikšmingas pasikeitimas objekto judėjime (E2, E3 atvejai). Tuomet nepatenkantys į saugumo zoną taškai įrašomi į trajektoriją. Modelis lengvai taikomas GSM, nes esant celėms saugios zonos būtų daug didesnės. Modelis galėtų būti pritaikytas ir WLAN, nors objekto buvimo vieta ir nėra tiksliai aprašoma. Čia reikėtų įvertinti maksimaliai galima paklaidą tarp realios objekto pozicijos ir nustatytos. Įvertinus paklaidą, galima objekto pozicija bus netaisyklinga geometrinė figūra. Saugių zonų sudarymas šiuo atveju taptų sudėtingesnis dėl didesnio resursų poreikio, o pačios saugios zonos nebūtų tokios taisyklingos, todėl daugeliu atveju reikėtų spėti ar objekto judesio pasikeitimas yra svarbus ar ne.

5. Suspaudimo metodų aspektai

Iš esmės galima teigti kad egzistuoja du vietos duomenų suspaudimo metodų tipai. Pirmajame priklausomai nuo informacijos svarbos keičiami judančių objektų vietos duomenų gavimo intervalai – kuo objektas greičiau ir labiau nenuspėjamai juda, tuo objekto vietos informacijos nuskaitymai turi būti dažnesni. Šia idėja remiasi atstumu ir grupavimu pagrįsti, o taip pat dinaminio duomenų surinkimo modeliai. Antrajame remiamasi ne laiko intervalų keitimu, tačiau judančio objekto pozicijų fiksavimu, priklausomai nuo informacijos svarbumo. Šią idėją geriausiai išnaudoja mišrus slenksčių modelis. Pirmas tipas taikant mobiliosioms ir bevielėms technologijoms yra geresnis, kadangi gaunamas didesnis trajektorijos tikslumas, kai tuo tarpu antrojo tipo atveju tikslumo maksimumas priklauso nuo sistemoje įrašyto (nekeičiamo) nuskaitymų intervalo. Tačiau abu tipai turi mobiliosioms ir bevielėms technologijoms tinkamų savybių, kurias apjungus galima sukurti visiškai naują modelį, apimantį abiejų tipų geriausias

savybes. Kadangi pirmojo tipo modeliuose objektui judant greičiau vietos informacijos nuskaitymų skaičius didinamas, o judant lėčiau – mažinamas, modeliai yra inertiški ir reikiamą nuskaitymo intervalą pasiekia tik po tam tikro laiko, t.y. nusistovi. Tačiau nusistovėjimo laikotarpiu visi duomenys apie judantį objektą yra įterpiami į trajektoriją nepaisant jų svarbumo. Šis svarbumas gali būti įvertintas pritaikant antrojo tipo modeliuose taikomas savybes. Tokiu būdu pirmojo tipo modelių savybių dėka galima didinti tikslumą, o antrojo tipo modelių - sutaupyti disko vietas.

6. Išvados ir tolimesni darbai

Šiame darbe buvo išnagrinėti duomenų apie judančius objektus suspaudimo metodai, įvertinant įvairias mobiliąsias ir bevielės technologijas, ištirtas galimas jų pritaikymas šioms technologijoms. Pagrindinis analizuotų suspaudimo metodų pritaikymo tam tikroms technologijoms trūkumas tas, kad duomenų bazėje turi būti saugoma paskutinė užfiksuota judančio objekto vieta, o taip pat reikalaujami didesni procesoriaus resursai skaičiuojant zonų persikirtimus. Jei sistemos resursai dideli, galima naudoti ir statinių modelį, kuris užtikrina didžiausią tikslumą, tačiau dažniausiai resursų trūksta, todėl šis modelis naudojamas tik palyginimui su kitais modeliais. GSM technologija dėl savo techninių savybių generuoja didesnę duomenų neapibrėžtumą ir dažniausiai nėra taikoma ten, kur reikalaujamas labai didelis tikslumas, todėl galima atlikti duomenų suspaudimą priklausomai nuo vartotojo reikalaujamo tikslumo. Bluetooth technologijai netinka nei vienas modelis, nes technologija nesiremia pasikartojančiais objekto vietos nustatymais, o paremta įvykiais (trigeriais), pvz. prisijungimas/atsijungimas. Tačiau ši technologija gali būti panaudojama kitomis technologijomis gautų vietos duomenų papildymui, arba objektų vietos nustatymui tam tikrose apibrėžtose erdvėse. WLAN gautiems vietos duomenims galima būtų taikyti daugelį duomenų suspaudimo modelių ir taikyti daug efektyviau negu Bluetooth technologiją, panaudojant ją įvairių su lokalia vieta susijusių paslaugų pateikimui.

Tolimesniuose darbuose planuojama sukurti hibridinį vietos duomenų suspaudimo modelį, leidžiantį suspausti vietos duomenis, kurie gali būti pateikiami iš visų analizuotų technologijų vienu metu, ir pateikti jo eksperimentinį įvertinimą.

Literatūros sąrašas

- [1] **A. Bar-Noy, I. Kessler, and M. Sidi.** Mobile users : To update or not to update? *Wireless Networks*, vol. 1, no. 2, pp. 187-196, 1995.
- [2] **I. F. Akyildiz and J. S. M. Ho.** Dynamic mobile user location update for wireless PCS networks. *Wireless Networks*, vol. 1 no. 2 pp. 187-196, 1995
- [3] **J. S. M. Ho and I. F. Akyildiz.** Mobile user location update and paging under delay constraints. *Wireless Networks*, vol. 1. no. 4, pp. 413-425, 1995
- [4] **A. Abutaleb and V. O. K. Li.** Location update optimization in personal communication systems. *Wireless Networks*, vol. 3, no. 3, pp. 205-216, 1997
- [5] **Zhao-Hong Liu, Xiao-Li Liu, Jun-Wei Ge, and Hae-Young Bae.** Indexing Large Moving Objects from Past to Future with PCFI+Index. 11th International Conference on Management of Data (COMAD 2005) (COMAD 2005), Goa, India, January 6-8, 2005
- [6] **Guting, R. H., B'OHLEN, M. H., Erwig, M., Jensen, C. S., Lorentzos, N. A., Schneider, M., and Vazirgiannis, M. A** Foundation for Representing and Querying Moving Objects. *ACM Transactions on Database Systems* 25, 1, pp1-42, Mar, 2000
- [7] **Santosh Pandey, B. Kim, F. Anjum, and P. Agrawal.** Client assisted location data acquisition scheme for secure enterprise wireless networks. *IEEE WCNC 2005*, 2005.
- [8] **Kyoung-Wook MIN and Jong-Hyun PARK.** Techniques for Acquisition of Moving Object Location in LBS. Paris, France, April 13-17, 2003.
- [9] **Michalis Potamias, Kostas Patroumpas, Timos K. Sellis.** Sampling Trajectory Streams with Spatiotemporal Criteria. *SSDBM 2006*: 275-284

- [10] **Santosh Pandey, B. Kim, F. Anjum, and P. Agrawal.** Client assisted location data acquisition scheme for secure enterprise wireless networks. In IEEE WCNC 2005, 2005.

The compression methods for mobile and wireless networks subscribers' location data

This article covers analysis of new location technologies, compression methods of location data, evaluation of suitability for applying these methods in getting location data using different mobile and wireless technologies. There are also given location data compression recommendations when applying wireless and mobile technologies.