



TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

MATTI VAATTOVAARA
**NELIPUUTIEORAKENNE JA SEN TOIMINTA KÄYTÄNNÖN
SOVELLUKSISSA**

Kandidaatintyö

Tarkastaja: Marko Helenius

TIIVISTELMÄ

Matti Vaattovaara: Nelipuutietorakenne ja sen toiminta käytännön sovelluksissa

Tampereen teknillinen yliopisto

Kandidaatintyö, 17 sivua, 0 liitesivua

Toukokuu 2018

Tietotekniikan diplomi-insinöörin tutkinto-ohjelma

Pääaine: Ohjelmistotuotanto

Tarkastaja: Marko Helenius

Avainsanat: nelipuu, tietorakenne

Erilaiset tietorakenteet ovat tietotekniikassa olennainen osa teknisiä toteutuksia ja niiden soveltuvuus eri käyttökohteisiin vaihtelee suuresti. Tyypillinen kaksiulotteista aluetta kuvaamaan käytetty rakenne on tasaisesti jaettu ruudukko, jossa alue on jaettu pysty- ja vaakasuunnassa kaikkialta yhtä suuriin neliön muotoisiin osiin. Tasaisesti jaettu ruudukko ei kuitenkaan ole ainoa tapa esittää ja jakaa alue, vaan sille on olemassa useita vaihtoehtoja, joilla on siihen nähden erilaisia hyöty- ja haittapuolia.

Tässä tutkimuksessa esitellään lyhyesti nelipuu käsitteenä ja siitä kehitetyn nelipuutietorakenteen eri tyyppisiä ja käyttökohteita. Tutkimuksessa esitellään lähteisiin pohjautuen kolme versiota nelipuutietorakenteesta toimintaperiaatteineen ja annetaan lyhyt esimerkki ohjelmallisesti toteutetusta nelipuutietorakenteesta. Lisäksi esitellään kolme kirjallisuudessa esiintyvää kuvausta nelipuun toiminnasta käytännön sovelluksissa: hiukkasliikkeen simuloinnissa, tsunamin mallinnuksessa ja ympäristön mallinnuksessa. Nelipuun soveltuvuutta, sen etuja ja siitä aiheutuvia haittoja arvioidaan kussakin käyttötarkoituksessa lähdetietojen pohjalta ja pyritään muodostamaan laajempi kuva siitä, millaisiin käyttötarkoituksiin nelipuu soveltuu hyvin. Arvioiden perusteella todetaan nelipuulla olevan selkeitä etuja kaksiulotteisen alueen kuvauksessa ja liikkeen mallinnuksessa, varsinkin kun tarkastelu halutaan kohdentaa johonkin osaan alueesta. Nelipuun etujen todetaan liittyvän sen mahdollistamaan mukautuvaan alueen jakamiseen ja vähäiseen tilankäyttöön. Nelipuun haittapuoliksi arvioidaan sen usein monimutkainen toteutus ja sen nopean käsittelyn vaatima suuri suorituskyky.

ABSTRACT

Matti Vaattovaara: Quadtree data structure and its function in practical applications

Tampere University of Technology

Bachelor of Science Thesis, 16 pages, 0 Appendix pages

April 2018

Master's Degree Programme in Information Technology

Major: Software Engineering

Examiner: Marko Helenius

Keywords: quadtree, data structure

In this study, the quadtree is briefly introduced as a concept and as different types of data structures derived from that concept. Some potentially practical uses for the data structures are also presented.

SISÄLLYSLUETTELO

1.	JOHDANTO	1
2.	NELIPUU	3
2.1	Nelipuu tietorakenteena	3
2.2	Nelipuun tyyppejä.....	3
2.2.1	Aluenelipuu	4
2.2.2	Pistenelipuu	5
2.2.3	PR-nelipuu.....	5
2.3	Esimerkki nelipuusta.....	6
3.	NELIPUUN KÄYTTÖ.....	8
3.1	Nelipuu hiukkasten liikkeen simuloinnissa.....	8
3.1.1	Tekninen hyödyntäminen	8
3.1.2	Soveltuvuus käyttökohteeseen.....	9
3.2	Nelipuu tsunamin mallinnuksessa	10
3.2.1	Tekninen hyödyntäminen	10
3.2.2	Soveltuvuus käyttökohteeseen.....	11
3.3	Nelipuu ympäristönmallinnuksessa	12
3.3.1	Tekninen hyödyntäminen	12
3.3.2	Soveltuvuus käyttökohteeseen.....	13
3.4	Muita käyttökohteita	13
4.	TULOSTEN ARVIOINTI.....	15
5.	YHTEENVETO.....	16

KUVALUETTELO

Kuva 1. <i>Mustiin ja valkoisiin osiin jaettu ruudukkomainen alue, jota kuvaa aluenelipuu. Nelipuun juuren lasten viereen on merkitty niiden suunta alueen keskipisteestä. Oikealla on esitetty kunkin hierarkia-asteen kuvaaman alueen suuruus. Perustuu lähteeseen [14].</i>	4
Kuva 2. <i>Pistenelipuu (oikealla) ja sen määrittämä alue. Pisteiden sijainnit on voitu määrittää esimerkiksi koordinaatein. Perustuu lähteeseen [14].</i>	5
Kuva 3. <i>Pisteiden mukaan jaoteltu alue ja sitä kuvaava PR-nelipuu. Muokattu alkuperäisestä [14, s. 2222].</i>	6
Kuva 4. <i>Kuvassa 1 esitetty alue kierrettynä 90 astetta myötäpäivään.</i>	7
Kuva 5. <i>Hiukkasten sijainteja alueella kuvaava nelipuu ennen (vas.) ja jälkeen siihen tehtävien päivitysten. Siniset nuolet osoittavat siirtymän kuvien välillä.</i>	9
Kuva 6. <i>Gerrisin käyttämän aluenelipuun ylimääräiset rajoitteet. Vihreät viivat kuvaavat kunkin ehdon täyttymiseen tarvittavia tarkennuksia ja punaiset viivat kiinteää rajaa. Perustuu lähteeseen [10, s. 4].</i>	10
Kuva 7. <i>Punaisilla viivoilla merkityt alueet arvioidaan jaettaviksi. Oikealla esimerkki mahdollisesta tarkennetusta alueesta jaon jälkeen. Perustuu lähteeseen [5].</i>	13

LYHENTEET JA MERKINNÄT

C++	Ohjelmointikieli C++
PR-nelipuu	PR-nelipuu (engl. Point, Region)
NE	Koillinen (engl. Northeast)
SE	Kaakko (engl. Southeast)
SW	Lounas (engl. Southwest)
NW	Luode (engl. Northwest)
Z-koodi	Tekniikka, jolla yksiulotteinen tieto voidaan esittää moniulotteisena
Bumblebee XB3	3D-kuvausjärjestelmä

1. JOHDANTO

Tietotekniikassa tietoa säilötään yleisesti erilaisiin tietorakenteisiin, jotka selkeyttävät tiedon säilytystä ja esimerkiksi helpottavat ja nopeuttavat tiedon käsittelyä. Eräs tietorakenteiden tyyppi ovat puurakenteet, jotka muistuttavat rakenteeltaan paljolti sukupuita: niillä on hierarkia ja solmuja, jotka ovat tietoa sisältäviä alkioita. Nelipuu on puurakenne, jossa jokaisella solmulla on joko nolla tai neljä siitä polveutuvaa solmua.

Nelipuuta on käsitelty kirjallisuudessa usein apuvälineenä joissakin sovelluksissa [5, 8, 11] ja toisaalla sen toteutus on esitelty teknisesti hyvinkin tarkasti [13, 14], mutta sen käytännön sovelluksia ei ole tällöin juurikaan avattu. Tästä syystä tässä tutkimuksessa perehdytään nelipuuhun laajemmin: käsitteen esittelystä ja tietorakenteen teknisestä toteutuksesta edetään ohjelmalliseen esimerkkiin ja nelipuun toimintaan käytännön sovel-luskohteissa.

Tämän tutkimuksen tavoite on vastata kysymykseen siitä, mitä nelipuu käsitteenä tarkoittaa, millaisia tietorakenneversioita siitä on tähän mennessä kehitetty, sekä mihin ja miten niitä voidaan käytännössä hyödyntää. Lisäksi tutkimuksessa esitetään arvioita nelipuun soveltumisesta kuhunkin käyttötarkoitukseen. Tutkimuksen luettuaan lukija ymmärtää perusteita nelipuusta yleisellä tasolla ja erilaisina tietorakenteina sekä kykenee hahmot-tamaan sille mahdollisesti sopivia käyttökohteita.

Tutkimus toteutettiin tekemällä laajahko kartoitus nelipuun esiintymisestä tieteen ja tekniikan alojen julkaisuissa, joita haettiin Tampereen teknillisen yliopiston kirjaston Andor-tiedonhakupalvelulla, sekä Web of Science ja Springer link -tietokannoista. Myös google-hakukonetta käytettiin aiheeseen tutustuttaessa ja alkuperäislähteitä etsittäessä. Tietoa haettaessa yleisimpiä hakusanoja tai sanayhdistelmiä olivat ”quadtree”, ”(hierarchical) data structure”, ”use of quadtree” ja ”quadtree application”. Lähteiden ja niiden julkaisualue-tojen luotettavuutta arvioitiin niihin kohdistuvien viittausten yleisyyden perusteella ja Ulrichsweb Serials Directory -tietokannan avulla.

Lähteissä nelipuusta on esitetty runsaasti erilaisia versioita, joiden tulkinnat ja esitystavat eroavat toisistaan välillä selvästikin. Tähän tutkimukseen valittiin tarkemmin esiteltäviksi tietoa haettaessa usein ilmenneitä versioita nelipuusta, joiden soveltumista eri käyttötarkoituksiin arvioitiin kirjallisuudessa esiintyviin lähteisiin perustuen.

Tämä tutkimus suhteutuu muihin aihetta käsitteleviin tutkimuksiin yhdistelevänä ja ko-koavana tekijänä sivuten aluksi nelipuuta esitteleviä tutkimuksia [13, 14] ja nojaten lo-puksi vahvasti nelipuusta jo tehtyihin, käytännön sovelluksiin liittyviin tutkimuksiin [5, 9, 11].

Luvussa kaksi pohjustetaan työtä: esitellään nelipuu käsitteenä, nelipuu tietorakenteena, sekä kolmen erilaisen nelipuutietorakennetyypin rakenteiden periaatteet. Luvussa kolme esitellään kolme eri käytännön sovellusta, joissa nelipuu on todettu käyttökelpoiseksi, esitellään nelipuun toimintaperiaatetta kussakin sovelluksessa, sekä arvioidaan nelipuun soveltuvuutta kuhunkin tarkoitukseen. Luvussa kolme esitellään lisäksi lyhyesti muita tietoa haettaessa löytyneitä nelipuun potentiaalisia sovelluskohteita. Tuloksia arvioidaan luvussa neljä ja yhteenveto tutkimuksen tuloksista esitetään luvussa viisi.

2. NELIPUU

Termin nelipuu (engl. quadtree) voidaan yleisesti katsoa käsittävän joukon tapoja kuvata geometrisia entiteettejä, jotka jakavat alueen kahden tai useamman ulottuvuuden avaruudessa rekursiivisesti pienempiin osiin, kunnes jokin ehto täyttyy. Tällaisia entiteettejä ovat esimerkiksi pisteet, janat ja alueet, jotka voivat kukin jakaa esimerkiksi neliön muotoisen alueen pienempiin osiin ja jälleen jakaa näin syntyneet alueet edellisen tavoin. Ehto, jonka täytyminen lopettaa kuvaillun kaltaisen jakamisen, voi olla esimerkiksi tiedetyn alueiden minimikoon saavuttaminen tai alueiden sallitun maksimimäärän ylittäminen. [14, s. 2219]

2.1 Nelipuu tietorakenteena

Tietorakenteiden viitekehyksessä nelipuu tarkoittaa puurakennetta, jossa jokaisella solmulla, jolla on lapsia, on niitä tarkalleen neljä, ja jota käytetään etenkin kaksiulotteisten alueiden kuvaamiseen. Tällöin nelipuun juurisolmu kuvaa koko kuvattavaa aluetta ja muut solmut osia vanhemmistaan. Sisärsolmut siis jakavat koko vanhempansa kuvaaman alueen neljään sitä pienempään osa-alueeseen ja muodostavat yhdessä vanhempansa koko alueen. Nelipuun solmuja, joilla ei ole lapsia, kutsutaan puun lehdiksi tai lehtisolmuiksi. [4, s. 380, 14, s. 2219]

Tässä työssä yhdistettäessä alue ja sitä kuvaava nelipuu, kunkin solmun lasten edustamat osa-alueet vasemmalta oikealle ovat ilmansuuntien mukaan luode (NW), koillinen (NE), kaakko (SE) ja lounas (SW). Näin ollen nelipuun graafisessa esityksessä solmun lapsista esimerkiksi kolmas vasemmalta kuvaa solmun kuvaaman alueen osa-alueita, joka sijoittuu ilmansuunnista kaakkoon eli alas ja oikealle.

2.2 Nelipuun tyyppejä

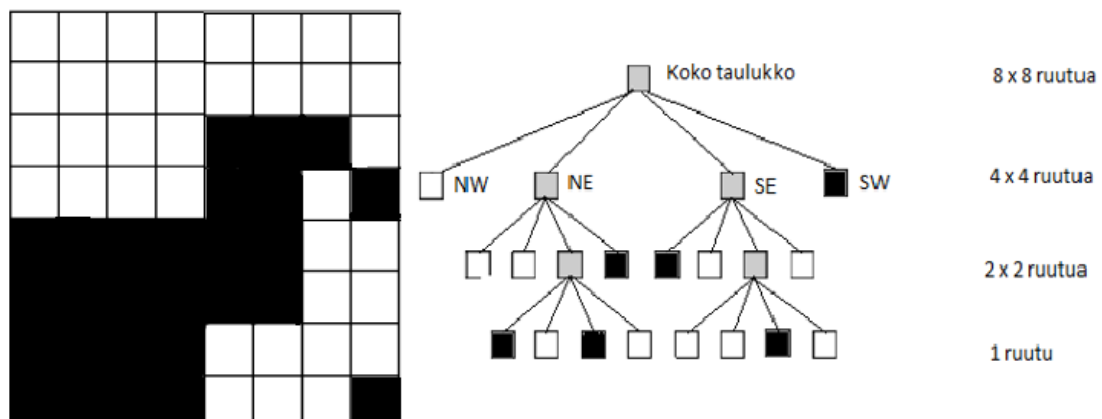
Nelipuita voidaan määrittellä useanlaisia. Määrittelyyn voi vaikuttaa esimerkiksi kuvatus avaruuden dimensioiden määrä, kuvatus datan tyyppi, alueen jakamisen kriteerit, sekä kuvatus alueen tai osa-alueiden koko, muoto tai sijainti. [13, 14, s. 2219]

Yleensä nelipuista puhuttaessa viitataan puurakenteisiin, jotka jakavat hierarkkisesti ja tasaisesti alueen pienempiin, usein kullakin jakokerralla yhtä suuriin osiin [13, 14]. Toisaalta alueen jako voi riippua jostakin syötteestä, jolloin on mahdollista, että jako ei ole tasainen alueen sisällä [13]. Tällaista rakennetta käyttäen päädytään usein tilanteeseen, jossa jokin alue on jaettu pienempiin, kooltaan ja sijainniltaan tunnettuihin osa-alueisiin. Nämä osa-alueet voivat olla erikokoisia ja ne voivat jakautua alueelle tasaisesti tai epätasaisesti. Eri osa-alueisiin voidaan myös luonnollisesti yhdistää tietoa muustakin kuin niiden sijoittumisesta nelipuuhun.

2.2.1 Aluenelipuu

Aluenelipuu on suosittu nelipuun tyyppi kuvien tallentamisessa ja käsittelyssä. Alue, jota aluenelipuu kuvaa, voidaan kuvata kaksiulotteisena listana pienempiä alueita, joissa kummankin dimension alkioiden määrä on 2^n , missä n on jokin luonnollinen luku. Yksinkertaistettuna tämä alue voisi koostua mustista ja valkoisista alkiosta, jolloin aluetta kuvavassa nelipuussa kukin solmu saisi arvonsa sen esittämän alueen väristä siten, että alueen ollessa valkoinen (musta) myös solmu olisi valkoinen (musta). Jos solmun alueella olisi sekä mustia että valkoisia solmuja, todettaisiin solmun väri harmaaksi ja jaettaisiin solmun kuvaama alue neljään yhtä suureen osaan. Näitä osia nelipuussa kuvaisivat kyseisen solmun lapset, joista kunkin väri jälleen pääteltäisiin edellä kuvatuin perustein erikseen. [14]

Aluenelipuun toteutuksesta on useita variaatioita, jotka sopivat eri ominaisuuksiensa takia erilaisiin käyttötarkoituksiin. Eräs osoittimiin pohjautuva toteutus aluenelipuusta sisältää yhdenlaisia solmuja, joihin tallennetaan osoitinarvoja, joilla määrätään kunkin solmun lapset, vanhemmat ja väriarvo. [14]



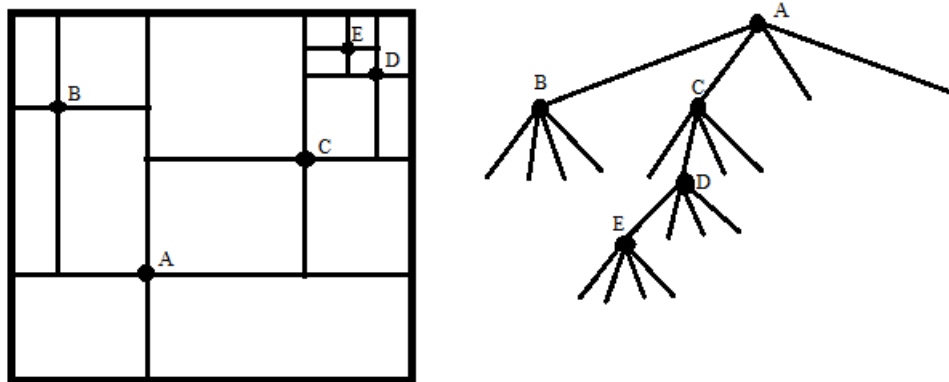
Kuva 1. Mustiin ja valkoisiin osiin jaettu ruudukkomainen alue, jota kuvaa aluenelipuu. Nelipuun juuren lasten viereen on merkitty niiden suunta alueen keskipisteestä. Oikealla on esitetty kunkin hierarkia-asteen kuvaaman alueen suuruus. Perustuu lähteeseen [14].

Aluenelipuuta käytetään esimerkiksi kuvapisteiden tai niiden muodostamien alueiden värien selvittämiseen ja kuvien uudelleenvärjäykseen, kieräyttämiseen, yhdistelyyn sekä muuhun muokkaukseen [14]. Kuvassa 1 on esitetty, miten aluenelipuu kuvaa mustavalkoista aluetta. Nelipuun graafisesta kuvauksesta voi helposti nähdä, että kuvatun alueen vasemmassa alanurkassa on yksinomaan mustia ruutuja. Kuvasta voidaan myös laskea, että nelipuuta käyttäen alue voidaan kuvata 21 solmulla, kun esimerkiksi kaksiulotteista listaa käytettäessä alkiota olisi 64.

2.2.2 Pistenelipuu

Pistenelipuussa solmun kuvaamaan alueeseen liitetään piste, jonka kautta kulkevat kaksi koordinaattiakselien suuntaista janaa jakavat alueen neljään osaan. Pisteiden sijainti alueella vaikuttaa siihen, miten alue jaetaan, joten alueet voivat olla hyvinkin erimuotoisia ja -kokoisia. Pistenelipuussa voidaan myös sallia useita pisteitä samalla koordinaatilla, jolloin kukin solmu pitää yllä listaa pisteistä, jotka siihen on talletettu. [14]

Pistenelipuuta käytetään muun muassa pisteiden indeksointiin eli esimerkiksi koordinaattien osoittaman pisteen tietojen tallentamiseen, sekä erilaisten kyselyjen tekoon, kuten tiedetyllä etäisyydellä jostain pisteestä sijaitsevien asioiden hakuun. Sen tehokkuus tällaisissa hakualgoritmeissa perustuu siihen, että haku voidaan keskittää vain haluttuun osaan hakualueesta. [14]

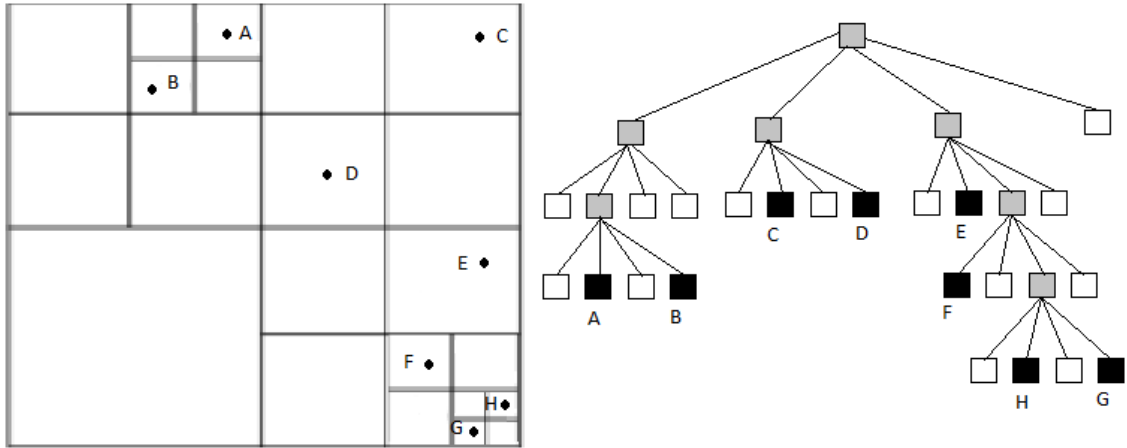


Kuva 2. Pistenelipuu (oikealla) ja sen määrittämä alue. Pisteiden sijainnit on voitu määrittää esimerkiksi koordinaatein. Perustuu lähteeseen [14].

Kuvassa 2 on esitetty esimerkki siitä, miten pistenelipuu voi jakaa alueen. Juurisolmuun liitetyn pisteen A kautta piirretyt janat jakavat solmun esittämän alueen neljään erisuuruiseen ja -muotoiseen osaan. Kuvasta nähdään esimerkiksi, että solmu, johon liitetään piste F, ei ole pisteeseen C liitetyn solmun lapsi, sillä pisteen F kautta piirretyt janat eivät jaa pisteen C kautta piirrettyjen janojen rajaamaa osa-alueita kokonaan.

2.2.3 PR-nelipuu

PR-nelipuussa (engl. Point, Region) on vaikutteita sekä alue- että pistenelipuusta. Siinä alueet on jaettu neljänneksiin aluenelipuun tapaan tasaisesti. Lehtisolmuilla voi puun määrittämisestä riippuen olla alueellaan yksi tai useampia pisteitä. Sellaiset lehtisolmut, joiden kuvaamalla alueella on vähintään yksi piste, ovat väriltään mustia, muut valkoisia. PR-nelipuussa alue ei siis jakaudu pisteiden kautta kulkevien janojen mukaan, mutta pisteiden sijainti vaikuttaa silti alueiden jakoon ja määrään: jos on määritetty, että kutakin lehtisolmuja vastaa yksi piste ja kaksi pistettä on hyvin lähellä toisiaan, on aluetta jaettava useaan kertaan, jolloin puusta tulee syvä. [14]



Kuva 3. Pisteiden mukaan jaoteltu alue ja sitä kuvaava PR-nelipuu. Muokattu alkuperäisestä [14, s. 2222].

Kuvassa 3 on esitetty alue, joka on pisteiden mukaan jaettu neliönmuotoisiin osa-alueisiin sekä aluetta kuvaava PR-nelipuu. Pisteiden tarkat sijainnit voitaisiin tallentaa tietorakenteessa koordinaatein tai vaikkapa etäisyyksinä niihin liitettyjen solmujen keskipisteistä.

2.3 Esimerkki nelipuusta

Nelipuu-tietorakenne voidaan ohjelmallisesti toteuttaa useilla eri menetelmillä riippuen puun tyypistä ja halutusta esitystavasta. Ohjelma 1 esittää olio-ohjelmoinnille mahdollisen luokan avulla tehtyä toteutusta aluenelipuusta: Nelipuusta on tehty luokka, jolla on juurisolmu-attribuutti. Solmun kuvaamista varten on määritelty tietorakenne, jonka yksi alkio sisältää tekstijonona kyseisen solmun värin ja ilmansuuntien mukaan nimetyt osoittimet sen lapsiin. Alustettaessa uusi solmu asetetaan sen osoittimien arvoiksi `nullptr`, mikä viittaa siihen, että kyseisellä solmulla ei ole lapsia.

```

struct solmu {

    //Solmun lapset englanninkielisten ilmansuuntien mukaan
    solmu *NW = nullptr;
    solmu *NE = nullptr;
    solmu *SE = nullptr;
    solmu *SW = nullptr;

    //Solmun väri
    std::string vari;
}

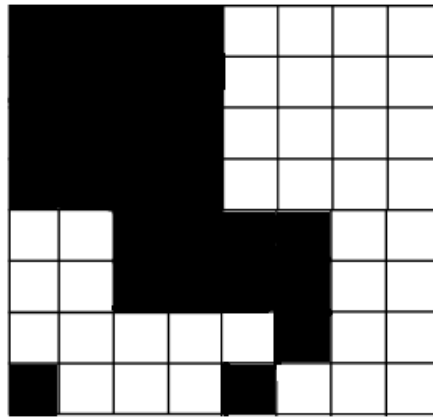
//Nelipuu-luokka
Class Nelipuu {

    public:
        solmu juuri;
        ...
}

```

Ohjelma 1. C++:lla tehty esimerkki nelipuusta luokkarakenteena. Kolme pistettä osoittavat, mihin muuta määrittelyä voitaisiin lisätä.

Ohjelma 1 on esitys pelkästä nelipuun rakenteesta. Nelipuu-luokalle voitaisiin antaa lisää määrittelyksiä ja toimintoja attribuutteina ja metodeina. Luokalle voitaisiin vaikkapa määrittellä metodi ”kieräytä”, joka muuttaa solmujen lasten ilmansuuntia yhden myötäpäivään. Tämä vastaisi kuvalle tehtävää 90 asteen kiertämistä myötäpäivään.



Kuva 4. Kuvassa 1 esitetty alue kierrettynä 90 astetta myötäpäivään.

Kuvassa 4 on havainnollistettu sitä, mikä vaikutus kuvassa 1 esitetyn nelipuun solmujen lasten ilmansuuntien siirtämisellä on sen kuvaamaan alueeseen. Kuvista nähdään, että alue on ikään kuin kierähtänyt 90 astetta myötäpäivään. Aluetta kuvaavassa nelipuussa on 21 solmua, joista 16 on lehtisolmuja, joten muutoksia olisi tehtävä jäljelle jäävissä 5 solmussa.

3. NELIPUUN KÄYTTÖ

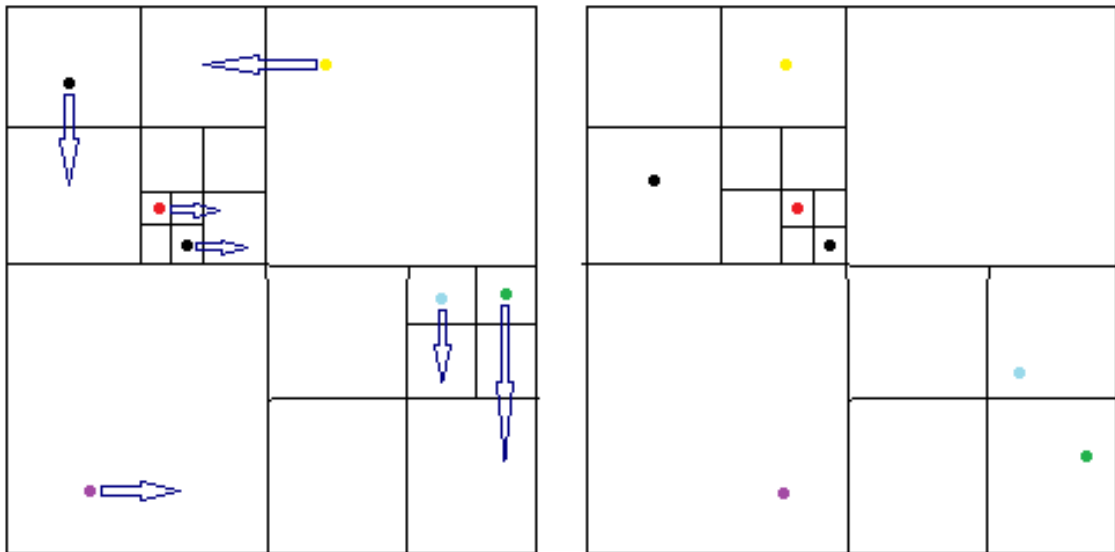
Nelipuutietorakenteen avulla voidaan luvussa 2 esitettyjen esimerkkien tavoin kuvata aluetta intuitiivisesti tietorakenteena. Tämä tekee nelipuusta potentiaalisen työkalun erilaisissa käytännön sovelluksissa, jotka tyypillisesti liittyvät alueen kuvaukseen tai pyrkiin myksiin tehostaa jotakin ohjelmallista operaatiota. Tässä luvussa esitellään nelipuun käyttöä kolmessa eri käytännön sovelluksessa: hiukkasten liikkeen simuloinnissa, tsunamiaallon mallinnuksessa ja ympäristömallinnuksessa. Lisäksi mainitaan lyhyesti muita kirjallisuudessa esitettyjä sovelluskohteita.

3.1 Nelipuu hiukkasten liikkeen simuloinnissa

Oğuz O., Durupınar F. ja Güdükbay U. esittelevät artikkelissaan [9] algoritmin, joka käyttää dynaamista PR-nelipuuta muokattavana tietorakenteena hiukkasten liikkeen simuloinnissa. Heidän mukaan käytännön sovelluskohteita simulaatiolle voisivat olla esimerkiksi väkijoukon, savun tai nesteiden liikkeiden tutkiminen.

3.1.1 Tekninen hyödyntäminen

Oğuz, Durupınar ja Güdükbay esittävät lähestymistavan, jossa jaetaan alue PR-nelipuuta käyttäen aliluvussa 2.2.3 esitettyyn tapaan ja asetetaan nelipuun lehtisolmuihin tallennetut pisteet kuvaamaan hiukkasia [9]. Hiukkasten massa, nopeus ja sijainti voidaan tallentaa nelipuun lehtisolmuihin. Hiukkasten liikettä kuvataan päivittämällä nelipuuta niiden sijaintien muuttuessa.



Kuva 5. Hiukkasten sijainteja alueella kuvaava nelipuu ennen (vas.) ja jälkeen siihen tehtävien päivitysten. Siniset nuolet osoittavat siirtymän kuvien välillä.

Kuvasta 5 havaitaan, miten nelipuu käyttäytyy hiukkasten liikkeessa alueella: tilanjako ja sen seurauksena koko nelipuun rakenne muuttuu siten, että jokaiselle hiukkaselle on oma solmunsaa. Kuvassa esitetty hiukkasten siirto ja siitä seuraava nelipuun päivittäminen eivät olisi kovinkaan tehokkaita, jos nelipuu päivitettäisiin jokaisen yksittäisen hiukkasen siirron jälkeen, koska sovelluksessa, johon Oğuz, Durupınar ja Güdükbay ovat algoritmin kehittäneet, jokaisen hiukkasen sijainti päivitetään jokaisella iteraatiolla. Siksi olennainen osa algoritmia on ensin päivittää kaikkien hiukkasten sijainti, sitten päivittää nelipuu kerralla. [9]

3.1.2 Soveltuvuus käyttökohteeseen

Oğuzin, Durupınarin ja Güdükbayn mukaan huonoimmassa skenaariossa käytettäessä tasaista ruudukkoa kaksiulotteisen alueen kuvauksessa tilavaatimus n datapisteelle on kokoluokkaa $O(n^2)$, kun käytettäessä PR-nelipuu se on $O(n)$ [9]. PR-nelipuu käyttämällä siis saadaan kuva tilanteesta joutumatta jakamaan koko aluetta esimerkiksi tasaisesti pieniin neliöihin. Jos hiukkaset ovat jakautuneet alueelle epätasaisesti, saadaan alueen eri osista havaintoja eri tarkkuuksilla ilman mittavaa tilantarvetta, joka ilmenisi tasaisesti jaettua ruudukkoa käytettäessä. Tilantarpeen kannalta on siis selvää, että PR-nelipuun käyttö kuvaillun kaltaisessa sovelluksessa kannattaa.

Pisteitä päivitettäessä PR-nelipuun käyttökelpoisuus esimerkiksi tasaisesti jaettuun ruudukkoon verrattuna ei ole yksiselitteistä. Jotkin operaatiot, esimerkiksi hiukkasten lisäys ja poisto, naapurin etsintä ja kyselyt, voivat olla ajallisesti vähemmän tehokkaita käytettäessä nelipuu kuin käytettäessä tasaisesti jaettua ruudukkoa [9]. Kuitenkin, Oğuz, Durupınar ja Güdükbay toteavat, että heidän kehittämänsä algoritmia käyttäen parhaassa

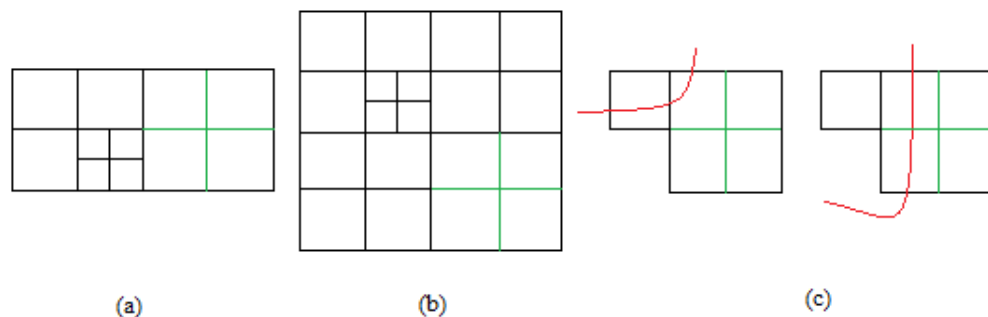
tapauksessa päästään datapisteiden määrällä n tehokkuuteen $O(n)$ ja huonoimmassa tapauksessa tehokkuuteen $O((d+1)n)$, missä d on nelipuun syvyys.

3.2 Nelipuu tsunamin mallinnuksessa

Popinet esittää tutkimuksessaan [11] tavan käyttää nelipuuta veden liikkeen ja sen peittämän alueen mallinnuksessa. Hän mainitsee potentiaalisiksi käyttökohteiksi jokien tulvimisen ja varsinkin laaja-alaisten tsunamien etenemisen hahmottamisen, mitä voidaan käyttää työkaluna riskien arvioinnissa, evakuoitaisuunnitelmissa ja reaaliaikaisissa ennusteissa tsunamihälytyksissä. Tutkimuksessaan Popinet esittää menetelmänsä toimintaa useissa testitapauksissa, sekä realistisessa vuoden 2004 Intian valtameren tsunamin mallinnuksessa.

3.2.1 Tekninen hyödyntäminen

Popinet'n esittämässä tekniikassa käytetään nelipuulle soveltuvia Saint-Venantin [12] yhtälöitä nesteen liikkeelle yhtälönratkaisuohjelmassa Gerris [10] hyödyntäen yleistettyä Adresse et al. [1] esittelemää tasapainotettua matalavan veden virtauksia kuvaavaa mallia [11]. Olennainen osa tekniikkaa on Gerrisin käyttämä nelipuun avulla tehtävä alueen jaottelu, joka mahdollistaa Popinet'n mukaan tehokkaan, mukautuvan hienosäädön. Menetelmä käyttää hieman muokattua versiota aliluvussa 2.2.1 esitellystä aluenelipuusta.



Kuva 6. Gerrisin käyttämän aluenelipuun ylimääräiset rajoitteet. Vihreät viivat kuvaavat kunkin ehdon täyttymiseen tarvittavia tarkennuksia ja punaiset viivat kiinteää rajaa. Perustuu lähteeseen [10, s. 4].

Kuvassa 6 on havainnollistukset kolmesta rajoitteesta, jotka Popinet on lisännyt koskemaan Gerrisin käyttämää nelipuuta. Nämä rajoitteet ovat seuraavat: (a) kaksi vierekkäistä solmua eivät voi erota syvyydeltään yhtä astetta enempää, (b) kaksi kulmistaan kohtaavaa solmua eivät voi erota syvyydeltään yhtä astetta enempää ja (c) kaikkien kiinteää solmua sivuavien solmujen syvyyden on oltava sama [10]. Kiinteä solmu määritellään tässä yhteydessä niin, että solmut, jotka leikkaa kiinteä raja, ovat kiinteitä. Popinet'n [10] mukaan näiden rajoitusten tarkoitus on yksinkertaistaa gradientin ja vuon laskentaa sovelluksessa.

Laskettaessa veden virtaamia alueelta toiselle Popinet'n kehittämä menetelmä määrittää (Eulerin yhtälöiden) muuttujien tilan kunkin osa-alueen (eli nelipuun solmun) keskikohdassa, jonka jälkeen voidaan yksinkertaisemmin laskea virtaus kahden pisteen välillä. Rajoitteiden seurauksena tilanne kahden solun esittämän alueen välillä on jompikumpi seuraavista: vierekkäiset solmut ovat yhtä suuria eli samalla syvyydellä tai suurempaa aluetta sivuaa kaksi pienempää aluetta, joiden solut ovat asteen syvemmällä [10]. Näin ollen algoritmin täytyy käytännössä osata laskutoimitukset vain yhdeltä alueelta toiselle ja kahdelta alueelta yhteensä.

3.2.2 Soveltuvuus käyttökohteeseen

Popinet'n [11] mukaan tsunamit ovat vaikeita mallintaa, koska ne tapahtuvat laajoilla, valtamerten kokoisilla alueilla, mutta niiden tuhoava vaikutus voi kuitenkin olla riippuvainen muutamien metrien korkuisista esteistä. Tästä syystä niiden käytännöllinen mallintaminen koordinaattipohjaisilla ruudukoilla, joissa alue on tasaisesti jaettu, vaatisi suuria määriä laskentatehoa yhdistettynä rinnakkaisohjelmointiin ja rajoittaisi tsunamiennusteiden kapasiteettia. Vaihtoehto alueen jakamiselle tasaisesti olisi käyttää epätasaisesti jaettua ruudukkoa, jossa esimerkiksi rannikkoalueet olisi tarkemmin rajattu kuin alueet syvällä merellä. Ongelma tässä lähestymistavassa on Popinet'n mukaan se, että tarkka tsunamiaallon korkeuden määrittäminen syvällä merellä voi vaatia tarkempaa alueen jaottelua myös siellä, jolloin pitäisi tietää etukäteen, minne tarkka jaottelu tulisi kohdistaa. [11, s. 1261]

Nelipuu mahdollistaa mukautuvan ja tehokkaan tavan jaotella aluetta eri tarkkuuksilla, joten se mahdollistaa tsunamimallinnuksen tarkkuudella, joka olisi epäkäytännöllinen tasaisesti jaetulla ruudukolla. Käytettäessä tasaista alueen jaottelua tilantarve on kertaluokkaa 10 suurempi kuin käytettäessä mukautuvaa alueen jaottelua. Merkillepantavaa on, että tilantarpeen kasvaessa eksponentiaalisesti, jaettaessa alue tasaisesti eksponentiksi tulee 2, kun taas jaettaessa alue mukautuvasti eksponentiksi tulee 1,4. Mitä tarkemmin aluetta siis kuvataan, sitä tehokkaammin sovelluksessa käytetty nelipuuhun perustuva malli käyttää muistia verrattuna alueen tasaisesti jakavaan malliin. Edellä mainitut tiedot on saatu Popinet'n tekemästä analyysistä, jossa hän on mallintanut vuonna 2004 Intian valtamereillä tapahtunutta tsunamia. Käytetyt luvut on saatu simulaation ajallisesta keskiarvosta. [11]

Nelipuun käyttö aiheuttaa Popinet'n mukaan myös ongelmia, koska nelipuuta käytettäessä alue on paikallisesti jaettu isotrooppisesti, eli saman kokosiin osa-alueisiin, mikä monimutkaistaa sen käyttöä ei-isotrooppisten virtausten laskennassa [10, s. 4]. Toinen nelipuuta koskeva ongelma liittyy Popinet'n mukaan siihen, että mukautuvaa alueen jaottelua käytettäessä tietokantaa on muutettava joka kerralla, kun jaottelua muutetaan, mikä asettaa vaatimuksia tietokannan tehokkuudelle [11, s. 1267-1268]. Tästä syystä sovelluksessa on jouduttu tekemään kompromissi liittyen vedenpinnan korkeuden ja kunkin

solun veden tilavuuden tallennuksessa [11, s. 1268]. Vaikka nelipuu siis mahdollistaa tarkemman ja tehokkaamman tsunaminmallinnuksen, se jossain määrin monimutkaistaa mallinnusta verrattuna esimerkiksi tasaisesti jaettuun ruudukkoon.

3.3 Nelipuu ympäristömallinnuksessa

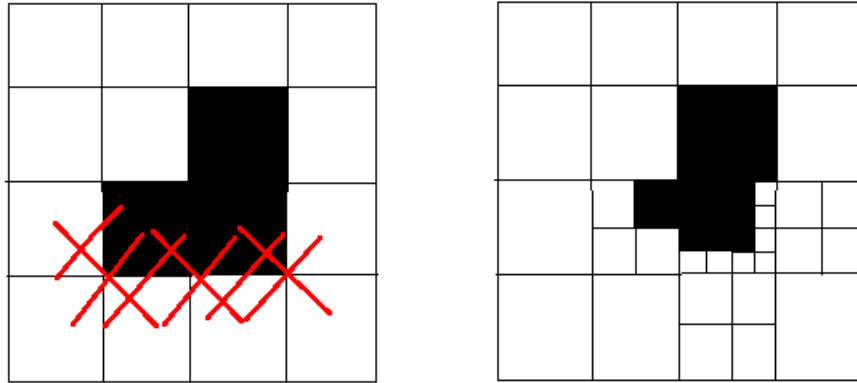
Tässä aliluvussa tarkoitetaan ympäristömallinnuksella kehittyneiden ajoneuvojen ja robottien luomaa mallia ympäristöstään. Esimerkiksi itseohjautuvat autot voivat olla esimerkki tällaisista kulkuvälineistä, jotka luovat sensoripohjaisen syötteen mukaan kuvaa ympäristöstään. Y. Li ja Y. Ruichek esittelevät artikkelissaan [5] nelipuuhun pohjautuvaa muuttuvan tarkkuuden ratkaisua ympäristömallinnukseen.

3.3.1 Tekninen hyödyntäminen

Li ja Ruichek käyttävät [3] esittelemän kaltaista nelipuuta, eli aliluvussa 2.2.1 esitellyn kaltaista aluenelipuuta, jonka kuhunkin lehtisolmuun tallennetaan sensorien mittaama etäisyys tiedetyssä (nelipuun juurisolmun kuvaamassa) suunnassa. Li ja Ruichek antavat käyttämänsä nelipuun lehtisolmuille yhden kolmesta arvosta: vapaa, ei havaintoa ja tukkeutunut. Nämä arvot viittaavat siihen onko lehtisolmun osoittamassa suunnassa esteitä. [5]

Nopeuttaakseen nelipuupohjaista ratkaisuaan Li ja Ruichek käyttävät niin kutsuttua ”Morton-” tai Z-koodia [8], joka mahdollistaa useamman ulottuvuuden tiedon muokkaamisen yhteen ulottuvuuteen lokaalisuuden kärsimättä. Piste Z-koodin haku perustuu siihen, että sen sijainnin osoittavien arvojen numerot limitetään keskenään. Piste (3, 2) haku tapahtuu siis yhdistelemällä limittäin lukujen 3 (11) ja 2 (10) binäärimuodon numerot, mistä saadaan luku 1110.

Li ja Ruichek muokkaavat esimerkissään nelipuuta kohdennettuun alueeseen, jolla nelipuun solmuja joko yhdistetään, jos solmun kuvaaman alueen tietojen tarkkuutta katsotaan voitavan vähentää, tai jaetaan pienempiin osiin, jos arvioidaan, että alueen tarkkuutta on lisättävä. Alueen jakaminen perustuu siihen, onko sen naapurialueilla arvioitu sama tukkeutuneisuus. Jokaiselle nelipuun lehtisolmun kuvaamalle alueelle lasketaan todennäköisyys, jolla siinä esiintyy esteitä, eli joilla se on tukkeutunut. Jos alue ja sen naapurit eroavat laskelmien perusteella, suoritetaan alueella jako pienempiin osakokonaisuuksiin, siis nelipuun solmu jaetaan neljään lapseen. Tätä toimenpidettä toistetaan, kunnes eroavaisuudet alueiden välillä ovat halutulla minitasolla. Jos taas useilla toistensa naapurialueilla havaitaan sama tukkeutuneisuus laskelmien pohjalta, ne yhdistetään, eli nelipuutasolla poistetaan lehtisolmut, jonka jälkeen niiden vanhempi esittää niiden kuvaamaa aluetta. Tällä tavoin muistintarve pienenee. [5]



Kuva 7. Punaisilla viivoilla merkityt alueet arvioidaan jaettaviksi. Oikealla esimerkki mahdollisesta tarkennetusta alueesta jaon jälkeen. Perustuu lähteeseen [5].

Kuvassa 7 on havainnollistus alueen jaosta tilanteessa, jossa katsotaan sille olevan tarvetta. Mustan alueen reunamilla on osa-alueita, jotka eroavat tarkkuuksiltaan naapureistaan. Näin ollen nämä alueet jaetaan pienempiin ja taas pienempiin osa-alueisiin, mikä nelipuuta käytettäessä tapahtuu lisäämällä niitä kuvaaville solmuille lapsisolmut, joille jälleen lasketaan uudelleen todennäköisyys tukkeutuneisuudelle.

3.3.2 Soveltuvuus käyttökohteeseen

Li ja Ruichek esittävät artikkelissa [5] tuloksia testistä, joka on suoritettu ajoneuvon päälle asennetun Bumblebee XB3 -järjestelmän antamalla syötteellä ympäristöstä ajettaessa Belfortissa Ranskassa. Heidän saamiensa tulosten perusteella nelipuulla jaotellun alueen tarkkuus on yli kymmenkertainen verrattuna vastaavalla syötteellä saatuun alueeseen käytettäessä tasaisesti jaettua ruudukkoa. Nelipuulla toteutettu aluejako vaikuttaa siis tässä tapauksessa selvästi yleisesti käytettyä tasaista ruudukkoa tarkemmalta.

Nelipuun heikkoudeksi verrattaessa tasaisesti jaettuun ruudukkoon Li ja Ruichek mainitsevat hitauden haettaessa solmua käsittelyyn. Vaikka he toteavatkin [5] Z-koodin käytön solmun haussa tehostavan operaatiota, mitään matemaattista vertailua tai osoitusta sen nopeudesta ei artikkelissa ole esitetty.

3.4 Muita käyttökohteita

Tässä luvussa esiteltyjen sovellusten lisäksi nelipuulle on muitakin käyttökohteita ja esityksiä sen hyödyntämiseksi. Yleisiä sovellusalueita ovat kuvankäsittely ja eri aineiden, lähinnä nesteiden liikkeen simulointi. Näissä sovelluskohteissa nelipuun eduksi katsotaan usein tehokas tiedon esitystapa ja sen käytön mahdollistamat algoritmit.

Mairal, Sapiro ja Elad hyödyntävät nelipuun tehokasta tiedonjakotapaa esityksessään [7] kuvien ja videoiden restaurointisovellusten parannukseksi. Cheng ja Li esittävät [2, s. 2441] kuvan kompressointia nelipuun avulla laskentateholtaan alhaisissa laitteissa sille soveltuvan vähäisten bittimäärien nopean käsittelyalgoritmin ansiosta. Liang ja Borthwick käyttävät [6] nelipuuta matalien virtausten, kuten tulvien, simuloinnissa topologisesti monimutkaisilla alueilla ja mainitsevat [6, s. 222] useita muita esityksiä nelipuun käytöstä vastaavanlaisissa käyttökohteissa. Tyypillisestä hieman poikkeava käyttökohde nelipuulle on Ungerin, Eckardtin ja Könken esittämä [15] nelipuuta hyödyntävä betoni-rakenteiden halkeamien kartoitusmenetelmä. Tässäkin tapauksessa nelipuun eduksi kuitenkin todetaan [15] sen mahdollistama halkeamien esitystavan tarkkuuden mukautuvuus.

4. TULOSTEN ARVIOINTI

Nelipuun käyttökelpoisuuden puolesta on luvussa 3 esitetty perusteluiksi laskennallisesti ja kokeellisten tulosten avulla saatuja tuloksia. Aliluvussa 3.1.2 käsitellyssä Oğuzin, Dupinarin ja Güdükbayn tutkimuksessa [9] on tultu tulokseen, että tilankäytöllisesti nelipuu on selvästi tasaisesti jaettua ruudukkoa tehokkaampi. Aliluvussa 3.2.2 käsitellyssä Popinet'n tutkimuksessa [11] on niin ikään on havaittu, että nelipuutietorakenne vähentää tilantarvetta merkittävästi varsinkin suurilla tietomäärillä. Myös aliluvussa 3.3.2 käsitelty Lin ja Ruichekin tutkimus [5] osoittaa, että nelipuun käytön mahdollistama kohdistettu tarkennus mahdollistaa huomattavasti tasaisesti jaettua ruudukkoa tarkemman ympäristön mallintamisen. Näiltä osin on osoitettu, että nelipuutietorakenteella on tasaisesti jaettuun ruudukkoon nähden joillain mittareilla mukautuvuutensa ansiosta selviä etuja erityyppisten ilmiöiden mallinnuksessa.

Nelipuun käytön mahdollisia haittapuolia ovat esimerkiksi aliluvussa 3.1.2 mainitut päivitysoperaatiot, aliluvussa 3.2.2 mainitut tehokkuusvaatimukset, sekä aliluvussa 3.2.3 mainittu hitaus solmun käsittelyyn haussa. Nämä ongelmat viittaavat siihen, että vaikka nelipuu vaikuttaa usein tilankäytön kannalta suotuisalta tekniikalta, se ei aina ole suorituskyvyn kannalta optimaalisin tekniikka.

5. YHTEENVETO

Tämän tutkimuksen tavoitteena oli käsitellä nelipuuta niin käsitteenä ja tietorakenteena yleisemmin kuin erilaisina käytännön sovelluksiin muokattuina versioina. Tarkoituksena oli, että tutkimuksen luettuaan lukija ymmärtää nelipuun sekä abstraktina käsitteenä että tarkemmin määriteltyinä tietorakenteina. Lisäksi tavoitteena oli, että lukija hahmottaa nelipuun toimintaa muutamissa käytännön tekniikoissa ja voi sitä kautta itse arvioida sille muitakin käyttökohteita.

Tutkimuksessa todetaan nelipuun voivan tarkoittaa montaa asiaa geometrisesta entiteetistä tarkasti määriteltyihin toisistaan eroaviin tietorakenteisiin, joilla kaikilla on kuitenkin yhteisiä piirteitä, joiden vuoksi niitä voidaan kutsua nelipuiksi. Näihin piirteisiin kuuluu vähäisimmällä määrittelyllä se, että nelipuulla on juurisolmu ja sen solmuilla on lapsia joko 4 tai 0. Tutkimuksessa esitellään tarkemmin kolme nelipuun tyyppiä: aluenelipuu, pistenelipuu ja PR-nelipuu, sekä joitain käyttökohteita näille. Lisäksi esitellään tarkemmin kolme käytännön sovellusta, joissa jonkinlaista versiota nelipuusta on käytetty. Nelipuun soveltumista näihin sovelluksiin arvioidaan ja todetaan sen olevan varteenotettava vaihtoehto ainakin tasaisesti jaetulle ruudukolle niin esitellyissä sovelluksissa kuin yleisemminkin. Nelipuun vahvuuksiksi katsotaan ainakin mukautuvuus ja sen mahdollistama tarkkailun kohdentaminen eri ilmiöitä mallinnettaessa, intuitiivisuus ja vähäinen tilankäyttö. Nelipuun heikkouksiksi voidaan katsoa toteutuksen monimutkaisuus, heikko suorituskyky joissain sovelluksissa, ja tietoalkioiden käyttöön hakemisen hitaus.

Tutkimuksen tuloksia tarkasteltaessa on otettava huomioon se, että lainatuissa teksteissä nelipuuta verrattiin lähinnä tasaisesti jaettuun ruudukkoon. Vaikka siis nelipuu osoitetaan niissä joiltain osa-alueilta paremmaksi vaihtoehtoehdoksi tiedon esittämiseen, tämän tutkimuksen pohjalta ei voida todeta sen olevan paras mahdollinen vaihtoehto.

Tiivistetysti voidaan sanoa, että nelipuuta kannattaa harkita käytettäväksi tilanteissa, joissa halutaan mallintaa kaksiulotteista geometrista entiteettiä, varsinkin jos käytettävä tieto on jakautunut alueelle epätasaisesti, tietoa on erittäin paljon tai jos mallinnus halutaan tehdä osalle alueesta tarkemmin kuin toiselle. Nelipuun selkein etu on tehokas mukautuva ilmaisu sen tilankäyttöön nähden. Nelipuun käyttöä harkittaessa kannattaa kiinnittää huomiota sen alkioiden käsittelyn ja sen muodon muuttamiseen tarvittavien algoritmien monimutkaisuuteen ja mahdolliseen epätehokkuuteen.

LÄHTEET

- [1] E. Audusse, F. Bouchut, M-O. Bristeau, R. Klein, B. Perthame, A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows. *SIAM J. Sci. Comput.*, vol. 25, issue 6, 2004, pp. 2050–2065.
- [2] H. Cheng, X. Li, Partial Encryption of Compressed Images and Videos, *IEEE Transactions on Signal Processing*, vol. 48, issue 8, 2000, pp. 2439-2451.
- [3] R. A. Finkel, J. L. Bentley, Quad Trees: a data structure for retrieval on composite keys, *Acta Informatica*, vol. 4, 1974, pp. 1-9. Saatavissa: https://www.researchgate.net/profile/Raphael_Finkel/publication/220197855_Quad_Trees_A_Data_Structure_for_Retrieval_on_Composite_Keys/links/0c9605273bba2ece7b000000.pdf
- [4] H. Karutz, W. B. Kraetzig, A quadtree data structure for the coupled finite-element /element-free Galerkin method, *International Journal for Numerical Methods in Engineering*, vol. 53, issue 2, 2002, pp. 375-391.
- [5] Y. Li, Y. Ruichek, Building variable resolution occupancy grid map from stereoscopic system — A Quadtree based Approach, *Intelligent Vehicles Symposium (IV)*, 2013, pp. 744-749. Saatavissa: https://www.researchgate.net/publication/261245976_Building_variable_resolution_occupancy_grid_map_from_stereoscopic_system_-_A_quadtree_based_approach
- [6] Q. Liang, A. G. L. Borthwick, Adaptive quadtree simulation of shallow flows with wet-dry fronts over complex topography, *Computers and Fluids*, vol. 38, issue 2, 2009, pp. 221-234.
- [7] J. Mairal, G. Sapiro, M. Elad, Learning Multiscale Sparse Representations for Image and Video restoration, *Multiscale Modeling and Simulation*, vol. 7, issue. 1, 2007, pp. 214-241.
- [8] G. M. Morton, A computer oriented geodetic data base and a new technique in file sequencing. International Business Machines Company, 1966. Saatavissa: <https://domino.research.ibm.com/library/cyberdig.nsf/0/0dabf9473b9c86d48525779800566a39?OpenDocument>
- [9] O. Oğuz, F. Durupinar, U. Güdükbay, Dynamic point-region quadtrees for particle simulations, *Information Sciences*, vol. 218, 2013, pp. 133-145. Saatavissa: <https://pdfs.semanticscholar.org/a026/3199aed83c23937b20974269f486ac3873c0.pdf>

- [10] S. Popinet, Gerris: a tree-based adaptive solver for the incompressible euler equations in complex geometries, *J Compute Phys* 190(2):572–600, 2003. Saatavissa: <http://gfs.sf.net/gerris.pdf>
- [11] S. Popinet, Quadtree-adaptive tsunami modelling, *Ocean Dynamics*, vol. 61, issue 9, 2011, pp. 1261-1285. Saatavissa: <https://hal.archives-ouvertes.fr/hal-01445423/document>
- [12] A. B. Saint-Venant, Théorie du mouvement nonpermanent des eaux, avec application aux crues des rivières et à l'introduction des marées dans leur lit, *C R Acad Sci Paris* 73, 1871, pp. 147–154.
- [13] H. Samet, The Quadtree and Related Hierarchical Data Structures, *ACM Computing Surveys*, vol. 16, issue 2, 1984, pp. 187-260. Saatavissa: <http://legacydirs.umiacs.umd.edu/~ramani/cmssc878R/p187-samet.pdf>
- [14] T. Tzouramanis, M. Vassilakopoulos, Quadtrees (and Family), *Encyclopedia of Database Systems*, 2009, pp. 2219-2225.
- [15] J. F. Unger, S. Eckardt, C. Könke, Modelling of cohesive crack growth in concrete structures with the extended finite element method, *Computer methods in applied mechanics and engineering*, vol. 196, issue 41-44, 2007, pp. 4087-4100. Saatavissa: <https://link.springer.com/article/10.1007%2Fs10704-009-9380-1>