

Jesse Kananen

Grafiikkasuorittimien kehityksen suuret linjat

Tietotekniikan kandidaatintutkielma

10. elokuuta 2016

Jyväskylän yliopisto

Tietotekniikan laitos

Tekijä: Jesse Kananen

Yhteystiedot: jesse.j.kananen@jyu.fi

Työn nimi: Grafiikkasuorittimien kehityksen suuret linjat

Title in English: Outline of the History of the Graphical Processing Unit

Työ: Kandidaatintutkielma

Sivumäärä: 23+0

Tiivistelmä: Näytönohjaimien ja niitä vauhdittavien grafiikkasuorittimien kehitys on viime vuosina ollut huomattavaa. Tässä opinnäytetyössä perehdytään suorittimien yleiseen kehityskulkuun sekä laitteiston kehitykseen alkeellisista 3D-laitteista moderneihin grafiikkasuorittimiin. Samalla etsitään tätä teknologiseen edistykseen merkittävästi myötävaikuttaneita uudistuksia. Päälähteenä toimivat alalta jo tehdyt historiakatsaukset, joita tukevat tutkimusmateriaali sekä valmistajien tekniset julkaisut. Lopputuloksena aineiston perusteella on kehityslinjan kannalta merkittäviksi tekijöiksi nostettavissa grafiikkasuorittimien oma prosessori, shaderit, ohjelmoitavuus sekä lopulta rinnakkaislaskenta.

Avainsanat: grafiikkasuorittimet, tietotekniikan historia, laitteisto

Abstract: The development of video cards and their graphical processing units (GPU) has been significant during the latest years. In this thesis the focus is on the general outline of the development of the hardware from simple 3D-frame buffers to modern GPUs and the development process in the general. The main goal is to find the most important innovations. The sources are mainly based on existing histories of the subject which are supported by other academic research proceedings and publications by the hardware manufacturer. The results are generalized to the significance of the GPU processors, shaders, programmability and parallel processing to the evolution of the whole industry.

Keywords: graphical processing unit, history, hardware

Kuviot

- Kuvio 1. Datan liukuhihnaprocessoinnin yleistetty malli. *Vertex Transformation* tarkoittaa kärkipisteiden muodostamista, *Primitive Assembly* kokoaamisvaihetta, *Rasterization and Interpolation* rasterointia ja väliarvolaskua, *Raster Operations* rasterointia ja *Frame Buffer* kuvan puskurimuistiin tallentamista. Lähde: <URL: <http://www.seas.upenn.edu/cis565/LECTURES/Lecture2%20New.pdf>> Viitattu 5.4.2016 4
- Kuvio 2. Shaderit osana datan prosessointia. *Programmable Vertex Shader* tarkoittaa ohjelmoitavaa kärkikolmioita käsittelevää shaderia, *Programmable Fragment Processor* taas ohjelmoitavaa sirpaleprosessoria ja *Texture Memory* tekstuuri-muistia. Muiden termien suomennotkatso kuvan 1 kuvateksti. Lähde: <URL: <http://www.seas.upenn.edu/cis565/LECTURES/Lecture>> Viitattu 5.4.201610

Sisältö

1	JOHDANTO	1
2	GRAFIIKAN PIIRTÄMISEN PERUSPERIAATTEET	3
	2.1 Kiinteä liukuhihnaprozessointi	4
	2.2 Ensimmäiset grafiikkapiirtimet	5
	2.3 3D-korteista kiihdyttimiin	6
3	GRAFIKKASUORITTIMIEN NOUSU	8
	3.1 Ensimmäinen grafiikkasuoritin	8
	3.2 Siirtyminen ohjelmoitavaan prosessointiin.....	9
4	GRAFIKKALAITTEISTON NYKYHETKI.....	11
	4.1 NVIDIA lanseeraa Tesla-arkkitehtuurin	11
	4.2 Siirtyminen rinnakkaisprosessointiin	12
	4.3 Rinnakkaislaskenta - tulevaisuus nyt?	13
5	YHTEENVETO	15
	KIRJALLISUUTTA	18

1 Johdanto

Tässä tutkielmassa keskitytään tarkastelemaan 3D-piirtämiseen soveltuvien grafiikkasuorittimien kehitystä sekä tutkimuskirjallisuuden kehityslinjan katsausten että aikalaislähteiden, kuten aiheeseen liittyvien artikkeleiden ja valmistajien esittelyiden, pohjalta. Kaksiulotteisen grafiikan sijaan kolmiulotteisuus on valittu tämän opinnäytetyön tarkasteluun grafiikkapiirron toteuttamiseen tarvittujen laskentavaatimusten vuoksi. Lisäksi käytännössä kaikki näyttöpäätteelle piirretty grafiikka on kaksiulotteista — kolmiulotteisesta mallista transformoitua — kuvaa. Laite, joka pystyy kolmiulotteiseen piirtämiseen, kykenee siis myös kaksiulotteiseen. Kolmiulotteisuus on myös markkinoilla vahvassa ja kasvavassa asemassa. Tästä syystä myös huomio kiinnittyy kuluttajille suunnattujen laitteiden kehitykseen.

Tutkimusmenetelmänä tässä opinnäytetyössä on kirjallisuuskartoitus, jonka perusteella löydetty lähdemateriaali jäsennellään, luokitellaan ja yhdistellään yhtenäiseksi kokonaisuudeksi. Lähdeaineiston perusteella laaditaan siis yleisesitys grafiikkasuorittimien synnystä, kehityksestä ja nykyhetkestä. Samalla nostetaan esille aineiston mukaan laitteiden kehitykseen merkittävimmin vaikuttaneita elementtejä.

Tässä opinnäytetyössä kirjallisuuskartoituksen ydin asettuu McClanahan (2010) grafiikkasuorittimen historiasta kertovan katsauksen hahmottelemana. Tämä tutkimus antaa eräänlaisen tarinarunon, jota täydennetään muilla aiheeseen oleellisesti liittyvillä tieteellisillä julkaisuilla. Näitä muita lähteitä ovat esimerkiksi alan aikakausjulkaisuissa ilmestyneet artikkelit, merkittävimmissä konferensseissa esitetyt keynote-puheet sekä laitevalmistajien omat julkaisut, kuten uusista innovaatioista julkistetut whitepaper-esittelyt.

Tämä opinnäytetyö pyrkii vastaamaan materiaalin pohjalta seuraaviin kysymyksiin: Kuinka grafiikkasuorittimet kehittyivät? Mitkä teknologiset innovaatiot veivät kehitystä eteenpäin? Mikä on grafiikkasuorittimien nykytilanne?

Tämän opinnäytetyön ensimmäisessä luvussa tarkastellaan grafiikkasuorittimen kehityksen alkuvaiheita. Toisessa luvussa syvennyttään ensimmäisiin varsinaisiin gra-

fiikkasuorittimiin ja sen hyödyntämään liukuhihnaprosessointiin. Kolmannessa luvussa perehdytään prosessointimallin ohjelmoitavuuden murrokseen ja sen tuomiin uusiin mahdollisuuksiin. Neljännessä luvussa taas paneudutaan grafiikkasuorittien uusimpiin tekniikoihin ja käyttötarkoituksiin. Lopuksi koko kehityskaari tuodaan yhteen johtopäätöksissä.

2 Grafiikan piirtämisen peruseriaatteen

Kolmiulotteista kuvaa on nykypäivän tietotekniikassa käytössä runsaasti aina peruskäyttöliittymästä lähtien upeisiin fysiikkamallinnuksella varustettuihin simuloituihin reaali maailmoihin asti. Tässä luvussa perehdytään 3D-kuvan piirtämisen perusteisiin yksittäisestä tietoarvosta kohti näytölle piirrettävää kokonaiskuvaa.

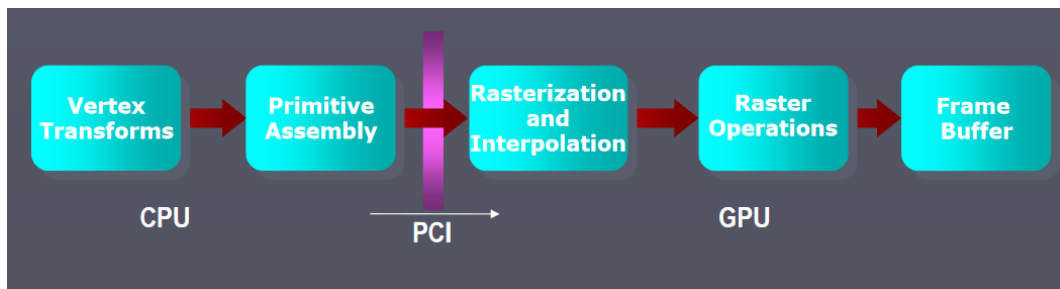
Piirtäminen datasta 3D-kuvaksi lähtee Owensin ym. mukaan grafiikkaa piirtävälle laitteistolle annetusta syöttödatasta, joka on siis lista alkukantaisia geometrisia koordinaatteja 3D-maailmaan. Tässä ensimmäisessä vaiheessa (Vertex Operations) tästä syöttöarvodatasta muodostetaan kärkipisteitä (vertice). Yksittäiselle kärkipisteelle lasketaan oma paikkansa piirrettävältä ruudulta. Samalla sille lasketaan myös sävytys valaistusarvojen mukaisesti. Toisessa eli kokoamisvaiheessa (Primitive Assembly) kärkipisteet yhdistetään toisiinsa ja niiden avulla luodaan pintoja usein kolmioista koostuen. Tämän jälkeen kolmioiden perusteella hahmotellaan mitkä kolmiot kuuluvat minkäkin pikselin alle näyttöalueella. Lopputuloksena rasterointivaiheesta (Rasterization) on datatyypinä sirpale (fragment), joka pitää sisällään jokaisen kolmion näytöltä peittämät pikselit. Huomioitavaa on, että pikseli voi olla useamman kolmion peittämä, joten sen väriarvo koostuu useasta eri sirpaleesta. Seuraavassa vaiheessa (Fragment Operations) sirpaleista ja mahdollisesti piirrettäville pinnoille asetettavista tekstuurereista muodostetaan kuva asettamalla jokaiselle sirpaleelle nyt oma valotuksensa aiemman laskennan perusteella. Lopuksi suoritetaan kompositio (Composition), jonka lopputuloksena on jokainen pikseli kerrallaan luotu 2D-kokonaiskuva. Tämä malli on säilynyt koko kehityskaaren ajan samanlaisena. (Owens, Houston, Luebke, Green, Stone & Phillips 2008, ss. 880–881)

Grafiikan piirtämiseen käytetty laitteisto koostuu pääosin näytönohjaimeen sisällytetystä grafiikkasuorittimesta, Video BIOS-ohjelmistosta ja erillisestä näyttömuistista. Grafiikkasuoritin tekee varsinaisen laskentatyön, VBIOS sisältää tiedot esimerkiksi suorittimen ja muistien jännitteistä ja näyttömuisti taas pitää sisällään erilaista tallennettavaa ja nopeasti käyttöön tarvittavaa tietoa.

Grafiikkalaitteistosta löytyvät lähes kaikki tietokoneen itsessään oleelliset elementit, mutta sen ero tietokoneen suorittimeen on datan käsittelyn ja syöttämisen tyylissä. Keskusyksikön arkkitehtuuri perustuu yksisäikeiseen datan kuljettamiseen läpi koko yksisäikeisen prosessoinnin yhtä muistirajapintaa käyttäen. Grafiikkasuorittimissa data tulee prosessoitavaksi ikään kuin massiivisena virtana. Näin ollen grafiikkasuoritin koostuu useista toisiinsa kiinteästi linkitetyistä datavirtaa käsittelevistä prosessoreista. Se siis toimii jatkuvasti kaikkia osiaan hyödyntäen. (Crow 2004, s.20) Tämä arkkitehtuuriero on johtanut merkittäviin eroihin myös suorittimen suorituskyvyssä.

2.1 Kiinteä liukuhihnaprosessointi

Data suodatetaan grafiikkalaitteiston läpi kiinteässä liukuhihnaprosessoinnissa (fixed function-pipeline). Siinä käsiteltävälle datalle suoritettavat operaatiot tehdään peräkkäisesti jokainen elementti kerrallaan. Data putkitetaan läpi grafiikkalaitteiston siten, että se on jälleen käyttäjän muokattavissa vasta koko prosessimallin läpikäytyään. Liukuhihnaprosessin malli on kuvattu yksinkertaistetusti kuvassa 1.



Kuvio 1. Datan liukuhihnaprosessin yleistetty malli. *Vertex Transformation* tarkoittaa kärkipisteiden muodostamista, *Primitive Assembly* ko-koamisvaihetta, *Rasterization and Interpolation* rasterointia ja väliarvolas-kaa, *Raster Operations* rasterointia ja *Frame Buffer* kuvan puskurimuistiin tallentamista. Lähde: <URL: <http://www.seas.upenn.edu/cis565/LECTURES/Lecture2%20New.pdf>> Viitattu 5.4.2016

2.2 Ensimmäiset grafiikkapiirtimet

Ennen varsinaisia grafiikkapiirtimiä ensimmäiset grafiikan piirtämistä tehostavat laitteet olivat lähinnä puskurimuisteja eli erillisiä muistipaikkoja piirrettäville kuville. Niiden tarvitsemat laskuoperaatiot suoritettiin täysin tietokoneen omalla prosessorilla ja se suoritti myös piirtämisen. Vain lasketut, piirrettävät kuvat asetettiin muistiin nopeampaa käyttöä varten. Hiljalleen varsinaista pikseleillä suoritettavaa piirtämistyötä jaettiin tietokoneen prosessorilta erillisten integroitujen grafiikkasuorittimien tehtäväksi. Tämä vapautti arvokkaita kellokierroksia varsinaiselta keskusyksiköltä. (McClanahan 2010, s.2)

Eniten rajoituksia 3D-piirtämisen nopeudessa 80-luvulla aiheuttivat niin sanotut floating point eli liukulukuarvot, sillä liukulukujen laskeminen pelkällä keskusyksiköllä oli hidasta. Tämä datatyyppejä toimi kuitenkin tarkkuutensa vuoksi objektien 3D-transformoinnin pohjana. Ongelman vaikutus väheni, kun liukulukuja laskevan yksikön käyttö tietokoneiden prosessoreiden rinnalla yleistyi. Liukulukuyksiköiden käyttöönotto mahdollisti pintojen sävyttämisen (shading) myötä siirtymistä kohti valokuvamaisempaa lopputulosta. (Grimes, Kohn & Bharadhwaj 1989)

Kehittyvät prosessorit antoivat ajatusta uudelle teknologialle myös grafiikkakorttien suhteen. Muuttuvasta ajatusmallista kohti erillistä grafiikkayksikköä kertoo esimerkiksi se, että vuonna 1989 julkaistussa, erillisen 3D-prosessorin sisältäneessä Intel i860-prosessorissa nähtiin potentiaalia myös muiden 3D-prosessorien pohjaksi erityisesti liukulukulaskentatehonsa vuoksi, mutta esimerkiksi Wang ym. näkivät prosessorin ongelmaksi liiallisen tarpeen konekielisille käskyille. Grafiikkalaitteiston hyödyntämisen tulisi olla yksinkertaisempaa (Wang, Mangaser & Srinivasan 1992).

Näitä liukulukuarvoja käyttämällä tietokoneen prosessori suorittaa alkuvalmistelut luomalla syötetystä kärkipistedatasta kolmiot, joille grafiikkasuoritin työstää muut tarvittavat operaatiot piirtämistä varten. Esimerkiksi IBM:n 1984 julkaisema omalla prosessorilla varustettu IBM XT-tietokoneeseen asennettava, erityisesti CAD-piirtämiseen tarkoitettu grafiikkakortti Professional Graphics Controller (PGA) vapautti tieto-

koneen prosessoria piirtotehtävistä PGA:n suorittaessa muun muassa kolmioiden sisäisten pikseleiden värien laskemisen ja täyttämisen. PGA:n oma erillinen prosessorinsa joka tapauksessa merkitsi uutta suuntaa grafiikkapiirtimien kehityksessä. Sen erillisprosessoriarkkitehtuuri vakuutti. Vuoteen 1987 mennessä grafiikkakorttien toiminnot lisääntyivät laskentatehon myötä ja uusissa grafiikkakorteissa oli muiden muassa "Shaded Solids"-, "Vertex lightning"-, "Rasterization of filled polygons"-, "Pixel depth buffer"- ja "Color blending"-toiminnot — edelleen kuitenkin jaettiin laskentaa varsinaisen keskusyksikön prosessorin kanssa. (McClanahan 2010, s.2)

Alkukantaisin 3D-piirtäminen tapahtui siis keskusyksikössä tehtävin laskutoimittuksin rankasti sen tehoja syöden. Esimerkiksi aiemmin mainittu PGA-grafiikkakortti oli jo askel eteenpäin prosessorin vapauttamisessa. Kuitenkin senkin päätehtävän sijoittuessa CAD-piirtoon ei se lopulta vaikuttanut radikaalisti kolmiulotteisen piirron mahdollisuuksiin. Vaillinaisina ajan kortit osoittivat siltikin suunnan, johon myös tulevaisuudessa siirryttiin: Laskentaa tuli siirtää enemmän pois tietokoneen suorittimelta.

2.3 3D-korteista kiihdyttimiin

Oman panoksensa 3D-laitteiden kehitykselle antoi Silicon Graphics Inc. (SGI). Geometriasiruja kehittänyt yritys julkaisi 90-luvun vaihteen jälkeen grafiikkapiirtoon keskittyneen laitteiston ja arkkitehtuurin. Se sai nimen RealityEngine. (McClanahan 2010) Varsinainen käyttötarkoitus grafiikkalaitteistolle oli supertietokoneiden parissa. Laitteisto kuitenkin määritteli pitkälti aiemmin mainitun kiinteän liukuhihnaprosessoinnin mallin, sillä sen arkkitehtuurissa oli oman komentoyksikkönsä lisäksi sekä erilliset geometriaa laskevat että pikseleitä rasteroivat kuvamoottorinsa. (Luebke 2008) SGI tarjosi RealityEngineen liittyen sovelluskehittäjille mahdollisuuden 3D-ohjelmointiin oman IRIS-grafiikkakielen kautta. Ongelmat IRIS-kielen ylläpidossa vaikuttivat SGI:n päätökseen rakentaa sen perusteella uusi ohjelmointirajapinta (API). SGI julkaisikin OpenGL API:n tammikuussa 1992 (Crow 2004, ss. 17–19). Näin otettiin tärkeitä, grafiikkalaitteiston myöhempää kehitystä tukevia aske-

leita myös sovelluspuolella.

3dfx Voodoo-korttia pidetään yleisesti ottaen ensimmäisenä 3D-grafiikkakiihdyttimenä. 3D-kiihdyttimellä tarkoitetaan erillistä, 3D-grafiikan piirtämistä varten suunniteltua lisälaitetta. Vuonna 1996 julkaistu PCI-väylään asennettava lisäkorttien tuotesarja erosi aiemmin markkinoilta saatavista grafiikkakiihdyttimistä sen kyvystä kolmiulotteisen grafiikan piirron nopeuttamisessa. Vanhemmat kiihdyttimet tukivat pääasiassa vain 2D-kiihdytystä. Ongelmalliseksi Voodoo-sarjan teki sen kykenemättömyys kaksiulotteiseen piirtämiseen: kortti tarvitsi edelleen rinnalle tätä tukevan näytönohjaimen. Vastaavasti kilpailevien valmistajien ATI:n ja NVIDIA:n Voodoo-kortteja haastaneet 3D-kiihdyttimet olivat arkkitehtuuriltaan yksisiruisia, mutta sekä 2D- että 3D-piirtoa nopeuttavia. Yksisiruiset kortit selvisivät kilvasta voittajina. (Römisch 2009, s.5)

Johtuen grafiikkakiihdyttimien yhdestä prosessorista kykeni laite tuottamaan yhden pikselin kerrallaan jokaista prosessorin kellojaksoa kohden. Tämänkaltainen arkkitehtuuri hidasti piirtämistä huomattavasti, josta johtuen laitteiston kehitystyössä siirryttiin lisäämään prosessointiin lisää rinnakkaisuutta. Ensimmäiset grafiikkasuorittimet jättivät yhä kärkipisteiden laskennan tietokoneen prosessorille, mutta ne kykenivät jo rasterointiin ja tekstuuriin piirtämiseen. (McClanahan 2010, s.2) Huolimatta rajoituksista, toivat ensimmäiset grafiikkakiihdyttimet mukanaan muun muassa z-puskurimuistin, joka taas mahdollisti syvyysnäkökulman ja täten tarkemman aiempaa tarkemman valaistuksen piirtämisen. (Haines 2006, s.77)

Kuten edellä esitetystä havaitaan, alkuvuosina kehittyvä grafiikkalaitteisto haki siis vielä todellista suuntaansa. Olennaisina innovaatioina tältä varsinaisia grafiikkasuorittimia edeltäneeltä ajalta olivat IBM:n PGA:n grafiikkapiirtokortista löytyvä oma erillinen grafiikkapiirinsä sekä SGI:n RealityEngine. Laitekehityksen ohella tärkeitä askeleita otettiin myös sovelluspuolella SGI:n julkaistua OpenGL-ohjelmointirajapinnan. Ne näyttivät tietä tulevasta, erillisellä prosessorilla varustetusta grafiikkasuorittimesta.

3 Grafiikkasuorittimien nousu

Vielä ennen vuosituhaten vaihdetta olivat 3D-grafiikkalaitteet toiminnaltaan pitkälti riippuvaisia tietokoneen keskusyksikön prosessorista sekä rajoittuneita yhä rasteroinnin ja tekstuurien piirtämisen toteutukseen. 3D-kortit ja -kiihdyttimet toimivat siis keskusyksikköä rajallisesti avustaen. Suunta eriytymiseen oli kuitenkin jo annettu ja se sisältämistään ongelmista huolimatta tulisi vahvistumaan tulevina vuosina.

3.1 Ensimmäinen grafiikkasuoritin

Grafiikkasuorittimista yleisesti käytettävä termi Graphical Processing Unit (GPU) tuli käyttöön NVIDIA:n vuonna 1999 lanseeraaman uuden GeForce 256-näytönohjaimen myötä. Eräs merkittävistä muutoksista oli siirtyminen yleisestä PCI-laiteväylästä erityisen Accelerated Graphics Port (AGP)-väylän käyttöön. Sukupolven laitteet esittelivät uusia piirtomalleja, kuten valaistuskartat ja pintamuotojen mallinnuksen sekä siirsivät geometrista muunnostyötä ja valaistuksen laskentaa laitteiston suorittavaksi. Uuden tekniikan myötä muutos täysin kiinteästi toteutettuun liukuhihnaprosessoinnin malliin oli tapahtunut. Siirtyminen toi kuitenkin myös uusia ongelmia: Grafiikkadatan siirryttyä laitteistolle prosessoitavaksi ei sitä ollut mahdollista muokata prosessoinnin aikana. Kehittäjät olivat siis edelleen laitteiden fyysisten suoritusten rajoittamia. (McClanahan 2010, ss.2-3) Tämän ongelman ratkaisussa kuitenkin vauhditti ja avasi koko laitteiston kehitystä seuraavalle tasolle.

Kun 90-luvun loppupuolella koordinaattien transformointia ja valaistuselementtejä siirrettiin tietokoneen prosessorilta GPU:lle, pitivät osa kehittäjistä tätä askeleena taaksepäin. Erityisesti kehittäjät pelkäsivät joustavuuden menettämistä, sillä aiemmin esimerkiksi valaistuksen laskentaan käytetty malli ei riippunut käytetystä grafiikkasuorittimesta, vaan se suoritettiin keskusyksikössä. Grafiikkasuorittimet taas tarjosivat kiinteitä valaistusmalleja ja -efektejä. Näitä olivat esimerkiksi ambient, hajautus, Phong-heijastusmalli ja sumu. Lisäksi muita ohjelmointirajapinnan tar-

joamia ominaisuuksia oli mahdollista käyttää esimerkiksi OpenGL:n laajennuksina, mutta kehittäjät eivät halunneet ohjelmoida vain tiettyjä kortteja silmällä pitäen. Näihin ongelmiin vastauksena alkoi erillisten ohjelmoitavien kärkikolmio- ja sirpalesuorittimien kehittäminen vaihtoehtoisiksi poluiksi normaalin datan kanavoinnin rinnalle. (Haines 2006, ss.77–78)

3.2 Siirtyminen ohjelmoitavaan prosessointiin

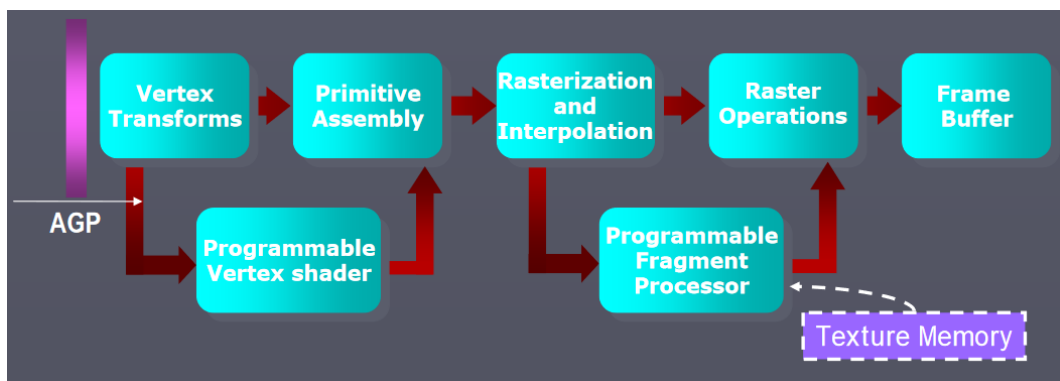
Grafiikkadatan prosessointimallin peruseriaate oli pysynyt lähes alusta asti samanlaisena aina nykypäivään saakka. Kuten luvussa 2 todettiin, data siirrettiin edelleen liukuhihnamaisesti prosessoitavaksi edelliseltä vaiheelta toiselle. Owensin ym. (2008) mukaan merkittävä muutos grafiikkalaitteiston kehityskaudessa syntyi vasta kun sovelluskehittäjät pääsivät lähemmäksi laitteistotasoa erityisesti ohjelmoinnin puolesta. Muutoksen mahdollisti siirtyminen kiinteästä liukuhihnakäsittelystä rinnakkaisen suorituksen mahdollistavaan käsittelyyn. Tämän avulla vastaavasti monimutkaisten valojen ja varjojen käyttämisen reaaliaikaisessa 3D-piirtämisessä tuli mahdolliseksi. Aiemmin laskentaoperaatiot suoritettiin vain kolmio ja sirpale kerrallaan käsiteltäen, mutta uuden prosessointimallin myötä ohjelmointirajapinnassa on mahdollista suorittaa useita operaatioita useille elementeille rinnakkain (Owens ym. 2008, ss.879–880). Ensimmäinen tätä tekniikkaa vielä rajoitetusti hyödyntävä grafiikkasuoritin vuonna 2001 julkaistusta NVIDIA:n GeForce 3-näytönohjaimesta. Vastaavasti ensimmäiset täysin ohjelmoitavissa olevat grafiikkasuorittimet olivat vuonna 2002 julkaistuissa ATI:n Radeon 9700- ja NVIDIA:n GeForce FX-näytönohjaimissa. (McClanahan 2010, s.3)

Modernien grafiikkasuorittinten dataprosessointi oli 2000-luvun ensimmäiselle vuosikymmenelle muuttunut. Nykyisessä mallissa varsinainen käytettävän sovelluksen ohjelmoija saa enemmän mahdollisuuksia toteuttaa haluamanlaistaan 3D-transformointia, valaistusta tai tekstuurialgoritmeja erillisillä piirtämiseen vaikuttavilla ohjelmilla. Näitä ohjelmia suoritetaan erillisissä shader-yksiköissä, jotka sijaitsevat grafiikkasuorittimessa. (Goodnight, Wang & Humphreys 2005, ss.12-13) Shiue, Jones & Peters (2005) määrittelevät shaderit grafiikkasuorittimien algoritmiseksi osaksi. Alkuvai-

heessa shadereita käytettiin prosessoinnissa syöttöarvojen ja sirpaleiden käsittelyssä Owens, Luebke, Govindaraju, Harris, Krüger, Lefohn & Purcell (2005). arvioivat vuonna 2005 jokaisen merkittävät grafiikkasuoritingeneraation tuovan ohjelmoitavaksi yhden uuden tason. Lopulta tämä tarkoitti koko prosessointimallia.

Shaderit muuttivat grafiikan piirtoa huomattavasti. Jo parissa vuodessa lähes kaikki liukuhihnaproessin aiemmin lukitut osat olivat tulleet ohjelmoitaviksi (Martinez-Zarzuela, Cabido-Valladolid, Sanz-Montemayor & Díaz-Pernas 2011, s.138). Lopulta shaderit ja niiden suorittaminen tulivat grafiikkasuorittimen arkkitehtuurisuunnittelun ytimeksi (Luebke 2008). Kuten Moya, Gonzalez, Roca, Fernandez & Espasa (2005) havainnoivat, keskittyi myös alan tutkimus pitkälti shader-yksikköjen toimintaan.

Edellä esitetyssä on havaittavissa, että grafiikkasuorittimien arkkitehtuurimallin tiivistyessä yhä vahvemmin varsinaiseksi kiinteäksi liukuhihnaksi, siirtyi kehittämissyö enemmän eri prosessointiosien välille. Ohjelmoitava prosessointi toi grafiikka-piirtoon uusia elementtejä, joista tärkeimpänä osoituksena ovat shaderit. Näiden merkitys korostuu myös jatkossa.



Kuvio 2. Shaderit osana datan prosessointia. *Programmable Vertex Shader* tarkoittaa ohjelmoitavaa kärkikolmioita käsittelevää shaderia, *Programmable Fragment Processor* taas ohjelmoitavaa sirpaleprosessoria ja *Texture Memory* tekstuurimuistia. Muiden termien suomennokset katso kuvan 1 kuvateksti. Lähde: <URL: <http://www.seas.upenn.edu/cis565/LECTURES/Lecture2%20New.pdf>> Viitattu 5.4.2016

4 Grafiikkalaitteiston nykyhetki

Uusien ominaisuuksien vauhdittamana koko grafiikkasuorittimen käyttö muuttui. Huolimatta samankaltaisuudesta oli grafiikkalaitteiden kehitysnopeus huomattavasti keskusyksikköjä nopeampaa. Tämä ero muodostui Owensin ym. mukaan perustavanlaatuisiin eroihin suorittimien arkkitehtuurissa. Owens ym. (2005) Keskusyksiköt suunniteltiin perättäistä koodia silmälläpitäen, kun taas grafiikkasuorittimien malli mahdollisti rinnakkaista prosessointia. Samassa ajassa samalla transistorien määrällä siis pystyttiin suorittamaan aritmeettisiä operaatioita merkittävästi enemmän.

4.1 NVIDIA lanseeraa Tesla-arkkitehtuurin

Seuraava suuri muutos alalla nähtiin NVIDIA:n GeForce 8-grafiikkasuorittinsarjan yhteydessä julkaistussa Tesla-arkkitehtuurissa. Arkkitehtuurin merkittävyys piili yhtenäistetyssä kärkipisteitä ja pikseleitä laskevassa prosessorissa. Aiemmin grafiikkalaitteistossa oli eri suorittimet molempien käsittelylle. Tämän vuoksi esimerkiksi laitteiston piirtäessä pikseleitä näyttöalueelle saattoi kärkipisteitä laskeva prosessori olla täysin toimeettomana. (Lindholm, Nickolls, Oberman & Montrym 2008, s.39)

Uudessa, yhdistetyn shader-suorittimen arkkitehtuurimallissa Lindholm & Oberman (2007) konferenssiesityksen mukaan jokainen shader kävi läpi saman ohjeituksen ja saman suorituksen samassa laitteen suoritusyksikössä. Alkuun shadereita käytettiin kiinteän liukuhihnamallin tavoin prosessoimalla data shaderilta toiselle putkimaiseen tyyliin. Vanhassa mallissa laitteiston tehot jäivät käyttämättä, mikäli yhdestä prosessiosasta tuli pullonkaula. Nyt grafiikkasuorittimen koko teho oli käytettävissä. Tesla-arkkitehtuurimallin myötä NVIDIA mahdollisti seuraavan uudistuksen — rinnakkaislaskennan — joka sai markkinoilla nimen Compute Unified Device Architecture-ohjelmointikieli (CUDA). Myöhemmin muut laitevalmistajat seurasivat trendiä muun muassa ATI:n ATI Stream- ja Microsoftin DirectX 10-ohjelmointikielien myötä. (McClanahan 2010, s.5)

4.2 Siirtyminen rinnakkaisprosessointiin

Rinnakkaislaskenta toi mukanaan merkittävän muutoksen. Itsessään ajatus rinnakkaisprosessoinnista ei kuitenkaan ollut uusi. Sitä oli hyödynnetty jo varhaisissa tietokoneissa, kun tehon lisäämiseen käytettiin toisen prosessorin lisäämistä. Rinnakkaisprosessointi pohjaa prosessorin moniytimisyyteen. Aiemmin nopeuden kehittäminen pohjasi transistorien pienentymiseen ja lisääntymiseen keskusyksikössä. Geerin mukaan tietokoneiden prosessoreiden nopeuden kehitys alkoi kuitenkin hidastumaan huomattavasti heti 2000-luvun vaihteen jälkeen. Vielä 90-luvun lopulla nopeuskehitys oli 60 % vuodessa, kun taas vuosituhannen vaihteessa se oli vain 40 % ja hidastui yhä vuoteen 2004 mennessä 20% vuosikehitykseen. Syy hänen mukaansa oli nimenomaisesti transistorikoon pienemisen rajoissa — pienentämistä ei ollut mahdollista suorittaa loputtomasti Geer (2005). Prosessorivalmistajien ratkaisu hidastumiseen oli laskevien ytimien lisääminen. Tämä ajatus siirtyi siis myös grafiikkasuorittimiin.

Perinteisessä liukuhihnaproseessoinnissa suoritin antoi siis datan käsiteltäväksi kokonaisuuksina eri prosessointitasoille. Dataa on kuitenkin mahdollista käsitellä yhtäaikaaisesti eri prosessointivaiheissa. Owens ym. (2008, s.881) määrittelevät tällaisen, yhtäaikaisen elementtien tai elementtiryhmän käsittelyn rinnakkaisprosessoinniksi. Siinä missä keskusyksikkö jakaa prosessointivaiheet ajallisesti on grafiikkasuorittimen toiminta jaettua fyysisellä tasolla. Grafiikkasuoritin siis jakaa käsittelemänsä datan suoraan prosessointivaiheeseen käytetystä yksiköstä seuraavaan. Grafiikkalaitteiden arkkitehtuuri tuki rinnakkaisprosessointia siirtymistä erityisen hyvin, koska laskentaa oli helppo tehostaa perinteiseen tyyliin yksinkertaisesti lisäämällä sitä kulloisessakin vaiheessa suoritettavien yksikköjen määrää.

Moya ym. (2005) esittävät, että grafiikkasuorittimet hyödyntävät rinnakkaisprosessointia neljällä tavalla: prosessointivaihetasolla, datatasolla, monisäikeisyydessä ja ohjeistustasolla. Prosessointitasolla tämä tarkoittaa prosessoinnin jakamista satoihin eri syklisiin vaiheeseen ja näin datan läpivienti nopeutuu ja prosessorin kellotaajuutta voidaan nostaa. Datatasolla taas prosessointitasot moninaistetaan, jotta useiden datatyyppien yhdenaikainen käsittely mahdollistuu. Monisäikeisyydessä taas

prosessointielementit talletetaan ja suoritetaan limittäisesti, jotta muistin viive eri prosessointivaiheissa voidaan piilottaa. Viimeiseksi ohjeistustasolla rinnakkaisprosessointia hyödynnetään yksinkertaisesti shader-ohjelmissa toisistaan riippumattomien suoritusohjeiden rinnakkaista suorittamista.

Ensimmäinen yhtenäistetyn prosessorin arkkitehtuuri nähtiin AMD:n Xenos grafiikkasuorittimessa. Se julkaistiin vuonna 2005 ja vastasi Xbox 360-pelikonsolin grafiikan piirtämisestä. Yhtenäistetyssä prosessorissa ei eri prosessointitasoja siis enää jaettu fyysisesti eri osiin, vaan ne suoritettiin yhdessä prosessorissa. (Owens ym. 2008, s.882) NVIDIA:n GeForce 8-sarjassa oli taas ensimmäinen grafiikkasuoritin, joka oli yhtenäistetyn arkkitehtuurin lisäksi myös täysin ohjelmoitavissa. Shadereista tuli nyt prosessorille eri aikaan suoritettavia säikeitä sen sijaan, että ne olisivat liukuhnaprosessoinnissa fyysisesti perättäisiä, erikseen suoritettavia ohjelmia. Uusi malli teki vanhasta, teknisestä liukuhinamallista suoritettavan ohjelmatasoisen abstraktion, sillä enää jokaista datan operaatioita ei suorita sitä varta vasten varattu laitteisto, vaan kaikki tapahtuu yhdessä suorittimessa useassa säikeessä. (McClanahan 2010, s.4)

4.3 Rinnakkaislaskenta - tulevaisuus nyt?

Grafiikkasuorittinten laskuteho erityisesti liukulukulaskentojen yhteydessä on niin merkittävä, että grafiikkasuorintinta hyödynnetään niin sanotussa näytönohjaimella suoritettavassa yleislaskennassa (General-Purpose Computing on Graphics Processing Units eli GPGPU). Tätä tekniikkaa hyödyntävät tänä päivänä ohjelmistorajapinnassa muun muassa OpenCL, Direct3D (DirectCompute) ja CUDA. (NVIDIA 2016a) Ensimmäisen kerran yleislaskennasta puhui Mark Harris vuonna 2002 (Martinez-Zarzuela ym. 2011, s.138). Owens ym. (2008) kirjoittivat vuonna 2008: "Rinnakkaislaskenta on tietotekniikan tulevaisuus".

Markkinoilla grafiikkalaitteiston päätehtäväksi nähtiin pelit ja niiden näyttävyyden taustalla vaadittava massiivinen laskenta. Tätä suuntaa NVIDIA:n toimitusjohtajan Huang (2009) hahmotteli Hot Chips-konferenssissä vuonna 2009 pitämässä

puheessaan laskentatehon uutena sovellutuksena yhä vaativampien maailmojen simuloiminnin. Peleissä fysiikkasimulaatio onkin jo näkynyt esimerkiksi Havok FX- ja NVIDIA:n PhysX-fysiikkamoottorien käytön yleistymisessä. Nämä pelimoottorit hyödyntävät grafiikkasuorittimien laskentatehoa erityisesti fysiikkasimuloiminnin raskaassa laskennassa. Owens ym. mukaan Havok FX-pelimoottori hyödyntää nimenomaisesti tietokoneen prosessorin ja grafiikkasuorittimen vahvuuksia laskiessaan esineiden törmäyksiä Owens ym. (2008, s.890). Pelit ovat siis jo hyödyntäneet tätä grafiikkasuorittimen merkittävää laskentatehoa.

Monikäyttöisyys ja ohjelmoitavuus ovat kuitenkin muokanneet grafiikkasuorittimen käyttösovellutuksia pois pelkästä 3D-laskentaan ja -piirtämiseen liittyvistä suoritteista. Myös tieteellisen tutkimuksen kiinnostus yleiskäyttöisiä grafiikkasuorittimia kohtaan on lisääntynyt huomattavasti. Esimerkiksi Mittal & Vetter (2014, s.19) mainitsee, että 500 parhaan supertietokoneen laskenta perustuu nimenomaisesti grafiikkasuorittimen suureen tehoon.

Viimeisinä vuosina laitevalmisjat ovat keskittyneet vieläkin enemmän prosessorin ja grafiikkasuorittimen väliseen työnjakoon. Lisäyksenä tähän jatkuvasti kehittyvään rinnakkaislaskentaan NVIDIA esitteli Kepler-arkkitehtuurin yhteydessä oman dynaamisen rinnakkaisuutensa. Tekniikassa on kysymys grafiikkasuorittimen ytimen omasta jakautumisesta uusiin suorittaviin säikeisiin. Enää ei siis tarvita varsinaisen prosessorin käskytystä uusille säikeille, vaan tämäkin työ siirretään grafiikkasuorittimen sisäiseksi tehtäväksi. (NVIDIA 2016b) Vuonna 2016 esitellyissä grafiikkasuorittimien arkkitehtuurimalleissa on jo hahmoteltu useita varsinaiseen grafiikkapiirtämiseen liittymättömiä ominaisuuksia. Näistä esimerkiksi NVIDIA:n Pascal-mallissa merkittävimpinä näyttäytyvät tekoälyyn liittyvän laskennan optimointi sekä neuroverkkoihin liittyvän koneellisen oppimisen huomattavaa nopeuttamista. (NVIDIA 2016c)

Laskentatehon kasvaessa ovat grafiikkasuorittimet nousseet tärkeään rooliin suhteessa tavalliseen prosessoriin. Tämä on nostanut vaatimustasoa ohjelmoijille, joiden tulee entistä enemmän hallita rinnakkaisuuksia ja laitteiden yhteislaskutehon hyödyntämistä.

5 Yhteenveto

Koko grafiikkalaitteiston alkuperäinen lähtökohta oli siis tukea tietokoneen varsinaista keskusyksikköä piirtämiseen liittyvissä raskaissa laskentatehtävissä. Aiemmin grafiikkapiirtoa suoritettiin tietokoneeseen sijoitetuilla geometriasiruilla. Hiljalleen laitteet erottautuivat kuitenkin erityiseksi, omaksi arkkitehtuurikseen. Tässä suhteessa muutos laitearkkitehtuurissa tapahtui, kun valmistajat kehittivät grafiikkakortteja, jotka kykenivät suorittamaan esimerkiksi pikselien piirtämisen ruudulle. Menetelmä helpotti tietokoneen prosessorin kuormittamista ja samalla lisälaitteina saatavat kortit synnyttivät käsitteen grafiikkalaitteista. 3D-kiihdyttimiksi nimetyt laitteet kykenivät vauhdittamaan piirtämistehtäviä riippumatta käytetystä prosessorista.

Kuten esimerkiksi Intel i860-prosessorin arvioinnin yhteydessä esitetyistä havainnoista on havaittavissa, jo ennen varsinaisia grafiikkasuorittimia suhtauduttiin alalla negatiivisesti kiinteään ja vaikeasti käskyttävään arkkitehtuuriin. Vapauttamalla laitteiston toimintaa hiljalleen ohjelmointirajapinnan mukautettavaksi, mahdollisti se laskentatehon yhä paremman hyödyntämisen ja täten grafiikan monimutkaistamisen. Samalla se antoi näkemyksen tulevasta kehityksestä Shader-mallina.

Seuraavaksi olennainen kehityslinja laitteiden kehityksessä erottui ensimmäisen varsinaisen grafiikkasuorittimen myötä. Grafiikkapiirtäminen siirtyi erillisen suorittimen tehtäväksi ja niin sanotun liukuhihnaprocessoinnin malli tuli grafiikkalaitteiden vastuulle. Näin koko piirtämisen prosessointi tapahtui grafiikkasuorittimessa — prosessorin merkitys varsinaiselle piirtämiselle oli nyt minimoitu.

Grafiikkalaitteiden tekniikassa otettiin 2000-luvun ensimmäisellä vuosikymmenellä merkittäviä askeleita. Oman erillisen arkkitehtuurin mahdollistamana laitteista pystyttiin nyt myös luomaan enemmän hallittavia ja ohjelmitavia. Käytännössä muutos tapahtui niin sanottujen shadereiden käyttöönotossa. Shader-yksiköt toimivat perinteisen arkkitehtuurimallin lomassa toteuttaen omissa yksiköissään prosessoitavalle datalle ohjelmitavat toimenpiteet. Muutoksen ohessa eri sovelluskehittä-

jät toivat markkinoille ohjelmointirajapintoja kuten Direct3D:n ja OpenGL:n. Lopulta datan prosessointi yhdistettiin kaikki tasot yhdessä prosessorissa suoritettavaan malliin. Niin sanottu Unified Shader Model teki tien täysin uudelle grafiikkasuorituskuvapolvelle. Shaderien ja yhdistetyn prosessorin mallin myötä grafiikkasuorittimien alkuperäinen liukuhihnaprososoinnin malli oli muuttunut kiinteästä fyysisestä muodosta ohjelmatason abstraktioksi. Vaikka perinteinen malli edelleen ohjasi prosessointia, tapahtui toiminta yhdessä ytimessä lähellä ohjelmointirajapintaa. Näin ollen myös suuntaviivat tulevalle oli jo annettu.

Terminä rinnakkaislaskenta ei ollut uusi, mutta se nousi käytännön syistä merkittävään rooliin. Rinnakkaislaskennan merkitys kasvoi, kun prosessorien valmistajat eivät enää kyenneet lisäämään laskutehoa vain transistorimäärän kasvattamisella. Johtuen grafiikkasuorittimien arkkitehtuuri- ja datankäsittelymallista soveltuivat ne hyvin uudelle rinnakkaislaskennan suuntaukselle. Suoritinvalmistajat osoittivat tämän lisäämällä grafiikkasuorittimien laskutehoa yhä kasvavalla tahdilla lopulta ohittaen varsinaisten keskusyksikköjen laskutehon nousutahdin.

Rinnakkaislaskenta on lisännyt ohjelmoijan vaatimustasoa. Olisi helppoa kuvitella, että aiemmin raskaiksi osoittautuneet ja laskentatehoa vaatineet grafiikkapiirtämiset mahdollistaisivat yhä helpomman toteuttamisen. Kuitenkin teho on kasvanut yksinkertaisuuden kustannuksella. Nykypäivän prosessorit ja grafiikkasuorittimet ovat moniytimisiä, useita säikeitä sisältäviä yksikköjä, joiden yhtäaikainen hallinta vaatii hyvää ymmärrystä rinnakkaislaskennan peruseräiteistä. Tätä onnistuneesti hyödyntämällä on modernista grafiikkasuorittimesta tullut myös monia muita laskutoimituksia vauhdittava, joissakin tapauksissa jopa tietokoneen prosessoria korvaava laite.

Lopulta onkin mielenkiintoista, että lähtökohtana ollut tietokoneen prosessorin vapauttaminen on kääntynyt sen ajoittaiseen, lähes täydelliseen korvaamiseen. Grafiikkalaitteiden käyttötarkoitus on vuosien aikana muuttunut alkuperäisestä merkittävästi kohti yhä yleiskäyttöisempää, erillistä suoritinta kohti. Olisiko jatkossa syytä puhua grafiikkasuorittimen sijaan laskentasuorittimesta?

Tulevaisuus näyttäisi siltä, että rinnakkaissuorittamisen ja -laskemisen ongelmakohdaksi ohjelmointirajapinnassa nähty monimutkaisuus siirtyisi yhä enemmän laitevalmistajien vastuulle. Tästä osoituksena on NVIDIA:n uusin grafiikkasuoritinarkkitehtuuri, jossa säikeisiin jakaminen tapahtuu dynaamisesti suorittimessa itsessään, eikä keskusyksikön käskytyksen mukaisesti. Tässä suhteessa suunta näyttää olevan kohti aiempaa hallitumpaa prosessointimallia — kuitenkin shaderien ohjelmoitavuutta unohtamatta.

Tehtyjen havaintojen perusteella olisi mielenkiintoista tutkia jatkossa esimerkiksi jonkin yksittäisen ominaisuuden, kuten yhdistetyn prosessorimallin, kehitykseen johtanutta tutkimusta. Toisaalta jatkotutkimusta olisi hedelmällistä tehdä piirtoon ja laskentaan käytettyjen algoritmisten menetelmien puolelta. Tämän opinäytetyön tyyppinen historiakatsaus menetelmien kehityksestä laajentaisi nyt avattua näkökulmaa suorittinten kehitykselle — mitkä menetelmälliset ratkaisut avustivat teknologiaa?

Kirjallisuutta

- Crow T. 2004. *Evolution of the Graphical Processing Unit*. Pro Gradu. Nevadan yliopisto, USA.
- Geer D. 2005. *Chip Makers Turn to Multicore Processors*. IEEE Computer, Vol. 38, Iss. 5, s. 11–13.
- Goodnight N., Wang R. & Humphreys G. 2005. *Computation on Programmable Graphics Hardware*. IEEE Computer Graphics and Applications, Vol. 25, Iss. 5, ss. 12–15.
- Grimes J. D., Kohn L. & Bharadhwaj R. 1989. *The Intel i860 64-bit processor: a general-purpose CPU with 3D graphics capabilities*. IEEE Computer Graphics and Applications, Vol. 9, Iss. 4, s. 85–94.
- Haines, E. 2006. *An Introductory Tour of Interactive Rendering*. IEEE Computer Graphics and Applications, Vol. 26, Iss. 1, s. 76–87.
- Huang J. 2009. *The GPU Computing: Tipping Point*. Konferenssiesitys. Hot Chips 2009.
- Lindholm E. & Oberman S. 2007. *The NVIDIA GeForce 8800 GPU*. Konferenssiesitys. Hot Chips 2007.
- Lindholm E., Nickolls J., Oberman S. & Montrym J. 2008. *NVIDIA Tesla: A Unified Graphics and Computing Architecture*. IEEE Micro, Vol. 28, Iss. 2, ss. 39–55.
- Luebke D. 2008. *GPU Architecture: Implications & Trends*. Konferenssiesitys. SIGGRAPH 2008.
- Martinez-Zarzuela, M., Cabido-Valladolid, R., Sanz-Montemayor, A. & Díaz-Pernas, F.J. 2011. *A Survey on the GPU Computing Revolution*. Teoksessa M.J. Davis (toim.) Computer Graphics. Hauppauge, US: Nova, s. 137–162.
- McClanahan C. 2010. *History and Evolution of GPU Architecture*. Survey Paper. Georgia Tech, College of Computing, USA.
- Mittal S. & Vetter J.S. 2014. *A Survey of Methods For Analyzing and Improving GPU Energy Efficiency*. ACM Computing Surveys, Vol. 47, No. 2, Article 19.
- Moya V., Gonzalez C., Roca J., Fernandez A. & Espasa R. 2005. *Shader Performance Analysis on a Modern GPU Architecture*. Konferenssijulkaisu. MICRO-38 - 38th Annual IEEE/ACM International Symposium on Microarchitecture.
- NVIDIA 2016a. *What Is GPU Computing?*. <URL:

<http://www.nvidia.com/object/what-is-gpu-computing.html>>.

Viitattu 4.4.2016.

NVIDIA 2016b. *Whitepaper. NVIDIA's Next Generation CUDA Compute Architecture: Kepler GK110/210.* <URL: <http://images.nvidia.com/content/pdf/tesla/NVIDIA-Kepler-GK110-GK210-Architecture-Whitepaper.pdf>>. Viitattu 6.4.2016.

NVIDIA 2016c. *Introducing the New NVIDIA Pascal Architecture.* <URL: http://www.nvidia.com/object/gpu-architecture.html#utm_source=shorturl&utm_medium=referrer&utm_campaign=pascal>. Viitattu 6.4.2016.

Owens, J.D. Luebke, N. Govindaraju, M. Harris, J. Krüger, A.E. Lefohn, T.J. Purcell 2005. *A Survey of General-Purpose Computation on Graphics Hardware.* Eurographics 2005, State of the Art Reports, s. 21–51.

Owens, J.D., Houston, M., Luebke, D., Green, S., Stone, J.E. & Phillips, J. 2008. *GPU Computing.* Proceedings of the IEEE, Vol. 96, No. 5, s. 879–899.

Römisch K. 2009. *Sparse Voxel Octree Ray Tracing on the GPU.* Pro gradu-opinnäytetyö. Aarhusin yliopisto, tietotekniikan laitos.

Shiue L., Jones I. & Peters I. 2005. *A Realtime GPU Subdivision Kernel.* ACM Transactions on Graphics, Vol. 24, Iss. 3, s. 1010–1015.

Wang Y., Mangaser A. & Srinivasan P. 1992. *A Processor Architecture for 3D Graphics.* IEEE Computer Graphics and Applications, Vol. 12, Iss. 5, s. 96–105.