

Jyrypekka Lehtinen

3D-animaation yhdistäminen live-kuvaan

3D-objektien ja live-kuvan dynaamisen valotilanteen vuorovaikutus

Metropolia Ammattikorkeakoulu

Medianomi

Viestinnän koulutusohjelma

Opinnäytetyö

12.5.2017

Tekijä(t) Otsikko	Jyrypekka Lehtinen 3D-animaation yhdistäminen live-kuvaan
Sivumäärä Aika	31 sivua + 1 liite 12.5.2017
Tutkinto	Medianomi
Koulutusohjelma	Viestinnän koulutusohjelma
Suuntautumisvaihtoehto	Graafinen suunnittelu
Ohjaaja(t)	Lehtori Lauri Huikuri
<p>Tämän opinnäytetyö tutkielman tavoitteena oli ensisijaisesti selvittää, mitä tulisi ottaa huomioon kuvakerronnan kohtauksen rakentamis- ja toteutusvaiheessa, kun tarkoitus on käyttää 3D-animaatiota upotettuna live-kuvaan. Yritin erityisesti rakenteellisesti avata dynaamisen interaktion rakentamista 3D-materiaalin ja oikean maailman valotilanteen välille. Sekä korostaa mahdollisuuksia käyttää 3D-kamera trakkäystä ja moderneja 3D-grafiikka työkaluja erilaisissa toteutuksissa, joissa muut tavat voisivat olla liian työläitä.</p> <p>Kävin aiheen teoriaa läpi osissa, jotka ovat karkeasti jaoteltu kohtauksen rakentamisen päävaiheisiin. Aloitin kuvausten valmistelusta ja päätin 3D-materiaalin ja live-kuvan lopulliseen yhdistämiseen. Yritin käsitellä teoriaosuutta yleisellä tasolla, joka ei ole sidottu mihinkään erityiseen ohjelmaan. Taustatyö perustuu laajalti omaan kokemukseeni aiheesta, mutta tukeutuu myös tunnettuun alan teokseen. Huomioitani tutkielman ohessa toteutetusta omasta käytännön työstäni kuljetan vaiheittain mukana lukujen lopussa. Oman projektini huomiot koskevat erityisesti sen toteuttamiseen käytettyä ohjelmistoa.</p> <p>Oma projektini on toteutettu Blender 3D -ohjelmistolla ja sen monipuolisilla moduuleilla. Blender 3D on avoimen lähdekoodin ohjelma, jota ei perinteisesti ole aiemmin käytetty visuaalisten efektien tuottamiseen elokuvissa, johtuen sen puutteellisista ominaisuuksista verrattuna muihin alan johtaviin ohjelmistoihin. Mahdollisuus kuitenkin toteuttaa projekti Blender 3D -ohjelmalla korostaa erilaisia tapoja päästä samaan lopputulokseen.</p> <p>Kirjoitin tämän tutkielman pääasiallisesti oman tietämykseni syventämiseksi 3D-animaation käyttämisestä visuaalisena efektinä elokuvassa. Sen lisäksi se voi avata aihetta muille, joilla on kiinnostusta, mutta ei niin paljon kokemusta kyseisestä toteutustavasta.</p>	
Avainsanat	3D-animaatio, Visuaaliset efektit, Blender 3D, 3D-trakkäys

Author(s) Title	Jyrypekka Lehtinen Combining 3D Animation with Film Footage
Number of Pages Date	31 Pages + 1 Appendix 12 May 2017
Degree	Bachelor of Culture and Arts
Degree Programme	Media
Specialisation option	Graphic Design
Instructor(s)	Lauri Huikuri, Senior Lecturer
<p>This thesis examines the requirements in combining 3D animation with film footage, to achieve a convincing illusion of unity between computer generated imagery and film footage of the real world. The main focus is to elaborate on the structural theory of recreating the real world perspective and lighting inside the virtual environment and highlighting the possibilities of using 3D camera tracking and modern 3D animation tools to create a visual effects shot, that might by other means be too work intensive to achieve.</p> <p>The theory of the subject is approached in segments divided roughly into the main stages of the work process in building the VFX shot. The thesis starts from shooting the footage and ends up in compositing the final product. It tries to convey the process in a way that is not committed to any specific software. Research relies mainly on the writer's primary knowledge and is backed up by a staple publication in the field. Observations on producing the accompanying demonstration project is carried alongside each stage, and mostly pertain to the specific software used.</p> <p>The demonstration project was produced with the Blender 3D software and its versatile modules. Blender 3D is an open source software that has conventionally not been used by VFX artists in the past, because of the lack of features in comparison to other leading programs in the industry. Achieving the desired result with Blender 3D further highlights the different paths possible to achieve the same end result.</p> <p>The thesis was mainly written to deepen the writer's own understanding of creating a 3D animation VFX shot for film, and possibly help others with interest but less experience to discover the potential in the methods used.</p>	
Keywords	3D animation, Visual Effects, Blender 3D, 3D Camera Tracking

Sisällys

1	Johdanto	1
2	Kuvausten valmistelu	2
2.1	Esineet ja pinnat	4
2.2	Valaistus	6
2.3	Träkkäyspisteet	8
2.4	Tekniikan rajoitteet	10
3	3D-träkkäys	11
3.1	2D-träkkäyspisteet ja 3D-merkit	13
3.2	Live-kuvan puhdistus	16
4	3D-ympäristön valmistelu	18
4.1	Valaistuksen määrittely	20
4.2	Interaktiivinen sijaisgeometria	22
4.3	Varjo- ja heijastuspinnat	24
5	Kompositointi	25
5.1	Kompositointikerrosten värisäätö ja suodatus	27
5.2	Efektit	27
6	Yhteenveto	29
	Lähteet	30

Liitteet

Liite 1. Produkti: *Oma projekti, Jyrypekka Lehtinen (videotiedosto)*

1 Johdanto

Tutkielman tavoitteena oli selvittää, mitä tulee ottaa huomioon kuvakerronnan kohtauksen rakentamis- ja toteutusvaiheessa, kun tarkoituksena on käyttää 3D-animaatiota upotettuna live-kuvaan. Yritin nimenomaan keskittyä 3D-materiaalin dynaamiseen interaktioon oikean maailman pintojen, objektien ja valotilanteen kanssa.

Tuotin tutkielman yhteydessä havainnollistavan teoksen käyttämällä itse kuvaamaani video-sekvenssiä, Blender 3D -ohjelman 3D Motion Tracking -sovellusta, 3D-mallinnuskomponenttia, Cycles Render Engine -piirtomodulia ja kompositointimoduulia sekä Adobe AfterEffects -ohjelmaa kuvasarjan yhdistämiseen.

Olen saanut käytännön kokemusta tutkielman eri osa-alueista aikaisempien projektien toteutuksessa, mutta tämä oli ensimmäinen kerta, kun kaikki nämä osa-alueet yhdistyvät tällä tavalla.

Avasin eri osa-alueita ja syvennyin niihin siinä määrin, kun näin sen olevan rakenteellisesti merkittävää dynaamisen vuorovaikutuksen rakentamisen kannalta. En kuitenkaan kuvaillut syvällisesti lavastuksen tai valaisun estetiikkaa, 3D-objektien mallintamista tai animointia.

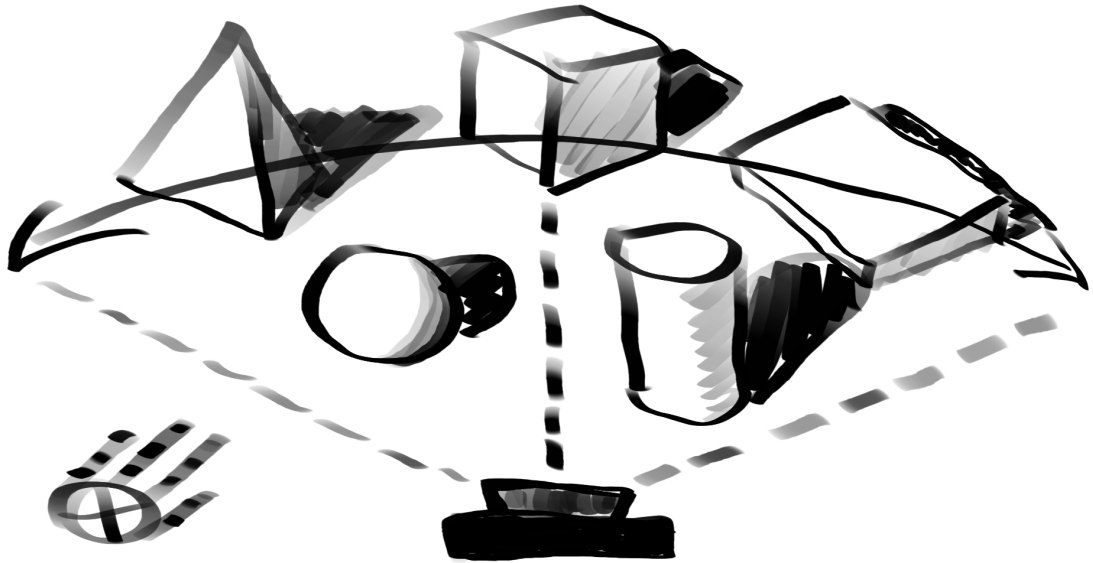
Käytännön hyöty tutkielmasta oli ensisijaisesti minulle itselleni, koska se oli luonnollinen jatke ja puuttuva palanen kiinnostuksestani CG-materiaalin (CG eli Computer Generated) yhdistämisestä live-kuvaan. CG-elementeillä viitataan tietokoneella tuotettuun grafiikkaan, tämän tutkielman yhteydessä se on nimenomaan kolmiulotteista grafiikkaa, joka on tuotettu 3D-grafiikkaohjelmalla. Kutsuin näitä elementtejä myös 3D-objekteiksi tai virtuaaliobjekteiksi. Oikealla kameralla kuvattavaa kuvamateriaalia kutsun live-kuvaksi. Toivoisin että tämä tutkielma myös hieman valaisisi kyseisen tekniikan mahdollisuuksia graafisen alan osaajille, jotka ovat ehkä tutustuneet 3D-mallinnukseen ja kameran liikkeen jäljittämiseen, mutta eivät välttämättä ole vielä tulleet syystä tai toisesta kosketuksiin syvempien 3D-ratkaisujen käyttämisessä graafisten pulmien ratkaisuun.

2 Kuvausten valmistelu

Päätös CG-elementtien yhdistämisestä kuvaan olisi hyvä olla selvillä jo kuvausten lavastusvaiheessa. Kun päätös CG-elementtien käytöstä on tehty, se asettaa tiettyjä rajoitteita ja ennakkovaatimuksia kuvauksessa käytettäville lavasteille. Jos työn lopputuleman on tarkoitus olla yksittäinen kuva, ovat rajoitteet ja huomioitavat asiat vähemmän merkittäviä lavastuksessa, koska niiden korjaaminen manuaalisesti myöhemmissä vaiheissa vie suhteellisen vähän aikaa. Tässä tutkielmassa kuitenkin keskityn nimenomaan siihen, mitä tulee ottaa huomioon, kun lopputulema on liikkuva kuva. Rajoitteita on enemmän ja niiden huomiointi lavastusvaiheessa voi ratkaisevasti karsia aikaa myöhemmistä työvaiheista.

Tarve CG-elementeille käy yleensä ilmi kuvakäsikirjoituksen tarkastelussa, jossa jotkin elementit ovat liian kalliita tai epäkäytännöllisiä toteuttaa perinteisillä kuvaus- ja leikkausmenetelmillä. Kuvakäsikirjoitusta tarkasteltaessa jaetaan kuvassa esiintyvät elementit perinteisillä elokuvan lavastusmenetelmillä toteutettaviin ja tietokonegrafiikalla toteutettaviin elementteihin. Seuraavat alaluvut syventyvät kuvausten valmistelussa huomioitaviin asioihin ja rajoitteisiin, kun tavoitteena on luoda dynaaminen vuorovaikutus CG-elementtien ja oikean maailman lavaste-elementtien välille.

Opinnäytetyöni rinnalle toteuttamani havainnollistava projekti alkoi videokuvan tallentamisella. Tähän ei ollut mitään erityisen syvällistä suunnitelmaa, vaan tarkoitus oli yksinkertaisesti tallentaa liikkuvaa kuvaa johon 3D-objektit tai -animaatio voidaan upottaa. Kuviossa 1 on yksinkertainen visualisointi kohtauksen suunnitelmasta. Poikittain kulkeva nuoli esittää kameran liikettä ja vasemman alareunan symboli esittää valon sijaintia.



Kuvio 1. Kuvakäsikirjoitus (illustraatio).

Tehdäkseni kuvan käyttämisestä 3D-grafiikan kanssa helpompaa, päädyin valitsemaan geometrisesti yksinkertaisia esineitä alueelle, jossa 3D-objektit tulisivat vaikuttamaan ympäristöön, jättäen monimutkaisemmat esineet kauemmaksi. Valaistuksessa käytin yhtä voimakasta valoa saadakseni aikaan yksinkertaisemmat varjot, jotka olisivat helpompi mallintaa 3D-grafiikkaohjelmassa.

Kuviossa 2 video-sekvenssin kuvauksen jälkeen tehty tarkempi suunnitelma lopullisessa kuvassa esiintyvistä esineistä. Tässä vaiheessa minulla oli selvillä minkälaisen hahmon halusin tarinassa esiintyvän.



Kuvio 2. Oikeat ja virtuaaliset (illustraatio).

Kuvassa suunnitelma oikeiden ja virtuaalisten elementtien erottelusta. Vihreällä on merkitty vain virtuaaliset objektit, sinisellä oikean maailman esineet joiden valotilanteeseen virtuaaliobjektit vaikuttavat ja punaisella oikean maailman esineet joihin virtuaaliset objektit eivät vaikuta.

2.1 Esineet ja pinnat

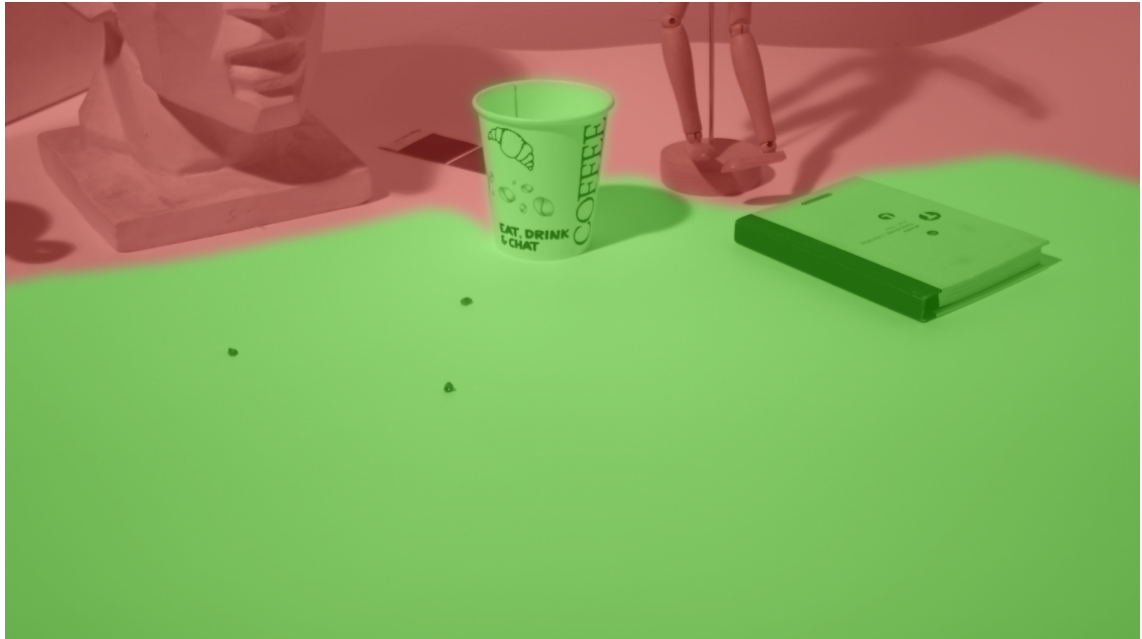
Tässä luvussa käsittelen esineitä ja pintoja, jotka ovat vuorovaikutuksessa CG-elementtien kanssa. Tässä tutkielmassa CG-elementti sijoittuu oikean maailman lavasteen sisälle, jota kutsun jatkossa live-ympäristöksi tai live-lavasteeksi. Live-ympäristö pitää sisällään kaikki perinteisillä lavastusmenetelmillä toteutettavat elementit, jonka keskiössä ovat esineet ja pinnat, joiden on tarkoitus olla suorassa vuorovaikutuksessa virtuaaliobjektien kanssa. Kutsun näitä keskeisiä pintoja ja esineitä vastaisuudessa kollektiivisesti tapahtumapaikaksi, mikä erottaa ne merkitykseltään muista lavastuselementeistä, joilla ei ole interaktiivista roolia kuvakerronnassa.

Kuvakäsikirjoituksesta voi ilmetä kuvakerronnan kannalta olennaisia tavoitteita lavaste-elementeille. Lavaste-elementin konkreettiseksi tavoitteeksi voi osoittautua esimerkiksi monimutkainen geometrinen muoto tai pinta, jolloin pitäisi harkita, onko se tarpeellista sijoittaa juuri vuorovaikutuksessa olevalle tapahtumapaikalle vai voitaisiinko se sijoittaa live-ympäristön taka-alalle.

Suorassa vuorovaikutuksessa olevat lavaste-elementit joudutaan myöhemmin uudelleenluomaan niin kutsuttuna sijaisgeometriana, joka on 3D-grafiikkaohjelman sisällä luotavan virtuaalilavasteen osa. Käsittelen sijaisgeometriaa syvemmin luvussa 4.2 Interaktiivinen sijaisgeometria. Käytännön nyrkkisääntönä pidän itse sitä, että mitä monimutkaisempi geometria, sitä enemmän sen virtuaalisen vastikkeen uudelleenluomisessa kuluu aikaa. Tähän kaksi suurinta vaikuttajaa ovat itse geometrian muotoilu sekä geometrinen muotojen sijainnin täsmäminen.

Yksi olennainen tavoite voisi olla myös CG-elementtien heijastuminen esimerkiksi kiiltävästä pinnasta, mutta on hyvä huomioida, että jokainen ylimääräinen samanaikainen näkökulma osittain virtuaaliseen asetelmaan voi potentiaalisesti viedä moninkertaisesti aikaa toteuttaa. CG-elementin kirkkaasti ympäristöä heijastava pinta voi luoda myös toisenlaisen haasteen ja tarpeen 3D-grafiikkasohjelman kuvapohjaiselle valaistukselle (*Image-based lighting* (IBL)), joka on tekniikka, jossa käytetään valokuvia valaisemaan virtuaaliobjekteja (Wikipedia 2015a). Kuvapohjainen valaistus voi muodostua ainoaksi ratkaisuksi, joka saa CG-elementit istumaan live-ympäristöön. Käyn tätä tekniikkaa tarkemmin läpi luvussa 4.1 Valaistuksen määrittely.

Oman projektini kuvassa pahvinen kahvikuppi ja muistilehtiö on aseteltu tapahtumapaikan taakse oikealle. Tämä on suunta johon kameran vasemmalta puolelta tuleva valo heittäisi kuva-alan keskellä olevien esineiden varjot, jos niitä olisi. Valoista lisää seuraavassa luvussa 2.2 Valaistus. Tapahtumapaikan keskelle asettelin lattialta löytämiäni pikkukiviä, joiden tarkoitus on havainnollistaa pöydän pinnan kulmaa suhteessa kameraan sekä auttaa 3D Motion Tracking -sovellusta. Siitä lisää luvussa 2.3 Träkkäyspisteet sekä luvussa 3. 3D-träkkäys. Pöydän pinnalle valitsin materiaalia, joka ei olisi kiiltävä. Aivan mattapintaista materiaalia ei ollut käytettävissä, joten tyydyin muovirullaan jossa oli röpelöinen pintakuvio. Tämä helpottaa virtuaalielementtien heijastuksien kanssa. Liian kiiltävä tai peilaava pinta aiheuttaisi lisää mallinnusta jotta 3D-objektien heijastukset vastaisivat tarkasti oikeita esineitä. Kuviossa 3 video-sekvenssin yksi kuva johon on eroteltu tapahtumapaikka ja tausta.



Kuvio 3. Tapahtumapaikka (väritetty kuvakaappaus).

Kuvakaappaus videon tapahtumapaikasta johon olen värittänyt pinnat joihin virtuaalieleментit voivat vaikuttaa. Ne on merkitty vihreällä. Punaisella merkittyjen pintojen tulisi olla vain live-kuvassa, eivätkä ole suorassa vuorovaikutuksessa virtuaalieleментtien kanssa.

2.2 Valaistus

Kuvauksessa käytettävien lavasteiden valaisulla on monia vaikutuksia CG-elementtien käyttöön. Pääpiirteittäin pitäisin tärkeimpinä asioina käytettävien valojen määrää, muotoa, sijaintia ja valaisusuuntaa. Koska tarkoituksena on luoda mahdollisimman tarkka kopio live-kuvan lavasteiden valotilanteesta myöhemmin rakennettavaan 3D-ympäristöön, on valaisuasetelman dokumentointi myös tärkeää. Vaikka lopuksi kaikki on varmennettava visuaalisesti, mitä enemmän ja mitä tarkempaa dataa on virtuaaliympäristön rakentamisvaiheessa käytettävissä, sitä nopeammin päästään hienosäätövaiheeseen.

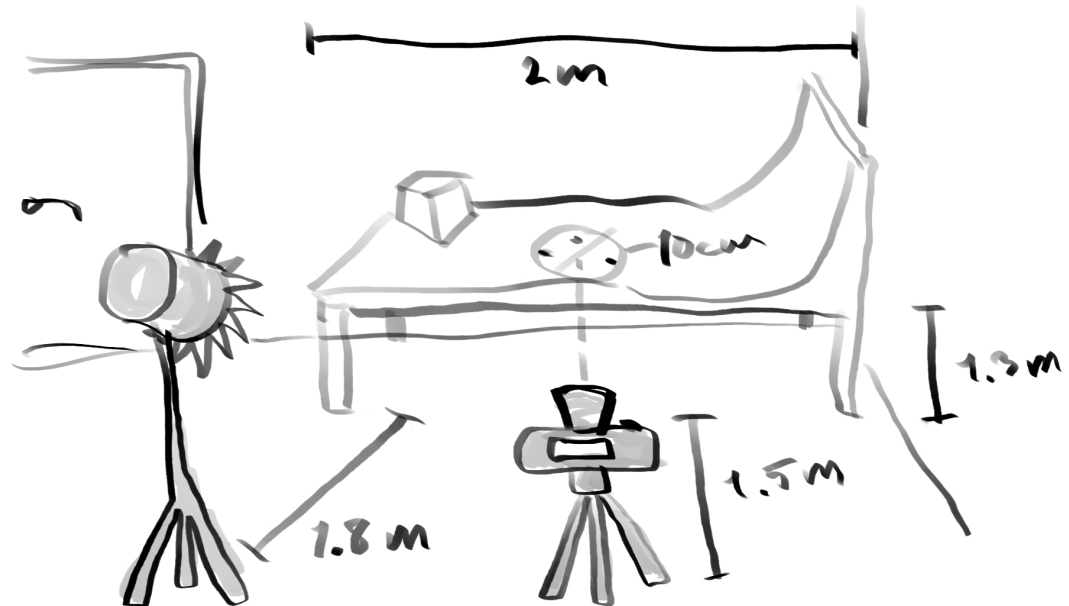
Valojen määrällä ei varsinaisesti ole rajoitteita, koska 3D-ympäristöön voidaan asettaa aivan yhtä monta tai useampi valo kuin live-ympäristössä. Valojen määrä kuitenkin lisää 3D-ympäristön rakentamiseen käytettävää aikaa sekä 3D-grafiikkaohjelman valosimulaation laskennallista aikaa. Laskennallinen aika lisääntyy suuremmilla kuvatarkkuuksilla ja voi aiheuttaa pitemmän videokuvan jälkituotannossa ongelman ajan tai rahan kanssa, riippuen käytettävistä resursseista. Jos heikkojen valolähteiden lukumäärä kasvaa kovin suureksi, voi olla tarpeellista harkita kuvapohjaista ympäristövalaistusta.

Käytettävien valojen muoto vaikuttaa virtuaalivastakappaleiden toteutukseen. 3D-grafiikkaohjelmien perusasetuksissa on monesti kyseisellä ohjelmalla hyvin toimivia valotyyppejä, vaikka valosimulaatioasetukset olisivat yksinkertaisemmat. Virtuaaliympäristöön voi myös rakentaa valon esineenä, ja myös pehmeiden täytevalojen rakentaminen onnistuu. Käyttäjän omasta osaamisesta 3D-grafiikkaohjelman valaisun rakentamisessa riippuu, kuinka merkittäväksi asiaksi tämä muodostuu.

Valojen sijainti erityisesti olisi hyvä dokumentoida. Sijainnin merkkäamisessa voi käyttää eri tapoja, esimerkiksi live-lavasteista voi valita koordinaatiston nollapisteeksi jonkun selkeästi näkyvän kohdan. Valojen sijainnista olisi hyvä pistää muistiin jalustan etäisyys nollapisteestä ja korkeus lattiasta.

Valaisusuunnan erityinen muistiinpano ei ehkä ole kuitenkaan tehokasta ajankäyttöä. Sen sijaan valaisusuunta on erittäin tärkeä huomioida varsinkin valoille, jotka heittävät lavasteiden esineistä teräviä varjoja. Koska kuvaan upotettavien virtuaaliobjektien on tarkoitus heittää varjoja live-kuvan lavasteiden pinnoille, pitää varmistua, että kaikki pinnat, joihin 3D-objektin varjon olisi tarkoitus osua, ovat projektin käytössä olevien resurssien puitteissa teknisesti toteutettavissa. Valojen suuntauksen voi helposti varmistaa lavasteiden valmisteluvaiheessa heiluttamalla jotain esinettä tai ihmistä samassa kohdassa, jossa virtuaalielementin olisi tarkoitus liikkua. Näin saadaan yleinen käsitys siitä, mitä sijaisgeometriaa virtuaaliympäristöön pitäisi rakentaa dynaamista valotilannetta varten.

Live-kuvassa käytin lavasteen valaisemiseen yksittäistä voimakkaampaa studiovalaisinta. Ympäristöstä kuitenkin paljon tulee pehmentävää yleisvaloa koska lavasteiden ympärillä on paljon vaaleita pintoja, jotka heijastavat päävalaisimen tuottamaa valoa edestakaisin. Muistiinpanona jälkepäin tehtävää virtuaaliversiota varten painoin muistiin kameran, valon ja ympäröivien isompien pintojen suhteelliset sijainnit. Tämä antaa lähtökohdan tarvittaville virtuaalimaailman valolähteille. Kuviossa 4 havainnollistettu kuvauksessa käytettyä tilaa.



Kuvio 4. Muistiinpanot valaistuksesta (illustraatio).

Kuvassa hieman karkeita arvioita lavasteista ja valaisusta. Muistiinpanojen tarkoitus on lähinnä auttaa jälkituotannon vaiheissa, joissa voi ilmetä tarve mallintaa kameran kuvan ulkopuolelle jäävien osien vaikutuksia virtuaaliobjekteihin.

2.3 Träkkäyspisteet

Träkkäyspisteitä kannattaa huomioida myös kuvausten valmisteluvaiheessa. Ne ovat kuvan liikkeen selvittämiseen käytettäviä seurantapisteitä, joita Motion Tracking tai 3D Camera Tracking -ohjelmat käyttävät (Dobbert 2012, 40). Niitä käytetään niin kaksiuotteiseen kuvan elementtien seuraamiseen kuin myös tämän tutkielman osalta kolmiuotteiseen kameran tai esineen liikkeen selvittämiseen, jota kutsun lyhyesti 3D-träkkäykseksi. Tämä on elokuvan jälkituotannon vaihe, jota englannin kielellä kutsutaan nimellä *Matchmoving*.

Yleisesti parhaiten videokuvasta poimittavista kiinnepisteistä, joita 3D-träkkäysohjelma seuraa, toimivat kuvassa olevat suuren kontrastin pisteet ja esineiden terävät kulmat. Selkeitä geometrisiä kuvioita tasaisilla pinnoilla voi myös käyttää tehokkaasti. Monissa sovelluksissa on erityisesti tasaisilla pinnoilla olevien kuvioiden jäljittämiseen tarkoitettu trækkerityyppi, joka pystyy tunnistamaan perspektiivimuutoksen kuvion muodossa ja pystyy jäljittämään kuvioita, vaikka ne kameran liikkeen takia vääristyvät. Näiden pisteiden on oltava täysin staattisia ympäristöön nähden, jotta ohjelma voi käyttää niitä kameran liikkeen selvittämiseen. Käsittelen 3D-träkkäystä tarkemmin luvussa 3. 3D-träkkäys, sekä trökkäyspisteitä ja niiden käytännön merkitystä luvussa 3.1 2D-trökkäyspisteet ja 3D-merkit. Suuren kontrastin omaavia pisteitä voi sijoittaa live-ympäristöön keinotekoisesti, jos ei niitä ole luonnollisesti valaistuksessa syntynyt tarpeeksi. Jos keinotekoisesti sijoitetut pisteet eivät sovi kuvattavaan kohtaukseen, mutta ovat tarpeellisia trökkäyksen onnistumisen kannalta, pitää ne jälkikäsitellyssä sitten poistaa tai korjata videomateriaalista, jota käsittelen enemmän luvussa 3.2 Live-kuvan puhdistus. Suositus on, että erityisesti 3D-trökkäykseen näitä pisteitä tulisi olla ainakin 8–12 samanaikaisesti näkyvillä läpi koko kuvattavan kohtauksen, jotta kameran liikkeen arviointi onnistuisi (Dobbert 2012, 48). On siis hyvä jo lavastusvaiheessa varmistaa, että sopivia pisteitä on runsaasti käytettävissä kuvan liikkeen analysointiin.

Se, mitkä pisteet lopulta live-kuvassa soveltuvat trökkäykseen, ei ole aivan täysin pääteltävissä ennen kuin trökkäysprosessi aloitetaan. Niiden tulisi olla kuitenkin niin suuria, etteivät ne häviä videotallenteen kohinaan, mutta niin pieniä, että niistä tulee selkeä keskipiste merkille. Kannattaisi myös varmistaa, että pisteiden valoisuus ei muuttuisi liikaa, koska kontrasti on normaali tapa trökkäysohjelmalle tunnistaa niitä. Olisi myös tärkeää, että pisteitä löytyisi eri etäisyyksiltä ja monipuolisesti ympäri kuvaa. (Dobbert 2012, 44.) Käsittelen tätä lisää luvussa 3. 3D-trökkäys ja 3.1 2D-trökkäyspisteet ja 3D-merkit.

Olisi myös hyödyllistä dokumentoida muutaman trökkäyspisteen fyysinen etäisyys toisistaan tai jonkin elementin korkeus tai leveys ainakin senttimetrin tarkuudella. Se helpottaa virtuaaliympäristön skaalan esiasettelua, valojen sekä objektien esiasettelua ja esimerkiksi 3D-grafiikkaohjelman sisäisten syvästerävyysasetusten käyttämistä. Jos mahdollista, tämä kannattaisi tehdä kahden trökkäyspisteen välille, jotka selvästi voi todeta toimiviksi ja joiden välinen akseli olisi mahdollisimman suoraan sivusta kuvattu kameran näkökulmassa.

Omassa projektissa lavasteen keskelle asettamani pikkukivet ovat tärkeä lisä, vaikka joudun ne poistamaan lopullisesta kuvasta. Ne ovat ainoat pisteet lähempänä kameraa, jotka rikkovat pöydän monotonisen vaalean pinnan ja tarjoavat referenssin virtuaaliympäristöön tällä alueella. Ne ovat myös tärkeä apu trakkäysohjelmalle perspektiivin laskennassa. Kivet ovat noin 10 senttimetrin päässä toisistaan ja kahden kiven välinen etäisyys toisistaan tarjoaa skaalan virtuaalilavasteille. Taaempänä lavasteessa löytyy myös tummia kulmia ja varjoja, jotka sisätilassa kuvatussa materiaalissa ovat soveltuvia, koska valo joka varjot heittää ei itsessään liiku videokuvan aikana. Kuviossa 5 video-sekvenssin yksi kuva jossa on korostettu punaisella värillä ympäröitynä trakkäyspisteiksi soveltuvia kohteita.



Kuvio 5. Live-kuva (kuvakaappaus).

2.4 Tekniikan rajoitteet

Oikean maailman valojen ja varjojen virtuaalisten vastineiden luomisessa suurin tekninen rajoite on laskennallinen aika, vaikka ei ole mahdotonta luoda monimutkaista virtuaaligeometriaa paremmin sijaistamaan live-ympäristön pintoja. Jokainen polygoni, josta se rakentuu, lisää laskennallista aikaa 3D-grafiikkaohjelman lopullisessa renderöinnissä(Wikipedia 2015b). Sama vaikutus on myös 3D-grafiikkaohjelman valosimulaation tarkkuudella: aina kun ohjelma joutuu laskemaan valon kimpoamista pinnasta toiseen, lisää se laskennallista aikaa(Wikipedia 2015c). Laskennallisen ajan lisäksi virtuaaliympäristön työstäminen ja muut prosessin vaiheet vievät ihmiseltä huomattavan määrän aikaa. Tekniikka ei sinänsä ole rajoite, jos projektin resurssit ovat

rajattomat. Rajatuilla resursseilla illuusion luominen on taito itsessään ja sen onnistuminen pitkälti katsojan subjektiivinen näkemys. Ohjelmistojen yhteensopivuus tulee myös selvittää ennen teknisen toteutuksen aloittamista. Kaikki 3D-träkkäyssovellukset eivät ole yhteensopivia kaikkien 3D-grafiikkaohjelmien kanssa.

Omassa projektissa käyttämäni Blender 3D on jatkuvasti kehittyvä avoimen lähdekoodin ohjelma. Ohjelmassa on useita moduuleita erilaisen CG-grafiikan käsittelyyn ja luomiseen. Merkittävin tekninen rajoite omalle projektilleni paljastui Blender 3D -ohjelmassa virtuaalisten valojen luomisessa. Live-kuvan lavastamisessa käytettyjen oikean maailman valojen kopioiden luominen virtuaaliympäristöön ei tuottanut ongelmia, mutta kun halusin lisätä pelkästään virtuaalisen valolähteen, monimutkaisti se asioita huomattavasti. Tämä johtui projektissa käyttämäni Blender-ohjelman version rajoitteesta moneen muuhun moderniin 3D-ohjelmaan verrattuna. Käyttämäni ohjelman versiossa ei ollut mahdollisuutta erikseen valikoida, mihin kaikkiin objekteihin valittu valolähde vaikuttaa. Löysin kyllä lopulta tavan kiertää rajoitteen, mutta se monimutkaisti projektia niin paljon, että päädyin olemaan käyttämättä niin sanotusti kokonaan virtuaalista valolähdettä ollenkaan. Mahdollisuus valita esineen heittävän varjon vain itseensä ja samalla estämään toisen esineen heittämää varjoa piirtymästä kyseisen esineen taakse, ei projektin tekovaiheessa ollut mahdollista Blender 3D -ohjelmalla.

3 3D-träkkäys

3D-träkkäyksen tärkein lopputulema on virtuaalisen kameran liikeanimaatio: mitä onnistuneempi träkkäys on, sitä tarkemmin liikeanimaatio vastaa live-kuvan kameran liikettä. Träkkäyspisteiden kaksikulotteinen liike kuvasta toiseen ja erityisesti niiden toisiinsa nähden suhteellinen liike on se, jota Motion Tracking -sovellus käyttää arvioimaan live-kuvassa käytetyn kameran liikettä, kennon kokoa, polttoväliä ja linssin vääristymää. Sovellus käyttää projektiivista geometriaa matemaattisessa laskelmassa kolmiulotteisten sijaintien löytämiseksi trakkättäville pisteille. Tim Dobbartin kirja *Matchmoving: The Invisible Art of Camera Tracking (2nd edition)* käsittelee laajemmin prosessia ja tarjoaa viitteet jopa tarkempisiin teknisiin tietoihin matemaattisista kaavoista. (Dobbert 2012, 24.) Tämän tutkielman tarkoitus ei kuitenkaan ole tarjota kattavaa teknistä tietämystä 3D-träkkäysohjelmien toiminnasta.

Oleellista on kuitenkin huomioida muutama avaintekijä 3D-träkkäyksen onnistumiseksi. Koska sovellus käyttää projektiivista geometriaa avaruudellisten sijaintien laskemiseksi, seurattavia pisteitä pitäisi löytyä kaikilta etäisyyksiltä ja alueilta, joille virtuaaliobjekteja aiotaan sijoittaa (Dobbert 2012, 47). Kun sovellus analysoi esimerkiksi kahta kuvaa samasta ympäristöstä, mutta eri kuvakulmista. Sovellus voi tehdä matemaattisen laskelman seurattavien pisteiden kolmiulotteisesta sijainnista. Käyttäjän on kuitenkin varmennettava, että sovelluksen tekemä ratkaisu on oikea. Yleinen suositus on, että ainakin 8–12 pistettä olisi seurannassa koko ajan: jos osa näistä poistuu kuvasta, tulisi korvaavien pisteiden tulla niiden sijalle. Dobbert suosittelee jo ensimmäisessä kirjassaan portaistettua pisteiden poistumista ja korvaamista (Dobbert 2005, 49). Vaikka Dobbertin julkaisu on tätä kirjoittaessani jo kymmenen vuotta vanha, oman kokemukseni perusteella tämä on vieläkin hyvä käytäntö. Jos pisteiden sijainnit edustavat monipuolisesti kuva-alaa ja niiden seuranta onnistuu hyvin, sivutuotteena kameran liikkeen selvittämiseksi sovellus pystyy myös arvioimaan kameran ja linssin muita ominaisuuksia. Arvioidut ominaisuudet ja arvion tarkkuus, jotka ohjelma voi laskea videomateriaalista, vaihtelevat riippuen käytettävästä sovelluksesta ja materiaalista. Käsittelen tarkemmin pisteiden valintaa ja seuranta tapaa luvussa 3.1 2D-träkkäyspisteet ja 3D-merkit.

Riippumatta siitä, onko työssä tarkoitus upottaa 3D-grafiikkaa live-kuvan sisään tai greenscreen lavasteissa kuvattuja ihmisiä tai esineitä 3D-grafiikalla tuotettuun maisemaan. Erityisesti tapauksissa joissa kamera liikkuu kuvauksen aikana, 3D-träkkäys on prosessi, joka mahdollistaa näiden kahden saumattoman yhdistämisen.

Omassa projektissa käyttämäni Blenderin 3D-träkkäysmoduuli on pätevä työkalu kameran liikkeen selvittämiseen. Ohjelmasta kuitenkin puuttuu muutamia ominaisuuksia, joita on kalliimmissa ammattilaisohjelmissa, kuten esimerkiksi vaihtuvan kuva-alan otoksien tunnistaminen, joissa on käytetty muuttuvaa linssin polttoväliä. Automaattinen seurantapisteen lisäystoiminto ei myöskään ole mielestäni aivan samalla tasolla kuin parhaissa kaupallisissa ohjelmissa. Omaan projektiin se sopii mainiosti, koska kuvaamassani videosekvenssissä kameran polttoväliä ei muuteta kuvan aikana, ja vain kameran sijaintia sekä kuvaus-suuntaa muutetaan. Käsini lisättyjen träkkäyspisteiden seuranta toimii myös moitteettomasti.

3.1 2D-träkkäyspisteet ja 3D-merkit

Kun videokuva tuodaan Motion Tracking -ohjelmaan on yleensä mahdollista, riippuen ohjelmasta, antaa ohjelman automaattisesti etsiä seurattavia pisteitä, tai käyttäjä voi itse valita, mitä pisteitä ohjelma rupeaa seuraamaan asettamalla 2D-merkin manuaalisesti. Kun piste tai pisteet on valittu, ohjelma analysoi kuva kвалта jokaisen pisteen sijainnin X- ja Y-akselilla suhteessa kuvamateriaalin reunoihin.

Analyysin jälkeen ohjelma ilmoittaa arvion siitä, kuinka tarkasti 2D-merkki pysyi kuvasta toiseen valitun pisteen kohdalla. Tuloksen ohjelma ilmoittaa virhemarginaalina, joka on merkitty pikseleiden määränä. Useamman 2D-merkin analyysin jälkeen ohjelma pystyy tekemään arvion pisteiden sijainnista 3D-avaruudessa. Mitä useampia pienen virhemarginaalin pisteitä kohtauksessa on, sitä paremmin ohjelma käsittää avaruudellisen tilan kokonaisuudessaan.(Dobbert 2012, 70.)

Manuaalinen träkkäyspisteiden tai seurantapisteen valinta on varmin tapa varmistaa seurannan onnistuminen. Seurantapisteksi kelpaavilla elementeillä on tiettyjä kriteereitä. Seurattavien elementtien tulisi olla täysin staattisia live-ympäristössä, on myös vahvasti suositeltavaa, että ne ovat oikeita fyysisiä sijainteja (Dobbert 2012, 47). Seurantapisteksi soveltuvat huonosti pisteet, joissa valoisuus muuttuu merkittävästi kuvan aikana. Valotilanteen muuttumisen takia seuranta ei välttämättä onnistu perinteisellä seurantamenetelmällä. Varmistuksen kuitenkin saa vasta kun sovellus yrittää seurata pistettä. Joissakin ohjelmissa on mahdollisuus erikseen erotella värikanava, jolla piste tunnistetaan, vain yhden värikanavan seuraaminen voi auttaa valotilanteen muutoksen takia häilyvän kohteen seuraamista. Vaihtoehtoisesti käyttäjä voi myös esikäsitellä trakkättävän materiaalin nostamalla kontrastia tai sulkemalla pois värikanavia. Esikäsitelyä voi myös hyödyntää kuvamateriaalin kohinan takia huonosti seurattavien pisteiden selkeyttämiseksi käyttämällä kompositointiohjelman kohinasuodinta.

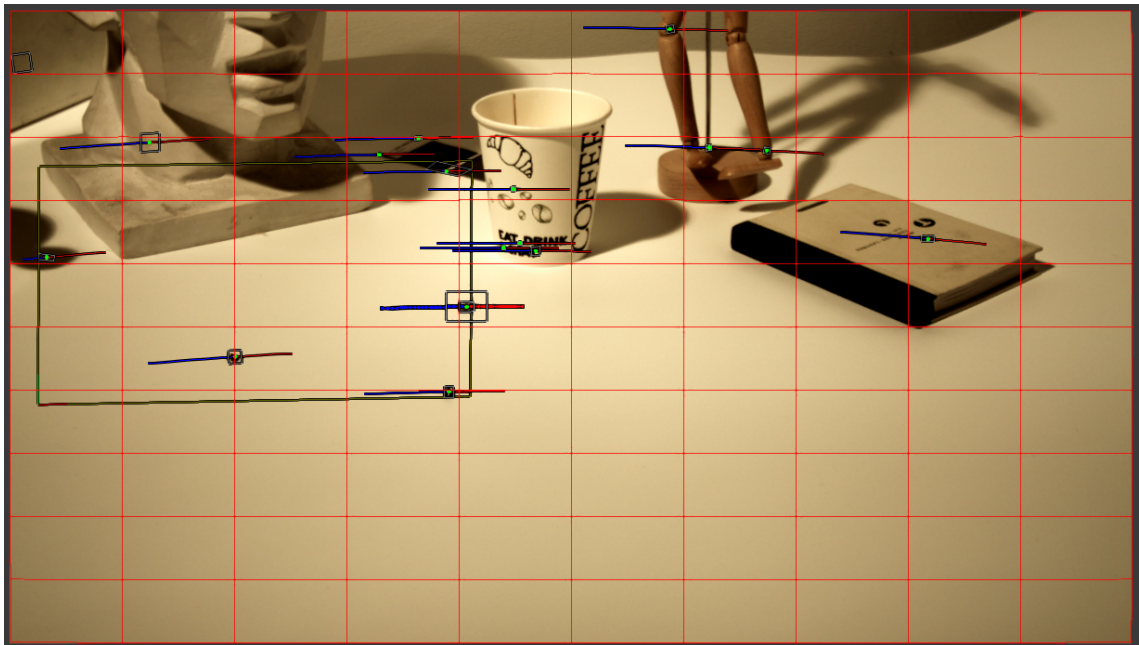
Automatisoitu seurantapisteen etsintä toiminto on käytännöllinen apuväline, mutta sen luotettavuus on vaihteleva (Dobbert 2012, 92). Yksinkertaisessa kuvassa, jossa on suurimmaksi osaksi vain staattisia objekteja, voi automatisoitua toimintoa käyttää jopa pääsääntöisesti. Ohjelman voi antaa etsiä seurattavia pisteitä itseksensä, mutta käyttäjän tulisi poistaa huonosti seuranneet 2D-merkit. Huonosti seuranneet 2D-merkit voivat olla korkean virhemarginaalin merkkejä tai vain lyhyen ajan seuranneet merkit.

Tämän prosessin voi toistaa monta kertaa ja potentiaalisena lopputuloksena on huomattava määrä automaattisesti lisättyjä seurantapisteitä. Automaation voi myös yhdistää manuaaliseen seurantapisteiden valintaan. Ennen lopullista ratkaisua, kun huonot seurantapisteet on poistettu yhtälöstä, tulisi käyttäjän vielä varmistaa, että kaikki seurattavat pisteet ovat olleet oikeita fyysisiä sijainteja live-ympäristöstä.

Monessa Motion Tracking -sovelluksessa on yleensä myös mahdollista antaa jokaiselle pisteelle niin sanottu painoarvo nollan ja yhden väliltä. Tämä määrittää merkin vaikutuksen lopputuloksen arvioinnissa. (Dobbert 2012, 95.) Tämä on hyödyllistä, koska se mahdollistaa huonomminkin paikallaan pysyvien 2D-merkkien käyttämisen viitteellisinä 3D-merkkeinä ilman, että se vaikuttaa negatiivisesti lopputulokseen.

Lopullinen varmennus käyttäjälle 3D-träkkäyksen onnistumisesta, saadaan vasta kun sovelluksen tekemä ratkaisu viehdään 3D-grafiikkaohjelman kolmiulotteiseen ympäristöön. 3D-ympäristössä käyttäjän tulisi ensin siirtyä kameran ulkopuoliseen näkökulmaan, jossa kameran liikeanimaatiota voidaan tarkkailla niin sanotusti ulkopuolelta. Silmämääräinen ulkopuolinen varmistus on ensimmäinen indikaatio siitä, onko animaation liike edes suunnilleen kuvattua kohtausta vastaava. Tarkemman käsityksen saa, kun suoritetun träkkäyksen 2D-merkit valitaan näkymään 3D-merkkeinä. 3D-merkit ovat träkkäysohjelman käsitys kuvasta seurattujen pisteiden avaruudellisista sijainneista. Kun kuvamateriaali ajetaan virtuaalikameran näkökulman taka-alalla samalla kun vain 3D-merkit on asetettu näkyviksi, käyttäjä pystyy varmistumaan, pysyvätkö ne videomateriaalin elementtien kanssa samassa paikassa. Lisävarmennetta voi hakea esimerkiksi asettamalla lattialla olevan merkin päälle jonkin testiobjektin. Kun animaatiota pyörittää testiobjektin kanssa, voi olla helpompi huomata, jos objekti ei näytä pysyvän omalla paikallaan. Vasta silloin, kun kaikki vaikuttaa tähän mennessä hyväksyttävältä, kannattaa alkaa luomaan tai sovittamaan geometriaa virtuaaliympäristöön. Lopuksi määritellään ympäristön skaala, joka toimii ennalta muistiinpanujen kahden keskeisen träkkäysmerkin fyysisen etäisyyden syöttämällä träkkäysohjelmaan. Riippuu täysin käytettävien ohjelmistojen yhteensopivuudesta, mikä kaikki selvitetystä tiedosta siirtyy lopulta käytettävään 3D-grafiikkasovellukseen. Käsittelen virtuaaliympäristön ja objektien asettelua tarkemmin luvussa 4. 3D-ympäristön valmistelu.

Ennen trakkäysprosessin aloittamista omassa projektissani kävin läpi live-kuvassa käytetyn kameran tiedettyjä ominaisuuksia. Käytin kuvaamiseen Blackmagic-merkkistä kameraa ja siihen asennettua toisen valmistajan 25 millimetristä linssiä. Kameran kennon koko löytyi helposti kameran valmistajan sivuilta ja pysyin syöttämään sen ohjelmalle. Linssi ei ollut tavanomainen, mutta lisäsin myös linssin polttovälin ohjelmalle aloitus arvoksi. Koska kuvaamani raakavideo oli hyvin haalea, käytin Adoben AfterEffects -ohjelmaa tehdäkseen uuden version videosta pelkästään trakkäämistä varten. Nostin videon uuden version kontrastia huomattavasti, jotta trakkäyspisteiden seuranta olisi ohjelmalle helpompaa. Kuviossa 6 Blender 3D -ohjelman trakkäysmoduulin näkymä jossa osa käyttämästäni trakkäysmerkeistä korkeamman kontrastin videossa.



Kuvio 6. Live-kuvan korkeampi kontrasti (kuvakaappaus).

Aloitin trakkäyspisteiden sijoittelun tapahtumapaikan ympäristöstä koska tämän alueen tarkat sijaintitiedot vaikuttaisivat eniten siihen miten hyvin virtuaaliobjektit pysyisivät oikeilla paikoillaan. Tämän alueen pisteet ovat myös keskeisimmässä osassa kuvaamaani videota koska kamera panoroi tapahtumapaikan vasemmalta oikealle ja takaisin. Trakkäyspisteiden seurantavaiheessa pisteet ovat vain kaksiulotteisia koordinaatteja kuvakehyksessä ja ohjelman voi komentaa seuraamaan niitä videon aikajanaa eteenpäin tai taaksepäin. Kun minusta tuntui siltä, että hyvin paikallaan pysyviä merkkejä oli tarpeeksi, asetin ohjelman ensimmäiseksi selvittämään pelkästään kameran liikettä. Tämä itsessään ei vielä tuottanut tyydyttävää tulosta, koska linssin vääristymä ja polttoväli eivät oletus asetuksilla täsmänneet. Muutama

träkkäyspisteistäni myös tuotti huomattavasti heikompia tuloksia. Ajoin seuraavaksi träkkäyksen muutaman kerran antaen ohjelman hienosäätää polttoväliä ja linssin vääristymää. Avasin liikekäyränäkymän, josta näkyi, kuinka yhteneväisiä pisteiden liikkeet olivat. Poistin huonosti seuranneet merkit, joiden liikekäyrä poikkesi villisti yleisestä trendistä, kokonaan tai vain aikajanan osiolta, jossa seuranta ei ollut onnistunut. Träkkäysmerkkejä oli lopulta yhteensä 24 kappaletta. Määritin kolme pöydän pinnalla ollutta träkkäysmerkkiä ohjelmalle tulkittavaksi yhdeksi pinnaksi. Lopputulos oli erittäin tyydyttävä 0.1709 pikselin virhemarginaali.

3.2 Live-kuvan puhdistus

Kuvamateriaalin niin sanottu puhdistaminen tai elokuva-alan kielessä yleisesti Wire Removal (Wikipedia 2015d). Puhdistaminen voidaan tehdä jo ennen virtuaaliympäristön luomista, jonka päälle virtuaalielementejä ja niiden sekundaarisia vaikutuksia voidaan sitten mallata. Kaikkia ongelmakohtia ei välttämättä voi kuitenkaan ennakoita, ja on mahdollista, että käyttäjä joutuu myöhemmin virtuaaliympäristön rakentamisen jälkeen palaamaan tähän. On myös mahdollista, että tämä vaihe osoittautuu järkevämmäksi tehdä kokonaan vasta myöhemmin. Mitä paremmin on mahdollista ennakoita, mitkä alueet kuvatusta live-ympäristöstä tulevat olemaan virtuaalielementtien vaikutuksen alaisia, sitä helpompi on tunnistaa ongelmakohtat.

Live-kuvan puhdistus ei ole aina välttämätöntä. Tapauksissa, joissa esimerkiksi luonnollisia träkkäyspisteitä on ollut kuvausvaiheessa vähän ja niitä on jouduttu keinotekoisesti lisäämään live-ympäristöön, voi näitä sopimattomia poikkeamia joutua poistamaan kuvamateriaalista. Myös valon heijastuksista voi koitua ongelmakohtia alueille, joissa virtuaaliobjektin varjo tai heijastus on tarkoitus näkyä. Näiden ongelmakohtien poistaminen voi myös olla tarpeellista, jotta lopullinen illuusio live-ympäristön ja virtuaaliobjektien valotilanteen interaktiosta säilyisi.

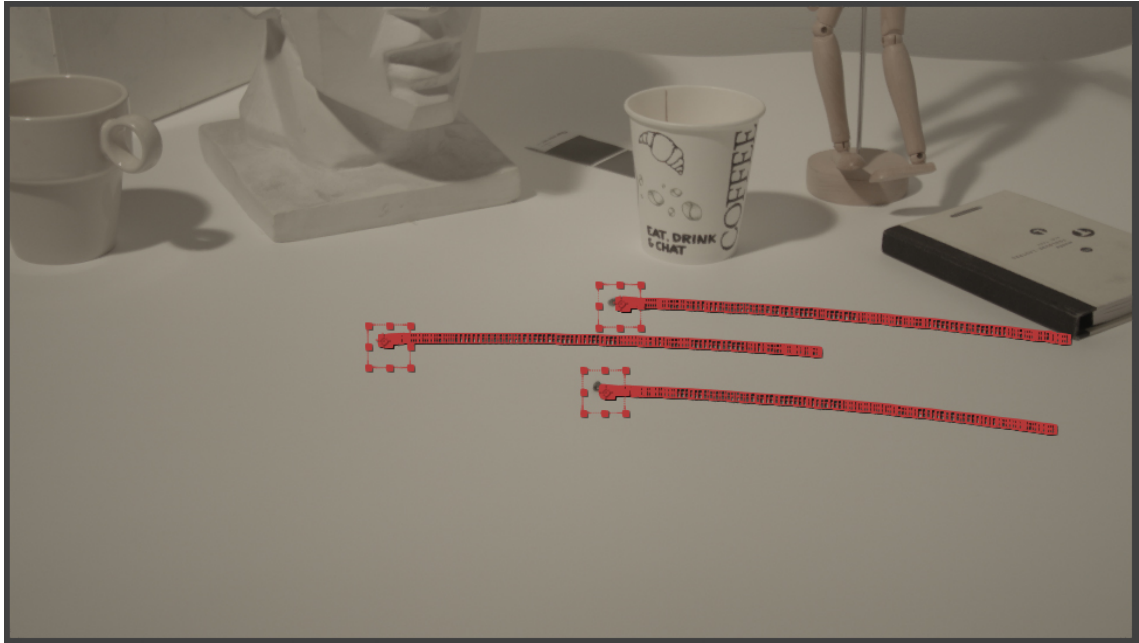
Koska lopputulos saavutetaan päällekkäistämällä alkuperäinen kuvamateriaali ja 3D-grafiikkaohjelmalla piirretty materiaali, ei alkuperäisen materiaalin valotilannetta muuteta, vaan siihen lisätään. Tästä syystä esimerkiksi pinta, jolle virtuaaliobjektin varjon olisi tarkoitus langeta, pitää etukäteen puhdistaa kirkkaista kiilteistä. Varsinkin kuvassa, jossa kameran sijainti vaihtelee huomattavasti, pitää myös huomoida, että kiilteitä voi ilmetä ajallisesti eri kohdissa kuvamateriaalia. Puhdistukseen tarvittavat

työkalut yleensä löytyvät kompositioiohjelman sisältä. Perinteisiä videokuvan puhdistuskeinoja ja työkaluja käyttämällä pääsee jo pitkälle.

Yksi tehokas apukeino korjata tai poistaa erityisesti tasaisilla pinnoilla olevia poikkeamia, kuten kirkkaita täpliä, on 2D-tasoträkkäys tai 2D Planar Tracking (Wright 2015). 2D-tasoträkkäyssovelluksia on useita. Käytännön ajatus on, että seurattava pinta määritellään valitsemalla kulmapisteet ongelmakohdan ympäriltä, minkä jälkeen ohjelma analysoi kulmapisteiden sisään jäävän pinnan kuvion perspektiivimuutoksen. Analyysin jälkeen kulmapisteiden 2D-koordinaatit voidaan viedä kompositointi ohjelmaan ja korvaava pintakuvio pystytään animoimaan videokuvan päälle samalla perspektiivimuutoksella. Tekniikka toimii parhaiten tasaisille pinnoille ja tasoille. Tekniikka voi olla ongelmallinen käyttää, jos valotilanne muuttuu voimakkaasti käsiteltävillä alueilla.

Toinen tehokas keino voi olla avaintaminen, eli *Keying*(Wikipedia 2015e). Tämä on erityisen hyödyllinen tekniikka, jos lavasteissa on kuvauksen aikana käytetty erikseen lisättyjä värillisiä merkkejä, erityisesti jos niitä on paljon. Avaintaminen mahdollistaa tietyn värin eristämisen kuvasta, joka voidaan sitten muuttaa kuvassa läpinäkyväksi. Tämän jälkeen taustalla voidaan pyörittää poistetun merkin ympärillä olevan pinnan väriä tai jopa samaa videokuvaa hieman siirrettynä. Avaintamista helpottaa roskamaski tai englanniksi Garbage Matte, jolla rajataan alue, johon avaintaminen vaikuttaa (Wikipedia 2015f).

Omasta live-kuvastani piti poistaa träkkäystä helpottamaan pöydälle asettamani pikkukivet. Käytin tässä AfterEffects-ohjelman sisäänrakennettua Track Motion -toimintoa seuraamaan jokaisen kiven kulman kärkeä. Kun pisteiden sijainnit oli selvillä asetin jokaisen pikkukiven päälle pehmeäreunaisen muodon, jotka asetin päästämään alla olevan kerroksen kuvan läpi ja seuraamaan träkkäysmerkkejä. Alempana kerroksena ajoin samaa videota hieman sivuun siirrettynä. Tämä korvasi kivien sijainnin viereisellä pöydän pinnalla, lopputulos on verrattavissa kuvankäsittelyohjelmien kloonaustryökaluun, joka seuraisi liikkuvassa kuvassa olevaa pistettä. Tämä on yhdenlainen automatisoitu tapa poistaa "roskat" suhteellisen tasaväriseltä pinnalta, ilman tarvetta manuaalisesti animoida roskamaskia. Kuviossa 7 AfterEffects -ohjelman näkymä, jossa kivien kulmat on pisteträkkätty Track Motion -toiminnolla.



Kuvio 7. AfterEffects pisteträkkäys (kuvakaappaus).

4 3D-ympäristön valmistelu

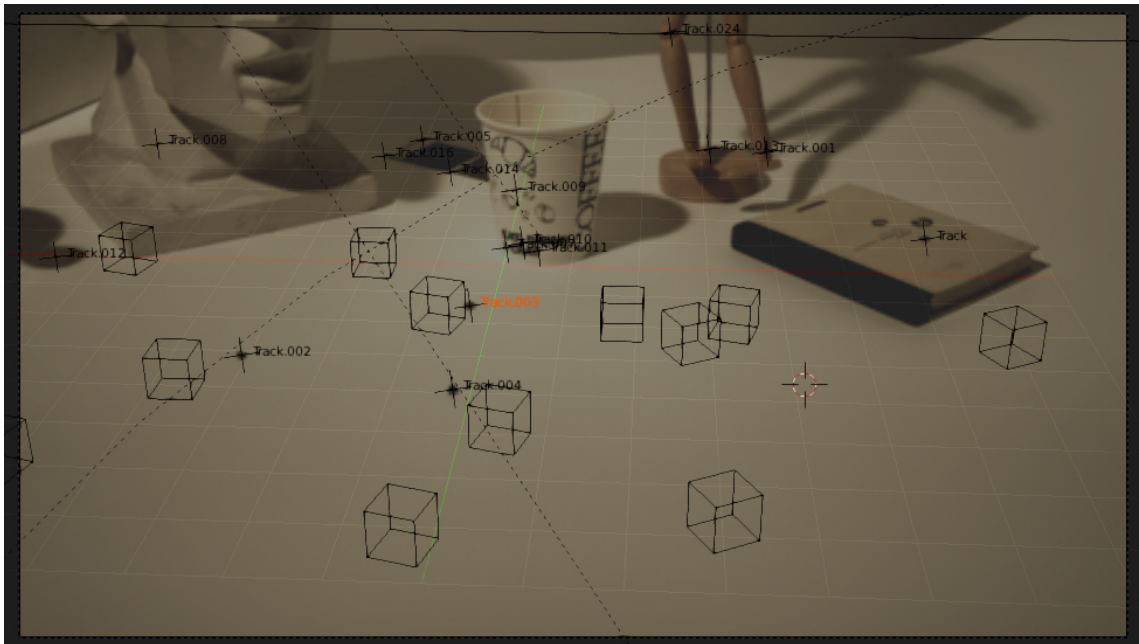
Kun 3D-träkkäysvaihe on onnistunut ja kuvaus on toteutunut suunnitelmien mukaan, kaikki edellytykset 3D-liikegrafiikan lisäämiselle otokseen ovat täyttyneet. Optimaalisessa tilanteessa 3D-träkkäyksestä ja kuvausten muistiinpanoista on saatu kameran liikeanimaatio, kameran kennon koko, linssin vääristymät, polttoväli ja valotusaika. Nämä tiedot ovat se, millä 3D-grafiikkaohjelman tuottama lopullinen kuva saadaan parhaiten vastaamaan live-kuvaa.

Virtuaaliympäristön skaala tulee yleisesti mukana, kun Motion Tracking -sovelluksessa tuotettu virtuaalikamera viedään 3D-grafiikkaohjelmaan. Jos näin ei ole, voidaan skaala kuitenkin määritellä 3D-grafiikkaohjelman sisällä. Virtuaalikamera ja sen animaatio on elementti 3D-grafiikkaohjelman sisällä kuten mikä muu tahansa objekti, ja sitä voidaan pienentää tai suurentaa tarpeen mukaan (Dobbert 2012, 139). 3D-grafiikkaohjelman näkökulmasta kaikki mittayksiköt ovat vain suhteellisia toisiinsa, mutta kun käyttäjä kertoo ohjelmalle yhden yksikön kuvastavan esimerkiksi yhtä oikean maailman metriä, sovelluksen kameran simulaatio osaa suhteuttaa esimerkiksi syvästerävyysasetukset määritellyyn kennon kokoon ja polttoväliin.

Liikeanimaatio on tärkein yksittäinen virtuaaliympäristön rakentamiseen tarvittava osa. Ilman tarkkaa live-kuvan kameraa jäljittelevää liikeanimaatiota olisi melkein mahdotonta saada kuvaan lisättäviä 3D-elementtejä pysymään paikoillaan. Liikeanimaation selvittämisen sivutuotteena saadaan myös avaruudellisia kiinnepisteitä. Kiinnepisteiden lisäksi virtuaalikameraan voidaan lisätä kuvasarja pyörimään samaan aikaan animaation kanssa. Kun taustalla pyörivä live-kuva yhdistetään 3D-merkkien sijainteihin, tulee virtuaaliympäristön muotoilusta ja sen todenmukaisuuden varmentamisesta huomattavasti helpompaa.

Kameran sekundaariset ominaisuudet kuten valotusaika liike-epäterävyyden simuloimiseen, linssivääristymät kuva-alan laidoilla sekä polttoväli ja kennon koko ovat myös muutettavissa. Tämän pitäisi kuitenkin olla tarpeellista vain tapauksissa, joissa tarkkoja muistiinpanoja kuvauksista ei ole käytettävissä ja 3D-träkkäysohjelman luoma arvio on jostain syystä puutteellinen.

Aloitin omassa projektissani virtuaaliympäristön valmistelun asettamalla kohtaukseen yksinkertaisen 3D-tason, jonka yritin asettaa samaan linjaan live-kuvan pöydän pinnasta seurattujen pisteiden kanssa. Tason päälle asetin 3D-kuutioita, jotka auttaisivat visuaalisessa sijaintien varmennuksessa. Tein 3D-tasosta näkymättömän ja laitoin live-kuva sekvenssin näkymään 3D-kuutioiden takana samalla kun pyöritin 3D-kameran liikeanimaatiota. Tällä tavalla pystyin silmämääräisesti näkemään kuinka luonnollisen näköisesti 3D-kuutiot tuntuivat pysyvän oikean maailman pöydän pinnassa. Muokkasin 3D-tason ja -kuutioiden sijaintia ja asentoa kunnes lopputulos tyydytti silmää. Tarkemman hienosäädön tekisin myöhemmin, kun sijaisgeometria live-kuvassa esiintyville monimutkaisemmille esineille on valmis. Sijaisgeometriasta lisää luvussa 4.2 Interaktiivinen sijaisgeometria. Seuraavaksi valitsin kaikki objektit mukaanlukien 3D-kameran ja linjasin tekemäni 3D-tason horisontaalisesti 3D-ohjelman koordinaatiston nolla korkeudelle. Tämä helpottaa uusien 3D-objektien luomista ja paikoilleen asettelua. Kuviossa 8 Blender 3D -ohjelman näkymä, jossa 3D-träkkäysmerkit sekä 3D-testiobjektit live-kuvan päällä.



Kuvio 8. Blender 3D koordinaatisto (kuvakaappaus).

4.1 Valaistuksen määrittely

Jos kuvausten valmisteluvaiheesta olevia muistiinpanoja on käytettävissä live-ympäristön valaisussa käytetyistä valoista ja niiden sijainneista, voi sitä nyt käyttää määrittelemään lähtökohdat virtuaaliympäristön vastineille. Muistiinpanot nopeuttavat prosessia mutta eivät ole pakollisia. Valojen uudelleen tuottaminen virtuaaliympäristöön vaatii joka tapauksessa hienosäätöä. 3D-grafiikkaohjelmissa ei ole kovin tarkkoja esiasetuksia valoille. Tarkin lopputulos muodostuu käyttäjän omasta visuaalisesta varmenteesta videomateriaalin valotilanteen ja virtuaaliympäristön valotilanteen vertailussa. Suuri vaikutus tähän vaiheeseen on myös 3D-grafiikkaohjelman piirtämän kuvan valosimulaatioasetuksilla (Blender Foundation 2015a).

Valokohtaisia asetuksia esimerkiksi sekundaaristen heijastuskimpoamien määrää voi muuttaa yksittäisille valoille. On kuitenkin ehkä järkevämpää aloittaa säätämällä valoasetukset yleisesti koskemaan kaikkia valoja, kokeilla valotilanteen käyttäytymistä joillain objekteilla ja vasta myöhemmin vähentää valosimulaation tarkkuutta yksittäisille valoille. Pelkästään suoravalaistus tai englanniksi *Direct Lighting* on luultavasti harvoin tämänkaltaiselle projektille riittävä. Toisaalta täysi globaali valaistus eli *Full Global Illumination* ei välttämättä ole myöskään tarpeellinen. Täysi globaali valaistus on 3D-grafiikkaohjelmissa esiintyvä esiasetus valosimulaatiolle. Sen tarkemmat asetukset vaihtelevat ohjelmakohtaisesti ja eri ohjelmat ylipäätensä käyttävät eri menetelmiä valojen simulointiin. Globaali valaistus käsittää yleisesti korkean määrän

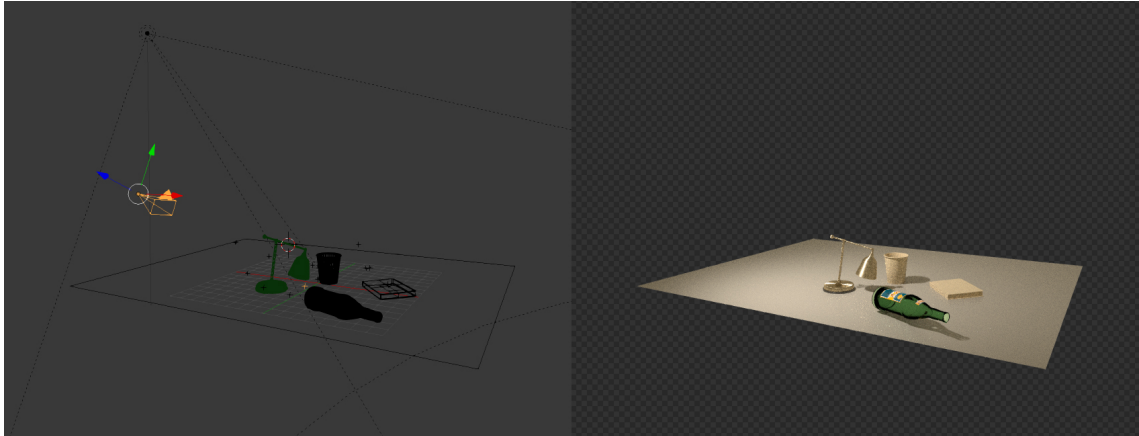
heijastuskimpoamia, heijastumien värjäytymistä, sekä mahdollisesti pintojen läpivalaisua ja kohdepinnan läpäisyä.(Wikipedia 2015g.)

Riittävä valosimulaation tarkkuus riippuu paljolti kuvauksissa käytettyjen valojen kirkkaudesta ja pintojen materiaaleista. Himmeämmillä valoilla kuvatussa live-kuvassa on yleensä luonnollisesti huomaamattomammat heijastukset. Sen lisäksi sekundaariset heijastukset eivät ole enää niin selkeästi näkyviä. Itse valolähteiden luomisessa on helppo aloittaa esiasetuksilla ja muokata niitä paremmin vastaamaan oikean maailman vastineita. Jos kuvauksissa on kuitenkin käytetty yleisvalaistukseen esimerkiksi valaistua kangasta, voi vastaavan kokoisen objektin kyllä muotoilla ja asettaa sen valolähteeksi.

Kuvapohjainen valaistus eli *Image-based lighting* voi tulla tarpeelliseksi esimerkiksi jos 3D-hahmo, joka kuvassa esiintyy, sisältää melkein peilimäistä heijastavaa pintaa(Wikipedia 2015h). Illuusion säilyttämiseksi heijastavasta pinnasta tulisi siis näkyä myös live-kameran kuvan ulkopuolelle jäävä ympäristö. Vaihtoehtoisesti live-ympäristön pehmeitä valolähteitä voi olla niin paljon, että niiden manuaalinen asettelu ei ole järkevää. Ympäristön valaisuun ja virtuaalisen näkökentän täydentämiseen käytetään parhaassa tapauksessa 360 astetta käsittävä HDRI-kuva live-lavasteista(Wikipedia 2015i). Kuva kietoutuu 3D-grafiikkaohjelmassa virtuaaliympäristön ympärille jokaiseen suuntaan. Kuvana voi myös käyttää erityisesti sitä varten koostettua mielikuvituksellista kuvaa, eikä sen ole pakko kattaa koko ympäristöä. Riippuen käyttötarkoituksesta sen ei välttämättä tarvitse olla olemassa mistään muusta syystä kuin tuomaan variaatiota valotilanteeseen.

Päätin käyttää omassa projektissani virtuaalista vastinetta isolle studiovalaisimelle sekä kuvapohjaista valaistusta värjäämään hieman virtuaaliobjekteja eri suunnista samaan tyyliin kuin live-ympäristön studion vaaleat seinät. Voimistin ympärillä olevan kahden seinän vaikutusta asettamalla muutaman himmeästi heijastavan 3D-tason suurin piirtein seinien sijaintia vastaaville paikoille. Täyden 360 astetta kattavan HDRI-kuvan sijasta käytin vain kuvakaappausta live-kuvasta värjäämään virtuaaliobjekteja. Studio oli sen verran himmeä lukuun ottamatta studiovalaisimen valoa, että pöydällä olevien esineiden heijastuksista tuskin näki mitään tapahtumapaikan ympäristön ulkopuolista. Erillisen 360° HDRI-kuvan rakentaminen 3D-mallina ei olisi tuonut merkittävää parannusta valotilanteeseen, eikä käytettävissä ollut kuvauspaikalta otettua oikeata 360° kuvaa. Studiovalaisimen virtuaalisen vastineen koon ja osoitus-suunnan pystyin

hienosäätämään, kun muotoilin sijaisgeometrian live-kuvassa olevalle pahvikupille. Muokkasin valon kokoa ja asentoa kunnes virtuaalisen pahvikupin varjo täsmäsi oikean maailman kahvikupin kanssa. Tästä lisää seuraavassa luvussa 4.2 Interaktiivinen sijaisgeometria. Kuviossa 9 Blender 3D -ohjelman näkymä, jossa virtuaaliympäristö kauempaata kuvattuna. Vasemmalla rautalankamalli ja oikealla sama kuvakulma valosimulaation ja pintamateriaalien kanssa.



Kuvio 9. Blender 3D (kuvakaappaus).

4.2 Interaktiivinen sijaisgeometria

Sijaisgeometria eli englanniksi *Proxy Geometry* on *Matchmoving* -artistien käyttämä termi, jolla kuvataan virtuaaliympäristöön rakennettavia pintoja ja objekteja, jotka ovat tavallaan kopioita live-kuvan oikean maailman esineille ja pinnoille. Sijaisgeometrian tarkoitus on isommissa tuotannoissa yleensä antaa muille CG-grafiikkaa projektissa työstäville ihmisille virtuaalinen sijaintireferenssi live-ympäristön tärkeille pinnoille ja avainobjekteille (Dobbert 2012, 135). Interaktiivinen sijaisgeometria on termi, jota käytän kuvaamaan sijaisgeometriaa, jonka on tarkoitus olla näennäisessä vuorovaikutuksessa kohtaukseen lisättävien puhtaasti virtuaalisten objektien kanssa, eli niin sanotusti reagoida virtuaaliobjektien heijastamaan valoon ja niiden langettamiin varjoihin. Tätä virtuaalista sijaisgeometrian pintaa itseään ei tule näkymään lopullisessa kuvassa, vaan tarkoitus on, että vain valot ja varjot, jotka virtuaaliobjekteista lankeaa pinnoille on näkyvää. Näillä pinnoilla voi myös olla virtuaalimaailman fyysisiä ominaisuuksia, esimerkiksi jos 3D-animaatiossa on valittu käyttää fysiikkamallinnusta, voivat ne toimia eräänlaisena virtuaalisena kosketuspintana. Pintojen piirtoasetuksista 3D-grafiikkaohjelmassa määritellään virtuaalikameralle näkymättömäksi kaikki muu paitsi pintaan heijastuva valo ja varjot (Blender Foundation 2015b).

Tarve sijaisgeometrialle voidaan alustavasti päätellä kuvakäsikirjoituksesta, mutta lopullinen tarve varmistuu vasta kun lopullinen objekti- tai hahmoanimaatio on 3D-grafiikkaohjelmassa paikallaan. Alustavasti on järkevää tuottaa erittäin yksinkertaisia geometrisia muotoja tapahtumapaikalle, jotta voi varmistua, mihin pintoihin CG-hahmon olemassaolon tulisi kuvassa vaikuttaa.

Sijaisgeometrian tuottamiseen paras apuväline ovat 3D-merkkireferenssit, joita 3D-träkkäysprosessista on syntynyt. Parhaassa tapauksessa niitä on riittävästi, että vähintään tasaiset pinnat voi melkein suoraan mallintaa näistä koordinaateista. Jos live-ympäristön pinnoissa on hienojakoisempaa pintamuodostelmaa mutta varjot ovat pehmeitä, tasainen pinta voi olla täysin riittävä illusion luomiseksi. Joissain tapauksissa voi kuitenkin olla tarpeellista rakentaa hienojakoisempi epätasaisuus, jolloin on luultavasti järkevämpää säätää pinnan kuviointia objektin tekstuuriasetuksista sen sijaan, että mallintaisi sen geometriana. Tekstuuriasetuksista hieman lisää luvussa 4.3 Varjo- ja heijastuspinnat.

Sijaisgeometrian muotoilussa ja paikalleen asettamisessa voi mennä huomattava määrä aikaa. Iteratiivinen elementtien muodon ja sijainnin jalostus ja lopputuloksen jatkuva arviointi on luultavasti tarpeellista. Yleisesti 3D-grafiikkaohjelmien asetuksissa pitäisi olla mahdollista säätää reaaliaikaisesti piirretyn kuvaportin kuva niin, että live-kuvaa voidaan pyörittää 3D-elementtien taustalla 3D-elementit piirrettynä osittaisella läpinäkyvydellä, tai live-kuva voidaan myös pyörittää etualalla osittaisella läpinäkyvydellä. Näin voidaan visuaalisesti varmentaa, että sijaisgeometria on oikean muotoista ja oikealla paikalla.

Omassa projektissani tärkein interaktiivinen sijaisgeometria on valkea pöytätaaso. Tasaisen kamerasta poispäin pakenevan pinnan tarkkaa sijaintia on vaikea todentaa ilman muita objekteja. Sain kuitenkin hyvän lähtötilanteen 3D-ympäristön valmistelusta. Tämä pohjatyö mahdollisti sen, että pystyin aloittamaan toiseksi tärkeimmän sijaisgeometria objektin, eli pahvisen kahvikupin, mallintamisen suoraan omalle paikalleen virtuaalisen pöydän pintaan. Tieto oikean maailman kahvikupin sylinterimäisen pyöreästä muodosta kuitenkin paljasti epätarkkuuden virtuaalipöydän kallistuskulmassa, kun virtuaalikuppi oli omalla paikallaan. Kun hienosäädön jälkeen virtuaalipöytä ja -kahvikuppi vastasivat tyydyttävästi live-ympäristön vastineitaan, pystyin helposti säätämään virtuaalivalaisimen oikealla paikalle ja oikeaan kulmaan täsmäämällä virtuaalikahvikupin varjon live-kuvan kahvikupin varjoon. Kun 3D-

ympäristö oli näin varmennettu, kolmannen sijaisgeometria objektin mallintamisessa pystyin luottamaan pöydän kallistuskulmaan muistilehtiön pohjan asettelussa, sekä lehtiön heittämään varjoon, hieman toispuoleisesti pöydästä ylös kurkottavan lehtiön kannen korkeuden säätämässä.

4.3 Varjo- ja heijastuspinnat

Kuten olen luvussa 4.2 Interaktiivinen sijaisgeometria aikaisemmin todennut, itse pintoja jotka esiintyvät live-ympäristössä ei virtuaaliympäristöstä piirretä. Sijaisgeometrian objektien piirtoasetuksista itse pinta säädetään virtuaalikameralle näkymättömäksi, ja vain asetukset, jotka antavat objektin vastaanottaa ja heittää varjoja sekä heijastaa valoa, jätetään näkyväksi. Jos heijastuvan valon värjäytymistä halutaan hyödyntää lopputuloksessa, tulee objektin väripinnan kuitenkin vastata live-ympäristön vastakappaletaan, vaikka se ei ole suoraan näkyvillä. Se, miten toimivat asetukset saavutetaan, on ohjelmakohtaista.

Illusion luomisessa tasaiset pinnat voivat tapauksesta riippuen olla riittäviä. Tapauksissa, joissa live-ympäristön pinnoilla on hienojakoisempia pintakuvioita, joita ei geometrian mallinnusvaiheessa ollut järkevää tuottaa, voidaan käyttää hyödyksi nykyaikaisten 3D-grafiikkaohjelmien kykyä tuottaa geometriaa tai kohokuviointia kuvista. Tässä voidaan käyttää hyödyksi live-kuvan lavasteista otettuja valokuvia tai kameraprojektiota. 3D-objektin pinnan geometriaan vaikuttavaa kuvapohjaista generointia kutsutaan nimellä *Displacement Mapping* ja kohokuviointia tai pinnan pelkästään valon käyttäytymiseen vaikuttavia yleisesti käytettyjä tekniikoita nimillä *Bump Mapping* ja *Normal Mapping* (Russell 2015). Itse kuva, jota tekniikoissa käytetään voi olla kuva oikeasta live-ympäristön pinnasta, joka on kuvauksien aikana otettu erikseen. Vaihtoehtoisesti koska live-kuvaa pyöritetään 3D-grafiikkaohjelmassa, voidaan siitä projisoida tekstuuri virtuaalielemeentien päälle suoraan, tätä kutsutaan englannin kielellä nimellä *Projective Texture Mapping* (Wikipedia 2015j). Projisoitu kuva voidaan asettaa vaikuttamaan kaikkiin pinnan ominaisuuksiin, mutta se voi muodostua ongelmalliseksi kohokuviointin osalta, jos pinnalla on live-kuvassa huomattavaa valotilanteen vaihtelua. Tämä johtuu siitä, että tummat kohdat pinnassa ohjelma ymmärtää syventyminä ja vaaleat ulontumina. Tällaisessa tapauksessa pintakuviointi pitäisi eristää jostain erityisestä kohdasta kuvasarjaa ja esikäsitellä paremmin vastaamaan pinnan tilaa suhteellisen neutraalissa valotilanteessa.

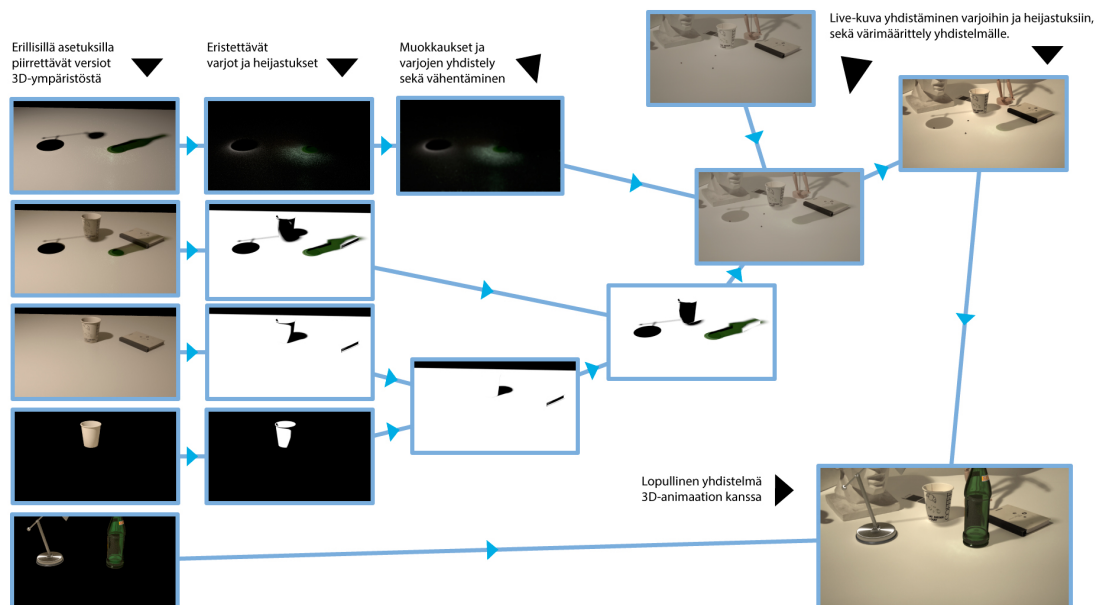
Oman projektini sijaisgeometrian materiaaliasetuksissa käytin tasaista väriä pöydän pinnan tekstuurina. Pahvisen kahvikupin ja muistilehtiön tekstuurin projisoin live-kuvasta, tämä ei tuottanut mielestäni kovin huomattavaa parannusta varjoihin ja heijastuksiin verrattuna hyvin valittuun yksittäiseen värisävyyn. Omasta mielestäni merkittävin vaikutus kuvan yhteneväisyydelle tuli nimenomaan live-kuvan kanssa samankaltaisten värisävyjen käyttämisestä. Itse 3D-animaation hahmojen materiaalit määritin ensin erillisinä vastaamaan kuvitteellisia oikean maailman esineitä. Virtuaalisen pöytälamppun materiaali simuloi harjattua terästä ja lasipullon materiaali värillistä lasia. Samalla ne myös heijastavat virtuaaliympäristön muita objekteja. Lisäsin 3D-hahmoille myös hienovaraisen heijastuksen live-kuvasta. Lasipullolle myös määritin live-kuvan näkymään lasin läpi. Ohjelman omasta lasimateriaali asetuksesta pystyi määrittämään kuinka paljon materiaali päästi läpi taustaa ja kuinka paljon se taittoi taustalta tulevaa valoa.

5 Kompositointi

Kompositointi eli englannin kielessä *Compositing* (Wikipedia 2015k). Kompositointi tarkoittaa elokuvan jälkikäsittelyssä eri lähteistä tulevien visuaalisten elementtien yhdistämistä. Vähimmäisvaatimus tämän tutkielman osalta koostettavalle materiaalille on live-kuva ja 3D-grafiikkaohjelman piirtämä kuva virtuaalikameran näkökulmasta, jossa 3D-hahmo on täysin näkyvä ja sijaisgeometrian pintatekstuuri on näkymätön, mutta varjot ja heijastukset ovat näkyviä. Kuitenkin voi olla järkevää jakaa 3D-grafiikkasovelluksen tuottama kuva useampaan osaan: esimerkiksi 3D-hahmo yhteen kuvasarjaan, varjot toiseen ja heijastukset kolmanteen. Riippuen käytetystä 3D-grafiikkaohjelmasta voi myös olla laskennallisesti nopeampaa tuottaa erilliset vedokset ja päällekkäistä ne myöhemmin. Koostettavien elementtien eristäminen omiksi kuvasarjoiksi tarjoaa myös enemmän hallinta- ja hienosäätömahdollisuuksia. Tämä voi hyvinkin olla tarpeellista, jos varjojen tai heijastusten sävyä tai voimakkuutta pitää muuttaa, jotta valotilanteen yhtenäisyys kokonaisuudessa säilyisi.

Kuten mainitsin luvussa 3.4 Tekniikan rajoitteet kun käsittelin omaa projektiani, puuttui Blender 3D -ohjelmasta projektin tekovaiheessa mahdollisuus valikoivasti hallita mihin objekteihin valot vaikuttavat. Tämän vuoksi jouduin omaan projektiini kehittämään tavan eristää haluamani valot ja varjot erillisinä vedoksina 3D-ympäristöstä piirrettävistä versioista, sekä niiden eristetyistä osista. Tässä hyödynsin Blender 3D

-ohjelman noodipohjaista kompositointimoduulia. Kuviossa 10 on havainnollistettu käyttämäni kompositointi prosessia.



Kuvio 10. Kaavio live-kuvan ja virtuaaliympäristön kompositoinnista.

Kuvassa on eroteltuna erilaiset versiot 3D-ympäristöstä ja niistä eristetyt heijastukset, varjot ja 3D-objektit. Kuvan tarkoitus on havainnollistaa, miten Blender 3D -ohjelmassa saavutetaan varjojen ja heijastuksien erottelu ja niiden päällekkäistäminen live-kuvaan 3D-animaation kanssa. Kuten kuvasta voi havaita 3D-ympäristöstä piirretään erillisiä versioita mistä kerrallaan näkyvillä on vain jokaiseen vedokseen tarvittavat objektit, heijastukset ja varjot. Näitä vedoksia yhdistelemällä ja toisistaan vähentämällä esimerkiksi varjojen tapauksessa poistetaan 3D-ohjelman piittämät varjot sijaisgeometrian esineiltä, joilla live-kuvassa on jo näkyvä varjo. Näin 3D-hahmojen varjot voidaan päällekkäistää live-kuvan kanssa ilman että varjo tummentuu kaksinkertaiseksi ja illuusio yhtenäisestä valotilanteesta syntyy. Tämä tapa on tavallaan kiertoratkaisu jolla Blender 3D -ohjelmassa saavutetaan ehkä muissa ohjelmissa yksinkertaisemmin toteutettava varjojen ja heijastuksien erottelu. Blender 3D -ohjelman tapa simuloida valoa ja piirtää kuva on kuitenkin mielestäni monella tapaa sopivampi yhdistelmä filmikameroiden tuottaman rakeisen kuvan kanssa.

5.1 Kompositointikerrosten värisäätö ja suodatus

Kun alkuperäinen videomateriaali ja 3D-animaation kuvasarjat ovat kerrostettu kompositointiohjelmassa, voidaan CG-materiaalin värisävyjä paremmin hienosäätää vastaamaan live-kuvan valotilannetta jos se on tarpeellista (Coots 2007). Värisäätö voidaan myös tehdä kaikille kerroksille yhteisesti määrittelemällä yleistä väritasapainoa, sävyä, kontrastia ja kirkkautta, tai erikseen jokaiselle kerrokselle. Näin voidaan vielä parantaa live-kuvan ja CG-materiaalin yhteneväisyyttä. Kompositoinnissa voidaan myös tehdä hienosäätöä eri kerrosten suhteelliseen vaikutukseen kokonaisuudessa. Yleisesti kompositointiohjelmissä on mahdollisuuksia valita eri kerroksille eri suodatustiloja, joilla esimerkiksi liian tummaksi mennyt varjokerros voidaan säätää päästämään kauttaaltaan alemman kerroksen kuvaa enemmän läpi tai erityisesti juuri valoisia kohtia hohtamaan hieman läpi. (Wikipedia 2015l.)

Omassa projektissani huomasin että hyvin valitut värisävyt 3D-objektien pinnoissa tuottivat jo kohtalaisen lopputuloksen, mutta tarvetta kuitenkin oli jokaisen kuvakerroksen värikorjaukselle, jotta lopputuloksena olisi yhtenäinen kokonaisuus. Blender 3D -ohjelman kompositointimoduuli tarjosi mielestäni täysin riittävät työkalut tähän, ja noodipohjainen järjestelmä mahdollisti väri käsittelyn jokaisessa välivaiheessa. Varjokerrosten päällekkäistäminen live-kuvan kanssa toimi mielestäni parhaiten nimenomaan erillistä ”yhdistä” suodatustilaa käyttämällä, sen sijaan että läpinäkyvyyttä pelkästään olisi muutettu.

5.2 Efektit

Efektien lisääminen kuvasarja tasolle voi koitua tarpeelliseksi. Jos esimerkiksi live-kuvamateriaalissa esiintyy kohinaa, on se mahdollista simuloida efekteillä myös CG-materiaaliin (Adobe 2015a). Näin vielä kerran voidaan vähentää eroja live-kuvan ja CG-materiaalin välillä. Kuten värisäädössä, tämä voidaan myös laittaa vaikuttamaan kaikkiin kerroksiin yhteneväisesti, live-kuva sekä CG-materiaali. Tilanteissa, joissa huomattavaa kohinaa ei alkuperäis materiaalissa ole eikä sitä CG-materiaaliin erillisenä lisätä, se voi luoda vielä voimakkaamman siteen materiaalien välille, jos se lisätään lopullisen komposition päälle. Kompositointiohjelman efektejä voidaan käyttää myös simuloimaan alkuperäisen materiaalin liike-epäterävyyttä (Adobe 2015b). Efekteillä voidaan myös simuloida syväterävyyttä ja vaikka lisätä linssiheijastuksen koostetun materiaalin päälle.

Omassa projektissani mielestäni toteutukseen tarvittavat efektit kuten syväterävyys 3D-objekteille, sekä virtuaaliobjektien varjoille ja heijastuksille, sai säädettyä kätevästi Blender -ohjelman kompositointimoduulissa. Vaikka lopputulosta voisi aina parantaa, en nähnyt erillistä tarvetta lähteä suuremmin käsittelemään Blender 3D -ohjelmasta ulos saamaani kuvasarjaa. Blender 3D -ohjelman Cycles -piirtomoottori tuottaa luontaisesti kohinaa laskiessaan valon kimpoamia, ja siten sopii mainiosti filmikameralla kuvattuun videomateriaaliin tehtävien visuaalisten efektien tuottamiseen. Kuviossa 11 Blender 3D -ohjelmalla tuottamani komposition yksittäinen kuva.



Kuvio 11. Blender 3D -ohjelman kompositio (kuvakaappaus).

Kuvasta näkyy kuinka vedokset päällekkäistettynä muodostavat illuusion valon ja varjon interaktiosta virtuaalielementtien ja live-ympäristön välille.

6 Yhteenveto

Koska 3D-träkkäys voidaan tuottaa vain muutamalla kuvalla että virtuaaliset koordinaatit elementeille saadaan käyttöön, tämän tutkielman tekniikat voivat olla työkustannusten kannalta tehokkaampi ratkaisu esimerkiksi realistisen still-kuvan rakentamiseen. Mielikuvituksellisen hahmon tai objektin istuttaminen valokuvaan tai kuvasarjaan on tekniikan kehittyessä jatkuvasti nopeampaa, ja uudet tekniikat syrjäyttävät vanhempia tekniikoita. Ei ole mitään yhtä oikeata tapaa toteuttaa dynaamista valotilannetta live-kuvan ja CG-materiaalin välille. Tärkeintä on mielestäni tekniikoiden ja ohjelmistojen luova yhdistely optimaalisen lopputuloksen saavuttamiseksi niillä resursseilla, mitä projektiin on käytettävissä.

Omaan projektiin käyttämäni monimutkaisemman kompositointi menetelmän kehittäminen oli mielestäni hyvin haastavaa. Tämä johtui siitä, että olemassa olevaa dokumentaatiota en vastaavalle menetelmälle löytänyt. Tämä menetelmä Blender 3D -ohjelmaa käyttäen kuitenkin tuottaa omasta mielestäni vakuuttavan lopputuloksen etenkin läheltä kuvattuun videokuvaan. Kun kaikki kuvan tuottamiseen vaadittavan ketjun osat on kerran määritelty, ohjelman voi suoraan laittaa piirtämään kaikki kuvat videosekvenssiin ilman että erillisiin kuviin tarvitsee tehdä mitään muutoksia. Uusien projektien rakentaminen tälle pohjalle on myös suhteellisen yksinkertaista. Live-kuva, sijaisgeometria ja 3D-hahmot vain korvataan uuden projektin vastaavilla osilla. 3D-träkkäys ja kuvasekvenssin piirtäminen vievät oman aikansa joka tapauksessa, riippumatta siitä millä menetelmällä lopputulos saavutetaan.

Kaiken kaikkiaan projektini onnistui omasta mielestäni hyvin, vaikka kompositointi menetelmän kehittämisessä meni pitkä aika. Menetelmän jatkokäyttömahdollisuudet ovat mielestäni hyvät ja tutkielman tekemisestä saamani syventävä tieto on varmasti korvaamaton apu tulevaisuudessa. Lopullisen kuvan onnistumisen arviointi on pitkälti subjektiivistä, mutta oikea perspektiivi sekä ainakin yksinkertaiset varjot ja heijastukset ovat merkittävä askel kohti visuaalisen realismin tavoittelua.

Lähteet

Adobe 2015a. Noise and Grain effects.

<<https://helpx.adobe.com/after-effects/using/noise-grain-effects.html>> (luettu 19.12.2015)

Adobe 2015b. Time effects.

<<https://helpx.adobe.com/after-effects/using/time-effects.html>> (luettu 19.12.2015)

Blender Foundation 2015a. Integrator.

<<https://www.blender.org/manual/render/cycles/settings/integrator.html>> (luettu 19.12.2015)

Blender Foundation 2015b. Light Paths.

<https://www.blender.org/manual/render/cycles/settings/light_paths.html> (luettu 19.12.2015)

Blender Foundation 2015c. Light Paths.

<https://www.blender.org/manual/render/cycles/settings/light_paths.html> (luettu 19.12.2015)

Coots, Brendan 2007. CreativeCOW.net.

<https://library.creativecow.net/articles/coots_brendan/greenscreen.php> (luettu 19.12.2015)

Dobbert, Tim 2005. Matchmoving: The Invisible Art of Camera Tracking. California: Sybex.

Dobbert, Tim 2012. Matchmoving: The Invisible Art of Camera Tracking (2nd Edition). New Jersey: John Wiley & Sons.

Russell, Eddie 2015. Understanding the Difference between Texture Maps. Digital-Tutors.

<<http://blog.digitaltutors.com/understanding-difference-texture-maps/>> (luettu 19.12.2015)

Wikipedia 2015a. Image-based lighting.

<https://en.wikipedia.org/wiki/Image-based_lighting> (luettu 19.12.2015)

Wikipedia 2015b. Polygonal modeling.

<https://en.wikipedia.org/wiki/Polygonal_modeling> (luettu 19.12.2015)

Wikipedia 2015c. Ray tracing (graphics).

<https://en.wikipedia.org/wiki/Ray_tracing_%28graphics%29> (luettu 19.12.2015)

Wikipedia 2015d. Wire removal.

<https://en.wikipedia.org/wiki/Wire_removal> (luettu 19.12.2015)

Wikipedia 2015e. Chroma key.

<https://en.wikipedia.org/wiki/Chroma_key> (luettu 19.12.2015)

Wikipedia 2015f. Matte (filmmaking).

<https://en.wikipedia.org/wiki/Matte_%28filmmaking%29> (luettu 19.12.2015)

Wikipedia 2015g. Global illumination.

<https://en.wikipedia.org/wiki/Global_illumination> (luettu 19.12.2015)

Wikipedia 2015h. Image-based lighting.

<https://en.wikipedia.org/wiki/Image-based_lighting> (luettu 19.12.2015)

Wikipedia 2015i. High-dynamic-range imaging.

<https://en.wikipedia.org/wiki/High-dynamic-range_imaging> (luettu 19.12.2015)

Wikipedia 2015j. Projective texture mapping.

<https://en.wikipedia.org/wiki/Projective_texture_mapping> (luettu 19.12.2015)

Wikipedia 2015k. Compositing.

<<https://en.wikipedia.org/wiki/Compositing>> (luettu 19.12.2015)

Wikipedia 2015l. Blend modes.

<https://en.wikipedia.org/wiki/Blend_modes> (luettu 19.12.2015)

Wright, Steve 2015. What is Planar Tracking? Imagineer Systems.

<<https://www.imagineersystems.com/videos/what-is-planar-tracking-an-intro-to-mocha-by-steve-wright/>> (luettu 19.12.2015)

Produkti

Oma projekti, Jyrypekka Lehtinen (videotiedosto)

<https://drive.google.com/open?id=0ByrtkLqyq_IVeVpOX1pLaGY1MGs>