

# **Konenäkö informaatiokioskissa**

Erno Mäkinen

Tampereen yliopisto  
Tietojenkäsittelytieteiden laitos  
Pro gradu -tutkielma  
Joulukuu 2001

Tampereen yliopisto

Tietojenkäsittelytieteiden laitos

Erno Mäkinen: Konenäkö informaatiokioskissa

Pro gradu -tutkielma, 49 sivua, 4 liitesivua

Joulukuu 2001

---

Konenäön kyvystä havaita ihminen on tehty useita tutkimuksia viime vuosina. Ihmisen ja tietokoneen välisessä vuorovaikutuksessa konenäköä ei kuitenkaan ole vielä juurikaan tutkittu. Mahdollisuuksia konenäön käytölle on kuitenkin paljon sitten, kun suurimmat aiheeseen liittyvät ongelmat saadaan ratkaistua. Tarkastelen tässä tutkielmassa konenäköä ja sen soveltamista informaatiokioskin käyttöliittymässä. Esittelen informaatiokioskeihin liittyviä ongelmia ja miten konenäköä voidaan käyttää ongelmien ratkaisuun. Tutkielmaan liittyy reaaliaikaisesti toimiva konenäkökomponentti, joka voidaan yhdistää informaatiokioskiin. Komponentin toiminta perustuu siihen, että informaatiokioskin edustasta luodaan malli, jota verrataan havainnoinnin aikana videokameralla otettuihin kuviin. Lisäksi kuvista etsitään ihonvärisiä alueita, joille suoritetaan minimi-maksimi-analyysi. Analyysin perusteella kuvasta löydetään kasvot ja kasvonpiirteistä silmät, nenä ja suu. Kehittämäni konenäkökomponentin toimintaa tutkittiin empiirisillä kokeilla ja kokeiden tuloksia esitellään tässä tutkielmassa.

Avainsanat- ja sanonnat:      Konenäkö, informaatiokioski, minimi-maksimi-analyysi, hahmontunnistustekniikat.

## Sisällys

1.	Johdanto .....	1
2.	Informaatiokioskit .....	3
2.1.	Erilaisia kioskeja.....	3
2.2.	Informaatiokioskien vaatimuksia .....	4
2.3.	Yhteenveto .....	4
3.	Konenäkö ja hahmontunnistus .....	5
3.1.	Konenäön historiaa .....	5
3.2.	Konenäön toiminnan vaiheita.....	7
3.3.	Ohjelmakirjastot konenäkösovelluksille .....	8
3.3.1.	Kirjastoesimerkkejä.....	8
3.3.2.	Konenäkökirjastojen vertailua .....	9
3.4.	Kuvankäsittelytekniikat .....	9
3.4.1.	Aritmeettiset operaatiot .....	10
3.4.2.	Keskiarvosuodatin.....	11
3.4.3.	Alueen löytäminen .....	12
3.5.	Hahmontunnistustekniikat.....	15
3.5.1.	Hahmontunnistus värin perusteella.....	16
3.5.2.	Hahmontunnistus muodon perusteella .....	17
3.5.3.	Hahmontunnistus piirteiden perusteella .....	17
3.5.4.	Hahmontunnistus liikkeen perusteella .....	22
3.6.	Yhteenveto .....	23
4.	Konenäön ja informaatiokioskin yhdistäminen .....	24
4.1.	Yksinkertaisia sovellusmahdollisuuksia .....	24
4.2.	Vaativampia sovellusmahdollisuuksia.....	25
4.3.	Yhteenveto .....	26
5.	Informaatiokioskin toteutus .....	28
5.1.	Käyttöliittymä.....	28
5.2.	Sovelluskehys.....	29

5.3.	Puhuva pää .....	30
5.4.	Kosketuskomponentti .....	31
5.5.	Konenäkökomponentti .....	32
5.5.1.	Toteutusympäristö.....	32
5.5.2.	Konenäkökomponentin rakenne ja toiminta.....	32
5.6.	Yhteenveto .....	34
6.	Konenäön empiiriset tutkimukset.....	35
6.1.	Konenäön nopeus ja kasvojen havaitseminen.....	35
6.1.1.	Konenäön nopeus.....	38
6.1.2.	Konenäön luotettavuus.....	39
6.2.	Kasvojen ja käsien erottaminen toisistaan .....	44
6.3.	Yhteenveto .....	45
7.	Lopuksi .....	46
	Viiteluettelo .....	47
	Liite 1 .....	50
	Liite 2 .....	51

## 1. Johdanto

Näköaisti on ihmiselle hyvin tärkeä väline toisen ihmisen kanssa kommunikoidessa. Kasvot, niiden ilmeet ja kehon erilaiset asennot kertovat keskustelijoille usein enemmän kuin mikään muu kommunikaation muoto. Ihmisen ja tietokoneen välinen vuorovaikutus tapahtuu kuitenkin perinteisesti näppäimistön ja hiiren välityksellä. Miksi siis näköaistia ei nykyään käytetä apuna ihmisen ja tietokoneen välisessä kommunikoinnissa? Tämä johtuu siitä, että konenäkö eli tietokoneen näkökyky ei ole vielä tarpeeksi kehittynyt, jotta sitä voitaisiin hyödyntää monipuolisesti ihmisen ja tietokoneen vuorovaikutuksessa.

Konenäön kyvystä havaita ja havainnoida ihmisiä on tehty useita tutkimuksia viime vuosina. Monissa tutkimuksissa on keskitytty konenäön kykyyn löytää ihminen videokuvasta [Wren *et al.*, 1997]. Toisissa tutkimuksissa taas oletetaan, että ihminen ja kasvot ovat tietyssä kuvan kohdassa ja tämän jälkeen yritetään tulkita ihmisen asentoa tai kasvoilla olevia ilmeitä [Bregler *et al.*, 1998]. Nämä eri tutkimusongelmat täytyisi kuitenkin saada ratkaistua ja yhdistettyä keskenään, jotta konenäön käyttöä ihmisen ja tietokoneen vuorovaikutuksessa päästäisiin kunnolla hyödyntämään.

Ihmisen löytäminen luotettavasti videokuvasta ja hänen toimintansa havainnointi peräkkäisten kuvien perusteella on siis yksi tutkimusongelma. Oma ongelmansa on kasvojen ja kasvonpiirteiden löytyminen sekä ilmeiden tulkinta. Lisäksi, vaikka näissä onnistuttaisiinkin, on jäljellä vielä yksi ongelma: miten tietokone käyttää havaintoja hyödyksi kommunikoidessaan ihmisen kanssa.

Tässä tutkielmassa käsittelen konenäköä ja esittelen kehittämäni konenäkökomponentin, jonka avulla videokuvasta voidaan samaan aikaan havainnoida useampia ihmisiä tai tarkemmin sanottuna heidän kasvojaan ja kasvonpiirteitään. Komponentti tullaan yhdistämään informaatiokioskiin, johon on liitetty myös *puhuva pää* (talking head), joka on kolmiulotteinen malli ihmisen päästä, ja se pystyy luonnollisesti puhumaan ja matkimaan kasvojen eleitä. Yhdistämisen jälkeen kioskin ja puhuvan pään kommunikointia tullaan kehittämään luonnollisemmaksi konenäön avulla. Käsittelen tässä tutkielmassa siten myös informaatiokioskeja ja konenäön käyttömahdollisuuksia informaatiokioskeissa.

Tämä tutkielma on jaettu seitsemään lukuun. Johdannon jälkeisessä toisessa luvussa käsitellään informaatiokioskeja. Informaatiokioskeista annetaan kaksi esimerkkiä, joista toinen on melko tyypillinen kioski ja toinen muistuttaa sitä kioskia, johon tässä tutkielmassa esitelty konenäkökomponentti liitetään. Toisessa luvussa kerrotaan myös informaatiokioskeihin liittyvistä vaatimuksista. Kolmannessa luvussa luodaan katsaus konenäköön, ja erilaisiin konenäkötekniikoihin, jotka liittyvät tämän tutkielman konstruktiviseen osuuteen. Neljännessä luvussa selvitetään, mitä hyötyä on konenäön yhdistä-

misestä informaatiokioskiin ja esitellään erilaisia sovellusmahdollisuuksia konenäölle informaatiokioskeissa. Viidennessä luvussa kerrotaan tarkemmin informaatiokioskista, johon konenäkökomponentti liitetään. Luvussa kerrotaan myös tarkemmin konenäkökomponentin rakenteesta ja toiminnasta. Konenäkökomponentin toimintaa kokeiltiin empiiristen tutkimusten avulla. Näistä kokeista on kerrottu kuudennessa luvussa. Viimeisessä eli seitsemännessä luvussa on tutkielman yhteenveto. Tässä luvussa esitellään myös jatkotutkimuksen aiheita.

## 2. Informaatiokioskit

Borchers *et al.* [1995] määrittelevät *kioskijärjestelmän* (kiosk system) siten, että se on tietokonepäätte, joka on julkisesti kenen tahansa käytettävissä. Heidän mukaansa tällainen järjestelmä on yleensä asennettu tiedon jakamista tai jonkin muun palvelun tarjoamista varten. Tyypillistä kioskijärjestelmälle on, että sillä ei ole tarkkaa käyttäjäryhmää. Sitä myös käytetään yleensä vähän aikaa ja sen takia käyttöliittymän tulisi olla yksinkertainen. Borchers *et al.* [1995] jakavat kioskijärjestelmät neljään ryhmään: informaatiokioskeihin, mainoskioskeihin, palvelukioskeihin ja viihdekioskeihin. Näillä ryhmillä on myös yhteisiä piirteitä.

### 2.1. Erilaisia kioskeja

*Informaatiokioskin* tärkein tehtävä on tarjota tietoa, yleensä tietystä aiheesta. Informaatiokioski voidaan sijoittaa esimerkiksi museoon tai kauppakeskukseen, joissa se tarjoaa tietoja museon vieraille ja kauppakeskuksen asiakkaille. *Mainoskioski* tarjoaa tietoa jostakin yrityksestä tai sen tuotteista, jolloin etenkin kioskin näytävyyys on tärkeää. *Palvelukioskin* avulla käyttäjä voi ostaa esimerkiksi junalipun tai varata hotellihuoneen. Palvelukioski tarjoaa käyttäjälle hyödyllisiä tietoja kuten informaatiokioski, mutta oleellista on myös käyttäjän kioskille syöttämä tieto. *Viihdekioskin* tehtävänä on nimensä mukaisesti viihdyttää käyttäjää. Tällainen kioskki voi sijaita esimerkiksi lentoterminalin odotustiloissa, jolloin lennon odottaminen on miellyttävämpää.

Informaatiokioskeja on kehitetty mitä erilaisimpiin paikkoihin ja tarkoituksiin. Meidän toteuttamaamme kioskia muistuttaa eniten Digital Smart Kiosk -projektin kioskki [Christian and Avery, 1998], jossa informaatio esitetään WWW-sivujen muodossa. Kommunikointia avustamassa on puhuva pää, josta myös Waters *et al.* [1998] kertovat. Kioskin toimintaa on lisäksi ohjaamassa konenäkökomponentti, joka eroaa toiminnaltaan hieman tähän tutkielmaan liittyvästä konenäkökomponentista. Se esimerkiksi havainnoi koko ihmistä ja toimiakseen kunnolla ihmisen tulee olla kokonaan näkyvässä. Christian ja Avery kokeilivat kioskia muun muassa Internet-kahvilassa. Käyttökokeesta kertyi tuhansia tunteja ja yleisesti ottaen käyttäjät pitivät kioskista [Christian and Avery, 2000].

Toinen ehkä tyypillisempi esimerkki informaatiokioskista on MINNELLI-kioskki [Steiger and Suter, 1994], joka on kehitetty Union Bank of Switzerlandissa. Kioskki tarjoaa tietoa pankista ja sen palveluista pankkikonttorissa vieraileville asiakkaille. Erikoista tässä kioskissa ovat piirretyt animaatiot, jotka houkuttelevat kioskin luokse ja tekevät kioskin käytöstä kiinnostavampaa. Kioskin prototyyppiä kokeiltiin yli vuosi, minkä perusteella selvisi muun muassa, että kioskki vähentää aikaa, jota tarvitaan neuvon antamiseen asiakkaille.

## 2.2. Informaatiokioskien vaatimuksia

Informaatiokioskeilla on monenlaisia vaatimuksia. Tähän on monia syitä. Ensinnäkin kioskien käyttäjillä on hyvin monenlaisia taustoja, eikä heitä voida kouluttaa etukäteen kioskin käyttöön. Joukossa voi olla myös käyttäjiä, joilla ei ole lainkaan kokemusta tietokoneiden käytöstä. Toiseksi kioskit voidaan sijoittaa hyvin erilaisiin paikkoihin ja ne voivat tarjota hyvin vaihtelevaa tietoa.

Esimerkiksi Waters *et al.* [1998] kertovat, että julkisissa tiloissa sijaitsevien kioskien tulee aktiivisesti aloittaa ja lopettaa vuorovaikutus käyttäjien kanssa sekä jakaa resursseja tasapuolisesti useiden käyttäjien kesken. Heidän mukaansa kioskin pitäisi myöskin toimia ja kommunikoida siten, että käyttäjä ymmärtää heti ja yksiselitteisesti sen tarkoituksen, koska etukäteiskoulutusta ei voida antaa ja informaatio saattaa olla käyttäjälle hyvinkin tärkeää.

Borchers *et al.* [1995] mukaan informaatiokioskien käyttäjät tarvitsevat tavallisesti tiettyä tietoa ja tällöin on tärkeää, että he löytävät tarvitsemansa tiedon nopeasti. Kioskin tulisi myöskin olla houkutteleva ja pitää käyttäjän kiinnostusta yllä. Toisaalta kioskki ei saa olla liian aktiivinen, jolloin mahdollinen käyttäjä saattaakin jättää kioskin käytön väliin. Näin esimerkiksi kävi aluksi MINNELLI-kioskin kanssa; moni pankin asiakas luuli, että heidän ei tarvitse tehdä mitään, kun kioskin näytöllä pyöri näyttävä animaatio. Näin siitäkin huolimatta, että kioskista kuului ääni, joka kehotti koskemaan näyttöä [Steiger and Suter, 1994].

## 2.3. Yhteenveto

Tässä luvussa määriteltiin informaatiokioskin käsite ja annettiin kaksi esimerkkiä informaatiokioskeista. Myös muista kioskityypeistä kerrottiin lyhyesti. Lopuksi kerrottiin informaatiokioskien vaatimuksista, joita ovat esimerkiksi helppo- ja nopeakäyttöisyys.

Luku ei ollut kovin pitkä, koska keskityin tässä tutkielmassa lähinnä konenäköosuuteen. Lisäksi informaatiokioskeja käsitellään myöhempienkin lukujen yhteydessä.



### 3. Konenäkö ja hahmontunnistus

*Konenäkö* tarkoittaa tietokoneella tai jollakin muulla koneella olevaa näkökykyä. Käytännössä tällaiseen laitteeseen on usein liitetty videokamera tai useampia videokameroita ja laite käsittelee kameralta tulevaa videokuvaa tilanteen vaatimalla tavalla. Yleistä joka tilanteeseen sopivaa konenäkötekniikkaa ei toistaiseksi ole pystytty kehittämään, mutta tietyissä tilanteissa konenäkö on saatu toimimaan hyvin [Shearer *et al.*, 2001]. Tässä tutkimuksessa käsitelen tarkemmin konenäköä, jossa tietokoneeseen on liitetty niin sanottu web-kamera, jonka kaappaamasta videokuvasta tietokone etsii ihmisiä ja heidän tarkempia piirteitään. Hahmontunnistus tai oikeastaan *hahmon havaitseminen* on kyseessä, kun kuvasta yritetään löytää ihmishahmoja. Termiä *hahmontunnistus* käytetään esimerkiksi yhteyksissä, joissa kuvassa on ihmiskasvot ja yritetään selvittää kenelle ne kuuluvat. Toisaalta termin hahmontunnistus alle voidaan kuitenkin sisällyttää sekä hahmon havaitseminen että varsinainen tunnistus. Tässä tutkielmassa on valittu tämä tulkinta.

#### 3.1. Konenäön historiaa

Konenäköä on tutkittu viime vuosina melko paljon ja aivan viime aikoina varsinkin ihmisten havaitsemista ja tunnistamista. Tämä on ymmärrettävää, koska konenäkö tarjoaa paljon erilaisia sovellusmahdollisuuksia ihmisen ja tietokoneen vuorovaikutuksessa. Osa ihmisen havainnointiin liittyvistä konenäkösovelluksista on onnistuttu jo kaupallistamaan. Esimerkkinä vaikkapa valvontakameran kuvista reaaliaikaisesti kasvoja tunnistava ohjelmisto FaceIt [FaceIt, 2001]. Ihmisten havainnointi on kuitenkin niin vaativaa, että tutkittavaa riittää vielä pitkälle tulevaisuuteen. Mihin tahansa sovellukseen sopivaa yleiskäyttöistä ihmisen havainnointimenetelmää pystytään tuskin kehittämään vielä pitkään aikaan.

Konenäön kykyä havaita ihmishahmo on tutkittu monin tavoin. Osa tutkijoista on keskittynyt konenäön kykyyn havaita koko ihminen, esimerkiksi Wren *et al.* [1997] sekä Rosales ja Sclaroff [1998], ja toiset vain kasvojen havainnointiin, esimerkiksi Sobottka ja Pitas [1996]. Osa tutkijoista on tutkinut myös kasvon ilmeiden tunnistamista [Matsuno *et al.*, 1995] ja osa esimerkiksi käsieleiden tunnistamista [Maggioni and Kämmerer, 1998].

Muutama vuosi sitten ihmishahmon havaitsemisesta on tehty myös kaksi hyvää katsausta: ”Human Motion Analysis: A Review” [Aggarwal and Cai, 1997] ja ”The Visual Analysis of Human Movement: A Survey” [Gavrila, 1999]. Gavrila on luokitellut ihmishahmon havaitsemisen tutkimukset kolmeen luokkaan: kaksiulotteiset lähestymistavat, joissa ihmisestä ei luoda tarkkaa mallia (2-D approaches without explicit shape models), kaksiulotteiset lähestymistavat, joissa luodaan malli tarkkailtavasta ihmisestä (2-D approaches with explicit shape models) ja kolmiulotteiset lähestymistavat (3-D approaches). Hänen mukaansa eri lähestymistavat sopivat erilaisiin sovelluksiin. Kaksiulotteiset lähestymistavat ovat hänen mukaansa parhaimmillaan sovelluksissa, joissa tarkkailtavan ihmisen asennontunnistus ei ole tarpeellista tai mahdollista huonon kuvanlaadun takia. Kolmiulotteiset lähestymistavat taas ovat parhaimmillaan sovelluksissa, joissa havainnoitavia ihmisiä voi olla paljon ja he voivat liikkua ilman rajoituksia. Hänen mukaansa kolmiulotteiset lähestymistavat vaativat kuitenkin kunnolla toimiakseen kaksi kameraa ja kolmiulotteiseen havainnointiin perustuvat järjestelmät ovat vielä hyvin rajoittuneita, eikä niitä voida tällä hetkellä yleisesti hyödyntää.

Aggarwal ja Cai lähestyvät asiaa eri suunnasta ja keskittyvät kolmeen alueeseen: liikkeen havainnointi ihmisen kehon osien avulla (motion analysis involving human body parts), ihmisen liikkeen havainnointi käyttämättä kehon osia apuna (tracking of human motion without using body parts) ja ihmisen toiminnan tunnistus kuvasarjoista (recognizing human activities from image sequences). Ensimmäisessä alueessa he keskittyvät kaksiulotteisiin ja kolmiulotteisiin lähestymistapoihin, kuten Gavrila. Toista aluetta he käsittelevät sen mukaan, onko havainnoinnissa käytetty yhtä vai useampaa kameraa. Kolmas alue eli ihmisen toiminnan tunnistus kuvasarjoista on heidän mukaansa korkeamman tason tehtävä kuin kaksi ensimmäistä ja se on mahdollista vain jos ihmisen liikettä onnistutaan havainnoimaan peräkkäisistä kuvista.

Myös kasvojen havaitsemisesta on tehty hyvä katsaus: ”A survey on Face Detection Methods” [Yang *et al.*, 2001]. Yang *et al.* keskittyvät lähinnä kasvojen havaitsemiseen yksittäisistä kuvista, mutta suurinta osaa katsauksessa mainituista tekniikoista voidaan käyttää myös videokuvan analysoinnissa. Katsauksessa on kerrottu muun muassa havaitsemistekniikoista, jotka perustuvat kasvonpiirteisiin, väriin, kaavaimiin ja neuroverkkoihin. Katsauksessa on myös lueteltu erilaisia ongelmia, joihin olisi hyvä keskittyä tulevaisissa tutkimuksissa. Näitä ovat erilaisten havainnointimenetelmien yhdistäminen, valaistusolosuhteiden huomioon ottaminen, kasvojen havaitseminen erilaisissa asennoissa, empiiriset kokeet, jotta menetelmiä voidaan vertailla keskenään ja tietokanta erityisesti kasvojen havaitsemisen tutkimusta varten.

Konenäön soveltamista ihmisen ja tietokoneen välisessä vuorovaikutuksessa on tutkittu jonkin verran. Informaatiokioskit ovat yksi mahdollinen sovellusalue, tosin aiheesta on käsittääkseni aiemmin tehty vain yksi tutkimusprojekti [Christian and Avery, 1998; Waters *et al.*, 1998]. Maes *et al.* [1996] kuvaavat kolmiulotteista ALIVE-virtuaaliympäristöä, jossa ihminen voi kommunikoida eleillä virtuaaliagenttien kanssa. Agentti voi olla esimerkiksi koira, joka istuu, kun ihminen tekee kädellään sopivan eleen. Konenäön avulla selvitetään ihmisen sijainti ja hänen tekemänsä eleet. Sengupta *et al.* [2000] toteuttivat kolme yksinkertaista tietokonepeiliä (esimerkiksi kivi-paperi-sakset) ja kokeilivat miten konenäkö toimii niiden kanssa. Tulokset olivat heidän mukaansa hyviä. Maggioni ja Kämmerer [1998] kuvailevat mahdollisia sovelluksia esimerkiksi lääketieteeseen, kuten kolmiulotteisia magneettikuvia esittävän taulun. Tällaisen taulun avulla voidaan muun muassa kallosta otettua kuvaa tarkkailla eri suunnista siten, että lääkäri liikuttaa päätään esimerkiksi sivulle. Seinään taulun yläpuolelle on asetettu videokamera ja konenäön avulla päätellään, mistä suunnasta kuva pitäisi näyttää.

### 3.2. Konenäön toiminnan vaiheita

Konenäön toiminta voidaan jakaa erilaisiin vaiheisiin. Jotta kameran ottamat kuvat saadaan ohjelmaan, tarvitaan ohjelmakomponentti, joka toimii varsinaisen sovellusohjelman ja kameran välissä. Tällainen *kamerakirjasto* voi esimerkiksi kaapata kuvia videokameralta, minkä jälkeen kuviin päästään käsiksi kirjaston tarjoaman ohjelmointirajapinnan kautta.

Kun kuva on saatu kameralta sovellusohjelmalle, kuvaa tai kameralta saatuja peräkkäisiä kuvia täytyy yleensä muokata monin eri tavoin, jotta niiden sisältämä informaatio saadaan sopivaan muotoon varsinaista hahmontunnistusta varten. Kuvan muokkauksen täytyy usein olla myös nopeaa varsinkin, jos hahmon havaitsemista käytetään reaaliaikajärjestelmässä. Näin on myös vuorovaikutteisen informaatiokioskin tapauksessa.

Varsinainen hahmon havaitseminen ja tunnistus voidaan aloittaa, kun riittävä määrä kuvia on saatu kameralta ohjelmalle ja kuvia on muokattu riittävästi. Joskus yksikin kuva voi riittää, ja toisinaan kuvia voidaan tarvita kymmeniä. Tarvittava kuvien määrä riippuu sovelluksesta ja kuvien laadusta.

Usein hahmon havaitseminen ei ole viimeinen konenäön työvaihe. Monet sovellukset vaativat, että luodaan malli esimerkiksi havainnointitilanteesta, tilanteeseen kuuluvista ihmisistä ja siitä miten he toimivat. Mallin perusteella voidaan päätellä vaikkapa se, mitä ihminen on tekemässä ja mitä hän seuraavaksi todennäköisesti tekee.

### 3.3. Ohjelmakirjastot konenäkösovelluksille

Konenäkösovelluksissa voidaan hyödyntää erilaisia ohjelmakirjastoja. Kirjasto voi esimerkiksi sisältää kuvankäsittelyominaisuuksia, jolloin sovelluksen tekijän ei tarvitse toteuttaa niitä itse. Kamerakirjastolla taas saadaan kaapattua kuvia videokameralta, jonka jälkeen kuvia voidaan muokata konenäkösovelluksessa. Pelkästään konenäkösovelluksiin tarkoitettut kirjastot sisältävät konenäköalgoritmeja, eivätkä ne useinkaan tarjoa apuvälineitä kuvankaappaukseen. Kaikkia edellä mainittuja kirjastoja voidaan mielestäni kutsua yhteisesti *konenäkökirjastoiksi*, koska niitä kaikkia voidaan käyttää konenäkösovelluksissa.

#### 3.3.1. Kirjastoesimerkkejä

Logitech on kehittänyt monipuolisen QuickCam SDK -kamerakirjaston [QuickSDK, 2000], mutta se toimii vain Logitechin omien kameroiden kanssa Microsoft Windows -käyttöjärjestelmässä, mikä on melko suuri rajoite. Kirjastoa voi käyttää yhdessä Microsoftin MFC -kirjaston tai Win API -rajapinnan kanssa. Logitech QuickCam SDK -ohjelmistokehityspaketin avulla yksittäisiä kuvia voi kaapata BMP-muodossa ja videokuvaa AVI-muodossa. Kirjasto ei kuitenkaan tarjoa näille kuvamuodoille muita valmiita operaatioita kuin tallennuksen muistiin tai levyille, sekä joitakin kirjaston asetuksiin vaikuttavia operaatioita: esimerkiksi videokuvan kaappausnopeuden asetuksen.

Microsoftin kehittämä Microsoft Vision SDK -kamerakirjasto [VisSDK, 2001] toimii vain MS Windows -käyttöjärjestelmissä, mutta sitä voi käyttää myös muiden kuin Logitechin kameroiden kanssa. Microsoft Vision SDK mahdollistaa kuvien kaappaamiseen kameralta. Se tukee erilaisia värillisiä RGB- ja YUVA-kuvatyyppejä sekä mustavalkoisia kuvatyyppejä. Kullekin näistä on oma C++-luokkansa. Osan se pystyy esittämään näytöllä, osan kaappaamaan kameralta ja osalle molemmat toiminnot ovat mahdollisia. Kuvien käsittelyyn on myös erilaisia funktiota. Esimerkiksi kuvien pikselien värit pystyy selvittämään luokkien funktioilla. Kirjastolla pystyy myös lukemaan ja tallentamaan kuvia levyille BMP-muodossa. JPEG-, TIFF- ja joidenkin muiden tyyppisten kuvien lukeminen ja tallennus onnistuu lisäkomponenttien avulla.

Microsoft Vision SDK ei kuitenkaan tarjoa kuvankäsittelyalgoritmeja tai muita ”korkeamman tason” palveluita. Sillä ei myöskään pysty asettamaan kameran parametreja, kuten kaappaustarkkuutta tai valotusaikaa. Toisaalta eri kameroilla on erilaisia asetuksia ja niiden asettaminen vaihtelee kamerasta toiseen, mikä hankaloittaa säätömahdollisuuksien lisäämistä mihin tahansa kamerakirjastoon.

Microsoft DirectShow on osa alhaisen tason DirectX-ohjelmointirajapintaperhettä [DirectX, 2001], joka toimii vain MS Windows -käyttöjärjestelmässä. Sen avulla pystyy kehittämään erilaisia videosovelluksia. DirectX-perheen käyttö edellyttää tietämystä esimerkiksi Microsoftin COM-mallista (Component Object Model) [COM, 2000]. Toisaalta esimerkiksi kameran asetuksiin pääsee DirectShow-kirjaston avulla hyvin käsiksi. DirectShow sopiikin hyvin juuri kamerakirjaston tehtävään.

Open Source Computer Vision Library [OpenCV, 2001], lyhyesti OpenCV, on Intelin kehittämä kirjasto, joka tarjoaa erilaisia kuvankäsittelyalgoritmeja. Se on suunniteltu nimenomaan konenäkösovelluksia varten. Se sisältää tuen esimerkiksi *Kalman-suodatimelle* (Kalman filter), *liikekaavaimille* (motion templates), *kuvapyramideille* (Image pyramids), *eigen-objekteille* (Eigen objects) ja *kätketyille Markovin malleille* (Hidden Markov models).

OpenCV-kirjasto on saatavissa sekä Windows- että Linux- käyttöjärjestelmälle ja algoritmit on toteutettu C++-kielellä. Kirjasto on toteutettu osittain assemblerilla ja optimoitu Intelin prosessoreille.

### 3.3.2. Konenäkökirjastojen vertailua

Kaikkien edellä mainittujen kirjastojen käyttö on ilmaista tutkimuskäytössä siitakin huolimatta, että jokainen on kaupallisen yrityksen tuote. Ominaisuuksiltaan kirjastot eroavat toisistaan melko paljon. Logitech QuickCam SDK ja Microsoft Vision SDK ovat ominaisuuksiltaan lähimpänä toisiaan. Ne on tarkoitettu kuvien kaappaamiseen videokameralta. Microsoft Vision SDK tarjoaa kuitenkin useita erilaisia kuvatyyppejä ja funktioita niiden käsittelyyn toisin kuin Logitech QuickCam SDK. Microsoft DirectShow taas on osa Microsoftin DirectX-perhettä ja on tarkoitettu pelkästään liikuvan kuvan kaappaamiseen. Kaikkia edellä mainittuja kirjastoja voidaan käyttää monissa erilaisissa ohjelmissa. OpenCV sopii käytettäväksi nimenomaan konenäkösovelluksissa, koska se tarjoaa hyvin monipuolisesti konenäköalgoritmeja.

Kirjastoja ei siis voida asettaa paremmuusjärjestykseen, koska ne ovat parhaimmillaan erilaisissa tilanteissa. Parhaiten konenäkösovelluksiin sopinevat kuitenkin Microsoft Vision SDK ja OpenCV, koska Vision SDK -ohjelmistokehityspaketin avulla kuvia pystyy helposti kaappaamaan ja OpenCV-kirjaston avulla analysoimaan.

### 3.4. Kuvankäsittelytekniikat

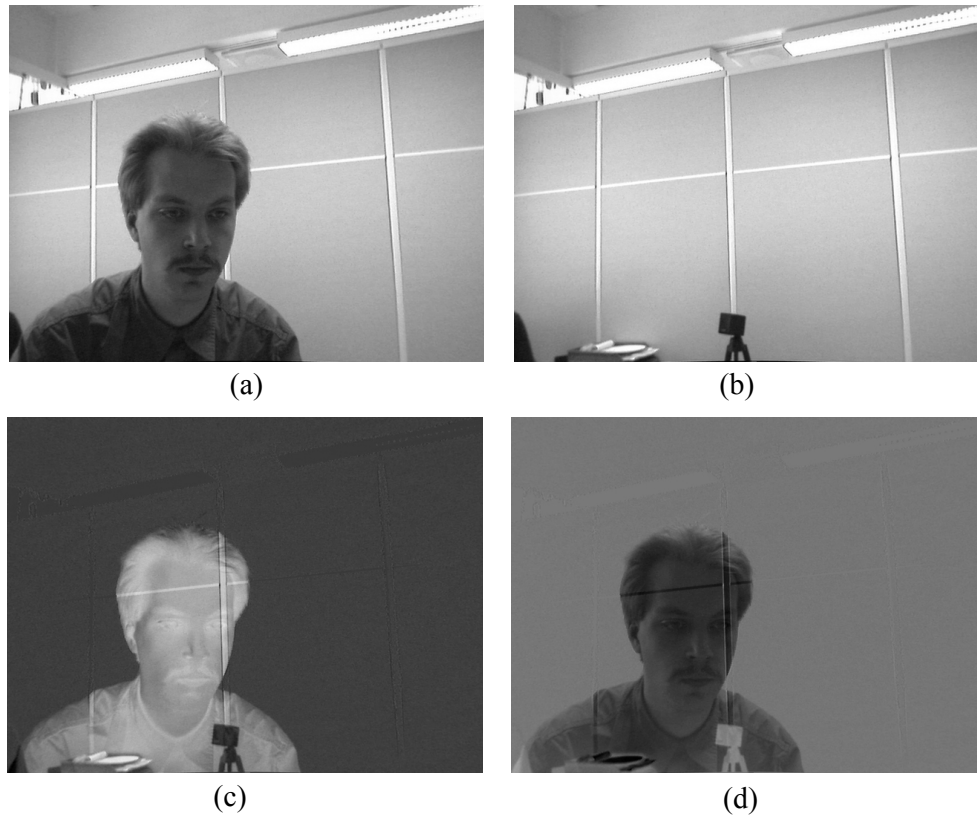
Kuvankäsittelytekniikoita on lukemattomia. Vaikka käsittelisin tässä vain niitä kuvankäsittelytekniikoita, joita usein käytetään konenäkösovelluksissa, olisi aihe silti liian laaja. Aionkin keskittyä niihin tekniikoihin, joita olen käyttänyt hahmontunnistukseen konenäkökomponentin toteutuksessa. Laajemmin aiheesta kiinnostuneen kannattaa tutustua esimerkiksi kirjaan ”Practical Algorithms for Image Analysis” [Seul *et al.*, 2000].

### 3.4.1. Aritmeettiset operaatiot

*Aritmeettisillä operaatioilla* saadaan kaksi kuvaa yhdistettyä yhdeksi uudeksi kuvaksi. Näitä operaatioita ovat esimerkiksi yhteenlasku, vähennyslasku ja jakolasku. Kuvien yhdistäminen tapahtuu soveltamalla jotakin operaatiota tiettyyn pikseliin tai joukkoon pikseleitä kummassakin kuvassa. Otetaan esimerkiksi kaksi kuvaa A ja B. Pikseleiden kirkkaus voi vaihdella välillä 0–255 (0 = musta pikseli ja 255 = valkoinen pikseli). Kuvassa A pikselin, joka sijaitsee koordinaateissa (15, 6) kirkkauden arvo on 145. Kuvassa B vastaavan pikselin kirkkauden arvo on 31. Jos suoritetaan vähennyslasku eli vähennetään vaikkapa B-kuvan pikselin kirkkaus A-kuvan pikselin kirkkaudesta saadaan  $145 - 31 = 114$ . Vähennyslaskun tulos voidaan antaa uuden kuvan koordinaateissa (15,6) sijaitsevan pikselin arvoksi.

Vähennyslaskun tulos voi olla myös negatiivinen. Edellä mainitussa esimerkissä näin olisi käynyt, jos A-kuvan pikselin kirkkaus olisi ollut vaikkapa 31 ja B-kuvan vastaavan pikselin arvo 145. Tällaista negatiivista kirkkauden arvoa ei voida esittää kuvassa. Voidaan kuitenkin sopia, että jos vähennyslaskun tulos olisi negatiivinen annetaan tulokselle arvo 0. Tämä on pienin mahdollinen kirkkauden arvo, joka pystytään esittämään kuvassa. Toinen vaihtoehto on käyttää tuloksen itseisarvoa, joka on aina positiivinen tai 0.

Kuvassa 3.1 on kaksi esimerkkiä vähennyslaskun käytöstä. Harmaasävykuvat 3.1a ja 3.1b on otettu samasta paikasta, mutta toisessa on kasvot ja toisessa vain tausta. Tämän jälkeen on suoritettu vähennyslasku. Kuva 3.1c on saatu vähentämällä kasvot sisältävä kuva taustakuvasta ja kuva 3.1d vähentämällä taustakuva kasvot sisältävästä kuvasta. Esimerkeissä vähennyslasku on tehty erikseen jokaiselle pikselille.



**Kuva 3.1.** Esimerkkejä vähennyslaskun käytöstä.

### 3.4.2. Keskiarvosuodatin

Suodattimia käytetään paitsi yleisessä kuvan käsittelyssä myös hahmontunnistuksessa. Tässä on kerrottu tarkemmin *keskiarvosuodattimesta* (average filter).

Keskiarvosuodatin laskee kullekin harmaasävykuvan pikselille uuden kirkkausarvon tiettyjen viereisten pikselien mukaan. Suodattimen koko voi olla esimerkiksi 3x3 pikseliä, jolloin uuteen kirkkausarvoon vaikuttaa yhteensä 9 pikseliä. Kuva 3.2 selventää asiaa. (a)-kohdassa keskimmäisen pikselin (lihavoitu) arvoa ei ole vielä muutettu. (b)-kohdassa keskimmäisen pikselin arvo on saatu kaikkien yhdeksän pikselin keskiarvosta. Muillekin pikseleille saataisiin arvo vastaavalla tavalla alkuperäisen kuvan pikselien kirkkauksia käyttämällä. Suodatuksessa muodostetaan uusi kuva ja alkuperäiseen ei tehdä muutoksia, koska muuten jo suodatetun pikselin arvo vaikuttaisi seuraavan suodatettavan pikselin arvoon. Toisin sanoen kaikki suodatetut pikselit tallennetaan uuteen kuvaan.

15	43	55
12	<b>34</b>	77
5	46	89

(a)

15	43	55
12	<b>42</b>	77
5	46	89

(b)

**Kuva 3.2.** Kooltaan 3x3 keskiarvosuodattimen käyttö. (a)-kohdassa keskimmäisen pikselin arvo ennen suodatusta ja (b)-kohdassa suodatuksen jälkeen.

### 3.4.3. Alueen löytäminen

Ihmisen on usein helppo löytää kuvasta yhtenäisiä alueita. Tietokoneelle asia ei kuitenkaan ole yhtä yksinkertainen, koska se ”näkee” kuvan vain vierekkäisinä pikseleinä, joille on määritelty tietty valoisuus ja mahdollisesti väri. Tietokone voi etsiä alueita kuvasta kahdella eri perusstrategialla [Seul *et al.*, 2000]. Alue voidaan löytää etsimällä sen reunat tai sitten vertaamalla kuvan lähekkäisiä pikseleitä keskenään ja merkitsemällä samaan ryhmään kuuluviksi ne pikselit, jotka täyttävät tietyt kriteerit. Kriteereitä voivat olla esimerkiksi välimatka pikselistä toiseen, pikselien kirkkaus ja värisävy. Olipa alueen etsinnässä käytetty tekniikka sitten alueen reunojen etsiminen tai alueen vierekkäisten pikselien vertaaminen keskenään, voidaan käyttää kummallekin yhteistä termiä *alueen löytäminen* (region detection).

Olen käyttänyt yleisesti tunnettua *yhtenäisen alueen nimeämisalgoritmia* (connected component labeling) [Dillencourt *et al.*, 1992] [Seul *et al.*, 2000] [Suzuki *et al.*, 2000], joka vertaa kuvan lähekkäisiä pikseleitä eli *naapureita* keskenään. Algoritmia käytetään yleensä kuviin, jotka on ensin *binärisoitu* (binarized), tosin itse olen ohittanut binärisointi vaiheen kokonaan. Binärisointi tarkoittaa sitä, että pikselille annetaan joko arvo päällä (ON) tai pois (OFF). Voidaan esimerkiksi sopia, että ON-pikseli tarkoittaa kuvassa mustaa pikseliä ja OFF-pikseli valkoista.

Kuvassa 3.3 on binärisoitu kuva, joka sisältää kolme ON-aluetta. Yhtenäisen alueen nimeämisalgoritmi löytäisi siis kolme aluetta ja antaisi kullekin oman nimen. Alla tarkempi algoritmin kuvaus.



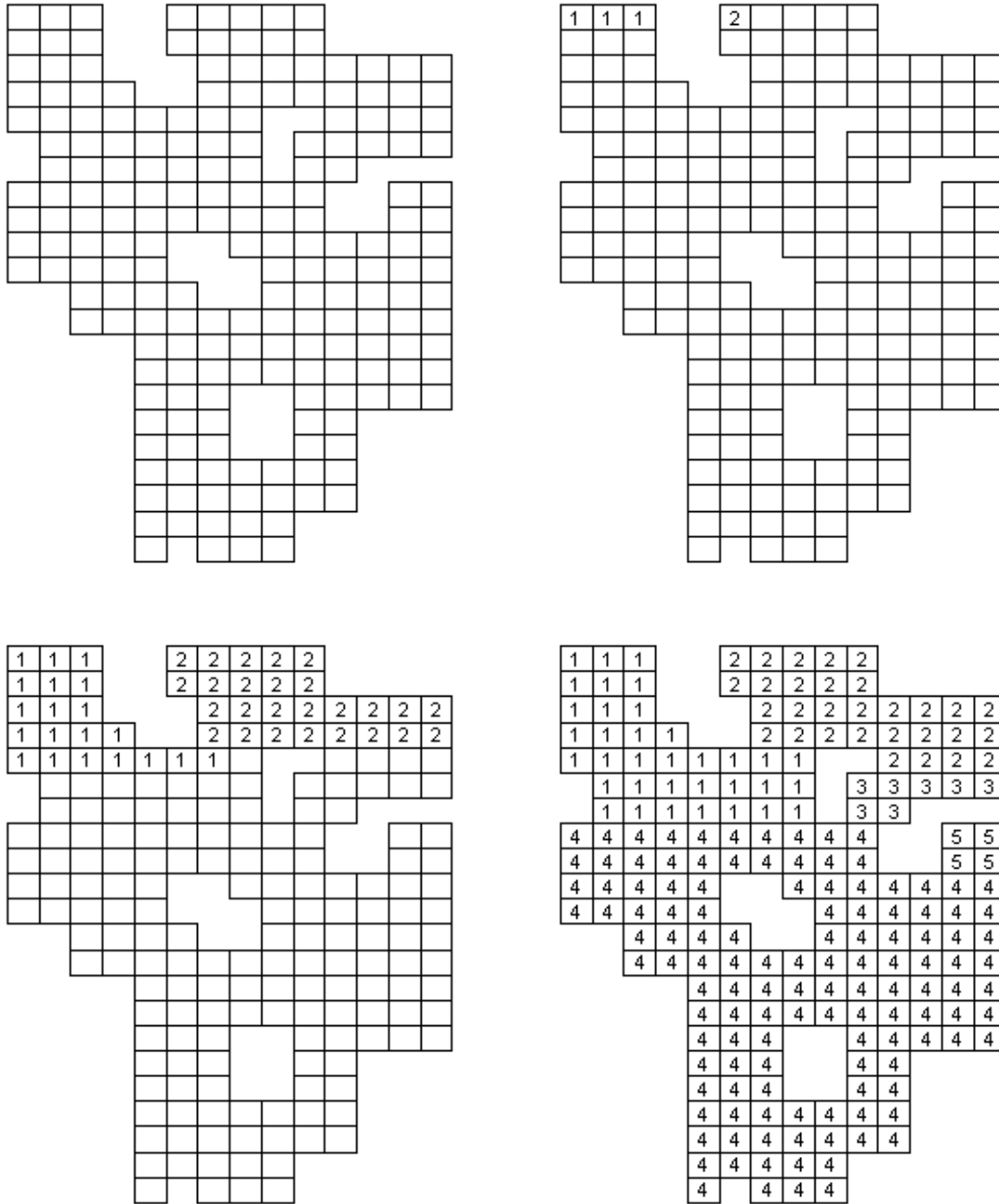


**Kuva 3.3.** Binärisoitu kuva.

1. Lähdetään liikkeelle kuvan vasemmasta yläreunasta. Tutkitaan kuva pikseli kerrallaan, kunnes saavutaan kuvan oikeaan alakulmaan.
2. Tutkitaan kunkin ON-pikselin lähin yläpuolella ja vasemmalla sijaitseva naapuri (jos sellainen on). Mikäli kummallekaan naapurille ei ole annettu nimeä, annetaan tutkittavalle pikselille uusi nimi. Mikäli vain toiselle naapurille on annettu nimi, annetaan tutkittavalle pikselille naapurin nimi. Mikäli kummallakin naapurilla on jo nimi ja nimet ovat samoja, annetaan tutkittavalle pikselille sama nimi kuin naapureilla. Mikäli kummallakin naapurilla on jo nimi, mutta ne eroavat toisistaan, annetaan tutkittavalle pikselille toisen naapurin nimi ja merkitään kumpikin nimi *yhtäsuuruusluokkaan* (equivalence class).
3. Käydään kuva toisen kerran läpi. Tällä kierroksella tutkitaan kunkin pikselin kohdalla, onko sen nimi yhtäsuuruusluokassa ja jos on, annetaan sille yhteinen nimi muiden samaan luokkaan kuuluvien pikseleiden kanssa.

Myös yhtäsuuruusluokan ylläpitämiseen on erilaisia algoritmeja. Yksi nopeimmista ellei nopein, UNION-FIND-algoritmi puu-rakenteella, on esitelty esimerkiksi kirjassa Introduction to Algorithms [Cormen *et al.*, 1989] ja algoritmia käytetään myös tässä tutkielmassa esitetyn informaatiokioskin tapauksessa. Dillencourt *et al.* mukaan mikä tahansa joukko UNION- ja FIND-operaatioita voidaan suorittaa ajassa  $O(m \alpha(m))$ , jossa  $\alpha$  on Ackermanin funktion käänteisfunktio ja kasvaa erittäin hitaasti. Heidän mukaansa lähes kaikissa käytännön tapauksissa  $\alpha(m) \leq 5$ .

Kuvassa 3.4 on esitelty yhtenäisen alueen nimeämisalgoritmin toimintaa käytännössä. Yhtenäinen alue (ON-pikselit) on esitetty ruutuina. Yksi ruutu tarkoittaa yhtä pikseliä (myös ruutujen välissä olevat tyhjät alueet sisältävät pikseleitä). Kun pikselille annetaan nimi, kyseinen nimi näkyy myös ruudussa. A-kohdassa yhtäkään pikseliä ei ole vielä tutkittu, joten ruuduissa ei näy nimiä.



**Kuva 3.4.** Yhtenäisen alueen nimeämisalgoritmin toiminnan vaiheita.

B-kohdassa on tutkittu kuusi pikseliä. Vasemman yläkulman pikselille on annettu nimi yksi, koska sen yläpuolella ja vasemmalla ei ole naapureita ja yhtään nimiä ei ole vielä annettu. Mikään ei estäisi valitsemasta jotakin toista nimeämiskäytäntöä. Toisen pikselin vasemmalla puolella on naapuri, joten sille annetaan sama nimi kuin naapurilla. Samoin tehdään kolmannelle pikselille. Neljäs ja viides pikseli eivät kuulu alueeseen (eivät ole ON-pikseleitä), joten niille ei anneta nimiä. Kuudennen pikselin yläpuolella ja vasemmalla puolella ei ole naapuria, joten sille annetaan uusi nimi.

C-kohdassa on törmätty pikseliin, jonka yläpuolisella ja vasemmanpuolisella naapurilla on eri nimet. Nyt tutkittavalle pikselille annetaan vasemman puoleisen pikselin nimi (mielivaltainen päätös). Nimet yksi ja kaksi täytyy kuitenkin lisätä yhtäsuuruusluokkaan, koska nimet vastaavat toisiaan (pikselit kuuluvat samaan alueeseen).

D-kohdassa on tutkittu koko kuva, kaikille pikseleille on annettu nimet ja kaikki nimet on lisätty samaan yhtäsuuruusluokkaan. Koko kuva käydään tämän nimeämisen jälkeen vielä toisen kerran läpi. Tällä kertaa annetaan sama nimi kaikille niille pikseleille, jotka kuuluvat samaan yhtäsuuruusluokkaan.

Tässä esitelty yhtenäisen alueen nimeämisalgoritmi ei ole ainoa mahdollinen. Alueet voitaisiin etsiä myös esimerkiksi rekursiivisesti tai rinnakkaisina prosesseina, mutta informaatiokioskin tapauksessa voidaan hyödyntää vain yhtä prosessoria (ja tietokonetta), joten rinnakkaisprosessoinnista ei ole hyötyä. Rekursio taas ei ole kovin nopea menetelmä.

### 3.5. Hahmontunnistustekniikat

Ihmisiä voidaan etsiä kuvasta sen jälkeen, kun taustaan kuulumattomat alueet on löydetty. Tähän käytettyjä hahmontunnistustekniikoita on erittäin runsaasti. Useat niistä kuitenkin toimivat liian hitaasti, jotta niitä voisi hyödyntää vuorovaikutteisen informaatiokioskin kanssa. Toisten tekniikoiden kohdalla taas on paljon vaatimuksia havainnointiympäristön tai jonkin muun asian suhteen. Tyypillisiä vaatimuksia ovat esimerkiksi seuraavat:

- tausta ei muutu,
- valaistusolosuhteet eivät muutu,
- kuvassa on vain yksi henkilö, ja
- ihminen on kokonaan näkyvissä.

Vaatimukset ovat ymmärrettäviä, koska usein niistä ei ole haittaa sovelluksen kannalta ja niiden täyttäminen tekee hahmontunnistuksesta huomattavasti helpompaa. Esimerkiksi yleistä hahmontunnistusalgoritmia, joka sallisi taustan tai valaistusolosuhteiden radikaalin muuttumisen, ei ole vielä pystytty kehittämään. Jos tausta voisi muuttua eli esimerkiksi kamera voisi liikkua, lisääntyisi analysoitavan informaation määrä todella paljon. Valaisen asiaa esimerkillä. Mikäli taustasta otettua kuvaa verrataan kuvaan, jossa on kasvat, jotka vievät vaikkapa 30 prosenttia kuva-alasta (usein kasvat vievät huomattavasti tätä vähemmän tilaa), saadaan aritmeettisen vähennyslaskun avulla loput kuva-alasta tiputettua pois ennen tarkempaa analyysiä. Koska aritmeettinen vähennyslasku on melko nopea toimenpide, voidaan ajallisesti saavuttaa hyvinkin suuri säästö, kun tarkempi analysointi täytyy suorittaa vain pienelle osalle kuvaa.

### 3.5.1. Hahmontunnistus värin perusteella

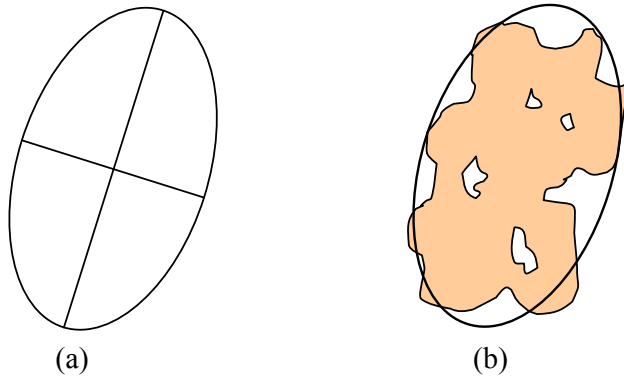
Väri on yksi parhaista keinoista ihmisen havaitsemiseen [Christian and Avery, 1998; Jones and Rehg, 2000; Sobottka and Pitas, 1996]. Tämä johtuu esimerkiksi siitä, että ihon *värisävy* (hue) ei juurikaan vaihtelee ihmisestä toiseen [Jones and Rehg, 2000]. Sen sijaan *värikylläisyys* (saturation) vaihtelee ihon pigmentin mukaan, joten jos sitä käytetään ihmisen havaitsemisessa apuna, on vaihtelu otettava huomioon. Eniten havaitsemista kuitenkin hankaloittavat vaihtelevat valaistuosuhteet, jotka vaikuttavat ihon *kirkkauteen* (brightness, intensity) kuvassa ja sitä kautta sekä värisävyyttä että värikylläisyyteen.

Jones ja Rehg [2000] ovat tehneet mielenkiintoisen tutkimuksen ihonväristä. He keräsivät Internetistä noin kolmelta miljoonalta sivulta yhteensä 18696 kuvaa, joiden avulla he loivat kartan ihmisen ihon väreistä. Kartan avulla he kehittivät luokittelijan, joka kykenee havaitsemaan ihonvärisistä alueista noin 80 prosenttia, kun vääriä positiivisia havaintoja samaan aikaan on 8,5 prosenttia.

Ihonvärin käyttöä erityisesti kasvojen havaitsemisessa on tutkittu melko paljon. Esimerkiksi Cho *et al.* [2001] esittelevät menetelmän, jolla ihonväriset alueet saadaan suodatettua kuvasta, jossa on muita lähes ihonvärisiä alueita. Se myös mukautuu eri kuvien välillä tapahtuviin muutoksiin, jotka tavallisissa menetelmissä aiheuttavat epätarkkuutta ihonväristen alueiden havaitsemisessa. Menetelmässä on heidän mukaansa kuitenkin myös heikkoja puolia. Ensinnäkin ihonvärisen alueen pitää olla suurempi kuin tausta-alueen, jonka väri on lähellä ihonväriä, jotta menetelmä toimisi kunnolla. Toiseksi menetelmää voi kyllä käyttää kaikkien ihotyyppeiden suodattamiseen, mutta vain yhteen kerrallaan. Nämä kaksi seikkaa tekevät menetelmän soveltamisen informaatiokioskissa lähes mahdottomaksi. Informaatiokioskin tapauksessa ei voida olettaa, että ihonvärinen alue olisi hallitseva videokuvassa tai, että kioskia käyttäisivät vain esimerkiksi valkoihoiset ihmiset.

### 3.5.2. Hahmontunnistus muodon perusteella

Muotoa voidaan käyttää esimerkiksi pään havaitsemiseen. Pää on melko lailla ellipsin muotoinen olipa se sitten missä asennossa tahansa. Sobottka ja Pitas [1996] ovat kuvanneet tähän tietoon perustuvan kasvojen havaitsemistekniikan, jossa tietyille kuvan alueille muodostetaan niitä parhaiten vastaava ellipsi. Ellipsi muodostetaan siten, että tutkittavalle alueelle lasketaan pienin ja suurin momentti. Näistä saadaan laskettua ellipsin isoakseli ja pikkuakseli, jotka kulkevat ellipsin keskipisteen kautta (ks. kuva 3.5a), sekä ellipsin kallistuskulma. Kun ellipsi on muodostettu, lasketaan, kuinka moni tutkittavan alueen pikseleistä on ellipsin ulkopuolella ja kuinka paljon ellipsin sisällä on pikseleitä, jotka eivät kuulu alueeseen (ks. kuva 3.5b). Laskettu arvo kuvaa kuinka hyvin alue vastaa ellipsiä. Riittävän lupaaville alueille voidaan sitten tehdä jatkoanalyysi esimerkiksi etsimällä tiettyjä kasvopiirteitä.



**Kuva 3.5.** (a) Aluetta parhaiten vastaava ellipsi ja ellipsin akselit. (b) Alue ja sitä parhaiten vastaava ellipsi.

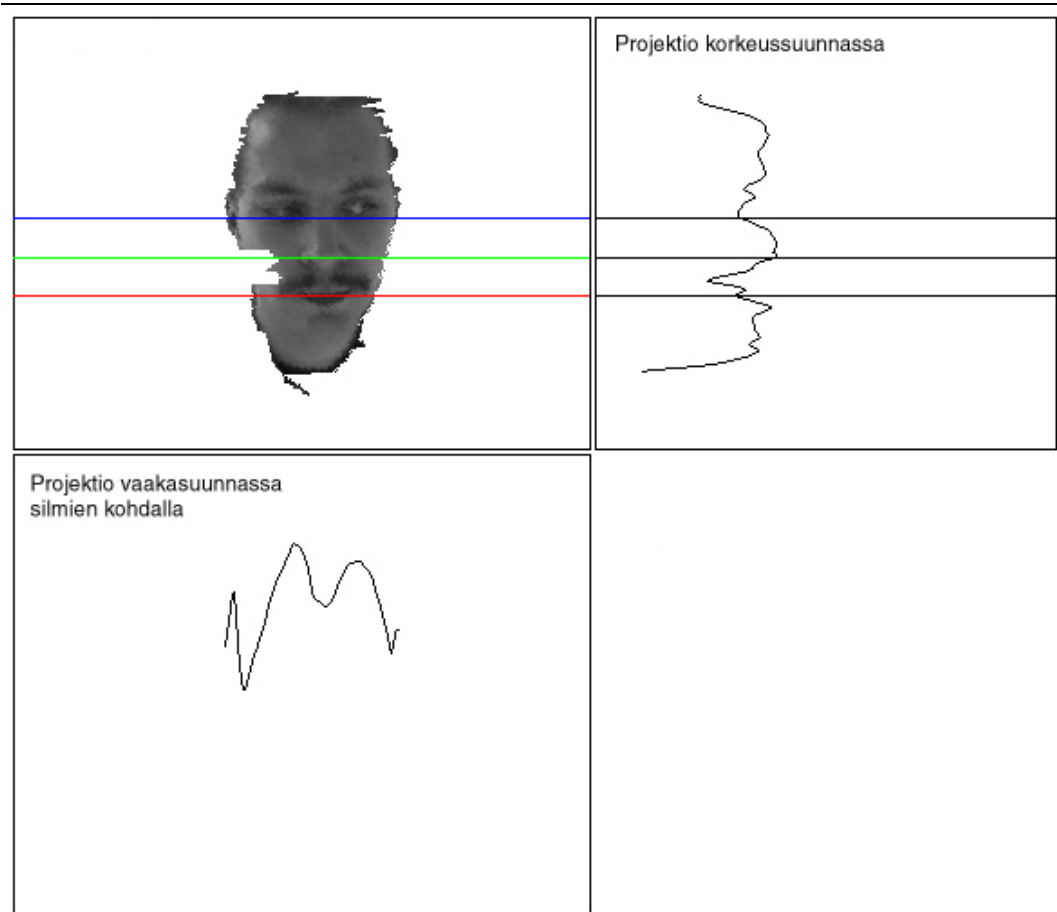
### 3.5.3. Hahmontunnistus piirteiden perusteella

Edellisessä alakohdassa mainituista alueista voidaan etsiä silmät, nenä ja suu. Sobottka ja Pitas [1996] ovat tutkineet kahta eri tapaa kasvojen piirteiden havaitsemiseen: *valuma-aluealgoritmia* (watershed algorithm) ja *minimi-maksimi-analyysiä* (min-max analysis). Käsittelen tässä tarkemmin minimi-maksimi-analyysiä, koska olen käyttänyt sitä konenäkökomponentin toteutuksessa.

Minimi-maksimi-analyysi perustuu sille, että tietylle kuvan alueelle muodostetaan projektio pystysuunnassa. Toisin sanoen kullekin alueen riville lasketaan kirkkauden keskiarvo. Rivien keskiarvoista muodostuu projektio pystysuunnassa. Projektiota voidaan vielä tasoittaa keskiarvosuodattimella. Sobottka ja Pitas ovat käyttäneet 1x3-kokoista suodatinta, mutta suurempaa suodatinta kannattaisi ehkä käyttää, mikäli tutkittava alue on suuri tai vastaavasti pienempää suodatinta, jos alue on pieni. En ole kuitenkaan tätä asiaa tarkemmin tutkinut, vaan olen käyttänyt informaatiokioskin kanssa samaa suodattimen kokoa kuin he.

Kasvoissa on silmien ja suun kohdalla keskimääräistä tummempia kohtia. Ne näkyvät pystysuuntaisessa projektiossa yleensä melko hyvin, mutta vastaavia kohtia voivat aiheuttaa esimerkiksi viikset, parta, hiukset ja silmälasit. Kasvot saattavat olla kuvassa myös kallistuneet, jolloin kuvan alue, jolla mahdollinen kasvo sijaitsee, on ensin käännettävä pystysuoraan. Kääntäminen onnistuu aluetta parhaiten vastaavan ellipsin kallistuskulman avulla. Kun rivit, joilla silmät tai suu saattavat sijaita, on löydetty, muodostetaan riveille vaakasuuntainen projektio, joka on tasoitettu pystysuunnassa 3x1-kokoisella keskiarvosuodattimella.

Kuvassa 3.6 näkyy kasvot ja kasvoista muodostettu pystysuuntainen projektio, joka on tasoitettu keskiarvosuodattimella. Kuvassa näkyy lisäksi projektio, joka on muodostettu vaakasuunnassa silmien kohdalta. Kasvoissa kyseiselle riville on piirretty sininen viiva, joka jatkuu vastaavassa kohdassa pystysuuntaisessa projektiossa. Myös nenälle ja suulle on vastaavat viivat, mutta niiden vaakasuuntaista projektioita ei ole näkyvissä.



**Kuva 3.6.** Kasvot ja siitä muodostettuja projektioita.

Kuvan projektiot ovat hyvin tyypillisiä kasvoille. Samanlaisia projektioita ovat kokeissaan saaneet muodostettua myös Sobottka ja Pitas. Pystysuuntaisessa projektiossa silmien, suun ja mahdollisten viiksien kohdalla on tummempi kohta, joka näkyy myös kuvan projektiossa. Tässä kuvassa myös kulmakarvojen kohdalla projektiossa näkyy notkahdus. Nenän kohdalla taas on keskimääräistä kirkkaampi kohta. Vaakasuuntaisessa projektiossa silmien kohdalla on tummemmat kohdat, jotka tässä esimerkissä näkyvät hyvin selvästi. Monesti tummia kohtia on kuitenkin useampia esimerkiksi hiusten takia. Projektioden paikallisia minimi- ja maksimikohtia käytetään kasvojen piirteiden havaitsemisessa, joten kaikki ”ylimääräiset” paikalliset minimi- ja maksimit hankaloittavat haluttujen piirteiden löytämistä.

Sobottka ja Pitas ovat käyttäneet seuraavia ehtoja silmien etsimiseen:

- silmien tulee sijaita kasvojen ylä- tai keskiosassa,
- vaakasuuntaisessa projektiossa kummallekin silmälle on merkittävä paikallinen minimi,
- minimeillä on samankaltaiset kirkkaudet,
- kahden minimin välissä on merkittävä paikallinen maksimi, ja
- silmien sijainti suhteessa pään leveyteen on tietyllä välillä.

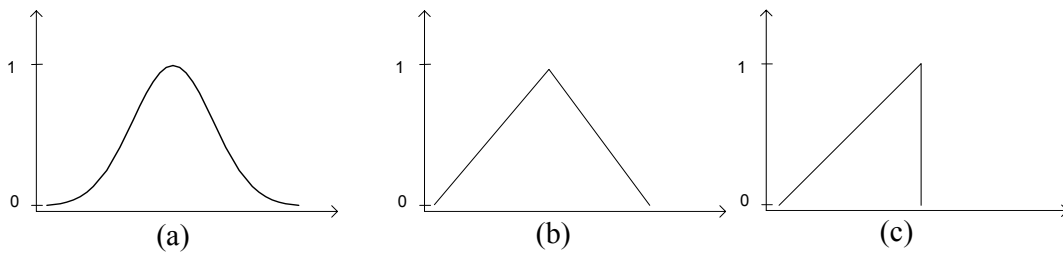
Suun havaitsemiseen he ovat käyttäneet seuraavia ehtoja:

- suu sijaitsee pään keski- tai alaosassa,
- vaakasuuntaisessa projektiossa on kaksi merkittävää paikallista maksimia, eikä suurempaa paikallista maksimia ole niiden välissä,
- merkittävä paikallinen minimi on kahden paikallisen maksimin välissä, ja
- suun leveys suhteessa pään leveyteen on tietyllä välillä.

Lisäksi he ovat käyttäneet vielä seuraavia ehtoja:

- silmien ja suun suhde päähän pystysuunnassa on tietyllä välillä, ja
- suu sijaitsee vaakasuunnassa silmien välissä.

Ehdot ovat sumeaan logiikkaan pohjautuvia eli kullakin ehdolla on niin sanottu *jäsenyysfunktio*. Jäsenyysfunktio kertoo, missä määrin jokin asia kuuluu tiettyyn sumeaan joukkoon: esimerkiksi missä määrin silmäehdokas kuuluu silmien joukkoon. Jäsenyysfunktio saa arvoja nolasta yhteen. Arvo nolla tarkoittaa, että silmäehdokas ei kuulu lainkaan silmien joukkoon ja yksi, että ehdokas kuuluu täysin silmien joukkoon. Jäsenyysfunktioilla on erilaisia kuvaajia. Kuvassa 3.7 on joitakin esimerkkejä jäsenyysfunktioiden kuvaajista. Kuvassa 3.7a kuvaaja on Gaussin käyrän muotoinen. Käyrän keskellä jäsenyyden arvo on yksi ja keskustasta poispäin mentäessä se lähestyy nollaa. Kuvan 3.7b kuvaaja ja kuvan 3.7c kuvaaja ovat *kolmiofunktion* kuvaajia. Tarkemmin sumeaan logiikkaan voi tutustua esimerkiksi kirjan Sumea Logiikka [Kosko, 2001] avulla.



**Kuva 3.7.** Jäsenyysfunktioiden kuvaajia.

Kasvonpiirteiden ehtoihin liittyvien jäsenyysfunktioiden perusteella lasketaan, miten hyvin silmä- ja suuehdokkaat toteuttavat ne. Näillä ehdoilla ja sopivilla jäsenyysfunktioiden määrittelyillä Sobottka ja Pitas kokeilivat menetelmän toimintaa kuvasarjan kohdalla, joka koostui 150 peräkkäisestä kuvasta. Kuvat oli ilmeisesti otettu uutistenlukutilanteesta. He onnistuivat löytämään silmät, suun tai molemmat 86 prosentissa kuvista. Niissä tapauksissa, missä silmät löytyivät, oli 98 prosenttia oikeita havaintoja. Suun tapauksessa 86 prosenttia oli oikeita havaintoja.

Olen käyttänyt monia edellä lueteltuja ehtoja kasvonpiirteiden havaitsemisessa, mutta silmien ja suun lisäksi olen etsinyt myös nenäehdokkaita. Myös muita eroavaisuuksia löytyy, kuten se, että en ole käyttänyt ellipsin epäkeskisyyttä kasvojen havaitsemisessa.

Ehtojen tulee olla hyvin valittu, jotta ne vastaisivat tyypillisiä kasvoja ja niiden perusteella löydettäisiin oikeat kasvonpiirteet. Toisaalta liian tiukat ehdot pudottavat oikeita ehdokkaita pois, koska yhdenkään ihmisen kasvot eivät vastaa täysin ”standardikasvoille” laskettuja mittoja. Kasvojen piirteiden kanssa käyttämäni suhteet ja niihin perustuvat ehdot pohjautuvat kasvojen *antropometriaan* [Farkas, 1994]. Antropometria on tieteen alue, jolla tutkitaan ihmiskehon mittoja ja tässä tapauksessa kasvonpiirteiden mittoja. Seuraavaksi tarkemmin kehittämistäni ehdoista.



Jotta tietyllä tutkittavan alueen rivillä voisi olla ehdokas silmille, suulle tai nenälle, täytyy sen täyttää tietyt ehdot. Silmillä on seuraavat ehdot:

- Rivin, jolla silmät sijaitsevat täytyy sijaita tutkittavan alueen puolivälissä tai sen yläpuolella.
- Rivin täytyy olla paikallinen minimi pystysuuntaisessa profiilissa ja sen kirkkauden täytyy olla profiilin keskiarvoa pienempi. Myös tällaisen minimin viereinen rivi sopii ehdokkaaksi.
- Vaakasuuntaisessa profiilissa täytyy olla ainakin kaksi paikallista minimiä (toinen vasemmalle ja toinen oikealle silmälle) ja näiden välissä ainakin yksi paikallinen maksimi.
- Silmäehdokasrivin paras paikallinen maksimi sijaitsee vaakasuunnassa mahdollisimman keskellä.
- Parhaat paikalliset minimit ovat ne, jotka ovat lähimmäksi silmän leveyden verran vaakasuunnassa alueen reunoista ja parhaasta paikallisesta maksimista. Silmien oletetaan sijaitsevan vaakasuunnassa näiden minimien kohdalla.

Nenällä (nenänpäällä) on seuraavat ehdot:

- Rivi sijaitsee pystysuunnassa alueella, joka on yli  $\frac{1}{4}$  korkeudesta ja alle  $\frac{3}{4}$  korkeudesta.
- Rivi on paikallinen maksimi pystysuuntaisessa profiilissa ja sen kirkkaus on keskiarvoa suurempi. Myös tällaisen rivin viereinen rivi sopii ehdokkaaksi.

Suulla on seuraavat ehdot:

- Rivin täytyy sijaita tutkittavan alueen puolivälin alapuolella.
- Rivin täytyy olla paikallinen minimi pystysuuntaisessa profiilissa ja sen kirkkauden täytyy olla profiilin keskiarvoa pienempi. Myös tällaisen minimin viereinen rivi käy ehdokkaaksi.
- Vaakasuuntaisessa profiilissa täytyy olla ainakin kaksi paikallista minimiä ja näiden kirkkauden tulee olla keskiarvon alapuolella.

Kun ehdokkaat on löydetty, muodostetaan silmä-, nenä-, ja suuehdokkaista kolmikoita ja lasketaan seuraavat jäsenyydet:

- silmien (kaksi paikallista minimiä vaakasuuntaisessa profiilissa) etäisyys toisistaan,
- silmien etäisyydet alueen reunoista vaakasuunnassa,
- silmien etäisyys vaakasuunnassa näiden välisestä maksimista,
- silmien sijainti pystysuunnassa,

- nenän sijainti pystysuunnassa,
- suun leveys (kahden paikallisen minimin välinen etäisyys toisistaan),
- suun sijainti pystysuunnassa,
- silmien ja nenän sijainti toisiinsa nähden pystysuunnassa,
- nenän ja suun sijainti toisiinsa nähden pystysuunnassa,
- silmien ja suun sijainti toisiinsa nähden vaakasuunnassa, ja
- silmien, nenän ja suun sijainti toisiinsa nähden pystysuunnassa.

Näiden jäsenyyksien perusteella lasketaan tietyillä painotuksilla todennäköisyys, että kyseessä on kasvot. Suurimman todennäköisyyden saanut kolmikko kertoo myös sen, missä kohtaa kasvoja silmät, nenä ja suu tällöin sijaitsevat.

Edellä kuvatut kasvonpiirteisiin perustuvat ehdot toimivat parhaiten, kun kasvot on kuvattu edestä päin. Kasvoista tulisi käytännössä olla kolmiulotteinen malli, jotta ehdot toimisivat hyvin silloinkin, kun kasvot on kuvattu mielivaltaisesta suunnasta. Kuten aiemmin mainittiin, kolmiulotteisten mallien käytössä on kuitenkin vielä suuria ongelmia. Lisäksi kioskien tapauksessa nykyiset ehdot ovat riittäviä, koska käyttäjä katsoo yleensä kioskiin päin, ja tällöin ehdot toimivat hyvin, kuten empiirisissä kokeissa olen todennut.

#### **3.5.4. Hahmontunnistus liikkeen perusteella**

Liikettä on käytetty apuna monissa hahmontunnistukseen liittyvissä tutkimuksissa. Ihmisen toiminta voidaan tunnistaa seuraamalla hänen liikkeitään kuvasta toiseen. Näin esimerkiksi urheilijan tai tanssijan toimintaa voidaan tutkia suorituksesta tallennetun videon perusteella tai lääkäri voi suorittaa kliinisiä tutkimuksia ortopedia-potilaalle. Myös käsieleiden tunnistus voi tapahtua liikkeen avulla. Tutustumalla aiemmin mainittuihin Gavrilan [1998] sekä Aggarwalin ja Cain [1997] katsauksiin saa aiheesta hyvän kuvan.

Liikkuvien kohteiden havaitsemisessa aritmeettinen vähennyslaskuoperaatio on kätevä. Jos esimerkiksi tietyssä huoneessa tapahtuvaa liikettä halutaan tarkkailla videokameran avulla, voidaan tilasta ottaa aluksi kuva, jossa ei ole ihmisiä, ja tämän jälkeen verrata videokuvaa aluksi otettuun kuvaan. Mikäli kuvia verratessa tietyllä alueella havaitaan riittävän suuri ero kirkkauksien välillä, voidaan olettaa, että ko. kohtaan on ilmestynyt jotakin alkuperäiseen kuvaan kuulumatonta. Myös peräkkäisiä videokuvia voidaan verrata ja yrittää löytää eroja niiden välillä. Jatkoanalyysillä voidaan sitten yrittää selvittää, mitä kyseisessä kuvan kohdassa todella on.

Edellä mainittua vähennyslaskumenetelmää käytän myös informaatiokioskin kanssa. Paikalle tulevat potentiaaliset käyttäjät havaitaan vertaamalla videokuvaa aluksi taustamalliin. Taustamalli on luotu ottamalla aluksi useampi peräkkäinen videokuva tyhjästä tilasta informaatiokioskin edessä, laskemalla kuvien pikselien värielementeille keskiarvot ja ottamalla lasketut keskiarvot talteen. Taustamallia myös päivitetään tarkkailun aikana, koska esimerkiksi valaistusolosuhteet voivat muuttua päivän mittaan. Kovin suuria muutoksia pienen ajan sisällä malliin ei kuitenkaan tehdä. Mikäli tarkkailun aikana kaapatun videokuvan jotkin alueet eivät sitten sovi taustamalliin, tehdään alueille jatkoanalyysi.

### 3.6. Yhteenveto

Tässä luvussa määrittelin termin konenäkö ja joitakin muita aihealueeseen liittyviä termejä. Loin lyhyen katsauksen siihen, mitä konenäön alueella on tähän mennessä tutkittu ja miten konenäköä on hyödynnetty ja voitaisiin hyödyntää ihmisen ja tietokoneen vuorovaikutuksessa.

Myös joitakin konenäkökirjastoja käsittelin lyhyesti. Kirjastot eroavat toisistaan melko paljon. Kolmea niistä voidaan käyttää kuvan kaappaukseen ja yksi sisältää konenäköalgoritmeja.

Kuvankäsittelytekniikoista esittelin kolme. Aritmeettisista operaatioista olen käyttänyt vähennyslaskua taustamallin kanssa siten, että havainnoinnin aikana kaapattua videokuvaa verrataan taustamalliin suorittamalla vähennyslasku kuvien pikseleille. Aritmeettista vähennyslaskua olen hyödyntänyt myös ihosta luodun mallin kanssa. Alueenlöytämisalgoritmin avulla etsitään yhtenäiset alueet kaapatuista kuvista, jotta alueille voidaan tehdä jatkoanalyysi hahmontunnistustekniikoihin kuuluvan minimi-maksimi-analyysin avulla. Keskiarvosuodatinta taas tarvitaan minimi-maksimi-analyysin kanssa, kun jostakin kuvan alueesta muodostetaan pysty- ja vaakasuuntaiset profiilit.

Hahmontunnistustekniikat jaoin neljään kategoriaan värin, muodon, piirteiden ja liikkeen perusteella. Muunkinlaiset kategorioinnit olisivat toki mahdollisia. Värin avulla etsin ihonväriset alueet videokuvasta. Ellipsin kallistuskulman avulla ihonväriset alueet käännetään pystysuoraan ennen jatkoanalyysiä. Piirteiden analysoinnissa olen käyttänyt yllä kuvattua minimi-maksimi-analyysiä. Liikettä en ole toistaiseksi juurikaan hahmontunnistuksessa hyödyntänyt paitsi taustamallin kanssa, mutta varsinkin *tapahtumapaikan* eli kioskin edustasta luodun mallin päivityksessä siitä olisi varmasti hyötyä.

## 4. Konenäön ja informaatiokioskin yhdistäminen

Luvussa 2 totesin, että kioskin käytön on oltava helppoa, nopeaa ja miellyttävää riippumatta käyttäjän taustasta ja kokemuksesta. Kioskin käyttöä voidaan yrittää helpottaa monella tapaa. Esimerkiksi monelle tietokoneen käyttöön tottumattomalle näppäimistön ja hiiren korvaaminen kosketusnäytöllä voi olla hyvä ratkaisu. Tällä ei kuitenkaan saada ratkaistua kaikkia kioskin käytön ongelmia. Tarvetta on siis myös muille kioskin käytettävyyttä parantaville ratkaisuille. Konenäöllä onkin monenlaisia informaatiokioskien käyttöä helpottavia ja nopeuttavia sovellusmahdollisuuksia. Seuraavaksi esittelen näitä sovellusmahdollisuuksia.

### 4.1. Yksinkertaisia sovellusmahdollisuuksia

Konenäön avulla käyttäjää voidaan tarkkailla häiritsemättä [Christian and Avery, 2000] ja hänestä voidaan saada selville monenlaisia asioita kuten sijainti kioskin ympäristössä, kasvonilmeet ja kehon eleet [Waters *et al.*, 1998]. Kun kioskiin yhdistetään konenäön lisäksi puhuva pää, saadaan kommunikoinnista luonnollisempaa. Hintakaan ei ole konenäön käytön este; koska web-kamerat ovat nykyään melko halpoja ja yleistymässä kotikäytössäkin, tarjoaa konenäkö mielenkiintoisia käyttömahdollisuuksia, myös muualla kuin kioskien yhteydessä.

Pelkkä paikalle tulevan ihmisen havaitseminen ja hänen sijaintinsa seuraaminen ovat käyttökelpoisia monissa sovelluksissa. Esimerkiksi, kun kioski huomaa käyttäjän vaikkapa viiden metrin päässä, voi se houkuttaa tätä luokseen. Toisaalta käyttäjien houkuttelu kioskin luokse voi olla häiritsevää, jos se tapahtuu liian tunkeilevasti. Houkuttelu voi tapahtua puheen avulla, ja jos käytössä on puhuva pää, niin myös sen avulla. Puhuvasta päästä saadaan luonnollisempi, jos se kääntyy tai katsoo käyttäjän suuntaan. Digital Smart Kiosk -projektin kioskissa [Christian and Avery, 1998] pää kääntyy käyttäjän suuntaan, kun käyttäjä on noin 4,5–6 metrin etäisyydellä kioskista ja kun käyttäjä tulee noin 1,5 metrin etäisyydelle kioskista, tervehtii kioski käyttäjää. Tervehdys valitaan kellon ajan mukaan, eli aamulla voidaan sanoa esimerkiksi ”hyvää huomenta” ja illalla ”hyvää iltaa”.

Kun käyttäjä on houkuteltu kioskin luokse, voidaan häntä alkaa heti opastaa kioskin käytössä. Tämä on tärkeää, koska informaatiokioskin käyttäjät eivät yleensä ole tutustuneet kioskiin aiemmin ja he luultavasti tarvitsevat ohjeita sen käytöstä. Jos käyttäjä on pidempään kioskin vieressä, niin samoja ohjeita ja kommentteja ei kannata toistaa, jotta ne eivät ala ärsyttää käyttäjää, paitsi jos toistolle on jokin erityisen hyvä syy. Kun käyttäjä poistuu kioskin luota, ei opastusta ole enää syytä jatkaa, ja uusia käyttäjiä voidaan alkaa houkutella kioskin luokse. Käyttäjä voidaan vielä lopuksi hyvästellä, kuten Digital Smart Kiosk -projektin kioski tekee.

Yllä olevien sovellusesimerkkien käytännön toteutusta on hankaloittamassa moni asia. Koska informaatiokioski sijaitsee yleensä julkisessa tilassa, kioskin edessä voi liikkua samaan aikaan paljon ihmisiä. Kioskin pitäisi erottaa käyttäjä muiden ihmisten joukosta. Tämän takia kioskin tulee rakentaa malli ympäristöstään, jotta se tietää, missä käyttäjä sijaitsee muihin ihmisiin ja ympäristöön nähden. Konenäöstä pitäisi saada myös melko luotettava, koska epäluotettavasti toimivan konenäön käyttö luultavasti vain hämmentää ja ärsyttää käyttäjää. Luotettavuuteen vaikuttaa moni asia, esimerkiksi havainnoinnin nopeus ja tarkkuus, havainnoitavan alueen laajuus ja konenäön sopeutuminen erilaisiin valaistusolosuhteisiin. Digital Smart Kiosk -projektin kioskissa oli aluksi ongelmia havainnoinnin nopeuden ja havainnoitavan alueen laajuuden kanssa. Havainnointi oli liian hidasta ja havainnoitava alue liian pieni, jolloin kioskin toiminta häiritsi käyttäjiä. Heistä nimittäin tuntui, että puhuva pää heräsi, kun kioskia lähestyi. Kameran linssi kuitenkin vaihdettiin laajempaa aluetta kuvaavaan ja konenäköä nopeutettiin 15 hertsiin, jolloin ongelmat poistuivat.

## 4.2. Vaativampia sovellusmahdollisuuksia

Mitä kehittyneempi konenäkö on, sitä enemmän se luonnollisesti tarjoaa sovellusmahdollisuuksia. Toisinaan useampi ihminen voi olla samaan aikaan käyttämässä informaatiokioskia. Mikäli konenäön avulla pystytään havainnoimaan samaan aikaan useita ihmisiä, voi puhuva pää katsoa vuoron perään eri ihmistä. Tällöin vuorovaikutuksesta saadaan hyvin toteutettuna luonnollisempaa kuin, jos pää katsoisi koko ajan vain yhtä ihmistä. Waters *et al.* [1998] ovatkin kokeilleet tätä tilanteessa, jossa kioski kertoi yksinkertaista tarinaa yhdelle ihmiselle ja antoi välillä kommentteja toiselle ihmiselle.

Jos konenäkö pystyy erottamaan ihmisen eri kehon osia, kuten pään ja kädet, voidaan kioskia ohjata yksinkertaisilla eleillä, esimerkiksi päätä nyökyttämällä tai kättä nostamalla ja laskemalla. Mitä tarkemmin ihmistä pystytään havainnoimaan, sen monipuolisempia eleitä ihminen voi käyttää. Monimutkaisempia eleitä voivat olla esimerkiksi suun avaus ja sulkeminen, silmien avaus ja sulkeminen, sekä erilaiset sormieleet. Myös kehon eleitä, kuten hyppimistä tai istumista, voidaan joissakin erikoistilanteissa hyödyntää.

Eleiden lisäksi kasvoilta voidaan havainnoida tunteita. Esimerkiksi ihmisen hymyn tunnistessaan puhuvalle päällekin voidaan antaa hymyilevä ilme. Kioski voi muuttaa toimintaansa myös muulla tavalla. Jos esimerkiksi käyttäjälle ollaan jo aiemmin annettu ohje jossakin tietyssä kioskin käyttötilanteessa, mutta käyttäjä näyttää turhautuvan tai ärsyyntyvän yrittäessään tehdä samaa asiaa uudelleen, hänelle voidaan antaa sama ohje uudestaan ja vaikkapa hieman eri tavalla esitettynä. Tunteiden havainnointi kasvoilta kuvien perusteella ja havainnoinnin mukaan kioskin toiminnan muokkaaminen on kuitenkin erittäin vaativaa, eikä asiaa ole tähän mennessä juurikaan tutkittu.

Liittämällä kioskiin konenäön lisäksi puheen havaitseminen ja tarvittaessa myös tunnistus lisääntyvät sovellusmahdollisuudet entisestään. Jos kioskin käyttäjä esimerkiksi keskustelelee vieressä seisovan ystävänsä kanssa ja kioski sanoo välissä jotakin, tämä luultavasti häiritsee käyttäjää. Siksi olisi hyvä, että kioski huomaisi käyttäjän puhuvan ja olisi tällöin hiljaa.

Jotta kioski osaisi edellä kerrotuissa tilanteissa toimia luonnollisesti, tarvitaan kioskille käyttäytymismalli, joka ohjaa kioskin toimintaa. Yksinkertaisimmillaan malli voi määrätä esimerkiksi silmien räpyttelytiheyden ja pään asennon vaihtumiset, mutta enemmän mallista on hyötyä, kun se toimii konenäön havainnoinnin ja kioskin käyttötilanteen perusteella. Toisin sanoen edellä mainitut kioskin luokse houkuttelu, käyttäjän opastus, useamman samanaikaisen käyttäjän kanssa kommunikointi ja tarvittaessa hiljaa oleminen edellyttävät käyttäytymismallia, jotta niistä saataisiin kunnolla toimivia.

### 4.3. Yhteenveto

Tässä luvussa käsiteltiin erilaisia sovellusmahdollisuuksia konenäön käytölle informatiokioskeissa. Osa sovelluksista on yksinkertaisia toteuttaa ja osa hyvin vaativia. Taulukossa 4.1 on vielä erilaiset sovellusmahdollisuudet yhteen koottuna.

Sovellus	Kuvaus
Kioskin luokse houkuttelu	Kioski houkuttelee luokseen, kun se havaitsee mahdollisen käyttäjän. Houkuttelu ei saa kuitenkaan olla liian häiritsevää.
Käyttäjän suuntaan katsominen	Puhuva pää katsoo käyttäjään tai mikäli käyttäjiä on useampia, tarvittaessa vuoron perään kutakin.

Käyttäjän tervehtiminen ja hyvästely	Kun käyttäjä tulee kioskin luokse, tervehditään käyttäjää esimerkiksi vuorokauden ajan mukaan. Käyttäjän poistuessa hänet hyvästellään.
Käyttäjän opastus	Käyttäjää aletaan opastaa heti, kun tämä tulee kioskin luokse ja opastusta annetaan aina tarvittaessa. Opastus toimii sitä paremmin, mitä paremmin käyttäjän toiminta ymmärretään.
Eleiden käyttö	Käyttäjä voi ohjata kioskia esimerkiksi käsi- ja sormieleillä tai päätään nyökyttämällä. Tämä edellyttää, että käyttäjän toiminnasta luodaan malli hänen liikkeidensä ja asentojensa perusteella.
Tunteiden havainnointi	Käyttäjän ilmeet, esimerkiksi hymyily tunnustetaan ja kioski muokkaa toimintaansa tämän perusteella. Myös kehon kieltä voitaisiin käyttää apuna tunteiden havaitsemisessa.
Hiljaisuus	Puhuva pää on hiljaa, kun huomaa käyttäjän puhuvan jollekin toiselle ihmiselle. Vaati konenäön lisäksi puheen havaitsemista.

**Taulukko 4.1.** Konenäön sovelluksia informaatiokioskeissa.

Konenäöllä on siis monenlaisia kioskin käyttöä helpottavia ja nopeuttavia sovellusmahdollisuuksia. Yksinkertaisimmat sovellukset onnistuvat pelkällä käyttäjän paikan havaitsemisella ja monimutkaisimmat sovellukset vaativat toimiakseen esimerkiksi käyttäjän tunteiden tulkintaa ja puheentunnistuksen yhdistämistä kioskiin.

## 5. Informaatiokioskin toteutus

Projektissa kehitetyn informaatiokioskin ydin on monipuolinen sovelluskehys. Siihen on liitetty erilaisia komponentteja, esimerkiksi selainkomponentti, puhuva pää ja kosketuskomponentti. Lähitulevaisuudessa siihen on tarkoitus liittää tämän tutkimuksen kannalta oleellisin osa eli konenäkökomponentti. Kioski esittää tietonsa WWW-sivujen muodossa. Tällä hetkellä kioskissa on tietoa Tampereen kaupungin museoista, mutta kioskin yleiskäyttöisen arkkitehtuurin ansiosta sisältö voidaan melko pienellä vaivalla vaihtaa toiseksi.

### 5.1. Käyttöliittymä

Kioskia ohjataan kosketusnäytön avulla, joko eleillä tai painikkeilla. Käyttöliittymässä ei ole hiirtä eikä näppäimistöä, koska monet mahdollisista käyttäjistä eivät ole niitä tottuneet käyttämään ja kosketusnäytön käyttö on tällöin helpompaa.

Kioskin käyttöliittymä näkyy kuvassa 5.1. Suurin osa kioskin näytöstä on varattu ikkunalle, jossa näytetään museoiden tietoja. Tämän ikkunan alapuolella on painikkeita, joiden avulla sivuilla voi liikkua. Liikkuminen onnistuu myös piirtämällä sormella eleitä oikeassa alakulmassa näkyvään ikkunaan. Eleikkunan yläpuolella on ohjeet eleiden käytölle. Puhuva pää on sijoitettu näytön oikeaan yläkulmaan. Käyttöliittymään ulkonäköä voidaan helposti muuttaa konfiguraatio-tiedostojen avulla. Tarvittaessa esimerkiksi navigointipainikkeet voidaan jättää pois.

Käyttöliittymässä ei näy videokameran ottamaa kuvaa. Digital Smart Kiosk -projektin kioskissa videokuva oli liitetty kioskin näytölle, koska Christianin ja Averyn [1998] mukaan käyttäjät pitivät itsensä näkemisestä ja se auttoi heitä ymmärtämään mitä kioski oli tekemässä. Videokuvan lisääminen meidänkin kioskiimme on mahdollista.



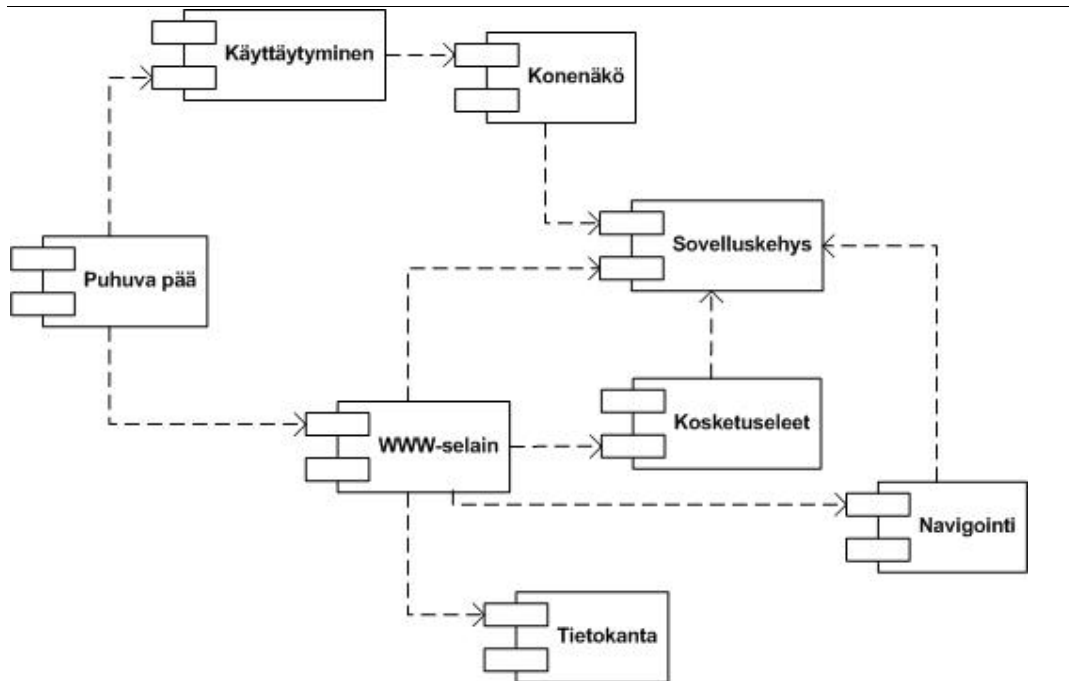


Kuva 5.1. Kuva kioskin käyttöliittymästä.

## 5.2. Sovelluskehys

Informaatiokioskin sovelluskehys on toteutettu C++ Builder -ympäristöllä. Kehyksen yhtenä tavoitteena on ollut, että siihen voidaan helposti liittää erilaisia ominaisuuksia. Kehys ja siihen liitettävät komponentit konfiguroidaan XML-tiedostojen avulla. XML-tiedostot jäsennetään kioskin käynnistysvaiheessa ja esimerkiksi käytettävät komponentit, käyttöliittymän ulkonäkö, sekä puhuvan pään kommentit määräytyvät tiedostoissa olevien asetusten perusteella.

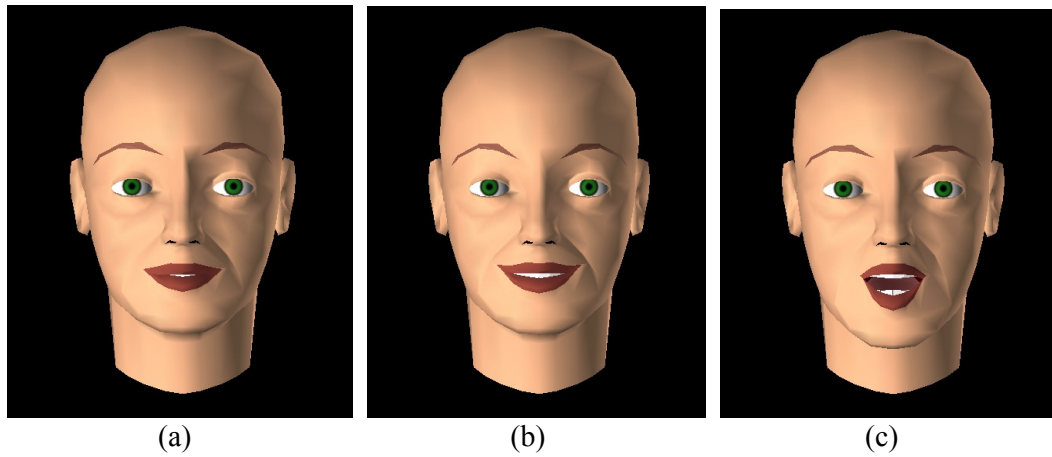
Kuvassa 5.2 on kioskin komponenttikaavio UML-notaation [Booch *et al.*, 1999] mukaan. Kioskin sovelluskehys on ydin, josta muut kioskin osat riippuvat. Kioskin käyttö tapahtuu kehykseen yhdistetyn WWW-selaimen avulla. WWW-selaimen näyttämät sivut haetaan tietokannasta. Kioskin eli lähinnä puhuvan pään käyttäytyminen riippuu käyttäjän WWW-selaimella tekemistä asioista. Käyttäjä käyttää WWW-selainta kosketuseleiden ja käyttöliittymässä näkyvien navigointipainikkeiden avulla. Myös konenäkö vaikuttaa tulevaisuudessa kioskin ja puhuvan pään käyttäytymiseen.



**Kuva 5.2.** Kioskin komponenttikaavio.

### 5.3. Puhuva pää

Kioskissa käytetty puhuva pää [Talker, 2001] on toteutettu alun perin Tampereen yliopistossa ja sen jälkeen sitä on kehitetty Teknillisessä korkeakoulussa. Päähän on liitetty Timehouse Oy:n Mikropuhe-syntetisaattori [Mikropuhe, 2001], joten pää puhuu suomea. Pää pystyy tuottamaan erilaisia ilmeitä ja erilaisia äänensävyjä. Se pystyy myös kääntymään ja katsomaan useaan eri suuntaan. Kuvassa 5.3 on tällä hetkellä kioskiin liitetty puhuva pää erilaisten ilmeiden kanssa.



**Kuva 5.3.** Kohdassa (a) puhuvalla päällä ”perusilme”. Kohdassa (b) pää hymyilee ja kohdassa (c) pää on sanomassa a-vokaalia.

Puhuvan pään käyttö on perusteltua, koska ihmiset reagoivat vahvasti ja välittömästi näkemiinsä kasvoihin varsinkin, jos ne kääntyvät katsomaan ihmistä ja seuraavat tämän liikkumista [Christian and Avery, 1998]. Toisin sanoen ihminen saadaan kiinnittämään huomio kioskien pään avulla, mikä on yleensä tarkoitus. Puhuva pää antaa kioskia käytettäessä erilaisia kommentteja, joskaan dialogia ei ole toistaiseksi kehitetty kovin pitkälle. Jatkossa tätäkin puolta olisi tarkoitus kehittää konenäön avulla. Christian ja Avery [1998] ovat kuitenkin sitä mieltä, että pään käyttäytymistä ohjaavan agentin kehitys voi olla vaikeaa, koska se vaatii agentin toteuttajalta ymmärrystä alueesta, josta kioskien tarkoitus antaa tietoa. Informaatiokioskin sisällöntuottajahan on usein eri henkilö kuin kioskien toteuttaja. Hyvin toteutettuna puhuva pää tekee kioskin käytöstä kuitenkin luonnollisempaa, koska esimerkiksi silmäkontakti puhuvan pään kanssa vastaa normaalia ihmisten välistä silmäkontaktia [Christian and Avery, 1998].

#### 5.4. Kosketuskomponentti

Kosketuskomponentin avulla informaatiokioskin käyttäjä voi esimerkiksi navigoida kioskin sisältämillä sivuilla piirtämällä sormellaan erilaisia merkkejä (eleitä). Käyttäjä pystyy myös ”nukuttamaan” ja ”herättämään” puhuvan pään eleiden avulla.

Eleet mahdollistavat kioskille monenlaisia toimintoja, joita voisi muuten olla vaikea sisällyttää kioskin käyttöliittymään. Näiden käyttö vain edellyttää, että ne ovat helppoja piirtää ja muistaa. Tämän takia kosketuskomponenttiin voidaan vaihtaa ja lisätä erilaisia eleitä.

Kosketuskomponenttia käsitellään tarkemmin Tuomisen [2002] tutkielmassa, ”Kosketuseleet informaatiokioskissa”. Hän on muun muassa tutkinut käytettävyytutkimusten avulla, miten käyttäjät suhtautuvat kosketuseleisiin ja helpottavatko eleet kioskin käyttöä.

## **5.5. Konenäkökomponentti**

Konenäkökomponentin tehtävänä on havainnoida informaatiokioskin ympäristöä videokameran avulla. Mikäli se havaitsee ihmisen tai ihmisiä videokuvassa, ilmoittaa se tästä sovelluskehyselle, joka edelleen toimii tilanteen vaatimalla tavalla. Konenäkökomponentti on tässä työssä esitettävä konstruktiivinen osuus.

Seuraavaksi esittelen konenäkökomponentin rakenteen ja kerron miten se on toteutettu. Selvitän myös, miten komponentin liittäminen informaatiokioskin sovelluskehukseen tapahtuu ja miten yhdistelmää voidaan hyödyntää käytännössä.

### **5.5.1. Toteutusympäristö**

Konenäkökomponentti on toteutettu MS Visual C++ 6.0 -ympäristöllä. Komponentin liittäminen sovelluskehukseen tapahtuu Microsoftin COM-mallin mukaisesti. Näin siksi, että sovelluskehys on toteutettu C++ Builder -ympäristöllä ja tavallisen MS Visual C++ -ympäristöllä tehdyn dynaamisen kirjaston tai staattisen kirjaston liittäminen siihen on erittäin hankalaa varsinkin, kun toteutus on olioperustainen. COM-malli ei ole kuitenkaan ongelma uudelleenkäyttämielessä, koska COM-malli on siirretty muun muassa UNIX-ympäristöön.

Konenäkökomponentti käyttää myös Microsoft Vision SDK -konenäkökirjastoa ja MFC-luokkakirjastoa, joten se toimii vain Microsoft Windows -käyttöjärjestelmissä. Toisaalta kioskin sovelluskehys on kehitetty Borland C++ Builder -ympäristöllä, joten kehyskin toimii vain MS Windows -käyttöjärjestelmissä. Riittää siis, että myös konenäkökomponentti toimii MS Windows -käyttöjärjestelmissä. Kamerakirjaston voi kuitenkin vaihtaa suhteellisen pienellä vaivalla, jolloin MFC-luokkakirjastonkaan käyttö ei olisi tarpeellista. Samalla komponenttia pystyisi hyödyntämään esimerkiksi Linux-käyttöjärjestelmässä.

### **5.5.2. Konenäkökomponentin rakenne ja toiminta**

Konenäkökomponentti on rakennettu olioperustaisesti. Liitteessä 1 on komponentin luokkakaavio UML-notaation mukaan. Liitteessä 2 on lyhyt kuvaus kustakin kaaviossa esiintyvistä luokasta ja rajapinnasta. Luokkien nimet ovat englanninkielisiä, koska komponentti on toteutettu englanninkielistä projektia varten. Kaaviossa ovat mukana vain tärkeimmät luokat ja rajapinnat. Monia vain toteutukseen liittyviä luokkia on jätetty pois ja mukana olevien luokkien kaikkia funktioita ei ole myöskään näkyvissä.

Konenäkökomponentin toiminta on pääpiirteissään seuraavanlainen:

1. Kioskikehys hakee COM-mallin avulla viittauksen konenäkökomponentin rajapintaan ja käynnistää havainnoinnin.
2. Kioskikehys rekisteröityy kuuntelemaan tapahtumapaikan mallin muutoksia. Toisin sanoen, jos konenäkökomponentti havaitsee kioskin eteen saapuvan uuden ihmisen, kioskin edessä liikkuvan ihmisen tai paikalta poistuvan ihmisen, se kertoo tästä kioskikehykselle. Kahdessa ensimmäisessä tilanteessa se samalla kertoo ihmisen sijainnin tapahtumapaikalla.
3. Konenäkökomponentti päivittää tapahtumapaikan mallia videokuvan analyysin perusteella ja ilmoittaa muutoksista kioskikehykselle, kunnes kioskikehys lopettaa havainnoinnin konenäkökomponentin rajapinnan kautta. Tarkkailijoita voi olla enemmän kuin yksi, jos sille on tarvetta.

Kuten edellä kuvatusta selviää, konenäkökomponentin ainoa tehtävä on luoda mahdollisimman hyvä malli kioskin edustasta ja kertoa mallissa tapahtuvista muutoksista kiinnostuneille kuuntelijoille. Kioskikehyksen tai paremminkin siihen liitetyn käyttäytymiskomponentin tehtäväksi jää sitten päättää mitä tehdä konenäkökomponentin antamilla tiedoilla.

Konenäkökomponentin tapahtumapaikasta luoman mallin hyvyys riippuu siitä, miten hyvin sen suorittama analyysi on toteutettu. Analyysin toiminta on pääpiirteissään seuraava:

1. Videokameran ottamien kuvien perusteella luodaan malli kioskin edustasta. Malliin tallennetaan löydettyjen ihmisten sijainnit kuvasta toiseen. Mallia päivitetään analysoimalla yksittäiset videokameran kaappaamat kuvat.
2. Jokaista kaapatun kuvan pikseliä verrataan taustamalliin aritmeettisen vähennyslaskun avulla. Aritmeettinen vähennyslasku suoritetaan pikselin punaiselle, vihreälle ja siniselle värille sekä kirkkaudelle. Mikäli jokin tai jotkin erot ylittävät määritellyn kynnsarvon oletetaan, että kaapatun kuvan pikseli ei kuulu taustamalliin.
3. Kun taustaan kuulumaton pikseli löydetään, tutkitaan edelleen ihonväristä luodun mallin avulla, onko kyseessä ihonvärinen pikseli. Mikäli on, annetaan pikselille nimi alueenlöytämisalgoritmin mukaisesti.
4. Kuva käydään lopuksi vielä toisen kerran läpi alueenlöytämisalgoritmin mukaisesti, jotta ihonväriset alueet saadaan yhdistettyä.
5. Kun kuvasta on löydetty ihonväriset alueet, niille lasketaan parhaiten vastaava ellipsi ja alue käännetään ellipsin kallistuskulman perusteella pystysuoraan. Tämän jälkeen alueille suoritetaan minimi-maksimi-analyysi.

6. Kaikki kuvan alueet, joille kasvojen todennäköisyydeksi on saatu minimi-maksimi-analyysin avulla suurempi kuin määritelty kynnyisarvo oletetaan kasvoiksi. Alueiden perusteella päivitetään tapahtumapaikan mallia siten, että edellisen analyysikierroksen tuottaman mallin kasvoille haetaan nyt löydetystä kasvoista vanhoja lähimpänä sijaitsevat parit. Vanhan mallin kasvot korvataan uusilla pareillaan. Mikäli jollekin vanhalle kasvolle ei useamman päivituksen jälkeen löydy paria, poistetaan se kokonaan mallista. Mikäli jollekin uudelle kasvolle ei löydy paria, tallennetaan se malliin uutena ja seuraavan kerran mallia päivitettäessä sille haetaan paria kuten muillekin mallin kasvoille.

Edellä kuvattu analyysi toimii yleensä hyvin. Sillä on kuitenkin joitakin vaatimuksia ja rajoitteita. Ensinnäkin analysoinnin pitää olla riittävän nopeaa, jotta mallin päivitys toimisi. Mikäli päivitys ei ole riittävän nopeaa, saattavat ihmiset ehtiä liikkumaan kuvasta toiseen niin paljon, että vanhan mallin ja uuden mallin kasvoista ei saada muodostettua oikeita pareja.

Toinen päivitykseen liittyvä ongelma on se, että mikäli käyttäjä liikkuu toisen ihmisen taakse, kuvittelee komponentti kyseessä olevan vain yhden ihmisen ja toimii sen mukaisesti. Ongelma olisi hyvin pitkälle ratkaistavissa laskemalla käyttäjän liikkumisen nopeus ja suunta mallista toiseen ja estimoimalla tämän perusteella, missä käyttäjä on seuraavassa kuvassa.

Ihonväri aiheuttaa omat rajoituksensa, eli jos kioskin käyttäjä on esimerkiksi maalannut kasvonsa, havainnointi ei toimi. Käytännössä tilanne on kuitenkin niin harvinainen, että siihen ei ole tarvetta varautua. Ihonväriin liittyy kuitenkin myös todellisempia ongelmia. Esimerkiksi ihon pigmentti voi vaihdella hyvinkin paljon. Tämän takia ihonväriä kuvaavan mallin pitää kattaa riittävän laajasti erilaisia pigmenttejä. Toisaalta liian laajastikaan se ei saisi niitä kattaa, koska silloin myös muita kuin ihonvärisiä alueita voidaan erehtyä luulemaan ihoksi. Taustamallin päivitykseenkin liittyy omat ongelmansa, mutta näitä on käsitelty jo luvussa 3.

## 5.6. Yhteenveto

Tässä luvussa esiteltiin Tampereen yliopistossa kehitetty informaatiokioski. Kioskia on helppo muokata lisäämällä ja poistamalla siihen erilaisia komponentteja. Jokaisesta kioskiin tällä hetkellä liitetystä komponentista kerrottiin luvussa lyhyesti. Tarkemmin käsiteltiin konenäkökomponentin rakennetta ja toimintaa.

Olen toteuttanut konenäkökomponentin olioperustaisesti ja se liitetään kioskiin COM-mallin avulla. Komponentti havainnoi tapahtumapaikkaa, päivittää tapahtumapaikasta luotua mallia ja ilmoittaa kioskiyehtykselle ihmisten liikkeistä tapahtumapaikalla. Tapahtumapaikan mallin päivitykseen liittyy joitakin ongelmia, joihin luvussa esitettiin ratkaisuehdotuksia.

## 6. Konenäön empiiriset tutkimukset

Luvussa 2 esitettiin erilaisia vaatimuksia informaatiokioskeille sekä sovellusmahdollisuuksia konenäön käytölle informaatiokioskeissa. Luvussa kerrottiin, että konenäön pitää olla luotettava ja toimia nopeasti, jotta siitä olisi hyötyä kioskin kanssa. Sen jälkeen, kun tämä tavoite on saavutettu, voidaan keskittyä varsinaiseen tavoitteeseen eli kioskin käytön nopeutukseen ja helpottamiseen. Tässä luvussa esittelen suorittamani empiiriset tutkimukset, joilla on tutkittu konenäön luotettavuutta ja nopeutta.

Tutkin konenäköä kahden kokeen avulla. Ensimmäisen kokeen tavoitteena oli selvittää konenäön nopeus ja miten luotettava konenäkö on tavallisessa kioskin käyttötilanteessa. Toisessa kokeessa tutkittiin pelkästään konenäön luotettavuutta.

### 6.1. Konenäön nopeus ja kasvojen havaitseminen

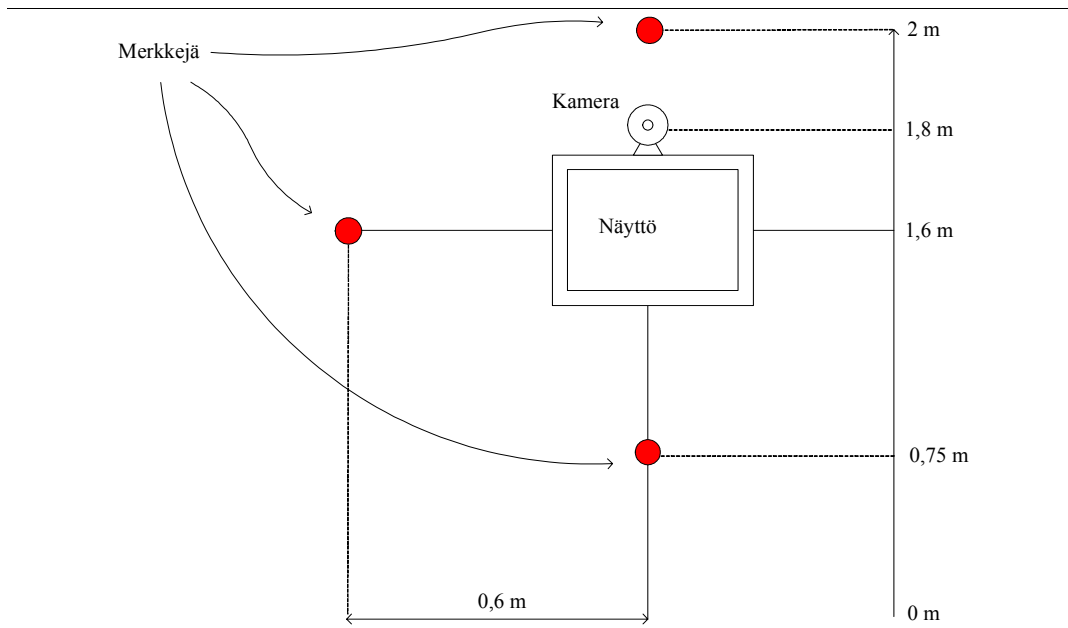
Ensimmäiseen kokeeseen osallistui kahdeksan koehenkilöä. Kuvassa 6.1 on kakkien koehenkilöiden kasvot. Kuvat on otettu koetilanteessa. Kasvot on järjestetty siten, että ylimpänä vasemmalla on ensimmäisen koehenkilön kuva, vieressä toinen koehenkilö, ja niin edelleen. Kaikkien koehenkilöiden ihon pigmentti on vaalea ja yhdellä oli koetilanteessa käytössä silmälasit. Mitään vaatimuksia koehenkilöiden vaatetukselle tai muulle pukeutumiselle ei asetettu.

Kutakin koehenkilöä pyydettiin kävelemään kulman takaa hieman yli 3 metrin päästä noin metrin päähän web-kamerasta (vaihe 1). Kävelyvaiheelle varattiin aikaa 8 sekuntia, jotta koehenkilö ehti kävellä rauhassa kameran eteen ja asettua paikoilleen. Lattialla oli jalan kuvat sillä kohtaa, mihin käyttäjän tuli sijoittua. Kamera oli sijoitettu 1,8 metrin korkeuteen näytön päälle samalle korkeudelle kuin se olisi tyypillisesti informaatiokioskin kanssa. Kun käyttäjä oli kävellyt kameran eteen, häntä pyydettiin aluksi katsomaan suoraan näytöllä näkyvää videokuvaa (vaihe 2). Tälle vaiheelle varattiin aikaa 3 sekuntia. Tämän jälkeen käyttäjää pyydettiin kääntämään päätään ylöspäin 2 metrin korkeuteen sijoitettua merkkiä kohden (vaihe 3), sitten noin 0,6 metrin päähän vasemmalle (vaihe 4) ja lopuksi alaspäin (vaihe 5). Kukin näistä vaiheista kesti kolme sekuntia. Koejärjestely on esitetty kuvassa 6.2. Kuvassa ei näy lattiassa olevia merkkejä, joiden päällä koehenkilöiden piti seistä. Merkit olivat kuitenkin suoraan kameran edessä.



**Kuva 6.1.** Koehenkilöt järjestyksessä vasemmalta oikealle ja ylhäältä alas.





**Kuva 6.2.** Koeasetelma.

Kamera otti koetilanteissa yhteensä 2631 kuvaa, joista tilan ja muistin säästämiseksi tallennettiin vain joka neljäs levyllä myöhempää tulosten analysointia varten. Kussakin vaiheessa tallennettiin seuraavat määrät kuvia:

- vaihe 1: 351 kuvaa (näistä 181 sisälsi ihmisen),
- vaihe 2: 118 kuvaa,
- vaihe 3: 55 kuvaa,
- vaihe 4: 83 kuvaa, ja
- vaihe 5: 122 kuvaa.

Lisäksi kustakin koetilanteesta luotiin loki, johon tallennettiin seuraavat asiat:

- kuinka monta kuvaa kone näkö analysoi koetilanteessa ja kuinka kauan siihen yhteensä kului aikaa,
- kustakin tallennetusta kuvasta löydettyjen ihonväristen alueiden lukumäärät ja todennäköisyys, että kyseessä on kasvat, ja
- jos kasvojen todennäköisyydeksi saatiin yli nolla prosenttia, tallennettiin lokiin myös silmien (kaksi koordinaattia), nenän (yksi koordinaatti) ja suun (kaksi koordinaattia) sijainnit.

Koetilanteen olosuhteet olivat melko vaativat. Katossa kameran yläpuolella oli loistevalo, joka aiheutti joissakin kuvissa otsan alueen ”palamisen”. Informaatiokioski voidaan kuitenkin sijoittaa lähes millaiseen ympäristöön tahansa, jolloin havainnointiolosuhteet saattavat olla huonot. Konenäön pitäisi toimia tällaisissakin tilanteissa, joten on perusteltua, että koe suoritettiin vastaavissa olosuhteissa.

### **6.1.1. Konenäön nopeus**

Konenäön nopeutta mittasin toteuttamalla yksinkertaisella C++-kielisellä profiloijalla, jonka yhdistin testiohjelmaan. Koetilanteessa profiloija laski havainnointifunktion kutsujen lukumäärän ja näiden kuluttaman ajan ja tallensi tiedot kokeen lopuksi lokiin. Tietojen perusteella saatiin laskettua, kuinka paljon konenäkö ehtii analysoida kuvia koetilanteen kaltaisessa tilanteessa, jossa yksi käyttäjä seisoo noin metrin päässä kioskista.

Konenäön nopeuden mittaukseen vaikuttaa eniten se, paljonko kuvasta löytyy ihonvärisiä alueita ja miten suuria ne ovat. Toisin sanoen, mitä enemmän kioskin edessä seisoo ihmisiä ja mitä lähempänä he ovat, sitä enemmän konenäöllä on analysoitavaa ja sitä kauemmin analysointi kestää. Tyypillisesti kioskia käyttää kuitenkin vain yksi ihminen kerrallaan, ja käyttäjä asettuu korkeintaan metrin päähän kioskista, joten toteutetun kokeen avulla saatiin melko hyvin selville konenäön käytännön nopeus.

Taulukossa 6.1 näkyvät profiloijan koetilanteissa laskemat luvut. Lukujen perusteella konenäön nopeudeksi saadaan hieman yli 23 hertsiä. Koetilanteessa kuitenkin noin 23 prosentissa kuvista ei sisältänyt ihmistä (koehenkilö ei ollut vielä tullut kulman takaa kuvaan). Tällaisissa kuvissa konenäöllä on hyvin vähän analysoitavaa, jolloin havainnointi-funktion kutsu on hyvin nopea. Jos oletetaan, että näissä tilanteissa havainnointi ei vie lainkaan aikaa ja halutaan selvittää, miten nopeasti muiden kuvien analysointi on tässä tapauksessa tapahtunut, voidaan funktion kutsujen lukumäärästä vähentää 23 prosenttia ja tällöin konenäön nopeudeksi tulee hieman alle 18 hertsiä. Tästä voidaan päätellä, että konenäön nopeus sijoittuu välille 18–23 hertsiä silloin, kun kameran edessä on yksi ihminen.

Koehenkilön numero	Havainnointi-funktion kutsujen lukumäärä	Funktion kutsujen yhteensä kuluttama aika (sekuntia)
1	322	14,2
2	338	14,2
3	334	14,3
4	311	14,2
5	366	13,8
6	304	14,3
7	325	14,1
8	331	14,1
<b>Yhteensä:</b>	2631	113,3

**Taulukko 6.1.** Profiloijan laskemia havainnoinnin nopeutta kuvaavia lukuja.

Pentlandin [1998] mukaan MIT Media Lab:ssa kehitetty havainnointimenetelmä, jolla pysytytään seuraamaan yhtä ihmistä, toimi 10 hertsin nopeudella. Digital Smart Kiosk -projektissa konenäön nopeus oli noin 15 hertsiä, joka oli Christianin ja Avernyn [1998] mukaan riittävästi. Tässä kokeessa mitattu nopeus 18–23 hertsiä pitäisi siis myös riittää meidän kioskimme tarpeisiin.

### 6.1.2. Konenäön luotettavuus

Ensimmäinen koe, jossa käytettiin useita koehenkilöitä, oli monipuolisempi kuin kohdassa 6.2 esitettävä koe, ja sen avulla pyrittiin selvittämään seuraavia asioita:

- miten luotettavasti konenäkö toimii eri tilanteissa (käyttäjän lähestyessä, katsoessa suoraan näyttöä, katsoessa pää käännettynä ylöspäin, katsoessa pää käännettynä sivulle ja katsoessa pää käännettynä alaspäin),
- onko eri tilanteiden välillä eroja konenäön luotettavuudessa,
- onko eri käyttäjien välillä eroa konenäön luotettavuudessa, ja
- miten hyvin konenäön laskema todennäköisyys kasvoille takaa, että silmät, nenä ja suu löydetään oikeista paikoista.

Konenäön luotettavuuteen vaikuttavat monet jo aikaisemmissa luvuissa mainitut asiat. Jotta kasvojen alue yleensä löydetäisiin videokuvasta, pitää ihosta luodun mallin olla riittävän hyvä, ja jotta kasvoista löydetäisiin oikeat piirteet, pitää lisäksi minimi-maksimi-analyysin olla toimiva. Kokeessa selvisi, että kasvojen alueen löytäminen ei tuottanut ongelmia. Konenäkö löysi kaikki kasvojen alueet, joita kuvissa esiintyi. Sen sijaan minimi-maksimi-analyysi laski monissa tilanteissa kasvojen todennäköisyydeksi nolla prosenttia, joten kasvot jäivät sen takia löytymättä.

Taulukosta 6.2 näkyy, kuinka monesta prosentista kuvia konenäkö on löytänyt kasvot eri vaiheissa eri koehenkilöiden osalta. Näissä vaiheissahan kasvot olivat aina löydettävissä kuvista. Vaiheiden väleillä on erittäin suuria eroja. Vaiheissa 1 ja 5 konenäkö on löytänyt kasvoista vain hieman yli 20 prosenttia, kun taas vaiheiden 2 ja 4 kohdalla löydettyjen kasvojen osuus on yli 90 prosenttia. Varsinkin lähestymisvaiheessa henkilöidenkin välillä on suuria eroja; kolmen henkilön tapauksessa heitä ei löytynyt yhdestäkään kuvasta, mutta yksi henkilö löytyi lähes 60 prosentista kuvia. Koehenkilö 2 piti päänsä suorassa vaiheen 3 aikana, joten hänen kohdaltaan kyseisestä vaiheesta ei ole tuloksia.

Koehenkilön numero	Vaihe 1 (lähestyminen)	Vaihe 2 (pää suorassa)	Vaihe 3 (pää ylöspäin)	Vaihe 4 (pää sivulle)	Vaihe 5 (pää alaspäin)
1	0,00 %	75,00 %	80,00 %	100,00 %	33,33 %
2	0,00 %	94,12 %	*	100,00 %	54,55 %
3	21,74 %	100,00 %	88,89 %	100,00 %	31,25 %
4	59,09 %	100,00 %	100,00 %	100,00 %	20,00 %
5	7,14 %	100,00 %	50,00 %	73,33 %	5,56 %
6	4,00 %	64,29 %	83,33 %	50,00 %	0,00 %
7	0,00 %	100,00 %	100,00 %	100,00 %	12,50 %
8	82,61 %	100,00 %	100,00 %	100,00 %	31,25 %
<b>Kasvojen löytyminen keskimäärin:</b>	21,82 %	91,68 %	86,03 %	90,42 %	23,55 %

**Taulukko 6.2.** Kasvojen löytyminen kuvista prosentuaalisesti.

Yksi selkeä syy, joka selittää vaiheen 1 ja vaiheen 5 alhaiset löytymisprosentit, on konenäkökomponentin toteutukseen liittyvä. Komponentti on nimittäin toteutettu siten, että jos ihonvärisestä alueesta löydetään esimerkiksi silmä- ja suuehdokkaat, mutta ei yhtään nenäehdokasta, lasketaan kasvojen todennäköisyydeksi nolla prosenttia riippumatta siitä, miten hyvin silmä- ja suuehdokkaat täyttävät niille asetetut ehdot. Toisin sanoen nykyinen toteutus vaatii, että kaikille, silmille, nenälle ja suulle, löytyy ainakin yksi ehdokas. Jos näin ei ole, kasvojen todennäköisyys on aina nolla prosenttia. Tämä on selvästi parannusta kaipaava kohta konenäkökomponentissa.

Taulukosta 6.3 näkyvät konenäön kasvojen piirteiden ehtojen jäsenyysfunktioiden perusteella kasvoille lasketut todennäköisyydet. Todennäköisyydet ovat melko alhaisia, joten kasvojen piirteiden ehtoja on tarvetta muuttaa, jotta ehdoille lasketuista jäsenyyksistä saadaan suurempia ja sitä kautta kasvojen todennäköisyydetkin kasvavat.

Koehenkilön numero	Vaihe 1 (lähestyminen)	Vaihe 2 (pää suorassa)	Vaihe 3 (pää ylöspäin)	Vaihe 4 (pää sivulle)	Vaihe 5 (pää alaspäin)
1	0,00 %	62,58 %	65,95 %	57,53 %	69,46 %
2	0,00 %	55,39 %	*	57,15 %	59,80 %
3	54,69 %	61,07 %	62,99 %	62,33 %	37,37 %
4	52,36 %	39,69 %	36,10 %	39,63 %	29,70 %
5	63,85 %	53,18 %	54,59 %	50,40 %	44,22 %
6	38,28 %	56,52 %	46,30 %	47,01 %	0,00 %
7	0,00 %	51,69 %	47,50 %	55,94 %	58,78 %
8	47,95 %	61,22 %	60,35 %	39,68 %	50,43 %
<b>Kasvoille laskettu todennäköisyys keskimäärin:</b>	32,14 %	55,17 %	53,40 %	51,21 %	43,72 %

**Taulukko 6.3.** Konenäön kasvoille laskemia todennäköisyyksiä.

Kasvojen löytymisen lisäksi koetilanteessa selvitettiin kasvopiirteiden löytymistä. Mikäli koetilanteessa jonkin kuvan kohdalla saatiin kasvoille suurempi todennäköisyys kuin nolla prosenttia (tällaisia kuvia oli yhteensä 291) tallennettiin lokiin merkintä silmien, nenän ja suun paikasta. Kävin nämä merkinnät läpi ja tarkistin kuvista, oliko piirteet merkitty oikeisiin paikkoihin.

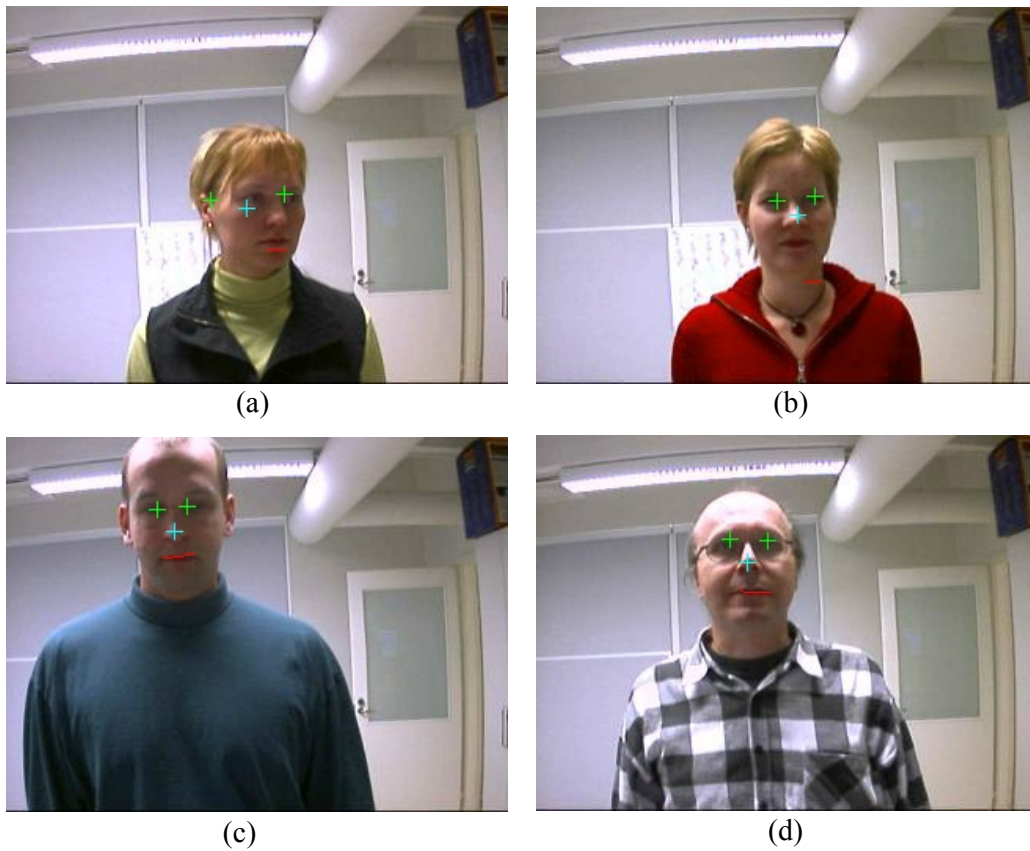
Taulukossa 6.4 on merkitty kunkin koehenkilön osalta oikeiden merkintöjen prosentuaalinen osuus kunkin kasvopiirteiden osalta. Taulukossa on kaikkien vaiheiden yhteenlaskettu keskiarvo, mutta laskuissa ei ole otettu huomioon lähestymisvaihetta, koska koehenkilöistä suurimman osan kohdalla kasvot jäivät havaitsematta tässä vaiheessa ja tämän takia eri henkilöiden lukuja on helpompi vertailla toisiinsa, kun vaihe ei ole mukana. Katsoin silmät oikein sijoitetuiksi vain, mikäli kumpikin silmä oli sijoitettu oikealle paikalleen.

Koehenkilön numero	Oikein sijoitetut silmät	Oikein sijoitettu nenä	Oikein sijoitettu suu
1	93,33 %	66,67 %	63,33 %
2	5,26 %	5,26 %	2,63 %
3	50,00 %	8,33 %	5,56 %
4	45,45 %	51,52 %	18,18 %
5	61,29 %	54,84 %	58,06 %
6	70,59 %	58,82 %	88,24 %
7	54,29 %	54,29 %	48,57 %
8	53,13 %	53,13 %	59,38 %
<b>Oikein keskimäärin:</b>	54,17 %	44,11 %	42,99 %

**Taulukko 6.4.** Kasvonpiirteiden sijoittuminen oikeille paikoilleen.

Kuten taulukosta huomataan, on koehenkilöiden ja kasvonpiirteiden kohdalla hyvin paljon eroja. Koehenkilön 2 huonot luvut selittyvät sillä, että ihonväristä luotu malli ei kyennyt erottamaan hiuksia ja ihoa toisistaan. Näin oli myös koehenkilön 3 tapauksessa; tosin hänen tapauksessaan silmien sijoitus kohdalleen onnistui huomattavasti paremmin kuin nenän ja suun. Koehenkilön 4 tapauksessa suun huono sijoitus johtuu suurehkosta paidan kaula-aukosta.

Kuvassa 6.3 on esimerkkejä väärin ja oikein sijoitetuista kasvojen piirteistä. Kuvan 6.3a esimerkissä ihonväriä ja hiuksien väriä ei ole erotettu toisistaan, ja tämän takia piirteet on löydetty vääristä paikoista. Kuvan 6.3b piirteet on sijoitettu suuta luukun ottamatta oikeille paikoilleen. Tässä tapauksessa paidan avoin kaula-aukko aiheutti virheelliseen havaintoon. Kuvissa 6.3c ja 6.3d piirteet on sijoitettu otsan palamisesta huolimatta oikeille paikoilleen.



**Kuva 6.3.** Kasvonpiirteiden onnistuneita ja epäonnistuneita havaintoja.

Eri vaiheiden kohdalla kasvonpiirteiden sijoituksessa on hyvin paljon eroa (keskihajonta vaiheiden välillä silmien osalta 26,38 prosenttiyksikköä, nenän osalta 23,80 prosenttiyksikköä ja suun osalta 5,99 prosenttiyksikköä). Tämä on ymmärrettävää, koska minimi-maksimi-analyysi toimii parhaiten, kun kasvot ovat suoraan edestä päin näkyvissä. Vaiheen 3 (pää suorassa ja kasvot näyttöä kohden) kasvonpiirteiden oikeiden sijoitusten prosentuaaliset osuudet on esitetty taulukosta 6.5.

Koehenkilön numero	Oikein sijoitetut silmät	Oikein sijoitettu nenä	Oikein sijoitettu suu
1	100,00 %	100,00 %	66,67 %
2	12,50 %	12,50 %	6,25 %
3	100,00 %	12,50 %	12,50 %
4	88,24 %	94,12 %	5,88 %
5	100,00 %	100,00 %	87,50 %
6	77,78 %	66,67 %	100,00 %
7	100,00 %	100,00 %	18,75 %
8	86,67 %	86,67 %	80,00 %
<b>Oikein keskimäärin:</b>	83,15 %	71,56 %	47,19 %

**Taulukko 6.5.** Kasvonpiirteiden sijoittuminen oikeille paikoilleen vaiheessa 3 (pää suorassa)

Tästä kuten edellisestäkin taulukosta huomataan, että koehenkilöiden 2 ja 3 tapauksessa hiusten väri on aiheuttanut suuria hankaluuksia piirteiden havaitsemisessa. Myös koehenkilön 4 tapauksessa suurehkon paidan kaula-aukon vaikutus on näkyvissä. Koehenkilöllä 7 suuta ei ole yleensä osattu sijoittaa oikein, vaan se on sijoitettu hieman liian ylös (kuitenkin nenän alapuolelle). Toisaalta monen koehenkilön kohdalla on tullut täysin oikeita kasvojen piirteiden sijoituksia.

Taulukon alimmalta riviltä nähdään kuinka usein kasvojen piirteet on sijoitettu oikein keskimäärin. Varsinkin silmien osalta tulos on hyvä, yli 83 prosenttia. Itse asiassa tulos olisi yli 93 prosenttia jos koehenkilö 2 ei olisi mukana laskuissa.

## 6.2. Kasvojen ja käsien erottaminen toisistaan

Toisen kokeen järjestely oli sellainen, että pidin kättäni erilaisissa asennoissa 60 sekuntia noin 0,3–1 metrin etäisyydellä kamerasta. Kaikki kameran ottamat kuvat tallennettiin levyille myöhempää tarkastelua varten. Lokiin tallennettiin kustakin kuvasta löydettyjen ihonväristen alueiden lukumäärät ja todennäköisyys sille, että kuvassa on kasvat.

Kuvia, joissa käsi oli näkyvissä, kertyi yhteensä 1111 kappaletta. Näistä 87,67 prosenttia hylättiin oikein eli kasvojen todennäköisyydeksi laskettiin nolla prosenttia. Jos tulosta verrataan oikeiden kasvojen löytymisen todennäköisyyksiin, huomataan niissä merkittävä ero, sillä kasvoista kaiken kaikkiaan noin 63 prosenttia havaittiin ja käsistä vain noin 12:ta prosenttia erehdyttiin luulemaan kasvoiksi.



Virheellisesti kasvoiksi luultujen havaintojen tapauksessa jäsenyyksien todennäköisyyksien keskiarvo kasvoille oli 46,89 prosenttia ja keskihajonta oli 10,43 prosenttiyksikköä. Ensimmäisessä kokeessa kasvoille laskettu jäsenyyksien keskiarvo oli 52,23 prosenttia ja keskihajonta 9,04 prosenttiyksikköä. Näiden perusteella voidaan laskea Z-testin arvoksi 5,13 (havaittujen kasvojen lukumäärä ensimmäisessä kokeessa oli 291 ja toisessa kokeessa se oli 137). Ero on siis tilastollisesti merkitsevä jopa 0,1 prosentin merkitsevyystasolla. Tämä tarkoittaa käytännössä sitä, että kasvot ja kädet voidaan tavallisesti erottaa toisistaan jäsenyyksien todennäköisyyksien perusteella. Täytyy kuitenkin huomata, että kun konenäkökomponentin toteutusta muutetaan aiemmin tässä luvussa mainitulla tavalla, tulevat edellä lasketut arvotkin todennäköisesti muuttumaan.

### **6.3. Yhteenveto**

Tässä luvussa kuvattiin kaksi eri koetta, joiden avulla selvitettiin alustavasti konenäkökomponentin luotettavuutta ja nopeutta. Nopeus on kokeiden perusteella nykyisellään riittävä, mutta luotettavuutta täytyy vielä parantaa. Kasvot ja kädet erotettiin toisistaan hyvin. Sen sijaan kasvojen havaitsemisessa oli hyvin paljon vaihtelua eri vaiheiden välillä, kasvoille lasketut todennäköisyydet olivat melko alhaisia ja kasvonpiirteiden sijoittaminen oikeille paikoilleen oli toisinaan hyvin epäluotettavaa.

Jatkossa suoritettavissa kokeissa tulisi olla suurempi joukko koehenkilöitä, joilla esimerkiksi ikä ja ihon pigmentaatio vaihtelee. Tällöin kokeen tulokset olisivat luotettavampia. Mahdollista olisi myös käyttää jotakin kuvatietokantaa kokeissa.

## 7. Lopuksi

Tutkimuksessa tutustuttiin konenäköön ja informaatiokioskeihin, sekä esiteltiin konenäkökomponentti, jolla pystytään havainnoimaan samanaikaisesti useiden ihmisten kasvoja ja kasvonpiirteitä videokuvasta. Konenäön toimintaa kokeiltiin kahden kokeen avulla, joissa selvisi, että konenäkö toimii ainakin yhden käyttäjän tapauksessa reaaliaikaisesti. Konenäön luotettavuudessa on kokeiden perusteella vielä kehittämistä vaikka se erottaakin esimerkiksi kasvot ja kädet toisistaan melko luotettavasti.

Minimi-maksimi-analyysin kohdalla mainitsin, että suodattimen koon vaihtaminen tarvittaessa voisi parantaa konenäön luotettavuutta. Tätä asiaa voisi tutkia jatkossa. Mainitsin myös, että kasvoille lasketaan nollan prosentin todennäköisyys, jos kaikkia kasvonpiirteitä ei löydetä. Tätä kohtaa voisi kehittää niin, että jos esimerkiksi silmät ja suu löydetään suurella todennäköisyydellä, näiden välistä haetaan nenäehdokasta hieman pienemmillä vaatimuksilla.

Luotettavuuden lisäksi konenäköä voidaan kehittää monin muin tavoin. Koko käyttäjän havainnointi pelkkien kasvojen sijaan voisi olla yksi tutkimussuunta. Koko käyttäjähän on aritmeettisen vähennyslaskun avulla helppo erottaa taustasta. Eri kehon osien erottaminen toisistaan ei ole kuitenkaan yhtä helppoa ja mallin luominen käyttäjän liikkeistä on vieläkin vaativampaa.

Tunteiden havainnointi kasvoista olisi myös yksi mielenkiintoinen vaikkakin vaativa tutkimusaihe. Mikäli tunteita onnistuttaisiin havainnoimaan luotettavasti, ja kioski osaisi mukautua käyttäjän tunteisiin, saataisiin kioskin käytöstä entistä luontevampaa.

Konenäkökomponentista on varmasti nykyiselläänkin hyötyä kioskin kanssa ja seuraavaksi komponentti on tarkoitus liittää kioskiin ja kioskin käyttäytymistä aletaan kehittää älykkäämmäksi.

Tulevaisuudessa suoritetaan myös empiirisiä tutkimuksia, joissa konenäkö on yhdistetty kioskiin. Konenäkökomponentin tehtävänä on nopeuttaa ja helpottaa kioskin käyttöä. Myös viihdyttävyyttä voidaan yrittää parantaa. Yksi tapa kokeilla, miten konenäkö onnistuu näissä tavoitteissa, ovat käytettävyytystutkimukset. Koejärjestely voisi olla esimerkiksi sellainen, että osa koehenkilöistä kokeilee kioskia ilman konenäkökomponenttia ja osa sen kanssa. Tämän jälkeen tutkitaan tilanteesta luotujen lokien ja koehenkilöille suoritettujen haastattelujen avulla, miten tavoitteissa on onnistuttu.

## Viiteluettelo

- [Aggarwal and Cai, 1997] J. K. Aggarwal and Q. Cai, Human Motion Analysis: A Review. In: *IEEE Nonrigid and Articulated Motion Workshop*, 90–102.
- [Booch *et al.*, 1999] G. Booch, J. Rumbaugh and I. Jacobson, *The Unified Modeling Language User Guide*. Addison-Wesley, 1999.
- [Borchers *et al.*, 1995] Jan Borchers, Oliver Deussen and Clemens Knörzer, Getting It Across: Layout Issues for Kiosk Systems. In: *Proceedings of the Workshop on W3-Based Online Kiosk Systems*, Third International World-Wide Web Conference, Darmstadt 1995. Reprinted in: *SIGCHI Bulletin* **27**, 4 (October 1995), 68–74. Also available as <http://citeseer.nj.nec.com/borchers95getting.html>.
- [Bregler *et al.*, 1998] C. Bregler, S.M. Omohundro, M. Covell, M. Slaney, S. Ahmad, D.A. Forsyth and J.A. Feldman, Probabilistic Models of Verbal and Body Gestures. In: Roberto Cipolla and Alex Pentland (eds.) *Computer Vision for Human-Machine Interaction*. Cambridge University Press, Cambridge, 1998, 267–290.
- [Christian and Avery, 1998] Andrew D. Christian and Brian L. Avery, Digital Smart Kiosk Project. In: *Proceedings of CHI '98* (1998), ACM Press, 155–162.
- [Christian and Avery, 2000] Andrew D. Christian and Brian L. Avery, Speak out and Annoy Someone: Experiences with Intelligent Kiosks, *CHI 2000*, 313–320.
- [Cho *et al.*, 2001] Kyung-Min Cho, Jeong-Hun Jang, Ki-Sang Hong, Adaptive skin-color filter, *Pattern Recognition* **34**, 5 (May 2001), 1067–1073.
- [COM, 2000] Component Object Model. Microsoft, 2000. See <http://www.microsoft.com/com/>.
- [Cormen *et al.*, 1989] Thomas H. Cormen, Charles E. Leiserson and Ronald L. Rivest, *Introduction to Algorithms*. MIT Press, 1989.
- [Dillencourt *et al.*, 1992] Michael B. Dillencourt, Hannan Samet and Markku Tamminen, A General Approach to Connected-Component Labeling for Arbitrary Image Representations. *J. ACM* **39**, 2 (Apr. 1992), 253–280.
- [DirectX, 2001] *Microsoft DirectX*. Microsoft, 2001. Available at <http://www.microsoft.com/directx/>.
- [FaceIt, 2001] *FaceIt*. Visionics Corporation, 2001. Available at <http://www.visionics.com/faceit/>.

- [Farkas, 1994] L.G. Farkas, *Anthropometry of the Head and Face*. RavenPress, 1994.
- [Gavrila, 1999] D. M. Gavrila, The Visual Analysis of Human Movement: A Survey, *Computer Vision and Image Understanding* **73**, 1 (1999), 82–98.
- [Jones and Rehg, 2000] Michael J. Jones and James M. Rehg, Skin Color Modeling and Detection. Available as <http://www.crl.research.digital.com/vision/humansensing/skin/default.htm>.
- [Kosko, 2001] Bart Kosko, *Sumea Logiikka*. Art House Oy, 2001.
- [Maes *et al.*, 1996] Pattie Maes, Trevor Darrell, Bruce Blumberg and Alex Pentland, The ALIVE System: Wireless, Full-body Interaction with Autonomous Agents. In: *Computer Animation '95 Proceedings*, IEEE Press, 11–18. Also available as <http://www.media.mit.edu/characters/papers/TR-257.pdf>.
- [Maggioni and Kämmerer, 1998] C. Maggioni and B. Kämmerer, GestureComputer – History, Design and Applications. In: Roberto Cipolla and Alex Pentland (eds.) *Computer Vision for Human-Machine Interaction*. Cambridge University Press, Cambridge, 1998, 23–51.
- [Matsuno *et al.*, 1995] K. Matsuno, C.-W. Lee, S. Kimura and S. Tsuji, Automatic Recognition of human facial expressions. In: *Proceedings of the Fifth International Conference on Computer Vision*, 1995, 352–359.
- [Mikropuhe, 2001] Mikropuhe, 2001. Katso <http://www.timehouse.fi/oy/mikropuhe.htm>.
- [OpenCV, 2001] *Open Source Computer Vision Library*. Intel, 2001. Available at <http://www.intel.com/research/mrl/research/opencv/>.
- [Rosales and Sclaroff, 1998] Rómer Rosales and Stan Sclaroff, Improved Tracking of Multiple Humans with Trajectory Prediction and Occlusion Modeling. In: *CVPR Workshop on the Interpretation of Visual Motion*, 1998.
- [Sengupta *et al.*, 2000] Kuntal Sengupta, Wong Hon Bing and Pankaj Kumar, Computer Vision Games Using a Cheap (< 100 \$) Webcam. Available as <http://www.ece.nus.edu.sg/stfpage/eleks/ICARCV'2000.pdf>.
- [Seul *et al.*, 2000] Michael Seul, Lawrence O’Gorman and Michael J. Sammon, *Practical Algorithms for Image Analysis: description, examples, and code*. Cambridge University Press, Cambridge, 2000.
- [Shearer *et al.*, 2001] Kim Shearer, Kirrily D. Wong and Svetha Venkatesh, Combining multiple tracking algorithms for improved general performance, *Pattern Recognition* **34**, 6 (June 2001), 1257–1269.

- [Sobottka and Pitas, 1996] K. Sobottka and I. Pitas, Looking for faces and facial features in color images, *Pattern Recognition and Image Analysis: Advances in Mathematical Theory and Applications*. Russian Academy of Sciences, 1996. Also available as <http://citeseer.nj.nec.com/sobottka96looking.html>.
- [Steiger and Suter, 1994] Patrick Steiger and Bettina Ansel Suter, MINNELLI – Experiences with an Interactive Information Kiosk for Casual Users. In: Bischofberger W.R., Frei H.P. (eds.) *Computer Science Research at UBILAB, Strategy and Projects; Proceedings of the UBILAB '94 Conference* (1994), 124–133. Also available as [http://www.ubilab.org/publications/print\\_versions/pdf/ste94b.pdf](http://www.ubilab.org/publications/print_versions/pdf/ste94b.pdf).
- [Suzuki *et al.*, 2000] Kenji Suzuki, Isao Horiba and Noboru Sugie, Fast Connected-Component Labeling Based on Sequential Local Operations in the Course of Forward Raster Scan Followed by Backward Raster Scan. In: *IPSJ J.* **41**, 11 (2000); *Proc. of the ICPR '00 Conference* (2000), IEEE, 434–437.
- [Talker, 2001] *Finnish-Talking Head*. Helsinki University of Technology, Laboratory of Computational Engineering, Cognitive Technology Team, 2001. Information available at <http://www.lce.hut.fi/research/face/index.html>.
- [Tuominen, 2002] Petri Tuominen, Kosketuseleet informaatiokioskissa. Pro gradu -tutkielma, Tampereen yliopisto, Tietojenkäsittelytieteiden laitos, Tammikuu 2002.
- [VisSDK, 2001] *Microsoft Vision SDK*. Microsoft, 2001. Available at <http://research.microsoft.com/visionsdk/>.
- [Waters *et al.*, 1998] K. Waters, J. Rehg, M. Loughlin, S.B. Kang and D. Terzopoulos, Visual Sensing of Humans for Active Public Interfaces. In: Roberto Cipolla and Alex Pentland (eds.) *Computer Vision for Human-Machine Interaction*. Cambridge University Press, Cambridge, 1998, 83–96.
- [Wren *et al.*, 1997] Christopher Wren, Ali Azarbayejani, Trevor Darrell and Alex Pentland, Pfunder: Real-Time Tracking of the Human Body. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**, 7 (July 1997), 780–785.
- [QuickSDK, 2000] Logitech QuickCam SDK. Logitech, 2000. Available at <http://developer.logitech.com/>.
- [Yang *et al.*, 2001] Ming-Hsuan Yang, Narendra Ahuja and David Kriegman, A Survey on Face Detection Methods. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Draft, 2001.



## Liite 2

Konenäkökomponentin tärkeimmät luokat ja rajapinnat

VisionFactory

VisionFactory on COM-mallin edellyttämä luokkaolio, joka toteuttaa IClassFactory-rajapinnan. Tämän rajapinnan avulla kioskikehyksestä päästään käsiksi konenäkökomponenttiin. VisionFactory luo pyydettyä MainVisionControl-olion ja palauttaa asiakkaalle (kioskikehykselle) osoittimen IVisionComponent-rajapintaan.

IVisionComponent

IVisionComponent-rajapinnan avulla asiakas aloittaa ja lopettaa konenäkökomponentin ajamisen. Rajapinta tarjoaa myös getIScene-funktion, joka palauttaa osoittimen IScene-rajapintaan. IVisionComponent on peritty COM-mallin IUnknown-rajapinnasta, joka kaikkien COM-mallin mukaisten rajapintojen täytyy periä (suoraan tai epäsuorasti).

IScene

IScene-rajapinnan avulla asiakas voi tarkkailla kioskiin eteen tulevia ja lähteviä ihmisiä. Esimerkiksi kioskin edessä olevien ihmisten lukumäärän selvitykseen, ja tietyn ihmisen paikan ja etäisyyden selvitykseen on omat funktionsa. Mikäli asiakas haluaa tarkkailla muutoksia tapahtumapaikalla, voi addSceneListener-funktiolla lisätä tarkkailijan (ISceneListener-rajapinnan toteuttava luokka) tarkkailemaan tapahtumapaikkaa (kioskin edessä olevaa tilaa). Rekisteröinnin jälkeen tarkkailijalle ilmoitetaan kaikista tapahtumapaikalla tapahtuvista muutoksista.

ISceneListener

ISceneListener sisältää kolme funktiota, joiden avulla tarkkailijalle kerrotaan, jos uusia ihmisiä tulee kameran eteen, kameran edessä olevat ihmiset liikkuvat tai joku ihminen lähtee pois.

MainVisionControl

MainVisionControl-luokka toteuttaa IVisionComponent-rajapinnan ja se huolehtii koko konenäkökomponentin hallinnasta. Se esimerkiksi luo muut Control-oliot ja käynnistää säikeen, jossa kioskin edessä olevan tilan tarkkailu tapahtuu.

SceneModel

SceneModel-luokka toteuttaa IScene-rajapinnan. Luokka tallentaa kulloisenkin tapahtumapaikan tilan, eli montako ihmistä tapahtumapaikalla on ja missä ihmiset sijaitsevat. Aina kun tilassa tapahtuu muutoksia, se ilmoittaa niistä rekisteröityneille ISceneListener-tarkkailijoille.

BlobControl

BlobControl-luokka hakee videokuvasta taustaan kuulumattomat alueet. Alueiden löytämiseen käytetyn algoritmin karkea kuvaus on esitelty alakohdassa 3.4.3. BlobControlin käyttämä algoritmi ei kuitenkaan tutki binärisoitua kuvaa vaan värikuvaa. Kullekin pikselille, joka ei kuulu taustaan (selvitetään BackgroundModel-luokan avulla) tutkitaan samat naapurit kuin perusalgoritmissa. Lisäehtona nimeämisessä on se, että mikäli tutkittava pikseli on ihonvärinen (selvitetään SkinModel-luokan avulla), naapurinkin pitää olla ihonvärinen, jotta tutkittavalle pikselille voitaisiin antaa sama nimi kuin naapurille tai se voitaisiin lisätä yhtäsuuruusluokkaan.

Siinä vaiheessa, kun kuva käydään toisen kerran läpi (ks. alueenlöytämisalgoritmi luvussa 3), kullekin alueelle luodaan oma Blob-olio, johon on tallennettu siihen kuuluvat pikselit. Lopuksi kaikkein pienimmät Blob-oliot poistetaan listasta, johon ne on tallennettu, koska tällaisia alueita on mahdotonta jatkoanalysoida ja ne saattavat olla kuvassa esiintyvää tavallista voimakkaampaa häiriösignaalia.

#### Blob

Blob-luokkaan tallennetaan siis kuvan tietyn yhtenäisen alueen pikselit. Blob-olioita käytetään muun muassa kasvojen etsinnässä.

#### BackgroundModel

BackgroundModel-luokka ylläpitää mallia kameran edessä olevasta muuttumattomasta eli staattisesta ympäristöstä. Tätä mallia kutsutaan tästä eteenpäin nimellä *taustamalli*. Taustamallissa on tallennettuna tieto kuvien taustasta, joka ei normaalisti muutu. Taustaan kuuluvat usein esim. seinät ja kiinteä sisustus. Taustamallia päivitetään jatkuvasti, joten vaikka esineitä siirrettäisiin tai kamera liikahtaisi, mukautuu taustamalli uuteen tilanteeseen.

#### SkinModel

SkinModel-luokkaan on mallinnettu tieto ihmisen ihon ominaisuuksista. Sen avulla voidaan selvittää, onko jokin pikseli ihonvärinen. Tarkkuus, johon mallilla päästään, riippuu määritellyistä värisävyn ja värikylläisyyden rajoista, valaistusolosuhteista ja käyttäjien ihonväristä.

#### CameraControl

CameraControl-luokka kaappaa kuvia kameralta. Se käyttää apuna Microsoft Vision SDK:ta. CameraControl-luokka ”paketoii” kaappaamansa kuvat Image-luokan sisälle, jolloin kamerakirjasto voidaan vaihtaa helposti siten, että muutoksia tulee vain CameraControl-, Image- ja RowPointer-luokkiin. Mihinkään muihin luokkiin ei tarvitse tällöin siis tehdä muutoksia.

#### Image



Image-luokka kapseloi Microsoft Vision SDK -ohjelmistokehityspaketin kuvien toteutuksen. Tarvittaessa toteutus voidaan vaihtaa konenäkökomponentin toimintaa rikkomatta, kunhan Image-luokan rajapintaa ei muuteta.

#### RowPointer

Kullekin kuvan riville on oma RowPointer-olionsa. Tämän avulla päästään käsiksi yksittäiseen pikseliin. Luokka kapseloi Microsoft Vision SDK -ohjelmistokehityspaketin vastaavan luokan.

#### PersonDetectionControl

PersonDetectionControl-luokka analysoi BlobControl-luokan löytämät alueet ja rakentaa niiden perusteella mallit kuvassa olevista ihmisistä, sekä päivittää SceneModel:a. Tällä hetkellä PersonDetectionControl rakentaa kuvassa olevien ihmisten mallit vain FaceDetectionControllin löytämien kasvojen perusteella, mutta sitä voidaan helposti laajentaa esimerkiksi siten, että siihen liitetään käsien havaitsemiseen erikoistunut luokka.

#### FaceDetectionControl

FaceDetectionControl on vastuussa kasvojen etsinnästä. Apunaan se tällä hetkellä käyttää MinMaxAnalysis-luokkaa, joka nimensä mukaisesti etsii kasvoja ja kasvonpiirteitä minimi-maksimi analyysin avulla (ks. alakohta 3.5.3).

#### MinMaxAnalysis

Koska Minimi-maksimi-analyysi vaatii melko paljon toiminnallisuutta (esimerkiksi su-mean logiikan jäsenyysfunktioita), on se erotettu omaksi luokakseen.

#### Person, Face, Torso ja Hand

Nämä luokat tuskin kovin paljon selityksiä vaativat. Person koostuu Face-, Torso- ja kahdesta Hand-oliosta. Itseasiassa Torso- ja Hand-luokka eivät sisällä tällä hetkellä toteutusta, mutta ne saatetaan toteuttaa myöhemmässä vaiheessa. Face-luokkaan tallennetaan luonnollisesti kasvonpiirteet eli esimerkiksi silmien, nenän ja suun sijainnit.