

# Mittarin lukeman digitalisointi

Juha-Pekka Ahonen

Opinnäytetyö  
Huhtikuu 2015

Automaatiotekniikan koulutusohjelma  
Tekniikan ja liikenteen ala





Tekijä(t) Ahonen, Juha-Pekka	Julkaisun laji Opinnäytetyö	Päivämäärä 24.4.2015
	Sivumäärä 52	Julkaisun kieli Suomi
		Verkojulkaisulupa myönnetty ( x)
Työn nimi <b>Mittarin lukeman digitalisointi</b>		
Koulutusohjelma Automaatiotekniikan koulutusohjelma		
Työn ohjaaja(t) Markku, Ström Seppo, Rantapuska		
Toimeksiantaja(t) ProSolve Oy		
Tiivistelmä <p>Opinnäytetyön toimeksiantajana oli ProSolve Oy, joka tuottaa erilaisia koneensuunnittelu-, kiinteistö- ja 3D-skannauspalveluita.</p> <p>Tavoitteena oli tehdä toimiva konenäkösovellus, joka mittaa analogisen mittarin tiedon ja lähettää sen Internetiin, sille varatulle salasanalla suojatulle sivulle. Työn tekotavaksi valittiin konenäkösovellus ja webkamera, sillä tämä tapa on helpoin ja halvin monistaa useisiin eri käyttökohteisiin. Ratkaisuvaihtoehtoina selvitettiin AD-muuntimen käyttöä tai että mittari vaihdettaisiin digitaaliseksi. Koneenäkösovellusta suunniteltaessa on ensin huomioitava ohjelmalliset vaatimukset kuvan laadulle ja valaistuksen vaikutus niihin. Sen jälkeen voidaan keskittyä konenäkösovelluksen tekemiseen.</p> <p>Opinnäytetyössä käytettiin PC-pohjaista älyä. Demotyön tekoon valittiin GNU Octave-sovellus sekä halpa webkamera, jotta konenäkösovellusta voitiin testata. Octave-sovelluksella saatiin kuvan laatu riittävän hyväksi, mutta ohjelman Hough -muunnos ei toiminut odotetulla tavalla.</p> <p>Työlle varattu aika ei ollut riittävä projektin valmistamiseen ilman ohjelman tuntevan henkilön avustusta. Tämän hetkinen ongelma on, että Hough -muunnos antaa samaa tulosta kaikilla kuvilla. Työtä jatketaan vielä opinnäytetyö palautuksen jälkeen.</p> <p>Kun Hough -muunnos saadaan toimimaan, jää konenäkösovellukseen vielä paljon parannettavaa: sitä voi optimoida, siihen voi tehdä graafisen käyttöliittymän ja tieto pitää myös jakaa Internetiin. Tällä hetkellä sovellus laskee Hough-muunnoksesta viisarin asennon ja pituuden, ottaen huomioon käyttäjän asettamat tiedot.</p>		
Avainsanat (asiasanat) Koneenäkö, GNU Octave, Analoginen mittari		
Muut tiedot		



Author(s) Ahonen, Juha-Pekka	Type of publication Bachelor's Thesis	Date 24.4.2014
	Pages 52	Language Finnish
		Permission for web publication ( x)
Title <b>Digitalization of meter reading</b>		
Degree Programme Automation Engineering		
Tutor(s) Ström, Markku Seppo, Rantapuska		
Assigned by ProSolve Oy		
<p>Abstract</p> <p>The thesis was assigned by ProSolve Oy. ProSolve Oy provides services in machine design, real estate and 3D scanning.</p> <p>The goal was to make a working program that will measure analogical needle position of an analogical meter and send it to the Internet, to a password secure website that is reserved for it.</p> <p>When planning a machine vision application the image quality requirements of the software must be considered as well as the effect of lighting on them. After that it is possible to focus on completing the software. Some basic image processing methods are also discussed in the thesis.</p> <p>When choosing the work method, a machine vision application and web camera were chosen, because they present the easiest and cheapest way to replicate this system to many different locations and within this choice, PC-based intelligence was used. As a demo for software GNU Octave and cheap web camera were chosen for programming. Other options were either AD-transformer or changing analog meter directly to digital meter.</p> <p>The time reserved for completing this project was not enough, without help of a person who is familiar with this program. The problem at the moment is that Hough-transform gives the same outcome on all images, thus, the work will be continued even after returning of the thesis.</p> <p>When the application is ready, there is still plenty of work: the software can be optimized, it is possible to create graphic user interface and the data must be shared as well. At the moment Hough-transform counts the radius and length of pointer in meter, taking in consideration what user inputs.</p>		
Keywords Machine Vision, GNU Octave, Analog meter		
Miscellaneous		

# SISÄLTÖ

<b>1</b>	<b>Opinnäytetyön lähtökohdat .....</b>	<b>1</b>
<b>2</b>	<b>Konenäkö.....</b>	<b>1</b>
2.1	Kuvanmuodostus .....	2
2.2	Optiikka.....	6
2.3	Kameroiden ja linssien eroja .....	12
2.4	PC-pohjainen äly vs. älykäs kamera.....	14
2.5	Mitoitus .....	16
2.6	Valaistus.....	18
2.7	Kuvan käsittely.....	22
<b>3</b>	<b>Opinnäytetyön ratkaisuvaihtoehdot .....</b>	<b>28</b>
3.1	AD-muunnin.....	28
3.2	Konenäkösovellus ja kamera .....	31
3.3	Mittarin vaihto.....	32
<b>4</b>	<b>Opinnäytetyön toiminnalliset valinnat.....</b>	<b>32</b>
4.1	Opinnäytetyön tekotapa .....	32
4.2	Konenäköohjelma.....	33
4.3	Kamera.....	34
<b>5</b>	<b>Toteutus .....</b>	<b>35</b>
5.1	Konenäkö .....	35
5.2	Kamera.....	50
5.3	Valaistus ja jalusta .....	50
5.4	Tiedon jakaminen turvallisesti.....	51
<b>6</b>	<b>Pohdinta .....</b>	<b>51</b>
	<b>Lähteet .....</b>	<b>53</b>
	<b>Liitteet .....</b>	<b>Virhe. Kirjanmerkkiä ei ole määritetty.</b>

- Liite 1. Together function, ohjelmointikoodi **Virhe. Kirjanmerkkiä ei ole määritetty.**
- Liite 2. Dial\_indicator function, ohjelmointikoodi.....**Virhe. Kirjanmerkkiä ei ole määritetty.**
- Liite 3. Background\_generator function, ohjelmointikoodi ...**Virhe. Kirjanmerkkiä ei ole määritetty.**
- Liite 4. Dial\_reader function, ohjelmointikoodi .....**Virhe. Kirjanmerkkiä ei ole määritetty.**
- Liite 5. Hought function, ohjelmointikoodi .. **Virhe. Kirjanmerkkiä ei ole määritetty.**
- Liite 6. Houghpeaks function, ohjelmointikoodi .....**Virhe. Kirjanmerkkiä ei ole määritetty.**
- Liite 7. Houghlines function, ohjelmointikoodi .....**Virhe. Kirjanmerkkiä ei ole määritetty.**
- Liite 8. Alkuperäinen ohjelmointikoodi ..... **Virhe. Kirjanmerkkiä ei ole määritetty.**

## Kuviot

Kuvio 1. Pikselien määrän vaikutus kuvan selkeyteen.....	3
Kuvio 2. Kameran parametrit, jotka muodostavat sen näkökentän.....	3
Kuvio 3. Kontrasti, jota kuvataan kanttiaaltona, ylhäällä sävy ja kanttiaallossa kontrastin vahvuus .....	4
Kuvio 4. Harmaasävyjen määrää vaikuttaa kuvan informaatiomäärään.....	5
Kuvio 5. Modulaationsiirtofunktion selvittämiseen tarvittava testauskuva.....	5
Kuvio 6. Kahdella eri linsillä otetut kuvat, oikean puoleisella kameralla on parempi modulaationsiirtofunktio .....	6
Kuvio 7. Valon käyttäytyminen objektiivissa.....	7
Kuvio 8. Tarkennuspisteen vaikutus terävyysalueeseen .....	8
Kuvio 9. Koko optiikkaa selventävä kuva .....	8
Kuvio 10. Suljin aukkoja kuvaava kuva.....	9
Kuvio 11. Valon määrää suhteessa kuvan vaakasuuntaisten pikselien määrään .....	10
Kuvio 12. Polttovälin pituuden vaikutus kuvan vääristymään.....	10
Kuvio 13. Esimerkki CCD-kennoisen matriisikamera toiminnasta .....	11
Kuvio 14. Kaksi esimerkkikuvaa CCD-kennoista, oikealla viiva- ja vasemmalla matriisikamera .....	12
Kuvio 15. Pyörähdyskappaleesta saatu kuva .....	13
Kuvio 16. Telesentrinen linssi (oikealla) poistaa kuvasta syvyyden.....	14
Kuvio 17. PC-pohjainen äly.....	15
Kuvio 18. Älykameran kuvan muodostus.....	16
Kuvio 19. Optiikan laskukaavaa hahmottava kuvio .....	17
Kuvio 20. Kulmavirhettä selventävä kuva .....	18

Kuvio 21. Esimerkki pikselivenymästä.....	19
Kuvio 22. Ylhäällä kuva Dome-valaistuksen rakenteesta. Alapuolella oikealla puolella on kuva, jossa käytetty Dome-valaistusta, kun vasemmalla puolella on normi valaistuksella otettu kuva.....	20
Kuvio 23. Esimerkki Darkfield-valaistuksesta.....	21
Kuvio 24. Esimerkki Rakenteellisesta valaistuksesta .....	22
Kuvio 25 Esimerkki maskista ja sen vaikutuksesta laajennuksen kohteeseen.....	23
Kuvio 26. Esimerkki dilaatiosta.....	23
Kuvio 27. Hough -muunnoksen esimerkki tulos.....	26
Kuvio 28. AD-muuntimen piirikaaviosymboli, vasemmalla analoginen ja oikealla digitaalinen signaali.....	28
Kuvio 29. Esimerkki liitäntä, $R_1 = 10\text{ k}\Omega$ ja $R_2 = 5\text{ k}\Omega$ , resoluutio 8bittinen .....	29
Kuvio 30. Perus lämpömittari, jossa ei ole sähköliittimiä .....	31
Kuvio 31. Herätyskello joka toimii testimittarina.....	36
Kuvio 32. Aloitusvaiheen testikuvat .....	37
Kuvio 33. HSV-kuvasta RGB-kuvaksi muunnetut kuvat.....	38
Kuvio 34. RGB kuvasta harmaasävykuvaksi muunnetut kuvat .....	39
Kuvio 35. Grayscale-kuvasta BW-kuvaksi muutettu kuva .....	40
Kuvio 36. Suoraan alkuperäisestä kuvasta im2bw-funktiolla muutettu mustavalkokuva.....	41
Kuvio 37. Kuvassa pisteet kuvastavat missä kuvan pisin viiva sovelluksen mielestä on. ....	43
Kuvio 38. Kuvassa pisteiden pitäisi kuvastaa kuvan suurimman viivan molempia reunoja. ....	44
Kuvio 39. Tuoreempi kuva, jossa jokainen kuvaaja näyttää, missä sen mittarin pisin viiva on, kuvaaja esittää toista kuvan pisimmistä viivoista.....	45
Kuvio 40. Hough -muunnos, joka antaa jokaisella mittarilla saman tuloksen .....	46
Kuvio 41. Testikuva, josta on poistettu mahdolliset viisaria isommat viivat ja ympyrät .....	47
Kuvio 42. Testikuvan Hough -muunnos, jossa vähemmän häiriötä mutta samat isot pisteet.....	48
Kuvio 43. Jamkin logo, jolle on tehty dilaatio ja Hough -muunnos.....	49
Kuvio 44. Prosolven logo, jolle on tehty dilaatio ja Hough -muunnos.....	50

## 1 Opinnäytetyön lähtökohdat

Opinnäytetyön toimeksiantajana oli insinööritoimisto ProSolve Oy. Yrityksen toimipisteet ovat Jyväskylässä ja Kotkassa, joista Jyväskylä on päätoimipaikka. Henkilömäärä on noin 30. Yritys tarjoaa monia erilaisia asiantuntija-, suunnittelu-, mittaus-, skannaus- ja mallinnuspalveluita. (ProSolve työpaikkana n.d.)

Yritys jakautuu kolmeen osaan toimialojen mukaan:

- ProLine-koneensuunnittelupalvelut
- ProKipa-kiinteistöpalvelut
- ProDigit-3D-skannauspalvelut. (Prosolve n.d.)

Prosolven asiakkaalla oli idea, että analogisia mittareita on paljon ja niistä saatavaa dataa olisi hyvä saada automatisoitua ja seurattua etäältä. On olemassa helpompi ja huokeampi tapa kerätä tieto analogisesta mittarista kuin vaihtaa koko mittari digitaaliseen.

Opinnäytetyön tavoitteena oli selvittää, miten analogisesta mittarista voidaan saada digitaalinen arvo. Internetissä on useita samankaltaisia sovelluksia, joissa on käytetty monia erilaisia ohjelmia, mutta niissä ei näy ohjelmointikoodia. Tässä opinnäytetyössä oli tarkoitus käyttää ilmaisohjelmaa kustannusten säästämiseksi. Käytetyksi ohjelmaksi valitsin GNU Octaven.

Opinnäytetyön tehtävänä oli tehdä esimerkkimittarille sovellus, joka voi lukea digitaaliseksi analogisen mittarin tiedon, millä tahansa ajan hetkellä. Toteutusvaihtoehtoina kartoitettiin A/D-muuntimen käyttö ja konenäkösovellus. Lisäksi tarkasteltiin analogisten mittareiden vaihtoa digitaalisiin taloudellisessa mielessä.

## 2 Konenäkö

Konenäöllä on tarkoitus matkia ihmisen näköaistia koneellisesti. Kameralle asetetaan haluttu kuvanottotaajuus eli se, kuinka usein kamera ottaa kuvan. Kuvanlaadun tarkaaminen edellyttää valaisuolosuhteita, jotka eivät muutu. Otetut kuvat digitoidaan

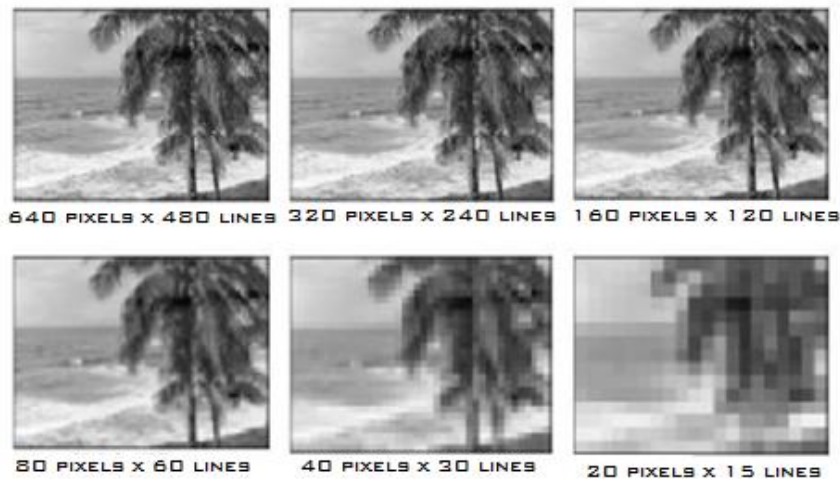
ja siirretään tietokoneelle, jossa ne analysoidaan jo ennalta ohjelmoidun sovelluksen mukaisesti. Analyysistä saatu tieto voidaan siirtää esim. tuotantolinjaan, joka toimii ohjelmoidulla tavalla. Näin pullonpalautuskone, joka joko hyväksyy tai hylkää pulloja, sen mukaan onko koneeseen syötetty esine pullon muotoinen ja mikä viivakoodin numero on. (Soini n.d.) Konenäkö muodostuu seuraavista asioista: optiikka, kamera, valaisin ja elektroniikka (Alroth 2010, 8).

## **2.1 Kuvanmuodostus**

Kuvankaappaus tehdään optiikalla ja kameran kennolla. Kuvan muodostuksen parametreihin kuuluu resoluutio, näkökenttä, kontrasti ja modulaation siirtokyky (Alroth 2010, 17).

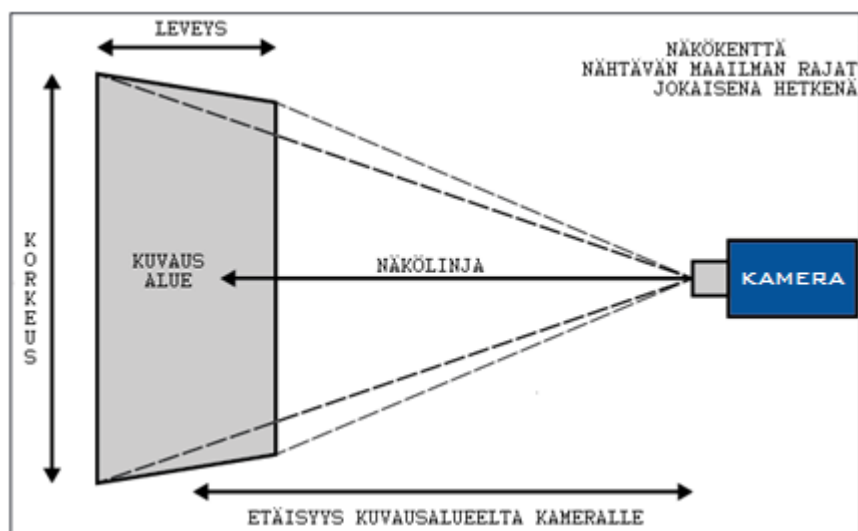
Resoluutio vastaa sitä, kuinka monta bittiä A/D-muunnoksessa käytetään. Bittien määrä on suoraan suhteessa harmaa- tai väritasojen määrään kameran tyypistä riippuen. Kuvan resoluutio myös vastaa siinä olevaa pikselimäärää. Mitä korkeampi resoluutio on, sitä tarkemmalta kuva näyttää olettaen, että kuva on tarkennettu oikein. (Rantapuska 2014.)

Kuten kuviosta 1 näkyy kuva, jossa on suurin määrä pikseleitä, on parhaan näköinen. Kuvassa, jossa on vähiten pikseleitä, voi erottaa pikselimäiset neliöt, joista kuva muodostuu.



Kuvio 1. Pikselien määrän vaikutus kuvan selkeyteen (Rantapuska 2014)

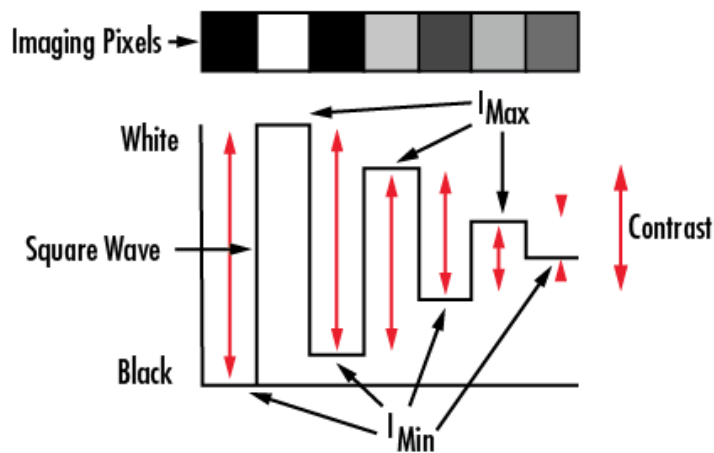
Näkökenttä, eli kameran kuva-alan suuruus, valitaan siten että kameran kuva alueelle mahtuu kuvattavat kohteet (Rantapuska 2014). Kuviosta 2 voidaan havaita, näkökenttä koostuu kuva-alueen leveydestä ja korkeudesta. Näihin vaikuttaa kameran etäisyys kuva-alueesta.



Kuvio 2. Kameran parametrit, jotka muodostavat sen näkökentän (Rantapuska 2014)

Kontrasti kuvaa, kuvan tummimman ja vaaleimman kohdan, valoisuuden eroa, ja sitä kuinka monta harmaatasoa näiden välissä on. (Introduction to Modulation Transfer Function n.d.)

Kuviossa 3, näkyvä kanttiaalto vaikuttaa yläpuolella olevien laatikoiden harmaasävyyn niin, että korkealla olevat luvut ovat vaaleampia ja alhaalla olevat luvut tummempia. Mitä enemmän luvut  $I_{Max}$  ja  $I_{Min}$  lähestyy toisiaan, sitä enemmän harmaasävyt ovat toisensa kaltaisia.



Kuvio 3. Kontrasti, jota kuvataan kanttiaaltona, ylhäällä sävy ja kanttiaallossa kontrastin vahvuus (Introduction to Modulation transfer Function n.d.)

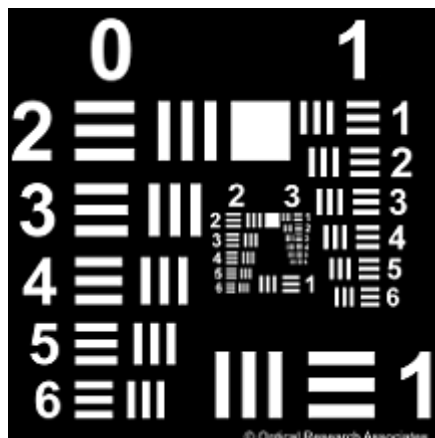
Kuten kuviosta 4 näkee, tämä luku vaikuttaa suoraan siihen paljonko informaatiota otetut kuvat sisältää, sillä luvun kasvaessa on kuvassa enemmän tietoa. Pienimmällä kontrastilla ei kuvassa ole kuin 2 tasoa, musta ja valkoinen, ilman erillisiä harmaita tasoja välissä.



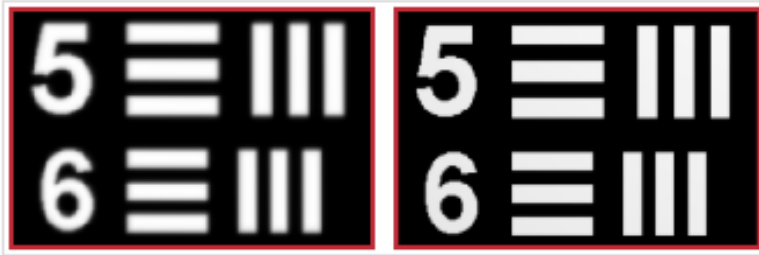
Kuvio 4. Harmaasävyjen määrää vaikuttaa kuvan informaatiomäärään (Rantapuska 2014)

Modulaationsiirtofunktio, on arvo siitä kuinka hyvin linssi voi siirtää kontrastia, tietyssä resoluutiassa, kohteesta kuvaan. (Introduction to Modulation Transfer Function n.d.)

Tämä voidaan määrittellä käyttämällä kuvaa 5, josta kameralla otetaan kuva, josta näkyy missä vaiheessa viivat alkavat mennä epäselviksi. Kuten kuvassa 6, vasemman puoleinen kuva numero 6 ja sen viivat ovat epäselkeämpiä kuin oikeanpuoleisessa kuvassa.



Kuvio 5. Modulaationsiirtofunktion selvittämiseen tarvittava testauskuva (Introduction to Modulation transfer Function n.d.)

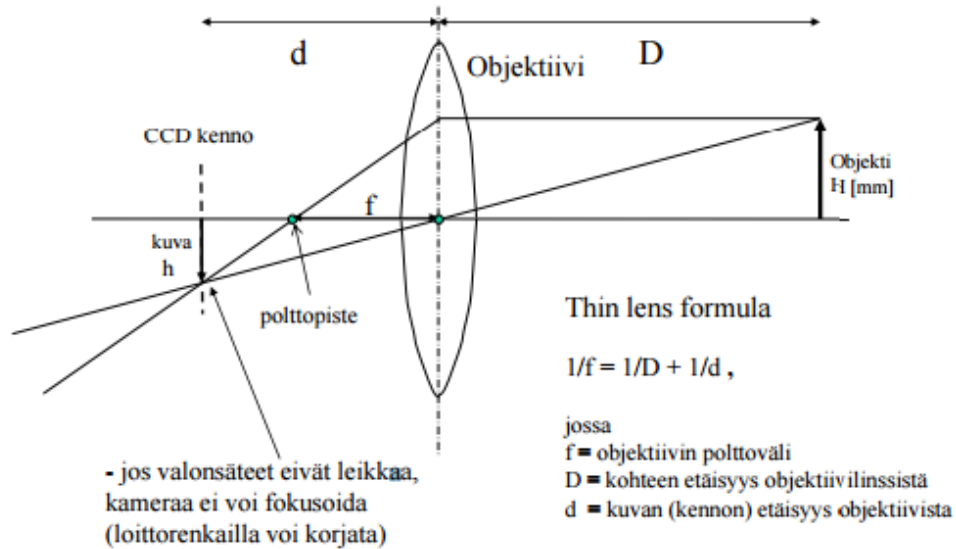


Kuvio 6. Kahdella eri linssillä otetut kuvat, oikean puoleisella kameralla on parempi modulaatiinsiirtofunktio (Introduction to Modulation transfer Function n.d.)

## 2.2 Optiikka

Optiikassa pitää ottaa huomioon polttoväli, syvyysterävyysalue ja suljinaukko. (Alroth 2010, 19).

Kuvio 7 selventää optiikkaa: Keskellä oleva soikio on kameran linssi. Oikealla puolella oleva nuoli ylöspäin kuvastaa objektin korkeutta millimetreinä. CCD-kennon ja linssin välinen viiva, joka on ainut linssin läpi menevä taittuva viiva, kuvastaa valonsädettä joka tulee objektiivin korkeimmasta kohdasta ja vielä yhdistyy CCD-kennoon. Tämän viivan keskellä olevan polttopisteen ja linssin keskellä oleva vihreä pisteen väli kuvastaa polttoväliä.

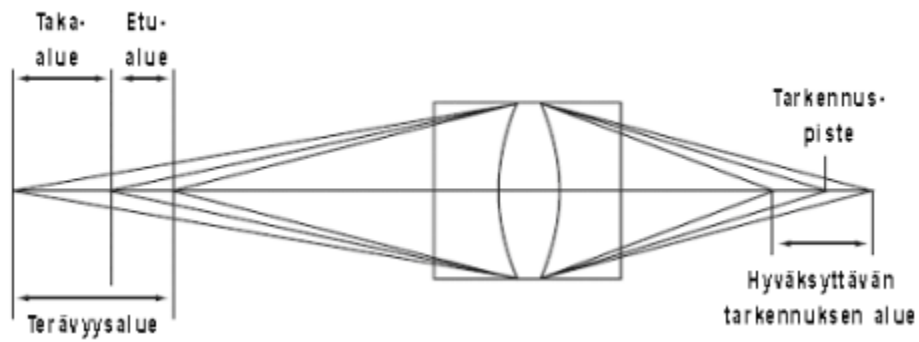


Kuvio 7. Valon käyttäytyminen objektiivissa (Rantapuska 2014)

Linssin polttoväli määräytyy kuvattavan sekä kohteen leveyden ja piteuden mukaan, että kuvakennon leveyden ja piteuden mukaan. Mitä pidempi linssin polttoväli on, sitä kapeampi on syvyysterävyysalue. Linssijärjestelmän polttoväliä, yleensä kuvataan millimetreinä. (Understanding Focal length n.d.)

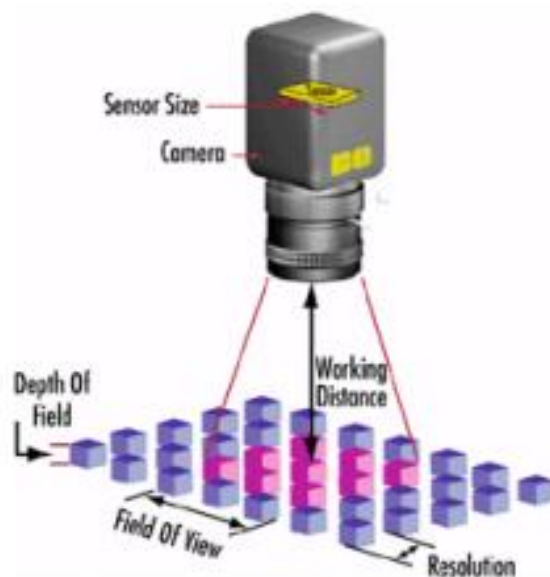
Syvyysterävyysalue on se alue, mihin optikka tarkentuu. Tämän alueen ulkopuolella olevat kohteet eivät kuvaudu terävästi. Kun kameras aukkoa pienennetään, syvyysterävyysalue vuorostaan kasvaa. (Alroth 2010, 20).

Kuviossa 8 selvennetään syvyysterävyysaluetta. Oikealla näkyvä tarkennuspiste ja sitä vastaava taka- ja etualueen välissä oleva viiva, näyttää sillä hetkellä tarkimmillaan olevan etäisyyden. Jos tarkennuspistettä siirretään toiseen suuntaan, tarkkuus muuttuu myös toiseen kohtaan. Näitä muutoksia rajoittaa tarkkuuden suhteen, terävyysalue ja tarkennuspisteen suhteen hyväksyttävän tarkennuksen alue.



Kuvio 8. Tarkennuspisteen vaikutus terävyysalueeseen (Alroth 2010, 20)

Kuviossa 9 näkyvät kaikki optiikan parametrit kokonaisuutena. Vaaleanpunaiset pikselit ovat näkökentässä (Field of View) ja tarkennettuja. Alhaalla olevien pikselien leveys kuvastaa resoluutiota eli sitä, miten tarkka kuvasta tulee (tässä tapauksessa 3 x 3 pikseliä). Samojen pikselien korkeus kuvastaa syvyysterävyysaluetta (Depth of Field) eli sitä, millä alueella kuvassa olevat kohteet voidaan tarkentaa polttoväliä muuttamalla.



Kuvio 9. Koko optiikkaa selventävä kuva (Rantapuska 2014)

Suljin säätelee, kuinka pitkän aikaa ja kuinka paljon valoa päästetään kohteesta kennon.

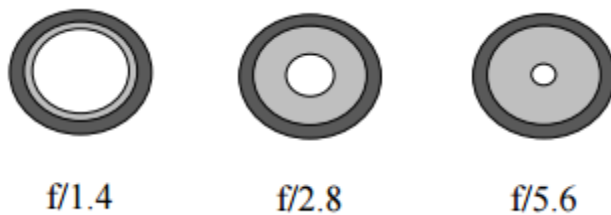
Aukon suuruus kuvaa suljinaukon kokoa suhteessa polttopisteeseen. Mitä suurempi aukko on, sitä enemmän valoa pääsee kennolle luomaan kuvaa. (Hirvisaari 2014.)

Kuvattaessa aukon suuruutta kuvataan f-numeroilla. F-numerossa "f" kuvastaa englannin kielisiä sanoja "focal length", joka kääntyy suomeksi kameran polttoväliksi.

Arvo lasketaan kaavalla:

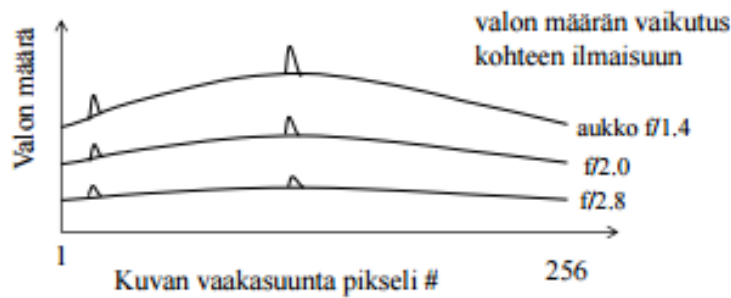
$$aukkano = \frac{\text{kameran polttoväli}}{\text{aukon halkaisija}} \quad (1)$$

eli mitä suurempi aukon halkaisija on, sitä pienempi aukko on. Tästä syystä pienempi numero tarkoittaa suurempaa aukkoa, kun taas suurempi numero tarkoittaa pienempää aukkoa, kuten kuviossa 10. (Karhulahti n.d.) Jos aukko on auki liian paljon, tulee kuvasta ylivalottunut. Aukon ollessa liian pieni jää suuri osa kuvasta aina pimeäksi.



Kuvio 10. Suljin aukkoja kuvaava kuva (Rantapuska 2014)

Kuviossa 11, näkyy, miten valon määrä kasvaa verrattuna aukon kokoon. Suurimmalla aukolla tulee korkeimmat valon määrät ja eniten hajautumaa valon määrään.

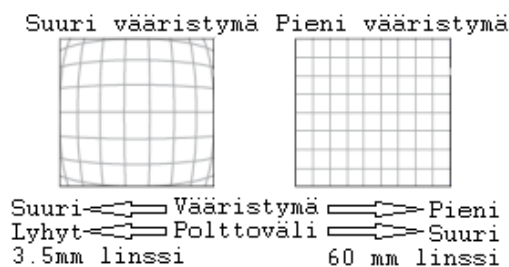


Kuvio 11. Valon määrää suhteessa kuvan vaakasuuntaisiin pikselien määrään (Rantapuska 2014)

Toinen parametri eli suljinaika on se aika, miten pitkään suljin pidetään auki. Myös tässä suurempi luku vastaa enemmän valotettua kuvaa. (Hirvisaari 2014.) Suljinaikaa joudutaan usein pitämään lyhyenä johtuen kuva-alueella olevasta liikkeestä. Jos suljinaika on pitkä, liike jää kuvaan, mutta haluttu kuva kohde jää piiloon.

Optiset vääristymät, ovat väistämättömiä. Kaikki linssit vääristävät kuvaa tietyn määrän. Vääristymät johtuvat valon taittumisen eroista linssin eri kohdissa. Eniten vääristymiä tulee linssin reunoilla. Tämä korostuu lyhyen polttovälin linsejä käyttäessä. Tunnetut vääristymät yleensä voidaan korjata erikoislinseillä tai ohjelmallisesti. (Alroth 2010, 21.)

Kuviosta 12 ilmenee, miten lyhyt polttoväli aiheuttaa suurempaa vääristymää linssissä.



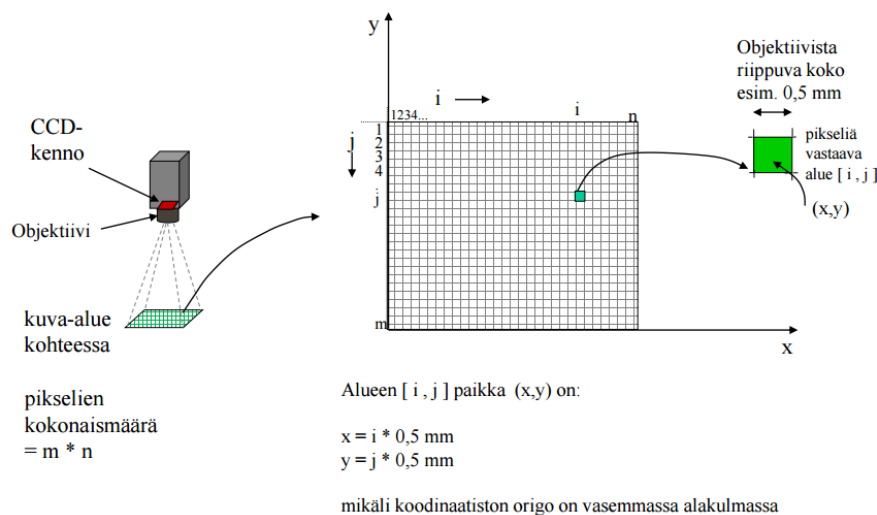
Kuvio 12. Polttovälin pituuden vaikutus kuvan vääristymään (Alroth 2010, 21, muokattu)

## Kameran kenno

Kameran kennot ovat joko CCD (Charge-Coupled Device)- tai CMOS (Complementary Metal-Oxide-Silicon)- kennoja (Rantapuska 2014).

CCD-kennot toimivat keräämällä valon irrottamia elektroneja pii-puolijohdeesta valoherkillä pikselialueilla. Kamera sitten muuntaa varauksen jännitteeksi ja siitä AD-muuntimella numeroarvoksi, joka vastaa pikseliin tullutta valonmäärää. Yhtä pikseliä vastaava alue riippuu objektiivista. (Rantapuska 2014)

Kuviossa 13 on esimerkki siitä, miten matriisikameran kuvat muodostuvat. Vasemmallalla nähdään miten kamera ottaa kuvaa alapuolellaan olevasta kuva-alueesta. Oikealla on sama kuva-alue, sieltä löytynyt vihreän pikselin varaus jää kuvaan. I ja j kuvastavat pikselien x ja y tietoa, esimerkissä yhden pikselin koko on 10  $\mu\text{m}$ .



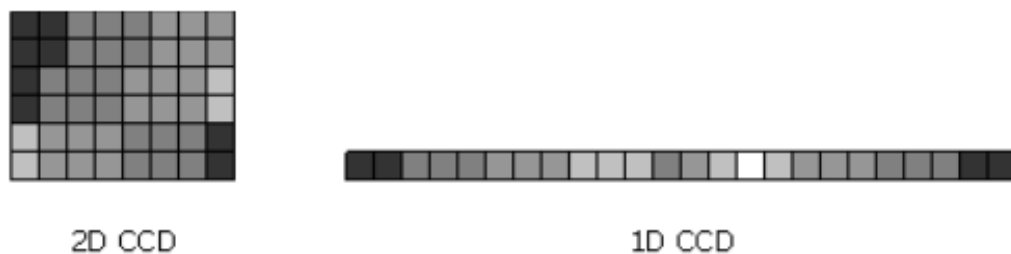
Kuvio 13. Esimerkki CCD-kennoisen matriisikamera toiminnasta (Rantapuska 2014)

CMOS-kennossa varaus muutetaan jännitteeksi erikseen joka pikselissä. Tästä johtuen varausta ei siirretä erikseen kennopiirillä ja CMOS-kennot yleensä kuluttavat vähemmän virtaa kuin CCD-kennot. (Hyyti 2014.)

## 2.3 Kameroiden ja linssien eroja

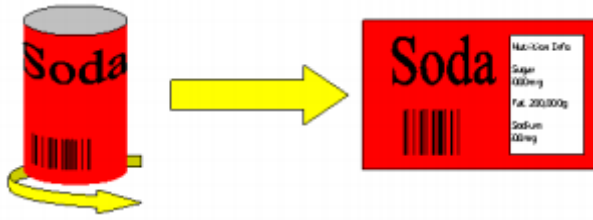
Matriisikameraa kutsutaan matriisiksi, koska sen ottama kuva muodostuu matriisiksi, johon tallentuu jokaiseen pikseliin kamerasta riippuva määrä tietoa. Harmaavärikameralla jokaiseen pikseliin tallennetaan binääriformaattissa tietoa, joka vastaa pikselin harmaasävyä. Matriisikamerat voivat yleensä ottaa 25 – 30 kuvaa sekunnissa, ja kameran valotusaika voi olla sitäkin lyhyempi. Lyhyt valotusaika voi olla tarpeellinen, jos kuvauskohde on liikkeessä. (Rantapuska 2014.)

Viivakameran kenno on viivamainen, siksi leveyssuunnassa on usein enemmän pikseleitä kuin matriisimuotoisessa kennossa. Tasainen valaistus on helpompi muodostaa viivakamerasovellukselle kuin matriisikamerasovellukselle, koska viivakameran kuvausalue on kapea (ks. kuvio 14). Viivakameran tyypilliset valotusajat vaihtelevat 12 mikrosekunnista 300 mikrosekuntiin. (Rantapuska 2014.)



Kuvio 14. Kaksi esimerkkikuvaa CCD-kennoista, oikealla viiva- ja vasemmalla matriisikamera (Alroth 2010, 26)

Viivakameraa voidaan käyttää pyörähdyskappaleen tai liikkuvan radan kuvaukseen. Tämä tapahtuu niin, että otetaan useita kuvia kohteen pyörähtäessä ja annetaan kameran tai tietokoneen yhdistää kuvat ennen kuvankäsittelyä. Esimerkkinä tästä on kuvio 15. (Rantapuska 2014)



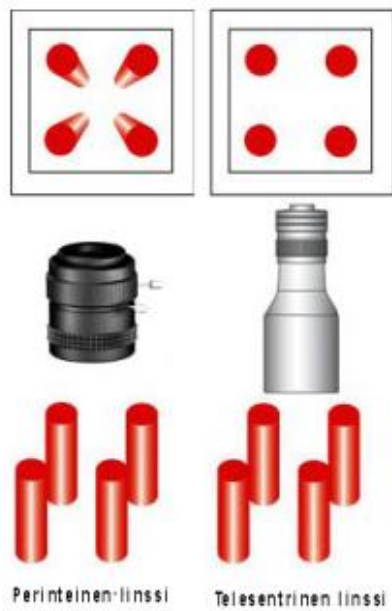
Kuvio 15. Pyörähdyskappaleesta saatu kuva (Alroth 2010, 27)

Optiikan tulokseen vaikuttavat myös mahdolliset erikoislinssit, joista seuraavassa esitellään vain pari yleisimpiin käyttökohteisiin tarkoitettua erikoislinssiä.

Perus konenäkölinssissä on kiinteä polttoväli, jolloin objektiivin polttoväli on aina sama. Tämä auttaa järjestelmän stabiiliudessa ja kuvien laadun varmistamisessa.

Zoom-linssissä polttovälin säätö on moottoriohjattu. Optiset ominaisuudet ovat usein heikompia kuin linseillä, joissa on kiinteä polttoväli. (Alroth 2010, 22.) Etuna zoom-linssissä on se, että polttoväliä ei tarvitse säätää käsin, jos polttoväliä täytyy vaihtaa usein, kun samaa kameraa käytetään moneen eri kohteeseen.

Telesentrisessä linssissä valonsäteet, jotka kohtaavat linssin kohtisuorassa, kohtaavat myös kennon kohtisuorassa ja mahdollistavat tarkat mittaukset poistamalla perspektiivin (Alroth 2010, 22). Kuten kuvio 16 ilmenee, telesentrinen linssi voi olla tarpeellinen, kun kuvataan kohteita, jotka ovat korkeita, ja halutaan nähdä ne ilman syvyyttä.



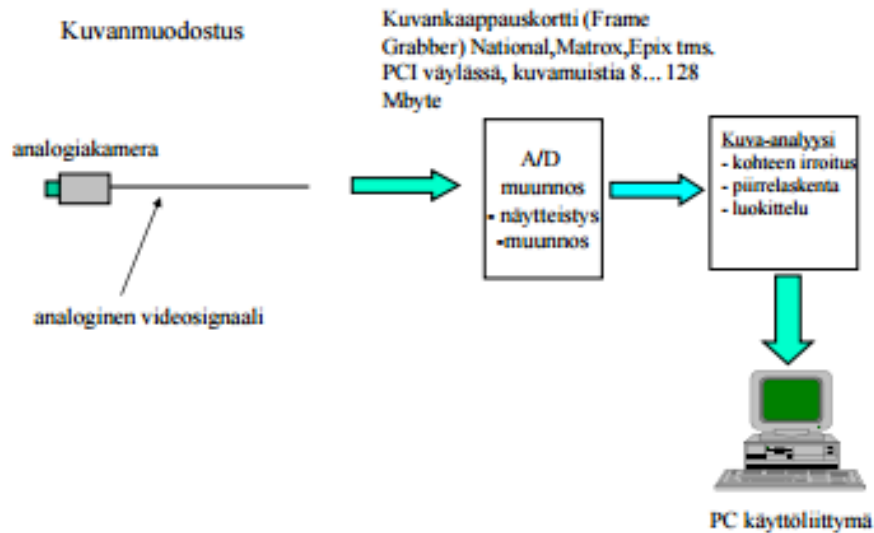
Kuvio 16. Telesentrinen linssi (oikealla) poistaa kuvasta syvyyden (Alroth 2010, 23)

Makrolinssi on optimoitu kuvaamaan kohdetta hyvin läheltä. Se näyttää hyvin tarkasti yksityiskohtia kuvauskohteesta. (Alroth 2010, 22.)

## 2.4 PC-pohjainen äly vs. älykäs kamera

PC-pohjainen äly tehdään konenäköjärjestelmällä. Tällöin järjestelmä koostuu tietokoneesta, kamerasta ja kuvankäsittelyohjelmasta. Ulkoisia liitäntöjä tällöin ovat esim. ethernet, I/O ja kenttäväylät. Etuna ovat PC:n laskentatehokkuus ja joustavuus. (Alroth 2010, 28.)

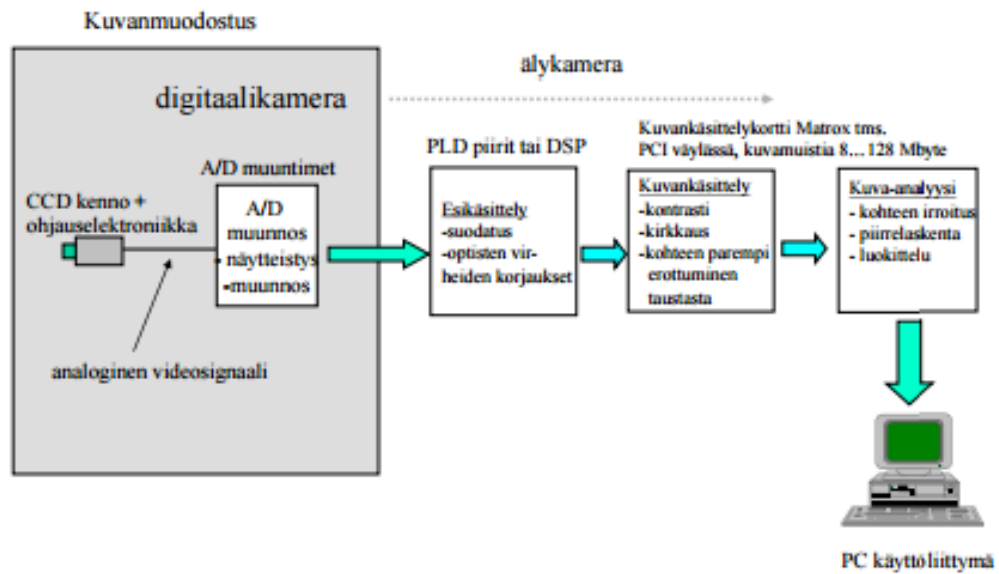
Kuviosta 17 voi nähdä, miten kuva muodostuu järjestelmässä, jossa on PC-pohjainen äly. Analogiakamera ottaa kuvan ja se siirretään analogisella videosignaalilla AD-muunnettavaksi. PC-käyttöliittymässä oleva ohjelma tekee sen jälkeen ohjelman määrittelemät kuva-analyysikäskyt.



Kuvio 17. PC-pohjainen äly (Rantapuska 2014)

Älykamera on itsenäinen ja kompakti laite, joka sisältää kuvankäsittelyohjelmiston, sulautetun kuvankäsittelyproessorin ja integroidun valaistuksen. Etuja älykameran käytössä ovat luotettavuus, yksinkertaisuus ja sovelluskehityksen nopeutus. (Alroth 2010, 29).

Kuten kuvio 18 ilmenee, älykamerassa kuvanmuodostus tapahtuu jo kameran sisällä. Digitaalikameran sisällä kuvalle tehdään AD-muunnos ja älykameran sisällä ohjelmoidut kuvankäsittelyt, ennen kuin kuva lähetetään tietokoneelle.



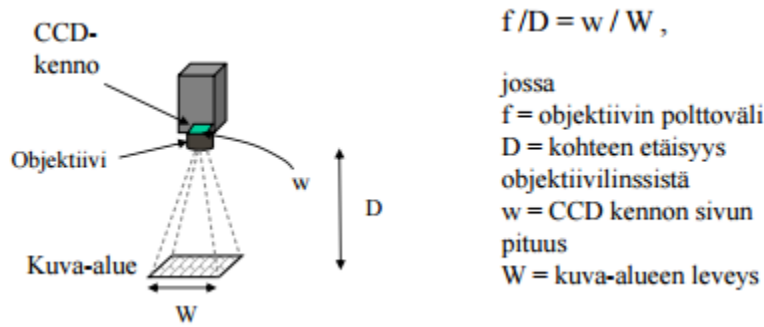
Kuvio 18. Älykameran kuvan muodostus (Rantapuska 2014)

## 2.5 Mitoitus

Konenäkösovelluksen optiikkaa mitoittaessa on hyvä aloittaa seuraavalla laskukaavalla (Rantapuska 2014).

$$\frac{\text{objektiivin polttoväli}}{\text{kohteen etäisyys}} = \frac{\text{CCD kennon koko}}{\text{kuva-alueen koko}} \quad (2)$$

Laskukaavan on helpompi hahmottaa kuvion 19 avulla.



Kuvio 19. Optiikan laskukaavaa hahmottava kuvio (Rantapuska 2014)

Kaavan perusta on se, että kuva muodostuu suhteellisen lähelle polttopistettä jääden vain vähän sen taakse. Lisäksi kohteen etäisyys on suurempi kuin objektiivin polttoväli. Jokaiseen koneeseen eri käyttökohteeseen on kuitenkin pyydettävä tarkempi laskenta kameran myyjältä, koska objektiivin valintaan vaikuttavat monet asiat kaavassa mainittujen lisäksi. Parhaimmillaan tarkkuus on yleensä 1 pikselin koko ja pahimmillaan 2 pikseliä. Kohteen liikkua tarkkuus huononee liikkeen suunnassa. (Rantapuska 2014.)

Kun ruvetaan kameran hankintaa miettimään, tiedetään halutun kuva-alueen leveys eli  $W$ . Myös kameran etäisyydestä pitäisi tässä vaiheessa olla jo ideaa eli tiedetään myös  $D$ , kohteen etäisyys linssistä. Kameraa valittaessa tarvitaan nämä tiedot ja sen lisäksi tulee päättää joko kameran objektiivin polttoväli tai CCD kennon sivun pituus. On parempi ensin päättää CCD kennon sivun pituus, koska polttoväliin voi vaikuttaa objektiivilinsillä.

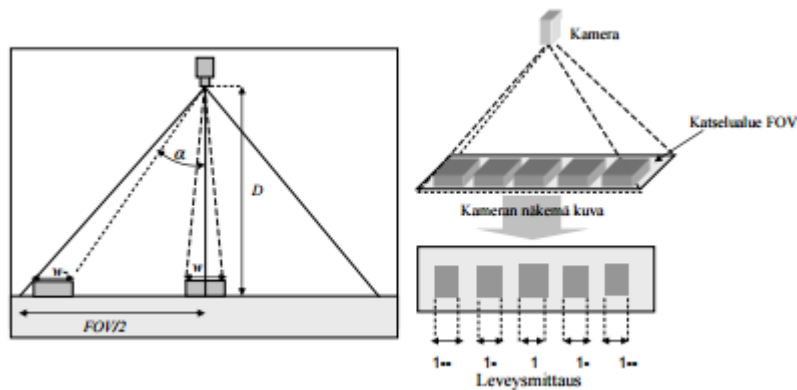
Esimerkiksi jos kuva-alueen leveys on 200 mm ja kamera asennetaan n. 700 mm:n päähän kohteesta, CCD-kennon sivun pituuden ollessa 4,83 saadaan polttoväliksi:

$$\left(\frac{X}{700}\right) = \left(\frac{4,83}{200}\right) \rightarrow z = 16,9\text{mm}$$

Tässä tapauksessa kannattaa järjestelmään ottaa 16 mm objektiivilinssi.

Aiemmin mainittujen asioiden lisäksi on huomioitava, että konekäyttösovelluksessa voi olla kulmavirhettä. Kun kamera on suuressa kulmassa kuvattavaan kohteeseen, luo se enemmän kulmavirhettä kuvaan. (Rantapuska 2014.)

Kuviossa 20 vasemman puoleisessa kuvassa, kulma  $\alpha$  on suoraan verrannollinen kulmavirheeseen. Oikean puoleisessa kuvassa sama asia hahmotetaan kameran näkökulmasta. Mitä lähempänä reunaan kohteet ovat, sitä enemmän on kulmavirhettä molemmilla puolilla.



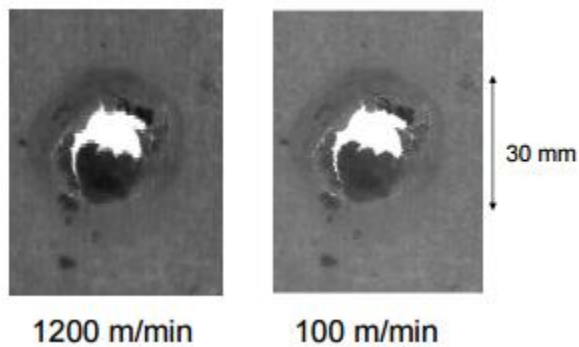
Kuvio 20. Kulmavirhettä selventävä kuva (Rantapuska 2014)

## 2.6 Valaistus

Valaistus on tärkeä osa konenäköä, sillä informaatio, jota kuvattavasta kohteesta saadaan, perustuu sen näkyviin piirteisiin. Huono valaistus johtuu yleensä joko yli- tai alivalaistuksesta, jolloin osa informaatiosta menee hukkaan. Tämä voi johtaa siihen, että konenäkö tarkistuksessa halutut piirteet eivät ole nähtävissä kuvaushetkellä ja tarkastus epäonnistuu. (Alroth 2010, 11.)

Valaistuksessa on huomioitava muut valaisimet, jotka valaisevat kohdetta, valon sijainti suhteessa kameraan, valon suuntaus, aallonpituus ja väri sekä konenäkösoveluksen vaatimukset ja valon käyttäytyminen kohteessa (Alroth 2010, 12). Kuten kuviossa 21 näkyy, huono valaistus voi myös johtaa pikselivenymään. Pikselivenymän aiheuttaa liian pitkä valotusaika kuvattaessa liikkuvaa kohdetta. Tästä syystä liikkuvassa kohteessa on hyvä käyttää Strobe-valoja valaistukseen.

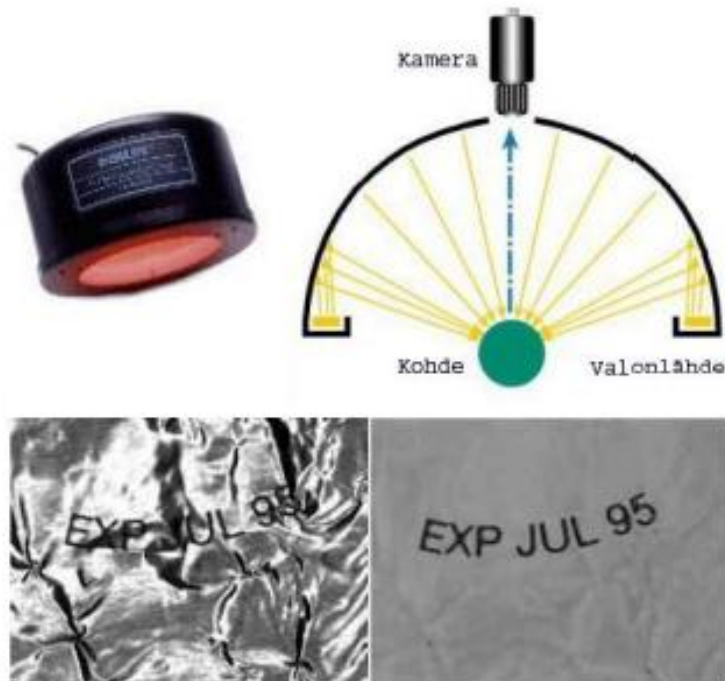
Strobovalot antavat käytännössä hyvin nopean valopulssin jokaisen otetun kuvan aikana, jotta nopeasti liikkuvakaan kohde ei liiku liian paljoa valotuksen aikana. Valotusajan ollessa liian pitkä aiheutuu kuvan otossa pikselivenymää. Strobovalojen käyttö voi helpottaa still-kuvan kaappaamisessa. (Rantapuska 2014.)



Kuvio 21. Esimerkki pikselivenymästä (Rantapuska 2014)

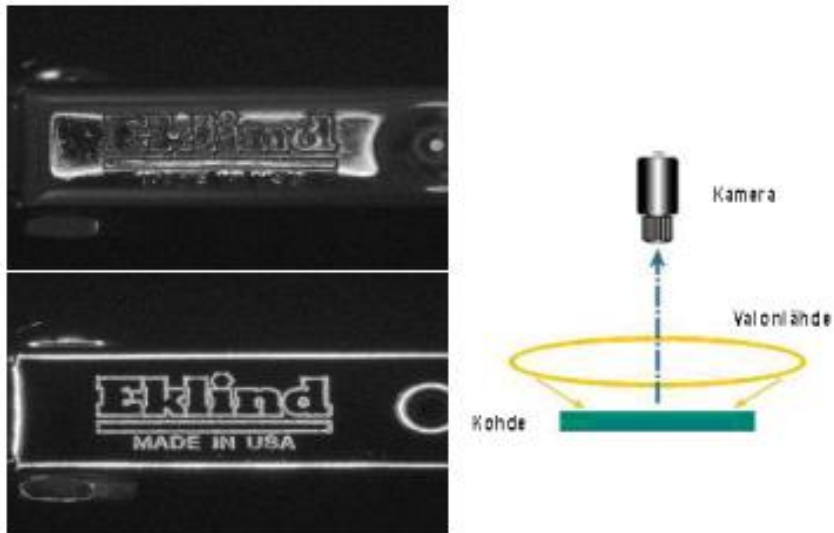
### **Esimerkkejä erilaisista erikoisvalaistuksista**

Dome-valaistus toimii hyvin voimakkaasti heijastavien pintojen tarkastukseen. Dome-valaistus tasoittaa pinnan heijastukset ja varjo hajauttaa valon voimakkaasti kohteeseen. (Alroth 2010, 13.) Kuten kuviossa 22 näkyy, Dome-valaistus tehdään kupolilla, jota valaistaan reunoilta. Kamera sijoitetaan suoraan kohteen yläpuolelle.



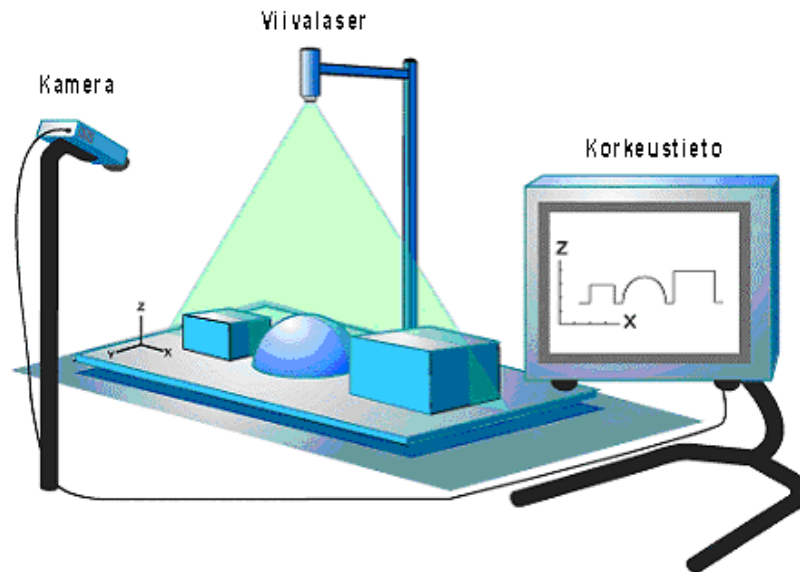
Kuvio 22. Ylhäällä kuva Dome-valaistuksen rakenteesta. Alapuolella oikealla puolella on kuva, jossa käytetty Dome-valaistusta, kun vasemmalla puolella on normi valaistuksella otettu kuva (Alroth 2010, 13)

Darkfield-valaistuksessa, poikkeamat pinnanmuodossa aiheuttavat kirkkaita heijastuksia kohteessa. Kamera sijoitetaan kohteen päälle ja valaistus toteutetaan pienessä tulokulmassa kohteen pintaan nähden. Tällöin vain pieni osa peiliheijastuneista säteistä saavuttaa kameran. (Alroth 2010, 14.) Kuten kuvion 23 oikeasta puoleisesta kuvasta näkee, valonlähde on kohteen ympärillä, niin että se valaisee kohteessa olevia poikkeamia pinnassa.



Kuvio 23. Esimerkki Darkfield-valaistuksesta (Alroth 2010, 14)

Rakenteellinen valaistus, auttaa hahmottamaan kappaleen muototiedon. Se tehdään yleensä laserilla. Valaistus syntyy kappaleeseen tunnetussa kulmassa heijastetusta valokuvion projektiosta. Valokuvio voi olla viiva, piste tai monimutkaisempi kuvio. (Alroth 2010, 15.) Kuten kuviosta 24 voi nähdä, rakenteellinen valaistus käyttää valaistuksena vain laseria, ja kamera kuvaa haluttuja kohteita niin, että kameraan rekisteröityy vain kohteiden kulmat.



Kuvio 24. Esimerkki Rakenteellisesta valaistuksesta (Alroth 2010, 15)

## 2.7 Kuvan käsittely

Käyn läpi tässä luvussa vain osan suuresta määrästä eri kuvankäsittelytapoja. Nämä mainitut kuvankäsittelytavat käydään kaikki läpi tässä projektissa.

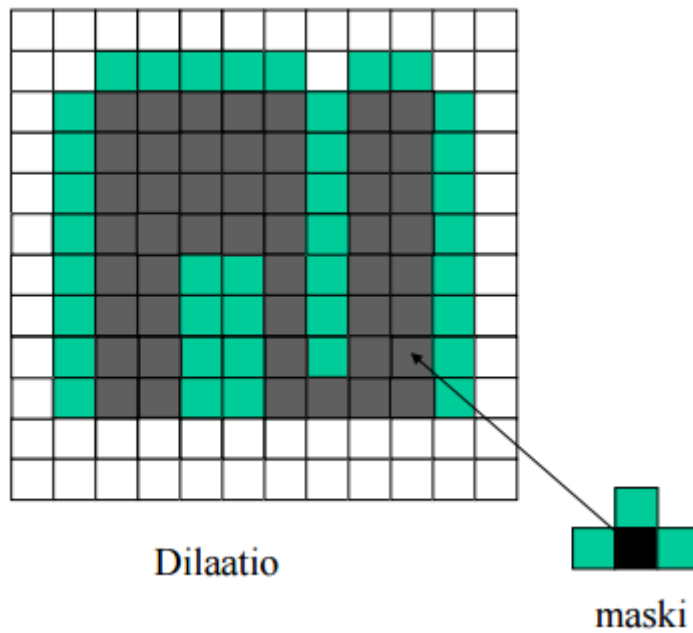
### Väri- tai harmaasävykuvasta mustavalkoiseksi

Turhan tiedon poistaminen kuvasta onnistuu esimerkiksi muutamalla värikuva mustavalkoiseksi. Tämä toimii valitsemalla kynnyisarvo, jota pienemmät arvot ja pikselit, jotka ovat vaaleampia kuin kynnyisarvo, muutetaan mustaksi ja kaikkien muiden pikselien arvot muutetaan valkoiseksi. Tämä vähentää tiedon määrää, koska jokaiseen pikseliin ei tarvitse varata enempää kuin yksi numero, joka vastaa väritietoa. Harmaasävykuvassa se, kuinka paljon jokainen pikseli tarvitsee tietoa, riippuu kuvan harmaasävymäärästä. (Azad, Gockel & Dillman 2008, 113.)

### Dilaatio eli laajennus

Kuvan mustavalkoistumisen jälkeen hajonneita oleellisia muotoja ja pikseleitä voidaan vahvistaa laajentamalla niitä. Tähän on olemassa yleinen kuvanmuodostusfunktio, jota kutsutaan nimellä "dilataatio". Se toimii niin, että käytetään maskia (kuvio 25), jonka origo asetetaan jokaisen mustan pikselin kohdalle laajentaen kaikkia kuvassa

olevia yksittäisiä ja toisiaan läheisiä pikseleitä. (Rantapuska 2014.) Kuviosta 26 näkyy, vasemman puoleinen kuva on käsittelemätön kuva ja vasemman puoleinen on laajennettu kuva.



Kuvio 25. Esimerkki maskista ja sen vaikutuksesta laajennuksen kohteeseen (Rantapuska 2014)



Kuvio 26. Esimerkki dilaatiosta (Rantapuska 2014)

### Reunanhaku, Sobel

Reunanhakufunktioilla on tarkoitus etsiä kuvasta kohta, jossa kuva alkaa. Näitä haku-toimintoja on monia, mutta minä valitsin opinnäytetyöhön Sobel-funktion.

$$h_1 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, h_2 = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & 1 & 0 \end{bmatrix}, h_3 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \dots$$

Sobel-funktiota käytetään usein reunan pystysuoraisuuden ja vaakasuoraisuuden tunnistamiseen. Tällöin käytetään vain maskeja  $h_1$  ja  $h_3$ . Jos  $h_1$ :n vastaus on  $y$  ja  $h_3$ :n vastaus on  $x$ , voidaan derivoida reunan vahvuutta kuin  $\sqrt{x^2 + y^2}$  tai  $|x| + |y|$  ja suuntana käyttää  $\arctan\left(\frac{y}{x}\right)$ . (Sonka, Hlavac & Boyle 2008, 136.)

### Hough muunnos linjan perusteella

Linjan perusteella toimivan Hough -muunnoksen pää tarkoitus on etsiä kuvasta pi-simmät linjat, niin että tämä tieto voidaan myöhemmin saada selville Hough -muunnoksesta.

Hough -muunnos on opinnäytetyön sovellukseen tarpeellinen, sillä työssä halutaan konenäkösovellus, jonka voi helposti asettaa monille eri mittareille. Kaikissa mittareissa ei ole samoja kiintopisteitä tai arvoja. Tästä syystä Hough -muunnoksen käyttö sovelluksessa vaikuttaa tarpeelliselta.

Hough -muunnos perustuu ideaan, että kuvan 2D-koordinaatit  $(u, v)$ , muutetaan parametriavaruuteen eli kuvitteelliseen tilaan, jossa on kaikki mahdolliset  $u$ :n ja  $v$ :n tulokset. Tästä parametriavaruudesta haluttu muoto pitäisi havaita. Tätä parametriavaruutta kutsutaan tässä tapauksessa Hough -avaruudeksi. Yleisesti yksi piste koordinaattiavaruudessa eli kuvan  $x, y$ -akselilla, vastaa loputonta määrää pisteitä Hough -avaruudessa. (Azad, Gockel & Dillman 2008, 123.)

Sopiva Hough -avaruus valitaan halutun tarkkuuden mukaan, ja jokainen piste tällä alueella ajatellaan kiinnityspisteinä. Jokaisesta käytettävän kuvan potentiaalisesta reunapikselistä tarkistetaan sopivien  $n_s$  kiinnityspisteiden määrä Hough -avaruudessa. Nämä tarkistetut mahdolliset kiinnityspisteiden määrät muistetaan ja

niitä verrataan toisiinsa. Tällä tavalla kiinnityspisteet, joilla on suuri määrä osumia eli ne jotka ovat luultavasti osa yhtä viivaa, korostuvat muiden joukosta. (Azad, Gockel & Dillman 2008, 123.) Tämä tapa toimii erittäin hyvin sekavilla kuvilla, mutta on suhteessa koneelle raskas, koska jokainen pistekoordinaatti avaruudessa ja suuri määrä pisteitä Hough -avaruudessa täytyy laskea. Todellinen laskennallinen työ riippuu Hough -avaruuden ulottuvuudesta. (Azad, Gockel & Dillman 2008, 123.)

Aluksi tutkitaan yleistä suoran viivan tunnistusta. Yksi mahdollinen tapa on ottaa tavanomainen suoran viivan laskentamalli, jossa kaltevuutta eli kulmaa x-akselin suhteen kuvataan muuttujalla  $m$  ja y-akselin leikkauspistettä kuvataan muuttujalla  $c$  (Azad, Gockel & Dillman 2008, 123):

$$v = m * u + c \quad (3)$$

Ratkaisemalla muuttuja  $c$  saadaan kaava, jolla lasketaan vastaavat pisteet yhdelle pikselille  $(u, v)$  kahden ulottuvuuden Hough -avaruudessa parametreilla  $m$  ja  $c$ :

$$c = -u * m + v \quad (4)$$

Tämän tavan huono puoli on se, että jos suoralla viivalla on loputon kaari  $m$ , sitä ei voida kuvastaa Hough -avaruudessa. Apuna voidaan käyttää toista Hough -muunnosta, jota kuvaa kaava 5.

$$u = m' * v + c \quad (5)$$

Tämä kaava poikkeaa kaavasta 3 niin, että muuttujalla  $m$  on heittomerkki. Tämä heittomerkki kuvaa sitä, että  $m$ -matriisi on käänteinen alkuperäiseen matriisiin verrattuna. Molempien Hough -muunnoksien tulokset pitäisi yhdistää, jotta voidaan laskea kaikki suorat linjat. Kaavalla 5 pystysuoralla viivalla ei ole loputonta kaarevuutta.

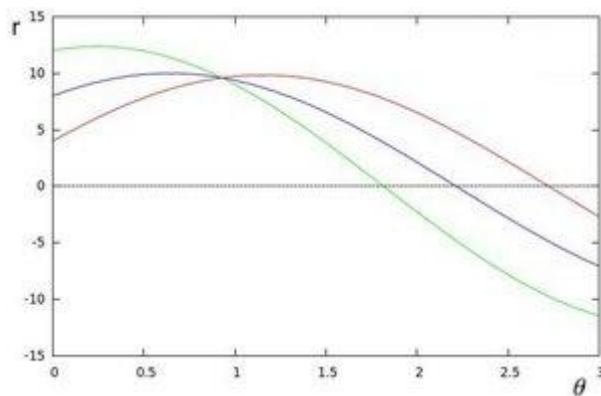
Hienovaraisempi tapa pohjautuu suoran linjan kaavalle, jonka apuna käytetään polaarisia koordinaatteja (Azad, Gockel & Dillman 2008, 123.)

$$r = u * \cos\theta + v * \sin\theta \quad (6)$$

Suora linjan määrittelee tällöin kulma  $\theta$  normaalista vektorista  $(\cos\theta \sin\theta)^T$  suoralla linjalla, jolla on  $u$ -akseli ja etumerkillinen etäisyys  $r$  alkupisteestä suoraan linjaan. Jokainen koordinaattipiste  $(u, v)$ , joka muutetaan Hough -avaruuteen. Tässä vaihees-

sa koordinaattipisteille tehdään prosessi, jossa laitetaan jatkuva ennalta asetettu joukko tietoa ryhmissä näihin koordinaattipisteisiin, jolloin saadaan erillinen numero kaikista mahdollisista koordinaattipisteiden prosessin asemista (discretization). Tämä tehdään kulmassa  $\theta$  välillä  $[0, \pi]$  ja vastaavien merkintöjen laskimia  $(\theta, r)$  Hough -avaruudessa kasvatetaan. (Azad, Gockel & Dillman 2008, 123.)

Nämä laskimet kuvastetaan Hough -muunnatussa kuvassa aaltoilevina viivoina. Useiden viivojen tavatessa samassa kohdassa rupeaa siinä kohdassa oleva arvo näkymään vahvempana kuin muut. Tällä tavalla katsotaan kuvan pisimmät viivat korostuvat Hough -muunnatussa kuvassa kirkkaampina pisteinä. Kuviossa 27 näemme Hough -muunnoksen tuloksen kuvalle, jossa on viivat  $L_0: x_0 = 8, y_0 = 6$ ,  $L_1: x_1 = 9, y_1 = 4$  sekä  $L_2: x_2 = 12, y_2 = 3$ . (Hough Line Transform n.d.)



Kuvio 27. Hough -muunnoksen esimerkki tulos (Hough Line Transform n.d)

Algoritmissä 1 Hough -muunnos suoralle viivalle, näytetään pseudokoodina, joka on väliaste ihmisen puhekielen ja tietokoneen ohjelmointikielen välillä. Käytettävä binäärikuva  $I$ , jolla on leveys  $w$  ja korkeus  $h$  on oletettu käyttäjän antamiksi tiedoiksi. Kuvasta  $I$  jokainen pikseli, jolla arvo  $q$ , on potentiaalinen reunapikseli. Algoritmista saadaan ulos 2-ulotteinen Hough -avaruus  $H \in N^{a,n}$ . Muuttuja  $a$  osoittaa discretizationin määrää kulmassa  $\theta \in [0, \pi)$ , ja  $n := 1 + 2 * \sqrt{w^2 + h^2}$ , discretizationin määrä  $r$ :lle, kun resoluutiona yksi pikseli. Tässä,  $\sqrt{w^2 + h^2}$  on suurin mahdollinen absoluut-



### 3 Opinnäytetyön ratkaisuvaihtoehdot

#### 3.1 AD-muunnin

AD-muuntimen nimi muodostuu sanoista Analog-to-Digital, ja sillä yleensä tarkoitetaan mikropiiriä, joka muuntaa analogista signaalia digitaalseksi, jotta tietokone tai mikroprosessori ymmärtäisi sitä. Kuviossa 28 AD-muuntimen piirikaaviosymboli (AD-muunnin n.d.)



Kuvio 28. AD-muuntimen piirikaaviosymboli, vasemmalla analoginen ja oikealla digitaalinen signaali (AD-muunnin n.d)

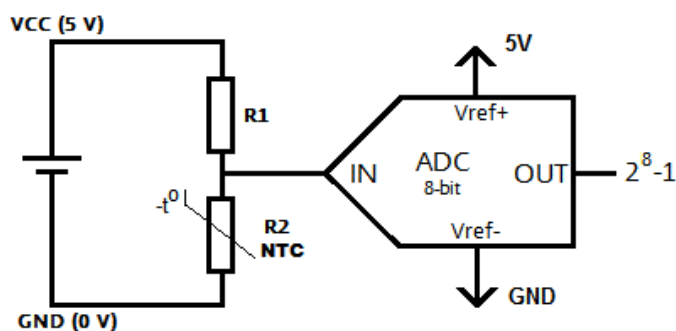
Tieto, jota muuntimelta luetaan, on muodossa  $2^N$ , joka vuorostaan kuvastaa kuinka tarkka kyseinen muunnin on. Tämä vastaa AD-muuntimen tarkkuutta, jota voidaan kutsua myös näytteenottosyvyydeksi. Resoluutio kuvaa, moneenko osaan analogijännite voidaan jakaa ja millä tarkkuudella mittatulos silloin saadaan esitettyä. (AD-muunnin n.d.)

Esimerkiksi 8-bittisen muuntimen tarkkuus on 256, sillä  $2^8 = 256$  - eli kahdeksan bitin avulla voidaan esittää 256 eri arvoa (AD-muunnin n.d.)

Toimintaan liittyy olennaisena osana referenssijännite (engl. Reference Voltage) eli, tieto mitkä ovat syötettävän signaalin minimi- ja maksimijännitteet. AD-muuntimeen syötettävän signaalin on oltava näiden lukujen sisäpuolella. Minimijännitettä kuvataan termillä  $V_{ref-}$  ja maksimijännitettä termillä  $V_{ref+}$ . (AD-muunnin n.d.)

Muunnin muuttaa täten siihen tulevan analogisen signaalin binäärimuotoiseksi dataksi suhteessa referenssjännitteisiin. Esim. kun sisääntuleva jännite on maksimi eli  $V_{ref+}$ , niin AD-muunnin antaa binäärisen arvon  $2^N-1$ . Kun sisääntulevan jännite on minimissään, niin binäärinen arvo on 0. (AD-muunnin n.d.)

Eli jos meillä on 8-bittinen AD-muunnin joka mittaa lämpötilaa (NTC-vastuksesta). Oletamme esimerkin vuoksi että NTC-vastus toimii lineaarisesti mittausalueella (, vaikka se ei oikeasti ole näin). Tästä liitännästä on esimerkki kuviossa 29. (AD-muunnin n.d.)



Kuvio 29. Esimerkki liitännästä,  $R1 = 10\text{ k}\Omega$  ja  $R2 = 5\text{ k}\Omega$ , resoluutio 8bittinen (AD-muunnin n.d)

Tässä tapauksessa AD-muunninta käytettäisiin liittämällä se analogiseen mittariin ja ottamalla siitä AD-muuntimen tulos tietokoneelle.

Tässä esimerkissä on käytössä 8-bittinen AD-muunnin, jolloin tarkkuus on (AD-muunnin n.d.)

Tarkkuus =  $2^{(\text{bit})} = 2^8 = 256$ , jossa ensimmäinen numero on 0 eli mittausalue on 0-255.

Kun referenssipiste  $V_{ref+}$  on kytketty viiteen volttiin ja  $V_{ref-}$  on kytketty maahan eli nollaan volttiin, jännitealue, jonka AD-muunnin voi mitata, on:

$V_{ref} = (V_{ref+}) - (V_{ref-})$ , josta saadaan:  $V_{ref} = 5\text{ V} - 0\text{ V} = 5\text{ V}$

Näin ollen AD-muuntimen jännitealue on 5 volttia. Tästä saadaan laskemalla selville se, kuinka suuri on yksi osa 256:sta, eli resoluutio voltteina:

$$\text{Resoluutio (V)} = V_{\text{ref}} / 2^N = 5 \text{ V} / 2^8 = 5 \text{ V} / 256 = 0,0195 \text{ V} \rightarrow \sim 19,5 \text{ mV}$$

Seuraavaksi analysoidaan tätä kytkentää. Oletetaan, että käytetyssä kytkennässä NTC-vastuksen lämpötilan ollessa 25 C astetta resistanssi on 5 k $\Omega$ . (AD-muunnin n.d.)

Täten AD-muuntimen kanavassa on jännite, joka voidaan laskea jännitteen jaosta kaavalla 4: (AD-muunnin n.d.)

$$V_{RI} = \frac{R_1}{R_1 + R_2} * V_{CC} \quad (4)$$

Käyttämällä kaavaa saadaan:

$$V_{R1} = \left( \frac{10}{10 + 5} \right) k\Omega * 5 \text{ V} = 3,33 \text{ V}$$

Nyt tiedetään, että vastuksen R2 yli jäämä on: 5 V - 3,33 V = 1,67 V. Arvo on sama kuin AD-muuntimen input-jännite  $V_{in}$ . Kun  $V_{in}$  on tiedossa, voidaan laskea miten paljon jännitettä saadaan AD-muuntimen lähdöstä: (AD-muunnin n.d.)

$$N = (2^8 - 1) * V_{in} / V_{\text{ref}} = 255 * 1,67 \text{ V} / 5 \text{ V} = 85,17 \rightarrow 85, \text{ joka binäärimuotoisena vastaa lukua } 1010101 \text{ (miinus yksi, koska laskenta lähtee liikkeelle nollasta).}$$

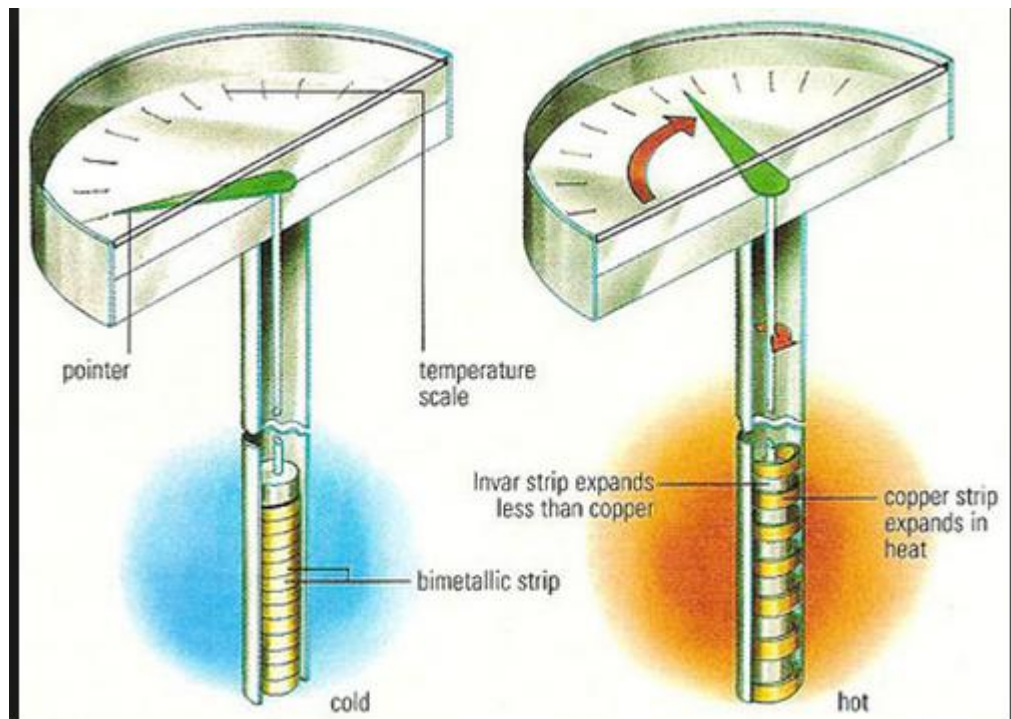
Voidaan siis todeta, että kun lämpötila on 25 astetta, niin kuviteltu AD-muunnin antaa luvun 85 ja anturilta saadaan 1,67 volttia jännitettä. Tällä tavalla voidaan halutessa AD-muuntimelle laskea myös kaikki muut mahdolliset jännitteet. (AD-muunnin n.d.)

Kun tiedetään käytetty referenssijännite ja AD-muuntimelta saatu muunnettu luku, mutta ei tiedetä kanavassa olevaa input-jännitettä, Se voidaan laskea edellisen esimerkin tapauksessa seuraavasti: (AD-muunnin n.d.)

$$V_{in} = N * (V_{\text{ref}} / (2^8 - 1)) = 85 * (5 \text{ V} / 255) = 1,666... \text{ V} \rightarrow 1,67 \text{ V}$$

Näin AD-muuntimelta saadaan laskettua tuntematon jännite, jota yleensä tarvitaan (AD-muunnin n.d.).

Opinnäytetyön yhden ratkaisuvaihtoehdon ideana oli että siirretään AD-muuntimella mittarista arvo tietokoneelle ja kone voi siitä skaalata mittarin arvon, jonka voi sen jälkeen julkistaa Internetiin. Tätä ei voitu toteuttaa, koska osa mittareista on sen tyyppisiä kuin kuviossa 30. Tämän tyyppisiin ei voida sähköjohdoilla kytkeä esiteltyä AD-muunninta.



Kuvio 30. Perus lämpömittari, jossa ei ole sähköliittimiä (Bimetallic strip thermometer n.d.)

### 3.2 Konenäkösovellus ja kamera

Ratkaisuvaihtoehto kaksi sisälsi sen, että otetaan joku konenäköohjelma ja tehdään sillä sovellus, joka katsoo kameran ottamasta kuvasta, missä kohdassa mittarin viisari on minä hyvänsä annettuna ajan hetkenä. Tämä toteutustapa vaatii, että valitaan ensin konenäköohjelma ja vaatimuksien mukaisen kameran. Myös valaistus ja asennus on otettava huomioon.

Tästä ratkaisuvaihtoehdosta löytyi eniten aiempia selvityksiä. Monet ihmiset eri puolilla maailmaa ovat saaneet samankaltaisen konenäkösovelluksen jo toimivaksi kotikäyttöön ohjelmoiden erilaisilla ohjelmilla ja kameroilla. Suurin osa löydöistä oli ikävä kyllä vain raportteja ja harvassa oli koodia sen teosta. Monet aiemmista selvitykset löytyivät erinäköisiltä foorumeilta. Vain yksi oli virallisemmän näköinen ja sitä myöhemmin rupesinkin käyttämään oman työni pohjana.

Konenäkösovelluksen on tarkoitus tässä PC-älyisessä järjestelmässä aluksi käsitellä kuvaa, jotta kuvasta saadaan helposti tarpeellinen tieto esille. Sen jälkeen muilla saman konenäkösovelluksen osilla selvittää tämä tieto kuvasta.

### **3.3 Mittarin vaihto**

Ratkaisuvaihtoehto 3 tarkoitti sitä, että vanha digitaalinen mittari poistetaan kokonaan ja sen tilalle asetetaan uusi digitaalinen mittari, joka voidaan kytkeä tietokoneeseen ja sitten lähettää siitä ja sille tietoa.

Mittarin vaihto on helppo tapa ratkaista ongelma, joka tässä on kyseessä, mutta se ei sopinut käyttötarkoitukseen. Jokaiselle kohteelle pitäisi etsiä uusi mittari, ja hinta nousee mittarien määrän mukaan nopeasti hyvin korkeaksi.

## **4 Opinnäytetyön toiminnalliset valinnat**

### **4.1 Opinnäytetyön tekotapa**

Luvussa 3 esitetyistä vaihtoehdoista valitsin analogisen mittarin digitalisointi tarkoitukseen konenäkösovelluksen, jota käytetään webkameran kanssa. Ratkaisu toimii kaikkien mittarityyppien kanssa, ja Internetistä löytyy tietoa, että siinä on jo onnistuttu. Sen lisäksi konenäkösovellusta on helppo muokata lisää ja kehittää sekä kameralle voidaan 3d-tulostimella tulostaa jalusta jokaista kohdetta varten erikseen. Prosolvella on tällainen tulostin.

Konenäköohjelmia löytyi Internetistä monia (ks. taulukko 1 luvussa 4.2). Kamerana riittää perinteinen web-kamera, koska kuvanottotiheyden ei tarvitse olla nopea.

Google haulla löydettiin useita raportteja vastaavista onnistuneista projekteista, ja jokainen oli hieman toisestaan erilaisia, vain Kenn Sebestan (n.d) projektissa oli ohjelmakoodi esillä.

## 4.2 Konenäköohjelma

Opinnäytetyön tekoon tutkin erilaisia konenäköohjelmia ja vertasin lähinnä niiden hintoja, sillä kaikki ohjelmat toimivat periaatteessa samalla tavalla.

Google-haulla löytyi useita eri konenäkösovelluksia (ks. taulukko 1), mutta harvoilla oli hinta näkyvillä. Valmistajat toivoivat, että heihin otetaan yhteyttä, jos hinta halutaan tietää. Kuitenkin monilla oli ilmaisia kokeiluajoja, mikä auttaa oikean ohjelman valinnassa. Lopulta valitsin ohjelmista GNU Octaven tämän opinnäytetyön tekemiseen.

Taulukko 1. Konenäköohjelmat

Valmistaja	Hinta	Verkkosivu
Microscan	Ei hintaa näkyvillä	<a href="http://www.microscan.com/en-us/products/machine-vision-systems/visionscape-machine-vision-software.aspx">http://www.microscan.com/en-us/products/machine-vision-systems/visionscape-machine-vision-software.aspx</a>
Matrox	Ei hintaa näkyvillä	<a href="http://www.matrox.com/imaging/en/products/software/design-assistant-machine-vision-software/order/">http://www.matrox.com/imaging/en/products/software/design-assistant-machine-vision-software/order/</a>
Cognex	Ei hintaa näkyvillä	<a href="http://www.cognex.com/products/machine-vision/visionpro-vision-software/">http://www.cognex.com/products/machine-vision/visionpro-vision-software/</a>
Datalogic	Ei hintaa näkyvillä	<a href="http://www.datalogic.com/eng/products/industrial-automation/machine-vision/impact-software-pd-605.html">http://www.datalogic.com/eng/products/industrial-automation/machine-vision/impact-software-pd-605.html</a>
Roborealm	159,95 €	<a href="http://www.roborealm.com/">http://www.roborealm.com/</a>
Teledyne	Ei hintaa näkyvillä	<a href="https://www.teledynedalsa.com/imaging/products/vision-systems/software/sherlock/">https://www.teledynedalsa.com/imaging/products/vision-systems/software/sherlock/</a>
Adaptive Vision	Ei hintaa näkyvillä	<a href="https://adaptive-vision.com/en/user_area/order_license/">https://adaptive-vision.com/en/user_area/order_license/</a>
MVTec	Ei hintaa näkyvillä	<a href="http://www.halcon.com/halcon/sales/">http://www.halcon.com/halcon/sales/</a>
TotalTurnKey	Ei hintaa näkyvillä	<a href="http://www.euresys.com/Distributors/Europe/EuropeMiddle.asp#Finland">http://www.euresys.com/Distributors/Europe/EuropeMiddle.asp#Finland</a>
Välittäjä	Ei hintoja näkyvillä	<a href="http://www.metric.fi/tuotteet/konenako-anturit-ja-lasermerkinta/konenakoratkaisut/konenakoohjelmistot/">http://www.metric.fi/tuotteet/konenako-anturit-ja-lasermerkinta/konenakoratkaisut/konenakoohjelmistot/</a>
NeuroCheck	Ei hintaa näkyvillä	<a href="http://www.neurocheck.com/template.php?MNID=2&amp;SID=5&amp;lid=31">http://www.neurocheck.com/template.php?MNID=2&amp;SID=5&amp;lid=31</a>
Aqsense	Ei hintaa näkyvillä	<a href="http://www.aqsense.com/products/3dexpress.html">http://www.aqsense.com/products/3dexpress.html</a>
Ni vision	2070-	<a href="http://www.ni.com/vision/software/">http://www.ni.com/vision/software/</a>

	3820€/vuosi	
GNU Octave	Ilmainen	<a href="https://www.gnu.org/software/octave/">https://www.gnu.org/software/octave/</a>

### 4.3 Kamera

Web-kameroiden valikoima on myös laaja Internetistä (Taulukko 2), mutta en valinnut niistä mitään, sillä Prosolvelta löytyi sopiva demo-tarkoitusta varten.

Lopulta tärkeää kameran valinnassa on vain se, että kuvanottotaajuus on tarpeeksi nopea ja kuvan laatu läheltä, kun polttoväli on asetettu, on tarpeeksi hyvä.

Taulukko 2. Web-kamerat

Valmistaja	Hinta	Verkkosivu
NetAnttila	29,95	<a href="http://www.netanttila.com/shop/fi/netanttila/web-kamerat/1237445—malli">http://www.netanttila.com/shop/fi/netanttila/web-kamerat/1237445—malli</a>
Expert	29	<a href="http://www.expert.fi/Tuotteet/Tietokoneet/Oheistuotteet/Webkamerat/(T3H-00012)-HD3000">http://www.expert.fi/Tuotteet/Tietokoneet/Oheistuotteet/Webkamerat/(T3H-00012)-HD3000</a>
Polarpanda	13,9	<a href="http://www.polarpanda.com/pp_fi/gadgets-electronics/cameras/clip-webcam-mikrofonilla.html">http://www.polarpanda.com/pp_fi/gadgets-electronics/cameras/clip-webcam-mikrofonilla.html</a>
Clas Ohlson	14,99	<a href="http://www.clasohlson.com/fi/Web-kamera/38-4754#moreinfo">http://www.clasohlson.com/fi/Web-kamera/38-4754#moreinfo</a>
Gigantti	19,95	<a href="http://www.gigantti.fi/product/tietokoneet/tietokonetarvikkeet/LWEBC/logik-1-3-mp-web-kamera-lw12mp11">http://www.gigantti.fi/product/tietokoneet/tietokonetarvikkeet/LWEBC/logik-1-3-mp-web-kamera-lw12mp11</a>
Biltema	5	<a href="http://www.biltema.fi/fi/Toimisto---Tekniikka/Tietokoneet/Multimedia/Web-kamera-24749/">http://www.biltema.fi/fi/Toimisto---Tekniikka/Tietokoneet/Multimedia/Web-kamera-24749/</a>
CDON	15,95	<a href="http://cdon.fi/kodin_elektroniikka/trust/trust-spotlight-webcam-pro-p10783732">http://cdon.fi/kodin_elektroniikka/trust/trust-spotlight-webcam-pro-p10783732</a>
Motonet	7,96	<a href="http://www.motonet.fi/fi/tuote/476475/Maxell-Ball-Webcam-kamera">http://www.motonet.fi/fi/tuote/476475/Maxell-Ball-Webcam-kamera</a>
Prisma	29,95	<a href="http://www.prisma.fi/fi/prisma/microsoft-lifecam-hd-3000-webkamera">http://www.prisma.fi/fi/prisma/microsoft-lifecam-hd-3000-webkamera</a>
Ostanetistä	6,19	<a href="http://ostanetista.net/web-kamera">http://ostanetista.net/web-kamera</a>
Logitech	19,95	<a href="http://www.logitech.com/fi-fi/product/webcam-c170?crd=34">http://www.logitech.com/fi-fi/product/webcam-c170?crd=34</a>
Matkantalo	14,95	<a href="http://www.markantalo.fi/product/tietotekniikka/tietokonetarvikkeet/LWEBC/logik-1-3-mp-web-kamera-lw12mp11">http://www.markantalo.fi/product/tietotekniikka/tietokonetarvikkeet/LWEBC/logik-1-3-mp-web-kamera-lw12mp11</a>
e.ville	6,95	<a href="http://www.e-ville.com/fi/39-nettikamerat/2935-nettikamera-2mpix.html">http://www.e-ville.com/fi/39-nettikamerat/2935-nettikamera-2mpix.html</a>
hintafi	3460,00	<a href="http://hintafi.fi/g108/webkamerat">http://hintafi.fi/g108/webkamerat</a>

## 5 Toteutus

### 5.1 Konenäkö

Analogisen mittarin tiedon digitalisoimiseen käytettävän konenäkösovelluksen alustava idea on se, että analogisesta mittarista otetaan kuva tasaisin väliajoin ja tämä kuva sitten asetetaan konenäkösovellukseen, josta digitaalinen arvo lähetetään Internetiin. Tämän digitaalisen arvon perusteella konenäkösovellus voi tehdä muutoksia ja arvoa voidaan seurata myös etäältä.

Aloitin sovelluksen teon sillä, että opiskelin Octaven ohjelmoinnin perusteita sekä luin sen ohjekirjaa ja katsoin ohjevideoita Youtubesta.

Kun minulla oli kohtalainen alustava käsitys Octaven ohjelmointikielestä, jatkoin konenäkösovelluksen luomista Kenn Sebestan (n.d) tekemän Matlab-sovelluksen pohjalta.

Aloitin ensin etsimällä, miten saan kuvan sovellukseen ja sen jälkeen käsiteltä kuvasta mahdollisimman selkeän. Aluksi kokeilin ohjelman perustoimintoja, mutta pian huomasin, että tarvitsen Octave Forgen paketit, ohjelman laajan listan lisäosia, jossa on lisää kuvankäsittelytoimintoja. Tämän löysin ohjelman vanhemmasta versiosta.

Tässä vaiheessa siis oli siis selvinnyt, että

```
Image = imread('C:/.../'haluttu_kuva.jpg');
```

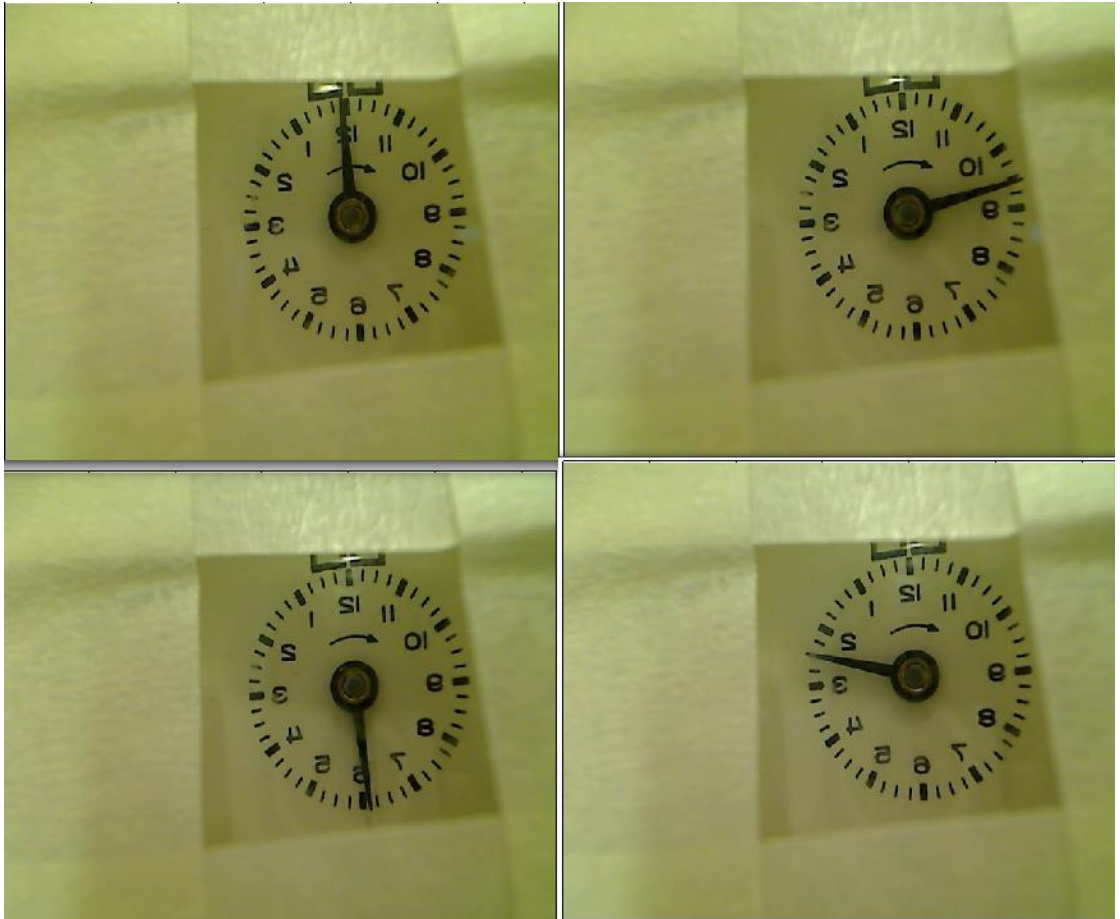
tallentaa Image-nimiseen muuttujaan 'haluttu\_kuva.jpg' kuvan ohjelmaan sovelluksen käsittelyä varten.

Kuten kuviossa 31 näkyy, käytin sovelluksen testaukseen hyvin yksinkertaista "mittaria", jolla oli tarkoitus vain testata, miten sovellus saadaan toimimaan. Tämä "mittari", jota testikuviin käytettiin, oli vanha kello, jonka viisarin asemaa voidaan muuttaa. Se toimi testaustarkoitukseen, sillä siitä voi vaihtaa viisarin asemaa ja tarkistaa toimiiko sovellus.



Kuvio 31. Herätyskello joka toimi testimittarina

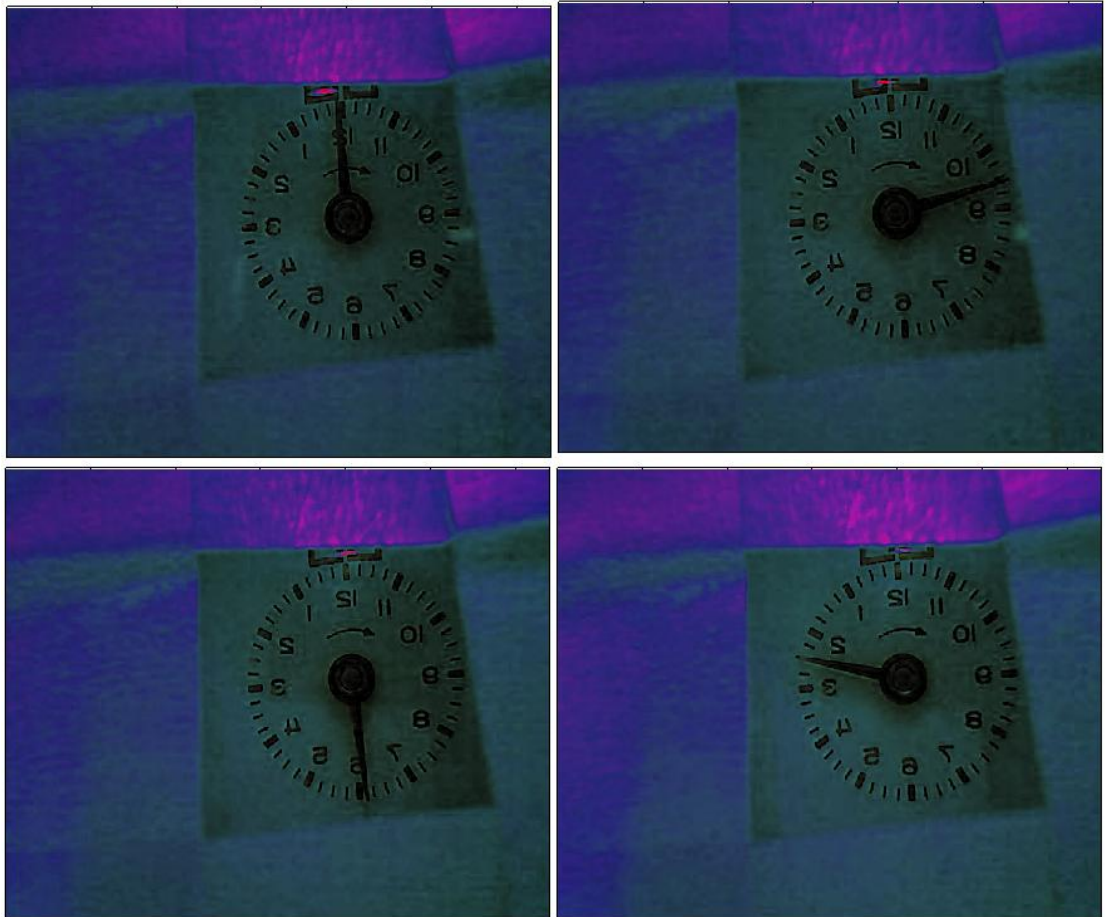
Päätin kuitenkin käyttää kellon yläosassa olevaa yhdellä viisarilla toimivaa osiota, itse testaukseen. Siitä syystä että, pienemmällä alueella on helpompi tässä toimisto valaistuksessa välttää valovirhettä ja kuvassa on vähemmän mahdollisia häiriöitä aiheuttavia asioita. Kuten kuvio 32 voi havaita, olen teipillä rajoittanut muita kuvaan muuten tulevia mahdollisia häiriön aiheuttajia.



Kuvio 32. Aloitusvaiheen testikuvat

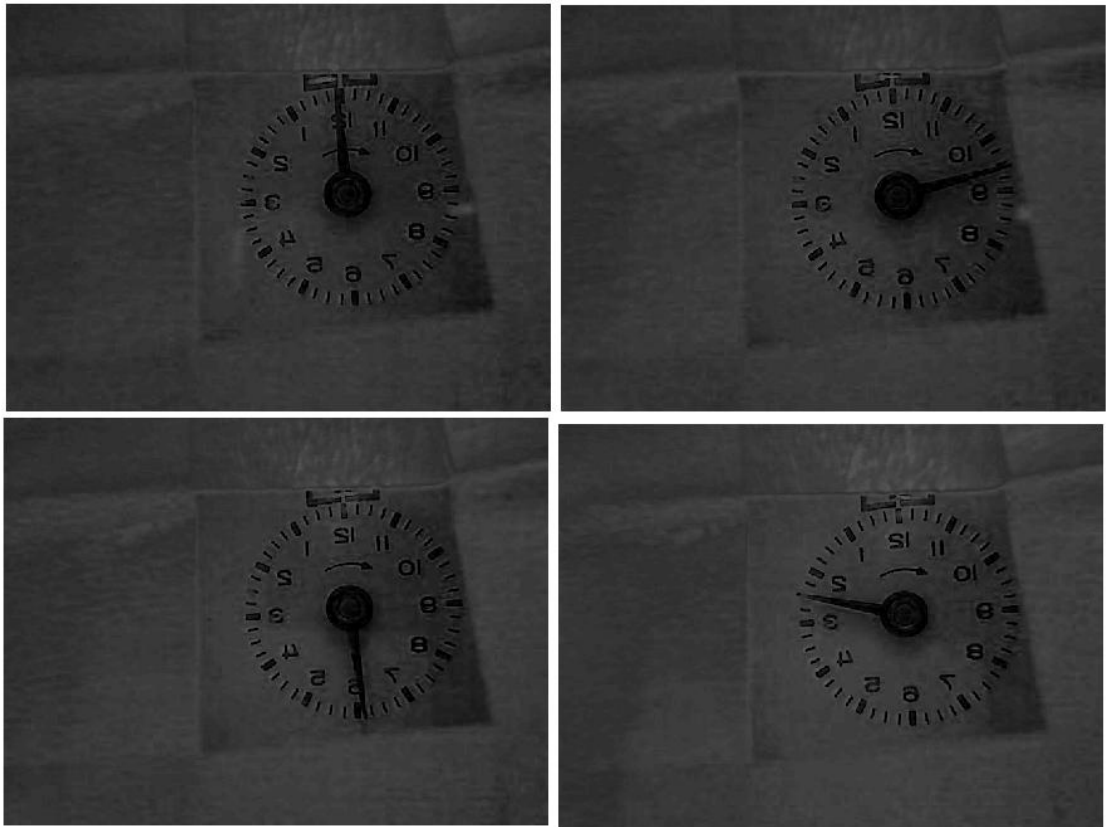
Tämän jälkeen otin kuvan ja muutin sen aluksi HSV-kuvasta (eli Hue, Saturation ja Value eli sävy, kylläisyys ja kirkkaus) RGB kuvaksi (red, green ja blue eli puna-vihreä-sinikuva) ja siitä vielä grayscale- eli harmaasävykuvaksi. Tämän kuvan lopulta muun-  
sin mustavalkoiseksi kuvaksi im2bw-funktiolla.

Kuten kuviosta 33 näkyy, RGB-kuvan värit ovat vääristyneet. Tämä johtuu siitä, että alkuperäinen kuva ei ollut HSV-kuva, kuten aluksi oletin.



Kuvio 33. HSV-kuvasta RGB-kuvaksi muunnetut kuvat

Kuviosta 34 näkyy, miten RGB-kuvasta harmaasävykuvaksi muutetussa kuvassa mitareiden oikealle alapuolelle jää tummempi kohta. Tämä tumma kohta johtuu väärin-tyneistä väritiedoista HSV-RGB-muunnoksessa.



Kuvio 34. RGB kuvasta harmaasävykuvaksi muunnetut kuvat

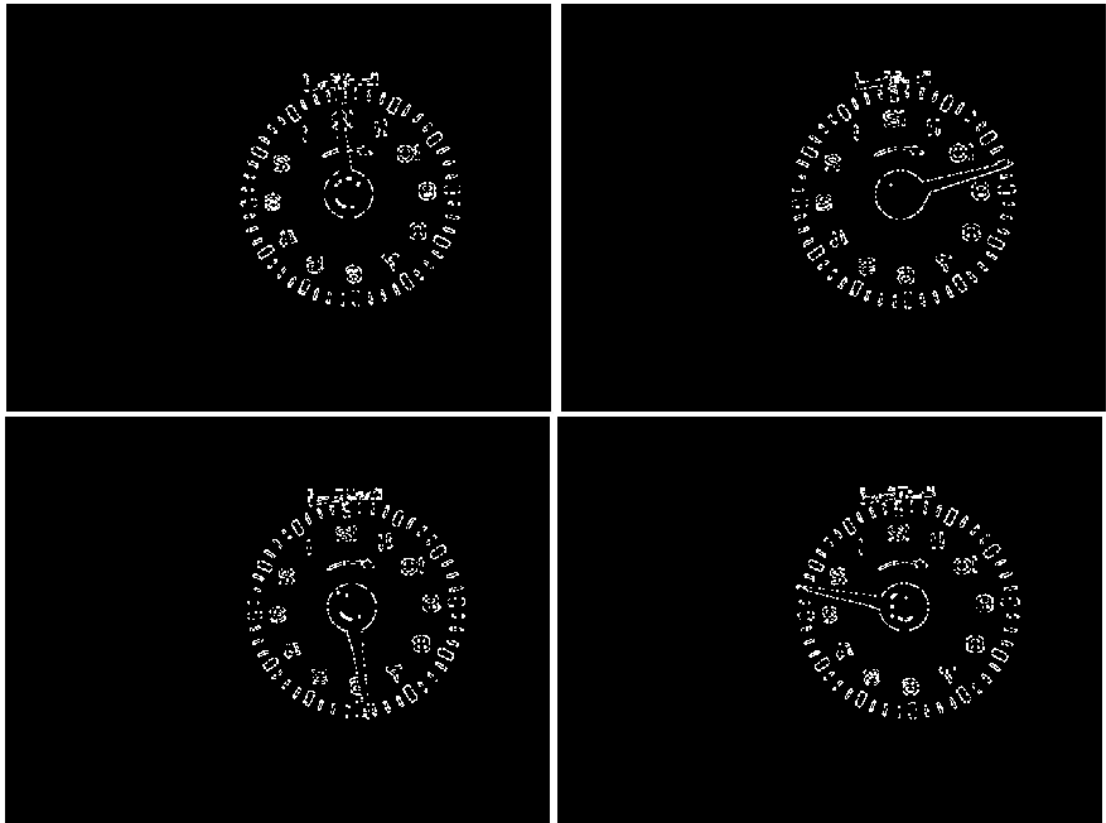
Kuten kuviossa 35 vasemmassa yläkulmassa olevassa kuvassa näkyy mittarin oikealla alapuolella, kuvissa on virheitä. Tästä syystä kokeilin eri bwmorph-funktion osia löytääkseni, mikä niistä on paras kyseisen virheen poistamiseen.



Kuvio 35. Grayscale-kuvasta BW-kuvaksi muutettu kuva

Lopulta huomasi, että `im2bw`-funktiota voi käyttää suoraan ohjelmaan lataamani kuvaan, minkä jälkeen kuvaan ei jäänyt samoja virheitä, kuten kuviossa 35 näkyy.

Kuten kuviossa 36 näkyy, nyt kuviin ei jää virhettä, joka tuli, kun kuva ensin muutettiin harmaasävykuvaksi. Kuvista myös ilmenee että tekemäni binäärikuvasta muutos tehdään reunanhakua hyväksikäyttäen, tästä syystä ei numeroiden sisäosat ole täysin valkoisia.



Kuvio 36. Suoraan alkuperäisestä kuvasta `im2bw`-funktiolla muutettu binäärikuva

Tämän jälkeen rupesin käyttämään Kenn Sebastan (n.d) tekemää koodia, ja rupesin tutkimaan Hough -muunnosta. Octavessa ei kuitenkaan ole kaikkia samoja muunnoksia, mitä Matlabista löytyy. Pienen hakemisen jälkeen huomasin, että uusimmalla Octaven versiolla 3.8.2, johon on Octaven paketit ladattuna, on olemassa enemmän Hough -muunnoksien versioita, mutta ei niitä, mitä minä tarvitsin, joten kokeilin uudenlaista lähestymistapaa.

Aluksi yritin ymmärtää Kenn Sebastan (n.d) koodia, mikä osoittautui haastavaksi. Kenn Sebasta ei ole kommentoinut koodiaan hyvin, ja välillä hän kirjoitti saman asian kahteen kertaan ilman hyvää syytä. Lopulta otin hänen koodistaan kaiken sen, mikä vaikutti oleelliselta. Toiminnan takaamiseksi latsin eri Matlabin Hough -muunnoksia Octaven käytettäväksi. Näitä jouduin kuitenkin muuttamaan hieman, jotta koodi toimi Octaven kanssa.

Tällä tavalla sain jossain määrin toimimaan dial\_indicator -funktionin (ks. liite 2) ja dial\_reader -funktionin (ks. liite 4).

Aloittaessani työskentelyn alkuperäisen background\_generator osion parissa (ks. liite 8) tuli vastaan uusi ongelma.

Tässä vaiheessa olin sovelluksen käyttänyt sekä Windowsilla Octaven versiota 3.6.4, että Octaven versiota 3.8.2. Octaven version 3.6.4 lataustiedon mukana tuli kaikki paketit, eli ylimääräisiä eri tarkoituksiin sisältyviä toimintoja sisältävät tiedostot. Ko-keilin toimintaa myös Linuxilla, jotta saisin video paketin toimimaan, mutta Image -pakettia ei ole päivitetty useaan vuoteen ja tästä syystä sitä ei saa toimimaan Linuxilla. Näissä myös kaipaamani videopaketti, jota en lopulta voinut sitä käyttää, sillä tarvitsin enemmän image-paketin uusinta versiota, ja sen lataamiseen tarvitsee olla vähintään Octaven versio 3.8.0.

Toinen Octave-paketti eli versio 3.8.2, jossa on myös mukana kaikki paketit. Tähän en saanut videopakettia vielä toimimaan, mutta kiersin sen käytön, mikä saattaa vaikuttaa hieman lopullisen kuvan laatuun. Jos videopaketti olisi ollut käytössä, olisin voinut käyttää Kenn Sebastian (n.d) background\_generaattoria ja poistaa kuvista turhaa dataa ja siten tehdä kuvan käsittelyn nopeammaksi. Laatimani lopullinen background\_generator on liitteessä 3.

Kuten Taulukosta 3 näkee, tässä vaiheessa sovellus siis toimi jossain määrin, joten tein together -funktion, joka löytyy liitteestä 1.

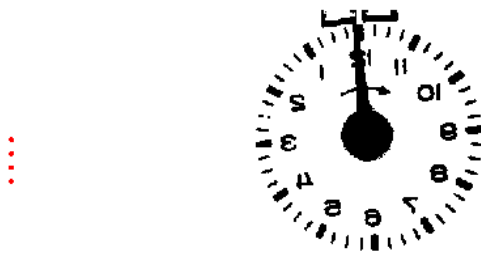
Taulukko 3. Koko konenäkösovelluksen läpi viedyn oikean ylämittarin mittaustulokset

Name	Class	Dimension	Value
I2	uint8	480x640x3	...
max2	double	1x1	12
measure2	double	1x2	[188.74, 162.35]
min2	double	1x1	0
rho2	double	1x1	114.24
theta02	double	1x1	-1.6613
theta2	double	1x2	[-2.3690, 3.8010...
x002	double	1x1	391.58
x02	double	1x1	401.90
y002	double	1x1	107.58
y02	double	1x1	221.36

Tässä vaiheessa havaittiin, että tulokset eivät vaikuta todennäköisiltä, ja tästä syystä alkoi virheen etsintä koodista. Virheen etsinnän aloitin käymällä koodin ja tulokset läpi.

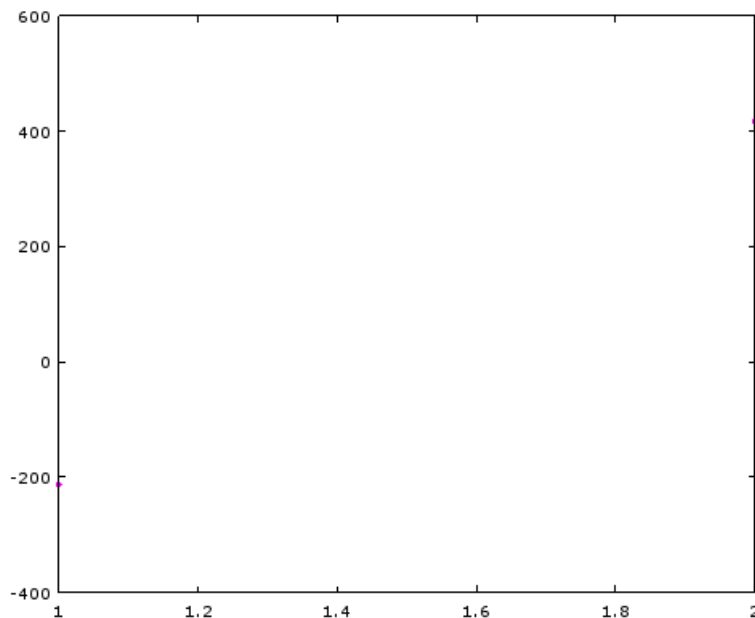
Huomasin pian, että koodissa se kohta, jonka pitäisi piirtää kuvaan viiva siihen kohtaan, missä linja on havaittu, ei toimi eli kuvassa ei näkynyt viivoja. Tästä syystä aloin selvittämään, mikä siinä on vikana. Pian selvisi että tämä osa koodia oli Matlab -koodia, jota Octave ei ymmärrä. Kuva piti ensin muuttaa binäärikuvaksi, ennen kuin viiva saatiin näkymään.

Korjasin vian aluksi jakamalla dial\_readerin kahteen osaan, josta ensimmäisen oletettavasti toimivan osan läpi käytyäni voin lyhyemmässä ajassa käydä läpi sovelluksessa ohjelmallisesti myös toisen osan. Samalla huomasin, että houghlines -funktion (ks. liite 7) pyydetyn viivan pituus oli selvästi liian lyhyt ja sitä kasvattamalla sain koodin toimimaan hieman nopeammin, sillä sovellus ei enää huomionnut lyhyitä viivoja. Tässä vaiheessa sain tuloksen, kuten kuviossa 37. Jostain syystä sovellus päättää, että viivan x:n pitäisi olla noin 1.



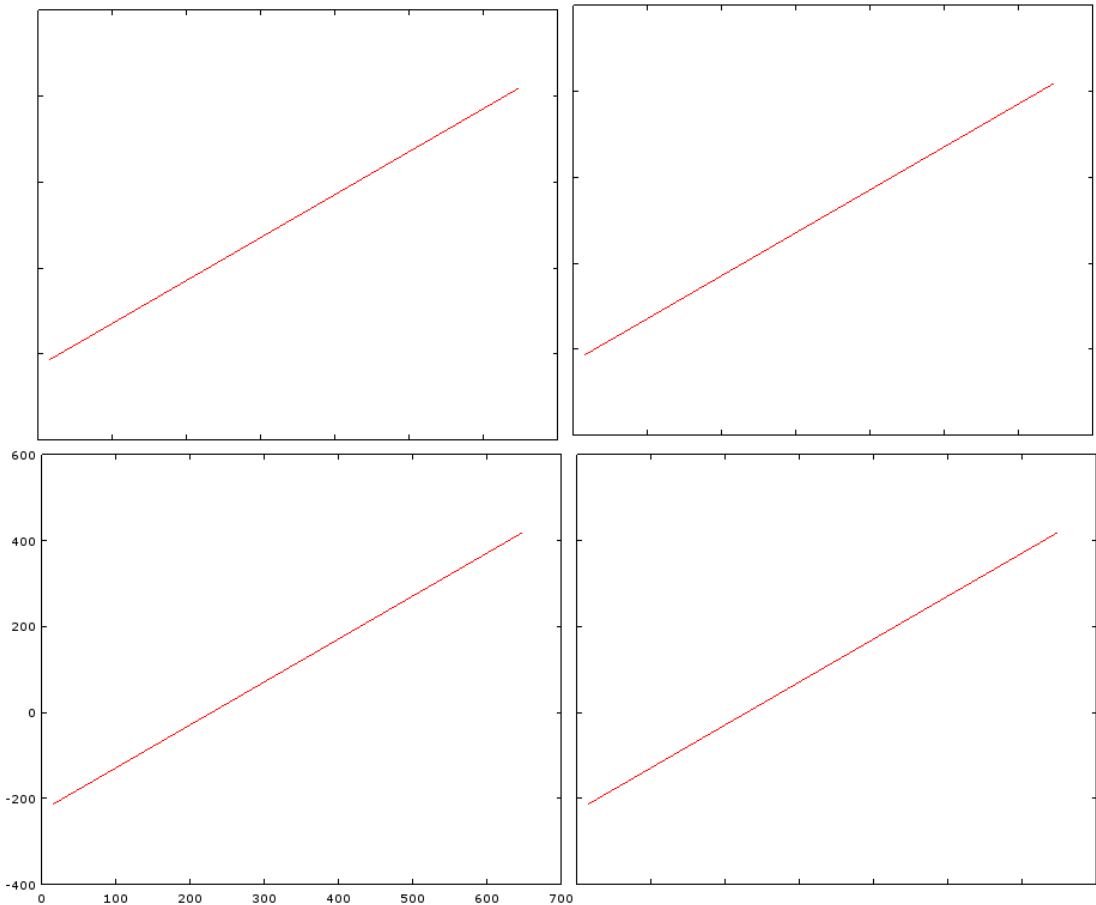
Kuvio 37. Kuvassa pisteet kuvastavat missä kuvan pisin viiva sovelluksen mielestä on.

Tässä vaiheessa myös päivitin viivan piirron uudemman koodin mukaan, jonka löysin Kenn Sebestan (2008) omalta sivulta. Aiemmin käyttämäni hackadayn sivulta (Eliot 2006) löytämäni linkissä olevan koodin sijaan, jossa hän käyttää plot -funktioita aiemman line -funktion sijaan. Kun kokeilin itsekin tätä funktiota, niin sain kuvion 38 esittämät kuvan. Tässä vaiheessa minulle selvisi, että Octavessa plot -funktio ei osaa piirtää kuin viimeisimmän sille määritellyn parametrin, vaikka Matlab osaa piirtää useita.



Kuvio 38. Kuvassa pisteiden pitäisi kuvastaa kuvan suurimman viivan molempia reunoja.

Tästä vielä edemmäksi menemällä sain kuvioon 39 kuviota 38 vastaavat kuvat, kuvan viivaa, esitetään viivana aiemman pistekoordinaattien sijasta ja x akseli ei enää lopu lukuun kaksi, vaan menee pidemmälle yli kuuden sadan.

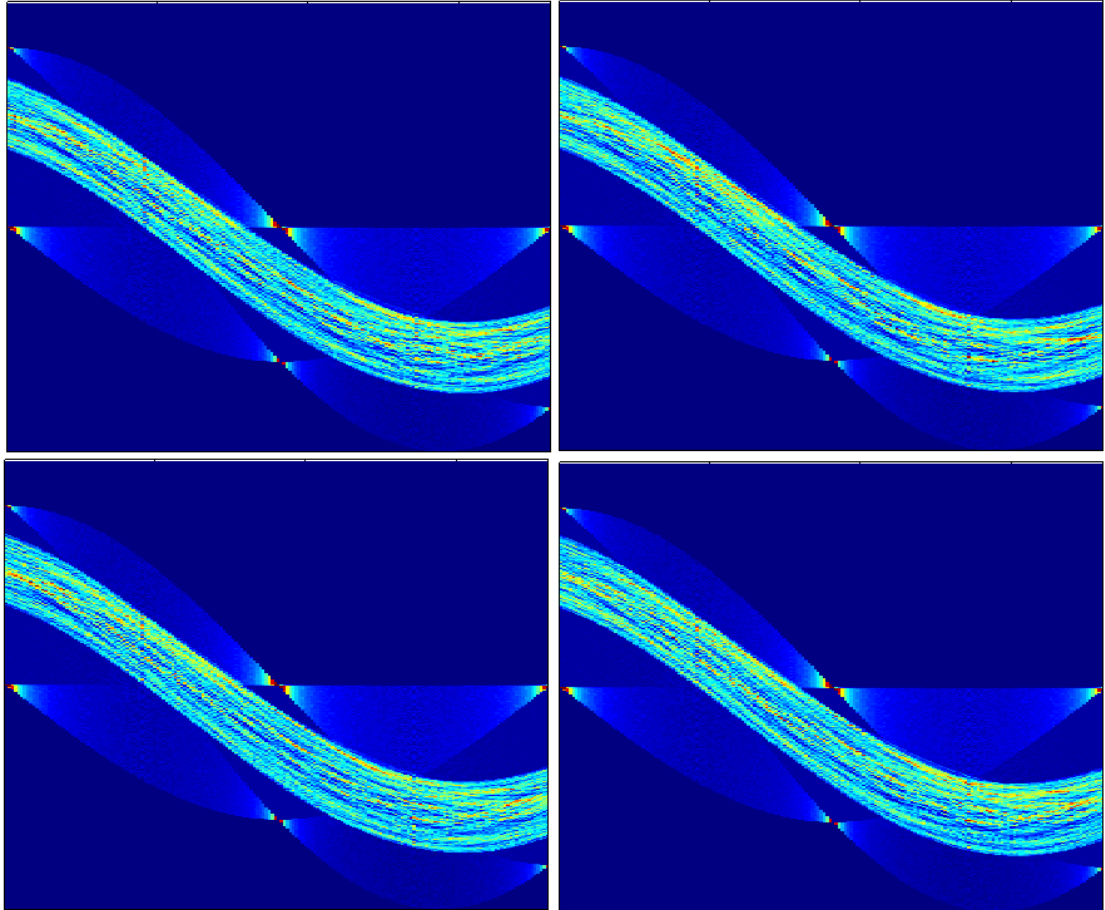


**Kuvio 39.** Tuoreempi kuva, jossa jokainen kuvaaja näyttää, missä sen mittarin pisin viiva on, kuvaaja esittää toista kuvan pisimmistä viivoista.

Tämän jälkeen käyvin vielä läpi houghlines (ks. liite 7) ja houghpeaks (ks. liite 6) funktiot, kunnes ne vaikuttivat toimivan oikein. Kun ne sain hyvin toimimaan, ilmeni, että ongelman juuri ei ollut kummassakaan näistä, vaan ensimmäisessä Hough -muunnoksessa (ks. liite 5). Vielä tällä hetkellä minulle tuntemattomasta syystä Hough -muunnos saa saman tuloksen jokaisesta eri kuvasta, ja koska koodi ei toimi ilman sitä, en voi edetä työssä, ilman että aloitan tekemisen täysin uudella tavalla tai ohjelmalla.

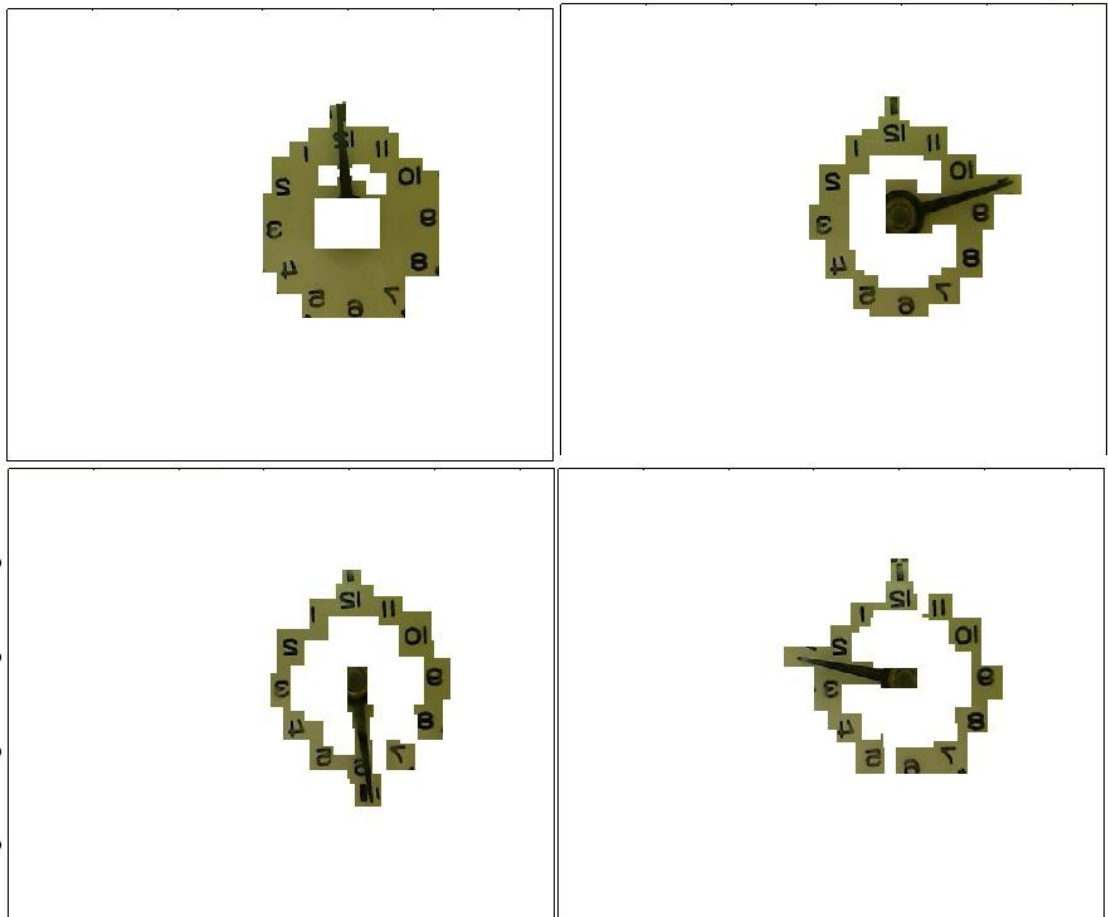
Kuviossa 40 on Hough -muunnoksen tulos jokaisesta kuvasta erikseen. Näissä kuvissa keskellä oleva virta monia erivärisiä viivoja kuvastaa kuvassa olevaa turhaa tietoa, joka ei ole tarpeeksi viivamaista, että siitä otettaisiin tulos. Suurinta viivaa vastaa kuviossa oleva kohta, jossa näkyy keskellä selkeä punainen väri, kun kaksi värin molemmilla puolilla olevaa muotoa kohtaa. Näitä on jokaisessa kuviossa kaksi, ja nämä

kaksi kuvastavat kuvion kahta suurinta viivaa, joiden pitäisi olla mittarin viisarin eri puolet.



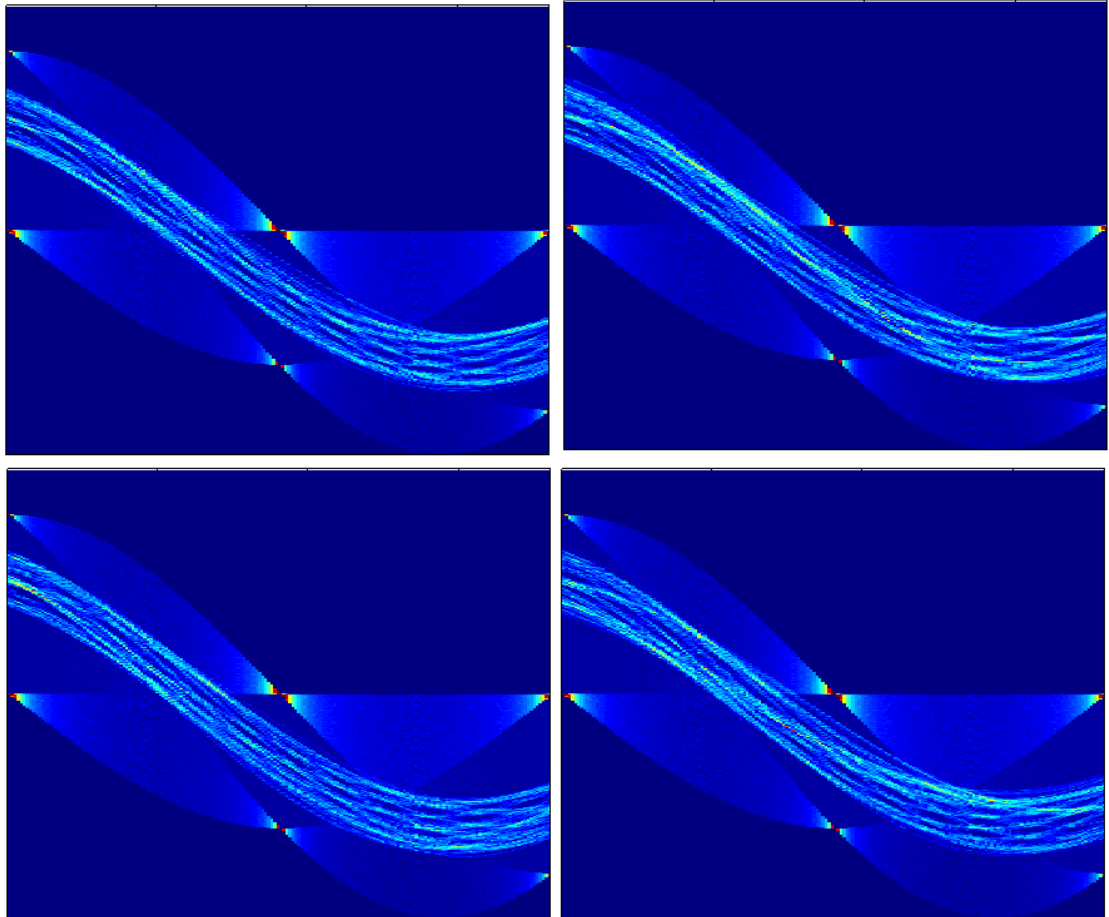
Kuvio 40. Hough -muunnos, joka antaa jokaisella mittarilla saman tuloksen

Hough -muunnos ongelmaan ei auttanut myöskään uusin kokeiluni, jossa poistin kuvasta kaiken ylimääräisen toivoen poistavani myös virheen aiheuttajan. Muokatut kuvat, joilla tätä kokeilin, ovat kuviossa 41.



Kuvio 41. Testikuva, josta on poistettu mahdolliset viisaria isommat viivat ja ympyrät

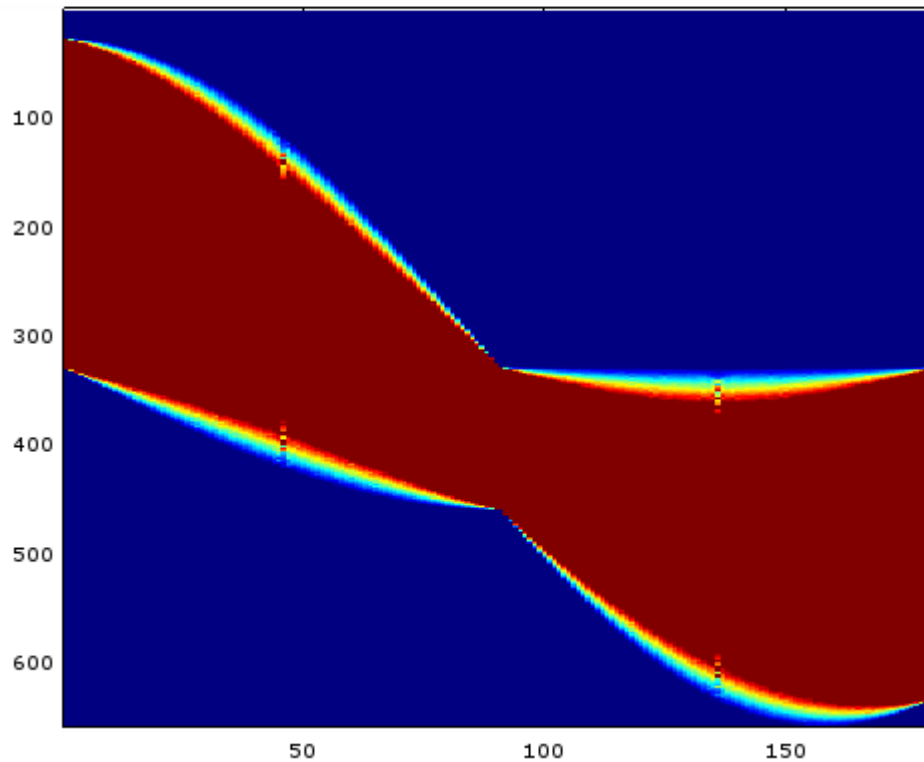
Kuviosta 42 nähdään näiden muokattujen kuvien Hough -muunnoksien tulokset. Kuten kuviosta näkee, vain keskellä oleva "häiriötieto" on vähentynyt mutta suurimmat punaisena näkyvät kohdat ovat identtiset jokaisessa kuvassa. Nämä pisteet ovat yhä samat kuin kuviossa 40, ja ne vastaavat samaa tietoa suurimmasta viivasta kuvassa.



Kuvio 42. Testikuvan Hough -muunnos, jossa vähemmän häiriötä mutta samat isot pisteet

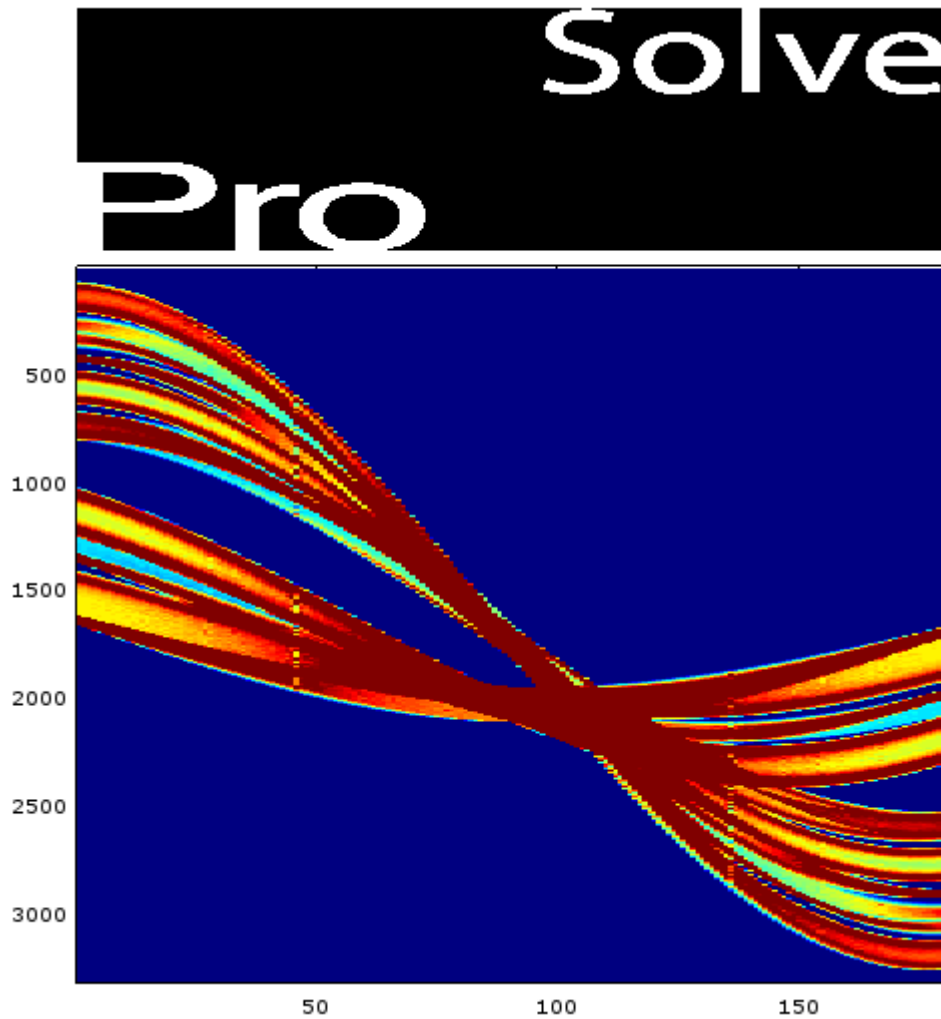
Tässä vaiheessa en tiennyt, toimiiko Hough -muunnos ollenkaan, joten testasin samaa muunnosta alustavalla dilaatiolla myös erilaisille kuville. Otin esimerkiksi Jyväskylän ammattikorkeakoulun logon ja Prosolven logon.

Kuten kuviosta 43 eli Jamkin logosta tehdystä Hough -muunnoksesta voi nähdä, ovat Hough -muunnoksen reunat hyvin lähellä aluetta, jossa mittarin Hough -muunnos näyttää vahvinta tulosta kuvassa olevasta viivasta. Tämän kuitenkin tarkoittaa täysin eri asiaa, koska Hough -muunnetussa kuvassa on vain yksi paksu aaltoileva viiva, tämä viittaa siihen että Jamkin logossa oli vain 1 tunnistettu viiva, joka on luultavasti ylöspäin suuntaava logon keskikohta.



Kuvio 43. Jamkin logo, jolle on tehty dilaatio ja Hough -muunnos

Kuten kuviosta 44 vuorostaan voi nähdä, on Prosolven logon Hough -muunnoksen tulos täysin erilainen, mikä vahvistaa sen että Hough -muunnos toimii Octave -ohjelmassa, mutta ei minun koodissa. Tästä Hough -muunnetusta kuvasta näkee että useat viivat kohtaavat samassa kohtaa ja "häiriötä" ei ole juurikaan, kun kaikki viivat kohtaa samassa keskikohdassa.



Kuvio 44. Prosolven logo, jolle on tehty dilaatio ja Hough -muunnos

## 5.2 Kamera

Demokamerana käytettiin hyvin perus web-kameraa. Koska sovellusta ei ole saatu vielä valmiiksi, en ole päässyt paremmin tutkimaan sovelluksen toimintaa eri kameroiden kanssa.

## 5.3 Valaistus ja jalusta

Erillinen valaistus on tarpeellista, koska osa mittareiden käyttö paikoista voi olla huoneissa, jotka yleensä pidetään pimeänä. Alustava idea on toteuttaa diffuusivalaistus led-valoilla, jotta mittarin kupu ei heijasta, niin että kuvasta saadaan selkeä. Tarkoitus oli, että näihin otetaan virta tietokoneelta erillisellä usb -johdolla.

Jalusta on taas tarpeellinen sen takia, että web-kameroita ei ole suunniteltu kuvaamaan analogisia mittareita. Parhaimmillaan jalusta toimii niin, että se myös hajauttaa

muut huoneen valot, niin että mittarin kupuun ei tule heijastuksia. Silti pitäisi jättää joku kohta, mistä lukua pystyy seuraamaan myös paikalle menemällä.

#### **5.4 Tiedon jakaminen turvallisesti**

Tarkoituksena on tehdä jokaiselle mittarille oma verkkosivu, josta mittarin arvo pystyy seuraamaan etäältä. Nämä sivustot tultaisi tekemään niin, että lukua pääsee seuraamaan vain kirjautumalla sisään. Sovellus ei ole valmis ja sen takia ei tätä ole vielä ruvettu kehittämään.

### **6 Pohdinta**

Tavoitteena oli toimiva sovellus, jolla voidaan ottaa digitaalinen arvo analogisesta mittarista, minä tahansa annettuna ajan hetkenä. Tässä en ole vielä suoriutunut. Omia tuloksia en saanut tuotettua, sillä sovellus ei vielä ole valmis ja sen takia en pysty tarkistamaan sovelluksen tarkkuutta verrattuna muiden tekijöiden tekemiin ohjelmiin. Onnistumiseksi voidaan laskea se, että ohjelma on nyt paljon tutumpi ja sitä on myöhemmin helppo käyttää. Sovellus itsessään ei vielä toimi.

Työ osoittautui hyvin haastavaksi, sillä tein sen ohjelmalla, jota en ollut ennen käyttänyt. Alustavasti uskoin, että Matlab- ja Octave -ohjelmat olisivat enemmän samankaltaisia ohjelmointikieleltään, mutta Octavea ei ole yhtä hyvin kehitetty. Työnteko vielä jatkuu, vaikka opinnäytetyön palautankin. Jos olisin käyttänyt ohjelmaa, jota on kehitetty pelkästään konenäkö-tarkoitukseen, luulen että olisin saanut sovelluksen valmiiksi.

Myös tekotapa, jossa en käytä Hough -muunnosta olisi voinut toimia, mutta olisi ollut käyttöönotossa raskaampi tehdä. Luulen että olisin saanut tämän toimimaan esimerkiksi NI vision assistantilla ja kiintopisteellä. Tämän hankalaksi tekee se, että kaikissa mittareissa ei ole haluttua kiintopistettä joka näkyy jokaisessa mittarin asennossa. Käyttöönotossa pitää käyttöönottajän määritellä tämä kiintopiste.

Saatuja tuloksia ei voi hyödyntää vielä mihinkään, mutta kun sovellus toimii, saadaan tulokset paremmiksi. Tällä hetkellä sovellus laskee Hough-muunnoksesta viisarin

asennon ja pituuden, ottaen huomioon käyttäjän asettamat tiedot. Nämä tiedot ovat kuitenkin väärin, koska Hough-muunnos ei toimi.

Aluksi pitää saada sovellus toimimaan, sen jälkeen varmistaa, että se toimii usealla erilaisilla mittareilla. Sen jälkeen pitäisi löytää, mikä kamera on paras, jos työn haluaa monistaa useaan eri käyttökohteeseen. Sitten voi hoitaa digitaalisen luvun Internetiin jakamisen turvallisesti ja siitä voidaan edetä mihin suuntaan vain. Työlle pitäisi tehdä aluksi selkeä asennus- ja käyttöohje. Tämän jälkeen kehitystä voi siirtää haluttuun suuntaan joko optimoimalla sovellusta tai tekemällä siitä enemmän käyttäjäystävällisen.

## Lähteet

AD-muunnin. N.d. Elektroniikasta, tietokoneista ja samankaltaisista kertova harrastajasivu. Viitattu 30.3.2015

[http://www.hutasu.net/index.php?sivu\\_id=43](http://www.hutasu.net/index.php?sivu_id=43)

Ahlroth, H. 2010. Konenäköjärjestelmät. Kappaletavara-automaatio opetuspaketti Aaltoyliopistolle. Viitattu 13.3.2015.

[https://noppa.aalto.fi/noppa/kurssi/as-116.1100/luennot/AS-116\\_1100\\_luentokalvot\\_konenako.pdf](https://noppa.aalto.fi/noppa/kurssi/as-116.1100/luennot/AS-116_1100_luentokalvot_konenako.pdf)

Azad, P., Gockel, T. & Dillman R. 2008. Computer Vision Principles and Practice. United Kingdoms: Elektor International Media BV.

Bimetallic strip thermometer. N.d. Kotisivu josta lainasin mittarikuva. Viitattu 28.3.2015 [http://www.daviddarling.info/encyclopedia/B/bimetallic\\_strip.html](http://www.daviddarling.info/encyclopedia/B/bimetallic_strip.html)

Eliot. 2006. Digitally reading analog gauges. Viitattu 22.4.2015

<http://hackaday.com/2006/02/18/digitally-reading-analog-gauges/>

Hirvisaari, M., 2014. Sanasto konenäön. Kotisivun rakennus sivustolla Spacebimbom tehty sanasto, joka selittää konenäön asioita selkokielellä. Viitattu 30.3.2015

<http://spacebimbom.com/kategoriat/uskonto/sanasto-konenaon.php>

Hough Line Transform. n.d. Koodauksessa auttava sivusto, jossa Hough -muunnoksesta tietoa selkokielellä. Viitattu 28.4.2015

[http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/hough\\_lines/hough\\_lines.html](http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/hough_lines/hough_lines.html)

Hyyti, H. 2014. Konenäkö robotin ohjauksessa. Automaatio- ja systeemiteknikan laboriotyöt. Aalto-Yliopisto Viitattu 30.3.2015

[https://noppa.aalto.fi/noppa/kurssi/as-0.2230/materiaali/AS-0\\_2230\\_tyon\\_3\\_tyoohje\\_ja\\_esiselostustehtavat.pdf](https://noppa.aalto.fi/noppa/kurssi/as-0.2230/materiaali/AS-0_2230_tyon_3_tyoohje_ja_esiselostustehtavat.pdf)

Introduction to Modulation transfer Function. N.d. Modulaationsiirtofunktiosta kertova sivu Edmund optics firman sivuilta. Viitattu 13.3.2015

<http://www.edmundoptics.com/technical-resources-center/optics/modulation-transfer-function/>

Karhulahti, M, n.d. Kuvaamisen perusteita. Kotisivu digikuvaamisesta, josta otin tietoa suljin aukoista. <http://pelivara.com/tutoriaalit/kuvaamisen-perusteita/>

Lens Calculator. N.d. Linssinlaskemiseen tehty sivusto, josta lainasin kuva. Viitattu 13.3.2015 <http://www.discount-security-cameras.net/lens-calculator.html>

Prosolve. N.d. Opinnäytetyön toimeksiantajan kotisivut Viitattu 31.3.2015

<http://www.prosolve.fi/>

Prosolve työpaikkana. N.d. Opinnäytetyön toimeksiantajan kotisivujen työpaikka osio  
Viitattu 31.3.2015 <http://www.prosolve.fi/prosolve/prosolve-tyopaikkana/>

Rantapuska, S. 2014. Konenäkö. Oppimateriaali. Jyväskylän ammattikorkeakoulu,  
Automaatiotekniikan koulutuslinja. Viitattu 3.4.2015. Optima oppiympäristö.

Sebasta, K. 2008. Automatic Dial Reader. Viitattu 22.4.2015  
[http://www.eissq.com/BallandPlate/appendix/dial\\_reader.html](http://www.eissq.com/BallandPlate/appendix/dial_reader.html)

Sebasta, K. N.d. A-D converter the hard (but cheap!) way. Viitattu 22.4.2015  
<http://www.eissq.com/DialADC.html>

Soini A. N.d. Konenäkö. Suomen Automaatioseura ry. Viitattu 13.3.2015  
<http://www.automaatioseura.fi/index/tiedostot/Konenako.pdf>

Sonka, M., Hlavac V. & Boyle R. 2008. Image Processing, Analysis, and Machine Vision.  
3rd edition., United States of America: Cengage Learning

Understanding Focal Length. N.d. Nikonin linssin tarkkuuden opetuspaketti. Viitattu  
29.3.2015 <http://www.nikonusa.com/en/Learn-And-Explore/Article/g3cu6o2o/understanding-focal-length.html>

