AALTO-YLIOPISTO
TEKNILLINEN KORKEAKOULU
Informaatio- ja luonnontieteiden tiedekunta
Tietotekniikan tutkinto-ohjelma

QR-koodin lukeminen kameralla otetusta kuvasta

Kandidaatintyö

Jussi Judin

Mediatekniikan laitos
Espoo 2010

AALTO-YLIOPISTO
TEKNILLINEN KORKEAKOULU
Informaatio- ja luonnontieteiden tiedekunta
Tietotekniikan tutkinto-ohjelma

KANDIDAATINTYÖN
TIIVISTELMÄ

Tekijä:	Jussi Judin	
Työn nimi:	QR-koodin lukeminen kameralla otetusta kuvasta	
Päiväys:	28. huhtikuuta 2010	**Sivumäärä:** 12 + 37
Pääaine:	Informaatiotekniikka	**Koodi:** T3006
Vastuuopettaja:	prof. Kai Puolamäki	
Työn ohjaaja:	prof. Kai Puolamäki	

QR-koodi on kaksiulotteinen matriisityyppinen viivakoodi, jota käytetään erityisesti Japanissa välittämään koneluettavaa dataa matkapuhelinsovelluksille. Tämän seurauksena QR-koodi usein luetaan matkapuhelimen kameralla otetusta kuvasta lähes reaaliaikaisesti, mikä asettaa rajoituksia lukemisongelman ratkaisussa käytettäville menetelmille.

Tässä työssä tarkastellaan QR-koodin ominaisuuksia, kuvan ottamisvaiheessa esiintyviä ilmiöitä ja tehdään kirjallisuuskatsaus QR-koodin lukemiseen liittyvistä julkaistuista tutkimuksista, joiden pohjalta käsitellään tarkemmin lukemisongelman vaiheiden toteutuksia. QR-koodin rakenne pyrkii sietämään osaa koodialueen lukukelvottomaksi tekevistä ilmiöistä ja eri esikäsittelyvaiheilla pyritään tekemään koodialueesta helpommin tunnistettava.

QR-koodin aluetta etsitään hakemalla koodille ominaisia etsinkuvioita, joko suoraan kuvasta tai esikäsittelyn seurauksena saaduilta ehdokasalueilta. Näiden perusteella tunnistetaan koodialueen tarkka sijainti ja asento, ja tälle alueelle tehdään vielä lukemistulosta mahdollisesti parantavia korjauksia.

Avainsanat:	kaksiulotteinen viivakoodi, QR-koodi, digitaalinen kuvankäsittely, konenäkö
Kieli:	Suomi

Sanasto

Kaikelle tässä työssä käytettävälle sanastolle ei välttämättä ole vakiintunutta suomalaista vastinetta, erityisesti kun lähdekirjallisuus on pääasiassa englanniksi, joten tässä on tekstissä esiintyvien vakiintumattomien sanojen käännöslista.

suomeksi	englanniksi
datan ja virheenkorjauksen koodisanat	data and error correction codewords
erotin	separator
etsinkuvio	finder pattern
formaattitiedot	format information
kiinalainen luotettava koodi	Chinese sensible code
kohdistuskuvio	alignment pattern
koodausalue	encoding region
pisteenleviämisfunktio	point spread function
rakenteen liittämistila	structure append mode
rauhoitettu alue	quiet zone
toiminnallinen kuvio	function pattern
versiotiedot	version information
viereisyyspuu	adjacency tree
ääriviivojen määrittäminen	contour detection

Sisältö

1 Johdanto **1**

2 QR-koodin rakenne ja ominaisuudet **3**

2.1 QR-koodin rakenne 3

2.2 Standardin määrittelemät muunnokset 5

2.3 Virheensieto-ominaisuudet 5

2.4 Standardin toteutus käytännössä 6

3 Koodin lukemista hankaloittavat ilmiöt **8**

3.1 Kuvan tarkkuus 8

3.2 Kohina 10

3.3 Sumeneminen 10

3.4 Epätasainen valaistus ja varjot 10

3.4.1 Vinjetointi 11

3.5 Epätasainen tausta 11

3.6 Vääristymät 12

3.7 Koodialuetta peittävät ilmiöt 12

3.8 Tunnistustilanteen asettamat rajoitukset 12

4 Aikaisempaa tutkimusta QR-koodin lukemisesta **14**

4.1 Ohbuchi, Hanaizumi ja Hock 2004 15

4.2 Yue ja Mingjun 2006 15

4.3 Aidong, Yan ja Caixing 2007 16

4.4 Chu, Yang ja Chen 2007 17
4.5 Chang, Chu ja Chen 2007 18
4.6 Yue, Bo ja Yu 2008 19
4.7 Alfthan 2008 19

5 QR-koodin lukemisessa käytettäviä menetelmiä 21
5.1 Harmaasävymuunnos 22
5.2 Kuvan kynnystys 22
5.3 Liikkeestä johtuva kuvan sumenemisen korjaus 24
5.3.1 Sumenemisparametrien arviointi 25
5.3.2 Sumenemiskorjaus 26
5.4 Viivakoodialueen haku 27
5.4.1 Etsinkuvioiden haku 27
5.4.2 Reunantunnistus 28
5.5 Käänteisperspektiivimuunnos 30
5.5.1 Kohdistuskuvioiden haku 32
5.6 Viivakoodin uudelleenrakennus 33

6 Yhteenveto 34

Kirjallisuutta 35

Luku 1

Johdanto[1]

QR-koodi eli nopean vasteen koodi (quick response code) on nopeaa ja yksinkertaista tunnistusta silmälläpitäen kehitetty kaksiulotteinen matriisityyppinen viivakoodi. Koodin kehittäjäyritys Denso Wave kehitti koodin vuonna 1994 ratkaisemaan yksiulotteisten viivakoodien ongelmat datan tallennustiheyden ja lukemisen luotettavuuden suhteen(Hara et al., 1998). Koodi standardoitiin ensimmäisen kerran vuonna 1997 Automaattisen tunnistamisen ja liikkuvuuden liiton (Association for Automatic Identification and Mobility, AIM) standardina ja viimeisin versio Kansainvälisen standardoimisjärjestön (International Organization for Standardization, ISO) standardista on vuodelta 2006(ISO, 2006). (Denso Wave, 2010)

QR-koodia käytetään teollisuustuotannon, laadunvalvonnan ja kaupan logistiikan ulkopuolella mitä moninaisimmissa sovelluksissa, jotka ovat mahdollistaneet halvat webbikamerat ja kameralla varustetut matkapuhelimet. QR-koodin lukeminen on osana japanilaisen NTT DOCOMO -matkapuhelinoperaattorin i-mode puhelintyyppien mahdollisia ominaisuuksia(NTT DOCOMO, 2010), minkä omistajia on Japanissa 48 miljoonaa(NTT DOCOMO, 2009). Tämä on johtanut QR-koodien käyttöön esimerkiksi mainoksissa kuvan 1.1 tapaisesti tekemään WWW-sivujen osoitteista koneluettavia. HTTP-osoitteiden välittämisen lisäksi yleisesti tuettuja käyttömuotoja QR-koodille on erilaisten yhteystietojen saattaminen koneluettavaan muotoon ja siten niiden helppo tuonti matkapuhelimen osoitekirjaan (Zxing-projektin wiki, 2010). Tämä asettaa monille tunnistussovelluksille vaatimuksia lähes reaaliaikaiselle suoritusajalle tunnistustarkkuuden kustannuksella.

Tässä työssä on tehdään katsaus erityisesti QR-koodin tunnistamiseen

[1] Tämä dokumentti sisältää värikuvia, jotka eivät näy oikein mustavalkotulosteissa.

Kuva 1.1: WWW-osoitteen välittäminen QR-koodin avulla.

liittyvästä kirjoitetusta tutkimuksesta, mikä liittyy osana yleisempään kaksiulotteisten viivakoodien tunnistamiseen. Työ jakautuu neljään sisältölukuun, joissa luvussa 2 kuvataan standardin määrittelemä QR-koodin rakenne, luvussa 3 kuvataan yleisesti ilmiöitä, mitkä hankaloittavat koodin lukemista kameralla otetuista kuvista, luvussa 4 käsitellään eri tutkimusten lähestymistapoja tämän ongelman ratkaisuun ja luvussa 5 kuvataan tarkemmin menetelmiä ja periaatteita, joiden avulla oman koodinlukijan luominen onnistuisi.

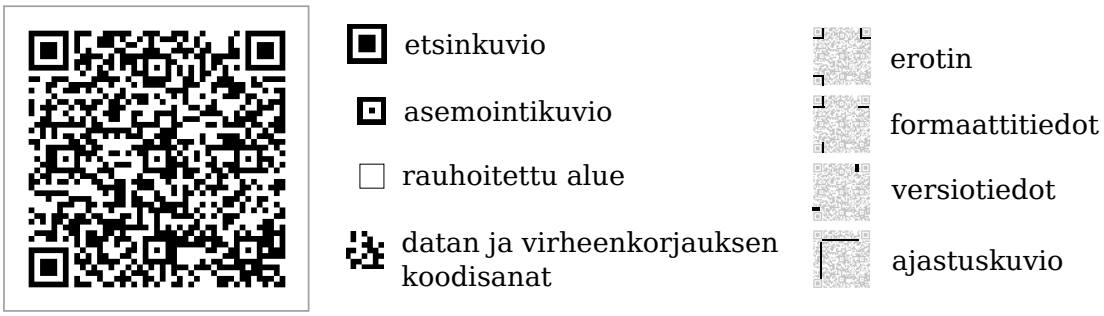
Luku 2

QR-koodin rakenne ja ominaisuudet

Tässä luvussa kuvataan ISO/IEC 18004:2006 standardin mukainen QR-koodin rakenne ja koodin tarjoamat ominaisuudet datan tallentamisen kannalta. Lisäksi tehdään lyhyt katsaus siihen kuinka standardi on käytännössä toteutettu ja miten standardin mahdollisuuksia hyödynnetään.

2.1 QR-koodin rakenne

QR-koodi on kaksiulotteinen neliön muotoinen matriisityyppinen viivakoodi, joka koostuu kuvan 2.1 mukaisista osista. Koodin pienin yksikkö on yhden bitin sisältävä tumma tai vaalea piste, eli moduuli (module) ja kaikki QR-koodin mitat määritellään yhden moduulin moninkertoina. Moduulien värien ei tarvitse olla välttämättä musta ja valkoinen, kunhan vaaleat moduulit ovat erotettavissa



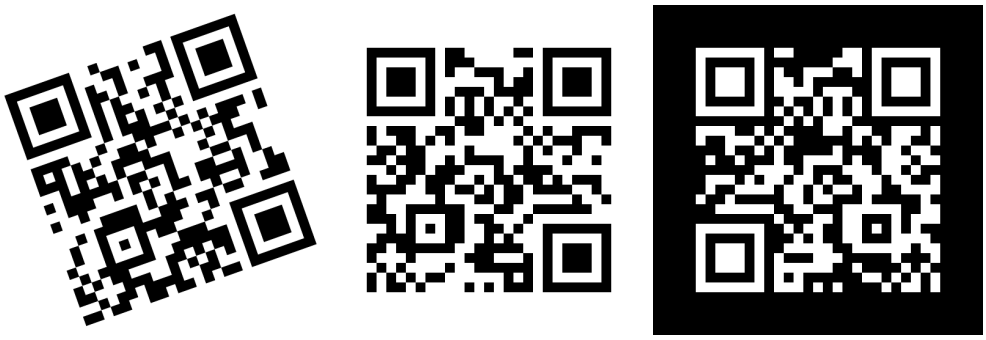
Kuva 2.1: Version 7 QR-koodin rakenne ja alueet.

tummista. Varsinainen moduuli voi koostua useammasta kuvapisteestä, riippuen siitä millä menetelmällä luettava koodialue on tuotettu ja minkä kokoisena se halutaan lukea.

Datantallennusominaisuudet vaihtelevat riippuen koodialueen koosta ja virheenkorjaustasosta. QR-koodin standardi määrittelee 40 eri versiota, joista pienin on versio 1, missä sivun pituus on 21 moduulia ja jonka kapasiteetti on 26 8-bittistä koodisanaa. Suurin version 40 koodi on sivunpituudeltaan 177 moduulia ja kapasiteetiltaan 3706 koodisanaa, eli yhden version kasvu kasvattaa koodialueen sivua aina 4 moduulia. Varsinainen data voi olla numeerista, alfanumeerista, 8-bittisiä tavuja tai 13-bittisiä kanjimerkkejä tai laajennetun kanavatulkinnan (Extended Channel Interpretations, ECI)(AIM) mukaisesti koodattuja eri merkistöjen merkkejä.

QR-koodin eri osat jaetaan toiminnallisiin kuvioihin (function pattern) ja koodausalueisiin (encoding region). Toiminnalliset kuviot määrittelevät QR-koodin alueen, jolta varsinainen data luetaan ja ne eivät täten itsessään sisällä dataa. Toiminnallisia kuvioita ovat erotin (separator), etsinkuviot (finder pattern), kohdistuskuviot (alignment pattern), ja ajastuskuviot (timing pattern). Koodausalueet sisältävät varsinaisen QR-koodin datan ja sen koodausmuodon ja näitä alueita ovat versiotiedot (version information), formaattitiedot (format information) ja datan ja virheenkorjauksen koodisanat (data and error correction codewords). Lisäksi varsinaisen koodialueen ympärille tulee jättää vähintään 4 moduulin levyinen vaalea rauhoitettu alue (quiet zone). Kaikki edellä mainitut osat löytyvät vähintään version 7 mukaisesta koodista. Versioista 2–6 puuttuu versiotietoalue, ja versiosta 1 puuttuu näiden lisäksi vielä kohdistuskuvio.

Etsinkuvioiden tarkoituksena on mahdollistaa QR-koodin nopea tunnistaminen kuvasta, ja niiden hakemista kuvataan tarkemmin luvussa 5.4.1. Erotin on yhden moduulin levyinen alue ja se erottaa etsinkuvion koodin muusta alueesta. Ajastuskuvion tarkoituksena on mahdollistaa QR-koodin version tunnistaminen ja koodisanojen moduulien sijainnin selvittäminen. Kohdistuskuvioilla taas on korkeampien versioiden kohdalla tarkoitus antaa viitepisteitä koodialueen muodon selvittämiseksi, jotta erilaiset vääristymät kuvassa eivät haittaisi niin paljon isojen koodin lukemista. Kohdistuskuvioiden käyttöä on kuvattu tarkemmin luvussa 5.5.1.



(a) Kierto. (b) Peilaus. (c) Käänteiset värit.

Kuva 2.2: Standardin asettamat vaatimukset muunnoksille, missä koodi pitää tunnistaa.

2.2 Standardin määrittelemät muunnokset

QR-koodi ei ole välttämättä aina lukemisen kannalta optimaalisimmassa perusasennossa, vaan se voi olla kuvassa muunnettuna eri tavoin, mistä QR-koodin tunnistavien ohjelmien pitäisi selvitä. Kuvassa 2.2 näkyy standardin määrittelemät muunnokset ja luvussa 3.6 kuvataan muita kuvatessa aiheutuvia vääristymiä.

QR-koodin tulisi olla luettavissa kaikista suunnista, mitä neliön muotoiset etsinkuviot tukevat. Lisäksi QR-koodi voi olla peilattu vaakatasossa, jolloin mahdollisesti tunnistettu koodialue voidaan peilata, josta voidaan yrittää tunnistamista uudelleen. Lisäksi koodi voi olla muodostettu käänteisin värein, jolloin tumma moduuli onkin vaalea ja vaalea moduuli on tumma. Tätä varten kuvasta voi kääntää värit ja yrittää tunnistamista uudelleen.

Lisäksi korkeampiversioinen QR-koodi voi muodostua korkeintaan 16 matalampiversioisesta QR-koodista, jotka luetaan yksitellen ja yhdistetään toisiinsa erityisessä rakenteen liittämistilassa (structure append mode).

2.3 Virheensieto-ominaisuudet

Koska QR-koodin lukemisessa voi tapahtua virheitä tai koodialueella saattaa olla roskia antamassa vääriä lukemia, QR-koodiin on rakennettu erilaisia virheensietoa parantavia ominaisuuksia. Versio- ja formaattitiedoissa on erillinen BCH-koodilla(Reed ja Chen, 1999) toteutettu virheenkorjaus ja samat tiedot sijaitsevat kahteen kertaan eri puolilla koodialuetta. Tämä mahdollistaa

Taulukko 2.1: QR-koodin virheenkorjaustasot ja arvioitu virheenkorjauskyky.

Virheenkorjaustaso	Virheenkorjauskyky prosentteina (suunnilleen)
L	7
M	15
Q	25
H	30

koodin lukemisen, vaikka yksi etsinkuvio ja sen lähiympäristö olisikin lukukelvoton. Koodisanat ovat tallennettuina mahdollisimman paljon suorakaiteen muotoisina, jotta yksittäiset pistemäisen läikät aiheuttaisivat mahdollisimman vähän tuhoa verrattuna siihen, että koodisanojen moduulit tallennettaisiin vaaka- tai pystysuoraan (Hara et al., 1998).

Varsinaiset dataa sisältävät koodisanat on maskattu niin, että niissä toistuu mahdollisimman vähän tummia tai vaaleita moduuleita, jotta moduulien rajoissa olisi mahdollisimman vähän epäselvyyksiä. Lisäksi datalle on määritelty neljä eri Reed-Solomon-koodeja(Reed ja Chen, 1999) käyttävää virheenkorjaustasoa, jotka nakyvät taulukossa 2.1. Tarkka virheenkorjauskyky riippuu kuitenkin QR-koodin versiosta ja virheen laadusta, riippuen siitä tunnistetaanko koodisana virheelliseksi vai ei.(ISO, 2006)

2.4 Standardin toteutus käytännössä

Kuva 2.3: QR-koodin virheensieto-ominaisuuksien käyttö logon sisällyttämiseen.

Sovelluskohtaisen tiedon tallentamisen lisäksi QR-koodien avulla välitetään dataa, jota QR-koodeja lukevat ohjelmat osaavat yleisesti tulkita ja avata tähän liittyviä sovelluksia. Näistä yleisin käyttötarkoitus on HTTP-osoitteiden välittäminen, mutta esimerkiksi sähköpostiosoitteita, puhelinnumeroita ja

MECARD-muotoisia yhteystietoja pystyy yleisesti välittämään QR-koodien avulla(Zxing-projektin wiki, 2010). Tämä mahdollistaa esimerkiksi suoraan käyntikortista yhteystietojen kopioinnin matkapuhelimen osoitekirjaan. Lisäksi QR-koodin virheenkorjausominaisuuksia voi käyttää hyödyksi esimerkiksi upottamaan logoja tai muita kuvioita osaksi koodia kuvan 2.3 tapaisesti. Tätä ominaisuutta ja houkuttelevampaa graafista ilmettä koodille tarjoavaa palvelua myydään design QR[1] -nimellä ja he käyttävät tarkemmin määrittelemätöntä menetelmää kuvioiden upottamiseen osaksi koodia.

Kaikkia standardin tarjoamia mahdollisuuksia ei käytännössä hyödynnetä kaikissa sovelluksissa johtuen usein laitteiston tai toteutuksen aiheuttamista rajoituksista. Matkapuhelimien kameroissa optiikan laatu ja kameran kennon tarkkuus rajoittaa käytännössä QR-koodin versiota ja esimerkiksi NTT Docomon i-mode -yhteensopivissa puhelimissa vaaditaan tuki vain versioon 10 asti(NTT DOCOMO, 2010). Lisäksi QR-koodin ISO-standardista on varhaisempi versio vuodelta 2000, tai vastaava japanilainen JIS X 0510 -standardi, jotka eivät määrittele koodin peilausta ja käänteisiä värejä sallituiksi muunnoksiksi, jolloin näiden standardien pohjalta tehdyt lukijat eivät välttämättä osaa ottaa näitä tilanteita huomioon.

[1] http://d-qr.net/

Luku 3

Koodin lukemista hankaloittavat ilmiöt

Kun QR-koodista otetaan kameralla kuva, niin kuvassa voi esiintyä kuvaustilanteesta ja -laitteistosta johtuvia erilaisia koodin tunnistusta haittaavia ilmiöitä, joiden seurauksena kuvaa joutuu käsittelemään tai tunnistusalgoritmeja täytyy muokata ottamaan huomioon halutut ilmiöt. Tässä luvussa esitellään koodin tunnistamista vaikeuttavia ilmiöitä ja niiden vaikutuksia tunnistusongelmaan. Lopuksi käydään läpi mitä kompromisseja tunnistusalgoritmien toteutuksessa täytyy tehdä riippuen tunnistustilanteesta ja -vaatimuksista.

3.1 Kuvan tarkkuus

Kuvan tarkkuus vaikuttaa tunnistustehtävän osalta siihen, kuinka suuren kuva-alan koodi voi peittää että se on vielä tunnistettavissa ja minkä kokoisena yksi moduuli esiintyy. Tähän liittyy myös se kuinka suuren kuva-alan koodi kattaa. Ohbuchi et al. (2004) käyttivät testilaitteistonaan 320x240 pikselin kokoisia kuvia tuottavaa matkapuhelimen kameraa, kun taas nykyään puhelimien kamerat kykenevät tuottamaan yli kymmenkertaisia kuvia sivukooltaan (Nokia, 2010). Tällainen kuvan tarkkuudesta, koodin versiosta ja kuva-alan peittävyydestä johtuva moduulikoon vaihtelu vaikuttaa sekä kuvan käsittelyaikaan että mahdollisiin tunnistusalgoritmin vaiheiden parametrien valintaan, esimerkiksi morfologisten operaatioiden maskien kokojen ja muiden kuvaa suodattavien menetelmien osalta. Alfthan (2008) mainitsee normaalin QR-koodeja tunnistavan algoritmin vaativan minimimoduulikooksi 2,5–3

(a) Sumeneminen.

(b) Epätasainen valaistus.

(c) Vinjetointi.

(d) Perspektiivi- ja muotoääristymä.

(e) Koodialueen peittyminen.

(f) Kohina.

Kuva 3.1: Kuvassa esiintyviä tunnistamista haittaavia ilmiöitä.

kuvapistettä sivultaan.

Koodialueen koon ja sijainnin vaihtelua voi pyrkiä vähentämään ohjaamalla käyttäjää kohdistamaan koodialuetta oikeaan kohtaan. Tämä onnistuu matkapuhelinkameroiden tapauksessa siten, että koodialueen keskelle piirretään suorakaide, mikä ohjaa käyttäjää kohdistamaan koodialueen tähän suorakaiteeseen. Lisäksi kameralta voi pyytää jonkin tietyn resoluution kuvan tai mahdollisesti muuttaa kuvan kokoa kuvaamisen jälkeen osana esikäsittelyvaihetta, jolloin voidaan paremmin ennustaa laskentavaatimuksia. Tarvittaessa voidaan myös rajoittaa tunnistettavaa QR-koodin versiota, mikä matkapuhelinsovelluksissa usein rajoitetaan versioihin 1–10 (Yue et al., 2008; NTT DOCOMO, 2010).

3.2 Kohina

Nykyisillä CCD-kennoihin (charge-coupled device) perustuvilla kameroilla kuvatessa aiheutuu eri tyyppisistä lähteistä kohinaa kuvaan. Kuluttajalaitteissa voimakkaimpana kohinanaiheuttajana on yhdelle kuvakennolle muodostuvien korkeamman energiatilan elektronien määrän vaihtelu (Bovik, 2005). Tämä on ongelma erityisesti otettaessa kuvia heikossa valaistuksessa ja valotusaikaa kasvattamalla voidaan vähentää kohinaa, mutta kuvauslaitteen mahdollinen liike taas pääsee vaikuttamaan voimakkaampana kuvan sumenemisena.

Kohina vaikuttaa erityisesti reunantunnistukseen ja muihin tasaisia alueita olettaviin operaatioihin kun moduulin koko on pieni ja tätä voidaan vähentää suodattamalla kuvaa joko erikseen tai osana jotain kuvankäsittelyllistä operaatiota, kuten Cannyn reunantunnistimessa(Canny, 1986). Kuvassa 3.1f näkyy heikon valaistuksen seurauksena syntyvää kohinaa.

3.3 Sumeneminen

Sumeassa kuvassa erityisesti etsinkuvioiden haku ei toimi luotettavasti hakemalla etsinkuvioille ominaista suhdetta(Alfthan, 2008) ja muutenkin sumeneminen vaikuttaa erilaisten koodialueen yksityiskohtien tunnistamiseen. Alfthan (2008) tarkastelee työssään QR-koodialueen hakemista sumeasta kuvasta ja esittää että sumeneminen voi tapahtua niissä tapauksissa, joissa kuva ei ole tarkentunut, kamera on liikkunut kuvan ottamisen aikana, kameran linssi on likainen tai kuvan resoluutio on pieni. Lisäksi heikossa valaistuksessa otetussa kuvassa on pitkästä valotusajasta johtuen herkemmin liikkeestä johtuvaa sumenemista.

Monia sumenemisilmiöitä voi mallintaa kuvan ja erilaisten sumenemismaskien konvoluutiolla(Bovik, 2005) ja koodialueen tunnistamisen tapauksessa Chu et al. (2007) esittävät yhden menetelmän liikkeestä johtuvan sumenemisen havaitsemiseen ja korjaamiseen. Kuvassa 3.1a näkyy tarkentamattomuudesta johtuvaa sumenemista.

3.4 Epätasainen valaistus ja varjot

Epätasainen valaistus vaikuttaa erityisesti kuvan kynnystykseen käytettäessä globaaleita kynnystysmenetelmiä. Tämän seurauksena jos koodialue sijaitsee huonosti tai hyvin valaistulla alueella verrattuna muuhun kuvaan, koodialue

saattaa kynnystäytyä kokonaan samaan arvoon globaalilla kynnystysmenetelmällä ja tällöin kynnystystuloksen käyttäminen jatkokäsittelyyn ei enää toimi. Vastaavanlainen osittainen koodialueen katoaminen kynnystyksen seurauksena voi tapahtua jos kuvassa on kuvan 3.1b mukaisesti voimakasta valaistuksen vaihtelua tai mahdollisesti varjo osalla koodialueesta.

Epätasaisen valaistuksen ja varjojen vaikutusta voidaan pienentää joko tasoittamalla kuvan kirkkaus keskimäärin samankaltaiseksi(Hsia et al., 2006) tai käyttämällä jotain paikallista kynnystysmenetelmää. Vaihtoehtoisesti kuvan valaistusominaisuudet voidaan arvioida kuvan histogrammista ja niiden perusteella valita kynnystykseen käytettävä menetelmä (Yue ja Mingjun, 2006; Ming et al., 2009).

3.4.1 Vinjetointi

Vinjetointi on linssistä johtuvaa kuvan kirkkauden tummenemista kuvan reunoja kohti (Sonka et al., 2007) ja tämä voi olla joillakin matkapuhelinten kameroilla hyvinkin voimakasta (Wagner ja Schmalstieg, 2007). Vinjetointi vaikuttaa erityisesti kuvan kynnystykseen epätasaisen valaistuksen tapaisesti ja vinjetointi voidaan ottaa huomioon joko erillisenä ilmiönä tai osana epätasaisen valaistuksen korjaamista. Esimerkki keinotekoisesti aiheutetusta voimakkaasta vinjetoinnista näkyy kuvassa 3.1c.

3.5 Epätasainen tausta

QR-koodia reunustaa vähintään neljän moduulin paksuinen rauhoitettu alue(ISO, 2006), mutta muuallakin kuvassa voi esiintyä eri tunnistusmenetelmillä esiintuotavia ilmiöitä. Esimerkiksi jos QR-koodi sijaitsee jollain tekstiä sisältävällä aluetta ja koodialuetta haetaan hyödyntämällä tietoa koodialueen suuresta reunantiheydestä, niin esimeriksi konekirjoitettua tekstiä sisältävä tausta voi sopia näihin kriteereihin ja asettua ehdokasalueeksi. Samoin morfologisten operaatioiden käytön seurauksena koodialue voi sulautua ympäröivään vaihtelevaan taustaan (Alfthan, 2008).

3.6 Vääristymät

Kuvassa voi olla pääasiassa erilaisia koodin muodosta ja sijainnista kameraan nähden johtuvia vääristymiä, jotka tekevät kuvassa näkyvästä koodialueen muodosta jonkin muun kuin neliön. Yleisin luvussa 4 otetuissa artikkeleissa esitelty vääristymä on kuvakulmasta johtuva lineaarinen koodialueen nelikulmiona pitävä vääristymä. Lisäksi jos koodialue on esimerkiksi sylinterin pinnalla tai rypistettynä kuvan 3.1d tapaisesti, niin koodialuetta ei saa suoraan yhdellä lineaarisella muunnoksella korjattua. Ohbuchi et al. (2004) esittelee myös taivutetun koodialueen huomioonottavan korjausmenetelmän periaatteen ja QR-koodin standardissa määritellyt kohdistuskuviot auttavat pienentämään erilaisten vääristymien vaikutuksia koodin uudelleenrakentamisen vaiheessa.

3.7 Koodialuetta peittävät ilmiöt

Koodialue voi olla osaksi toisen kappaleen peittämä. Tällöin mitkään kuvaan kohdistettavat korjaukset eivät pysty auttamaan koodialueen datan lukemisessa, vaan tällöin täytyy luottaa QR-koodin virheenkorjausominaisuuksiin, jotta koodi tunnistuisi oikein. Lisäksi koodialuetta peittävien kappaleiden vaikutuksen voimakkuus riippuu tunnistusmenetelmästä, vaikka QR-koodin virheenkorjaus sietäisikin kyseisen virheen. Esimerkiksi jos koodialuetta haetaan konveksin nelikulmion muotoisena ja koodin muoto muuttuukin peittävän kappaleen takia useampikulmaiseksi ei-konveksiksi kappaleeksi, tunnistin ei pysty tunnistamaan koodialuetta. Sama ongelma ilmenee jos koodialueen tunnistus perustuu pelkästään kolmen etsinkuvioin hakuun ja jokin etsinkuvioista on peittynyt.

Jos koodialue on tehty heijastavasta materiaalista tai sijaitsee heijastavan materiaalin alla, niin tällöin tästä aiheutuvat voimakkaat heijastumat saattavat myös aiheutta koodialuetta peittävän vaikutuksen. Kuvassa 3.1e näkyy salaman aiheuttama heijastuma, joka peittää koodialueen sen verran suurelta alueelta, että tämä tekee koodista lukukelvottoman.

3.8 Tunnistustilanteen asettamat rajoitukset

Kun QR-koodia tunnistetaan, niin tunnistustilanne asettaa usein erilaisia rajoituksia siihen, mitä kaikkea koodin tunnistamisessa täytyy ottaa huomioon ja mitä kaikkea tunnistamisen vaiheessa ehtii tekemään. Koodin lukemisessa ensiksi täytyy saada otettua kuva, jolloin erilaiset tässä luvussa luetelluiksi tulleet ilmiöt vaikuttavat kuvan ottamiseen ja siihen millainen kuva muodostuu.

Lisäksi kuvaustilanteessa käytettävissä oleva laitteisto ja aikavaatimukset vaikuttavat siihen mitä kaikkea kuvalle voi tehdä sen ottamisen jälkeen.

Kuvaustilanteessa kuvan laatuun vaikuttavat kameran optiikan ja CCD-kennon ominaisuudet, valaistus, koodialueen sijoittelu ja koodialueen materiaali. Valaistuksen vaikutus riippuu siitä, onko kuvaamislaite aktiivisesti valaiseva laite vai ympäristön valaistuksesta riippuva passiivinen laite. Aktiivinen laite voi valaista esimerkiksi salamavalon avulla tai viivakoodien lukemiseen tehtyjen lukijalaitteiden tapauksessa valaistuksessa voidaan tuottaa jonkin tietyn aallonpituusalueen valoa, joka on yhteensopiva koodialueen materiaalin heijastavuuden kanssa(ISO, 2006). Koodialueen tai kameran sijoittelu vaikuttaa siihen, että pystyykö sijoittelua muuttamalla vaikuttamaan valaistukseen, koodialueen kokoon, koodialueen vääristymiin tai sumenemiseen. Koodialueen materiaali taas vaikuttaa siihen että aiheutuuko materiaalista voimakkaita heijastuksia, jotka peittävät osan koodialueesta.

Kuvan ottamisen jälkeen alkaa koodialueen tunnistaminen ja tunnistuksen tuloksen perusteella toimiminen. Koodialueen tunnistamiseen käytettävissä oleva aika ja tarkkuusvaatimukset riippuvat kuvaustilanteesta ja laitteistosta ja vaikuttavat sitä kautta mitä kaikkea QR-koodin tunnistava algoritmi voi tehdä. Jos käsitellään etukäteen otettuja kuvia ja niiden käsittelyyn ei ole mitään erikoisempia aikavaatimuksia, niin voidaan keskittyä tekemään algoritmeista suureen tunnistamistarkkuuteen pyrkiviä ajankäytön kustannuksella. Jos koodialueen tunnistus täytyy olla lähes reaaliaikaista suoraan matkapuhelimen kameran videokuvasta, niin tunnistustarkkuudessa voidaan tinkiä käsittelyn nopeuden kustannuksella. Käyttäjää voidaan tässä tapauksessa ohjata tässä tapauksessa kuvaamaan koodialuetta niin kauan, että se tunnistuu oikein. Lisäksi matkapuhelimissa käytössä olevat viivakoodeja tunnistavat sovellukset pystyvät mahdollisesti tunnistamaan useita eri viivakoodityyppejä kuvasta, jolloin käytettävissä oleva tunnistusaika on usein hyvin rajallinen. Lisäksi tunnistusajan kestoon vaikuttaa käytettävissä olevan laitteiston tarjoama suoritinteho ja sen vaihtuvuus eri ohjelman tukemissa laitteissa.

Luku 4

Aikaisempaa tutkimusta QR-koodin lukemisesta

QR-koodin tunnistaminen kuvasta on kuvankäsittelyllinen ongelma, jonka ratkaisemiseksi löytyy lukuisia erilaisia menetelmiä, jotka soveltuvat muuhunkin kuin pelkästään QR-koodin tunnistukseen. Tässä luvussa esitellään erityisesti QR-koodin tunnistamiseen liittyvää tutkimusta ja menetelmiä, joista muodostuu luvun 5 sisältö. Näiden esiteltyjen menetelmien lisäksi olisi mahdollista tarkastella muiden kaksiulotteisten viivakoodityyppien tunnistamisesta ja olemassaolevien QR-koodia tunnistavien avoimen lähdekoodin sovellusten toimintaa, mutta näitä ei tässä juurikaan käsitellä laajuuden rajaamiseksi.

Kaikissa tutkimuksissa käsitellään harmaasävykuvia, joiden muodostamismenetelmää ei mainita tai jotka muodostetaan luminanssikanavan arvojen perusteella. Tämän takia mahdollinen harmaasävymuunnosvaihe jätetään pois menetelmäkuvauksista. Lisäksi lopuksi tehtävä moduulien arvojen luku on usein sivuutettu artikkeleissa määrittelemättä tarkemmin menetelmää, mutta ISO-standardi antaa tähän tavan, joten tämäkin sivutetaan, ellei tätä erikseen mainita artikkelissa.

Algoritmikuvauksissa esiintyy viittaus neljänteen kulmapisteeseen, mikä on perusmuotoisen koodialueen oikea alakulma. Tässä kohdassa ei välttämättä ole mustaa moduulia, joten tätä neljättä kulmaa ei yleensä saa suoraan haettua. Tästä johtuen neljäs kulmapiste on virtuaalinen kulmapiste verrattuna kolmeen muuhun koodialueen kulmapisteeseen, jotka saadaan hakemalla etsinkuvioita.

4.1 Ohbuchi, Hanaizumi ja Hock 2004

Ohbuchi et al. (2004) olettavat että koodialue sijaitsee noin suunnilleen kuvan keskellä ja hakevat koodialueen rajat pyyhkäisemällä kuvan ulkolaidoilta sisälle kahdeksaan eri suuntaan 45 asteen välein. Tämän menetelmän tarkoituksena on olla riittävän tehokas, että koodin reaaliaikainen tunnistaminen onnistuu testiympäristön hitaalla prosessorilla varustetulla laitteella. Tämä menetelmä sisältää seuraavat vaiheet:

- kuvan kynnystys
- kuvan koon puolitus
- koodialueen kulmapisteiden haku pienennetystä kuvasta
- kulmapisteiden sijaintien tarkennus alkuperäisestä kuvasta
- neljännen kulman haku
- käänteisperspektiivimuunnos.

Tässä kuvan kynnystys tapahtuu valitsemalla kuvasta 60x60 pikselin kokoinen alue läheltä kuvan keskustaa, jakamalla tämä alue 9 osaan, laskemalla kaikille näille osille erikseen mediaanitummuus ja valitsemalla globaaliksi kynnystysrajaksi pienin näistä mediaaneista. Seuraavaksi kuvan koko puolitetaan koodialueen kulmien hakemista varten ja koon puolittamisessa yksi puolitetun kuvan piste muodostetaan valitsemalla tummin alkuperäisen kuvan neljästä kuvapisteestä. Koodialueen kulmien paikat määritetään etsinkuvioiden avulla puolitetusta kuvasta pyyhkäisemällä viivoilla kuvan yli kahdeksaan eri suuntaan, mutta tästä kohdasta lähtien algoritmikuvaus on epäselvä. Näiden arvioiden perusteella tarkennetaan kolmen etsinkuvien uloimman kulman paikat alkuperäisestä kuvasta ja haetaan neljäs kulma olettamalla tälle paikka alueen ulkopuolelta ja viemällä sitä kohti alueen keskipistettä, kunnes neljännen kulman ja kahden muun kulman väliset janat koskettavat koodialuetta. Tämän jälkeen koodialueen moduulien arvot luetaan käänteisperspektiivimuunnoksen kanssa samanaikaisesti.

4.2 Yue ja Mingjun 2006

Yue ja Mingjun (2006) keskittyvät tunnistusmenetelmän lisäksi on kehittämään kynnystysalgoritmia, joka on nopea ja selviää epätasaisista valaistusolosuhteista. Artikkeli on vajaa varsinaisten toteuttavien menetelmien suhteen, mutta kuvaa seuraavat vaiheet tunnistusmenetelmässä:

- kuvan kynnystys
- etsinkuvioiden haku
- (kuvan oikaisu)
- kohdistuskuvioiden haku.

Kuvan kynnystyksessä lasketaan ensiksi kuvan suodatettu histogrammi ja histogrammin huipukkuuden perusteella päätetään kynnystysmentelmä. Yksi- tai kaksihuippuisissa histogrammeissa käytetään globaaleja kynnystysmenetelmiä ja monihuippuisen histogrammin tapauksessa kuvan tulkitaan olevan joko epätasaisessa valaistuksessa tai monimutkaisella taustalla ja tällöin käytetään jotain paikallista kynnystysmenetelmää.

Etsinkuvioiden haku tapahtuu hyödyntämällä etsinkuvioiden 1:1:3:1:1 mustan ja valkoisen vuorottelevaa suhdetta. Kun etsinkuvioiden keskikohdat on löydetty, kuvalle ehkä suoritetaan jo tässä vaiheessa käänteisperspektiivimuunnos tai vasta kohdistuskuvioiden hakemisen jälkeen, mutta tämä ei käy selväksi tekstistä. Käänteisperspektiivimuunnoksen pisteiden selvittämisessä käytetään tietoa etsinkuvioiden keskikohtien yhdistävien janojen muodostaman kolmion suurimmasta kulmasta, minkä oletetaan olevan perusasentoisen QR-koodin vasemman yläkulman etsinkuvion kohdalla. Kohdistuskuvioiden haku suoritetaan ennustamalla kohdistuskuvion paikka ja tutkimalla kahdeksaan eri suuntaan tämän paikan lähiympäristöstä sopivia ehdokaspisteitä.

4.3 Aidong, Yan ja Caixing 2007

Aidong et al. (2007) esittelevät algoritmin QR-koodin tunnistamiseksi, missä tunnistaminen perustuu etsinkuvioiden ulkoisten ääriviivojen hakuun (external contour detection). Tässä menetelmässä on seuraavat vaiheet:

- reunantunnistus Cannyn reunantunnistimella
- ulkoisten ääriviivojen haku
- käänteisperspektiivimuunnos arvioidun neljännen kulmapisteen avulla
- neljännen kulmapisteen sijainnin tarkennus
- käänteisperspektiivimuunnos.

Algoritmin ideana on käyttää Cannyn reunantunnistimen kohinanvähennysominaisuutta suodataamaan kuvaa ja käyttää reunainformaatiota muodostamaan kuvaan alueille vertekseistä koostuvat ääriviivat, joiden avulla määritetään etsinkuvioiden sijainnit. Tämän jälkeen

haetaan viereisyyspuun (adjacency tree) avulla uloimmat alueiden reunaviivat, joista tämän jälkeen valitaan etsinkuvioehdokasalueiksi vain konveksit 4 verteksiä sisältävät alueet, joiden kahden verteksin välinen maksimikulma on korkeintaan 120 astetta ja minimikulma on 70 astetta. Tämän jälkeen kuvassa oletetaan olevan kolme etsinkuvioista aiheutuvaa suunnikasaluetta, joita käytetään arvioimaan ensiksi neljäs kulmapiste, jonka avulla tehdään ensimmäinen käänteisperspektiivimuunnos. Tämän jälkeen neljännen kulmapisteen sijaintia tarkennetaan ja tehdään lopullinen käänteisperspektiivimuunnos.

4.4 Chu, Yang ja Chen 2007

Chu et al. (2007) ottavat huomioon kameran tärähdyksestä johtuvan suoraviivaisen liikkeen osana tunnistusprosessia. Menetelmässä on seuraavat vaiheet:

- sumenemisparametrien arviointi
- sumenemisen korjaus
- kuvan kynnystys
- reunantunnistus
- reunojen dilaatio
- alue-ehdokkaiden merkitseminen
- etsinkuvioiden haku ehdokasalueilta
- käänteisperspektiivimuunnos
- koodiarvojen luku.

Tässä algoritmissa oletetaan, että kuvaan tulee suoraviivaisesta liikkeestä johtuvaa sumenemista ja ensimmäinen vaihe on arvioida sumenemisen suuruus ja suunta muuntamalla kuva kepstritasoon ja hakemalla kepstritasosta negatiivisia piikkejä Hough-muunnoksen avulla. Näiden arvioitujen parametrien avulla sumeneminen korjataan kirjoittamalla konvoluutio lohkollisen Toeplitz-matriisin avulla ja muodostamalla tästä regularisoitu minimoitava yhtälö.

Seuraavaksi kuva kynnystetään ja kynnystetystä kuvasta tunnistetaan reunat tarkemmin määrittelemättömin algoritmein. Tämän jälkeen näihin reunaviivoihin kohdistetaan morfologinen dilaatio, minkä avulla saadaan ehdokasalueet etsinkuvioiden hakua varten. Etsinkuvioita haetaan ehdokasalueilla hakemalla vaakatasossa etsinkuvioiden 1:1:3:1:1 mustan ja

valkoisen vuorottelevaa suhdetta. Tämän jälkeen kuva korjataan käänteisperspektiivimuunnoksella käyttäen tietoa koodialueen kulmista, mutta kulmien sijaintien hakemista ei kuvata. Tämän jälkeen kuvasta lasketaan koodin versio ja moduulin leveys ja moduulin arvo päätetään sen perusteella ylittääkö lasketulla moduulin alueella olevien mustien pisteiden määrä jonkun ennaltamäärätyn raja-arvon.

4.5 Chang, Chu ja Chen 2007

Chang et al. (2007) käsittelevät erityisesti kuvan epätasaisen valaistuksen tasoittamista parantamaan kynnystystulosta. Menetelmä sisältää seuraavat vaiheet:

- kuvan valoisuuden tasoitus
- kynnystys
- etsinkuvioiden haku
- kuvan jakaminen osiin
- reunantunnistus
- etsinkuvioiden perusteella alueiden ja niiden kulmien etsiminen
- käänteisperspektiivimuunnos.

Artikkelin ideana on ensiksi tasoittaa artikkelin (Hsia et al., 2006) mukaisesti kuvan valaistus pyyhkäisemällä kuvan yli vaakasuorilla alueilla, mitkä on aluksi jaettu N:nään osaan ja mistä lasketaan vaakasuora kirkkauden muutos ja jos kirkkaus muuttuu liikaa alueen sisällä, näitä alueita jaetaan vielä pienempiin osiin. Tämän jälkeen jokaisen jaetun alueen kirkkautta muutetaan lineaarisesti alueen rajojen suhteen muuttamalla kuvapisteen kirkkautta riippuen sen sijainnista vaakatasossa.

Tämän jälkeen kuva kynnystetään valitsemalla kynnystysarvoksi harmaasävyjen puoliväli. Kynnystetystä kuvasta haetaan etsinkuviot etsimällä 1:1:3:1:1 mustan ja valkoisen vuorottelevaa suhdetta vaaka- ja pystytasossa, mikä muodostaa etsinkuvioihin ristin muotoisen alueen. Näistä päätellään moduulin koko ja kuva segmentoidaan CRLA-algoritmilla(Wahl et al., 1982) arvioidun moduulikoon perusteella vaaka- ja pystytasossa, millä saadaan ehdokaskoodialueet.

Viivakoodialueiden hakemisen jälkeen alueiden reunat selvitetään Cannyn reunantunnistimella, minkä tulosta lähdetään tutkimaan viivakoodialueiden ääriviivojen selvittämiseksi, joista saadaan alueen kulmapisteet tietoon käänteisperspektiivimuunnosta varten, minkä he rajoittavat vain alueille, joiden

pisimmän ja lyhimmän sivun suhde on korkeintaan 1.5. Chang ja kumppanit tekevät käänteisperspektiivimuunnoksen alkuperäisen kuvan kuvapisteitä hyödyntäen, mikä jättää käänteismuunnettuun kuvaan aukkoja, jotka he sitten täyttävät CRLA-algoritmilla.

4.6 Yue, Bo ja Yu 2008

Yue et al. (2008) esittelevät suhteellisen yleisen tason lähestymistavan QR-koodin tunnistamiseksi ja jättävät varsinaisen menetelmän toteutuksen suhteellisen epäselväksi. Yhtenä tutkijana tässä on luvun 4.2 algoritmin(Yue ja Mingjun, 2006) kehittänyt henkilö. Tämä algoritmi menee seuraavasti:

- harmaasävykuvan suodatus morfologisella sulkeumalla
- koodialueen summittainen haku
- koodialueen kynnystys Otsun menetelmällä
- koodialueen tarkan suunnan ja sijainnin haku etsin- ja ajastuskuvioiden perusteella
- kohdistuskuvioden haku jakamalla koodialue pieniin palasiin.

Ensiksi harmaasävykuvalle tehdään morfologinen sulkeuma vähentämään kuvassa olevan kohinan vaikutusta. Seuraavaksi koodialue arvioidaan jotenkin hakemalla nopeasti vaihtuvia tummien ja vaaleiden pisteiden alueita. Tämän jälkeen koodialue kynnystetään Otsun menetelmällä(Otsu, 1979). Tämän jälkeen on tarkoitus hakea etsinkuvioita niiden 1:1:3:1:1 mustan ja valkoisen vuorottelevan suhteen perusteella ja vielä hyödyntää ajastuskuviota tässä haussa, jos yksi etsinkuvio on lukukelvoton. Tämän jälkeen haetaan kohdistuskuvioita arvioidusta pisteestä kahdeksaan eri suuntaan ja etsitään ehdokaspisteitä hakemalla kohdistuskuvion 1:1:1:1:1 mustan ja valkoisen vuorottelevan suhteen perusteella. Lisäksi kohdistuskuvioiden etsimisessä otetaan huomioon se, että mustaa keskipistettä ympyröivää valkoista aluetta ympyröi vielä musta alue.

4.7 Alfthan 2008

Alfthan (2008) keskittyy koodialueen etsimiseen sumeasta kuvasta, missä yhden moduulin koko on vähän yli yhden kuvapisteen, jolloin esimerkiksi koodialueen haku etsinkuvioita kuvasta hakemalla ei juurikaan toimi. Hän käsittelee

yksinkertaistettua havainnointitilannetta, missä koodi kuvataan ylhäältä päin hyvässä valaistuksessa siten, että koodialue on aina neliön muotoinen. Tältä alueelta on tarkoitus pelkästään hakea koodialue, jonka lukeminen voidaan jättää myöhemmin suoritettavan tutkimuksen huoleksi. Varsinaista koodialuetta haetaan kahdella eri menetelmällä, joiden tulokset yhdistetään lopullisen havainnointituloksen saavuttamiseksi.

Ensimmäisen menetelmän, gradienttimenetelmän, tarkoitus on hakea kuvasta ehdokasalueita, joita toisella menetelmällä voidaan tarkemmin hakea ehdokasalueiksi:

- Sobel-maskien soveltaminen kuvaan paikallisen vaihtelun esiintuomiseksi
- morfologinen dilaatio ja eroosio eri maskein
- ennaltamäärätyn rajan ylittävien ehdokasalueiden haku.

Tässä menetelmässä kuvaan käytetään neljää eri Sobel-maskia, jotka kynnystetään ennaltapäätetyllä gradientin voimakkuuden raja-arvolla. Tämän jälkeen jokaiselle maskille tehdään erikseen morfologinen dilaatio 2 kuvapisteen kokoisilla maskeilla gradientin suuntaisesti ja dilatoituihin alueisiin ja eroosio isommalla neliömäisellä maskilla. Tämän jälkeen käydään yhtenäiset alueet läpi ja valitaan koodin ehdokasalueiksi ne joiden koko ylittää tietyn ennaltamäärätyn koon.

Toisessa menetelmässä suoritetaan neliön muotoisten alueiden haku, minkä tarkoituksena on varmistaa, että aikaisemmin esitellyssä menetelmässä löydetyt ehdokasalueet ovat neliön muotoisia:

- morfologinen eroosio
- reunantunnistus Cannyn reunantunnistimella
- neliön muotoisten alueiden haku Hough-muunnoksella.

Kuvan käsittely aloittaa morfologisella eroosiolla, millä koodialueen sisältä on tarkoitus poistaa reunat hyödyntäen koodin suhteellisen tasaista tummien ja vaaleiden pisteiden jakaumaa. Tämän jälkeen kuvaan kohdistetaan Cannyn reunantunnistin, millä tarkoituksena on saada koodialueiden reunat näkyviin, sillä eroosion jälkeen koodialueiden sisältä ei pitäisi enää reunoja juurikaan löytyä. Kuvalle tehdään tämän jälkeen Hough-muunnos, jolla kuvasta haetaan neliön muotoisia alueita. Näiden alueiden hakemisen jälkeen alueita verrataan gradienttimenetelmällä löydettyihin alueisiin ja jos ne ovat riittävän samanlaisia, kuvassa tulkitaan olevan QR-koodi.

Luku 5

QR-koodin lukemisessa käytettäviä menetelmiä

Luvussa 4 esiteltiin eri tapoja lähestyä QR-koodin lukemisongelmaa. Tässä luvussa käydään tarkemmin läpi eri tunnistamisen vaiheissa käytettäviä menetelmiä ja QR-koodin ominaisuuksia, joita voidaan käyttää hyödyksi tunnistamisen ja lukemisen osana. QR-koodin lukeminen voidaan karkeasti jakaa neljään eri vaiheeseen, joita ovat kuvan esikäsittely, koodialueen haku, koodialueelle suoritettavat korjaukset ja moduulien arvojen lukemiseen. Eri menetelmien luokittelu kuhunkin vaiheeseen on karkeaa ja riippuu täysin kuinka niitä sovelletaan osana tunnistamista.

Kameralla otettu kuva on oletuksena kolmikanavainen värikuva, joka saatetaan helpommin käsiteltävään muotoon harmaasävymuunnoksella. Tämän jälkeen kuvalle voidaan tehdä muita esikäsittelyvaiheita, kuten sumenemiskorjaus, kynnystys, kuvan koon pienentäminen, reunojentunnistus ja morfologinen dilaatio ja eroosio. Tämän jälkeen voidaan osana koodialueen hakua suorittaa kuvalle Hough-muunnos neliöin muotoisten alueiden etsimiseksi, reunojentunnistus koodialueiden ääriviivojen määrittämiseksi tai hakea etsinkuvioita joko suoraan kuvasta tai koodialue-ehdokkailta. Kun koodialue on saatu määriteltyä, niin tämän jälkeen koodialue voidaan suoristaa käänteisperspektiivimuunnoksella tai jos koodialuetta etsitään pienempiresoluutioisesta kuvasta, niin sen paikan voi tarkentaa alkuperäisestä kuvasta ja suorittaa vasta tämän jälkeen korjaukset. Kohdistuskuvioiden hakemisella voidaan koodialuelle tehtävät korjaukset saattaa vähemmän karkeiksi ja täten parantaa lukemisen onnistumista. Ennen lukemisvaihetta on myös mahdollista suorittaa kynnystys, jos koodialue luetaan sellaisenaan harmaasävykuvasta ja tällöin kynnystämisessä ei välttämättä tarvitse ottaa

huomioon niin rajuja valaistuksen vaihteluita. Lopulta koodialueelta luetaan moduulien arvot ja uudelleenrakennettua koodia voidaan tällöin lähteä käsittelemään QR-koodin standardin määrittelemin tavoin.

5.1 Harmaasävymuunnos

Monet kuvankäsittelyalgoritmit käsittelevät vain harmaasävykuvia perusmuodossaan, vaikka niiden laajentaminen onnistuisikin monispektrikuviin. Kameralla otetuissa kuvissa on yleensä punainen, vihreä ja sininen värikanava tekemässä kuvasta monispektrikuvan. Harmaasävymuunnokset voidaan jakaa globaaleihin ja lokaaleihin, joissa lokaaleissa muutoksissa on usein tarkoitus löytää ihmiselle miellyttävämpi muunnos(Cadik, 2008). QR-koodin tunnistamisessa tarkka tieto väreistä ei välttämättä ole olennaista, sillä QR-koodin tunnistamisessa kiinnostus on tummien moduulien erottamisessa vaaleista, vaikka koodissa käytettäisiin muita värejä kuin mustaa ja valkoista .

QR-koodista otettu kuva muutetaan kaikissa luvussa 4 esiintyneissä lähteissä harmaasävymuotoon. Chang et al. (2007) antavat NTSC-televisiostandardin mukaisen lineaarisen harmaasävymuunnoksen YIQ-väriavaruuden luminanssikanavan Y avulla. YIQ-väriavaruuden luminanssikanava on $Y = 0.29889531 * R + 0.58662247 * G + 0.11448223 * B$, missä alkuperäisen kuvan värikanavia ovat ovat: R punainen, G vihreä ja B sininen. Tämän muunnoksen avulla RGB-kuvasta 5.1a harmaasävykuvaksi muunnettu kuva näkyy kuvassa 5.1b.

5.2 Kuvan kynnystys

Kuvan kynnystyksellä kuva muutetaan kaksiarvoiseksi binäärikuvaksi, millä tarkoituksena on erottaa tummat alueet vaaleista, tai vaihtoehtoisesti tausta itse kappaleesta. Tämä mahdollistaa esimerkiksi helpomman etsinkuvioiden haun ja moduulien arvojen lukemisen jossain vaiheessa QR-koodin tunnistusta(ISO, 2006). Kuvan kynnystysmenetelmät jakautuvat paikallisiin ja globaaleihin kynnystysmenetelmiin, joista globaaleissa kynnystysmenetelmissä koko kuvalle annetaan yksi kynnysarvo, jonka perusteella kuvapisteiden arvot jaetaan tummiin ja vaaleisiin pisteisiin. Paikallisissa kynnystysmenetelmissä taas kynnysarvo voi vaihdella riippuen kynnystettävän kuvapisteen ympäristöstä.

Kynnystys voidaan tehdä osana esikäsittelyvaihetta, mikä on useimmissa artikkeleissa suosittu tapa, tai vasta siinä vaiheessa kun koodin arvoja

(a) Alkuperäinen värikuva. (b) Harmaasävykuva luminanssikanavalla. (c) Otsun menetelmällä kynnystetty kuva.

Kuva 5.1: Koko kuvalle tehtäviä värimuunnoksia.

luetaan(Alfthan, 2008; Yue et al., 2008). Riippumatta kynnystyksen suoritusvaiheesta, globaalilla kynnystysmenetelmällä kynnysarvon valinta vaikuttaa kynnystystulokseen, erityisesti jos kuvan valaistus tai värit vaihtelevat paljon. Paikalliset menetelmät taas voivat selviytyä epätasaisesta valaistuksesta ja varjoista paremmin ja niissä algoritmeissa missä paikallisia menetelmiä ehdotetaan käytettäväksi, usein päätetään paikallisen ja globaalin menetelmän välillä kuvan histogrammmin huipukkuuden perusteella(Yue ja Mingjun, 2006; Ming et al., 2009).

QR-koodin ISO-standardin esimerkkialgoritmissa kynnystysmenetelmäksi ehdotetaan kirkkaimman ja tummimman kuvapisteen puoliväliä. Tässä esitetään lähteessä (Yue et al., 2008) ja kiinalaisen luotettavan koodin (Chinese sensible code) analyysissä(Ming et al., 2009) mainittu globaali Otsun kynnystysmenetelmä(Otsu, 1979; Morse, 2000), jolla myös kuva 5.1c on tuotettu. Otsun menetelmässä lasketaan ensiksi kuvan histogrammi ja sen perusteella valitaan yksi tai useampia kynnysarvoja kuvalle.

Otsun menetelmässä minimoidaan taustaan ja kappaleeseen kuuluvien histogrammiarvojen luokkien sisäisten varianssien painotettua summaa $\sigma^2_{sisainen}(T) = n_B(T)\sigma^2_B(T) + n_O(T)\sigma^2_O(T)$. Tässä kuvassa on L eri harmaasävytasoa väliltä $[0, L-1]$ ja T on kynnysarvo. Alaindeksi B viittaa kynnysarvon alittaviin pisteisiin, eli taustan pisteisiin ja alaindeksi O viittaa kynnysrajalla oleviin tai sen ylittäviin kuvapisteisiin, eli kappaleen pisteisiin. $\sigma^2_i(T)$ viittaa kunkin pistejoukon varianssiin ja

$$\begin{aligned} n_B(T) &= \Sigma_{i=0}^{T-1} p(i) \\ n_O(T) &= \Sigma_{i=T}^{L-1} p(i) \end{aligned},$$

missä $p(i)$ on intensiteetillä i olevien kuvapisteiden suhteellinen määrä. Tämän laskeminen jokaiselle intensiteettitasolle on raskasta, joten voidaan määritellä histogrammiarvojen luokkien ulkoisten varianssien summa:

$$\sigma^2_{ulkoinen}(T) = \sigma^2 - \sigma^2_{sisainen}(T) = n_B(T)n_O(T)[\mu_B(T) - \mu_O(T)]^2,$$

missä μ_i on joko kappaleeseen tai taustaan kuuluvien histogrammiarvojen keskiarvo. Tämä laskenta voidaan tehdä iteratiivisesti asettamalla $T = 1$, $n_B = n_0$, $n_O = 1 - n_0$ ja hakemalla suurinta $\sigma^2_{ulkoinen}$ arvoa seuraavien päivityssääntöjen avulla:

$$\begin{aligned} n_B(T+1) &= n_B(T) + n_T \\ n_O(T+1) &= n_O(T) - n_T \\ \mu_B(T+1) &= \tfrac{\mu_B(T)n_B(T)+n_T T}{n_B(T+1)} \\ \mu_O(T+1) &= \tfrac{\mu_O(T)n_O(T)-n_T T}{n_O(T+1)}, \end{aligned}$$

missä n_T on histogrammissa harmaasävytasolla T olevien kuvapisteiden suhteellinen määrä.

5.3 Liikkeestä johtuva kuvan sumenemisen korjaus

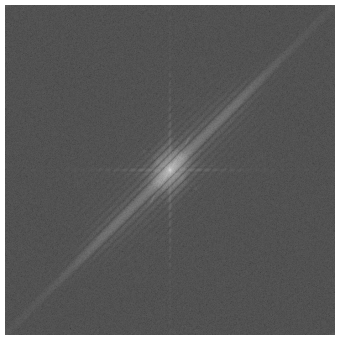
Otettaessa kameralla kuvaa tukemattomassa tilanteessa, saattaa kamera heilua johtuen käden epävakaudesta, tai jos kuvan ottaminen vaatii erikseen napin painamista, niin tämä saattaa myös aiheuttaa kevyillä kuvaamislaitteilla kameran liikkumista kuvaushetkellä. Lisäksi erityisesti matkapuhelinten kameroilla linssi ei välttämättä kykene kuvan automaattiseen tarkentamiseen ja näistä seikoista johtuen kuvaan voi tulla herkästi liikkeestä johtuvaa sumenemista.(Chu et al., 2007)

Yleinen menetelmäperhe kuvaa sumentavien ilmiöiden tunnistamiseksi ja korjaamiseksi tunnetaan sokeana dekonvoluutiona, missä sumennettu kuva muodostuu alkuperäisen kuvan kanssa konvoloituna sumentavana maskina ja

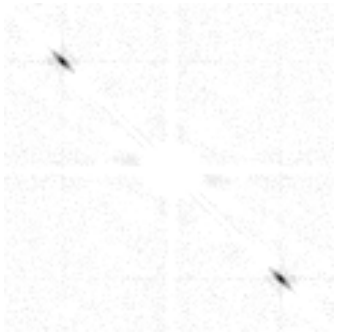
lisättynä kohinana. Tässä esitetään suoraviivaisesta liikkestä johtuvaa kuvan sumenemista kuvaava malli ja artikkeleissa (Chu et al., 2007; Lokhande et al., 2006) esiintyvä kepstritasoon perustuva menetelmä tämän sumenemismallin parametrien arvioimiseksi ja tämän mallin mukaisen sumenemisen korjaaminen.

(a) Alkuperäinen kuva.

(b) Sumennettu kuva.



(c) Sumennetun kuvan Fourier-tason arvot.



(d) Kepstritasossa näkyvät negatiiviset arvot origon lähellä.

(e) Sumenemiskorjattu kuva.

Kuva 5.2: Liikkeestä johtuva sumeneminen ja sen korjaus.

5.3.1 Sumenemisparametrien arviointi

Olettaen että kuvan sumeneminen aiheutuu suoraviivaisesta L:n pituisesta liikkeestä suuntaan θ kuvan valotusaikana, tätä voidaan mallintaa konvoluutiolla $g = f * h + n$, missä g on lopputuloksena saatava kuva, f on alkuperäinen sumenematon kuva, h on kuvan pisteenleviämisfunktio (point spread function, PSF) ja n on satunnaista nollakeskiarvoista normaalijakautunutta valkoista kohinaa (Bovik, 2005; Chu et al., 2007). Pisteenleviämisfunktion arvo on

$$h(x,y) = \begin{cases} \frac{1}{L}, & \text{kun } \sqrt{x^2+y^2} \leq \frac{L}{2} \text{ ja } \frac{x}{y} = -\tan\theta \\ 0 & \text{muulloin} \end{cases}$$

Kun tämä muutetaan Fourier-tasoon, niin maskin muunnoksen arvot ovat $H(u,v) = \text{sinc}(\pi L f(u,v))e^{-j\pi L f(u,v)}$ ja $|H(u,v)| = \text{sinc}(\pi L f(u,v))$, missä $f(u,v) = u\cos\theta + v\sin\theta$ (Lokhande et al., 2006). Pisteenleviämisfunktiolle voidaan laskea kepstri (cepstrum) $\text{C}(h) = \text{F}^{-1}(\log|\text{F}(h)|)$, missä F ilmaisee Fourier-muunnosta ja F^{-1} ilmaisee Fourier-käänteismuunnoksen. Tällöin jos kepstrin keskittää samoin kuin Fourier-muunnoksille tehdään, niin kepstrin origon läheisyyteen muodostuu kaksi negatiivista piikkiä etäisyydelle L origosta kulmassa $-\theta$ (Chu et al., 2007).

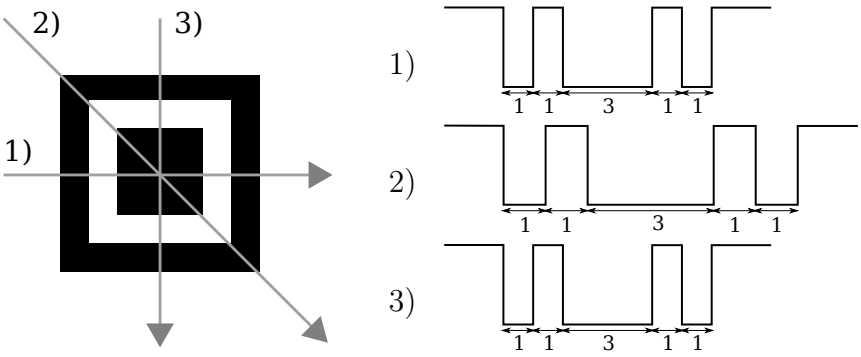
Kuva 5.2a on alkuperäinen kuva, minkä keinotekoisesti sumennettu versio näkyy kuvassa 5.2b. Sumennetun kuvan Fourier-muunnoksen logaritmiarvoinen vaste näkyy kuvassa 5.2c ja tästä voidaan nähdä jaksollisia $\text{sinc}(\pi L f(u,v))$-tyyppisiä saarekkeita. Sumennetun kuvan kepstrimuunnoksessa kuvassa 5.2d näkyy taas kaksi negatiivista piikkiä lähellä origoa sumenemisen suunnan ja pituuden mukaisessa kohdassa. Näiden paikat voidaan hakea esimerkiksi hakemalla origon kautta kulkevien suorien haulla Hough-muunnoksen avulla (Chu et al., 2007).

5.3.2 Sumenemiskorjaus

Dekonvoluutiopohjaiseen sumenemiskorjaukseen löytyy vaihtoehtoisia menetelmiä, jotka on johdettu eri lähtökohdista ja jotka pyrkivät erilaisiin korjaustoimenpiteisiin (Bovik, 2005). Lokhande et al. (2006) esittävät Wiener-suodatukseen perustuvan menetelmän liikkeestä johtuvan sumenemisen korjaamiseen, missä $\hat{F}(x,y)$ on sumenemiskorjatun kuvan Fourier-muunnos, joka lasketaan seuraavasti:

$$\hat{F}(x,y) = \frac{H^*(x,y)}{|H(x,y)|^2 + K} G(x,y),$$

missä $H(x,y)$ on kuvaa sumentavan maskin $h(x,y)$ Fourier-muunnos, $H^*(x,y)$ on tämän arvon kompleksikonjugaatti, $G(x,y)$ on sumentuneen kuvan Fourier-muunnoksen arvo ja K on jokin vakio. Tässä tässä ei oteta Wiener-suodatukselle tyypillisen alkuperäisen signaalin voimaspektrin arviointia, joten Lokhande et al. (2006) esittävät tälle vakiolle arvon $K = \frac{1}{w}$, missä w on kuvan leveys. Kuvassa 5.2e näkyy sumennetun kuvan 5.2b korjattu versio ja tämän sumenemiskorjausmenetelmän aiheuttamat häiriöt.



Kuva 5.3: Etsinkuvion keskipisteen lävistävät viivat aiheuttavat 1:1:3:1:1 suhteisen vasteen kirkkauden muutoksissa suunnasta riippumatta.

5.4 Viivakoodialueen haku

Varsinaisen viivakoodialueen haussa on eri artikkeleissa esitetty kaksi päämenetelmää, joko haetaan suoraan etsinkuvioita tai ensiksi haetaan ehdokkaita koodialueiksi ja tämän jälkeen ehdokasalueilta haetaan etsinkuviot. Eräs tapa jolla mahdollisia ehdokasalueita voidaan hakea esitetään lähteissä (Alfthan, 2008; Chu et al., 2007), joissa ensiksi tehdään kuvaan reunantunnistus ja tämän jälkeen hyödynnetään viivakoodialueiden suuria reunatiheyksiä kohdistamalla kuvaan esimerkiksi sopivan kokoinen morfologinen dilaatio tai eroosio. Vastaavasti (Chang et al., 2007) käyttää alueiden merkitsemiseen CRLA-algoritmia(Wahl et al., 1982). Näiden tarkoituksena on tuoda esiin mahdolliset viivakoodialueet, jotka voidaan ottaa tunnistamisen alue-ehdokkaiksi ja joille voidaan tehdä tarkempaa tutkimusta ilman, että koko kuvaa tarvitsee tutkia.

5.4.1 Etsinkuvioiden haku

Etsinkuviota hakemalla tarkoituksena on mahdollistaa viivakoodialueen nopea löytäminen (Hara et al., 1998) ja moduulikoon arviointi. Etsinkuvioiden haku tapahtuu hakemalla alueita, joissa on kuvan 5.3 mukainen 1:1:3:1:1-suhteinen vaste vuorottelevalle tummalle ja vaalealle värille. Koska varsinaisessa valokuvassa pisteet eivät jakaudu välttämättä näin, niin QR-koodin standardi(ISO, 2006) suosittelee 0,5:n yksikön toleranssia, jolloin etsitään leveyden 1 sijaan 0,5–1,5 ja leveyden 3 sijaan 2,5–3,5 yksikön kokoisia vasteita.

Etsinkuvioiden hakemiseen (Aidong et al., 2007) antaa algoritmin, jossa haketaan suoraan neljästä reunasta koostuvia konvekseja alueita, eikä huomioida mahdollista 1:1:3:1:1-suhdetta. (Chu et al., 2007; Ohbuchi et al., 2004) taas pyyhkäisevät viivoilla kuvaa eri suuntiin ja hakevat edellä mainittua etsinkuvioiden tumman ja vaalean suhdetta useammasta suunnasta ja yhdistävät tulokset paremman varmuuden saamiseksi.

5.4.2 Reunantunnistus

QR-koodissa vaihtelevat moduulien arvot suhteellisen tiheään tumman ja vaalean välillä, mikä tekee QR-koodin alueesta otollista reunantunnistuksen käyttämiselle. Aidong et al. (2007) käyttävät reunantunnistusta etsinkuvioiden hakuun ja Chang et al. (2007); Alfthan (2008) hyödyntävät tiheitä reunaesiintymiä viivakoodialuekandidaattien hakuun. Tässä esiteltävä Cannyn reunantunnistin(Canny, 1986) on optimaalinen reunantunnistin valkoisen kohinan seurauksena vääristyneille porrasreunoille(Sonka et al., 2007) ja edellämainittujen reunantunnistusta käyttävien menetelmien valinta reunantunnistimeksi. Cannyn reunantunnistimen vaiheiden toteutus vaihtelee hieman riippuen lähteestä(Sonka et al., 2007; Bovik, 2005) ja tämä kuvaus on yhdistelmä näiden kahden lähteen esityksestä.

Cannyn reunantunnistimessa ensimmäisenä vaiheena on vähintää kohinan vaikutusta konvoluoimalla kuva f gaussisella maskilla

$$G(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}},$$

missä σ^2 on valittavissa oleva maskin varianssi. σ^2 määrää minkä mittakaavan reunoja reunantunnistimella on tarkoitus tunnistaa. Lisäksi käytännön toteutuksissa tietyn koon jälkeen maskin reunoilla olevat kertoimet ovat niin lähellä nollaa, että konvoluution laskenta hidastuu turhaan, jos maskin kokoa ei ole rajoitettu.

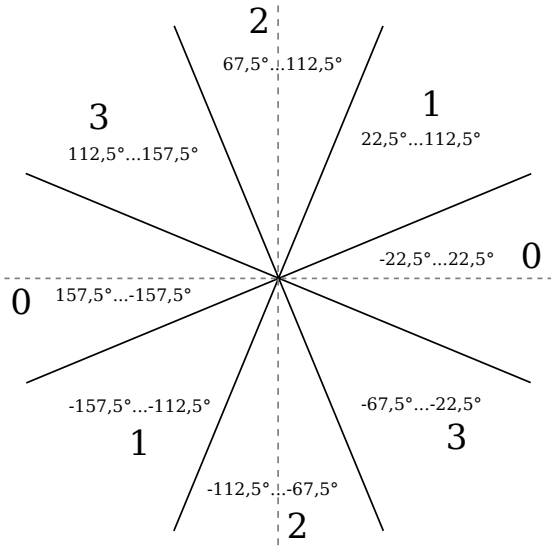
Seuraavaksi konvoluoidusta kuva $G * f$ arvioidaan kohtisuoraa reunaa vastaan olevan gradientin suunta jokaiselle kuvapisteelle laskemalla gradientti $\nabla(G * f)$. Gradientin voi laskea esimerkiksi konvoluoimalla kuva x- ja y-suuntaisilla suuntaisilla Sobel-maskeilla

$$h_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad h_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}.$$

Näistä voidaan muodostaa x- ja y-suuntaiset gradienttikuvat $f_x = h_x * (G * f)$ ja $f_y = h_y * (G * f)$. Gradientin suunta saadaan laskemalla jokaiselle kuvapisteelle kulma $\theta(x, y) = \tan^{-1} \frac{f_y(x,y)}{f_x(x,y)}$.

Seuraavaksi tarkoituksena on etsiä ehdokaspisteet reunoille ja muodostaa reunakartta ei-maksimiarvojen tukahduttamisella (non-maximal suppression). Ei-maksimiarvojen tukahduttamisella tarkoituksena on löytää kohtisuoraa reunaa vastaan olevien gradienttien voimakkuuksien maksimikohdat tai vastaavasti gradientin suuntaisen toisen asteen derivaatan nollakohdan ylitykset ja täten saada vain yksi piste reunavasteelle. Aluksi jokaiselle kuvapisteelle laskettu gradientin suunta kvantisoidaan kuvan 5.4 mukaisesti neljään eri suuntaan. Tämän jälkeen jokaiselle nollasta poikkeavalle gradientin arvolle tarkastetaan kvantisoitujen suuntien perusteella että onko kyseisessä pisteessä olevan gradientin voimakkuus $|\nabla(G * f)|$ suurempi kuin kahden viereisen. Jos jompikumpi kvantisoitujen suuntien mukaisista viereisistä gradienttien voimakkuuksista on suurempi kuin tarkasteltavan pisteen, tällöin kyseinen piste merkitään poistettavaksi. Kun kaikki kuvan pisteet on käyty läpi, niin muutetaan kaikkien poistettavaksi merkittyjen kuvapisteiden gradientin voimakkuuden arvo nollaksi.

Ei-maksimiarvojen tukahduttamisen jälkeen merkitään varsinaiset reunapisteet kahden kynnysarvon avulla hystereesipohjaisella suodatuksella, jonka tarkoituksena on vähentää katkonaisten reunojen määrää. Jos reunapisteen gradientin voimakkuus ylittää reunaehdokkuuden rajan t_1, niin piste merkitään



Kuva 5.4: Ei-maksimiarvojen tukahduttamisen kvantisointiarvojen alueet.

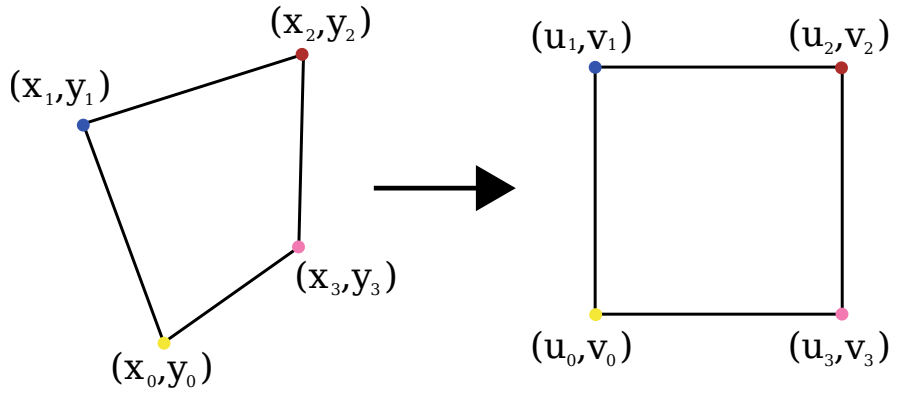
osaksi reunakarttaa. Seuraavaksi käydään kaikki pisteet, jotka ovat reunakartassa olevien pisteiden vieressä ja joiden gradientin voimakkuus on välillä $[t_0, t_1]$ ja merkitään nämä osaksi reunakarttaa. Tätä vaihetta toistetaan niin kauan että uusia reunapisteitä ei enää tule. Canny (1986) suosittelee kynnysarvoille arvosuhdetta, että t_1 on 2–3 kertaa t_0:n arvo.

Lisäksi Cannyn reunantunnistimessa on yhtenä lisäaskeleena mahdollista tehdä reunantunnistus kasvavilla gaussisen maskin variansseilla ja yhdistää nämä tulokset, mutta yleensä tämä jätetään sovelluksissa toteuttamatta(Sonka et al., 2007). Lisäksi kun reunoja kuvasta tunnistetaan, niin Cannyn reunantunnistimessa on kolme avointa parametria, jotka pitää valita joko etukäteen tai päätellä kuvasta. Gaussisen maskin varianssi σ^2 määrää reunavasteen herkkyyden, jolloin tämän valinta riippuu siitä minkä kokoisia koodialueita odotetaan tunnistettavan. Varsinaisten reunapisteiden merkinnässä käytettävät kynnysarvot t_0 ja t_1 taas riippuvat reunavasteiden suuruuksien oletetuista arvoista, jotka taas riippuvat tummien ja vaaleiden moduulien arvojen erotuksesta. Tutkituista lähteistä ainoastaan (Alfthan, 2008) ottaa kantaa Cannyn reunantunnistimen parametrien valintaan ja siinä niiden arvot syntyvät kokeilun tuloksena.

5.5 Käänteisperspektiivimuunnos

Käsivaralta tehtävässä kuvaustilanteessa QR-koodin kuva on mahdollisesti vinossa ja kiertyneenä, jolloin koodin muoto ei ole neliömäinen, mistä johtuen koodin moduulien arvojen lukeminen ei onnistu enää helposti suoraan. Käänteisperspektiivimuunnoksella voidaan kuvan 5.5 mukaisesti muuntaa epämuodostunut tai kierretty koodialue tunnettujen kuvapisteiden mukaisesti toiseen koordinaatistoon, QR-koodin tapauksessa koordinaattiakseleiden suuntaan asetetuksi neliöksi. Tässä esiteltävä lineaarinen koordinaatistomuunnos ei pyri korjaamaan kuvan mahdollisia epälineaarisia vääristymiä, kuten esimerkiksi sylinterin pinnanmuodosta aiheutuvia tai paperin taittumisesta johtuvia vääristymiä. Kuitenkin kohdistuskuvioiden sijaintia hyödyntämällä näiden vääristymien aiheuttamia haittavaikutuksia voi yrittää vähentää.

Eräs tapa tehdä Koordinaatistojen välinen muunnos on käyttää homogeenisiä koordinaatteja, joilla kuvapistevektorit ilmaistaan muodossa $\mathbf{p} = [x_i y_i 1]^{\mathrm{T}}$, eli koordinaattien lisäksi kuvapistevektorissa on arvo 1(Sjöberg, 2010). Tällöin muunnos voidaan esittää lineaarisena matriisiyhtälönä $\mathbf{p}_{uv} = \mathbf{C}\mathbf{p}_{xy}$, missä $\mathbf{p}_{xy}$ on kuvapisteen arvo alkuperäisessä koordinaatistossa, $\mathbf{p}_{uv}$ on kuvapisteen arvo uudessa koordinaatistossa ja $\mathbf{C}$ on $3x3$ muunnosmatriisi. Aukikirjoitettuna yhtälö on seuraavanlainen:



Kuva 5.5: Lineaarinen koordinaatistomuunnos kahden koordinaatiston välillä.

$$\begin{bmatrix} w_i u_i \\ w_i v_i \\ w_i \end{bmatrix} = \begin{bmatrix} c_{00} & c_{01} & c_{02} \\ c_{10} & c_{11} & c_{12} \\ c_{20} & c_{21} & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}.$$

Tästä ratkaisemalla u_i ja v_i saadaan yhtälöt

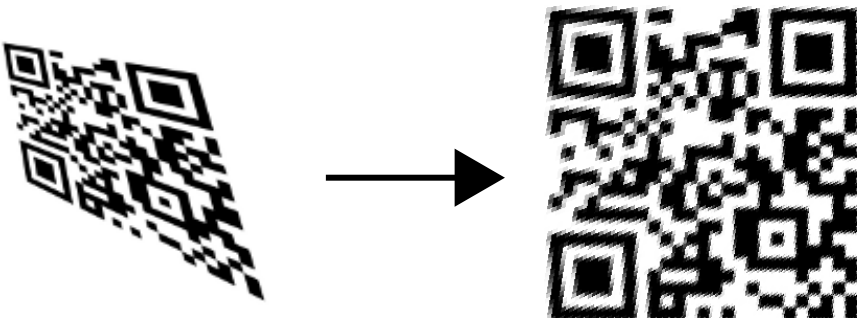
$$\begin{aligned} u_i &= \tfrac{c_{00}x_i + c_{01}y_i + c_{02}}{c_{20}x_i + c_{21}y_i + 1}, \\ v_i &= \tfrac{c_{10}x_i + c_{11}y_i + c_{12}}{c_{20}x_i + c_{21}y_i + 1}. \end{aligned}$$

Muunnosmatriisin $\mathbf{C}$ arvot voidaan ratkaista yhtälöstä $\mathbf{d} = \mathbf{Q}\mathbf{c}$, missä m:lle koordinaattiparille $\mathbf{d} = [u_0 \cdots u_m v_0 \cdots v_m]^{\mathrm{T}}$ sisältää kaikki kohdekoordinaattiparit, $\mathbf{c} = [c_{00} \;\; c_{01} \;\; c_{02} \;\; c_{10} \;\; c_{11} \;\; c_{12} \;\; c_{20} \;\; c_{21}]^{\mathrm{T}}$ sisältää muunnosmatriisin kertoimet ja

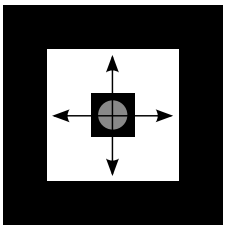
$$\mathbf{Q} = \begin{bmatrix} x_0 & y_0 & 1 & 0 & 0 & 0 & -x_0u_0 & -y_0u_0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_m & y_m & 1 & 0 & 0 & 0 & -x_mu_m & -y_mu_m \\ 0 & 0 & 0 & x_0 & y_0 & 1 & -x_0v_0 & -y_0v_0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & x_m & y_m & 1 & -x_mv_m & -y_mv_m \end{bmatrix}$$

sisältää aukikirjoitetun kuvapistevektorin arvojen oikean puolen muotojen

$$\begin{aligned} u_i &= c_{00}x_i + c_{01}y_i + c_{02} - c_{20}x_iu_i - c_{21}y_iu_i \\ v_i &= c_{10}x_i + c_{11}y_i + c_{12} - c_{20}x_iv_i - c_{21}y_iv_i \end{aligned}$$



Kuva 5.6: Käänteisperspektiivimuunnettu QR-koodi, jossa näkyy alkuperäisen kuvan alhaisesta resoluutiosta johtuvaa sahalaitaisuutta.



Kuva 5.7: Kohdistuskuvion keskikohta voidaan hakea sisäisen valkoisen alueen rajojen avulla.

arvot ilman kertoimia c_{ij}, jotka sisältyvät vektoriin $\mathbf{c}$ (Aidong et al., 2007; Sonka et al., 2007). Hakemalla koodialueen kulmia saadaan 4 tunnettua kuvapistettä, jolloin kerroinvektori $\mathbf{c}$ ratkeaa suoraan matriisi-inversiolla

$$\mathbf{c} = \mathbf{Q}^{-1}\mathbf{d}.$$

Kuvassa 5.6 näkyy käänteisperspektiivimuunnosta lähimmän naapurin interpolaation kanssa tuotettu kuva. Muunnetussa kuvassa näkyy alkuperäisen kuvan alhaisesta resoluutiosta ja interpolaatiomenetelmästä johtuvaa sumentumista ja sahalaitaisuutta, erityisesti pienempikokoisten moduulien kohdalla vasemmassa yläkulmassa.

5.5.1 Kohdistuskuvioiden haku

Kohdistuskuvioiden avulla voidaan parantaa koodialueen lukemisen tulosta suurikokoisilla koodialueilla, sillä kohdistuskuviot mahdollistavat koodialueen jakamisen pienempiin osiin ja täten erilaisten vääristymien paremman huomioonottamisen. ISO-standardissa kohdistuskuvioiden keskikohdan

hakemiselle annetaan ohje, että kohdistuskuvion keskikohdan sijainnille tulisi laskea arvio ja tämän avulla hakea kuvan 5.7 mukaisesti sisällä olevan valkoisen alueen ulkoääriviivat. Näiden valkoisen alueen ulkoääriviivojen perusteella voi laskea kohdistuskuvion keskikohdan.

Lähteissä (Yue ja Mingjun, 2006; Yue et al., 2008) annetaan kohdistuskuvioiden hakemiselle hieman tarkempi lähestymistapa, jossa lähdetään hakemaan kohdistuskuviota kahdeksaan eri suuntaan arvioidusta kohdistuskuvion paikasta ja etsitään mahdollisia 1:1:1:1:1 tumman ja vaalean alueen suhteita täyttäviä paikkoja. Lisäksi Yue et al. (2008) mainitsevat, että mustien ääriviivojen tulisi ympyröidä kohdistuskuvion keskikohdan ehdokaspisteen valkoista aluetta, mikä on käytännössä sama ehto kuin QR-koodin standardissa.

Kohdistuskuvioiden paikkojen hakemisella alue varsinainen viivakoodialue voidaan jakaa useampaan osaan, joille jokaiselle voi tehdä käänteisperspektiivimuunnoksen ja moduulien arvojen lukemisen erikseen. Tämän tarkoitus on parantaa suurien viivakoodialueiden lukemisvarmuutta tilanteissa, joissa viivakoodialue on niin suuri, että joko viivakoodialueen muoto tai kuvan vääristymät vaikeuttavat lukemista. Esimerkkinä tällaisesta vääristävästä muodosta on paperin taittuminen tai tölkin kylki.

5.6 Viivakoodin uudelleenrakennus

Kun viivakoodialue on oikaistu tai muuten siinä muodossa, että viivakoodialueen rajat ja moduulien koot tiedetään, koodialueelle voidaan muodostaa hila, jonka perusteella varsinaiset moduulien arvot luetaan. QR-koodin ISO-standardissa(ISO, 2006) moduulien arvojen lukemiselle annetaan ohje, että moduuleille tulisi arvioida kuvasta keskikohta ja moduulin oletetusta keskikohdasta tulisi lukea kuvapisteen arvo ja päättää onko tämä tumma vai vaalea. Artikkelissa (Chu et al., 2007) esitellään menetelmä, että moduuleille tulisi arvioida niiden sijaintialueet ja laskea kullekin alueelle tummien ja vaaleiden pisteiden lukumäärä ja päättää moduulin arvo sen perusteella, että ylittääkö tummien pisteiden määrä etukäteen määritellyn rajan.

Luku 6

Yhteenveto

Tässä työssä esiteltiin muutaman kirjallisuudessa esiintyvän QR-koodin tunnistamisessa käytettävän algoritmin yleisperiaate ja menetelmiä algoritmien vaiheiden saavuttamiseksi. Esitellyt algoritmit ja menetelmät eivät ole ainoita mahdollisia ja erityisesti kaupallisessa käytössä olevista QR-koodin tunnistavien ohjelmien algoritmeista harvemmin on tietoa muilla kuin ohjelman kehittäneellä yrityksellä. Lisäksi QR-koodi ei ole ainoa yleisessä käytössä oleva kaksiulotteinen viivakoodityyppi ja QR-koodin ja muiden vastaavien viivakoodityyppien tunnistamisessa voi hyödyntää suoraan samoja menetelmiä aina koodin tulkitsemiseen asti.

Kameralla otettuja kuvia voi käsitellä ja analysoida mielivaltaisen paljon, mutta jokainen käsittely vaatii aikaa, eivätkä kaikki kuville tehtävät toiminnot välttämättä johda korkeampaan tunnistustulokseen. Se mitä kaikkea kuvalle kannattaa tehdä, riippuu tunnistusohjelman käytössä olevasta laitteistosta ja kuvantunnistussovelluksen vaatimuksista sekä siitä, mitä sovelluksen käyttäjältä voidaan odottaa.

Tässä työssä tehtiin vain kirjallisuuskatsaus eri menetelmiin, eikä niiden suoritusnopeutta tai vaikutusta tunnistustarkkuuteen pyritty mittaamaan. Suurimmassa osassa tarkastelluista QR-koodin tunnistamiseen liittyvistä tutkimuksista liittyi erityisesti huonolaatuisilla kameroilla ja hitailla suorittimilla varustettuihin laitteisiin. Lisäksi kaksiulotteisia viivakoodeja tunnistavat ohjelmat pystyvät monesti lukemaan useita erilaisia viivakoodityyppejä, joten erityyppisten viivakoodien erottaminen toisistaan ja tarkka tunnistaminen huonolaatuisesta kuvasta mahdollisimman vähällä suoritinajan käytöllä on edelleen avoin tutkimuksen ja tuotekehityksen kohde.

Kirjallisuutta

AIM ECI1. Extended Channel Interpretations (ECI) - Part One. Association for Automatic Identification and Mobility. URL `https://www.aimglobal.org/Estore/ProductDetails.aspx?productID=88`.

ISO/IEC 18004:2006. Information technology – Automatic identification and data capture techniques – QR Code 2005 bar code symbology specification. International Organization for Standardization/International Electrotechnical Commission, Geneva, Switzerland, elokuu 2006. 114 s.

Sun Aidong, Sun Yan ja Liu Caixing. The qr-code reorganization in illegible snapshots taken by mobile phones. sivut 532 – 538, elokuu 2007. doi: 10.1109/ICCSA.2007.86.

Jonas Alfthan. Robust detection of two-dimensional barcodes in blurry images. Pro gradu, Royal Institute of Technology, School of Computer Science and Engineering, Stockholm, Sweden, 2008. 73 s.

Al Bovik, toimittaja. *Handbook of Image and Video Processing. Second edition.* Academic Press, toinen painos, 2005. ISBN 0-12-119792-1. 1384 s.

Martin Cadik. Perceptual evaluation of color-to-grayscale image conversions. *Computer Graphics Forum*, 27(7):1745–1754, 2008. ISSN 1467-8659. doi: 10.1111/j.1467-8659.2008.01319.x.

John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986. ISSN 0162-8828. doi: 10.1109/TPAMI.1986.4767851.

Yu-Hsuan Chang, Chung-Hua Chu ja Ming-Syan Chen. A general scheme for extracting qr code from a non-uniform background in camera phones and applications. sivut 123–130, joulukuu 2007. doi: 10.1109/ISM.2007.4412365.

Chung-Hua Chu, De-Nian Yang ja Ming-Syan Chen. Image stabilization for 2d barcode in handheld devices. *MULTIMEDIA '07: Proceedings of the 15th*

international conference on Multimedia, sivut 697–706, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-702-5. doi: 10.1145/1291233.1291394.

Denso Wave. About qrcode.com, 2010. [Online]. 9.3.2010. [Viitattu 8.4.2010]. Saatavissa: `http://www.denso-wave.com/qrcode/index-e.html`.

Masahiro Hara, Motoaki Watabe, Tadao Nojiri, Takayuki Nagaya ja Yuji Uchiyama. Us5726435. optically readable two-dimensional code and method and apparatus using the same, maaliskuu 1998. 30 s.

Shih-Chang Hsia, Ming-Huei Chen ja Yu-Min Chen. A cost-effective line-based light-balancing technique using adaptive processing. *Image Processing, IEEE Transactions on*, 15(9):2719–2729, syyskuu 2006. ISSN 1057-7149. doi: 10.1109/TIP.2006.877354.

R. Lokhande, K. V. Arya ja P. Gupta. Identification of parameters and restoration of motion blurred images. *SAC '06: Proceedings of the 2006 ACM symposium on Applied computing*, sivut 301–305, New York, NY, USA, 2006. ACM. ISBN 1-59593-108-2. doi: 10.1145/1141277.1141347.

Sun Ming, Fang Zhenkun, Fu Longsheng ja Zhang Shuhuai. Research of automatic recognition algorithm of chinese-sensible code. *Intelligent Systems, 2009. GCIS '09. WRI Global Congress on*, osa 4, sivut 492–499, toukokuu 2009. doi: 10.1109/GCIS.2009.259.

Bryan S. Morse. Thresholding, tammikuu 2000. [Online]. 15.1.2004. [Viitattu 10.4.2010]. Saatavissa: `http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MORSE/threshold.pdf`.

Nokia. Nokia n86 8mp - tarkka tuoteseloste, 2010. [Online]. 15.4.2010. [Viitattu 22.4.2010]. Saatavissa: http://www.nokia.fi/tuotteet/kaikki-puhelimet/nokia-n86-8mp/tarkka-tuoteseloste.

NTT DOCOMO. Information on i-mode technical topics – bar code, 2010. [Online]. [Viitattu 8.4.2010]. Saatavissa: `http://www.nttdocomo.co.jp/english/service/imode/make/content/barcode/`.

NTT DOCOMO. Ntt docomo secures 79.59 percent of net mobile shares, joulukuu 2009. [Online]. 3.12.2009. [Viitattu 8.4.2010]. Saatavissa: `http://www.nttdocomo.com/pr/2009/001463.html`.

Eisaku Ohbuchi, Hiroshi Hanaizumi ja Lim Ah Hock. Barcode readers using the camera device in mobile phones. sivut 260 – 265, marraskuu 2004. doi: 10.1109/CW.2004.23.

Noboyuki Otsu. A threshold selection method from gray-level histograms. *Systems, Man and Cybernetics, IEEE Transactions on*, 9(1):62–66, tammikuu 1979. ISSN 0018-9472. doi: 10.1109/TSMC.1979.4310076.

Irving S. Reed ja Xuemin Chen. *Error-Control Coding for Data Networks*. Kluwer Academic Publishers., toinen painos, 1999. ISBN 0-7923-8528-4. 572 s.

Mats Sjöberg. T-61.5070 computer vision, exercise 2/2010. camera calibration, stereo vision, and surface modeling., 2010. [Online]. 19.2.2010. [Viitattu 8.4.2010]. Saatavissa: `http://www.cis.hut.fi/mats/cv/ex2/answers2.pdf`.

Milan Sonka, Vaclav Hlavac ja Roger Boyle. *Image Processing, Analysis and Machine Vision*. Thomson-Engineering, 2007. ISBN 978-0-495-24438-7.

Daniel Wagner ja Dieter Schmalstieg. Artoolkitplus for pose tracking on mobile devices. *Proceedings of 12th Computer Vision Winter Workshop (CVWW'07)*, sivut 139–146, helmikuu 2007.

Friedrich M. Wahl, Kwan Y. Wong ja Richard G. Casey. Block segmentation and text extraction in mixed text/image documents. *Computer Graphics and Image Processing*, 20(4):375–390, 1982. ISSN 0146-664X. doi: 10.1016/0146-664X(82)90059-4.

Liu Yue ja Liu Mingjun. Automatic recognition algorithm of quick response code based on embedded system. osa 2, sivut 783 – 788, lokakuu 2006. doi: 10.1109/ISDA.2006.253712.

Liu Yue, Yang Bo ja Yang Ju. Bar code recognition in complex scenes by camera phones. *International Conference on Natural Computation*, 5: 462–466, 2008. doi: 10.1109/ICNC.2008.288.

Zxing-projektin wiki. Barcode contents, maaliskuu 2010. [Online]. 30.3.2010. [Viitattu 30.3.2010]. Saatavissa: `http://code.google.com/p/zxing/wiki/BarcodeContents`.