Institutionen för systemteknik
Department of Electrical Engineering

Examensarbete

Modellering av delvis kända objekt med hjälp av bilder

Examensarbete utfört i Informationsteori
vid Tekniska högskolan i Linköping
av

Linus Fredriksson, Martin Gyllensten

LITH-ISY-EX--06/3689--SE

Linköping 2006

Linköpings universitet
TEKNISKA HÖGSKOLAN

Department of Electrical Engineering
Linköpings universitet
SE-581 83 Linköping, Sweden

Linköpings tekniska högskola
Linköpings universitet
581 83 Linköping

Modellering av delvis kända objekt med hjälp av bilder

Examensarbete utfört i Informationsteori
vid Tekniska högskolan i Linköping
av

Linus Fredriksson, Martin Gyllensten

LITH-ISY-EX--06/3689--SE

Handledare: **Viiveke Fåk**
ISY, Linköpings universitet

Examinator: **Viiveke Fåk**
ISY, Linköpings universitet

Linköping, 13 juni, 2006

Avdelning, Institution
Division, Department

Informationsteori
Department of Electrical Engineering
Linköpings universitet
S-581 83 Linköping, Sweden

Datum
Date

2006-06-13

Språk
Language

☒ Svenska/Swedish
☐ Engelska/English
☐ _____

Rapporttyp
Report category

☐ Licentiatavhandling
☒ Examensarbete
☐ C-uppsats
☐ D-uppsats
☐ Övrig rapport
☐ _____

ISBN
—

ISRN
LITH-ISY-EX--06/3689--SE

Serietitel och serienummer **ISSN**
Title of series, numbering —

URL för elektronisk version
http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-6724

Titel
Title

Modellering av delvis kända objekt med hjälp av bilder

Creating 3D-models of partly known objects from unidirectional pictures

Författare
Author

Linus Fredriksson, Martin Gyllensten

Sammanfattning
Abstract

Today many different types of lock systems are used all over the world. Passwords, finger prints and magnetic strips are a few examples. When the key is a three dimensional object, lika a physical metal key, the security depends, in a high degree, on that you can't copy the object without access to it. One approach to copying the object is photographing the object, and from these photographs create a copy. If you can only get photographs from one side of the object more information will be needed.

Methods for digital image processing have been examined to be able to answer the question; can an attack against an ordinary door lock be performed by taking photographs of the key that opens the lock, and from the pictures re-create the key? Focus lies on extracting as much information about the object as possible from the pictures. This is achieved by generating a high resolution image with more details than each of the original pictures in a low resolution image series.

The experiment has been performed with camera equipment that is inexpensive and easy to get hold of. This is done to show that the attack can be performed even with only basic camera equipment.

Nyckelord
Keywords

lås, nyckel, säkerhet, bildregistrering, upplösningsförbättring, bildbehandling

Abstract

Today many different types of lock systems are used all over the world. Passwords, finger prints and magnetic strips are a few examples. When the key is a three dimensional object, lika a physical metal key, the security depends, in a high degree, on that you can't copy the object without access to it. One approach to copying the object is photographing the object, and from these photographs create a copy. If you can only get photographs from one side of the object more information will be needed.

Methods for digital image processing have been examined to be able to answer the question; can an attack against an ordinary door lock be performed by taking photographs of the key that opens the lock, and from the pictures re-create the key? Focus lies on extracting as much information about the object as possible from the pictures. This is achieved by generating a high resolution image with more details than each of the original pictures in a low resolution image series.

The experiment has been performed with camera equipment that is inexpensive and easy to get hold of. This is done to show that the attack can be performed even with only basic camera equipment.

Sammanfattning

I låssystem av olika slag används en mängd olika typer av nycklar. Lösenord, fingeravtryck och magnetremsor är några exempel. Då nyckeln utgörs av ett tredimensionellt objekt, så som en metallnyckel, beror mycket av systemets säkerhet på att detta objekt inte kan kopieras utan tillgång till det. Ett tillvägagångssätt för att lyckas kopiera objektet är att ta fotografier av det och sedan utifrån dessa skapa en kopia. Här krävs ytterligare information om objektet om man exempelvis bara kan få tag i fotografier tagna från ena sidan.

Bildbehandlingsmetoder för digitala bilder har undersökts för att svara på frågan om en attack mot ett vanligt dörrlås kan utföras genom att på avstånd fotografera nyckeln som öppnar låset och sedan kopiera nyckeln utifrån bilderna. Fokus ligger på att extrahera så mycket information som möjligt ur bilderna genom att från en serie bilder generera en bild med mer detaljer än var och en av orginalbilderna.

Försök har utförts med billig och lättåtkomlig kamerautrustning för att visa en låg tekniknivå för vilken attacken kan utföras.

Tack

Först skulle vi vilja tacka alla på ISY, speciellt vår examinator Viiveke Fåk, som även agerade handledare under arbetets slutfas, och Jacob Löfvenberg för handledning under den första delen av arbetet.

Vi skulle även vilja tacka följande personer som hjälpt till med diverse saker; Johan Gyllensten, Harald Nautsch, Dan Söderström och Tore Jungnelius.

Innehåll

1 Inledning **1**
1.1 Bakgrund . . . 1
1.2 Mål . . . 3
1.3 Syfte . . . 3
1.4 Metod . . . 3
1.5 Begränsningar i metoden . . . 4
1.6 Målgrupp . . . 4
1.7 Läsanvisning . . . 4

2 Lås och nycklar **7**
2.1 Stiftcylinderlås . . . 7
2.2 Regler för lås och nycklar . . . 10
2.3 Mekaniska lås som kryptosystem . . . 10
2.4 Huvudnycklar . . . 11

3 Bildkvalitetförbättring **13**

4 Bildregistrering **15**
4.1 Transformationer . . . 16
4.1.1 Euklidisk transformation . . . 16
4.1.2 Affin transformation . . . 17
4.1.3 Perspektivtransformation . . . 17
4.1.4 Polynomisk transformation . . . 18
4.2 Bildvariationer . . . 18
4.3 Korskorrelation . . . 19
4.4 Fouriermetoder . . . 20
4.5 Point-mapping . . . 22
4.6 Vald metod . . . 22

5 Upplösningsförbättring **29**
5.1 Iterative Backprojection (IBP) . . . 29
5.2 Projection Onto Convex Sets (POCS) . . . 31
5.3 Maximum A-Posteriori (MAP) . . . 31
5.4 Frekvensdomänmetoder . . . 32
5.5 Vald metod . . . 32

6 Bildfiltrering **35**
6.1 Point Spread Function 37
6.2 Medelvärdesfilter 37
6.3 Medianvärdesfilter 37
6.4 Gaussfilter 38
6.5 Adaptivt gaussfilter 38
6.6 Wienerfilter 39
6.7 Vald metod 41

7 Kantdetektering **43**
7.1 Sobels metod 45
7.2 Laplacefiltrering 46
7.3 Cannys metod 46
7.4 Vald metod 48

8 Experimentbeskrivning **49**
8.1 Fotografering 49
8.2 Bildbehandling 52
8.3 Registrering och upplösningshöjning 53
8.4 Filtrering av högupplöst bild 54
8.5 Kantdetektering och konturoptimering 54
8.6 Mätning 55
8.7 Nyckeltillverkning 56

9 Resultat **59**
9.1 Digitalkamera 59
9.2 Mobiltelefonkamera 61

10 Diskussion och analys **63**
10.1 Experimentmetod 63
10.1.1 Fotografering 63
10.1.2 Bildförbättring 65
10.1.3 Filtrering av högupplöst bild 67
10.1.4 Extraktion av information 68
10.1.5 Nyckelåterskapning 71
10.2 Resultat 71
10.3 Säkerhet hos låssystem 72

11 Slutsats **73**
11.1 Slutsats 73
11.2 Rekommendationer för fortsatt arbete 74
11.2.1 Fotografering 74
11.2.2 Bildförbättring 75
11.2.3 Lås och nycklar 75

Litteraturförteckning **77**

A Utrustning **81**

B Experimentdetaljer **84**
B.1 Fotografering 84
B.2 Bildbehandling 85

C Programkod **87**
C.1 runscript.m 87
C.2 register.m 88
C.3 enhance.m 90
C.4 contour.m 92

D Detaljerade resultat **96**
D.1 Bilder 96
D.2 Karaktäristiska kurvor 102
D.3 Registrering 103
D.4 Bildjämförelse 105

Figurer

2.1 Låsets olika delar. 8
2.2 Rätt nyckel i ett lås. 8
2.3 Felaktig nyckel i ett lås. 8
2.4 Beskrivning av understift. 9
2.5 En nyckel sedd från sidan och framifrån. 9

3.1 Bild med låg upplösning. 13
3.2 Beskrivning av förloppet vid upplösningshöjning. 14

4.1 Ett objekt ska representeras i låg upplösning 16
4.2 Olika transformationer. 17
4.3 Två bilder av objekt som förskjutits i förhållande till varandra. . . 23
4.4 Karaktäristisk kurva f_x för bilden A i figur 4.3. 24
4.5 Karaktäristisk kurva f_y för bilden A i figur 4.3. 24
4.6 Karaktäristisk kurva f_x för bilden B i figur 4.3. 24
4.7 Karaktäristisk kurva f_y för bilden B i figur 4.3. 24
4.8 Två bilder med identiska karaktäristiska kurvor. 25
4.9 Karaktäristiska kurvor f_x för bild A repektive B i figur 4.3. 25

5.1 Högupplöst pixelvärde ur två lågupplösta bilder 33

6.1 Illustration av hur ett pixelvärde beräknas i en filtrering. 36
6.2 Enkelt medelvärdesfilter 37
6.3 Approximation av gaussfördelning. 39
6.4 Filtreringsmask för gaussfiltrering. 39

7.1 Signal som representerar en kant i en bild. 44
7.2 Förstaderivatan av signalen i figur 7.1 44
7.3 Andraderivatan av signalen i figur 7.1 44
7.4 Filtreringsmasker för Sobels metod. 45
7.5 Filtreringsmask för laplacefiltrering. 46
7.6 Avrundning av riktningsvektorer i Cannys metod. 47
7.7 Exempelbild i vilken kanter ska detekteras. 48
7.8 Detekterade kanter i figur 7.7. 48

8.1 En överblick av processen att kopiera en nyckel. 50
8.2 Kameruppställning, scenario A. 51
8.3 Kameruppställning, scenario B. 51
8.4 Fotografi enligt scenario A. 52
8.5 Fotografi enligt scenario B. 53
8.6 Exempel på hur en bilds ointressanta delar beskärs. 53
8.7 Exempel på hur en kontur ritas i en bild. 55

10.1 Konturanpassning för slitna nycklar. 69

A.1 Panasonic DMC-LC70. 81
A.2 Samsung SGH-E700. 82
A.3 Digitalt skjutmått. 83

D.1 Fotografi taget enligt scenario A(1m). 97
D.2 Fotografi taget enligt scenario A(2m). 97
D.3 Fotografi taget enligt scenario A(3m). 98
D.4 Fotografi taget enligt scenario A(4m). 98
D.5 Fotografi taget enligt scenario A(6.5m). 99
D.6 Fotografi taget enligt scenario B(10cm). 99
D.7 Fotografi taget enligt scenario B(20cm). 100
D.8 Fotografi taget enligt scenario B(30cm). 100
D.9 Fotografi taget enligt scenario B(50cm). 101
D.10 Fotografi taget enligt scenario B(100cm). 101
D.11 Karaktäristiska kurvor f_x för två fotografier. 102
D.12 Karaktäristiska kurvor f_y för två fotografier. 102
D.13 Orginalbild(3m) 105
D.14 Upplösningsförbättrad bild 105
D.15 Upplösningsförbättrad och filtrerad bild. 105
D.16 Bild med kontur inritad. 106
D.17 Kontur anpassad efter givna regler. 106

Tabeller

8.1 Försök med digitalkamera. 50
8.2 Försök med mobiltelefonkamera. 51

9.1 Resultat av experiment med digitalkamera. 60
9.2 Antal tillverkade nycklar efter experiment med digitalkamera. . . . 60
9.3 Resultat av experiment med mobiltelefonkamera. 61
9.4 Antal tillverkade nycklar efter experiment med mobilkamera. . . . 61

D.1 Registreringsresultat för digitalkamera(1m). 103
D.2 Registreringsresultat för digitalkamera(3m). 103
D.3 Registreringsresultat för mobiltelefonkamera(10cm). 104
D.4 Registreringsresultat för mobiltelefonkamera(50cm). 104

Kapitel 1

Inledning

Detta kapitel ger en introduktion av rapportens frågeställning. Mål, syfte, metod och målgrupp presenteras. Begränsningar i metoden diskuteras, och en läsanvisning ger läsaren en kort beskrivning av varje kapitel.

1.1 Bakgrund

I vardagslivet stöter man ofta på system där en viss säkerhet krävs och förutsätts. När man exempelvis tar ut pengar från en bankomat förväntar man sig att man är den enda som kan komma åt pengarna på det egna kontot samt att man själv inte kan komma åt pengar på någon annans konto. En tidningsartikel[15] i den svenska kvällspressen berättar dock om hur man genom att montera en extra kortläsare över bankomatens kortläsare och en falsk knappsats, har lyckats tömma folks konton. Problemet tycks vara att systemet endast är säkert om den enda som känner till koden och den enda som har tillgång till bankomatkortet är kontoinnehavaren. Lyckas någon annan komma åt kortet och koden är pengarna helt oskyddade. Att koden bör skyddas så man är den enda som känner till den inser nog de flesta, men bankomatkortet är ju ett fysiskt föremål och så länge man har det i sin plånbok kan väl ingen annan ha det också, eller? Hela systemet bygger på att kortet inte kan kopieras utan kortägarens tillåtelse men det tillvägagångssätt som beskrivs i kvällspressen visar att detta faktiskt är ganska enkelt.

Det finns många andra system vars säkerhet beror på hur svårt det är att kopiera något som endast de som ska ha tillgång till systemet har tillgång till. I Sandströms rapport[30] beskrivs hur man enkelt kan kopiera någons fingeravtryck och på så vis utge sig för att vara någon annan i ett system där just fingeravtryck fungerar som nyckel. Att anta att en fysisk nyckel inte kan kopieras verkar vara vanskligt, men vanligt. När man låser sin ytterdörr förväntar man sig att ingen som inte har nyckeln ska kunna öppna den och att ingen man inte gett en nyckel till ska ha en sådan.

Att man kan ge en nyckel till en låssmed och få en kopia vet alla, så uppenbarligen kan nyckeln kopieras. Det är ju heller inte så konstigt eftersom den endast består av en liten metallbricka formad på ett visst sätt.

I nättidningen philly.com[31] berättas om en man som stal innehållet i folks postboxar genom att kopiera brevbärarens huvudnyckel. En filmkamera placerades så den zoomade in huvudnyckeln när låset öppnades och gärningsmannen kunde efteråt tillverka en fungerande nyckel med filmen som ritning. Tillvägagångssättet påminner mycket om det som användes för att kopiera bankomatkort men en stor skillnad kan identifieras. Bankomatkorten kopierades exakt, vilket var möjligt då deras innehåll är en sifferkod som lagras på en magnetremsa, medan metallnycklarna kopierades så bra det var möjligt med den teknik som användes.

Även om nyckelkopieringen inte var så sofistikerad så är tekniken intressant. En bild tas på avstånd av ett föremål och sedan kopieras föremålet utifrån bilden. När föremålet är en nyckel av något slag blir tekniken extra intressant, eftersom denna inte ska gå att kopiera utan att ägaren tillåter detta.

Vill man utveckla tekniken finns i huvudsak två frågor att begrunda. Hur får man en så bra bild som möjligt av föremålet och vilken information är det man behöver i bilden? Frågorna är intressanta på olika sätt. Det första man tänker på när det gäller bildkvaliteten är kanske att man bör ta bilden med en bra kamera på ett sådant avstånd så att många detaljer kan utläsas ur bilden, men vad som är mer intressant är ju om det finns enkla tekniker som möjliggör för en lekman att på ett billigt sätt få en tillräckligt bra bild av föremålet.

När det gäller vilken information som behövs i bilden så gäller det här att ha en viss kunskap om föremålet som ska kopieras. I fallet med fingeravtryck, där föremålet kan sägas vara ett finger, gäller exempelvis att alla fingrar består av samma material så information om detta kan anses vara given. Det är dock viktigt att även denna information tas hänsyn till vid kopieringen då de flesta fingeravtrycksbaserade lås inte skulle kunna låsas upp av exempelvis ett träfinger, även om det hade rätt fingeravtryck.

När det gäller metallnycklar är låstypen avgörande. De flesta lås tar ingen hänsyn till nyckelns material utan endast till dess form. Vet man vilken typ av lås som nyckeln ska öppna är det plötsligt mycket information som är given. Tänker man sig exempelvis den vanligaste typen av dörrlås ser de flesta nycklar ungefär likadana ut, så där kan mycket information ses som given.

I den här rapporten beskrivs tekniker som underlättar kopiering av tredimensionella objekt givet tvådimensionella bilder av objekten samt viss ytterligare information som kan anses vara given beroende på vilken typ av objekt det rör sig om. Fokus ligger på kopiering av metallnycklar, då dessa utgör ett perfekt exempel på tredimensionella objekt som man kan säga ganska mycket om utan att studera ett specifikt objekt. Det är även en typ av objekt som är starkt kopplade till säkerhet, vilket gör dem extra intressanta.

1.2 Mål

Den här rapporten innehåller två olika delar; en teoridel och en exprimentdel. Den teoretiska delen diskuterar olika metoder för att förbättra möjligheten att extrahera sökt information ur bilder, samt ger en överblick över hur lås och nycklar är konstruerade.

Experimentdelen undersöker om det är möjligt att på ett enkelt sätt och med enkel utrustning kopiera en nyckel utifrån en serie bilder av nyckeln. Mer information om metoden finns i avsnitt 1.4. Mer specifikt kan sägas att målet under experimenten är att ta reda på från vilket avstånd nyckeln kan fotograferas, och tillräckligt högkvalitativa bilder genereras, för att nyckelkopieringen ska vara möjlig. I slutändan innebär detta att bestämma vilken bildkvalitet som krävs.

1.3 Syfte

Syftet med den här rapporten är att undersöka metoder som ökar mängden detaljer man kan avläsa om tredimensionella objekt i tvådimensionella bilder av objekten. Som exempel på tillämpning studeras möjligheten att kopiera en metallnyckel genom att fotografera den och hämta information ur bilderna. Då det enligt philly.com[31] redan utförts bedrägerier där nyckelkopiering genomförts genom nyckelfotografering är det intressant att se hur svårt detta är, och på vilka avstånd fotograferingen kan ske.

1.4 Metod

Den teoretiska studien har utförts genom en litteraturstudie. Artiklar och böcker har lästs, diskuterats och analyserats. Olika användningsområden för vedertagna metoder har undersökts och jämförts.

Experimentet gick ut på att fotografera en serie bilder av en nyckel och utifrån bilderna återskapa denna nyckel tillräckligt bra för att den ska kunna öppna motsvarande lås. Utgångspunkten för experimentet var att undersöka på hur långt avstånd det var möjligt att placera en kamera men ändå få tillräckligt detaljerade bilder. Försöket utfördes med både en mobiltelefonkamera och en vanlig digitalkamera. Hur detta genomfördes, samt detaljer om experimentets olika delar återfinns i kapitel 8.

1.5 Begränsningar i metoden

Den här rapporten ger en överblick över olika metoder som kan användas för att extrahera intressant information ur tvådimensionella bilder, men inriktar sig mer detaljerat mot en viss metod då flera finns att välja mellan. Andra metoder skulle kunna användas för att uppnå liknande mål, men de beskrivs bara kortfattat i denna rapport, och testas inte i praktiken. Det praktiska experimentet begränsas av att det endast undersöker en viss nyckeltyp samt att endast relativt billig experimentutrustning används.

1.6 Målgrupp

Den här rapporten har ett antal olika målgrupper:

- Personer som är intresserade av bildbehandling och i synnerhet upplösningshöjning av bilder. Rapporten pekar på intressanta användningsområden av tidigare kända tekniker.
- Personer intresserade av datasäkerhet. Rapporten belyser hur vanliga lås kan liknas vid, och attackeras som kryptosystem.
- Tillverkare av låssystem.
- Låssmeder och andra personer i låsbranchen.

Eftersom rapporten har flera olika målgrupper kan olika delar av rapporten vara intressanta för olika grupper. Läsaren förutsätts inte ha omfattande kunskap om lås eller bildbehandling. En viss matematisk kunskap krävs dock för att i detalj kunna tillgodogöra sig vissa metoder. Avsnitt 1.7 rekommenderas för läsare som snabbt vill finna den information som är relevant för just honom/henne.

1.7 Läsanvisning

Det här avsnittet ger en kort beskrivning av alla kapitel och bilagor i rapporten.

- Kapitel 1 innehåller en kort beskrivning av rapportens frågeställning. Mål, syfte, metod och målgrupper presenteras. Vissa begränsningar i metoden diskuteras också.
- Kapitel 2 ger läsaren en beskrivning av hur lås och nycklar är konstruerade.
- Kapitel 3 ger läsaren en kort introduktion till bildkvalitetsförbättring, samt är en bra utgångspunkt för efterföljande kapitel som behandlar just detta.

- Kapitel 4 diskuterar olika metoder för bildregistrering och begreppet förklaras. Här ges också en mer detaljerad beskrivning av den metod som valts.
- Kapitel 5 diskuterar olika metoder för upplösningsförbättring, samt ger en mer detaljerad beskrivning av den metod som valts.
- Kapitel 6 diskuterar olika metoder för bildfiltrering, samt ger en mer detaljerad beskrivning av den metod som valts.
- Kapitel 7 diskuterar olika metoder för kantdetektering, samt ger en mer detaljerad beskrivning av den metod som valts.
- Kapitel 8 beskriver metoder som användes vid experimentet, kopiering av en nyckel utifrån en serie bilder, och de verktyg som användes under detta experiment.
- Kapitel 9 presenterar resultatet av experimentet beskrivet i föregående kapitel.
- Kapitel 10 analyserar och diskuterar de metoder som användes i experimentet, samt de resultat som experimentet gav.
- Kapitel 11 presenterar en slutsats och ger rekommendationer för framtida arbete.
- Bilaga A ger en detaljerad beskrivning av den utrustning som användes i experimentet.
- Bilaga B beskriver i detalj hur delar av experimentet genomfördes.
- Bilaga C presenterar den implementation som gjorts i MATLAB.
- Bilaga D presenterar detaljerade resultat från experimentet i form av bilder, diagram och tabeller.

Kapitel 2

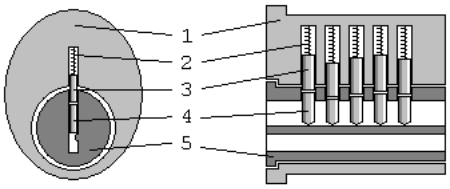
Lås och nycklar

Överallt där man behöver någon form av säkerhet används idag olika typer av lås. Allt ifrån hänglås, lås på bilar och hus, till lås på stora bankvalv. Eftersom lås används i så stor utsträckning är det också naturligt att det finns många olika varianter, både vad det gäller utformning och säkerhetsfaktorer.

En låstyp som är mycket vanlig idag, och som vi fokuserar på i den här rapporten, är stiftcylinderlås. Den variant av stiftcylinderlås som rapporten avser är relativt enkel, men då den är vanligt förekommande gör denna begränsning inte rapporten mindre intressant. Exempel på låstyper som inte behandlas i denna rapport är kombinationslås, tillhållarlås och skivcylinderlås. Stiftcylinderlås kan varieras mycket för att öka graden av säkerhet, och detta görs också där säkrare system krävs.

2.1 Stiftcylinderlås

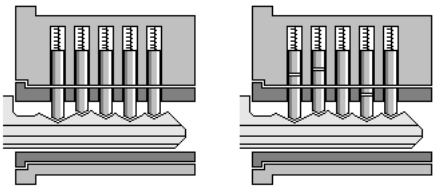
Stiftcylinderlås är den typ av lås som sitter i de flesta dörrar där man kräver lite högre grad av säkerhet, till exempel ytterdörren på ett bostadshus. Låset i det vanligaste utförandet består av följande delar. Ett cylinderhus är den yttre delen av låset och rör sig inte under upplåsning. Nyckelkanalen är där man för in nyckeln i låset. En cylindertrumma sitter inne i cylinderhuset och är den del som roterar när man vrider om nyckeln. I cylinderhuset finns också ett antal spärrstiftskanaler. I varje sådan kanal finns en stiftfjäder som trycker stiftet nedåt. Varje spärrstift består av ett överstift och ett understift. I vissa fall då en huvudnyckel ska kunna användas i låset finns det även ett mellanstift. De olika delarna i ett lås kan ses i figur 2.1.



1. Cylinderhus
2. Stiftfjäder
3. Överstift
4. Understift
5. Cylindertrumma

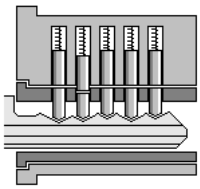
Figur 2.1. Låsets olika delar.

När rätt nyckel förts in i låset lyfts spärrstiften så att delningen av stiften ligger i linje med cylinderhusets och cylindertrummans delning. För de lås där det finns huvudnycklar finns det flera delningar och alltså flera olika kombinationer som gör att en delning ligger i linje. När alla stift befinner sig i en sådan position kan cylindertrumman roteras vilket öppnar låset. Exempel på hur det kan se ut när en korrekt nyckel har förts in i ett lås kan ses i figur 2.2.



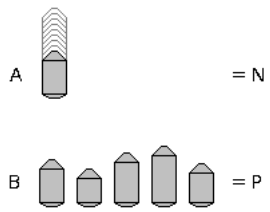
Figur 2.2. Rätt nyckel i ett lås vilket gör att spärrstiften ligger i linje. Till vänster ett lås som endast går att öppna med en nyckel. Till höger ett lås ingående i ett huvudnyckelsystem.

Om en felaktig nyckel används kommer någon eller några av spärrstiften att ligga över eller under cylindertrummans och cylinderhusets delning och genom detta hindra cylindertrumman från att rotera. Detta kan ses i figur 2.3.



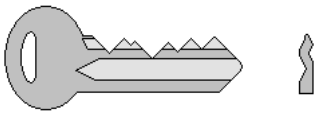
Figur 2.3. Felaktig nyckel i ett lås vilket gör att spärrstift hindrar cylindertrumman från att rotera.

Antalet nyckelkombinationer som finns för stiftcylinderlås bestäms av två faktorer. Dels antalet spärrstift i låset och dels hur många olika lägen i höjdled spärrstiften kan anta. De olika lägen som kan antas beror på hur många olika längder på understift som finns tillgängliga. Teoretiskt skulle ett oändligt antal längder kunna användas, men av praktiska skäl tillverkas understiften bara av ett visst antal olika längder. Det antal olika lägen i höjdled som kan antas betecknas N, antalet spärrstift P. Det betyder att det finns N^P nyckelkombinationer. Exakt vilka värden N och P har beror på vilken typ av lås det är och på vilket företag som tillverkat låset. Av dessa försvinner ett antal kombinationer på grund av att för att låset ska vara godkänt enligt de säkerhetsklasser, som används av till exempel försäkringsbolag, ställs vissa krav på hur nyckeln ska vara skuren. Vad som krävs för att lås ska vara godkända beskrivs av SIS[32]. Intressanta delar av dessa regler finns i avsnitt 2.2.



Figur 2.4. Beskrivning av understift.
A. Antalet olika längder av understift, N. B. Antalet stift, P.

För att nyckeln ska kunna föras in i nyckelkanalen måste nyckeln ha rätt form för att passa i låset. Även här har man av praktiska skäl inte obegränsade valmöjligheter. Varje låstillverkare tillverkar sina lås med ett antal olika utseenden, och till varje låstyp tillverkas blanka nycklar som passar i dessa. De blanka nycklarna kallas nyckelämnen. För att sedan låset ska kunna öppnas slipas nyckelämnet så att nedskärningarna stämmer överens med låsets stift. Att det bara finns ett visst antal nyckelämnen gör det också lätt för en låssmed att kunna kopiera en nyckel då det bara är nyckelns nedskärningar som behöver återskapas.



Figur 2.5. En nyckel sedd från sidan och framifrån.

2.2 Regler för lås och nycklar

För att ett lås ska klassas som säkert ur försäkringsbolagens synvinkel finns det vissa krav som måste uppfyllas. Dessa krav har arbetats fram av SIS och beskrivs utförligt i deras dokument för svensk standard[32, 33, 34]. De krav som kan vara intressant att känna till när man ska återskapa nycklar är de krav som ger information om hur nycklar och lås måste vara utformade.

- För att ett lås ska vara godkänt ska det ha ett nyckelvariationstal på minst 10 000. Det vill säga, det måste finnas minst 10 000 praktiskt användbara kombinationer av nycklar för just den låstypen.
- Ett lås måste ha minst sex stycken spärrelement (spärrstift för stiftcylinderlås).
- Spärrelementen får inte bilda en kontur som gör det möjligt att identifiera nyckelns skärning. Detta avser alltså när nyckeln inte befinner sig i låset.
- På en nyckel ska nedskärningsdjupet förändras minst tre gånger och nycklar till cylinderlås måste också ha minst tre olika nedskärningsdjup.
- På en nyckel måste det också finnas en sträcka på sex på varandra följande spärrelement där högst två intilliggande nedskärningar får vara lika.
- En nyckel där nedskärningsdjupet i en nedskärning skiljer sig från rätt nyckel ska givetvis inte kunna öppna låset.

Detta ger en hel del extra information om hur en nyckel ser ut. Om man försöker skapa en nyckel utifrån en relativt otydlig bild kan man, om man har tur, eliminera ett par av de möjliga nycklarna efter att ha applicerat dessa restriktioner.

Det finns även regler för hur lås ska klara fysiska attacker och slitage, men det är inget som är relevant i det här sammanhanget.

2.3 Mekaniska lås som kryptosystem

En nyckel av den typ som beskrivs i avsnitt 2.1 har ett antal nedskärningar av olika djup. Man kan se dessa som den kryptografiska nyckeln som dekrypterar ett krypterat meddelande, låset. Och man kan tänka sig nyckeln som en sekvens av siffror där varje siffra representerar hur djupt skuren nyckeln är i den positionen. För ett lås med fem spärrstift skulle en nyckel kunna vara till exempel 13264. Antalet möjliga nycklar är N^P med beteckningar från avsnitt 2.1, P spärrstift som vardera kan skäras på N nivåer.

Man inser direkt att en brute force-attack på låset där samtliga nycklar tillverkas och testas är mer komplicerad än motsvarande attack av ett lösenordssystem där en dator kan testa lösenord på bråkdelen av en sekund.

För $P = 7$ och $N = 5$ måste exempelvis 78125 nycklar testas innan en fungerande nyckel kan garanteras, givet att endast en nyckel kan öppna låset. Antalet möjliga nycklar begränsas dock en aning av reglerna i avsnitt 2.2, men fortfarande är antalet alldeles för stort i förhållande till tiden ett enskilt test tar.

Antagandet att samtliga nycklar måste testas och tillverkas gäller inte i praktiken. Det finns verktyg som konstruerats för att dyrka lås och som används av låssmeder när dessa öppnar ett lås som saknar nyckel. Dyrkning behandlas inte i denna rapport.

En attack av det mekaniska låset går ut på att tillgodogöra sig information om nyckeln som gör att antalet möjliga nycklar minskas. I ett kryptosystem kan sådan information exempelvis vara delar av kryptonyckeln, dess längd och teckenuppsättning om det är fråga om ett lösenord. En metallnyckel är mer publik än ett lösenord i den bemärkelsen att man ofta inte döljer dess utseende lika noga. Problemet att kopiera nyckeln utgör istället en stor del av säkerheten hos låssystemet, men då kopieringen är fullt möjlig för exempelvis en låssmed bortses här från detta och attacken betraktas som helt lyckad om tillräckligt mycket information om nyckeln kan fås fram.

2.4 Huvudnycklar

Betraktas ett låssystem enligt avsnitt 2.3 kan man inse att det finns allvarliga brister hos vissa system. Ett designmisstag i den vanligaste typen av huvudnyckelsystem gör det möjligt att gissa och verifiera beskrivningen av huvudnyckeln en nedskärning i taget.

I ett system där det finns en huvudnyckel används flera delningar på ett eller flera av spärrstiften; en delning som används för huvudnyckeln och en som används för den specifika nyckeln. Man skulle kunna tänka sig ett lås med delningarna 11111 och 22222, och ett lås med 11111 och 33333. Då skulle nyckeln 11111 vara huvudnyckel i det låssystemet. Svagheten i det här låssystemet är att alla möjliga kombinationer av 1 och 2 skulle öppna det första låset. 12222, 11222 och 22111 är alltså giltiga nycklar för detta lås. Representationen av nyckeln som ett femsiffrigt tal följer beskrivningen i avsnitt 2.3.

Denna typ av huvudnyckelsystem möjliggör komplexa låshirarkier. I ett hyreshus ska exempelvis alla hyresgästers nycklar kunna låsa upp ytterdörren och sin egen lägenhetsdörr. Samtidigt får inte en nyckel kunna låsa upp mer än en lägenhet och kanske delar flera trapphus en tvättstuga dit hela husets invånare, som bor i olika trappuppgångar, ska ha tillträde. Brevbäraren behöver en nyckel som öppnar samtliga trappuppgångar men inga lägenheter.

Matt Blaze beskriver i en artikel[4] en metod som gör det möjligt att genom att bara testa ett fåtal olika nycklar kunna ta reda på hur huvudnyckeln ser ut. Anta att man har tillgång till ett lås med tillhörande nyckel, och att systemet har P spärrstift och N möjliga skärningar. Detta ger N^P möjliga nycklar.

Skapa nu, för varje stiftposition p, nycklar som är identiska med ursprungsnyckeln i alla andra stiftpositioner och som antar alla tillåtna skärningsdjup för stiftposition p. Prova nu alla nycklar för varje stiftposition och om en nyckel fungerar, anteckna stiftposition och skärningsdjup. Om huvudnyckeln inte har något stift gemensamt med ursprungsnyckeln kommer två djup att fungera för varje position. Och eftersom man känner till ursprungsnyckeln så är det djup som inte stämmer överens med denna huvudnyckelns skärning. Genom detta har antalet nycklar som behöver testas minskat från N^P till $N{\cdot}P$. Men man kan också modifiera nycklarna mellan varje försök och på så sätt bara använda ett nyckelämne för varje stift. Det innebär att det enda man behöver för att skapa en huvudnyckel är ett lås med fungerande nyckel och P nyckelämnen för det låset.

När Blaze publicerade sin artikel visade det sig att många i låsbranchen redan kände till problemet men att de istället för att försöka göra någonting åt det försökte skydda sig genom att hålla informationen hemlig. Ur ett datorsäkerhetssynsätt är denna typ av "*security through obscurity*" inte så populär. Mekaniska lås kan dock inte uppdateras på samma enkla sätt som till exempel datorprogram kan, och att byta alla låssystem som är sårbara för en sådan attack skulle inte vara praktiskt genomförbart.

Förfarandet att kopiera huvudnycklar kan även vara intressant när det gäller att kopiera nycklar med hjälp av bilder på nycklarna. Att få fotografier på en huvudnyckel kan vara svårt i vissa sammanhang. Då kan det vara lättare att istället ta bilder på "vanliga" nycklar och sedan med hjälp av Blaze metod återskapa huvudnyckeln.

En annan sak som kan underlätta nyckelkopieringen är att man, i de fall där huvudnyckelsystem används, ökar antalet möjliga nyckelkombinationer som öppnar låset. På så sätt gör man, genom att använda ett huvudnyckelsystem, det enklare för en angripare att kopiera nyckeln för ett specifikt lås.

Man kan också, på samma sätt som Blaze beskrev, fila ner nyckeln om det är ett fåtal nedskärningar man är osäker på. Är det många nedskärningar som man inte kan bestämma exakt inses det lätt att det blir för tidskrävande att prova alla olika möjliga kombinationer.

Kapitel 3

Bildkvalitetförbättring

När det gäller forskning på området bildkvalitetförbättring finns det en hel del gjort. Begreppet bildkvalitet är dock inte entydigt, då man med hög kvalitet kan mena både avsaknad av element såsom brus i bilden, eller att bilden helt enkelt har hög upplösning. Här koncentrerar vi oss på sådant som gör att detaljer lättare kan ses i bilderna.

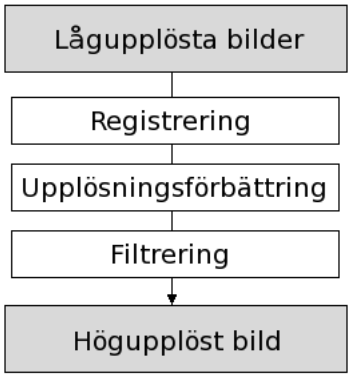
Har man en bild av en viss upplösning är det inte möjligt att öka den och på så vis få fram nya detaljer i bilden. Man skulle då behöva skapa detaljer som helt enkelt inte finns där. Man kan dock tänka sig att man kan gissa sig till hur vissa detaljer skulle se ut vid hög upplösning om man vet lite mer om vad bilden innehåller för typ av objekt. Figur 3.1 visar en bild av en nyckelprofil där upplösningen inte är tillräcklig för att några noggrannare mätningar ska kunna göras av djupet på nyckelns nedskärningar. Annan information kan dock vara möjlig att extrahera ur bilden, exempelvis hur många nedskärningar nyckeln har.

Figur 3.1. Exempel på en bild som inte har tillräckligt hög upplösning för att all information man vill ha om objektet i bilden ska kunna utläsas.

Har man en serie bilder av samma objekt så innehåller varje bild för sig inte nog med information för att man ska kunna öka upplösningen eller skapa en mer detaljerad bild av objektet, men vad gäller för hela serien? Bilder av samma objekt tagna från olika håll säger ju exempelvis mer om objektet än varje bild för sig.

Det är detta, att skapa en detaljerad bild givet en serie bilder, som rapporten fokuserar på och det är forskning kring detta som framför allt har undersökts.

Det finns ett stort antal vetenskapliga artiklar om framtagandet av högupplösta bilder utifrån lågupplösta bildserier, skrivna för olika ändamål. Tillvägagångssättet är ofta liknande, men många olika metoder används vid olika delar av processen. Den vanligaste principiella metoden, som i kapitel 4 till 6 genomgås i detalj, är att först utföra en bildregistrering. Den bestämmer hur bilderna man har att tillgå förhåller sig till varandra rent positionsmässigt. När den är gjord kan man utföra den faktiska upplösningshöjningen, och man avslutar med att filtrera resultatbilden. Då minimerar man bruset i bilden och man ökar skärpan, som ofta sänks under upplösningshöjningen. I kapitel 4 till 6 väljs och motiveras också metoder för det praktiska experimentet. Det principiella förloppet beskrivs schematiskt i figur 3.2.



Figur 3.2. Beskrivning av förloppet vid upplösningshöjning.

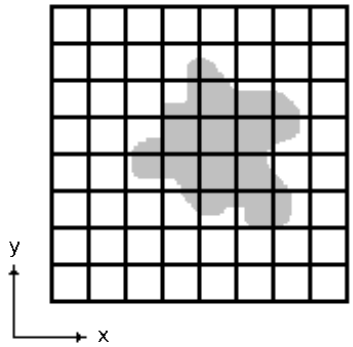
Kapitel 4

Bildregistrering

Då information om ett och samma objekt ska extraheras ur flera bilder gäller det att först bestämma var i bilderna objektet återfinns. Det kan exempelvis röra sig om en serie flygfoton tagna från ett flygplan i rörelse och man vill studera marken mer detaljerat än vad som är möjligt i en av seriens bilder, eller studera om någon viss position på marken har förändrats. Det första man måste göra här är att avgöra hur bilderna förhåller sig till varandra så man vet vilken punkt i en bild som motsvaras av vilken punkt i en annan. Det är uppenbart att flygplanets rörelse givit en förskjutning av markytan i bilderna och känner man bara till precis hur stor denna förskjutning är så kan man fixera samma punkt i olika bilder. Mer allmänt kallas detta att registrera bilder och det kan definieras som matchning av två bilder med avseende på både positioner och intensitet.

Bildregistrering går ut på att finna den optimala positions- och intensitetstransformationen som mappar en bild på en annan med syfte att antingen bestämma transformationsparametrar, markytans förskjutning i exemplet ovan, eller påvisa intressanta skillnader mellan bilderna. Intressanta skillnader i flygfotoexemplet kan vara en bils rörelse på en väg.

För upplösningshöjning, då man vill hämta olika information ur olika bilder i en bildserie av digitala fotografier, måste registreringen av bilderna ske på subpixelnivå. Detta inses om man betänker att positionen av ett objekt i en bild inte bestäms med ett helt antal pixlar från bildkanten. Bilden är snarare en approximation av hur objektet ser ut, vilket framgår av figur 4.1. Där illustreras hur ett objekt fotograferas digitalt till en bild med upplösningen 8×8 pixlar. Om objektet sedan förflyttar sig en aning innan nästa bild i bildserien tas så kommer denna förflyttning inte kunna bestämmas till ett helt antal pixlar. Om man bortser från förskjutningen kommer de två bilderna inte vara identiska eftersom de är approximationer av objektets utseende. Det är förskjutningar som dessa, på subpixelnivå, som gör det möjligt att hämta olika information om samma objekt från olika bilder.



Figur 4.1. Ett objekt ska representeras i en digital bild utan oändlig upplösning.

Här beskrivs ett antal olika transformationstyper följt av en kort genomgång av olika typer av variationer mellan bilder. Man skiljer på de variationerna man vill detektera och distorsion. Distorsion är sådana variationer som man vill avlägsna genom bildregistrering. Därpå följer i avsnitt 4.3 till 4.6 en beskrivning av olika registreringsmetoder, och en metod väljs för det praktiska experimentet.

4.1 Transformationer

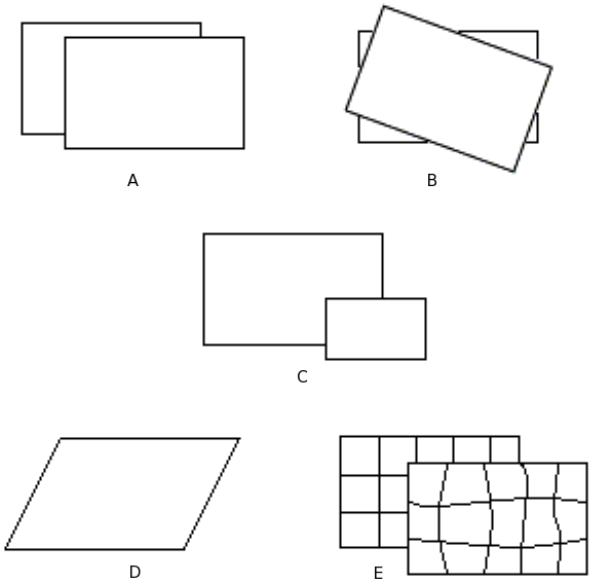
En fundamental egenskap hos en bildregistreringsmetod är vilka typer av transformationer som görs av bilderna. Oavsett om olika variationer kan förekomma är det viktigt att transformationen mappar bilderna med avseende på position utan att förstöra sådan information som man vill extrahera ur bilderna. Det gäller att välja rätt klass av transformation beroende på vilka typer av variationer mellan bilder som kan förekomma. En bild kan exempelvis vara förskjuten och roterad. De vanligaste transformationsklasserna beskrivs här kortfattat.

4.1.1 Euklidisk transformation

En euklidisk eller stel transformation motsvarar objekt- eller kamerarörelser som bevarar relativ storlek, vinklar och parallellitet. Transformationen består av en rotation, en förskjutning och en skalning(A, B och C i figur 4.2).

Euklidiska transformationer är de vanligaste i bildregistreringssammanhang, eftersom de kan matcha två bilder tagna från olika positioner men där inga perspektiv ändrats. Kameran kan även roteras kring sin optiska axel. Transformationen kan uttryckas med fyra parametrar enligt formeln nedan

$$\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} t_x \\ t_y \end{pmatrix} + s \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}$$



Figur 4.2. Olika transformationer.

4.1.2 Affin transformation

En affin transformation är mer generell än den euklidiska och tillåter skjuvning av ett objekt, alltså att vinklar ändras. Dock bevaras även i denna typ av transformation parallellitet. D i figur 4.2 är ett exempel på en affin transformation. Då transformationen är mer generell än den euklidiska krävs fler parametrar. Alla euklidiska transformationer är affina medan det ombytta inte gäller.

$$\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} a_{13} \\ a_{23} \end{pmatrix} + \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}$$

En uppenbar egenskap är att x- och y-axeln kan skalas oberoende av varandra.

4.1.3 Perspektivtransformation

Perspektivtransformationer bevarar inte objekts storlek, vinklar eller parallellitet men linjer mappas på linjer. De uppträder då objekt i 3D projiceras på ett plan. För att kunna använda denna typ av transformationer vid bildregistrering krävs att man känner till avståndet mellan kameran och objekten. Ett specialfall, där scenen som ska projiceras är ett plan, kallas projektiv transformation. Detta kan exempelvis användas för registrering av flygfotografier tagna av samma område men från olika vinklar.

4.1.4 Polynomisk transformation

Polynomiska transformationer står för mer olinjära skillnader mellan bilder. För att mappa två bilder på varandra använder man sig av kontrollpunkter. E i figur 4.2 är ett exempel på en polynomisk transformation.

Att bestämma en polynomisk transformation är möjligt genom att uttrycka den generellt och sedan lösa det ekvationssystem man får om man sätter in sina kontrollpunkter, enligt Brown[5]. Dessa punkter är sådana som kan bestämmas i båda bilderna, och precisionen med vilken dessa kan bestämmas avgör till stor del hur väl registreringen kan genomföras. Graden på polynomen är en avvägningsfråga vad gäller exakthet och beräkningshastighet men i praktiken är ofta andra eller tredje gradens polynom tillräckligt. I allmänhet kan man även säga att distorsion av hög frekvens mellan bilder gör registreringen svår eftersom polynom av hög grad beter sig ganska oförutsägbart. Givetvis hittar man heller i allmänhet ingen exakt lösning till ekvationssystemet. Parametrar som löser systemet så väl som möjligt blir istället en approximation till transformationen. En polynomisk transformation uttrycks av Brown[5] enligt följande formler

$$u = \sum_{i=0}^{m} \sum_{j=0}^{i} a_{ij} x^i y^{j-i}$$

$$v = \sum_{i=0}^{m} \sum_{j=0}^{i} b_{ij} x^i y^{j-i}$$

där (x, y) indexerar referensbilden, (u, v) indexerar bilden som ska registreras och a_{ij} och b_{ij} är de konstanta polynomkoefficienter som ska bestämmas.

4.2 Bildvariationer

Typiskt kan variationer i bilder delas in i tre grupper; rättad distorsion, ej rättad distorsion och intressanta variationer. Rättad distorsion utgörs av sådana variationer som kan modelleras och det är dessa variationer som avgör vilka transformationsklasser som ska användas vid registrering av bilder. Källan till dessa variationer är exempelvis kamerarörelser. Ej rättad distorsion är mer svårmodellerad och kan utgöras av skillnader i väderförhållanden och skuggor. I allmänhet kan man säga att denna typ av distorsion ofta beror på sjäva scenen snarare än hur scenen betraktas, och att registreringen försvåras av dessa variationer. Intressanta variationer är skillnader mellan bilder som man genom registreringen vill detektera. Det kan exempelvis röra sig om förflyttning av ett objekt i bilderna, såsom en bils rörelse på en väg i det tidigare flygfotoexemplet i inledningen av kapitlet. När man väljer registreringsmetod bör man ta hänsyn till samtliga typer av variationer och extra viktigt är det att skilja på ej rättad distorsion och intressanta variationer om båda dessa kan uppträda lokalt i bilden.

4.3 Korskorrelation

Den enklaste formen av registrering är korskorrelation, som använder ett statistiskt angripssätt. I sig själv är korskorrelation inte en metod för registrering, utan den beräknar ett mått på hur lika två signaler, eller i det här fallet bilder är. Detta använd som det viktigaste verktyget i flera metoder för registrering.

För att korskorrelation ska vara användbart för registrering krävs att bilderna som ska analyseras är relativt lika. Då det bara handlar om en förskjutning kan korskorrelation ge ett mycket gott resultat, men vid mer komplicerade transformationer är metoden inte lika användbar. För två bilder, I och T, kommer den tvådimensionella normaliserade korskorrelation-funktionen att ge ett mått på likhet för varje förskjutning:

$$C(u,v) = \frac{\sum_x \sum_y T(x,y) I(x-u, y-v)}{\sqrt{\sum_x \sum_y I^2(x-u, y-v)}}$$

Om bilderna matchar varandra, förutom i intensitet, med en förskjutning (i,j), kommer korskorrelationen ha en topp i $C(i,j)$. Genom att beräkna C över alla möjliga förskjutningar kan den sökta förskjutningen mellan bilderna således beräknas. För att inte intensiteten ska påverka beräkningarna måste korskorrelationen normaliseras.

Ett liknande sätt att beräkna förskjutningen är genom att beräkna summan av differansen i kvadrat i varje position i bilderna:

$$D(u,v) = \sum_x \sum_y (T(x,y) - I(x-u, y-v))^2.$$

D kommer här att minska då graden av likhet mellan bilderna ökar. Genom att minimera D kan man således beräkna förskjutningen. Även här måste man normalisera för att inte intensiteten ska störa beräkningarna, och eftersom $\sum_x \sum_y T^2(x,y)$ är konstant för varje position (u,v) kan, liksom tidigare, den lokala bildenergin $\sum_x \sum_y I^2(x-u, y-v)$ användas för normalisering.

En användbar egenskap hos korrelation ges av korrelationsteoremet. Det säger att fouriertransformen av korrelationen mellan två bilder är produkten av fouriertransformen av den ena bilden och komplexkonjugatet för fouriertransformen av den andra. Detta ger oss ett annat sätt att beräkna korrelationen mellan två bilder. Istället för att representera bildfunktionen i den spatiala domänen kan man istället representera den som en fouriertransform i frekvensdomänen. Detta görs för att fouriertransformerna kan beräknas effektivare än de tidigare beskrivna metoderna. Det vanligaste tillvägagångssättet är att fouriertransformen för korskorrelationen beräknas för varje tillåten transformation.

Ett problem är dock att då det finns många tillåtna transformationer blir beräkningarna snabbt för tunga för att utföras. Men om det bara handlar om ett fåtal möjliga transformationer, till exempel en förskjutning, är korrelationsberäkningar ett utmärkt verktyg. För ytterligare fördjupning i korskorrelation rekommenderas "A survey of image registration techniques" av Brown[5].

4.4 Fouriermetoder

De metoder som beskrivs i detta avsnitt använder flera egenskaper hos fouriertransformer för att registrera bilder. Förskjutning, rotation, reflektion och skalning går alla att upptäcka med fourieranalys. Med fourieranalys arbetar man med registreringen i frekvensdomänen till skillnad från i avsnitt 4.3 där fouriertransformer enbart användes som ett verktyg medan själva registreringen utfördes i den spatiala domänen. Genom att använda frekvensdomänen kan man uppnå god rubusthet vid korrelerat och frekvensberoende brus. Fouriermetoder klarar dock bara av bilder med en euklidisk transformation relativt varandra. I det här avsnittet beskrivs först den enklaste metoden som använder fourieranalys föreslagen av Kuglin och Hines[20]. Den kallas faskorrelation och kan användas för att registrera bilder som förskjutits relativt varandra. Sedan beskrivs en utvidgning av denna metod som hanterar bilder som har blivit både skiftade och roterade i förhållande till varandra. För att kunna beskriva faskorrelation, beskrivs först de uttryck som behövs inom fourieranalysen. Mer information om fouriertransformmetoder ges av Brown[5].

Fouriertransformen av en bild $f(x, y)$ är en komplex funktion. Varje funktionsvärde har en reell del $R(\omega_x, \omega_y)$ och en imaginär del $I(\omega_x, \omega_y)$ vid varje frekvens (ω_x, ω_y) av frekvensspektrumet:

$$F(\omega_x, \omega_y) = R(\omega_x, \omega_y) + iI(\omega_x, \omega_y).$$

Detta kan alternativt uttryckas i exponentialform som

$$F(\omega_x, \omega_y) = |F(\omega_x, \omega_y)|e^{i\phi(\omega_x, \omega_y)}$$

där $|F(\omega_x, \omega_y)|$ är fouriertransformens amplitud och $\phi(\omega_x, \omega_y)$ dess fasvinkel. Amplituden i kvadrat beskriver energin i varje frekvens av bilden och uttrycks

$$|F(\omega_x, \omega_y)|^2 = R^2(\omega_x, \omega_y) + I^2(\omega_x, \omega_y).$$

Fasvinkeln beskriver fasskiftning vid varje frekvens och definieras som

$$\phi(\omega_x, \omega_y) = \tan^{-1} \frac{I(\omega_x, \omega_y)}{R(\omega_x, \omega_y)}$$

Faskorrelation använder sig av förskjutningsegenskapen i fouriertransformen, ibland kallad skiftteoremet. Givet två bilder f_1 och f_2 som bara skiljer sig åt med en förskjutning (d_x, d_y),

$$f_2(x, y) = f_1(x - d_x, y - d_y),$$

kommer motsvarande fouriertransformer att vara relaterade enligt

$$F_2(\omega_x, \omega_y) = e^{-j(\omega_x d_x + \omega_y d_y)} F_1(\omega_x, \omega_y).$$

Det vill säga, de två bilderna har samma fouriermagnitud men en fasskillnad direkt relaterad till deras förskjutning. Denna fasförskjutning är given av $e^{j(\phi_1 - \phi_2)}$.

Om man beräknar krysseffektspektrumet för de två bilderna, definierat som

$$\frac{F_1(\omega_x,\omega_y)F_2^*(\omega_x,\omega_y)}{|F_1(\omega_x,\omega_y)F_2^*(\omega_x,\omega_y)|} = e^{(\omega_x d_x + \omega_y d_y)}$$

där F^* är komplexkonjugatet till F, är fasen av krysseffektspektrumet lika med fasskillnaden mellan bilderna enligt skiftteoremet. För att beräkna hur mycket förskjutna bilderna är beräknas inverstransformen av krysseffektspektrumets fas. Denna inverstransform blir en impuls, det vill säga den är approximativt noll överallt utom i den punkt som motsvarar bildernas förskjutning. Så genom att beräkna placeringen av toppen på inverstransformen kan man beräkna hur mycket bilderna är förskjutna. Eftersom fasskillnaden för varje frekvens påverkar lika mycket kommer brus inom små frekvensintervall inte störa beräkningarna. Detta gör också att denna metod lämpar sig väl då man har olika yttre förhållanden i de olika bilderna, till exempel olika belysning i bilderna eller att bilderna är tagna av olika sensorer. Däremot är metoden mycket känslig för vitt brus (begreppet förklaras i kapitel 6) eftersom detta påverkar alla frekvenser och på så sätt ger en felaktig fasförskjutning på varje frekvens. För att begränsa inverkan av vitt brus kan man införa en viktningsfunktion för fasskillnaden innan man tar inversen av fouriertransformen, enligt Kuglin och Hines[20].

För att kunna registrera bilder som är både förskjutna och roterade föreslog De Castro och Morandi[7] en metod som till stor del bygger på faskorrelation. Bara en rotation kan hanteras på ett liknande sätt som en förskjutning genom att representera rotationen som en förskjutning i polära koordinater. Både rotation och förskjutning tillsammans är en betydligt mer komplicerad transformation. För att lösa detta utförs beräkningar i två steg. Först beräknas rotationsvinkeln och sen förskjutningen. En rotation av en bild roterar fouriertransformen av bilden med samma vinkel. Om vinkeln är känd kan krysseffektspektrumet roteras och förskjutningen bestämmas enligt faskorrelationsmetoden. Men eftersom vinkeln inte är känd så beräknas fasen av krysseffektspektrumet som en funktion av rotationsvinkeln ϕ och polära koordinater (r,θ) används för att förenkla ekvationen. Detta ger en funktion

$$G(r,\theta;\phi) = \frac{F_1(r,\theta)F_2^*(r,\theta-\phi)}{|F_1(r,\theta)F_2^*(r,\theta-\phi)|}$$

som för den sökta rotationsvinkeln har en form som stämmer överens med vad som kan förväntas för en bild med enbart en förskjutning.

Med denna metod kan man alltså beräkna förskjutningen i två steg. Först bestäms vinkeln som ger den inversa fouriertransformen, av fasen för krysseffektspektrumet, den bästa approximationen av en impuls. Sedan beräknas förskjutningen genom att titta på placeringen av denna puls.

4.5 Point-mapping

Registrering av bilder till en referensbild där typen av transformation inte är känd förekommer ofta. Man behöver då en registreringsmetod som inte är beroende av transformationen och att det blir ett problem med alla tidigare beskrivna metoder inses lätt. Det kan exempelvis röra sig om fotografier tagna av en scen från olika vinklar. En perspektivtransformation skulle kunna användas här men ofta är avstånd i bilderna olika och okända.

Den vanligaste lösningen i liknande situationer är point-mapping, eller landmark-mapping som den också kallas. Det är en metod som går ut på att man först hittar vissa punkter, kontrollpunkter, i referensbilden som kan mappas mot punkter i bilden som ska registreras. Koordinaterna för dessa punkter låter man sedan ingå i ett ekvationssystem, vilket exemplifieras i avsnitt 4.1.4

Point-mapping används ofta men för de bilder som ska registreras under detta projekt passar metoden absolut inte, av den enkla anledning att kontrollpunkter som tillåter registrering på subpixelnivå inte kan hittas i bilderna. Point-mapping är lämpligare där registreringen har andra syften än att förbättre bildupplösningen. Det kan exempelvis röra sig om att detektera förändringar i naturen, hitta intressanta skillnader mellan olika röntgenbilder eller jämföra kartor med flygfotografier. Även om man i fallet upplösningsförbättring finner det svårt att motivera användning av point-mapping så bör det nämnas att det är en väldigt vanligt förekommande metodik vid bildregistrering. 1976 använde Bernstein[2] point-mapping för att få bort distorsion, orsakad av jordens krökning och kameraförflyttning, i satellitbilder. Maguire med flera[21] använde liknande metoder 1990 för att identifiera organ i röntgenbilder.

4.6 Vald metod

I den valda metoden, som påminner mycket om den metod som Koo med flera[19] beskriver, utförs registreringen genom att låta förskjutningen variera medan skillnaden mellan bildernas endimensionella karaktär i både x och y-led minimeras.

Metoden är i princip en korskorrelationsbaserad metod som beskrivs i avsnitt 4.3, men istället för de tunga beräkningar som krävs i dessa delar man upp beräkningarna och utför dem för en dimension i taget. De funktioner som bilden, som är tvådimensionell, delas upp i benämns av Koo[19] som bildens karaktäristiska kurvor. Att man utför beräkningarna i en dimension i taget gör att mer avancerade transformationer, till exempel rotationer, inte går att registrera. Men då objekten som ska registreras under experimentet förutsätts röra sig endast i x- och y-led utgör inte detta något problem.

Metoden föreslås av Kim med flera[18] för att bestämma rörelsers karaktär i digital video. Där motiveras den med att de tvådimensionella bildernas endimensionella karaktär var för sig studeras snabbare än vad som görs i vedertagna metoder. Dessa baseras ofta på gradienten hos intensiteten i bilder, blockmatchning eller

beräkningar i frekvensdomänen, där tidskrävande beräkningar görs över hela de tvådimensionella bilderna. Det exakta förfarandet vid beräkning av en bilds karaktäristiska kurvor beskrivs av Koo[19] och Kim[18]. Kim[18] behandlar inte områden lika nära vårt eget som Koo[19] men visar många intressanta tillämpningar och varianter av metoden. Metoden föreslås för videokomprimering, detektering och spårning av rörelser samt digitala gyroskop, alltså stabilisering av digital video.

Metoden förklaras enklast genom ett litet exempel. Bilderna A och B i figur 4.3 påminner mycket om varandra. De föreställer samma objekt men i den ena är objektet förskjutet en aning om man jämför med den andra. Dessa förskjutningar uppkommer typiskt då kameran rör sig över den fotograferade scenen, eller som här, när bakgrunden är enfärgad, också då objektet rör sig mot bakgrunden. Uppgiften är att ta reda på hur mycket objekten är förskjutna i förhållande till varandra.

Figur 4.3. Två bilder av objekt som förskjutits i förhållande till varandra.

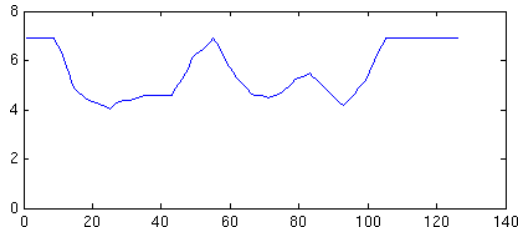
Om vi antar att vi har en digital bild med $M \times N$ pixlar kan bildens två karaktäristiska kurvor beräknas enligt följande formler.

$$f_x(j) = \frac{1}{M} \sum_{i=1}^{M} a_i l_{ij}$$

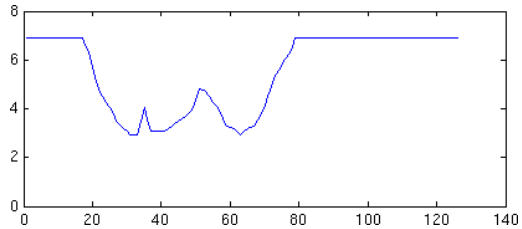
$$f_y(i) = \frac{1}{N} \sum_{j=1}^{N} b_j l_{ij}$$

f_x är alltså en kurva som för varje x antar värdet av en viktad och normaliserad summa av pixelvärden, l, i motsvarande pixelkolumn. På samma sätt är värdena hos f_y summor över pixelraderna. För bilder där det inte är objektets rörelse som orsakar förskjutningarna utan snarare kameran, kan vikterna a och b sättas till 1 och hela bilderna viktas lika. I övriga fall kan resultatet förbättras om man med vikterna plockar ut det objekt i bilden man är intresserad av.

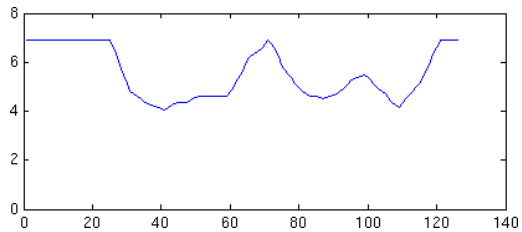
För A och B i figur 4.3 ser de karaktäristiska kurvorna ut enligt figur 4.4 till 4.7.



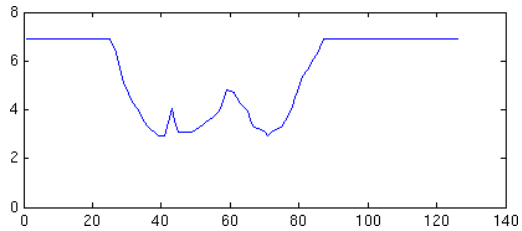
Figur 4.4. Karaktäristisk kurva f_x för bilden A i figur 4.3.



Figur 4.5. Karaktäristisk kurva f_y för bilden A i figur 4.3.



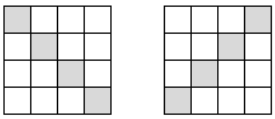
Figur 4.6. Karaktäristisk kurva f_x för bilden B i figur 4.3.



Figur 4.7. Karaktäristisk kurva f_y för bilden B i figur 4.3.

Lägger man kurvorna i figur 4.4 och 4.6 samman, vilket gjorts i figur 4.9 ser man att de båda exempelbildernas karaktäristiska kurvor f_x är identiska sånär som på förskjutningen dem emellan. Detta beror på att bilderna är identiska sånär som på en förskjutning. I ett mer realistiskt exempel med digitala fotografier skulle kurvorna inte vara helt lika, men starkt korrelerade och givetvis gäller samma sak för kurvorna f_y.

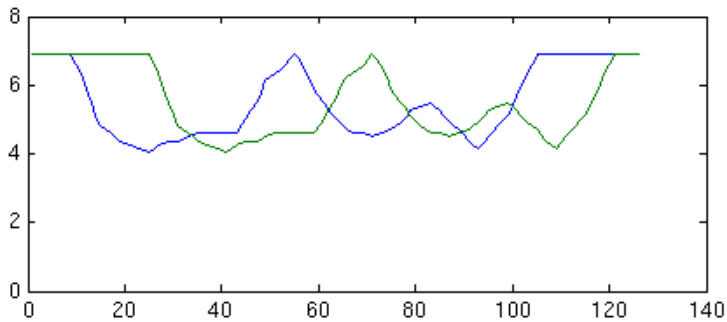
Två helt identiska bilder ger helt identiska karaktäristiska kurvor, f_x och f_y. Det omvända gäller inte, då det kan liknas vid att försöka lösa $M \cdot N$ variabler ur $M+N$ ekvationer. De aktuella bilderna är dock starkt korrelerade och endast en aning förskjutna så i praktiken gäller även förhållandet omvänt för den här tillämpningen. Att det inte gäller i det allmäna fallet blir uppenbart om man betraktar de två bilderna i figur 4.8, vilka har precis identiska karaktäristiska kurvor med vikterna a_i och b_j satta till 1.



Figur 4.8. Två bilder med identiska karaktäristiska kurvor.

I fallet med starkt korrelerade bilder och i denna tillämpning gäller att om två karaktäristiska kurvor är identiska har de skapats utifrån identiska bilder och man kan därför använda dem för att detektera skillnader mellan bilder.

Kurvorna f_x och f_y för två digitala bilder kan skjutas relativt varandra och för varje förskjutning kan en skillnad mellan dem beräknas. Denna skillnad kan så minimeras för x-kurvorna för sig och y-kurvorna för sig och man beräknar på så vis förskjutningen mellan de digitala bilderna i x respektive y-led. Då man som i figur 4.9 har två identiska men förskjutna karaktäristiska kurvor minimeras skillnaden dem emellan till noll.



Figur 4.9. Karaktäristiska kurvor f_x för bild A repektive B i figur 4.3.

För bilder som inte är identiska men starkt korrelerade, så som två digitala fotografier av samma scen, gäller inte detta men genom att minimera kurvskillnaden registreras bilderna. Följande formler gäller för registrering enligt denna metod.

$$P_x(s) = \frac{1}{N-s} \sum_{i=\max[1,-s]}^{\min[N-s,N]} |f_x(i) - f'_x(i-s)|$$

$$P_x(s_x) = \min_s P_x(s)$$

$$P_y(s) = \frac{1}{M-s} \sum_{j=\max[1,-s]}^{\min[M-s,M]} |f_y(j) - f'_y(j-s)|$$

$$P_y(s_y) = \min_s P_y(s)$$

N och M är bildupplösning i x respektive y-led. s_x och s_y, som minimerar uttrycken för skillnaden mellan de karaktäristiska kurvorna för referensbilden (kurvorna f_x, f_y) och den bild som registreras (kurvorna f'_x, f'_y), är den horisontella respektive vertikala förskjutningen mellan bilderna.

Koo[19] beskriver en hårdvarunära implementation som beräknar förskjutningen med en kvarts pixel och kan köras i realtid på bilder tagna med exempelvis en videokamera. För experimentet som utförts har en implementation i MATLAB gjorts.

Kim med flera[18] utnyttjar karaktäristiska kurvor för videokomprimering. Skillnaden mellan kurvorna för två på varandra följande bilder beräknas och resultatet är två nya kurvor med nollskillda värden i de punkter där bilderna skiljer sig. Detta kan användas vid exempelvis videokonferanser då realtidsvideo är nödvändigt och man vill skicka så lite data som möjligt mellan de aktiva parterna. Ansikten som samtalar är även exempel på video där en stor del av de på varandra följande bilderna är lika eller tillräckligt lika för att kunna betraktas som lika. De kurvor man beräknat som skillnaden mellan två bilders karaktäristiska kurvor har typiskt spikar vid munnen eller andra delar av ansiktet på de personer som videokonfererar, men är i övrigt noll. Spikarna i kurvorna talar om vilka regioner av bilderna som behöver skickas för att en bild ska kunna skapas givet föregående bild.

Den presenterade metoden lämpar sig bra för detta ändamål då realtidsvideo är ett måste, och tyngre beräkningar för att välja intressanta regioner i filmen skulle kräva mer och dyrare hårdvara. Detta är även ett exempel på rörelsedetektering då spikarna i kurvorna talar om både att det förekommer en rörelse mellan bilderna men också var någonstans. En tillämpning föreslås där en rörlig kamera matar video till ett system som centrerar kameran kring den position där rörelse förekommer, och på så vis låter kameran följa vad som rör sig i dess närhet.

Ett digitalt gyroskop kan användas när en kamera vibrerar kraftigt. Det gäller då att plocka ut den del av videon som är stabil, alltså delar av videon som ryms inom flera efterföljande bilder trots att kameran rört sig och Kim[18] resonerar kort kring hur detta kan göras genom att studera de karaktäristiska kurvorna.

Kapitel 5

Upplösningsförbättring

Låg upplösning är ofta ett problem när det gäller digitala bilder. Man vill ofta se mer detaljer i bilderna än vad som är möjligt i de bilder man har att tillgå. Om detta beror på att kameror med tillräckligt hög upplösning inte finns eller om det är för att man inte har råd med en tillräckligt högupplöst kamera spelar mindre roll, då problemet är detsamma.

Det finns metoder för att höja upplösningen hos en digital bild genom att interpolera fram nya pixelvärden, men korrekta värden till nya pixlar kan man i allmänhet inte konstruera ur en enstaka lågupplöst bild. Däremot kan man ur en serie bilder av samma objekt extrahera mer information än ur en enda bild, om bilderna av objektet är aningen förskjutna i förhållande till varandra på subpixelnivå, se kapitel 3. Detta gör dock inte att man automatiskt kan konstruera högupplösta bilder perfekt. I praktiken har man aldrig bilder med precis rätt förskjutningar och man har ofta heller inte tillräckligt många bilder för att all högupplöst information ska finnas representerad i de lågupplösta bilderna. Det finns olika metoder för att interpolera fram högupplösta bilder och de mest omtalade metoderna i publicerade artiklar är Iterative Backprojection, Projection Onto Convex Sets, Maximum Apostiori och frekvensdomänmetoder. Dessa metoder presenteras här och en metod väljs för det praktiska experimentet.

5.1 Iterative Backprojection (IBP)

Som namnet antyder är metoden iterativ och i varje steg resulterar en approximation av den högupplösta bilden och samtliga de lågupplösta bilder man har att tillgå i en ny approximation som används i iterationens nästa steg. Målet är att hitta en högupplöst bild från vilken lågupplösta bilder kan skapas genom simulering av hur de riktiga lågupplösta bilderna skapats och att de simulerade och tillgängliga bilderna är lika. Då de är lika har man en god approximation av den högupplösta bilden.

Man får alltså anta att man vet hur de lågupplösta bilderna skapats samt att man kan registrera dem på subpixelnivå, enligt Peleg med flera[26]. Som initialt värde på approximationen till den högupplösta bilden brukar man välja en av de lågupplösta bilderna projicerad på den högre upplösningen.

Optimeringsprocessen beskrivs av Peleg[26] med följande notation. Med en mängd lågupplösta bilder, $\{O_i(x,y)\}$, och en intial gissning av den högupplösta bilden, $G^{(0)}(x,y)$, simuleras en mängd lågupplösta bilder, $\{L_i^{(0)}(x,y)\}$. Felet, $E^{(0)}$, kan nu beräknas.

$$E^{(0)} = \sum_i \sum_{(x,y)} |L_i^{(0)}(x,y) - O_i(x,y)|$$

I varje pixel med gissningen $G(x,y) = l$, undersöks så felet $E^{(0)}$ för $\{l-1, l, l+1\}$. Ändringen i L_i måste alltså beräknas, och $G(x,y)$ ges det värde som minimerar felet.

Peleg[26] antar att nedsamplingen från den högupplösta bilden görs genom att den delas in i rutor om 4×4 pixlar, från vilka ett viktat medelvärde beräknas och ges som pixelvärde i en lågupplöst bild. Viktmatrisen ges av A, indelningen av en lågupplöst pixel av P, och p' ger värdet av en lågupplöst pixel.

$$A = \frac{1}{36}\begin{bmatrix} 1 & 2 & 2 & 1 \\ 2 & 4 & 4 & 2 \\ 2 & 4 & 4 & 2 \\ 1 & 2 & 2 & 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ p_{41} & p_{42} & p_{43} & p_{44} \end{bmatrix}$$

$$p' = \sum_{i=1}^{4} \sum_{j=1}^{4} a_{ij} p_{ij}$$

Beroende av hur den högupplösta bilden delas in kan alltså 16 olika lågupplösta bilder konstrueras från en högupplöst. Metoden testas för 4, 8 och samtliga 16 lågupplösta bilder men det sista fallet konstateras vara inget annat än att öka skärpan hos en suddig variant av den högupplösta bilden. Vanliga metoder för höjning av skärpa kan användas i det fallet men när inte samtliga lågupplösta bilder används krävs IBPs iterativa steg. Beroende på hur skillnaden mellan simulerade och observerade lågupplösta bilder minimeras i IBP, bör man betänka om lokala optimum kan förekomma.

5.2 Projection Onto Convex Sets (POCS)

Metoder som är baserade på POCS bildar ett vektorrum innehållande en konvex villkorsmängd där den sökta högupplösta bilden ligger i vektorrummet och omsluts av villkorsmängden. Varje pixel i sekvensen av de lågupplösta bilderna som används vid upplösningsförbättringen utgör tillsammans en villkorsmängd. Associerad med varje villkorsmängd finns också en projektionsoperation vilken mappar en godtycklig punkt i vektorrummet till den närmaste punkten i villkorsmängden. En godtycklig uppskattning av bilden projiceras iterativt på villkorsmängden och den bild som fås fram genom itereringen konvergerar mot en punkt innesluten av villkorsmängden. Denna punkt är dock i allmänhet inte unik utan beror på den första gissningen av den sökta bilden.

Patti med flera[25] visar hur metoden kan användas för att konvertera lågupplöst video till högupplösta bilder. Stark och Oskoui[37] beskriver metoden utförligt och motiverar den med att framtagandet av projektionerna för alla villkorsmängder ofta är mycket lättare att finna än den projektion som i ett enda steg ger lösningen. De använder POCS för att rekonstruera en bild utifrån ofullständig information om denna, och jämför olika antal iterationer med en tom bild som initial gissning. Hur brus påverkar restaureringen diskuteras också.

5.3 Maximum A-Posteriori (MAP)

Maximum A-Posteriori (MAP) är en stokastisk metod med vilken man utnyttjar Bayesianska sannolikhetsmetoder för att exempelvis rekonstruera bilder. En mest trolig högupplöst bild kan skattas givet de lågupplösta bilderna. Altunbasak och Patti[1] använder MAP för att höja upplösningen på en stillbild tagen ur MPEG video och de motiverar metoden med att den ger en statistiskt optimal rekonstruktion. MAP-skattningen formuleras här enligt följande formel.

$$\tilde{f} = \max_{y} P(y|g_1, g_2, ..., g_N)$$

Den högupplösta bilden kallas $\tilde{f}$ och g_i representerar DCT-koefficienter (Discrete Cosine Transform). Därpå appliceras Bayes regel och beräkningen utförs med ickelinjära optimeringsmetoder. Verveer och Jovin[39] använder istället MAP för att utföra bildrestaurering på bilder tagna med mikroskop. Algoritmer för suddiga bilder som är störda av antingen gauss eller poisson-fördelat brus presenteras och testas, med gott resultat.

5.4 Frekvensdomänmetoder

När ekvationssystemet för den högupplösta bilden ställs upp i den spatiella domänen får man en ekvation för varje högupplöst pixel och lika många okända. Dessa är alla beroende av varandra så för stora bilder blir ekvationssystemen väldigt krävande att lösa. Om man vill höja upplösningen med en faktor L tar man fouriertransformen av ekvationssystemet och då får man istället L^2 ekvationssystem för varje frekvenspar i fouriertransformen för den högupplösta bilden $F(\omega_1, \omega_2)$. Ekvationssystemen för varje frekvenspar är inte beroende av ekvationer för andra frekvenspar och $F(\omega_1, \omega_2)$ kan därför återskapas för varje frekvenspar oberoende av de andra. Eftersom det finns additivt brus i bilden kan inte ekvationerna lösas exakt utan får lösas med minsta kvadratmetoden. Tekalp med flera[38] beskriver mer utförligt frekvensdomänmetoder som tar hänsyn till både aliasingeffekter och brus.

5.5 Vald metod

Samtliga nämnda metoder ger högupplösta bilder som är suddiga i jämförelse med en skarp bild från vilken man skulle kunna nedsampla fram de lågupplösta bilder som använts. Elad och Hel-Or[9] visar att den högupplösta bilden kan fås genom pixelvis medelvärdesberäkning från de lågupplösta bilderna följt av en reducering av den högupplösta bildens suddighet om följande villkor gäller:

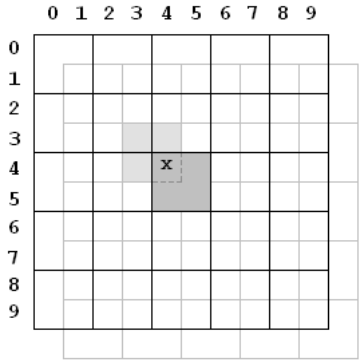
- Alla operationer med vilka upplösningen har minskats är samma. Detta gäller typiskt om alla lågupplösta bilder är tagna med samma kamera eller kameror som samplar bilder på samma sätt.
- Alla operationer med vilka en bild görs suddig är samma. Detta är typiskt beroende av kamerans, som de lågupplösta bilderna är tagna med, PSF (Point Spread Function). Begreppet PSF förklaras i avsnitt 6.1.
- Alla spatiella transformationer ska vara enkla förskjutningar. Alla förskjutningar är heltal men då de representeras i den höga upplösningen utgör denna avrundning inte något märkbart fel i resultatet.
- Det additiva bruset i bilderna är vitt och lika fördelat för samtliga lågupplösta bilder.

Detta har även tidigare gjorts av Gillete med flera[11], men då mer intuitivt och utan matematisk motivering. Att dela upp upplösningsförbättringsoperationen i sammanslagningen av de lågupplösta bilderna och reduceringen av resultatets suddighet innebär en ordentlig förenkling av själva problemet, och då ovanstående villkor uppfylls i det praktiska försök som beskrivs i denna rapport motiverar Elad[9] väl användandet av metoden för försökets upplösningsförbättring.

För att generera bilder med dubbelt så hög upplösning (4 gånger så många pixlar) som de bilder man utgått ifrån krävs registrering på subpixelnivå med en halv pixels noggrannhet i både x och y-led, enligt Elad[9]. Högupplösta pixlar fylls med medelvärdet av pixelvärden från de lågupplösta bilderna. Pixelvärden från rätt pixlar i de lågupplösta bildern hämtas enkelt när man känner till förskjutningen bilderna emellan enligt figur 5.1. Där ses ett exempel där värdet på en pixel, markerad x, kan beräknas enligt följande formel.

$$p(4,4) = \frac{p_1(2,2) + p_2(1,1)}{2}$$

$p(4,4)$ är den högupplösta pixeln, markerad med x, i figur 5.1. p_1 och p_2 är de lågupplösta bilderna.



Figur 5.1. Illustration av hur en högupplöst pixel x får bidrag ur två lågupplösta pixlar då bilder förskjutits en halv pixel i förhållande till varandra.

Det är uppenbart att man på motsvarande sätt som i figur 5.1 kan slå samman fler bilder än två, genom att beräkna medelvärden från fler pixlar. Implementationen tillåter även höjning av upplösning med mer än två gånger i ett steg. Man måste då registrera de aktuella bilderna med högre noggrannhet men annars hämtas pixelbidragen i analogi med figur 5.1.

Kapitel 6

Bildfiltrering

Brus i digitala bilder brukar delas in i två grupper; oberoende brus och brus som beror av data. Det oberoende bruset kan ofta enkelt modelleras medan den andra typen av brus medför mer komplicerade matematiska, olinjära, modeller. Därför antar man så långt det är möjligt att bruset som förekommer i de bilder man ska behandla är oberoende. Denna typ av brus brukar beskrivas med den additiva brusmodellen nedan där c är bilden, a är den sanna bilden och n är funktionen som motsvarar delen brus i varje pixel, enligt Fisher med flera[10].

$$c(i,j) = a(i,j) + n(i,j)$$

Additivt brus har ofta medelvärde noll och kan beskrivas med dess varians σ_n^2. Hur mycket en bild påverkas av bruset anges med SNR (Signal to Noise Ratio) som beräknas enligt följande formel.

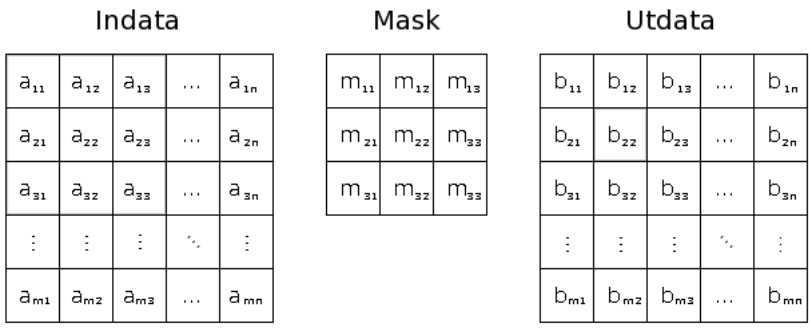
$$SNR = \frac{\sigma_a}{\sigma_n} = \sqrt{\frac{\sigma_c^2}{\sigma_n^2} - 1}$$

σ_c^2 och σ_a^2 är bildens och den sanna bildens varianser.

Typiskt är bruset av den art att det är jämnt fördelat över frekvensdomänen medan bilden i övrigt består av mestadels lågfrekvenskomponenter. Denna brustyp kallas för vitt brus och rent intuitivt kan man filtrera bort det med ett lågpassfilter. För digitala bilder görs detta antingen med ett frekvensfilter eller ett spatialt filter. Det sistnämnda föredras ofta eftersom det kan göras enklare, ur datorkraftsynpunkt. En vanlig källa till vitt brus är det faktum att fotografering handlar om att räkna fotoner, vilket är behäftat med ett slumpmässigt fel som är distribuerat enligt en gaussfördelning med medelvärde noll, enligt Fisher[10]. Det innebär alltså att varje pixel i den brusiga bilden är summan av värdet, som pixeln borde ha, och ett gaussdistribuerat slumptal.

En annan typ av brus som också är oberoende av datat i bilden är intensitetsspikar, vilka uppkommer på grund av fel i dataöverföring av bilden. Denna typ av brus yttrar sig ofta som pixlar med antingen maximalt eller minimalt värde, alltså svarta eller vita, medan övriga pixlar inte påverkas alls. Man brukar ange brusets styrka som den procent av alla pixlar som är korrupta.

En väldigt enkel metod att filtrera digitala bilder är att för varje pixel beräkna ett nytt pixelvärde som funktion av värdet i orginalbilden på dels den pixeln men även kringliggande pixlar. Denna typ av filter kallas spatiala filter. Typiskt låter man en fyrkant glida över bilden och resultatpixeln i mitten av fyrkanten beräknas utifrån värdena på samtliga pixlar i fyrkanten, enligt Green[13]. Förfarandet åskådliggörs i figur 6.1.



Figur 6.1. Illustration av hur ett pixelvärde beräknas i en filtrering.

Som exempel filtreras pixeln a_{22}, och resultatet b_{22} beräknas enligt följande formel.

$$\begin{aligned} b_{22} = \quad & (a_{11}m_{11}) + (a_{12}m_{12}) + (a_{13}m_{13}) + \\ & (a_{21}m_{21}) + (a_{22}m_{22}) + (a_{23}m_{23}) + \\ & (a_{31}m_{31}) + (a_{32}m_{32}) + (a_{33}m_{33}) \end{aligned}$$

Operationen kan varieras mycket med olika vikter m_{ij} och olika stora filtreringsmasker. Filtreringen kan även utföras i flera steg för vissa önskade effekter.

Här presenteras ett antal vanliga filtertyper som används för reducering av brus i bilder. Ofta kan de anges med sin viktmatris. Ett problem vid bildfiltrering är att man, när man tar bort brus också gör bilden mindre skarp. Bildsignalens intressanta högfrekventa delar särskiljs inte från det högfrekventa bruset. Redan vid fotograferingen minskas skärpan i en bild på grund av egenskaper hos kameran som beskrivs i avsnitt 6.1.

6.1 Point Spread Function

Om man betraktar en ljuspunkt i en scen som ska fotograferas, så skulle den med perfekt kameraoptik representeras av en punkt även i fotografiet. I verkligheten distribueras istället intensiteten hos punkten kring en punkt i fotografiet enligt kamerans Point Spread Function. Denna kallas även kamerasystemets impulssvar. När en bild skapas filtreras den enligt beskrivningen av spatiala filter i kapitlets inledning med kamerans PSF.

6.2 Medelvärdesfilter

Ett enkelt exempel på ett spatialt filter är ett filter som för varje pixel beräknar medelvärdet av samtliga pixlar i den fyrkant som skjuts över bilden. Detta ger en resultatbild som är suddig, och en större fyrkant ger en suddigare bild. För intensitetsspikar är inte denna typ av filter så lämplig eftersom en pixel med ett värde som skljer sig mycket från övriga pixlar i fyrkanten påverkar medelvärdet för mycket. En annan sak som ska noteras är att kanter i bilden suddas ut, vilket kan var oacceptabelt i vissa applikationer. I figur 6.2 ses ett exempel på ett medelvärdesfilter.

$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$

Figur 6.2. Enkelt medelvärdesfilter givet av en 3×3 pixlar stor mask.

6.3 Medianvärdesfilter

För problemen som medelvärdesfiltreringen innebär är en självklar lösning att istället för att i varje pixel beräkna medelvärdet, beräkna medianen av fyrkantens pixlar. Detta görs genom att sortera pixlarna efter värde och sedan välja värdet av den mellersta. Om antalet pixlar är jämnt väljs medelvärdet av de två mellersta pixlarna. Genom detta förfarande kommer enstaka pixlar som skiljer sig mycket från de övriga i fyrkanten inte påverka resultatet på samma sätt som vid medelvärdesfiltrering och resultatvärdet är alltid ett värde som finns i den brusiga bilden. Detta innebär att man inte suddar ut kanter på samma sätt som vid medelvärdesfiltrering då exempelvis en bild som endast innehåller svarta och vita pixlar ger ett filtrerat resultat där detsamma gäller. Man skapar inte nya pixelvärden, undantaget det fall då fyrkanten innehåller ett jämnt antal pixlar.

Det finns dock även ett problem med denna typ av filter och det är att man i bilder ofta har vitt brus, alltså brus i samtliga pixlar. Medianvärdesfilter fungerar bra när mindre än hälften av de omkringliggande pixlarna är korrupta men när bildens samtliga pixlar är det påverkas medianen av bruset.

6.4 Gaussfilter

Istället för den enkla fyrkant som beskrevs ovan kan pixlarna viktas på olika sätt i filtret. Låter man till exempel vikterna i området som skjuts över bilden anta formen av en normalfördelningskurva, eller gausskurva, kallas filtret för ett gaussfilter. En gausskurvas egenskaper kan anges med σ, standardavvikelsen. I filtersammanhang anges ibland istället egenskaperna med $FWHM$, Full Width at Half Maximum, vilket är vidden på kurvan där den nått halva sin höjd. Förhållandet mellan σ och $FWHM$ anges i följande formel.

$$FWHM = \sigma\sqrt{8\log 2}$$

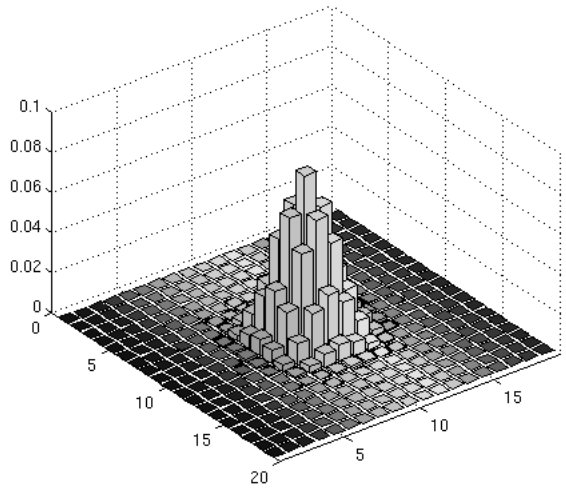
I praktiken skapar man en approximation till gausskurvan och representerar denna med en fyrkant innehållande vikter. I allmänhet vill man undvika en global skalning av pixelvärdena så man låter summan av vikterna vara 1. Filtreringen utförs genom förskjutning av pixelmasken enligt kapitlets inledning, där det också framgår varför gaussdistributionen är intressant i denna rapport.

Förhållandet mellan gaussfördelningen och motsvarande digitala filter visas i figur 6.3 där en approximation av en tvådimensionell och normaliserad gaussfördelning med $\sigma = 1.4$ plottats över 20×20 pixlar. En grövre approximation av samma gaussfördelning ses i figur 6.4, där representerad av en filtreringsmask på 5×5 pixlar.

Gaussfilter filtrerar bort gaussdistribuerat brus, men notera att även bildens högfrekvenskomponenter, små detaljer, undertrycks då det i princip rör sig om en lågpassfiltrering.

6.5 Adaptivt gaussfilter

Gomez[12] presenterar ett adaptivt gaussfilter som, beroende på bildens egenskaper i den punkt som filtreras, sätter σ till ett lämpligt värde mellan fördefinierade min- och max-värden. Om filtret befinner sig nära en diskontinuitet minskas σ och om det istället befinner sig i en mer uniform del av bilden ökas σ. Detta görs för att bevara bildens intressanta diskontinuiteter såsom kanter, men samtidigt effektivt filtrera bort brus. Resultatet som presenteras i artikeln visar att metoden står sig väl emot, eller till och med överträffar, de iterativa metoder man jämfört med. Metoden är även avsevärt snabbare då den endast behandlar varje pixel en gång.



Figur 6.3. Approximation av gaussfördelning, $\sigma = 1.4$

$$\frac{1}{115}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

Figur 6.4. Filtreringsmask för gaussfiltrering.

6.6 Wienerfilter

Filtren som behandlats i avsnitt 6.2 till 6.5 kan användas för att reducera brus i bilder, men de gör samtidigt bilderna suddiga. Det optimala filtret, med avseende på minsta kvadratskillnaden mellan den brusiga och den icke brusiga bilden är wienerfiltret, enligt Young med flera[41]. I fourierdomänen kan ett wienerfilter för signaloberoende additivt brus utryckas enligt följande formel.

$$H_w = \frac{S_{aa}}{S_{aa} + S_{nn}}$$

S_{aa} är energispektrumet för den obrusade bilden och S_{nn} för det additiva bruset. I praktiken har man ofta inte tillgång till S_{aa} men det kan skattas.

I praktiken, men inte i ovanstående modell, är additivt signaloberoende brus sällan den enda distorsionen man har i en bild. Vanligare är att den även är oskarp på grund av kamerarörelser, linsen i kameran, med vilken bilden togs, eller annat. En intressant källa till oskärpa i denna rapport är upplösningsförbättring, framför allt genom medelvärdesbildning.

Om man istället inte hade något additivt brus i distorsionen, utan bilden endast genomgått ett filter av känd karaktär skulle en inversfiltrering kunna genomföras.

$$H(m,n)H_{inv}(m,n) = 1$$

$$H(m,n)F(m,n)H_{inv}(m,n) = F(m,n)$$

Detta behandlas av Biemond med flera[3] och det konstateras att det sällan är praktiskt genomförbart då det inte finns en invers till alla filter samt att additivt brus väldigt ofta förekommer. Brusets karaktär gör att det ofta förstärks om man inte räknar med det i sin modell. Wienerfiltret ovan utbyggt för att ta hänsyn till båda typerna av distorsion kan se ut enligt följande formel.

$$H_w = \frac{H_0^* S_{aa}}{H_0^2 S_{aa} + S_{nn}}$$

eller, som det ofta utrycks

$$H_w = \frac{H_0^*}{H_0^2 + \frac{S_{nn}}{S_{aa}}}$$

H_0 är fouriertransformen av det linjära system som behäftat bilden med oskärpa innan det additiva bruset lades på, och H_0^* är dess komplexkonjugat. Detta följer av den vedertagna modellen för vilken distorsion som förekommer i digitala fotografier, enligt Young[41].

$$c(m,n) = a(m,n)h_0(m,n) + n(m,n)$$

Den bild som observeras är c, och den består av a som genomgått ett filter h_0 samt adderats brus n. Utrycket för H_w fås genom att man minimerar det minsta kvadratfel som vid bildrestaureringen, enligt Biemond[3], kan uttryckas enligt följande formel.

$$E_r = E(|A(m,n) - \hat{C}(m,n)|^2) = E(|A(m,n) - c(m,n)H_w(m,n)|^2)$$

$E(...)$ är väntevärdet över en mängd bilder, A orginalbild och $\hat{C}$ den restaurerade bilden.

Andra modeller för distorsion har också behandlats inom bildbehandlingen. Rudin och Osher[28] presenterar en metod, Total Variation, som även fungerar för multiplikativt brus på följande former.

$$c(m,n) = [a(m,n)h_0(m,n)]n(m,n)$$
$$c(m,n) = a(m,n)h_0(m,n) + a(m,n)n(m,n)$$

I denna rapport är dock bilderna av en sådan natur att den enklare modellen för distorsion kan användas.

Wienerfilter hittar man i många kommersiella bildbehandlingsprogram. Vetenskapliga artiklar som behandlar wienerfilter angriper ofta problematiken med det skattade värdet av egenskaper hos bilden utan distorsion, som behövs vid filtreringen. Hillery och Chin[16] föreslår som alternativ till att skatta variansen ur en enda bild med distorsion en iterativ metod, kallad iterativ wienerfiltrering. Den använder den wienerfiltrerade signalen som en förbättrad version av bilden med distorsion, ur vilken en bättre skattning kan göras. Det visas också att det iterativa filtret konvergerar mot en signal. Konvergens är inte självklart hos iterativa algoritmer som ersätter eller förbättrar wienerfilter. Biemond[3] behandlar speciellt detta.

6.7 Vald metod

För filtrering har wienerfiltrering valts då det är viktigt att inte förlora de högfrekventa delarna i bilderna samtidigt som man vill undertrycka additivt brus och öka skärpan. De högfrekventa delarna är just de detaljer i bilderna som ska mätas så noga som möjligt. Wienerfiltrering lämpar sig också väl för den modell av distorsionen som använts, och som är vedertagen för digitala fotografier, enligt Young[41].

För utförandet av filtreringen har programmet Image Analyzer valts då detta är ett verktyg som är gratis, medan motsvarande Adobe Photoshop plug-ins är relativt dyra. Image Analyzer utför även de variansskattningar som behövs.

Under experimenten gavs ibland bättre resultat av enklare filter, utan brusreducering, och i dessa fall görs ingen wienerfiltrering.

Kapitel 7

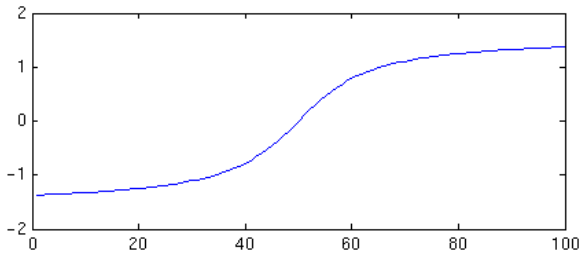
Kantdetektering

En kant i en digital bild är ett hopp i intensitet från en pixel till nästa, och processen att detektera kanter i bilder är en viktig del av bildanalysen eftersom man genom att filtrera ut kanter minskar datamängden och tar bort ointressant information. Samtidigt bevaras bildens strukturella egenskaper, och i allmänhet har man väldigt mycket information i bilder som är värdelös för just det man ska använda bilden till. I fallet med bilder av nyckelprofiler, som studeras i denna rapport, är det endast formen på nyckelns kontur som är av intresse. All annan information bör filtreras bort om detta är möjligt.

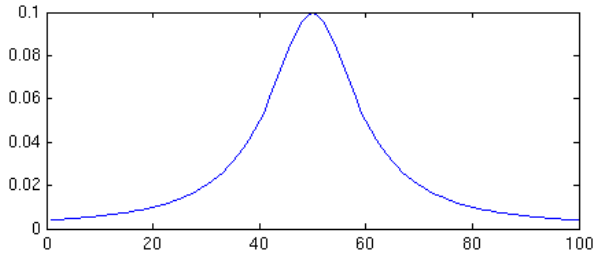
Ett stort problem vid kantdetektering är att det ofta inte bara är bildens faktiska kanter man vill urskilja utan objekten i bilden. Det är väldigt svårt för ett program att särskilja en kant mellan två objekt och en kant som beror av hur ytan på ett objekt ser ut. Människor å andra sidan, har man visat, delar tidigt i bildbehandlingsprocessen i hjärnan in kanter i enstaka kanter och grupper av kanter som hör samman, enligt Grigorescu med flera[14].

De flesta traditionella metoder för kantdetektering kan delas in i två kategorier; *gradient*-metoder och *laplace*-metoder. De förstnämnda detekterar kanter genom att studera förstaderivatan i bilden. Där förstaderivatan överstiger eller understiger ett visst gränsvärde säger man sig ha en kant i bilden. Detta är alltså där bildsignalen har som störst lutning, eller med mer passande termer, där skillnaden i intensitet mellan intilliggande pixlar är stor. Laplacemetoder letar istället efter nollställen i bildsignalens andraderivata, vilka uppträder där förstaderivatan har lokala maximum och minimum, alltså bildsignalens kanter men utan att ta hänsyn till vissa gränsvärden. En signalkant, dess förstaderivata och andraderivata kan ses i figur 7.1, 7.2 och 7.3. En endimensionell signal exemplifieras men samma sak gäller för signaler av högre dimension, i varje dimension för sig.

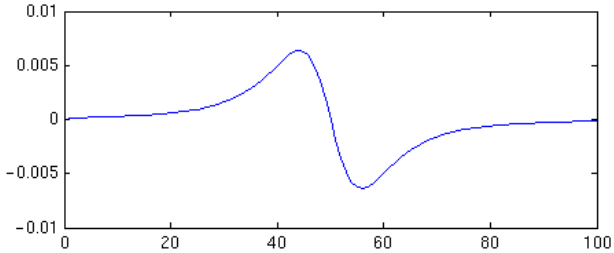
Det har forskats mycket på ämnet kantdetektering, och det är ett område som inte bara bildanalytiker intresserar sig för. När det gäller detektering av kanter är hjärnans syncentra svårslaget och långt innan algoritmer för kantdetektering utvecklades inom den digitala bildbehandlingen studerade läkare hur människor



Figur 7.1. Signal med stor lutning, vilket representerar en kant i en bild.



Figur 7.2. Förstaderivatan av signalen i figur 7.1



Figur 7.3. Andraderivatan av signalen i figur 7.1

och djur kan avgöra var det finns konturer i synfältet, och vilka konturer som hör samman. I början av 1960-talet började man klassificera syncentrats neuroner och man kunde på apor och katter visa att en typ av neuron, kallad simple cell, är känslig för polariteten hos linjer och kanter, enligt Grigorescu[14]. Området har sedan dess varit föremål för aktiv forskning och 1981 delades Nobelpriset i medicin ut till Hubel[17] och Wiesel för deras forskning kring hjärnans syncentra.

Grigorescu[14] presenterar två kantdetekteringsalgoritmer som förbättrar simuleringen av syncentrats simple cells med gaborfilter, vilket ger bättre resultat än traditionella kantdetekteringsmetoder ofta använda inom datorseende.

Den största skillnaden i resultat är att de biologiskt motiverade algoritmerna och användandet av gaborfilter som approximerar ett tidigt filtreringssteg i hjärnan, precis som syncentrat hos människor och apor, bättre skiljer på konturer i texturer och verkliga konturer i röriga landskapsbilder. Detta är svårt med traditionella metoder.

På grund av sin likhet med celler i däggdjurs syncentra har gaborfilter fått ganska mycket uppmärksamhet. De har använts bland annat vid kantdetektering, detektering av egenskaper i fingeravtryck, enligt Yang med flera[40] , ansikten och näthinnor, enligt Smeraldi och Bigun[35] , och detektering av olika texturer i digitala bilder, enligt Weldon med flera[23]. De är linjära och består av en komplex sinusoid, kallad bärare, som är modulerad med ett gaussiskt skal. Mer information om gaborfilter presenteras av Prakash[27].

I avsnitt 7.1 till 7.3 följer mer detaljerade beskrivningar av några av de mer traditionella kantdetekteringsmetoderna och metoden för kantdetektering i det praktiska experiment som beskrivs i rapporten framgår av avsnitt 7.4.

7.1 Sobels metod

Sobels metod är en välanvänd och enkel gradientmetod där man approximerar varje pixels gradient genom att utföra en spatiell filtrering med respektive 3×3 pixlar stor pixelmask, vilka kan ses i figur 7.4, för gradienten i x respektive y-led var för sig.

-1	0	+1
-2	0	+2
-1	0	+1

X

+1	+2	+1
0	0	0
-1	-2	-1

Y

Figur 7.4. Filtreringsmasker för Sobels metod.

Filtreringsförfarandet för den här typen av filter, spatiala filter, beskrivs i inledningen av kapitel 6 med figur 6.1. Resultatet är två nya bilder, en för varje filtreringsmask, där pixelvärdena utgörs av orginalbildens gradient i x-led respektive y-led. Genom att jämföra värdena i resultatbilderna med ett gränsvärde detekteras kanter.

Ett problem vid användning av Sobels metod, och alla gradientmetoder, är just gränsvärdet gradienten måste överstiga för att en kant ska anses detekterad. Om man har en bild ur vilken man kan detektera kanter med ett visst gränsvärde och sedan dubblar bildens storlek gäller inte längre samma gränsvärde. Även om kanterna är de samma så har gradienten halverats genom storleksfördubblingen.

7.2 Laplacefiltrering

På liknande sätt som i Sobels metod kan man approximera en bilds andraderivata. En mask som ofta används för detta är 5×5 pixlar stor och ses i figur 7.5. Vid en spatiell filtrering med denna filtreringsmask fås som resultat en bild där pixelvärdena utgörs av approximationer av orginalbildens andraderivata och är på grund av detta naturligt bruskänsliga, enligt Green[13].

-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	24	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1

Figur 7.5. Filtreringsmask för laplacefiltrering.

Ett par fördelar med denna metod kan identifieras. Man slipper gränsvärden med alla problem som de medför. Istället letar man bara efter positioner där signalen byter tecken. Detta betyder att alla konturer som detekteras är slutna, likt nivåkurvor. En uppenbar fördel som detta ger användandet av bildsignalens andraderivata framför förstaderivata är att objekt enklare kan bestämmas i bilden. Det är dock inte naturligt att sluta alla kanter i alla bilder, vilket istället visar på en nackdel hos metoden.

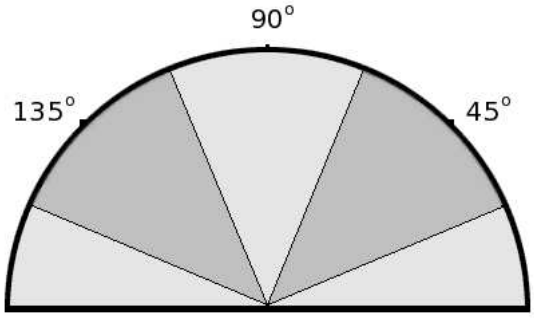
7.3 Cannys metod

En metod som ofta anses vara den bästa av alla traditionella kantdetekteringsmetoder är den som Canny presenterade i en artikel 1986[6]. Han kombinerade tidigare metoder samt gjorde de förbättringar han tyckte sig behöva för att optimera kantdetekteringen enligt följande kriterier.

- Maximera Signal to Noise Ratio, SNR (förklaras i kapitel 6). Detta ökar chansen att en korrekt kant detekteras.
- Ge kanterna deras rätta positioner.
- Minimera antalet detekteringar av en och samma kant. Detta minskar chansen att inkorrekta kanter detekters.

Metodens första steg är en brusreducering som sker genom en spatiell gaussfiltrering, enligt avsnitt 6.4. En positiv bieffekt av detta är att man med filtreringen i viss mån kan välja detaljrikedomen hos resultatet.

Steg två är att med Sobels metod ta fram bildens gradient i x- och y-led. Dessa läggs för varje punkt ihop, med vektoraddition, till en kantmagnitud vars belopp är summan av komponentbeloppen och riktning fås av $\arctan(\frac{y}{x})$. Riktningen avrundas till att peka på en av fyra angränsande pixlar, då arctan har ett värdeintervall av 180 grader. Om 0 grader sägs peka mot den till höger om given pixel liggande pixeln måste även 180 grader peka på denna pixel. 45 grader pekar på pixeln upp till höger, 90 grader uppåt och 135 grader på den pixel som ligger upp till vänster om den givna pixeln. Avrundningar görs enligt figur 7.6 till dessa vinklar. Beloppet av kantmagnituden avgör om en pixel är en kant.

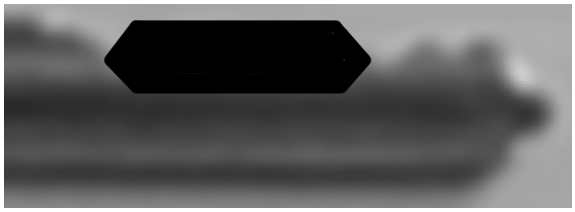


Figur 7.6. Avrundning av riktningsvektorer i Cannys metod.

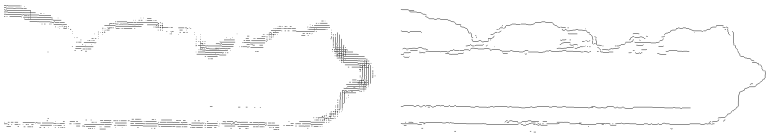
För varje pixel som är en kant undersöker man den pixel som pekas ut av denna och sätter den till noll om den inte är en kant. På så vis fås tunna kantlinjer i resultatbilden. Slutligen markeras pixlar som överstiger ett gränsvärde T_1 som kanter och pixlar som gränsar till en kant markeras som kanter om de ej understiger ett gränsvärde T_2. Denna hysteres motverkar att kantlinjer blir streckade om deras intensitet ligger nära gränsvärdet och man har brus i bilden, en effekt som lätt fås om man som i Sobels metod endast använder ett gränsvärde.

Variationer av Cannys metod för kantdetektering har gjorts av bland andra Deriche[8], som 1987 presenterade en optimal metod med avseende på Cannys kriterier för kantdetektering, och som utnyttjar två rekursiva filter som filtrerar en bild från motsatta håll.

I figur 7.7 ses en exempelbild, en nyckelkontur som censurerats, och motsvarande kanter detekterade i MATLAB med Sobels respektive Cannys metod ses i figur 7.8.



Figur 7.7. Exempelbild i vilken kanter ska detekteras.



Figur 7.8. Sobels metod till vänster och Cannys metod till höger.

7.4 Vald metod

Kantdetektering är ett komplicerat problem där det mänskliga syncentrat är svårt att överträffa med en algoritm även om goda, biologiskt motiverade, resultat uppnåtts av Grigorescu med flera[14]. Resultat som ligger på den nivå som ses i figur 7.8 kan inte anses tillräckligt bra för att dessa traditionella metoder som testats ska användas. Kanterna som kunde detekteras ses bättre i orginalbilden som den är och då experimentet endast kräver kantdetektering i ett fåtal bilder kan detta genomföras manuellt i ett bildbehandlingssystem så som Adobe Photoshop.

En nackdel med att att inte detektera kanter automatiskt är att det är svårare att kompensera för fel manuellt. Systematiska fel vid automatisk kantdetektering kan man rätta genom att ändra detaljer i implementationen, medan man i det manuella fallet istället tvingas lära sig att placera konturen rätt genom övning.

Kapitel 8

Experimentbeskrivning

Det här kapitlet beskriver hur det praktiska experimentet, som beskrivs i korthet i avsnitt 1.4, utförs med de metoder som valts i kapitel 3 till 7. En schematisk överblick av experimentutförandet ses i figur 8.1 och avsnitt 8.1 till 8.7 behandlar detaljer om experimentets olika delar. Ytterligare detaljer om experimentutrustningen hittas i bilaga A och detaljer kring hur denna användes finns i bilaga B.

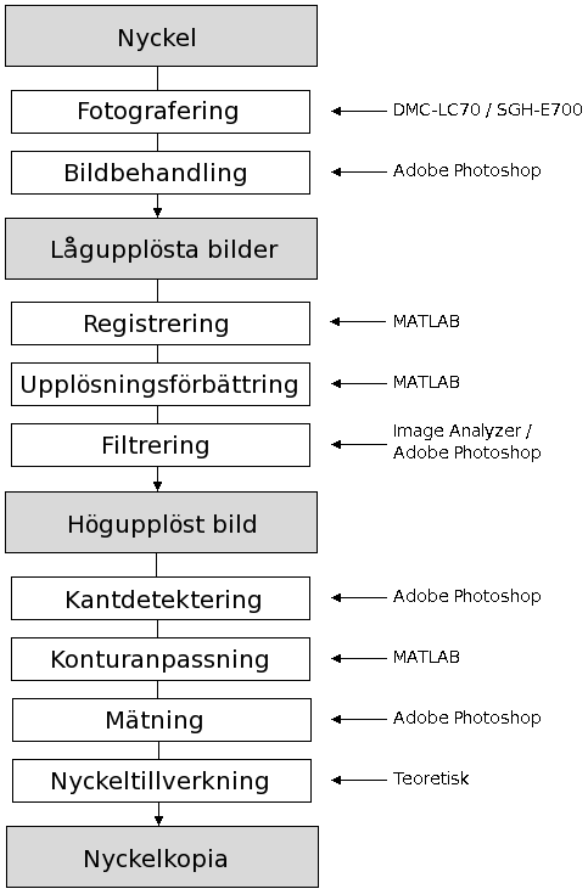
Den låstyp som experimentet avser är stiftcylinderlås av den enkla typ som beskrivs i avsnitt 2.1. Detta motiveras av att just denna låstyp är vanligt förekommande och används exempelvis för vanliga dörrlås i bostadsytterdörrar.

8.1 Fotografering

Fotograferingen syftade till att undersöka på vilket avstånd man kan ta tillräckligt bra bilder av en nyckel för att denna ska kunna kopieras med hjälp av bilderna, enligt två givna scenarion. I det ena scenariot fotograferas nyckeln när någon låser upp en dörr och i det andra när nycklar ligger på ett bord.

Scenario A

Då fotograferingen ska ske när ett lås låses upp är det enklast att placera kameran i höjd med låset med fokus på nyckelhålet enligt figur 8.2. Precis innan nyckeln förs in i låskanalen är den positionerad vågrät och vinkelrät mot kameran, och det är då fotograferingen sker. Kameran som används är en enkel digitalkamera som beskrivs i bilaga A. Detaljer om använda inställningar finns i bilaga B. Scenario A approximeras här genom att fotografera en nyckel monterad mot lämplig bakgrund med kameran på stativ.

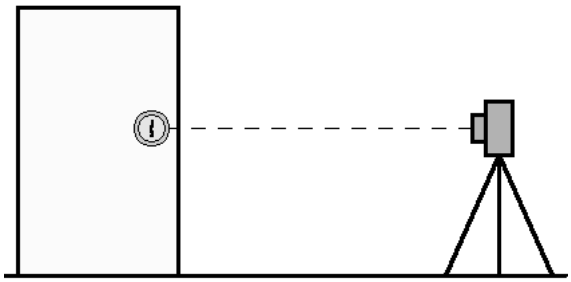


Figur 8.1. En överblick av processen att kopiera en nyckel.

Tabell 8.1 visar vilka försök som utfördes enligt scenario A.

Försök nr	Avstånd (m)
1	1
2	2
3	3
4	4
5	6.5

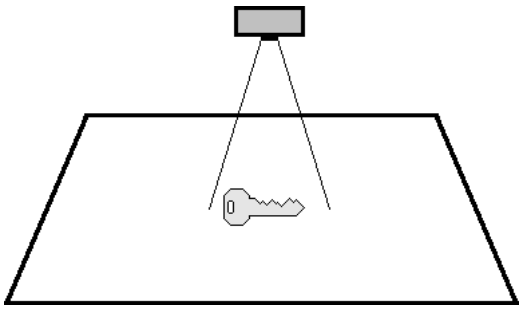
Tabell 8.1. Försök med digitalkamera.



Figur 8.2. Kameruppställning, scenario A.

Scenario B

I scenariot när nycklar ligger på ett bord kan man ta bilder på nära håll, vilket gör att resultatet är intressantare om en mobiltelefonkamera används. I figur 8.3 ses hur fotograferingen utfördes. Den valda kameramodellen beskrivs i bilaga A. Detaljer om använda inställningar finns i bilaga B.



Figur 8.3. Kameruppställning, scenario B.

Tabell 8.2 visar vilka försök som utfördes enligt scenario B.

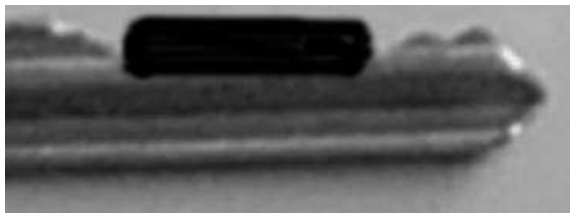
Försök nr	Avstånd (cm)
1	10
2	20
3	30
4	50
5	100

Tabell 8.2. Försök med mobiltelefonkamera.

I både scenario A och B användes relativt enkla kameror som kan köpas billigt i vilken elektronikaffär som helst. Ett lyckat resultat i experimentet betyder därför att kamerorna inte begränsar vem som kan utföra den föreslagna attacken mot mekaniska lås. I båda scenariona gäller även relativt ideala förhållanden. I praktiken måste man räkna med att ljusförhållanden och kamerans placering inte kan väljas fritt. Det är exempelvis inte troligt att den som låser upp dörren skulle låta sig fotograferas helt öppet på nära håll, eller så skulle det i alla fall väcka hans misstankar. Nycklarna på ett bord kan i viss mån anses vara mer realistiskt men om säkerheten är viktig lämnas antagligen inte nycklarna obevakade.

Samtliga bildserier är tagna med kamerornas multitagningsfunktion, vilket finns i de flesta digitalkameror. För både digitalkameran och mobiltelefonkameran har bildserier om 9 bilder använts. En knapptryckning resulterar i att en serie bilder tas och detta kan motverka att kameran skakar vid fotograferingen, speciellt i fallet med den monterade digitalkameran. För mobiltelefonkameran gäller det att hålla den så stilla som möjligt under fotograferingen. Då nyckeln inte rör sig är det kamerarörelserna som ger upphov till de förskjutningar som utnyttjas för upplösningsförbättringen.

Två exempel på bilder tagna med den monterade kameran respektive mobiltelefonkameran kan ses i figur 8.4 och 8.5. Som synes är de fotograferade objekten inte alltid placerade vinkelrät mot fotografiets kant, något som inte är optimalt för senare delar av experimentet. Figur 8.4 är inzoomad och beskuren för att visa just detta. Den lilla rotationen ger en mer realistisk approximation av scenariot, även om transformationer mellan bilder i en bildserie fortfarande inte är annat än förskjutningar i x och y-led. Bilderna måste roteras en aning för att nyckelkonturen ska kunna anpassas, men det lämnas till avsnitt 8.4 som behandlar detta.



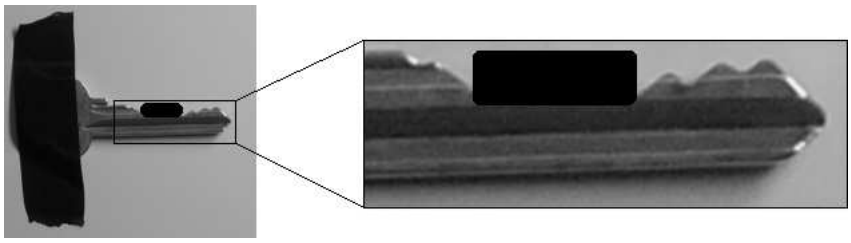
Figur 8.4. Fotografi enligt scenario A.

8.2 Bildbehandling

Innan bildregistreringen utförs bör man beskära det bildmaterial man har så att endast intressanta delar av bilderna finns kvar. Detta utförs exempelvis i Adobe Photoshop. Genom att beskära bilderna minskar registreringsprogrammets exekveringstid avsevärt, samt underlättas en lyckad registrering eftersom en större del av bilderna utgörs av de intressanta objekten.

Figur 8.5. Fotografi enligt scenario B.

Varje bild beskars likt exemplet i figur 8.6. Observera att alla beskurna bilder måste ha samma upplösning i den implementation som skrivits.



Figur 8.6. Exempel på hur en bilds ointressanta delar beskärs.

8.3 Registrering och upplösningshöjning

När man skapar en bild utifrån två bilder som är förskjutna i förhållande till varandra är det naturligt att man i kanterna på den skapade bilden inte har pixelvärden att använda. De överlappande delarna av de lågupplösta bilderna täcker ju inte lika mycket av den fotograferade scenen som de lågupplösta bilderna var för sig. Detta gör att om man vill använda exempelvis två genererade högupplösta bilder, som skapats av lågupplösta bilder som var mycket förskjutna, för att skapa en bild med ytterligare höjd upplösning bör man beskära kanterna innan en ny registrering görs, något man behöver tänka på endast om man utför upplösningshöjningen i flera steg. Registrering av redan upplösningsförbättrade bilder görs på detta sätt av Koo med flera[19]. Alternativet är att registrera bilderna med högre noggrannhet och sedan höja upplösningen så mycket man vill i ett enda steg.

Den implementerade registreringen talar för det sistnämnda, då den vid stegvis upplösningshöjning och små bildserier skulle utföra fler registreringar utan att för den skull prestera ett bättre resultat.

Med en implementation som istället registrerar alla bilder mot en referensbild och grupperar dem optimalt efter hur de är förskjutna inbördes, skulle stegvis upplösningshöjning vara interssant. Större bildserier skulle inte betyda att stegvis upplösningshöjning är bättre trots att registreringen i varje steg kan göras med lägre noggrannhet.

Resonemanget ovan räcker som motivation för att registreringen ska utföras i ett steg. Exempel på körskript för en bildserie, som upplösningshöjs i flera steg, finns i bilaga C.1, och exempel på resultat från registreringar av bildserier finns i bilaga D.

8.4 Filtrering av högupplöst bild

Man har nu skapat en högupplöst bild som innehåller information från samtliga observerade lågupplösta bilder. Kvar att göra är att höja dess skärpa då upplösningshöjningen gjorde bilden suddig. För detta användes wienerfilter som inte bara ökar skärpan utan också reducerar brus. I vissa fall blev resultatet bättre med en enkel höjning av skärpan, och i dessa fall wienerfiltrerades inte bilderna.

Programmet Image Analyzer valdes för filtreringen eftersom det är gratis och relativt lättanvänt, vilket gör det till ett bra alternativ till dyrare system. Detaljerad information om vilka inställningar som användes finns att läsa i bilaga B.

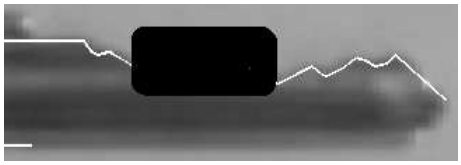
I de fall där Image Analyzer inte gav ett gott resultat, användes istället enklare funktioner i Adobe Photoshop för att höja skärpan.

Om man har en bild där nyckeln inte ligger vågrät i bilden måste den roteras för att nästa del av experimentprocessen ska kunna genomföras. Detta hör egentligen inte till filtreringen men eftersom det måste göras i anslutning till denna nämns det här. Bildrotationer kan enkelt utföras i Adobe Photoshop.

8.5 Kantdetektering och konturoptimering

Att automatiskt beräkna konturen av nyckeln, vilket kan göras med metoder liknande de som beskrivs i kapitel 7, visade sig vara svårt att genomföra i praktiken. Dåliga bilder och bilder där bakgrunden ibland inte skiljer sig mycket från objektet, vars kanter man vill bestämma, gör att parametrar till de enklare kantdetekteringsalgoritmerna inte kan bestämmas. I de fall där man lyckas någorlunda är det även svårt att hitta parametervärden som kan användas över hela bilder. Som nämndes i kapitel 7 är det komplicerat att filtrera ut kanter som tillhör ett visst objekt medan man med det mänskliga ögat gör det utan att tänka på det. Även om fördelaktiga bakgrunder valdes under försöket uppnåddes det bästa resultatet om kanterna ritades i manuellt i Adobe Photoshop.

För optimeringen av nyckelkonturen kräver implementationen att konturen ritas i med vitt enligt exemplet i figur 8.7. Mellan en topp och en dal i nyckelkonturen måste den inritade konturen vara strängt avtagande. Motsatsen gäller mellan en dal och en topp.



Figur 8.7. Exempel på hur en kontur ritas i en bild.

Den nyckelkontur som nu tagits fram tar endast hänsyn till hur bilderna, eller snarare den högupplösta bilden, av nyckeln ser ut men då bildkvaliten kan vara dålig gäller det att även anpassa nyckelkonturen till de regler som finns för hur nycklar till den låstyp man behandlar ser ut. Detta är information som till viss del beskrivs i avsnitt 2.2.

I experimentet valdes att även ta hänsyn till antagandet att intilliggande nedskärningar i nyckeln är placerade på lika avstånd från varandra, samt att alla nedskärningar är gjorda med samma verktyg. Det vill säga att de har samma form. Nedskärningarna antas även vara 7 till antalet. Detta är antaganden som gäller för den undersökta låstypen men som inte kan antas gälla för alla stiftcylinderlås.

8.6 Mätning

I det ideala fallet kan man utifrån den högupplösta bilden av nyckeln exakt bestämma alla dess mått. Detta är inte vad som gäller i praktiken utan det gäller istället att hitta tillräckligt bra värden. Vad som är tillräckligt bra bestäms av antalet nedskärningsnivåer vilket beskrivs i avsnitt 2.1.

Den högupplösta bildens skala kan beräknas givet kamerainställningar och avstånd mellan kamera och nyckel vid fotograferingen, men detta kräver väldigt exakta mätningar när försöket utförs, och minskar likheten mellan experimentet och ett tänkbart verkligt scenario i onödan. Enklare är istället att betrakta en nyckel för samma typ av lås, vilket man kan anta att man har tillgång till då det rör sig om en väldigt vanlig låstyp, och mäta avståndet mellan två nedskärningar i den. Med detta värde, som också gäller för avståndet mellan nedskärningar i den nyckel som ska kopieras, bestäms skalan så man vet hur många pixlar som motsvarar exempelvis en millimeter. Lämplig måttenhet avgörs av vilken noggrannhet som krävs.

Om man inte anser sig kunna bestämma tillräckligt bra värden för att kunna skapa en fungerande kopia bör man istället betänka hur liten man kan göra den nyckelmängd i vilken minst en nyckel öppnar låset.

Givet information om minsta möjliga skillnader i nedskärningsdjup kan man kanske säga att en viss nedskärning är någon av exempelvis tre olika förslag. En sådan osäkerhet för endast en nedskärning innebär att en av tre föreslagna nycklar öppnar låset, något som man säkert kan acceptera i praktiken. Antalet möjliga nycklar ökar dock snabbt om fler nedskärningar är osäkra. Låt antalet troliga nedskärningsdjup för en nyckel med i spärrstift vara a_i. Detta ger antalet möjliga nycklar, A, enligt följande formel.

$$A = \prod_i a_i$$

För en nyckel med 7 spärrstift och tre möjliga nedskärningsdjup i varje position har man alltså $3^7 = 2187$ möjliga nycklar. Även om reglerna i avsnitt 2.2 antagligen kan minsta antalet ytterligare så är detta för många för att man ska kunna säga att man har tillräckligt bra resultat för att kopiera nyckeln.

8.7 Nyckeltillverkning

För att jämföra försökens resultat bör det definieras vad som är ett lyckat genomfört experiment. Följande fråga har besvarats för samtliga genererade nyckelkonturer:

> *Om de N mest troliga nycklarna tillverkas, kommer man då att tillverka den korrekta nyckeln?*

När N ökas undersöks alltså vilken mätning som är mest osäker, det vill säga vilken som ligger närmst en annan möjlig nedskärningsnivå än den man bestämt sig för tidigare. Ett exempel belyser enklast resonemanget. Antag att de möjliga nivåerna är 1.0 mm, 2.0 mm, 3.0 mm, 7.0 mm och 9.0 mm. Mätningar på den genererade nyckelkonturen visar 1.2 mm, 1.9 mm, 3.0 mm och 5.9 mm. Då dessa värden avrundas mot sitt närmsta möjliga nedskärningsdjup fås nyckeln 1234 (notation enligt avsnitt 2.3). Den mest osäkra mätningen är den första eftersom den ligger 0.8 mm från nivå 2. Mätning 2 har motsvarande osäkerhetstal 0.9 mm. Den säkraste mätningen är den fjärde eftersom dess osäkerhetstal är 2.9 mm, vilket är mätningens avstånd till nivå 3. Observera dock att den fjärde mätningen är den med störst absolut fel, men beroende på vilka möjliga nedskärningsnivåer den aktuella låstypen har, behöver detta inte automatiskt innebära att mätningen är den mest osäkra.

Hur stort N man kan ha och fortfarande anse experimentet lyckat beror på experimentutrustningen i övrigt. Har man väldigt billig experimentutrustning är det inte rimligt att spendera mycket arbete och pengar på själva nyckeltillverkningen.

Med de enkla kameror som använts i det genomförda experimentet är det endast rimligt att tillverka ett fåtal nycklar, och därför definieras ett lyckat experiment här som ett positivt svar på den tidigare ställda frågan med $N = 10$.

Hur nyckeltillverkningen går till rent praktiskt är inte speciellt intressant. Givet de möjliga nedskärningsnivåerna, typ av nyckelämne och vilka nedskärningsdjup experimentet resulterar i, kan en låssmed utföra tillverkningen.

Kapitel 9

Resultat

Detta kapitel beskriver resultatet av de experiment som utförts. Delar av de data som producerades under försöken finns i bilaga D; bilder, karaktäristiska kurvor, registreringsresultat och skillnader mellan data före och efter bildbehandlingen. Det sistnämnda i form av ett exempel där samma nyckeldetalj i både låg och hög upplösning, samt med den anpassade nyckelkonturen inritad, presenteras. Detta exempel visar att mängden detaljer som kan extraheras ur bilden ökar genom bildbehandlingsprocessen, och motiverar alltså de använda metoderna.

Under fotograferingen togs för varje avstånd en serie på 9 bilder som sedan förbättrades enligt kapitel 8. Nedskärningsdjup mättes och analyserades enligt avsnitt 8.7. Liksom i avsnitt 8.7 betecknar också här N det antal nycklar som måste tillverkas för att den korrekta nyckeln ska finnas bland dessa.

9.1 Digitalkamera

Resultatet från experimentet med digitalkamera på långt avstånd (scenario A i kapitel 8) finns beskrivet i tabell 9.1 och 9.2. För varje nedskärning i nyckeln anges det i tabell 9.1 om den mättes korrekt ($\surd$), osäkert (?) eller fel ($\times$).

Observera skillnaden i resultat på avståndet 4 meter för en upplösningsförbättrad bild och en bild som endast uppsamplats till den höga upplösningen (markerad med * i tabell 9.1 och 9.2). Den upplösningsförbättrade bilden ger ett lyckat resultat enligt definitionen i avsnitt 8.7 medan den uppsamplade bilden inte gör det. Detta motiverar hela bildbehandlingsprocessen.

Observera även skillnaden i resultat på avståndet 3 meter för en konturanpassad bild och en bild där konturen endast ritats för hand (markerad med ** i tabell 9.1 och 9.2). Denna skillnad motiverar väl optimeringen av nyckelkonturen, som görs för att få konturen att uppfylla vissa krav, specifika för det aktuella låssystemet.

Tabell 9.2 visar dels om rätt nyckel skulle tillverkas då endast en nyckel tillverkas, men också ett mer realistiskt lägsta värde på N för de olika bildserierna. Anledningen till att detta görs är de osäkert uppmätta nedskärningsdjupen som man i en praktisk situation inte skulle lita på, även om de i ett specifikt fall kan avrundas rätt.

Avstånd	Nedskärning						
(m)	1	2	3	4	5	6	7
1	√	√	√	√	√	√	√
2	√	√	√	√	√	?	√
3	√	√	√	√	?	?	√
3**	√	×	×	?	?	×	√
4	?	√	√	√	?	×	√
4*	×	?	√	√	√	×	?
6.5	?	?	√	√	?	?	?

Tabell 9.1. Resultat av experiment med digitalkamera.

Avstånd (m)	$N = 1$	N
1	√	1
2	√	2
3	×	4
3**	×	288
4	×	8
4*	×	16
6.5	×	32

Tabell 9.2. Antal tillverkade nycklar efter experiment med digitalkamera.

9.2 Mobiltelefonkamera

Resultatet från experimentet med mobiltelefonkamera (scenario B i kapitel 8) finns beskrivet i tabell 9.3 och 9.4. Det presenteras på samma sätt som resultatet för digitalkamera i avsnitt 9.1.

Enligt definitionen av ett lyckat experiment i avsnitt 8.7 lyckas de utförda experimenten för ett avstånd på upp till 50 centimeter med den mobiltelefonkamera som använts. För bildserien tagen på 100 centimeters avstånd kunde inga detaljer utläsas ur bilden och försöket ansågs misslyckat.

Avstånd	Nedskärning						
(cm)	1	2	3	4	5	6	7
10	√	√	√	√	√	√	√
20	√	?	√	√	√	√	√
30	√	√	√	√	√	?	√
50	√	√	√	√	√	×	√
100	×	×	×	×	×	×	×

Tabell 9.3. Resultat av experiment med mobiltelefonkamera.

Avstånd (cm)	$N = 1$	N
10	√	1
20	√	2
30	√	2
50	×	4
100	×	×

Tabell 9.4. Antal tillverkade nycklar efter experiment med mobilkamera.

Kapitel 10

Diskussion och analys

Detta kapitel diskuterar och analyserar de metoder som använts under bildbehandlingsprocessen samt hur dessa ökade mängden detaljer som kunde läsas ur bilderna. Intressanta delar av de resultat som experimenten gav belyses och en kort diskussion kring låssäkerhet finns i avsnitt 10.3.

10.1 Experimentmetod

Eftersom experimentmetoden som användes inkluderar så många olika steg, och en del utrustning, kan metoden förbättras eller varieras på ett flertal sätt. Vissa steg och viss utrustning kan vara viktigare än andra att välja så optimalt som möjligt.

10.1.1 Fotografering

Fotograferingen av objektet som ska återskapas är kanske den viktigaste processen i återskapandet. Utan bra bilder blir det mycket svårt att utvinna tillräckligt mycket information för att kunna återskapa objektet. I fallet då nycklar ska återskapas måste nyckelns ungefärliga skärning tydligt framgå även innan man börjar med bildförbättringen. Även om de presenterade metoderna baseras på små förskjutningar mellan de fotograferade objekten bör kameran, vid försök liknande de i denna rapport, hållas så stilla som möjligt. Tillräckliga förskjutningar uppkommer ändå, och bilden blir skarp, vilket är en av de saker som är viktigast att prioritera under fotograferingen. Stativ eller andra anordningar för att minska vibrationer föredras.

Kamera

Den kamera som användes vid experimentet var relativt billig och lämpar sig bättre för att ta semesterbilder än att fånga små detaljer på avstånd. Den ansågs dock ge en god uppskattning av hur bra experimentet kan genomföras av en person utan tillgång till avancerad utrustning. För att få så bra bilder som möjligt är det självklart att en digital systemkamera med teleobjektiv borde användas. Med en sådan kamera skulle tillräckligt bra bilder kunna tas från mycket längre avstånd.

Vid det andra experimentet, med en mobiltelefonkamera, höll den använda kameran mycket låg standard. Utvecklingen inom detta område har gått snabbt framåt och de kameror som sitter i de mobiltelefoner som säljs idag håller en mycket högre klass. Kameran som användes i experimentet hade en upplösning på 0.3 mega-pixels, men de senaste protyperna av mobiltelefonkameror kan klara av ända upp till 7 mega-pixels. Därför kan resultatet, som uppnåddes vid mycket goda förhållanden, ses som representativt för en modern mobiltelefonkamera vid normala eller rent av dåliga förhållanden. Information om den utrustning som användes i experimentet kan hittas i bilaga A. Man kan även låta mobiltelefonkameran representera en mindre kamera, som kan gömmas intill objektet som ska fotograferas, och som inte kan prestera bilder av högre kvalitet.

Yttre omständigheter

För att kunna ta en så bra bild som möjligt av föremålet är det viktigt att de yttre omständigheterna också är så goda som möjligt. God belysning kan vara svårt att uppnå, speciellt om fotograferingen sker utomhus. Att det är ljust är viktigt då bilderna, om man inte har en mycket bra kamera, ofta blir lite suddigare när den fotograferade scenen är mörk. Är det däremot för ljust är det lätt att det blir reflexer och blänk i det fotograferade objektet, vilket gör att det kan vara mycket svårt att se exempelvis en nyckels skärning. Felaktigt ljus kan också göra att man har en skugga precis bakom objektet på bilden. Då det inte är säkert att skuggan rör sig exakt likformigt med objektet i bilderna kan detta störa registreringen. Att ta en så snabb bildserie att varken objekt eller skuggor hinner förflytta sig löser till viss del problemet men givetvis är inte detta alltid möjligt. Enklare är att lösa problemet genom att beskära bildmaterialet, och eliminera ointressanta delar av bilderna.

Ett problem som kan uppkomma är att personen som ska låsa upp låset blockerar sikten till nyckeln på ett sådant sätt att det inte går att ta bilder av nyckeln rakt från sidan. Om så är fallet kanske det är möjligt att ta bilderna från en annan vinkel. För att kunna mäta i dessa bilder kan perspektivtransformationer vara nödvändiga. Experiment där vinkeln mellan det fotograferade objektet och kameran måste ändras har här inte utförts men behandlas av Mann och Picard[22].

Även objektets bakgrund har betydelse för hur bra bildkvaliteten hos en bildserie kan anses. Om bakgrunden är för framträdande kan den störa registreringen av objektet i bilderna, då objektet ofta rör sig i förhållande till bakgrunden.

Om så är fallet kan man via enkel bildbehandling neutralisera bakgrunden först, genom att man till exempel suddar ut den i ett bildbehandlingssystem. Innan bildserien kan genomgå registrering måste man också tänka på att bakgrunden bör vara lika i alla bilder. Detta är inte helt enkelt att uppnå men det viktiga är att det intressanta objektet som ska undersökas dominerar bilderna under registreringen. Om bakgrunden kring objektets kanter skiftar spelar mindre roll. I praktiken kan det vara svårt att påverka hur bakgrunden ser ut då man inte kan sätta upp skärmar hur som helst vid det lås som nyckeln ska kopieras vid, utan att personen som innehar nyckeln blir misstänksam. Om man har möjlighet att påverka hur bakgrunden ser ut är en enfärgad bakgrund att föredra då det fotograferade objektet i det fallet kan registreras med avseende endast på sig själv i referensbilden.

10.1.2 Bildförbättring

Registrering

Registreringen är ett område där det finns många möjligheter att välja mellan olika metoder. Den variant av korskorrelation som valdes fungerade bra då bilderna var av hög kvalitet och det antogs att den enda rörelse i bilderna som förekom var en förskjutning av nyckeln i x- och y-led. I verkligheten kan man tänka sig att det skulle kunna finnas flera transformationer, till exempel en rotation under tiden nyckeln förs mot låset. Vi antog att rotationen var så pass liten att den kunde försummas. Om rotationen skulle visa sig vara betydlig kan man innan man kör registreringen förbereda bilderna så att nyckeln ligger helt vågrätt i samtliga bilder, genom att rotera dem i ett bildbehandlingsprogram. I det utförda experimentet var detta dock inte något problem. Det finns även metoder som tillåter att man registrerar både förskjutningar och rotationer samtidigt. Ett par av dessa metoder beskrivs av Brown[5] och Mann[22].

Om bilderna hade en varierande bakgrund neutraliserades den, enligt avsnitt 10.1.1. En annan sak som också löser problemet med för dominerande bakgrunder är att beskära bilderna tillräckligt innan registreringen påbörjas.

I de tester som gjordes under implementeringen av registreringsmetoden visade det sig att exakt rätt förkjutning kunde registreras då bilderna var konstruerade helt utan additivt brus. Detta verifierade algoritmen.

För bilder med mycket störningar och låg upplösning, typiskt fotografier tagna på långt avstånd, var misslyckade registreringar vanligare förekommande. Detta visade sig genom registreringsresultat där förskjutningarna av de tre färgkanalerna kunde skilja sig mycket från varandra . Det är möjligt att andra metoder hade gett bättre registreringsresultat. Men eftersom denna typ av fel förekom när bilderna var av så låg kvalitet att några betydande detaljer inte bedömdes möjliga att avläsa, sågs det inte som något stort problem. Typiskt gällde det i situationer då man inte kunnat gissa att en av de lågupplösta bilderna föreställde en nyckel.

I de fall där registreringen inte kan anses misslyckad förekom också små skillnader i förskjutningarna mellan de tre färgkanalerna. Detta ses i bilaga D, där detaljerade registreringsresultat presenteras. Fenomenet diskuteras ytterligare under upplösningsförbättring i detta kapitel.

Om andra bildregistreringsmetoder fungerar lika bra för den enkla form av sub-pixelregistrering av förskjutning som utfördes under experimentet kan man istället betrakta andra aspekter av metodens kvalitet, så som svårigheter vid implementation samt exekveringstid. Den metod som implementerades är exempelvis lätt att följa även utan goda kunskaper i signalteori och det kan vara en fördel då vi i rapporten även vänder oss till personer i låsbranchen, där dessa kunskaper är relativt ovanliga. Å andra sidan kan man utan dessa kunskaper tillgodogöra sig idén om sub-pixelförskjutningar och behöver inte förstå exakt hur registreringen utförs.

Exekveringstiden för implementationen av registreringen ökar linjärt med noggrannheten multiplicerat med antalet lågupplösta pixlar som ska behandlas. Noggrannheten avser här antalet minsta förskjutningar som ryms i en pixel. Detta innebär att så små bilder som möjligt är att föredra, vilket är en av anledningarna till att man beskär dem innan registreringen. Att ställa in en mindre maximal förskjutning i körskriptet minskar också exekveringstiden vilket gör att bildmaterialet bör beskäras så hårt som det är möjligt utan att eliminera intressanta regioner i bilderna.

En sak man skulle kunna ifrågasätta i den algoritm som Koo[19] presenterar, och som också gäller i den implementerade registreringen, är hur skillnaden mellan de karaktäristiska kurvorna minimeras. En minsta kvadratlösning känns mer intuitiv, men gav efter tester inte ett bättre utan precis samma resultat som de summerade skillnadsbeloppen. Specialfall kan konstrueras där minsta kvadratlösningar presterar bättre, men med fotografier lika de som använts under experimentet kan metoden, som Koo[19] presenterar, tillämpas.

Upplösningsförbättring

Förbättringar finns att göra när det gäller upplösningsförbättringsmetoden. En sak man skulle kunna förbättra är valet av vilka lågupplösta bilder som används. Exempelvis vill man för en fördubbling av upplösningen (antalet pixlar ökar då fyra gånger) ha tillgång till fyra bilder tagna med sådan relativ förskjutning på subpixelnivå att de helt kan återskapa en suddig version av den högupplösta bilden, enligt Peleg[26].

Vad man istället har tillgång till är ofta ett större eller ibland mindre antal bilder men med icke optimala förskjutningar. Genom att välja bilder efter hur deras förskjutningar ser ut kan man minska programmets exekveringstid utan att försämra resultatet, och då det gäller medelvärdesbildning kan det även förbättra resultatet att inte ha för många bilder, eftersom alla bilder ger bidrag till den högupplösta bilden. I praktiken är det dock svårt att minska programmets totala exekveringstid genom detta resonemang, då det rör sig om relativt små bildserier.

Kvalitetsaspekten bör istället vara motiveringen till att bilder plockas bort från bildserierna.

Som implementationen ser ut är förskjutningarna mellan de lågupplösta bilderna som beräknades under registreringen givna för röd, grön och blå färgkanal. Bilder med 24 bitars färgdjup förutsätts genom hela experimentet. Dessa färgkanaler har visat sig kunna skilja sig åt för samma bild, vilket kan tyckas märkligt. Tre lösningar på problemet är uppenbara, och de diskuteras här eftersom motsvarande del av implementationen återfinns i anknytning till upplösningshöjningen i programkoden.

Man kan konvertera alla bilder till gråskala så man endast har ett värde för varje pixel att ta hänsyn till, alternativt slå samman färgkanalerna för att inte förlora färgdjup i processen. Man kan välja en av färgkanalerna och alltid betrakta förskjutningen av denna som bildens förskjutning. Den tredje, vilken också är den lösning som gäller för den implementerade upplösningsförbättringen, är att beräkna medianen av färgkanalernas förskjutning och låta detta värde gälla. Fördelen med detta är att när man vid vissa registreringar, för dåliga bilder som ger ett felaktigt värde för en enda av färgkanalerna, kan utnyttja att medianen ger ett av de två korrekta värdena som resultat. När registreringen ger lika värden, eller relativt lika beroende på vald noggrannhet, spelar det mindre roll vilket av dem man väljer. En nackdel kan vara att man för olika bilder kan förkasta olika information och på så vis eventuellt påverka upplösningshöjningen negativt. Detta ses inte som något problem eftersom man som regel får relativt lika förskjutningar av samtliga färgkanaler.

De detaljerade registreringsresultaten som presenteras i bilaga D talar för att vissa bilder inte bör ingå i upplösningshöjningen eftersom deras färgkanaler är olika förskjutna i förhållande till referensbilden. Detta löses enkelt genom att man exekverar körskriptet två gånger, och den andra gången har man plockat bort de bilder ur bildserien som gav dåliga registreringar vid första exekveringen. Givetvis skulle man även kunna detektera dessa automatiskt efter registreringen men det görs inte i experimentimplementationen.

10.1.3 Filtrering av högupplöst bild

Wienerfilter valdes för att de effektivt kombinerar borttagande av oskärpa med brusreducering. Vikten av att få bort det additiva bruset i bilderna kan diskuteras då det ofta inte avgör hur väl man kan utföra mätningar. Samtidigt kan man argumentera att all kvalitetshöjning är eller kan vara till nytta och bör ske om möjligt. Det är också svårt att höja skärpan utan att förstärka bruset så man bör ta hänsyn även till detta. När det gäller tillgängliga implementationer av wienerfilter är det ofta svårt att koppla valda parametrar till parametrar man är van vid från filterteori. Fokus ligger ofta istället på att man enkelt, och utan filterkunskaper, ska kunna testa sig fram till ett gott resultat.

I vissa situationer visade sig ett enklare filter ge ett bättre resultat än wienerfilter, och när så var fallet användes även andra filter. Detta finns beskrivet i detalj i bilaga B.

10.1.4 Extraktion av information

Den intressanta informationen i den högupplösta bilden är i det utförda experimentet konturen av nyckeln. Den ges av dels den information man direkt kan få ur bilden, nämnligen var nyckelns kanter finns, men också av den information man har om den aktuella nyckeltypen. Bilden man betraktar är inte perfekt så att bara ta hänsyn till den behöver inte ge det bästa resultatet, och gör det heller inte, visas i kapitel 9.

Kantdetektering

Då arbetet startade avsågs att använda en automatiserad metod för att ta fram konturen av nyckeln som fotograferades. Detta visade sig vara svårare än väntat och ingen av de metoder som testades gav tillräckligt goda resultat. Manuell utritning av konturer gav istället ett gott resultat. Att automatisera processen helt, även om man skulle hitta en metod som på ett bra sätt detekterade kanter, skulle inte bli trivialt då inte alla kanter är intressanta. Ett problem som återstår att lösa är att, givet alla kanter, bestämma vilka kanter som motsvarar vilka i en optimal nyckelmodell.

Eftersom ljus och bakgrund ofta skiljer sig mellan varje serie är det svårt att hitta parametervärden som ger ett gott resultat på alla möjliga bilder. Bara att hitta värden som fungerade över en hel bild visade sig vara mycket svårt. Kanterna är ofta olika tydliga i olika regioner i bilden. Exempel på automatisk kantdetektering med traditionella metoder finns i avsnitt 7.3.

Ett alternativ till helt manuell kantdetektering, som kan vara interssant att testa, är att kombinera Cannys metod med en manuell anpassning av den genererade konturen. Målet med detta skulle vara att minska de systematiska felen genom automatiseringen, men samtidigt slippa de otillräckliga kanterna som Cannys metod presterar. En klar nackdel är att det blir ganska mycket arbete för varje bild som ska kantdetekterats, och detta har inte testats under de utförda experimenten.

Konturberäkning

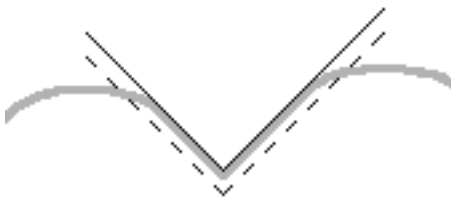
När alla kanter ritats in enligt föregående avsnitt optimeras så nyckelkonturen med avseende på den information man har om nyckeln. I implementationen förutsätts vissa saker som till exempel att alla nycklar har 7 nedskärningar, att två närliggande nedskärningar alltid har samma avstånd mellan sig, och att samtliga dessa görs med samma verktyg, vilket är en V-formad fil med ett visst utseende.

En förbättring av implementationen vore att ge all nyckelinformation på parameterform för att på så vis förenkla konturberäkningar för fler nyckeltyper. Experimentet avser dock endast att visa att en kontur kan beräknas för en viss nyckel så allt detta ansågs överflödigt.

Under optimeringsprocessen varieras nyckelkonturens längd, dess position, och nedskärningarnas djup var för sig. Med att konturens position varieras menas att den skjuts i sidled. Att optimeringen av nyckelkonturen förbättrar resultatet råder det inga tvivel om. Det ses i kapitel 9. Visserligen kan man ta hänsyn till vissa egenskaper hos nycklar då konturen ritas för hand, men att automatisera anpassningen av konturen efter givna regler bedöms som mycket viktigt.

Ytterligare information man kan ta hänsyn till vid nyckelkonturanpassningen är de olika nedskärningsnivåer som man känner till för den aktuella låstypen. Den implementation som gjorts tar inte hänsyn till detta, men givet nivåerna kan man enkelt avrunda dem åt rätt håll manuellt. När man gör det är det också naturligt att avgöra om man säkert kan bestämma nedskärningsnivån eller om man måste låta den vara obestämd bland exempelvis två nivåer. Ofta, när man har sämre bilder att utgå ifrån, kan nyckelns utseende inte bestämmas exakt utan ett antal nyckelkonturer får ges som resultat, bland vilka exakt en är den rätta.

Ett problem som kan uppkomma när man ska generera konturen av nyckeln är att nycklar slits. Då man simulerar att det V-formade verktyget sänks mot nyckeln kan detta komma att hamna fel om nyckeln är för mycket sliten och man liksom i experimentimplementationen minimerar skillnaden mellan verktyget och konturen med minsta kvadratmetoden. Skillnaden mellan verktyget och nyckelkonturen beräknas mellan nyckelkonturens toppar och hur detta kan ge ett felaktigt nedskärningsdjup ses i figur 10.1. En minimering enligt minsta kvadratmetoden talar för att verktyget ska sänkas ner under det egentliga nedskärningsdjupet, om topparna är rundade.



Figur 10.1. Konturanpassning för slitna nycklar.

Om man istället börjar sin anpassning en bit in på varje nedskärning kan man komma runt problemet. Dock uppstår då ett problem vid de nedskärningar som har väldigt kort sträcka till en topp, så hur optimeringsområdet på verktyget ska väljas är inte självklart. För enkel konturanpassning krävs därför att nycklar som ska kopieras är i gott skick eller att man på något sätt kompenserar för slitaget i den kontur som tagits fram innan simuleringen av skärningen utförs, något som till viss del kan göras om den ritas in för hand.

Man skulle också kunna tänka sig en implementation där man för varje nedskärning manuellt anger var på verktyget optimeringen ska utföras.

Mätning i bilden

Vid mätning måste man ha mycket god kunskap om vilken typ av objekt det är man mäter, så man vet vad som är intressant att mäta. I fallet nyckelkopiering behövs information om den låstyp man ska kopiera en nyckel till. Utan beräkningar av mått med hjälp av avståndet mellan nyckeln och kameran kan det vara problematiskt att bestämma ett exakt mått i en bild, från vilket övriga mått kan beräknas. Vilken låstyp nyckeln ska öppna är avgörande för hur tätt nedskärningarna i nycklarna sitter, och då detta är konstant för en låstyp, är det viktigt att det mäts så exakt som möjligt. Avståndet mellan den första och den sista nedskärningen är alltså ett känt mått i nyckeln, som med fördel kan användas för beräkning av övriga mått, då det är relativt långt och därför minskar felet.

Om man inte har tillgång till professionell hjälp, av exempelvis en låssmed, är det extra viktigt att noggranna mätningar kan göras på nycklar till den aktuella låstypen. Detta för att så mycket information om den okända nyckeln som möjligt blir känd innan den fotograferas. Denna information är typiskt avstånd mellan nedskärningar och möjliga nedskärningsdjup. För det sistnämnda kan flera nycklar behöva undersökas om inte alla djup finns representerade i en nyckel, och skillnaden i nedskärningsdjup mellan olika nedskärningsnivåer skiljer sig. Alltså om exempelvis skillnaden mellan nivå 1 och nivå 2 är 0.2 mm men skillnaden mellan nivå 2 och nivå 3 är 0.3 mm. Den här typen av information anses vara given då försök utförs med hjälp av låssmed.

När denna typ av information är känd, oavsett hur man erhållit den, är det möjligheten att bestämma nedskärningsdjup som är intressant. När en mätning gjorts jämförs den med de möjliga nedskärningsnivåerna och om mätningen ger ett resultat som stämmer väl överens med endast en av dem anser man sig säker på sin nedskärning. Om man istället mäter en nedskärning i bilden som ligger mitt emellan två möjliga nedskärningsnivåer anser man sig säker om man väljer båda dessa. Man har dock ökat antalet nycklar som måste tillverkas för att man ska vara säker på att få upp låset. Observera att även om man, enligt denna definition, är säker på att den korrekta nyckeln finns bland dem man tycker sig behöva tillverka, måste så inte vara fallet. Man kan ha mätt fel, på grund av att bilden inte är tillräckligt detaljerad. Detta resonemang resulterar i den definition av ett lyckat utfört experiment som presenteras i avsnitt 8.7.

Försökens mätvärden presenteras här inte i detalj, då syftet med det utförda experimentet inte är att underlätta inbrott i bostadshus, försedda med den undersökta låstypen, utan endast att belysa brister i säkerheten hos mekaniska lås.

10.1.5 Nyckelåterskapning

För att lyckas med det genomförda experimentet, och för att göra det mer intressant, kontaktades ett antal låssmeder. Målet med detta var att få tillgång till all den information som kan användas utöver den som fås ur själva bilderna. Alltså detaljer kring nyckeltillverkning som allmänheten inte känner till. Denna verksamhet är av tradition en aning hemlighållen så till vida att många låssmeder ogärna talar om hur nycklar kopieras, tillverkas etc. Sveriges Låssmedsmästares Riksförbund, SLR, har i sina stadgar[36] ett antal etiska regler som gäller för deras medlemmar och som ämnar att upprätthålla allmänhetens förtroende för låsmästarkåren. Bland annat finns en regel om tystnadsplikt som säger följande:

> *Medlem eller anställd i Låsmästarföretag får ej för utomstående omtala förhållandet då det gäller utförda arbeten, låssystem, delningsuppgifter, låsöppningsmetoder m.m. Systemhandlingar ska förvaras på ett betryggande sätt då de inte är under uppsikt. Fakturor eller följesedlar får inte förses med exemplevis delningsuppgifter eller liknande noteringar.*

Man kan alltså förstå att vissa låssmeder inte vill befatta sig med ett projekt som ämnar att visa på låsens sårbarhet, då regeln ovan påbjuder att man inte avslöjar låsöppningsmetoder. Detta var också fallet. Ett antal av de tillfrågade låssmederna avböjde med motiveringen att projektet skadar dem mer än hjälper dem, medan andra tyckte iden var god men ville först diskutera experimentet med andra personer i branchen, så som låstillverkare.

Kanske kan det anses sundare för en säkerhetsbranch att man istället uppmuntrar ifrågasättande och okonventionella tester av de system som ska vara säkra. Detta borde utveckla branchen till både branchutövares och allmänhetens nytta.

10.2 Resultat

Enligt de resultat som presenteras i kapitel 9 lyckades de utförda experimenten med digitalkamera på upp till 4 meters avstånd och med mobiltelefonkamera på upp till 50 centimeters avstånd. Detta är värt att kommentera.

Om man tycker 4 meter är ett för litet avstånd för att den föreslagna attacken mot stiftcylinderlås ska kunna anses vara effektiv, bör man betänka vilken kamera som använts. En mer avancerad kamera kan ge mycket bättre bilder vilket gör att attacken kan utföras på längre avstånd. Observera också att även om mer avancerade kameror inte testats under det utförda experimentet så kan man, genom att studera bildkvaliteten hos sitt fotograferade material och jämföra med bildkvaliteten i de lyckade försöken, uppskatta om en annan kamera vid ett annat avstånd kan ge ett lyckat resultat. Detta resonemang gäller givetvis också för mobiltelefonkameror.

Resultatet visar också att de olika stegen i bildbehandlingsprocessen är motiverade, då de gör att den föreslagna attacken mot ett mekaniskt lås kan utföras på längre avstånd.

10.3 Säkerhet hos låssystem

Säkerheten hos ett låssystem med fysisk nyckel är beroende av att inte fel personer har tillgång till nyckeln. Ju mer information, som underlättar nyckelkopiering, som personer har desto sämre säkerhet måste man anse att låssystemet ger. Detta kan liknas vid ett lösenordssystem där en pressumtiv inkräktare exempelvis vet att lösenordet bara innehåller små bokstäver, är sju tecken långt, eller att lösenordets tredje tecken är A. Den här typen av information minskar antalet möjliga lösenord precis som information om en fysisk nyckel minskar antalet fall att undersöka vid exempelvis en brute force-attack. En sådan attack kanske inte är trolig då väldigt många möjliga nycklar återstår, men med mycket information om låset och nyckeln minskar antalet snabbt.

Att man genom att visa nyckeln läcker den här typen av information måste anses vara en svaghet hos systemet. Samma sak gäller för koder och lösenord som knappas in där andra kan se dem eller delar av dem.

Om man ska föreslå några alternativ till den här typen av system så är kanske det första man tänker på elektroniska lås med elektroniska nycklar. Informationen man får om innehållet på en magnetremsa enbart genom att titta på den är begränsad men samtidigt har enkla attacker mot den här typen av nycklar också genomförts, enligt Aftonbladet[15]. Likaså har fingeravtryckslås attackerats av Sandström[30]. Det intressanta är kanske om detta projekt visat på något man skulle kunna förbättra hos just mekaniska lås.

Noggrannheten med vilken nyckeln måste vara skuren för att kunna öppna det avsedda låset är en sak som direkt avgör hur svår den är att kopiera. Det är också direkt avgörande för priset på låset, vilket är en sak som skiljer sig från vad man annars kunde dra en parallell till; ökning av antalet nyckelbitar vid kryptering. Ett lås med låg tolerans för fel hos nyckeln gör att bättre bilder av den behövs för att den ska gå att kopiera, bättre verktyg behövs vid kopieringen, etc. Precis som datorns utveckling gör att längre och längre nycklar krävs vid kryptering, gör utvecklingen av digitala kameror att större exakthet vid nyckeltillverkning krävs för att ett lås ska motstå den föreslagna attacken.

Kapitel 11

Slutsats

Detta kapitel ger en slutsats om det utförda arbetet men också rekommendationer och spekulationer kring fortsatt arbete med kopiering av delvis kända objekt i allmänhet och nyckelkopiering i synnerhet.

11.1 Slutsats

Vi har i denna rapport visat ett antal metoder som kan användas för att extrahera detaljer ur bilder. Metoderna underlättar kopiering av tredimensionella objekt så som metallnycklar om man har tillräcklig ytterligare information om objekten, samt tvådimensionella bilder. Ett bedrägeri där denna typ av kopiering utnyttjas har tidigare utförts, vilket uppmärksammats av philly.com[31]. Metoder som beskrivs i denna rapport förenklar denna typ av bedrägerier, men självklart blir inte lås värdelösa bara för att kopieringen av dess nycklar, utan nyckelägarens tillåtelse, kan förenklas. Man bör dock tänka på att en nyckels utseende är något som i vissa fall bör hållas hemligt. Man går inte omkring och ropar ut sina lösenord och bankomatkoder och på samma sätt bör alla typer av nycklar behandlas diskret, om man vill kunna lita på de säkra system till vilka de används.

Den grad med vilken nyckelkopiering, enligt den föreslagna attacken mot lås, är möjlig beror mycket på den utrustning man har att tillgå. Givet en viss kamera är det dock enklare att kopiera nyckeln vid ett givet avstånd om de föreslagna bildbehandlingsstegen utförs. Detta visas genom experimentresultatet i kapitel 9. I bilaga D finns en bildjämförelse där samma detalj i en nyckelkontur ses lågupplöst, högupplöst, filtrerad, med kontur inritad samt med nyckelkonturen anpassad efter givna regler för nycklars utseende. Denna jämförelse visar också att bildbehandlingsprocessen gett resultat.

Resultatet är också mycket beroende av vilken låstyp det är som attackeras, men slutsatsen att mängden detaljer som kan extraheras ur den högupplösta bilden är större än mängden detaljer som kan extraheras ur de lågupplösta bilderna var för sig kan dras. Detta gäller för tredimensionella objekt i allmänhet även om det här endast testats för metallnycklar.

Om man betraktar bildkvaliteten hos det fotograferade bildmaterialet som använts under försöken kan man dra slutsatser om andra kameror och andra avstånd, genom att jämföra med bilder tagna med dessa. Det gör att begränsningen i försöksutrustning inte minskar betydelsen av resultaten mycket.

11.2 Rekommendationer för fortsatt arbete

Under arbetet med den här rapporten har det fokuserats på att återskapa nycklar, men det finns givetvis andra objekt som kan vara intressanta att titta närmare på. Ett liknanade exempel inom säkerhet, eller brist på sådan, är kontokort. Dessa skulle på samma sätt kunna fotograferas på avstånd, och informationen som finns skriven på kortet skulle kunna avläsas. Denna information är ofta tillräcklig för att handla i en internetbutik, även om man nu för tiden ofta behöver information från båda sidorna av kortet. Kan man utnyttja automatisk bokstavsigenkänning för att ta fram informationen? Vilken typ av information känner man till utan att betrakta kortets utseende? Dessa frågor samt var en attack av den här typen skulle kunna genomföras kan diskuteras. Kontokort kan här inte betraktas som tredimensionella objekt men resultaten i denna rapport talar ändå för att internetbedrägerier skulle kunna underlättas av den föreslagna bildbehandlingen.

Följande avsnitt behandlar framtida arbete som kan vara intressant inom de områden som rapporten har behandlat.

11.2.1 Fotografering

Vill man i framtiden utföra ett liknande projekt med målet att förbättra resultatet så är fotograferingen ett moment som kan varieras på flera sätt. Vill man ha relevantare resultat kan man montera digitalkameran mer realistiskt under försöken och kanske utföra hela experimentet exakt enligt ett scenario. Val av kamera är också intressant. Här har en enkel digitalkamera och en mobiltelefonkamera testats. Vad kan man uppnå med exempelvis en digital systemkamera eller kan man rent av få hjälp av en tidningsfotograf med försöket? En professionell fotograf skulle antagligen kunna prestera bättre bilder än en lekman även med en billigare kamera, då fotografering är något som kräver en viss skicklighet.

Kanske är det lättare att motivera en riktigt dyr kamera då ett säkrare system än ett dörrlås ska attackeras, men även för dörrlås är olika och bättre kameror intressant, speciellt eftersom kameror utvecklas så snabbt. En kamera som är modern och dyr idag är antagligen inte dyr alls om fem år.

Samtidigt är sannolikheten att vanliga fysiska dörrlås ser likadana ut om fem år som de gör idag stor. Man kan också undersöka om det går att skapa bättre bilder med en mindre kamera snarare än med en bättre. En liten kamera kan kanske gömmas precis i närheten av det objekt man vill kopiera, vilket exempelvis kan utnyttjas för att kopiera informationen på folks kontokort när de tar ut pengar ur en bankomat.

Intressant vore också att försöka sig på en beräkning av vilken kamera som krävs för att utföra experimentet med ett visst avstånd mellan kamera och objekt. Detta kan sedan jämföras med vilka kameror som över huvud taget existerar, och hur utvecklingen av digitala kameror sett ut hittills.

Ett problem som här inte diskuterats är det faktum att någon måste utföra fotograferingen manuellt. Detta kan vara en stor begränsning om man vill gömma kamerautrustningen. En automatisk rörelsedetektering kan användas, kanske med hjälp av karaktäristiska kurvor föreslaget av Kim[18], eller enklare kommersiella alternativ. I kombination med en monterad digital videokamera kan detta bättre visa vissa typer av säkerhetsbrister.

11.2.2 Bildförbättring

Flera av de använda metoderna kan varieras och andra metoder kan testas i praktiken. Även om de metoder som valts kan motiveras så har det inte visats att dessa presterar bäst i en praktisk situation. Registrering i frekvensdomänen är en sak som borde undersökas, då de antagna transformationerna lämpar sig väl för detta. Upplösningshöjning med MAP eller POCS skulle vara intressant för att undersöka hur antalet lågupplösta bilder påverkar resultatet.

Den manuella kantdetekteringen kan bytas ut mot en automatiserad. Intressant vore också att inrikta hela experimentet mer på kantdetektering än upplösningshöjning. Man skulle kunna undersöka vilket resultat som kan uppnås genom att i varje lågupplöst bild detektera nyckelkonturen och sedan beräkna en mest trolig nyckelkontur med exempelvis MAP utifrån dessa. Man skulle då ha mindre information att ta hänsyn till när själva optimeringssteget utförs, vilket inte måste betyda ett sämre resultat. Samtidigt kan man räkna med avsevärt snabbare beräkningar.

Detaljer som skulle kunna göras bättre i experimentimplementationen diskuteras i avsnitt 10.1.2. Dessa är snarare tips för ökad användarvänlighet än rekommendationer för fortsatt arbete.

11.2.3 Lås och nycklar

Då enbart stiftcylinderlås av en viss typ har undersökts finns det här stora möjligheter för fortsatt arbete. Flera typer av lås, många mycket mer avancerade än de som undersökts, finns på marknaden.

Alla nycklar går inte att återskapa bara för att man har en bild av nyckeln från sidan, oavsett hur detaljerad den är. Ytterligare arbete behövs för att till exempel kontrollera hur svårt det är att återskapa nycklar där nedskärningarna inte sitter jämnt fördelade och inte är skurna vinkelrätt mot nyckeln samt där de avgörande måtten hos nyckeln inte kan ses från en enda sida.

Inom låsbranchen borde det också ske ett fortsatt arbete. Bedrägeriet som beskrevs av philly.com[31] är i praktiken ganska enkelt att genomföra. Det kan dessutom förenklas genom olika bildbehandlingsmetoder, vilka presenterats i denna rapport, som gör det möjligt att extrahera ytterligare information ur bilderna. Det bör därför ligga i låsbranchens intresse att försöka utveckla nycklar som inte går att kopiera lika enkelt. Det ideala vore om man inte fick någon information om nyckeln och låset genom att studera nyckeln. Detta för tankarna direkt till elektroniska lås men även fysiska nycklar som är mycket säkrare än dagens vanliga dörrnycklar finns. Det är en avvägningsfråga mellan hur lättillverkade nycklarna ska vara och vilken säkerhet som krävs.

I många fall finns svagare länkar än låsmekanismen när det gäller möjligheten att ta sig in i ett låst utrymme. Låsets och dörrens hållfasthet, glasfönster och möjligheten att komma över nycklar utan att kopiera dem exempelvis. Mängden arbete och utrustning som krävs vid den föreslagna attacken blir snabbt stor i proportion till värdet av att ta sig in i det låsta utrymmet.

Litteraturförteckning

[1] Yucel Altunbasak and Andrew J. Patti. A maximum a posteriori estimator for high resolution videoreconstruction from mpeg video. *Image Processing, 2000. Proceedings. 2000 International Conference on*, 2:649–652, 2000.

[2] R. Bernstein. Digital image processing of earth observation sensor data. *IBM Journal of Research and Development*, 20:40–67, January 1976.

[3] Jan Biemond, Reginald L. Lagendijk, and Russell M. Merserau. Iterative methods for image deblurring. *Proceedings of the IEEE*, 78(5):856–883, May 1990.

[4] Matt Blaze. Cryptology and physical security: Rights amplification in master-keyed mechanical locks. 2002. URL: `http://www.crypto.com/papers/mk.pdf` [01/03/06].

[5] Lisa Gottesfeld Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):327–372, December 1992.

[6] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, November 1986.

[7] E. De Castro and C. Morandi. Registration of translated and rotated images using finite fourier transforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 700–703, September 1987.

[8] Rachid Deriche. Using canny's criteria to derive a recursively implemented optimal detector. *International Journal of Computer Vision*, 1(2):167–187, June 1987.

[9] Michael Elad and Yacov Hel-Or. A fast super-resolution reconstruction algorithm for pure translational motion and common space-invariant blur. *IEEE Transactions on Image Processing*, 10(8):1187–1193, August 2001.

[10] Bob Fisher, Simon Perkins, Ashley Walker, and Erik Wolfart. Image processing operation reference. URL: `http://www.cee.hw.ac.uk/hipr/html/hipr_top.html`. [01/03/06], 2004.

[11] J. C. Gillete, T. M. Stadtmiller, and R. C. Hardie. Aliasing reduction in staring infrared imagers utilizing subpixel techniques. *Opt. Eng*, 34(11):3130, November 1995.

[12] Giovani Gomez. Estimating local variance for gaussian filtering. URL: `http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/GOMEZ1/` [01/03/06], 2001.

[13] Bill Green. Autonomous systems lab. URL: `http://www.pages.drexel.edu/~weg22/index.html` [01/03/06], 2002.

[14] Cosmin Grigorescu, Nicolai Petkov, and Michel A. Westenberg. Contour detection based on nonclassical receptive field inhibition. *IEEE Transactions on Image Processing*, 12(7):729–739, July 2003.

[15] Anders Haraldsson. Så stjäl de dina pengar. URL: `http://www.aftonbladet.se/vss/nyheter/story/0,2789,766338,00.html` [10/02/06], 2006.

[16] Allen D. Hillery and Roland T. Chin. Iterative wiener filters for image restoration. *Signal Processing, IEEE Transactions on*, 39(8):1892–1899, August 1991.

[17] D.H. Hubel. Explorations of the primary visual cortex, 1955-1978. *Nature*, 299:515–524, 1982.

[18] Min-Kyu Kim, Ealgoo Kim, Daeyn Shim, Seong-Ik Jang, Gyudong Kim, and Wonchan Kim. An efficient global motion characterization method for image processing application. *IEEE Transaction on Consumer Electronics*, 43(4), November 1997.

[19] Yido Koo and Wonchan Kim. An image resolution enhancing technique using adaptive sub-pixel interpolation for digital still camera system. *IEEE Transactions on Consumer Electronics*, 45(1):118–123, February 1999.

[20] C.D. Kuglin and D.C. Hines. The phase correlation image alignment method. *Proceedings of the IEEE 1975 International Conference on Cybernetics and Society.*, pages 163–165, September 1975. New York.

[21] G.Q. Maguire, M.E. Noz Jr., and H. Rusinek. Software tools to standardize and automate the correlation of images with and between diagnostic modalities. *IEEE Computer Graphics and Applications*, 1990.

[22] Steve Mann and Rosalind W. Picard. Virtual bellows: Constructing high quality stills from video. *M.I.T. Media Laboratory Perceptual Computing Section Technical Report No. 259*, 1994.

[23] Thomas P, Weldon, William E. Higgins, and Dennis F. Dunn. Efficient gabor filter design for texture segmentation. 1996.

[24] Panasonic. *Bruksanvisning, Digitalkamera, DMC-LC70EGM.*

[25] Andrew J. Patti, M. Ibrahim Sezan, and A. Murat Tekalp. High resolution standards conversion of low resolution video. *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, 4:2197–2200, May 1995.

[26] Shmuel Peleg, Danny Keren, and Limor Schweitzer. Improving image resolution using subpixel motion. *Pattern Recognition Letters*, 5:223–226, March 1987.

[27] Arul Prakash. Script segmentation of multi-script documents. URL: `http://students.iiit.net/~arul/report/report.html` [20/04/06], 2002.

[28] Leonid I. Rudin and Stanley Osher. Total variation based image restoration with free local constraints. *Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference*, 1:31–35, November 1994.

[29] Samsung. *Specifcation, SGH-E700*. URL: `http://nordic.samsungmobile.com/eng/mobile_phone/sgh-e700/specification.jsp`. [30/04/06].

[30] Marie Sandström. Liveness detection in fingerprint recognition systems. LITH-ISY-EX-3557-2004, Linköping University, Linköping, Sweden, 2004.

[31] John Shiffman. Check was in the mail, but thieves were at mailbox. URL: `http://www.philly.com/mld/philly/news/local/12554094.htm` [10/02/06], 2005.

[32] Standardiseringskommissionen i Sverige SIS. Svensk standard ss 3079. 1993.

[33] Standardiseringskommissionen i Sverige SIS. Svensk standard ss 1303. 1998.

[34] Standardiseringskommissionen i Sverige SIS. Svensk standard ss 3522. 2000.

[35] F. Smeraldi and J. Bigun. Retinal vision applied to facial features detection and face authentication. *Pattern Recognition Letters*, 23, 2002.

[36] SLRs stadgar. Etiska regler för medlemmar och företag anslutna till slr, bilaga 5.

[37] Henry Stark and Peyma Oskoui. High-resolution image recovery from image-plane arrays, using convex projections. *Journal of the Optical Society of America A*, 6(11):1715–1726, November 1989.

[38] A. Murat Tekalp, Mehmet K. Ozkan, and M. Ibrahim Sezan. High-resolution image reconstruction from lower-resolution imagesequences and space-varying image restoration. *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, 3:169–173, March 1992.

[39] Peter J. Verveer and Thomas M. Jovin. Efficient superresolution restoration algorithms using maximum a posteriori estimations with application to fluorescence microscopy. *Journal of the Optical Society of America A*, 14(8):1696–1706, August 1997.

[40] Jainwei Yang, Lifeng Liu, Tianzi Jiang, and Yong Fan. A modified gabor filter design method for fingerprint image enhancement. *Pattern Recognition Letters*, 24, 2003.

[41] I.T. Young, J.J. Gerbrands, and L.J. van Vliet. Image processing fundamentals. URL: `http://www.ph.tn.tudelft.nl/Courses/FIP/noframes/fip-Basic.html` [27/04/06].

Bilaga A

Utrustning

Denna bilaga innehåller detaljerad information om den utrustning som användes i experimentet.

Digitalkamera

För experimentet där en nyckel fotograferades på långt avstånd användes en DMC-LC70EGM[24] från Panasonic. Här följer detaljerad information om denna kamera, och den ses i figur A.1.

Figur A.1. Panasonic DMC-LC70.

- *Modell:* Panasonic DMC-LC70EGM
- *Kamerans effektiva pixlar:* 4.000.000
- *Bildsensor:* 1/2.5 tum CCD, totalt antal pixlar 4.230.000
- *Lins:* Optisk 3x zoom, f=5.8-17.4mm (detta motsvarar 35-105mm för en 35mm-filmkamera)
- *Digital zoom:* 3x

- *Fokusområde:* 50cm-∞, Makroläge: 10cm-∞
- *Inspelning av bildsekvenser:* 3.3 bilder/sekund (hög hastighet), 2 bilder/sekund (låg hastighet), Max 9 bilder för 2304×1728 pixlar
- *Bildstorlek:* 2304×1728 pixlar.
- *ISO Känslighet:* Auto och 50, 100 och 200.

Mobiltelefonkamera

För experimentet med fotografering på nära håll med en mobiltelefonkamera användes en SGH-E700[29] från Samsung. Här följer detaljerad information om kameran, och mobiltelefonen ses i figur A.2.

Figur A.2. Samsung SGH-E700.

- *Modell:* Samsung SGH-E700
- *Kamerans effektiva pixlar:* 307.200
- *Bildstorlek:* 640×480 pixlar.
- *Digital zoom:* 5x
- *Inspelning av bildsekvenser:* 15 respektive 9 bilder per serie.

Implementationsspråk

För implementationen i exprimentet användes MATLAB, Copyright 1984-2005 The MathWorks, version 7.1.0.183 (R14) Service Pack 3.

Bildbehandlingsmjukvara

För beskärningar, borttagning av bakgrund och liknande enkla operationer användes Adobe® Photoshop® CS, version 8.0.

Filtreringsmjukvara

Filtreringar utfördes i Image Analyzer 1.25, Freeware for Windows 98/ME/2000/XP (http://meesoft.logicnet.dk/).

Mätutrustning

För mätning av avstånd mellan kameran och det fotograferade objektet användes ett vanligt måttband. För de mer noggranna mätningar som gjordes av nyckelnedskärningar användes ett digitalt skjutmått från Biltema med följande tekniska specifikationer. Skjutmåttet ses i figur A.3.

- *Modell:* Biltema digitalt skjutmått
- *Mätområde:* 0-150mm
- *Upplösning:* 0.01mm
- *Mätnoggrannhet:* ±0.03mm

Figur A.3. Digitalt skjutmått.

Bilaga B

Experimentdetaljer

Denna bilaga innehåller information om hur vissa delar av experimentet utfördes, mer i detalj.

B.1 Fotografering

En digitalkamera, Panasonic DMC-LC70, med följande inställningar användes för att ta fotografier av nyckeln enligt scenario A i experimentbeskrivningen i kapitel 8.

- Multi-tagning, hög hastighet, 9 bilder.
- Ej blixt.
- Porträttläge.
- Automatisk vitbalans.
- ISO-känslighet 50.
- Bildstorlek 2304 (2304×1728 pixlar).
- Bildkvalitet fin (ickekomprimerade bilder).
- Autofokusering med punktområde.
- Optisk zoom 3 gånger.
- Digital zoom 3 gånger (maximal zoom).
- Bildjustering vivid (skarpa bilder).

För mobiltelefonkameran, SGH-E700, i scenario B, användes följande inställningar:

- Fotoformat: VGA (640×480 pixlar).
- Fotokvalitet: Superfin
- Kameraläge: Multi-tagning, normal hastighet, 9 bilder.

B.2 Bildbehandling

De delar som inte kunde automatiseras inom bildbehandlingen beskrivs närmare här.

Adobe® Photoshop® CS

För enkla operationer på bilderna användes Adobe Photoshop. För varje bildserie bedömdes vilka punkter som behövde genomföras för att resultatet skulle bli så detaljerat som möjligt.

- Det första som gjordes var att beskära bilderna så att bara de intressanta delarna av nyckeln var kvar. Detta gjordes genom att markera det intressanta området och sedan använda funktionen crop.

  Image → Crop

- För att nyckeln skulle hamna rakt i bilden roterades bilden genom att använda en rotationsfunktion med rätt vinkel som parameter.

  Image → Rotate Canvas → Arbitrary...

- För att neutralisera bakgrunden, i de fall där det var nödvändigt, raderades framträdande delar med penselverktyget.

- I de fall där wienerfiltrering inte gav ett bra resultat ökades bildens skarphet genom att använda funktionen unsharpen mask.

  Filter → Sharpen → Unsharpen Mask...

  Eftersom bilderna kan skilja sig mycket från varandra, varierade de värden som gav ett gott resultat mellan de olika bildserierna. Så för att få fram det bästa resultatet provades många olika värden.

- Konturer ritades ut på bilderna genom att använda linjeverktyget.

Image Analyzer

För att göra bilderna skarpare filtrerades dessa i Image Analyzer. Eftersom bilderna skiljer sig mycket från varandra kan inte heller här fasta värden, som ger ett gott resultat, anges. Funktionen som användes var deconvolution.

Filters → Restoration by Deconvolution...

För att flera olika parametrar skulle kunna provas på samma gång användes funktionen Test, som skapade bilder för alla olika värden mellan satta parametrar. Genom att variera modell och inparametrar kunde bilden göras bättre i de flesta fall.

Bilaga C

Programkod

Här följer en handledning i användning av den implementation som gjorts. Koden är skriven för MATLAB 7.1 och är uppdelad i filerna *runscript.m*, *register.m*, *enhance.m* och *contour.m*. De presenteras i avsnitten C.1 till C.4.

C.1 runscript.m

Detta är ett skript som förenklar körningen av resterande delar av implementationen och det är i denna fil som alla parametrar specifika för körningen anges. Dessa är som följer:

- `MaxDiff` - Den maximala förskjutningen mellan objekt i två lågupplösta bilder, räknat i pixlar.
- `files` - Samtliga bilder som ska användas för upplösningsförbättringen. Dessa ska vara i bitmap-format med 24 bitars färgdjup.
- `outfile` - Fil att spara den högupplösta bilden i. Även denna är i bitmap-format med 24 bitars färgdjup.
- `N` - Antalet bilder angivna i `files`. I det utförda experimentet var denna parameter 9 för samtliga bildserier utom i de fall då enstaka bilder kasserades på grund av registreringssvårigheter.
- `M` - Antal gånger som upplösningen förbättras. 2 betyder exempelvis att upplösningen fördubblas i båda dimensionerna och alltså att den högupplösta bilden får 4 gånger så många pixlar som en av de lågupplösta. I det utförda experimentet valdes `M` till 5, vilket alltså innebär registrering med en noggrannhet av 0.2 pixlar.

Här föjer ett exempel på hur ett körskript kan se ut, och här ses hur man genom att köra *register.m* och *enhance.m* flera gånger, med utdata från en körning som indata till nästa, kan öka upplösningen i flera steg om man har ett stort antal bilder. I det utförda experimentet utfördes upplösningshöjningen i ett enda steg.

```
MaxDiff=10
files=['key0001.bmp';'key0002.bmp';'key0003.bmp';'key0004.bmp']
outfile='series1.bmp'
N=4
M=2
register
enhance

MaxDiff=10
files=['key0005.bmp';'key0006.bmp';'key0007.bmp';'key0008.bmp']
outfile='series2.bmp'
N=4
M=2
register
enhance

MaxDiff=20
files=['series1.bmp';'series2.bmp']
outfile='result.bmp'
N=2
M=2
register
enhance
```

C.2 register.m

Denna fil behöver aldrig köras manuellt utan körs endast av *runscript.m*, men om innehållet kan sägas att det är här som den inbördes förskjutningen mellan bilderna angivna i *runscript.m* beräknas enligt metoden beskriven i avsnitt 4.6.

```
% Parameters (variables which have to be set to run this program part)
% N - The number of images used.
% M - The sub-pixel resolution (2 means every low-res pixel is 2*2=4
% high-res pixels).
% files - A vector with file names of low-res images.

Imgres=imread(files(1,:), 'bmp');
```

```
% Extract image resolution from the first image (all images have the
% same resolution).
Xres=get(image(Imgres), 'xdata');
Yres=get(image(Imgres), 'ydata');
Xr=Xres(2);
Yr=Yres(2);

Md=2*M*MaxDiff;
Xd=0;
Yd=0;

A=zeros(Yr,Xr,3,N); % 3, RGB

for n = 1:N,
  A(:,:,:,n)=imread(files(n,:), 'bmp');
end

A=double(A);

Afx = double(zeros(Xr*M-1, N));
Afy = double(zeros(Yr*M-1, N));

Px = double(zeros(Md+1, N));
sx = double(zeros(Md+1, N));

Py = double(zeros(Md+1, N));
sy = double(zeros(Md+1, N));

% Calculate displacement, all images against image 1.
for color = 1:3,
  for n = 1:N,
    % Calculate characteristic curves.
    for j = 1:Xr-1,
      for i = 1:Yr-1,
        for k = 0:M-1,
          Afx((j-1)*M+k+1, n)=Afx((j-1)*M+k+1,n)+
         ((M-k)/(Yr*M))*A(i, j, color, n)+(k/(Yr*M))*A(i, j+1, color, n);
          Afy((i-1)*M+k+1, n)=Afy((i-1)*M+k+1,n)+
         ((M-k)/(Xr*M))*A(i, j, color, n)+(k/(Xr*M))*A(i+1, j, color, n);
        end
      end
    end

    % X displacement from curve 1.
    for i = 1:Md+1,
      sx(i, n)=i-((Md/2)+1);
```

```
    end
    for i = 1:Md+1,
      sum=0;
     for j=max((Md/2)+1,-sx(i,n)):min(Xr*M-(Md/2)-sx(i,n),Xr*M-(Md/2)),
        sum=sum+abs(Afx(j, 1)-Afx(j+sx(i, n), n));
      end
      Px(i, n)=sum;
    end

    % Y displacement from curve 1.
    for i = 1:Md+1,
      sy(i, n)=i-((Md/2)+1);
    end
    for i = 1:Md+1,
      sum=0;
     for j=max((Md/2)+1,-sy(i,n)):min(Yr*M-(Md/2)-sy(i,n),Yr*M-(Md/2)),
        sum=sum+abs(Afy(j, 1)-Afy(j+sy(i, n), n));
      end
      Py(i, n)=sum;
    end

    % Take displacement according to the smallest curve difference.
    [Xdiff,Xi]=min(Px(:, n));
    [Ydiff,Yi]=min(Py(:, n));
    Xd(color, n)=-(Xi-(Md/2+1))/M;
    Yd(color, n)=-(Yi-(Md/2+1))/M;
  end
end

% Print the result.
Xd
Yd
```

C.3 enhance.m

Inte heller denna fil körs någonsin manuellt men här skapas den högupplösta bilden och sparas under det filnamn som angetts i `outfile` i *runscript.m*.

```
% Parameters (variables which have to be set to run this program part)
% N - The number of images used.
% M - The sub-pixel resolution (2 means every low-res pixel is 2*2=4
% high-res pixels).
% outfile is the file where the result image will be saved.
```

```
% Color channels will be translated the same amount of sub-pixels.
% The median is used to ensure a valid offset of sub-pixels is used.
for n = 1:N,
  Xdiff(n)=median(Xd(:, n));
  Ydiff(n)=median(Yd(:, n));
end

W=double(zeros(Yr*M, Xr*M, 3));

% Builds high-res picture from low-res pictures.
for color=1:3,
  for x=1:Xr,
    for y=1:Yr,
      flag1=0;
      for n = 1:N,
        if (((Xdiff(n)>0) & (x<ceil(Xdiff(n)+1))) |
           ((Ydiff(n)>0) & (y<ceil(Ydiff(n)+1))) |
           ((Xdiff(n)<0) & (x>ceil(Xr+Xdiff(n)-1))) |
           ((Ydiff(n)<0) &  (y>ceil(Yr+Ydiff(n)-1))))
          % Not possible to grab information from all pictures.
          flag1=1;
        end
      end
      if flag1==1
        for subx = 0:M-1,
          for suby = 0:M-1,
            W(y*M-suby, x*M-suby, color)=A(y, x, color, 1);
          end
        end
      else
        for subx = 0:M-1,
          for suby = 0:M-1,
            sum=A(y, x, color, 1);
            for n = 2:N,
              % Build sub-pixel from corresponding pixels in
              % the low res pictures.
              sum=sum+A(ceil((y*M-suby-M*Ydiff(n))/M),
                        ceil((x*M-subx-M*Xdiff(n))/M),color, n);
            end
            W(y*M-suby, x*M-subx, color)=round(sum/N);
          end
        end
      end
    end
  end
end
```

```
% Write result
imwrite(uint8(W), outfile, 'bmp');
```

C.4 contour.m

Denna fil innehåller skapandet av nyckelkonturen, givet en handritad nyckelkontur. Att detta sker i ett separat program som inte startas från *runscript.m* beror på att man först måste rita konturen i den högupplösta bilden innan beräkningen kan ske. Parametrar till *contour.m* är följande:

- `infile` - Ritad nyckelkontur i bitmap-format med 24 bitars färgdjup.
- `outfile` - Resultatfil i bitmap-format med 8 bitars gråskala.
- `Cang` - Vinkel för nyckelns nedskärningar.

Ett exempel på hur en ritad nyckelkontur kan se ut samt resultatet av konturanpassningen kan ses i bilaga D.

```
% infile is the name of the image
% outfile is the file to write to
% Cang is the cutting angle of the key.

% Load the file from disk.
A=imread(infile, 'bmp');

Xres=get(image(A), 'xdata');
Yres=get(image(A), 'ydata');
Xr=Xres(2);
Yr=Yres(2);

A=double(A);

B=zeros(Yr,Xr);
down=0;
up=0;
points=zeros(15,2);
point=zeros(15,2);
pointcounter=1;
distances=zeros(7,1);

% Register key points in the picture.
for xi=1:Xr,
  cdone=0;
```

```
for yi=1:Yr,
 if A(yi,xi,1)==255 && A(yi,xi,2)==255 && A(yi,xi,3)==255 && cdone==0
    if xi==1
      last=yi;
    end

    cdone=1;
    B(yi,xi)=255;
    Yvalue(xi)=yi;


    if down==0 && up==0 && yi>last
      down=1;
      up=0;
      points(pointcounter,1)=last;
      points(pointcounter,2)=xi-1;
      pointcounter=pointcounter+1;

    elseif down==1 && up==0
      if yi>last % down
        plainmark1=xi;
      end
      if yi<last %up
        plainmark2=xi-1;
        points(pointcounter,1)=last;
        points(pointcounter,2)=plainmark1+
                                round((plainmark2-plainmark1)/2);
        pointcounter=pointcounter+1;
        down=0;
        up=1;
        plainmark1=xi;
      end

    elseif down==0 && up==1
      if yi<last % up
        plainmark1=xi;
      end
      if yi>last %down
        plainmark2=xi-1;
        points(pointcounter,1)=last;
        points(pointcounter,2)=plainmark1+
                                round((plainmark2-plainmark1)/2);
        pointcounter=pointcounter+1;
        down=1;
        up=0;
        plainmark1=xi;
```

```
        end
      end

      last=yi;
    end
  end
end

% Mark points in gray
for p=1:15,
  B(points(p,1),points(p,2))=128;
end

% Calculate distance between first and last cut in the key.
distance=0;
distances(1)=0;
for p=2:7,
  distances(p)=points(2*p,2)-points(2*p-2,2);
  distance=distance+points(2*p,2)-points(2*p-2,2);
end

k=1/tan(Cang/360*2*pi);
Quadsums=zeros(31,7,21,21); %ym,i,d,f

% Calculate quadratic sums for the distance between the drawn
% contour and possible keys contours.
for f=-10:10,
  for d=-10:10,
    point(2,2)=points(2,2)+f;
    for p=2:7,
      point(2*p,2)=point(2,2)+round(((distance+d)/6)*(p-1));
    end

    for i=1:7,
      for ym=-15:15,
        for x=points(i*2-1,2):points(i*2+1,2),
          x1=point(i*2,2);
          x2=abs(x1-x);
          Quadsums(16+ym,i,d+11,f+11)=Quadsums(16+ym,i,d+11,f+11)+
                                  (Yvalue(x)+ym+x2*k-points(i*2,1))*
                                  (Yvalue(x)+ym+x2*k-points(i*2,1));
        end
      end
    end
  end
end
```

```
Quadvect=zeros(21*21,8);

% Pick the best solution
for f=-10:10,
  for d=-10:10,
    [val,index]=min(Quadsums(:,:,d+11,f+11));
    Quadvect((f+10)*21+d+11,1)=sum(val);
    Quadvect((f+10)*21+d+11,2:8)=index;
  end
end

[val,index]=min(Quadvect(:,1));

f=floor(index/21)-11
d=mod(index,21)-11
Ym=Quadvect(index,2:8)-16

% Draw the correct key contour.
point(2,2)=points(2,2)+f;
for p=2:7,
  point(2*p,2)=point(2,2)+round(((distance+d)/6)*(p-1));
end

for i=1:7,
  for x=points(i*2-1,2):points(i*2+1,2),
        x1=point(i*2,2);
        x2=abs(x1-x);
        B(round(-x2*k+points(i*2,1))-Ym(i),x)=128;
  end
end

% Write the contour to disk.
imwrite(uint8(B), outfile, 'bmp');
```

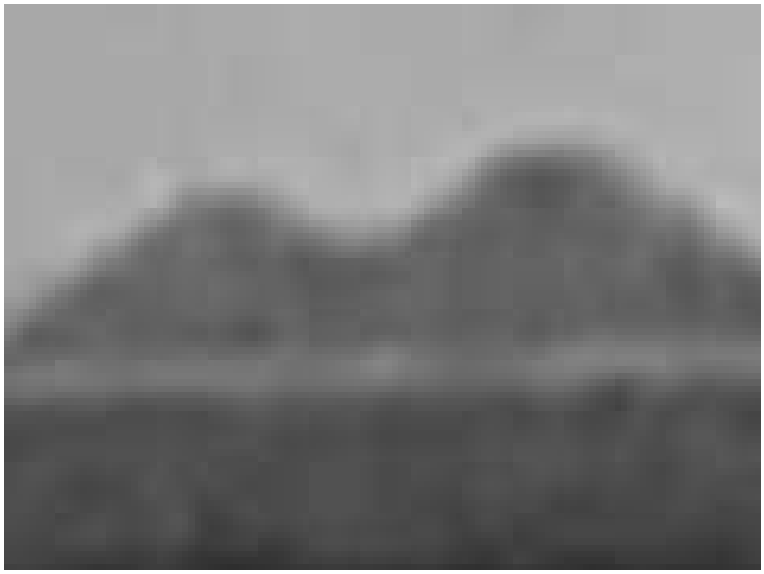
Bilaga D

Detaljerade resultat

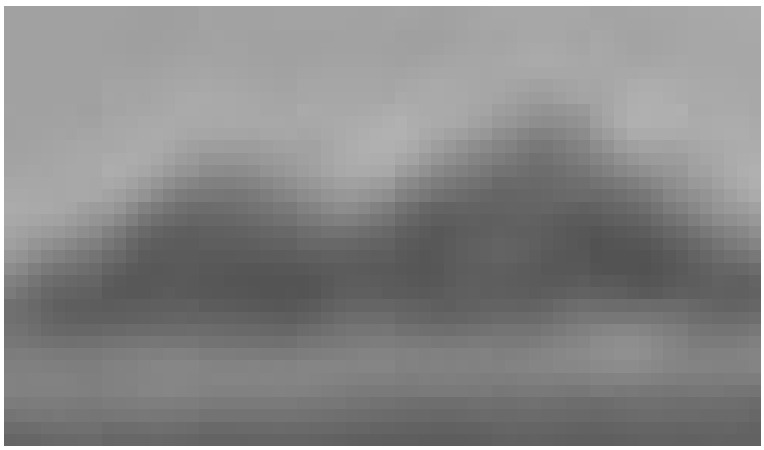
I denna bilaga presenteras detaljerade resultat från de olika delarna av experimentet.

D.1 Bilder

Här presenteras exempel på bilder ur samtliga bildserier. I figur D.1 till D.4 samt figur D.6 till D.9 är detaljer i bilderna utskurna för att belysa skillnaden i bildkvalitet. Dessa bilder har alltså inte upplösningsförbättrats utan endast beskurits och förstorats.

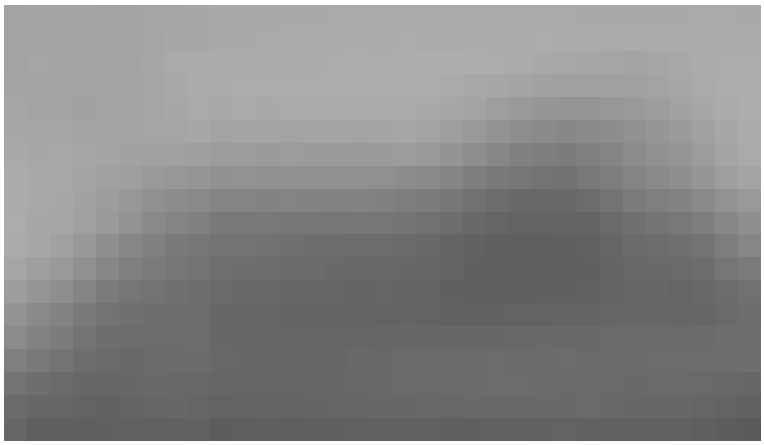


Figur D.1. Fotografi taget på 1 meters avstånd enligt scenario A i kapitel 8.

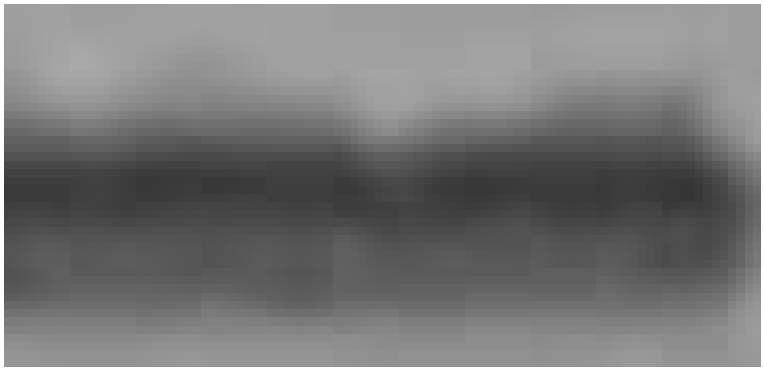


Figur D.2. Fotografi taget på 2 meters avstånd enligt scenario A i kapitel 8.

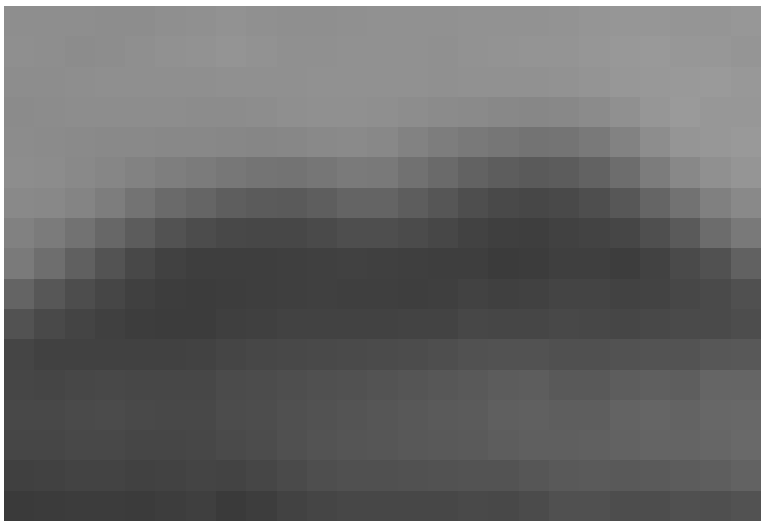
Figur D.3. Fotografi taget på 3 meters avstånd enligt scenario A i kapitel 8.



Figur D.4. Fotografi taget på 4 meters avstånd enligt scenario A i kapitel 8.

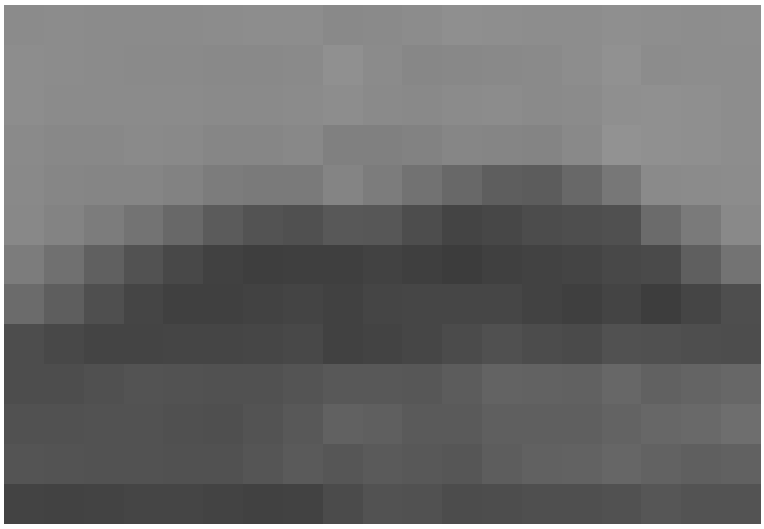


Figur D.5. Fotografi taget på 6.5 meters avstånd enligt scenario A i kapitel 8.

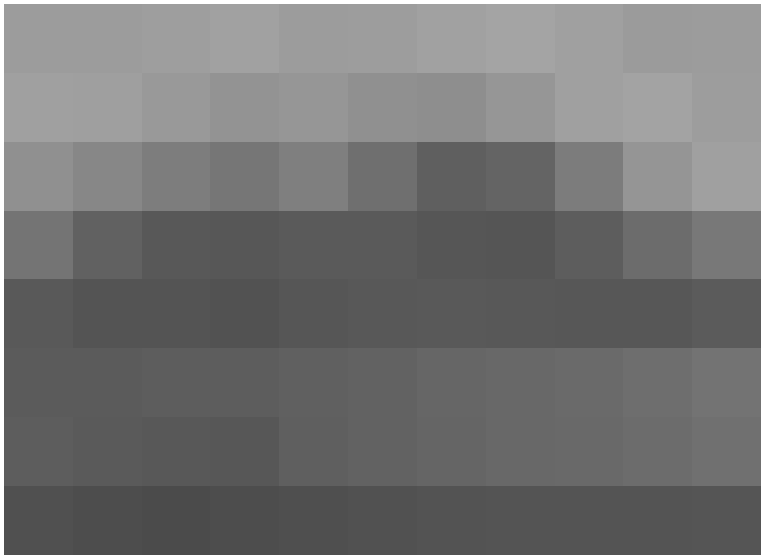


Figur D.6. Fotografi taget på 10 centimeters avstånd enligt scenario B i kapitel 8.

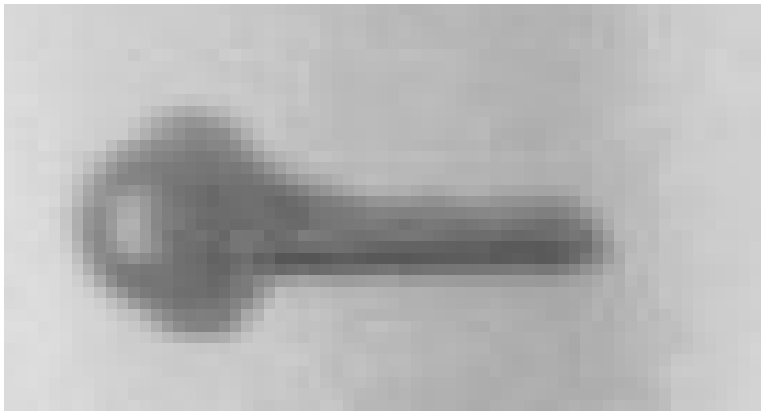
Figur D.7. Fotografi taget på 20 centimeters avstånd enligt scenario B i kapitel 8.



Figur D.8. Fotografi taget på 30 centimeters avstånd enligt scenario B i kapitel 8.



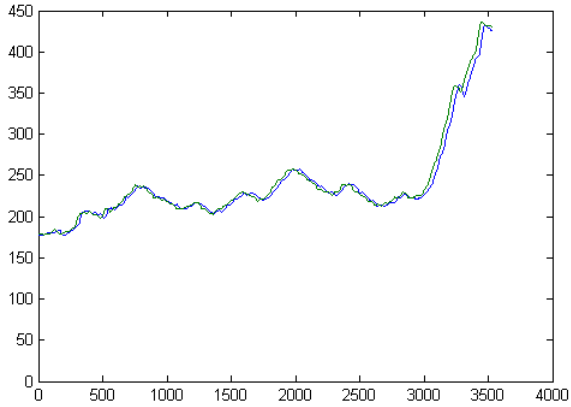
Figur D.9. Fotografi taget på 50 centimeters avstånd enligt scenario B i kapitel 8.



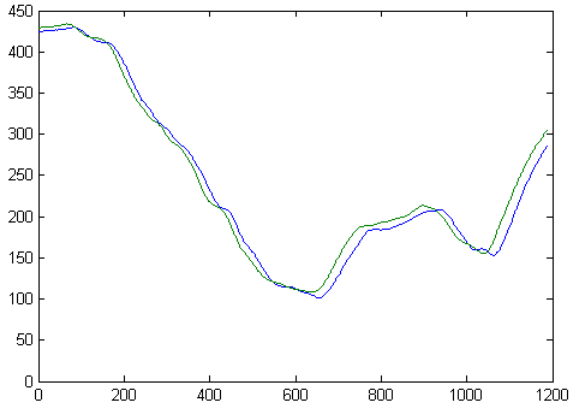
Figur D.10. Fotografi taget på 100 centimeters avstånd enligt scenario B i kapitel 8.

D.2 Karaktäristiska kurvor

Här presenteras exempel på karaktäristiska kurvor för två beskurna fotografier från försök 1 enligt scenario A, som beskrivs i kapitel 8. Kurvorna har genererats utifrån de två första bilderna i bildserien. Som synes är kurvorna i figur D.11 respektive D.12 starkt korrelerade.



Figur D.11. Karaktäristiska kurvor f_x för två fotografier.



Figur D.12. Karaktäristiska kurvor f_y för två fotografier.

D.3 Registrering

Registreringsresultat för två bildserier tagna med digitalkameran kan ses i tabell D.1 och D.2, och registreringsresultat för bilder tagna med mobiltelefonkameran kan ses i tabell D.3 och D.4.

Bild nr	Förskjutning x-led (pixlar)			Förskjutning y-led (pixlar)		
	R	G	B	R	G	B
1	5.60	6.00	6.00	3.20	3.00	2.80
2	8.80	8.60	7.20	5.20	5.60	6.20
3	6.60	6.60	6.60	3.80	3.80	3.80
4	6.00	5.40	4.40	1.40	1.80	2.20
5	4.80	5.00	5.40	6.40	6.20	6.00
6	8.20	8.20	7.00	3.40	3.80	4.20
7	3.80	4.00	3.80	4.60	4.40	4.40
8	5.60	6.20	6.40	4.20	4.00	4.00

Tabell D.1. Resultat av registreringen för bilder tagna med digitalkameran DMC-LC70 på 1 meters avstånd.

Bild nr	Förskjutning x-led (pixlar)			Förskjutning y-led (pixlar)		
	R	G	B	R	G	B
1	1.80	1.60	1.80	0.40	0.40	0.40
2	0.40	1.20	2.20	1.00	1.00	1.00
3	3.40	3.00	3.40	-0.20	-0.20	-0.20
4	-0.60	0.00	0.40	0.80	0.80	0.80
5	-1.00	-1.00	-1.20	1.80	1.80	1.80
6	2.00	2.20	2.40	1.60	1.60	1.40
7	1.40	1.20	1.00	0.60	0.60	0.60
8	1.40	1.60	2.40	0.40	0.20	0.20

Tabell D.2. Resultat av registreringen för bilder tagna med digitalkameran DMC-LC70 på 3 meters avstånd.

Bild nr	Förskjutning x-led (pixlar)			Förskjutning y-led (pixlar)		
	R	G	B	R	G	B
1	-2.40	-2.80	-3.20	0.80	0.80	1.00
2	-3.40	-3.60	-3.60	-1.20	-1.20	-1.00
3	-3.00	-3.20	-3.20	-0.80	-0.80	-0.80
4	-4.40	-4.80	-5.20	-1.40	-1.40	-1.20
5	-4.20	-3.80	-4.00	-1.00	-1.00	-1.00
6	-3.00	-3.20	-3.60	-1.80	-1.80	-1.80
7	-3.60	-3.40	-3.60	-1.00	-1.00	-1.00
8	-2.80	-3.00	-3.20	-0.60	-0.80	-0.60

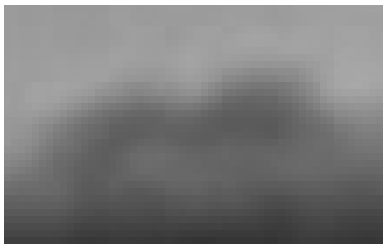
Tabell D.3. Resultat av registreringen för bilder tagna med mobiltelefonkameran SGH-E700 på 10 centimeters avstånd.

Bild nr	Förskjutning x-led (pixlar)			Förskjutning y-led (pixlar)		
	R	G	B	R	G	B
1	0.20	0.80	1.00	-0.20	-0.20	-0.40
2	-2.00	-1.80	-1.80	-0.60	-0.80	-0.80
3	-1.20	-1.00	-0.60	0.60	0.40	0.20
4	-0.20	-0.20	-0.20	1.20	1.20	1.20
5	-0.8	-0.40	0.00	0.80	0.60	0.60
6	-1.20	-0.80	-1.00	0.60	0.40	0.20
7	-0.60	-0.20	0.20	0.00	-0.20	-0.20
8	0.00	-0.20	-0.20	-1.60	-2.00	-2.00

Tabell D.4. Resultat av registreringen för bilder tagna med mobiltelefonkameran SGH-E700 på 50 centimeters avstånd.

D.4 Bildjämförelse

Här följer en detalj i en nyckelkontur, under olika delar av bildbehandlingsprocessen. Figur D.13 visar detaljen i ett lågupplöst fotografi taget på avståndet 3m, med digitalkamera. Figur D.14 visar samma detalj men i den upplösningsförbättrade bilden genererad ur hela motsvarande bildserie. I figur D.15 ses en filtrerad variant av den högupplösta bilden och i figur D.16 har konturen ritats i. Figur D.17 visar den nyckelkontur som anpassats efter dels den inritade konturen men också efter de givna reglerna för godkända nyckelutseenden.



Figur D.13. Fotografi taget på 3 meters avstånd enligt scenario A i kapitel 8.

Figur D.14. Samma detalj som i figur D.13, men här i den upplösningsförbättrade bilden.

Figur D.15. Samma detalj som i figur D.13, men här i den upplösningsförbättrade och filtrerade bilden.

Figur D.16. Samma detalj som i figur D.13, men här med konturen inritad i den upplösningsförbättrade och filtrerade bilden.

Figur D.17. Samma detalj som i figur D.13, men här med konturen anpassad efter givna regler.

LINKÖPING UNIVERSITY
ELECTRONIC PRESS

Linköpings universitet

Upphovsrätt

Detta dokument hålls tillgängligt på Internet — eller dess framtida ersättare — under 25 år från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida `http://www.ep.liu.se/`

Copyright

The publishers will keep this document online on the Internet — or its possible replacement — for a period of 25 years from the date of publication barring exceptional circumstances.

The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for your own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page: `http://www.ep.liu.se/`

© Linus Fredriksson, Martin Gyllensten