

Automatisk bildanalys av foton på Rubiks kub

ANDRÉ GRÄSMAN
och RASMUS GÖRANSSON



**KTH Datavetenskap
och kommunikation**

Automatisk bildanalys av foton på Rubiks kub

A N D R É G R Ä S M A N
o c h R A S M U S G Ö R A N S S O N

Examensarbete i datalogi om 15 högskolepoäng
vid Programmet för datateknik
Kungliga Tekniska Högskolan år 2011
Handledare på CSC var Lars Kjell Dahl
Examinator var Mads Dam

URL: [www.csc.kth.se/utbildning/kandidatexjobb/datateknik/2011/
grasman_andre_OCH_goransson_rasmus_K11046.pdf](http://www.csc.kth.se/utbildning/kandidatexjobb/datateknik/2011/grasman_andre_OCH_goransson_rasmus_K11046.pdf)

Kungliga tekniska högskolan
Skolan för datavetenskap och kommunikation

KTH CSC
100 44 Stockholm

URL: www.kth.se/csc

Abstract

This paper deals with various techniques used in the context of automatic image analysis. Content is intended to provide an overview of techniques used to extract information from images, with a concrete example. The paper describes how a program extracts information from two photographs of the Rubik's cube to determine the state of the cube. The program's work is divided into distinct steps with each step explained individually. Techniques covered include edge detection, shape classification of polygons, white balancing and comparison of colors.

An important part of the work is based on an understanding of how unstructured data can be processed into structured data. The process makes assumptions and estimates, so the information can be structured and used in the next step. The document shows how information about the analyzed object can be used to detect errors in the assumptions and estimates made. For example, the labels must not have colors that result in physically impossible pieces of the cube. The paper contains a result part where the program is demonstrated for photos of different quality.

Sammanfattning

Dokumentet behandlar olika tekniker som används i samband med automatisk bildanalys. Innehållets syfte är att ge en överblick på tekniker som används för att extrahera information ur bilder, med ett konkret exempel. Dokumentet beskriver hur ett program extraherar information från två fotografier på Rubiks kub för att fastställa tillståndet hos kuben. Programmets arbete delas upp i avgränsade steg där varje steg förklaras individuellt. Tekniker som berörs är bland andra kantigenkänning, formkategorisering av polygoner, vitbalansering och jämförelse av färger.

En viktig del av arbetet bygger på förståelsen om hur ostrukturerad information kan processas till strukturerad data. Processen gör antaganden och uppskattningar så informationen kan struktureras och användas i nästkommande steg. Dokumentet visar hur information om det analyserade objektet kan användas för att upptäcka felaktigheter i de antaganden och uppskattningar som har gjorts. Till exempel kan etiketter inte ha färger som resulterar i fysiskt omöjliga bitar på kuben. Dokumentet innehåller en resultatdel där programmet demonstreras för foton av olika kvalitet.

Förord

Denna rapport och dess tillhörande projekt tillhör ett kandidatexamensarbete (6Hp) inom datorvetenskap och kommunikation (CSC) på Kungliga Tekniska högskolan i Stockholm. Projektet och rapporten har utförts och författats av André Gräsman och Rasmus Göransson med Lars Kjeldahl som handledare. Inledningsvis har André gjort grunden till rapporten medan Rasmus fokuserat på implementationen. Tekniker och strategier har diskuterats tillsammans och arbetet har slutförts gemensamt.

Innehållsförteckning

1.	Introduktion.....	1
2.	Programmets flöde.....	3
2.1	Hitta polygoner i fotografierna.....	4
2.1.1	Areaberäkning av en polygon.....	6
2.1.2	Masscentrumberäkning för en yta.....	7
2.2	Kategorisering av etiketter.....	8
2.3	Identifiering av sidornas etiketter.....	10
2.4	Anpassning av koordinatsystem.....	12
2.5	Identifiering av etiketternas färger.....	14
2.6	Hopparning av etiketter med samma färg.....	16
2.6.1	Algoritm för ihop parning av etiketter med samma färg.....	17
2.7	Formatera resultatet som en sträng.....	18
3	Resultat.....	19
3.1	Första kubén.....	21
3.2	Andra kubén.....	22
3.3	Tredje kubén.....	23
3.4	Fjärde kubén.....	24
3.5	Femte kubén.....	25
4	Slutsats.....	26
5	Referenser.....	28
5.1	Böcker.....	28
5.2	Webbsidor.....	28

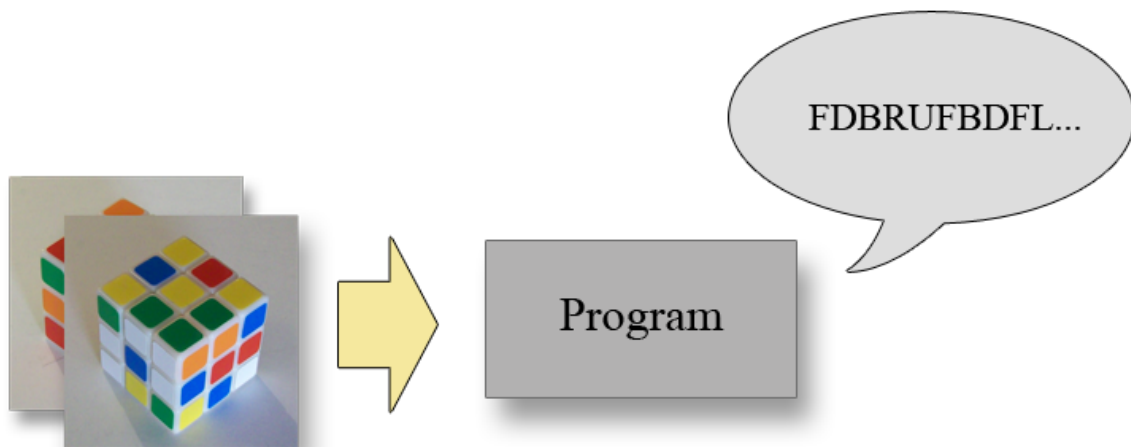
1. Introduktion

Intresset för datorseende och bildigenkänning är stort bland forskare, studenter och hobbyister. Ämnet har diskuterats i flera år och har potential att lyfta olika tekniker ett stort steg framåt. Framförallt har ämnet diskuterats inom medicinteknik, robotik och övervakning [2]. Google har nyligen offentliggjort en söktjänst som tillåter användaren att söka bilder på internet som liknar en bild som skickas som fråga [4]. Läkare har hjälp av datorer för att hitta skador och intressanta detaljer hos röntgenbilder [5], medan vägkameror hittar och tolkar registreringsnummer bland förbipasserande bilar för att registrera trafikskatt automatiskt [6]. Det hålls flera kurser som behandlar ämnet, bland annat på KTH [8].

För en dator kan bildtolkning vara ett svårt problem, även då objektet eller informationen som söks är känd på förhand. Svårigheten beror på mängden information som finns i bilden, men också hur lika det eftersökta objektet är gentemot andra "störande" objekt som kan förekomma.

Ämnet har intresserat oss som är författare till detta dokument under flera år. Bildigenkänning är ett stort område som har känts ogripbart att starta ett eget projekt i, utan en väl avgränsad uppgift att lösa. Det är tack vare ett verktyg på internet som idén till projektet uppstod. Verktöget heter Cube Explorer [7] och är ett program som används för att hitta lösningar till Rubiks kub. Programmet erbjuder flera sätt att ange hur kuben är blandad men inget sätt är särskilt snabbt och smidigt. Programmet har en funktion för att använda webbkamera för att avläsa kubens tillstånd. Dock krävs ett foto på varje sida av kuben i en viss förutbestämd ordning. Kuben måste vara placerad så att färgerna läses på rätt ställen. För att färgerna ska tolkas rätt behöver programmet kalibreras.

Vår idé bygger på en mer automatiserad igenkänning och innebär att ta fram kubens tillstånd från två foton utan färgkalibrering. På varje foto kan tre av kubens sex sidor vara synliga. Datorns uppgift är att hitta kuben och bestämma dess tillstånd endast utifrån fotografierna. Se figur 1.1.



Figur 1.1 Programmets uppgift.

I det här dokumentet får läsaren ta del av hur datorprogrammet kan bestämma tillståndet på en Rubiks kub utifrån endast två fotografier. Varje fotografi visar upp tre av kubens sex sidor så att färginformationen om alla etiketter är tillgänglig.

Svårigheten med problem av den här typen är att informationen är diffus. Ett fotografi är inte strukturerad information då det inte finns en mall för hur informationen skall tolkas. Informationen behöver struktureras och filtreras för att bli användbar.

En sådan process behöver göra mätningar, antaganden, gissningar och rimlighetsbedömningar för att behandla felaktig, missvisande, irrelevant, bristfällig eller saknad information. Data måste kompletteras och filtreras på sin väg genom programmet för att nå ett slutgiltigt resultat.

För att lösa problemet används kunskap om objektet vi söker. En Rubiks kub består till stor del av fyrkantiga etiketter. På varje foto förväntas att det finns tre sidor med nio etiketter vardera. Etiketterna på varje sida liknar varandra till form och storlek. De är enfärgade och framträder mot kubens bottenfärg och omgivning. Etiketterna är väl strukturerade i förhållande till varandra och är positionerade i rader och kolumner för varje sida.

Etiketterna kan bara ha sex olika färger och måste vara uppdelade på olika bitar. Varje mittenetikett för varje sida har en egen färg som hör ihop med åtta andra etiketter på kubens.

Eftersom etiketterna har en central roll i problemet är ett första steg att identifiera dessa i bilden. När de är funna kan processen gå vidare med att sortera dem och bestämma deras färger.

Fysiska restriktioner hos kubens ger möjligheter att guida programmet till rätt beslut även om tidigare slutsatser varit felaktiga. Det gäller att ständigt vara kritisk mot informationen som programmet hanterar. Informationen kvalitetsbedöms kontinuerligt. Den säkraste informationen får bestämma framför information som är mer osäker.

Programmet lyckas bra med sin uppgift så länge fotografierna har rimlig kvalitet. Om fotografierna är så suddiga att kanterna mellan etiketterna blir otydliga får programmet svårt att skilja ut etiketterna. Om allt för många etiketter saknas misslyckas programmet med sin uppgift.

2. Programmets flöde

Målet med det första steget är att identifiera polygoner som kan utgöra etiketter på kuben i bilden. Detta görs genom att identifiera slutna ytor med kantigenkänning. En kant upptäcks genom att jämföra färgskillnaden mellan punkter i bilden.

I nästa steg granskas alla polygoner som hittades i steget ett. Polygonerna tilldelas en sida på kuben som den potentiellt kan utgöra en etikett för. Polygoner i bilden som verkar lämpliga väljs som etiketter.

I det tredje steget identifieras tre grupper av etiketter som bildar var sin sida på kuben. Etiketterna som ingår i varje grupp tilldelas rad och kolumn.

Eftersom det kan saknas etiketter behövs en metod för att välja var färgavläsningen ska ske för saknade etiketter. Detta görs i steg fyra genom att anpassa ett koordinatsystem per sida till de funna etiketterna och avläsa färgerna där koordinatsystemet anger att etiketterna finns.

I steg fem bearbetas färgerna så att de lättare kan jämföras med varandra. De justerade färgerna sparas i en struktur där kubens alla etiketter är sorterade efter position.

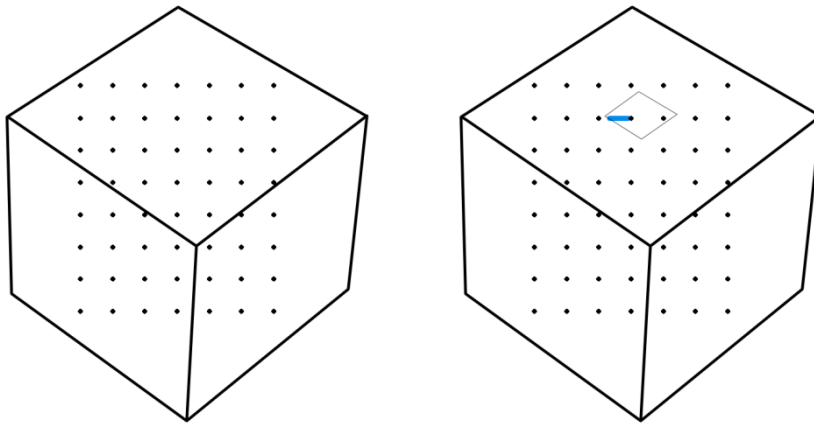
Under steg sex tilldelas varje etikett en bokstav som representerar dess färg. Etiketterna i strukturen paras ihop med andra etiketter av samma färg och får samma bokstav. Detta görs för att kunna tilldela alla etiketter av samma färg en sida där de hör hemma då kuben är löst.

Slutligen när kubens tillstånd är bestämt uttrycks tillståndet i form av en textsträng som enkelt kan bearbetas för att koppla ihop svaret med andra system.

2.1 Hitta polygoner i fotografierna

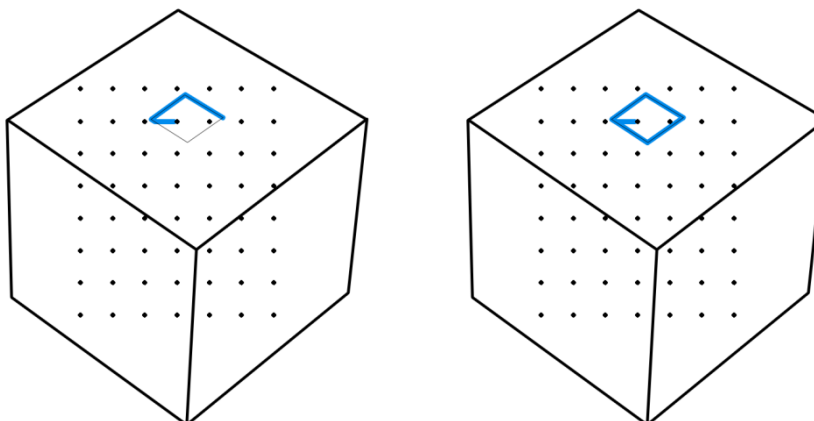
Målet med det första steget är att identifiera polygoner som kan utgöra etiketter på kuben i varje bild. Detta görs genom att identifiera slutna ytor med kantigenkänning. En kant upptäcks genom att jämföra färgskillnaden mellan punkter i bilden.

I bilden placeras ett antal sökkoordinater där letandet efter etiketter skall påbörjas. Algoritmen börjar i en koordinat och letar sig i en rak riktning tills den upptäcker en kant i bilden. Kanten upptäcks genom att jämföra färgskillnaden mellan den aktuella positionen och omgivande positioner. Om skillnaden är stor har algoritmen hittat en kant [10]. Figur 2.1.1 visar hur de förutbestämda sökkoordinaterna är placerade i förhållande till kuben i bilden. Till höger i figuren visas hur kantsökningen startar i en sökkoordinat och hittar en kant för en illustrativ etikett.



Figur 2.1.1 Kantsökningens startskede.

När den första kanten har upptäckts så registreras positionen för upptäckten. Algoritmen följer sedan kanten tills kantvandringen kommer tillbaka till samma position där kanten hittades. Detta illustreras i figur 3.1.2. Det inringade området bildar en polygon vars area, masscentrum och andra parametrar beräknas under själva inringningen. Vi återkommer till innebörden och nyttan av dessa parametrar längre fram.

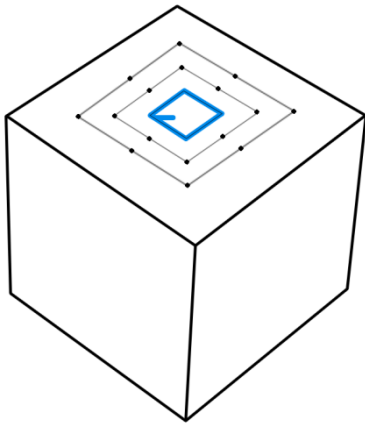


Figur 2.1.2 Kantvandring.

Programmet förbrukar en sökkoordinat i varje iteration och stannar först när alla sökkoordinater har undersökts. För varje yta som identifieras i bilden så adderas nya sökkoordinater att undersöka.

För att samma polygon inte ska hittas flera gånger ritas varje polygon i en bild. Om den första kanten som stöts på redan finns ritad i bilden avbryts sökningen. Eftersom det finns ett begränsat antal ytor i bilden kommer sökningen bli klar.

Centrumetiketterna på kuben är de etiketter som har flest etikettgrannar. De nya sökkoordinaterna i bilden placeras så alla övriga etikettgrannar upptäcks om den upptäckta etiketten skulle var en centrum bit. Figur 2.1.3 visar hur sökkoordinaterna läggs till runt den hittade etikettens kanter och diagonaler så de sannolikt hamnar innanför andra etiketter i bilden.

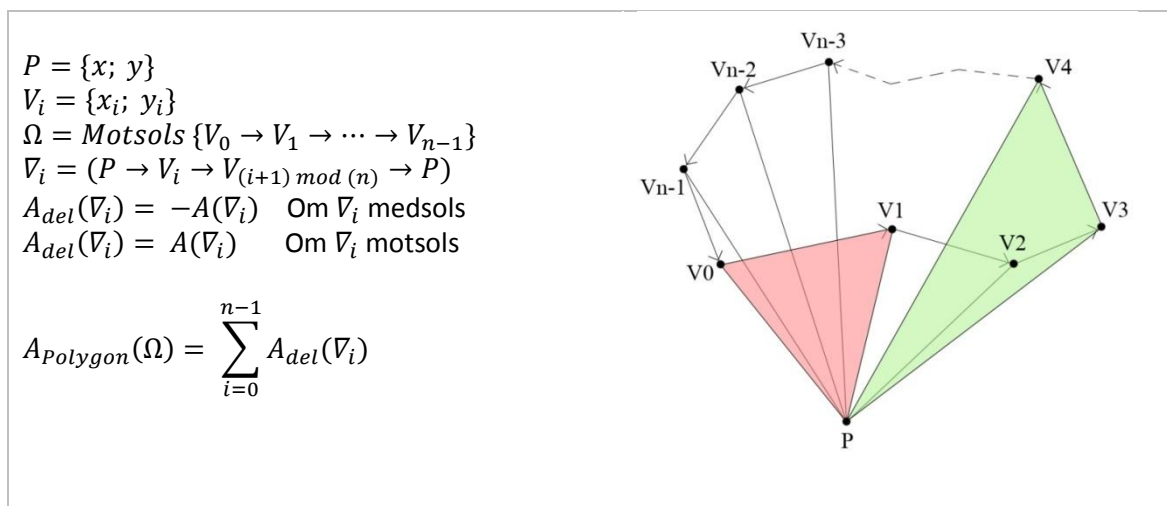


Figur 2.1.3 Nya sökkoordinaterna placeras i bilden.

2.1.1 Areaberäkning av en polygon

Arean för en identifierad polygon är hjälpsam för att avgöra ifall polygonen är en potentiell etikett eller inte. Polygoner med areor som är för små eller stora utgör sannolikt inte någon etikett. Arealen är dessutom viktig för att identifiera polygonens typ och avgöra vilken sida den eventuellt utgör en etikett för.

Ett enkelt sätt att beräkna ytan för en polygon är att dela upp polygonen i trianglar och summera över trianglarnas area [3]. Trianglarna skapas av en referenspunkt P och två på varandra följande noder V_i och V_{i+1} i polygonen. Om triangelns riktning $P \rightarrow V_i \rightarrow V_{i+1} \rightarrow P$ är medsols för en polygon där $V_0 \rightarrow V_1 \rightarrow \dots \rightarrow V_{n-1} \rightarrow V_0$ är motsols så räknas arean av triangeln negativt och i annat fall positivt. Beräkningen av arean kan genomföras i samband med kantvandringen då polygonen upptäcks. Skärningspunkterna mellan pixlarna längs med kantvandringen bildar polygonens hörn.



Figur 2.1.4 Areaberäkning av en polygon. Den röda triangeln ∇_0 bildar en negativ area. Den gröna triangeln ∇_3 bildar en positiv area.

2.1.2 Masscentrumberäkning för en yta

Masscentrum för en yta beskriver ytans tyngdpunkt, som om ytan hade en jämnt fördelad massa. Tyngdpunkten används som positionsbestämning för polygonen, vilket är praktiskt då färgavläsningen skall genomföras.

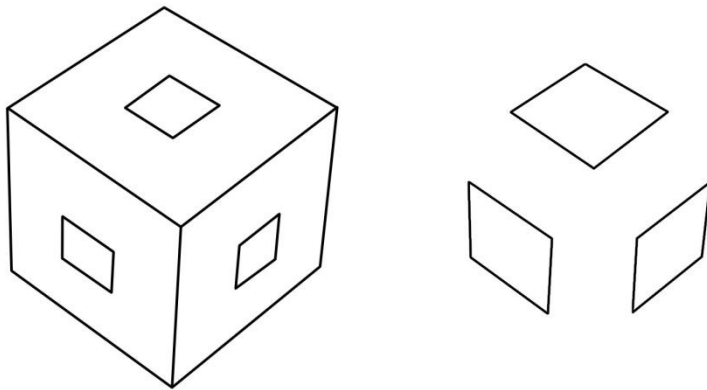
Vi beräknar masscentret genom att dela upp ytan i trianglar på samma sätt som tidigare. För varje triangel beräknas x- respektive y-koordinaten av triangelns tyngdpunkt från medelvärdet av triangelns hörn [9]. Masscentrum för hela polygonen i x-led bestäms genom att summera över produkten mellan triangelns tyngdpunkt i x-led och arean för triangeln och sedan dela med den totala arean för polygonen. Beräkningen i y-led sker på liknande sätt.

Programmet beräknar masscentrum för x- och y-led i samband med beräkningen av polygonens area. Trots att vissa koordinater viktas med arean för trianglar som sträcker sig utanför polygonen ges rätt resultat. Faktum är att vissa produkter under summeringen är negativa så det hela jämnar ut sig. En x-koordinat som multipliceras med en för stor area balanseras av en annan produkt med en negativ area.

2.2 Kategorisering av etiketter

I det här steget granskas alla polygoner som hittades i förra steget. Polygonerna tilldelas en sida på kuben som den potentiellt kan utgöra en etikett för. Polygoner i bilden som verkar lämpliga väljs som etiketter.

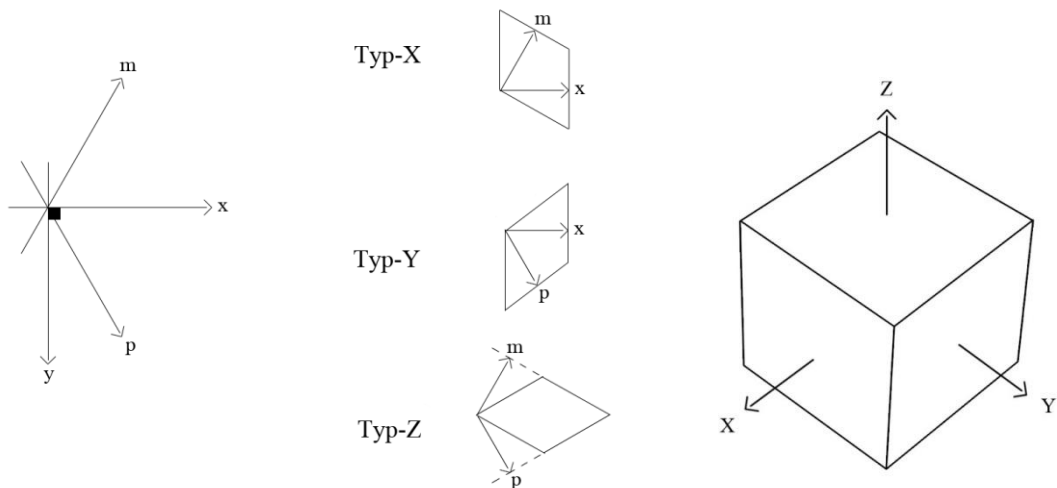
I det förra steget hittade programmet en mängd polygoner i bilden som alla är potentiella etiketter. För att avgöra vilka av polygonerna som är etiketter jämförs polygonens yta med ytan av en idealisk etikett. En idealisk etikett kan beskrivas av en parallelogram vars form är unik för en sida på kuben. Den unika formen gör det möjligt att jämföra vilken idealisk form en hittad polygon är mest lik. En polygon med någon av de identifierade formerna kommer att refereras till som en etikett av en viss typ. De olika formerna visas i figur 2.2.1 nedan.



Figur 2.2.1 Idealiskt formade etiketter.

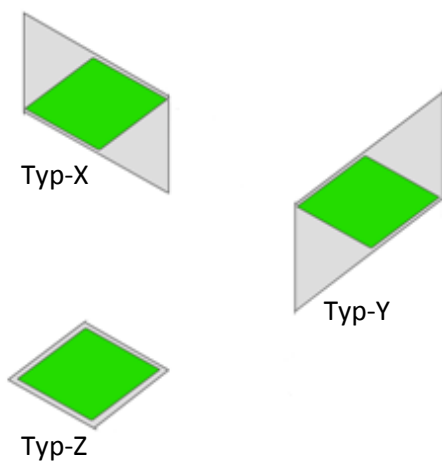
Formen identifieras genom att anpassa olika parallelogram så tätt som möjligt runt polygonen. Vinklarna på parallelogrammerna är förutbestämda och motsvarar vinklarna hos de ideala etiketterna.

I avsnitt 2.1 under kantigenkännings genomförande lagrades ett antal parametrar om varje polygon. Förutom arean och masscentrum lagras åtta andra parametrar som ligger till grund för anpassningen av parallelogrammerna. De åtta parametrarna heter $xMin$, $xMax$, $yMin$, $yMax$, $pMin$, $pMax$, $mMin$, $mMax$. Parametrarna är min- och maxvärden i fyra koordinatriktningar som svarar mot vinklarna i de ideala etiketterna. Parametrarna bestäms då kantigenkänningen i avsnitt 2.1 vandrar runt kanten på ytan av en potentiell etikett. För varje pixel som kantigenkänningsalgoritmen besöker justeras min och max för de fyra riktningarna ifall det behövs. Koordinatriktningarna visas i figur 2.2.2 nedan.



Figur 2.2.2 Koordinataxlar och typer.

Koordinataxlarnas lutningar utgörs av förbestämda konstanter i programmet. Lutningskonstanterna är anpassade så de bildar parallelogram som efterliknar etiketter i en idealisk bild. Det är informationen om vilka två par av min- och max-värden som avslöjar på vilken sida den potentiella etiketten hör hemma. Polygonens form kategoriseras genom att välja den parallelogram som ger minst area och därför bäst anpassning. Figur 2.2.3 visas en polygon som anpassas bäst av en parallelogram längs med p- och m-axlarna. En polygon med någon av de identifierade formerna kommer att refereras till som en etikett av en viss typ. Om en rektangel anpassar polygonen bäst får polygonen ingen typ utan slängs istället bort. De olika typerna visas i figur 2.2.2.



Figur 2.2.3 Anpassning av parallelogrammer.

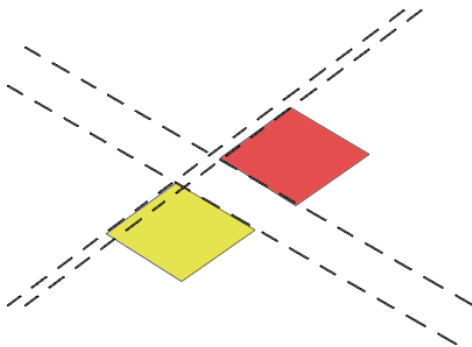
2.3 Identifiering av sidornas etiketter

I det här steget identifieras tre grupper av etiketter som bildar var sin sida på kuben. Etiketterna som ingår i varje grupp tilldelas rad och kolumn.

Alla polygoner som har hittats och blivit tilldelade en typ behöver inte vara etiketter. För att välja ut en grupp av etiketter som tillsammans bildar en sida på kuben grupperas de ytor som ligger nära varandra och är av samma typ. Etiketterna delas först in efter typ varefter varje grupp av etiketter av samma typ behandlas för sig. Varje yta identifierar vilka andra ytor som ligger så nära i bilden att de kan vara etikettgrannar. Avgörandet om närhet bygger på en jämförelse av två etiketters max- och minvärden för deras karaktäriserande koordinataxlar.

Med avseende på koordinataxeln p anses etiketterna A och B vara grannar om något av absolutbeloppen nedan är tillräckligt litet. Båda karaktäriserande koordinataxlarna måste uppfylla detta samband för att räknas som grannar. Figur 2.3.1 visar två etiketter som är grannar.

$$|A.pMax - B.pMax| \text{ eller } |A.pMax - B.pMin| \text{ eller } |A.pMin - B.pMax| \text{ eller } |A.pMin - B.pMin|$$



Figur 2.3.1 Två grannar

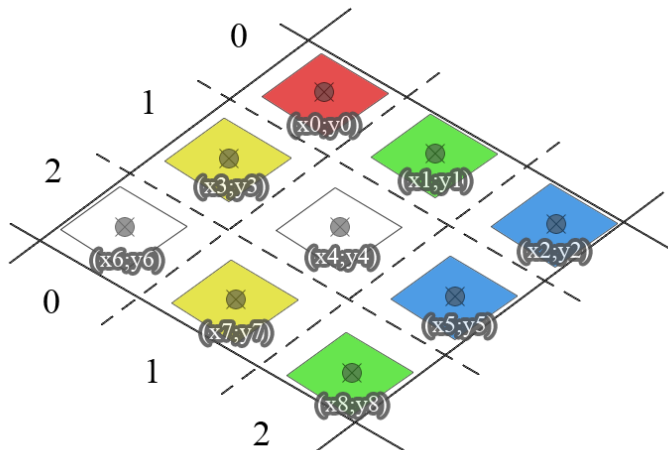
När alla ytor har identifierat sina grannar kan relationen mellan alla ytor beskrivas av en graf. I grafen representerar noder etiketter och kanter grannar. Den största komponenten i grafen beskriver det största sammanhängande grannskapet av etiketter. Om kuben dominerar i fotografiet är den största komponenten av varje typ de etiketter som representerar en sida. Även om störande objekt i bilden är orsak till skapandet av falska etiketter måste flera av dessa ligga nära varandra för att bilda en större komponent än den komponent som utgörs av riktiga etiketter. Dock kan falska etiketter komma med i anslutning till de riktiga etiketterna.

När den största komponenten har identifierats reduceras antalet etiketter i komponenten till nio. Detta görs för att en sida på kuben inte har fler än nio etiketter. Ett större antal än nio etiketter innebär falska etiketter. Om antalet etiketter är mindre än nio finns det saknade etiketter som kompenseras för i nästa steg.

Reduceringen av etiketter kan göras på flera sätt. Vi har valt att favorisera de etiketter med störst area. Andra sätt att välja etiketter som skall vara kvar är t.ex. att jämföra hur väl ytan är anpassad till etikettens typ.

När antalet etiketter har reducerats ringas hela komponenten in av två par av tangentlinjer med samma lutning som etiketternas karaktäriserande tangentlinjer. Figur 2.3.1 visar hur det inringade

området i bilden delas i nio lika stora områden så formen liknar en sida på kuben. Varje område där en etikett bör finnas indexeras med rad och kolumn. Etiketterna i den identifierade komponenten går igenom och rad och kolumn bestäms. Bestämningen görs genom att identifiera i vilket område etiketten har sitt masscentrum. Resultatet beskriver sambandet mellan rader, kolumner, x- och y-värden i fotot för varje sida på kuben som en tabell med en rad för varje utvald etikett.



Figur 2.3.1 Indelning i rader och kolumner.

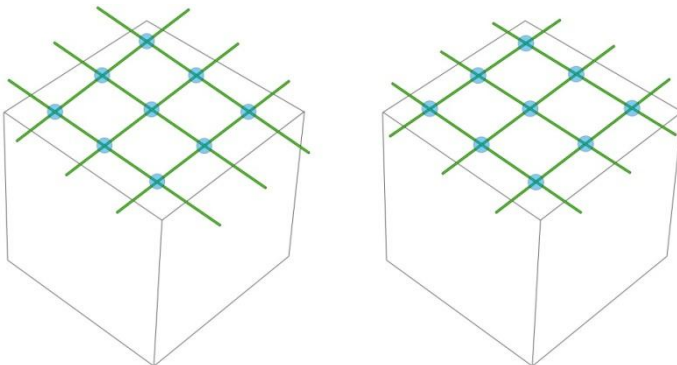
Om en etikett är ungefär lika bred som två tredjedelar av sidans område gissar programmet att etiketten är en sammanslagning av två riktiga etiketter. Kolumnens koordinat blir därmed ett bråk, antingen $\frac{1}{2}$ eller $\frac{3}{2}$, som beskriver om etiketten är en sammanslagning mellan mitten och vänster kolumn eller mitten och höger kolumn. Är etiketten bred som tre tredjedelar av sidan behövs ingen specialbehandling eftersom mitten av etiketten hamnar rätt ändå. Detta test görs även för indelning i rader. Sammanslagningar av etiketter innebär att information förloras men på detta sätt kan mycket av informationen användas ändå.

2.4 Anpassning av koordinatsystem

Eftersom det kan saknas etiketter från tidigare steg behövs en metod för att välja var färgavläsningen ska ske för saknade etiketter. Detta görs genom att anpassa ett koordinatsystem per sida till de funna etiketterna och avläsa färgerna där koordinatsystemet anger att etiketterna finns.

I det här steget bestäms alla etiketter färgavläsningskoordinater utifrån de etiketter som i förra steget blev tilldelade rader och kolumner. Detta görs genom att skapa ett koordinatsystem för varje sida där etiketternas positioner kan beräknas. Koordinatsystemet är viktigt ifall någon etikett på någon sida saknas. Etiketter som saknas har inget masscentrum och därmed ingen koordinat för färgavläsning. Genom att anpassa ett koordinatsystem till de etiketter som har identifierats blir färgavläsningen för de saknade etiketterna möjlig.

Varje sida på kuben har ett koordinatsystem med samma koordinataxlar som sidans tillhörande parallelogram vid kategoriseringen i avsnitt 3.2. Varje sidas koordinatsystem är förutbestämt i programmet. Det förutbestämda koordinatsystemet tar inte hänsyn till perspektivistiska förkrympningar i bilden, vilket gör att passformen är dålig om bilden är tagen på kort avstånd eller med lite sned vinkel. Figur 2.4.1 visar två likadana kuber med två olika koordinatsystem för att ange etiketternas centrum. Det vänstra koordinatsystemets axlar är parallella med varandra, men följer inte kubens sidor särskilt bra. Det högra koordinatsystemet är inte parallellt men följer kubens sidor bra.



Figur 2.4.1 Olika koordinatsystem.

För att skapa ett koordinatsystem som tar hänsyn till förkrympningarna måste linjerna mellan etiketternas masscentrum kunna ha olika riktningar. Linjer i fotot kan tyckas se parallella ut utan att vara det. I bilden till vänster är linjerna helt parallella med varandra och därför inte helt anpassade till kubens sidor. Problemet löses genom att anpassa kvadratiska funktioner till de funna etiketternas masscentrum. Två funktioner används, en för x-koordinat och en för y-koordinat i fotot.

Funktionerna tar emot rad och kolumn för en etikett som parametrar och returnerar var etikettens masscentrum befinner sig. Om funktionerna utgörs av kvadratiska polynom har polynomen sex konstanter. De sex konstanterna ger möjlighet att beskriva tre godtyckliga punkter på en kurva även om rad eller kolumn är fixerad. Funktionerna visas nedan.

$$f_x(x, y) = a_0 + a_1i + a_2j + a_3ij + a_4i^2 + a_5j^2$$

$$f_y(x, y) = b_0 + b_1i + b_2j + b_3ij + b_4i^2 + b_5j^2$$

Den första ekvationen tar emot parametrarna x och y som kolumn och rad för etiketter och returnerar x -koordinaten för etiketten i bilden. Den andra funktionen bestämmer y -koordinaten på samma sätt. Konstanterna a_0 till a_5 och b_0 till b_5 måste bestämmas innan koordinatsystemet kan användas. Varje identifierad etikett har en masscentrumskoordinat som beskriver var i bilden etiketten finns. Dessa x och y -koordinater utgör funktionsvärdena för de två funktionerna. Samma etikett har identifierats med kolumn och rad index i ett tidigare steg. Etikettens kolumn och rad index utgör parametrar till funktionerna.

Varje ekvation har sex okända konstanter vilket innebär att minst sex etiketter på en kub-sida måste identifieras för att ekvationssystemet skall ha en lösning. Om fler än sex etiketter har identifierats så blir systemet överbestämt och lösas med fördel med minsta-kvadrat-anpassning [11]. Om färre än sex etiketter identifieras får systemet flera lösningar. Bland lösningarna väljs då lösningen där de sista konstanterna är noll. Genom att sätta de sista konstanterna till noll reduceras graden av polynomet och linjerna böjer sig inte. Figur 2.4.2 och 2.4.3 visar hur ekvationssystemen ställs upp när alla etiketter på en sida har hittats.

När koordinatsystemen för varje sida på kuben har bestämts går det att beräkna alla etiketter bildkoordinater. Det går även att hitta koordinater till kanterna mellan etiketter, vilket kan användas i nästa steg.

$$\begin{aligned}
 x_0 &= f_x(0,0) = a_0 \\
 x_1 &= f_x(1,0) = a_0 + a_1 + a_4 \\
 x_2 &= f_x(2,0) = a_0 + 2a_1 + 4a_4 \\
 x_3 &= f_x(0,1) = a_0 + a_2 + a_5 \\
 x_4 &= f_x(1,1) = a_0 + a_1 + a_2 + a_3 + a_4 + a_5 \\
 x_5 &= f_x(2,1) = a_0 + a_1 \cdot 2 + a_2 + 2a_3 + 4a_4 + a_5 \\
 x_6 &= f_x(0,2) = a_0 + 2a_2 + 4a_5 \\
 x_7 &= f_x(1,2) = a_0 + a_1 + 2a_2 + a_3 \cdot 2 + a_4 + 4a_5 \\
 x_8 &= f_x(2,2) = a_0 + 2a_1 + 2a_2 + 4a_3 + 4a_4 + 4a_5
 \end{aligned}$$

Figur 2.4.2 Ekvationssystemet för f_x .

$$\begin{aligned}
 y_0 &= f_y(0,0) = b_0 \\
 y_1 &= f_y(1,0) = b_0 + b_1 + b_4 \\
 y_2 &= f_y(2,0) = b_0 + 2b_1 + 4b_4 \\
 y_3 &= f_y(0,1) = b_0 + b_2 + b_5 \\
 y_4 &= f_y(1,1) = b_0 + b_1 + b_2 + b_3 + b_4 + b_5 \\
 y_5 &= f_y(2,1) = b_0 + b_1 \cdot 2 + b_2 + 2b_3 + 4b_4 + b_5 \\
 y_6 &= f_y(0,2) = b_0 + 2b_2 + 4b_5 \\
 y_7 &= f_y(1,2) = b_0 + b_1 + 2b_2 + b_3 \cdot 2 + b_4 + 4b_5 \\
 y_8 &= f_y(2,2) = b_0 + 2b_1 + 2b_2 + 4b_3 + 4b_4 + 4b_5
 \end{aligned}$$

Figur 2.4.3 Ekvationssystemet för f_y .

2.5 Identifiering av etiketternas färger

I föregående steg fastställdes alla etiketter bildkoordinater. I detta steg läses alla individuella etiketter färger. Färgerna bearbetas så att de lättare kan jämföras i nästa steg.

Färgen för en etikett bestäms av ett medelvärde från färgavläsningar runt etikettens bildkoordinater. En färg identifieras av sitt RGB-värde [1], [13]. Genom att avläsa flera färgvärden i bilden och beräkna ett medelvärde kan en viss grad av brus kompenseras. Färgavläsningen i bilden resulterar i tre värden som anger mängden av Rött, Grönt och Blått. Medelvärdet för etikettens färg bestäms av medelvärdet för mängderna rött, grönt och blått var för sig. Se figur 2.5.1 nedan.

$$R = \frac{\sum_{i=0}^n r_i}{n}$$
$$G = \frac{\sum_{i=0}^n g_i}{n}$$
$$B = \frac{\sum_{i=0}^n b_i}{n}$$

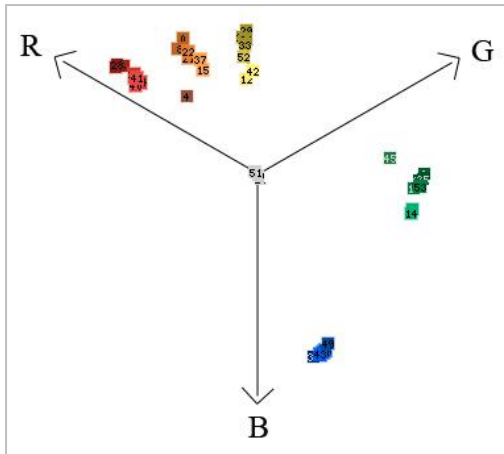
Figur 2.5.1 Medelvärdesberäkning av färg.

När färgerna för alla individuella etiketter är avlästa skall etiketter med samma färg identifieras. Detta är inte ett helt trivialt problem då två lika färgade etiketter kan resultera i två skilda RGB-värden. En av de påverkande faktorerna till fenomenet är ljusets olika intensitet på olika sidor av kuben. Till exempel då en ljuskälla belyser någon sida starkt och/eller lämnar någon sida i skugga.

En annan påverkande faktor är ljus av olika färg som faller mot olika sidor på kuben. Till exempel då kuben belyses med flera olika färgade ljuskällor eller färgas av inkommande ljus från färgstarka objekt i omgivningen.

För att slippa problemet med olika ljusintensitet projiceras informationen om ljusstyrka bort innan jämförelsen. RGB-systemet kan beskriva ett ortonormalt koordinatsystem [12] som spänner en tredimensionell färgrymd. I rymden kan en godtycklig färg beskrivas som en linjärkombination av R, G, B. Rymddiagonalen utgör en gråskala från vitt längs bort till svart i origo. Genom att projicera färgrymden till ett plan där rymddiagonalen fixeras till en punkt i mitten och vinklarna ut från rymddiagonalen bestämmer färgernas position försvinner all information om färgernas ljusstyrka.

Resultatet av projektionen till färgplanet beskriver vilken balans av R, G, B varje färg har. Det innebär att två färger med samma balans projiceras på samma punkt i färgplanet trots olika ljusstyrka. Om två färger med samma nyans har olika grad av mättnad skiljer endast avståndet från origo i färgplanet, riktningen ut från origo är densamma. Färger med hög mättnad hamnar långt ut från origo. Eftersom ljusstyrkan projiceras bort hamnar även mörka färger med relativt hög mättnad långt ut. Figur 2.5.2 visar ett exempel på etiketter projektion till färgplanet.

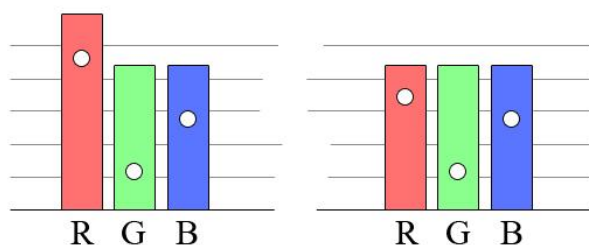


Figur 2.5.2 Etiketter på färgplanet.

Ett resultat av projektionen innebär att svarta och vita etiketter inte kan skiljas åt. En svart färg kommer att projiceras till planets origo, likaså en vit eller grå färg. Detta medför att detta nästa steg inte kommer fungera för kuber som har både svarta och vita etiketter. Däremot gör projektionen det enklare att avgöra vilka etiketter som har samma färg trots skillnad i nyans och ljusintensitet.

När olika sidor på kuben färgas av olika ljus behöver färgavläsningen korrigeras så den blir likvärdig för alla sidor. Färgen hos det infallande ljuset påverkar RGB-värdet vilket kan göra att etiketten ser ut att vara av en annan färg. Om det infallande ljuset för en sida är rödare än för de övriga sidorna kommer balansen hos färgerna vara olika. Alla etiketters RGB-värden på den rödaktiga sidan kommer ha högre andel R jämfört med etiketter av samma färg på de övriga sidorna.

För vita kuber kan detta kompenseras genom att avläsa vitbalansen för kubens olika sidor. Vitbalansen avslöjar vilka färger som dominerar i ljuset som belyser sidan. Hos en vit färg som belyses med vitt ljus är balansen mellan R, G, B precis lika. Om en vit yta belyses av rött ljus förändras balansen hos färgen så andelen rött dominerar. RGB-värdena från varje sida av kuben kan skalas om med skalningskonstanter som är anpassade så att en neutral referens på samma sida får en jämn balans. Skalningskonstanterna anger i vilken grad R, G, B skall skalas så färgtolkningen blir neutral. Staplarna i figur 2.5.3 visar balansen av rött, grönt och blått hos vit-referensen för en sida på kuben medan cirkelarna illustrerar färgbalansen hos en etikett. Den vänstra grafen visar mätvärden hos en rödbelyst sida innan kompensation, den högra efter kompensation.



Figur 2.5.3 Vitbalansering.

De neutrala referenserna bestäms utifrån medelvärdet av fyra färgavläsningar per kubsida i bilden. Avläsningen sker i hörnen mellan etiketterna på kuben där kubens bakgrundsfärg är synlig. Om bakgrundsfärgen på kuben är svart fungerar inte kompensationen eftersom den svarta färgen har för låga värden på R, G, B för att en neutral balans ska kunna avläsas.

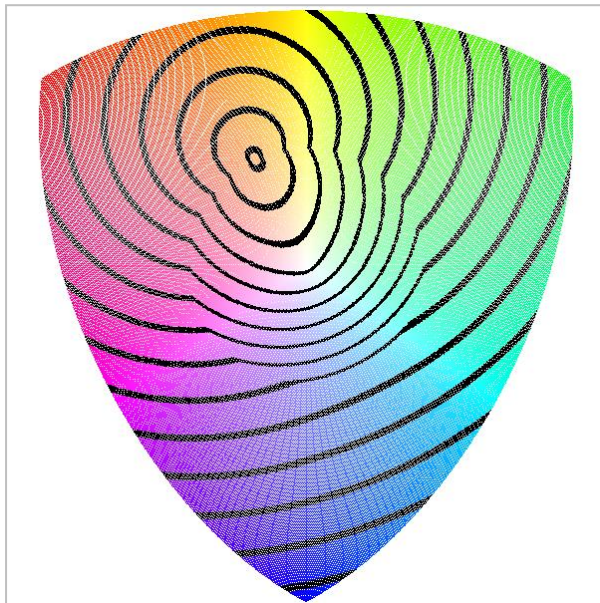
2.6 Hopparring av etiketter med samma färg

I de tidigare stegen har programmet arbetat fram en struktur där varje etikett på kuben finns representerad med sin justerade färg. I detta steg tilldelas varje etikett en bokstav som representerar dess färg. Etiketterna i strukturen paras ihop med andra etiketter av samma färg och får samma bokstav. Detta görs för att kunna tilldela alla etiketter av samma färg en sida där de hör hemma då kuben är löst.

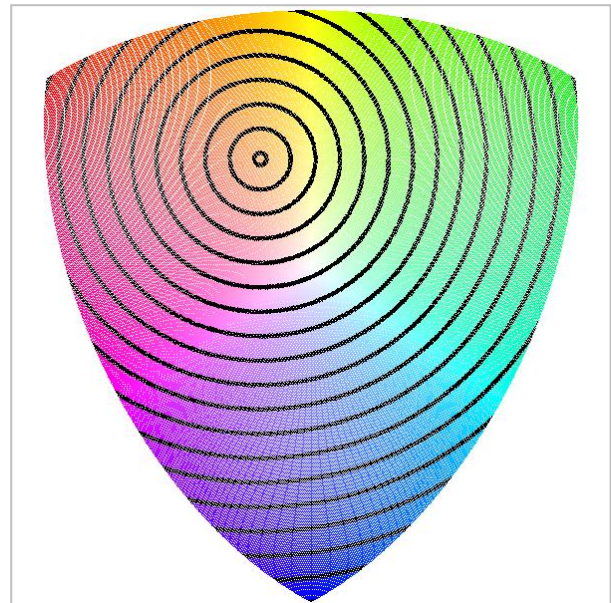
För att prioritera mellan möjliga hoppningar används något vi kallar för det kromatiska avståndet. Det kromatiska avståndet bestäms av det euklidiska avståndet mellan två färger i färgplanet minus skillnaden i avstånd till origo delat med en konstant. Se formel nedan.

$$K(x_1, y_1, x_2, y_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} - \frac{(\sqrt{x_1^2 + y_1^2} - \sqrt{x_2^2 + y_2^2})^2}{2,5}$$

Avdraget av skillnaden i avstånd till origo görs för att färger med samma vinkel gentemot koordinataxlarna skall favoriseras. Etiketter som hör ihop tenderar att variera i mättnad snarare än i nyans och rör sig därför närmare eller längre ifrån origo snarare än runtom. Favoriseringen illustreras med nivågrafer i figur 2.6.1 och kan jämföras med det euklidiska avståndet som illustreras i figur 2.6.2.



Figur 2.6.1 Kromatiskt avstånd i färgplanet.



Figur 2.6.2 Euklidiskt avstånd i färgplanet.

Hoppningen av etiketter innebär att jämföra två etiketters färger och besluta om dessa är hör ihop. Detta är ett problem i sig då färgavläsningen inte ger perfekt resultat. Det kan vara så att olika färger ligger nära varandra och det är svårt att veta vilka som egentligen hör ihop. För att göra smarta hoppningar av färger används kubens struktur och information om andra hoppningar för att minska risken för fel.

En restriktion hos kuben är att två etiketter som tillhör samma rörliga bit omöjligt kan ha samma färg. Om en sådan bit skulle finnas innebär detta att kuben har två lika färgade sidor. Vilket kuben inte har. Därför kan två etikettfärger omöjligt höra ihop ifall etiketterna tillhör samma bit på kubens. En annan restriktion hos kuben förbjuder bitar vars etikettfärger hör hemma på motsatta sidor av kubens. Det är inte tillåtet att en etikett A tilldelas en viss sida om dess motstående sidas färg redan finns på biten som A tillhör.

Vidare innehåller kubens inga dubletter av bitar. Det ger restriktionen att endast en tilldelning av varje unik kombination av två eller tre bokstäver kan göras på kubens alla bitar. Därför kan inte en etikett tilldelas en bokstav så att den tillsammans med de andra etiketterna på biten bildar en bit som redan finns representerad någon annan stans på kubens.

Vid hoppningen av etiketter utses referensfärger för varje sida på kubens. Referensfärgerna tilldelas färgen av centrumbitarnas etiketter. Programmet skapar en lista som innehåller alla övriga etiketter i sex upplagor. Varje upplaga hör ihop med en referensfärg och tilldelas ett kromatiskt avstånd gentemot referensfärgen. Listan sorteras med hänsyn till det kromatiska avståndet så den bästa matchningen hamnar först. Programmet går igenom listan och försöker para ihop etiketter med hänsyn till restriktionerna hos kubens. Om en tilldelning förbjuds på grund av restriktioner som uppstår från tidigare tilldelningar fortsätter genomgången med nästa element i listan. Metodiken gör det möjligt att utnyttja uteslutningsmetoden för osäkra tilldelningar då tidigare (säkrare) tilldelningar skapar så många restriktioner som möjligt. När programmet har tilldelat bokstäver till alla etiketter är tilldelningen klar och steget avslutas. Om programmet lyckas gå igenom hela listan utan att tilldela bokstäver till alla etiketter har tilldelningen misslyckats. Då utförs processen en andra gång. Den andra iterationen använder sig av referensfärger satta till de genomsnittliga färgerna av de etiketter som lyckades tilldelas i första iterationen. Om tilldelningen inte heller lyckas denna gång körs ännu en iteration med nya referensfärger. Men det finns en begränsning som gör att programmet ger upp och presenterar de etiketter som lyckades tilldelas efter ett visst antal iterationer.

2.6.1 Algoritm för ihop parning av etiketter med samma färg

In: Etiketter med färg, rad och kolumn

Ut: Kubens tillstånd

1. Låt mittenbitarnas färger vara referensfärger
2. Lista alla möjliga tilldelningar av etikett till referensfärg
3. Sortera möjliga tilldelningar efter kromatiskt avstånd
4. Så länge det går att göra fler tilldelningar
 - a. Gör bästa möjliga tilldelning
5. Om resultatet är dåligt
 - a. Uppdatera referensfärgerna till medelvärdet av tidigare tilldelningar
 - b. Börja om från 2

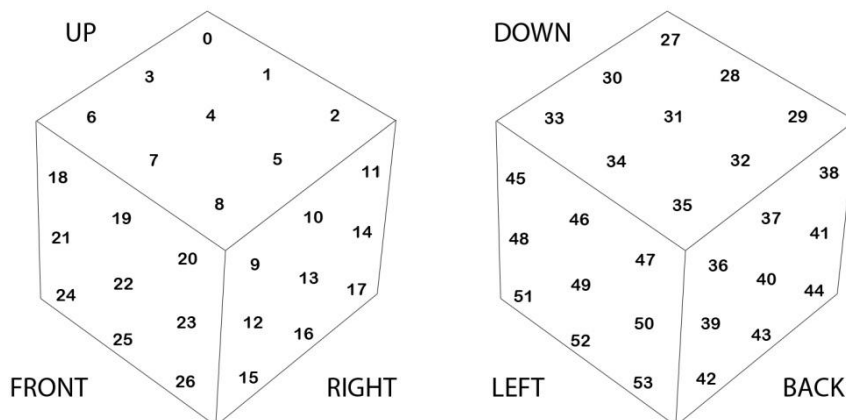
2.7 Formatera resultatet som en sträng

Det tidigare steget har bestämt kubens tillstånd. Alla etiketter med samma färg har en relation till varandra genom bokstaven de har blivit tilldelade. I detta steg formuleras tillståndet på ett användbart format och presenteras för användaren. Vi har valt att uttrycka tillståndet i form av en textsträng som enkelt kan bearbetas för att koppla ihop svaret med andra system.

Textsträngen innehåller sex olika bokstäver och är 54 tecken lång. Bokstäverna som används är *U, R, F, D, B* och *L* som står för *Up, Right, Front, Down, Back, och Left*, vilket motsvarar kubens sidor. Förra steget använder sig av samma bokstäver. Antalet tecken i strängen är detsamma som antalet etiketter på kuben. Varje etiketts bokstav beskriver på vilken sida den hör hemma när kuben är löst.

Textsträngen skapas genom att gå igenom alla etiketter i en förbestämd ordning och skriva ut bokstaven som identifierar på vilken sida etiketten hör hemma.

Figur 2.7.1 visar den bestämda etikett- och sidordningen för de två fotografierna som analyserats. Den vänstra kuben har sidorna *Up, Right* och *Front* där etiketter med index 0 – 26 finns angivna. Den högra kuben har sidorna *Down, Back* och *Left* där etiketter med index 27 – 53 finns angivna.



Figur 2.7.1 Etiketternas ordning.

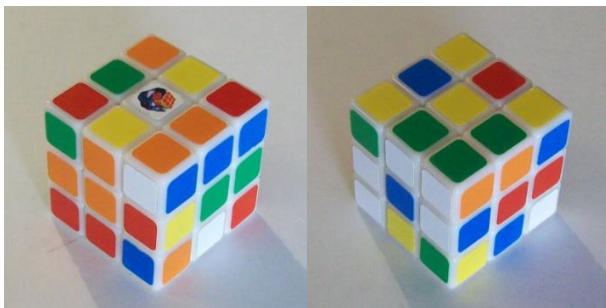
3 Resultat

I denna del kommer resultatet från flera programkörningar med olika foton att redovisas. Först redovisas resultatet av en programkörning och hur resultatet tolkas och kan användas. Sedan redovisas interna beteenden hos programmet som uppstår utifrån olika fotografier. Programmets prestanda kommer demonstreras utifrån fem olika par av foton.

Programmet tar emot två original foton från exempelvis en digitalkamera och levererar en textsträng som beskriver tillståndet. Resultatet kan exempelvis användas för att generera en lösning till kuben från det givna tillståndet som åskådliggörs av bilderna.

Figur 3.1 visar två beskurna foton som skickas till programmet. Programmet har inget krav på att bilderna skall vara beskurna. Resultatet som programmet ger ifrån sig ser ut så här:

```
FDBRUFBDLFLLDLRRFUBRFUFFBBBLDBDLDRDRRFFLLBBDLURUUULUUDR
```



Figur 3.1 Programmets indata.

För att illustrera hur resultatsträngen hör ihop med fotonerna kan etikettidentifieringen renderas av programmet. Programmet placerar ut information i bilden om etiketters index och vilken sida de hör hemma på. Figur 3.2 visar resultatet av programmet i bildform.



Figur 3.2 Resultatet i bildform.

Programmets prestanda demonstreras av fem olika par av fotografier som skickats till programmet. Det första exemplet har bra foton med tydliga kanter och färger. Det andra exemplet störs av reflektioner från en röd duk. Det tredje är taget på en svart kub med störande element närvarande. Exempel fyra och fem har lågupplösta och oskarpa foton.

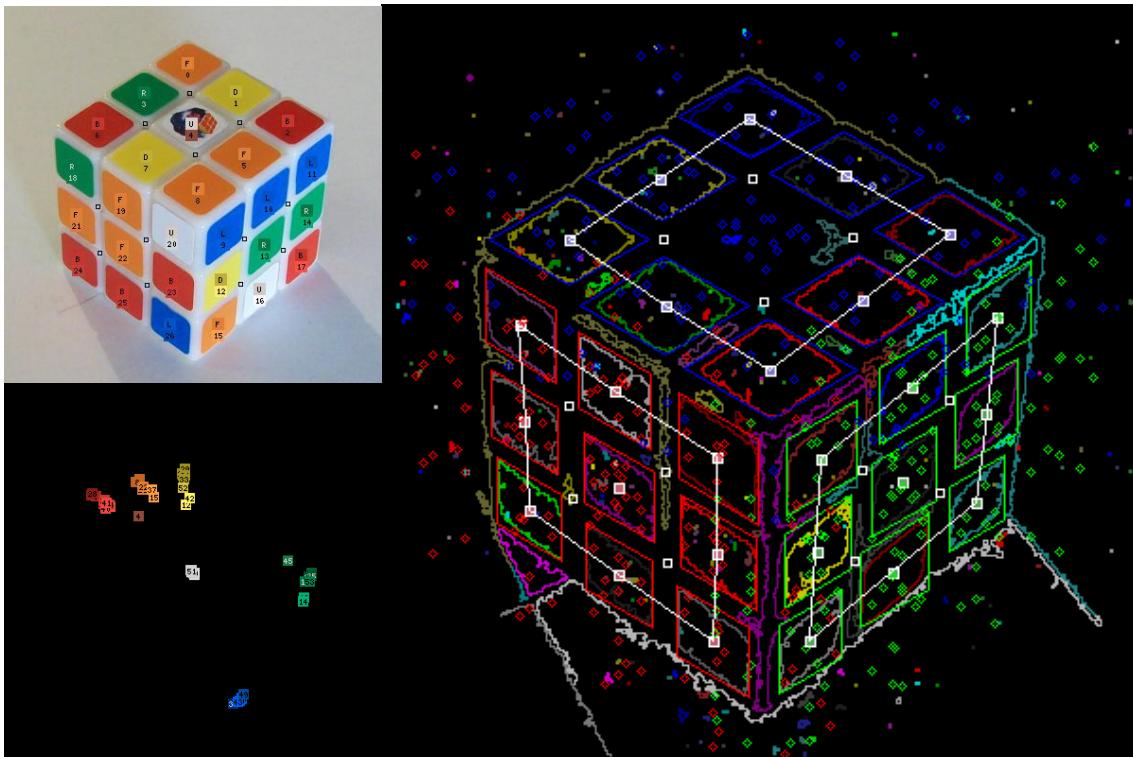
I varje exempel demonstreras resultatet för ett av fotografierna i paret som skickades till programmet. Exemplena består av tre renderade bilder som illustrerar det interna arbetet i programmet. Den första bilden uppe till vänster utgör ett beskuret foto där etiketterna har identifierats och markerats i fotot. Varje etikett har en bokstav som anger vilken sida etiketten tillhör. Bakgrundsfärgen för bokstaven utgör den genomsnittliga färgen för etiketter som tilldelats samma sida. Varje etikett har desutom en siffra som anger vilket index etiketten utgör på kuben. Indexvärdet har den vitbalanskorrigerade färgen för etiketten som bakgrund.

Den stora bilden till höger i exemplen visar resultatet av kantigenkänningen som linjer i olika godtyckliga färger. De ytor som har identifierats som etiketter har sina karaktäriserande parallelogrammer utritade runt sig. De olika typerna x, y och z avgör vilken av färgerna röd, grön och blå som parallelogrammen ritas i. De små färgade punkterna i bilden är romber som symboliserar tillagda sök koordinater. Tyngdpunkterna för de identifierade etiketterna som användes för att anpassa koordinatsystemet i avsnitt 2.4 är markerade med vita kvadrater. Resultatet av anpassningen syns som vita fyrkanter på varje sida av kuben. Avläsningspunkterna för vitbalanskorrigeringen syns som vita fyrkanter nära centrumbitarnas hörn.

Den tredje bilden visar båda fotografiernas etiketter utritad i färgplanet.

3.1 Första kuben

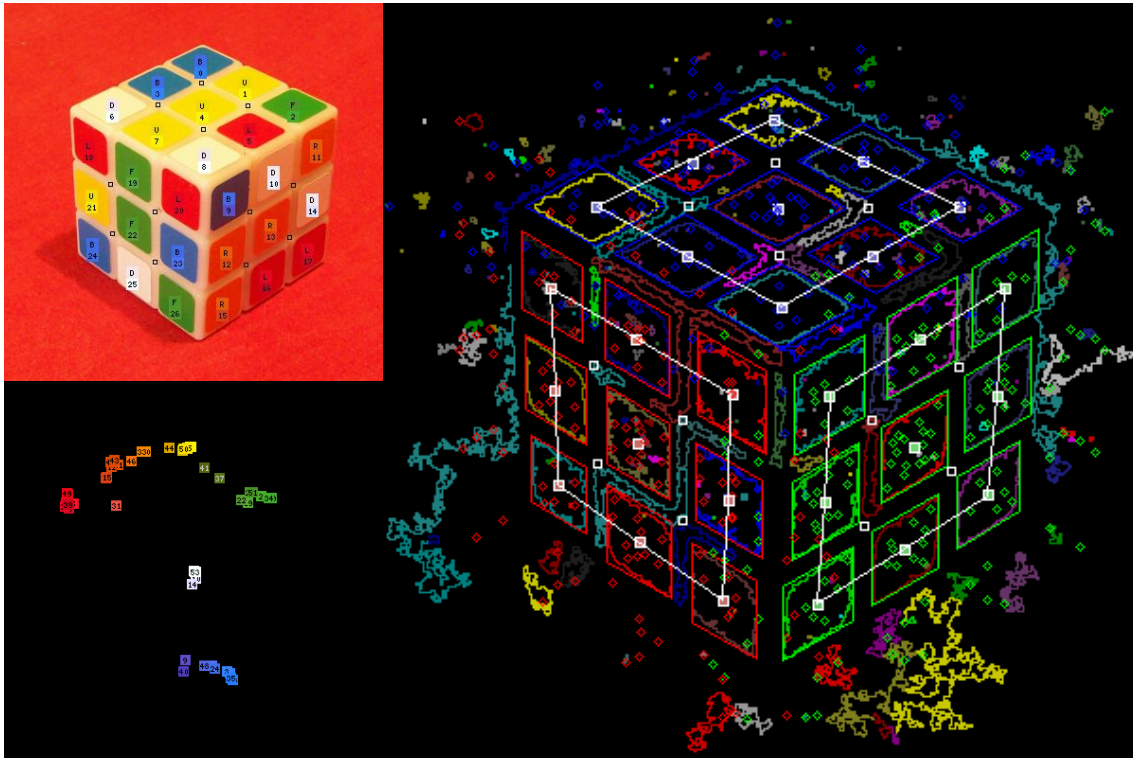
Den första kuben är fotograferad mot en vit bakgrund utan störande element. Den vita centrumbiten på ovasidan har en liten logo på sig. Steg ett hittar alla etiketter förutom den vita centrumbitens etikett. Steg två och tre kategoriserar och grupperar de omgivande etiketterna på ovasidan. Steg fyra anpassar koordinatsystemet så att den saknade etiketten på ovasidan ändå får koordinater. När färgerna läses i steg fem får den saknade etiketten en rödbrun färg. Trots detta lyckas steg sex med att bestämma vilken sida alla etiketter tillhör och resultatet blir korrekt.



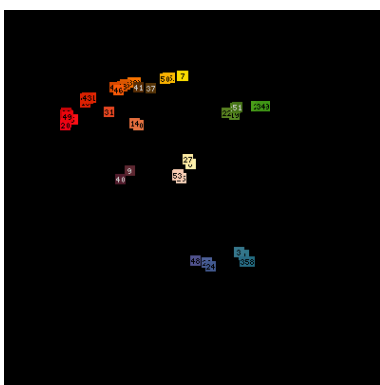
Figur 3.1.1 Resultat.

3.2 Andra kuben

Den andra kuben är fotad på en röd duk vilket gör att sidorna får olika värden på vit referenserna. Den högra sidan av kuben är kraftigt påverkad av det reflekterade röda ljuset. Vitbalanskorrigeringen i steg fem kompenserar så att de vita etiketterna läses som vita trots det färgade ljuset. Figur 3.2.2 visar färgplanet utan vitbalanskorrigering. Etikett 41 och 37 hamnar utan vitbalanskorrigering bland de orangea färgerna på färgplanet. Efter korrigering har de flyttas längre till höger så att steg sex klarar att tilldela alla sidor korrekt. På färgplanet syns att etiketternas färger är utspridda men inte överlappar varandra.



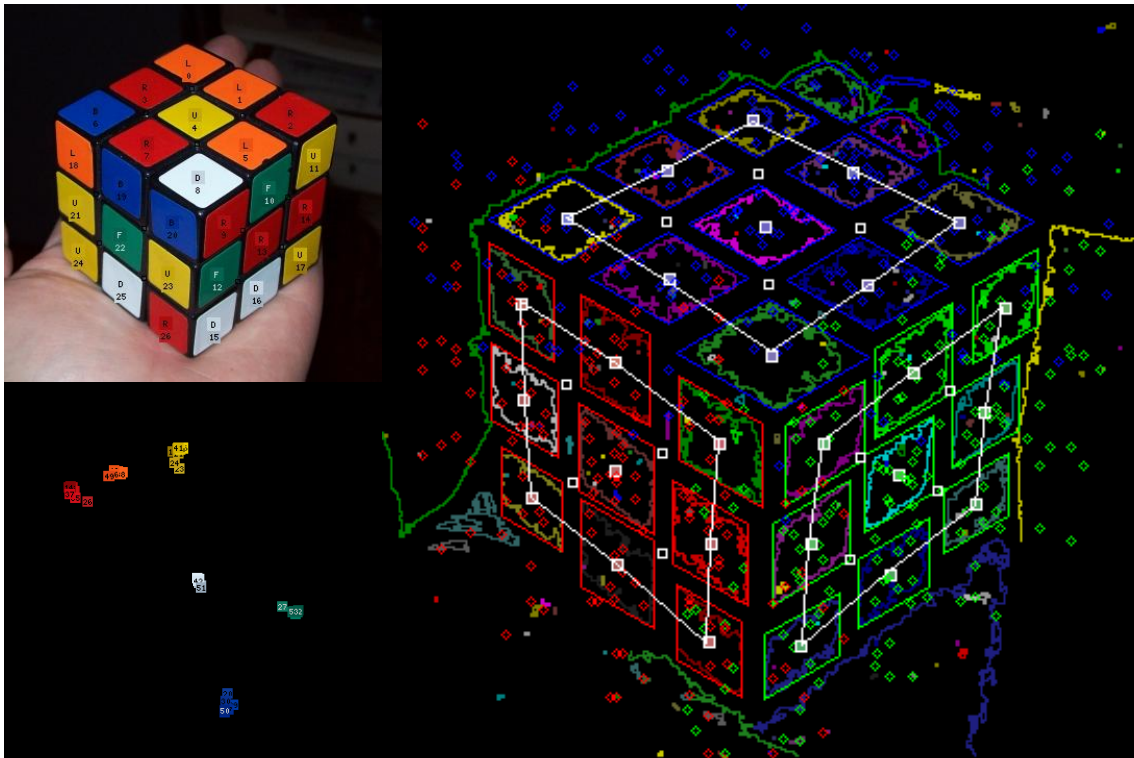
Figur 3.2.1 Resultat.



Figur 3.2.2 Färgplanet utan vitbalanskorrigering.

3.3 Tredje kuben

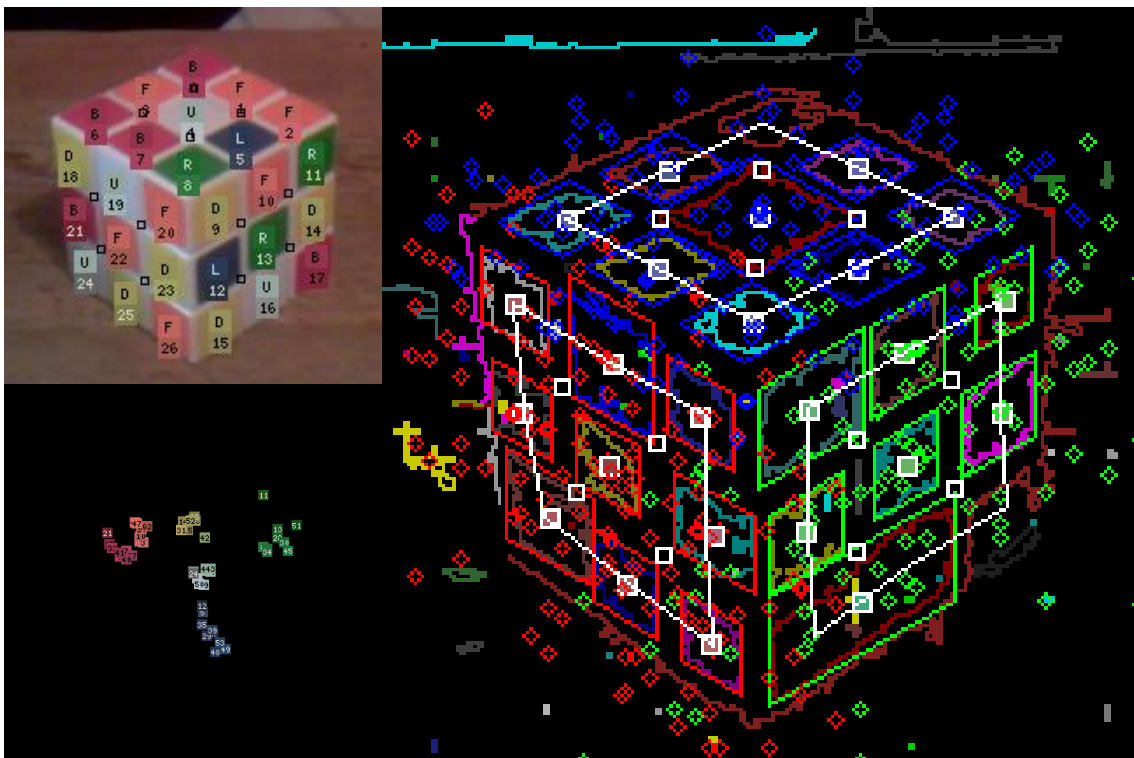
Den tredje kuben är en svart kub fotad med blixst, hållen i en hand med två fingertoppar som sticker upp bakom kuben. Fingertopparna hittas och klassificeras som etiketter i steg ett och två men sållas bort under steg tre då det finns nio andra etiketter i samma komponent som är större. Blixten gör att färgerna blir väldigt tydliga och sidorna tilldelas korrekt utan problem.



Figur 3.3.1 Resultat.

3.4 Fjärde kuben

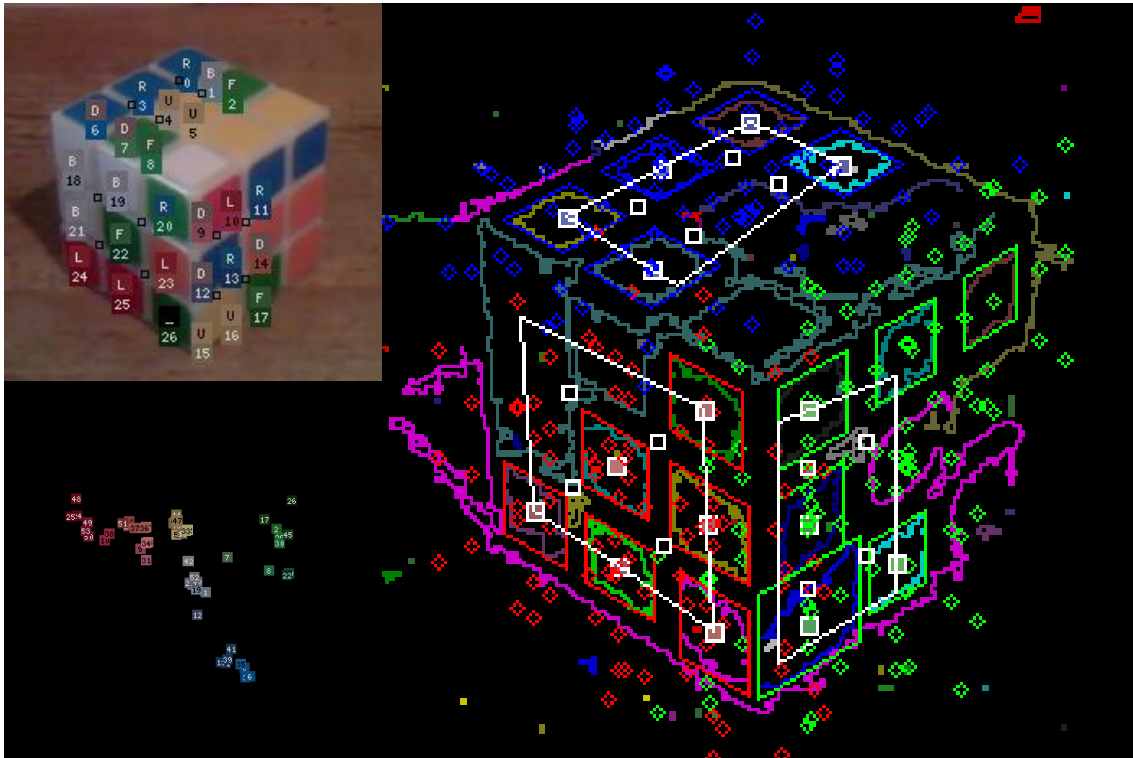
Den fjärde kuben står på ett bord och är fotad med en webbkamera. Fotot har låg upplösning och är oskarpt. Färgerna är bleka och bakgrunden är ett träbord. Steg ett lyckas inte identifiera alla etiketter. Några saknas. Två etiketter har avlästs som en bred och flera har lästs som större än vad de egentligen är. Anpassningen av parallelogrammen i steg två är dålig men ger ändå en korrekt typ kategorisering. När steg tre identifierar den sammanslagna etiketten och räknar den som att den befinner sig mellan två kolumner. Anpassningen av koordinatsystem i steg fyra lyckas hitta bra koordinater för de saknade och sammanslagna etiketterna så att färgavläsningen träffar alla etiketter rätt. I färgplanet syns att färgerna är bleka eftersom de projiceras nära mitten. Dessutom är varje grupp av färger utspridd och grupperna ligger nära varandra. Ändå klarar steg sex att tilldela alla sidor korrekt.



Figur 3.4.1 Resultat.

3.5 Femte kuben

Den femte kuben liknar den fjärde men har en beige bottenfärg (färgen på kuben mellan etiketterna). Här hittas inte många etiketter. Så få att hela rader och kolumner saknas vilket gör att steg tre misslyckas med att bestämma vilka rader och kolumner etiketterna tillhör. Det medför att färgerna läses på fel ställen. (Mätpunkterna för vitbalans klarar sig dock.) Steg sex klarar inte av att kompensera för alla tidigare fel. Kuben blir därmed inte avläst korrekt.



Figur 3.5.1 Resultat.

4 Slutsats

Trots programmets avgränsade uppgift att hitta en kub i två fotografier och bestämma dess tillstånd är problemet komplext. Lösningen består av flera steg och många algoritmer. Linjär algebra används på flera ställen och algoritmer och datastrukturer behöver tas stor hänsyn till vid implementeringen av programmet. Det finns många delar som går lösa på olika sätt med olika krav på arbetsinsats för datorn.

Exempel på detta är kantvandringen som går i linjär tid med avseende på etiketternas omkrets. Om etiketterna istället identifierades genom att titta på alla pixlar innanför kanterna skulle algoritmen gå i linjär tid i förhållande till etiketternas area. Eftersom arean förhåller sig kvadratisk gentemot omkretsen skalar den senare varianten sämre.

Det andra exemplet gäller tilldelningen av etiketternas sidor. Om problemet ses som att gruppera närstående punkter i ett plan erhålls ett väldigt svårt problem. Det kan ta väldigt lång tid att lösa om etiketternas färger är lika varandra. Men genom att utnyttja kunskaper om kubens restriktioner kan en enklare algoritm användas som inte riskerar att ta för lång tid.

Vårt program hittar tillståndet på kuben på ca 1 sekund. Programmet är skrivet i PHP för att användas som en webbapplikation. Trots att språket inte är gjort för tunga beräkningar bearbetas fotona relativt snabbt. Den största delen av tiden går åt till att hitta etiketternas kanter.

Programmet ger utmärkta resultat så länge fotona är av rimlig kvalitet. Det finns inga parametrar som behöver kalibreras så länge färgerna på kuben kan skiljas åt med antingen mättnad eller nyans. Programmet klarar av att gruppera etiketter och ange vilka av dessa som har samma färg.

Programmet är relativt okänsligt för störande objekt i bakgrunden, eftersom kategoriseringen av etiketter och grupperingen till stor grad utesluter former som inte passar in. Svagheterna kommer fram när bilderna är oskarpa vilket resulterar i att få etiketter hittas. Eftersom programmet delar upp etiketter sida för sida, räcker det att en hel kolumn eller rad saknas för att koordinatsystemet ska anpassas felaktigt.

Om kantigenkänningen skulle förstå hur linjer hänger ihop utan att vara sammanhängande hela vägen, skulle analysen av riktigt dåliga bilder kunna lyckas. Detta skulle kunna göras genom att skapa en bild på alla identifierade linjesegment och förlänga dessa. Linjernas riktning kan beräknas och extrapoleras så linjerna blir längre och växer ihop. Då skulle hål i linjerna fyllas igen och bilda slutna sammanhängande ytor, som sedan kan processas som vanligt.

Eftersom programmet delar upp grupperingen och anpassningen av koordinatsystem sida för sida finns det information som inte utnyttjas. Genom att göra en kameramodell i 3d och anpassa den till de funna etiketterna skulle även foton med väldigt få funna etiketter kunna anpassas bra. Vi har utfört prov med kamerakalibrering som gett lovande resultat. Kamerakalibreringen lyckades generera ett korrekt resultat för exempel fem i resultatdelen som det nuvarande programmet misslyckas på. Kamerakalibreringen utnyttjade då endast de funna etiketterna på vänster sida och beräknade koordinater för alla andra etiketter i samma foto utifrån dem.

För att 3d-anpassningen verkligen skulle ge stora fördelar behöver algoritmen för att dela in alla funna etiketter i rader och kolumner bytas ut. Om etiketterna inte kan ges rätt index blir hela Anpassningen lidande. Då felaktiga rader och kolumner på etiketter utgör felaktig information som påverkar hela fotots Anpassning och inte bara den aktuella sidan av kuben. Därmed läses färgerna på fel ställen och kubens tillstånd kan inte hittas.

Problemet med indelning i rader och kolumner skulle kunna förbättras genom att sortera etiketterna med hänsyn till andra sidor. Om varje kant i grafen som visar etiketterns relationer fick en typ som anger i vilken riktning den andra etiketten befinner sig t.ex. "granne till höger" skulle en sökning kunna utföras för att hitta rader och kolumner. Genom att börja i en etikett och följa dess kanter i grafen skulle det då gå att räkna sig fram till vilka rader och kolumner etiketterna ska ha. För varje steg till höger ökar man index för kolumnen. För varje steg till vänster minskar man och så vidare. När två etiketter av olika typ är bredvid varandra går det att avläsa var sidorna möts. Genom att utnyttja avstånden mellan grannarna skulle det även gå att hitta var ensamma etiketter hör hemma på kuben.

5 Referenser

Ett av uppslagsverken som har använts är Wikipedia. Det är förmodligen ett av nutidens största nätbaserade uppslagsverk. Wikipedias koncept bygger på samspel med användarna. Vem som helst kan skicka in redogörande uppslag som granskas av andra användare och i behov rättas till. Fel uppgifter kan förekomma men det sker sällan. Sidan skall vara opartisk och skildra vetenskaplig fakta, vilket man som läsare får känslan av att den gör. Vi har valt att hänvisa till Wikipedia för att läsaren enkelt ska kunna ta del av detaljer som vi har valt att utesluta.

5.1 Böcker

[1] Digital Image Processing

Används som kurslitteratur i kursen DD2423 Image Processing and Computer Vision på KTH.

Prentice Hall 2007-01-31

ISBN: 9780131687288

Författare: Rafael. C. Gonzalez, Richard. E. Woods

[2] Emerging Topics in Computer Vision

Prentice Hall PTR Upper Saddle River, NJ, USA ©2004

ISBN: 0131013661

Författare: Gerard Medioni, Sing Bing Kang

5.2 Websidor

Dan Sunday's Geometry Algorithms

[3] Polygon area computing (2011-04-10)

[http://softsurfer.com/Archive/algorithm_0101/algorithm_0101.htm#2D Polygons](http://softsurfer.com/Archive/algorithm_0101/algorithm_0101.htm#2D%20Polygons)

Google

[4] Similar images labs (2011-04-14)

<http://similar-images.googlelabs.com/>

Högskolan I Gävle

[5] A comparison of medical image analysis algorithms for edge detection (2011-03-05)

<http://hig.diva-portal.org/smash/record.jsf?pid=diva2:324629>

IBM

[6] Vägkameror (2011-04-13)

<http://www-05.ibm.com/se/smarterplanet/traffic/stockholm.html>

Kociemba

[7] Cube Explorer (2011-02-18)

<http://kociemba.org/cube.htm>

Kungliga Tekniska Högskolan

[8] Computer Vision & Active Perception lab (2011-04-13)

<http://www.nada.kth.se/cvap/courses.htm>

Wikipedia

[9] Centroid (2011-04-14)

<http://en.wikipedia.org/wiki/Centroid>

[10] Edge detection (2011-02-18)

http://en.wikipedia.org/wiki/Edge_detection

[11] Least Squares (2011-02-18)

http://en.wikipedia.org/wiki/Least_squares

[12] Ortonormalt Bas (2011-04-14)

http://sv.wikipedia.org/wiki/Ortonormerad_bas

[13] RGB (2011-04-10)

<http://sv.wikipedia.org/wiki/RGB>

