Loggbok

Eric Arnebäck

1 september 2011

Jag mötte idag min handledare Aref och började skriva i loggboken.

Det här projektarbetet kommer att handla om hur bildformat är uppbyggda. Bildformat handlar egentligen bara om hur man får en lång sekvens av nummer(som är hur all data lagras i en dator) att representera en bild.

Fokuset kommer att läggas på formatens uppbyggnad och deras kompressionsalgoritmer. Det är högst troligt att jag bara kommer att hinna med tre bildformat: TGA, GIF och PNG. Jobbet PNG är troligen vad majoriteten av tiden kommer att spenderas på.

Arbetet kommer att vara strukturerat som mer en lärobok. I stället för att fakta bara själlöst rabblas upp ska fokuset ligga på att faktiskt *lära* läsaren något. Innehåller kommer att struktureras upp i kapitel. Språket för den kommer att vara engelska för att så många som möjligt ska kunna läsa den.

Jag hade redan startat arbetet i förväg innan jag hade börjat skriva i den här loggboken. Jag har redan börjat arbeta på det första kapitlet i boken. Det handlar om hur färger representeras i en dator.

För varje bildformat kommer jag att skriva ett program som tolkar datan i en bildfil och dumpar den råa färgdatan och bildens egenskaper i en fil.

Så här kommer det att se ut:

```
Width:2
Height:2
(255,255,255)(0,0,0)
(0,0,0)(255,255,255)
```

Innan jag hade börjat skriva den här loggboken så hade jag redan börjat jobba på programmet som gör det här uppgiften för TGA formatet. Men än så länger analyserar programmet bara header-delen av TGA-filer.

Jag har också de senaste dagarna letat efter källor och information om hur kompressionsalgoritmer fungerar. Jag har hittat några bra källor som jag har organiserat i en textfil.

Sedan igår natt har också arbetat på ett program som demonstrerar den extremt simpla kompressionsalgoritmen RLE. Mer att läsa om denna algoritm finns i den skriftliga delen av arbetet.

Och i dag slutförde jag även detta demoprogrammet.

2 september 2011

För att göra de skriftliga delen och loggboken använder jag faktiskt inte Word, utan LaTeX. Jag föredrar LaTeXför att det är extremt flexibelt, producerar mycket vackrare dokument än Word och för att det tilltalar mig mer programmerare. Om du inte vet vad LaTeXså kan det bli svårt att förklara det...

Detta är den bästa förklaring för den oinvigda som jag har hittat:

http://scottmcpeak.com/latex/whatislatex.html

Men i alla fall, idag listade jag ut hur jag kan skapa vackert formatterade algoritmer i LaTeXdokument. För detta använder jag ett paket som heter `algorithmx`.

Här är till exempel Euklides algoritm för att finna största gemensamma nämnaren av två tal:

Algorithm 1 Euklides algoritm

```
1: procedure EUCLID(a, b)
2:     r ← a mod b
3:     while r ≠ 0 do
4:         a ← b
5:         b ← r
6:         r ← a mod b
7:     end while
8:     return b
9: end procedure
```

Kapitel 2 i detta arbetet kommer att handla om kompressionsalgoritmer. Jag har beskrivit hur RLE fungerar och har även börjat skapat pseudokoden för den.

Fixade också flera fel och buggar i demoprogrammet för RLE.

3 september 2011

Gjorde färdigt pseudokoden för RLE och spenderade även hel del tid med att finputsa algoritmerna och att göra dem enklare att förstå.

Men jag berättade inte riktigt hela historen om RLE. Det finns en bättre, mer effektiv, version av RLE, som TGA använder. Denna bättre version som TGA använder kommer vi att kalla RLE2 från och med nu.

Idag implementerade jag ett demoprogram för RLE2.

Hur mycket jag hade skrivit om digital färglära innan jag hade börjat skriva den här logboken har jag ännu inte sagt. Men jag hade redan fått tag på rätt så bra källor om ämnet. Jag hade skrivit om RGB, alfakanaler och färgdjup. Idag renskrev jag sektionerna om de här ämnena och gjorde några figurer för att förklara bättre. Dessutom skrev jag idag en helt ny sektion om hur gråskala färg representeras digitalt.

Idag fann jag äntligen lite tid till jobba på programmet som laddar TGA filer. Och efter lite arbete kan den ladda okomprimerade gråskalebilder. Men TGA formatet stödjer flera andra bildtyper som det måste implementeras stöd för.

5 september 2011

En bilds metadata beskriver saker som bildens skapare, datumet den skapades, datum den har ändrats, bildkommentarer och så vidare.

Idag skrev jag koden som laddar metadatan från en TGA-fil.

Dessutom kan nu TGA-laddaren ladda *komprimerade* gråskalebilder. TGA använder som bekant RLE2 för sin kompression.

7 september 2011

Efter dagens arbete kan TGA-laddaren ladda 24-bitars okomprimerade och komprimerade true-color bilder. True-color betyder ungefär att färgdatan finns direkt i bilden och inte använder någon palett.

11 september 2011

Ett litet framsteg idag också: lyckades efter lite problem lista ut hur 32-bitars bilder lagrar sin färgdata och implementera stöd för att ladda bilder med 32-bitars färgdjup.

12 september 2011

Implementerade stöd för 16-bitars bilder i TGA-laddaren.

Att implementera stöd för bilder med paletter tog längre tid än jag hade väntat, på grund utav att jag hade skapat flera irriterande buggar under kodningen. Efter att ha fixat alla buggar kunde jag äntligen implementera stöd för bilder med paletter. Med det här så har jag täckt så mycket av TGA formatet som jag tycket att jag behöver göra. Fortfarande finns det delar av TGA formatet jag inte täkt, men de är obskyra och jag har aldrig träffat på några bilder som använder dem.

Imorgon kommer jag att städa upp koden och fixa alla återstående buggar om det finns några.

13 september 2011

Städade upp koden för att göra den enklare att översätta till mer generell pseudokod samt fixade flera buggar.

Resten av dagen planerade jag att dedicera till att beskriva TGA formatet skriftligt. Men sen insåg jag att det skulle vara mycket enklare för läsaren att förstå bildformat om jag först gick igenom hur de i allmänhet är uppbyggda.

Jag började med ett beskriva ett imaginärt bildformat IM som har delar som är gemensamma för *alla* bildformat. På så sätt får läsaren en mycket bättre allmän bild av bildformat.

Detta kommer att göra det så mycket enklare att förklara TGA formatet.

14 september 2011

Renskrev och städade upp kapitlet om IM och gjorde det lite mer lättförståerligt.

Jag har ännu inte disskuterat RLE2 i boken. Så idag skrev jag introduktionen till RLE2 och förklarade algoritmen så väl jag kunde. Nästa gång kommer jag även att skriva pseudokoden för den.

17 september 2011

Gjorde ett högst preliminärt veckoschema för resten av projektet.

Letade efter källor som kommer att hjälpa mig att förstå GIF-formatet och dess kompressionsalgoritm LZW. hittade flera bra källor och organiserade dem i en textfil.

18 september 2011

RLE2 koden har översatts till pseudokod. Jag har även gjort mitt bästa för att förklara algoritmen skriftligt, men jag misstänker att förklaringen behöver renskrivas ett flertal gånger, för en är rätt så rörig.

Har påbörjat renskrivningen av det upp till nu skrivna materialet .

Först så har jag renskrivit för kapitel 1 om digital färglära. Jag lade även till lite nytt material som handlar om färgmodellen CMYK.

Jag renskrev också kapitel 2(kapitlet om IM) och fixade ett flertal syftningsfel och grammatikfel.

19 september 2011

Har påbörjat den långa renskrivningen av kapitel 3, kapitlet om kompression. Men kapitlet är i sig långt från klart och har bara täckt en av de tre algoritmer jag vill gå igenom.

Idag fann ett utmärkt källa för kompressionsalgoritmer som kommer att vara mig användbar hela projektet igenom: *"Data Compression: The Complete Reference"*.

Resten av dagen planerar jag att tillägna till att renskriva kapitel 3.

Under renskrivningen fann jag faktiskt en bugg i mitt demoprogram för RLE2. Jag fixade buggen och även pseudokoden.

Kapitel 3 är renskrivet.

21 september 2011

Har kommit igång med att beskriva TGA formatet. Har än så länge bara börjat att beskriva file header delen av formatet och mycket jobb återstår.

TGA formatet består av så kallade *fält* som lagrar information om bilden. Dessa fält delas i sin tur upp i olika *sektioner*. File Headern är en sådan sektion och den förekommer alltid i början av bildfilen.

24 september 2011

Har fortsatt att jobba med att beskriva TGA formatet. Idag så har jag rätt så mycket bestämt mig för vilka delar jag ska täcka. Det blir bara sektionerna som är väsentliga för att ladda själva färgdatan: file headern och färgdata-sektionen. De andra delarna kan läsaren utforska själv, ty TGA specifikationen är relativt lättläst.

Jobbade lite på att beskriva file header-delen av formatet. Har bara hunnit beskriva några få fält idag.

25 september 2011

Har idag skriftligt beskrivit de färgdjup(gråskala, 16-,24- och 32-bitars färg) jag än så länge träffat på.

Har skrivit pseudokoden för att ladda färgdatan i en TGA fil. Nu har jag beskrivit så mycket av TGA formatet som jag tycker behövs. Det finns vissa mer invecklade fält jag ännu inte har beskrivit. Nu återstår en lång renskrivning, sen måste jag beskriva de sista fälten.

Använde resten av tiden för läsa på lite om LZW algoritmen. Den verkar rätt så invecklad, men ändå helt hanterbar att lära sig på en vecka.

De här källorna kommer att vara mig ovärderliga för att förstå LZW:

```
LZW Data Compression
http://marknelson.us/1989/10/01/lzw-data-compression/

Data Compression - The Complete Reference

A Technique for High-Performance Data Compressio
```

25 september 2011

Påbörjade renskrivningen idag.

Renskrivningen är nästan färdig. Imorgon tänker jag på jobba på att beskriva ett rätt så invecklat fält.

26 september 2011

Fältet som jag jobbar med att beskriva just nu är ett fält som heter image descriptor, på svenska blir det ungefär bild deskriptor.

Den är en byte stor, vilket är ett binärt tal med åtta siffror, som tex 01110110. De separata siffrorna i det binära talet brukar vi inom datorer kalla för bitar. Det intressanta med bild deskriptorn är att olika bitar i den beskriver olika typer av data. Bitarna 0–3(vi börjar räkna från 0 och från höger), 4–5 och 6–7 beskriver olika sorts data.

Det invecklade med detta är hur man får ut de separata bitarnas värden.

27 september 2011

Har beskrivit bild descriptorn och har just renskrivit hela kapitlet om TGA. Imorgon kommer jag att börja verkligen sätta min in i hur LZW fungerar.

1 oktober 2011

Börjat jobba med att försöka lära mig LZW algoritmen. Har inte skrivit mycket kod än utan har mest läst och försökt förstå mig på den.

2 oktober 2011

Efter flera timmars jobb lyckades jag äntligen förstå hur LZW fungerar och implementerade därefter algoritmen i kod. Koden är dock rätt så(eller snarare mycket) ineffektiv och kommer senare att kräva lite ytterligare optimering.

Nu är det dags för mig att implementera den motsvarande dekompressionsalgoritmen.

3 oktober 2011

Och idag så har jag även implementerat dekompressionsalgoritmen för LZW. Det finns ett flertal fel med min implementation som jag kommer att få fixa senare, och den är också rätt så ineffektiv.

6 oktober 2011

Jobbade lite idag med att förbättra min implementation av dekompressionsalgoitmen för LZW. Jag fick inte mycket gjort alls, men har jag listat ut vad jag behöver göra för att förbättra den.

7 oktober 2011

Har förbättrat och optimerat mina implementationer av LZW kompression och dekompression. Filer som förut tog minuter att komprimera tar nu bara några få sekunder. Ja, min förra implementation var rent ut sagt för jävlig och jag skäms nästan över hur dålig den var.

8 oktober 2011

Städade upp koden i mina implementationer av LZW kompression och dekompression.

10 oktober 2011

Har gett en skriftlig förklaring av LZW kompressionsalgoritmen. Nu återstår det dock att jag förklarar den tekniska biten, det vill säga hur man implementerar detta i ett riktigt programmeringsspråk. Har börjat lite smått på det här idag.

11 oktober 2011

Är färdig med att beskriva den tekniska biten av LZW kompression. Nu återstår det att beskriva LZW dekompression. Det hann jag påbörja idag.

12 oktober 2011

Gjort det mesta av jobbet med att förklara LZW dekompression. Lite jobb återstår dock.

15 oktober 2011

Nästan färdig med att förklara LZW dekompression.

Har gjort en undersökning med LZW algoritmen genom att testa hur mycket den kunde komprimera olika sorters filer. Dessutom ändrade jag vissa parametrar på algoritmen för att testa hur de påverkade den slutgiltiga kompressionen. Resultatet av undersökningen har jag sammanfattat skriftligt i en rätt så stor tabell.

17 oktober 2011

Ska nu börja jobba med förstå mig på bildformatet GIF. Först läste jag GIF specifikation. Sen började jag att jobba på ett program som tolkar nummer datan och

i GIF filer och dumpar den råa färgdatan i en fil. Än så länge har jag inte hunnit med något jobb med att tolka själva färgdatan. Har bara hunnit med tolka andra strukturer i formatet. De strukturer jag har hunnit med kallas ”Header”, ”Logical Screen Descriptor”, ”Global Color Table”, ”Graphic Control” och ”Image Descriptor”. Att man förstår innehållet i dessa strukturerna är helt nödvändigt för att man ska kunna förstå hur man ska tolka och okomprimera själva färgdatan i GIF filer.

18 oktober 2011

Har implementerat okomprimering av färgdatan i GIF programmet. Nu kan programmet med andra ord helt och hållet analysera och tolka väldigt enkla GIF filer. Jag har inte testat detta så väl än, så det finns förmodligen ett flertal buggas som jag kommer att behöva fixa.

19 oktober 2011

Majoriteten av dagen spenderades med att lösa ett riktigt klurigt problem. I en GIF fil lagras färg datan i så kallade koder, dvs binära tal som tex 01101. Dessa koder varierar dessutom ofta i sin längd och därför är det lite lurigt att läsa in dem en efter en. Implementation jag gjorde för detta igår fungerade inte för alla sorts koder av olika länger. Därför fick jag skriva om denna implemenation idag, vilket var mycket svårt. Det retsamma är att den fungerande lösningen jag fick fram bara var runt 20 rader kod.

22 oktober 2011

Har spenderat hela dagen med att verkligen finputsa GIF laddaren. Har fixad ett flertal buggar i programmat; några var svåra att fixa, andra var mycket lätta att fixa. Dock återstår det en del buggar som jag kommer att få fixa nästa jag får tid till det här. När jag väl har fixat alla buggar så är jag rätt så mycket färdig med GIF laddaren.

30 oktober 2011

Alla buggar och fel har fixats i GIF laddaren. Jag har testat den ett flertal gånger och inte hittat ett enda fel. GIF laddaren är alltså nu klar.

1 november 2011

Idag har börjat att jobba med att sammanfatta GIF formatet skriftligt. Men jag spenderade dock majoriteten av dagen med att verkligen läsa mig in på alla källor som jag hade hittat innan jag började skriva.

Nu har jag börjat skriva lite smått och har beskrivit historien bakom GIF formatet och lite om hur de minsta datatyperna i formatet är uppbyggda.

5 november 2011

Har börjat beskriva de olika strukturerna som finns i GIF formatet. Har mest fokuserat på att förklara de mer enkla strukturera idag och har sparat de svårare att förklara till det sista.

13 november 2011

Gjorde färdigt min skriftliga beskrivning av GIF formatet.

Eftersom att jag imorgon kommer att påbörja jobbet på DEFLATE-algoritmen, så har jag börjat att söka efter källor som beskriver den så bra som möjligt. Efter lite efterforskningar så har jag funnit att DEFLATE-algoritmen är en kombination av kompressionsalgoritmerna Huffmankodning och LZ77. Jag kommer att utforska och implementera dessa two algoritmerna separat och sedan kommer jag göra sista slutgiltig implementation där jag implementerar DEFLATE genom att kombinera LZ77 och Huffmankodning.

14 november 2011

Har idag påbörjat jobbet med att implementera kompressionsalgoritmen Huffmankodning. Jag lyckades i kodningen få en hel del gjort idag, men en hel del jobb återstår fortfarande. Jag är dock mycket optimistiskt till att jag kommer att hinna klart arbetet på en vecka. Sen spenderar nästa vecka på kompressionsalgoritmen LZ77 och veckan efter det kombinerar jag från de två tidigare veckorna arbete i den slutgiltiga DEFLATE algoritmen.

15 november 2011

Spenderade hela dagen med att fixa en otroligt jobbig bugg. Fick inte mycket mer gjort.

17 november 2011

Är nästan färdig med att implementera Huffmankodning nu. Sen återstår det bara att implementera den motsvarande dekompressionsalgoritmen.

18 november 2011

Var lite osäker på hur man skulle implementera vissa detaljer i Huffmankodning, så jag började idag att leta efter källor och läsa på i detalj hur DEFLATE algoritmen verkligen fungerar.

19 november 2011

För att jag inte kunde förstå hur LZ77 och Huffmankodning skulle kombineras till DEFLATE så har börjat att jobba på att implementera DEFLATE algoritmen idag och börjat med att förstå den överskådligt. Men hann inte skriva så mycket kod idag, utan läste mest de källor jag fått tag på och gjorde efterforskningar.

20 november 2011

Jag spenderade lite tid på att städa upp koden i lite gamla program.

Hur fortsuttit arbete med att försöka slutgöra algoritmen för Huffmankodning. Det jag har problem med är man ska kombinera Huffmankodning med LZ77 och bilda DEFLATE algoritmen. De källor jag har fått tag på som beskriver det här är rätt så otydliga. Har nu fått på några källor som jag verkligen tror kan hjälpa mig att slutföra arbetet med Huffmankodning. Ska läsa igenom dem imorgon.

21 november 2011

Jag har bestämt mig att bara för att implementera dekompression av DEFLATE algoritmen så länge. Jag funnit det mycket svårt att kunna implementera DEFLATE kompression och jag tänker nu fokusera på att implementera bara dekompressionsversionen. Får jag tid över senare tänker jag även implementera kompressionsversionen men nu tänker jag fokusera på dekompressionsversionen. Jag förklarar mina anledningar senare.

Så idag påbörjade jag arbetet med att implementera dekompressionsalgoritmen för DEFLATE.

22 november 2011

Har fortsuttit med implementeringen av dekompressionsalgoritmen för DEFLATE. Arbetet går sakta men säkert framåt och jag mycket säker på att jag hinner bli färdig med kodningen den här veckan.

26 november 2011

Är nästan färdig med implementeringen nu.

27 november 2011

Är Färdig med implementeringen av dekompressionsalgoritmen för DEFLATE. Nästa sak jag tänker att göra att utforska hur färgdatan är lagrad i ett annat bildformat: PNG. PNG använder DEFLATE komprimering för att komprimera sin färgdata. Jag har därför börjat att läsa alla de källor jag har fått tag som handlar om hur PNG formatet är uppbyggd.

28 november 2011

Har fortsuttit med mina efterforskningar om hur PNG formatet är uppbyggt. Nu har jag fått en rätt så bra översiktlig bild om hur PNG formatet är uppbyggt, så nästa gång tänker på börja jobba på ett program som analyserar och tolkar innehållet i PNG filer.

29 november 2011

Påbörjat arbetet med att skriva det förutnämnda programmet.

30 november 2011

Jobbat vidare med kodningen.

4 december 2011

För att kunna förstå mig på vissa delar av PNG formatet fann jag det nödvändigt att fördjupa mig i färglära. Därför har jag mest läst en massa idag. Jag har hittat mängder av källor på det här området och jag har ännu inte hunnit sammanställa eller ens läst igenom alla.

5 december 2011

Har fortsuttit med arbetet jag påbörjade igår.