Laborationer i kursmomentet Datoranvändning E1

Laboration nr 6: Filformat för bilder

98-10-26 Thomas Hallgren

1 Introduktion

Denna laboration knyter an till föreläsning 5 i kursen, där jag pratade om hur olika typer av information kan representeras digitalt. Vi koncentrerar oss här på hur bilder representeras. Vilka filformat är lämpliga för olika typer av bilder? Hur bra kompression ger olika filformat? Vilken bildkvalité får man? Vi jämför formaten PPM, GIF, JPEG och PostScript. Som hjälpmedel använder programmen `xv`, MATLAB och FrameMaker.

2 Uppgifter

I denna laboration är det lite färre uppgifter än i de tidigare, så förhoppningsvis går det att hinna med allihop på två timmar. Skriv ner svaren på frågorna och visa för en handledare för att bli godkänd.

Uppgift 1. Titta på bilder med xv

Vi börjar med att titta på bilden `~hallgren/Intro/blomvas.ppm`. Som ändelsen `.ppm` antyder är bilden sparad i PPM-format. Detta är ett mycket enkelt format utan någon form av kompression. Hur stor är filen? ____________________________

Starta programmet `xv` och titta på bilden. Man kan göra det på två sätt:

- Man kan ange namnet på bildfilen på kommandoraden när man startar `xv`.
- Man kan starta `xv` utan att ange en bild. Genom att klicka med höger musknapp i `xv`:s fönster kan man få fram en kontrollpanel där bland annat kan trycka på knappen `Load` för att välja en ny bild som ska visas.

I PPM-formatet lagras färgen på varje pixel i RGB-format (dvs man lagrar hur mycket av primärfärgerna rött, grönt och blått som ingår) i 3 bytes per pixel. Om man trycker på tangenten `i` i `xv`-fönstret visas ett informationsfönster, där man bland annat kan se hur många pixels bred och hög bilden är (resolution).

- Hur många pixlar bred och hög är bilden? ___________
- Hur många pixlar blir det? ____________
- Hur många bytes är det i PPM-filen utöver det som behövs för pixlarna? _______

(För den nyfikne: titta hur filen ser ut om man öppnar den med `nedit`. Läs manualbladet för PPM-formatet med kommandot `man ppm`.)

Uppgift 2. Vi undersöker JPEG-formatet

`xv` kan visa bilder lagrade i många olika format, och även spara i många olika format, så man kan alltså använda `xv` för att konvertera mellan bildformat. Vi ska utnyttja detta för att spara bilden från förra uppgiften i JPEG-format och GIF-format och jämföra resultaten.

Vi börjar med att prova JPEG-formatet. Tag fram `xv`:s kontrollpanel (om du inte redan gjort det). Tryck på knappen `Save`. Det dyker upp ett filvalsfönster. Gå till labbgruppens hemkatalog (om det inte redan är den katalogen som visas), välj JPEG i `Format`-menyn, kontrollera att det står `blomvas.jpg` filnamnsfältet längst ner (ändra om det behövs) och tryck på OK. Det dyker upp ytterligare ett fönster, där man kan välja vilken bildkvalité man vill ha, men låt de förvalda inställningarna vara kvar och tryck på bara på OK.

- Hur stor plats tar bilden när den lagras i JPEG-format? ____________
- Hur många procent av den PPM-filens storlek är det? ___________

Man undrar förstås nu vilken effekt detta har på bildkvalitén. JPEG är ju ett format som sparar plats genom att ta bort detaljer ur bilden baserat på vad som är lätt att se med blotta ögat. Vi får ta en titt på JPEG-bilden. I `xv`:s kontrollpanelfönster finns en ruta där namnen på använda filer visas. JPEG-bilden vi just sparade borde finnas med där. Dubbelklicka på den för att visa den i bildfönstret. Kontrollera bildkvaliten. Dubbelkicka på den ursprungliga PPM-bilden igen och jämför. Syns det någon stor/liten skillnad? ______________________________

(För den nyfikne: man kan alltså välja bildkvalité när man sparar i JPEG-format. Prova att spara med högre/lägre kvalité och undersök vad detta får för effekt på filstorleken och bildkvalitén.)

> **Anm.** När man jämför bilder kanske man vill se båda samtidgt. `xv` kan bara visa en bild i taget, men man kan ju starta två kopior av `xv`, om man vill...

Vi ska även undersöka vad JPEG-kompressionen har för effekt på en annan typ av bild.

- Öppna bilden `~hallgren/Intro/screendump1.ppm` och spara den i JPEG-format. Hur stor blev filen (i procent av PPM-filens storlek)? ________
- Syns det några skillnader mellan originalet och den komprimerade bilden i detta fall? _____
- Prova att titta på bilden med 2 gångers förstoring. (Välj alternativet `Double Size` i menyn `Image Size` i `xv`:s kontrollfönster. Notera vilken tangent man kan använda för att göra samma sak). Försök säga något om hur JPEG-kompressionen har påverkat bilden: ____

Uppgift 3. Vi undersöker GIF-formatet

Klicka fram den första bilden igen (`blomvas.ppm`) och spara den i GIF-format.

- Hur stor blev GIF-filen? ____________

- Hur många procent av PPM-filens storlek är det? ___________
- Hur många procent av JPEG-filens storlek är det? ___________
- Jämför JPEG-bilden och GIF-bildens kvalité. Är någon av dem betydligt bättre än den andra, och vilken i så fall? ____________________
- Vilket format är lämpligast för denna bild? ______________

Klicka även fram den andra bilden (`screendump1.ppm`) och spara den i GIF-format.

- Hur stor blev GIF-filen? ____________
- Hur många procent av PPM-filens storlek är det? ___________
- Vilket format är lämpligast för denna bild? ______________

> **Anm.** Även GIF-formatet komprimerar bildinformationen, men till skillnad från JPEG-formatet används en förlustfri kompression. Färgen på varje pixel bevaras alltså exakt. Men GIF-formatet är begränsat till bilder med höst 256 olika färger (så att det räcker att lagra en byte per pixel). För enkla dator/handgjorda bilder, som screendump1 ger GIF-formatet en exakt kopia av originalbilden. För mer färgrika bilder, som t ex foton, måste man alltså på något sätt reducera antalet färger, vilket kan mer eller mindre märkbart försämra bilden. Bilden blomvas innehåller tydligen inte så många olika färger så GIF-versionen blev hyfsat bra, men det kan alltså bli betydligt sämre.

Uppgift 4. Vi undersöker PostScript-formatet

PPM, GIF och JPEG är exempel på pixelbaserade bildlagringsformat. Detta betyder att bilden delas in i punkter och färgen på varje punkt lagras. Ju finare indelning (dvs ju fler punkter) man använder, desto högre blir bildkvalitén, men storleken på filerna växer förstås också. Därför lagrar man inte bilderna med fler punkter än nödvändigt för ändamålet. Typiska datorbildskärmar visar 75dpi (dots per inch, punkter per tum) eller lite mer.

PostScript är ett objektbaserat bildbeskrivningsformat, dvs man lagrar vilka grafiska objekt bilden är uppbyggd av, eller, om man så vill, hur bilden ritades. Detta passar bra för bilder som framställs på dator med något slags ritprogram, men förstås mindre bra för foton. Storleken på filerna beror på hur komplicerade bilder man lagrar, men är oberoende av storleken på bilden.

> **Anm.** När PostScript utvecklades ville man ha ett utökningsbar standard för bildbeskrivning. Man löste detta genom att göra PostScript till ett programmeringsspråk! Detta ger förstås mycket stor frihet.

Vi ska nu jämföra en pixelbaserad bild som skapats för att visas på en bildskärm och en PostScript-version av samma bild, genom att rita en sinuskurva med MATLAB, importera den på två olika sätt i FrameMaker och skriva ut dem.

- Starta MATLAB och rita en jämn och fin sinuskurva som i Uppgift 3 i Laboration nr 5.
- Spara kurvan i PostScriptformat med MATLAB-kommandot

```
print sinuskurva.epsi -depsc2 -epsi
```

Den sista flaggan, `-epsi`, gör att en pixelbaserad kopia av bilden bakas in i PostScriptfilen. Detta gör att filen blir onödigt stor (i det här fallet 100KB istället för 7KB, eller något sånt). Man kan hoppa över detta, men då visas en grå ruta istället för bilden när man importerar den i FrameMaker. (Det blir rätt när man skriver ut dokumentet.)

- Gör en skärmavbildning av sinuskurvan. Man kan använda xv för detta genom att trycka på knappen `Grab` i kontrollpanelen. Det dyker då upp ett nytt fönster där det står att man kan avbilda ett fönster genom att kicka i det eller en godtycklig rektangel. Tryck på `Grab`-knappen i det fönstret och klicka sedan i fönstret med sinuskurvan. Spara bilden i GIF-format i filen `sinuskurva.gif`.

  Anm. I Laboration nr 4 gjorde vi en skärmavbildning med hjälp av FrameMaker. Två fördelar med att göra det med xv är att man lätt kan avbilda ett helt fönster och att man kan spara i valfritt filformat, t ex GIF, istället för XWD-formatet, som är okomprimerat och därmed ger väldigt stora filer.

Vi har nu tillverkat de två versionerna av sinuskurvan, så starta FrameMaker och skapa ett nytt dokument i stående A4. Importera de båda bilderna och justera storleken på dem så att de blir ungefär lika stora och precis får plats på en sida.

- När man tittar på dokumentet på bildskärmen, är bildkvalitén på någon av de båda sinuskurvorna betydligt bättre än den andra, och i så fall vilken? ____________

Spara dokumentet. Skriv ut det.

- Undersök utskriften. Är bildkvalitén på någon av de båda sinuskurvorna betydligt bättre än den andra, och i så fall vilken? _______________

3 Avslutande kommentarer

Anledningen till att vi tittade på just JPEG och GIF är de är de två vanligaste formaten för bilder på webben. Ett nyare format, som utvecklades för att ersätta GIF är PNG. Det använder också förlustfri kompression, men är inte begränsat till 256 färger. Det finns även andra fördelar med PNG.

Anledningen till att vi tittade på PostScript var att illustrera att pixelbaserade bilder som är skapade för att se bra ut på bildskärmen inte är det bästa när man skriver ut på skrivare. Detta beror på att skrivare har högre upplösning, vanligen 300dpi eller 600dpi.

Mer information om olika bildformat kan man förstås hitta på webben. En bra sida att starta från är:

```
http://www.dcs.ed.ac.uk/~mxr/gfx/
```