

Bildkomprimering med JPEG, Fraktaler och Krusningar (Wavelets)

Projektarbete i Bildanalys av

Jacob Ström, D-91

Handledare: Sven Spanne

maj 1994

1 Inledning

Färgrika datorbilder av hög upplösning tar stor plats att lagra. Allt eftersom våra skärmar blir bättre och bättre klarar de av att visa högre upplösning och fler färger och datamängderna för bilderna bara ökar. Därför är det viktigt med effektiva algoritmer för bildkomprimering. Eftersom det är människor som skall titta på bilderna behöver inte de inte återskapas helt exakt och detta öppnar möjligheter för att komprimera mycket mer effektivt (så kallade lossy crunchers). I mitt projektarbete har jag studerat tre sätt att komprimera bilder på: En beprövad variant (JPEG) och två lite mer udda, fraktalkomprimering och komprimering med hjälp av krusningar (Wavelets). Alla använder sig av lossy crunching, dvs en del av information från ursprungsbilden slängs bort.

2 Hur fungerar det?

2.1 JPEG

JPEG står för "Joint Photographic Experts Group", och kommer från den kommité som skrev standarden. JPEG delar in bilden i rutor om $8 \cdot 8 = 64$ pixels. Dessa undergår en diskret cosinus transform, som är en speglad variant av den diskreta fouriertransformen. De koefficienter man nu får ut kan exakt återskapa de 64 pixlarna. Nu spar man emellertid bara de stora koefficienterna helt; de små slängs bort eller kodas med sämre noggrannhet.

2.2 Fraktalkomprimering

Fraktal komprimering är den andra metoden. Grovt förenklat kan man säga att den utnyttjar att det i en bild ofta finns självliknande områden: En del av bilden liknar en helt annan del av bilden, fast är lite mindre och kanske speglad. Då kan man återskapa den mindre delen genom att spegla och förminska den stora. Nu visar det sig att den enda information man behöver spara är själva transformeringarna, dvs hur man skall förminska och spegla. Kodar man dessa effektivt, kan man få en bra packning.

2.3 Komprimering med Krusningar

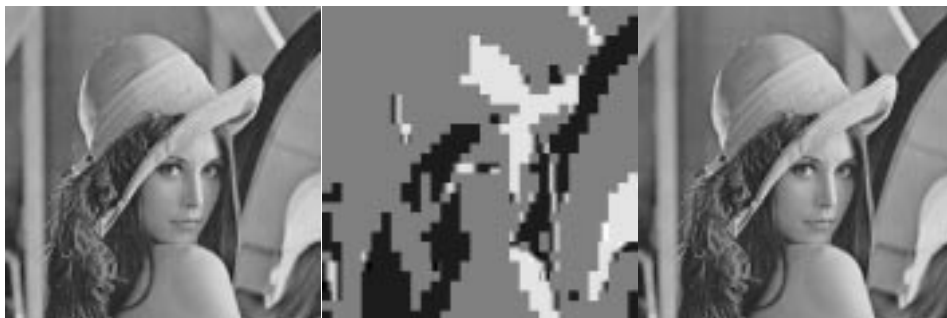
Krusningar, eller Wavelets som det heter på engelska, är ett relativt nytt forskningsområde som blivit väldigt populärt den senaste tiden. Väldigt förenklat kan man säga att medan fouriertransformen (JPEG) arbetar helt i frekvensplanet, lever krusningarna lite grann i både frekvens och tidsplanet. Man hittar ett ortogonalt system av krusningar, delar upp bilden med hjälp av dem och slänger (eller kodar dåligt) de koefficienter som är små.

3 Tillvägagångssätt

För att jämföra de olika metoderna utgick jag från bilden Lena¹ och packade ner denna olika hårt och fick därmed skiftande kvalitet. Sedan bedömde jag dessa dels subjektivt ("okulär besiktning") samt mätte jag minstakvadratfelet för att få ett något mer objektivt mått.

¹Bilden är $256 \cdot 256$ i 256 gräskalor och väldigt använd inom bildanalys

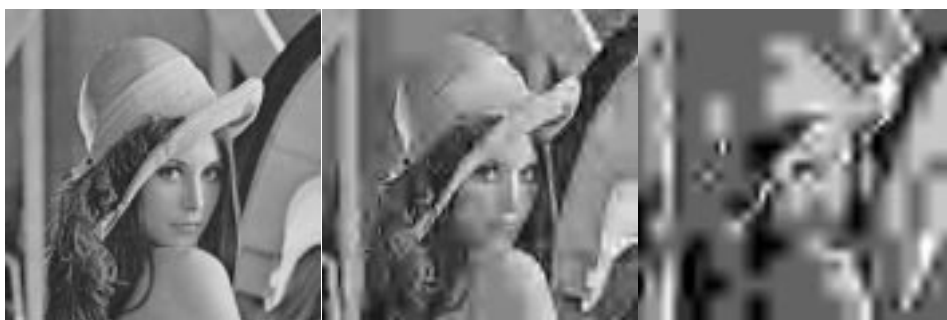
Här följer först originalbilden Lena samt de komprimerade versionerna för JPEG, fraktaler och krusningar. Reproduktionen tillåter inte någon noggrannare särskådning av bilderna men jag hoppas ändå att man kan se skillnaderna mellan de olika formaten.



Figur 1: JPEG, 5.7 ggr Figur 2: JPG, 49.6 ggr Figur 3: Original, 1 ggr



Figur 4: Fraktal, 5.9 ggr Figur 5: Fraktal, 49.6 ggr Figur 6: Fraktal, 93.2 ggr



Figur 7: Wave, 5.5 ggr Figur 8: Wave, 49.4 ggr Figur 9: Wave, 93.5 ggr

Här kan man se att skillnaderna är små för måttliga kompressionsfaktorer (5-6) medan de det vid 50 gångers kompression är stor skillnad. Här är krusnings-metoden märkbart bättre än de två andra. Till slut har vi c:a 100 gångers packning. Här

vinner fraktalkomprimeringen - å andra sidan finns det inte så mycket bild kvar att tala om. Till den subjektiva bedömningen måste läggas att krusningskomprimerade bilder har en egenskap att bevara ytor med jämna övergångar bättre, något som ökar realismen i bilden. JPEG tenderar istället att, med ökande kompressionsgrad, ta bort allt fler gråskalor i bilden tills den blir "kornig" och till slut alltför kontrastrik. Både JPEG- och fraktalkomprimerade bilder har dessutom nackdelen att ett $8 \cdot 8$ mönster blir synbart vid ökad kompressionsgrad.

3.1 Minsta kvadratfel

För att få ett lite mer objektivt mått på bildkvaliteten mätte jag upp minsta kvadratfelet för de tre metoderna med varierande packningstäthet. Jag plottade $-\log(\text{kvadratfelet})$ som funktion av packningstätheten ($\frac{\text{orig.storlek}}{\text{storlek}}$). Resultatet kan man se i figur 10. Här ser man att Wavelet och JPEG är jämförbara i början av kurvan, dvs vid låg packningstäthet. Någonstans mellan 20 och 30 gångers packning börjar JPEG sacka och mister stort i kvalitet vid ökande packningstäthet - kurvan sluttar brant. Högre packning än 50 gånger gick inte alls att uppnå med den JPEG packare jag använde². Fraktal-kurvan börjar väldigt dåligt men kommer igen på slutet, vid runt 70 gångers packning börjar det löna sig att fraktalkomprimera. Om man särskådar kurvorna kan man se att JPEG är bäst från faktor 1.0 till 2.5, sedan tar krusningarna över fram till 10. Återigen är JPEG bäst fram till ca 17, sedan krusningar fram till 70. Nu skall man inte övertolka grafen, den är gjort på ett mindre antal komprimeringar, - ca 15 - och dessutom bara på samma bild hela tiden.

3.2 Informationshantering

Man kan även se på de tre metoderna ur informationshänseende. JPEG är här klart sämst eftersom varje $8 \cdot 8$ -ruta behandlas för sig. Inget flöde av information över gränserna, alltså. Fraktal-komprimeringen är lite svårare att tolka, visserligen tas bara information från en $16 \cdot 16$ -cell och skalas ner till en $8 \cdot 8$, men å andra sidan gås hela bilden igenom vid nerkodningen för att hitta den bästa av alla möjliga $16 \cdot 16$ celler, så här utnyttjas hela bilden lite bättre. Krusningarna kan sägas vara ett optimalt sätt att hantera informationen i bilden.

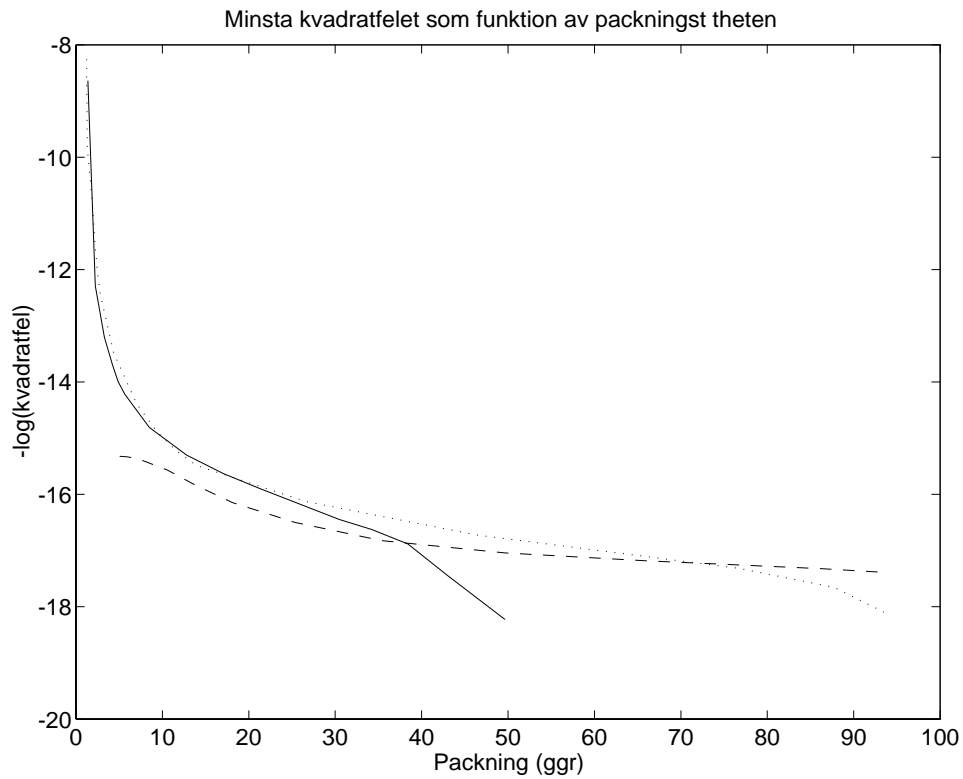
3.3 Tidsaspekten

För de flesta tillämpningar är inte bara packningstätheter intressanta, utan även hur snabbt det går att packa och packa upp sina bilder. Skall metoderna användas för tex bildtelefoni går det inte att ha en latent tid på 3 minuter innan mottagaren får den i andra änden. Jag har inte gjort några noggrannare studier här, utan har nöjt mig med att konstatera att fraktalkomprimering är i särklass sämst. Det tar i storleksordningen minut att komprimera Lena och åtskilliga sekunder att packa upp. Krusnings- och JPEG- metoderna är ungefär lika snabba, både vid komprimering och upppackning. Detta är anmärkningsvärt eftersom JPEG funnits en längre tid och rutinerna som här används borde vara starkt optimerade.

3.4 Sammanfattning

Krusningar har kommit för att stanna. Eftersom skillnaderna mot JPEG inte är så våldsamma inom det mest högkvalitativa området (1-20 ggr packning) kommer nog JPEG att förbli standard inom de flesta områdena. Där man däremot kan

²Jag använde mig av xv:s save-funktion.



Figur 10: Hur minsta kvadratfelet varierar med packningstätheten. JPEG är hel-
 dragen linje, krusningarna är prickade och fraktalkurvan är streckad

tumma lite på bildkvalitén vinner man på att byta teknik, ett exempel är bildte-
 lefoni. Att använda krusningar för att komprimera animationer är också lockande,
 eftersom informationen mellan bilderna kanske kan utnyttjas bättre än i JPEGs
 motsvarighet MPEG. MPEG lär nämligen utnyttja informationen mellan bilderna
 rätt dåligt. Går man ner en dimension istället, kan man säkert ha användning för
 krusningskompression i talöverföring. Detta hamnar dock utanför ämnet bildanalys.