

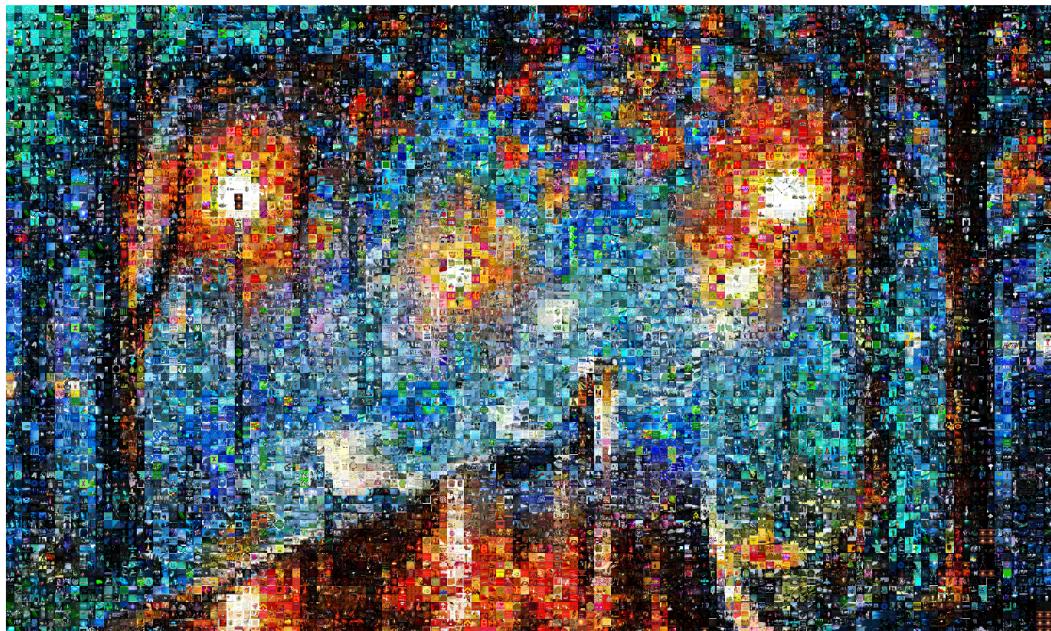
BILDMOSAIK

BILDDATABASER, TNM025

Linköpings Universitet

Anna FLISBERG
Linnéa MELLBLOM

annfl042
linme882



28 maj 2015

Innehåll

1	Introduktion	2
2	Metod	2
2.1	Features	2
2.2	Representation av features	2
2.3	LBP	2
2.4	PCA	2
2.5	Databas	3
2.6	Mosaikbild	3
3	Resultat	3
4	Diskussion	3
5	Förslag till förbättringar	4
A		
	Resultatbilder	5

1 Introduktion

Målet med projektet i kursen TNM025 är att skapa ett program i *Matlab* som kan komponera en så kallad mosaikbild. Givet en inbild ska programmet försöka återskapa denna bild genom att ersätta områden i inbilden med olika databasbilder. Mosaikbilden som skapas kommer därmed innehålla en mängd olika databasbilder. På nära avstånd kommer varje enskild databasbild att synas men på ett längre avstånd kommer mosaikbilden i sin helhet att efterlikna orginalbilden.

2 Metod

För att skapa en mosiskbild behövs en databas med en stor mängd varierade bilder. Bilderna i databasen analyseras för att ta fram huvudkomponenter som beskriver egenskaper som kan jämföras. Den generella strategin för att bygga upp databasen är följande:

1. Välj ut features som ska mäts, exempelvis färgrymd.
2. Välj representation för features, vanligen histogram.
3. Komprimera representationen genom PCA.

För att skapa mosaikbilden delas orginalbilden upp i mindre områden som ska ersättas med en bild från databasen. Analys görs på varje område och jämförs med de huvudkomponenter som beskriver egenskaperna för bilderna i databasen. Jämförelsen kan ses som en distans och beräknas genom exempelvis euklidisk avstånd. Genom detta mått kan den bild i databasen som är mest lik varje område väljas ut.

2.1 Features

Features kallas de egenskaper som väljs ut för att ge beskrivande information om bilderna, exempelvis färgrymd eller struktur. Det är med dessa features som jämförelse sker för att hitta bilder i databasen matchar området i inbilden. I detta projekt används *CIELAB*, *LBP* och *W* som features.

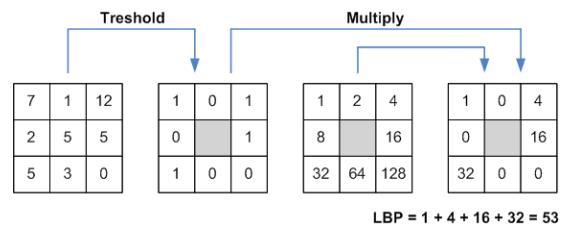
2.2 Representation av features

Histogram används för att visa fördelningarna *CIELAB* och *LBP*. För att få en statistisk representation normaliseras histogrammen.

W är en representation för mättnad och nyans i enhetsskivan med komplexa värden. Denna implementation är given av Reiner Lenzs föreläsningar i kurserna TNM025, 2015.

2.3 LBP

Local Binary Pattern, förkortat *LBP* är en effektiv metod för att beskriva texturer [1]. *LBP* kategoriseringen av pixlarna i en bild genom att tröskla de 8 grannarna av varje pixel där tröskelvärdet är pixelvärdet, multiplicera med en given matris och betrakta resultatet som ett tal i denna pixel, se figur 2.3. Det är över detta tal som histogrammet beräknas.



Figur 1: Metod för LBP.

2.4 PCA

Principal Component Analysis, förkortat *PCA*, är en linjär ortogonal transform som konverterar ett dataset med variabler som möjligt är korrelerade till ett dataset där variablerna säkerligen är okorrelerade [2]. Detta gör det möjligt att med endast ett fåtal huvudkomponenter beskriva ett dataset av en högre dimension.

Korrelationsmatriser som beskriver bilderna i databasen beräknas för de features som representeras med hjälp av histogram 1, där *Hist* är histogram över de valda features.

$$Korr = Hist * Hist' \quad (1)$$

Histogrammen har dimensionerna $Y \times n$, där Y är antal bins från histogrammet och n är antal bilder i databasen som ska jämföras. Detta ger en korrelationsmatris av dimension $Y \times Y$. Ur denna tas sedan egenvärden och egenvektorer fram. Egenvektorerna bygger nu upp ett nytt koordinatsystem där egenvärdena utgör varianserna längs med de nya koordinataxlarna.

Nu beräknas featurevektorer för alla valda features enligt 2

$$featureV = Hist * eigVectors; \quad (2)$$

Featurevektorerna kan nu ses som en komprimerad version av histogrammen då de representeras av ett färre antal dimensioner i det nya koordinatsystemet.

2.5 Databas

Databasen byggs upp genom att läsa in n antal bilder. För varje bild beräknas histogram utifrån de features som valts. Dessa histogram lagras i en stor matris, som blir $antal_bins \times antal_bilder$, där histogrammen ger $antal_bins$. Det är utifrån denna stora matris som korrelationsmatrisen enligt ekvation 1 beräknas. Sedan beräknas PCA som ger huvudkomponenterna för varje bild i databasen för varje feature. När databasen byggs upp, beräknas även dessa huvudkomponenter i kvadrat. Denna kvadrat används när distansberäkning ska ske vid jämförelse mellan en querybild och en databasbild. Förberäkningen sker för att snabba på processen.

2.6 Mosaikbild

För att bygga upp den nya bilden bryts originalbilden ned i ett antal områden. För varje område beräknas sedan samma features ($QHist$) som när databasen beräknades. Sedan beräknas också featurevektorer för varje område enligt ekvation 3 där $eigVectors$ är de huvudkomponenter vi fått från databasen.

$$QfeatureV_n = QHist_n * eigVectors \quad (3)$$

$QfeatureV_i$ jämförs sedan med alla bilder databasen för att hitta den bild som är mest lik området. Jämförelsen sker genom en euklidisk distansfunktion, $L2$ -norm.

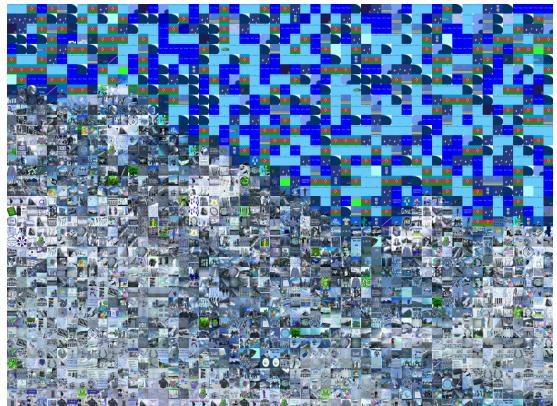
$$\begin{aligned} distans_i &= db_i^2 - 2 * q * db_i \\ q &= \text{område} \\ db_i &= \text{bild nummer } i, \text{ i databasen} \end{aligned} \quad (4)$$

För varje feature beräknas en distans enligt ekvation 4. Genom att kombinera dessa distanser och vikta dem olika får en slutgiltig distansen fram. Den bild i databasen som är mest lik området får genom att ta den distans som är minst. Databasbilder ersätter varje område och en mosaikbild är skapad!

3 Resultat



Figur 2: En tiger.



Figur 3: Resultat av berg som inte blev så bra.

Mer resultatbilder finns i bilaga A.

4 Diskussion

Att välja vilka features som skulle beräknas för att få det bästa resultatet var något som vi funderade mycket kring. När det gäller färgrymd testades först RGB men vi insåg ganska snabbt att det blev svårt att få till en vettig distansfunktion. HSV fungerade inte heller som vi ville eftersom mättnad representeras av en vinkel. Med $CIELAB$ fick vi vårt bästa resultat, eftersom det är en ungefärligt jämnfordelad färgrymd.

Att väga in W gör att mättnaden förbättras lite, dock med nackdelen att beräkningarna för W tar väldigt lång tid. W gör också att homogena områden hanteras bättre, vilket är ett problem som fortfarande återstår. I figur 3 kan man tydligt se att homogena blåa områden ersätts med gröna bilder.

LBP använde vi för att kunna väga in mönster i beräkningen av vilken bild som skulle plockas från databasen. Detta för att mosaikbilden ska få liknande struktur.

Att hitta en viktning mellan $CIELAB$, W och LBP som gav ett resultat som vi var nöjda med, var något som vi kämpade med. Det var svårt att förutse hur olika viktningar skulle påverka mosaikbilderna och i slutändan så testade vi oss fram för att få ett bra resultat.

Storleken på områdena som plockas ut från inbilden avgör också hur pass bra resultatet blir. Om området blir för stort försätter med stor sannolikhet detaljer i bilden. För små områden ger troligtvis mer homogena områden vilket vårt program som sagt inte alltid klarar av.

Att läsa in databasen är det som tar mest tid. 100 000 bilder med alla features ($CIELAB$, W , LBP) tog cirka 35 minuter. Efter förbättringar och optimeringar är nu inläsningstiden förkortas till 14 minuter. Då det är en engångsföreteelse så känns tiden rimlig. Väljer vi bort att beräkna W så halveras inläsningstiden för databasen men som vi nämnde ovan hjälper W oss med mätnad och homogena områden.

Storleken på databasen spelar stor roll i hur resultatet blir. Databasbilderna ska helst representera så många färger som möjligt för att ett område ska kunna ersättas med en representativ bild. Detta upptäcktes i början när färre bilder

användes och resultatet inte blev så bra.

5 Förslag till förbättringar

Några förbättringar som skulle kunna göras i projektet är att en databasbild har en begränsning på hur många gånger den får förekomma i en mosaikbild. På detta sätt kan man ta bort upprepningar. Dessa upprepningar är lätt för oss människor att upptäcka och om vi då tar bort dessa så kommer troligtvis bilden att upplevas bättre.

En annan förbättring som vore trevlig är att inbilden analyseras i förväg så att områdena som väljs ut och ska ersättas har olika storlek beroende på om inbilden är detaljerad eller inte i just sagda område. På detta sätt kan stora homogena områden ersättas av en större bild medan mer detaljrika områden delas upp i flera mindre.

Referenser

- [1] Beskrivning av metoden Local Binary Pattern. http://www.scholarpedia.org/article/Local_Binary_Patterns 28 maj 2015
- [2] Beskrivning av metoden Principal component analysis. http://en.wikipedia.org/wiki/Principal_component_analysis 28 maj 2015

A

Resultatbilder

