MSPEL Föreläsning 3

Färgmodeller och digitala bildformat

DSV Peter Mozelius

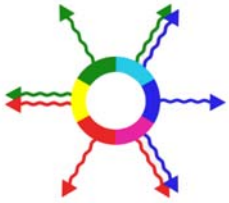
1

Ljus och våglängder

- Det synliga spektrumet ca 400-800 nm
- Ultraviolett - Synligt ljus - Infrarött
  - Violett 390 - 430 nm
  - Blått 430 - 500 nm
  - Grönt 500 - 580 nm
  - Gult 580 - 600 nm
  - Rött 600 - 750 nm

2

Färgcirkeln



Red
Green
Blue
Yellow
Magenta
Cyan

Komplementfärger är två färger mitt emot varandra i en färgcirkel och om de adderas ger de upphov till vitt ljus

3

Färgmodeller

- RGB-modellen, ett sätt att härma ögat
- Färgkänsliga tappar i näthinnan
- Stavarna registrerar ljusstyrkan
- RGB för dataskärmar, scanners mm
- CMYK för tryckprocesser
- Additiv och Subtraktiv färgblandning
- HSB för digital bildbehandling mm

4

Färgmodeller - RGB

- RGB - modell för ljus
- Standard för skärmar
- Standard för scanners
- Standard i HTML
- Standard i Javascript
- En färgmodell i Java
- En färgmodell i PhotoShop

5

Färgmodeller - CMYK

Subtraktiv färgblandning

Vid färgblandning i tryck är utgångspunkten vitt papper och en ljuskälla som avger vitt ljus. Pappret reflekterar ljuskällans hela färgspektrum. Olika kulörer uppstår när delar av ljuskällans spektrum filtreras bort av de transparenta tryckfärgerna. Tekniken kallas *Subtraktiv färgblandning (CMY).*

K = Key color

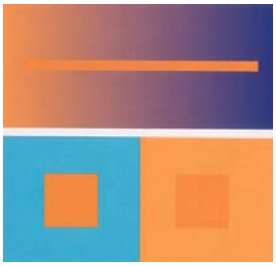
6

Färg – designregler

- Färguppfattningen är individuell
  MEN
- Undvik komplementfärger på samma sida om det inte finns en klar anledning
  - Genomgående i Microsoftprodukter
- Undvik även att blanda alltför många färger på samma sida om det finns viktig information att läsa

7

Färgkombinationer



Hur påverkas färger av omgivande färger ?

Finns det färger på denna sida som INTE harmonierar ?

8

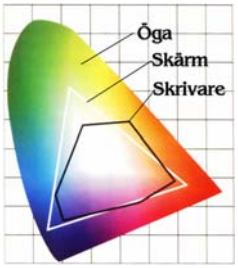
Bilders färgdjup

- Varje pixel i en punktuppbyggd bild innehåller information om sin färg
- Hur många bitar som går åt till detta kallas för **bildens färgdjup**
  - 1 bit, 0 eller 1för en s/v bild (*strecktecknin*g)
  - 4 bitars = 2*2*2*2 = 16 färger
  - 8 bitar ger 256 färger
  - 16 bitar ger **High Color**
  - 24|32 bitar ger **True Color**

9

Färgkalibrering

- Bildskärm - Skrivare
- RGB - CMYK
- Bildskärmen kan återge fler färger än skrivaren
- Olika färggamuter
- *Perceptuell* matchning
- *Kolometrisk* matchning



10

Webbpalett 216

- Plattformsoberoende med 216 säkra färger som återges likadant oavsett om det är på en PC, en Mac eller en UNIX-dator
- Mindre färger, snabbare att ladda sidan
- 24-bitars färgdjup ger distorsion på datorer med dåliga (föråldrade) grafikkort

PAUS 15 min

11

Bilder och bildformat

- Vektorgrafik eller punktgrafik
- Vektorgrafik: grafiken beskrivs med formler och noder (*Bézierkurvor*)
- Framställs i t ex Adobe Illustrator
- Skalbart och utrymmessnålt
- Fungerar ej för fotografier
- **SVG** eller **SWF?**

  `http://www.sitepoint.com/article/v-svg-which-should-choose`

12

Bilder och bildformat

- Punktgrafik innebär att bilderna är uppbyggda av pixlar (bitmapping)
  - Filstorlek = antalet pixlar x färgdjupet
  - Utskriftsstorlek = pixlar / upplösningen
  - *Resampling*: Upsampling-Downsampling
  - *Uppsampling* = interpolering
  - Jämna multiplar vid inscanning-uppsampling

13

Bilder och bildformat

Bitmappade bildformat som t ex:

- bildfil.**bmp**
- bildfil.**tiff**
- bildfil.**psd**

Hög kvalitet och metainformation ger stora filer som tar tid att ladda via nätet

14

Digitala bilder

- Då datorskärmarnas upplösning fortfarande är så låg som 96 dpi så kan vi på datorn ändå inte tillgodagöra oss den höga kvaliteten.
- Det är också nödvändigt att minska bildstorleken vid internetpublicering.
- Stora filer ger lång nedladdningstid.

- Lösningen är **komprimeringsalgoritmer**.
  - Ett exempel är GIF-bildens LZW-komprimering.

15

Bildformat för Internet

- GIF, en gammal trotjänare
- GIF87a och den uppdaterade GIF89a
- 8 bitars färgdjup - 256 färger
- Transparens för 1 färg

- Interlace (sammanflätning)
- Animering genom en serie av GIF-bilder
- Passar bra för diagram, ikoner och teckningar

16

Bildformat för Internet

- JPEG-formatet
- Framtaget av **J**oint **P**hotographic **E**xperts **G**roup
- 24-bitars färgdjup för fotorealistiska bilder
- Förstörande irreversibel kompression
- Passar för fotografier, målningar och liknande
- Progressiva JPEG-bilder i stil med GIF-interlace
- Stödjs precis som GIF av nästan alla webb-läsare

17

Bildformat för Internet

- PNG-bilder
  - **P**ortable **N**etwork **G**raphics
- **PNG** = GIF + JPEG + lite till
- Icke-förstörande kompression utan *ägare* med bättre packratio än GIF/LZW
- 16-bitars alfa-kanal för transparens
- Framtidens bildformat?

18

Framtidens format

- Det har i flera år talats om att PNG ska ta över
- Det har gått lite trögt

- Nu finns också **JPEG2000**
- En vidareutveckling av JPEG
- Förbättrad komprimering:
  - högre packratio – mindre filer
  - högre kvalitet vid kraftig komprimering

19

RLE - en gammal algoritm

- **R**un **L**enght **E**ncoding

- I en bitmappad bild är färgen på varje enskild pixel representerad med ett visst bitmönster
- Om detta bitmönster upprepar sig t ex 224 ggr kan RLE-kodningen bli: 11001000 (x) 11100000
- Detta minskar i princip storleken från 224 bytes till 2 bytes

20

Ett (för mig) nytt problem

Vilket är nästa tal i följande talserie:

1, 11, 21, 1211, 111221, 312211, 13112221 …

PAUS 15 min

21

Var kan man hitta bilder?

2000 nya bilder/min

2 miljoner geotaggade fotografier

- http://www.flickr.com/

22

Flickr - geotagging

+ 20 000 bilder från Sri Lanka

http://www.flickr.com/map/

22 stycken under *"Urban Photos"*

23

Aliasing

Alias

24

Aliasing

- I bilder **< 500KB**
- Ögat kan se pixeluppbyggnaden
- Framför allt problem med nästan horisontella eller nästan vertikala linjer
- Syns tydligt vid skarpa linjer och detaljer med hög kontrast mot bakgrunden
- Som motmedel finns : **anti-aliasing**

25

Bilder i Java

javax.swing.ImageIcon

Läs mera på:
`http://java.sun.com/docs/books/tutorial/uiswing/misc/icon.html`

java.awt.Image

Läs mera på:
`http://www.particle.kth.se/~lindsey/JavaCourse/Book/Part1/Java/Chapter06/images.html`

26

Java 2D API

- Ett senare tillkommet klassbibliotek för mer avancerad grafik
  - Flera former som kan ritas ut
  - De kan ritas ut på **olika** sätt
  - Det som ritas ut kan *transformeras*
  - Det går att sätta *rendering hints*
- Att slå på/av rendering hints påverkar motsättningen mellan **kvalitet** och **snabbhet**

27

Java 2D API

- För att kunna få tillgång till geometriska figurer och rittekniker i Java 2D API så krävs följande typomvandling:

```
public void paintComponent(Graphics g){
   super.paintComponent(g);
   Graphics2D g2D = (Graphics2D)g;
   g2D…
```

28

Java 2D API

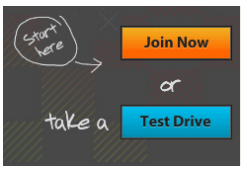
- För att kunna styra utritningen när det gäller motsättningen mellan kvalitet och snabbhet så finns en klass som heter **RenderingHints**

```
RenderingHints hints =
       new RenderingHints(null);
hints.put(key, value);
g2D.setRenderingHints(hints);
```

29

Bildhantering på nätet

- Adobe ***PhotoShop Express***
- 2 GB gratis lagringsutrymme



https://www.photoshop.com/express/

www.photoshopexpresstechniques.com/

30

Bildhantering i Java

- Java Image Editor
  - Allt är skrivet i Java
  - Bara att ladda hem från
    `http://www.jhlabs.com/ie/index.html`
- Att skriva egna bildfilter i Java
  - Uppgift3e
  - Uppgift3f

31

Bildhantering i Java

```
import java.applet.*;
import java.awt.*;
import javax.swing.*;


public class Föreläsning3 extends JApplet {
  private Image bild;
  private BildPanel bildPanel;

  public void init(){
    setSize(300,300);
    bild = getImage(getDocumentBase(),
                              ”minbild.typ”));
```

32

Bildhantering i Java

```
     MediaTracker mt = new MediaTracker(this);
     mt.addImage(bild,1);
     try{
         mt.waitForAll();
     }catch(Exception e){
         System.out.println("BILDPROBLEM:" + e);
         System.exit(1);
     }
    bildPanel = new BildPanel(bild);
    this.getContentPane().add(bildPanel);
  }//init

}//Föreläsning3
```

33

Bildhantering i Java

```
public class BildPanel extends JPanel{
  private Image bild;

  public BildPanel(Image bild){
    super();
    this.bild = bild;
    this.setBackground(new Color(0,0,0));
  }

  public void paintComponent(Graphics g){
    super.paintComponent(g);
    g.drawImage(bild,10,30,this);
  }
}//BildPanel
```

34

Allt för idag

- Missa inte morgondagens lektion!
- Ska vi bjuda hit någon från Adobe?

35