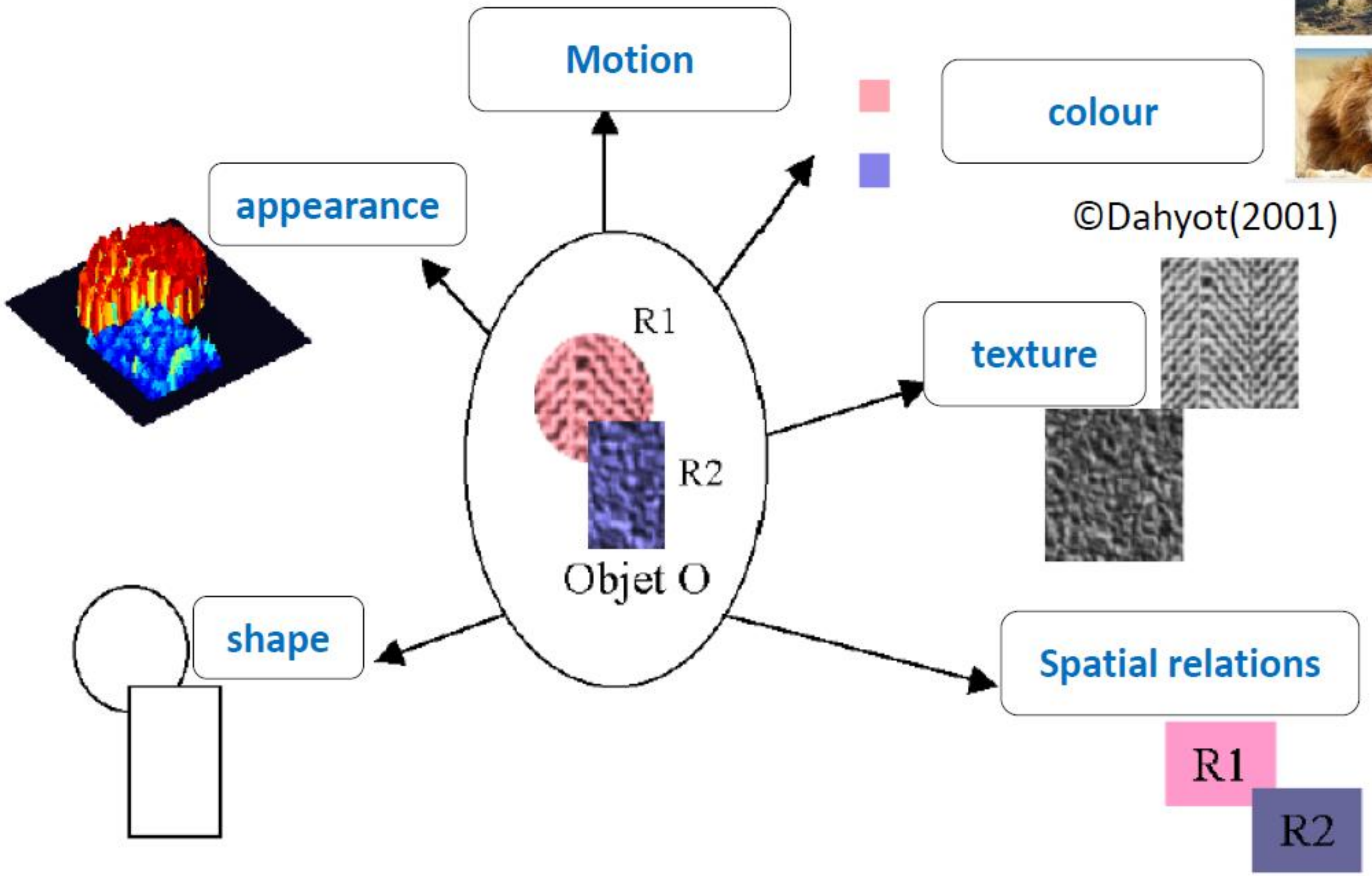
Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

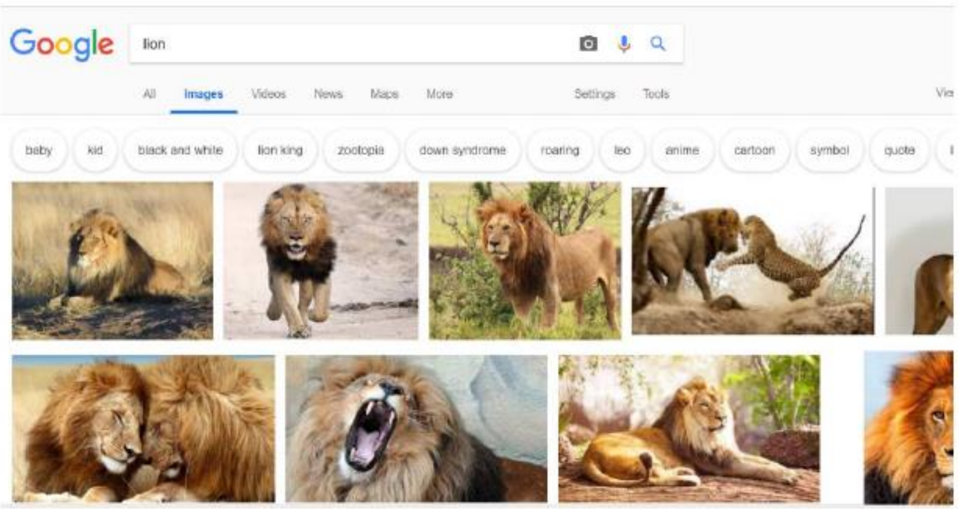
CS7GV1 Computer vision

Visual attributes

Dr. Martin Alain

Visual attributes





Example of application:
Indexing and retrieval of image database

©Szeliski(2020)

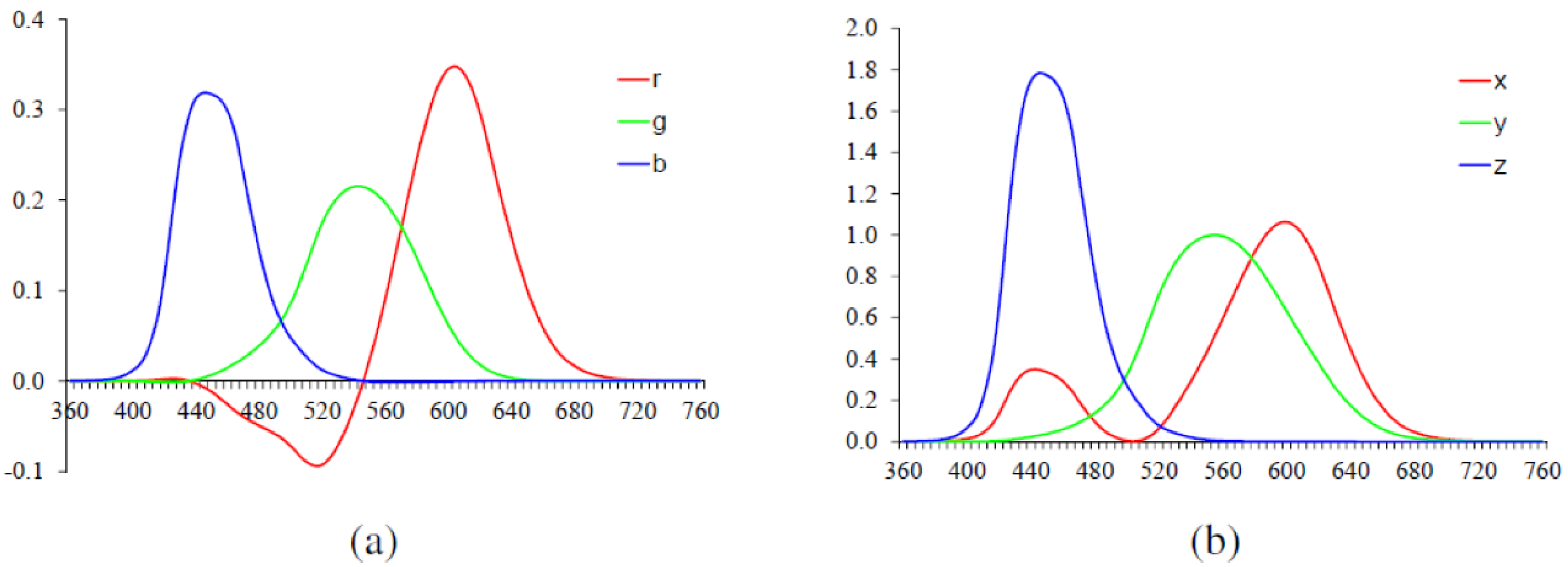


Figure 2.29 *Standard CIE color matching functions: (a) $\bar{r}(\lambda)$, $\bar{g}(\lambda)$, $\bar{b}(\lambda)$ color spectra obtained from matching pure colors to the R=700.0nm, G=546.1nm, and B=435.8nm primaries; (b) $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, $\bar{z}(\lambda)$ color matching functions, which are linear combinations of the $(\bar{r}(\lambda), \bar{g}(\lambda), \bar{b}(\lambda))$ spectra.*

https://github.com/Roznn/Detection-of-Changing-Objects-in-Camera-in-Motion-Video

Octave-Matlab

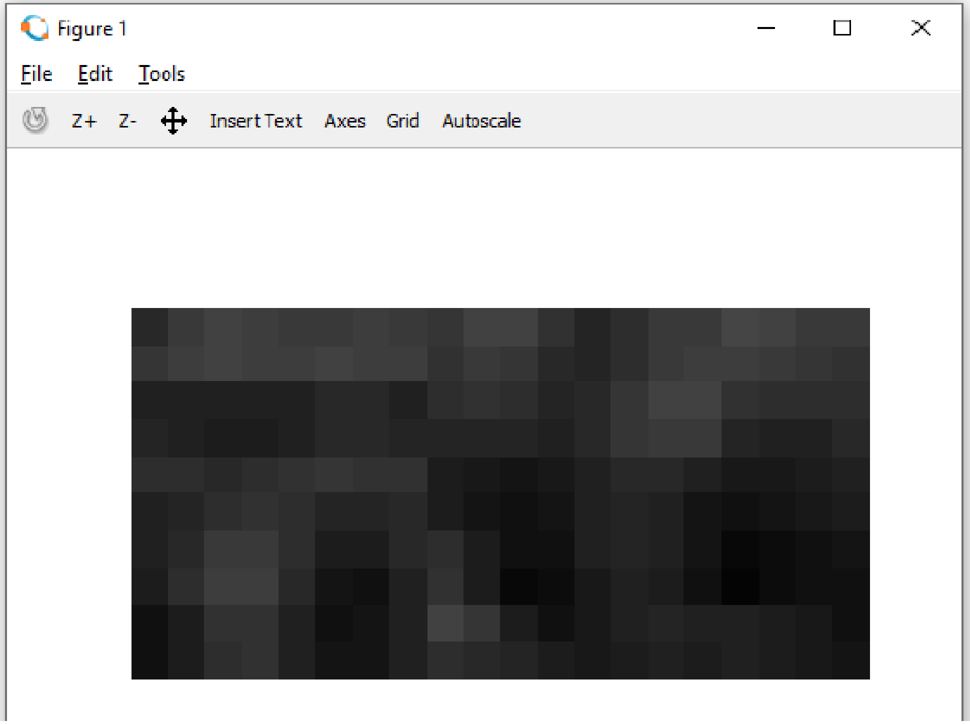
Red channel

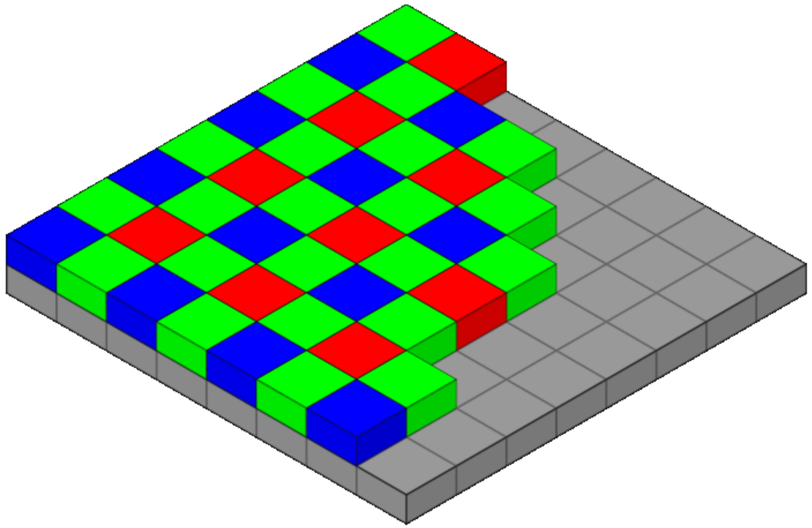
```
figure;imshow(I0(:,:,1))
```

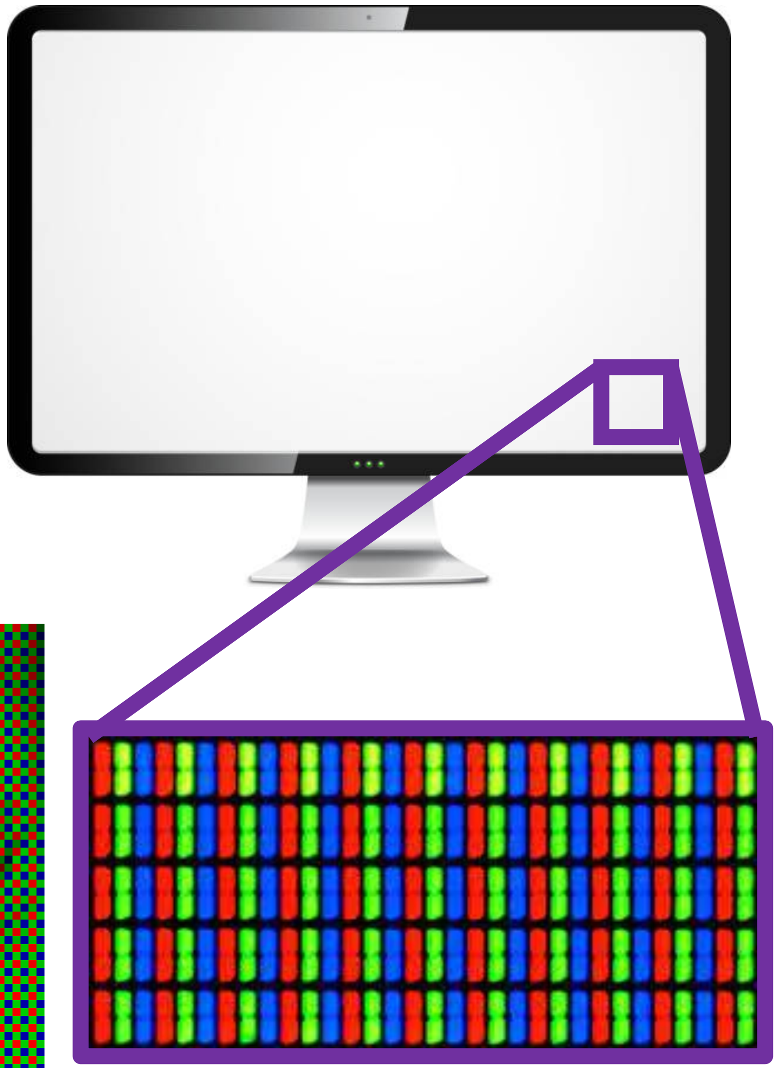
```
I0(1:10,1:20,3)
imshow(I0(1:10,1:20,3))
```

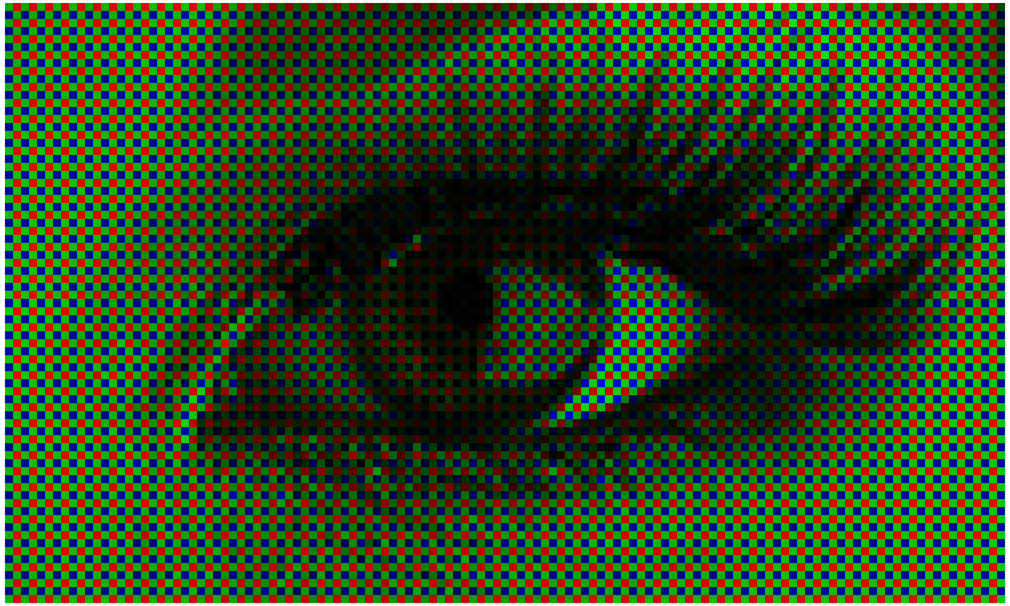
Blue channel

```
>> I0(1:10,1:20,3)
ans =

   43   56   64   60   57   58   60   58   55   67   65   48   39   47   57   58   71   64   59   56
   54   62   66   60   60   64   63   60   48   56   55   41   36   47   59   62   63   57   53   51
   35   35   34   32   34   40   40   33   44   50   47   39   41   53   65   67   51   46   44   46
   37   34   30   31   35   41   41   37   37   38   36   35   40   53   59   58   37   33   35   41
   47   44   42   46   50   52   50   51   28   24   22   25   33   42   42   35   26   26   28   35
   34   37   44   48   44   36   37   43   31   22   16   20   33   39   33   23   17   20   25   30
   32   42   56   59   46   28   29   40   45   29   16   18   32   38   32   22   10   15   19   22
   29   44   61   62   42   20   19   33   51   28    9   12   25   33   28   19    5   12   16   16
   16   31   49   51   35   19   21   32   66   53   31   19   24   35   39   34   34   30   24   18
   17   31   47   50   35   20   22   33   44   43   36   28   27   31   33   29   33   31   27   23

>> imshow(I0(1:10,1:20,3))
>>
```









resulting RGB image: (human) visible light image

false colour representation of „bee-colours“

Color spaces: RGB

Default color space

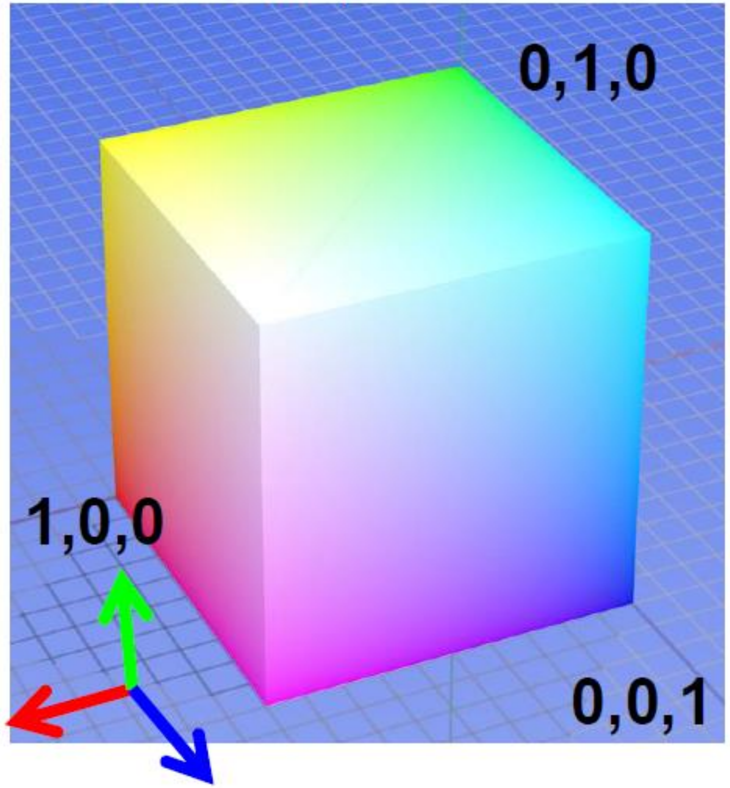
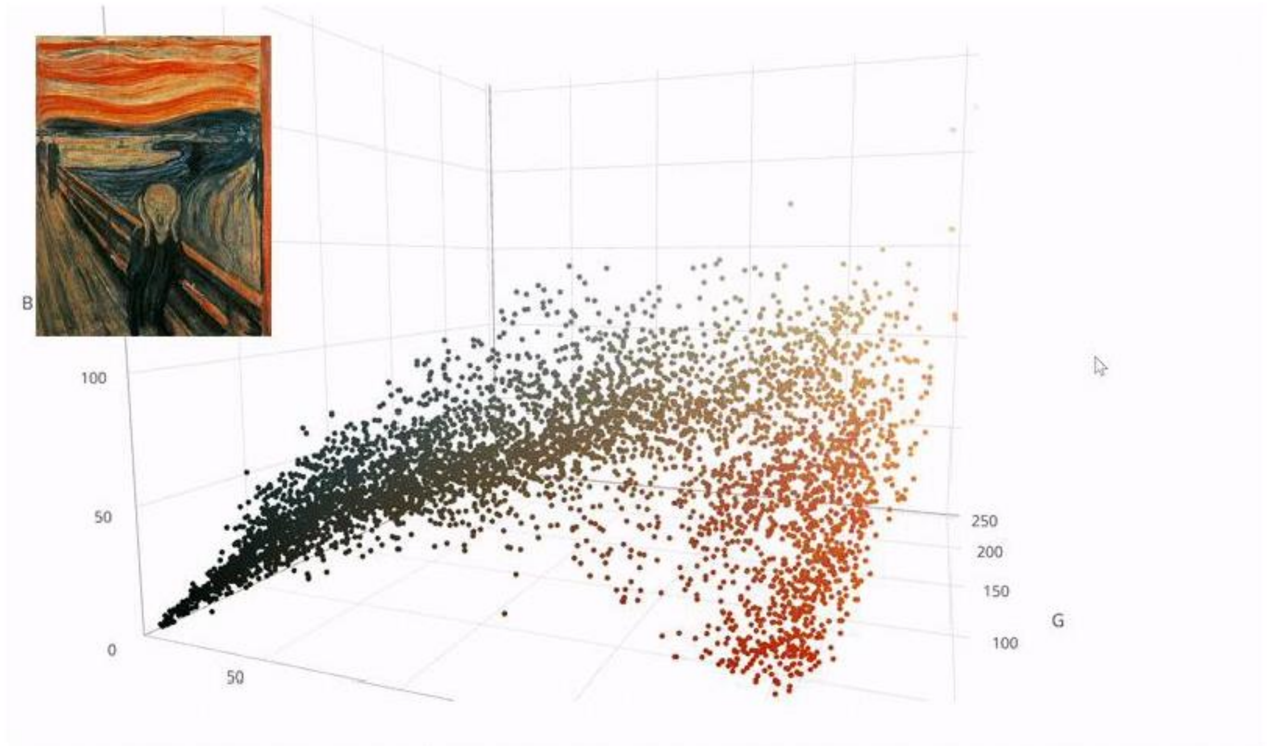


Image from: http://en.wikipedia.org/wiki/File:RGB_color_solid_cube.png



Credit animation: https://github.com/zumbov2/colorfindr

Some drawbacks
- Strongly correlated channels
- Non-perceptual

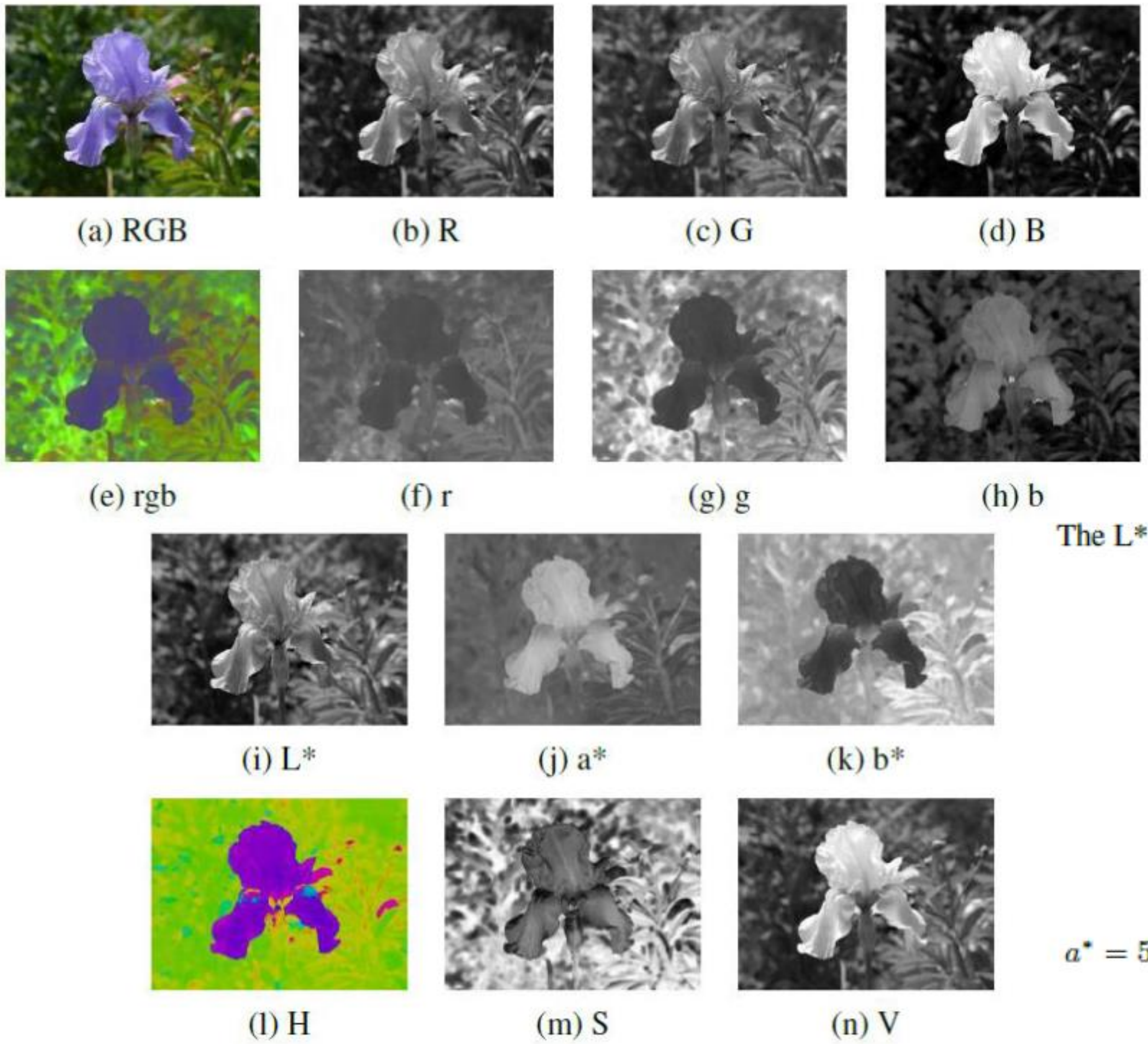


Figure 2.33 *Color space transformations: (a–d) RGB; (e–h) rgb. (i–k) L*a*b*; (l–n) HSV. Note that the rgb, L*a*b*, and HSV values are all re-scaled to fit the dynamic range of the printed page.*

color ratios:

$$r = \frac{R}{R+G+B}, \quad g = \frac{G}{R+G+B}, \quad b = \frac{B}{R+G+B} \tag{2.117}$$

The L* component of *lightness* is defined as

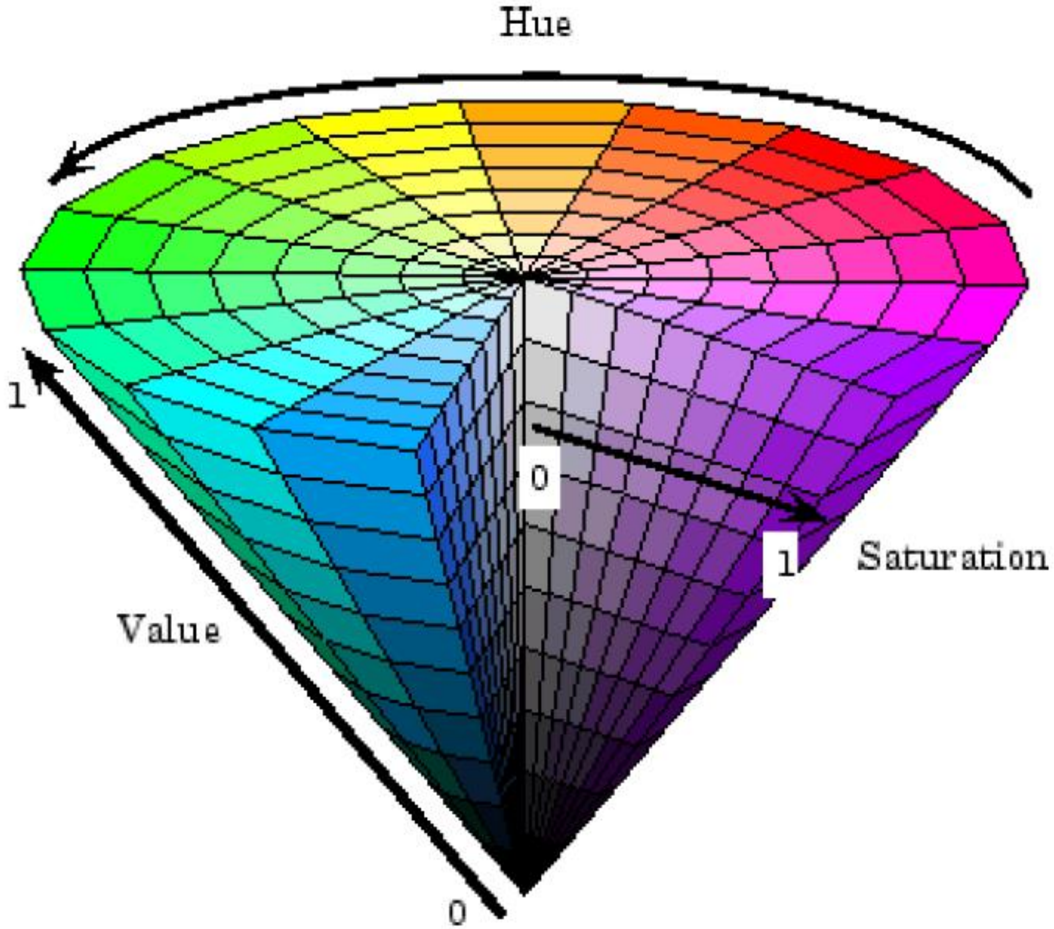
$$L^* = 116 f\left(\frac{Y}{Y_n}\right), \tag{2.106}$$

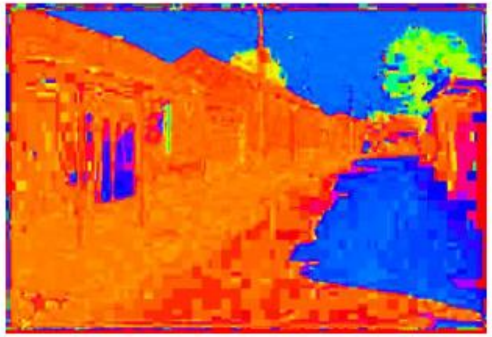
$$f(t) = \begin{cases} t^{1/3} & t > \delta^3 \\ t/(3\delta^2) + 2\delta/3 & \text{else}, \end{cases} \tag{2.107}$$

$$a^* = 500\left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right] \quad \text{and} \quad b^* = 200\left[f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right], \tag{2.108}$$

Color spaces: HSV

Intuitive color space





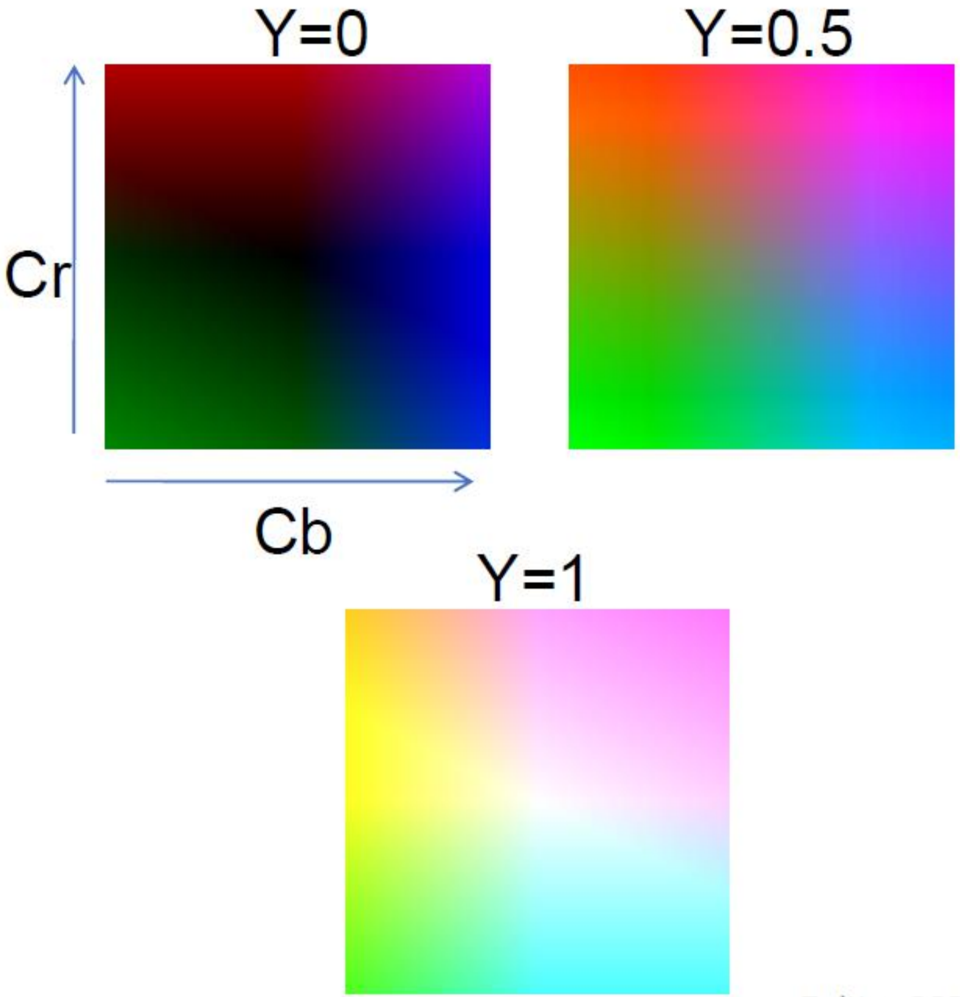
H
(S=1,V=1)

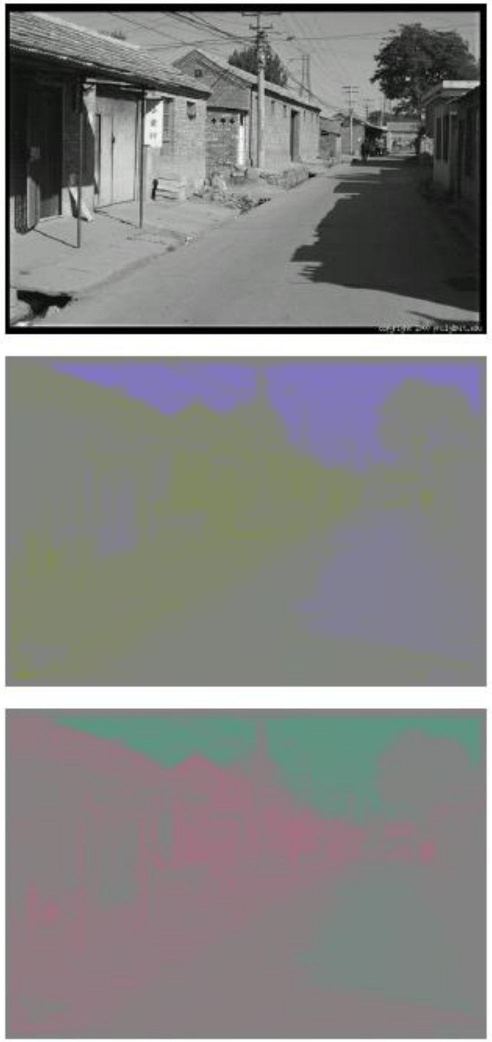
S
(H=1,V=1)

V
(H=1,S=0)

Fast to compute, good for compression, used by TV

Color spaces: YCbCr





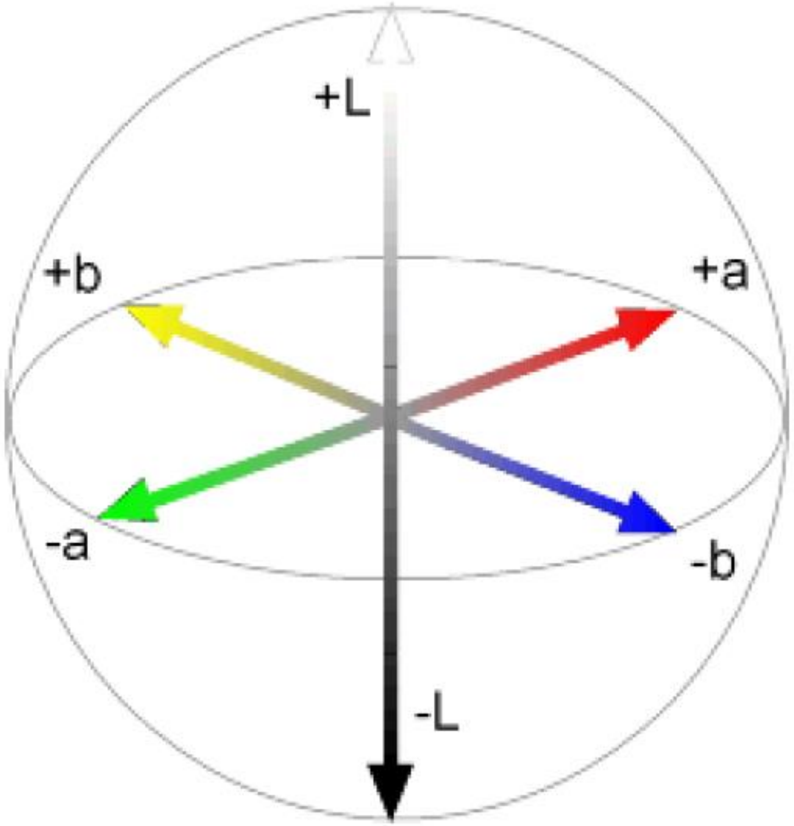
Y
(Cb=0.5,Cr=0.5)

Cb
(Y=0.5,Cr=0.5)

Cr
(Y=0.5,Cb=05)

“Perceptually uniform”* color space

Color spaces: L*a*b*



L
(a=0,b=0)

a
(L=65,b=0)

b
(L=65,a=0)

Visual attribute: shape

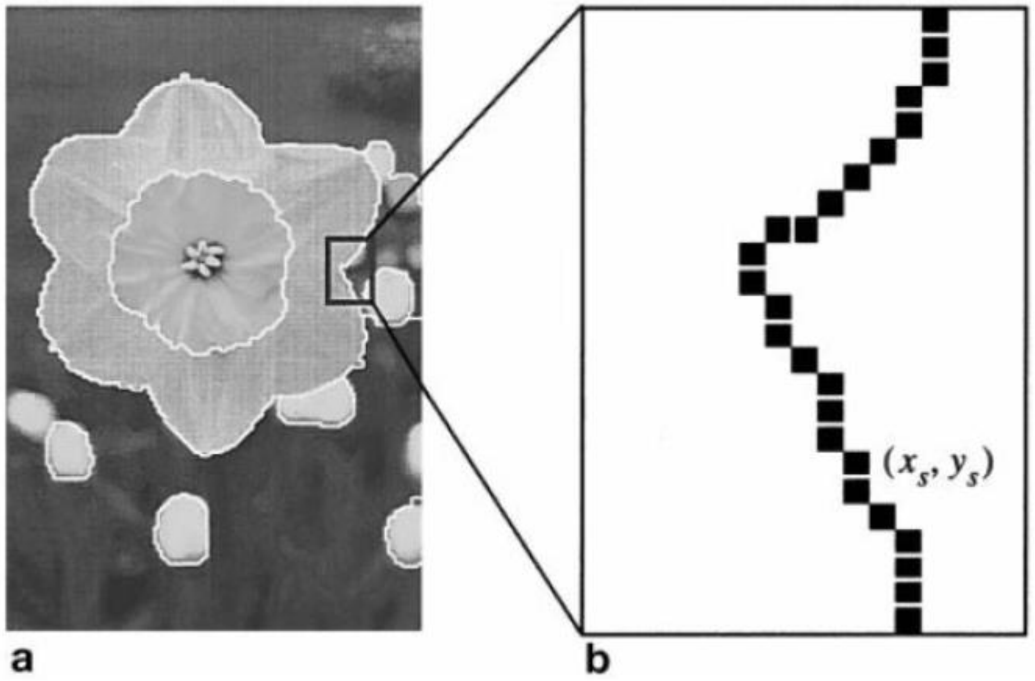
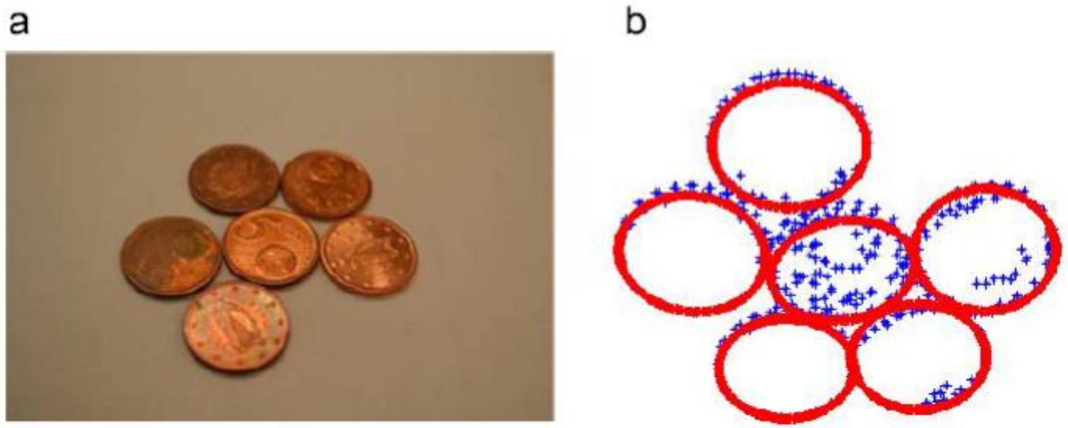


Fig. 3a,b. The object boundary is represented as a discrete coordinate chain. **a** A segmented flower image. **b** A small portion of the outer boundary of flower

Shapes are often described as curves matching the contours of the object, or local descriptors of the contours.

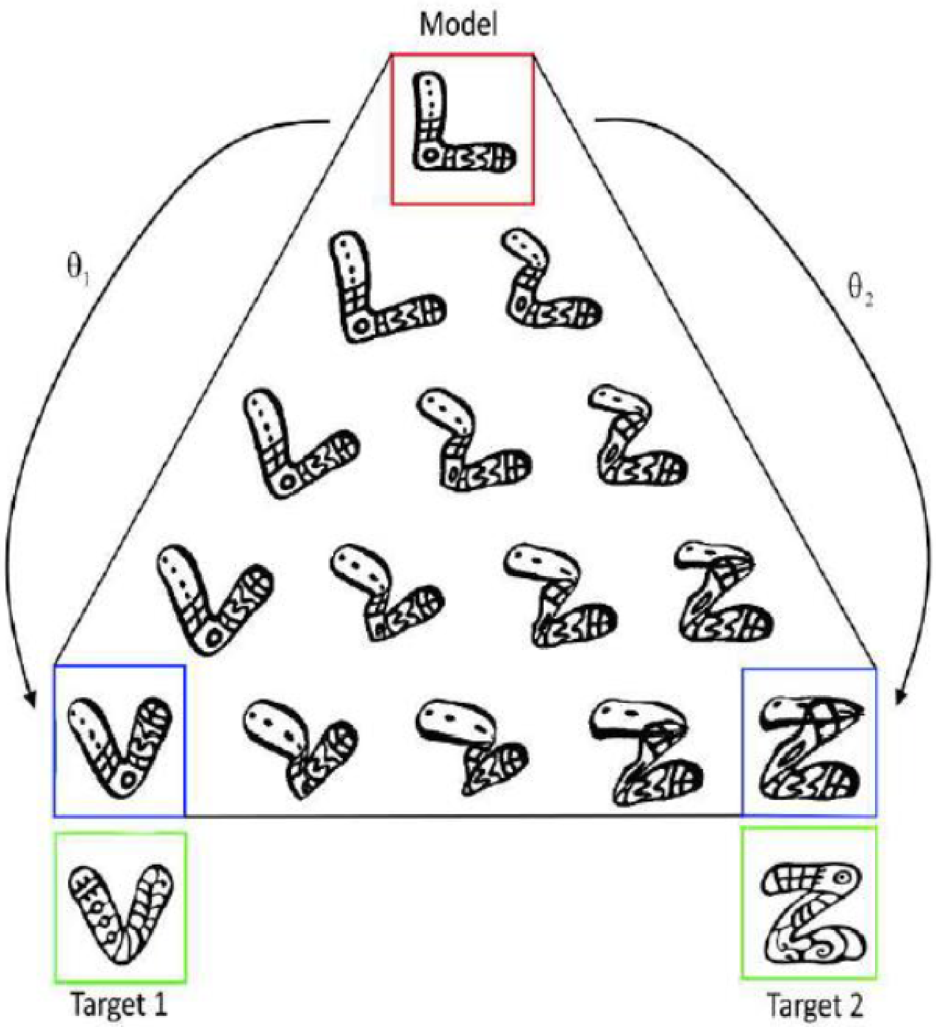
NeTra: A toolbox for navigating large image databases, Wei-Ying Ma, B. S. Manjunath Multimedia Systems 7: 184–198 (1999) https://vision.ece.ucsb.edu/sites/vision.ece.ucsb.edu/files/publications/99ACMNeTra.pdf

Visual attribute: shape



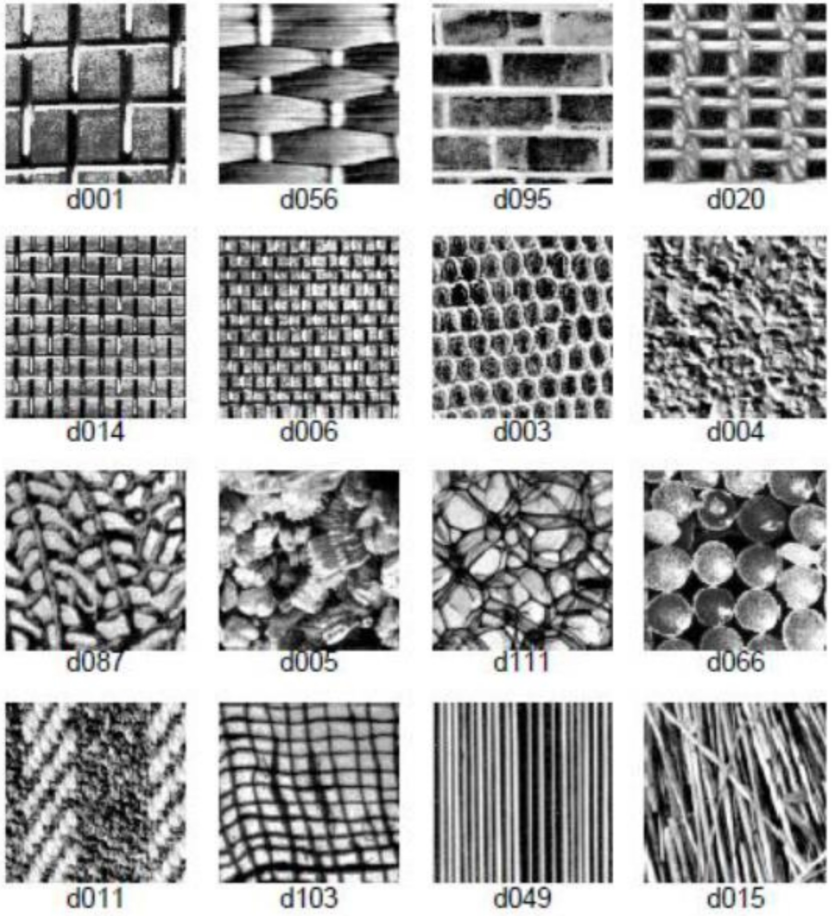
[Robust Ellipse Detection With Gaussian Mixture Models]()
C. Arellano and R. Dahyot, Pattern Recognition, Volume 58, pages 12-26, October 2016

Visual attribute: shape



Shape Registration with Directional Data
M. Grogan and R. Dahyot, Pattern Recognition, Volume 79, July 2018, Pages 452-466
[DOI:10.1016/j.patcog.2018.02.021](https://doi.org/10.1016/j.patcog.2018.02.021)

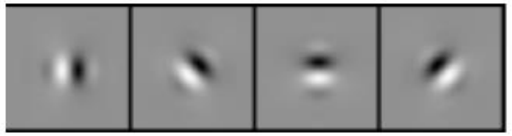
Visual attribute: texture



J.R. Smith, *Integrated Spatial and Feature Image Systems: Retrieval Analysis and Compression*, Ph.D. thesis, Columbia University, USA, 1997
http://www.ee.columbia.edu/dvmm/publications/PhD_theses/jrsmith-thesis.pdf

https://multibandtexture.recherche.usherbrooke.ca/original_brodatz.html

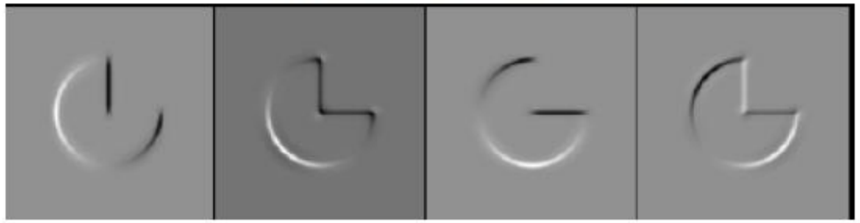
Texture



Steerable filters

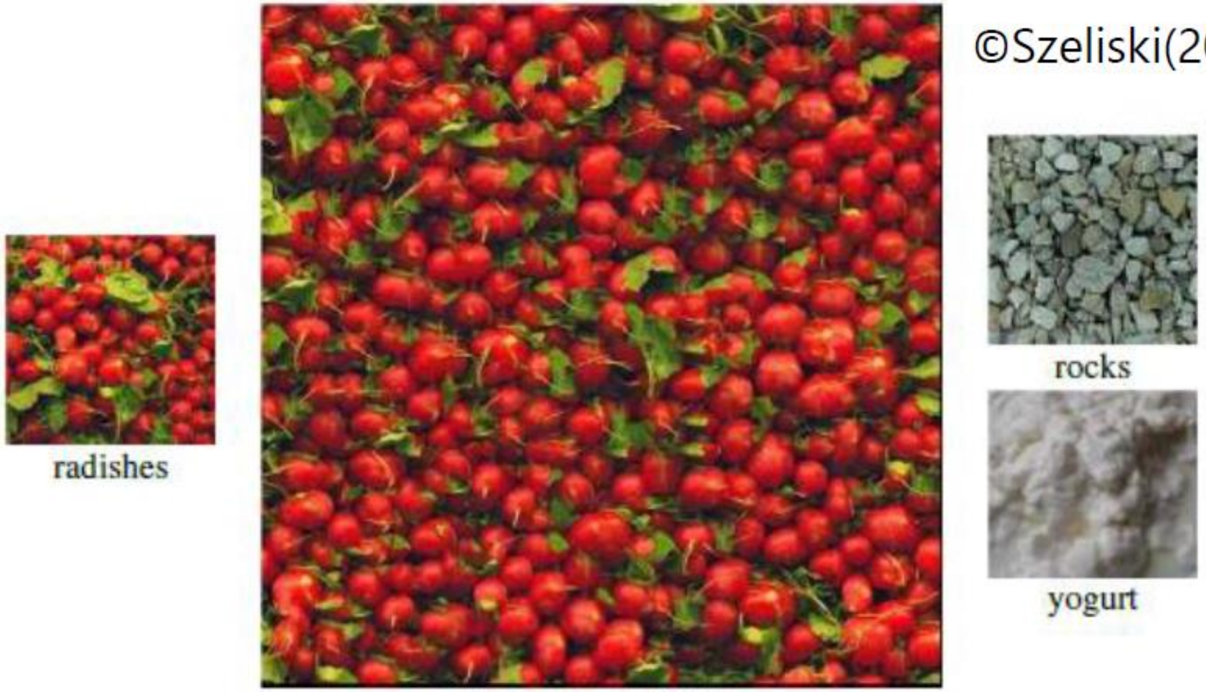
image

Image * Steerable filters



10.5 Texture analysis and synthesis 641

©Szeliski(2020)



(a) (b) (c)

Figure 10.50 *Texture synthesis: (a) given a small patch of texture, the task is to synthesize (b) a similar-looking larger patch; (c) other semi-structured textures that are challenging to synthesize. (Images courtesy of Alyosha Efros.)*

Texture

Steerable filters /wavelets are pre-set filters (no machine learning) that are used to extract local features on the input image using the operation '**convolution**'.

MathWorks® Products Solutions Academia Support Community Events

Wavelet Toolbox Search MathWorks

Wavelet Toolbox™ provides functions and apps for analyzing and synthesizing signals and images. The toolbox includes algorithms for continuous wavelet analysis, wavelet coherence, synchrosqueezing, and data-adaptive time-frequency analysis. The toolbox also includes apps and functions for decimated and nondecimated discrete wavelet analysis of signals and images, including wavelet packets and dual-tree transforms.

https://uk.mathworks.com/products/wavelet.html

PyWavelets - Wavelet Transforms in Python

PyWavelets is open source wavelet transform software for Python. It combines a simple high level interface with low level C and Cython performance.

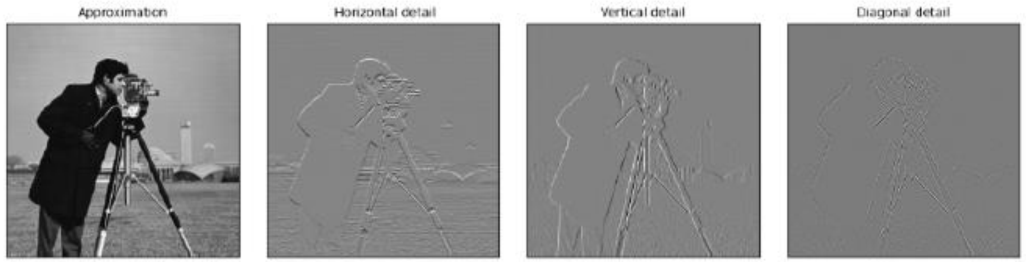
PyWavelets is very easy to use and get started with. Just install the package, open the Python interactive shell and type:

```
>>> import pywt
>>> cA, cD = pywt.dwt([1, 2, 3, 4], 'db1')
```

Voilà! Computing wavelet transforms has never been so simple :)

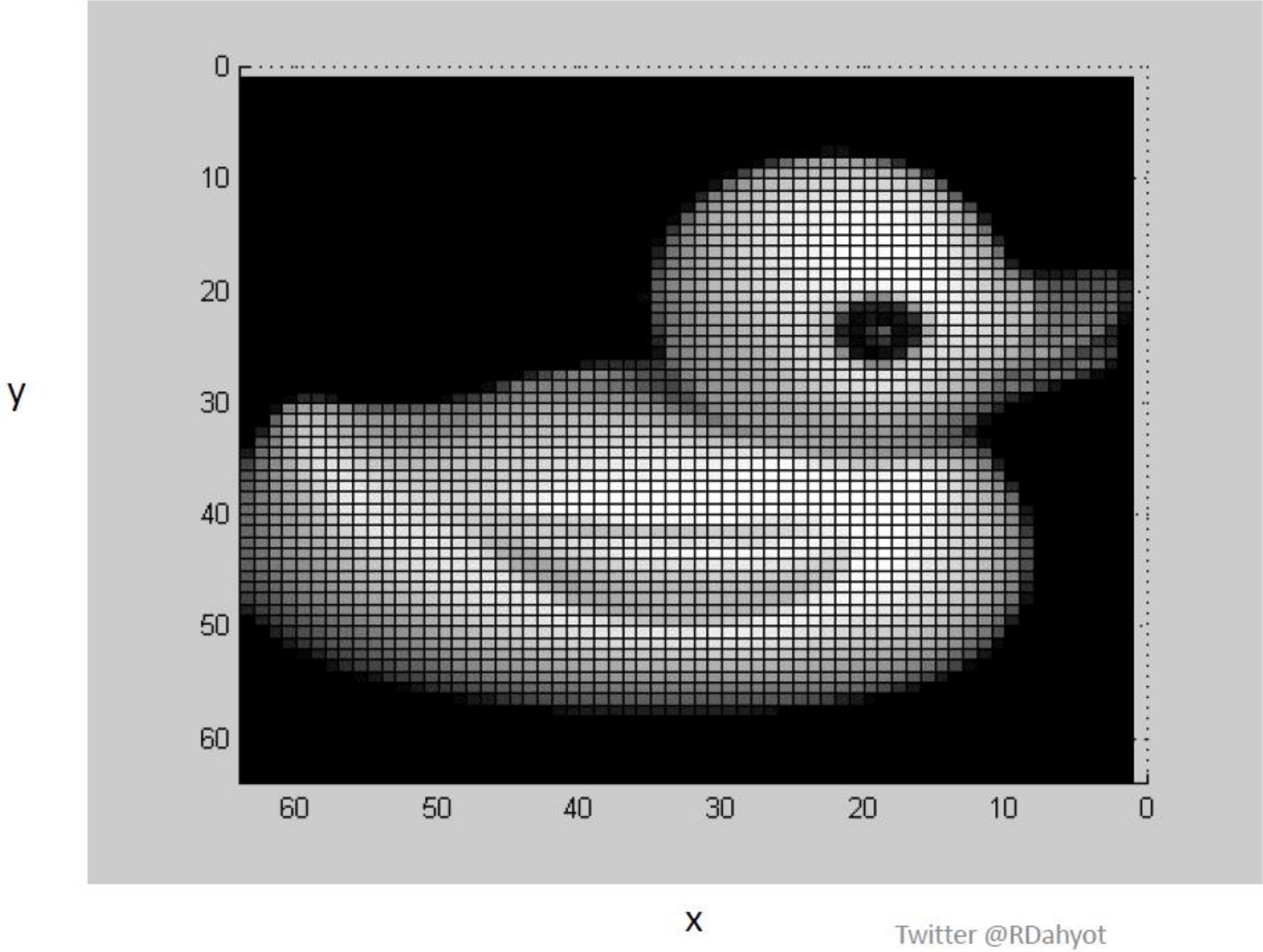
Here is a slightly more involved example of applying a digital wavelet transform to an image:

```
import numpy as np
import matplotlib.pyplot as plt

import pywt
import pywt.data


# Load image
original = pywt.data.camera()

# Wavelet transform of image, and plot approximation and details
titles = ['Approximation', ' Horizontal detail',
          'Vertical detail', 'Diagonal detail']
coeffs2 = pywt.dwt2(original, 'bior1.3')
LL, (LH, HL, HH) = coeffs2
fig = plt.figure(figsize=(12, 3))
for i, a in enumerate([LL, LH, HL, HH]):
    ax = fig.add_subplot(1, 4, i + 1)
    ax.imshow(a, interpolation="nearest", cmap=plt.cm.gray)
    ax.set_title(titles[i], fontsize=10)
    ax.set_xticks([])
    ax.set_yticks([])

fig.tight_layout()
plt.show()
```



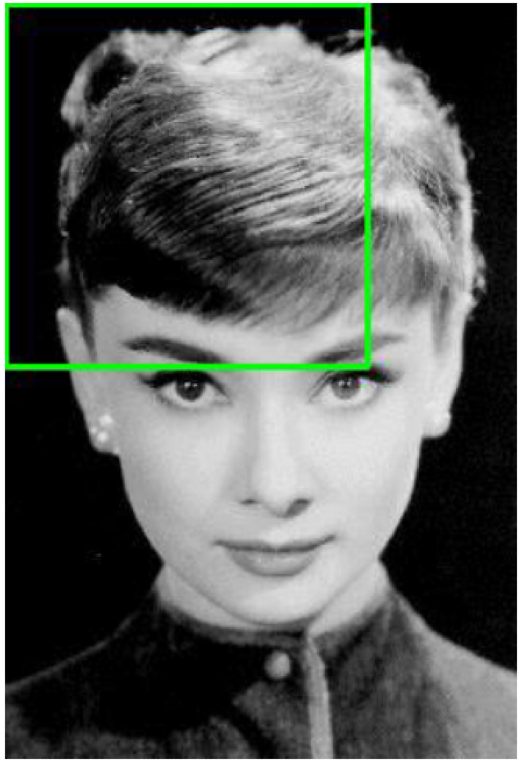
https://pywavelets.readthedocs.io/en/latest/

Visual attribute: appearance



Appearance considers an **image I(x,y) as a surface** that can be described globally (e.g. as a vector containing all pixel intensities), or locally considering spatial derivatives of the surface image.

Visual attribute: appearance



Example of sliding window

Template matching: Having a representative image (of size of the green box) of a class of objects of interest (this image acts as a template for the class), template matching is a technique that measure a similarity between the template and the content of the sliding window at a given position. This computation of similarity is repeated for each position of the sliding window providing a heat map that helps in localising the area of the image that is the most similar to the template.

Visual attribute: appearance

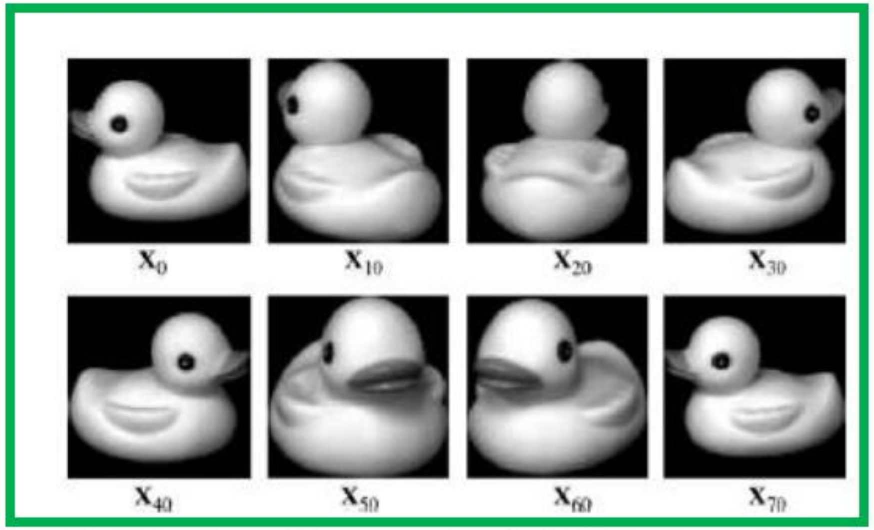
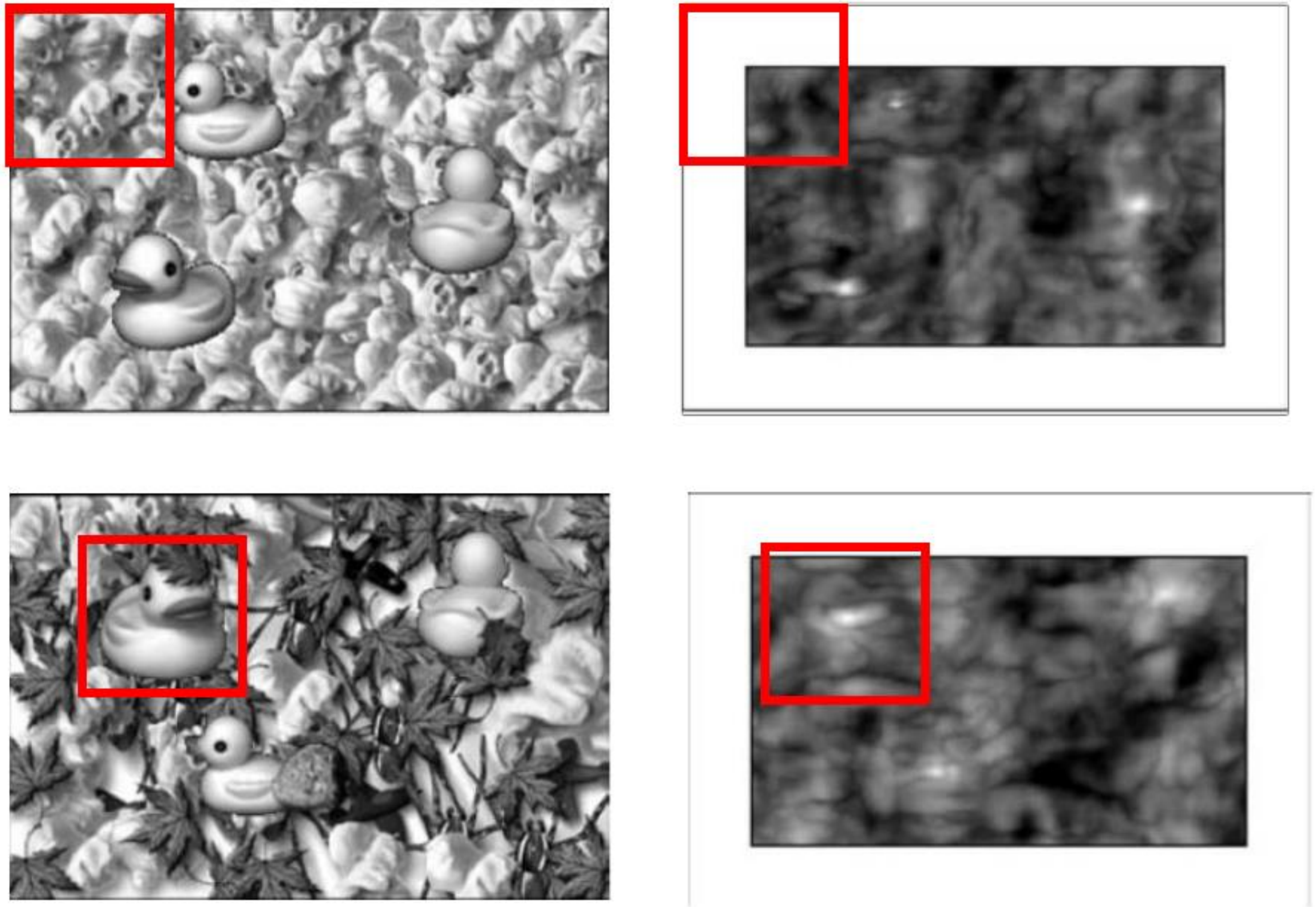


Fig. 2 Sample training images from the COIL database [25]

Detection (heat) maps



[A Bayesian approach to object detection using probabilistic appearance-based models](https://www.scss.tcd.ie/Rozenn.Dahyot/pdf/article2333.pdf)
R. Dahyot, P. Charbonnier and F. Heitz, Pattern Analysis and Applications, Vol. 7, No 3, pp. 317-332, December 2004.
DOI:10.1007/s10044-004-0230-5

https://www.scss.tcd.ie/Rozenn.Dahyot/pdf/article2333.pdf

Visual attribute: spatial relations

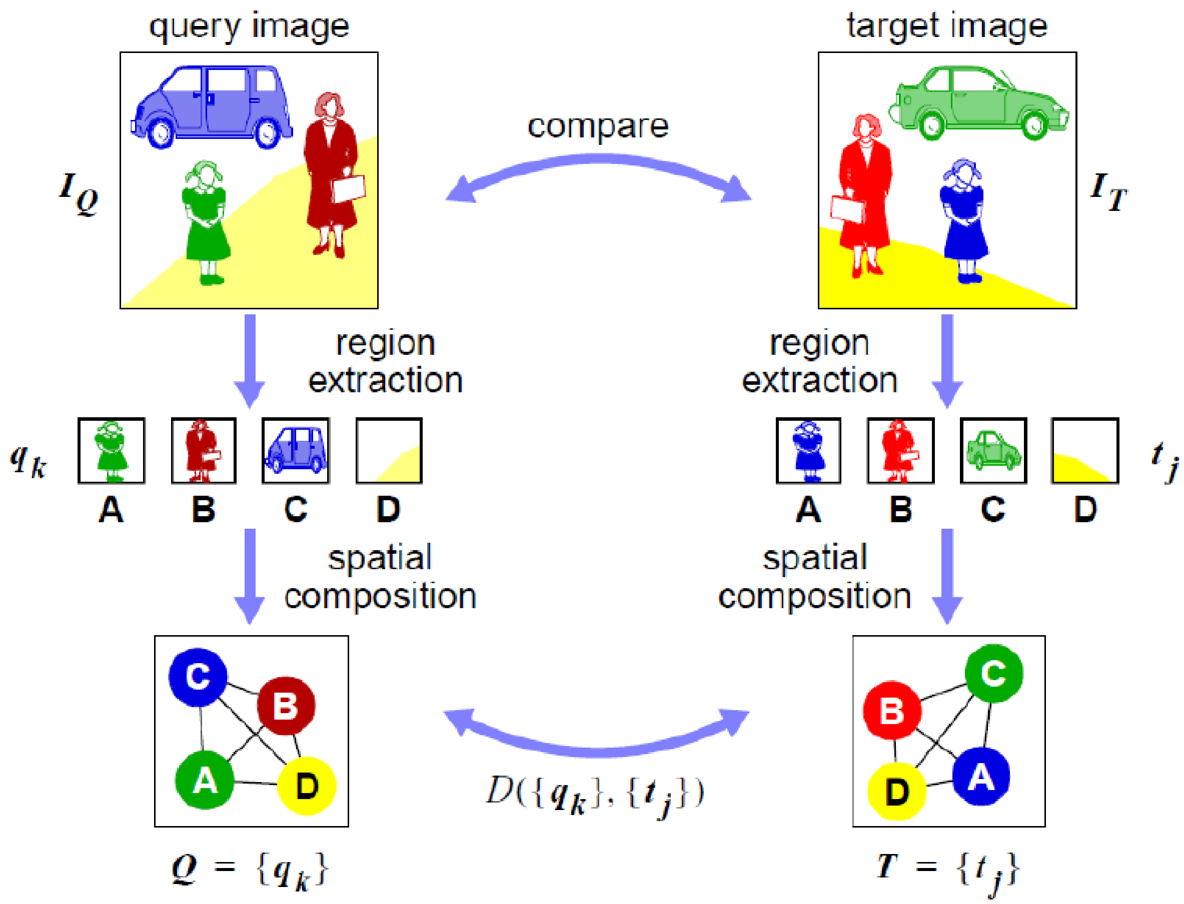


Figure 1-5: Integrated spatial and feature query.

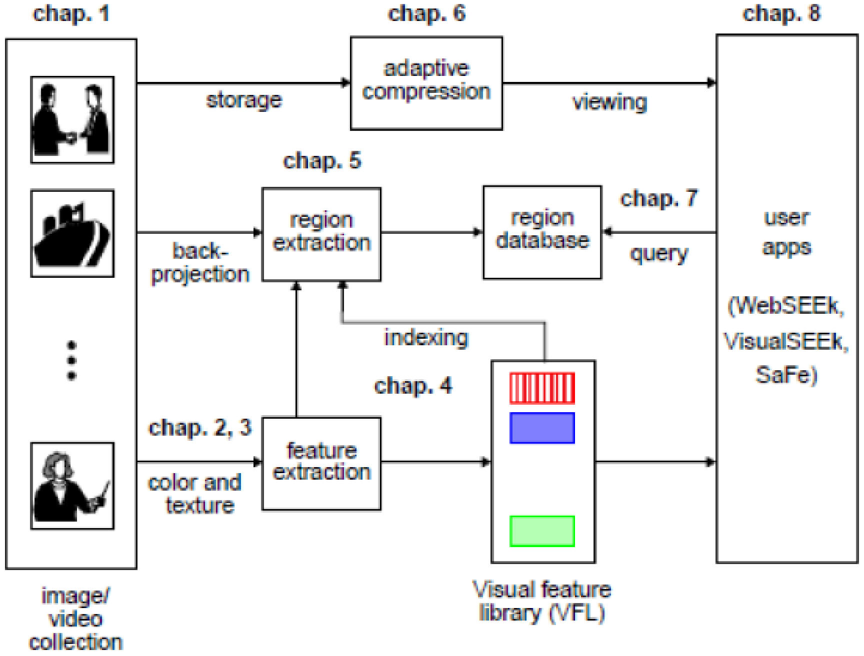
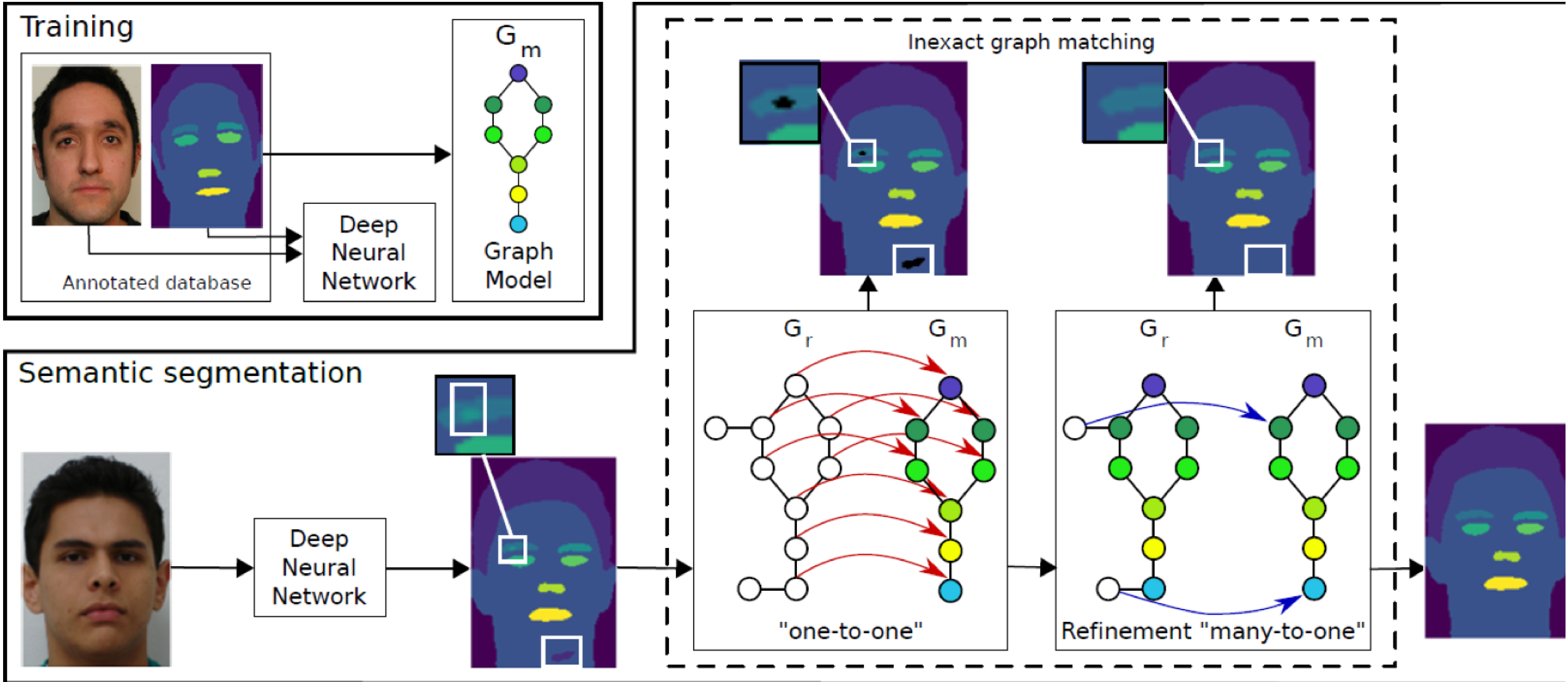


Figure 1-6: Map outlining the work addressed in this thesis.

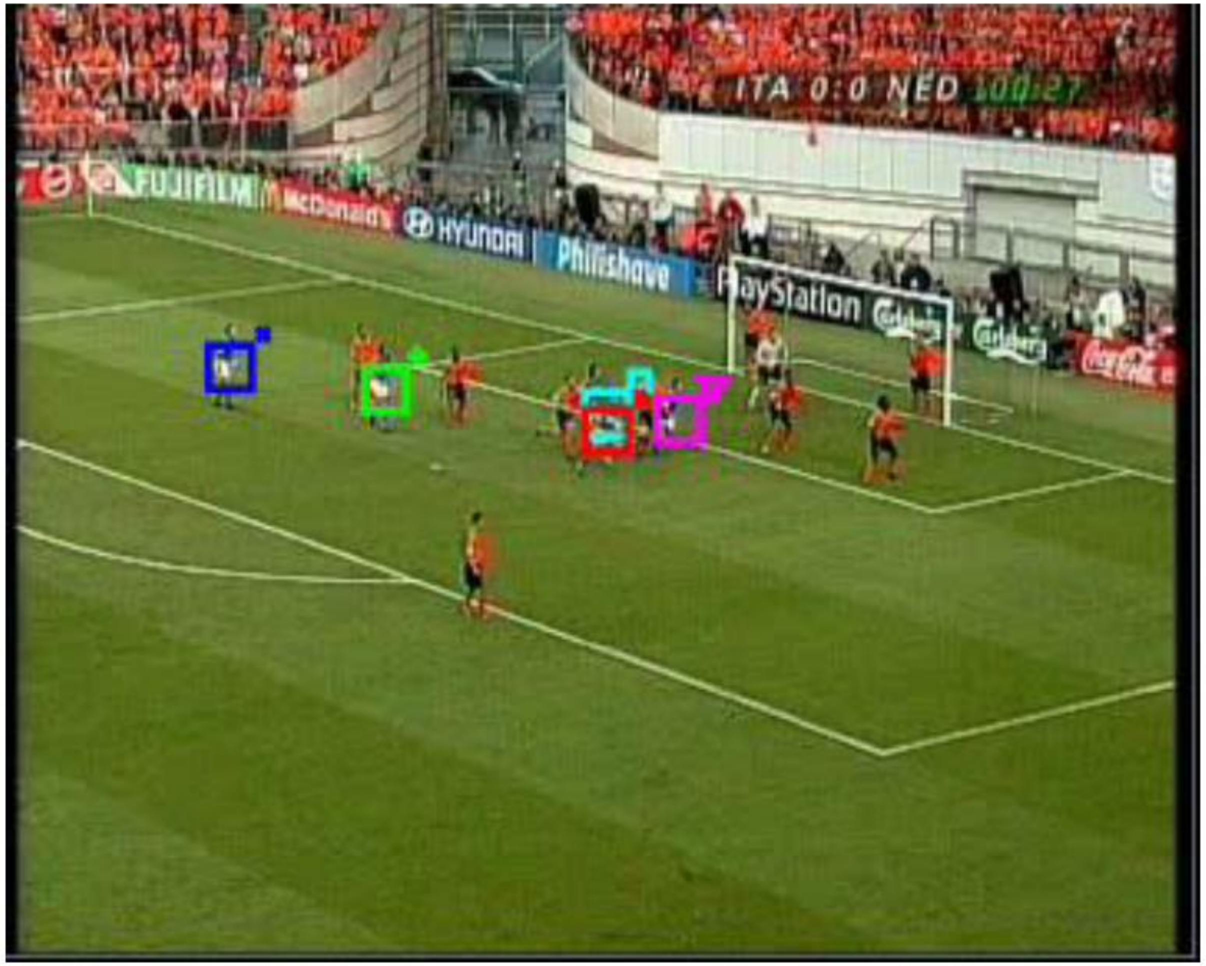
J.R. Smith, *Integrated Spatial and Feature Image Systems: Retrieval Analysis and Compression*, Ph.D. thesis, Columbia University, USA, 1997
http://www.ee.columbia.edu/dvmm/publications/PhD_theses/jrsmith-thesis.pdf

Spatial relations

Semantic image segmentation based on spatial relationships and inexact graph matching
J. Chopin, J.-B. Fasquel, H. Mouchere, R. Dahyot and I. Bloch,
International Conference on Image Processing Theory, Tools and Applications (IPTA 2020), 2020
https://hal.archives-ouvertes.fr/hal-02916165/



Visual attribute: motion



Off-line Multiple Object Tracking using Candidate Selection and the Viterbi Algorithm
F. Pitie, S-A. Berrani, R. Dahyot and A. Kokaram, in IEEE International Conference on Image Processing (**ICIP'05**), Genoa, Italy. DOI:10.1109/ICIP.2005.1530340