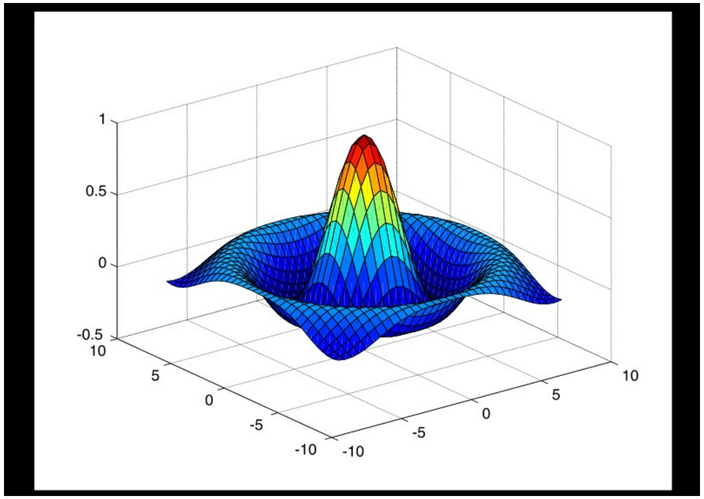
INF555

Fundamentals of 3D



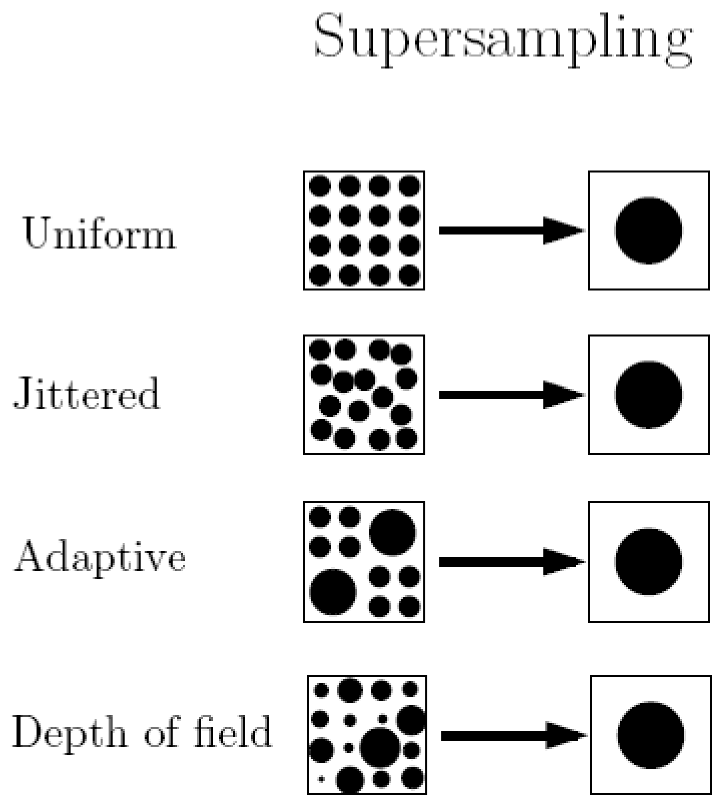
Lecture 7:
Colors
Randomized algorithms
RANSAC/MINIBALL

2nd November 2011

Frank Nielsen
nielsen@lix.polytechnique.fr

Supersampling: Averaging

The essence of pixels: Integration !



Uniform versus non-uniform sampling
(raytracing software: Povray, Blender)

Color and perception

Commission Internationale de
l' ´ Eclairage **in French, CIE, *http:// www.cie.co.at/***

$$\lambda_{\text{red}} = 700\ nm,$$
$$\lambda_{\text{green}} = 546.1\ nm,$$
$$\lambda_{\text{blue}} = 435.8\ nm.$$

Violet	380–440 nm
Blue	440–485 nm
Cyan	485–500 nm
Green	500–565 nm
Yellow	565–590 nm
Orange	590–625 nm
Red	625–740 nm

Color and measurements

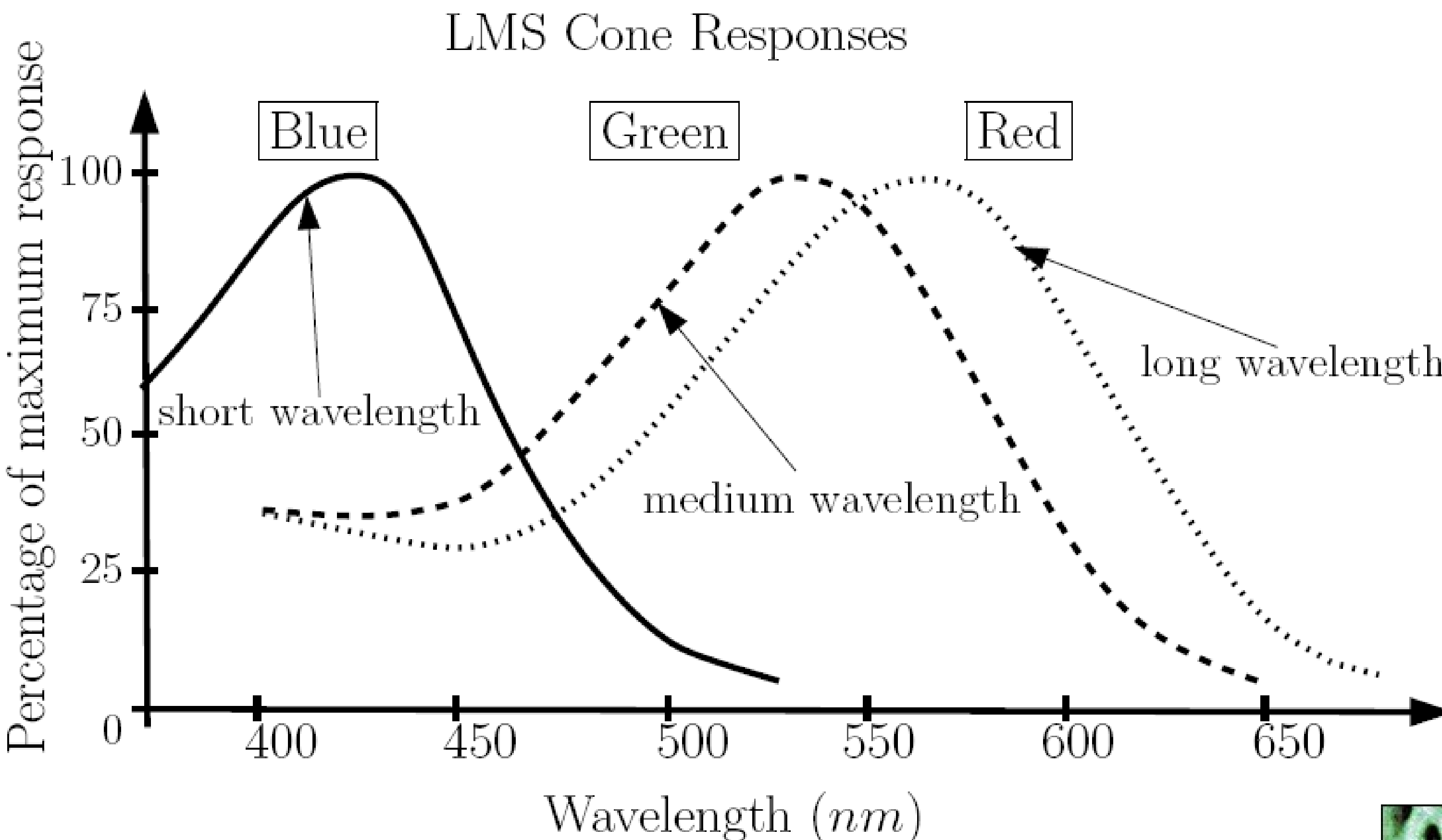
A **candela unit** is defined as the luminous intensity of a light source emitting a monochromatic light of frequency 540 × 10^12 Hz that has a radiant intensity of 1 683 watt per steradian.

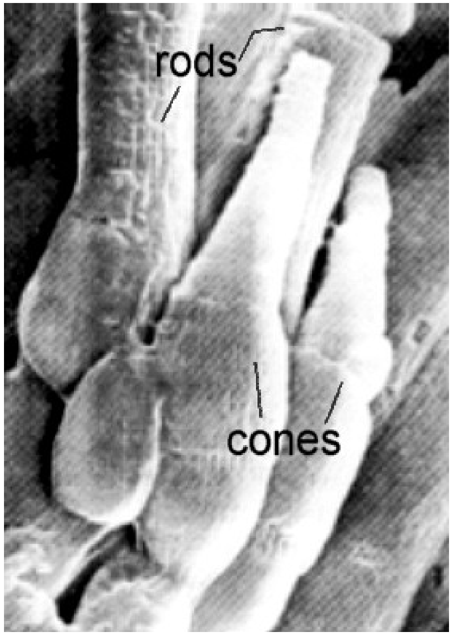
A luminous flux is measured in ***lumen*** units (lm).
A lumen unit is defined as the amount of light that falls on a unit area at unit distance from a light source of one candela.

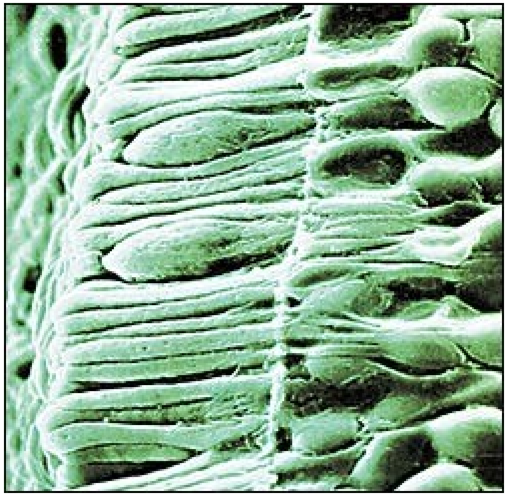
The ***lux*** (lx) represents the luminance and defined as one lumen per square meter.

Radiance	$\frac{W}{sr.m^2}$	Watt per steradian per meter square
Luminance	$\frac{cd}{m^2}$	Candela per meter square
Irradiance	$\frac{W}{m}$	Watt per meter
Luminance	lx	Lux

Cone responses in human eyes





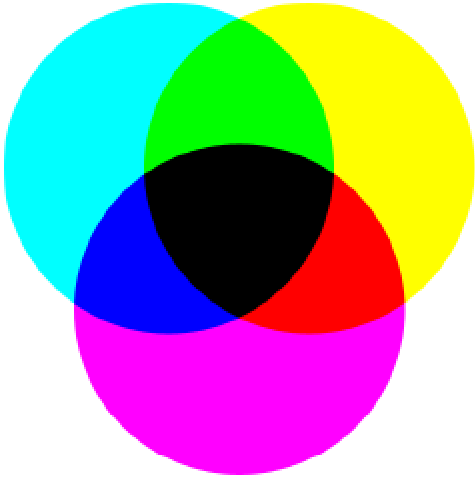


Cones (color) and rods (B&W) in human eyes

Additive and subtractive color modes

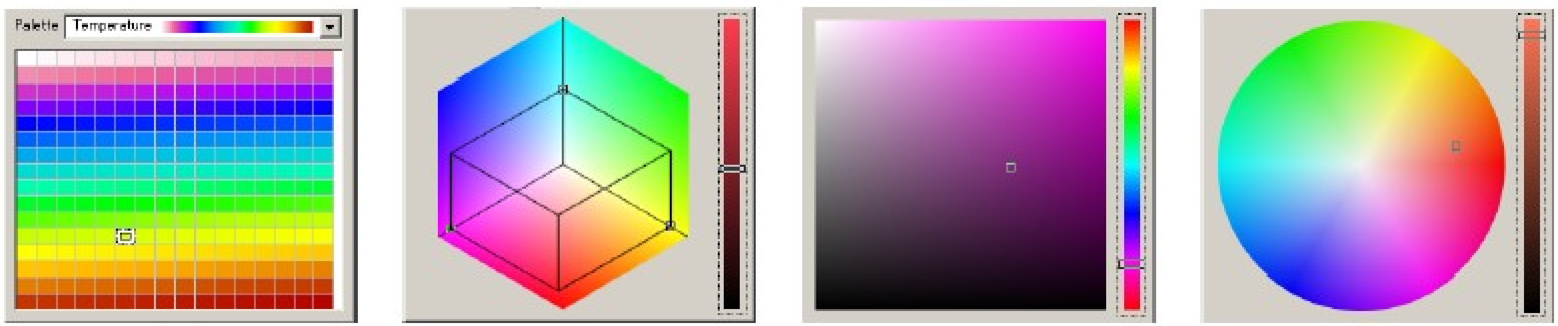
Display/printing industries

(a) 

(b) 

Color checking in printing industry

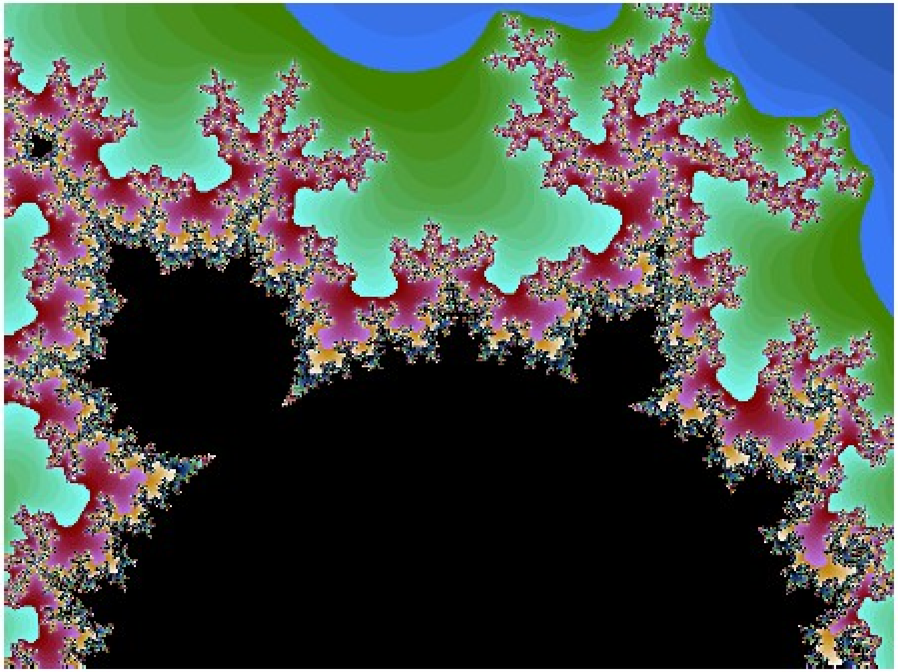
Choosing colors: Colors pick-up

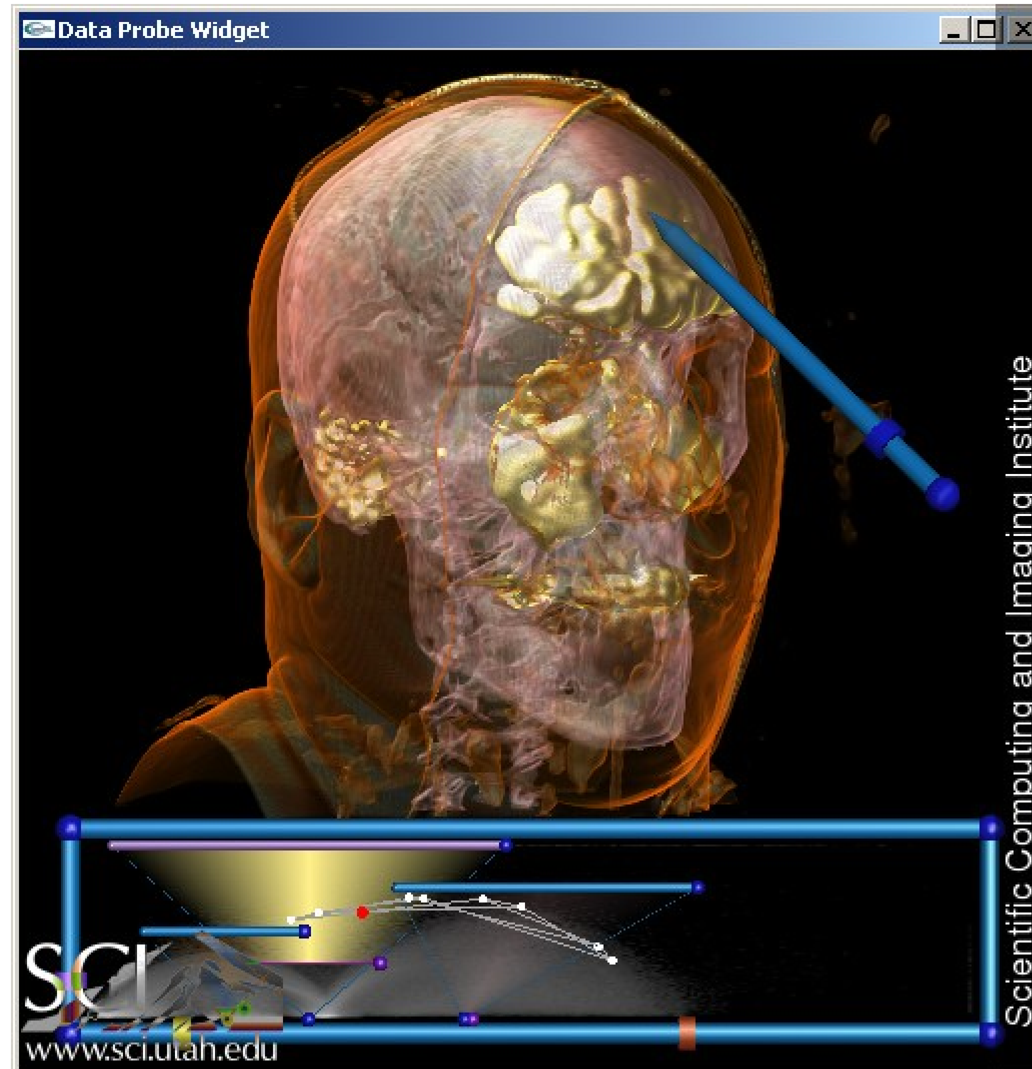


(a) (b) (c) (d)

COLOR PLATE VIII *Examples of common user interfaces for picking colors: (a) palette picker, (b) RGB picker, (c) 2D picker, and (d) HSV picker.*

Pseudo-coloring





Volume rendering and transfer function

Color spaces

converts the spectrum absorption of light *L* to XYZ colors

$$X(L) = k \int_{\lambda_{\min}}^{\lambda_{\max}} x(\lambda) L(\lambda) d\lambda,$$

$$Y(L) = k \int_{\lambda_{\min}}^{\lambda_{\max}} y(\lambda) L(\lambda) d\lambda,$$

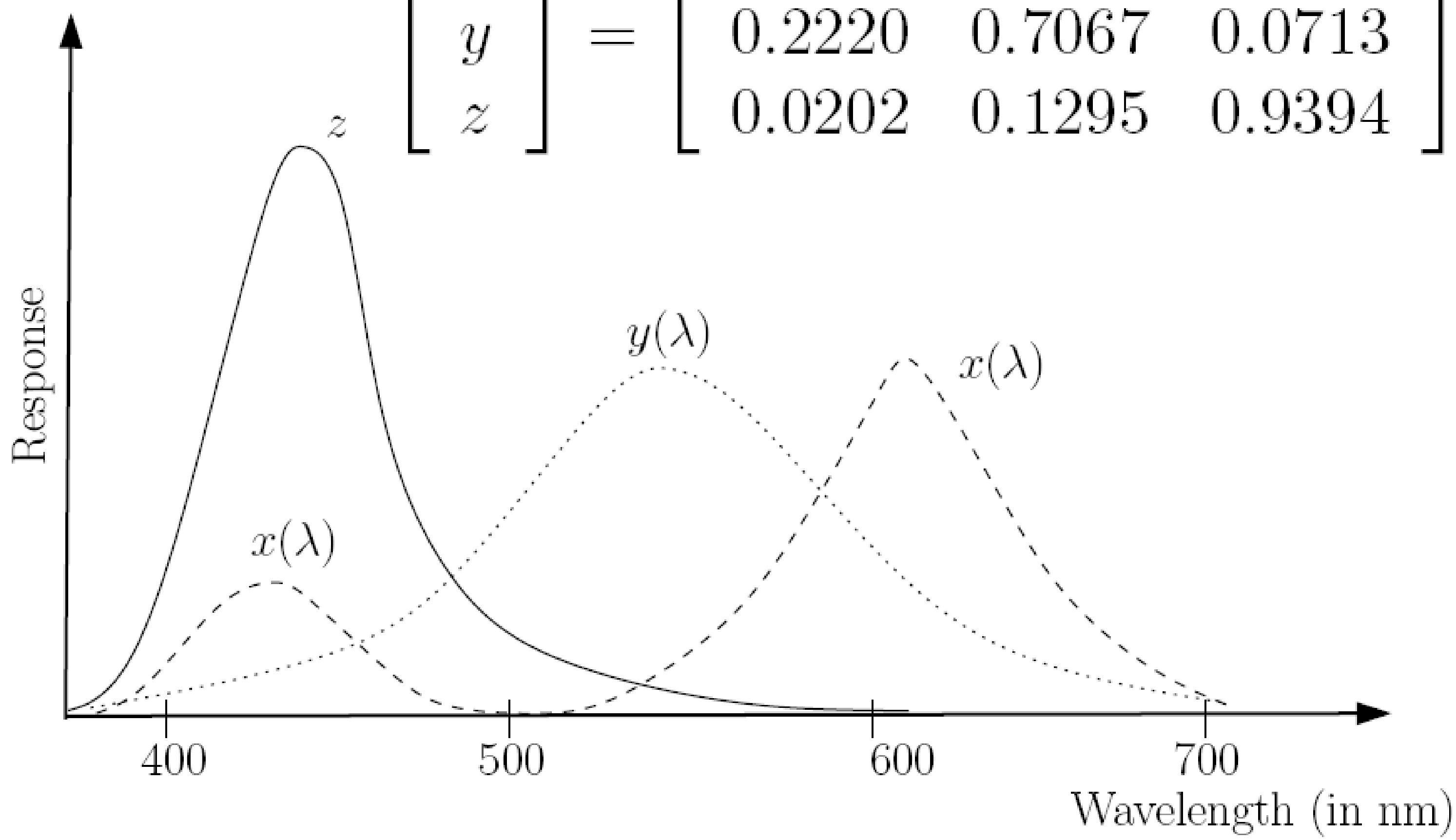
$$Z(L) = k \int_{\lambda_{\min}}^{\lambda_{\max}} z(\lambda) L(\lambda) d\lambda,$$

Normalizing coefficient

$$k = \frac{100}{\int_{\lambda_{\min}}^{\lambda_{\max}} y(\lambda) W(\lambda) d\lambda}$$

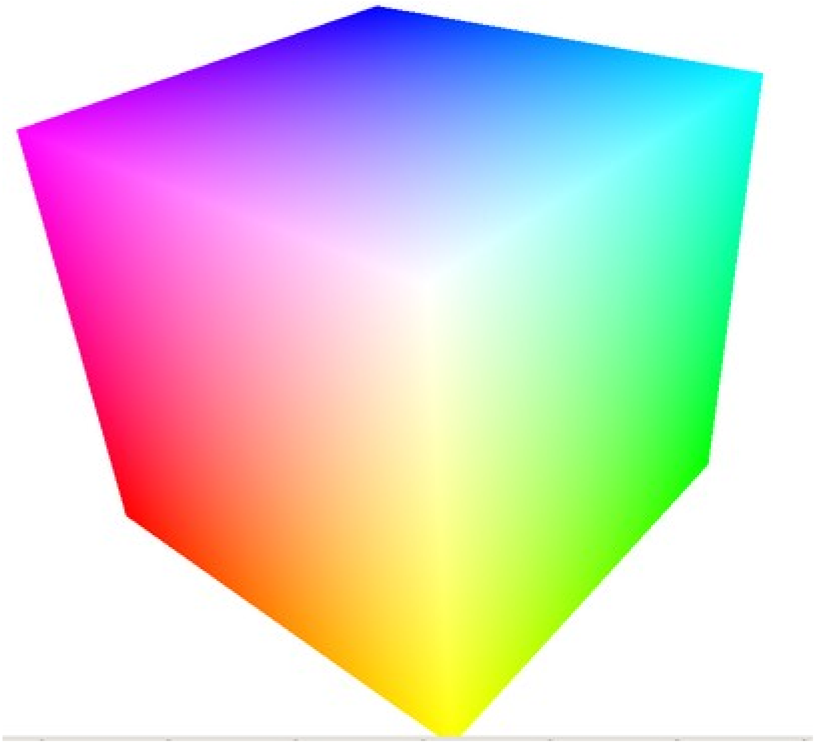
$$\begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} 3.0527 & -1.3928 & -0.4759 \\ -0.9689 & 1.8756 & 0.0417 \\ 0.0585 & -0.2286 & 1.0690 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0.4305 & 0.3415 & 0.1784 \\ 0.2220 & 0.7067 & 0.0713 \\ 0.0202 & 0.1295 & 0.9394 \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix}$$



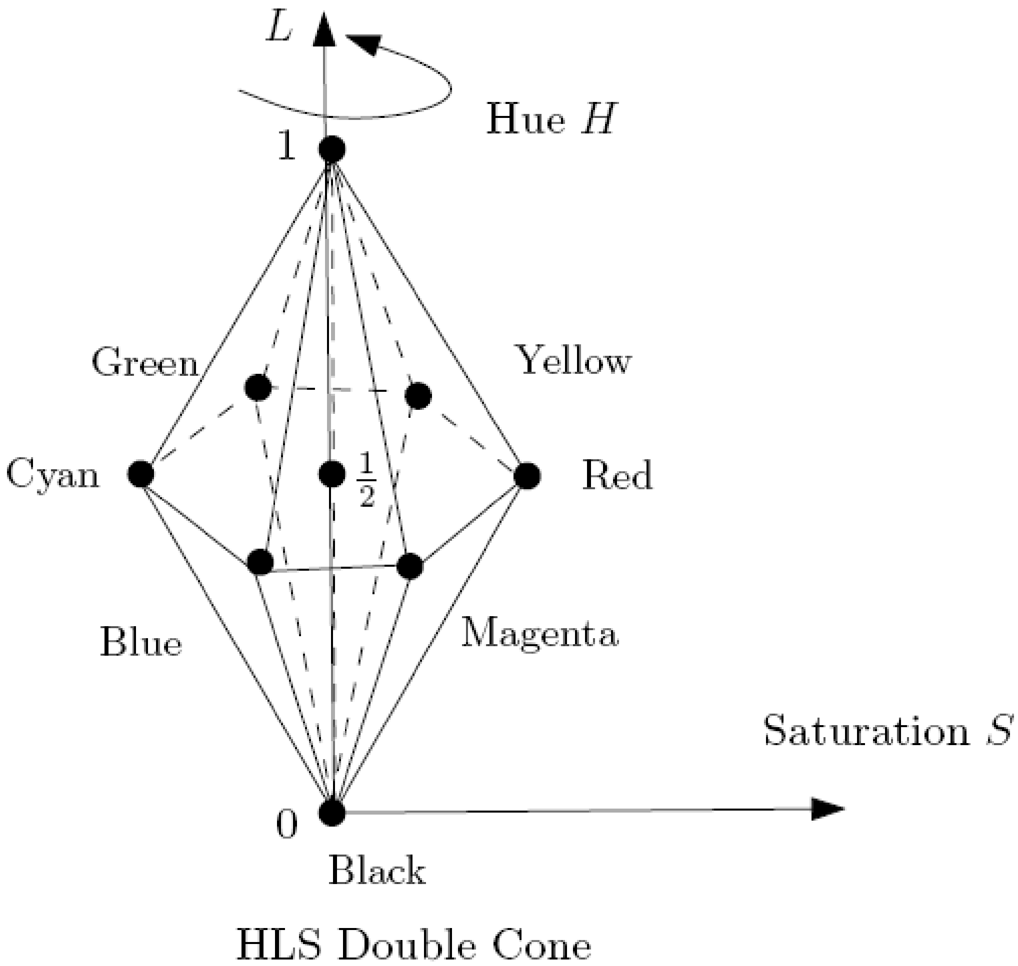
The basic response functions for CIE XYZ hypothetical colors. Observe the artificial response curve x(·) of color X.

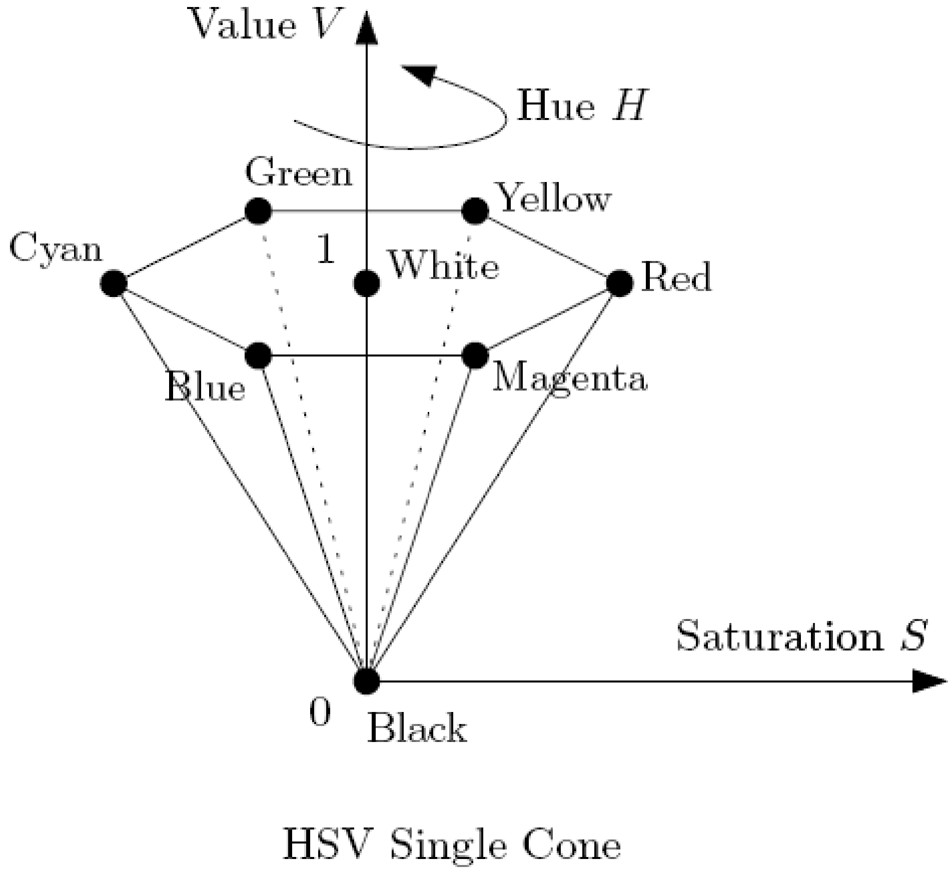
Color cube



Vertex position	
$(0,0,0)$	Black
$(1,1,1)$	White
$(1,0,0)$	Red
$(0,1,0)$	Green
$(0,0,1)$	Blue
$(1,1,0)$	Yellow
$(0,1,1)$	Cyan
$(1,0,1)$	Magenta

HLS/HSV color spaces





(hue, lightness, saturation

(hue, saturation, value)

RGBtoHSV(r, g, b)

$m \leftarrow \min\{r,g,b\}; V \leftarrow \max\{r,g,b\}$
if $m > 0$
  then $S \leftarrow (V-m)/V$
  else $S \leftarrow 0; H \leftarrow$ undefined
switch
  case $V = r$:
    $H = \frac{g-b}{V-m}$
  case $V = g$:
    $H = 2 + \frac{b-r}{V-m}$
  case $V = b$:
    $H = 4 + \frac{r-g}{V-m}$
$H \leftarrow 60H$
if $H < 0$
  then $H \leftarrow 360 + H$

RGB	*Name*	*HSV*
$\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$	Black	$\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$
$\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$	Red	$\begin{bmatrix} 0 & 1 & 1 \end{bmatrix}$
$\begin{bmatrix} 1 & 1 & 0 \end{bmatrix}$	Yellow	$\begin{bmatrix} 60 & 1 & 1 \end{bmatrix}$
$\begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$	Green	$\begin{bmatrix} 120 & 1 & 1 \end{bmatrix}$
$\begin{bmatrix} 0 & 1 & 1 \end{bmatrix}$	Cyan	$\begin{bmatrix} 180 & 1 & 1 \end{bmatrix}$
$\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$	Blue	$\begin{bmatrix} 240 & 1 & 1 \end{bmatrix}$
$\begin{bmatrix} 1 & 0 & 1 \end{bmatrix}$	Magenta	$\begin{bmatrix} 300 & 1 & 1 \end{bmatrix}$
$\begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$	White	$\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$

CMYK color space (for printing)

$$c' = 1 - r$$

$$m' = 1 - g$$

$$y' = 1 - b$$

$$k = \min\{c', m', y'\}$$

$$c = \frac{c' - k}{1 - k}, \qquad m = \frac{m' - k}{1 - k}, \qquad y = \frac{y' - k}{1 - k}$$

CIE L*a*b (where Euclidean distance make sense)

$$L = 25\frac{100Y}{Y_W}^{\frac{1}{3}} - 16,$$

CIE L*a*b

$$a = 500\left(\left(\frac{X}{X_W}\right)^{\frac{1}{3}} - \left(\frac{Y}{Y_W}\right)^{\frac{1}{3}}\right)$$

$$b = 200\left(\left(\frac{Y}{Y_W}\right)^{\frac{1}{3}} - \left(\frac{Z}{Z_0}\right)^{\frac{1}{3}}\right)$$

Perceptual difference between two colors:

$$\Delta C = \sqrt{\Delta L^2 + \Delta a^2 + \Delta b^2}$$

Many standards....

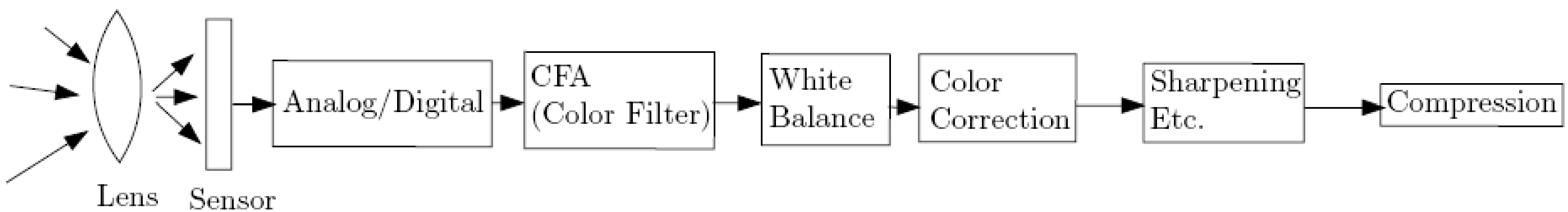
$$Y = 0.2125R + 0.7154G + 0.0721B$$

$$\begin{bmatrix} y \\ u \\ v \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.1 \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix}$$

$$\begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.140 \\ 1 & -0.396 & -0.581 \\ 1 & 2.029 & 0 \end{bmatrix} \begin{bmatrix} y \\ u \\ v \end{bmatrix}$$

$$\begin{bmatrix} y \\ i \\ q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix}$$

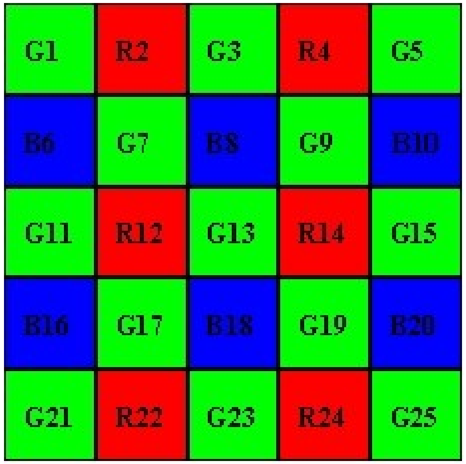
$$\begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} 1 & 0.956 & 0.621 \\ 1 & -0.272 & -0.647 \\ 1 & -1.105 & 1.702 \end{bmatrix} \begin{bmatrix} y \\ i \\ q \end{bmatrix}$$



Gamma color correction:

$$i_\gamma = 255 \left(\frac{i}{255} \right)^{\frac{1}{\gamma}} + \frac{1}{2}$$

Bayer tile



Half-toning and dithering

t=127

Random t

Dithering: Trade space for grey level perception

```
DITHERINGBINARIZATION(I, D, B)
    ◁ I is the input image ▷
    ◁ D is the dither cell ▷
    ◁ B is the binary image ▷
    for j ← 1 to h
        do for i ← 1 to w
                do
                    if I[i, j] > D[i mod k, j mod k]
                       then B[i, j] ← 1
                       else  B[i, j] ← 0
```

$$\begin{bmatrix} 1 & 33 & 9 & 41 & 3 & 35 & 11 & 43 \\ 49 & 17 & 57 & 25 & 51 & 19 & 59 & 27 \\ 13 & 45 & 5 & 37 & 15 & 47 & 7 & 39 \\ 61 & 29 & 53 & 21 & 63 & 31 & 55 & 23 \\ 4 & 36 & 12 & 44 & 2 & 34 & 10 & 42 \\ 52 & 20 & 60 & 28 & 50 & 18 & 58 & 26 \\ 16 & 48 & 8 & 40 & 14 & 46 & 6 & 38 \\ 64 & 32 & 56 & 24 & 62 & 30 & 54 & 22 \end{bmatrix}$$

$$\begin{array}{|c|c|c|c|c|c|} \hline \dots & \dots & \dots & \dots & \dots & \dots \\ \hline \dots & 128 & 128 & 128 & 128 & \dots \\ \hline \dots & 128 & 128 & 128 & 128 & \dots \\ \hline \dots & 128 & 128 & 128 & 128 & \dots \\ \hline \dots & 128 & 128 & 128 & 128 & \dots \\ \hline \dots & \dots & \dots & \dots & \dots & \dots \\ \hline \end{array} > \begin{array}{|c|c|c|c|c|c|} \hline \dots & \dots & \dots & \dots & \dots & \dots \\ \hline \dots & 48 & 144 & 48 & 144 & \dots \\ \hline \dots & 192 & 96 & 192 & 96 & \dots \\ \hline \dots & 48 & 144 & 48 & 144 & \dots \\ \hline \dots & 192 & 96 & 192 & 96 & \dots \\ \hline \dots & \dots & \dots & \dots & \dots & \dots \\ \hline \end{array} = \begin{array}{|c|c|c|c|c|c|} \hline \dots & \dots & \dots & \dots & \dots & \dots \\ \hline \dots & 1 & 0 & 1 & 0 & \dots \\ \hline \dots & 0 & 1 & 0 & 1 & \dots \\ \hline \dots & 1 & 0 & 1 & 0 & \dots \\ \hline \dots & 0 & 1 & 0 & 1 & \dots \\ \hline \dots & \dots & \dots & \dots & \dots & \dots \\ \hline \end{array}$$

$$\mathbf{D} = \begin{bmatrix} 48 & 144 \\ 192 & 96 \end{bmatrix} \qquad \begin{bmatrix} 178 & 51 & 153 \\ 229 & 25 & 76 \\ 127 & 102 & 204 \end{bmatrix}$$

Error-diffusion process: Floyd-Steinberg

$$\mathbf{E} = \frac{1}{16}\begin{bmatrix} \square & \square & 7 \\ 3 & 5 & 1 \end{bmatrix}$$

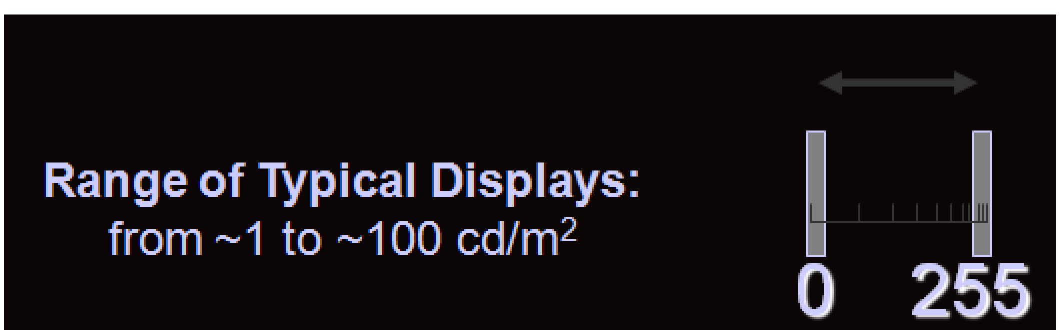
$$\mathbf{E} = \frac{1}{48}\begin{bmatrix} \square & \square & \square & 7 & 5 \\ 3 & 5 & 7 & 5 & 3 \\ 1 & 3 & 5 & 3 & 1 \end{bmatrix}$$

$\textsc{FloydSteinbergBinarization}(\mathbf{I})$

- **for** $i \leftarrow 1$ **to** h
  - **do for** $j \leftarrow 1$ **to** w
    - **do if** $\mathbf{I}[i,j] < 0.5$
      - **then** $\mathbf{B}[i,j] = 0$
      - **else** $\mathbf{B}[i,j] = 1$
    - error $= \mathbf{I}[i,j] - \mathbf{B}[i,j]$
    - $\triangleleft$ Error diffusion to neighbors. See Eq. 4.105 $\triangleright$
    - $\mathbf{I}[i,j+1] \leftarrow \mathbf{I}[i,j+1] + \frac{7}{16}\text{error}$
    - $\mathbf{I}[i+1,j-1] \leftarrow \mathbf{I}[i+1,j-1] + \frac{3}{16}\text{error}$
    - $\mathbf{I}[i+1,j] \leftarrow \mathbf{I}[i+1,j] + \frac{5}{16}\text{error}$
    - $\mathbf{I}[i+1,j+1] \leftarrow \mathbf{I}[i+1,j+1] + \frac{1}{16}\text{error}$

Low dynamic range images (LDRs) versus High dynamic range images (HDRs)

- The range of luminances is more than 10^14 candela/m2



Low dynamic range image
High dynamic range image
Tone mapping

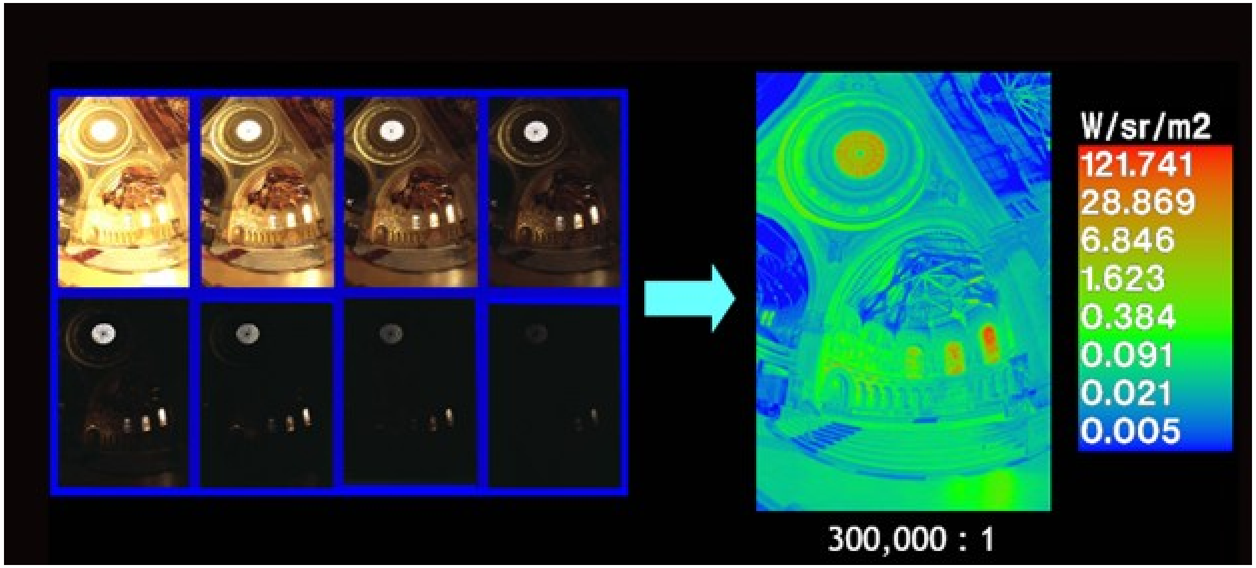
(a) $\Delta t = \frac{1}{40}s$, F3.5 Stop

(b) $\Delta t = \frac{1}{400}s$, F5.6 Stop

(c) $\Delta t = \frac{1}{500}s$, F8.0 Stop

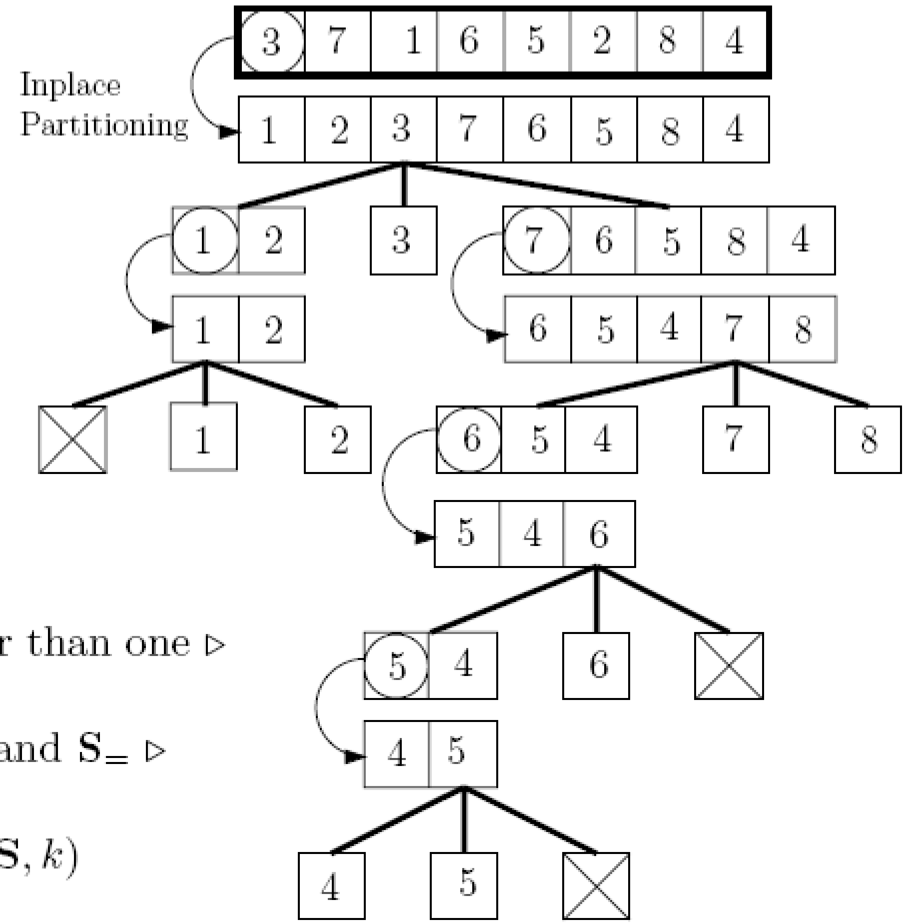
(d) Tone mapped

Tone mapping



Quicksort:

O(n log n) expected
O(n^2) worst-case
O(n) best case



QuickSort(S)

- ◁ We only need to sort arrays of size strictly greater than one ▷
- **if** $|S| > 1$
  - **then** ◁ Partition in place the array into $S_<$, $S_>$ and $S_=$ ▷
    - Choose a random pivot index k
    - $S_<, S_=, S_> \leftarrow$ PartitionArrayInPlace(S, k)
    - ◁ Recursive calls ▷
    - QuickSort($S_<$)
    - QuickSort($S_>$)

RandomPermutation(S)

- **for** $i \in 1$ **to** n
  - **do**
    - $S[i] = i$
    - ◁ Draw a random number in $[\![1, i]\!]$ ▷
    - $j = \mathrm{RandomNumber}(1, i)$
    - swap($S[j], S[i]$)

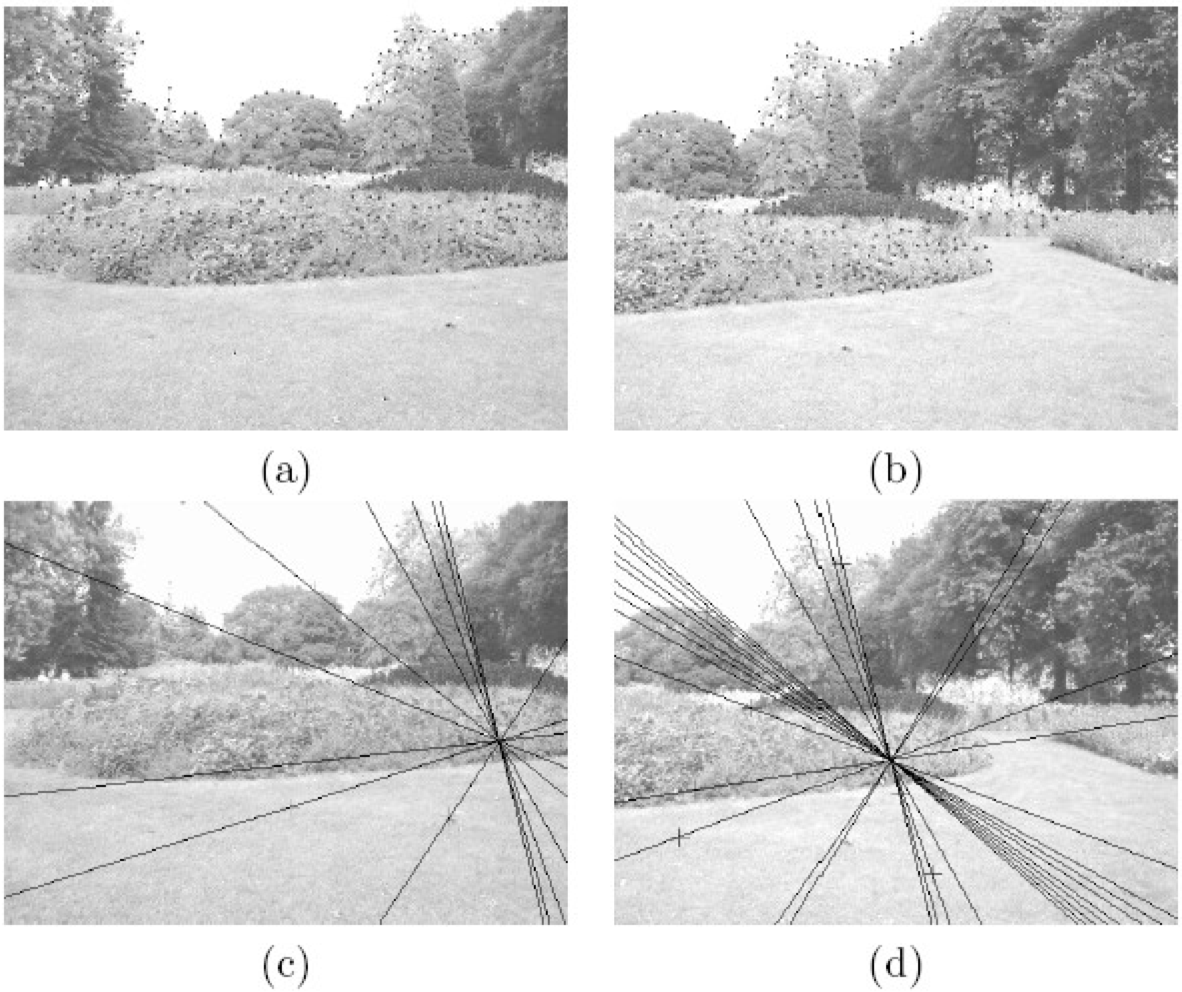
Randomization: A Powerful principle

TossingACoin()
repeat
    Face $\leftarrow$ Toss a black/white coin
until Face=White

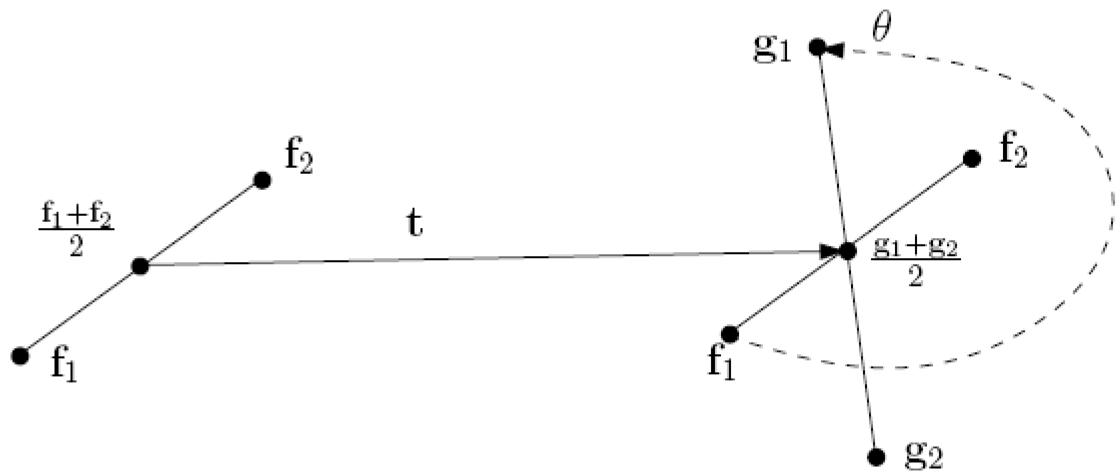
Analysis: $$\sum_{i=1}^{\infty} \frac{1}{2^i} i = 2.$$

SelectElement($\mathbf{S}, k$)
$\triangleleft$ Select the kth smallest element of an array $\mathbf{S}$ of n elements $\triangleright$
if $|\mathbf{S}| = 1$
    then return $\mathbf{S}[1]$
    else Choose a random pivot index $k \in [\![1, |\mathbf{S}|]\!]$
        $\triangleleft$ Partition inplace array $\mathbf{S}$ $\triangleright$
        $\mathbf{S}_<, \mathbf{S}_=, \mathbf{S}_> \leftarrow$ PartitionArray($\mathbf{S}, k$)
        if $k \leq |\mathbf{S}_<|$
            then return SelectElement($\mathbf{S}_<, k$)
            else **if** $k > |\mathbf{S}_<| + |\mathbf{S}_=|$
                then return SelectElement($\mathbf{S}_>, k - |\mathbf{S}_<| - |\mathbf{S}_=|$)
                else $\triangleleft$ The kth smallest element of $\mathbf{S}$ is inside $\mathbf{S}_=$ $\triangleright$
                    $\triangleleft$ $\mathbf{S}_=[1]$ is stored at $\mathbf{S}[k]$ (inplace partitioning) $\triangleright$
                    return $\mathbf{S}[k]$

RANSAC: Random Sample Consensus



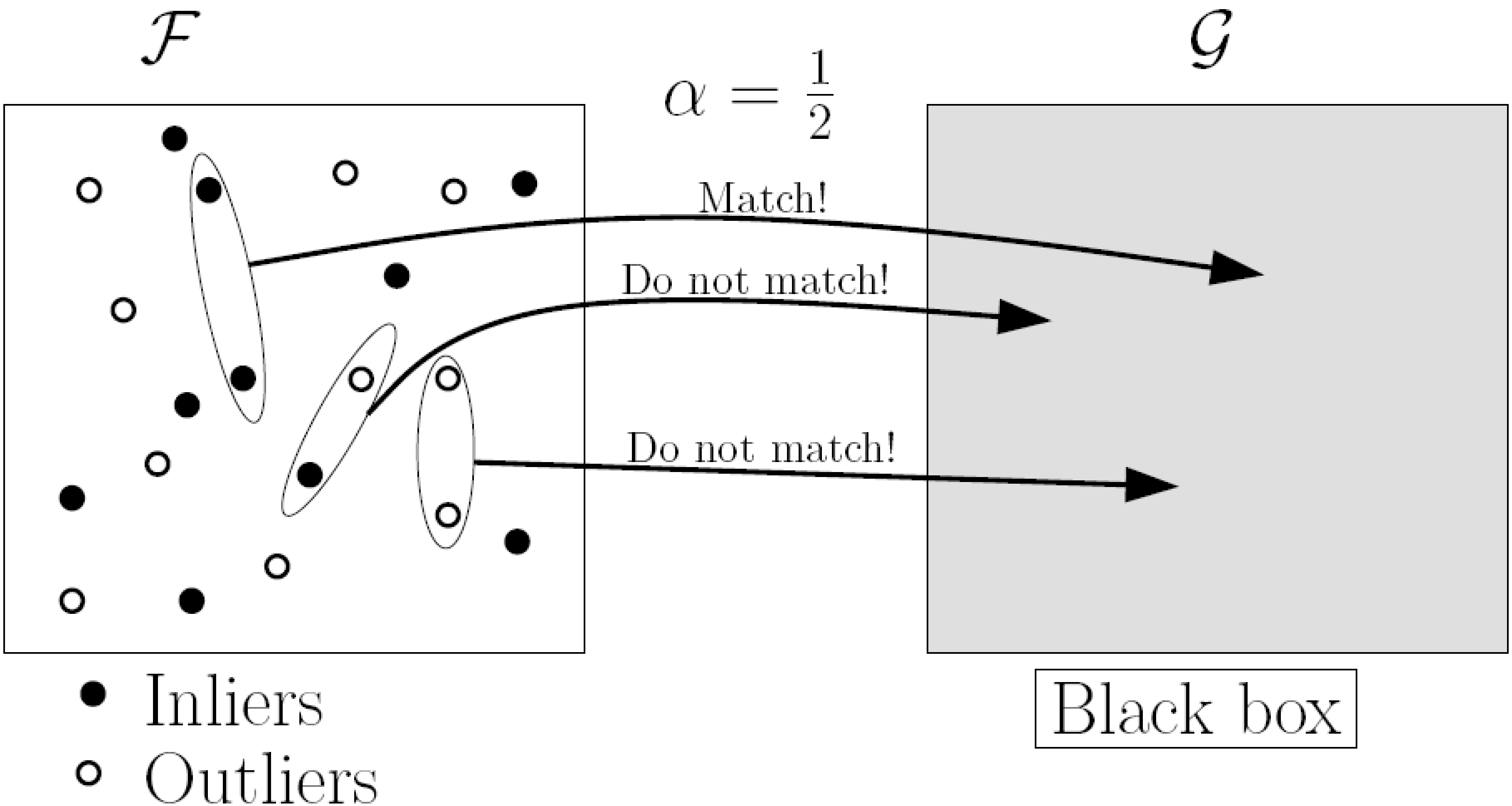
RANSAC on Harris-Stephens points.
Epipolar points intersect on their epipoles



$$\mathsf{M} = \mathsf{M}[f_1, f_2; g_1, g_2] = \mathsf{T}_{\frac{g_1+g_2}{2}} \mathsf{R}_\theta \mathsf{S}_s \mathsf{T}_{-\frac{f_1+f_2}{2}}$$

$$\mathsf{M}[f_1, f_2; g_1, g_2] = \begin{bmatrix} \mathbf{I} & \frac{g_1+g_2}{2} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\frac{f_1+f_2}{2} \\ \mathbf{0}^T & 1 \end{bmatrix}$$

Macthing points: Inliers/outliers



Probability of failure of picking two inliers $1 - \alpha^2$

RANSAC: Bounding failure probability

Probability of failure after one round $1-\alpha^2$

Probability of failure after **k independent rounds** $(1-\alpha^2)^k$

$$e^{k\log(1-\alpha^2)} \leq f.$$

$$k = \left\lceil \frac{\log f}{\log(1-\alpha^2)} \right\rceil = \lceil \log_{1-\alpha^2} f \rceil$$

$$k = \left\lceil \frac{\log f}{\log(1-\alpha^s)} \right\rceil = \lceil \log_{1-\alpha^s} f \rceil$$

TABLE 7.1 *Number of rounds required by RANSAC to ensure a failure probability below 1%.*

Number of pairs / Application example		Outliers ratio $(1-\alpha)$		
		10%	30%	50%
2	Similitude	3	7	17
3	Affine	4	11	35
4	Homography	5	17	72
6	Trifocal tensor	7	37	293
7	Fundamental matrix	8	54	588

Estimating homographies with RANSAC

HomographyRANSAC($\mathcal{P}_1, \mathcal{P}_2, n, f, \alpha$)
$\triangleleft$ n: number of points of $\mathcal{P}_1$ and $\mathcal{P}_2$ $\triangleright$
$\triangleleft$ f: probability of failure $\triangleright$
$\triangleleft$ α: a priori inlier ratio $\triangleright$
$k = \left\lceil \frac{\log f}{\log(1-\alpha^4)} \right\rceil$
$\triangleleft$ c_m: maximum consensus set found by RANSAC $\triangleright$
$c_m = 0$
for $i \leftarrow 1$ **to** k
  do Draw a random sample $\mathcal{S}_1$ of 4 elements from $\mathcal{P}_1$
    Check with all other 4-elements of $\mathcal{S}_2$.
    For each correspondence sets, calculate the free parameters.
    Compute the consensus set size c
    if $c > c_m$
      then $c_m = c$
        Save current transformation and largest consensus set
$\triangleleft$ Final stage: compute the homography given correspondence pairs $\triangleright$
Estimate the homography with the largest found consensus sample (inliers)

Adaptive RANSAC: Guessing the ratio of inliers

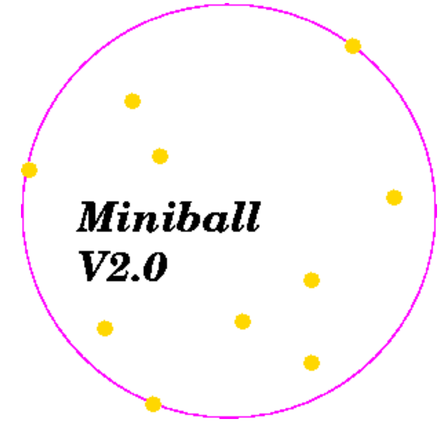
$\textsc{AdaptiveRANSAC}(n, f)$

$\triangleleft$ n: data set size $\triangleright$
$\triangleleft$ s: number of correspondences required (free parameters) $\triangleright$
$\triangleleft$ f: probability of failure $\triangleright$
$\triangleleft$ Initialize k to a very large number $\triangleright$
$k = \infty$
$\triangleleft$ d: current number of independent random draws $\triangleright$
$d = 0$
$\triangleleft$ c_m: maximum consensus set found so far by RANSAC $\triangleright$
$c_m = 0$
while $k > d$
    do Draw a random sample
        Compute the consensus set size c
        if $c > c_m$
            then $\triangleleft$ Update the proportion of inliers $\triangleright$
                $c_m = c$
                $\alpha = \frac{c}{n}$
                $\triangleleft$ Lower the number of rounds $\triangleright$
                $k = \left\lceil \frac{\log f}{\log(1-\alpha^s)} \right\rceil$
                $d = d + 1$

Demo of autostich:

Bounding sphere hierarchy

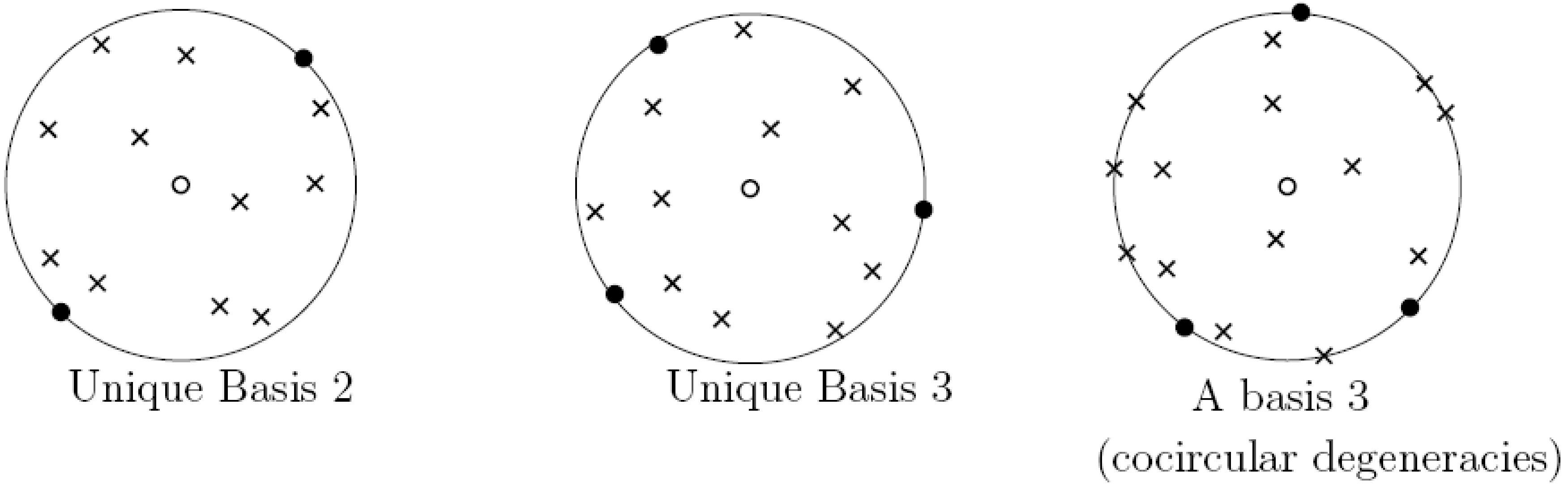
MINIBALL: Smallest enclosing ball UNIQUE



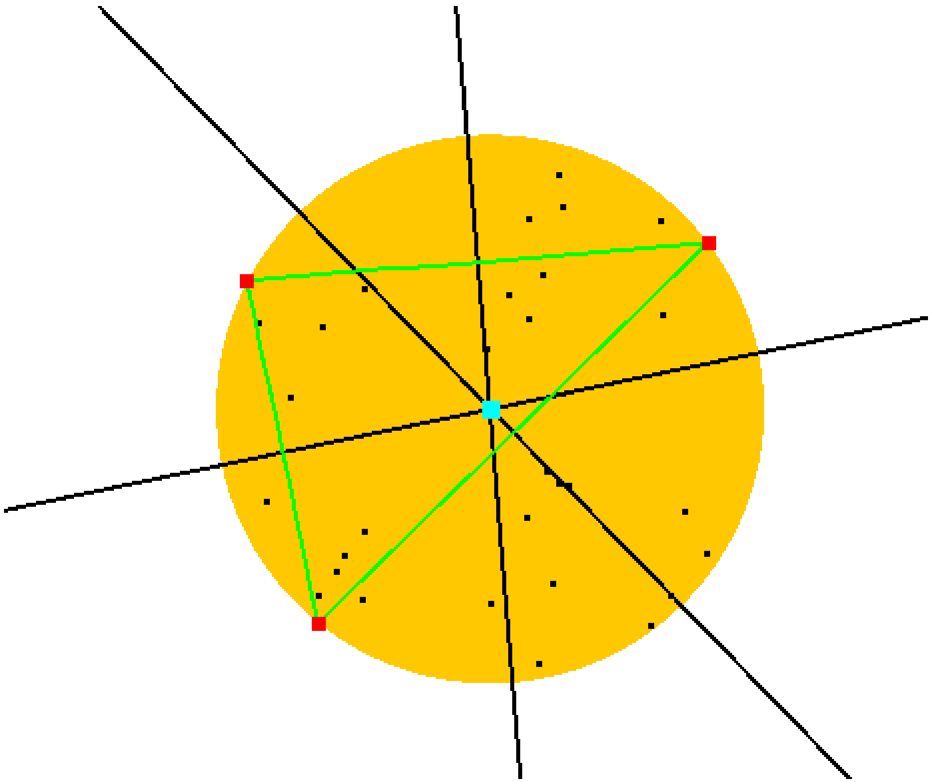
$\textsc{MiniBall}(\mathcal{P} = \{\mathbf{p}_1, ..., \mathbf{p}_n\}, \mathcal{B})$

$\triangleleft$ Initially the basis $\mathcal{B}$ is empty $\triangleright$
$\triangleleft$ Output: $\mathcal{B}$ contains a basis solution $\triangleright$
$\triangleleft$ That is, two or three points defining the minimum enclosing ball $\triangleright$
$\triangleleft$ Function $\textsc{MiniBall}$ returns the smallest enclosing ball B^* $\triangleright$
if $|\mathcal{B}| = 3$
  then return $B = \textsc{SolveBasis}(\mathcal{B})$
  else
    if $|\mathcal{P} \cup \mathcal{B}| \leq 3$
      then return $B = \textsc{SolveBasis}(\mathcal{P} \cup \mathcal{B})$
      else
        Select at random point $\mathbf{p} \in \mathcal{P}$
        $B = \textsc{MiniBall}(\mathcal{P}\backslash\{\mathbf{p}\}, \mathcal{B})$
        if $\mathbf{p} \notin B$
          then $\triangleleft$ Then $\mathbf{p}$ belongs to the boundary of $B^*(\mathcal{P})$ $\triangleright$
            return $B = \textsc{MiniBall}(\mathcal{P}\backslash\{\mathbf{p}\}, \mathcal{B} \cup \{\mathbf{p}\})$

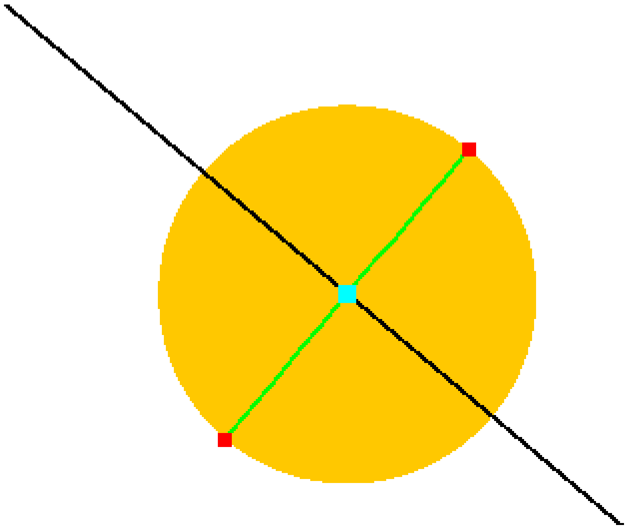
http://www.inf.ethz.ch/personal/gaertner/miniball.html



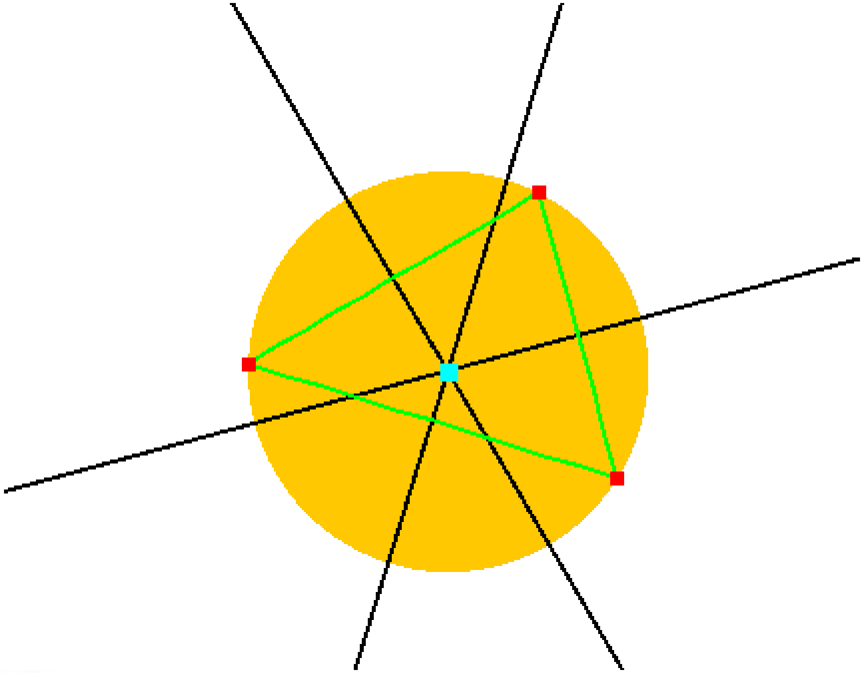
LP-type problem in optimizati



MINIBALL: Solving for basis of 2 and 3 points



Intersection of bisector/segment
linking the two points:
Circumcenter=half-point



Intersection of bisectors
(use cross-product on homogeneous
coordinates)