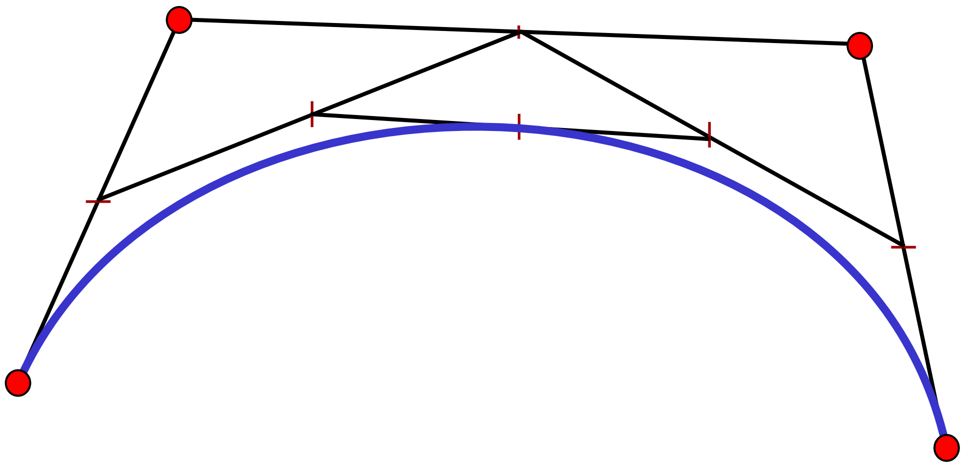
Introduction to Visualization and Computer Graphics
DH2320

Prof. Dr. Tino Weinkauf

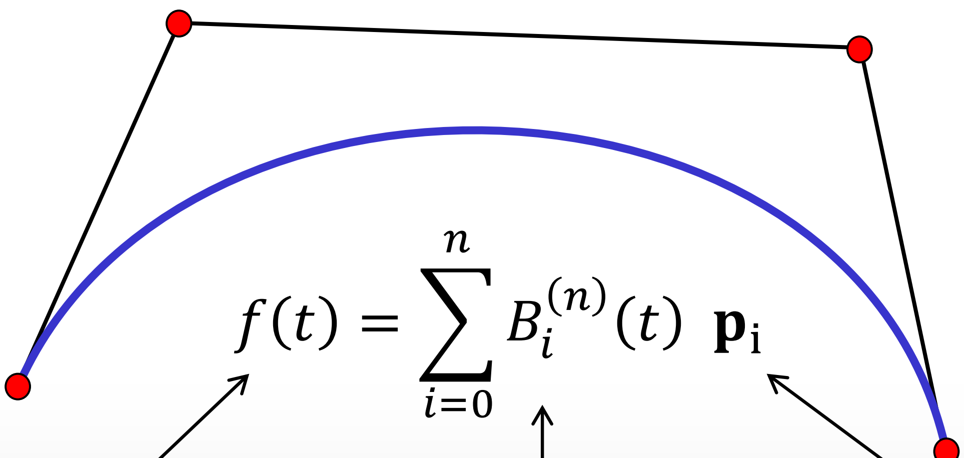
Geometric Modeling II

Spline Surfaces
Subdivision

Bernstein Basis



de Casteljau algorithm

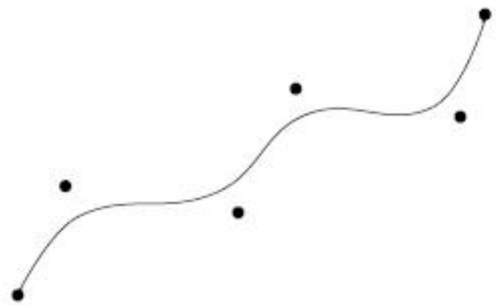


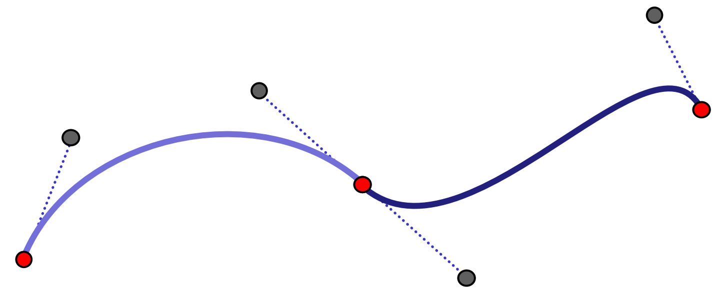
Bernstein form

$$f(t) = \sum_{i=0}^{n} B_i^{(n)}(t)\ \mathbf{p}_\mathrm{i}$$

curve | basis function | control point

Towards Bezier Splines





Approximation → Interpolation

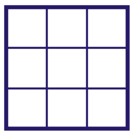
Spline Surfaces

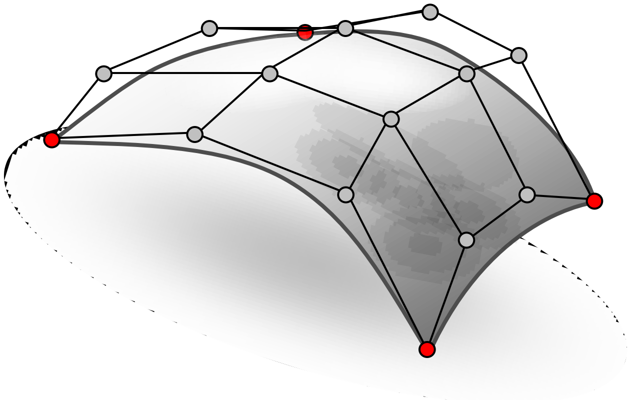
~~exam~~

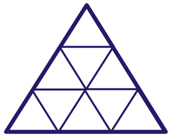
Spline Surfaces

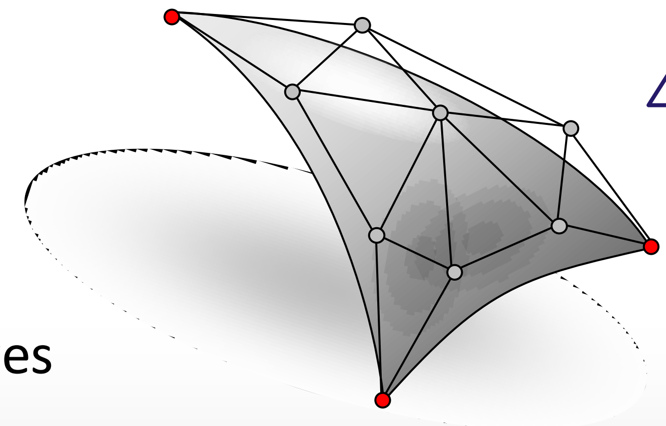
Two different approaches

- Tensor product surfaces
  - Simple construction
  - Everything carries over from curve case
  - Quad patches
  - Degree anisotropy
- Total degree surfaces
  - Not as straightforward
  - Isotropic degree
  - Triangle patches
  - “Natural” generalization of curves







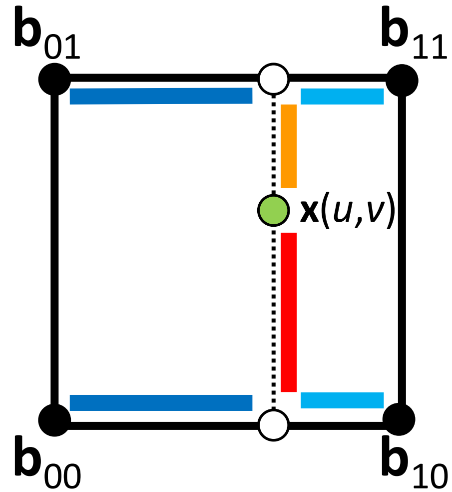


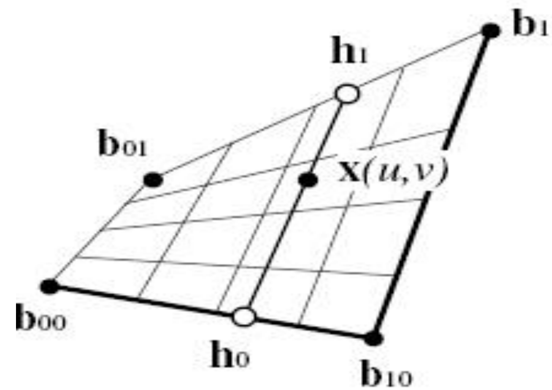
Tensor Product Surfaces

Tensor Product Bezier Surfaces

Bezier curves:
repeated linear interpolation

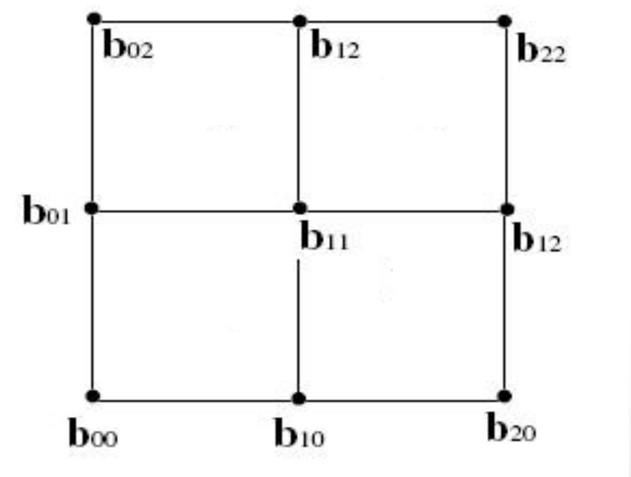
now a different setup:
4 points $\mathbf{b}_{00}$, $\mathbf{b}_{10}$, $\mathbf{b}_{11}$, $\mathbf{b}_{01}$
parameter area $[0,1] \times [0,1]$





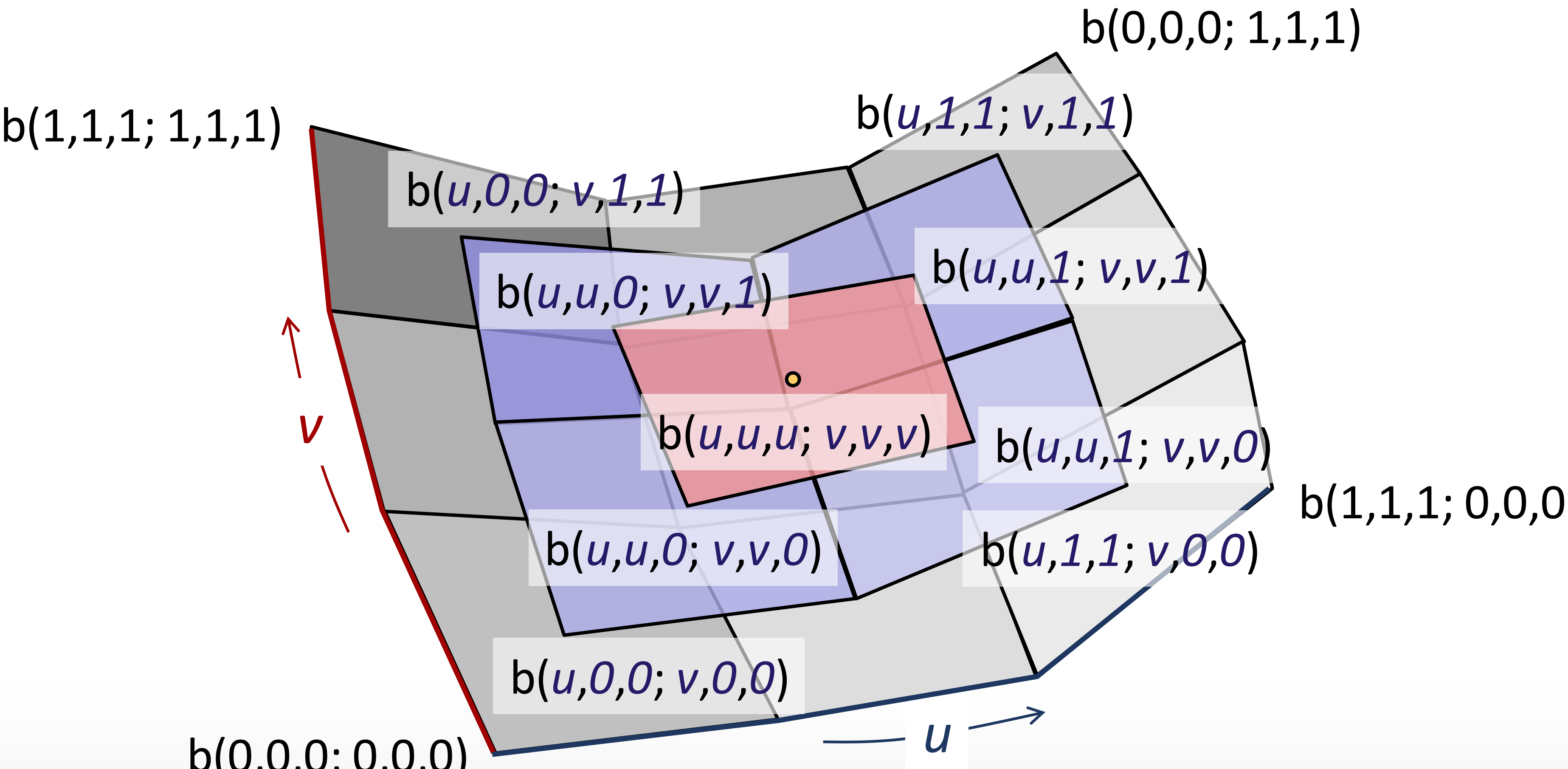
bilinear interpolation:
repeated linear interpolation

repeated bilinear interpolation:
gives us tensor product Bezier surfaces
(example shows quadratic Bezier surface)



De Casteljau Algorithm

De Casteljau algorithm for tensor product surfaces:



Tensor Product Surfaces

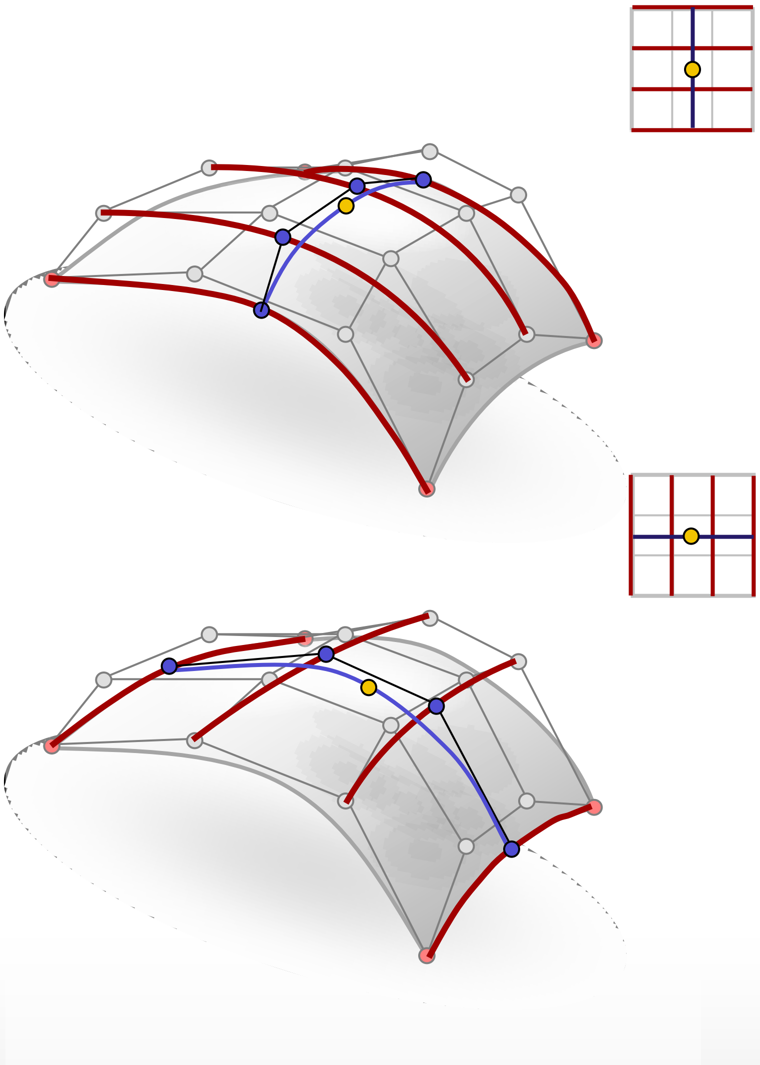
Tensor Product Surfaces:

$$\mathbf{f}(u,v) = \sum_{i=1}^{n}\sum_{j=1}^{n} b_i(u) b_j(v) \mathbf{p}_{i,j}$$

$$= \sum_{i=1}^{n} b_i(u) \sum_{j=1}^{n} b_j(v) \mathbf{p}_{i,j}$$

$$= \sum_{j=1}^{n} b_j(u) \sum_{i=1}^{n} b_i(v) \mathbf{p}_{i,j}$$

- "Curves of Curves"
- Order does not matter



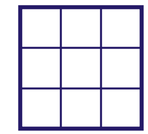
Tensor Product Surfaces

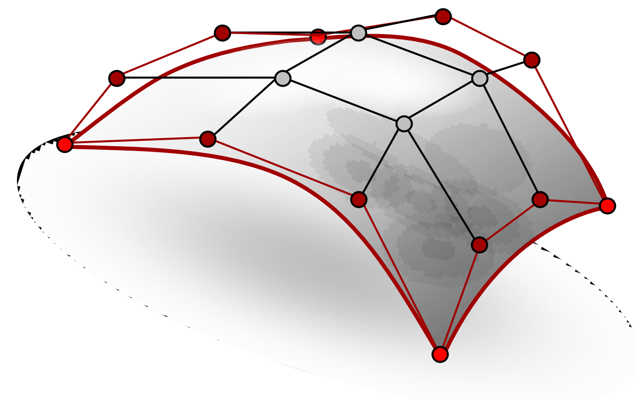
Bezier Patches

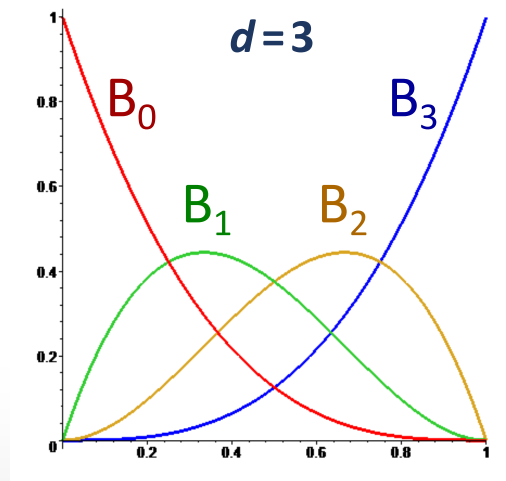
Bezier Patches

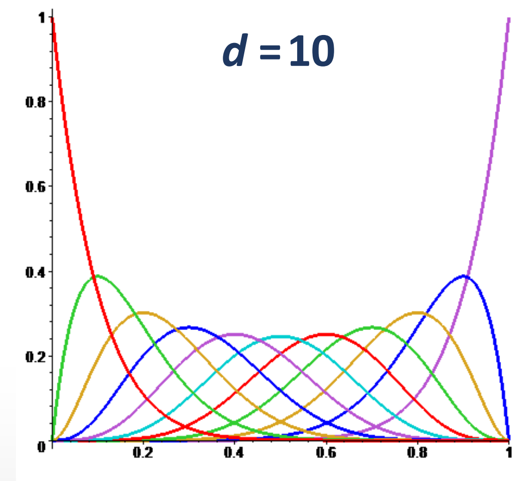
Bezier Patches:

- Remember endpoint interpolation:
  - Boundary curves are Bezier curves of the boundary control points









Continuity Conditions

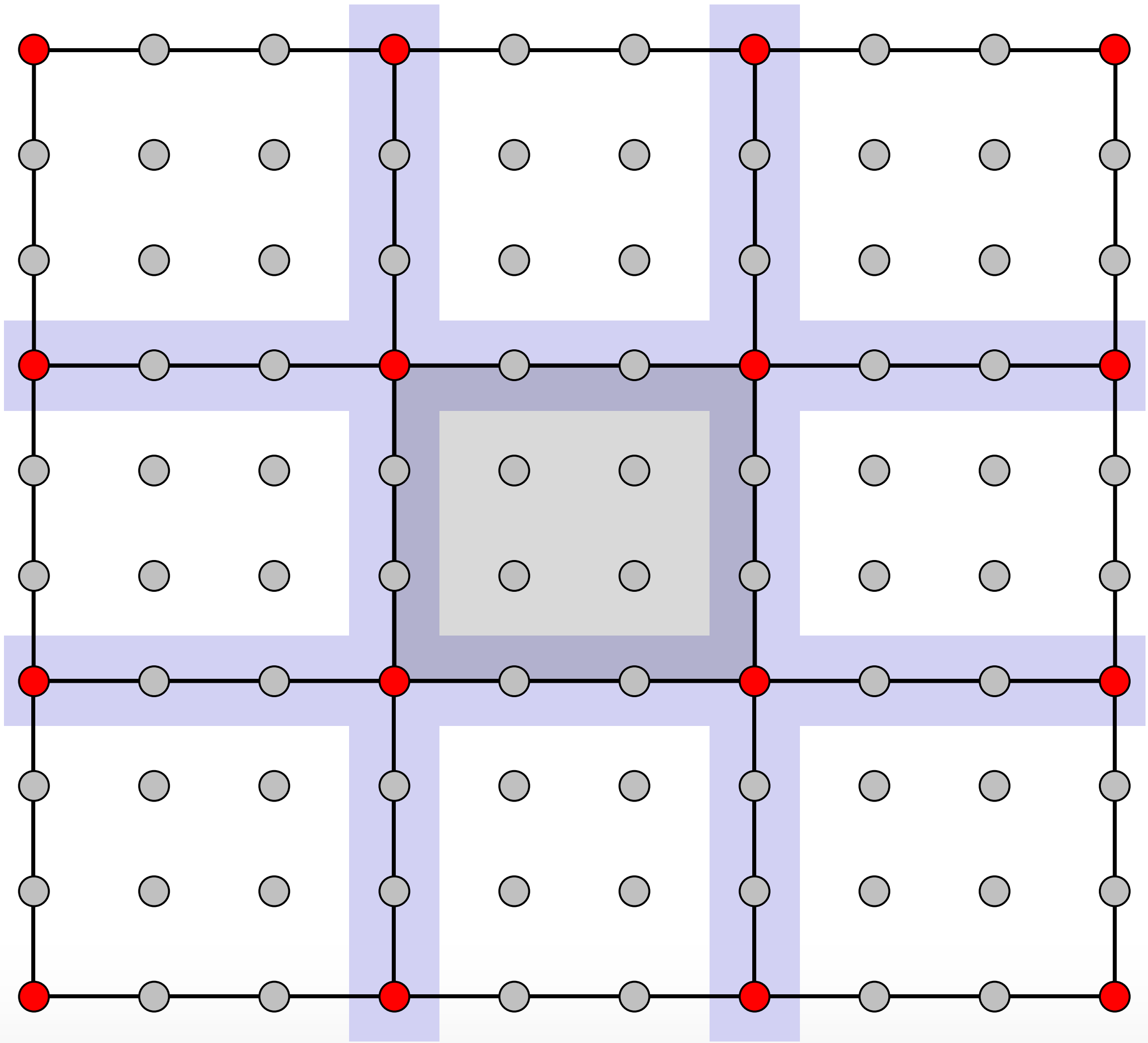
For C^0 continuity:

- Boundary control points must match

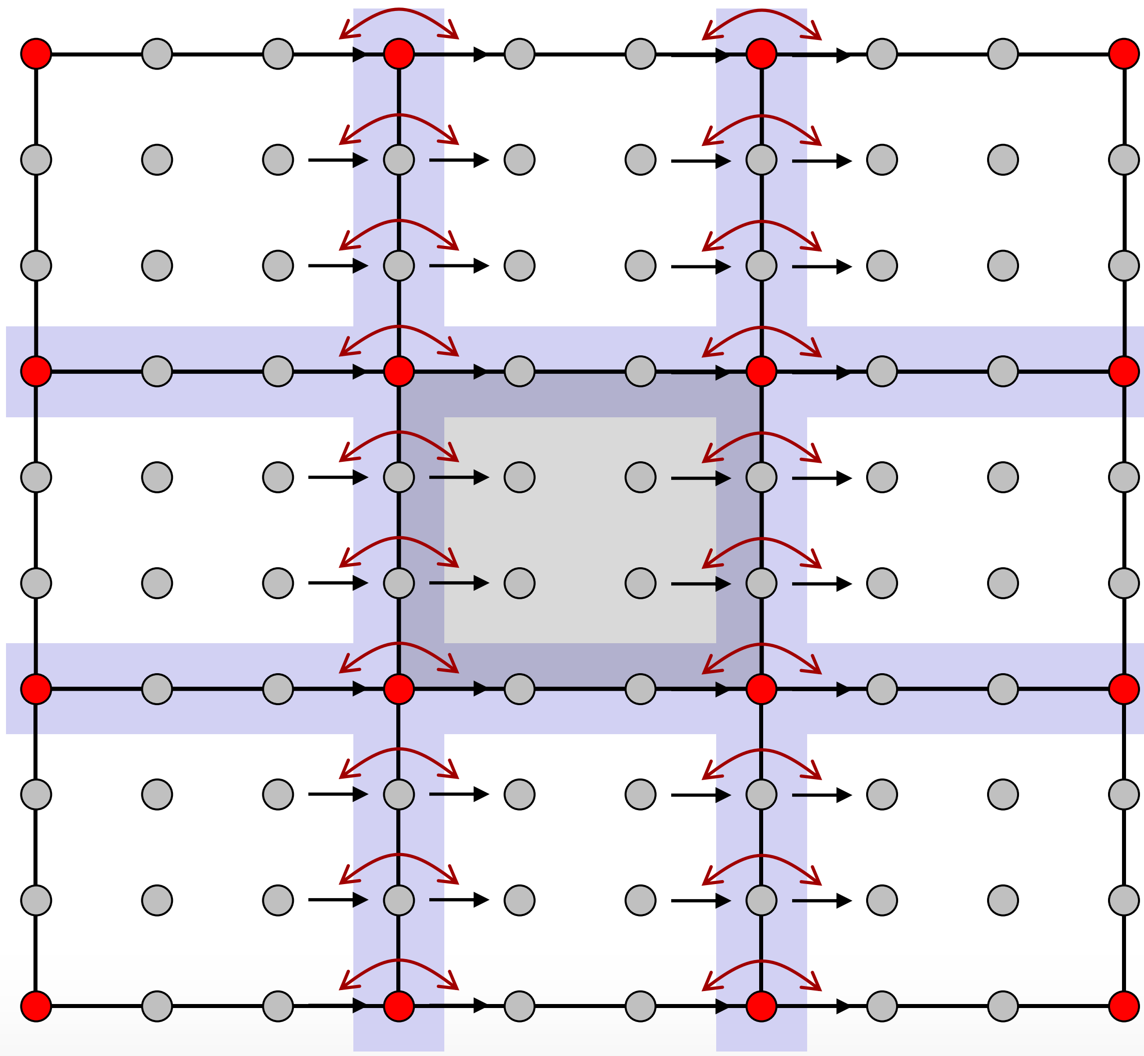
For C^1 continuity:

- Difference vectors must match at the boundary

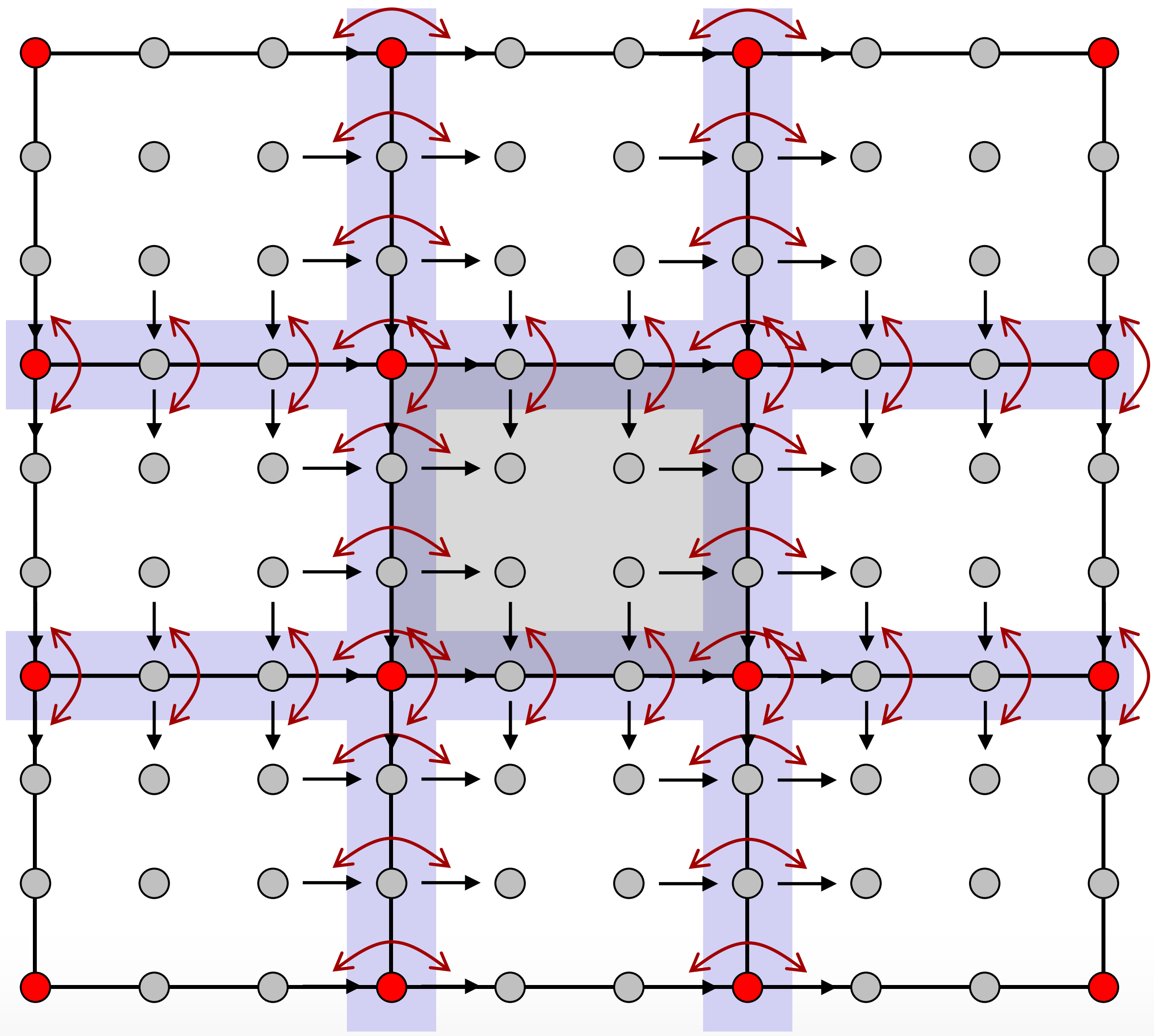
C^0 Continuity



C^1 Continuity



C^1 Continuity

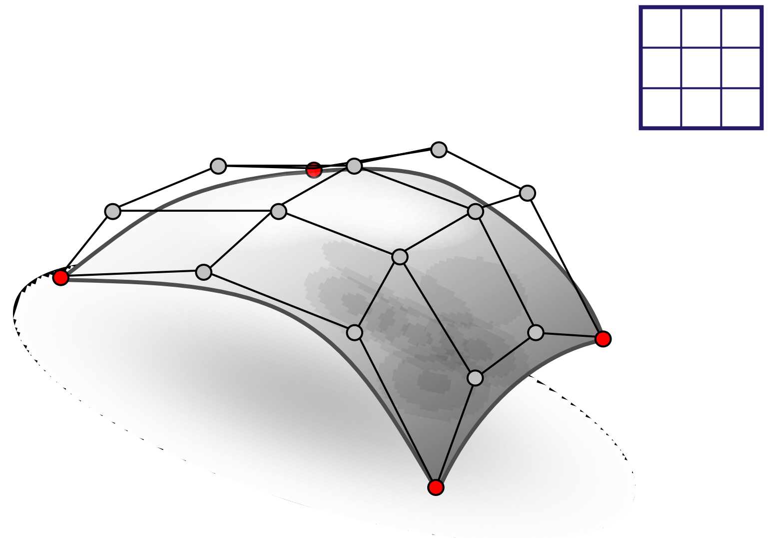


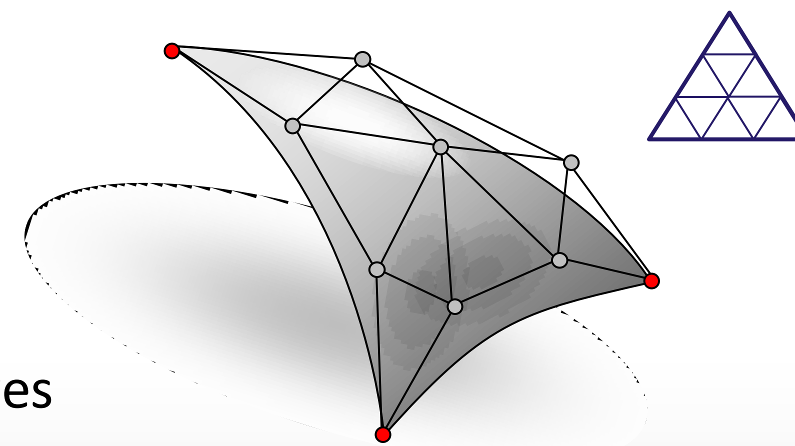
Total Degree Surfaces

Spline Surfaces

Two different approaches

- Tensor product surfaces
  - Simple construction
  - Everything carries over from curve case
  - Quad patches
  - Degree anisotropy
- Total degree surfaces
  - Not as straightforward
  - Isotropic degree
  - Triangle patches
  - “Natural” generalization of curves

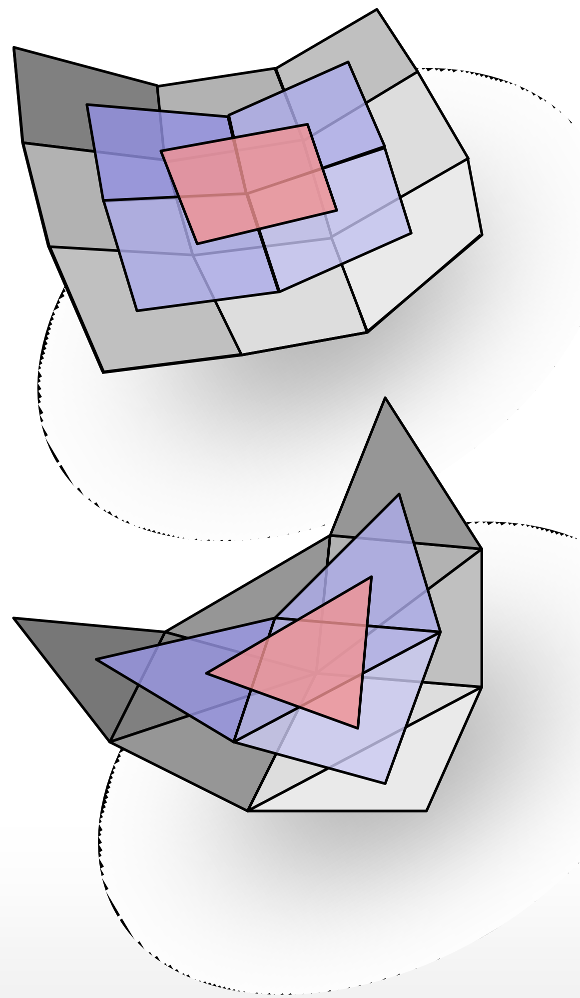




Bezier Triangles

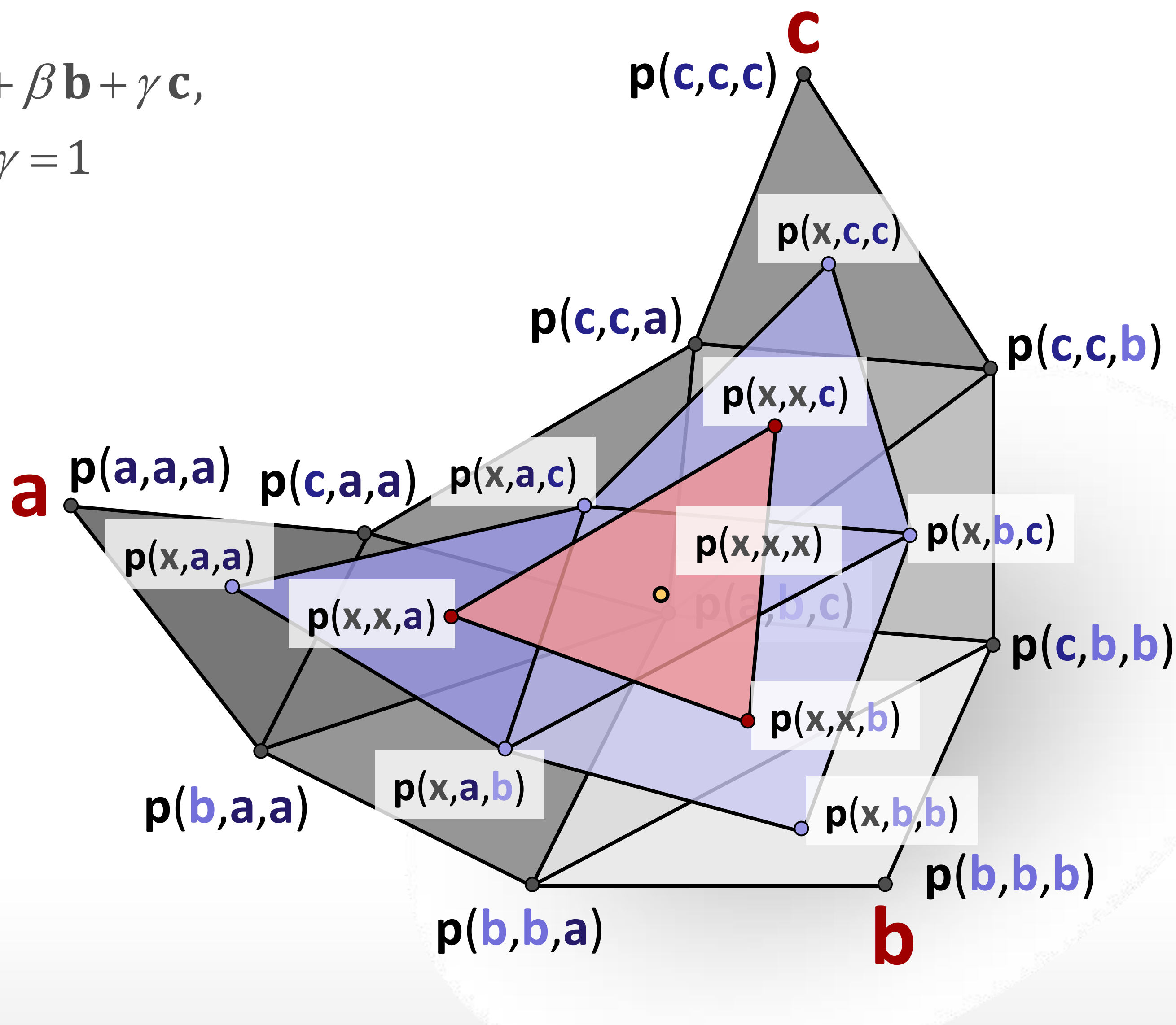
Alternative surface definition: Bezier triangles

- Constructed according to given total degree
  - Completely symmetric: No degree anisotropy
- Can be derived using a triangular de Casteljau algorithm
  - Barycentric interpolation



De Casteljau Algorithm

$$\mathbf{x} = \alpha\,\mathbf{a} + \beta\,\mathbf{b} + \gamma\,\mathbf{c},$$
$$\alpha + \beta + \gamma = 1$$



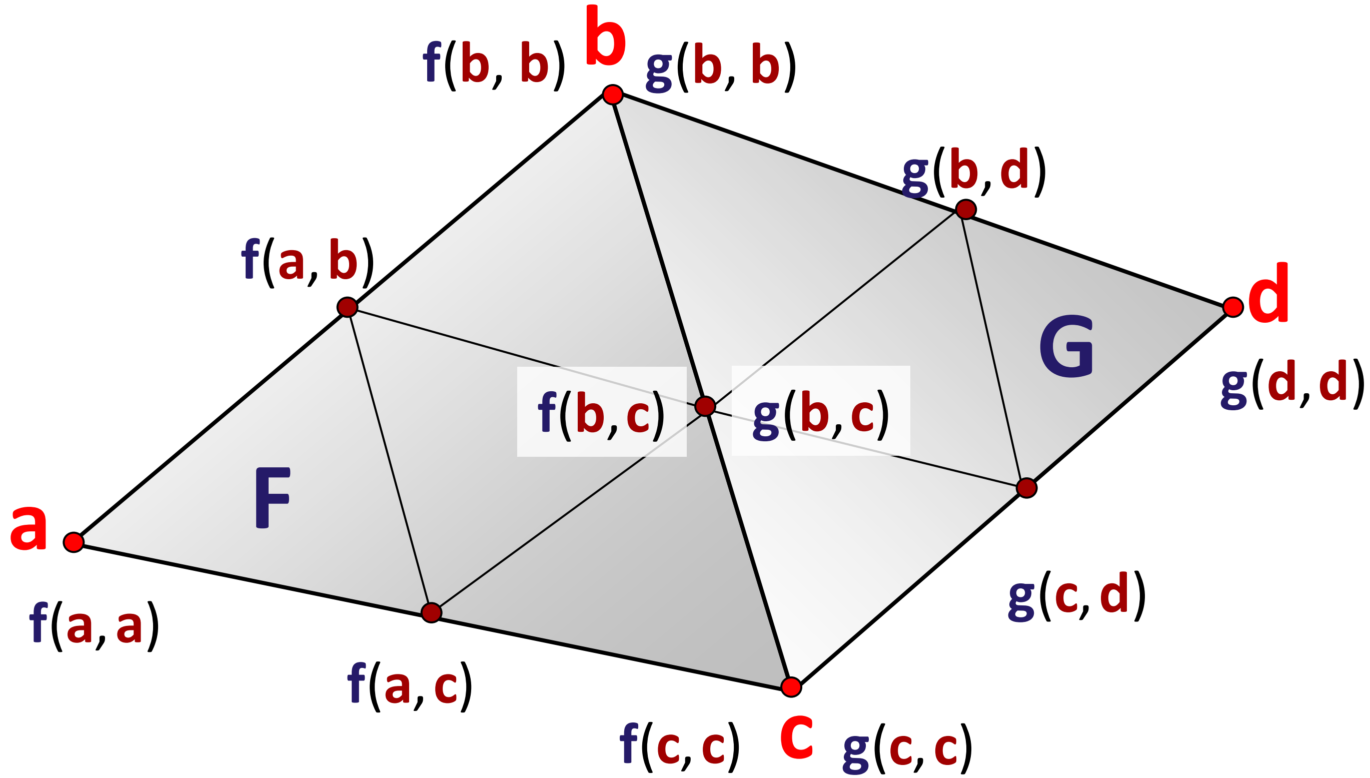
Continuity

We need to assemble Bezier triangles continuously:

- What are the conditions for C^0, C^1 continuity?

Continuity

Situation:



Subdivision Surfaces

Introduction

Subdivision Surfaces

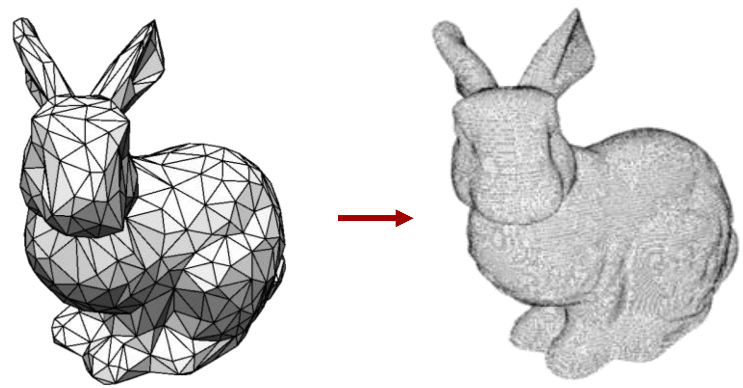
Problems with Spline Patches:

- A continuous tensor product spline surface is only defined on a regular grid of quads as parametrization domain
- Thus, the topology of the object is restricted
- Assembling multiple parameter domains to a single surface is tedious, hard to get continuity guarantees
- Handling trimming curves is not that straightforward

Question: Can we do better?

Subdivision Surfaces

Simple Idea:



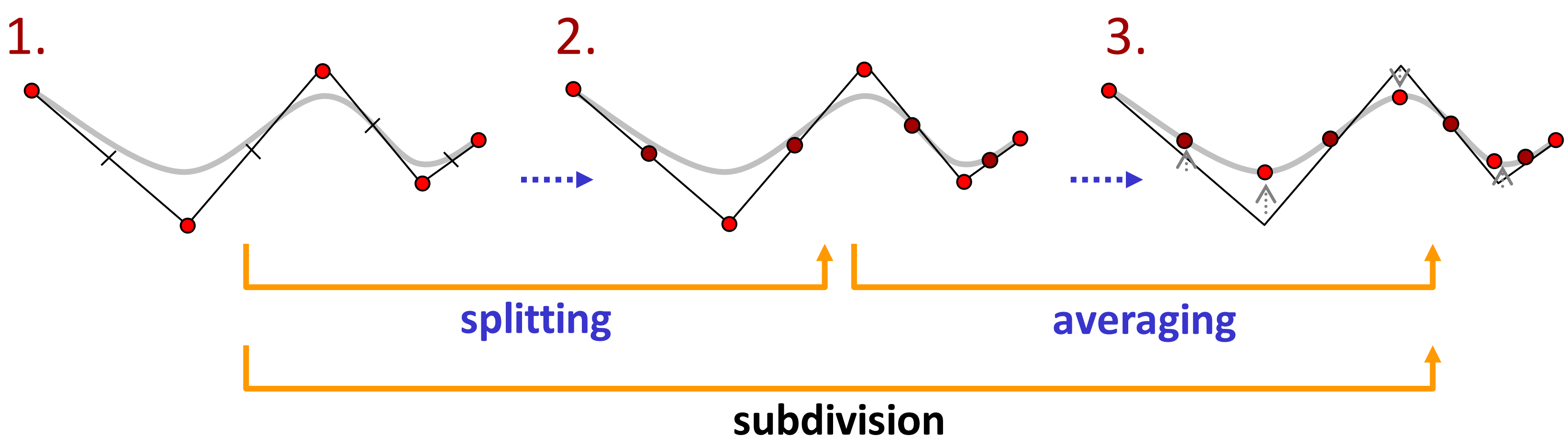
- Use a triangle mesh / quad mesh itself as a parametrization domain ("base mesh")
- Use 1:4 splits to refine the base mesh (subdivision connectivity meshes)
- Find an interpolation rule to create a smooth surface

This basic idea leads to subdivision surfaces.

Basic Scheme

Subdivision Curves & Surface: Three Steps

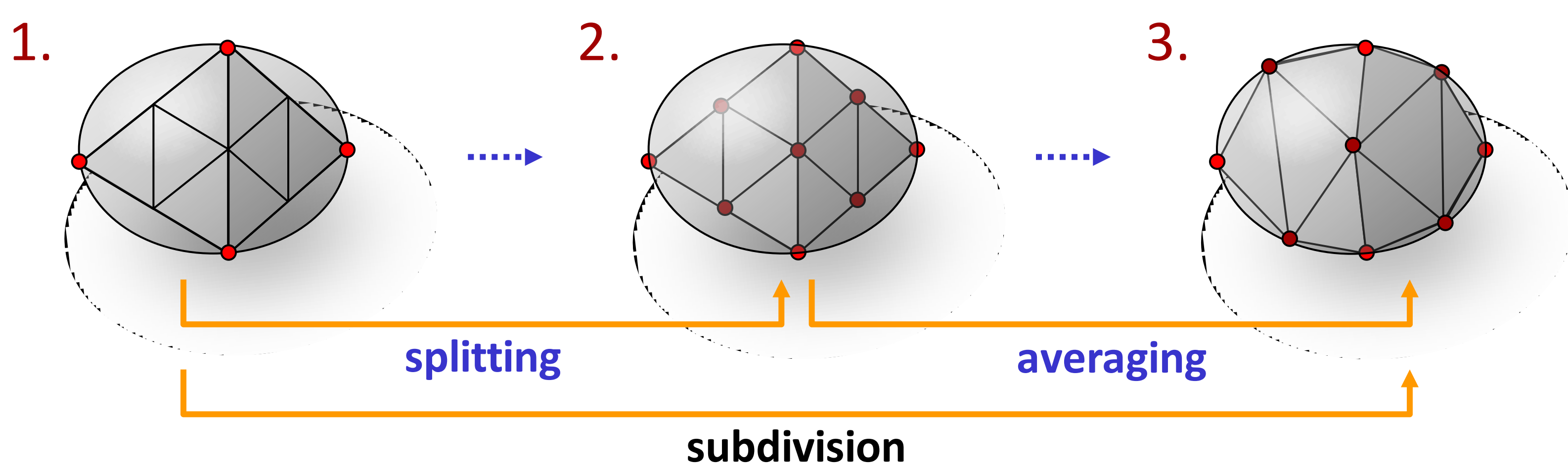
1. Subdivide current mesh
2. Insert linearly interpolated points (*splitting*)
3. Move points: Local weighted average (*averaging*)
   - To all points – approximating scheme
   - To new points only – interpolating scheme



Basic Scheme

Subdivision Curves & Surface: Three Steps

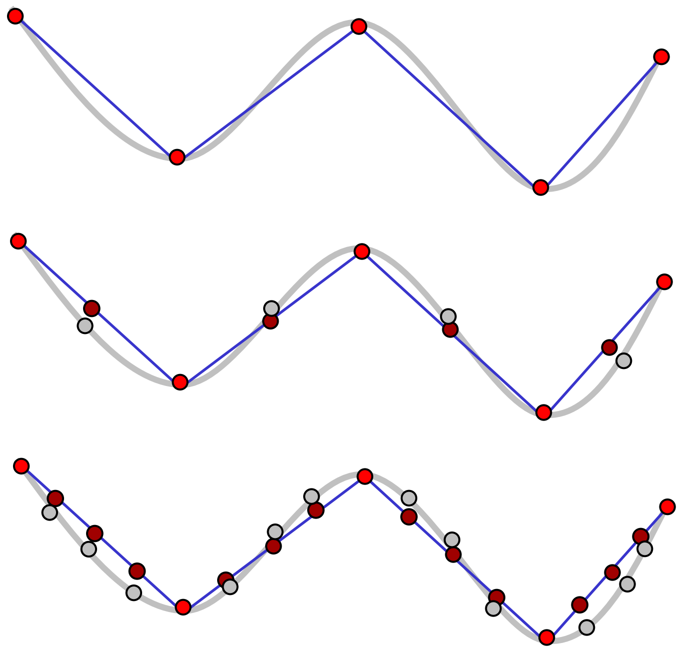
1. Subdivide current mesh
2. Insert linearly interpolated points (*splitting*)
3. Move points: Local weighted average (*averaging*)
   - To all points – approximating scheme
   - To new points only – interpolating scheme

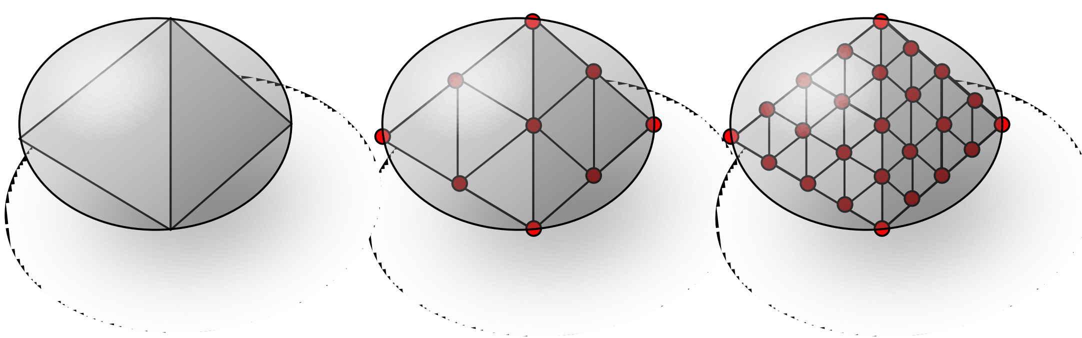


Subdivision Surfaces

The main question is:

- How should we place the new points to create a smooth surface? (interpolating scheme)
- Respectively: How should we alter the points in each subdivision step to create a smooth surface? (approximating scheme)





Subdivision Schemes

More precisely:

- What are good *averaging masks*?
- The averaging mask determines the weights by which the new point positions are computed

Interesting observation:

- Most averaging schemes do not converge (in particular interpolating schemes – try this at home).
- We need to be very careful to design a good averaging mask.
- How can we guarantee C^1, C^2 surfaces?

Subdivision surfaces – History

de Rahm described a 2D (curve) subdivision scheme in 1947; rediscovered in 1974 by Chaikin

Concept extended to 3D (surface) schemes by two separate groups during 1978:

- Doo and Sabin found a biquadratic surface
- Catmull and Clark found a bicubic surface

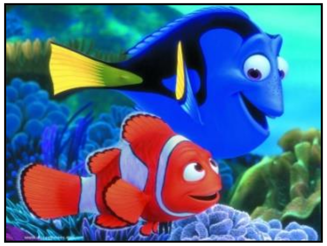
Subsequent work in the 1980s (Loop, 1987; Dyn [Butterfly subdivision], 1990) led to tools suitable for CAD/CAM and animation

Subdivision surfaces and the movies

Pixar first demonstrated subdivision surfaces in 1997 with Geri's Game.

- Up until then they'd done everything in NURBS (Toy Story, A Bug's Life.)
- From 1999 onwards everything they did was with subdivision surfaces (Toy Story 2, Monsters Inc, Finding Nemo...)

It's not clear what Dreamworks uses, but they have recent patents on subdivision techniques.



Video about Subdivision Surfaces

http://youtu.be/ckOTI2GcS-E

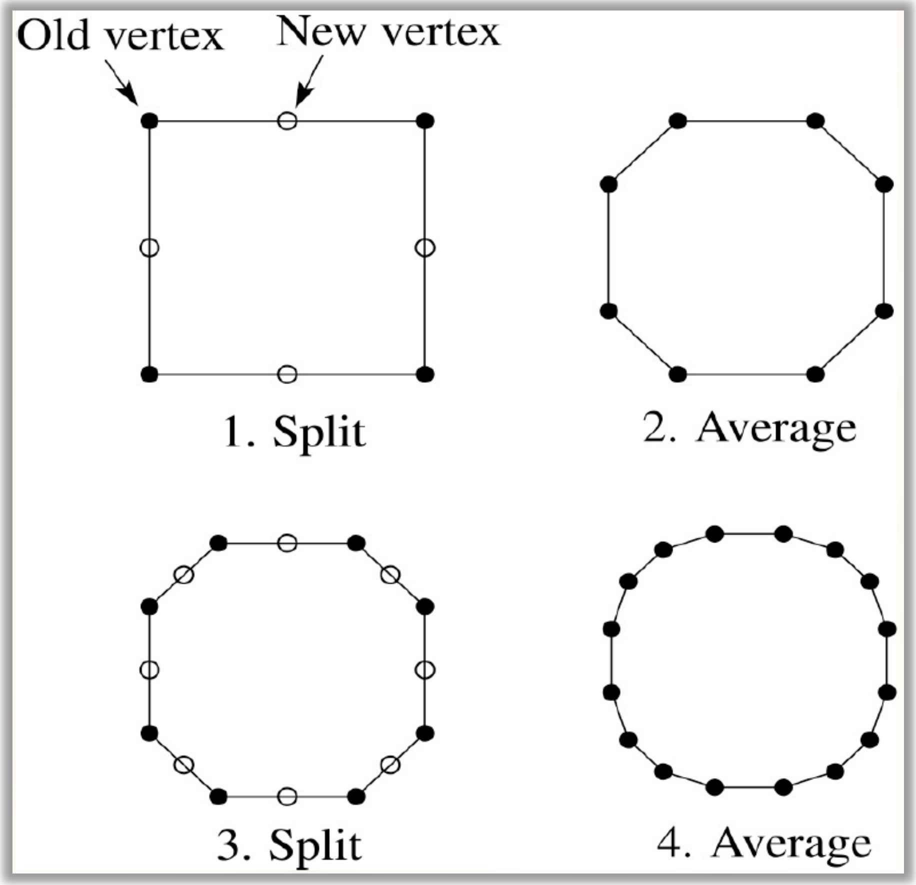
Example: Corner Cutting

very important

Corner Cutting Splines [Chaikin 1974]:

1. Split each line segment in half.
2. Average every point with its next neighbor (clock-wise).
3. Repeat.

- Converges to quadratic B-Spline curve



Matrix Notation

Curve subdivision in matrix notation:

- Control points

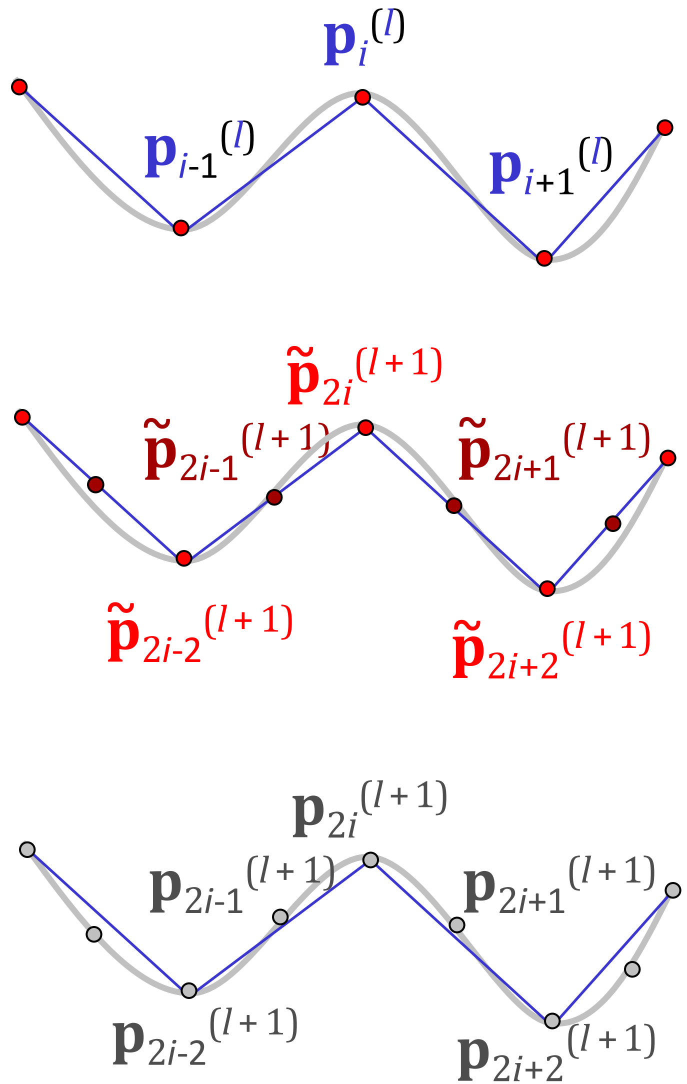
  at level l: $\mathbf{p}_i^{(l)}$

- "Splitted" points

  at level $l+1$: $\tilde{\mathbf{p}}_i^{(l+1)}$

- "Averaged" control points

  at level $l+1$: $\mathbf{p}_i^{(l+1)}$



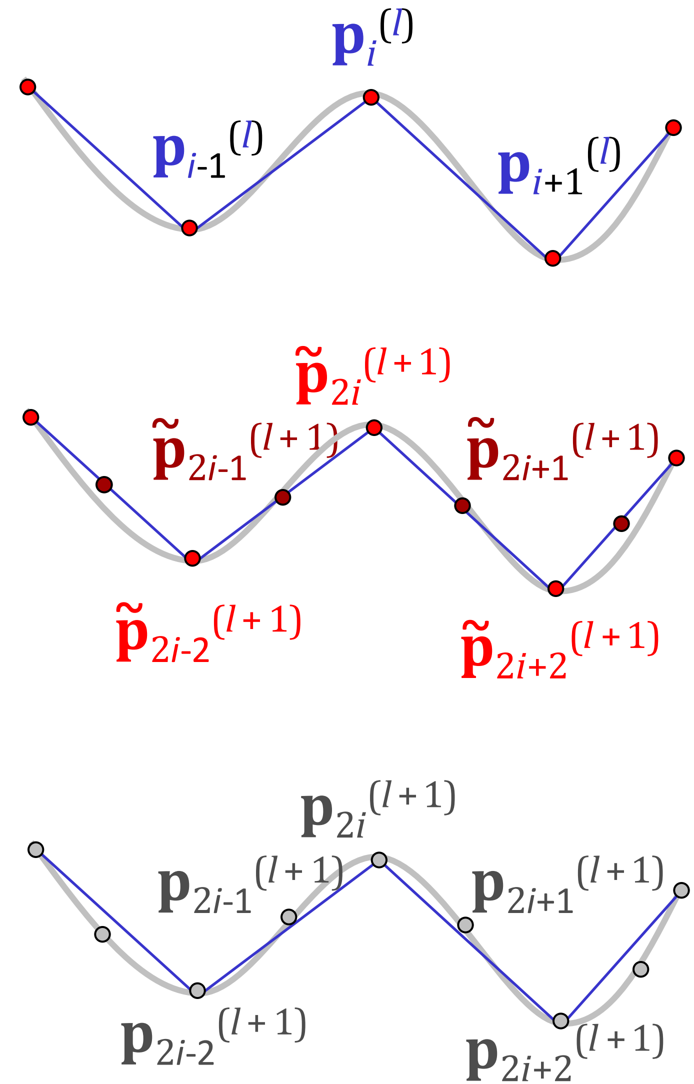
Matrix Notation

Splitting in matrix notation:

$$2n\left\{\begin{pmatrix}\vdots\\ \tilde{x}_{2i}^{(l+1)}\\ \tilde{x}_{2i+1}^{(l+1)}\\ \vdots\end{pmatrix}\right. = 2n\left\{\underbrace{\begin{pmatrix}\ddots &&&&&\\ &1&&&&\\ &1/2&1/2&&&\\ &&&1&&\\ &&&1/2&1/2&\\ &&&&&\ddots\end{pmatrix}}_{n}\right.\left.\begin{pmatrix}\vdots\\ x_i^{(l)}\\ x_{i+1}^{(l)}\\ \vdots\end{pmatrix}\right\}n$$

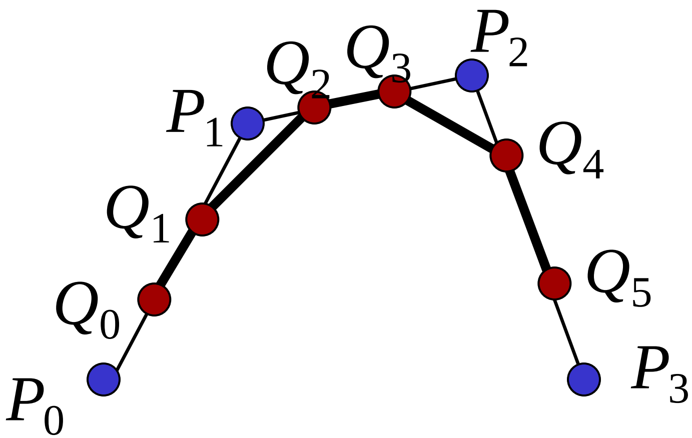
Averaging in matrix notation:

$$2n\left\{\begin{pmatrix}\vdots\\ x_{2i}^{(l+1)}\\ x_{2i+1}^{(l+1)}\\ \vdots\end{pmatrix}\right. = 2n\left\{\underbrace{\begin{pmatrix}\ddots &&&\\ &\frac{1}{2}&\frac{1}{2}&\\ &&\frac{1}{2}&\frac{1}{2}\\ &&&\ddots\end{pmatrix}}_{2n}\right.\left.\begin{pmatrix}\vdots\\ \tilde{x}_{2i}^{(l+1)}\\ \tilde{x}_{2i+1}^{(l+1)}\\ \vdots\end{pmatrix}\right\}2n$$



a different view on the same algorithm…

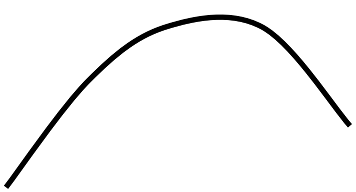
Chaikin's Corner Cutting



Apply Iterated Function System

$$Q_{2i} = \frac{3}{4} P_i + \frac{1}{4} P_{i+1}$$

$$Q_{2i+1} = \frac{1}{4} P_i + \frac{3}{4} P_{i+1}$$



Limit Curve Surface

$$Q_0 = \frac{3}{4} P_0 + \frac{1}{4} P_1$$

$$Q_1 = \frac{1}{4} P_0 + \frac{3}{4} P_1$$

$$Q_2 = \frac{3}{4} P_1 + \frac{1}{4} P_2$$

$$Q_3 = \frac{1}{4} P_1 + \frac{3}{4} P_2$$

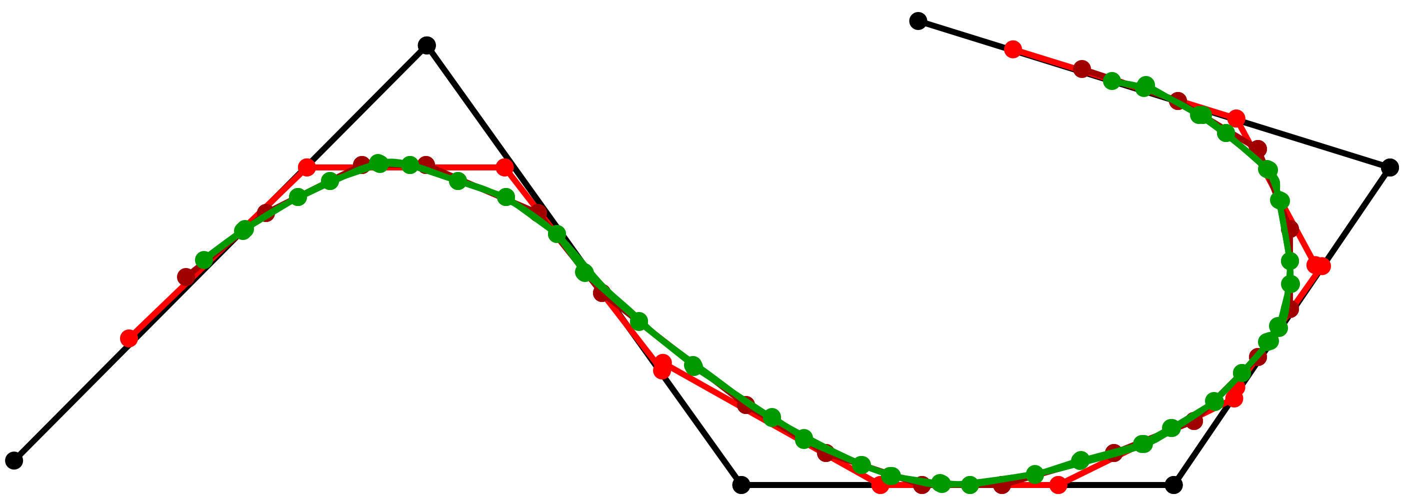
$$Q_4 = \frac{3}{4} P_2 + \frac{1}{4} P_3$$

$$Q_5 = \frac{1}{4} P_2 + \frac{3}{4} P_3$$

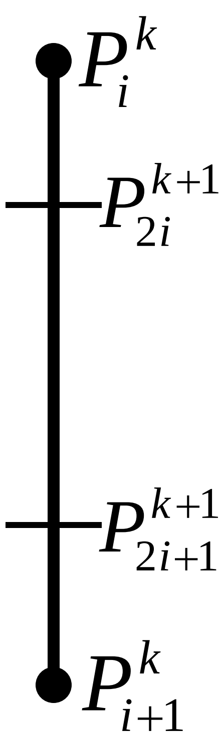
Chaikin‘s Corner Cutting

Chaikin curve subdivision (2D)

- On each edge, insert new control points at ¼ and ¾ between old vertices; delete the old points
- The *limit curve* is C1 everywhere



Chaikin's Corner Cutting



Chaikin can be written programmatically as:

$$P_{2i}^{k+1} = (3/4)P_i^k + (1/4)P_{i+1}^k \quad \leftarrow \textit{Even}$$

$$P_{2i+1}^{k+1} = (1/4)P_i^k + (3/4)P_{i+1}^k \quad \leftarrow \textit{Odd}$$

- …where *k* is the 'generation'; each generation will have twice as many control points as before.
- Notice the different treatment of generating odd and even control points.
- Borders (terminal points) are a special case.

Chaikin's Corner Cutting

Chaikin can be written in vector notation as:

$$
\begin{bmatrix} \vdots \\ P_{2i-2}^{k+1} \\ P_{2i-1}^{k+1} \\ P_{2i}^{k+1} \\ P_{2i+1}^{k+1} \\ P_{2i+2}^{k+1} \\ P_{2i+3}^{k+1} \\ \vdots \end{bmatrix}
= \frac{1}{4}
\begin{bmatrix}
\ddots & & & & & & & \iddots \\
& 0 & 3 & 1 & 0 & 0 & 0 & \\
& 0 & 1 & 3 & 0 & 0 & 0 & \\
& 0 & 0 & 3 & 1 & 0 & 0 & \\
& 0 & 0 & 1 & 3 & 0 & 0 & \\
& 0 & 0 & 0 & 3 & 1 & 0 & \\
& 0 & 0 & 0 & 1 & 3 & 0 & \\
\iddots & & & & & & & \ddots
\end{bmatrix}
\begin{bmatrix} \vdots \\ P_{i-2}^{k} \\ P_{i-1}^{k} \\ P_{i}^{k} \\ P_{i+1}^{k} \\ P_{i+2}^{k} \\ P_{i+3}^{k} \\ \vdots \end{bmatrix}
$$

Chaikin‘s Corner Cutting

The standard notation compresses the scheme to a *kernel*:

- h =(1/4)[…,0,0,1,3,3,1,0,0,…]

The kernel interlaces the odd and even rules.

It also makes matrix analysis possible: eigenanalysis of the matrix form can be used to prove the continuity of the subdivision limit surface.

The limit curve of Chaikin is a quadratic B-spline!

Cubic B-Spline Subdivision Scheme

Cubic Subdivision

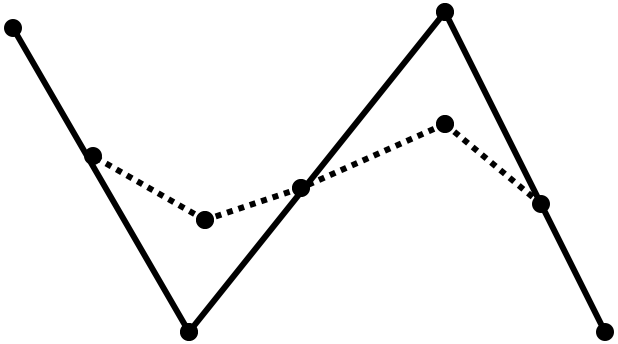
Consider the kernel

- h=(1/8)[…,0,0,1,4,6,4,1,0,0,…]

You would read this as

$$P_{2i}^{k+1} = (1/8)(P_{i-1}^{k} + 6P_{i}^{k} + P_{i+1}^{k})$$

$$P_{2i+1}^{k+1} = (1/8)(4P_{i}^{k} + 4P_{i+1}^{k})$$



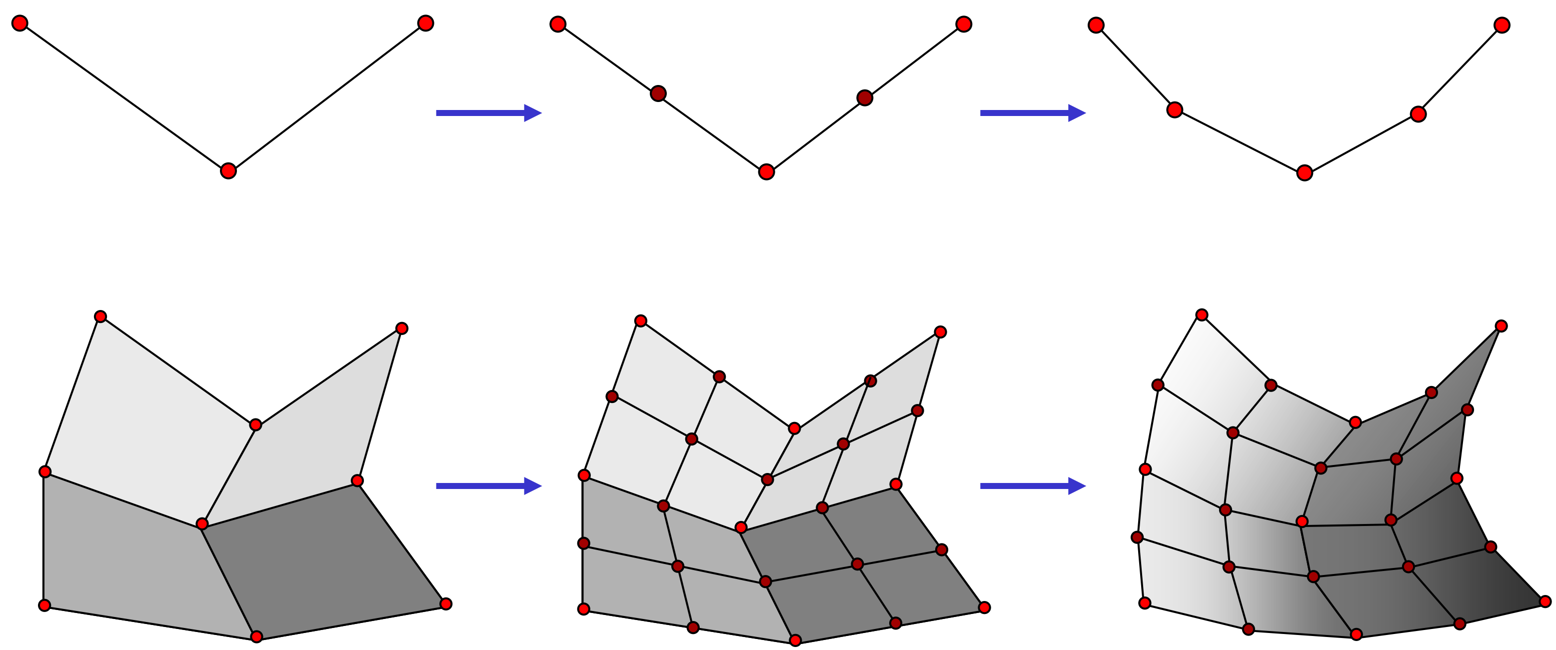
The limit curve is provably C2-continuous.

B-Spline Subdivision Surfaces

B-Spline Subdivision Surfaces

B-Spline Subdivision Surfaces

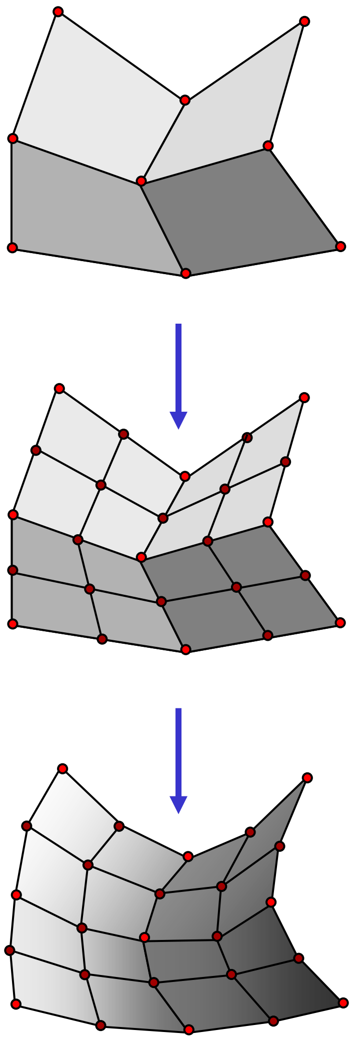
- We can apply the tensor product construction to obtain subdivision surfaces:



B-Spline Subdivision Surfaces

Tensor Product B-Spline Subdivision Surfaces:

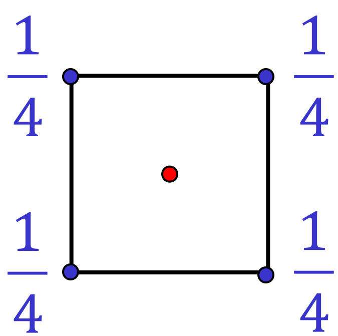
- Start with a regular quad mesh (will be relaxed later)
- In each subdivision step:
  - Divide each quad in four (quadtree subdivision)
  - Place linearly interpolated vertices
  - Apply 2-dimensional averaging mask



Subdivision and Averaging Masks

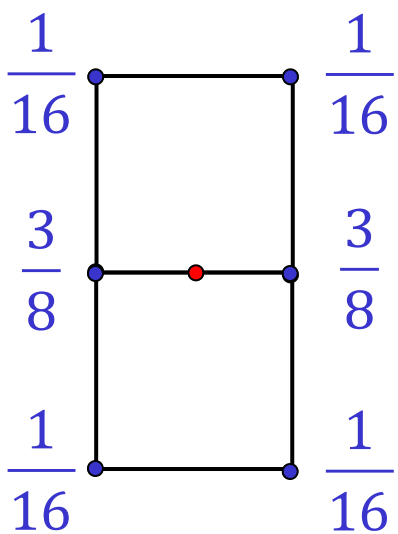
What is the subdivision mask?

- Can be derived from tensor product construction:



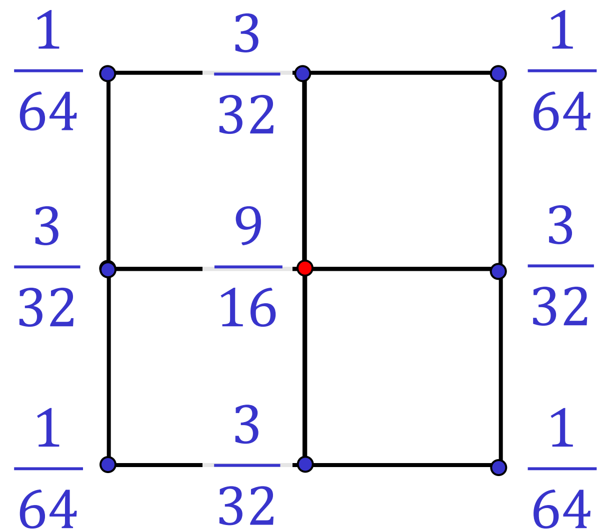
face midpoint
(odd/odd)

$$\begin{pmatrix}\frac{1}{2}\\ \frac{1}{2}\end{pmatrix}\cdot\left[\frac{1}{2},\frac{1}{2}\right]$$



edge midpoint
(even/odd)

$$\begin{pmatrix}\frac{1}{2}\\ \frac{1}{2}\end{pmatrix}\cdot\left[\frac{1}{8},\frac{3}{4},\frac{1}{8}\right]$$



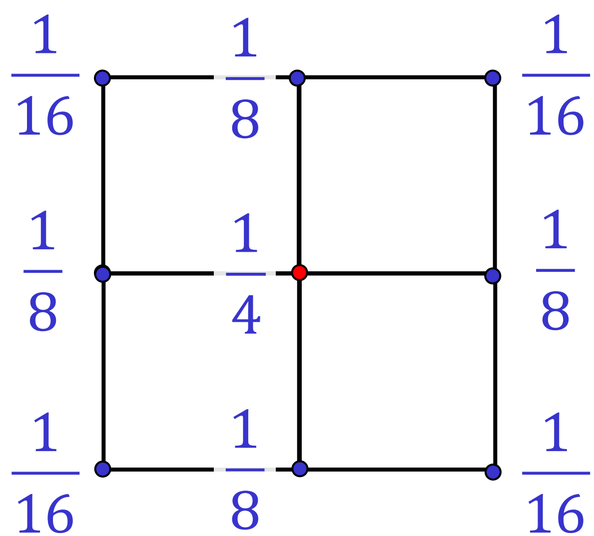
original vertex
(even/even)

$$\begin{pmatrix}\frac{1}{8}\\ \frac{3}{4}\\ \frac{1}{8}\end{pmatrix}\cdot\left[\frac{1}{8},\frac{3}{4},\frac{1}{8}\right]$$

Subdivision and Averaging Masks

What is the averaging mask?

- Can be derived from tensor product construction, too:



any (split) vertex

$$\begin{pmatrix} \frac{1}{4} \\ \frac{1}{2} \\ \frac{1}{4} \end{pmatrix} \cdot \left[\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\right]$$

Generalizations

Generalizations:

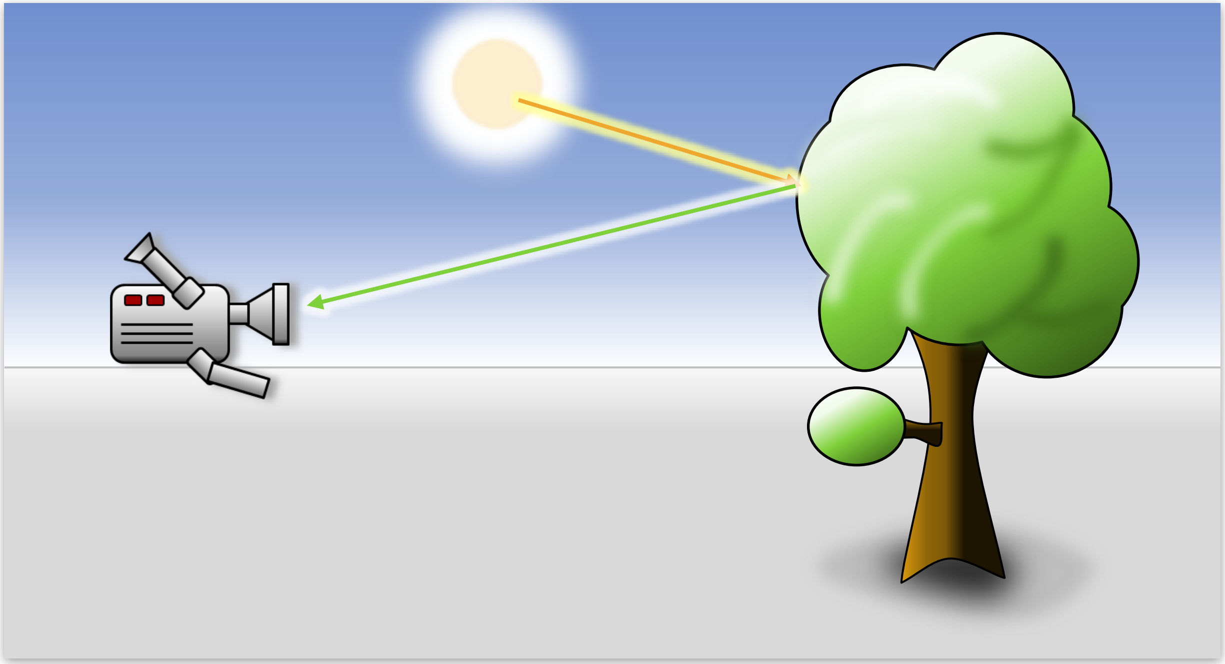
- General degree B-spline tensor product surface subdivision rules can be derived in the same way
- Just use the 1-dimensional subdivision / averaging masks and form a tensor product mask
- Guaranteed to converge to a C^{d-1}-smooth surface
- Extraordinary vertices need to be treated properly:
  - Catmull-Clark subdivision scheme, and others

Color

Physics, Biology

Ray Optics & Color

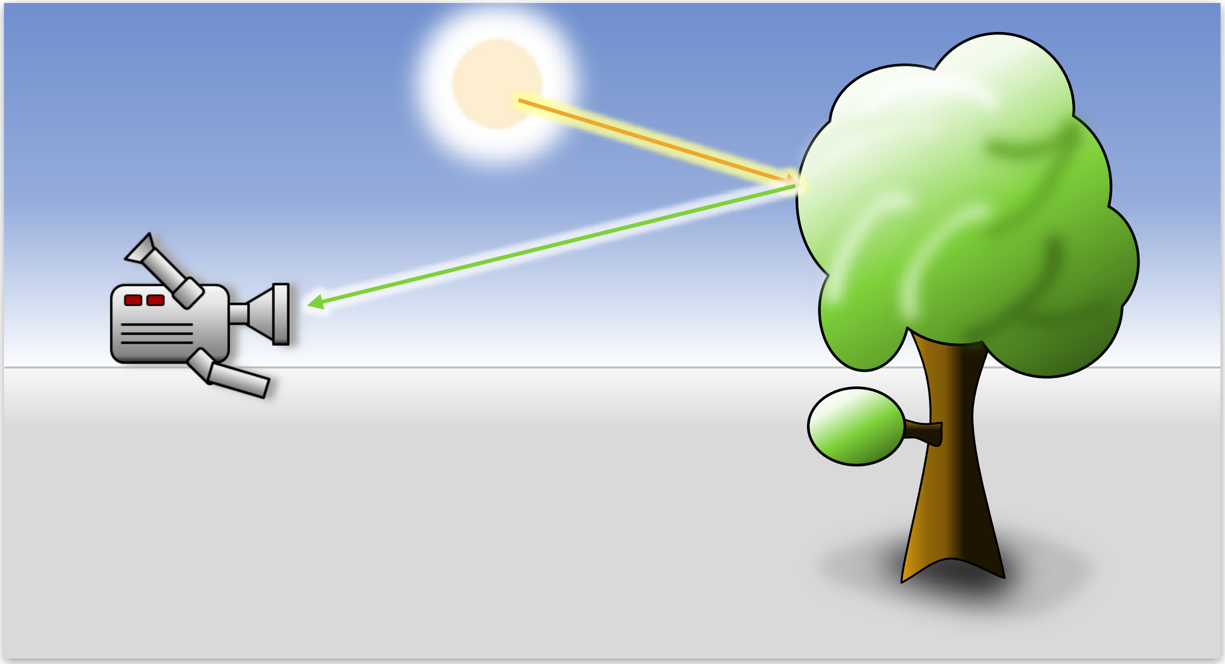
Ray Optics



Geometric ray model

- Light travels along rays

Ray Optics



Geometric ray model

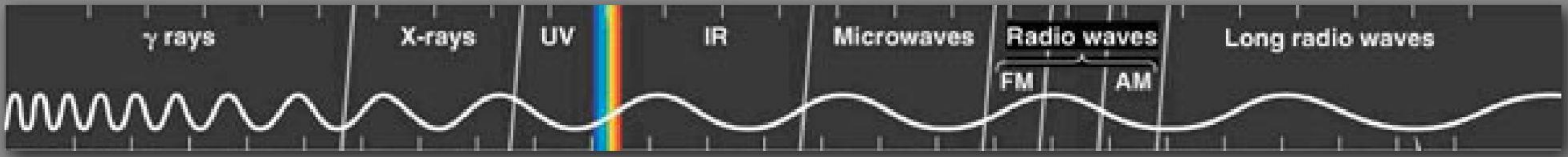
- Rays have “intensity” and “color”

What is COLOR?

- Next slides mostly from Kristi Potter (U Utah)

The Electromagnetic Spectrum

Range of all possible frequencies of electromagnetic radiation

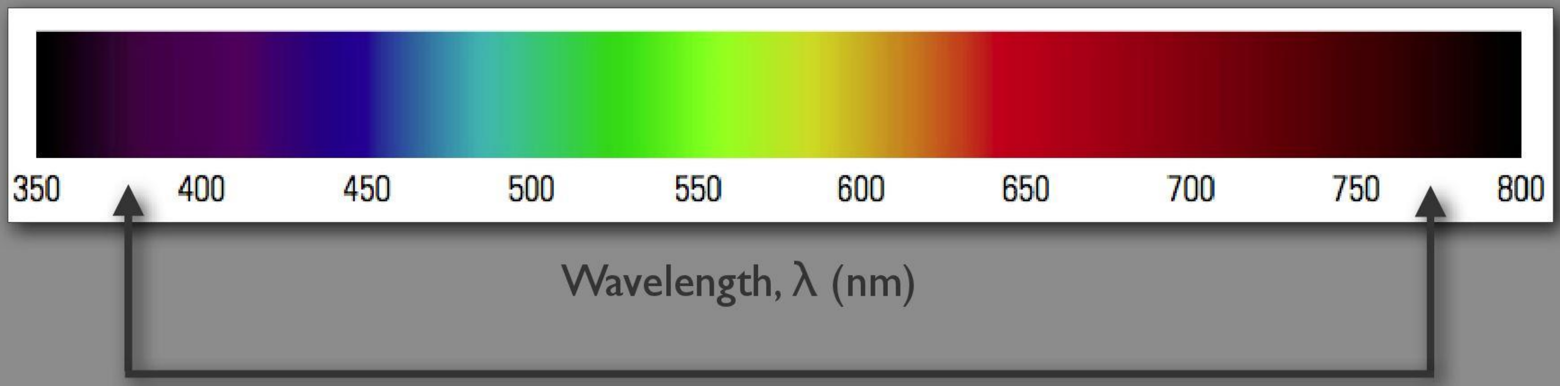


10^{-16} 10^{-7} 10^{8}

Wavelength, λ (m)

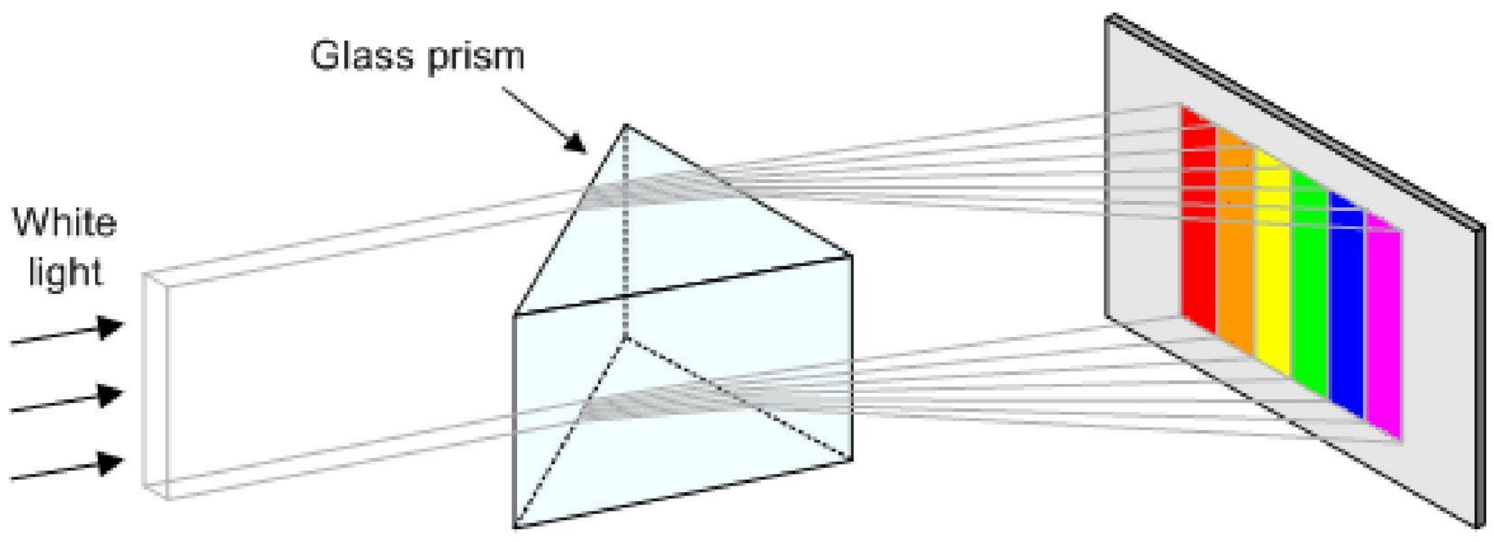
The Visible Spectrum

Human Visual System Sensitive to 380-780 nm



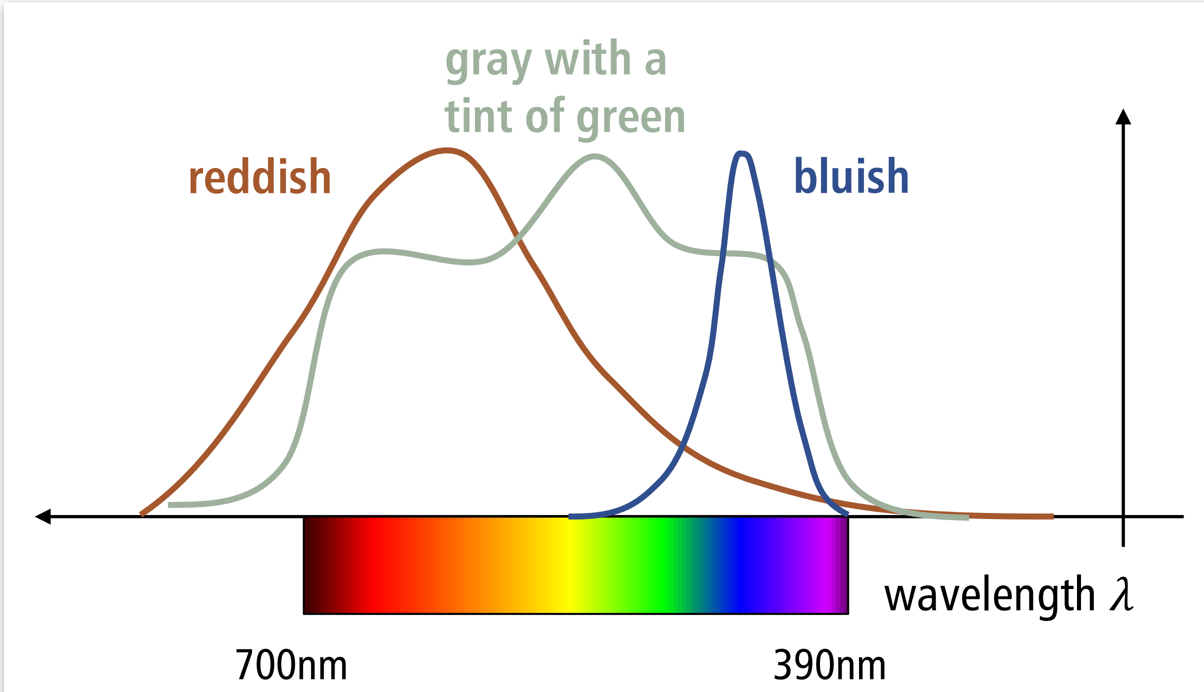
Isaac Newton

Objects appear colored by the character of the light they reflect



Newton's experiment for splitting white light into a spectrum

Ray Optics



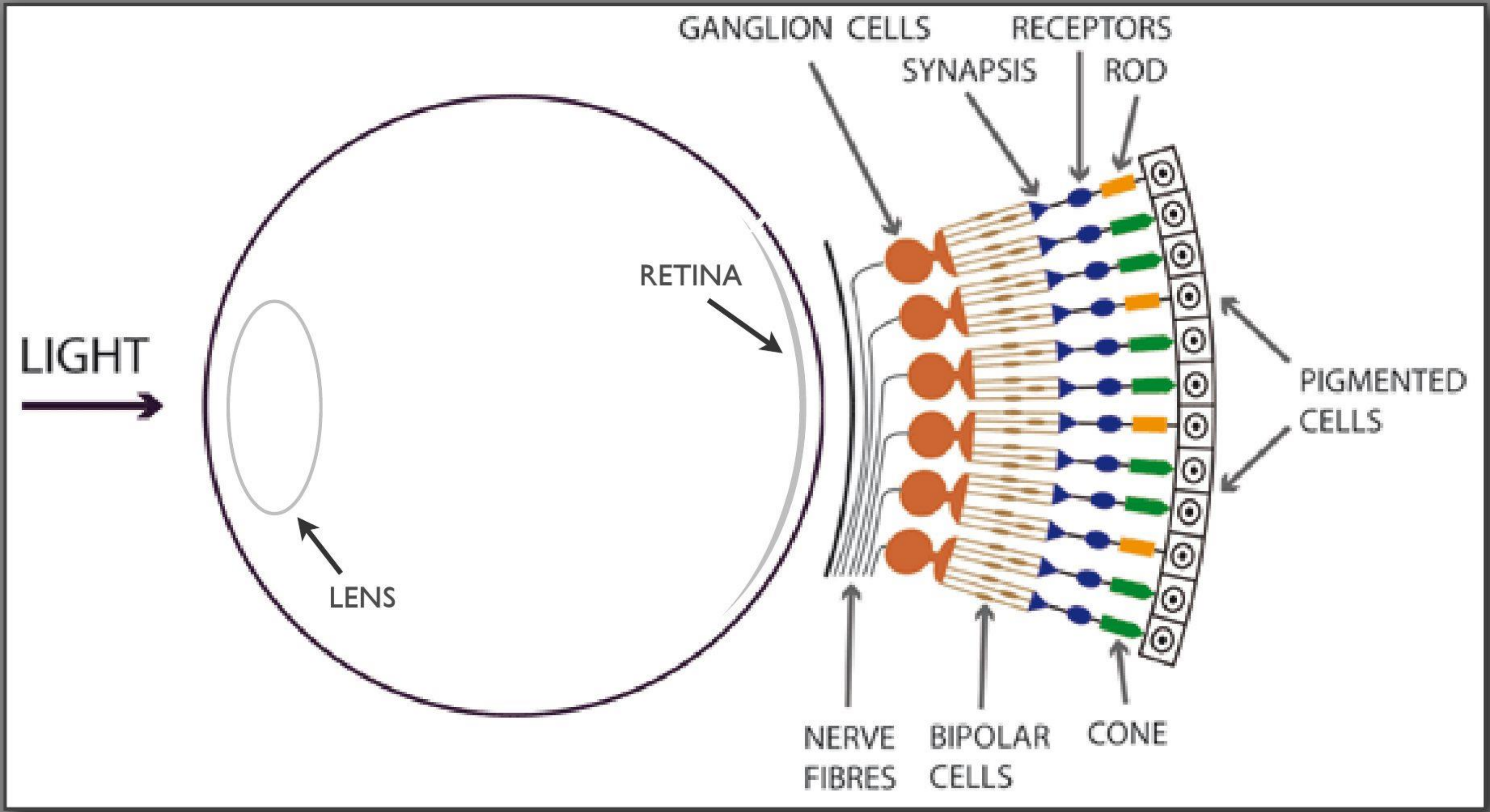
Color spectrum

- Continuous spectrum
- Intensity for each wavelength

Human Color Perception

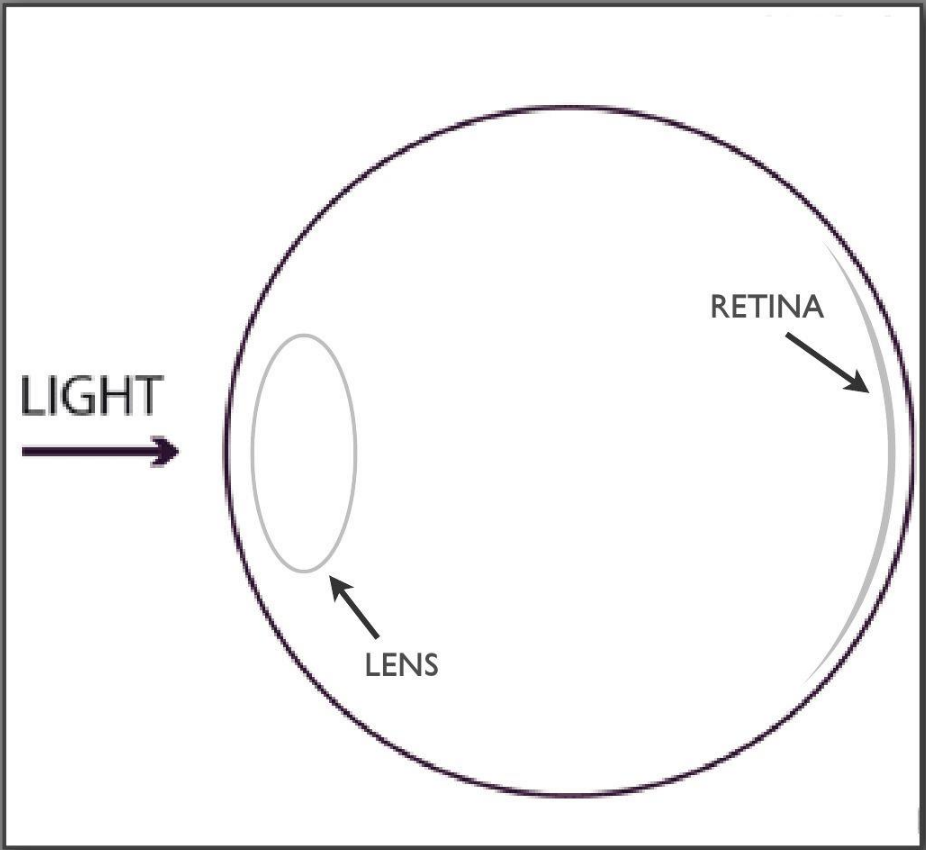
The Eye

Not like a camera



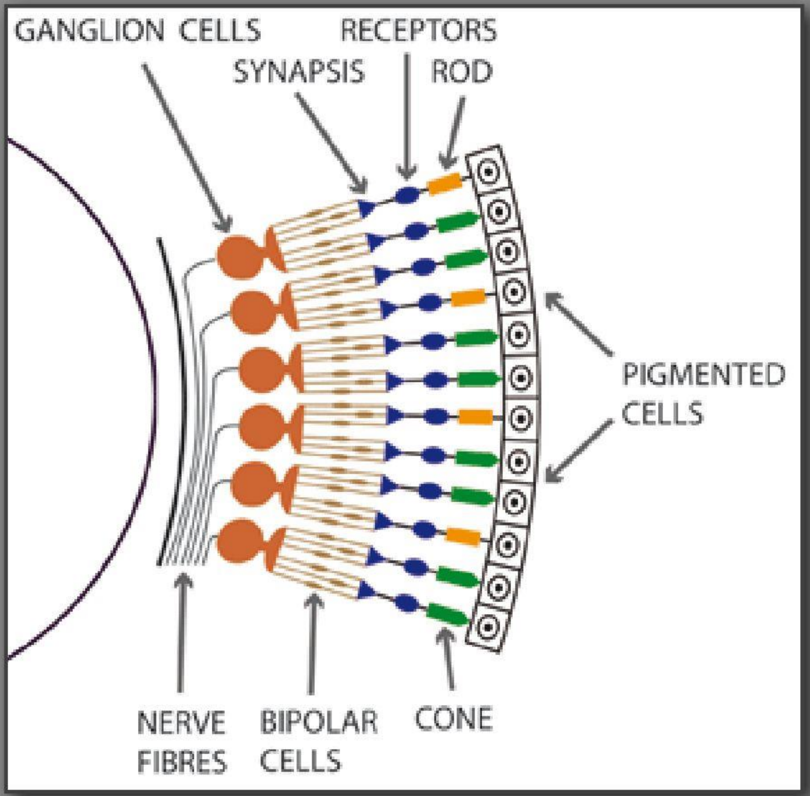
Passage of Light

- Light ⟶ Lens
- Lens ⟶ Retina



Retina

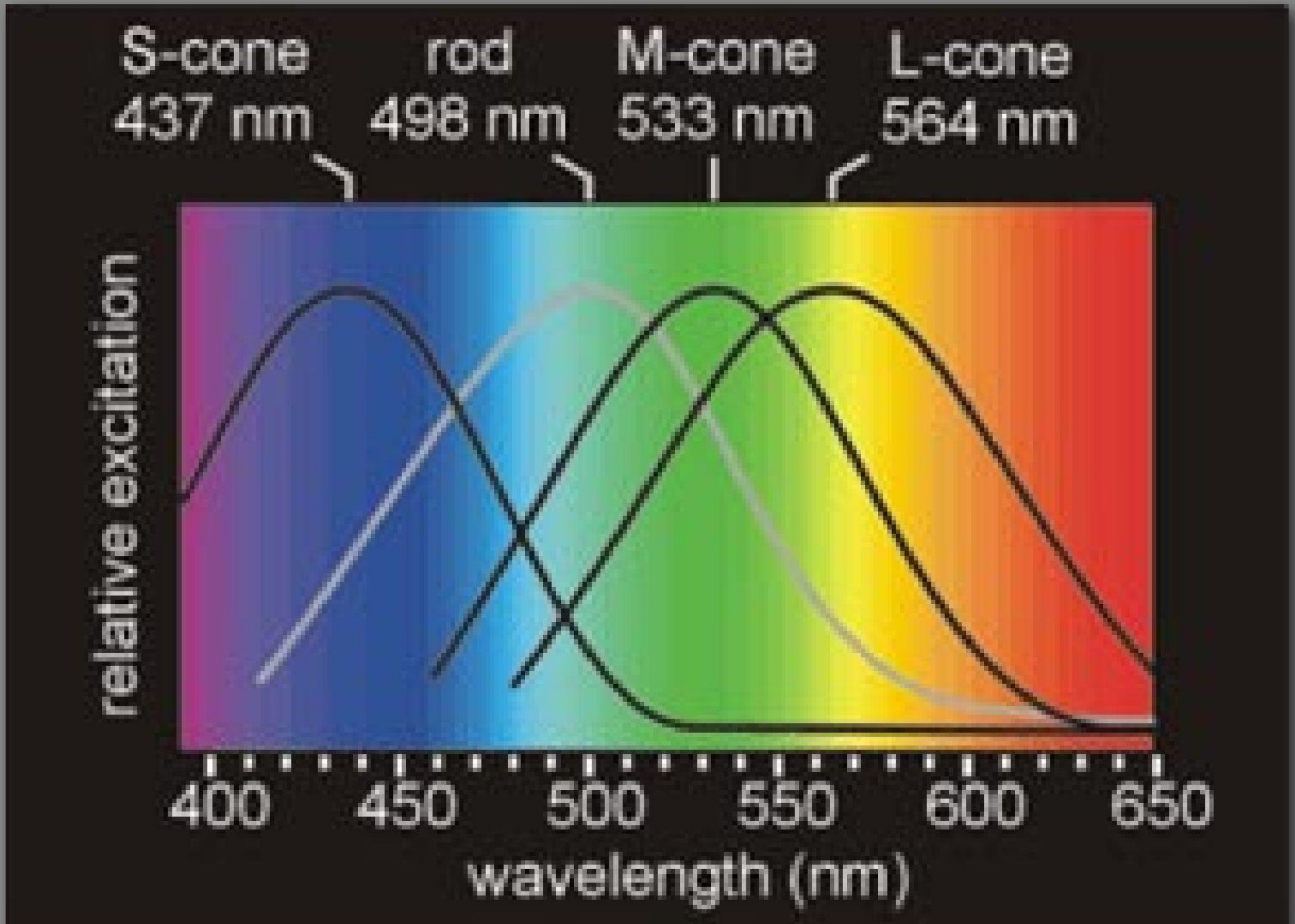
- Optimized for acuity (rather than light sensitivity)
- Initiate information extraction
- Pigmented cells absorb light



Photoreceptors

- Cones for day vision (small, medium, long)
- Rods for night vision
- Binary signal on/off
- Info indicated by which cell & how often

Color Coded through Signal Comparison



Why 3?

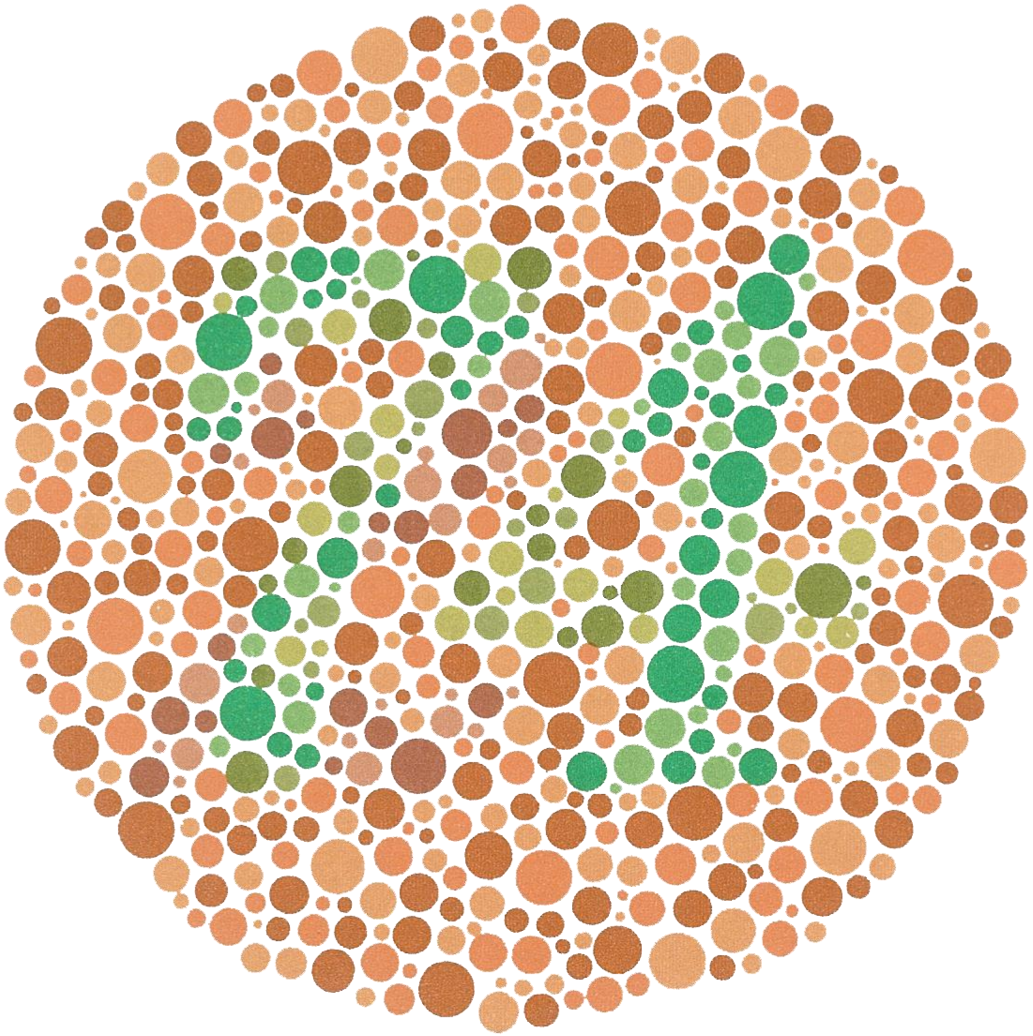
- Our 3 cones cover the visible spectrum
- Theoretically possible with only 2 cones
- Most birds, some fish, reptiles & insects have 4

- Color is:
  - A spectral distribution of light
  - Perceptual response to spectral distribution of light
  - A way of encoding a spectral distribution of light

Computer Graphics → Visualization ← Vision Science / Neuro Science

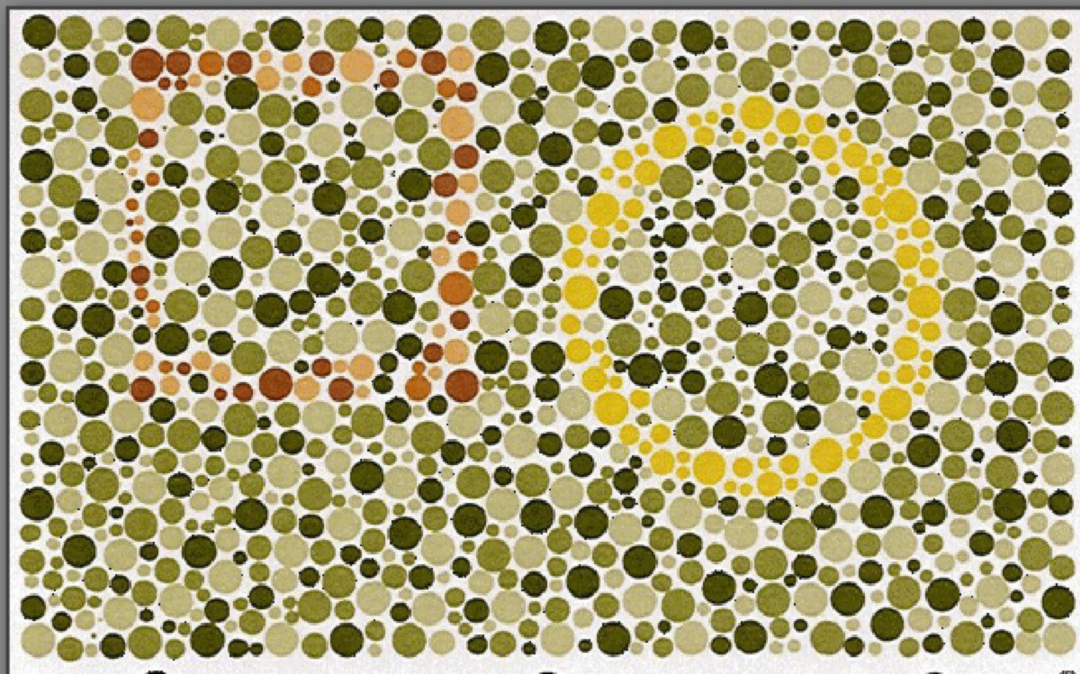
- It would be too simplistic to describe color just as
  - A particular wavelength of light
  - RGB

Color Blindness



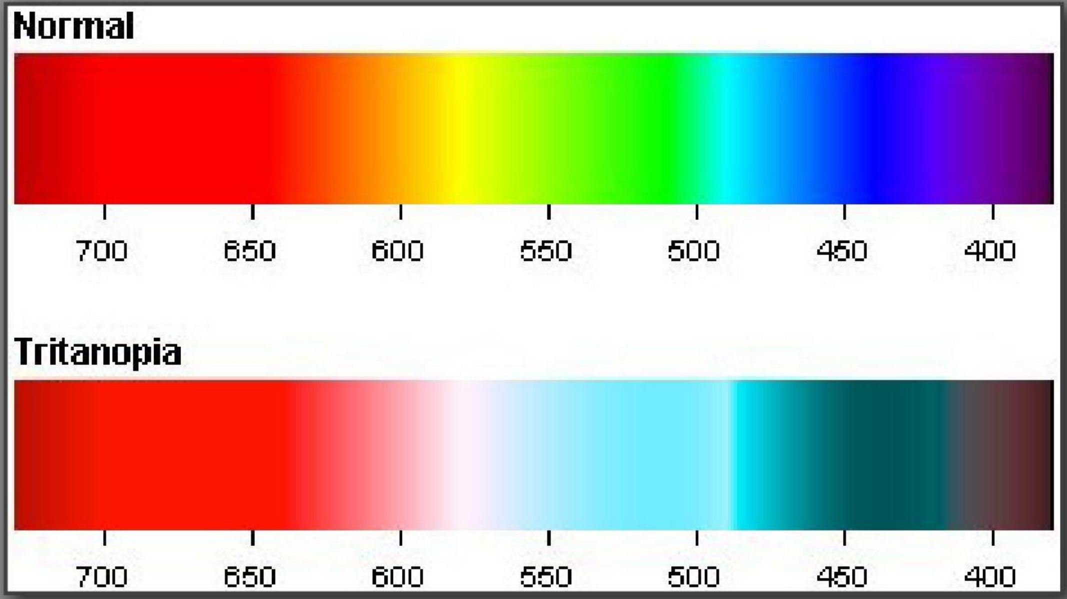
The numeral "74" should be clearly visible to viewers with normal color vision.
Viewers with dichromacy or anomalous trichromacy may read it as "21“.
Viewers with achromatopsia may not see numbers.
From http://en.wikipedia.org/wiki/Color_blindness

Red/Green

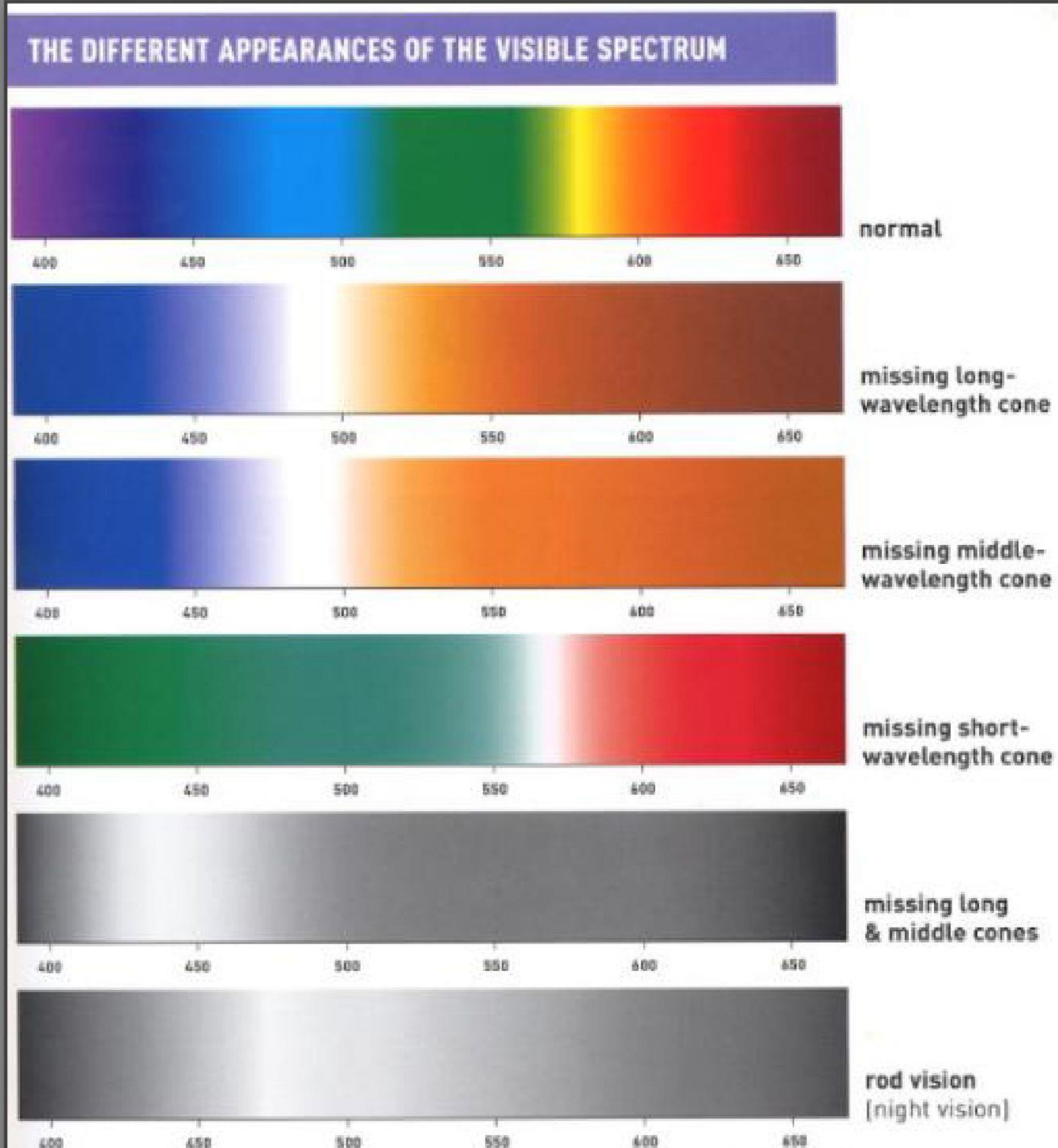


- Lack of or mutations in red or green cones
- Genes located on the X chromosome (women have 2, men have 1)
- 10% of men, less than 1% of women

Blue/Yellow



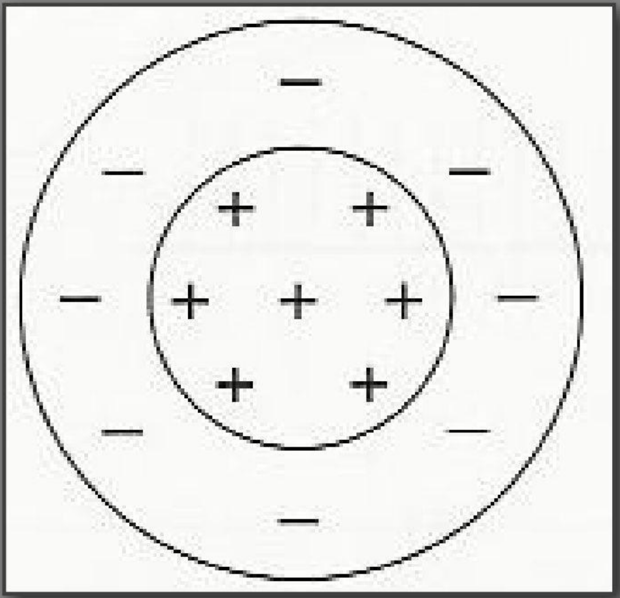
- Equally found in men & women
- mutation in short wave cone


THE DIFFERENT APPEARANCES OF THE VISIBLE SPECTRUM
normal
missing long-wavelength cone
missing middle-wavelength cone
missing short-wavelength cone
missing long & middle cones
rod vision
(night vision)
400
450
500
550
600
650

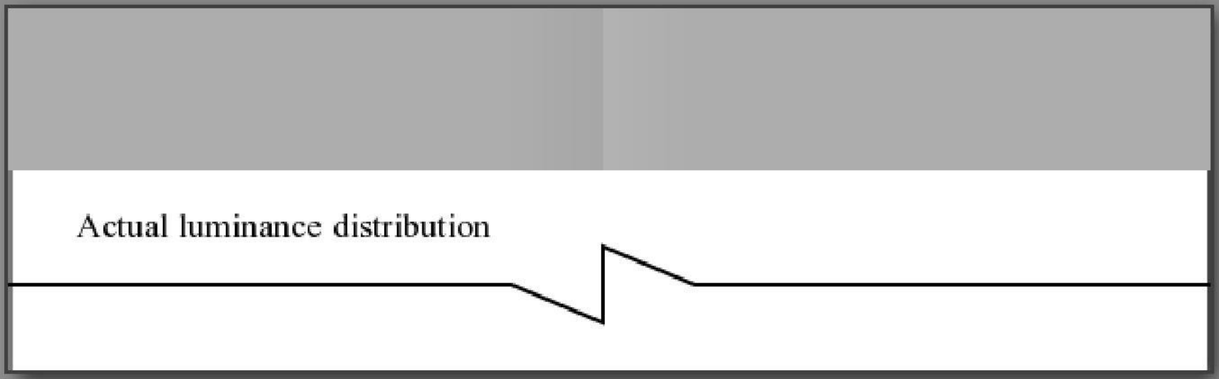
Color Illusions

Center/Surround

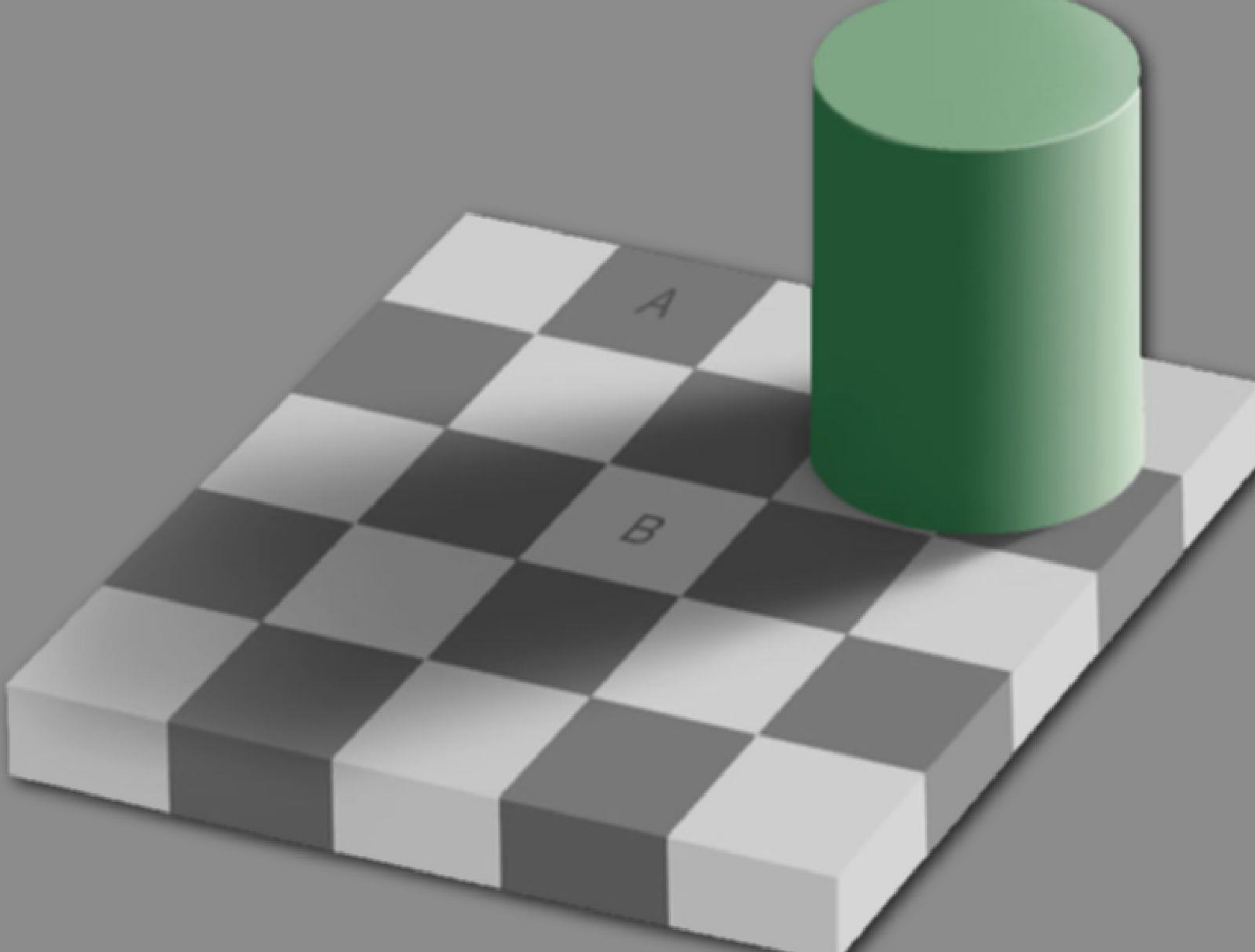
- Retinal ganglion cells
- First stage of visual processing
- Triggered by light in the center suppressed by light in the surround
- Selectively sensitive to discontinuities in light

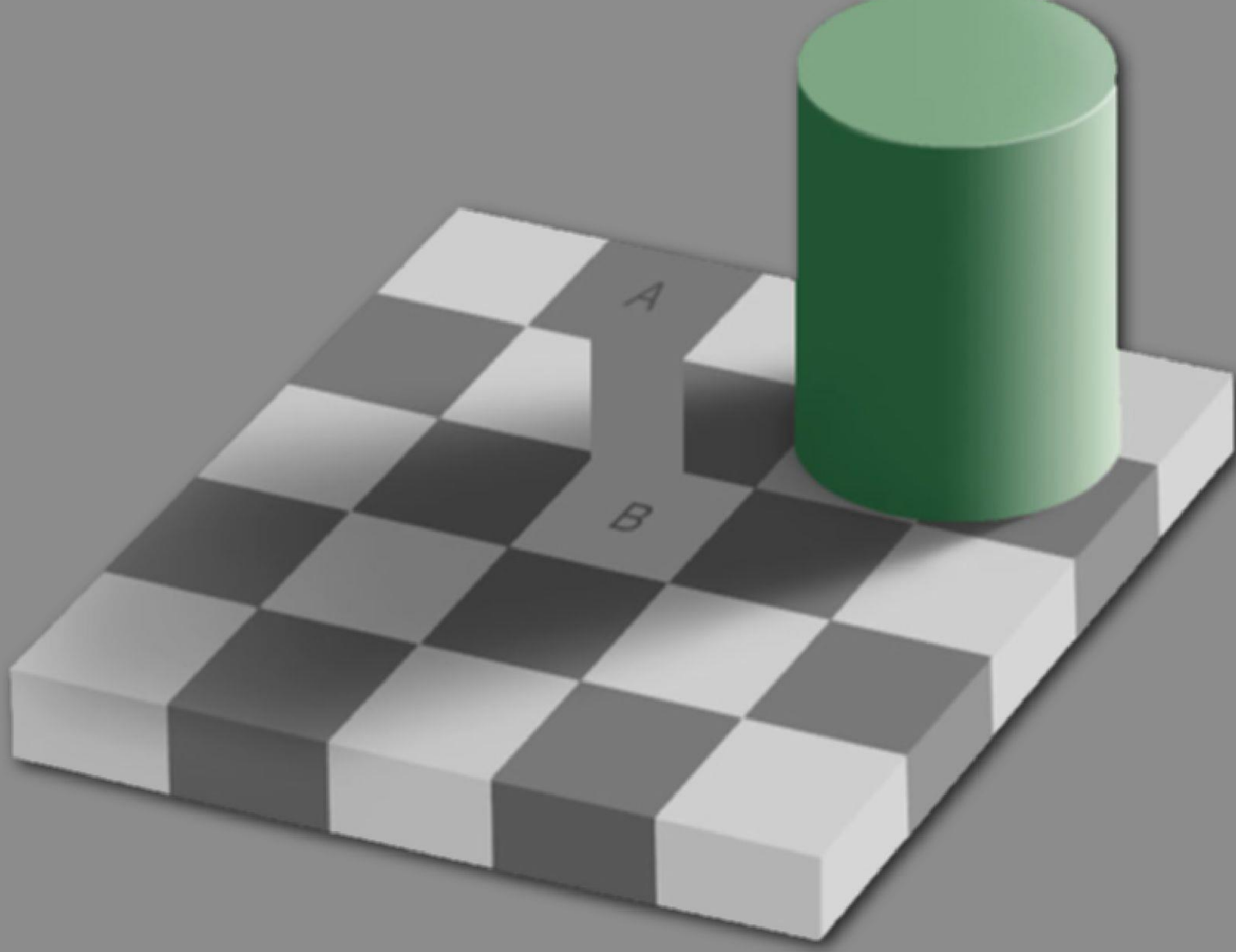


Cornsweet Illusion



- Luminance the same at the ends
- Many perceptions more sensitive to abrupt change (luminance, color, motion, depth)
- Attributed to center/surround organization


A
B

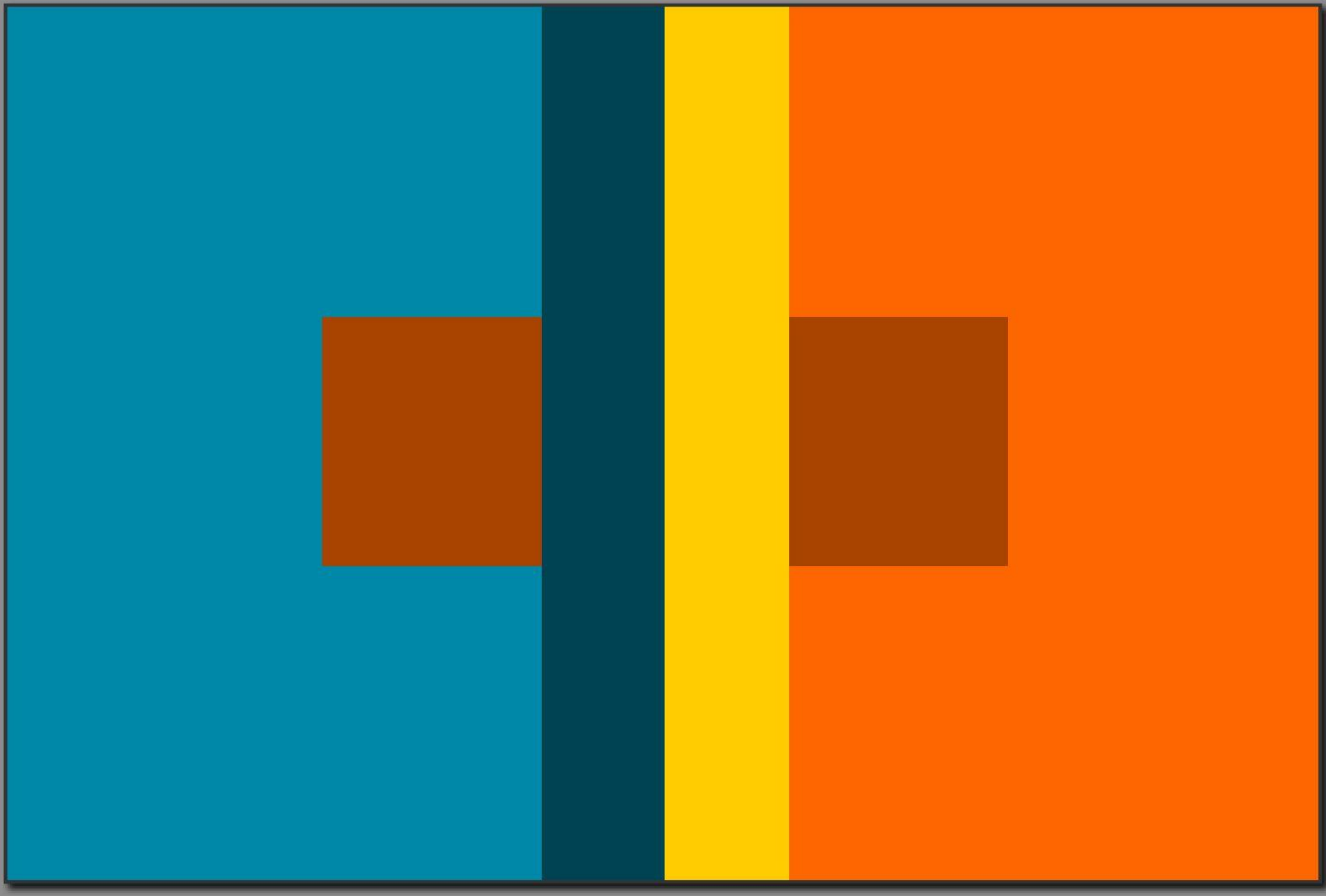

A
B

Contrast Effects

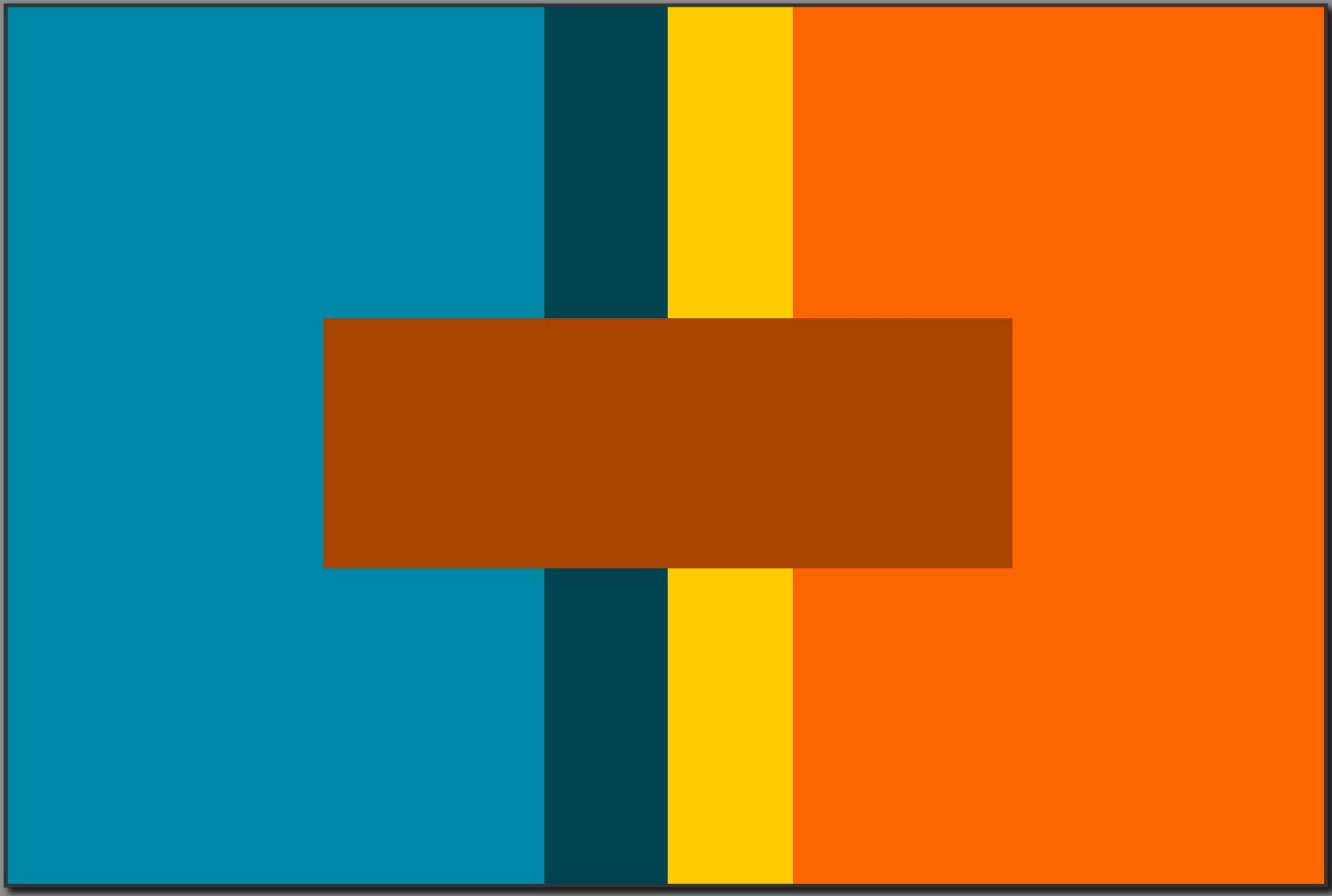
- Result of center/surround cells
- Simultaneous or successive
- Juxtaposition of colors effects our perception of them
- Complimentary colors often most effected

- The terms "simultaneous contrast" and "successive contrast" refer to visual effects in which the appearance of a patch of light (the "test field") is affected by other light patches ("inducing fields") that are nearby in space and time, respectively.
- The names are somewhat misleading since both simultaneous and successive contrast involve inducing fields that are close in both time and space.

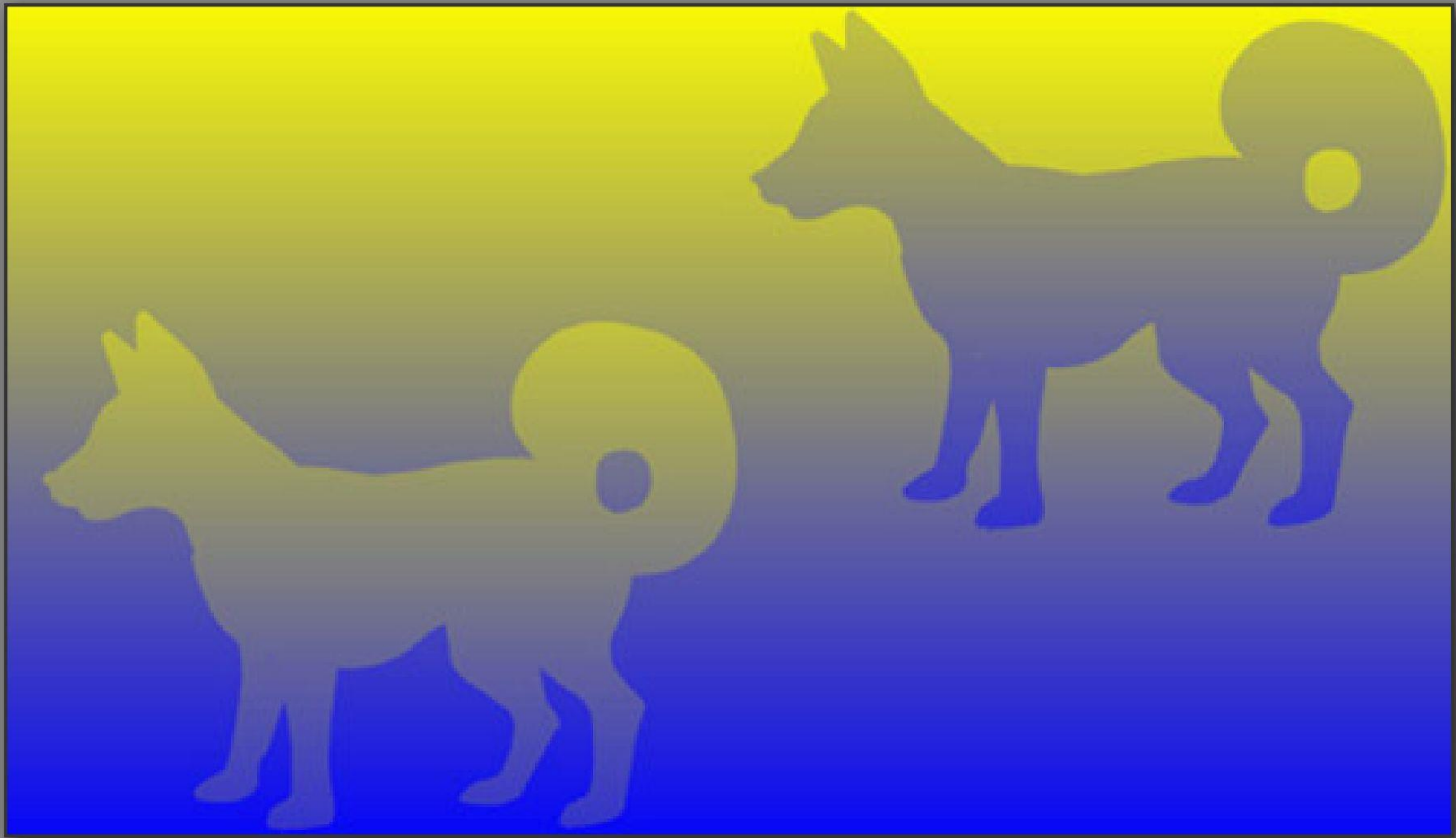
Simultaneous Contrast



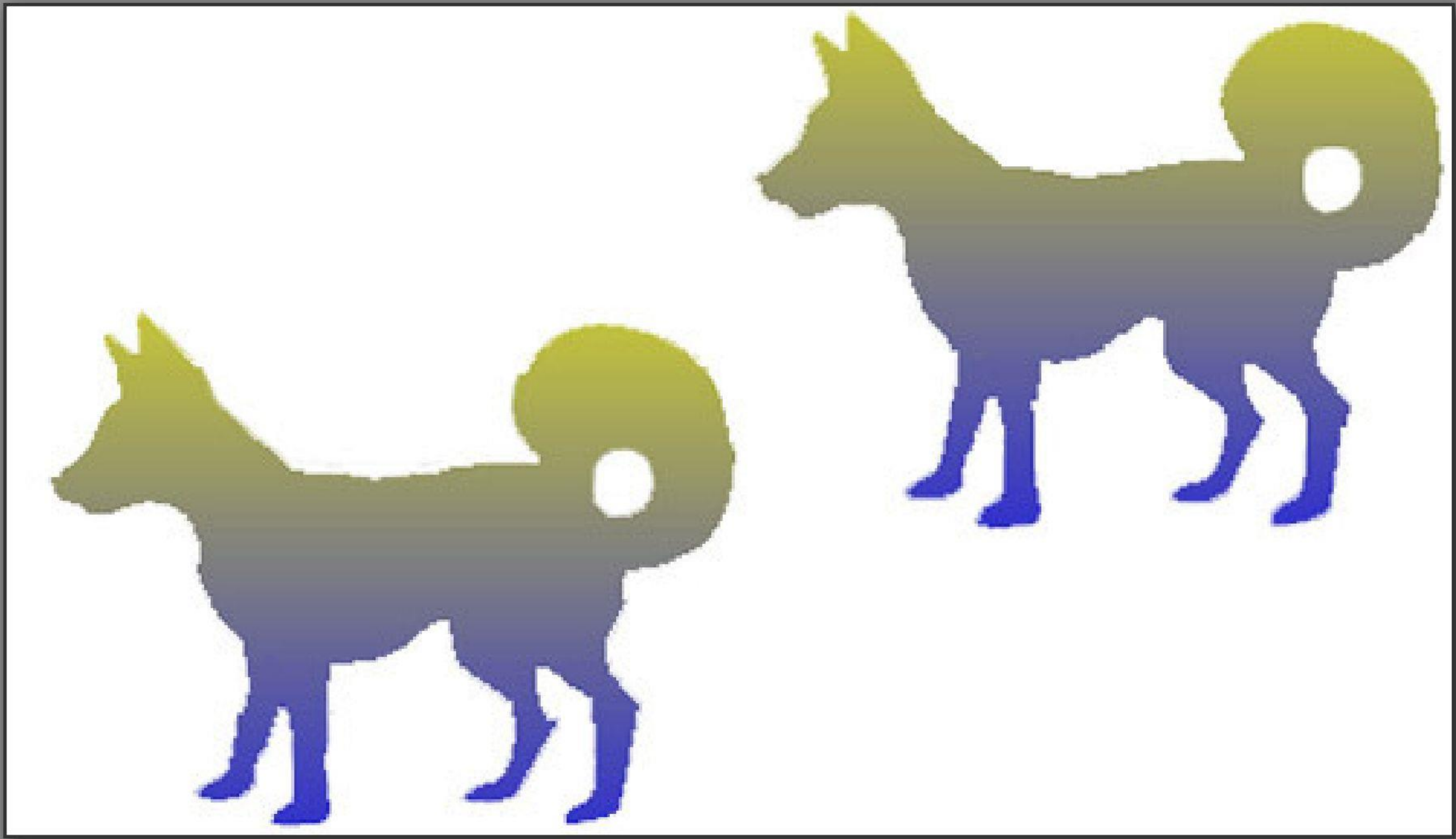
Simultaneous Contrast



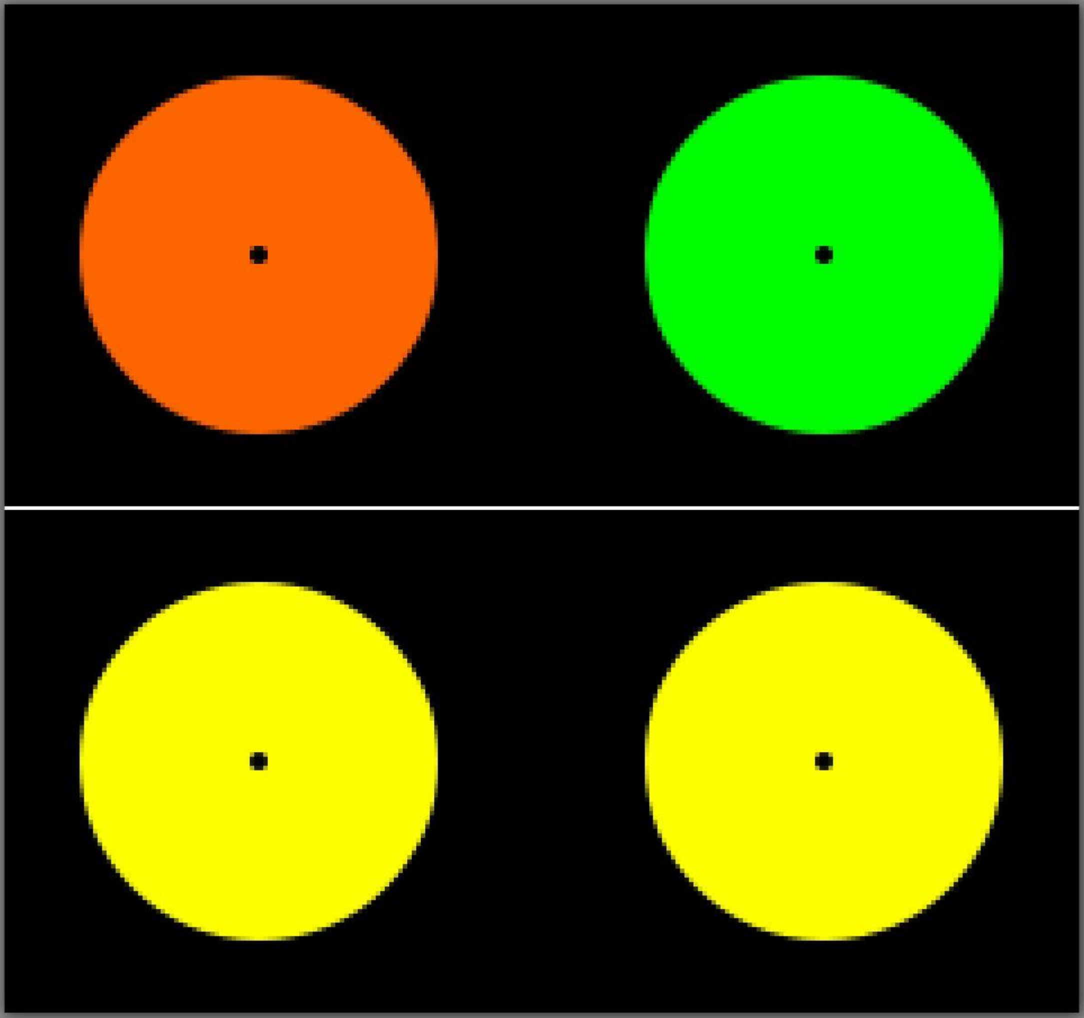
Simultaneous Contrast



Simultaneous Contrast



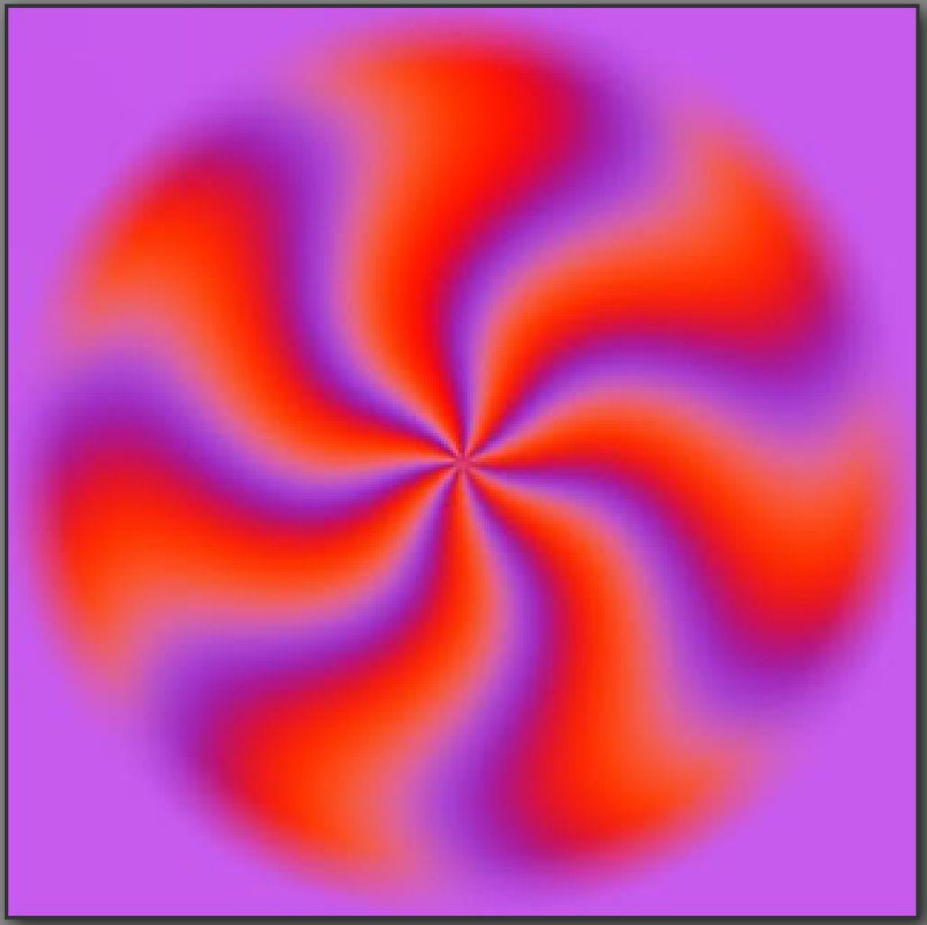
Successive Contrast



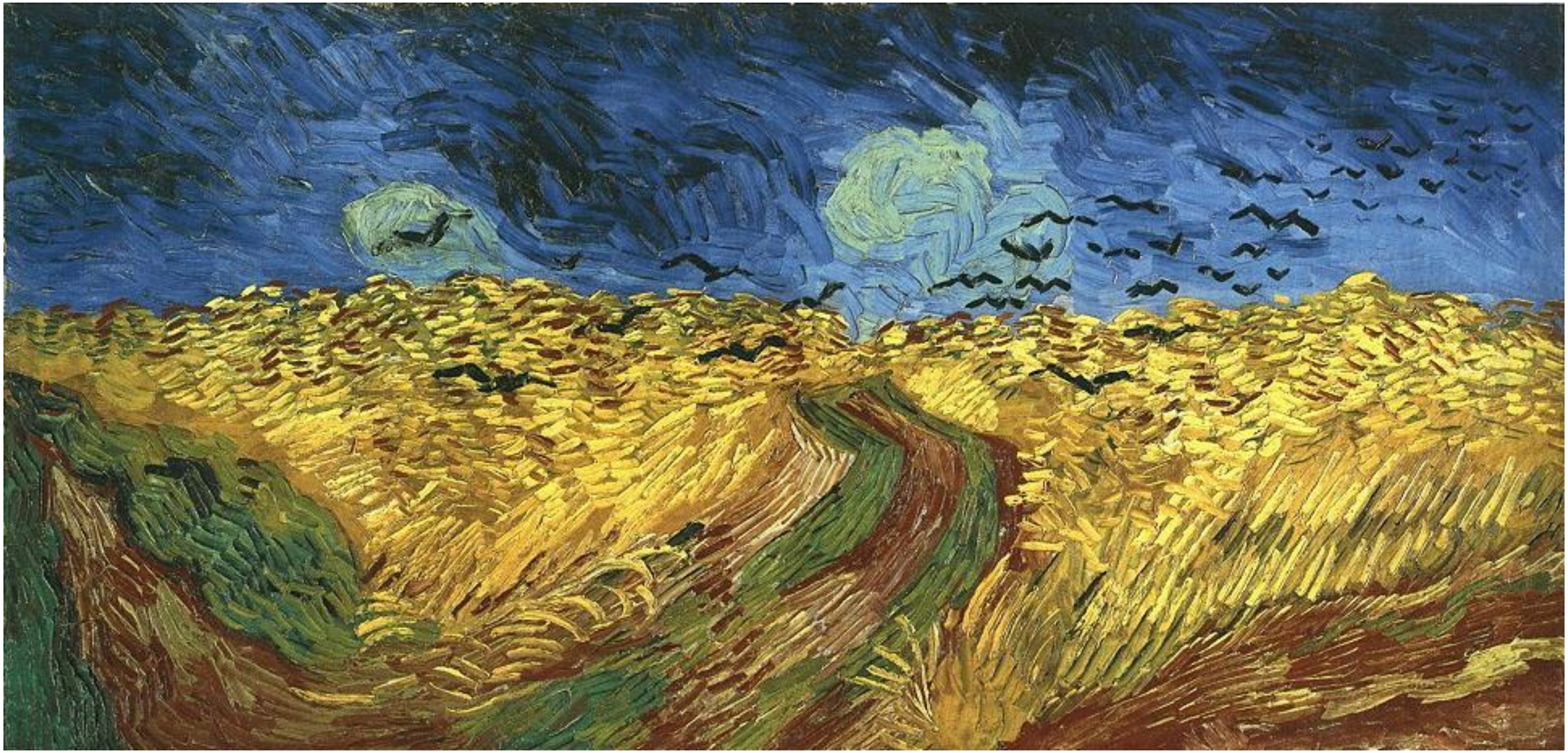
Successive Contrast

Equiluminant Colors

- Strong contrast causes shapes to be seen by color sensitive cell
- Equiluminance hides positions from light sensitive cells
- Flickering/movement caused by this disconnect

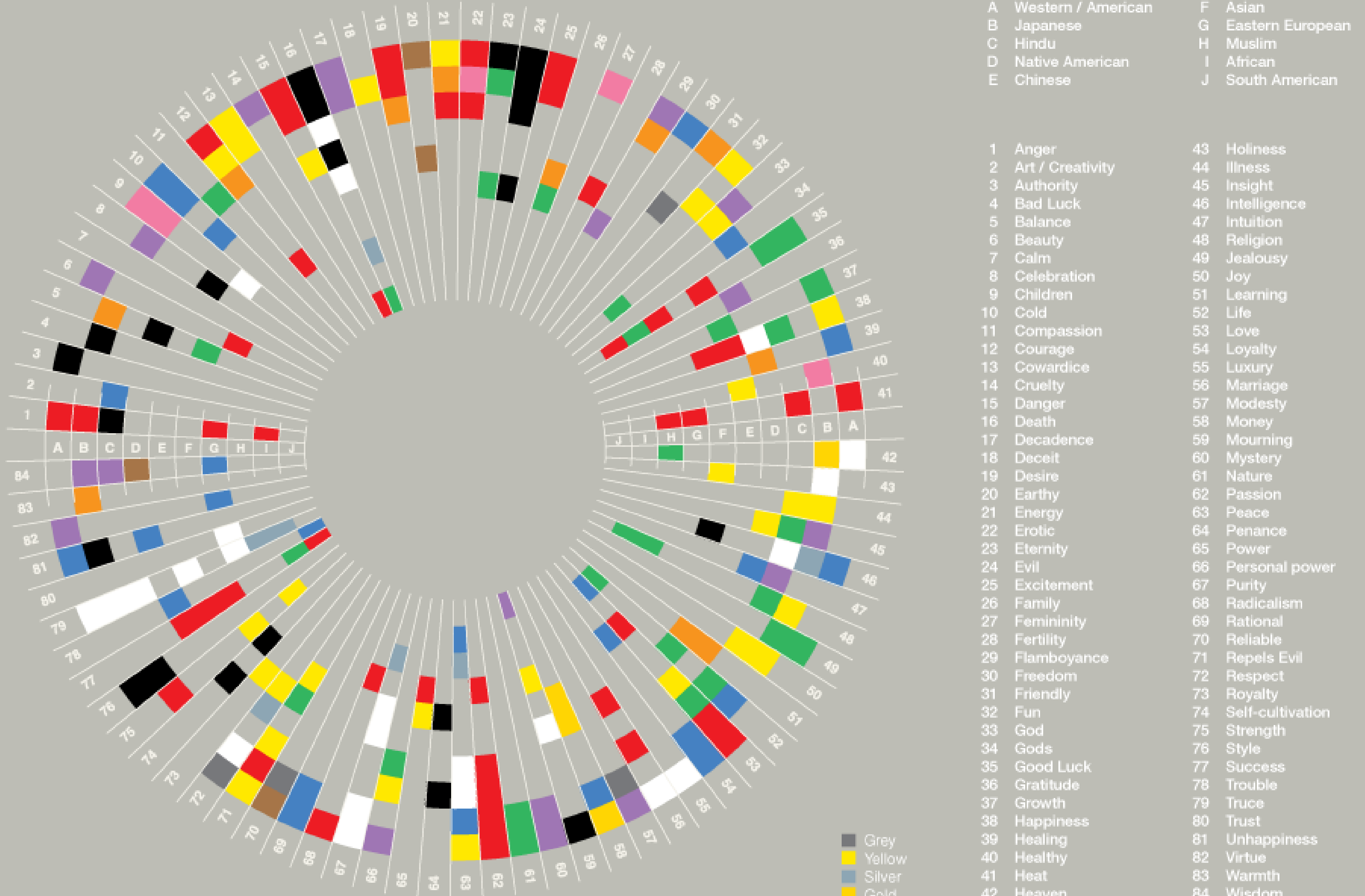


Other perceptual aspects



„This is Van Gogh's last painting before he committed suicide.“

Colours In Culture

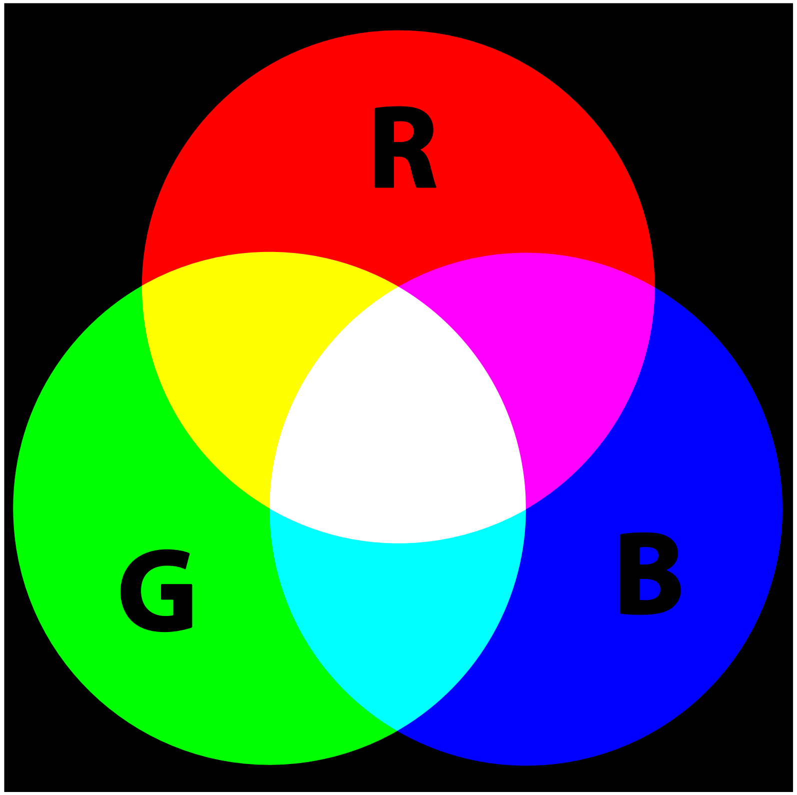


David McCandless & AlwaysWithHonor.com // v1.0 // Apr 09 // InformationIsBeautiful.net

source: Pantone, ColorMatters & web sources

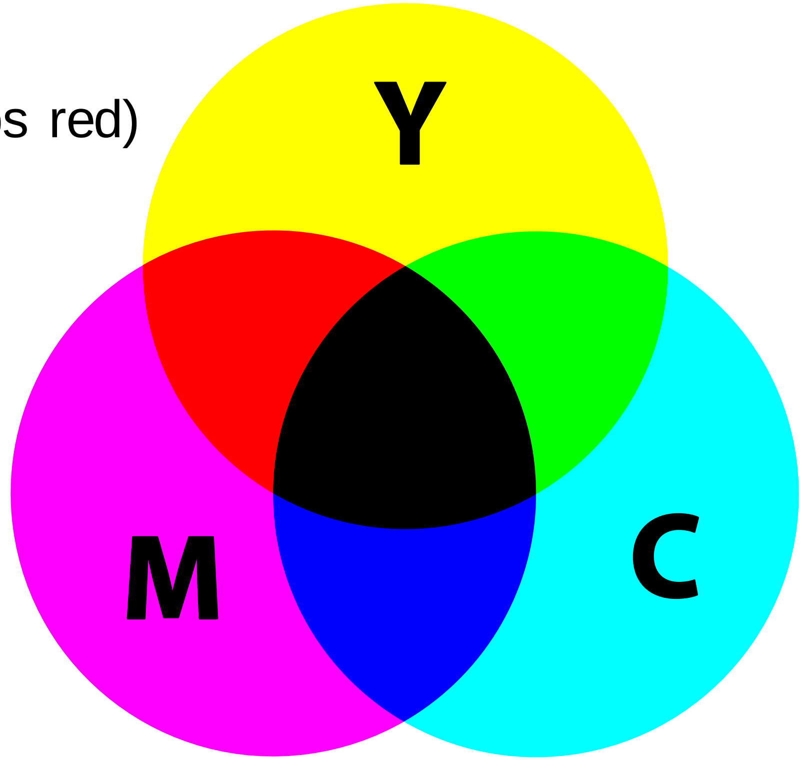
Color Mixing, Color Models, Color Interpolation

- Additive color mixing:
  - Light rays with different spectra of light come together
  - The spectra add up
  - The result is a different spectrum of light, i.e., color.
- Example RGB:
  - Monitors
- Note: Additive color mixing can also be done with other color models, including CMYK



- Subtractive color mixing:
  - A light ray with a (white) spectrum of light hits a surface
  - It is being reflected
  - The surface **absorbs** some wavelengths of light
  - The result is a different spectrum of light, i.e., color.

- Example CMY(K):
  - Cyan: complement of red (= absorbs red)
  - Magenta: complement of green
  - Yellow: complement of blue
  - K = black ink to hide color mixing imperfections
  - Note: Subtractive color mixing can also be done with other color models, including RGB



- Color Models are a way to encode a spectrum of light
  - HSL
  - HSV
  - RGB
  - CMYK
  - many more…

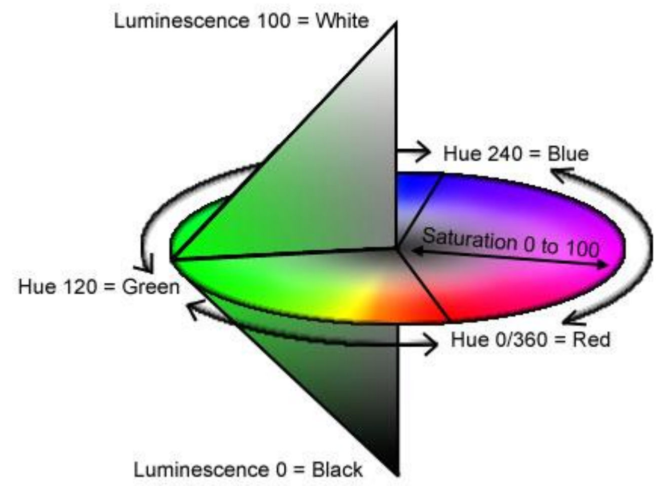
HLS System

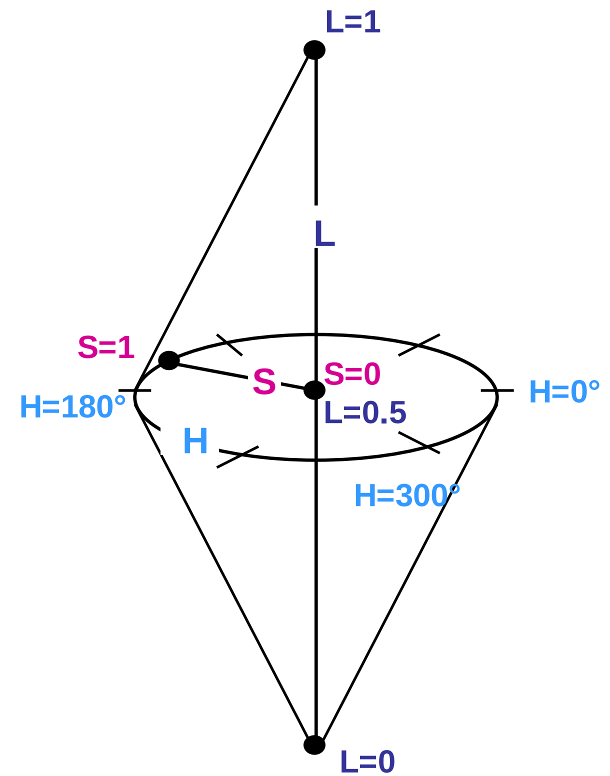
- **H**ue
  classifies a color as red, green, blue, or mixture of these. The hues are given on a circle.

- **L**ightness
  depends on the amount of light

- **S**aturation
  describes the gray portion of the color

- Perception-oriented color system





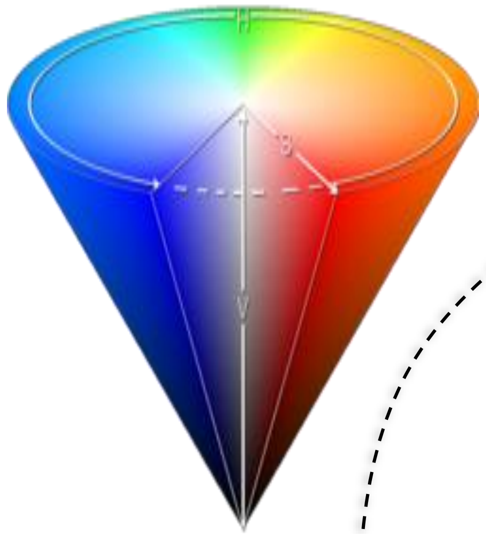
HSV System

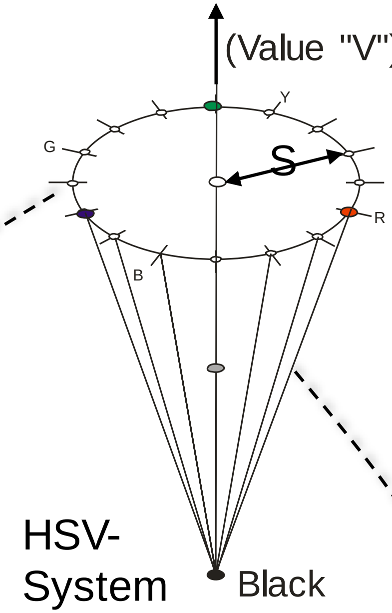
- **H**ue
  classifies a color as red, green, blue, or mixture of these. The hues are given on a circle.

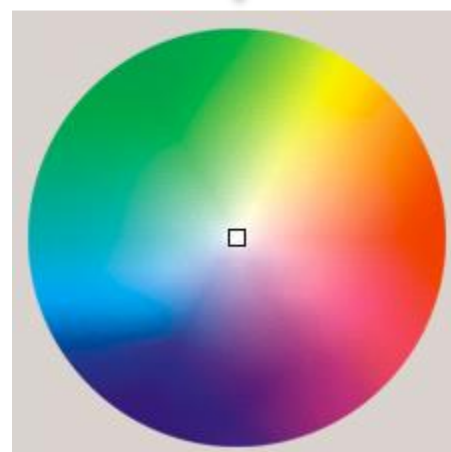
- **S**aturation
  describes the gray portion of the color

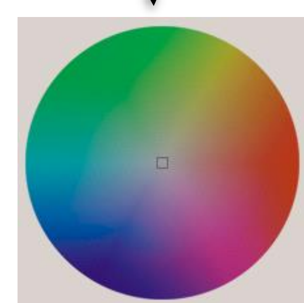
- **V**alue
  depends on the amount of light

- Perception-oriented color system





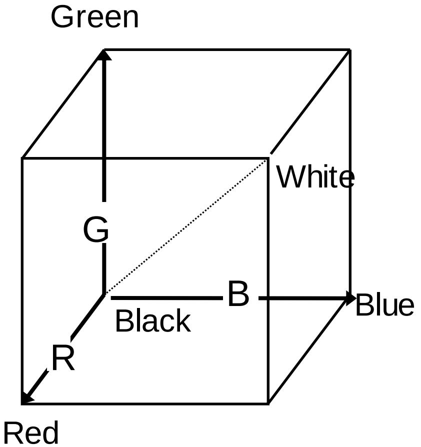




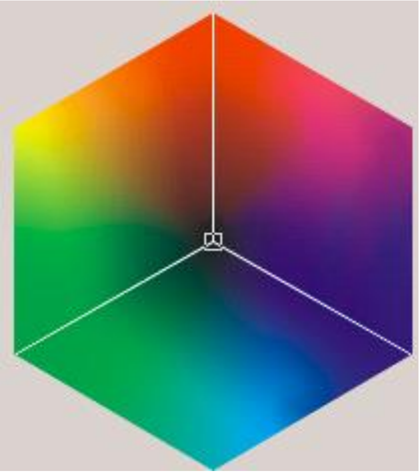
cuts through the HSV cone at v=1 and v=0.5

RGB System

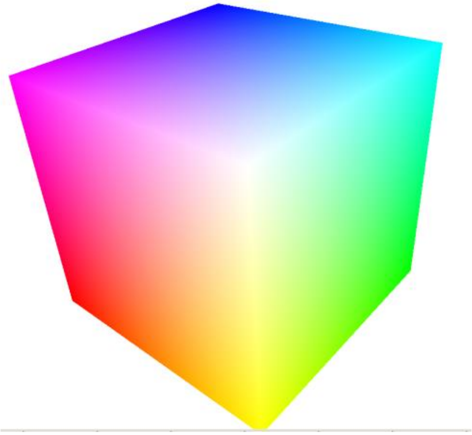
- **R**ed
- **G**reen
- **B**lue

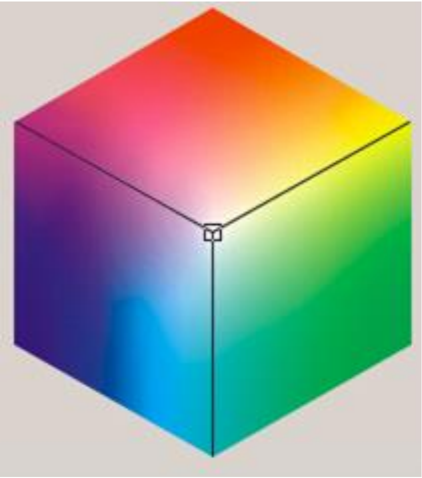


RGB color cube

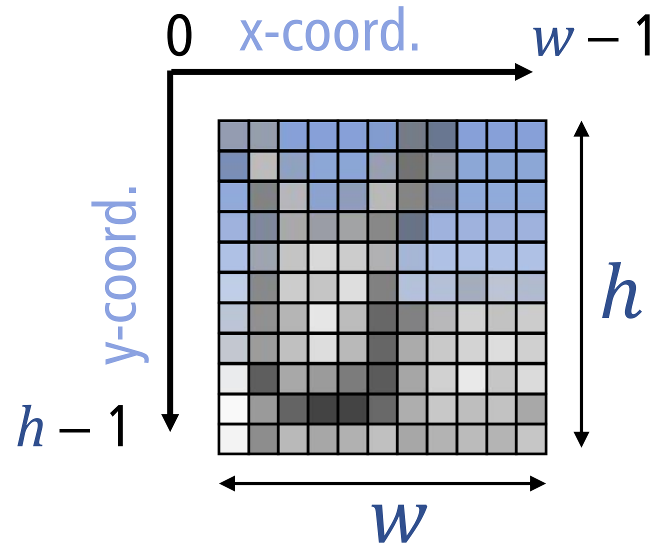


- Technology-oriented color system
- Describes a color by mixing three primary colors





RGB Model

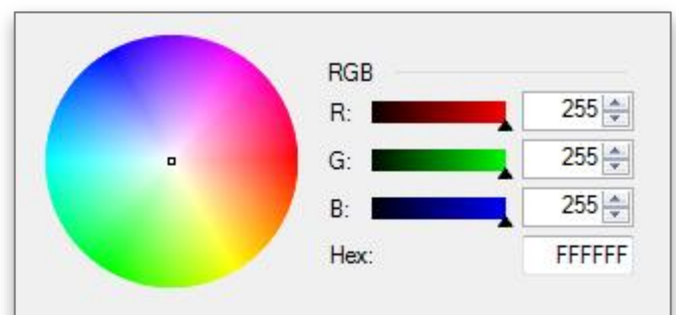


Bitmap (Pixel Display)

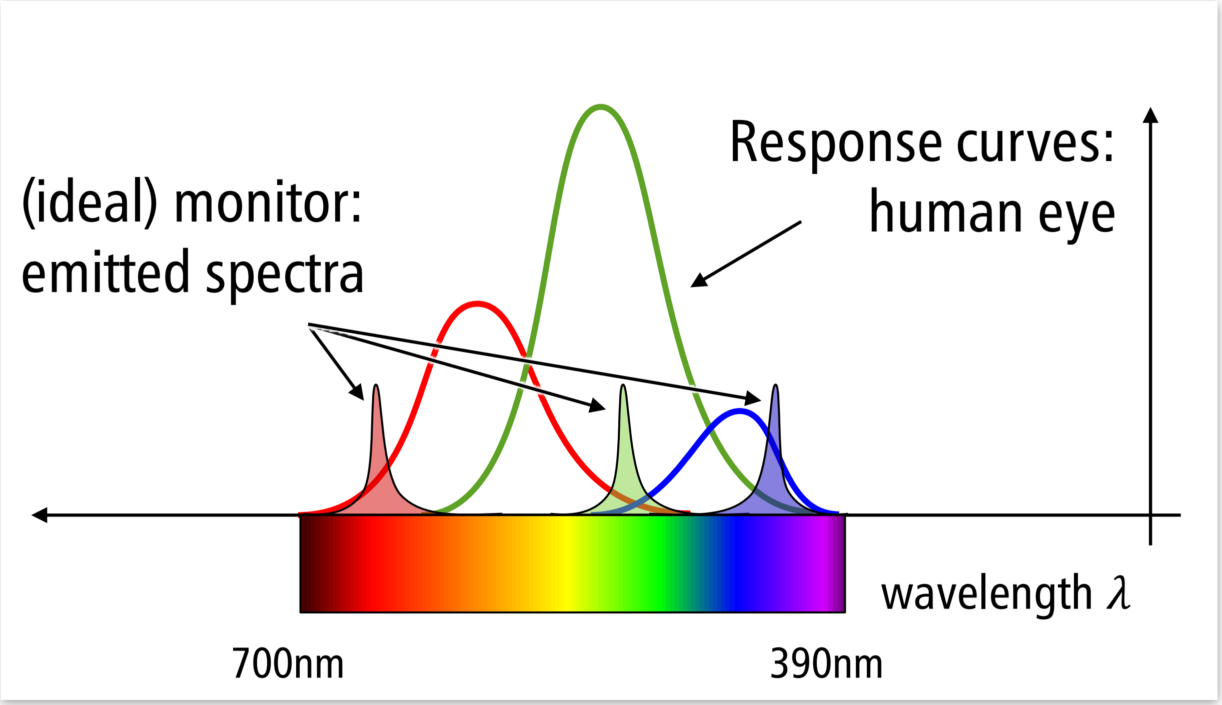
- Screen: $w \cdot h$ discrete pixels
    - Origin: usually upper left
- Varying color per pixel

RGB Model

- Every pixel can emit *red*, *green*, *blue* light
- Intensity range:
    - Usually: bytes 0…255
    - $0 =$ dark
    - $255 =$ maximum brightness



Human Vision



(**curves:** schematic, not accurate)

Create color impressions

- Basis for three-dimensional color space
- Wide spacing, narrow bands: purer colors
  - Otherwise: washed out colors

amounts of *R*, *G*, and *B* to reach the proper lightness or value.[19]

From HSV [edit]

Given a color with hue $H \in [0°, 360°]$, saturation $S_{HSV} \in [0, 1]$, and value $V \in [0, 1]$, we first find chroma:

$$C = V \times S_{HSV}$$

Then we can find a point (R_1, G_1, B_1) along the bottom three faces of the RGB cube, with the same hue and chroma as our color (using the intermediate value *X* for the second largest component of this color):

$$H' = \frac{H}{60°}$$

$$X = C(1 - |H' \bmod 2 - 1|)$$

$$(R_1, G_1, B_1) = \begin{cases} (0,0,0) & \text{if } H \text{ is undefined} \\ (C,X,0) & \text{if } 0 \leq H' < 1 \\ (X,C,0) & \text{if } 1 \leq H' < 2 \\ (0,C,X) & \text{if } 2 \leq H' < 3 \\ (0,X,C) & \text{if } 3 \leq H' < 4 \\ (X,0,C) & \text{if } 4 \leq H' < 5 \\ (C,0,X) & \text{if } 5 \leq H' < 6 \end{cases}$$

Finally, we can find *R*, *G*, and *B* by adding the same amount to each component, to match value:

$$m = V - C$$

$$(R, G, B) = (R_1 + m, G_1 + m, B_1 + m)$$

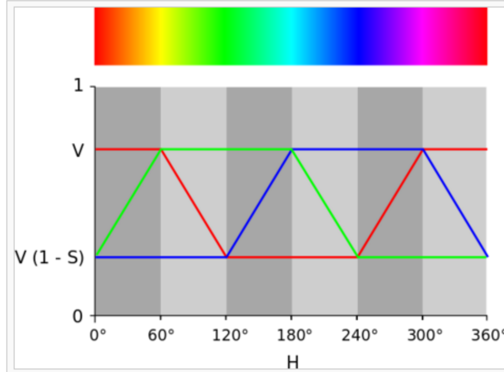


Fig. 24. A graphical representation of RGB coordinates given values for HSV.

From HSL [edit]

- When using a specific color model, we can interpolate between two colors by treating them like vectors and using linear interpolation.
  Example:

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = (1-t)\begin{pmatrix} R_1 \\ G_1 \\ B_1 \end{pmatrix} + t\begin{pmatrix} R_2 \\ G_2 \\ B_2 \end{pmatrix}$$

- It is often perceptually better, to interpolate in the HSV or other perception-based models!
  Example of *changing the saturation:*

$$\begin{pmatrix} H \\ S \\ V \end{pmatrix} = \begin{pmatrix} H \\ (1-t)S_1 + tS_2 \\ V \end{pmatrix}$$

Transparency

Transparency

- "Alpha-blending"
- α = "opacity"
- Color + opacity: RGBα

Blending

- Mix in α of front color, keep $1-\alpha$ of back color

$$\mathbf{c} = \alpha \cdot \mathbf{c}_{front} + (1-\alpha) \cdot \mathbf{c}_{back}$$

- Not commutative! (order matters)
  - unless monochrome