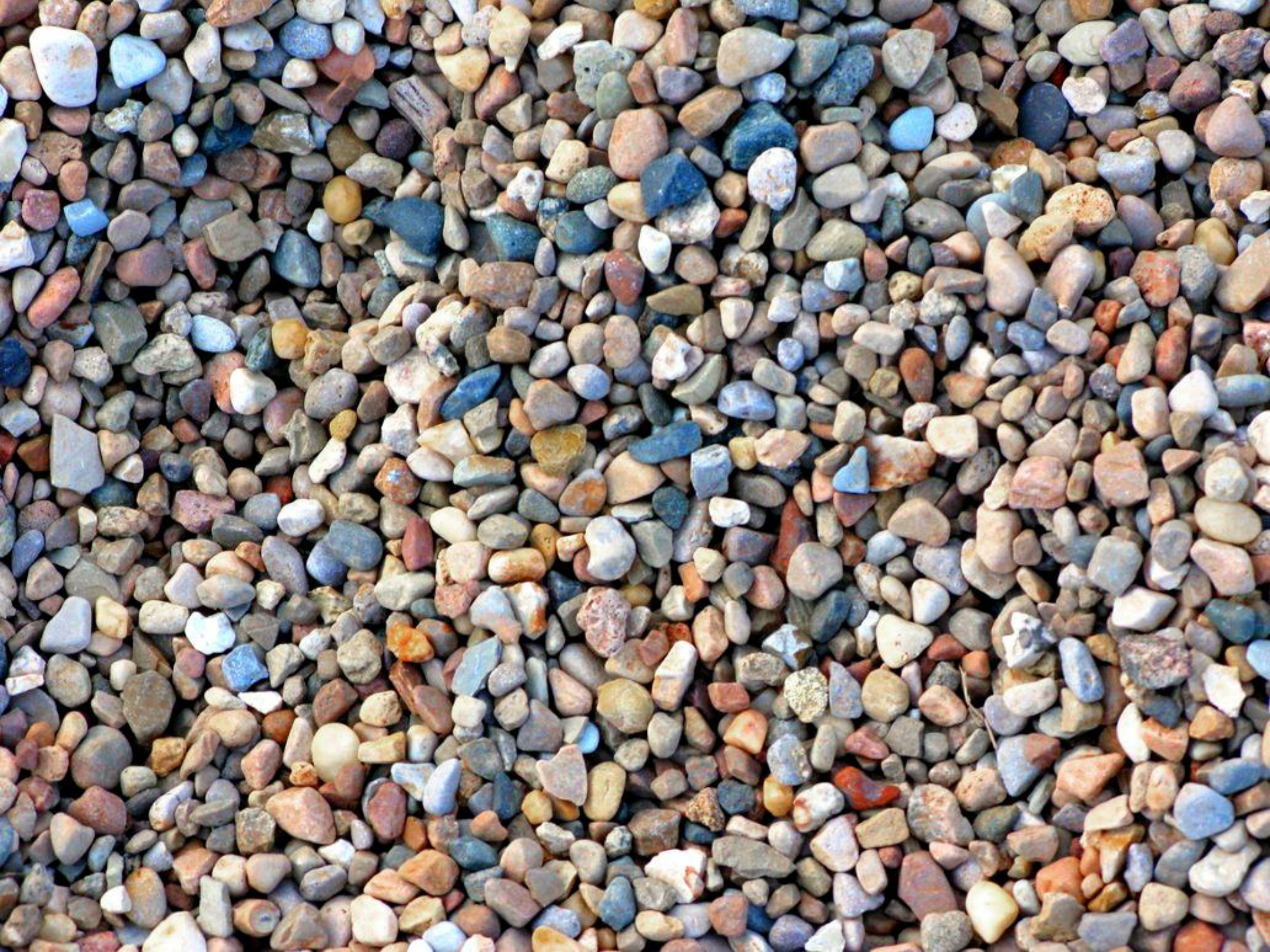
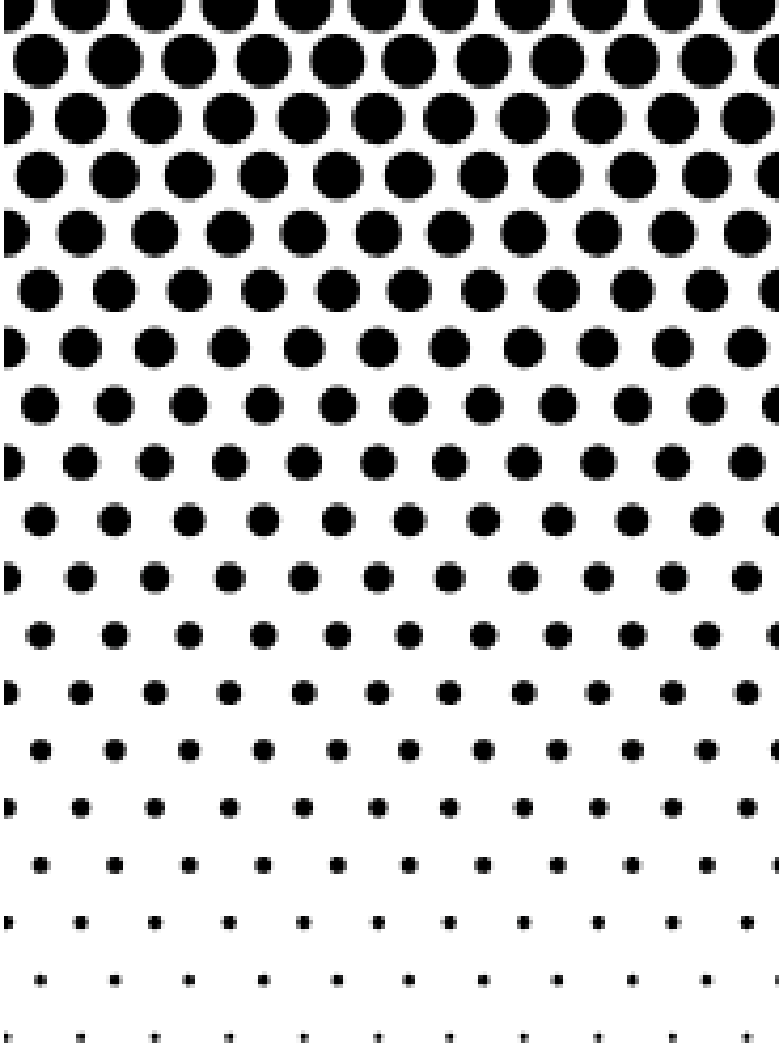
Двухмерная графика ИЗНУТРИ

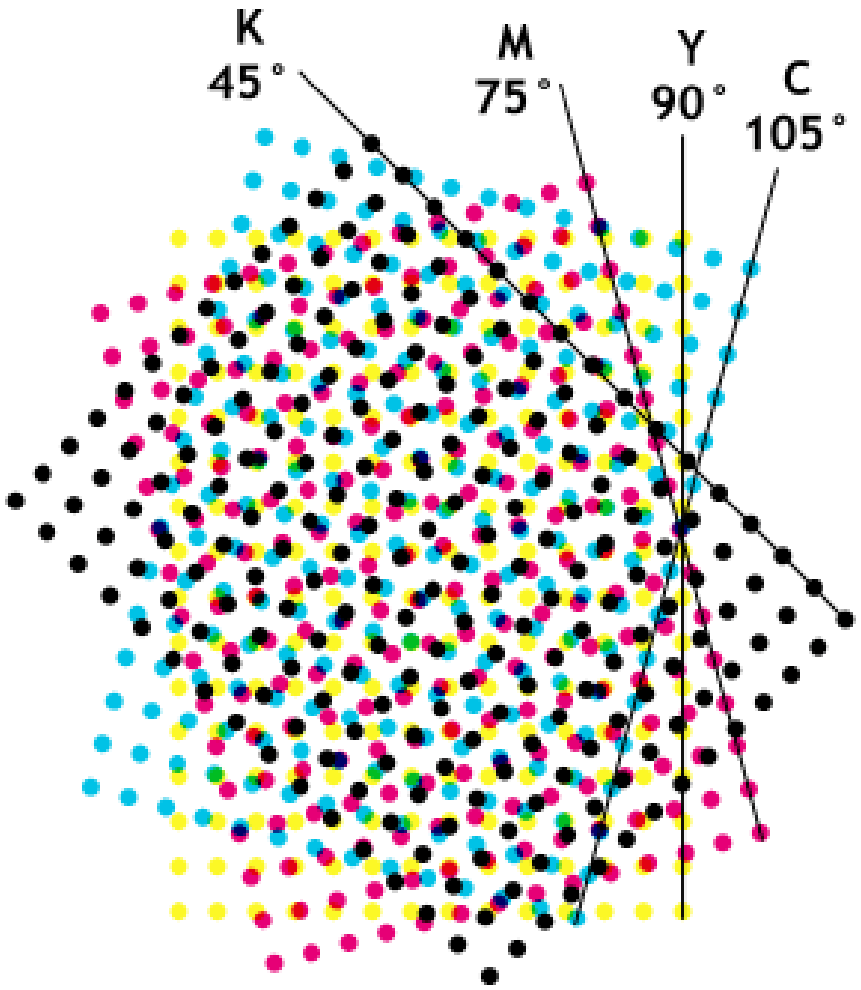
Александр Бурт



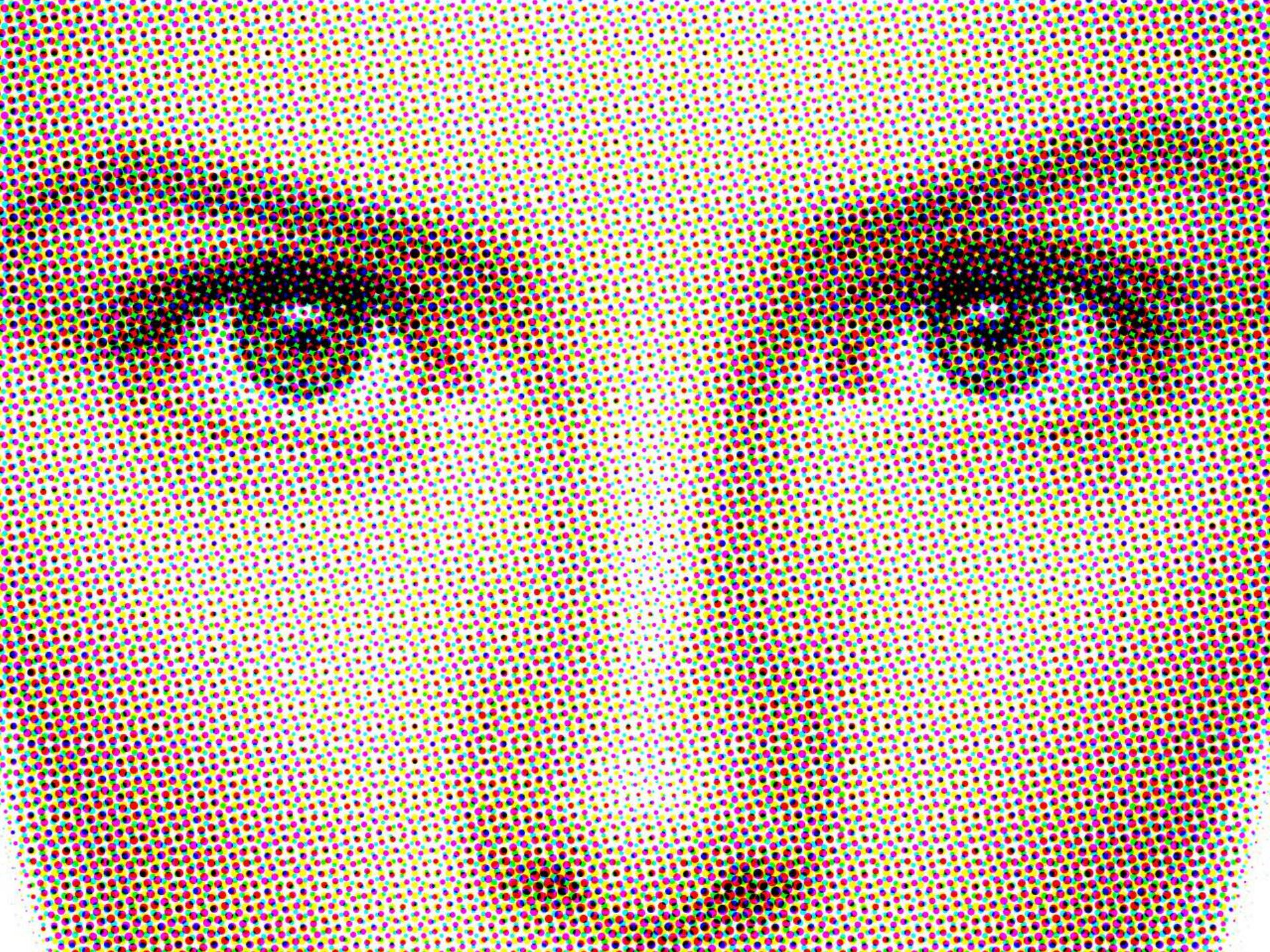

ПЕЧАТЬ

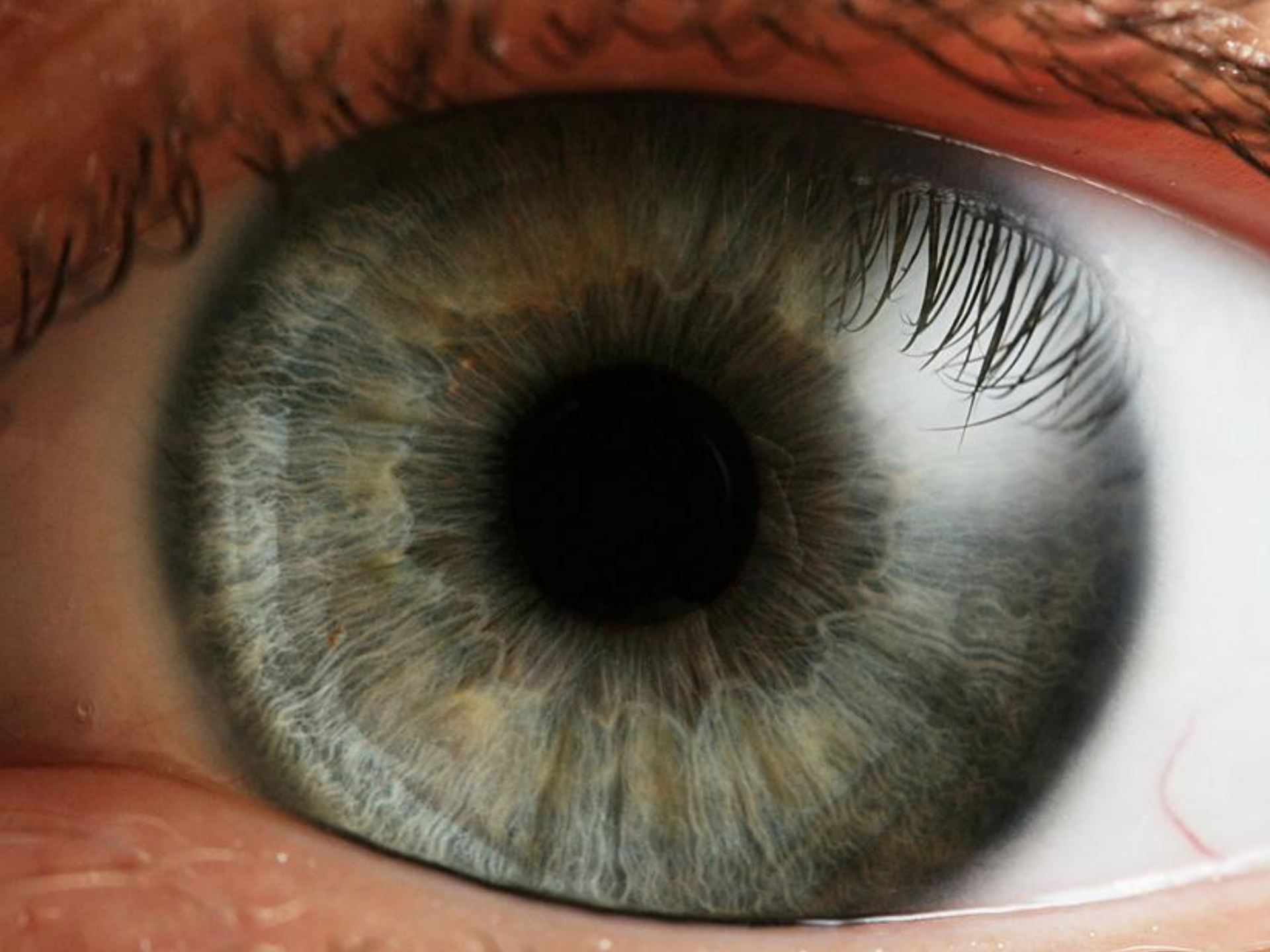


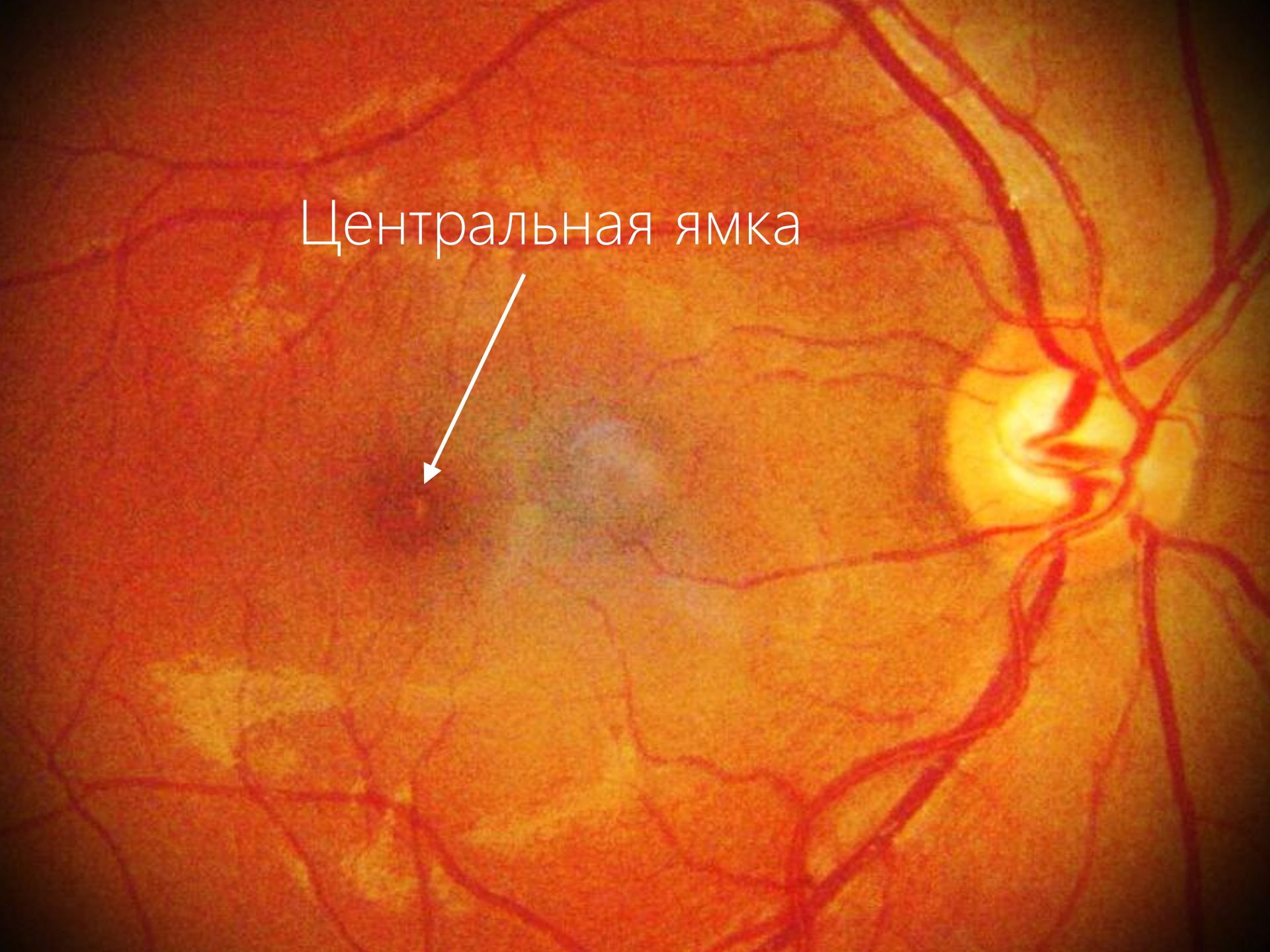
Полутоновый экран

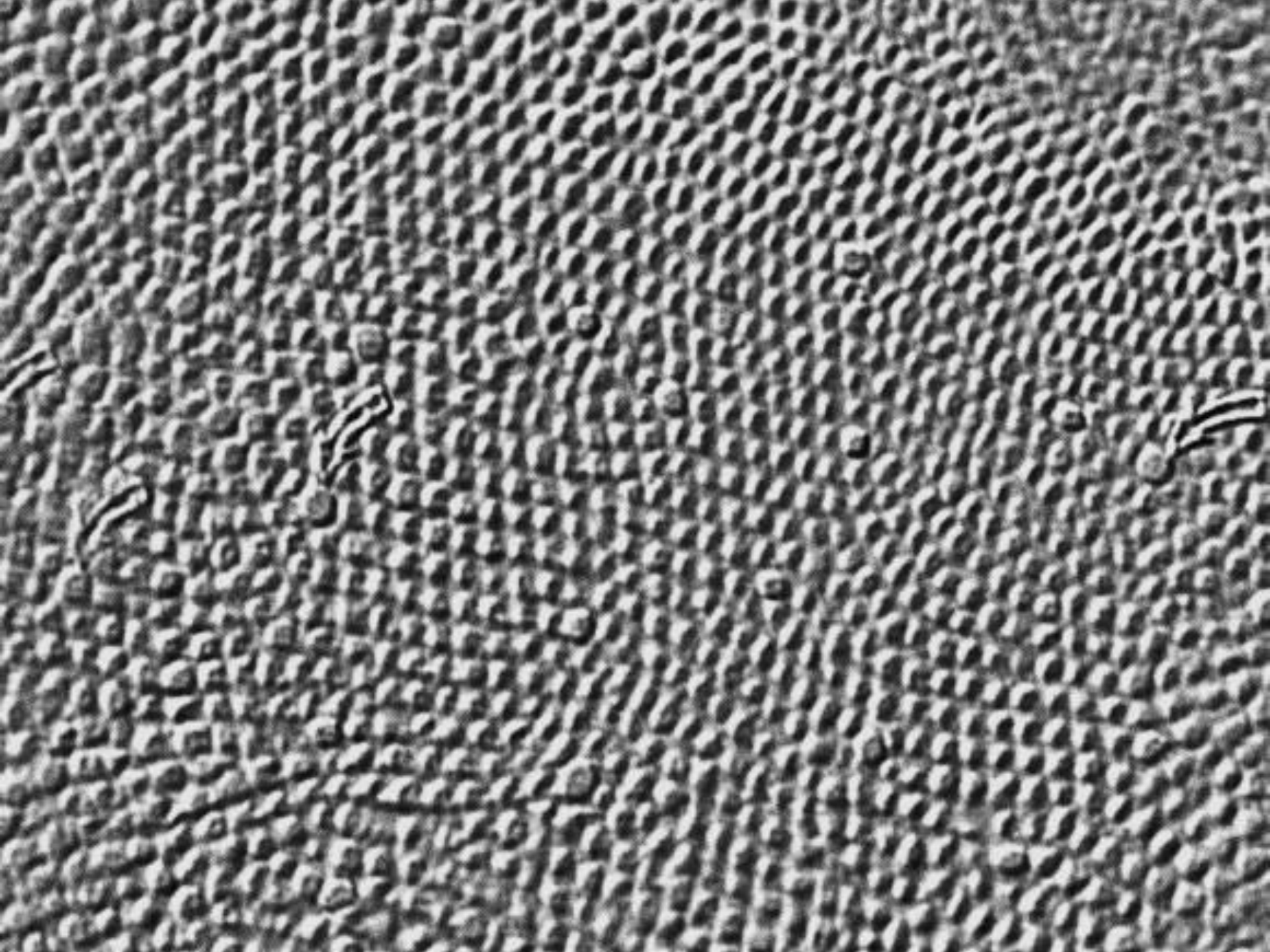


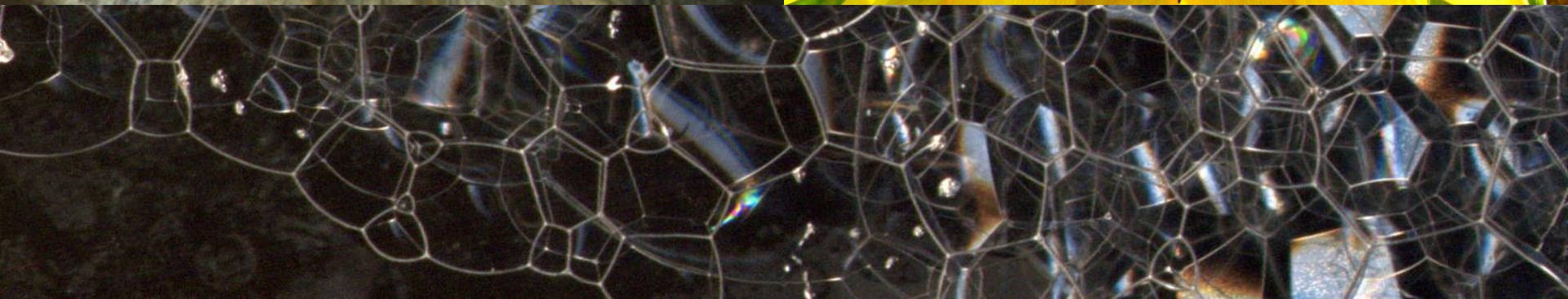
Углы экранов CMYK

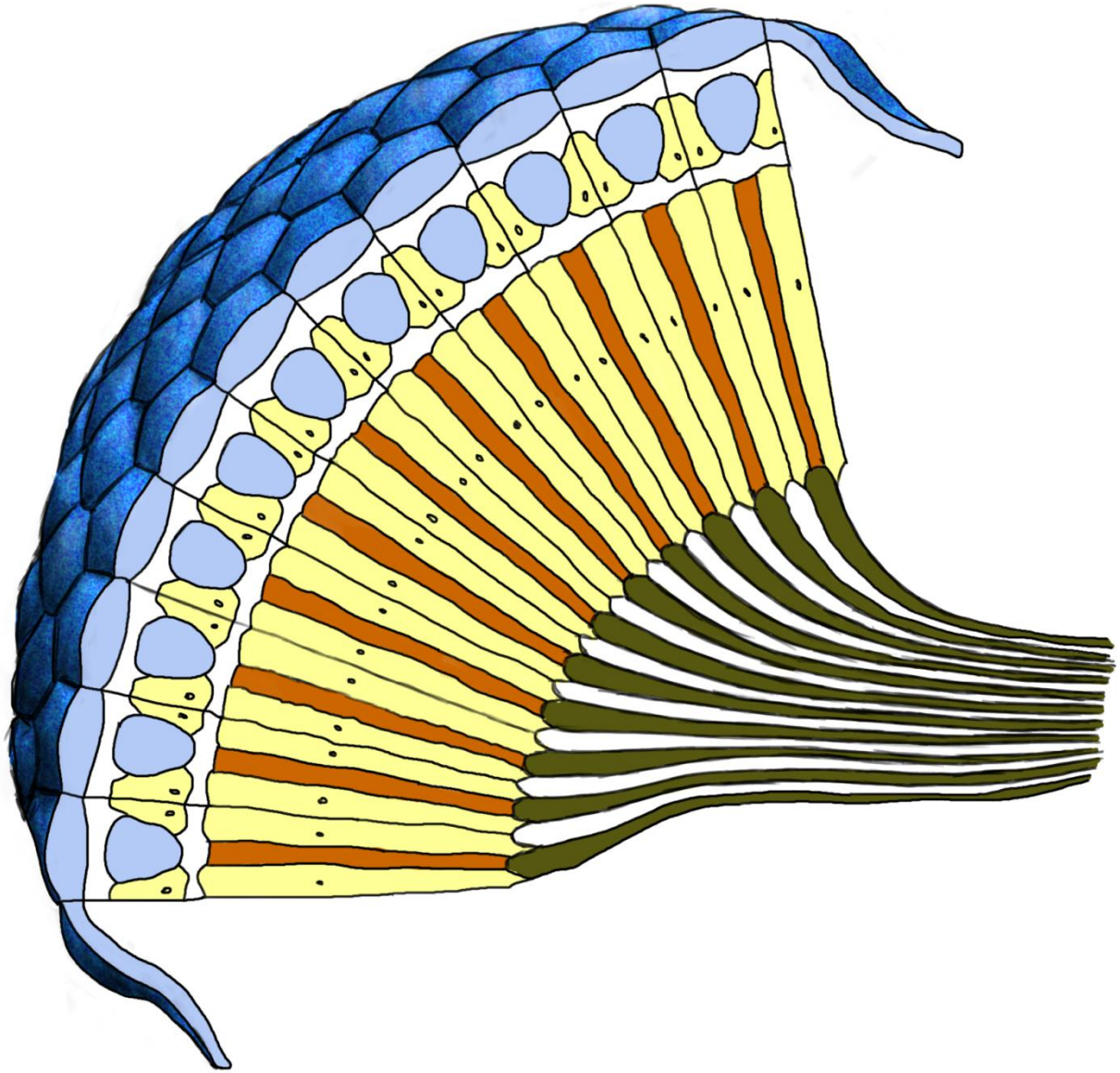


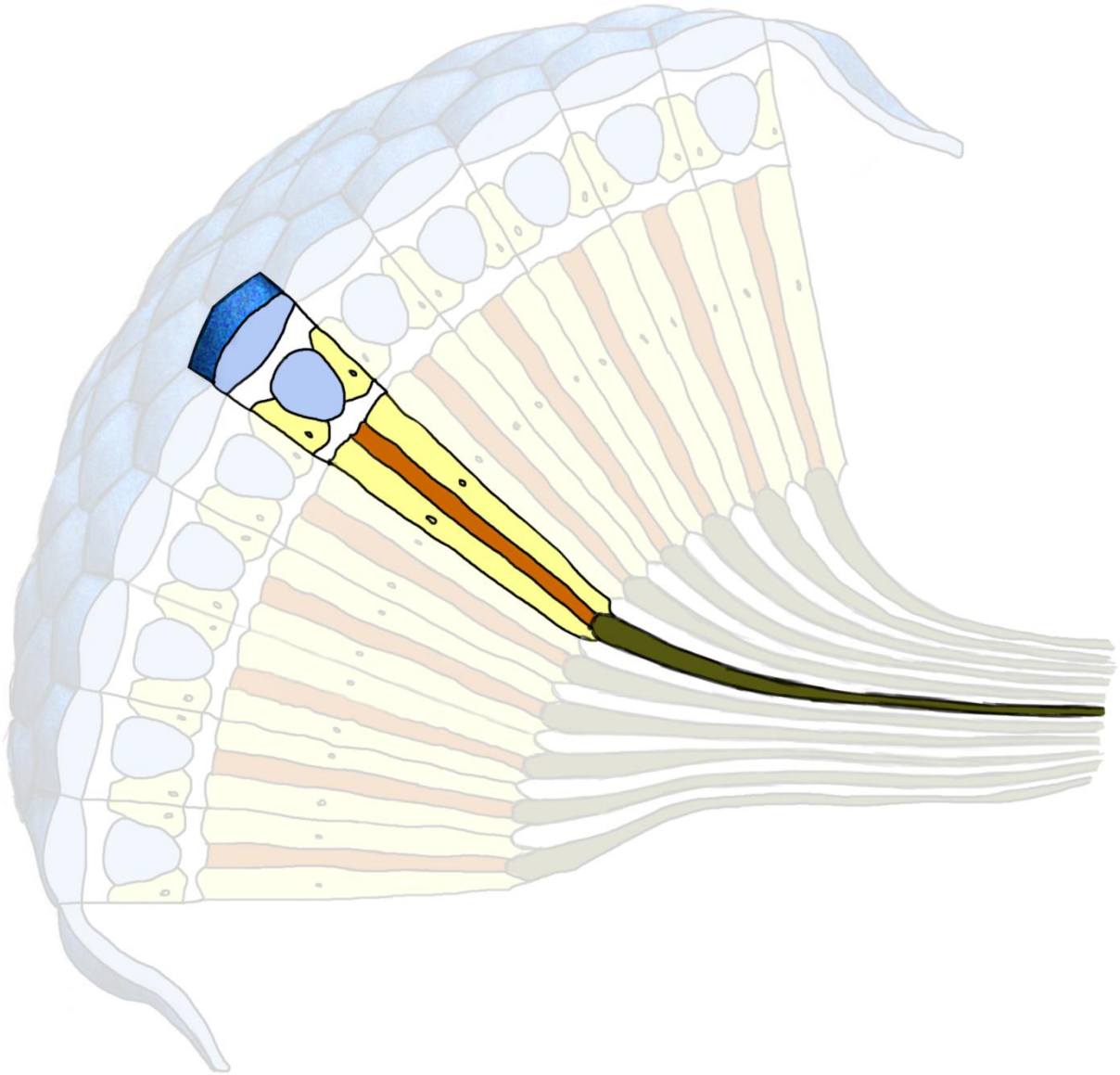


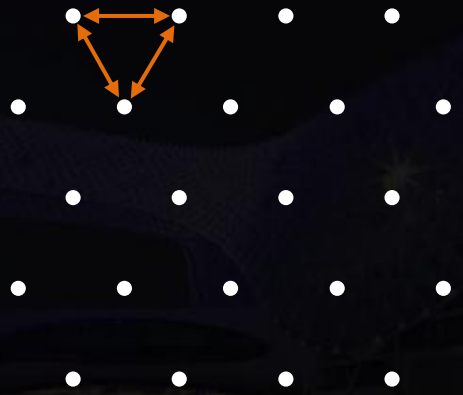

Центральная ямка

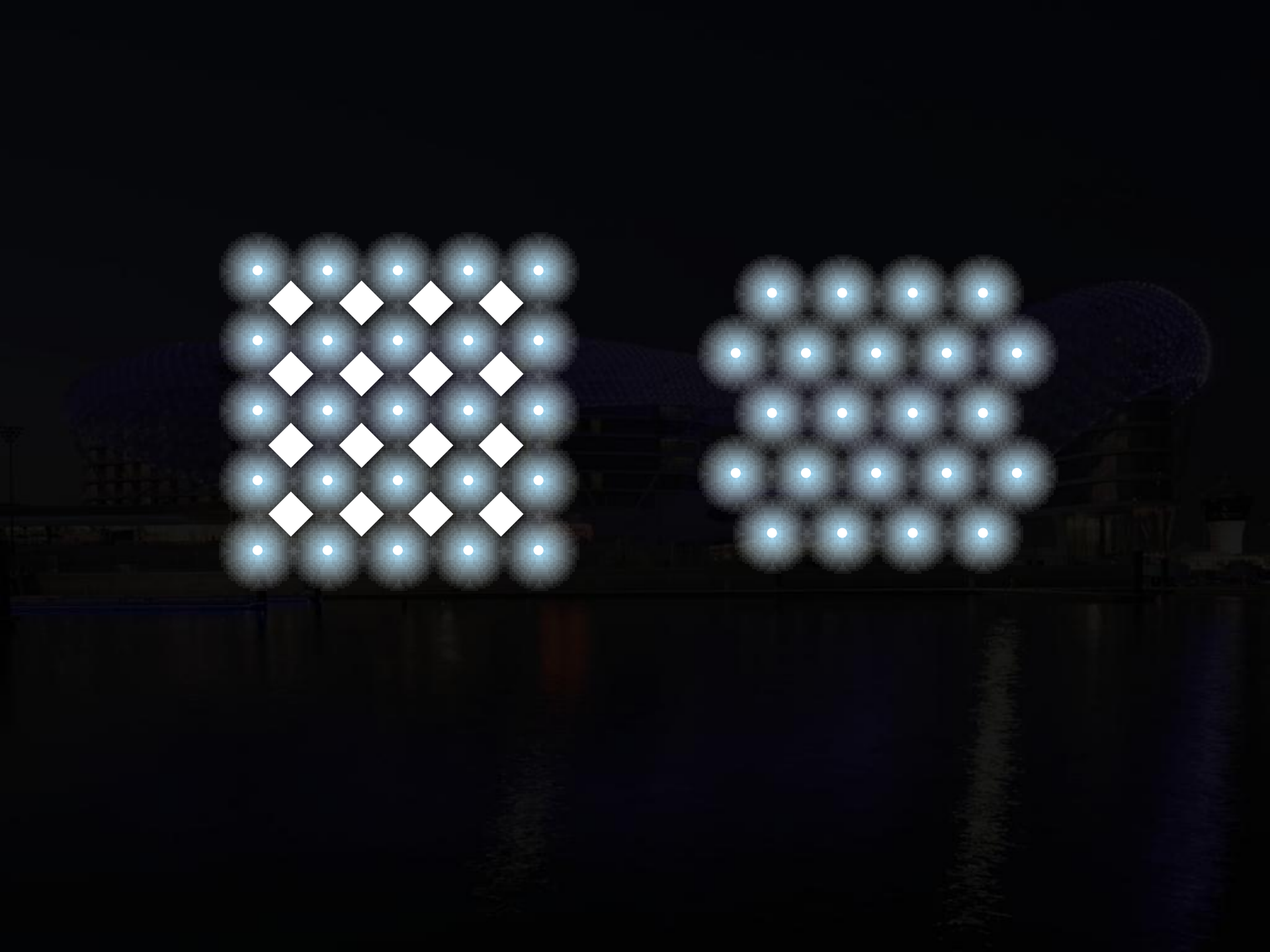


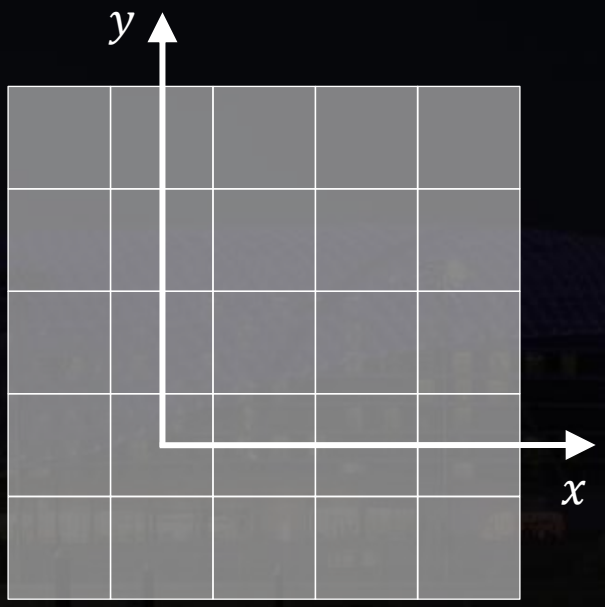


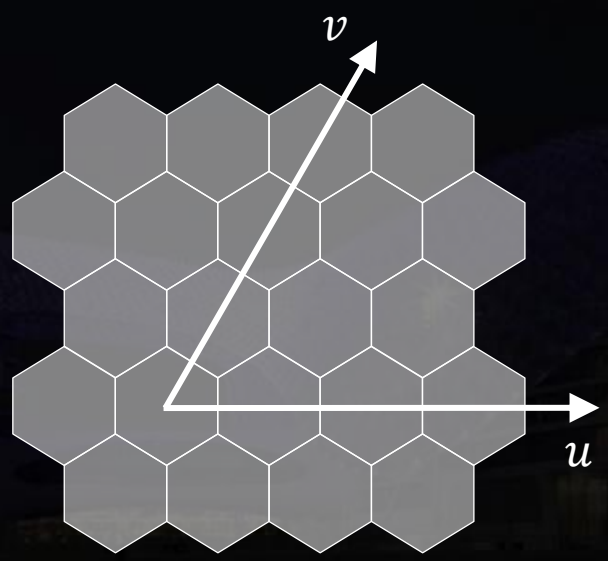










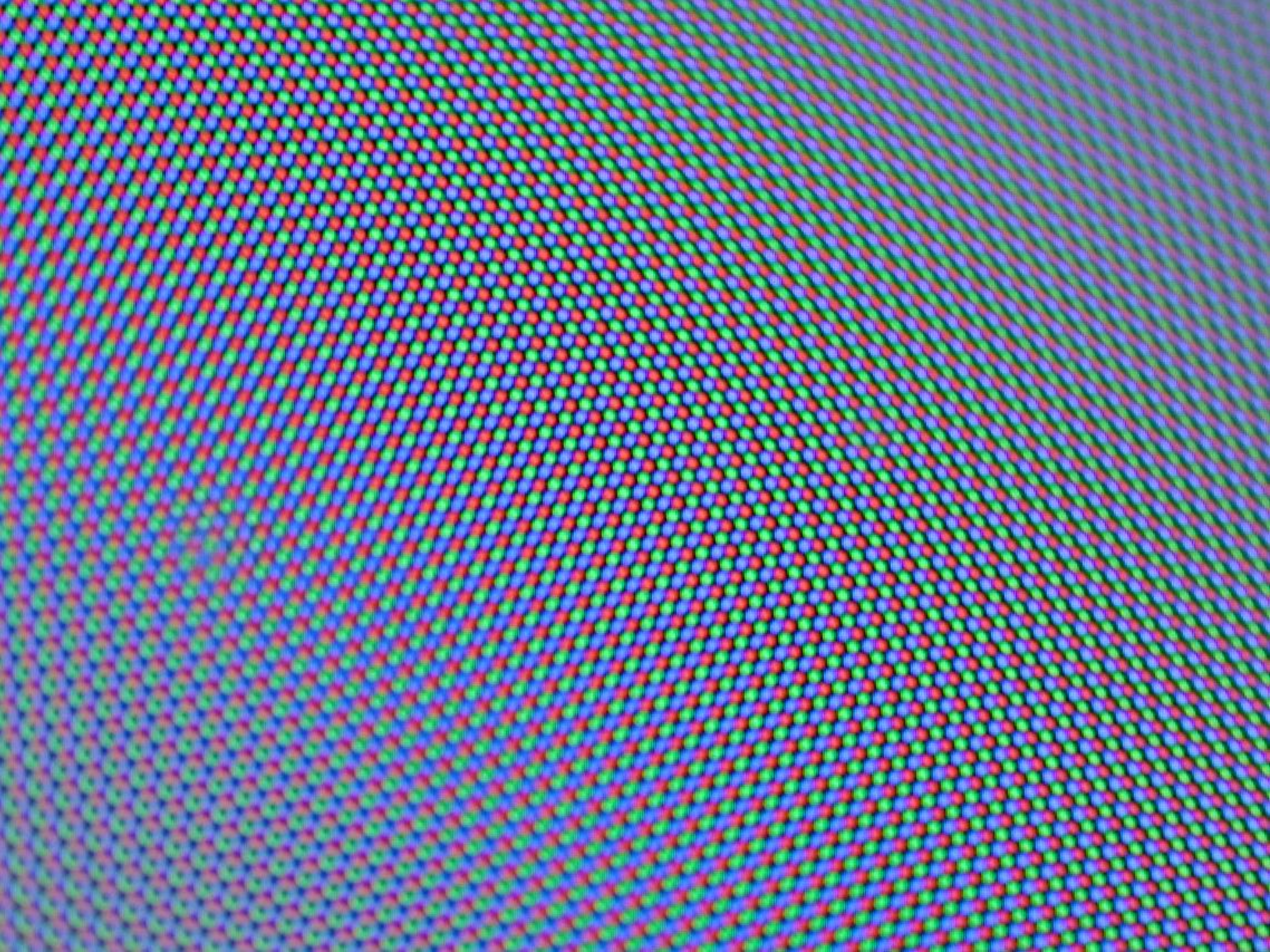


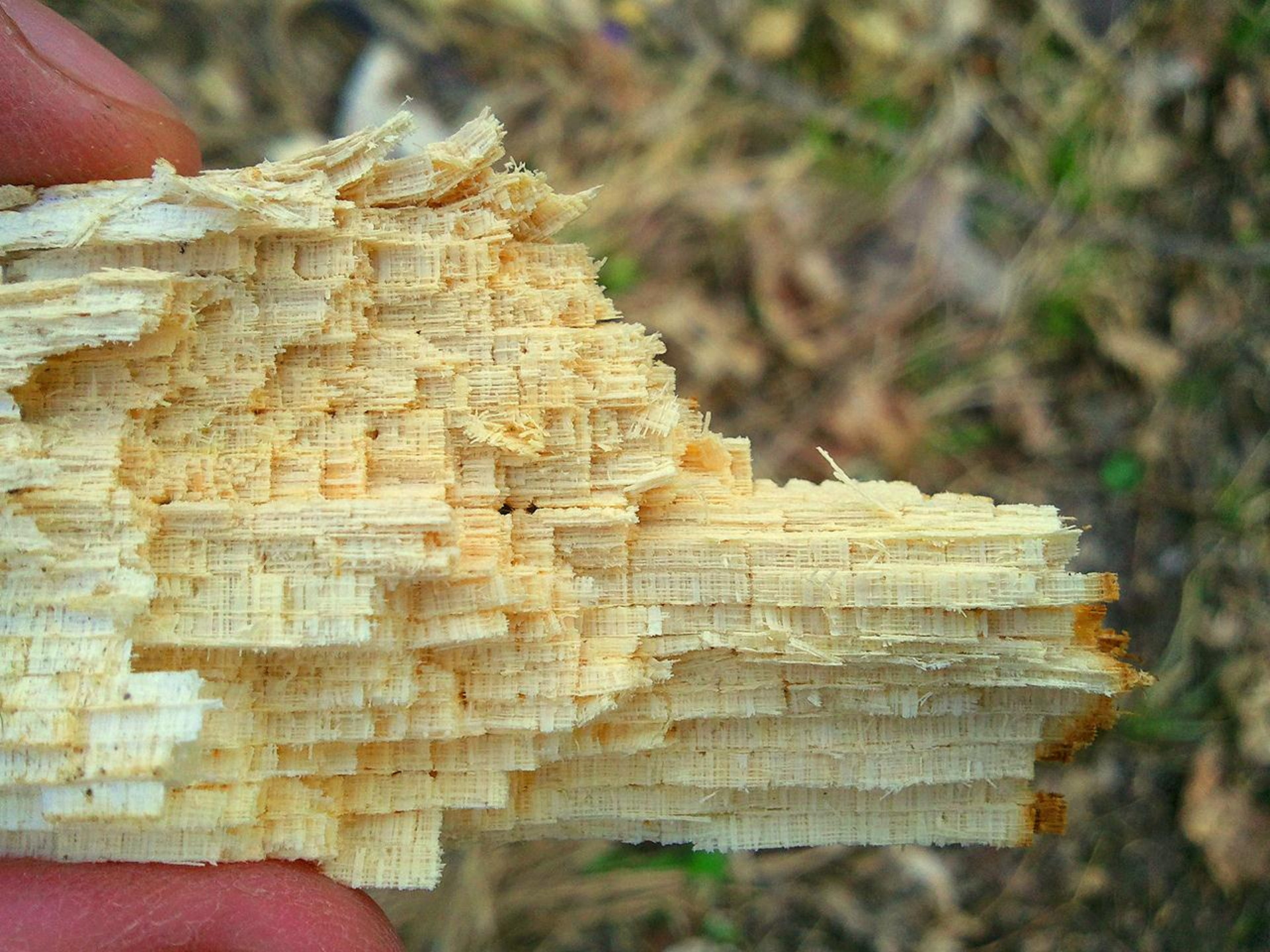
$$x = \mathrm{u} + \frac{v}{2}$$

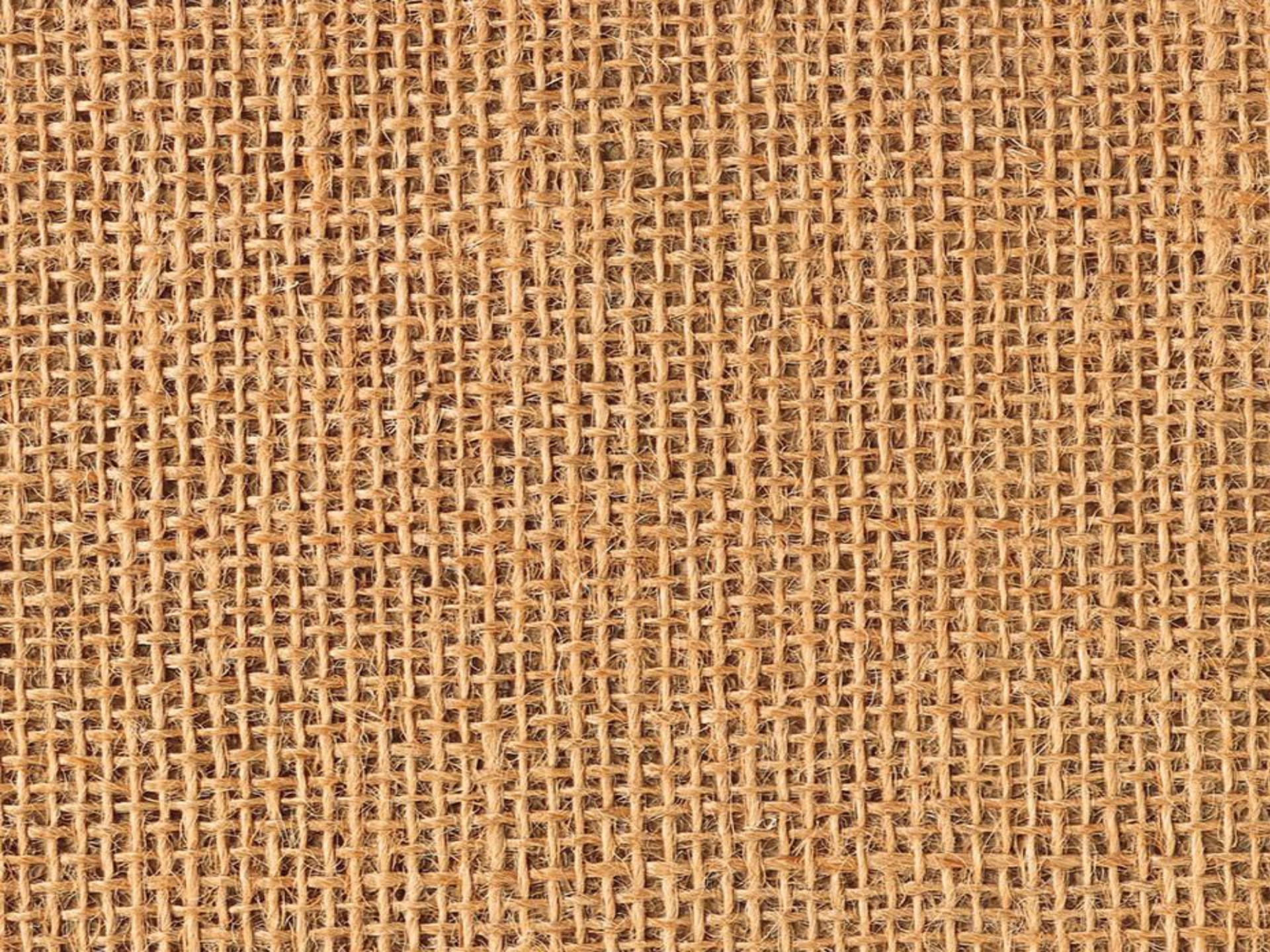
$$y = \frac{\sqrt{3}v}{2}$$

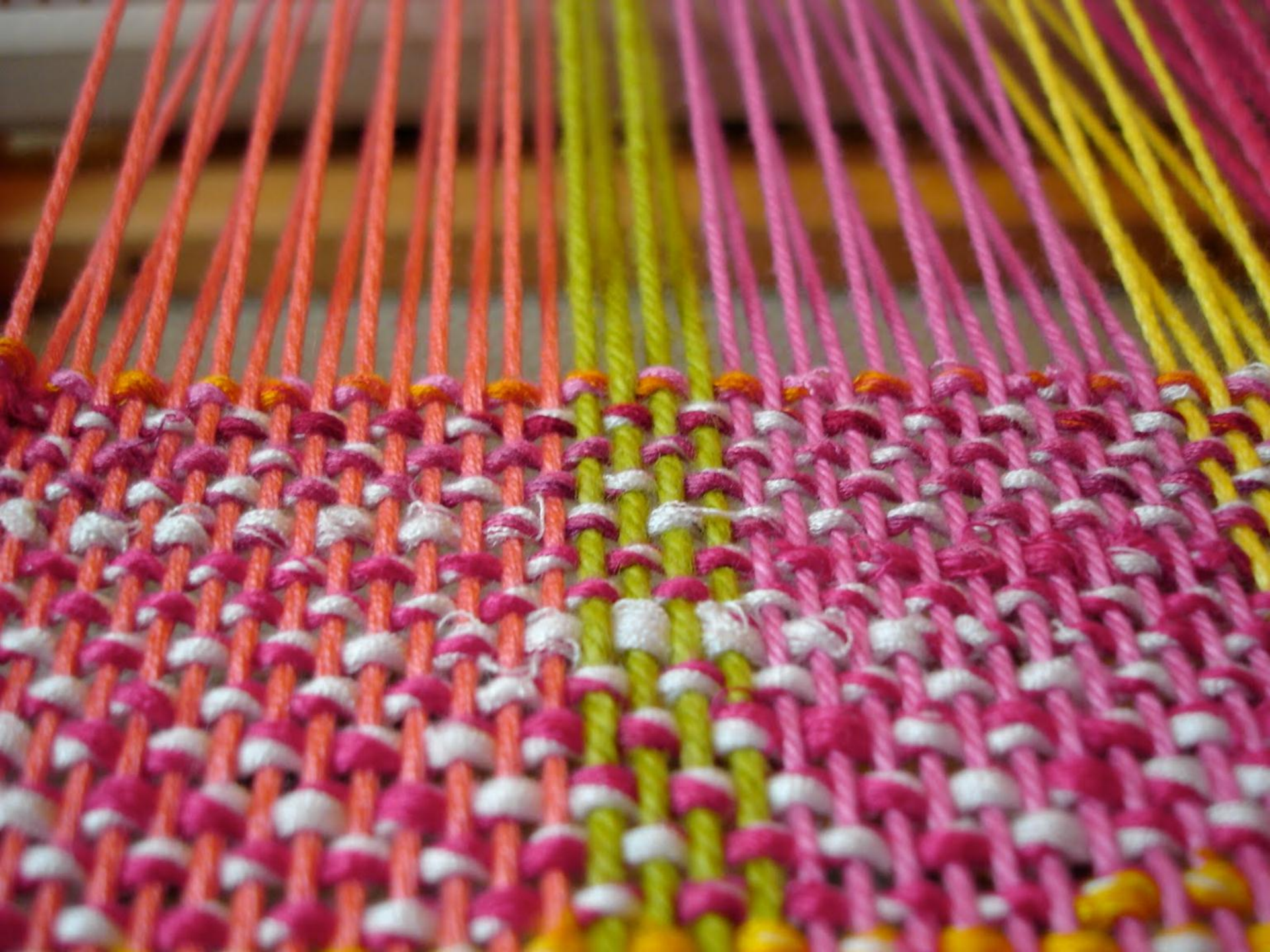
$$u = \mathrm{x} - \frac{y}{\sqrt{3}}$$

$$v = 2\frac{y}{\sqrt{3}}$$



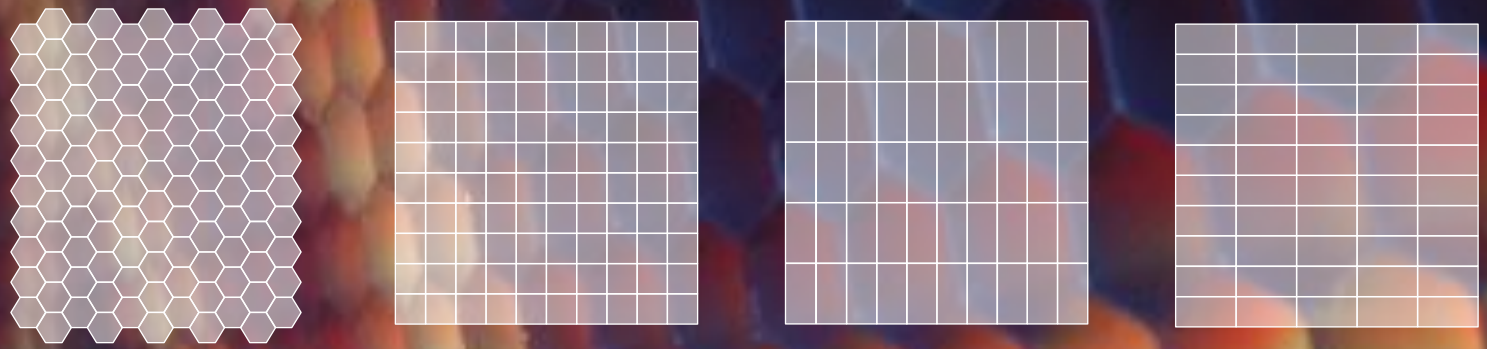




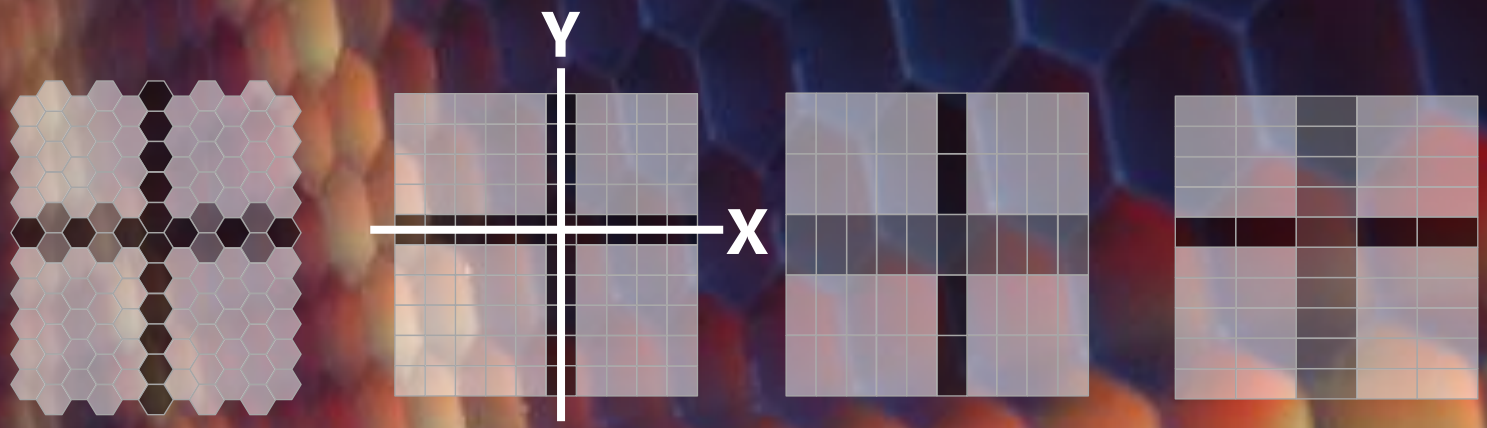


Формы пикселей

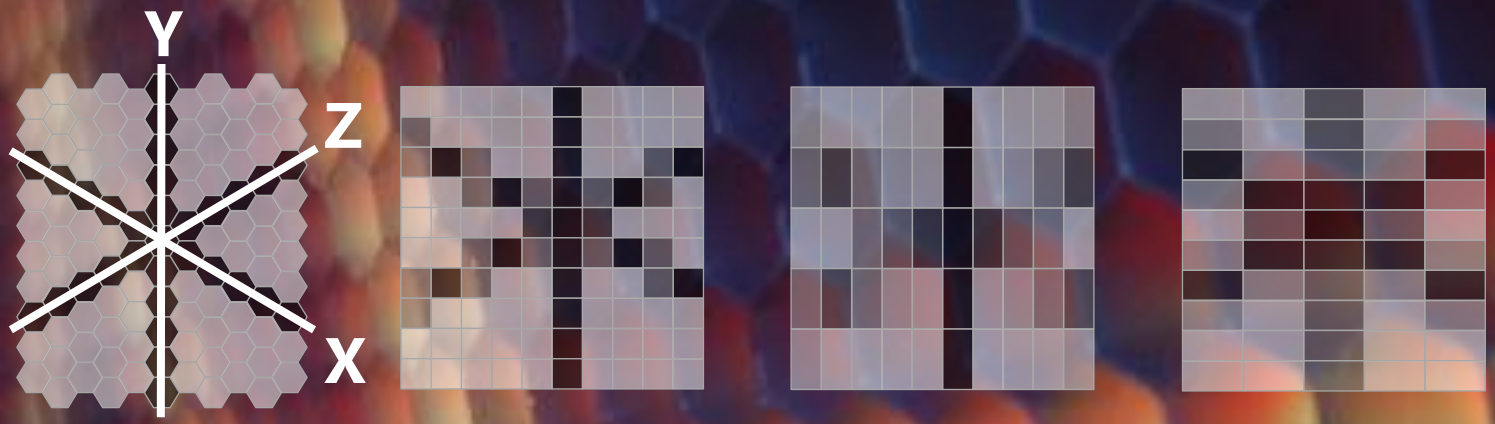
Плюсы и минусы



Плюсы и минусы

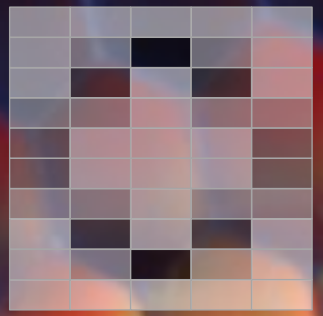


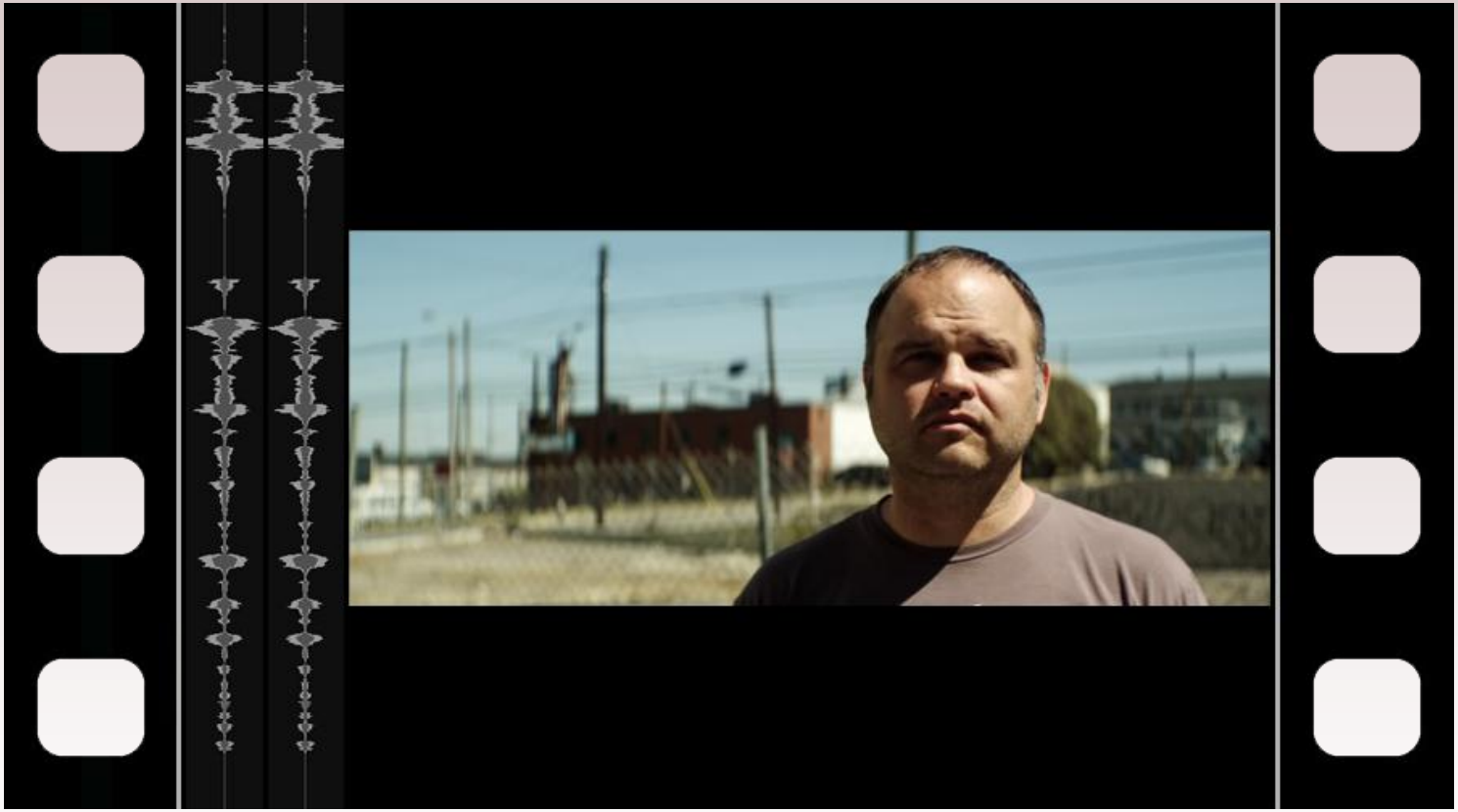
Плюсы и минусы



Плюсы и минусы

Прямоугольные
ПИКСЕЛИ

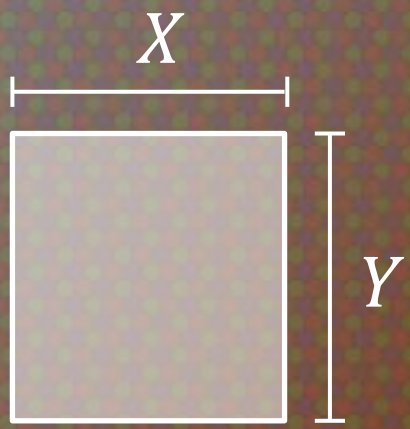





F=50 1:2
ЛОМО


БАС37-22-1
2/22
820001

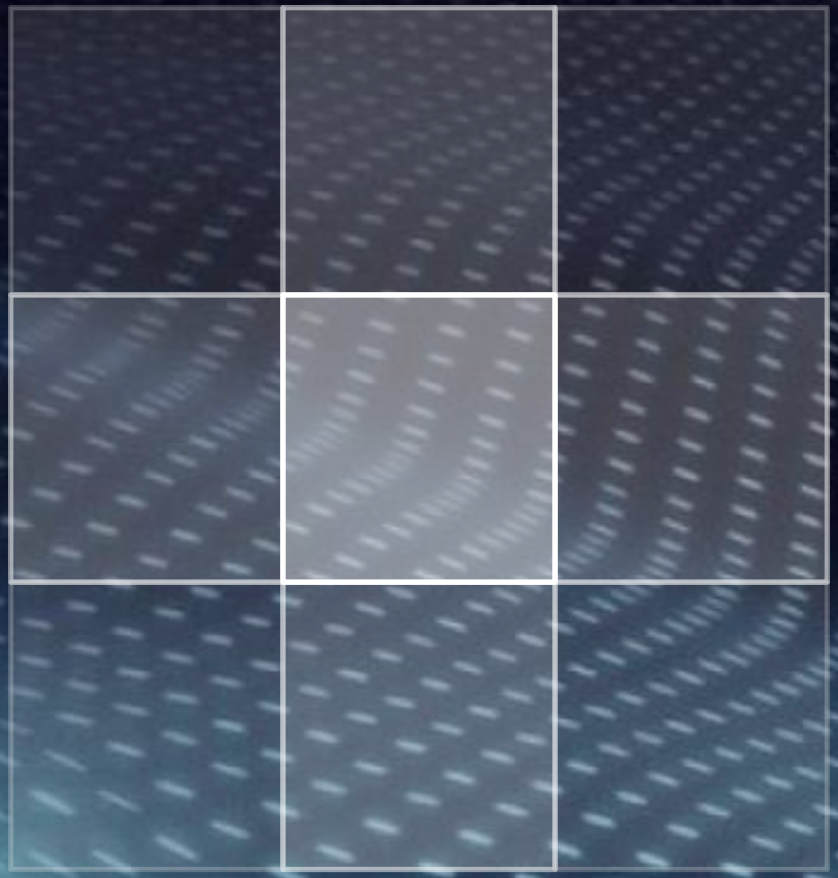
Соотношение сторон

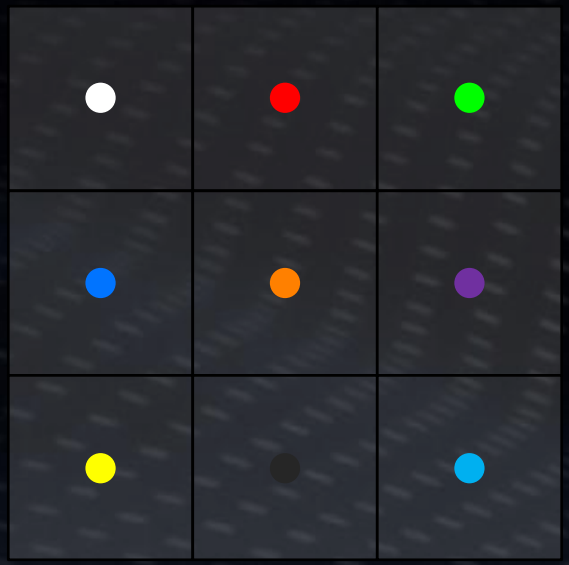


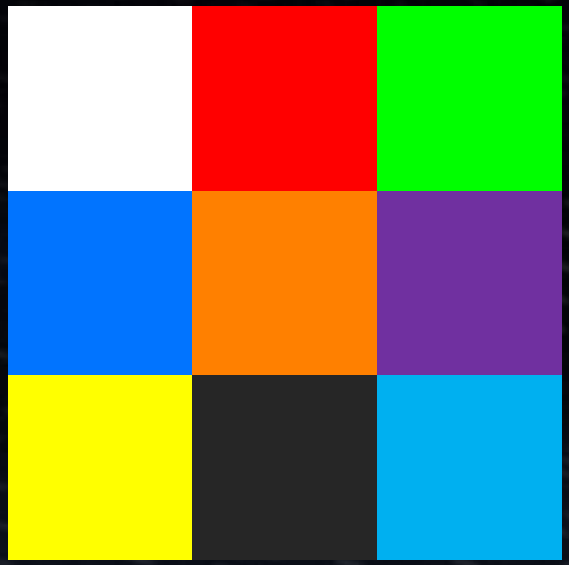
$$\frac{X}{Y}:$$

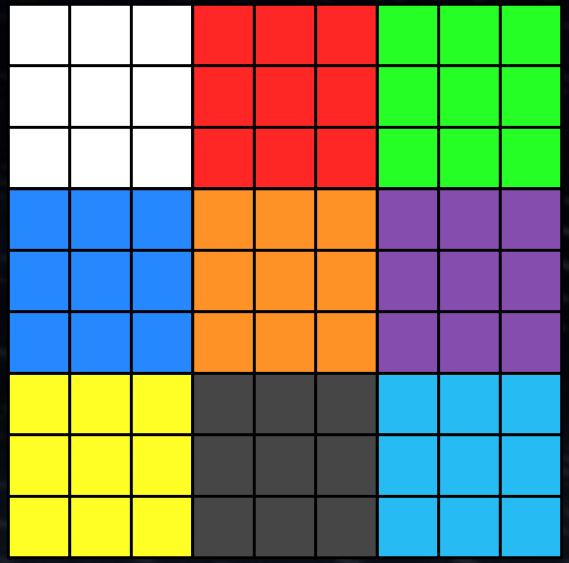
Монитор	1.0
PAL	1.09
PAL Widescreen	1.46
NTSC	0.91
NTSC Widescreen	1.21
Anamorphic	2.0
Cinemascope	2.39

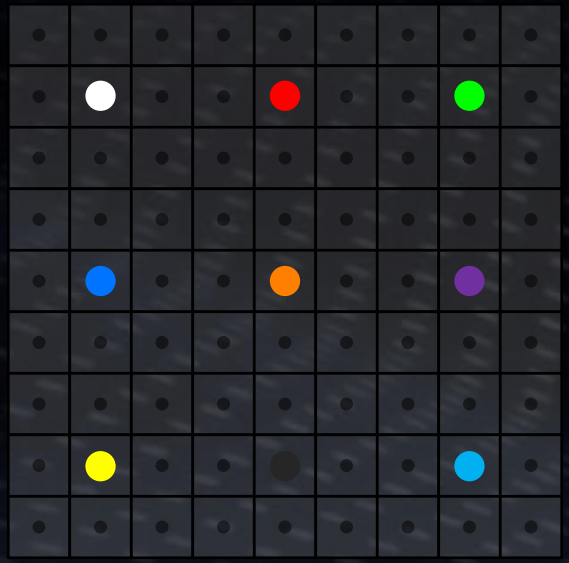
ЧТО ТАКОЕ ПИКСЕЛЬ?

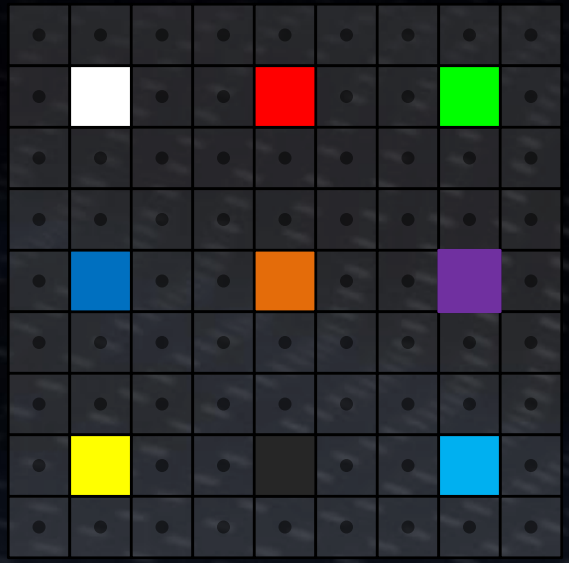


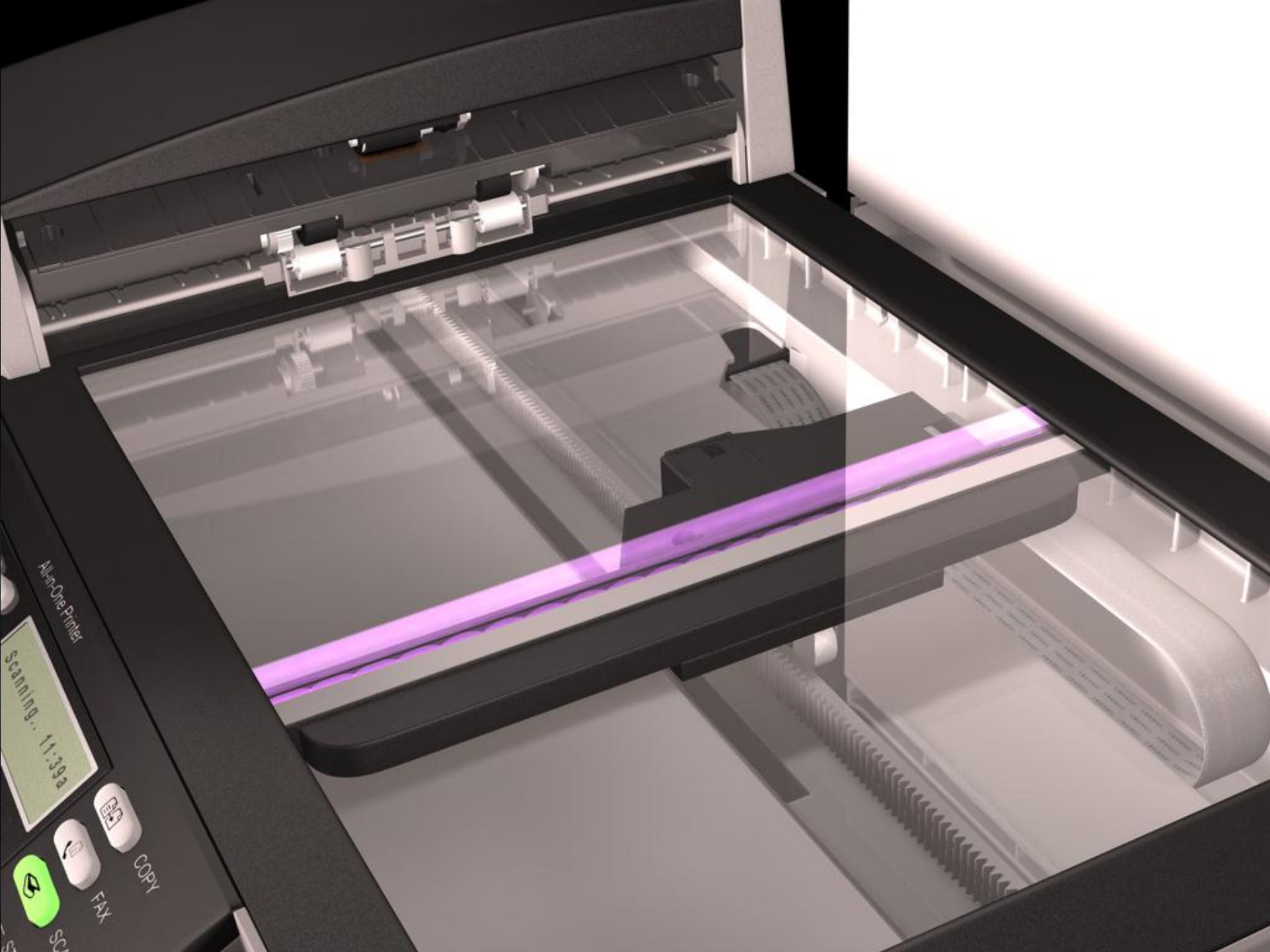


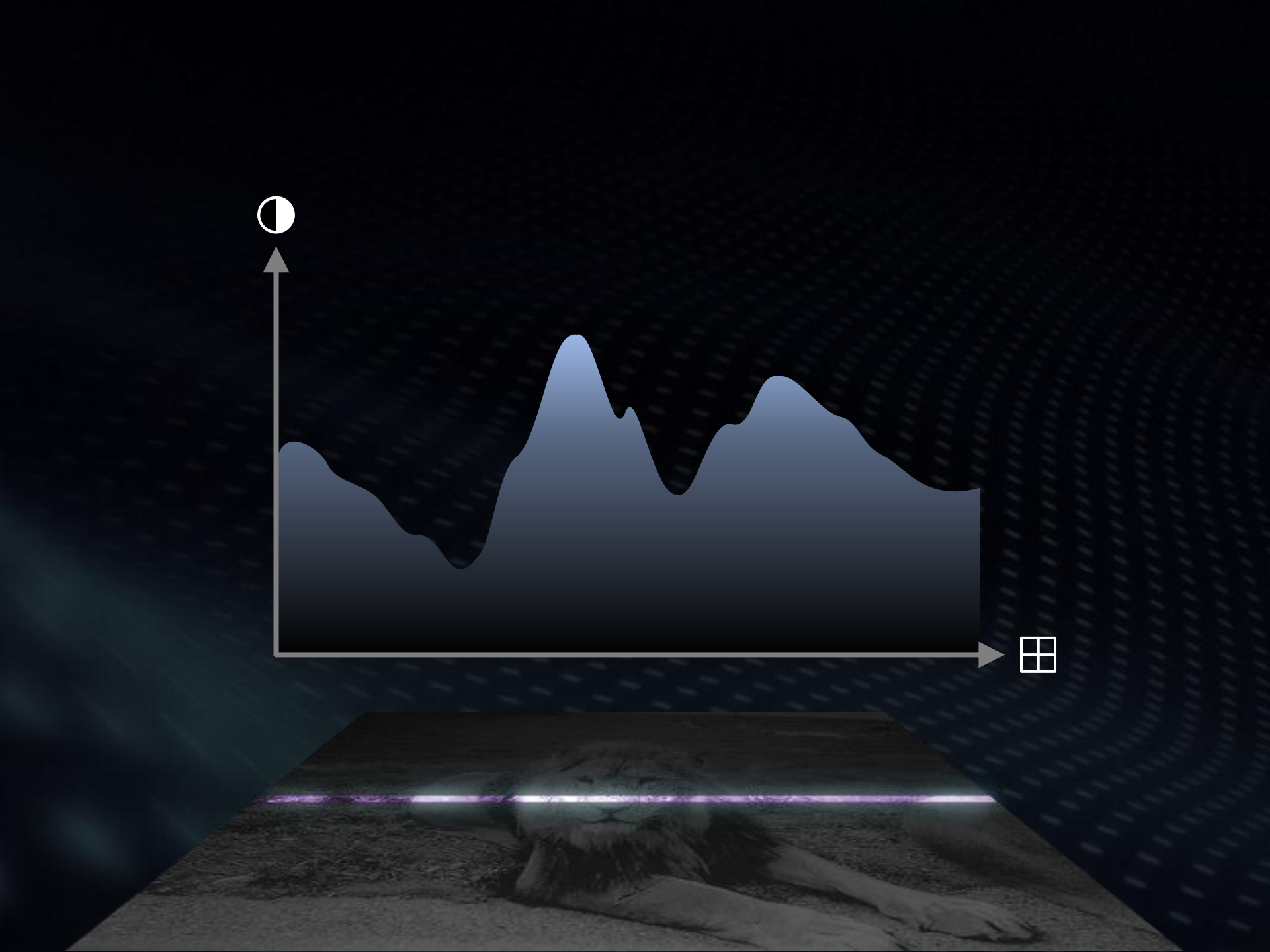


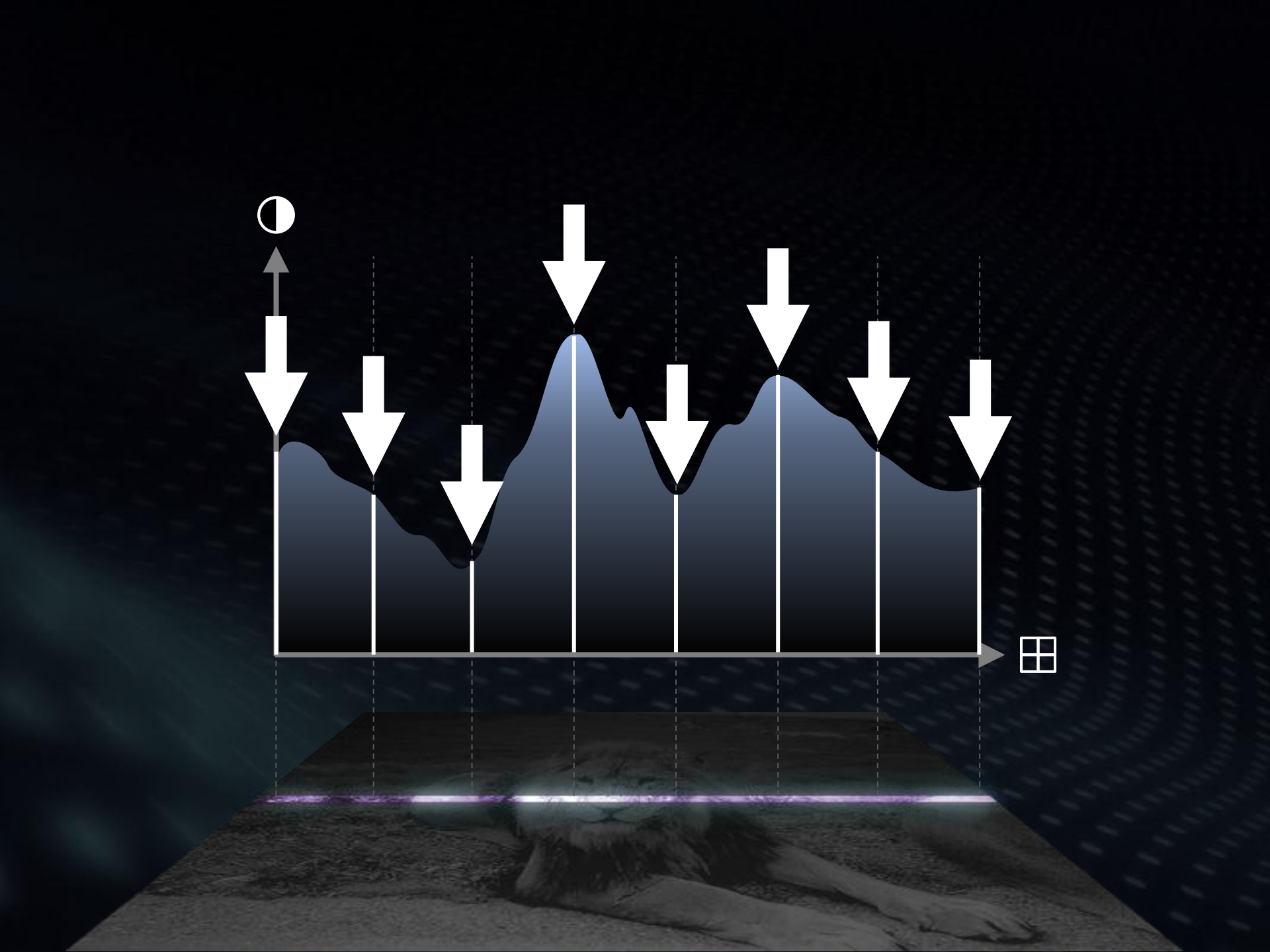


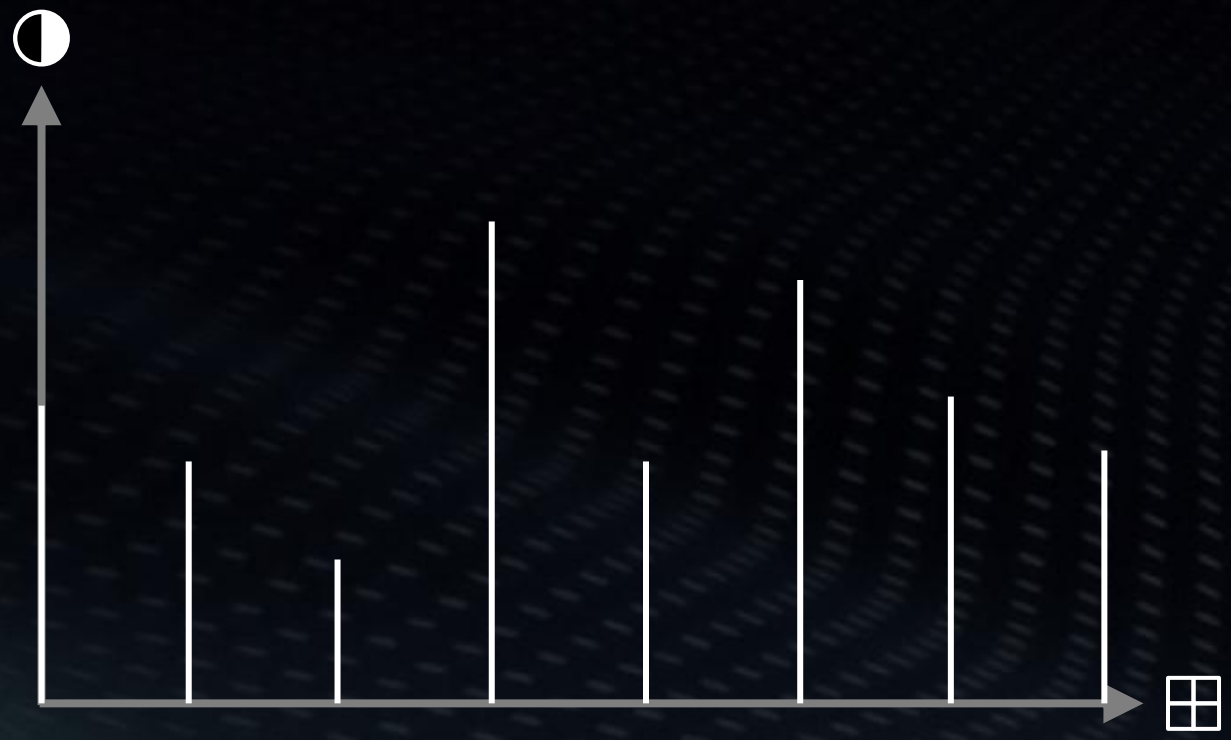


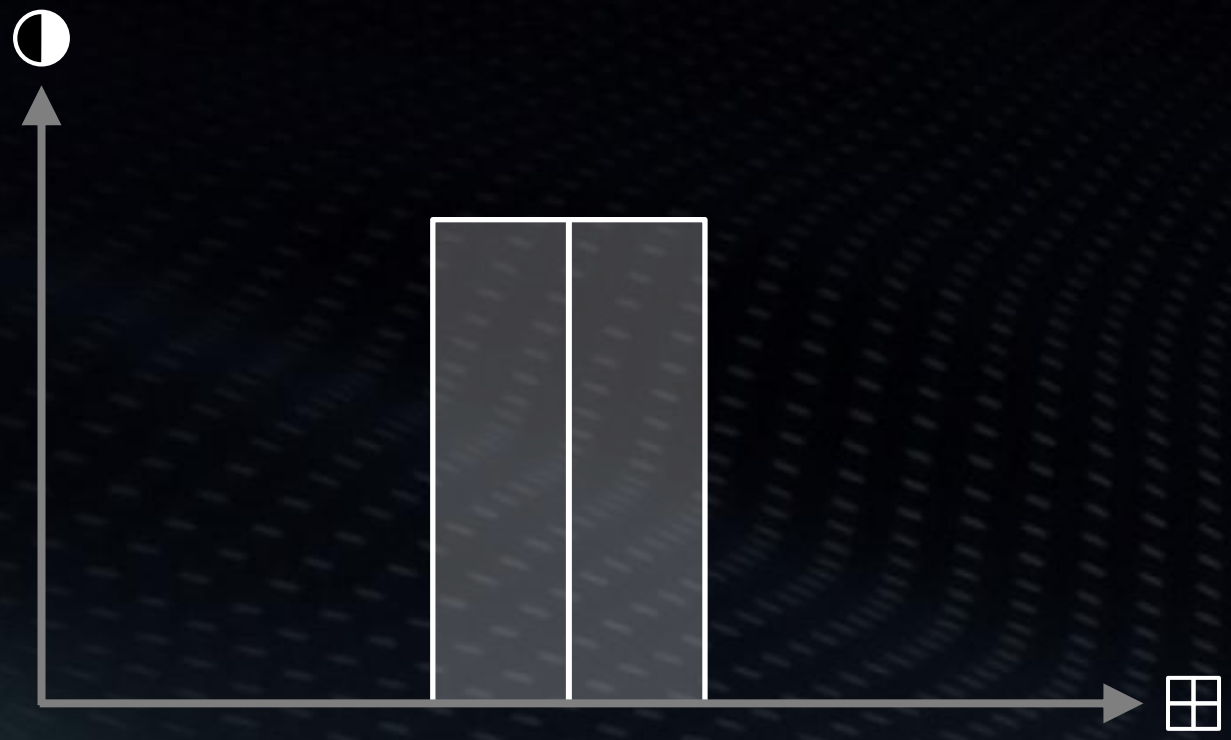


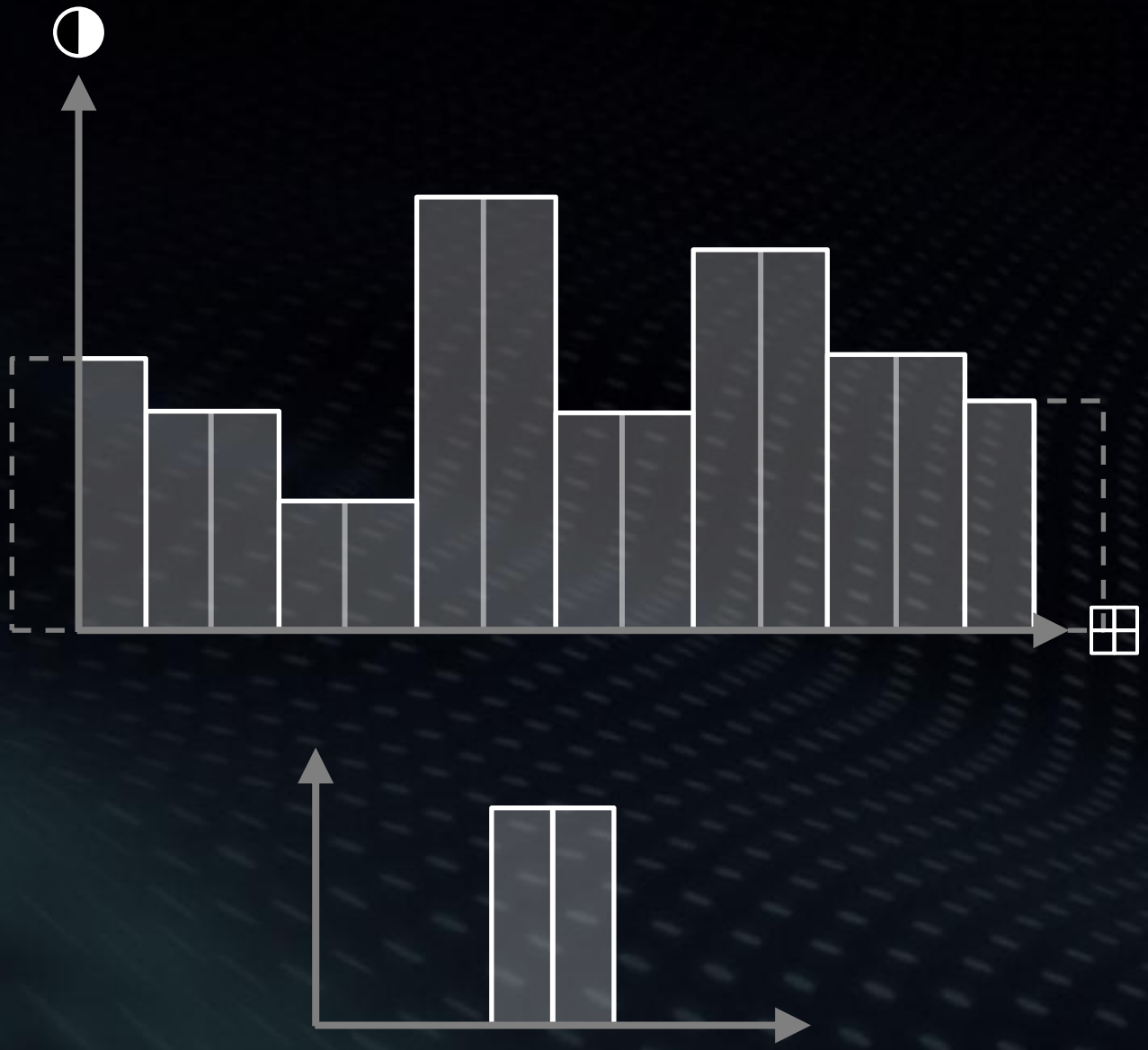

All-in-One Printer
Scanning.. 11:39a
COPY
FAX

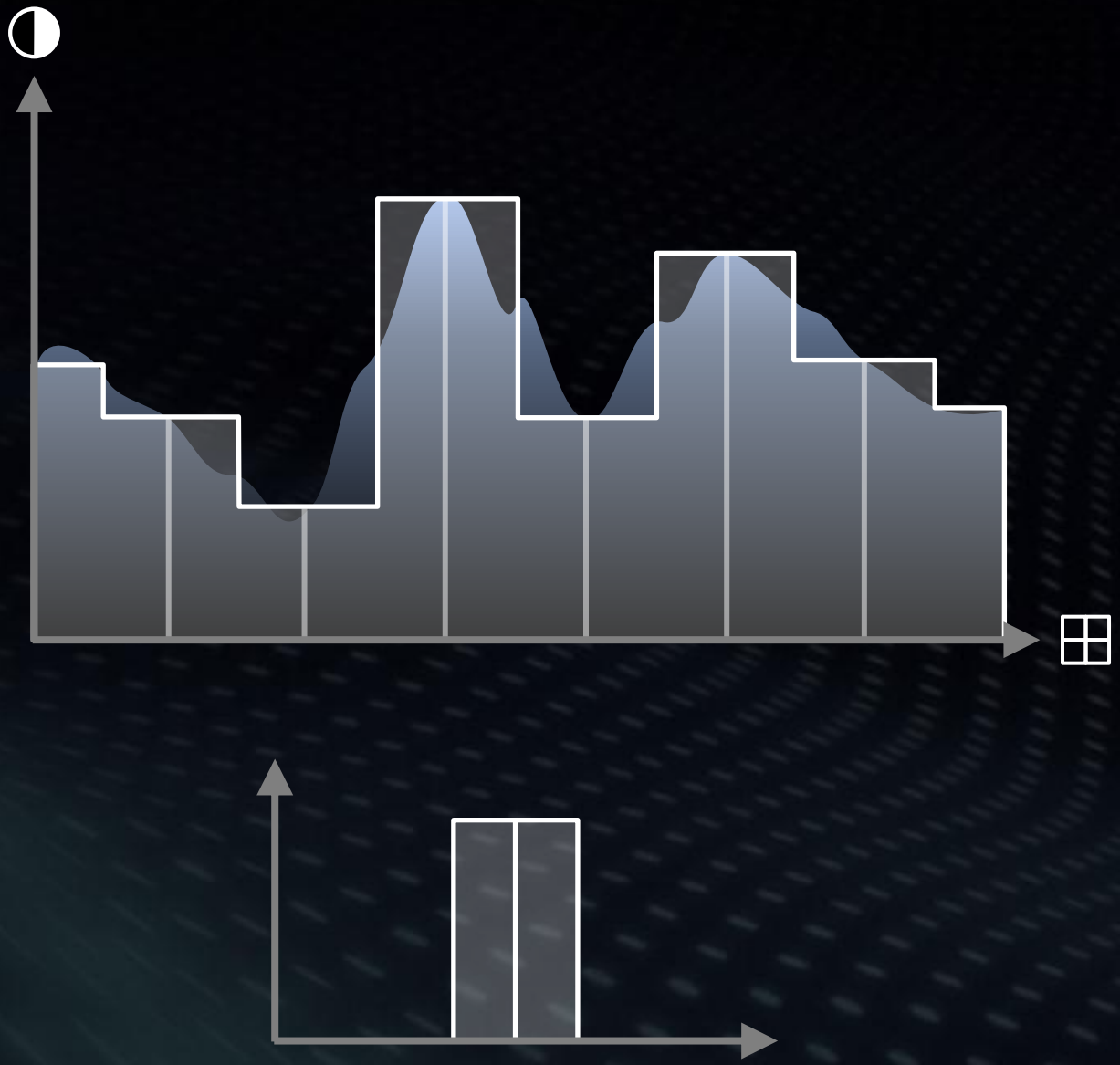


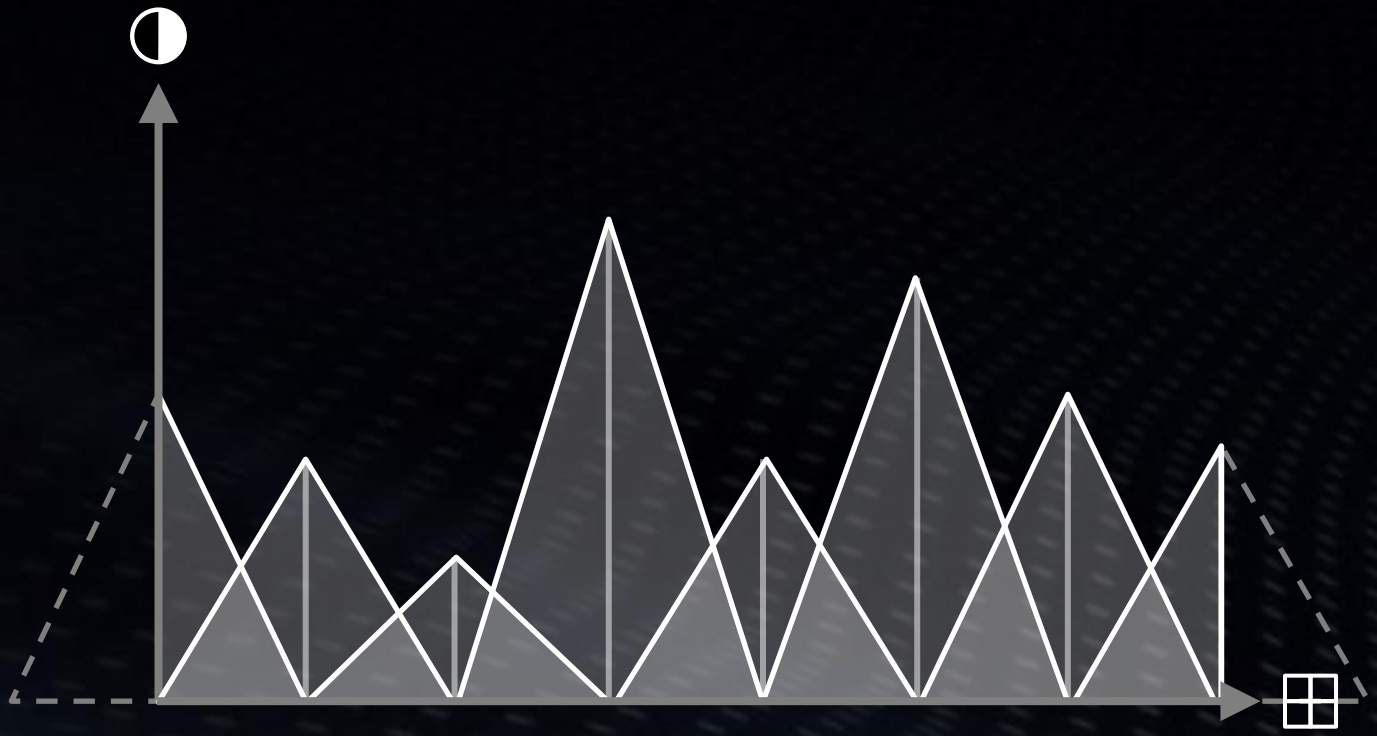


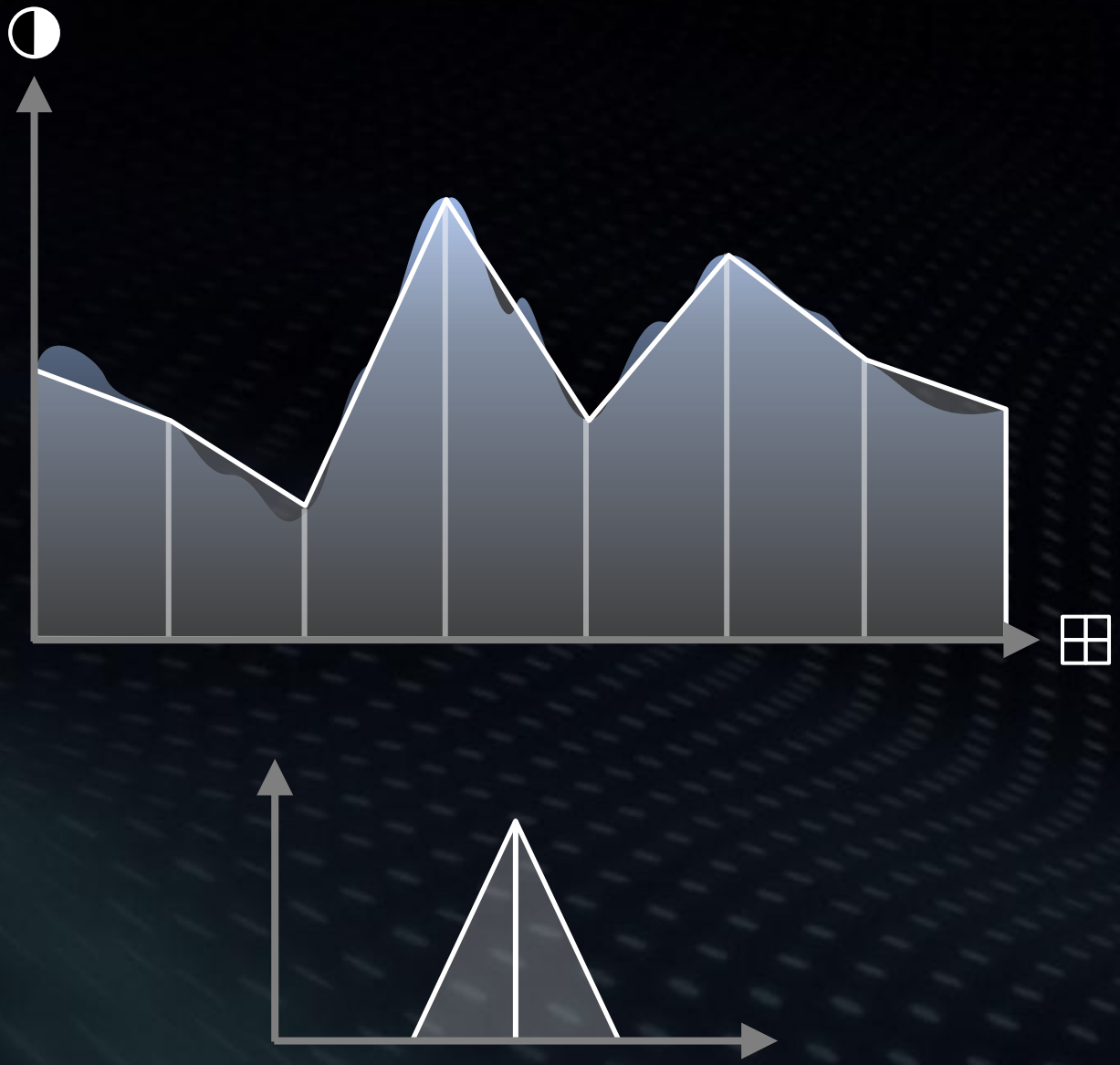


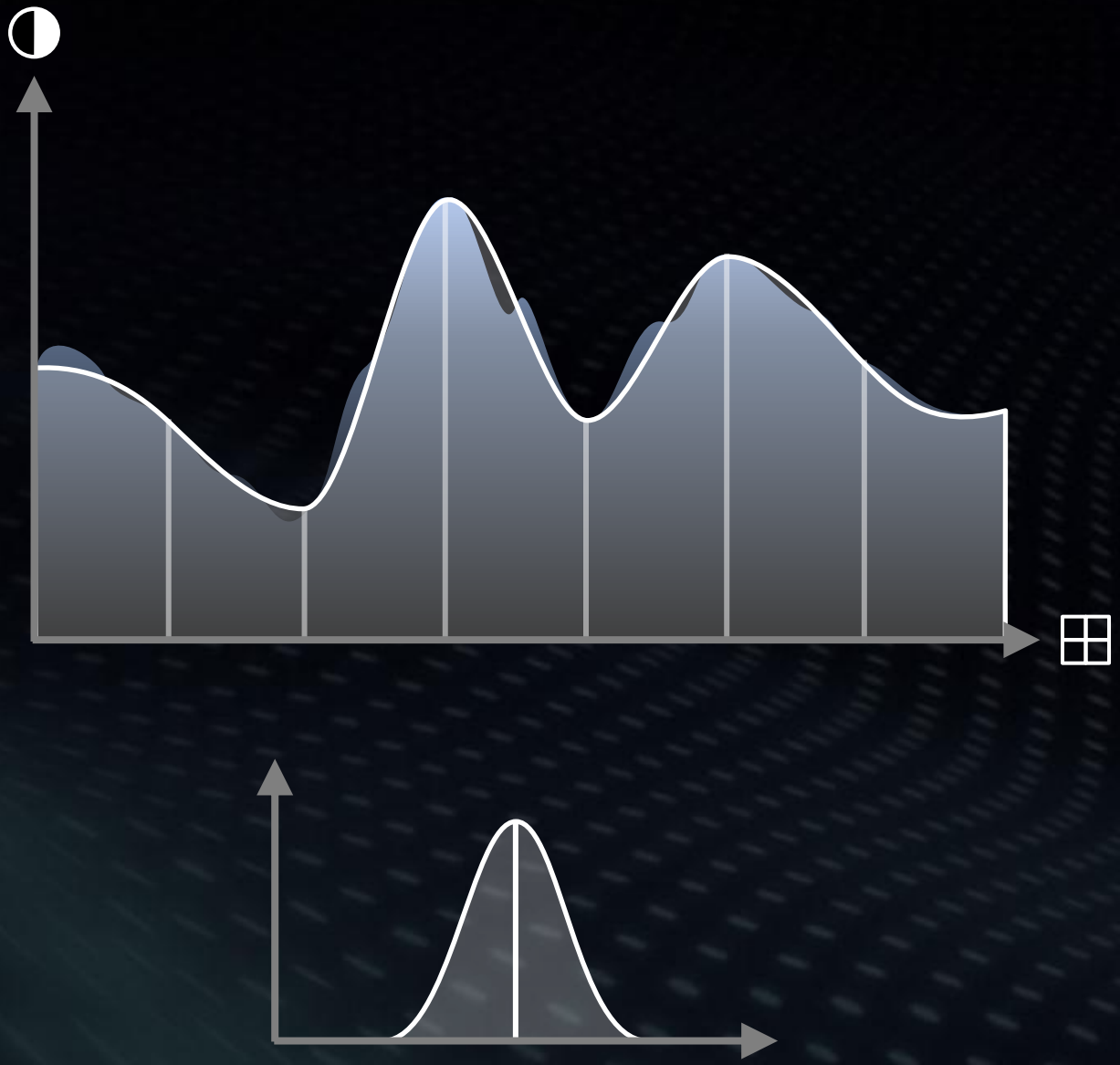


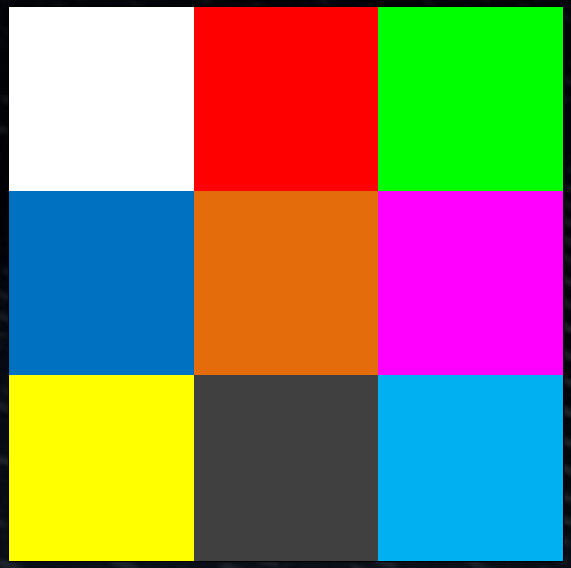


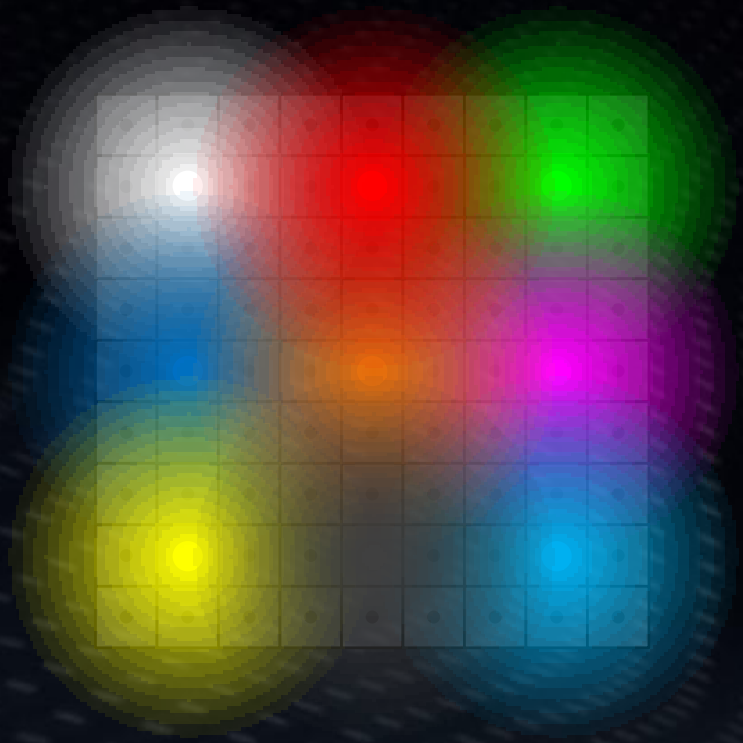


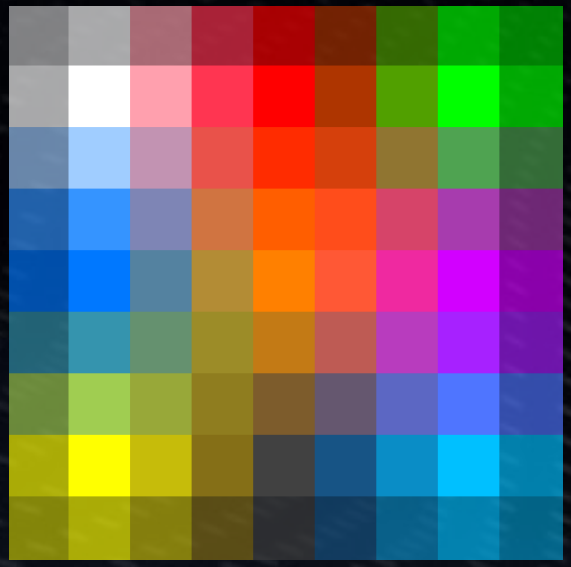


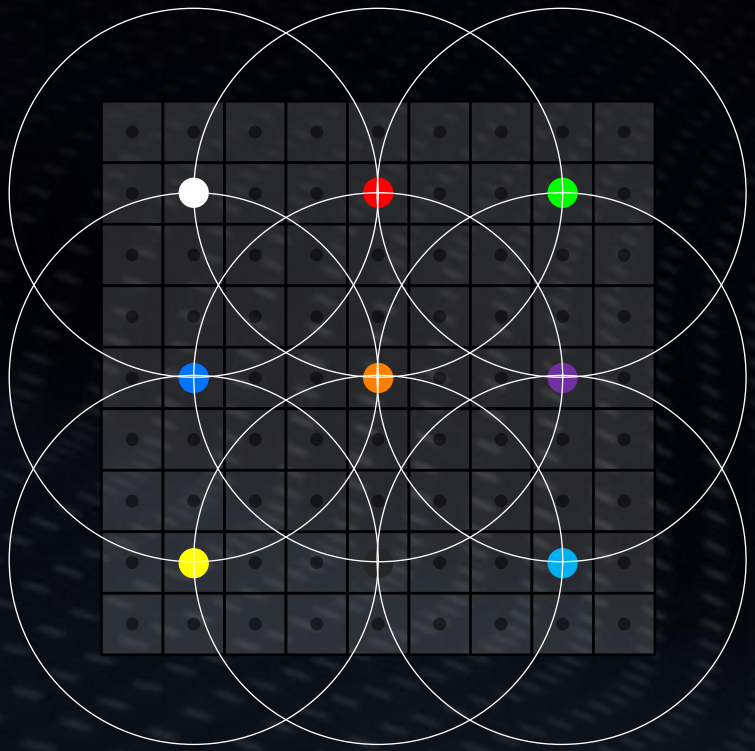


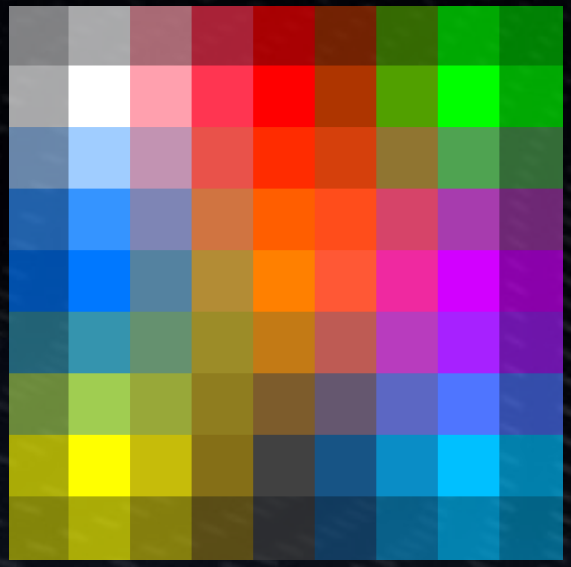






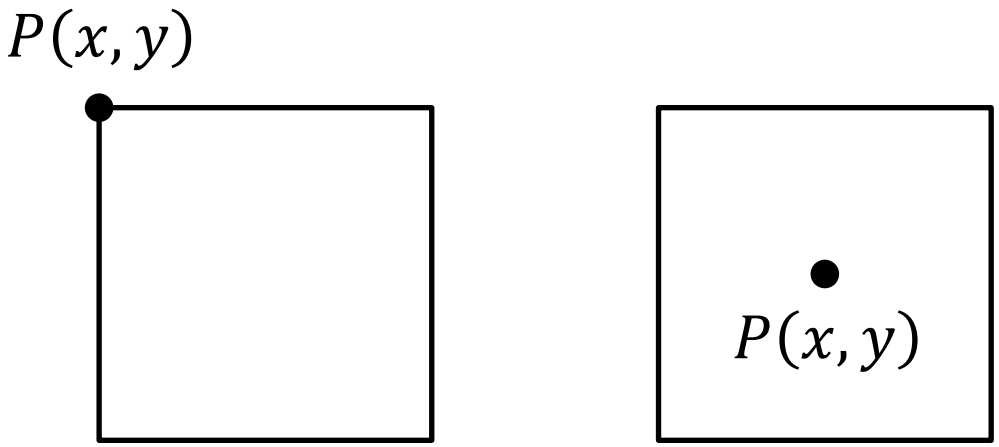


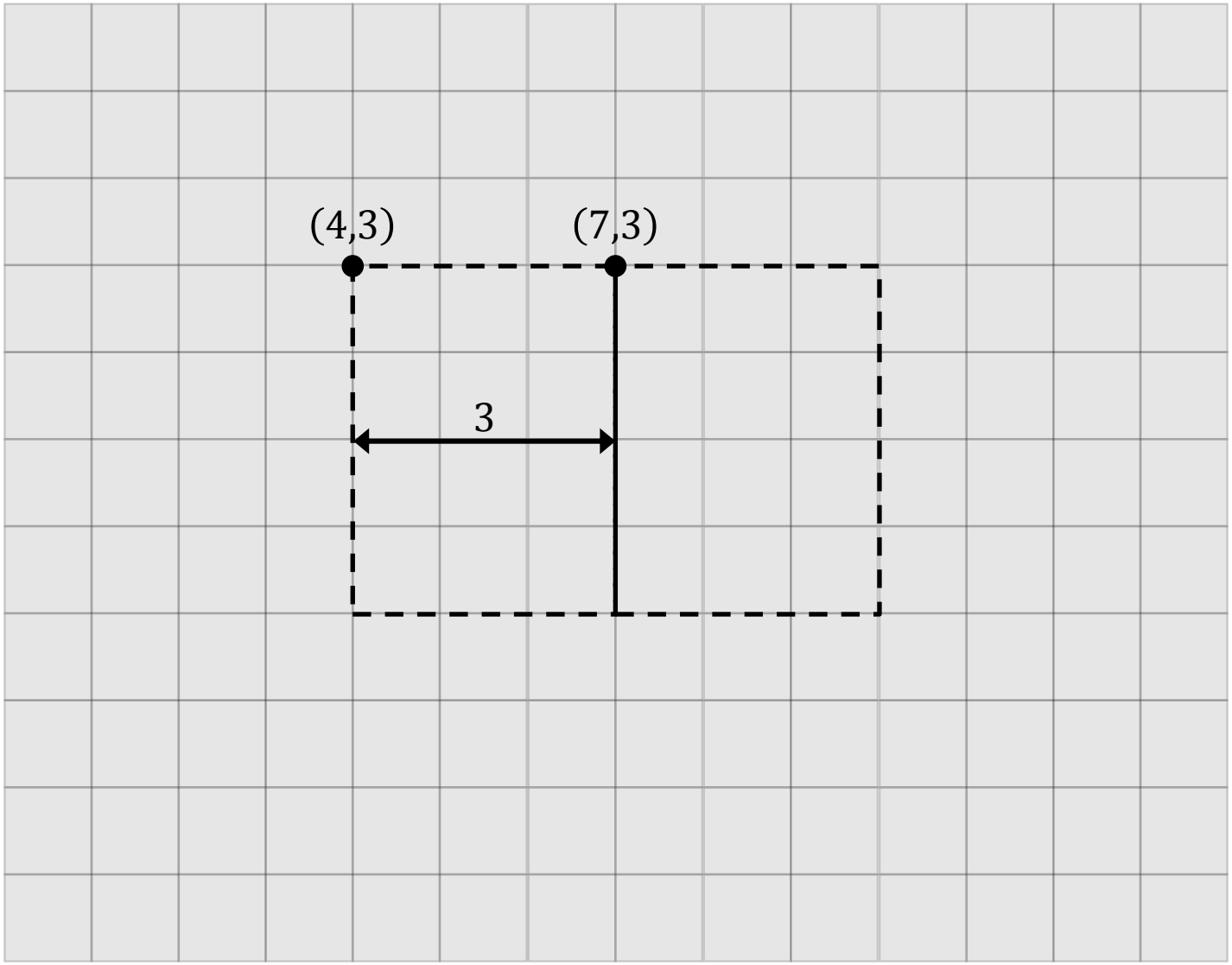




ко ор ди

на ты

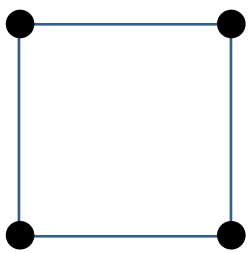

$P(x, y)$
$P(x, y)$

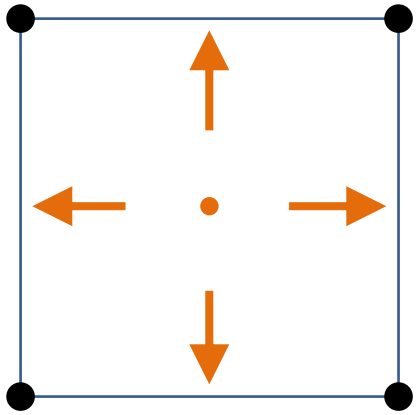

(4,3)
(7,3)
3

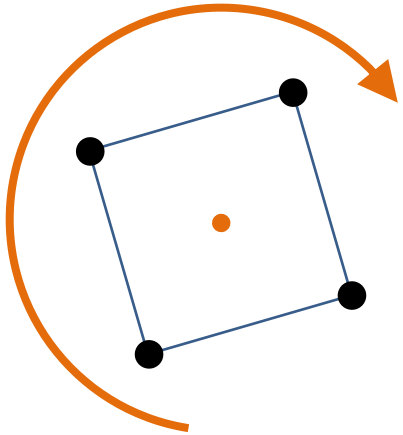
Преобразование координат

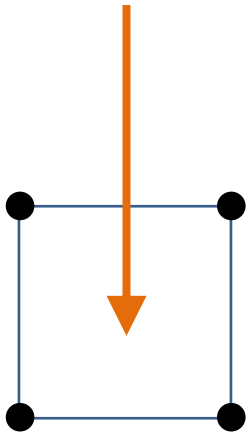
масштаб

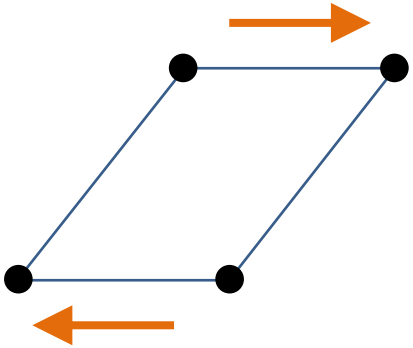
вращение

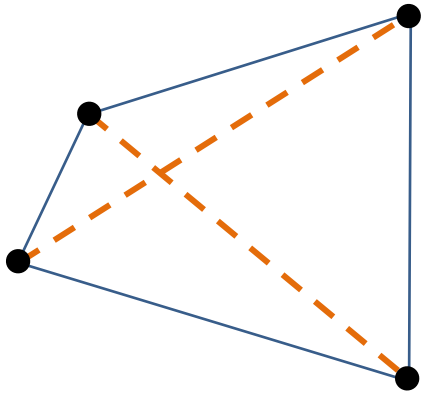








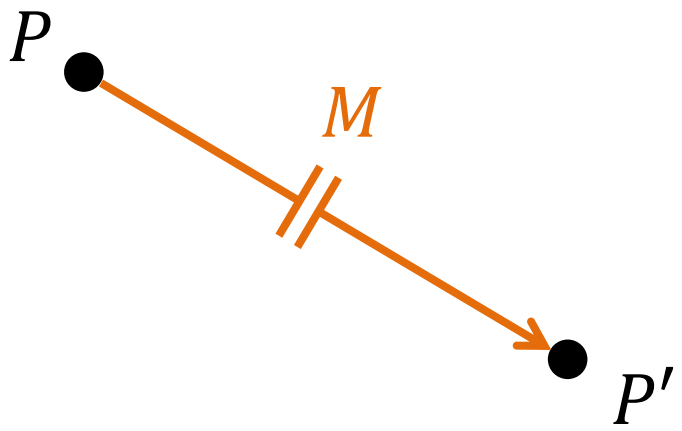




смещение

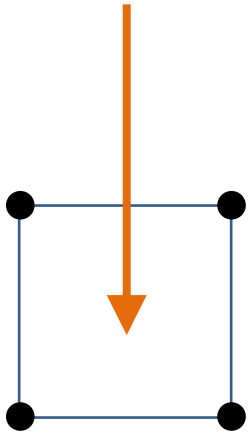
искажение

перспективная проекция



$$\vec{P}' = \vec{P} \cdot M$$

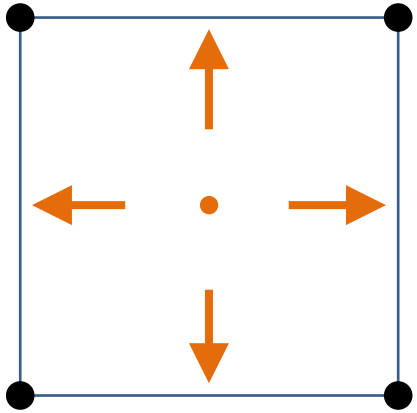
$$\begin{pmatrix} x' \\ y' \\ 0 \end{pmatrix} = \begin{pmatrix} x \\ y \\ 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$



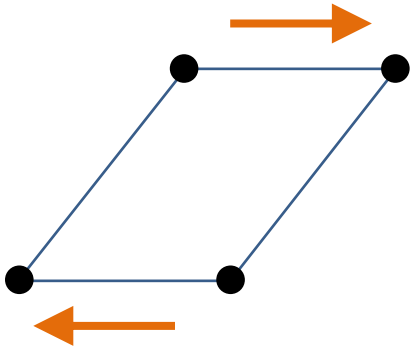
смещение

$$T(x,y) = \begin{pmatrix} 1 & 0 & x \\ 0 & 1 & y \\ 0 & 0 & 1 \end{pmatrix}$$

масштаб



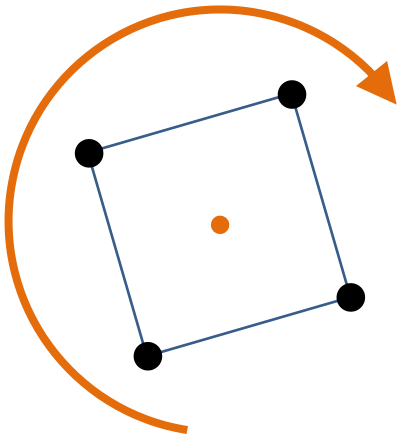
$$S(x,y) = \begin{pmatrix} x & 0 & 0 \\ 0 & y & 0 \\ 0 & 0 & 1 \end{pmatrix}$$



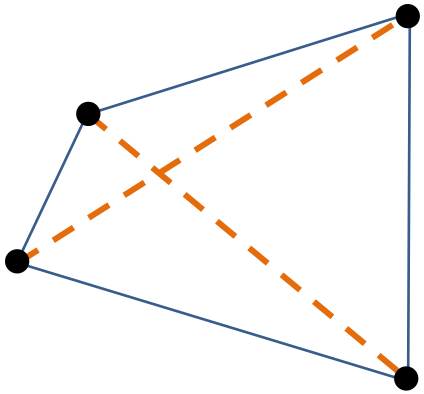
смещение

$$H(s) = \begin{pmatrix} 1 & s & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

вращение



$$R(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$



перспективная
проекция

$$S = P_0 - P_3 + P_2 - P_1$$
$$\Delta_1 = P_3 - P_2$$
$$\Delta_2 = P_1 - P_2$$
$$D = \Delta x_1 \Delta y_2 - \Delta x_2 \Delta y_1$$

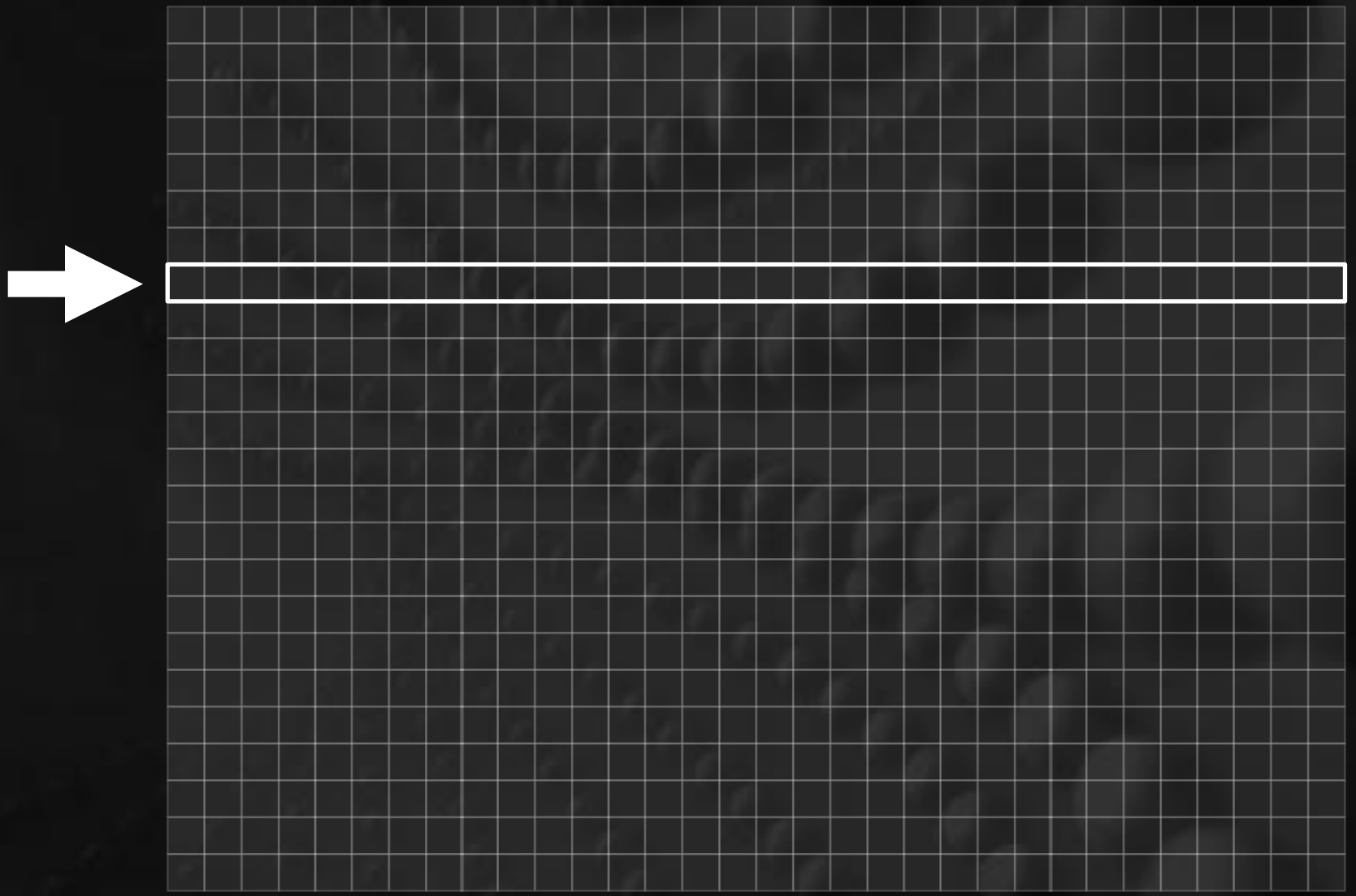
$$a = x_3 - x_0 + g x_3$$
$$b = x_1 - x_0 + h x_1$$
$$c = x_0$$
$$d = y_3 - y_0 + g y_3$$
$$e = y_1 - y_0 + h y_1$$
$$f = y_0$$
$$g = \frac{S_x \Delta y_2 - S_y \Delta x_2}{D}$$
$$h = \frac{S_y \Delta x_1 - S_x \Delta y_1}{D}$$

$$P(P_1, P_2, P_3, P_4) = \begin{pmatrix} a & d & g \\ b & e & h \\ c & f & i \end{pmatrix}$$

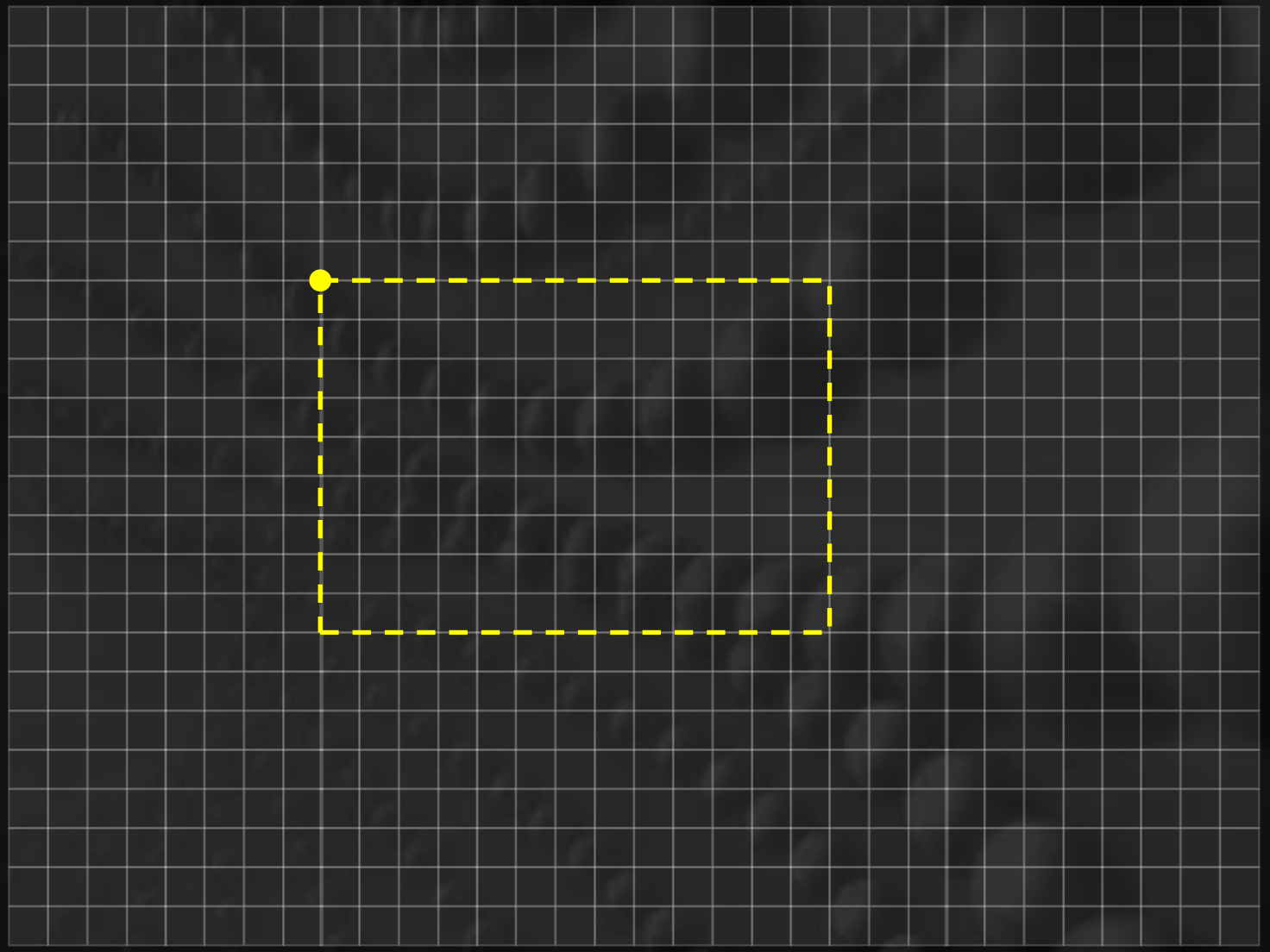

iPad
63%
9:59
Saturday, May 1
THE RAPTURE ECHOES
FOO FIGHTERS
ECHOES, SILENCE, PATIENCE & GRACE
Echoes, Silence, Patience & Grace
Foo Fighters
slide to unlock

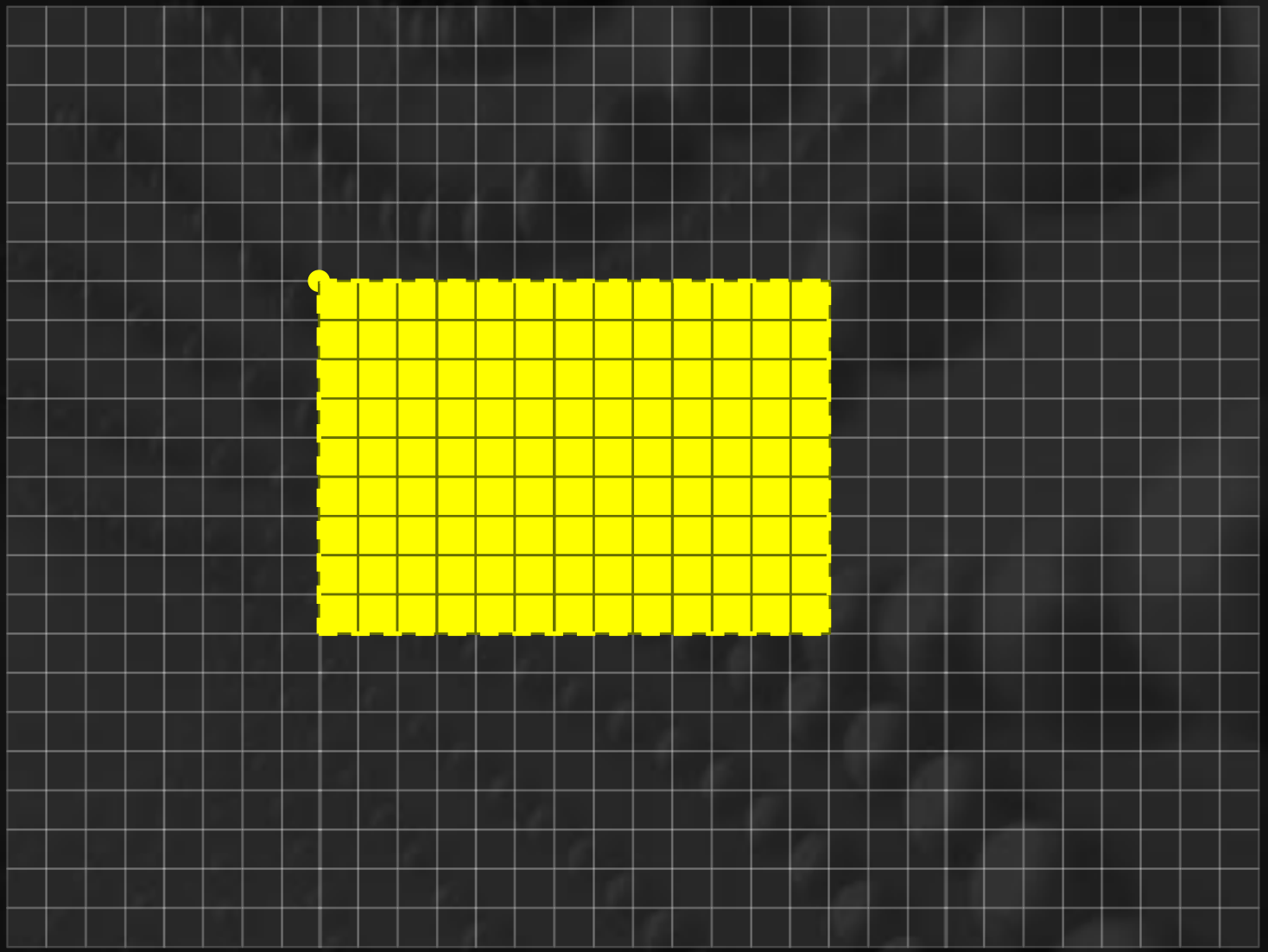
РАС
ТЕР
ИЗА
ЦИЯ

Сканлинии



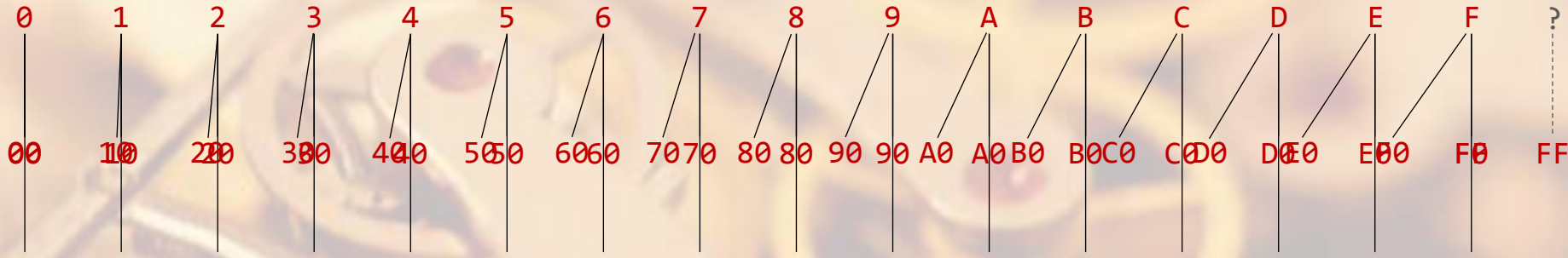
Сканлинии



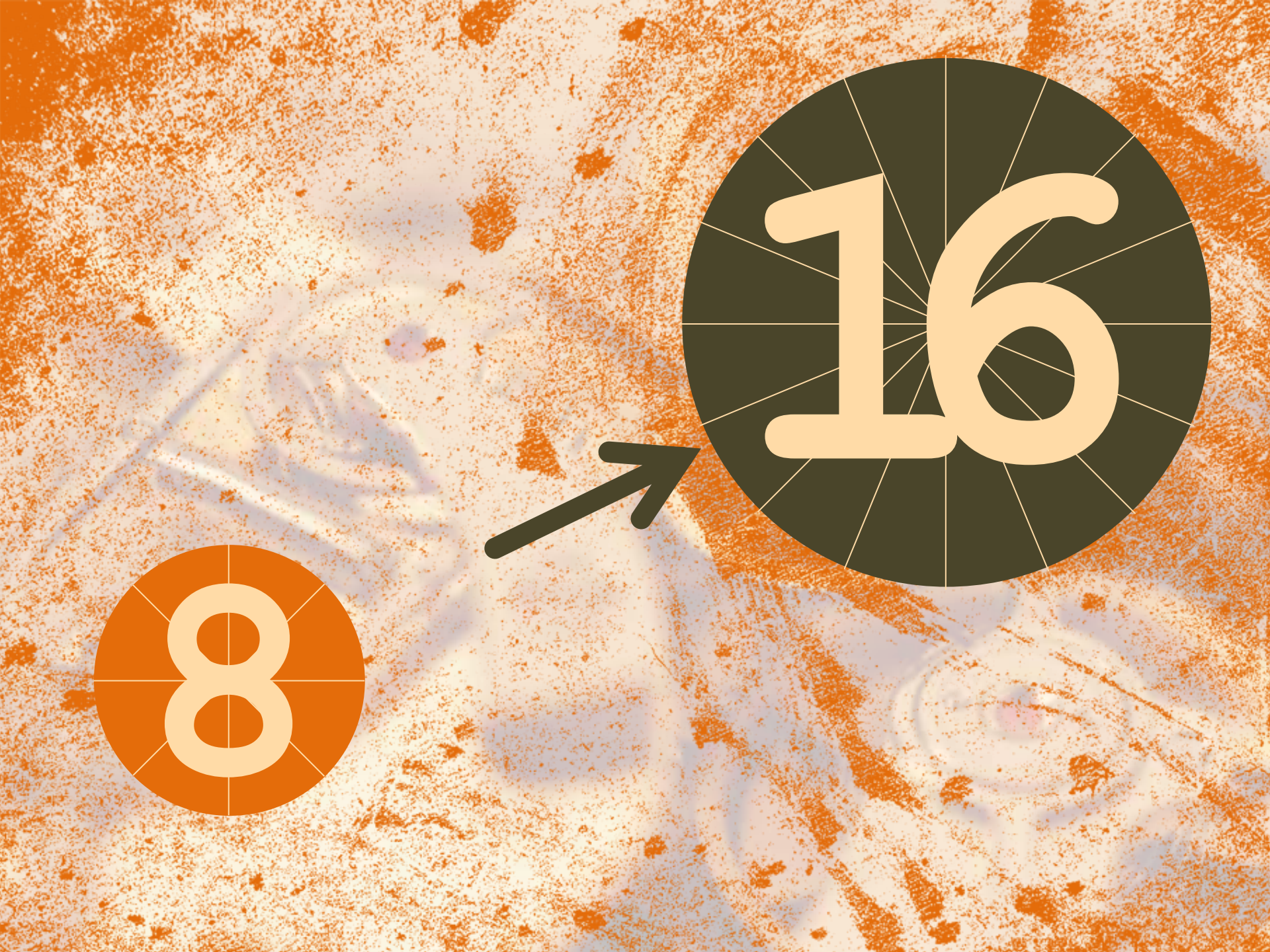


битность

```
byte   b = 0xFF;   // 8 битов
ushort w = 0xFFFF; // 16 битов
```


0 1 2 3 4 5 6 7 8 9 A B C D E F ?
00 10 20 30 40 50 60 70 80 90 A0 B0 C0 D0 E0 F0 FF

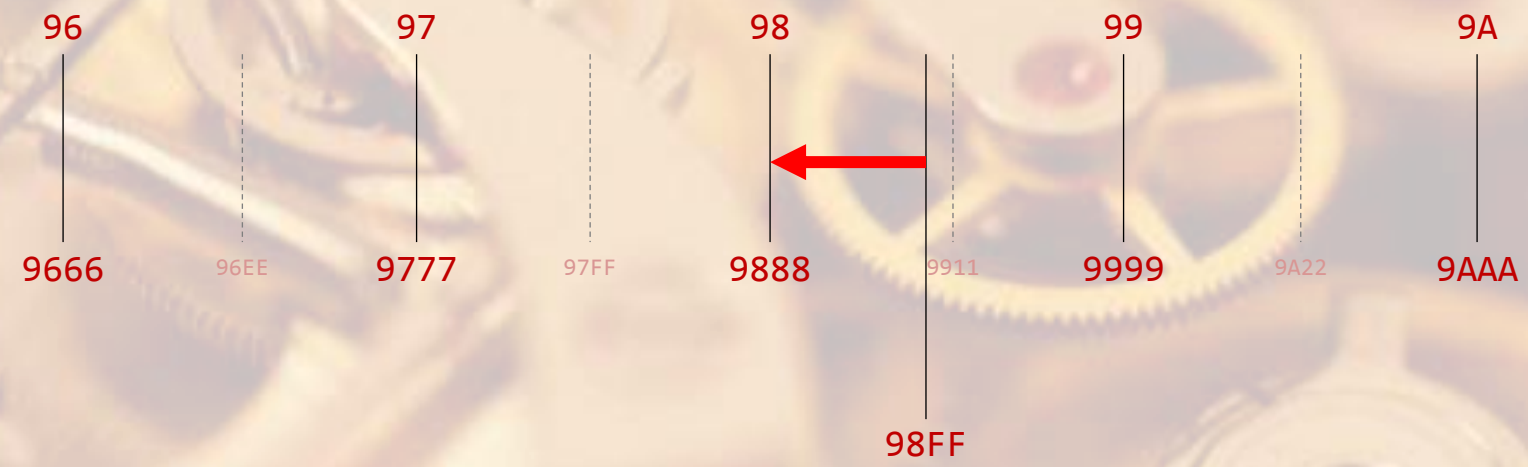

0 1 2 3 4 5 6 7 8 9 A B C D E F
00 10 20 30 40 50 60 70 80 90 A0 B0 C0 D0 E0 F0 FF
00 FF/F 22 33 44 55 66 77 88 99 AA BB CC DD EE FF
000 111 222 333 444 555 666 777 888 999 AAA BBB CCC DDD EEE FFF
0000 1111 2222 3333 4444 5555 6666 7777 8888 9999 AAAA BBBB CCCC DDDD EEEE FFFF
#fff = #ffffff


16
8

```
byte b = 0x98;
ushort w = b << 8; // 0x9800
ushort w = (b << 8) | b; // 0x9898
```

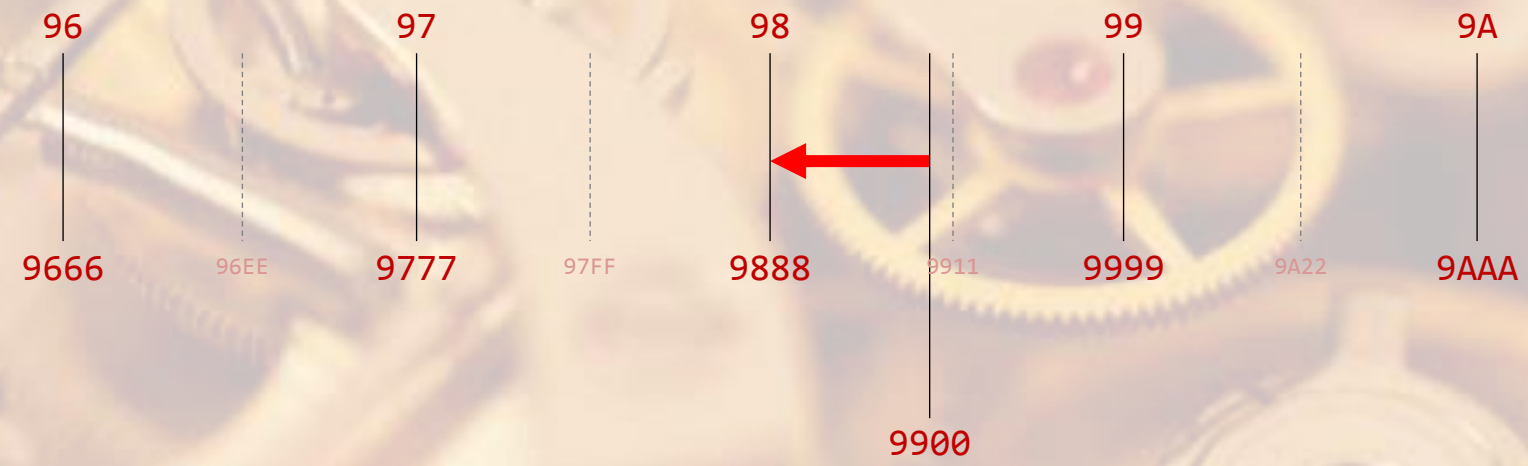

16
8

```
ushort w = 0x98FF;
byte b = w >> 8; // 0x98
byte b = (w + 0x80) >> 8; // 0x997F >> 8 = 0x99
```

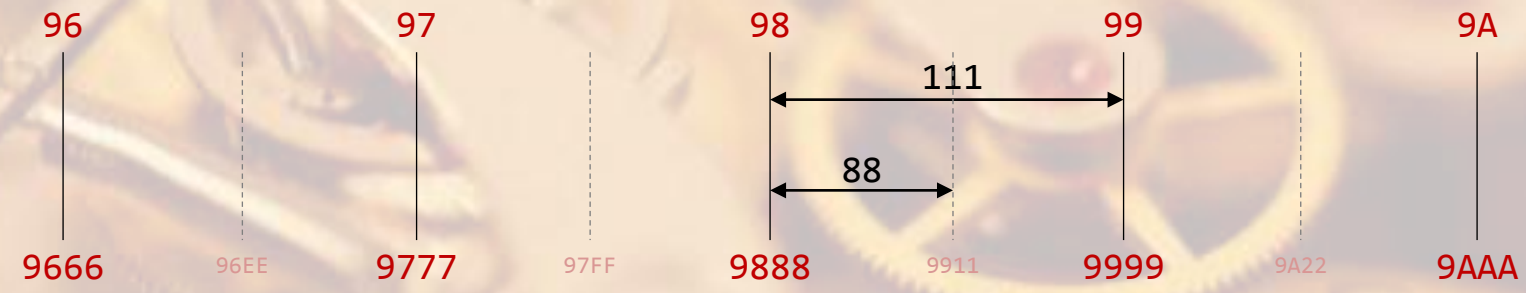

96
97
98
99
9A
9666
96EE
9777
97FF
9888
9911
9999
9A22
9AAA
98FF

```
ushort w = 0x98FF;
byte b = w >> 8; // 0x98
byte b = (w + 0x80) >> 8; // 0x997F >> 8 = 0x99
```

```
ushort w = 0x9900;
byte b = w >> 8; // 0x99
byte b = (w + 0x80) >> 8; // 0x9980 >> 8 = 0x99
```

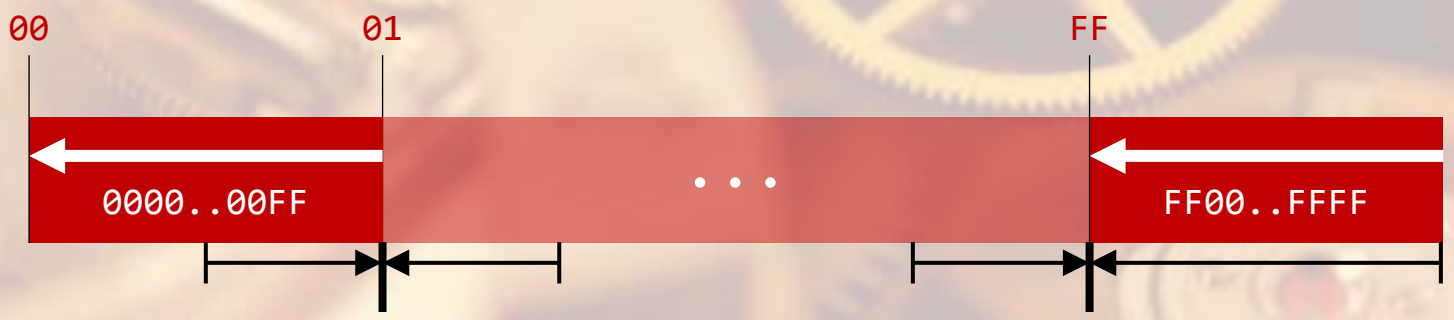

96
97
98
99
9A
9666
96EE
9777
97FF
9888
9911
9999
9A22
9AAA
9900

```
ushort w = 0x9900;
byte b = w >> 8; // 0x99
byte b = (w + 0x80) >> 8; // 0x9980 >> 8 = 0x99
```


96
97
98
99
9A
111
88
9666
96EE
9777
97FF
9888
9911
9999
9A22
9AAA

```
ushort w = 0x9900;

byte b =
        ((w & 0xF000) >> 8)
      | ((w & 0x0FFF) + 0x88) / 0x111; // 0x98
```

```
ushort w = 0x9900;
byte b = w >> 8;
byte b = (w + 0x80) >> 8;
```



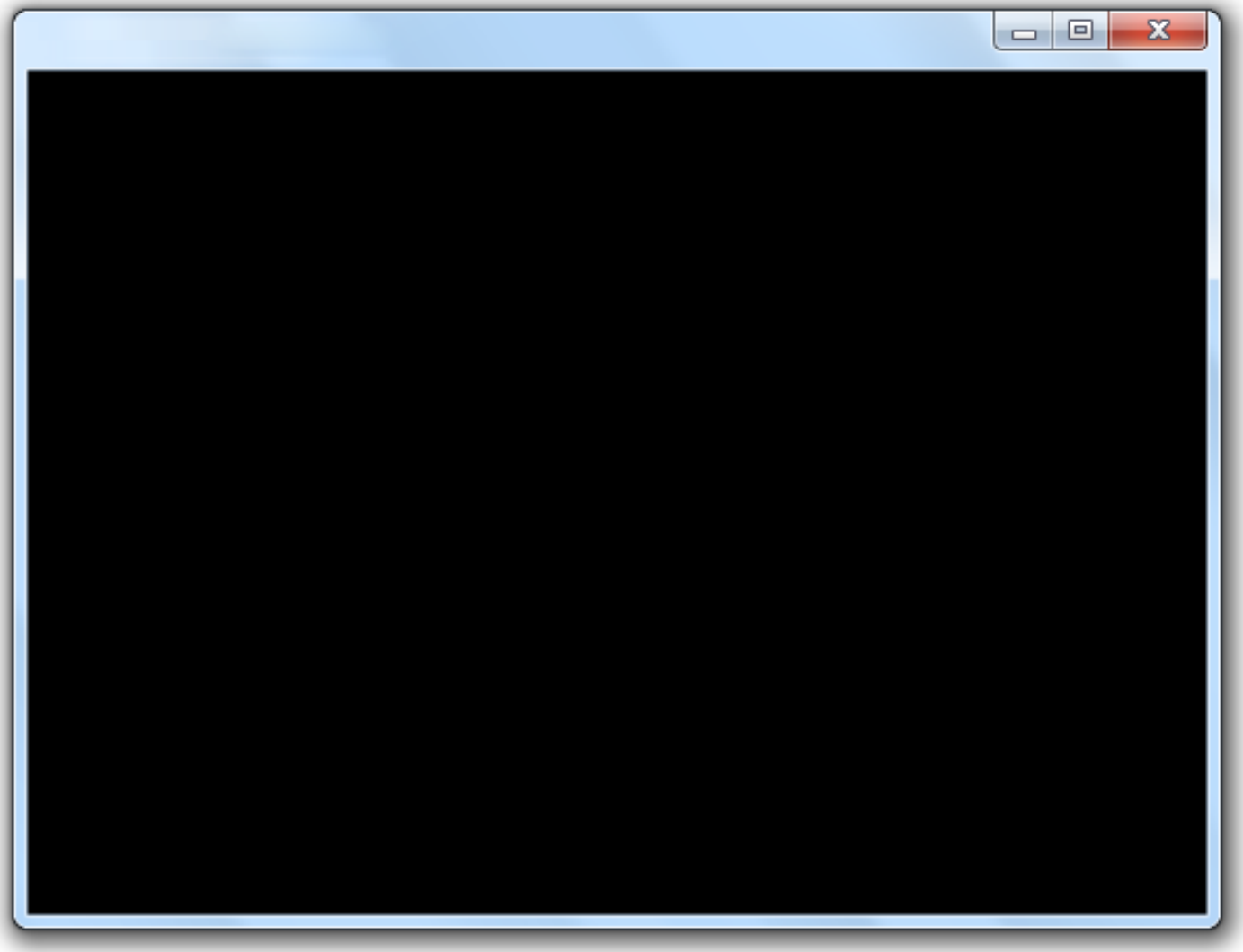
Image

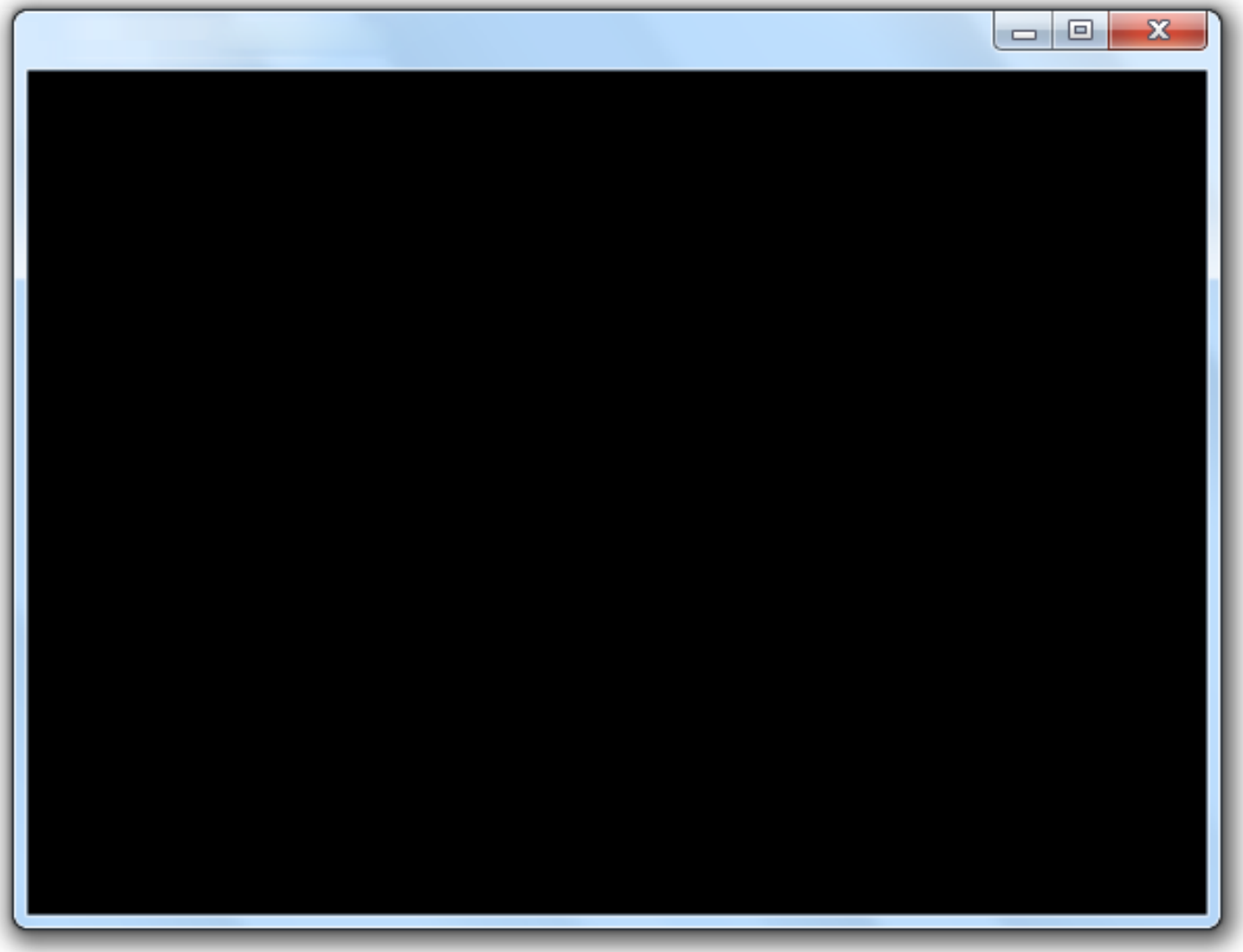
```
class Image
{
public:
    Image(uint width, uint height)
        : m_width(width)
        , m_height(height)
    {
        m_buffer = (pix8*)calloc(m_width * m_height, sizeof(pix8));
    }

    ~Image() { free(m_buffer); }

    uint  GetWidth()  { return m_width; }
    uint  GetHeight() { return m_height; }

    pix8* GetBuffer() const { return m_buffer; }

private:
    uint  m_width;
    uint  m_height;

    pix8* m_buffer;
};
```

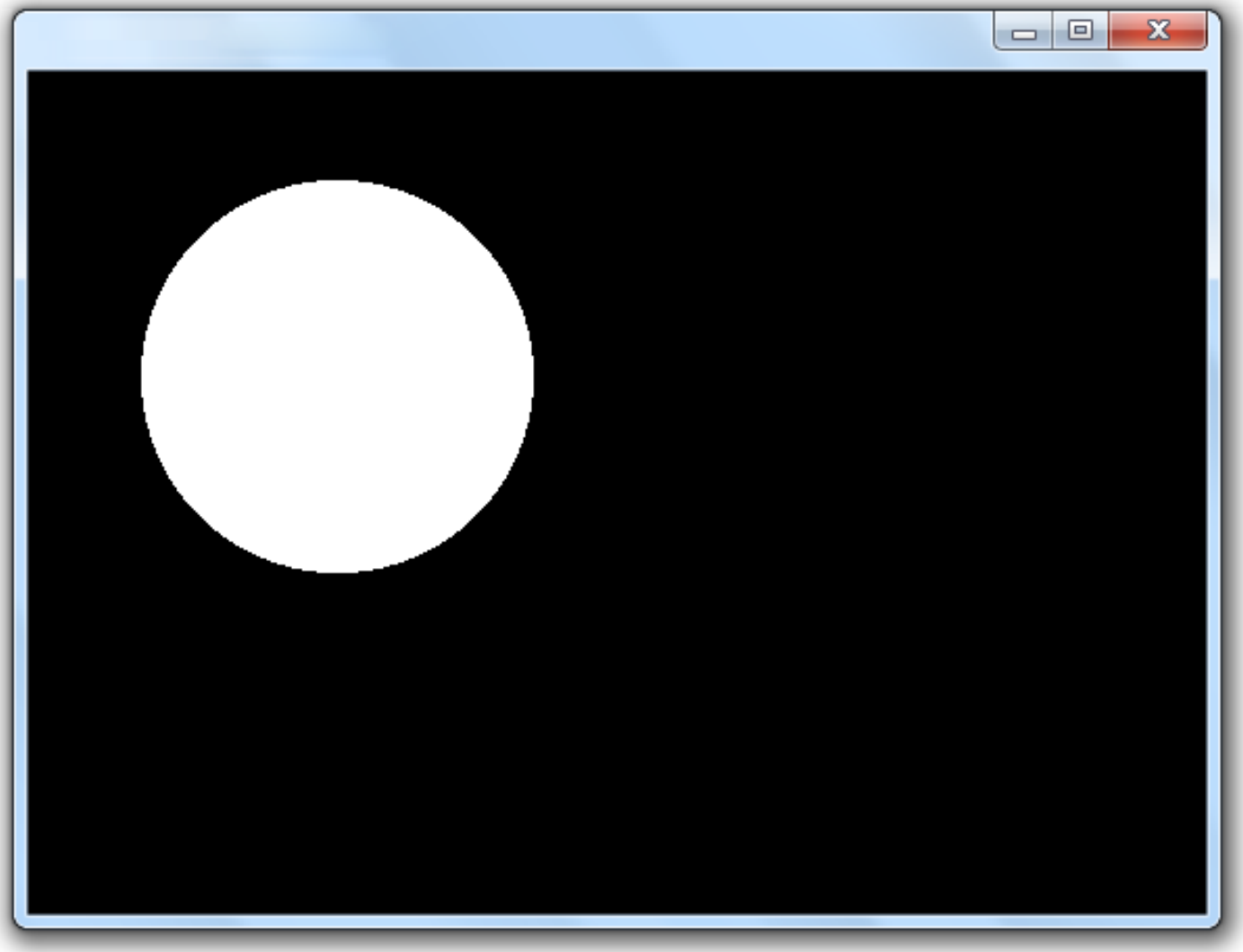
Graphics

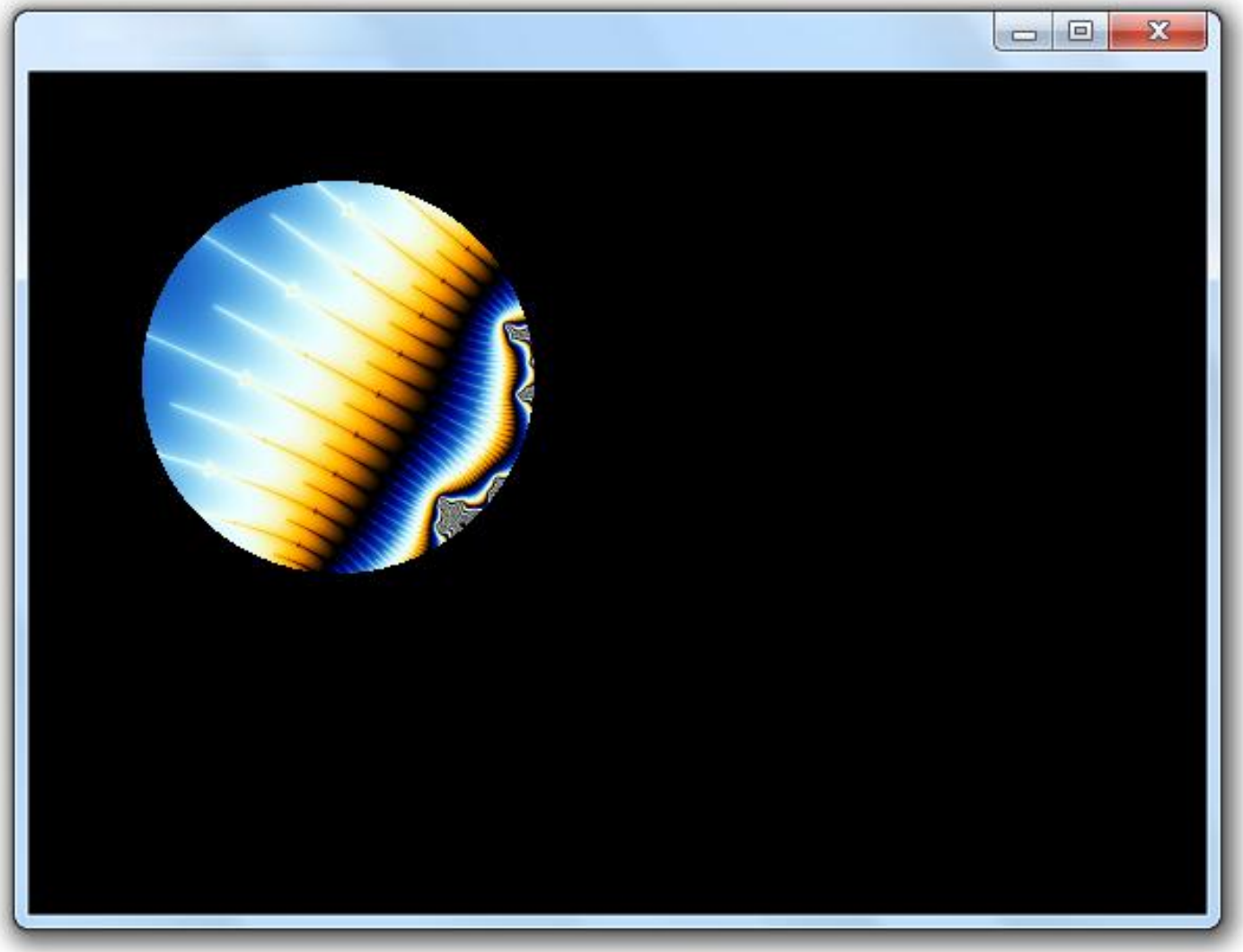
```
class Graphics
{
public:
    Graphics(Image* image) : m_image(image) {}

    void FillRect(int x, int y, int width, int height, const Brush& brush);
    void DrawImage(int x, int y, Image* image, float opacity);

    /*
     *
     *
     *
     *                              ...etc...
     *
     *
     *
     */

private:
    Image* m_image;
};
```

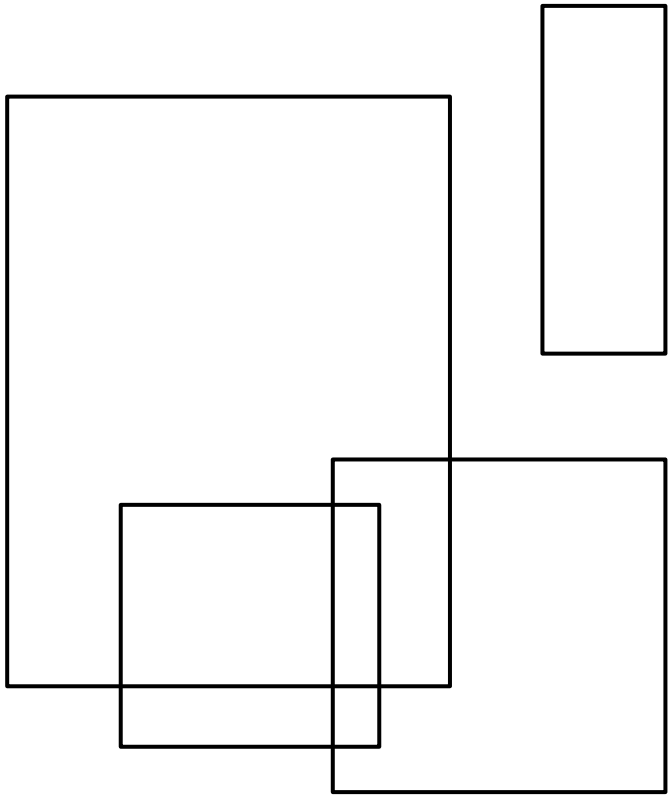
РЕГИОНЫ

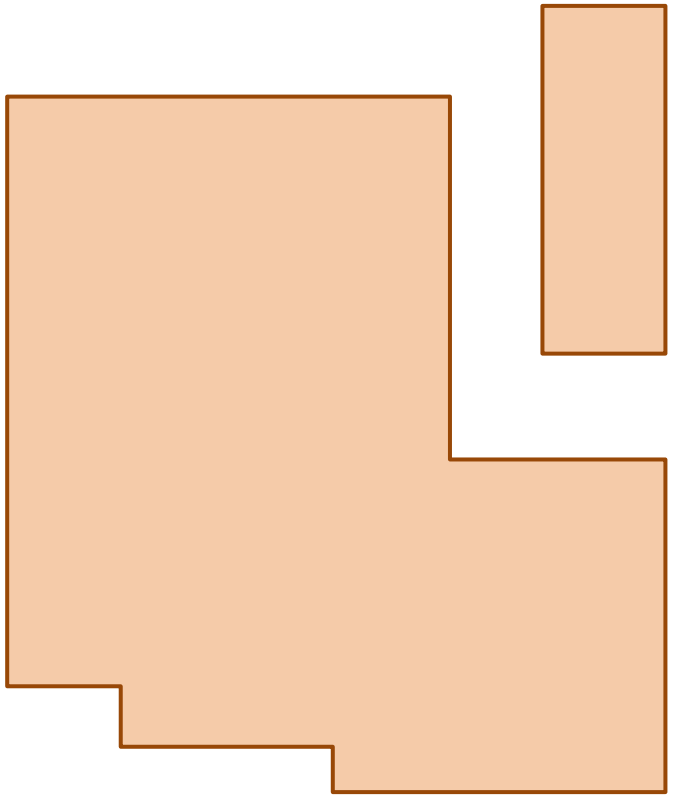


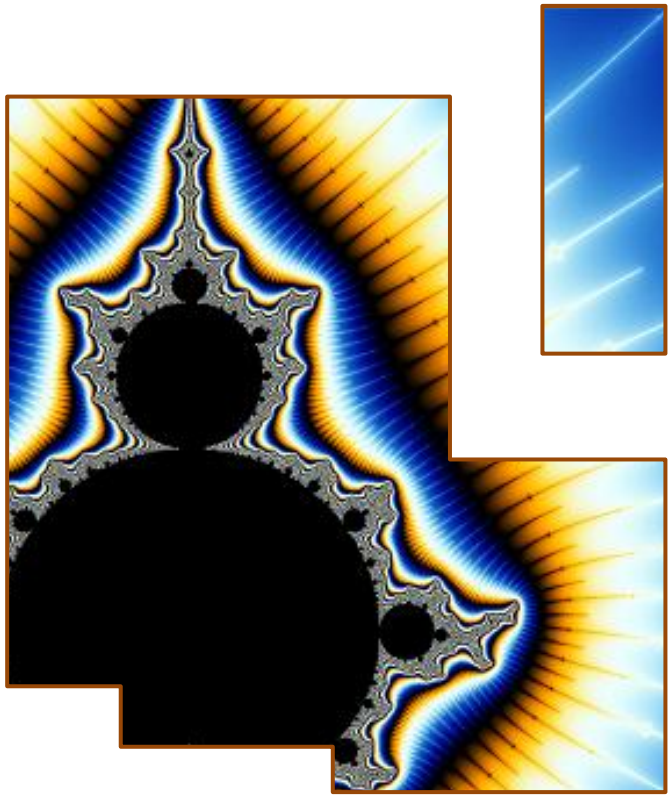


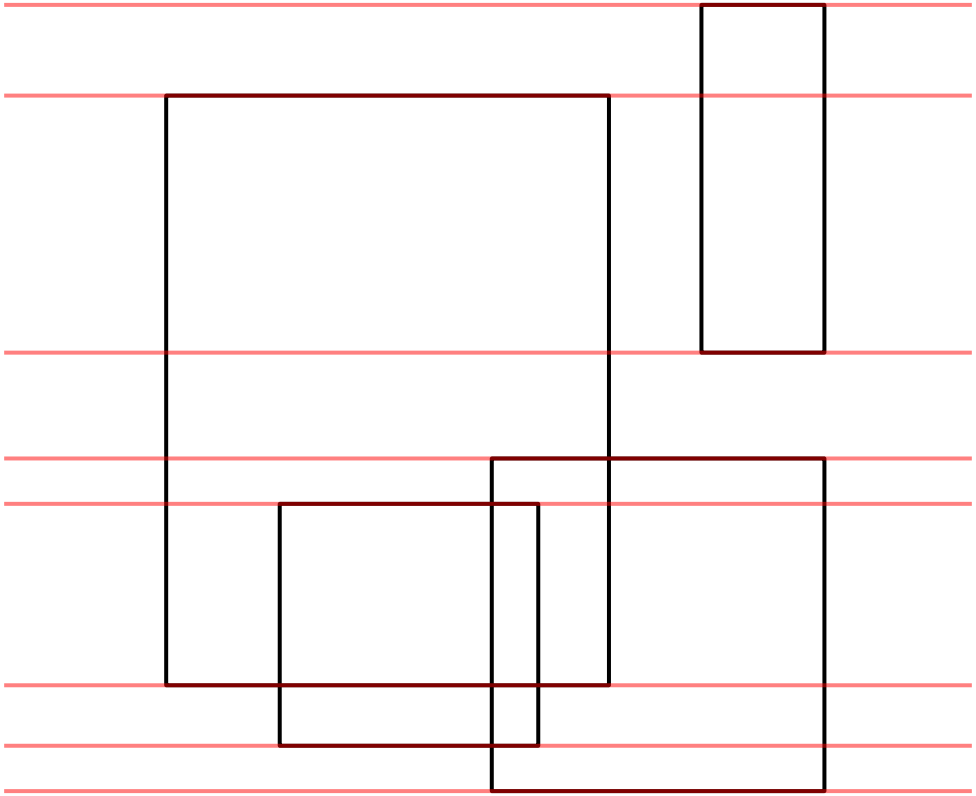


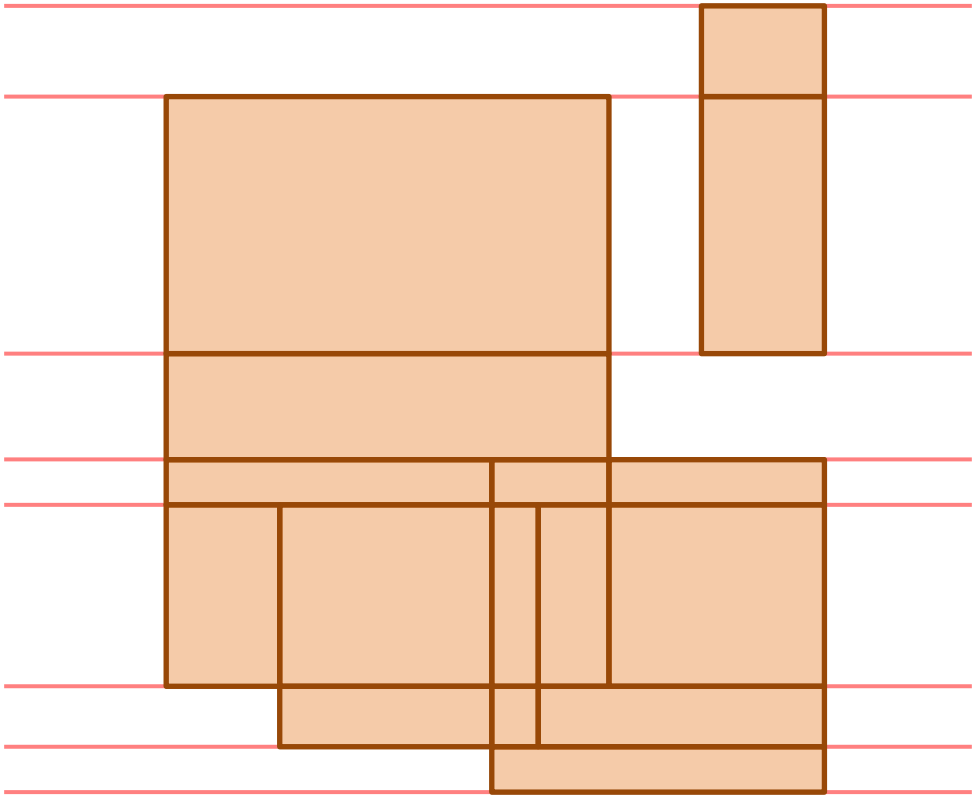


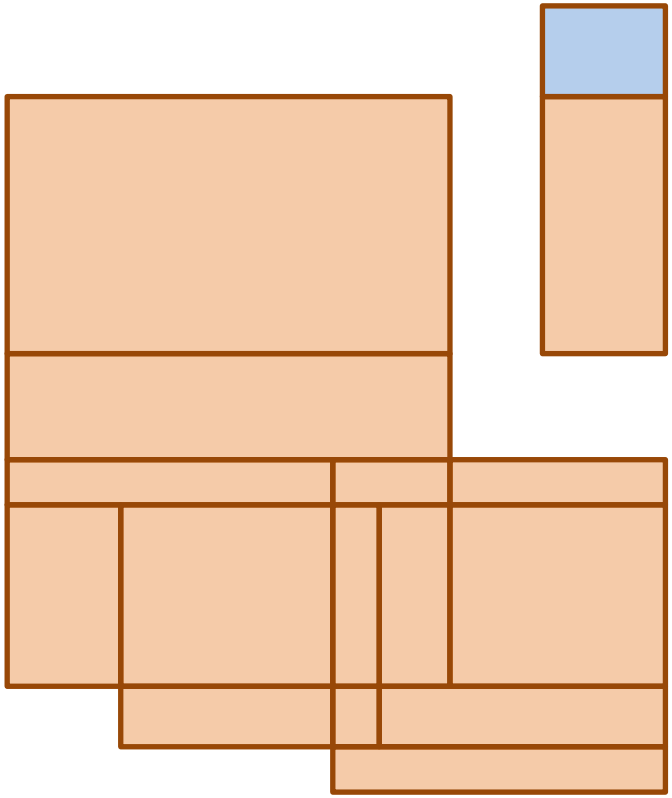


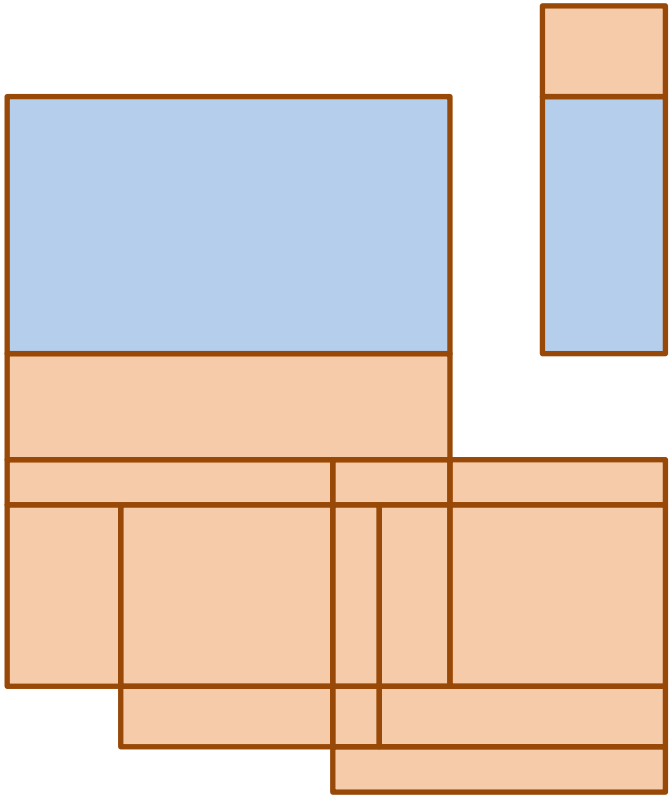


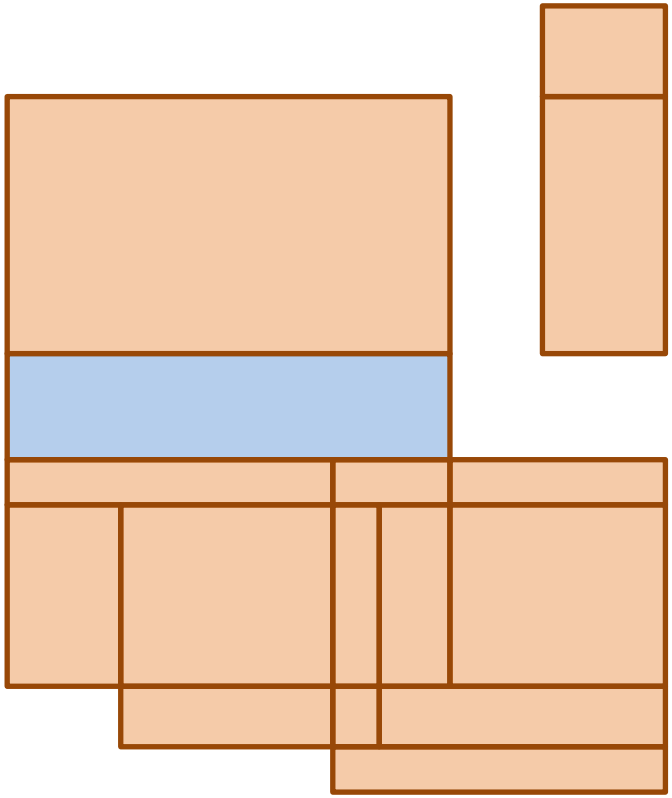


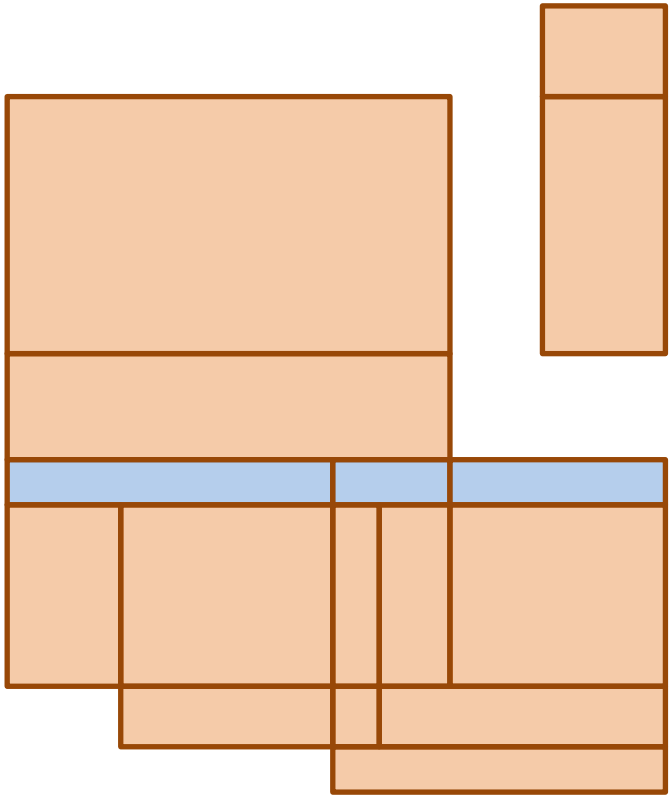


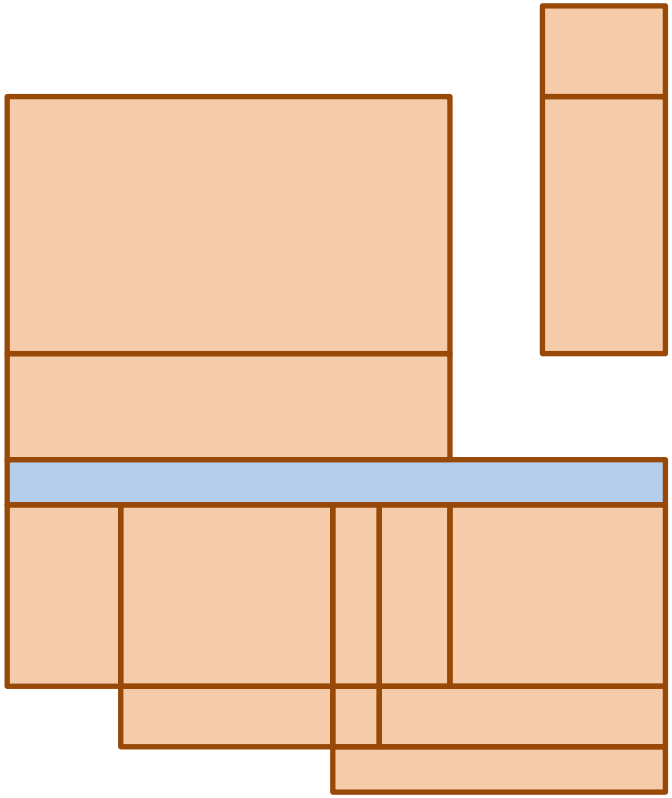


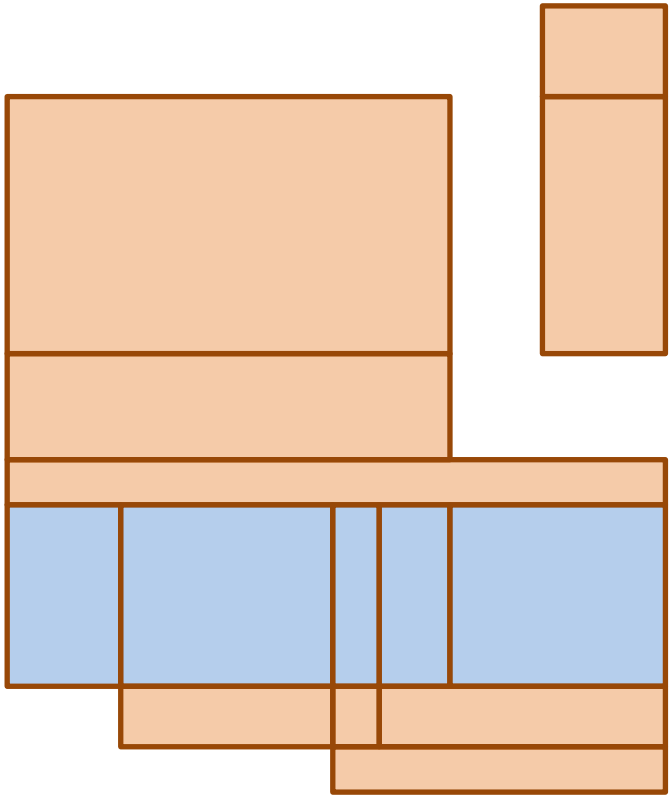


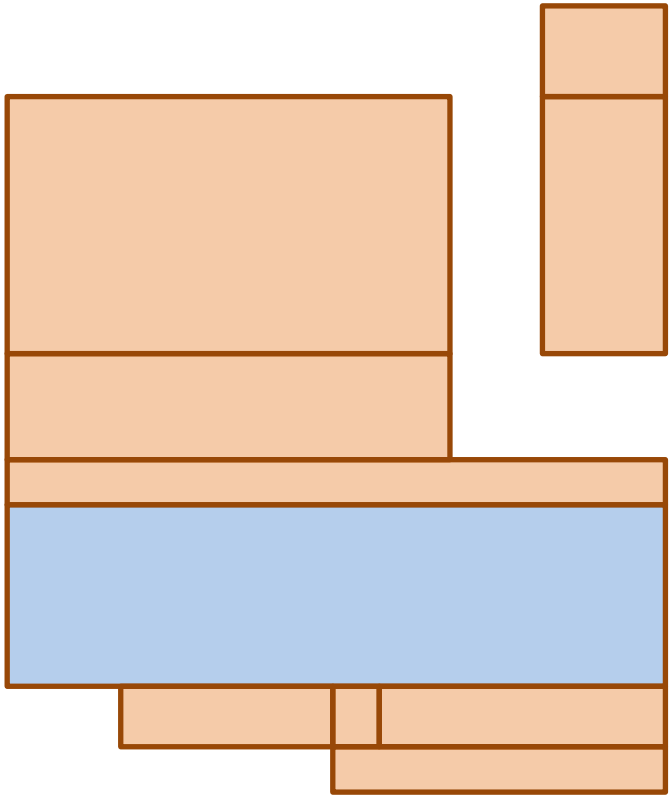


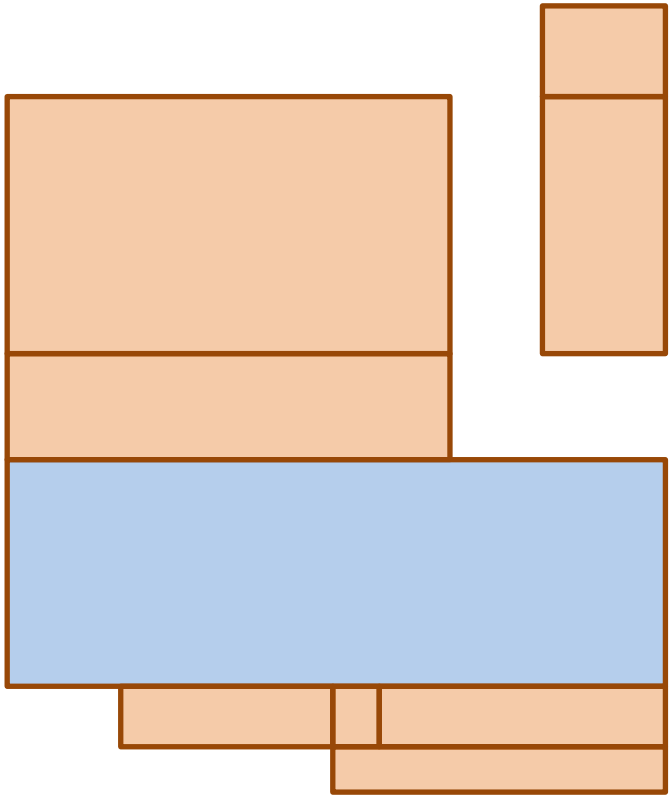


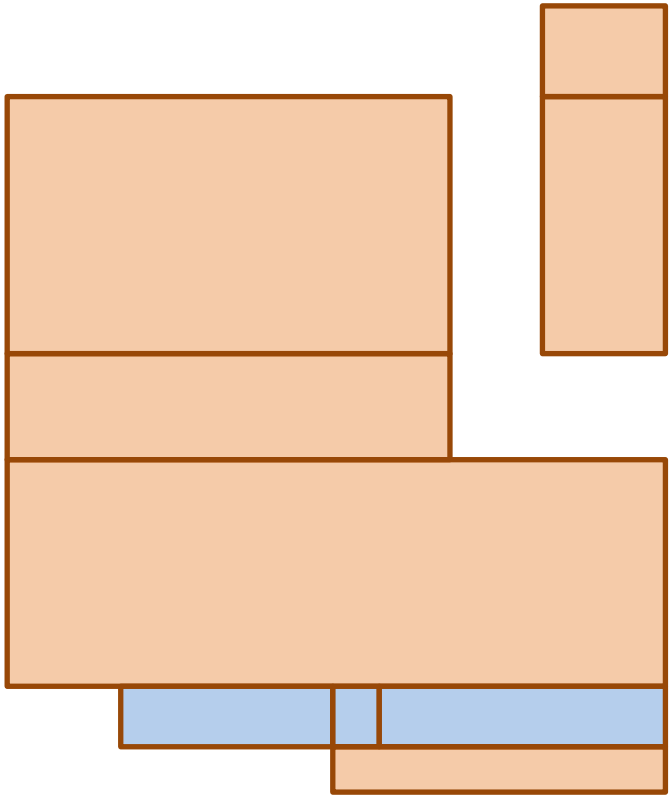


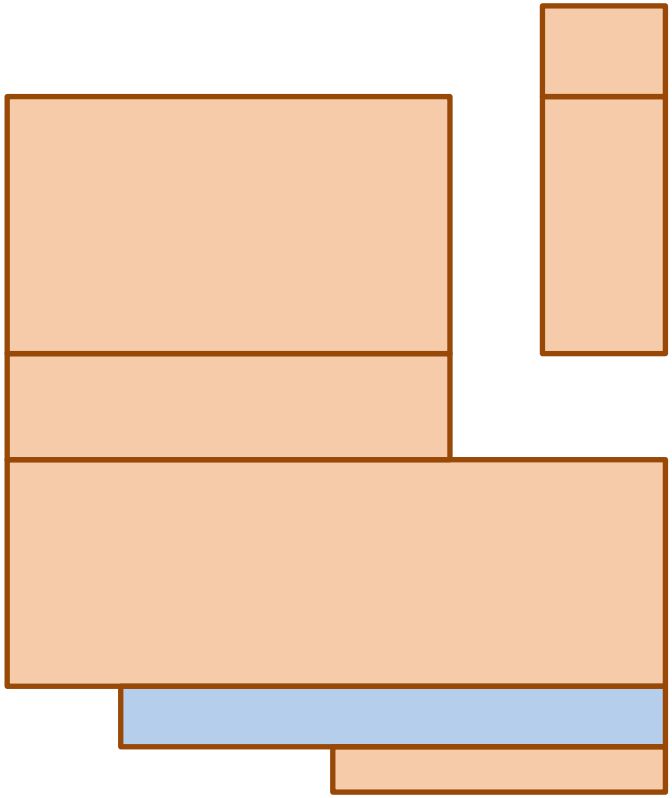


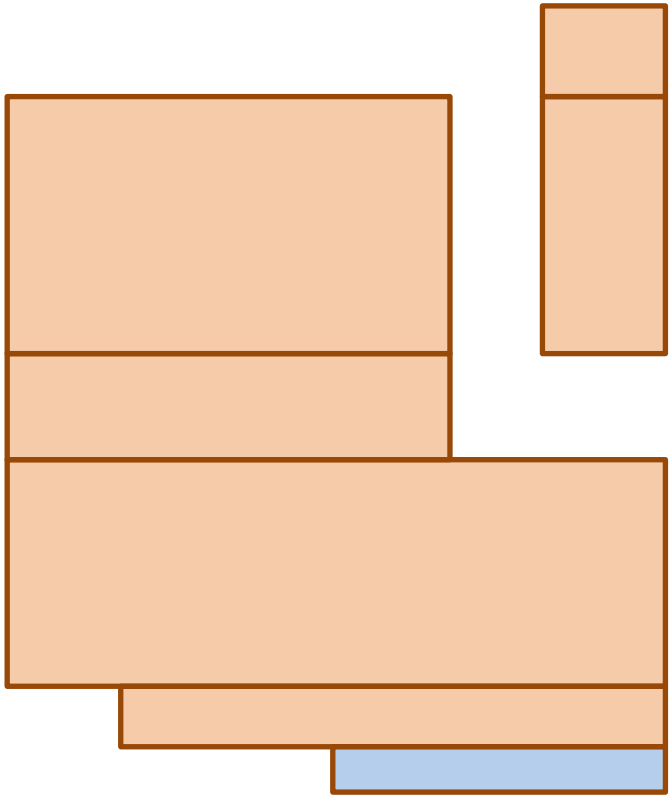


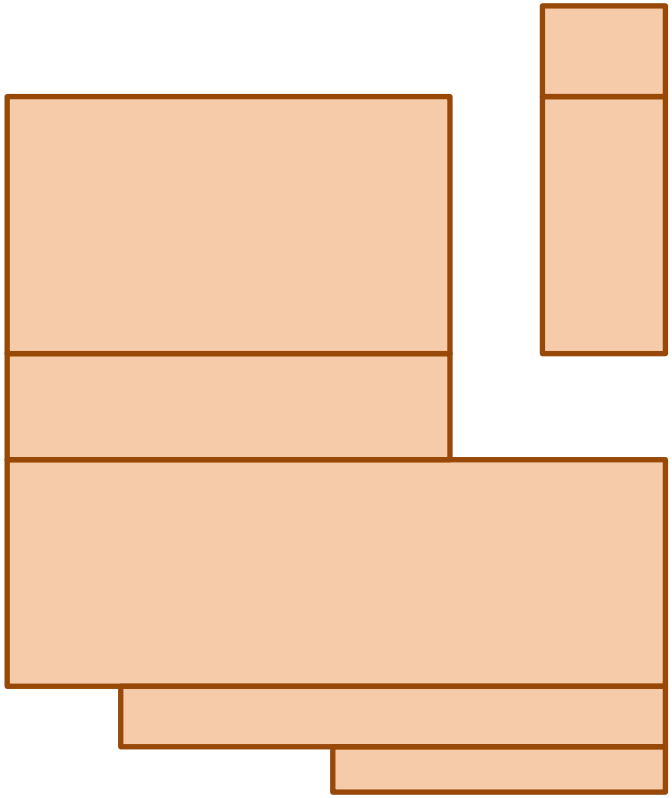


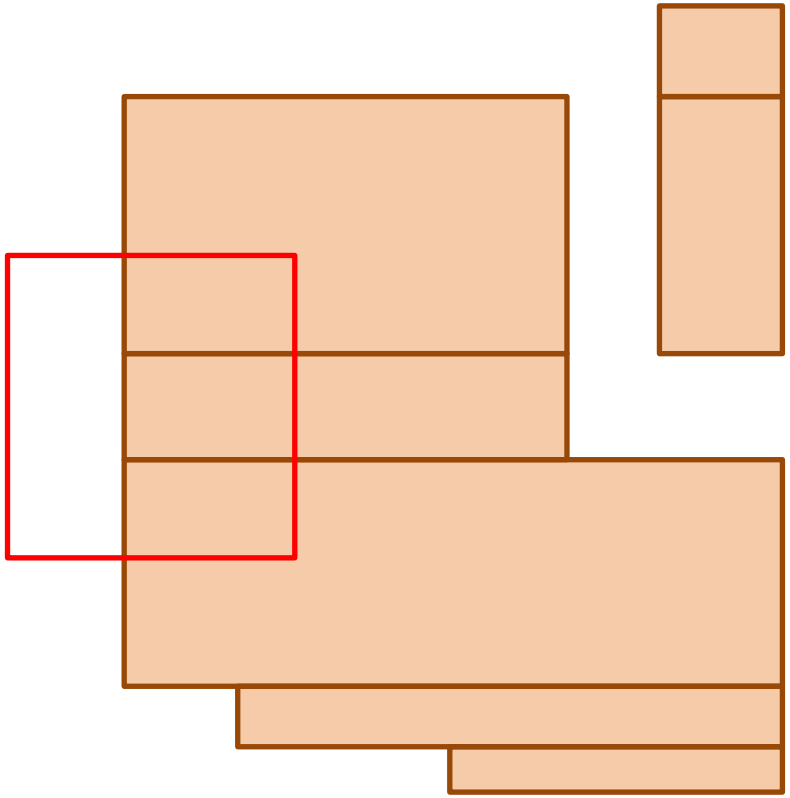


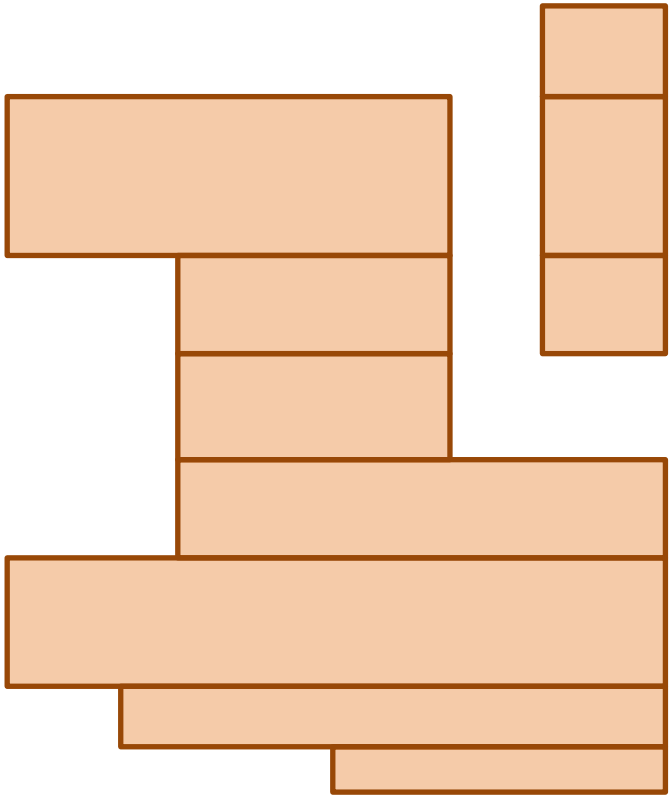


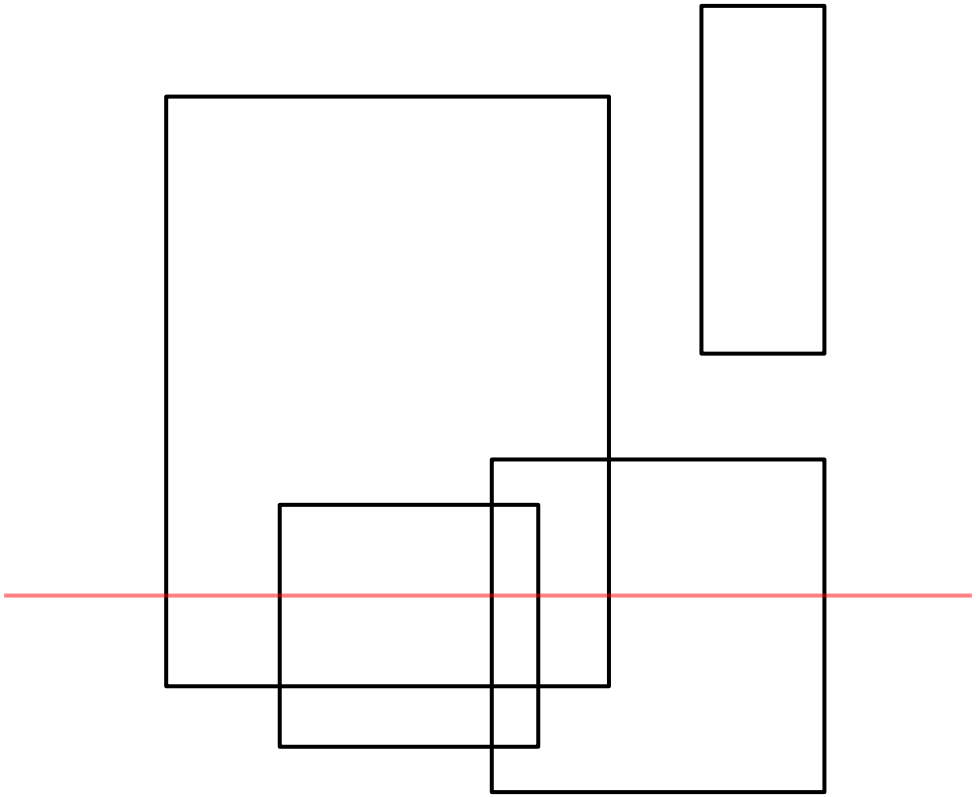


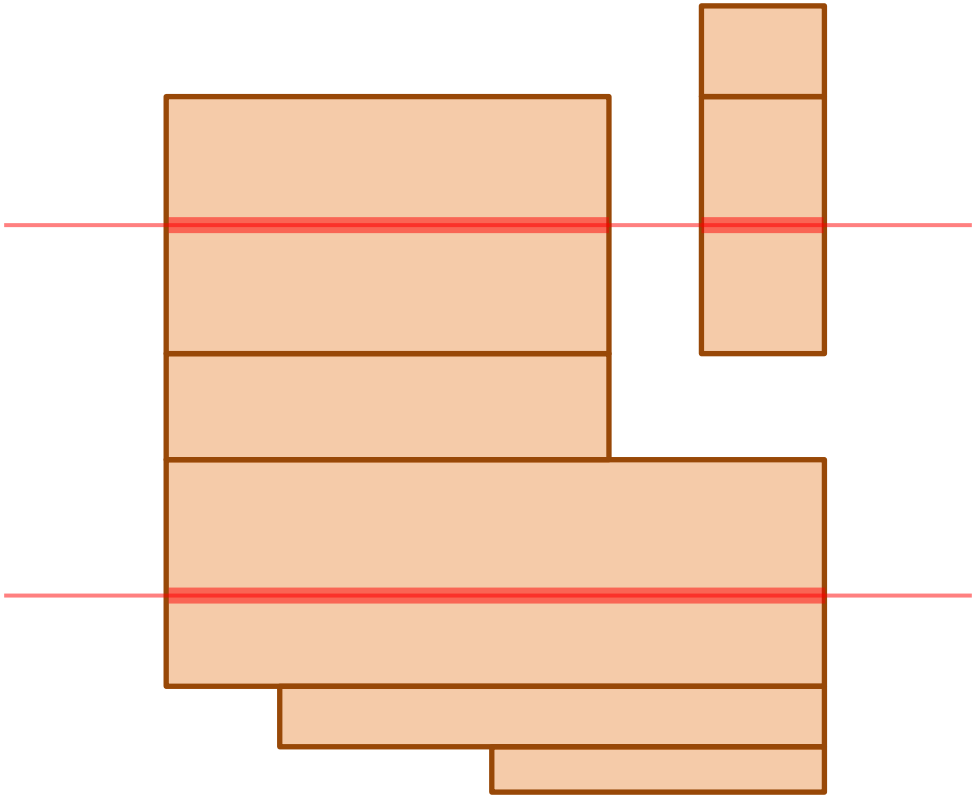


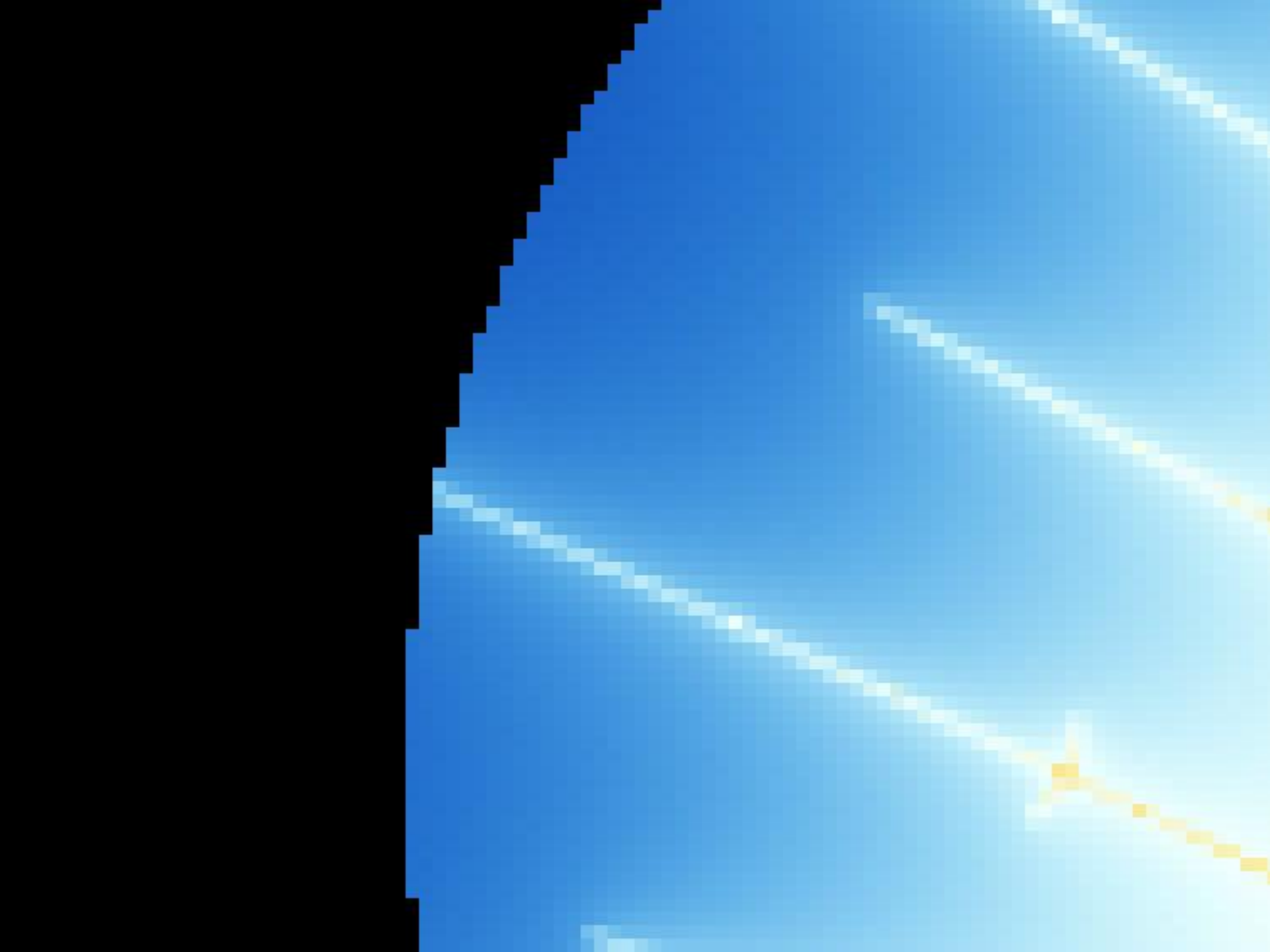












MENU

oments of creation

Graeme Murphy brings a
d energy to Romeo and
iet, dancers, set makers
d costume designer Akira...

terview: John
aters

erybody wants to be an
sider now,' John Waters.

ursed with success

st year, Jessica Chastain
s a complete unknown.
w she's in everything -
h everyone, writes STEV...

THE SYDNEY MORN

Moments of

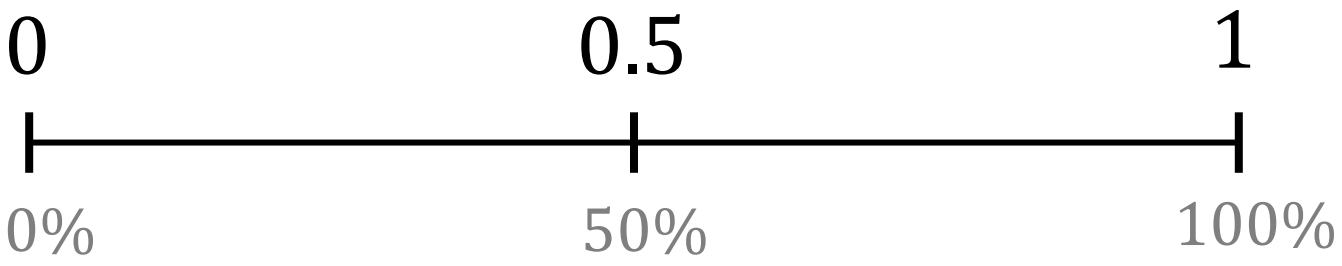
As Graeme Murphy brings
qy to *Romeo and Juliet*
kers and costume de
"It has sogawa are swept al
the fi PHILIPPA HAWKER
his

"It has to seem as if it's happe
the first time," Graeme Murph
his new ballet, *Romeo and Juli*

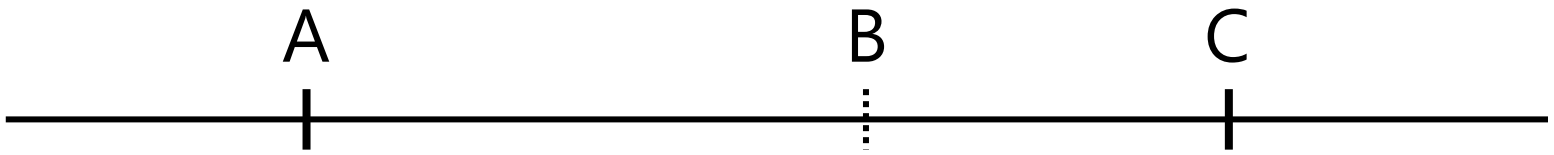
The Australian Ballet's produc
culmination of a long, complex
process involving painstaking

Единичный отрезок

$$[0,1]$$



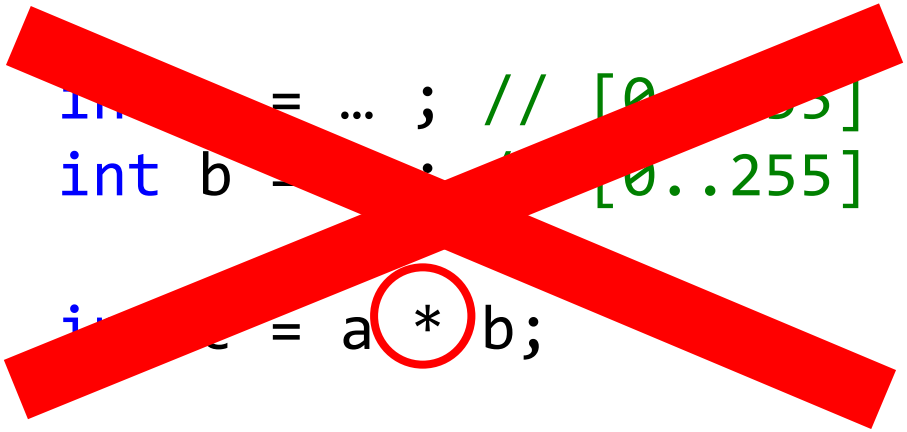
Интерполяция



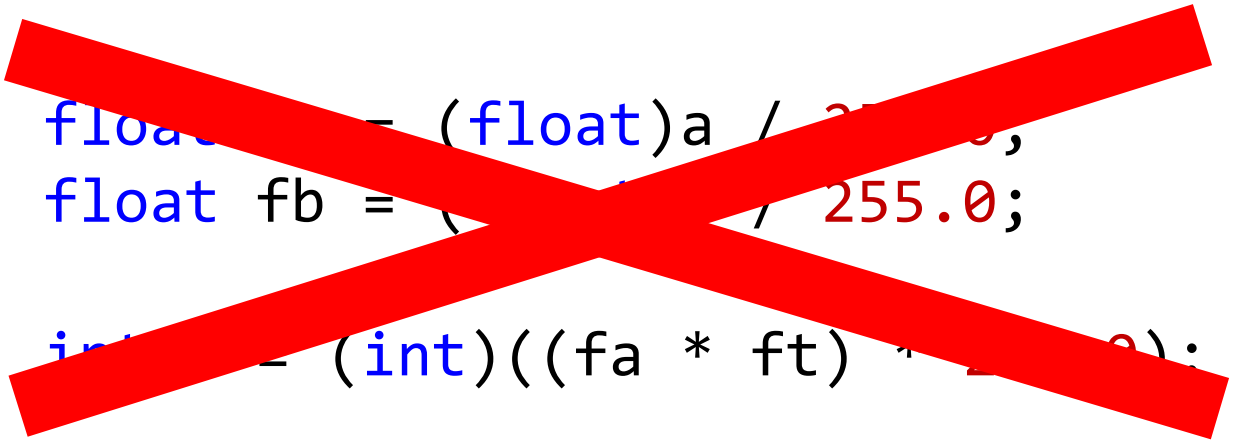
$$B = A + t(C - A)$$

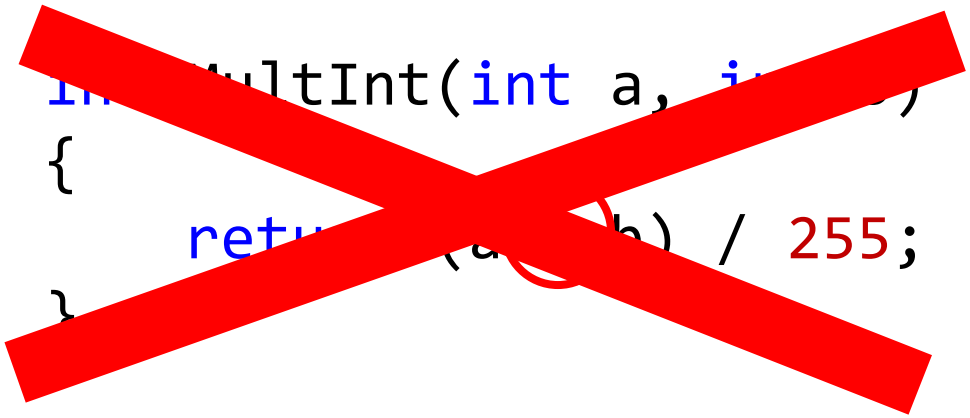
Целые числа

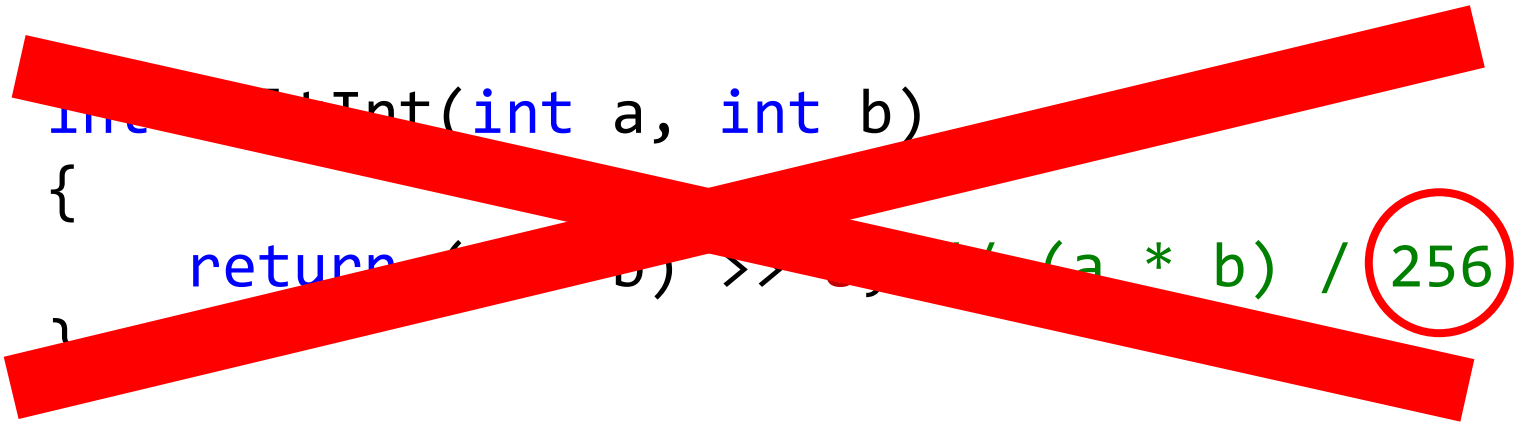
```
float a = … ; // [0..1]
float b = … ; // [0..1]

float c = a * b;
```



$$255 \times 255 = 65025$$

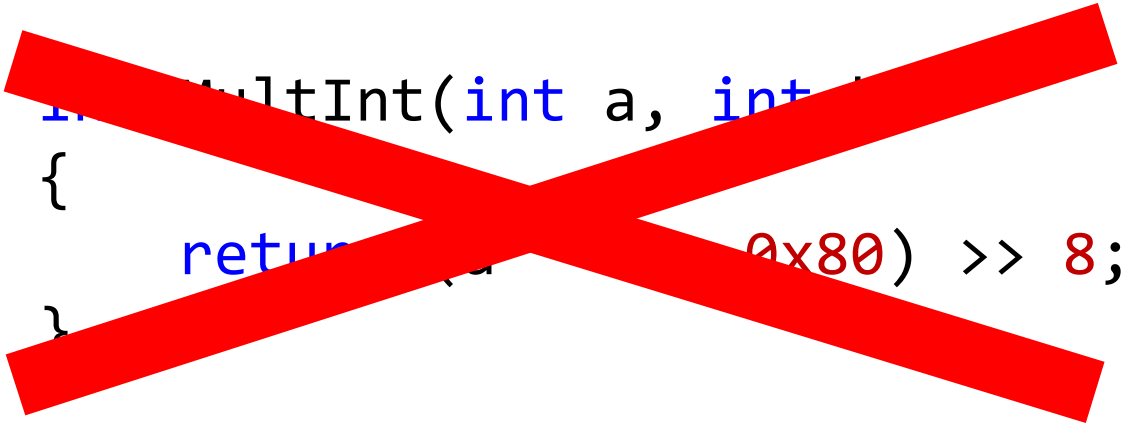

float = (float)a /
float fb = (/ 255.0;
= (int)((fa * ft)


ultInt(int a,
{
/ 255;
}



$$255 \times 255 = 65025$$

$$\frac{65025}{256} = 254.00390625$$



$$255 \times 255 + 128 = 65153$$

$$\frac{65153}{256} = 254.50390625$$

```
int MultInt(int a, int b)
{
    int temp = a * b + 0x80;
    return (temp + (temp >> 8)) >> 8;
}
```

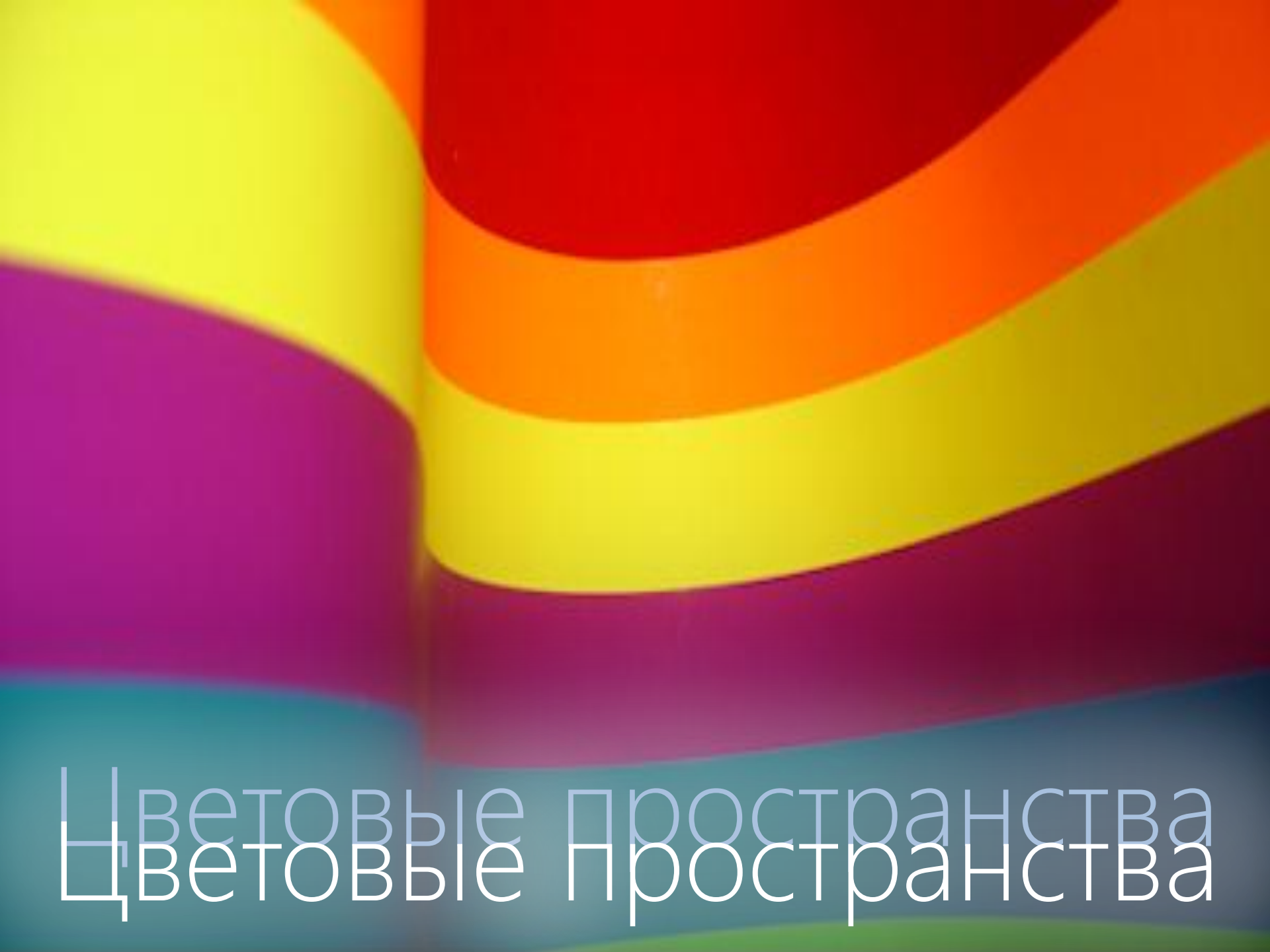
$$\frac{\frac{65153}{256} + 65153}{256} = 255.49806\ldots$$

Многабукф

```
int a = … ; // [0..255]
int b = … ; // [0..255]
int t = … ; // [0..255]

int c = a + IntMult(b - a, t);
```

```
int p1 = … ; // 0xAARRGGBB
int p2 = … ; // 0xAARRGGBB
int t  = … ; // [0..255]

int r1 = (p1 >> 16) & 0xFF;
int g1 = (p1 >>  8) & 0xFF;
int b1 = (p1      ) & 0xFF;
int a1 = (p1 >> 24) & 0xFF;

int r2 = (p2 >> 16) & 0xFF;
int g2 = (p2 >>  8) & 0xFF;
int b2 = (p2      ) & 0xFF;
int a2 = (p2 >> 24) & 0xFF;

int r3 = r1 + MultInt(r2 - r1, t);
int g3 = g1 + MultInt(g2 - g1, t);
int b3 = b1 + MultInt(b2 - b1, t);
int a3 = a1 + MultInt(a2 - a1, t);

int p3 =
      (r3 << 16)
    | (g3 <<  8)
    | (b3      )
    | (a3 << 24);
```

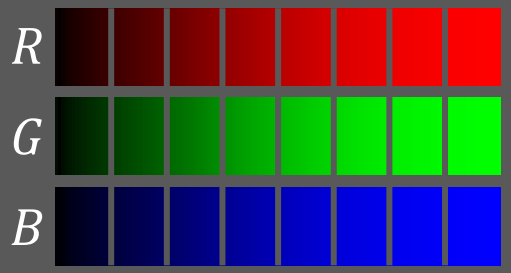
```
union pix8
{
    int RGBA;
    struct { byte R, G, B, A };
};

//
// ...
//

pix8& p1 = (pix8&)… ; // 0xAARRGGBB
pix8& p2 = (pix8&)… ; // 0xAARRGGBB
int t = … ; // [0..255]

pix8 p3;

p3.R = p1.R + MultInt(p2.R - p1.R, t);
p3.G = p1.G + MultInt(p2.G - p1.G, t);
p3.B = p1.B + MultInt(p2.B - p1.B, t);
p3.A = p1.A + MultInt(p2.A - p1.A, t);
```



Цветовые пространства

компонентные

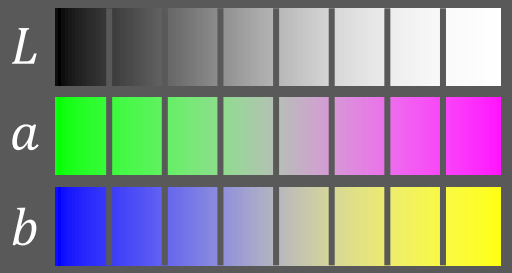
RGB



$CMYK$

хроматические

Lab



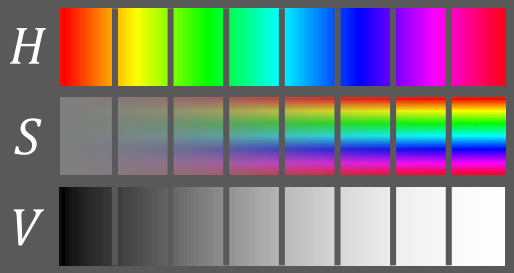
LUV

YUV

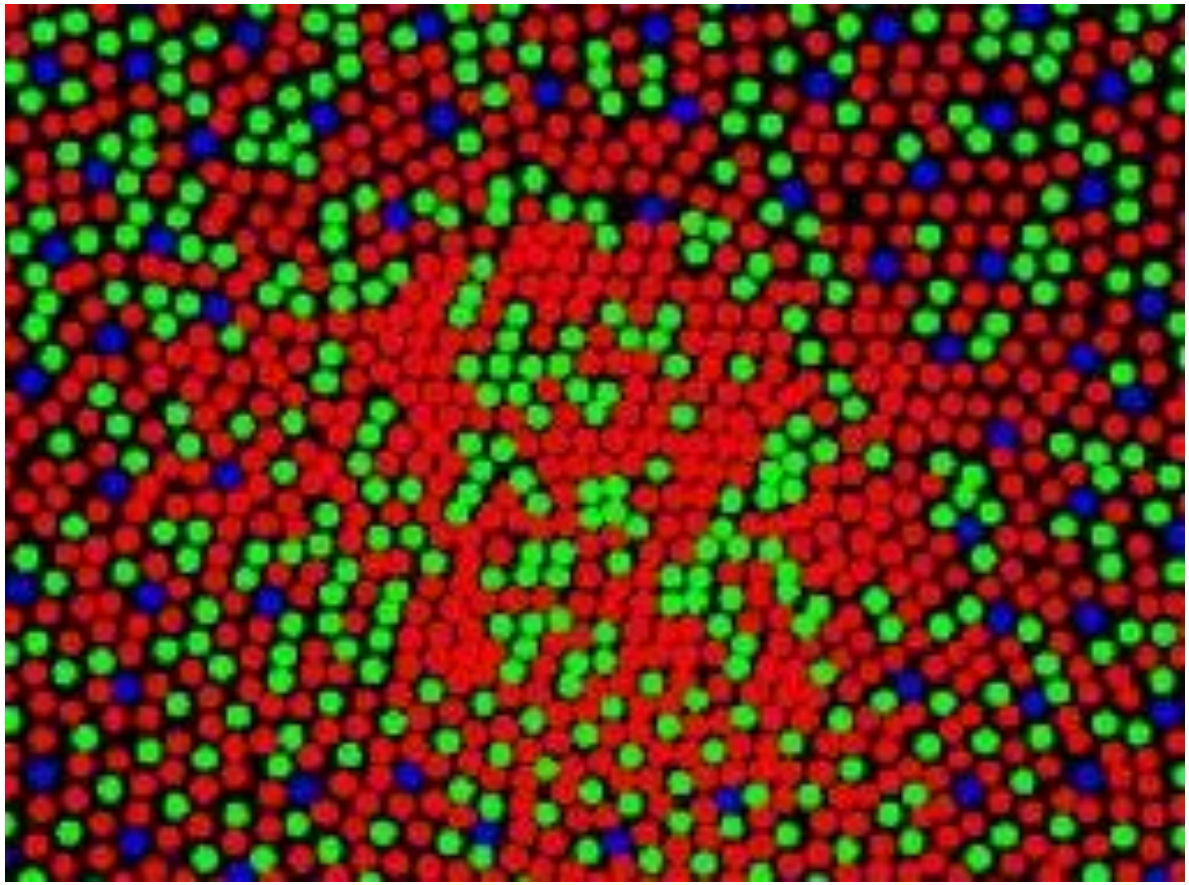
YP_BP_R

концептуальные

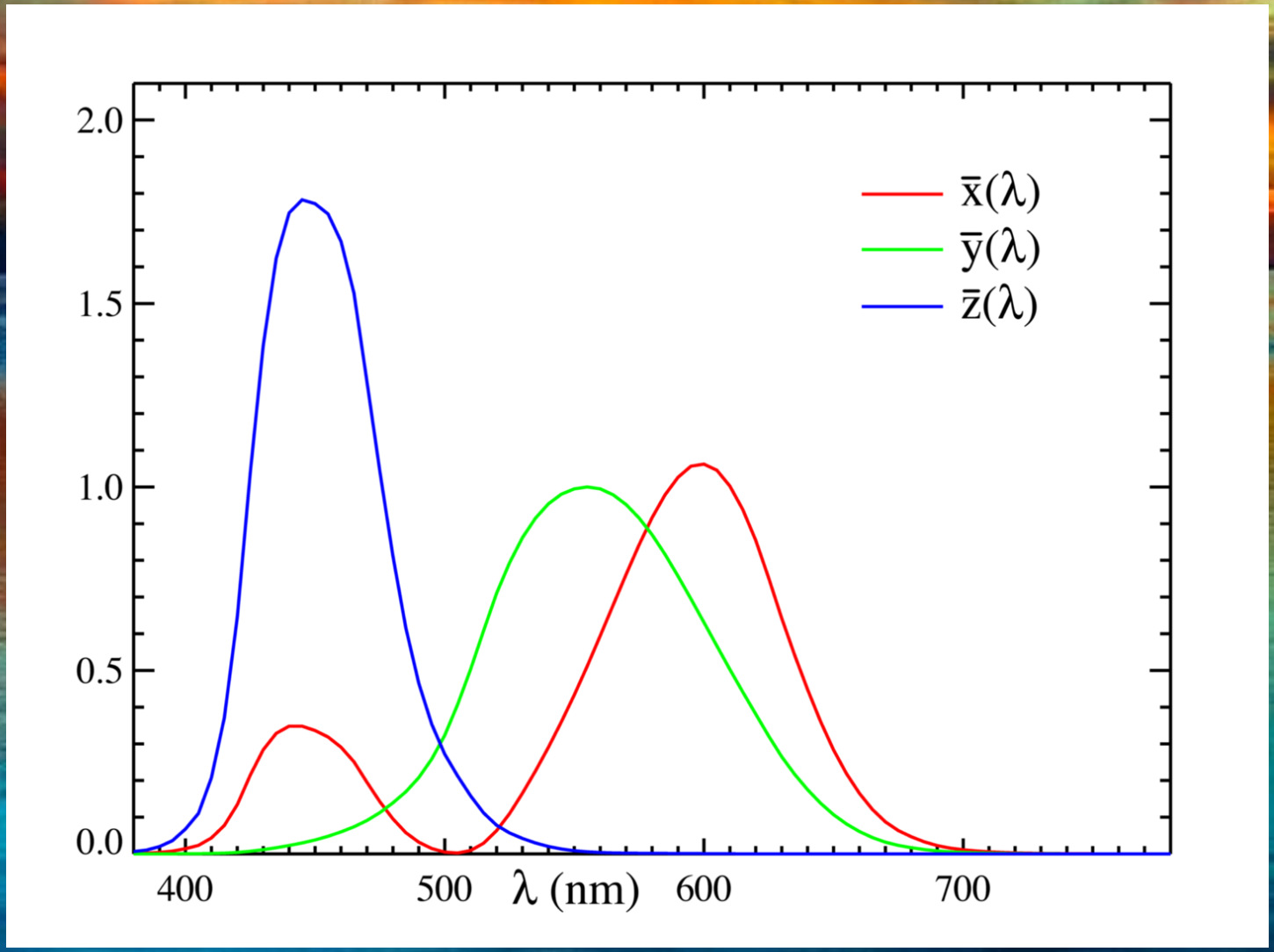
HSV



HSL

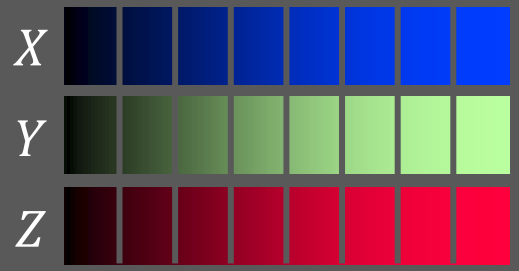


Распределение конусов



CIE 1931 2° Стандартный наблюдатель

CIE XYZ



$$x = \frac{X}{X+Y+Z} \%$$

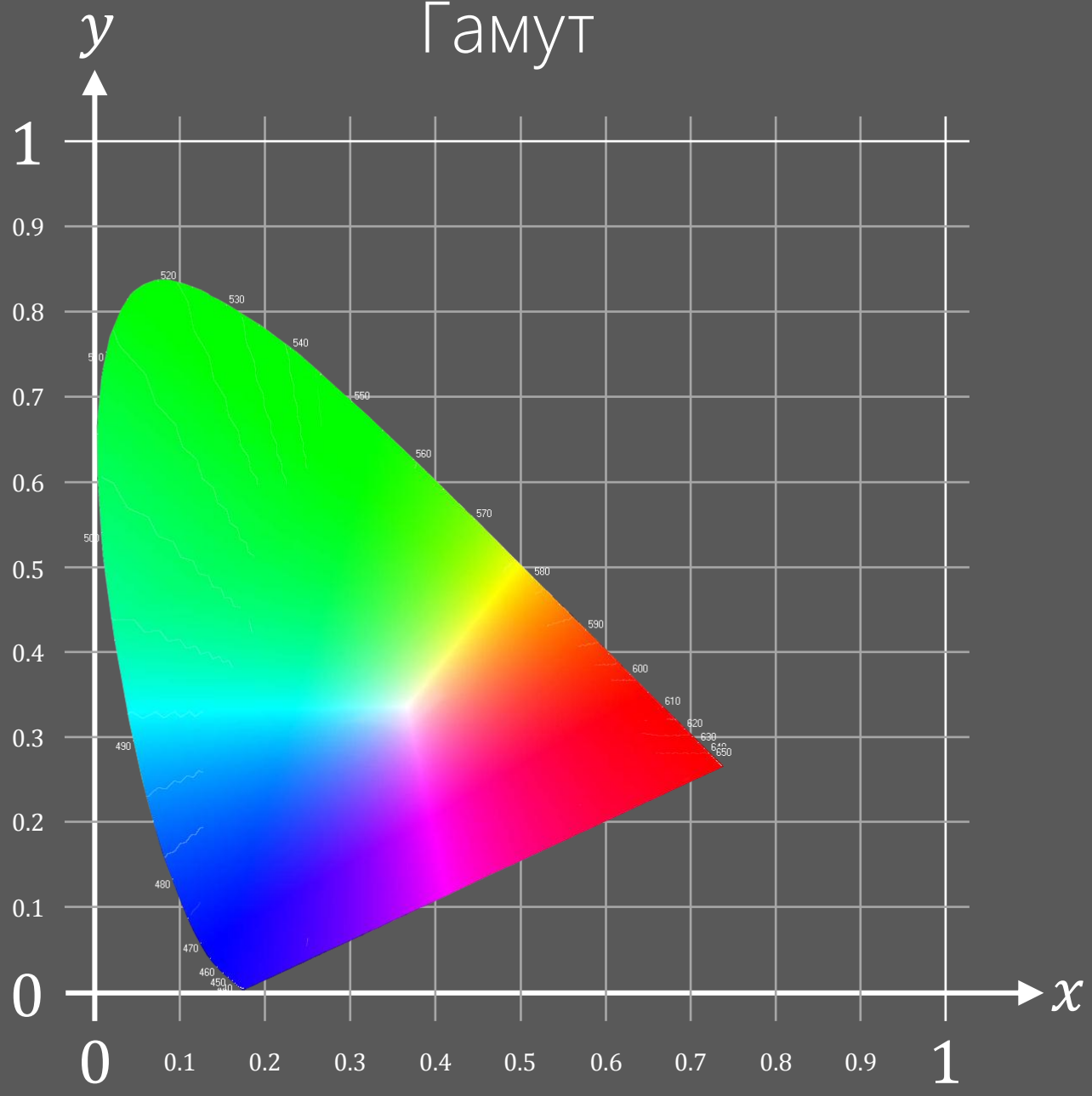
$$X = \frac{Y}{y} x$$

$$y = \frac{Y}{X+Y+Z} \%$$

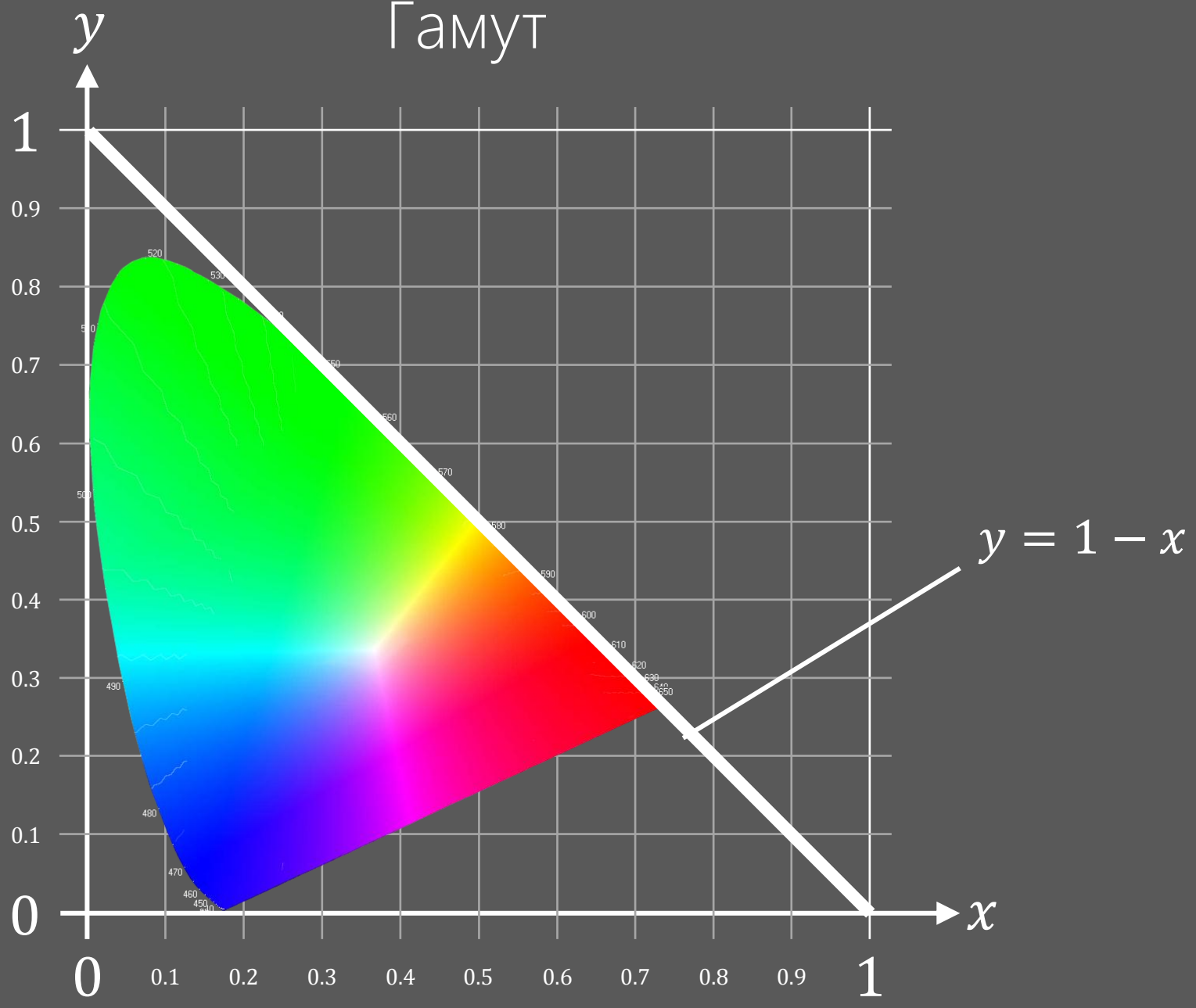
$$Z = \frac{Y}{y}(1-x-y)$$

$$z = \frac{Z}{X+Y+Z} = 1-x-y \%$$

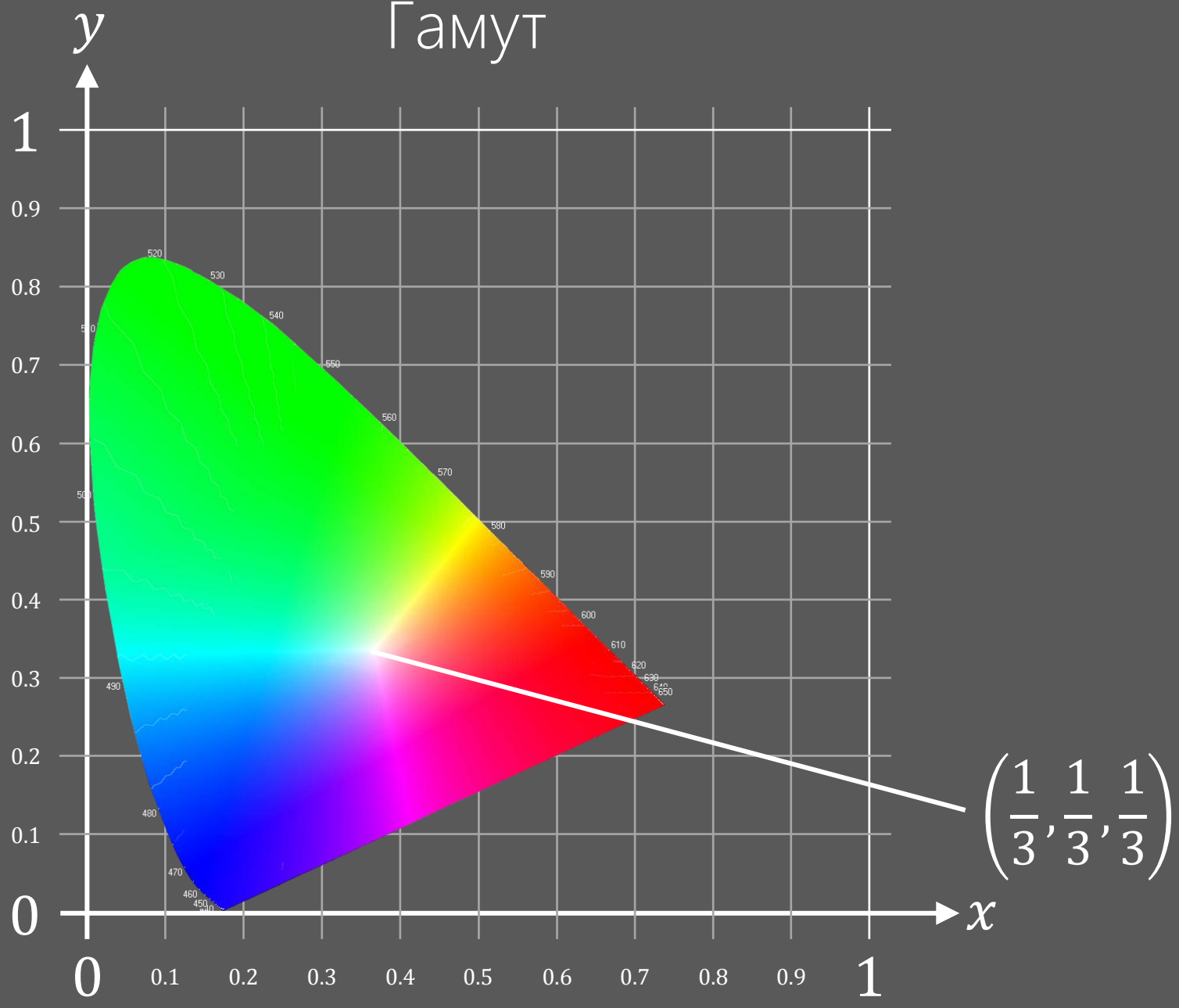
Гамут


y
x
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
520
530
540
550
560
570
580
590
600
610
620
630
650
490
480
470
460
450

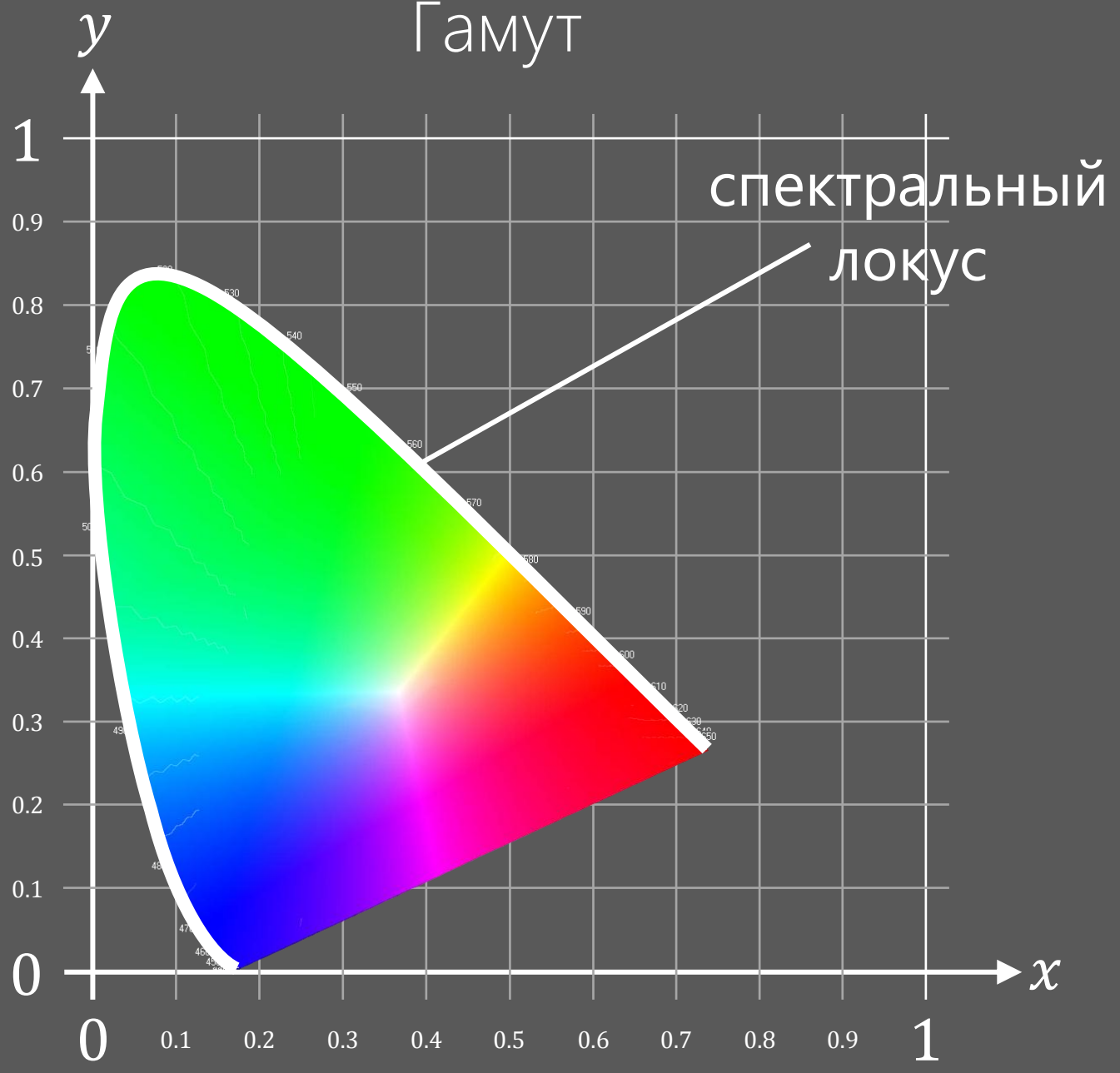
Гамут


y
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
x
y = 1 − x
520
530
560
570
580
590
600
610
620
630
490
480
470
460
450

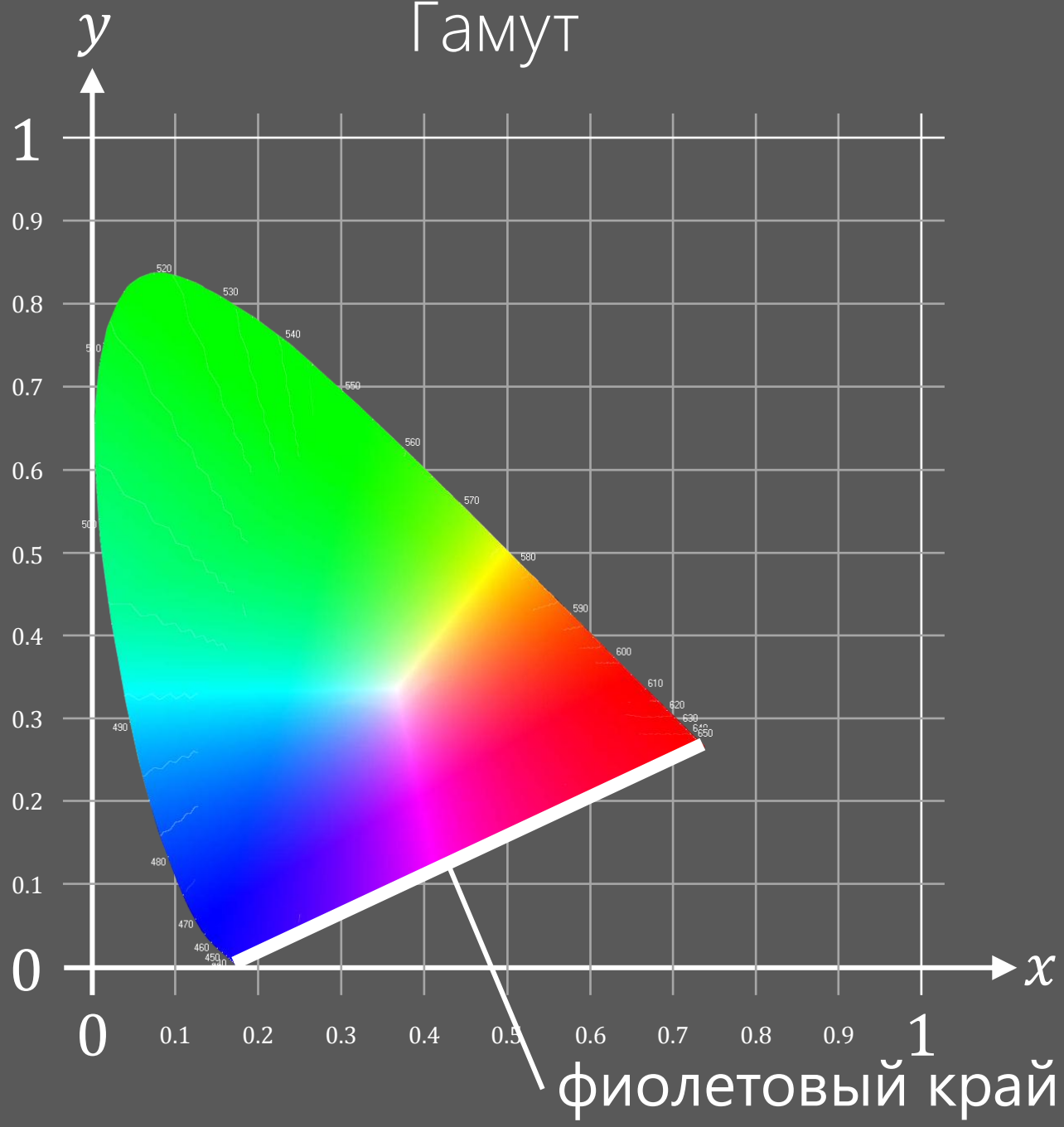
Гамут


y
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
x
520
530
540
550
560
570
580
590
600
610
620
630
640
650
490
480
470
460
450
$\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$

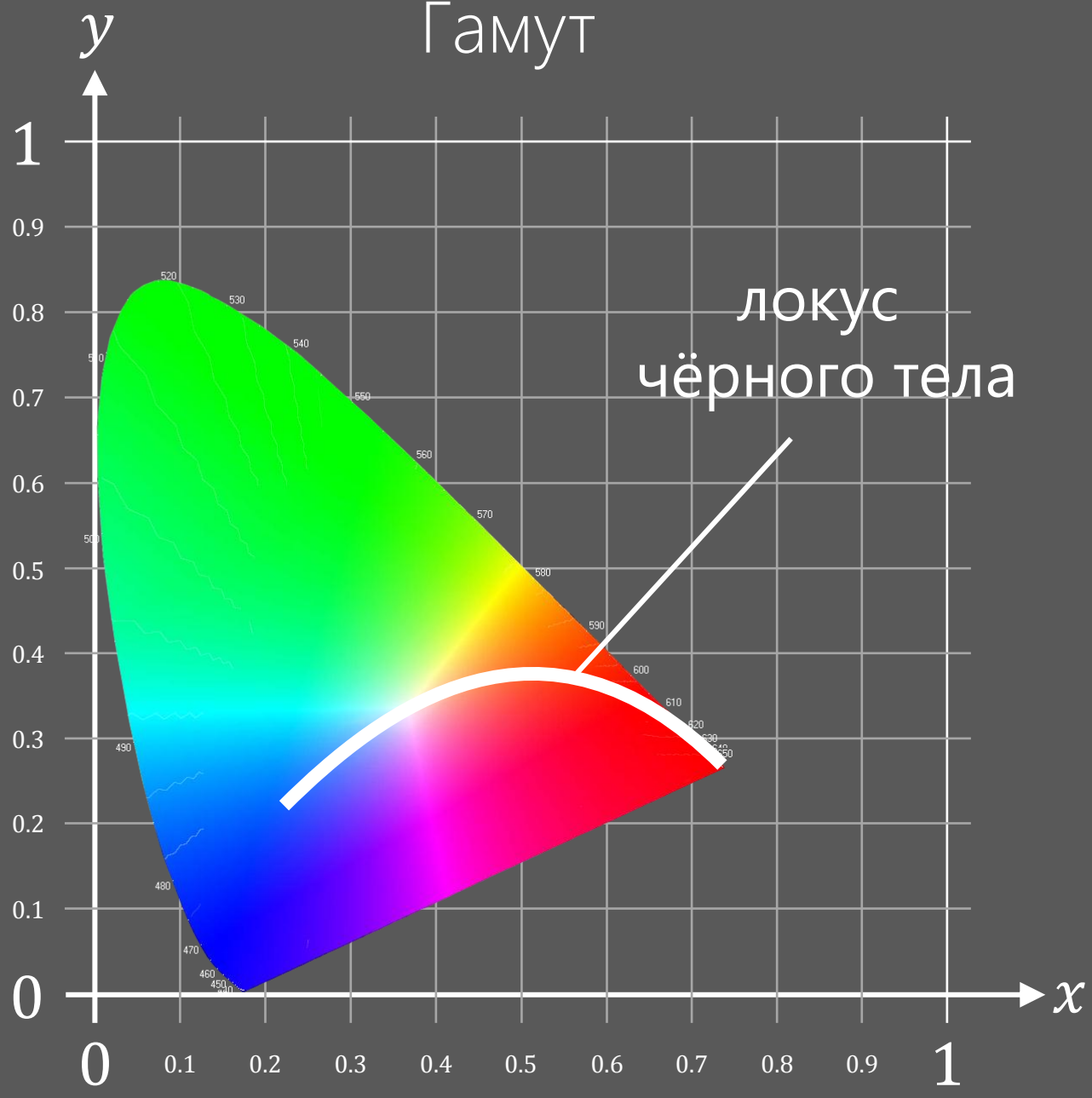
Гамут


y
x
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
спектральный
локус

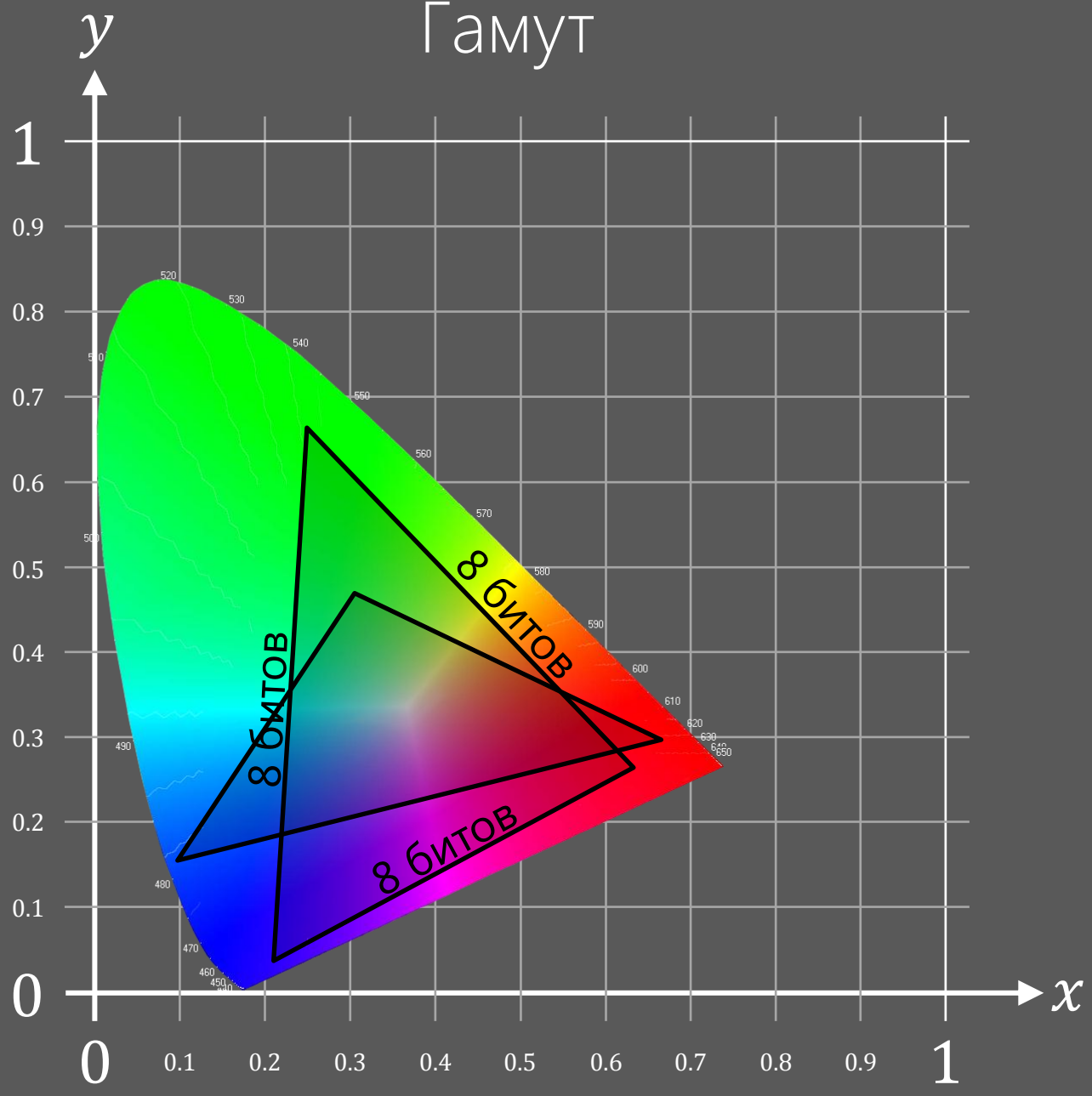
Гамут


y
x
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
520
530
540
550
560
570
580
590
600
610
620
630
640
650
500
490
480
470
460
450
фиолетовый край

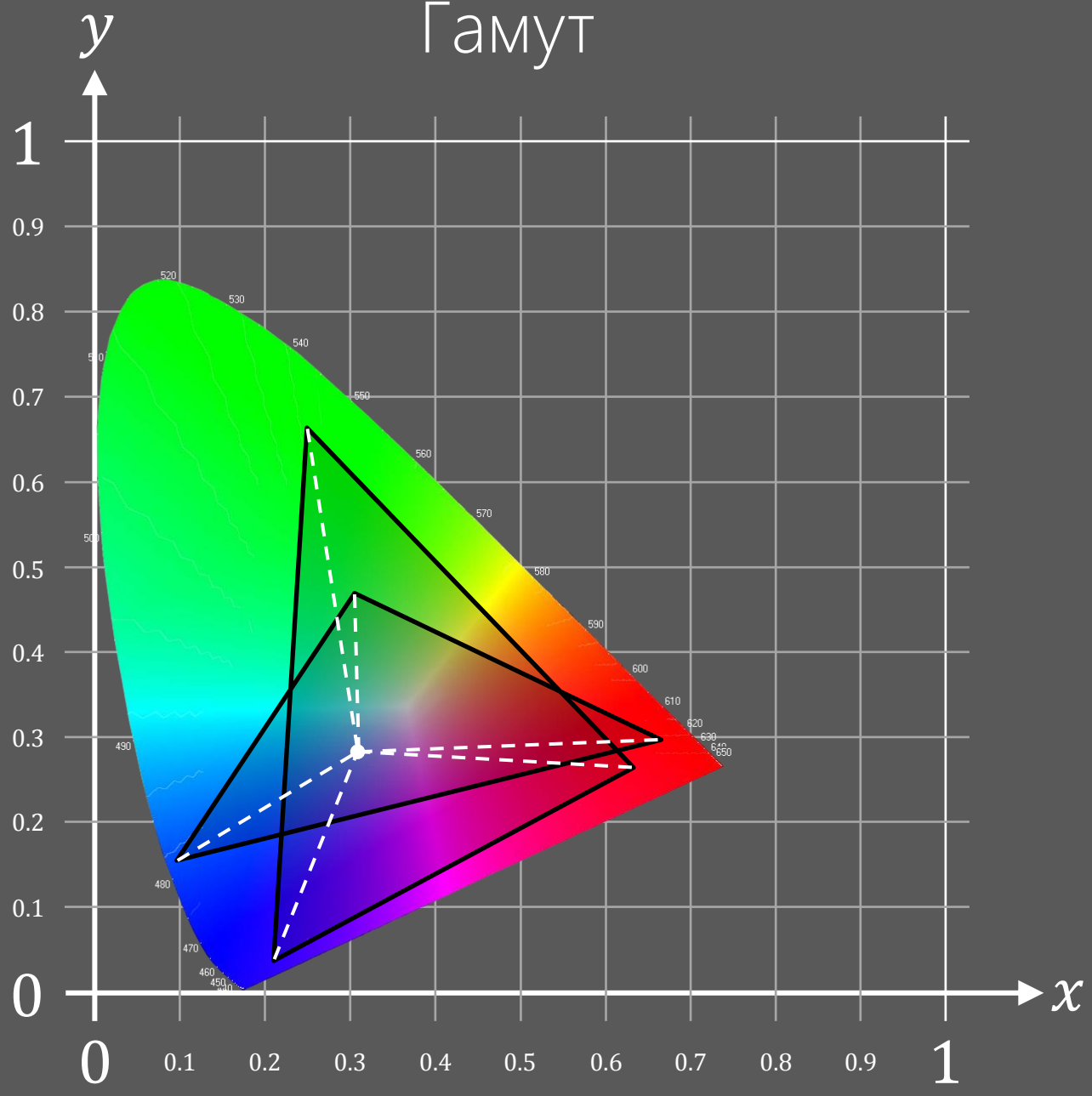
Гамут

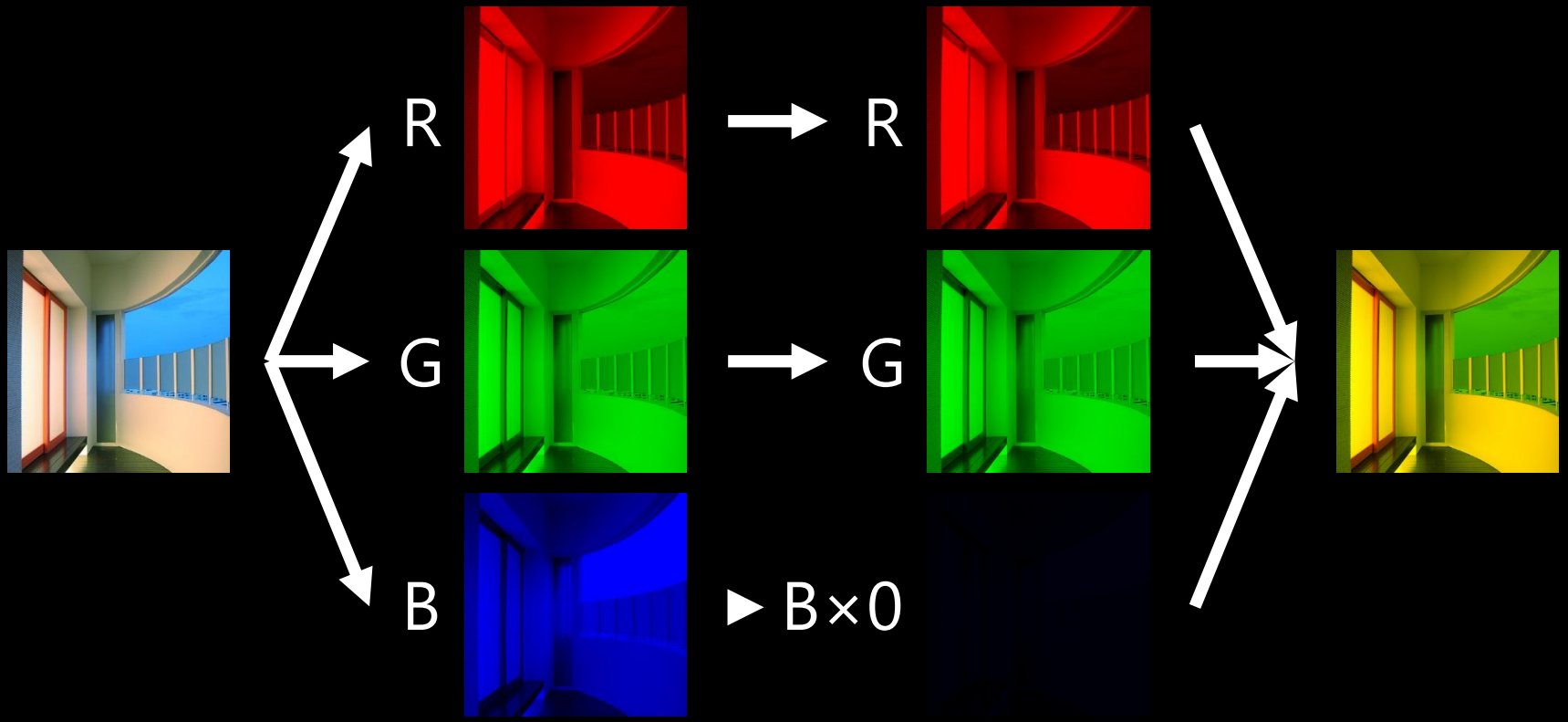


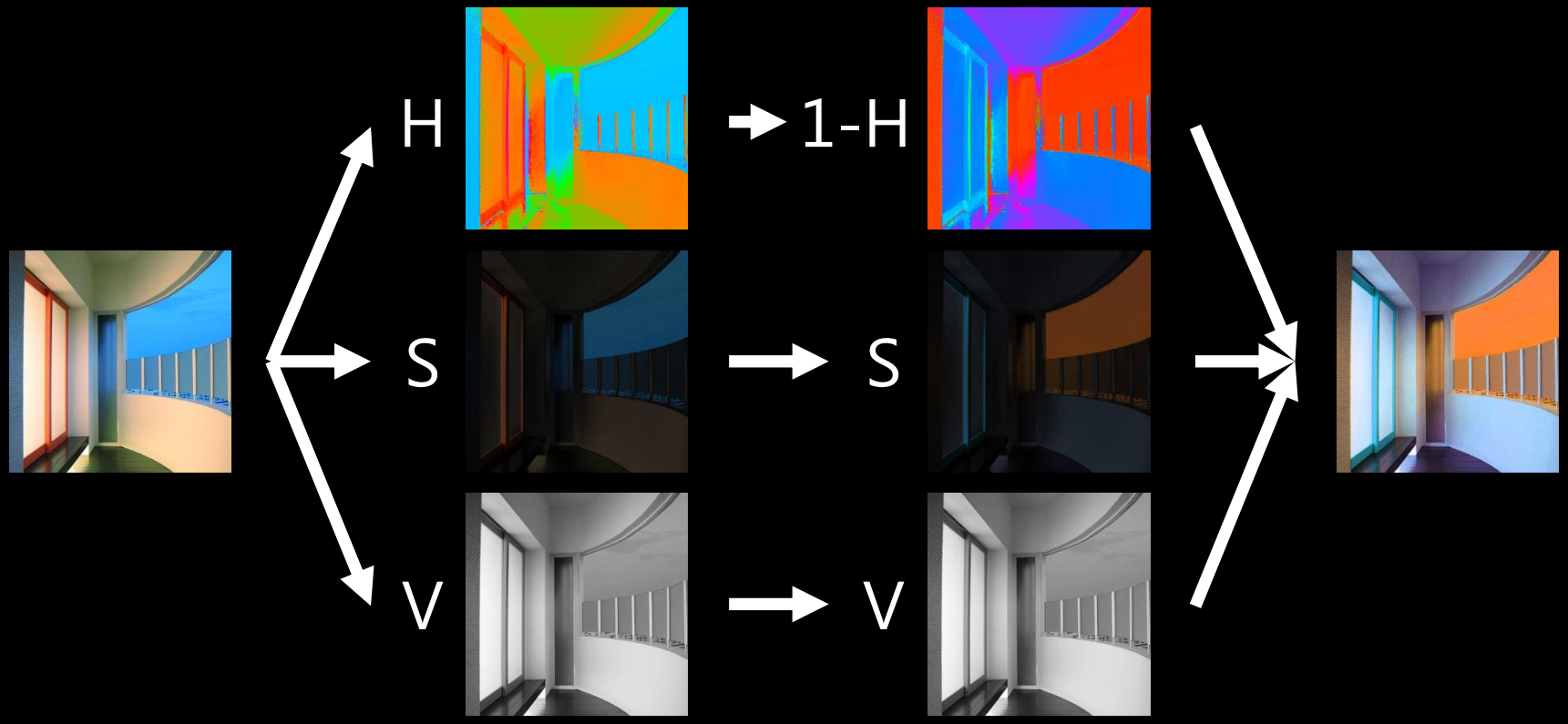
Гамут



Гамут


y
x
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
520
530
540
550
560
570
580
590
600
610
620
630
650
490
480
470
460
450


R
G
B
R
G
B×0


H
1-H
S
S
V
V


POWER
X PETS PET SHOP
BARK TWICE
Café Olé
DRINK

$t = 0$


POWER
X PETS PET SHOP
BARK TWICE
Café Olé
DRINK

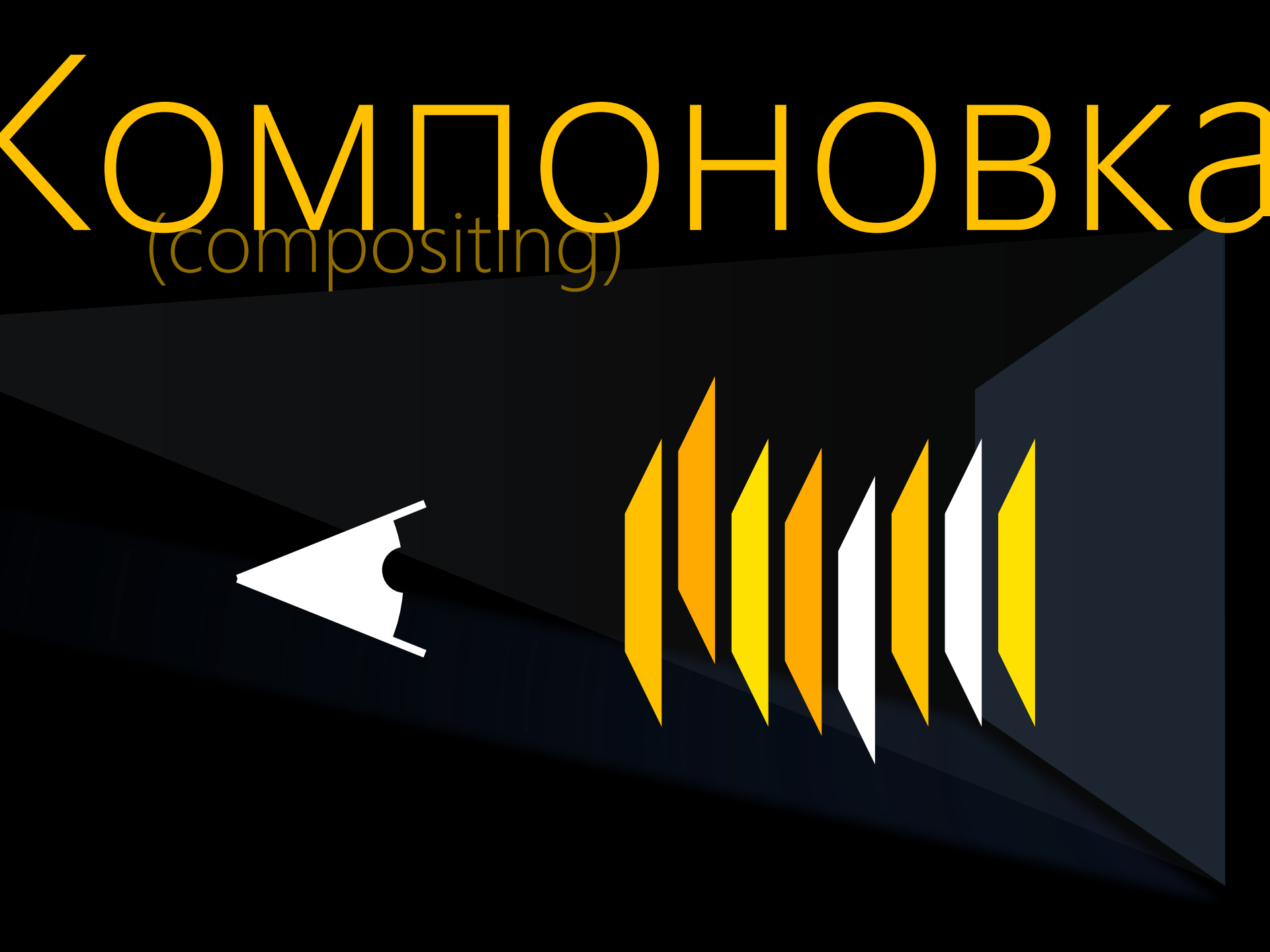
$t = 1$

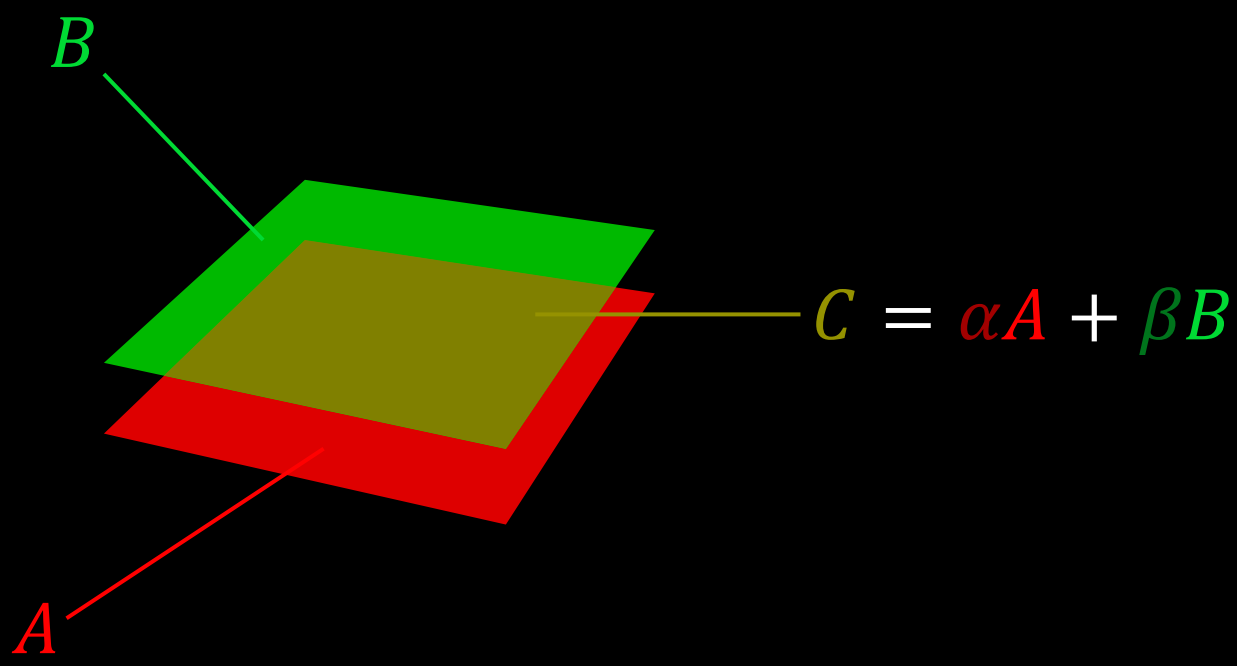

POWER
X PETS PET SHOP
BARK TWICE
Café Olé
DRINK

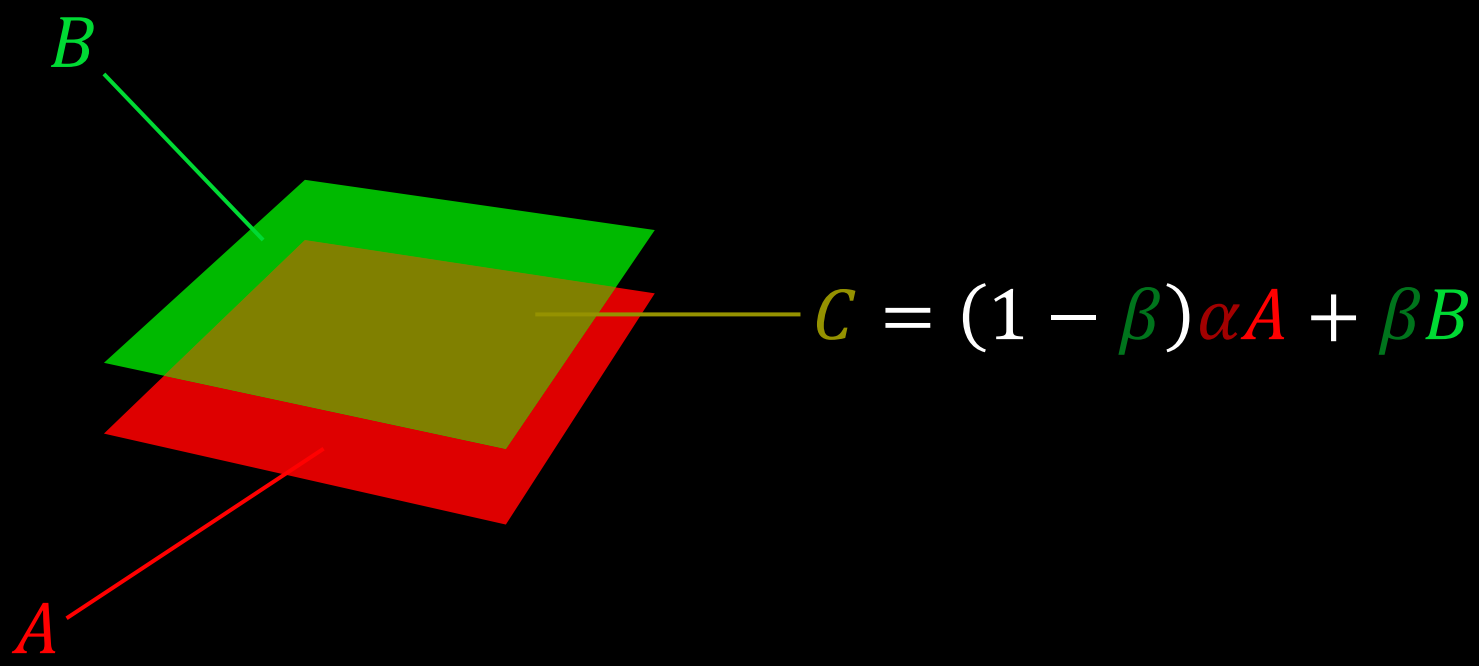
$t = 2$

Компоновка

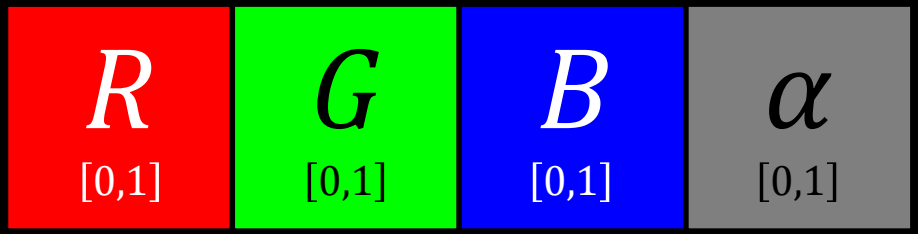
(compositing)



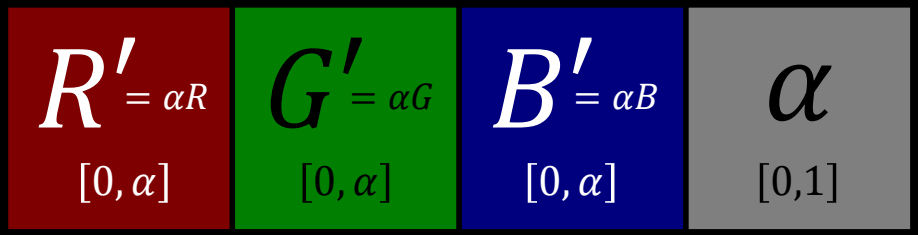

B
C = αA + βB
A


B
A
C = (1 − β)αA + βB

Перемножение

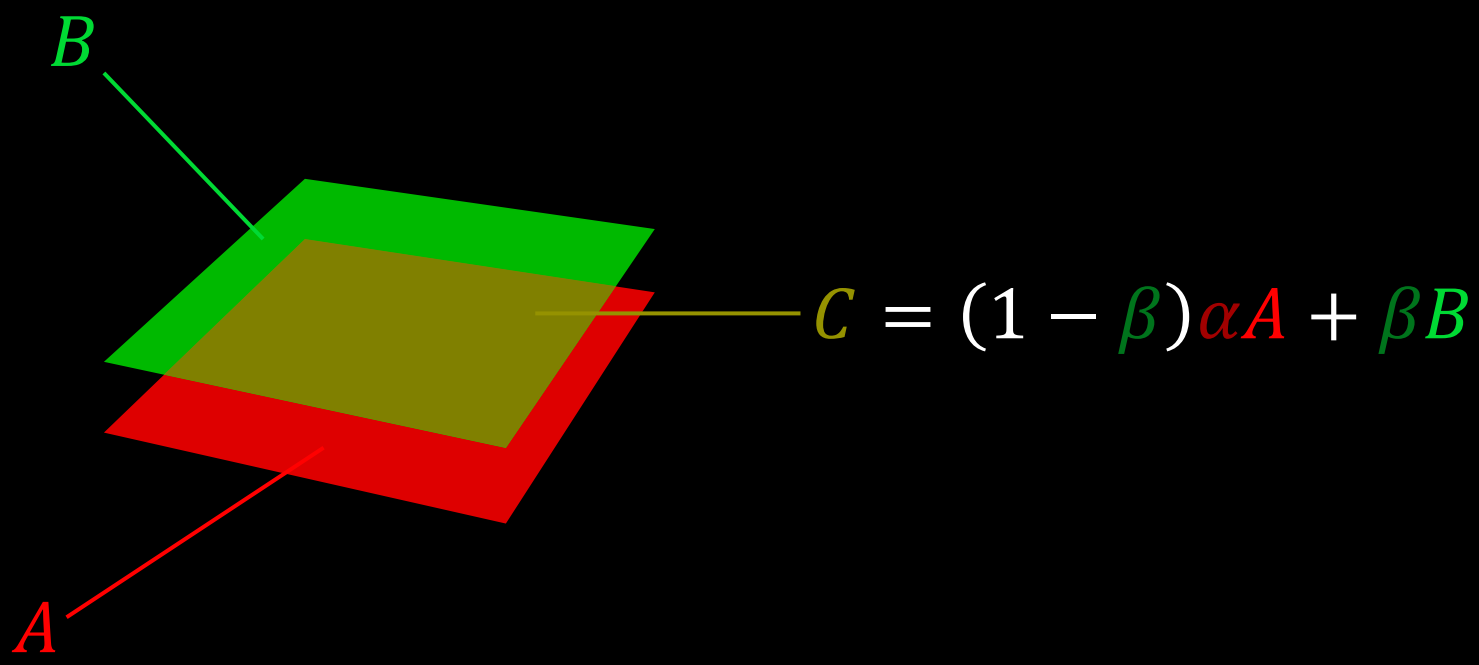


неперемноженный
(unpremultiplied)

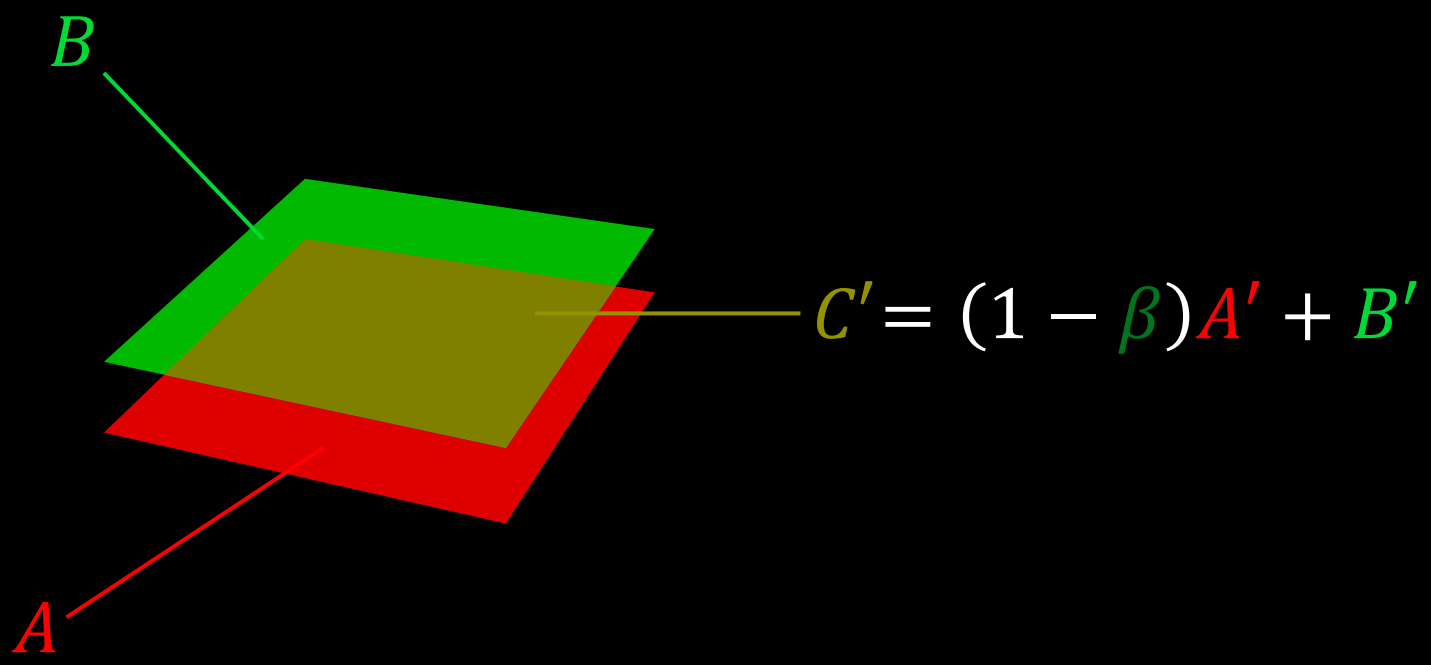


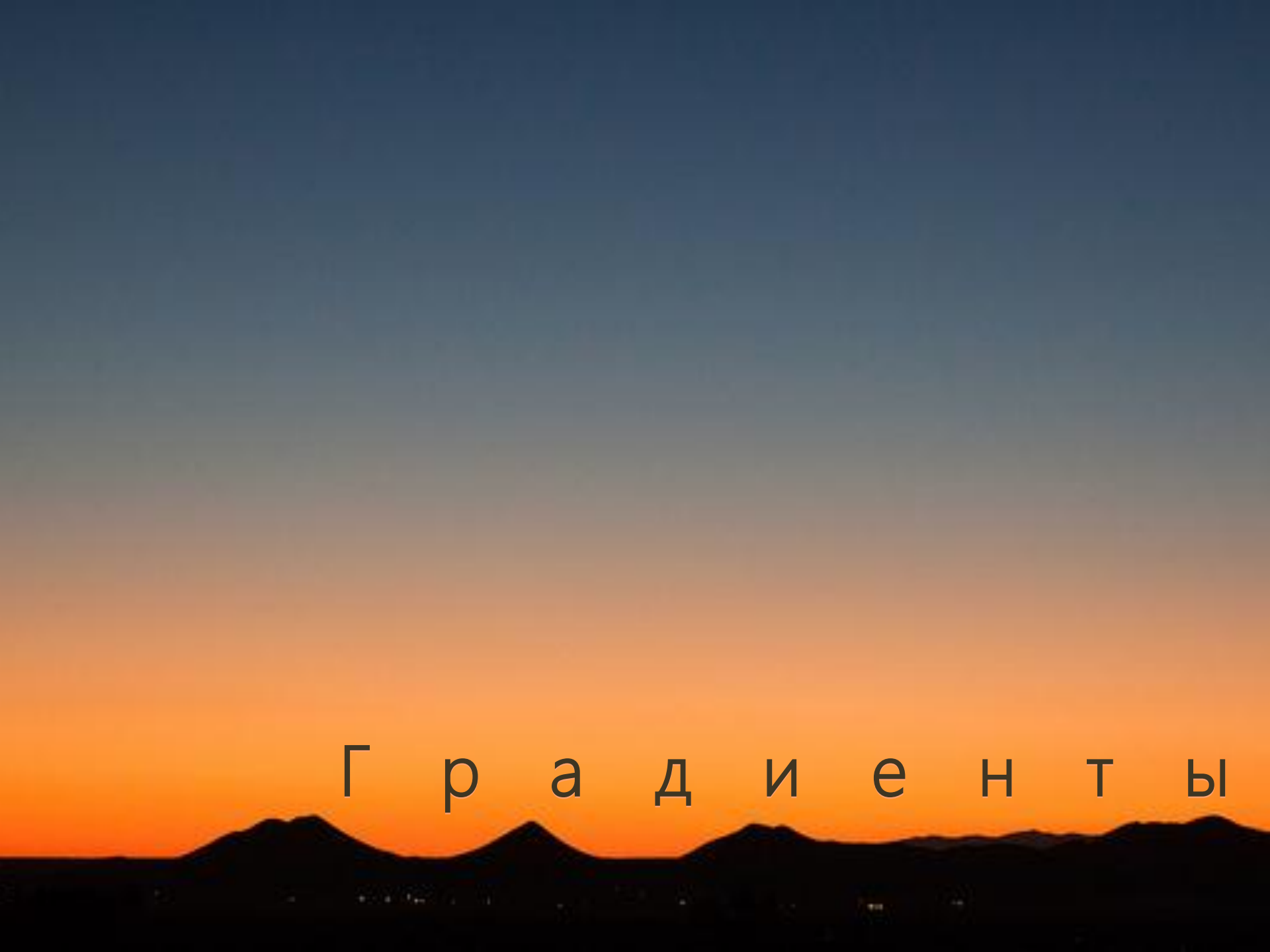
перемноженный
(premultiplied)

Перемножение

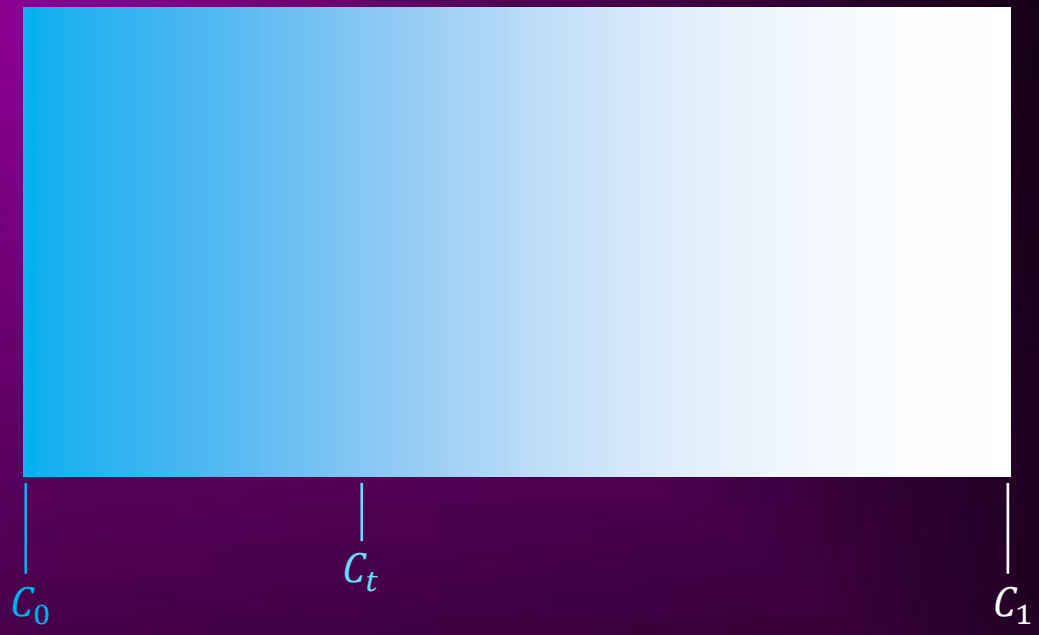


Перемножение



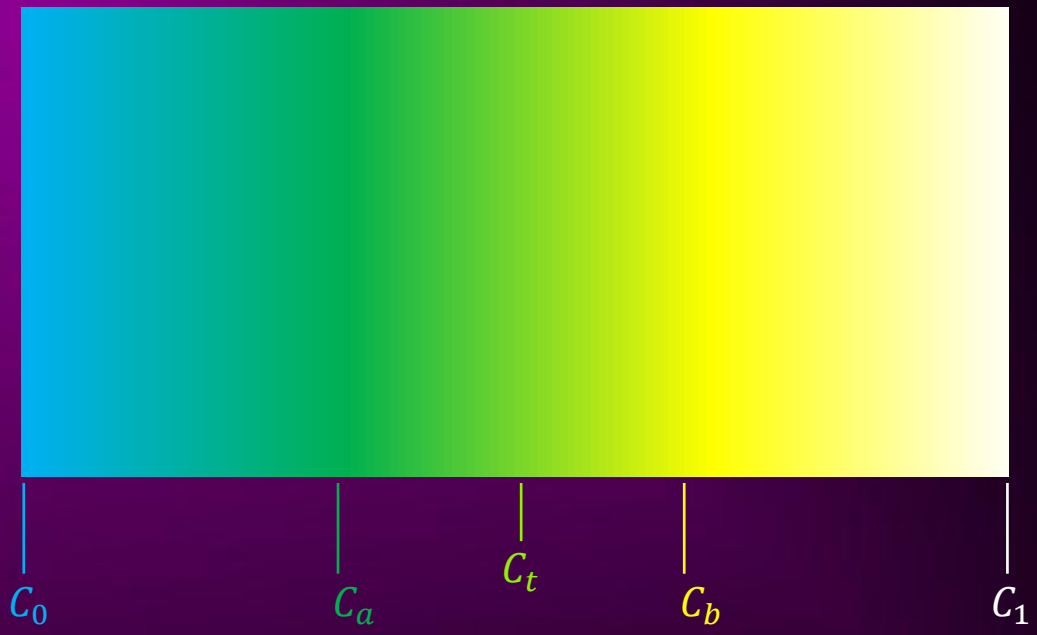

Г р а д и е н т ы

2 цвета



$$C_t = C_0 + t(C_1 - C_0)$$

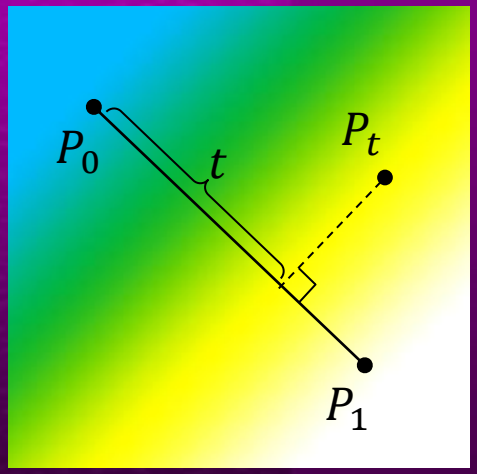
4 цвета



$$t' = \frac{t - a}{b - a}$$

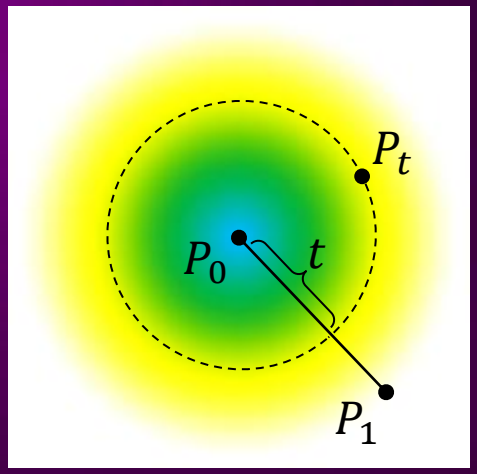
$$C_t = C_a + t'(C_b - C_a), \qquad a < t' < b$$

линейный



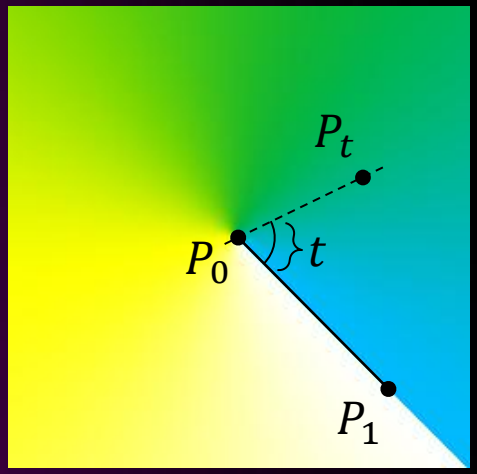
$$t = \frac{(x_t - x_0)(x_1 - x_0) - (y_1 - y_0)(y_0 - y_t)}{(x_1 - x_0)^2 + (y_1 - y_0)(y_1 - 2y_t + y_0)}$$

радиальный

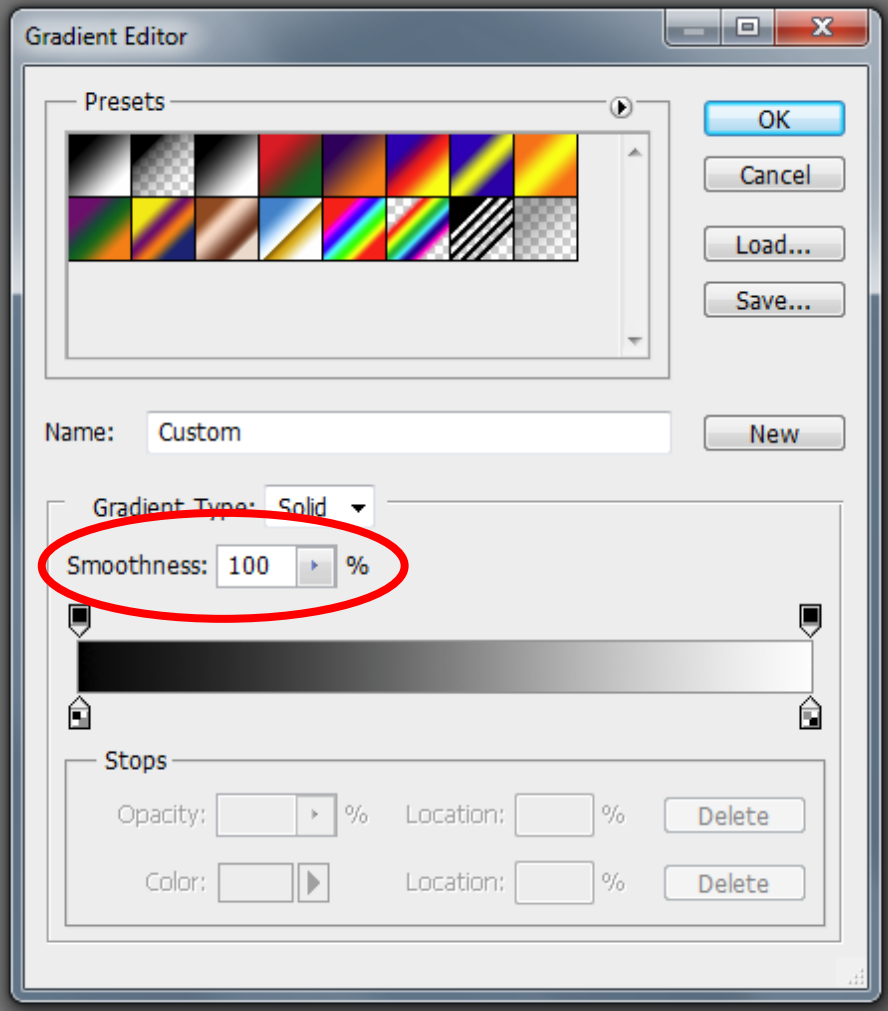


$$t = \frac{\sqrt{(x_t - x_0)^2 + (y_t - y_0)^2}}{\sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2}}$$

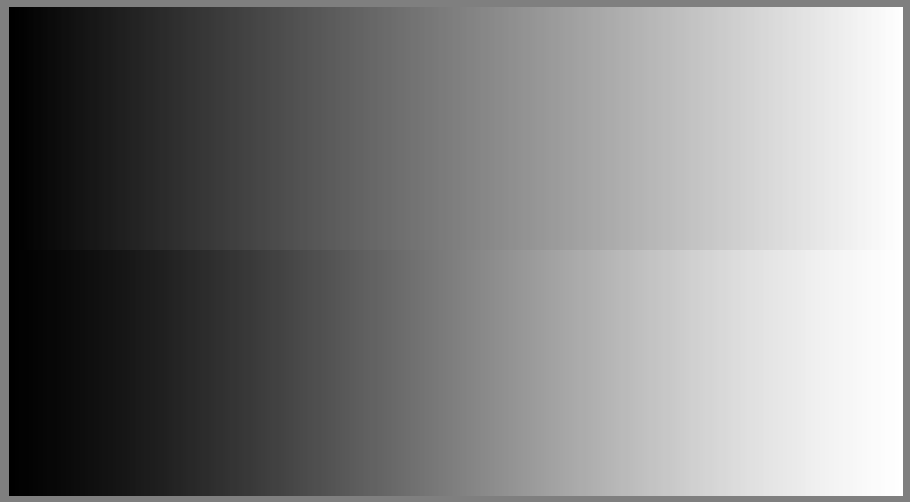
угловой

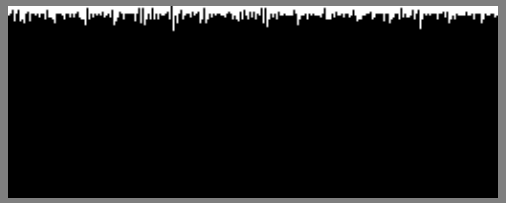


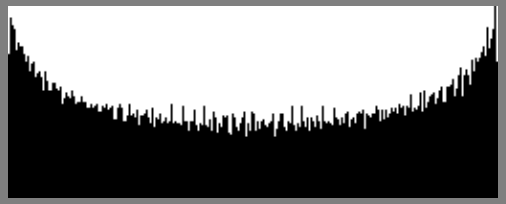
$$t = \frac{\tan^{-1}\left(\frac{y_t - y_0}{x_t - x_0}\right)}{2\pi \tan^{-1}\left(\frac{y_1 - y_0}{x_1 - x_0}\right)}$$


Gradient Editor
Presets
OK
Cancel
Load...
Save...
Name:
Custom
New
Gradient Type:
Solid
Smoothness:
100
%
Stops
Opacity:
%
Location:
%
Delete
Color:
Location:
%
Delete

smoothness = 0%

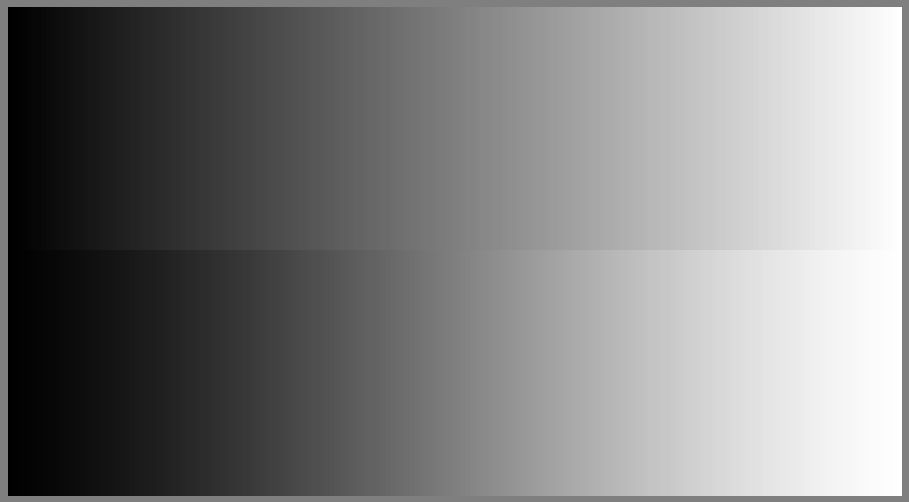


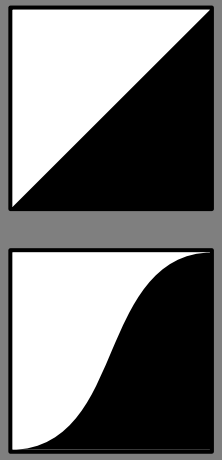




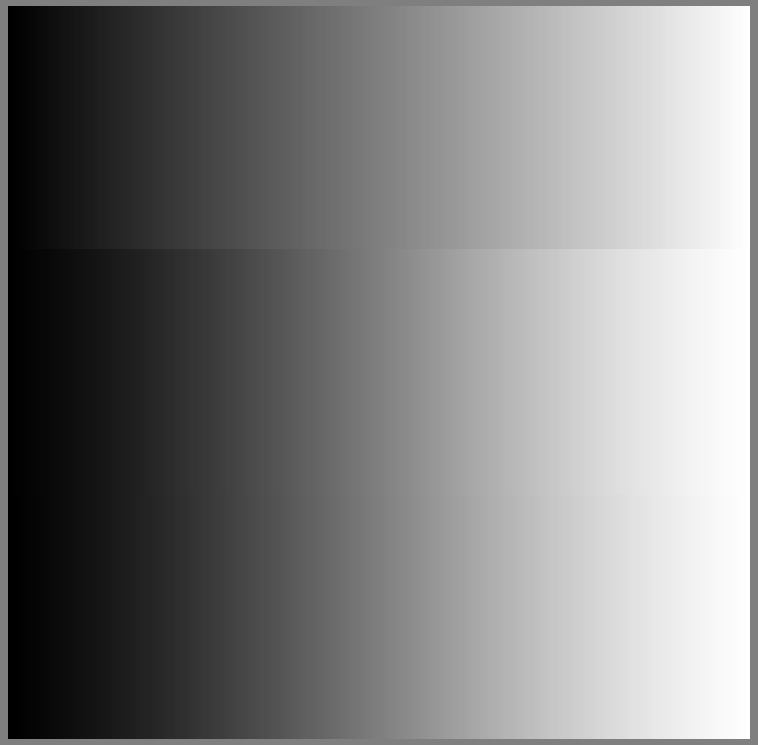
smoothness = 100%

smoothness = 0%





smoothness = 100%



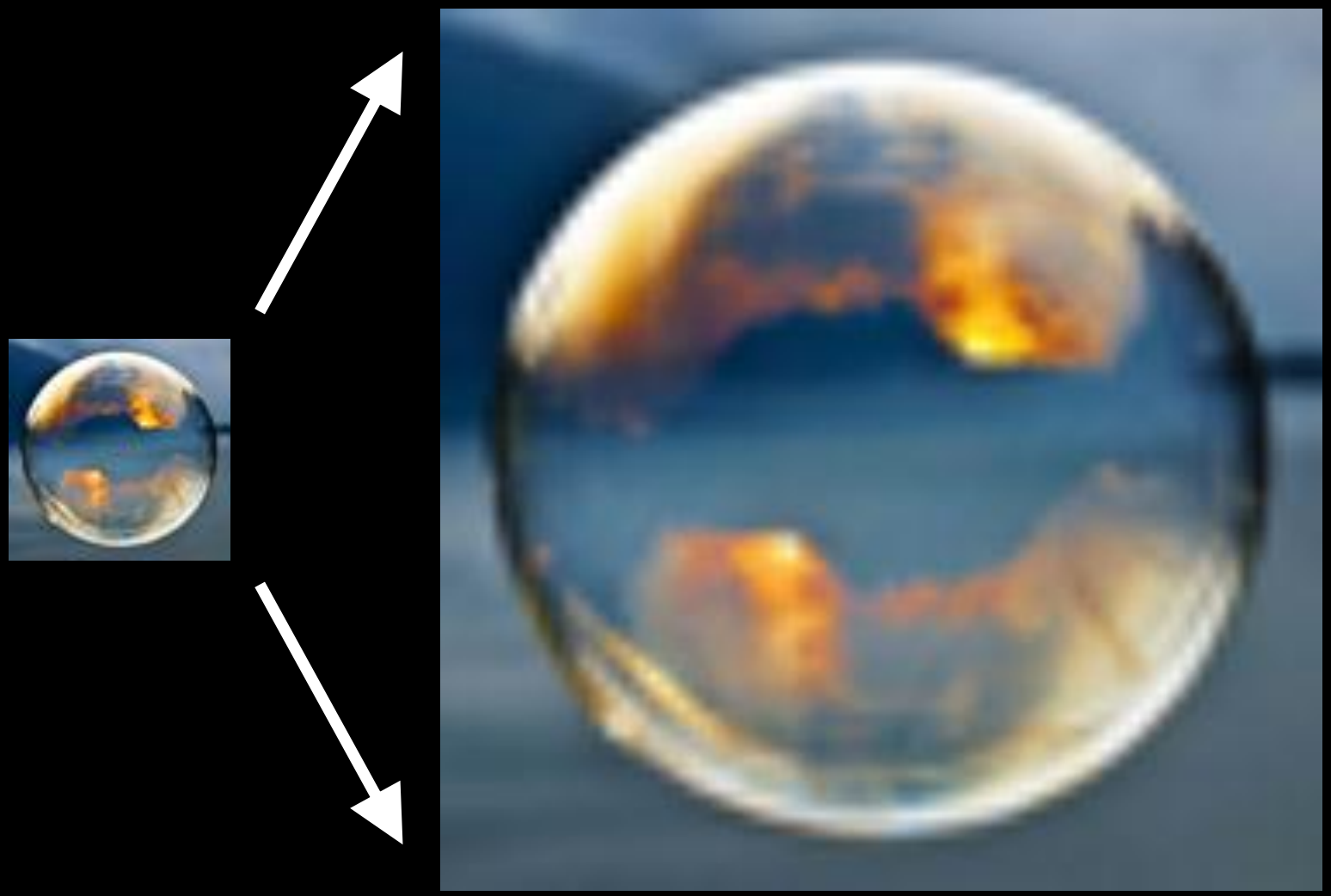
$$L(C) = C_0 + t(C_1 - C_0)$$

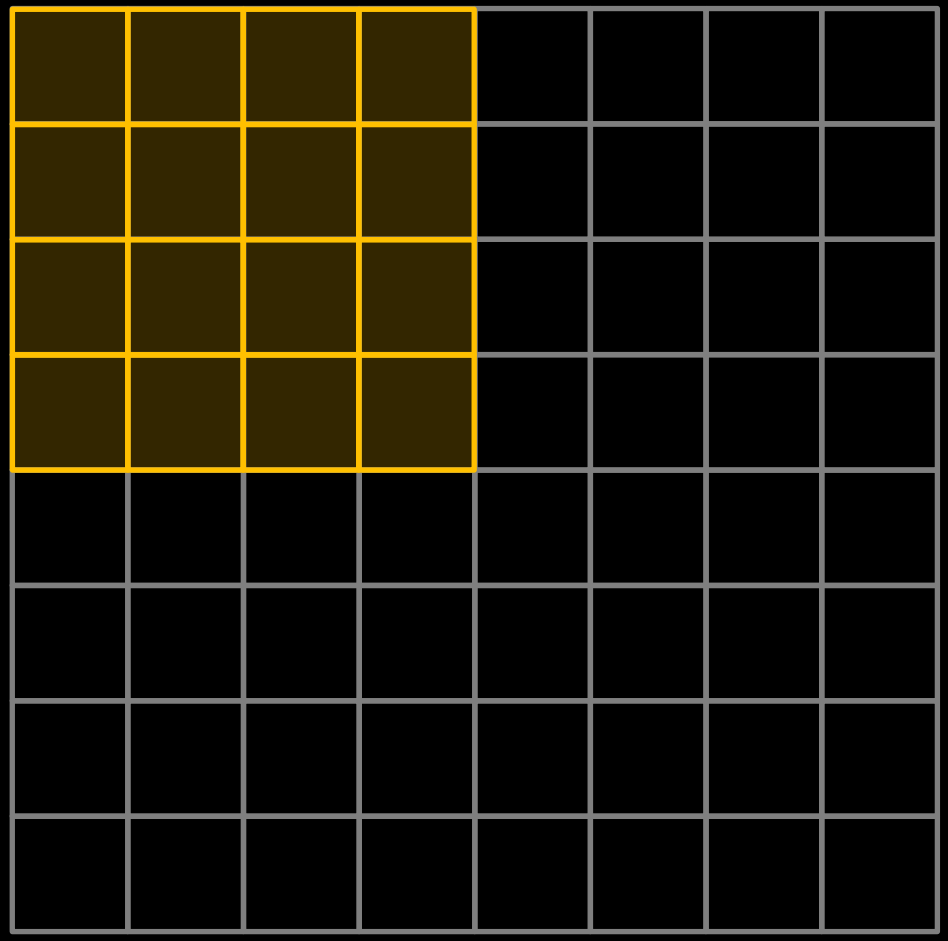
smoothness = 100%

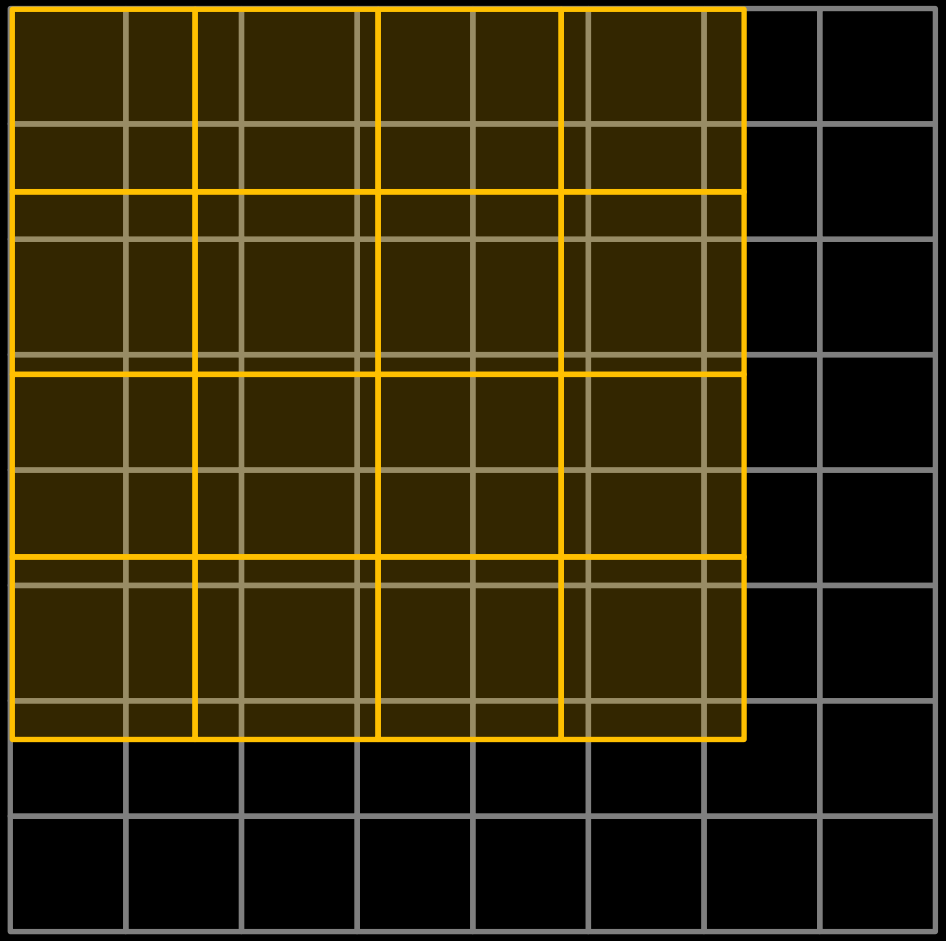
$$S(C) = C_0 + \left(t + 0.42 \left(\frac{1 - \cos(\pi t)}{2} - t \right) \right) (C_1 - C_0)$$

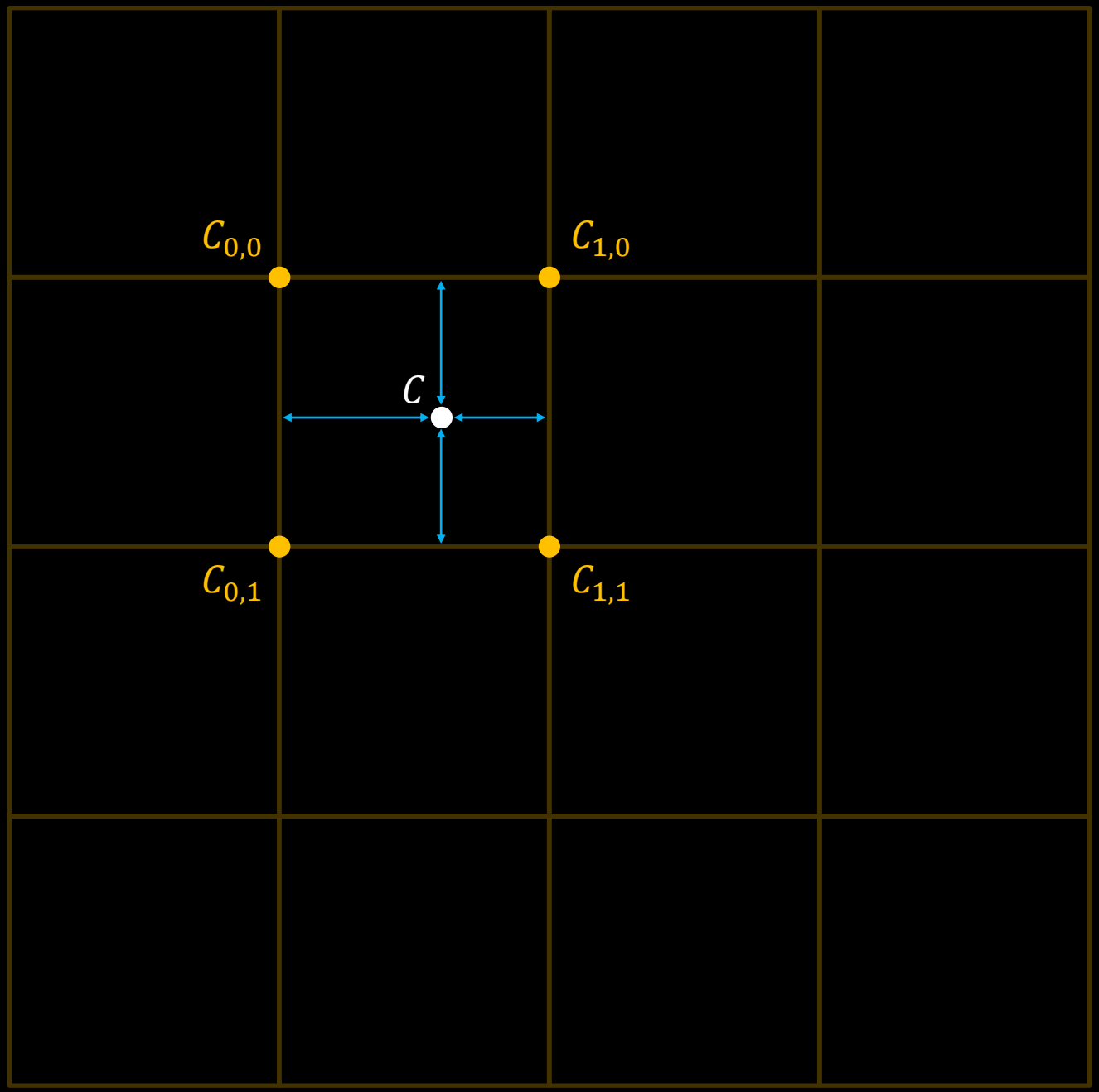
Изображения

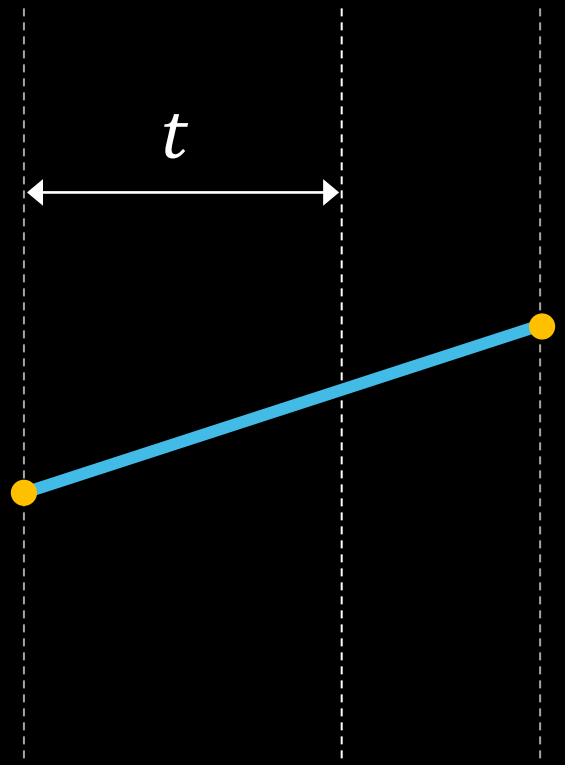
Интерполяция

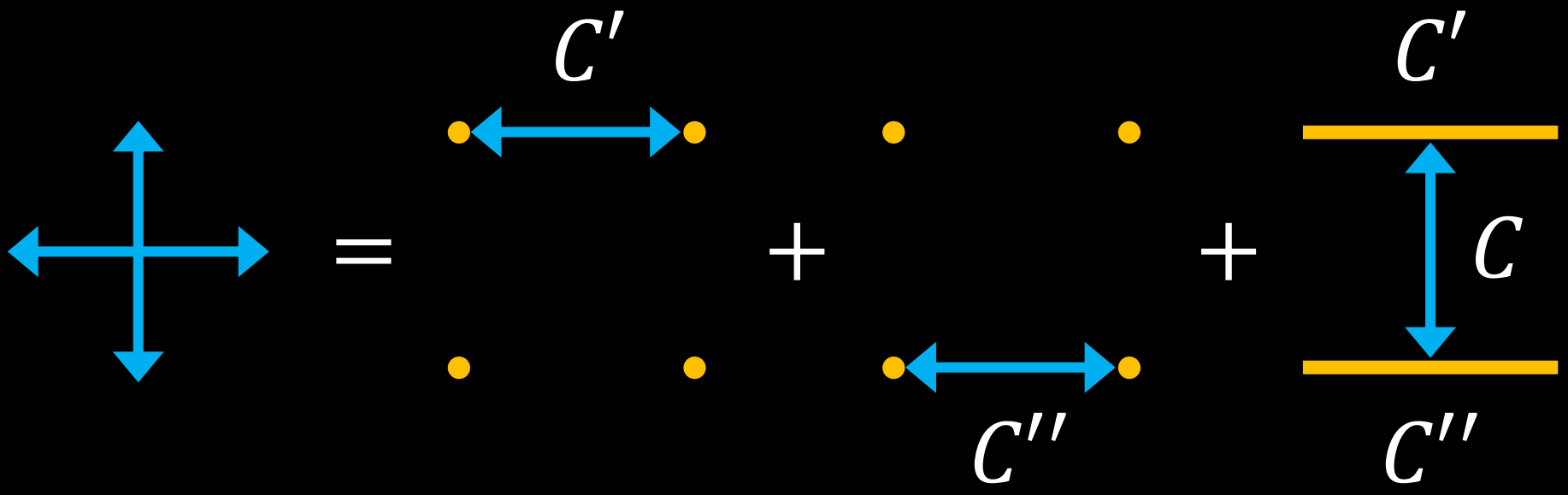







$C_{0,0}$
$C_{1,0}$
C
$C_{0,1}$
$C_{1,1}$

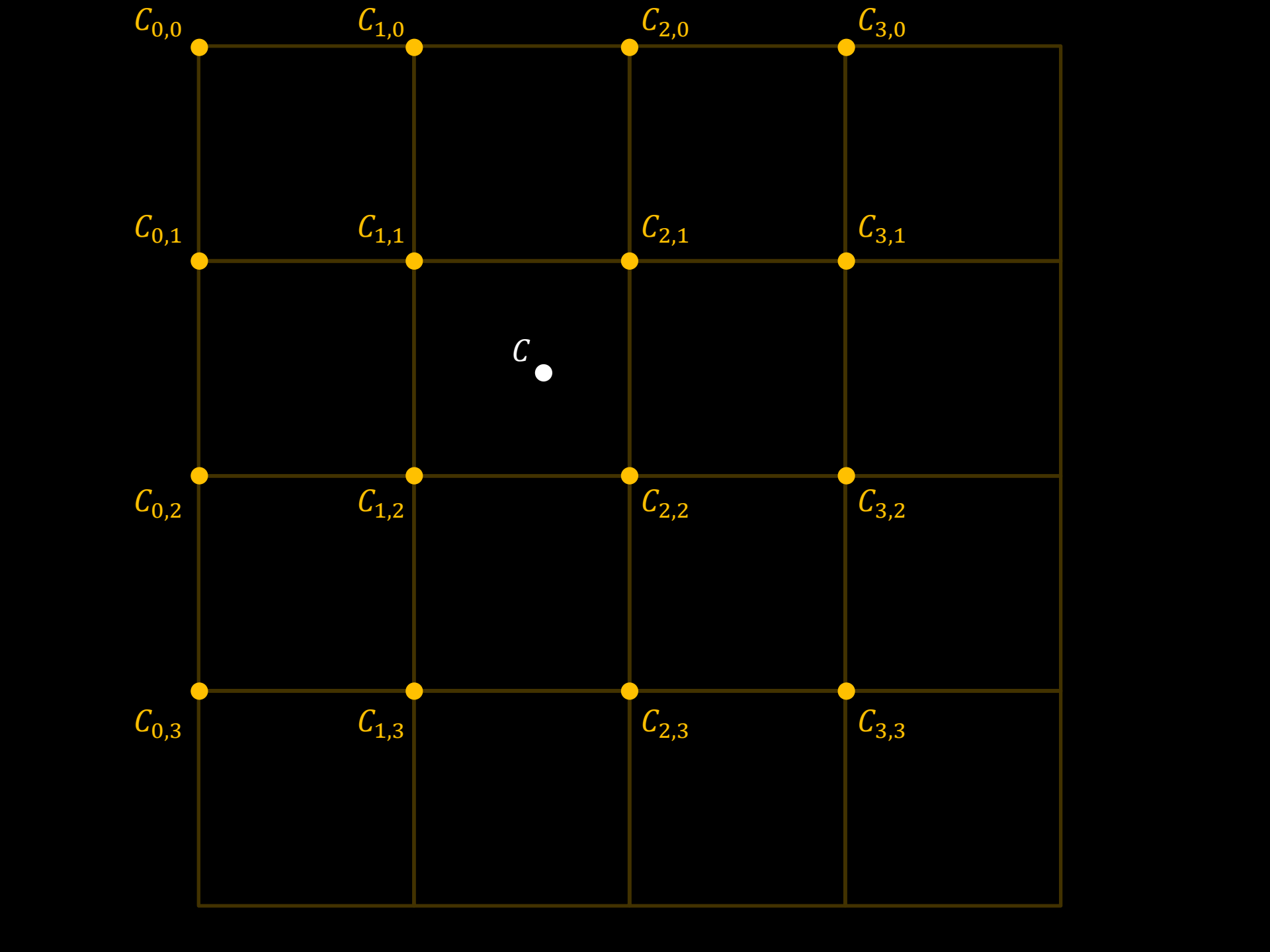

t

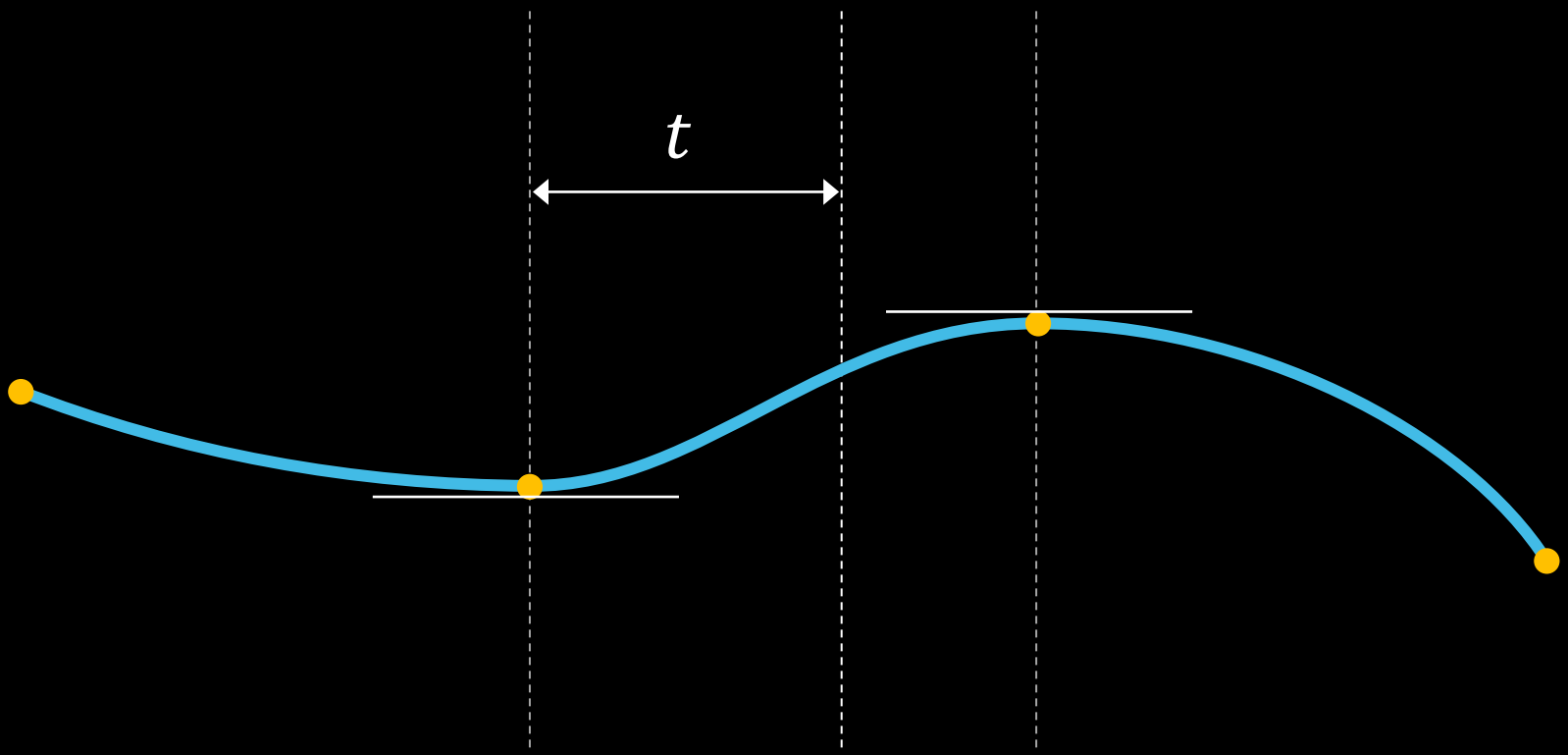

C′
C′
C
C″
C″

$$C' = C_{0,0} + t_x\left(C_{1,0} - C_{0,0}\right)$$

$$C'' = C_{0,1} + t_x\left(C_{1,1} - C_{0,1}\right)$$

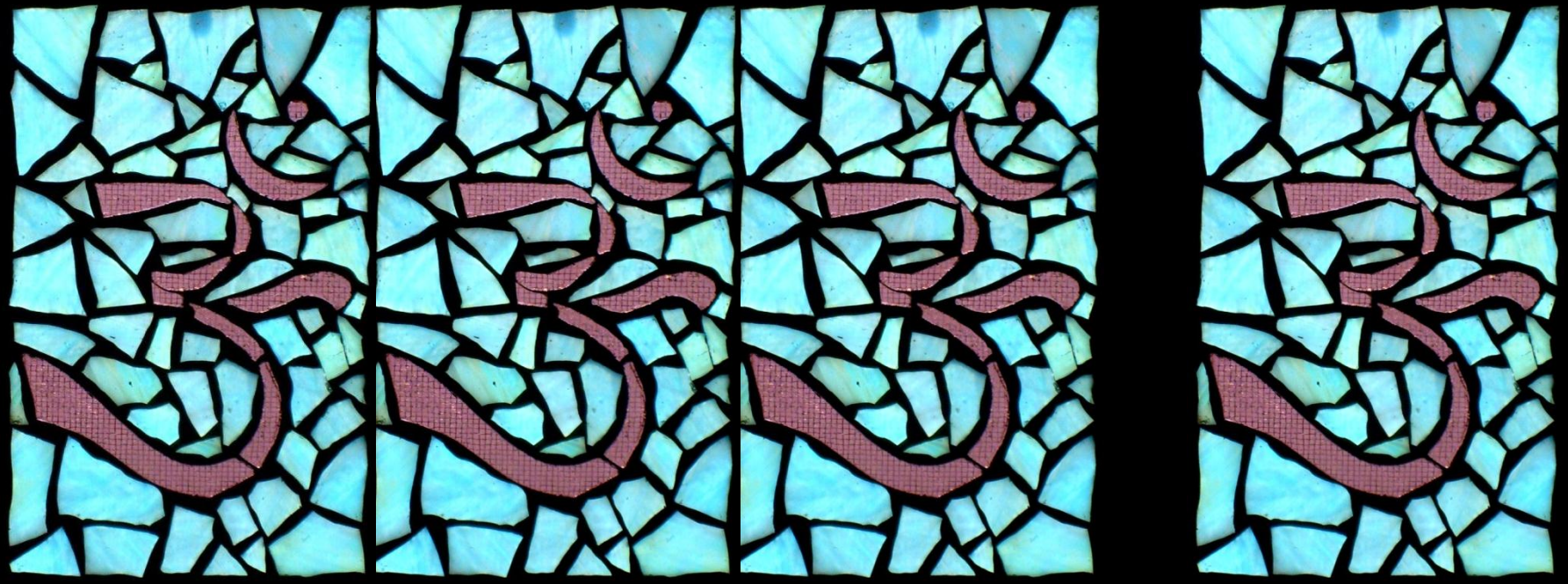
$$C = C' + t_y(C'' - C')$$

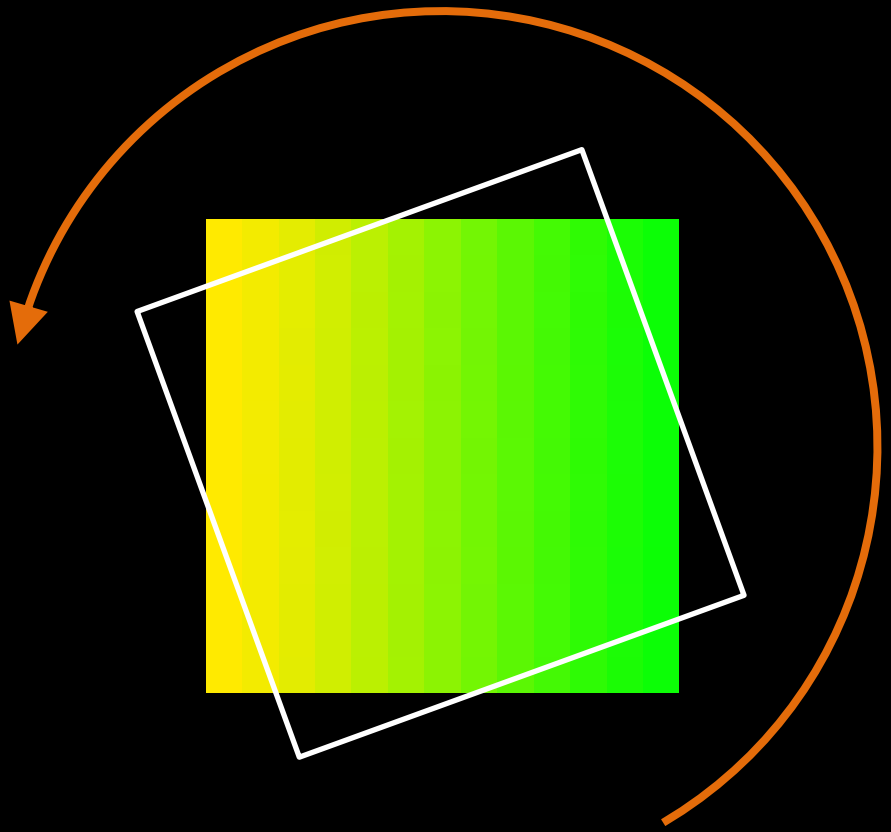

$C_{0,0}$
$C_{1,0}$
$C_{2,0}$
$C_{3,0}$
$C_{0,1}$
$C_{1,1}$
$C_{2,1}$
$C_{3,1}$
C
$C_{0,2}$
$C_{1,2}$
$C_{2,2}$
$C_{3,2}$
$C_{0,3}$
$C_{1,3}$
$C_{2,3}$
$C_{3,3}$

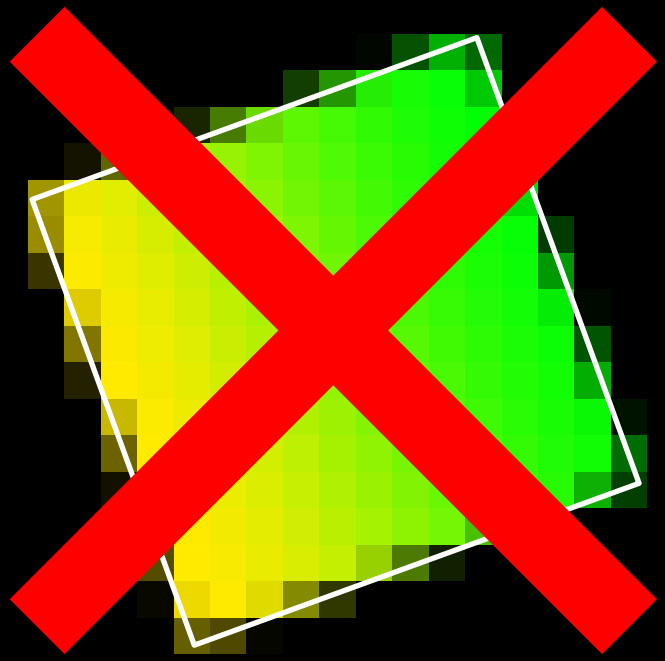

t

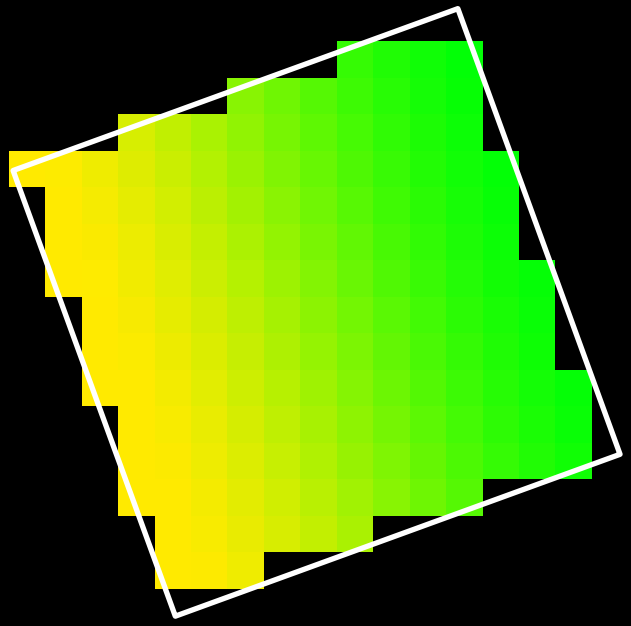
Нарочный

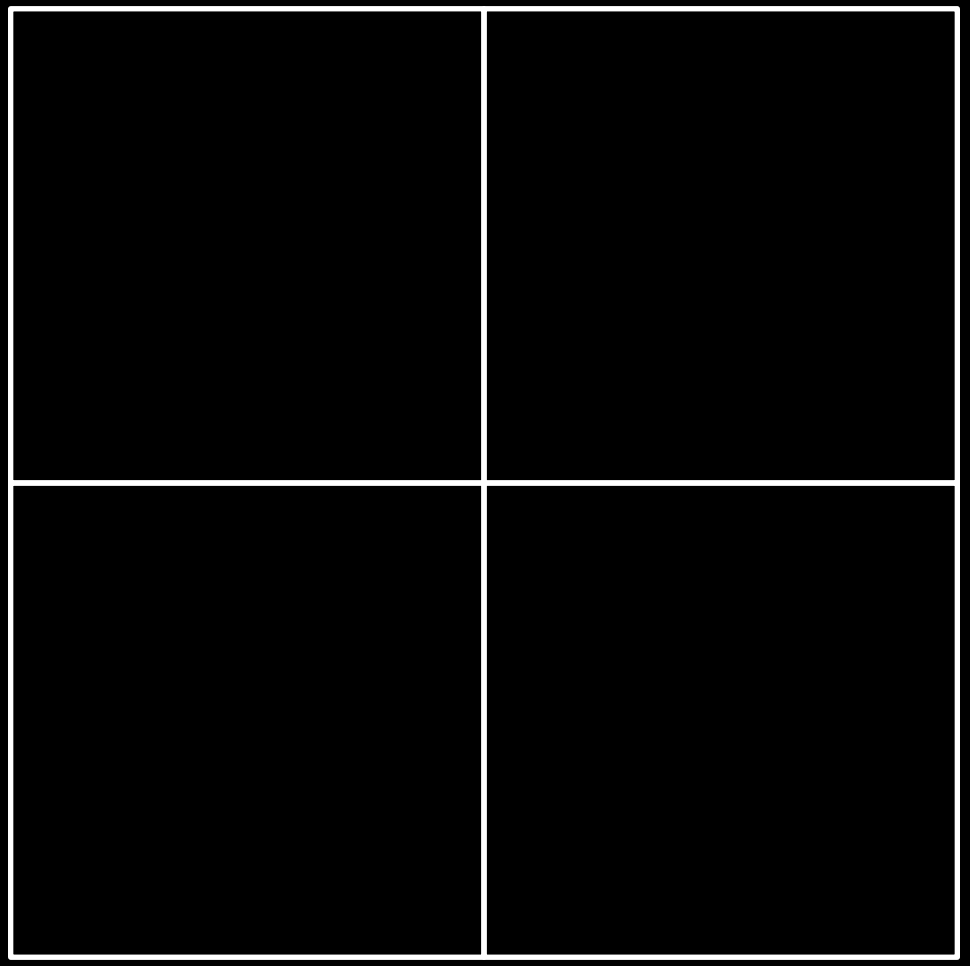
~~анти~~алиасинг

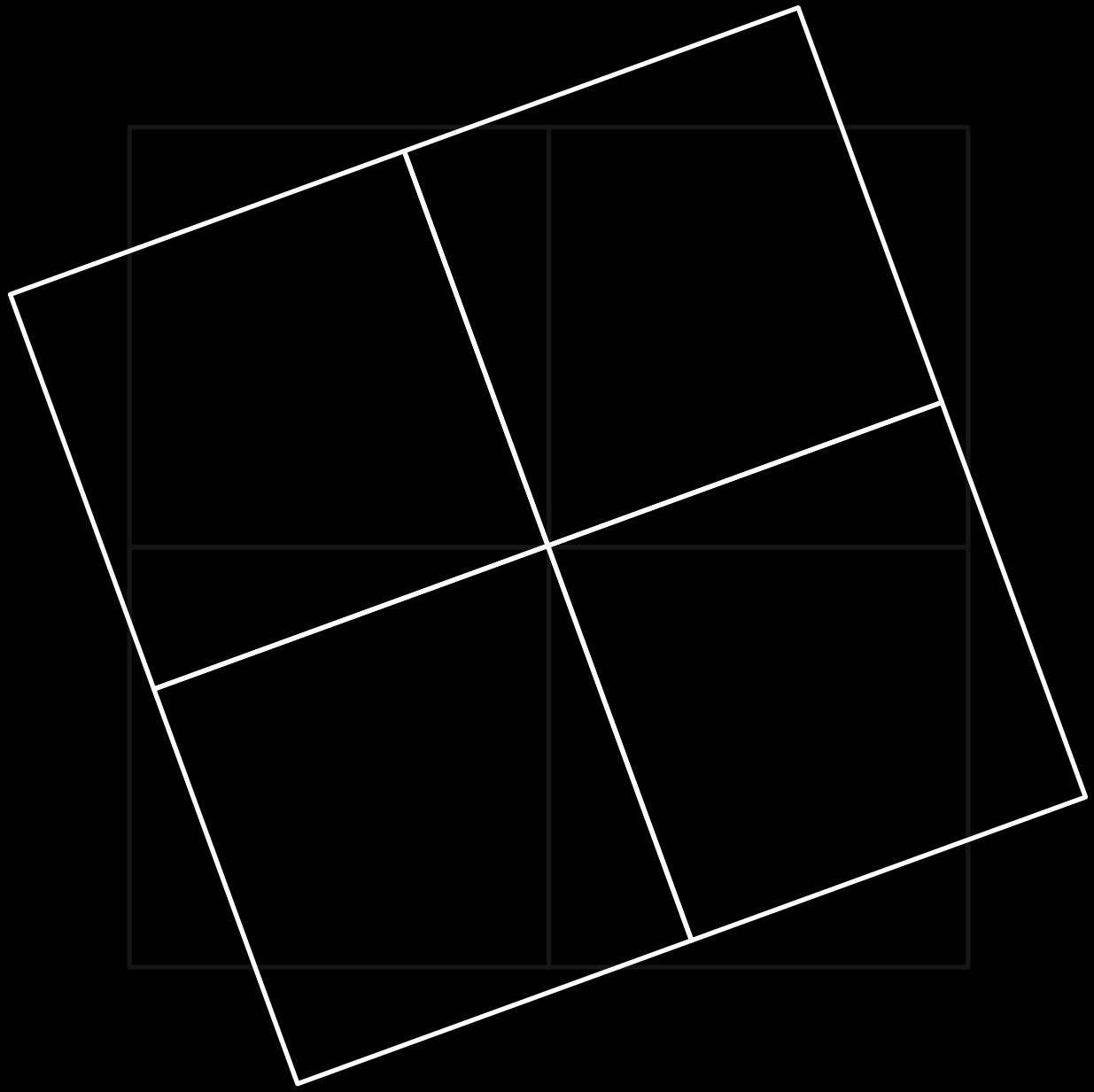


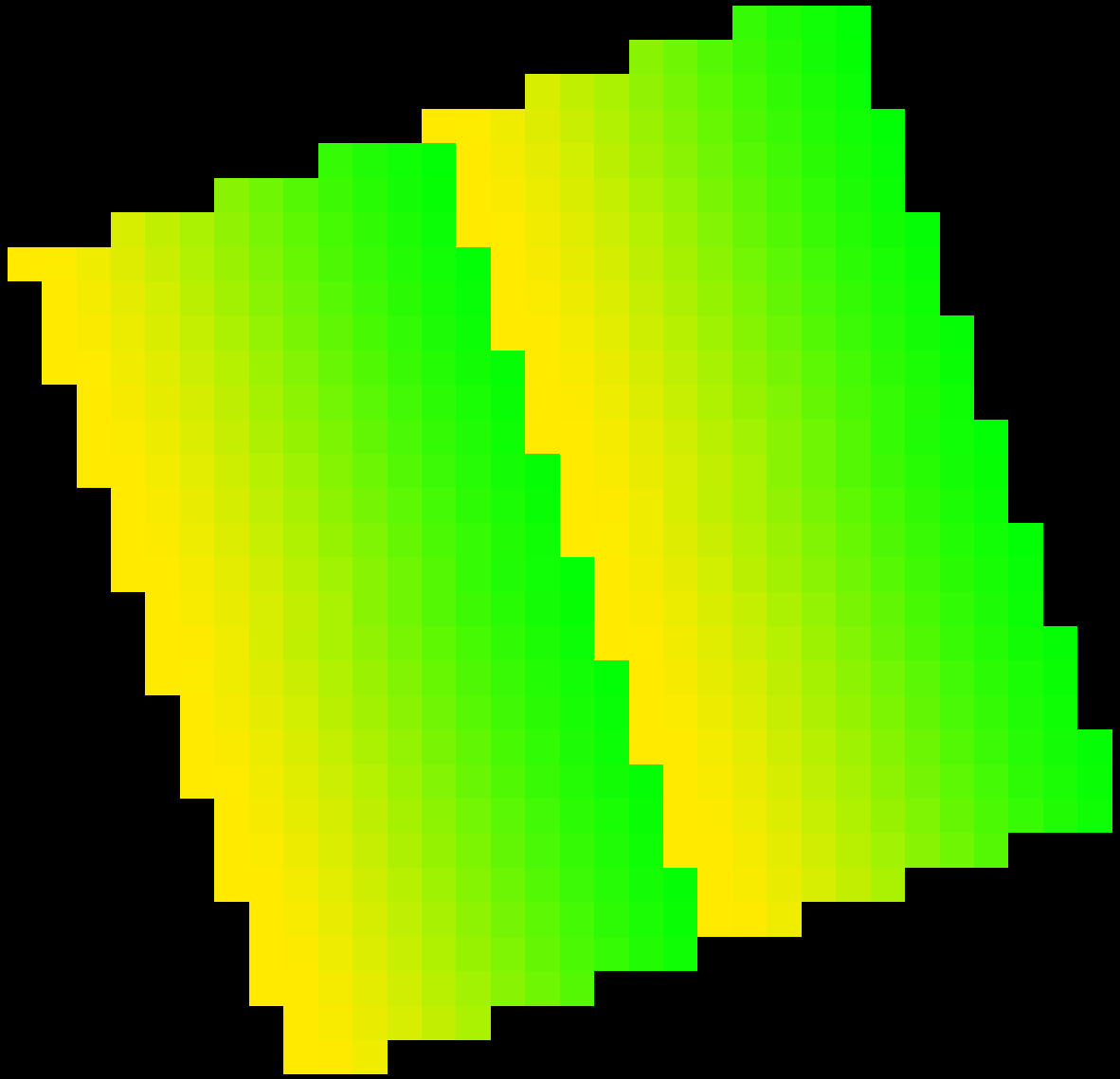


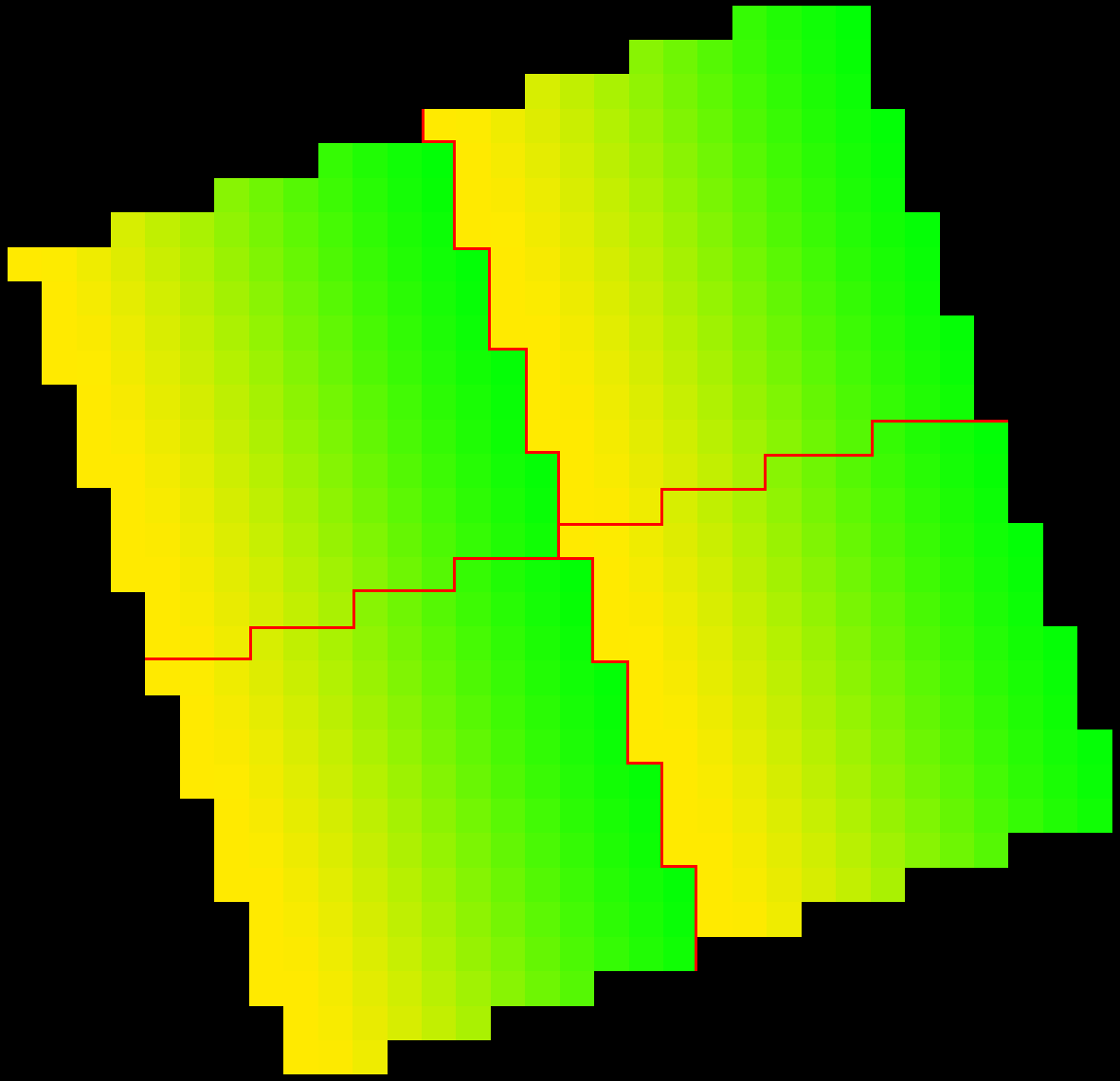


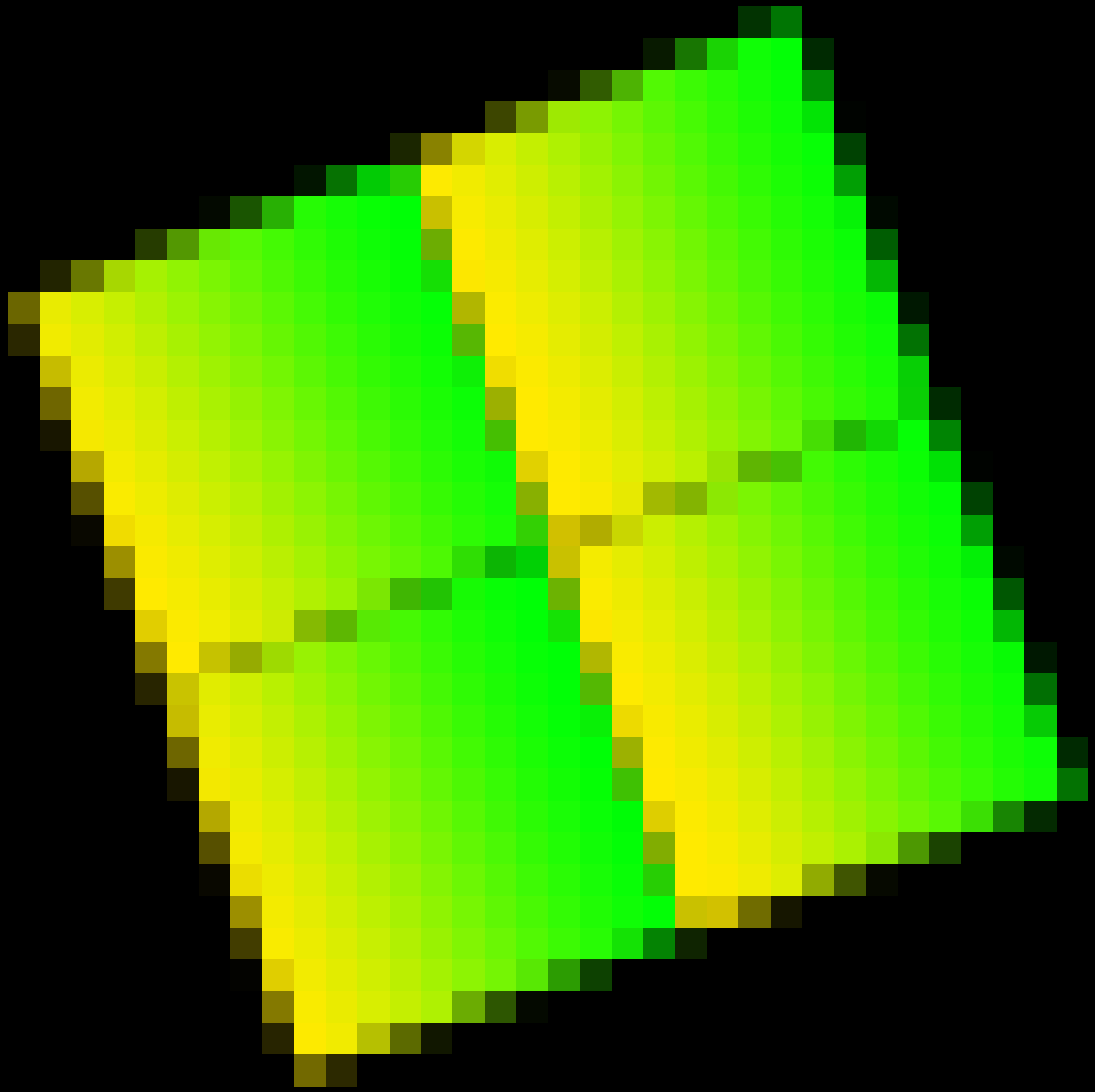


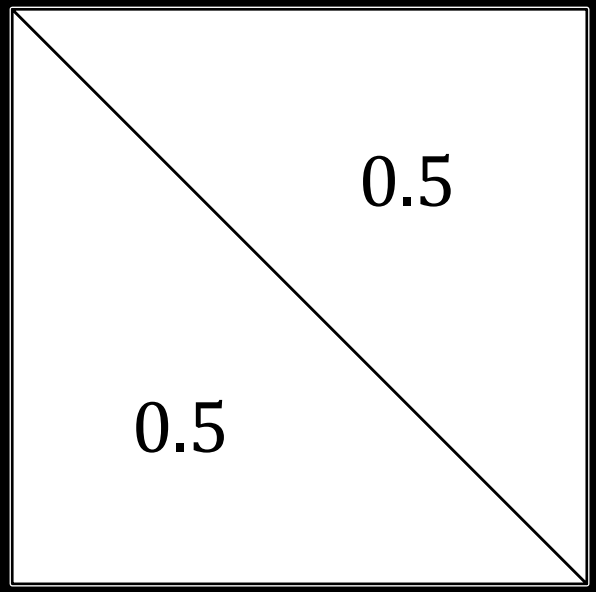


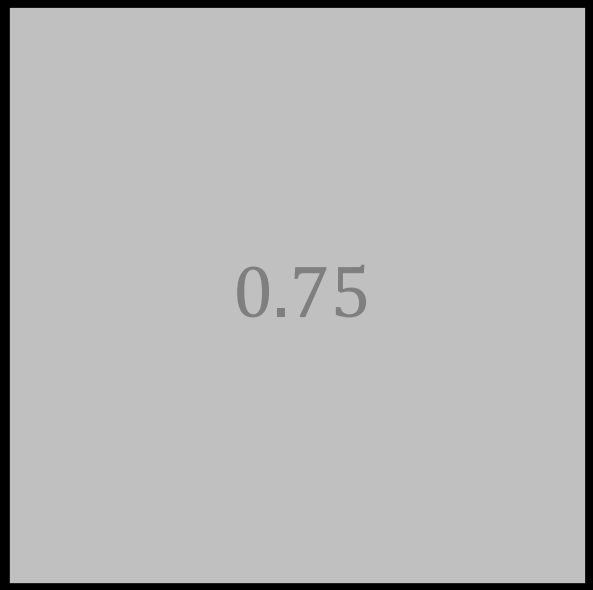






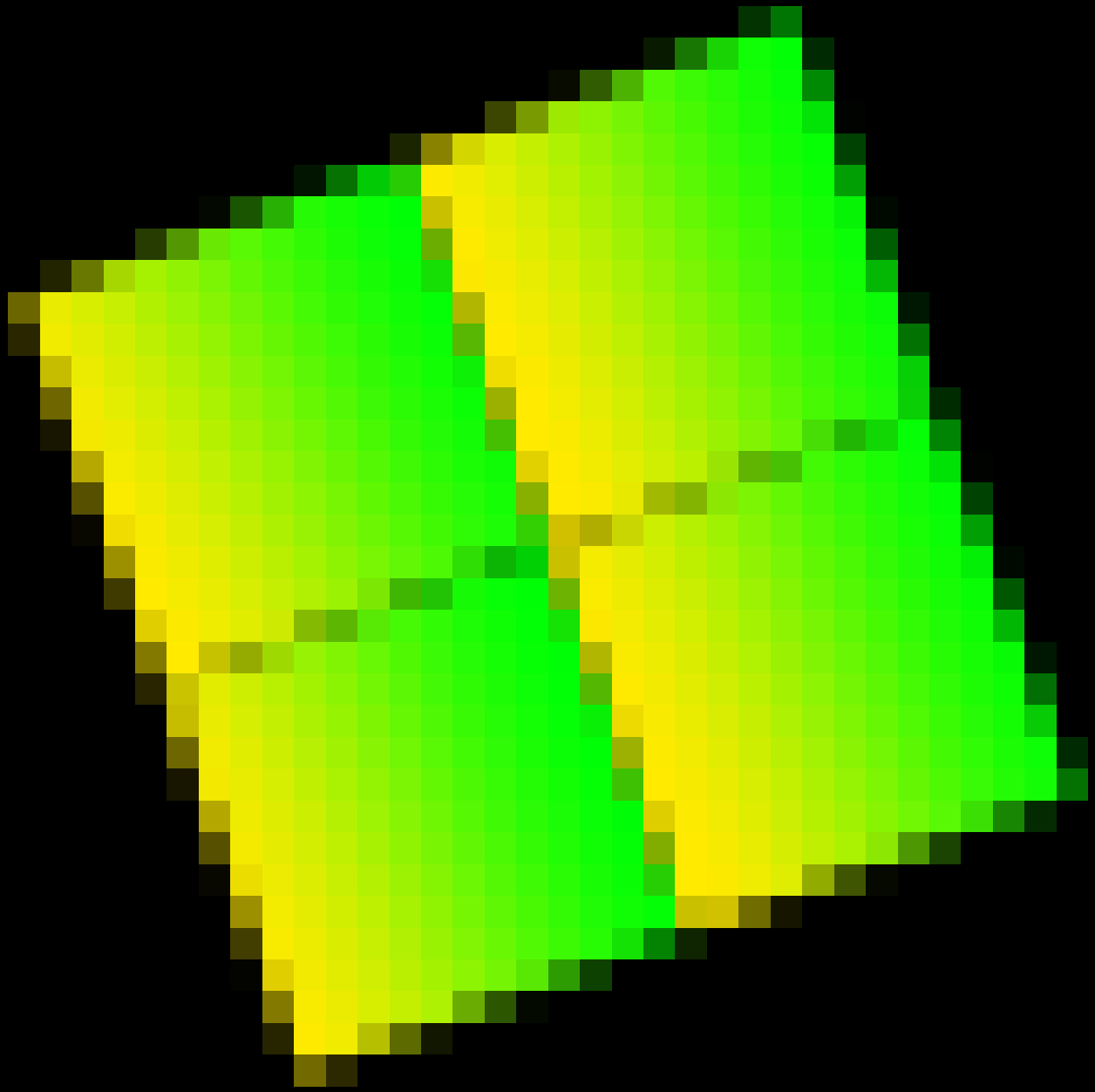


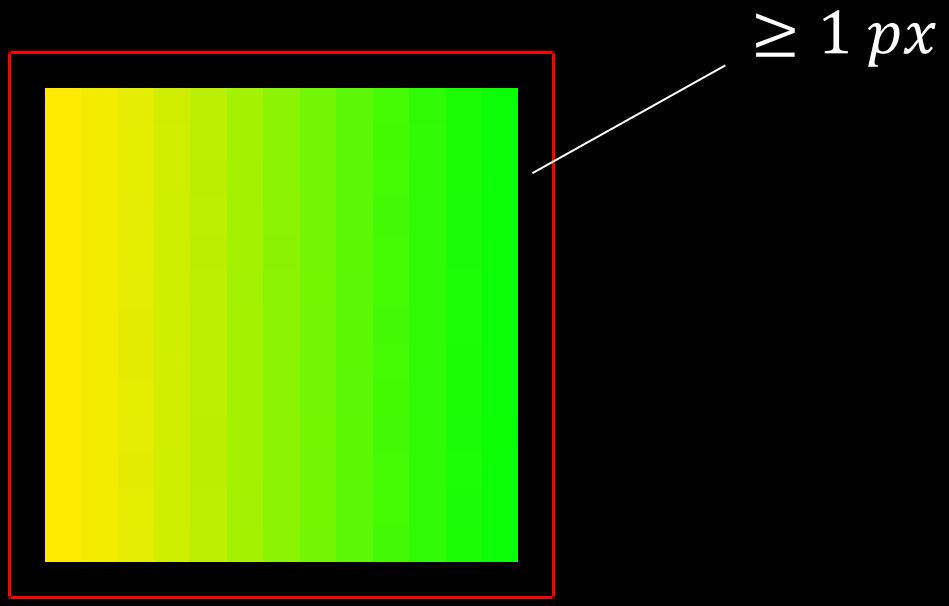


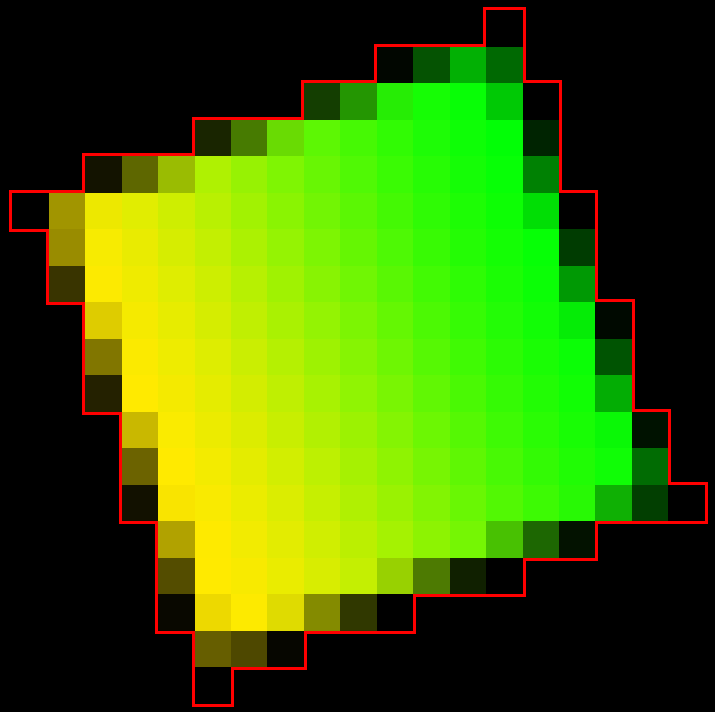


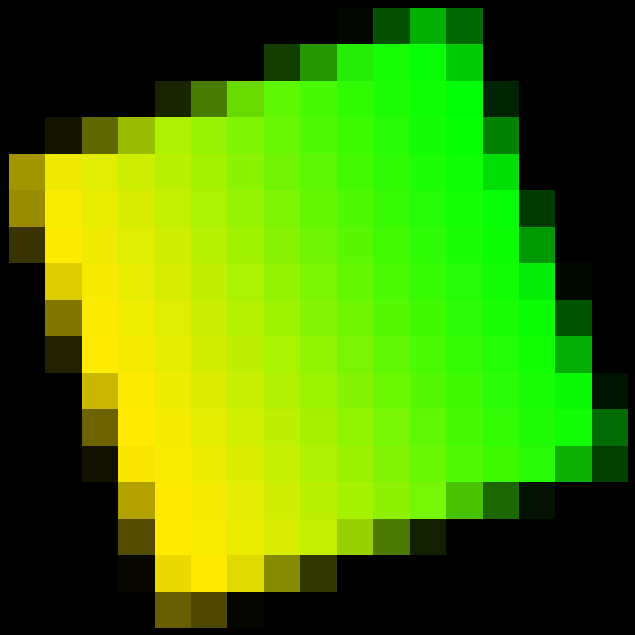
$$0 + 0.5(1 - 0) = 0.5$$

$$0.5 + 0.5(1 - 0.5) = 0.75$$




$\geq 1\,px$





узоры

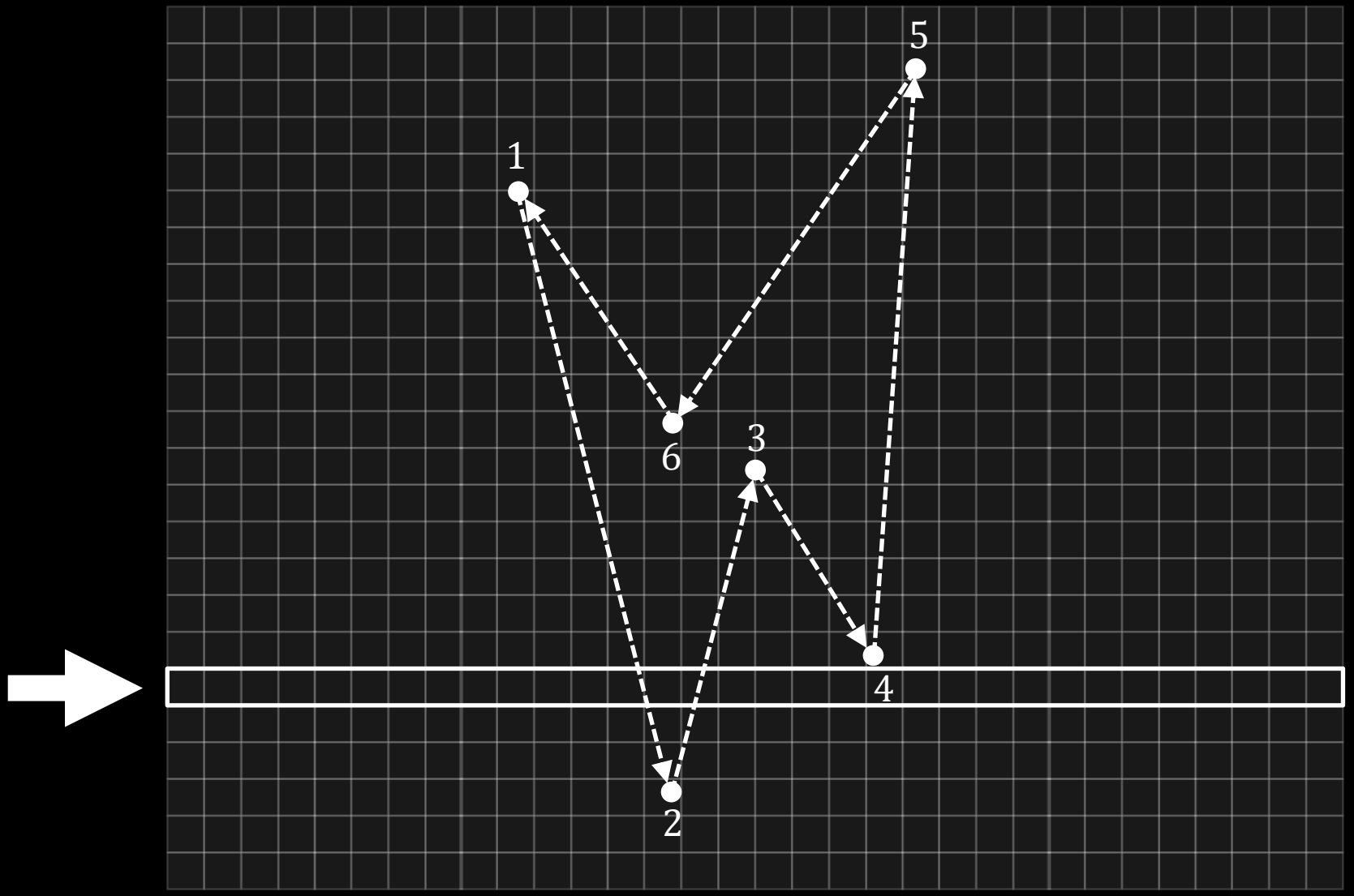
Шейдеры

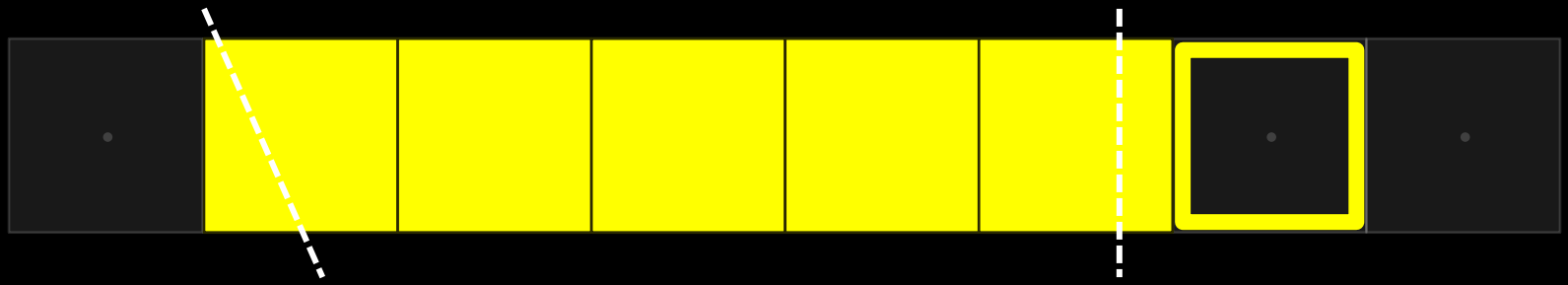
```
class ColorBrush : Brush
{
public:
    ColorBrush(const Color& color) : m_color(color) {}

    // virtual
    void FillScanline(Image& img, int y, int[] edges)
    {
        assert(edges.Count % 2 == 0);

        for (int i = 0; i < edges.Count; i += 2)
        {
            for (int x = edges[i]; x < edges[i + 1]; x++)
            {
                pix8& pix = img.Buffer[y * img.Width + x];
                pix = Over(pix8(m_color), pix);
            }
        }
    {

private:
    Color m_color;
};
```

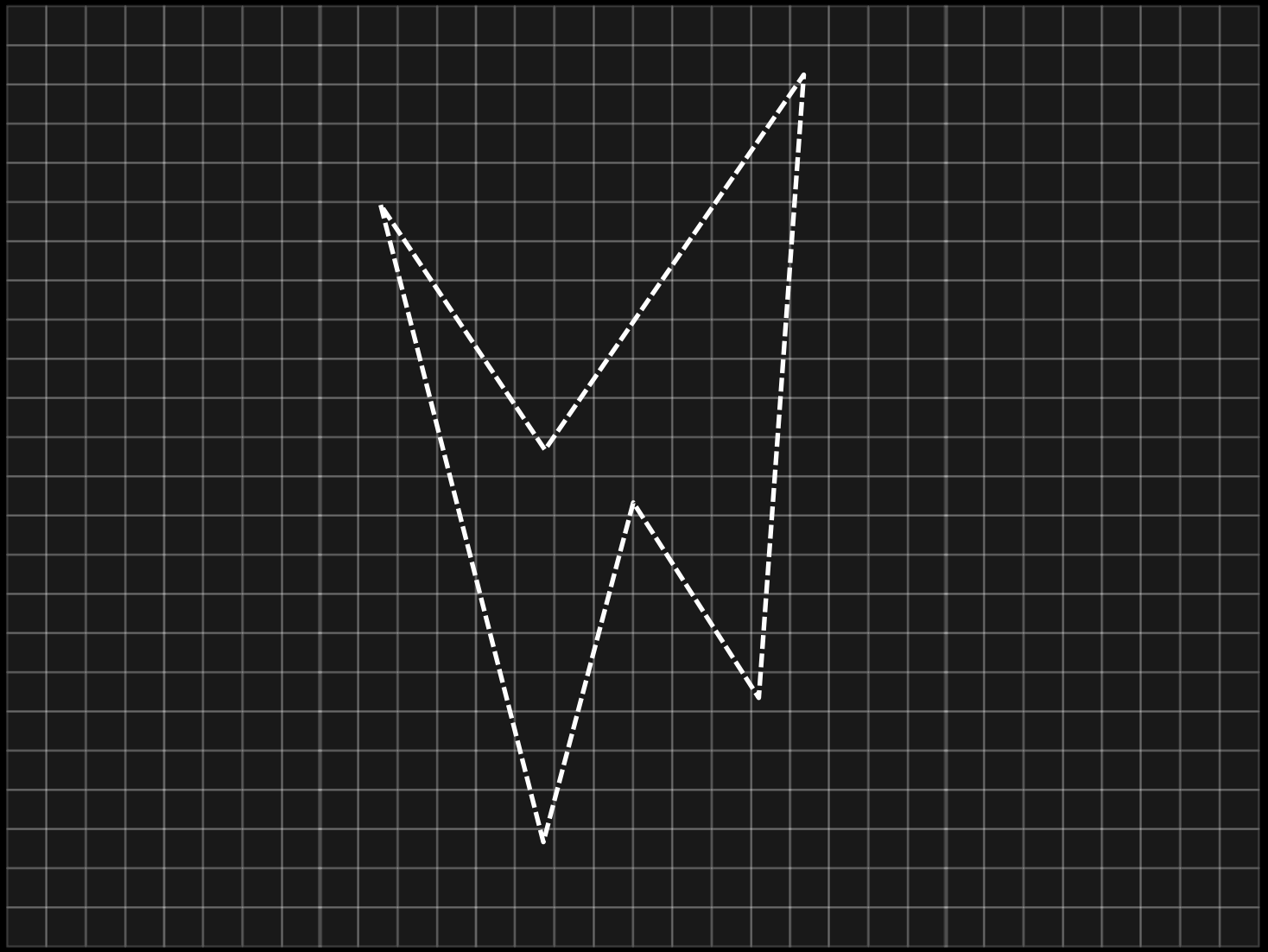

Brush
Color Brush
Gradient Brush
Linear Gradient Brush
Radial Gradient Brush
Angular Gradient Brush
Pattern Brush
Noise Brush
...

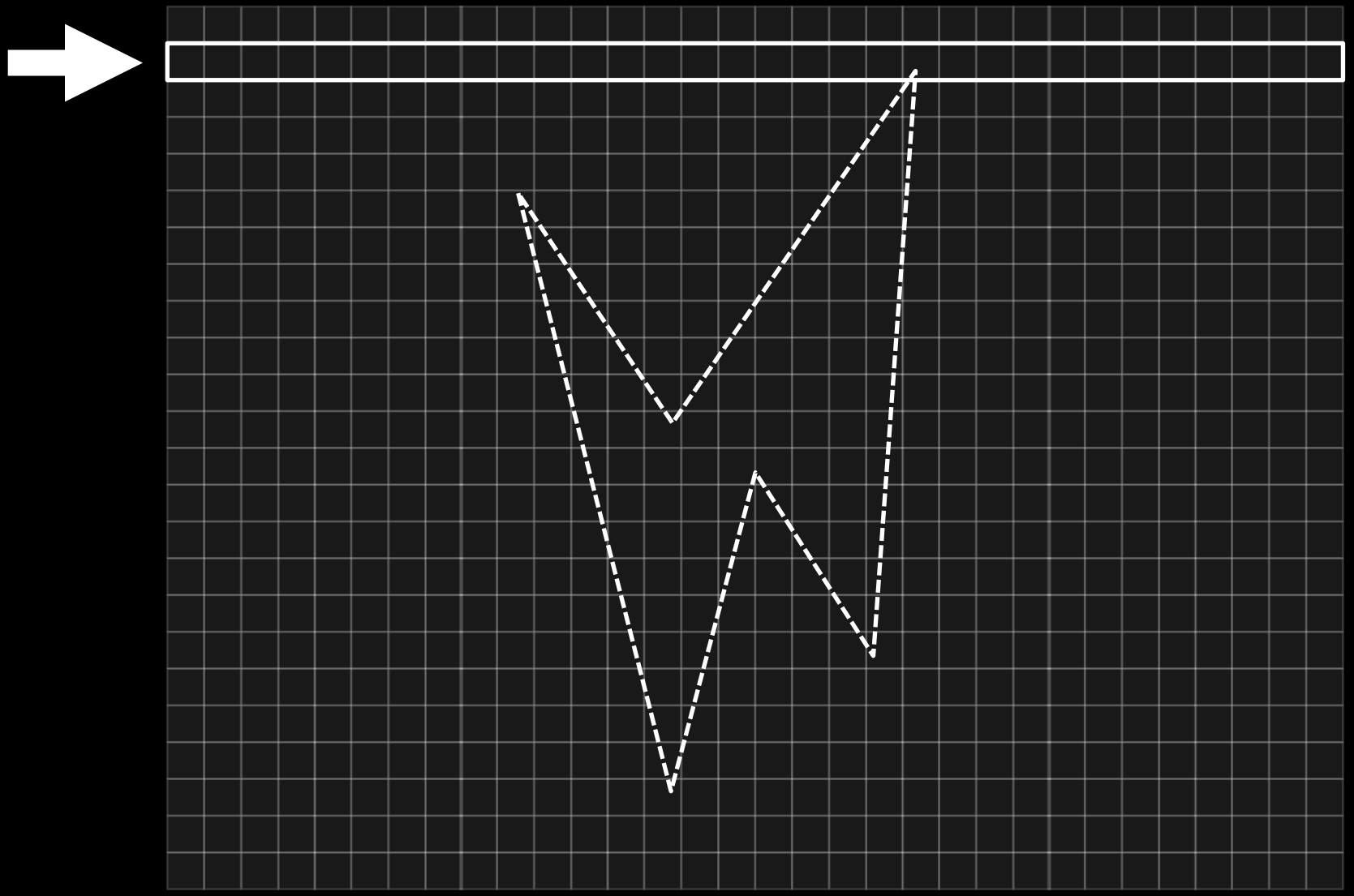
Векторная графика

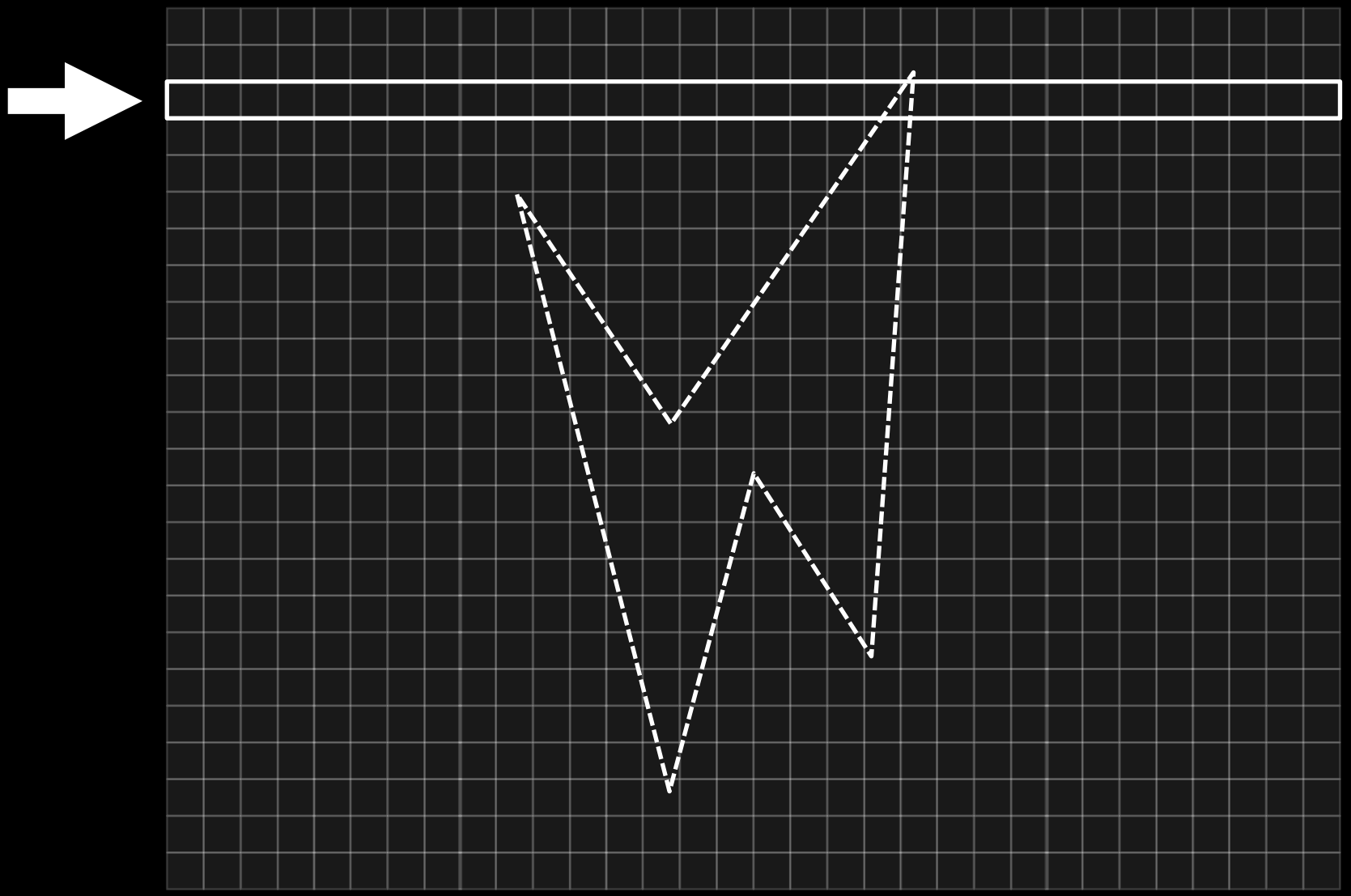
Полигоны

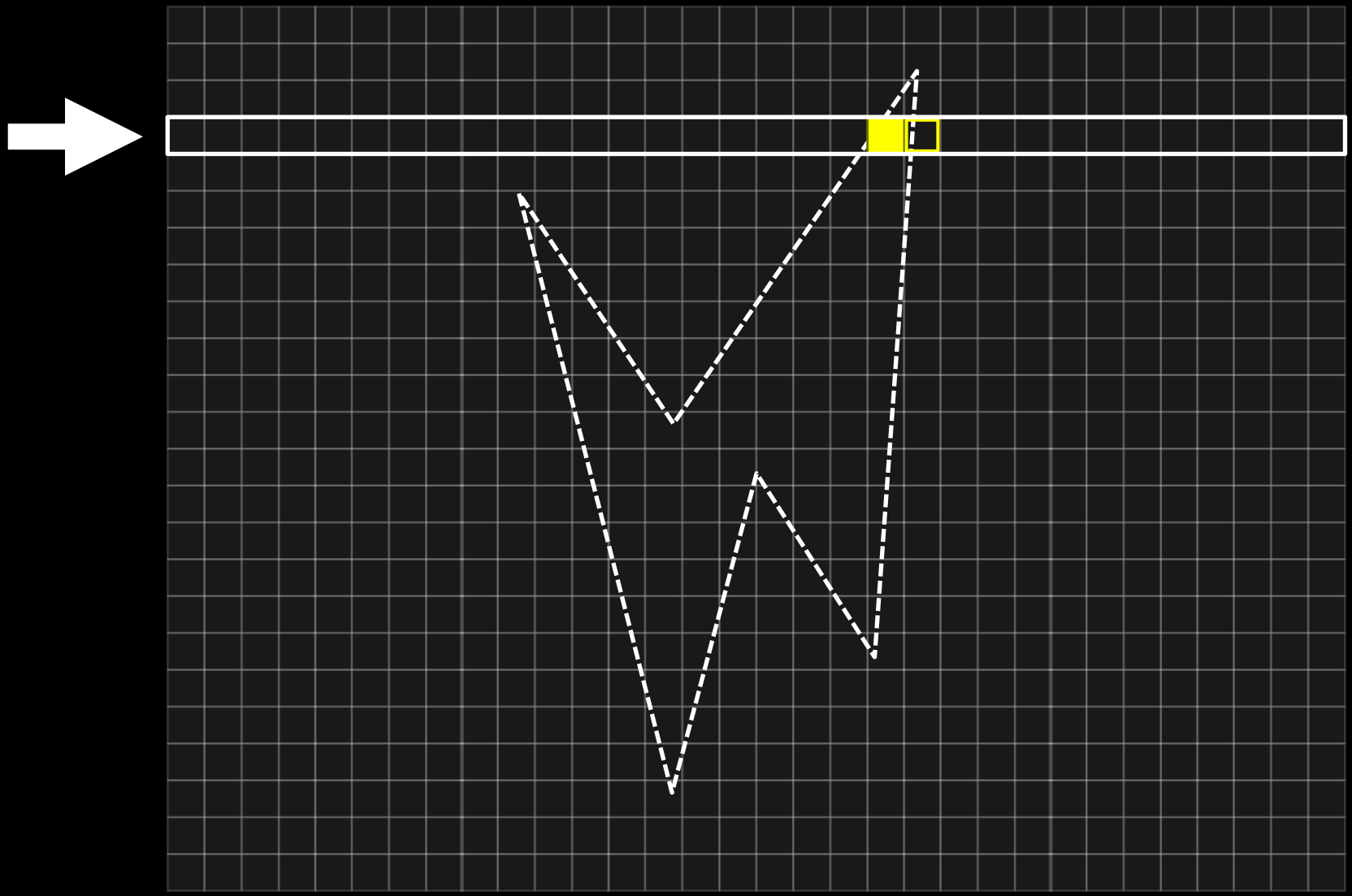


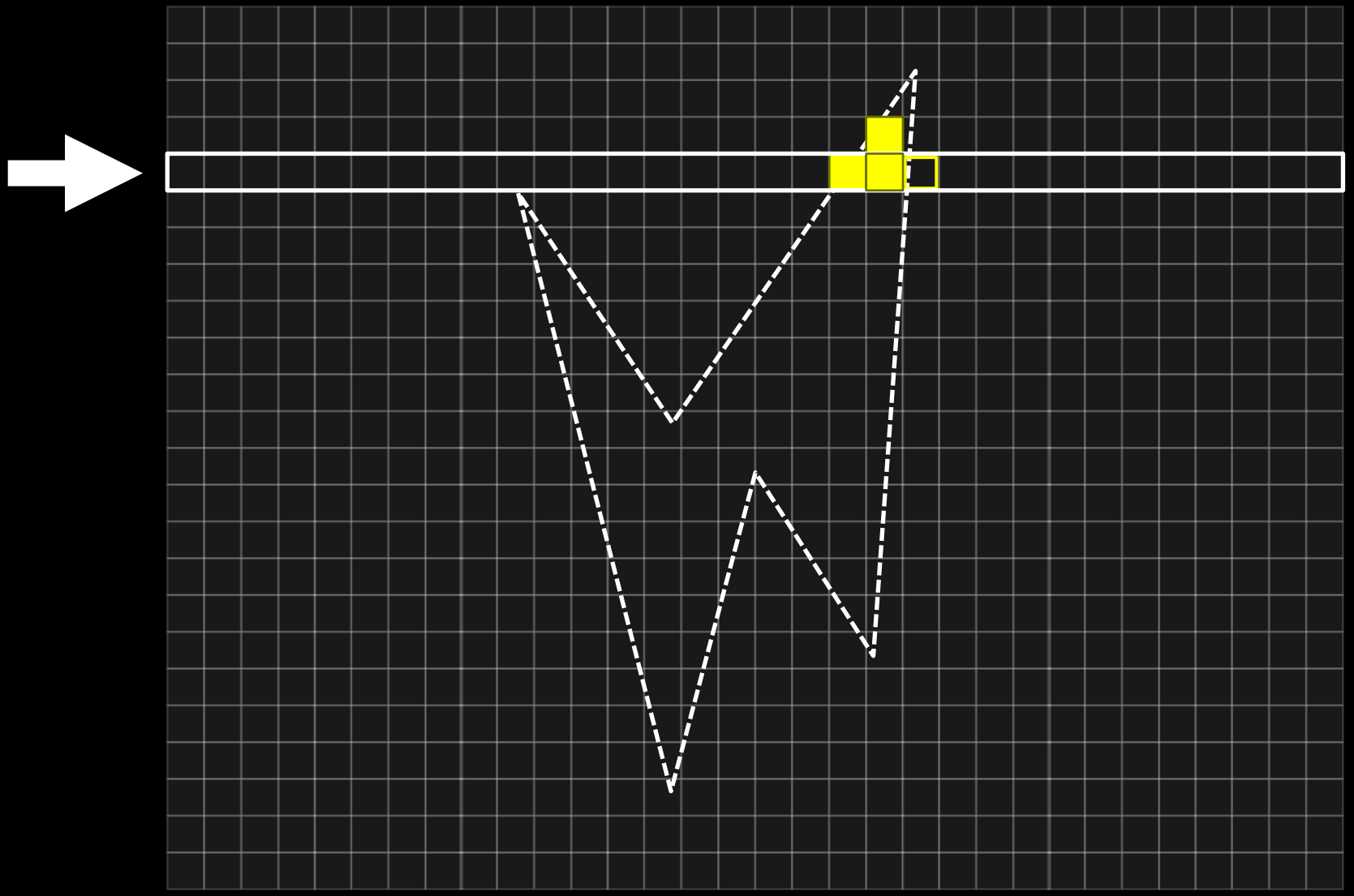


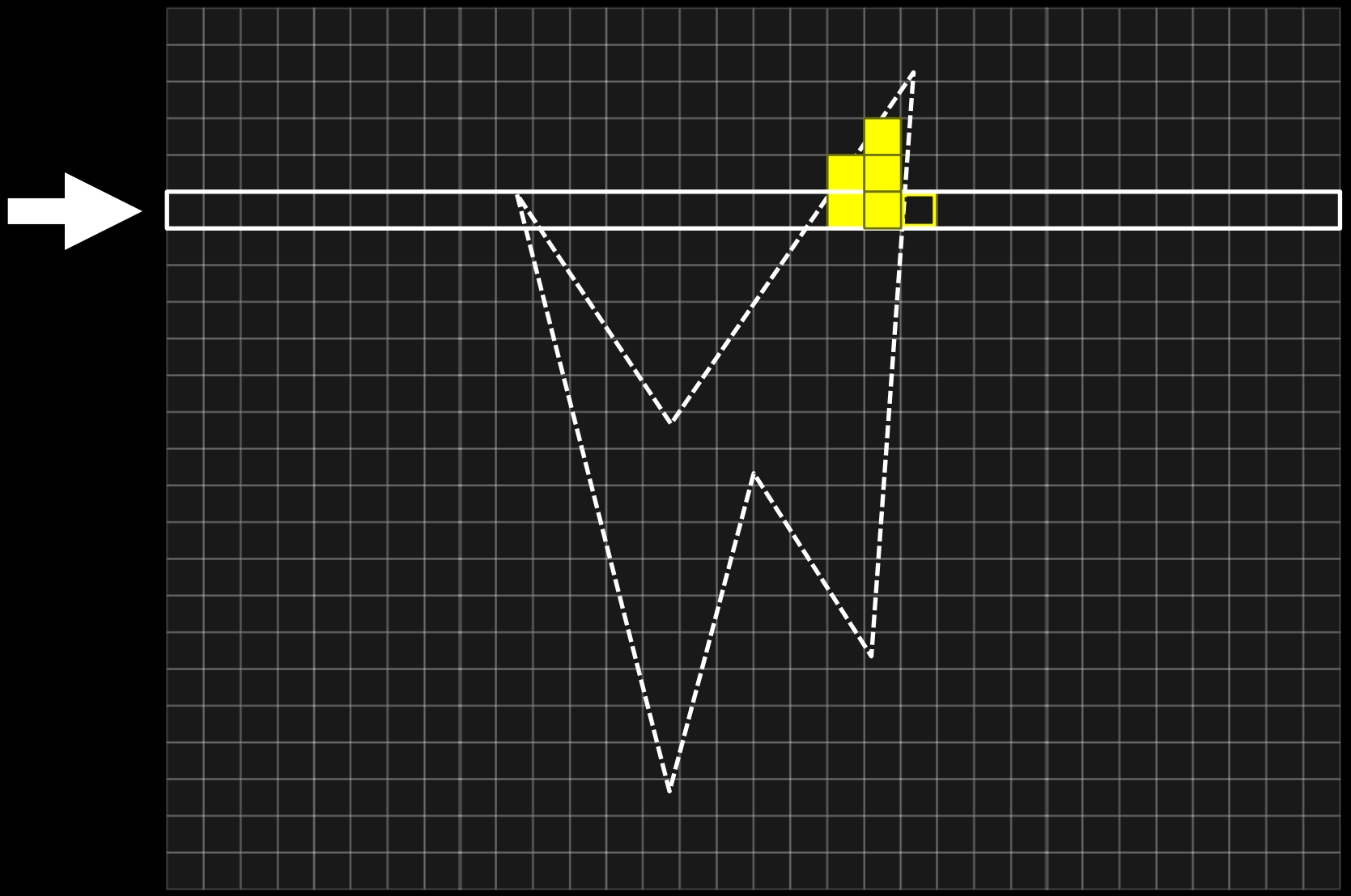


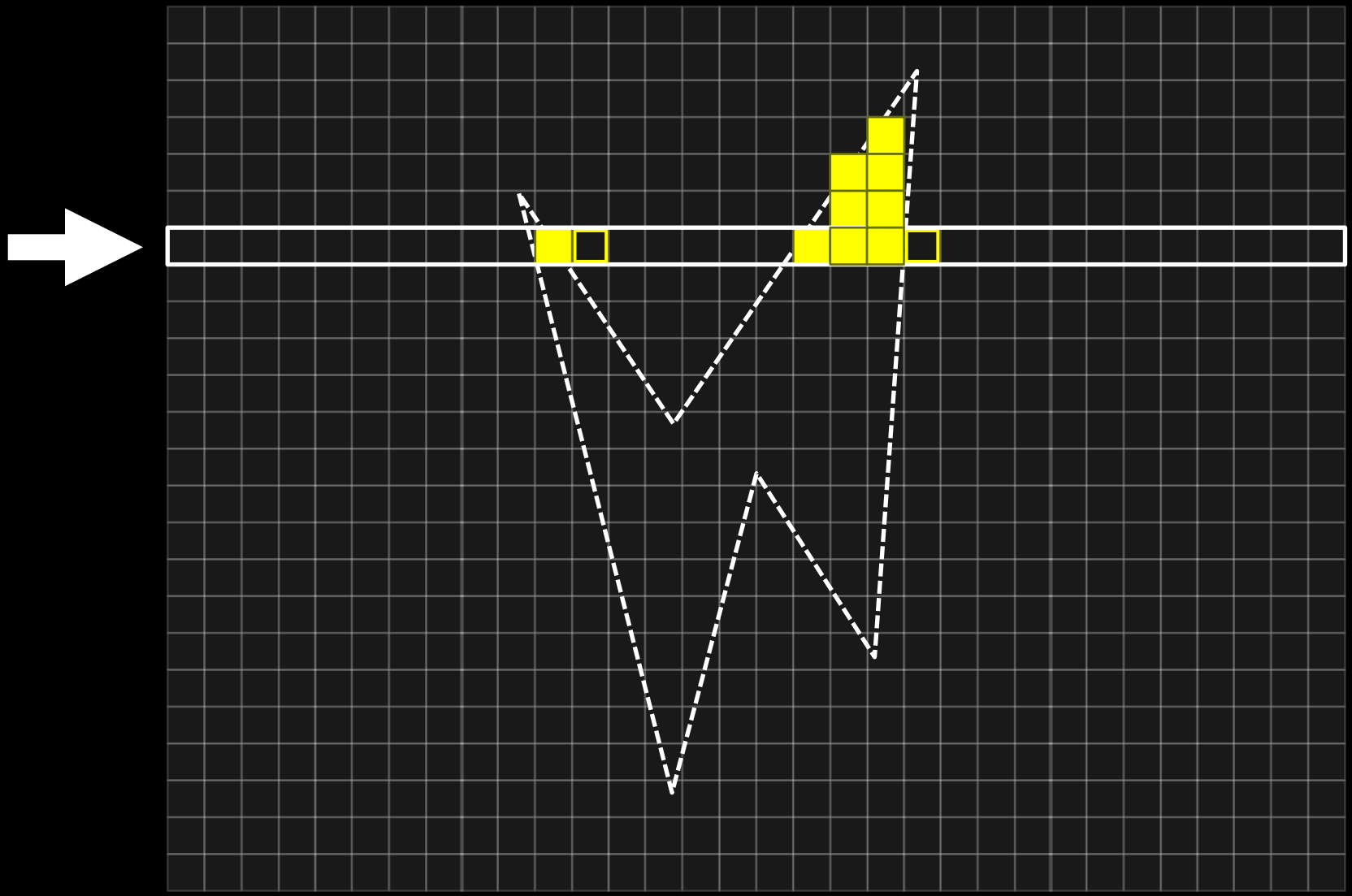


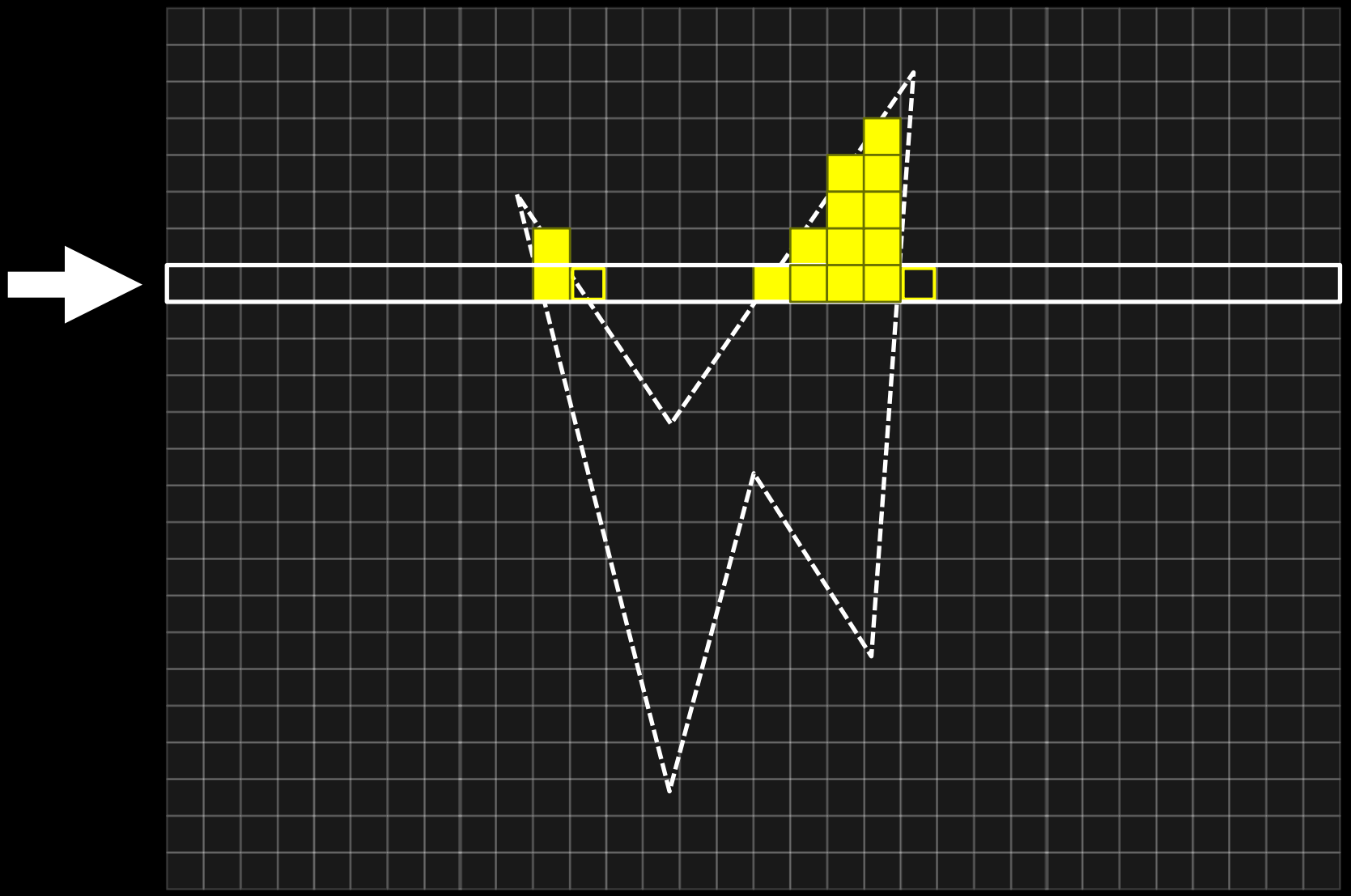


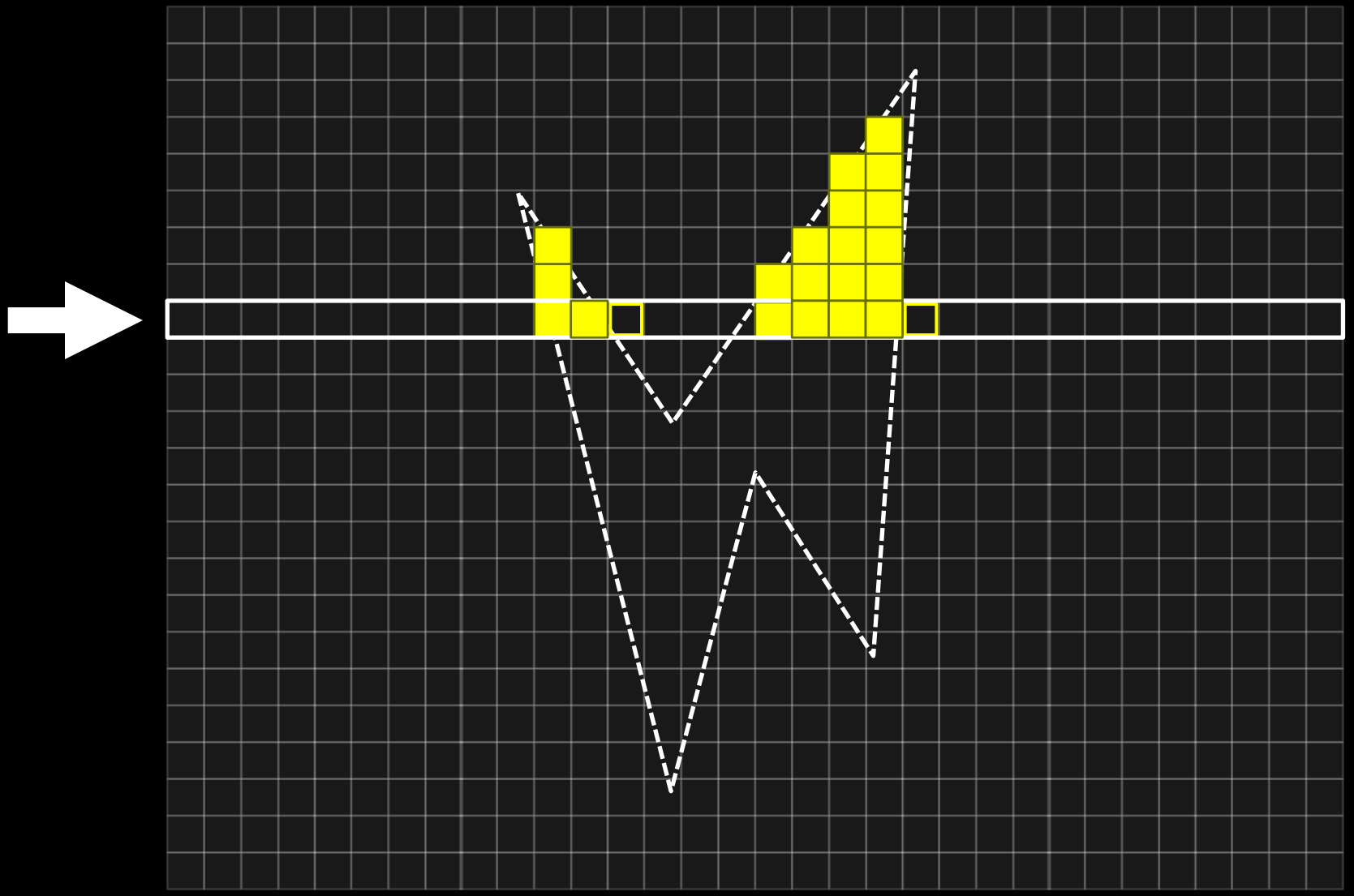


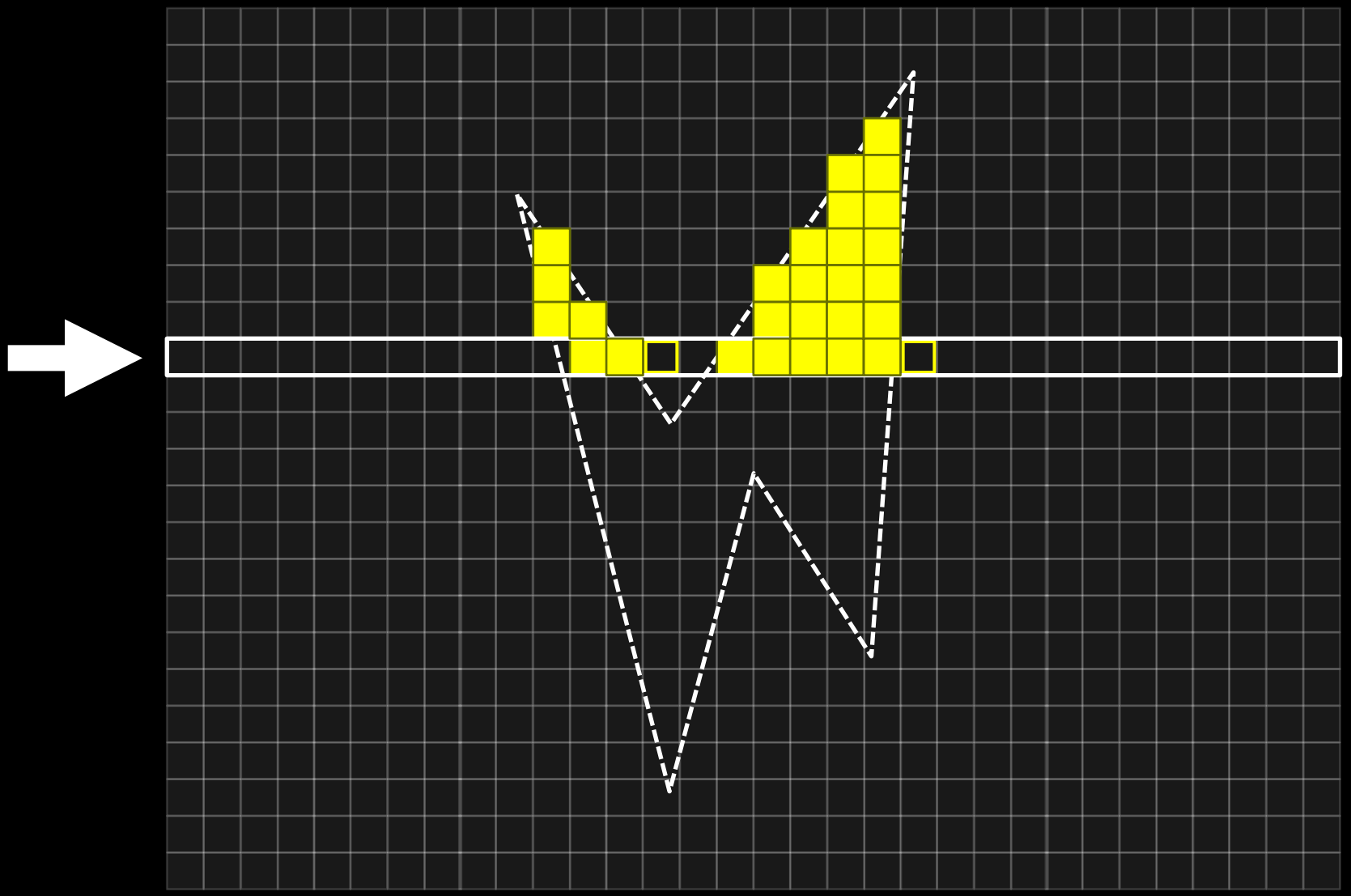


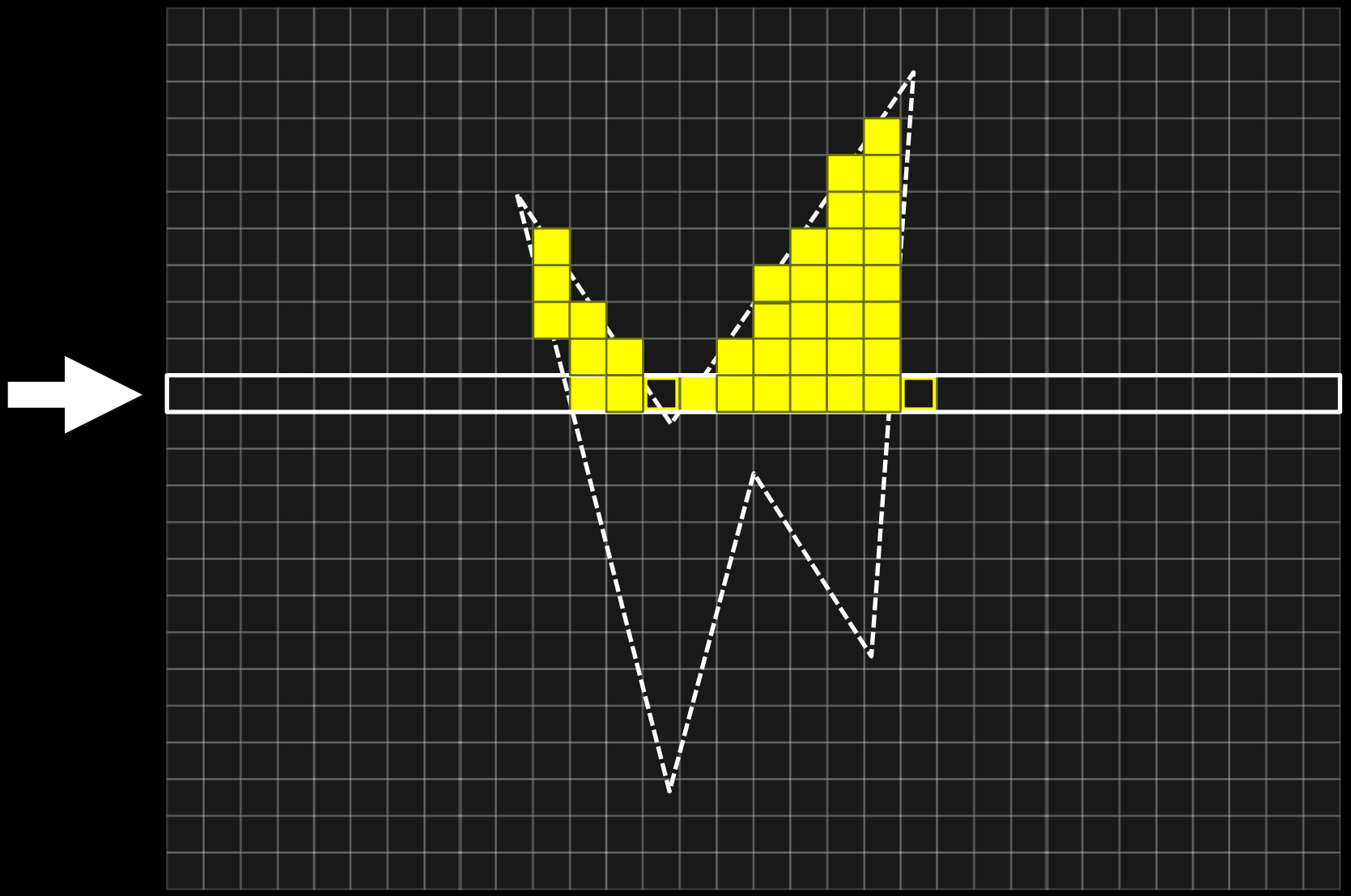


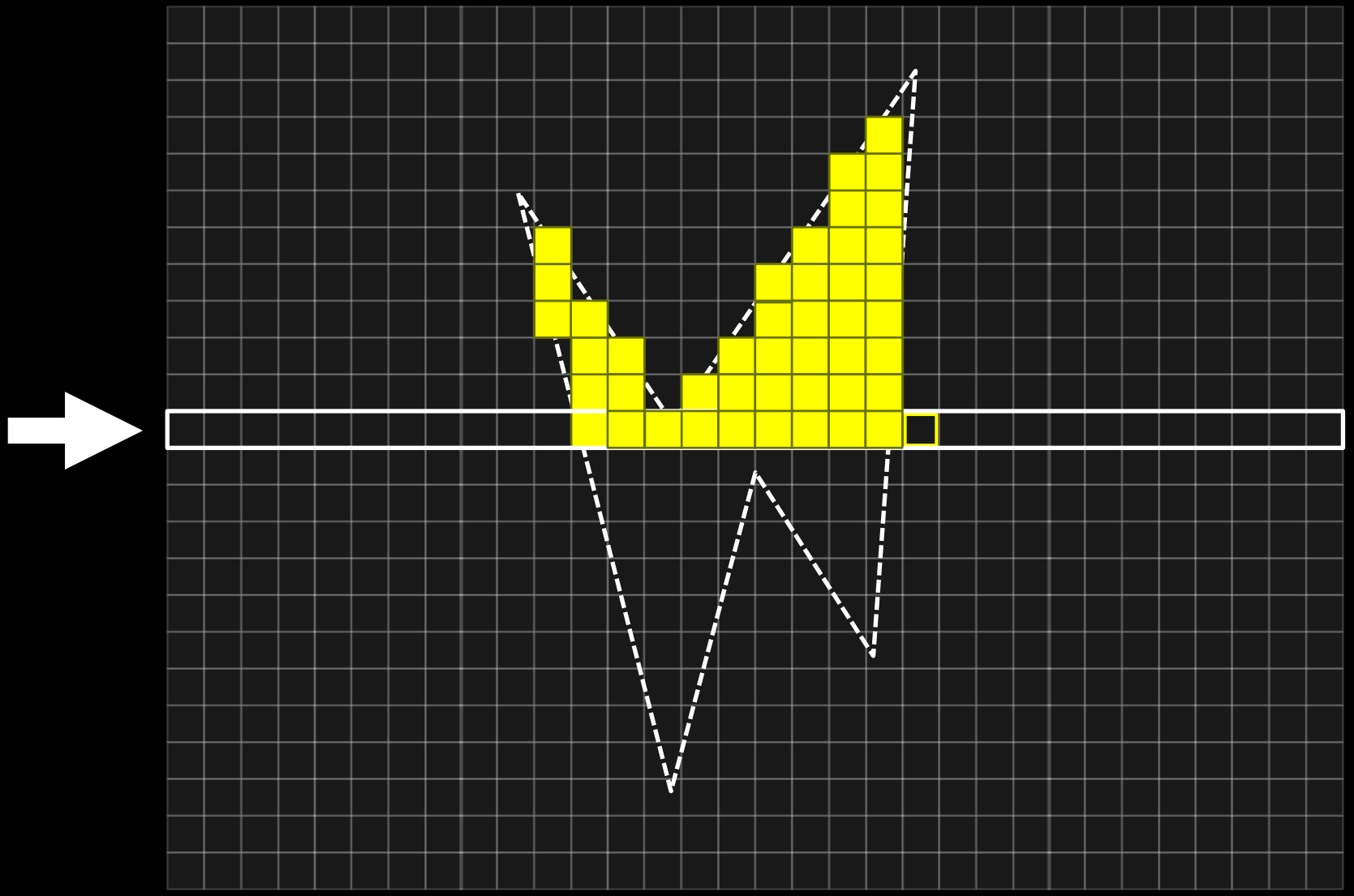


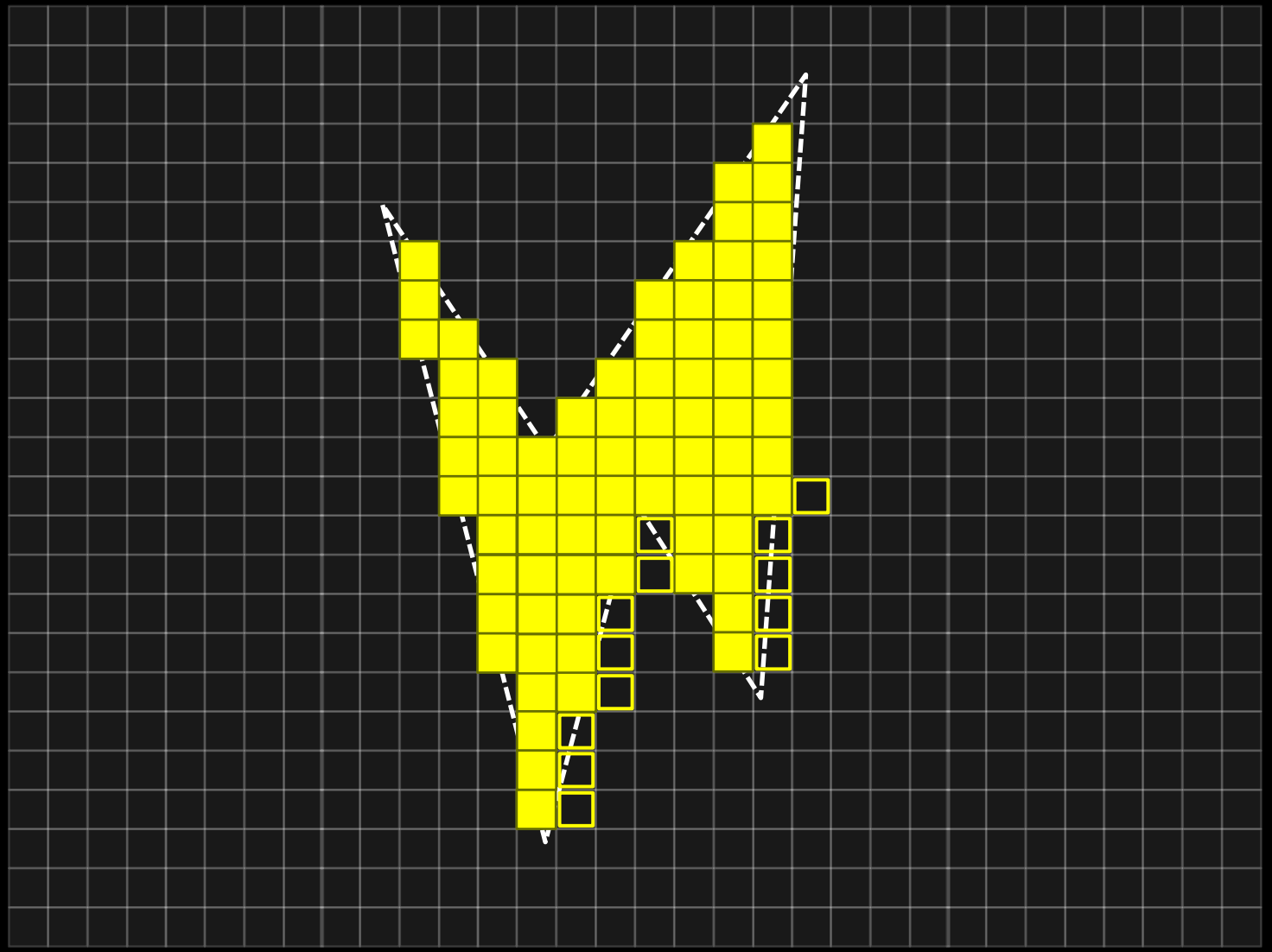




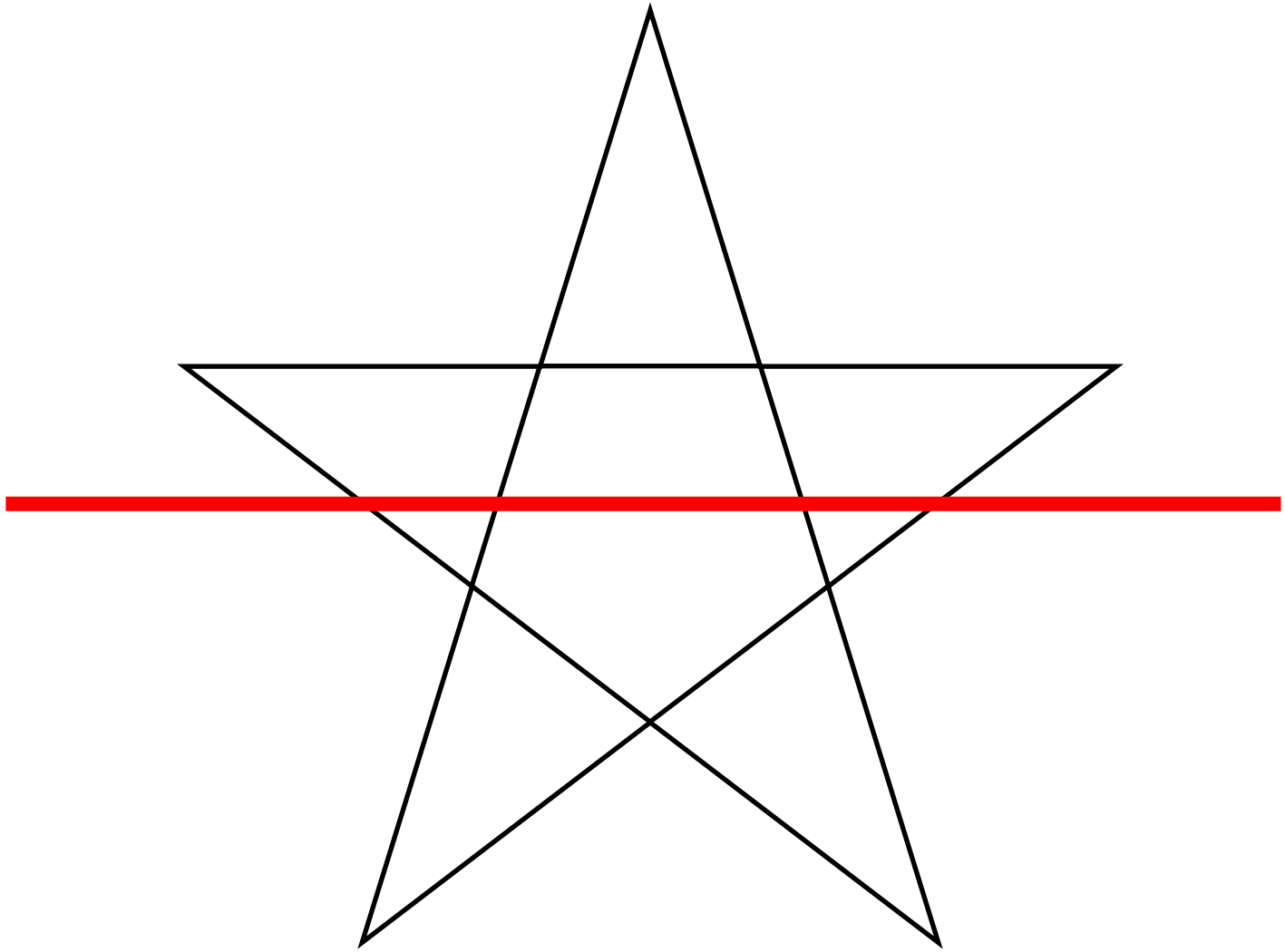




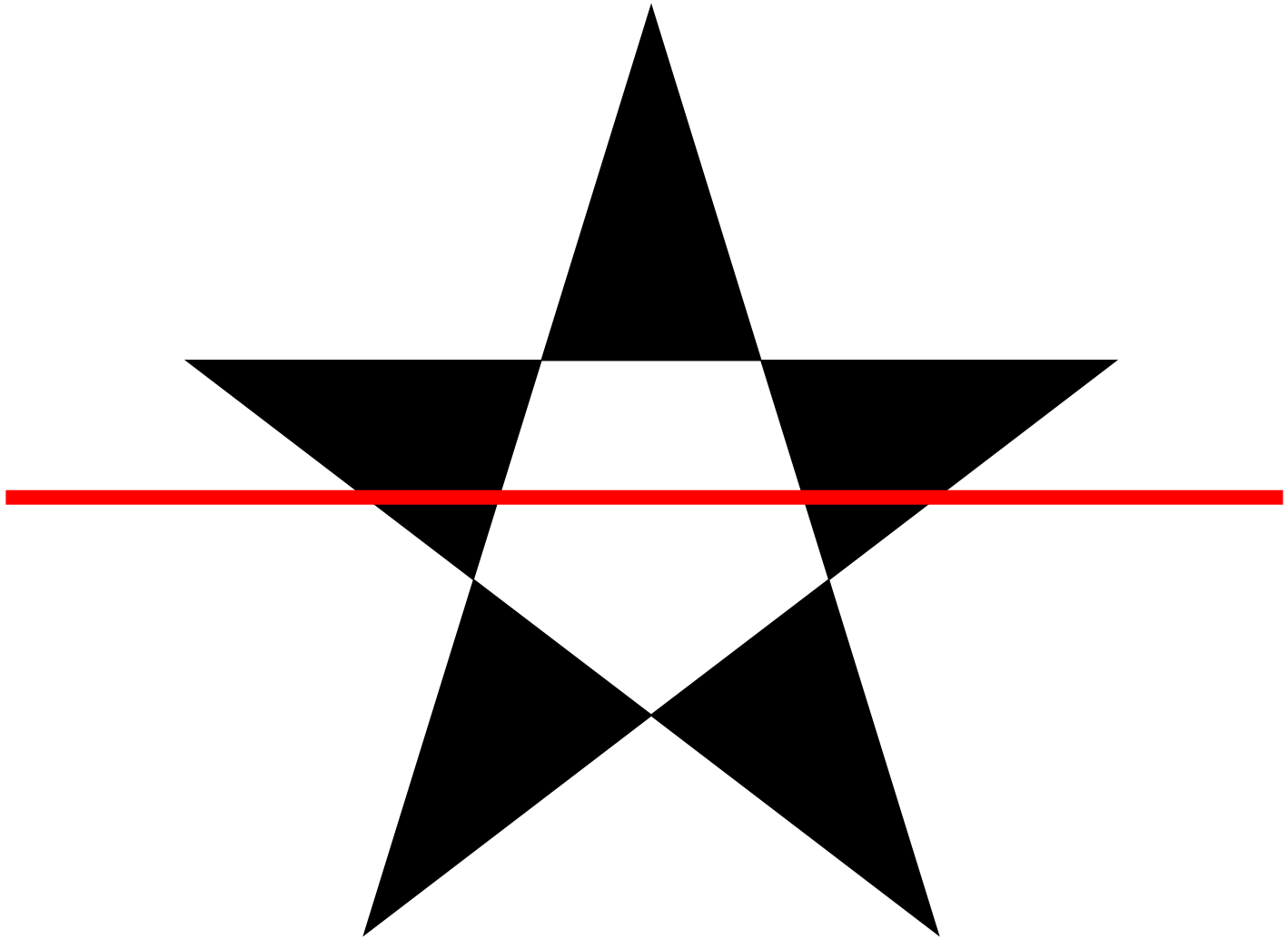


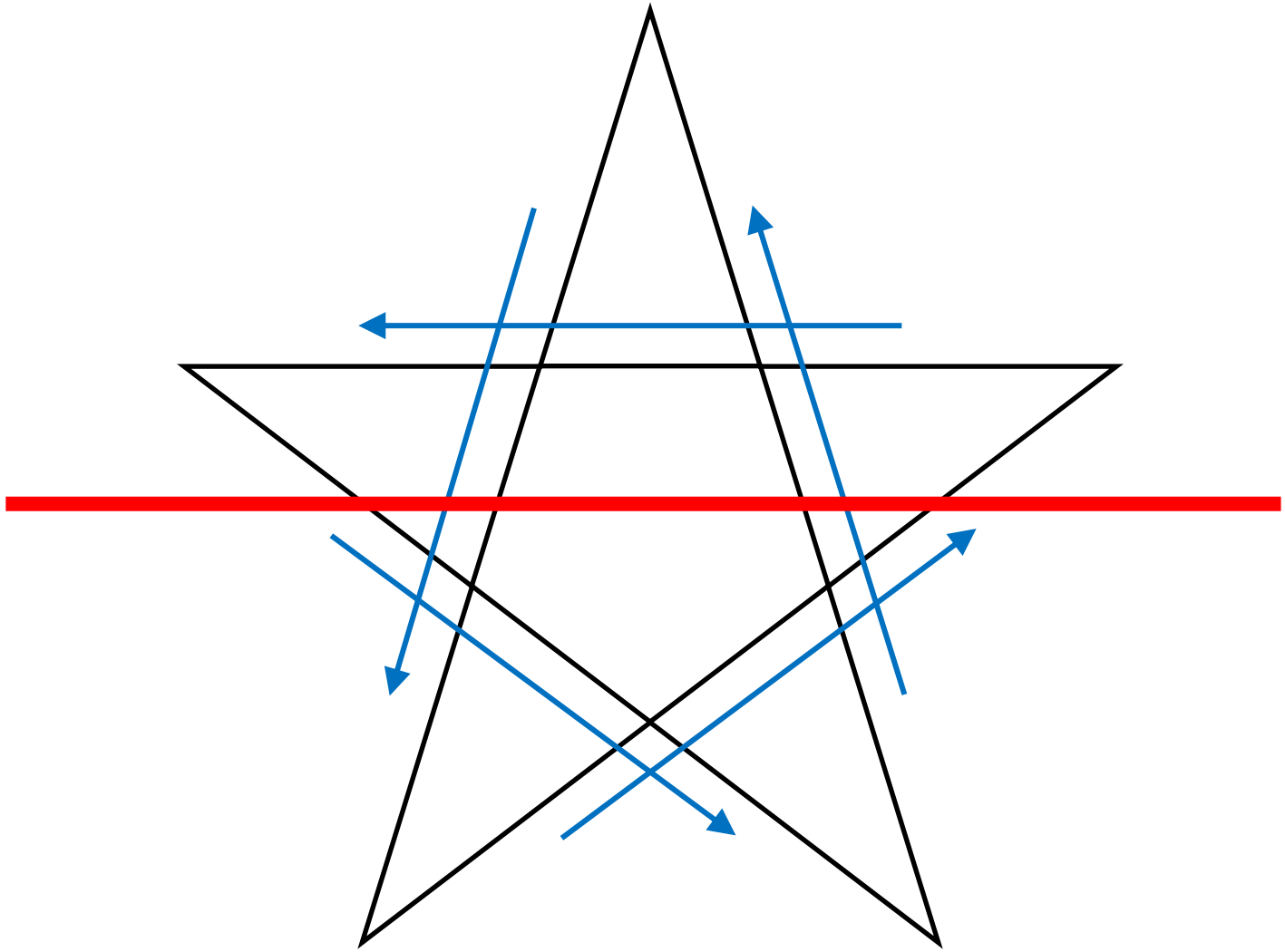


Чётное-нечётное

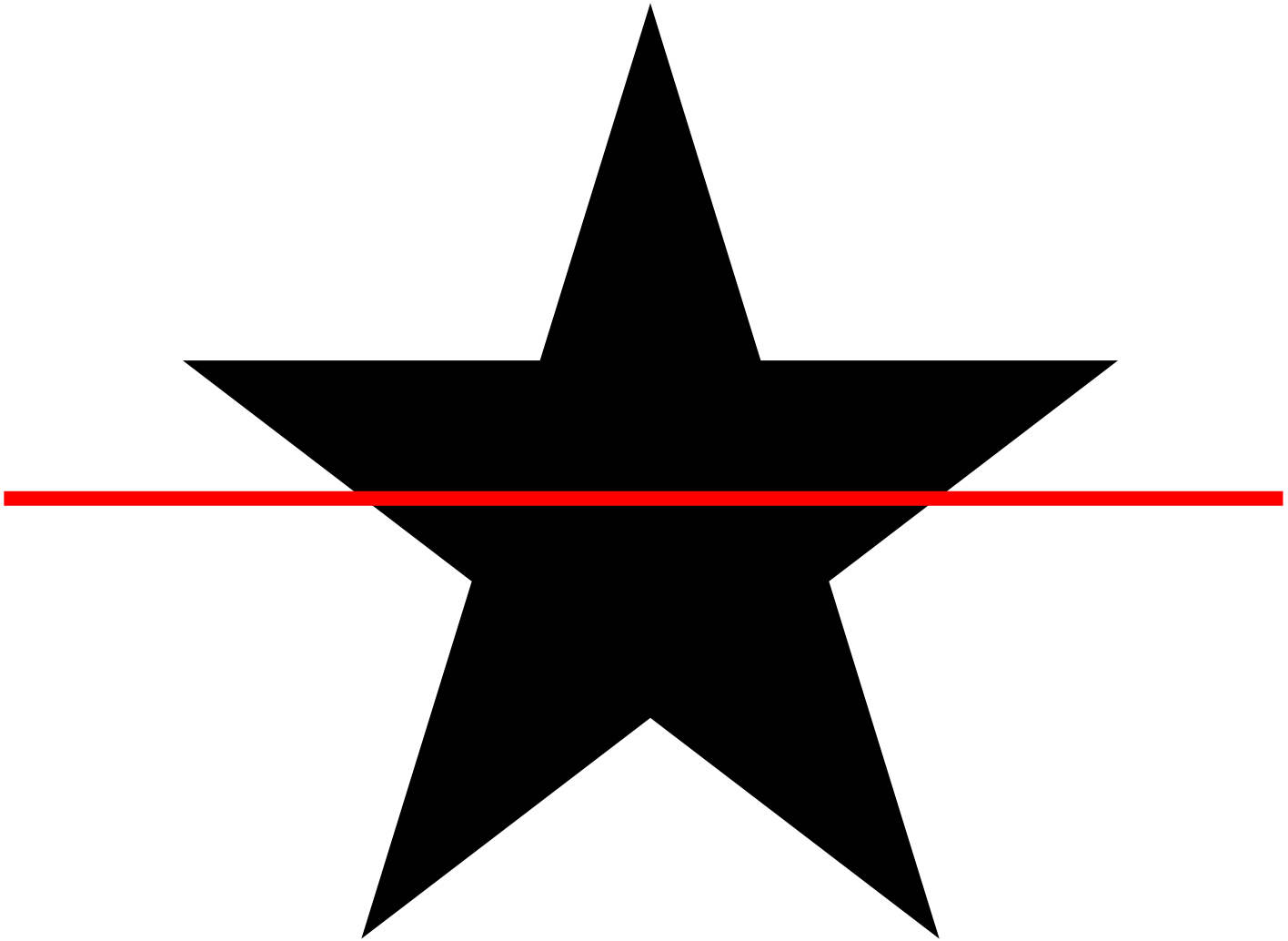


Чётное-нечётное

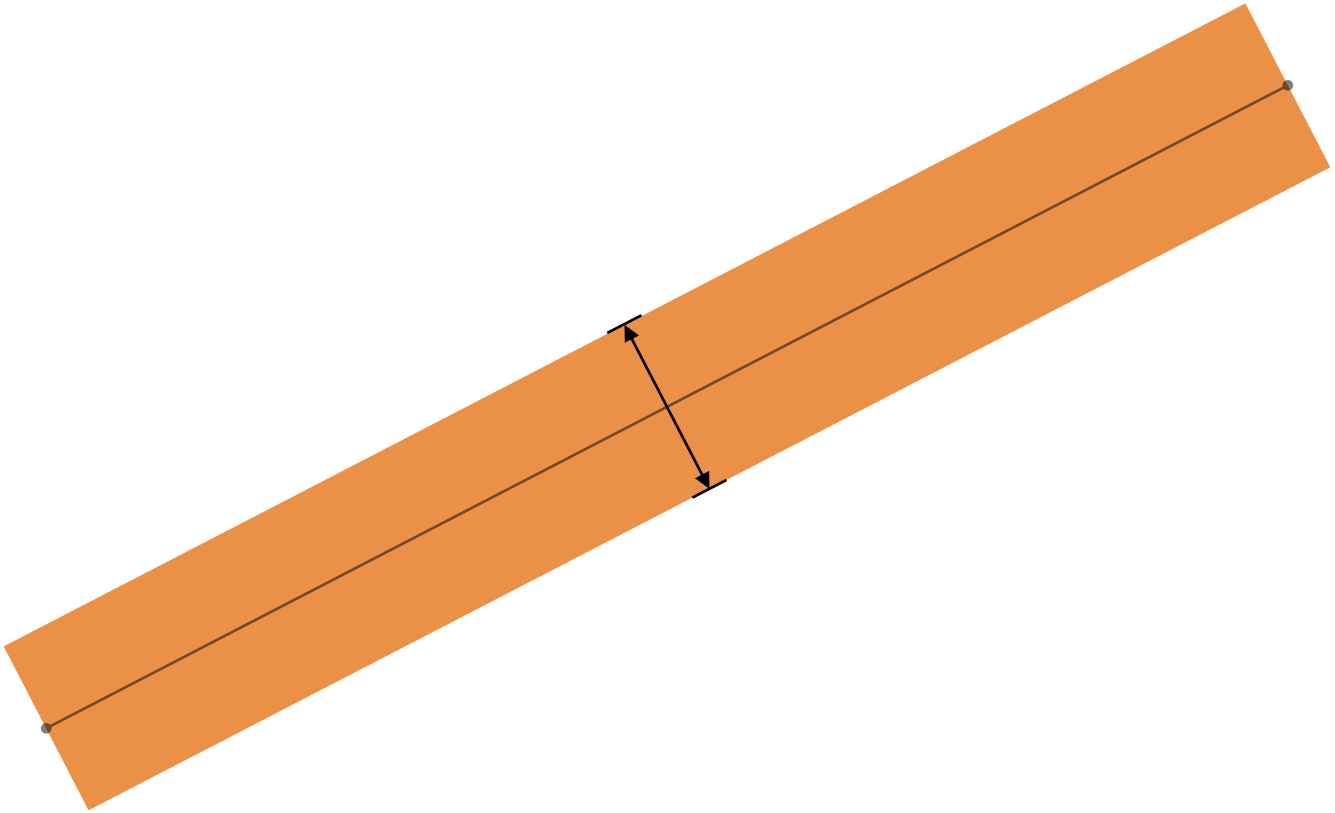




Ненулевая намотка



Толстые линии



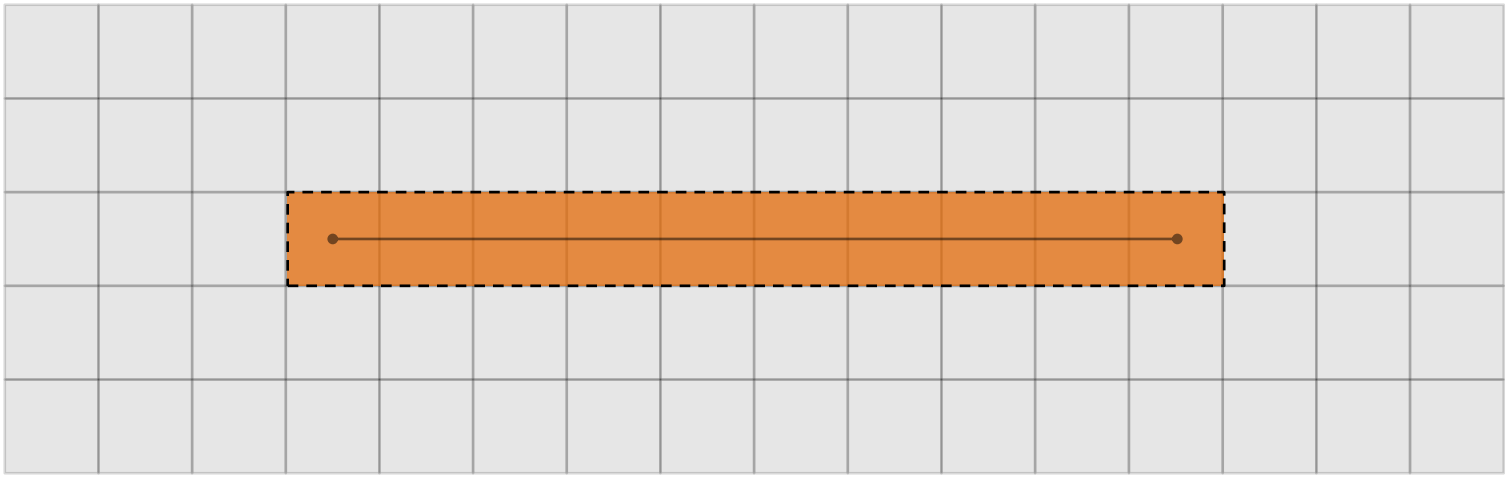
Концы

тупые

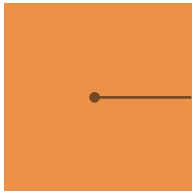
квадратные

круглые

Опять координаты

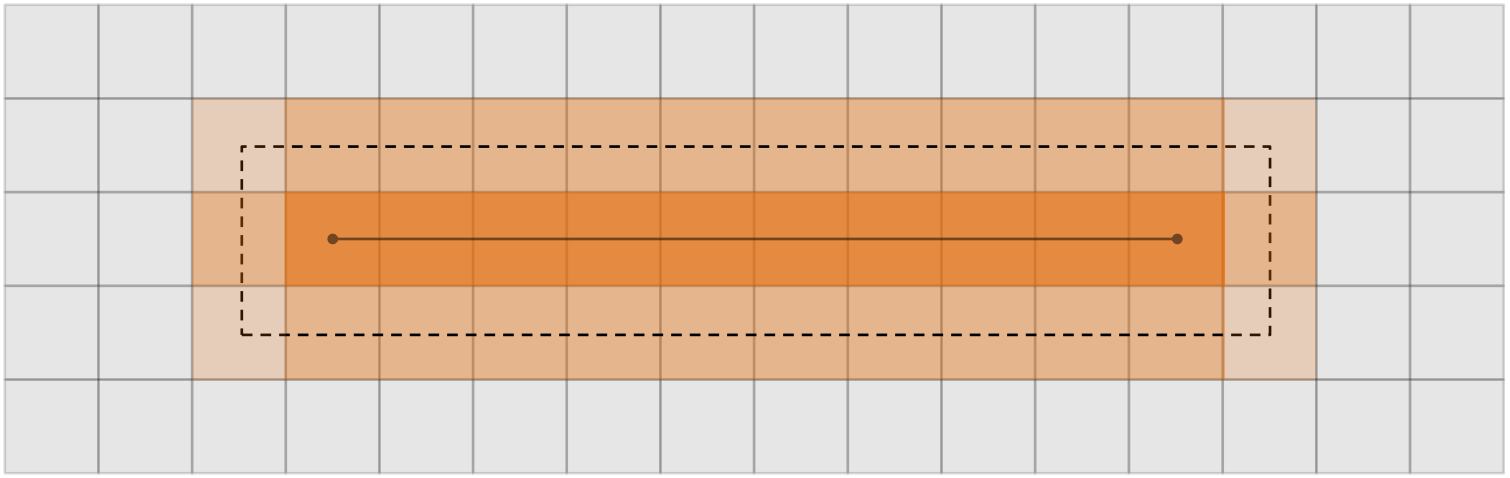


```
void DrawLine(FPoint p1, FPoint p2, float width)
{
    p1 += FPoint(0.5, 0.5);
    p2 += FPoint(0.5, 0.5);

    // ...
}
```

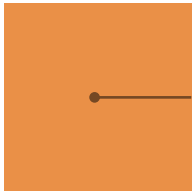


```
DrawLine(FPoint(3, 2), FPoint(12, 2), 1);
```

Опять координаты

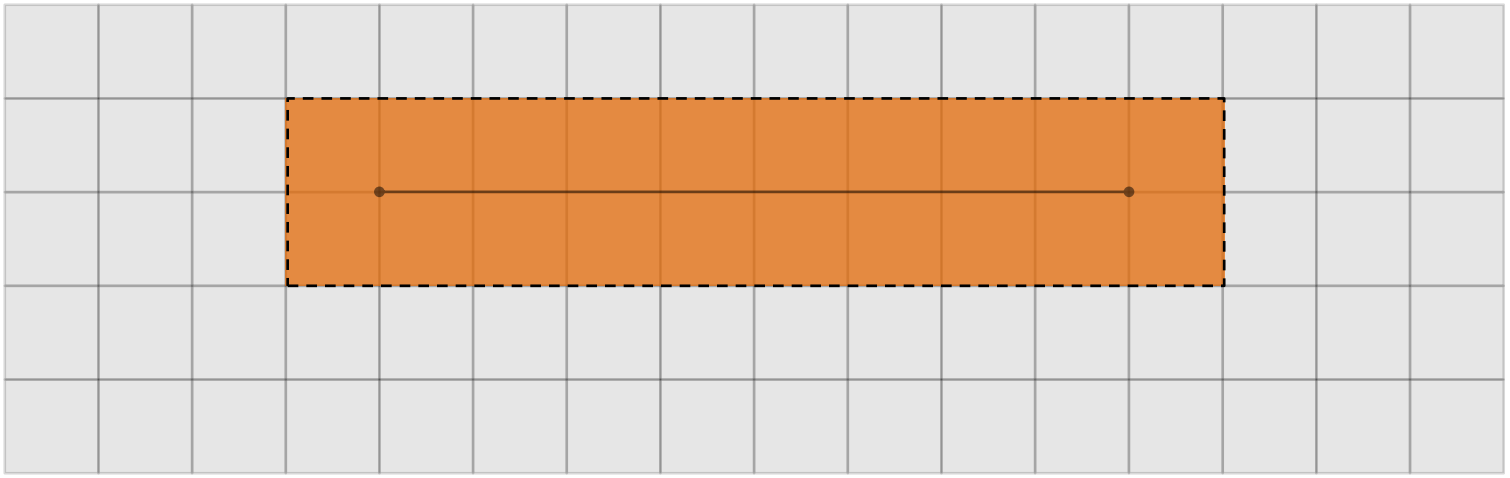


```
void DrawLine(FPoint p1, FPoint p2, float width)
{
    p1 += FPoint(0.5, 0.5);
    p2 += FPoint(0.5, 0.5);

    // ...
}
```



```
DrawLine(FPoint(3, 2), FPoint(12, 2), 2);
```

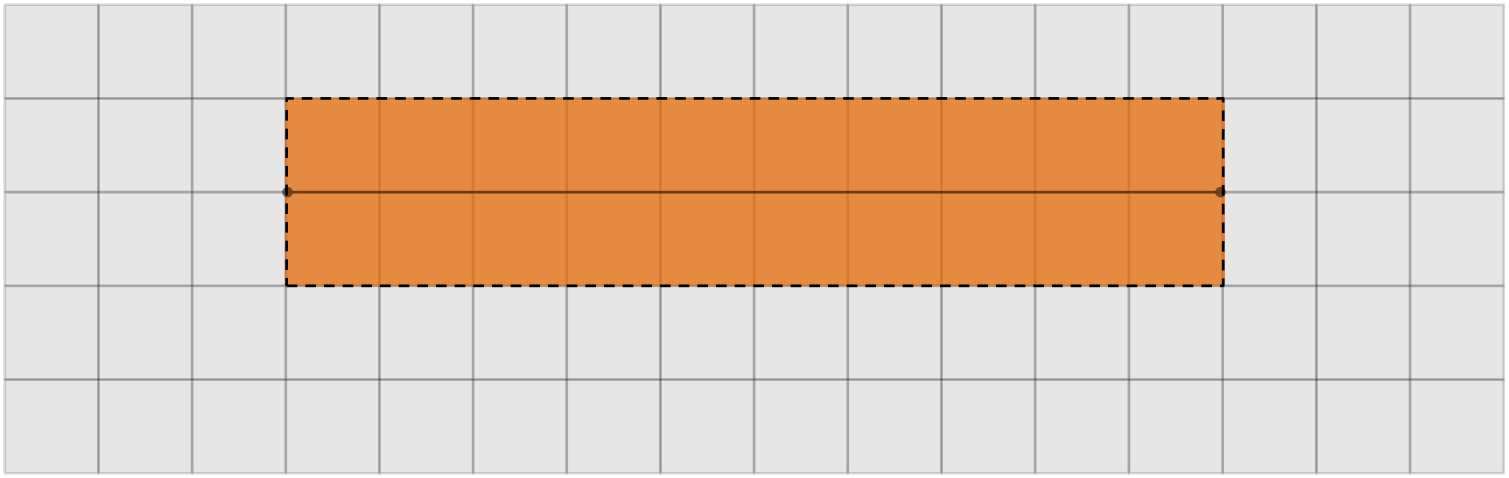
Опять координаты



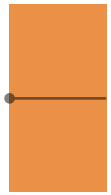
```
void DrawLine(FPoint p1, FPoint p2, float width)
{
    p1 += FPoint(0.5, 0.5);
    p2 += FPoint(0.5, 0.5);

    // ...
}
```

```
DrawLine(FPoint(3.5, 1.5), FPoint(11.5, 1.5), 2);
```

Опять координаты

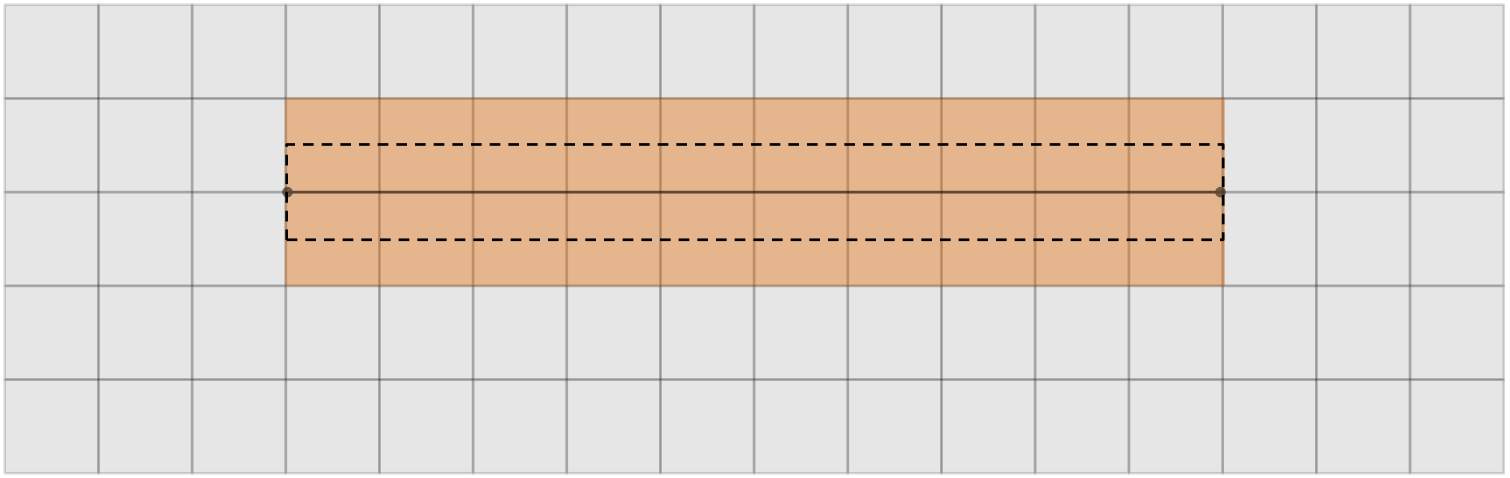


```
void DrawLine(FPoint p1, FPoint p2, float width)
{
    // ...
}
```

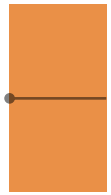


```
DrawLine(FPoint(3, 2), FPoint(13, 2), 2);
```

Опять координаты

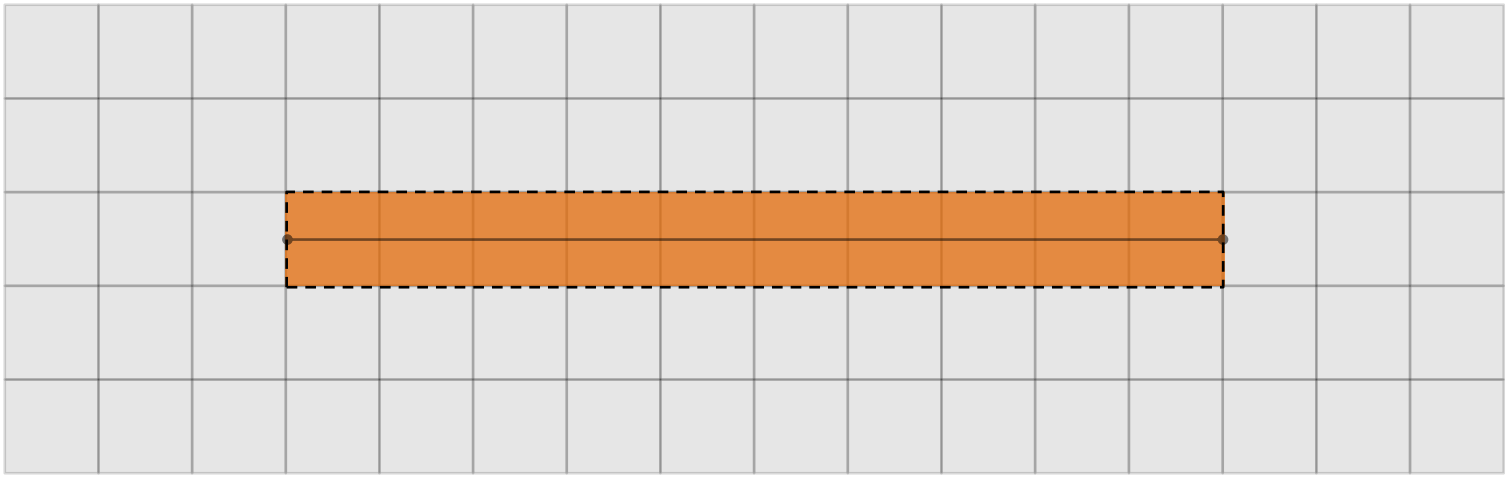


```
void DrawLine(FPoint p1, FPoint p2, float width)
{
    // ...
}
```

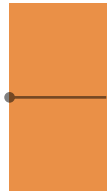


```
DrawLine(FPoint(3, 2), FPoint(13, 2), 1);
```

Опять координаты



```
void DrawLine(FPoint p1, FPoint p2, float width)
{
    // ...
}
```



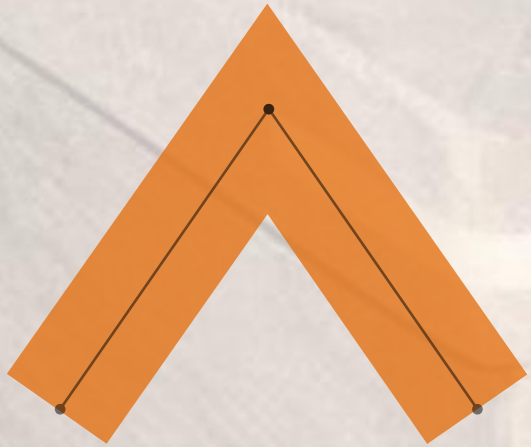
```
DrawLine(FPoint(3, 2.5), FPoint(13, 2.5), 1);
```

Толстые полилинии

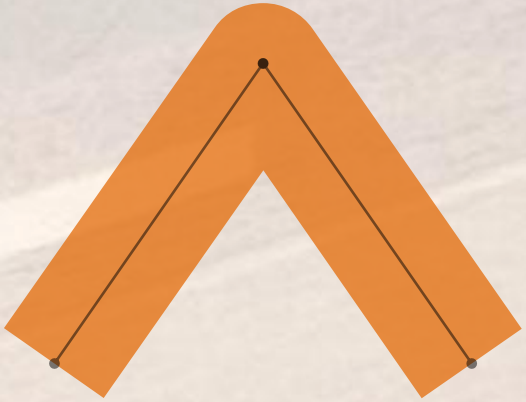
Стыки

фасковый

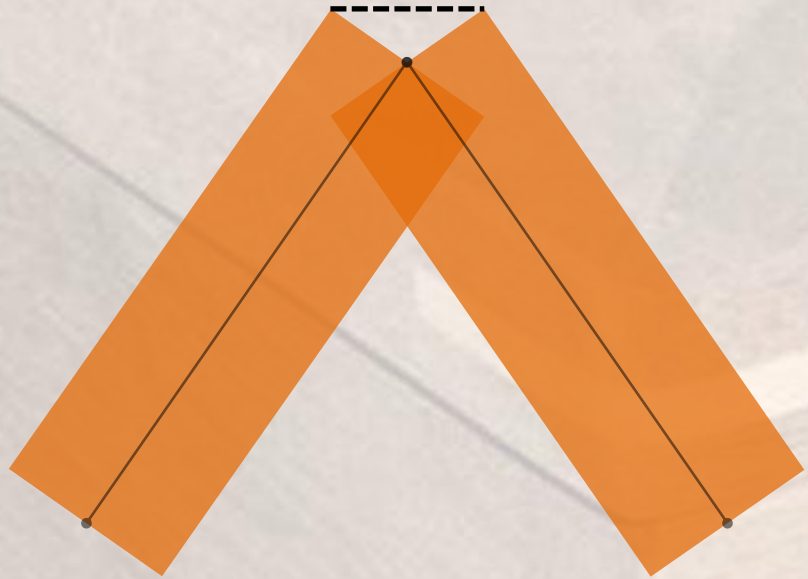
угольный



круглый

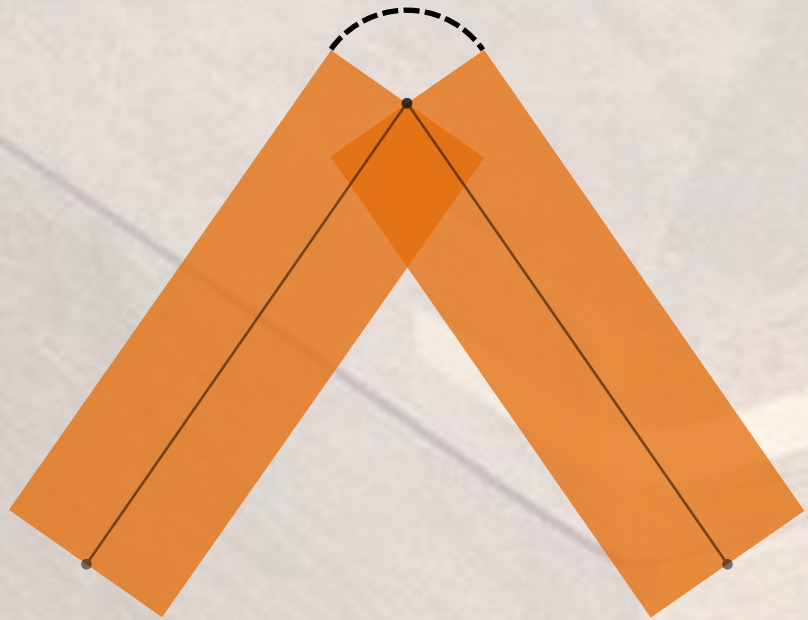


Фасковый стык



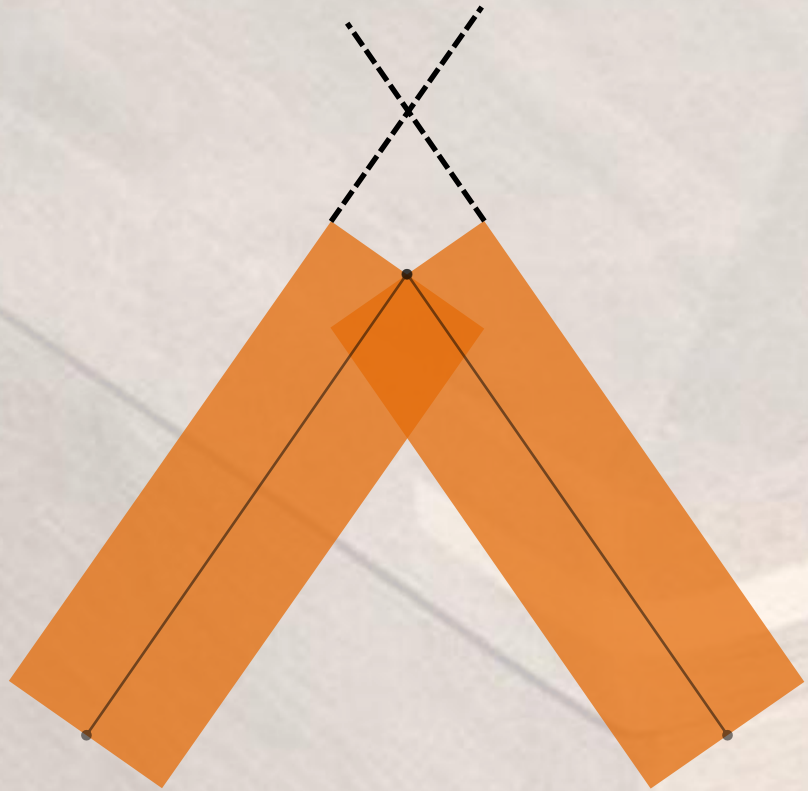
Фасковый стык

Круглый стык



Круглый стык

Угольный стык

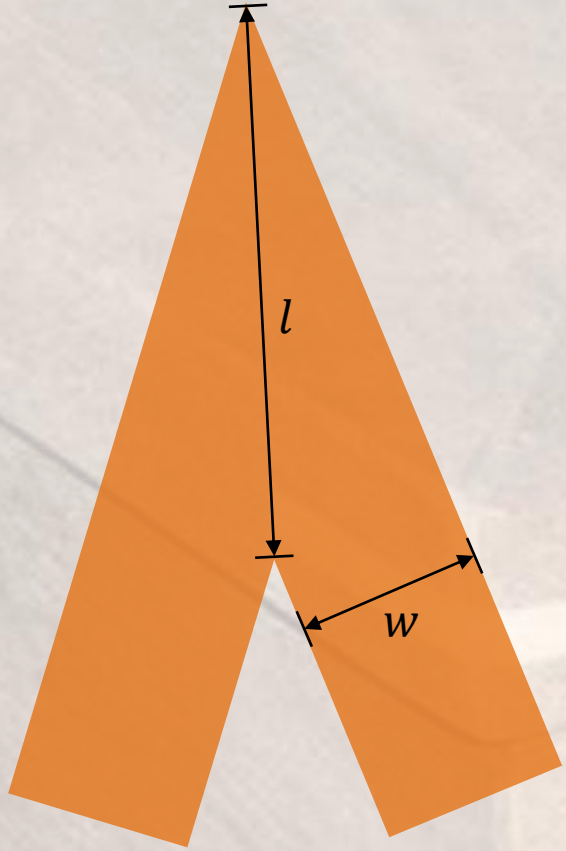


Угольный стык

Угольный стык

Угольный стык

Митровое соотношение

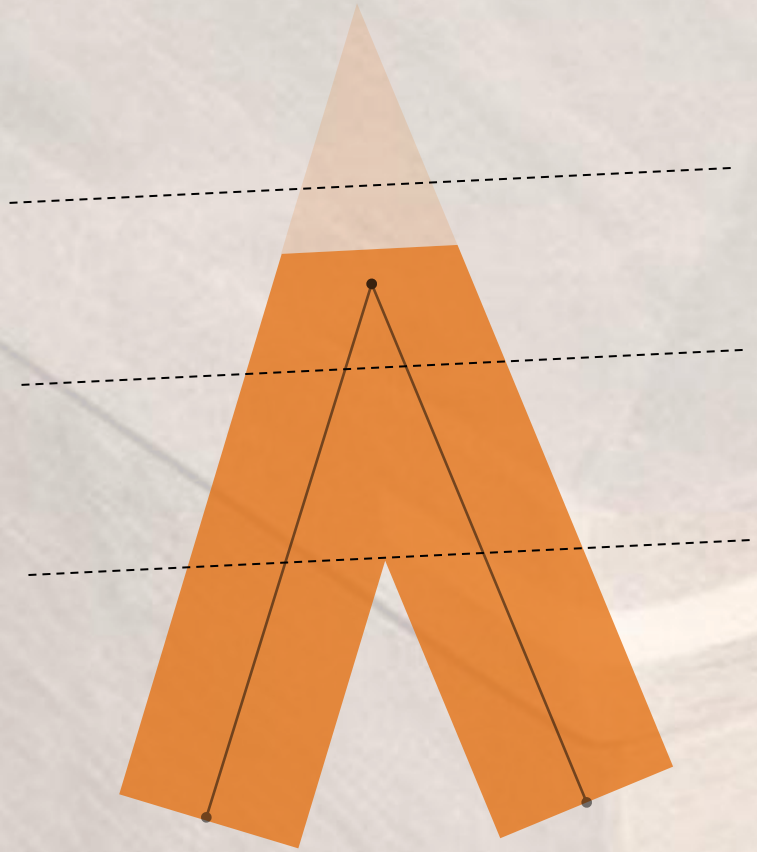


$$M = \frac{l}{w}$$

Митровый лимит

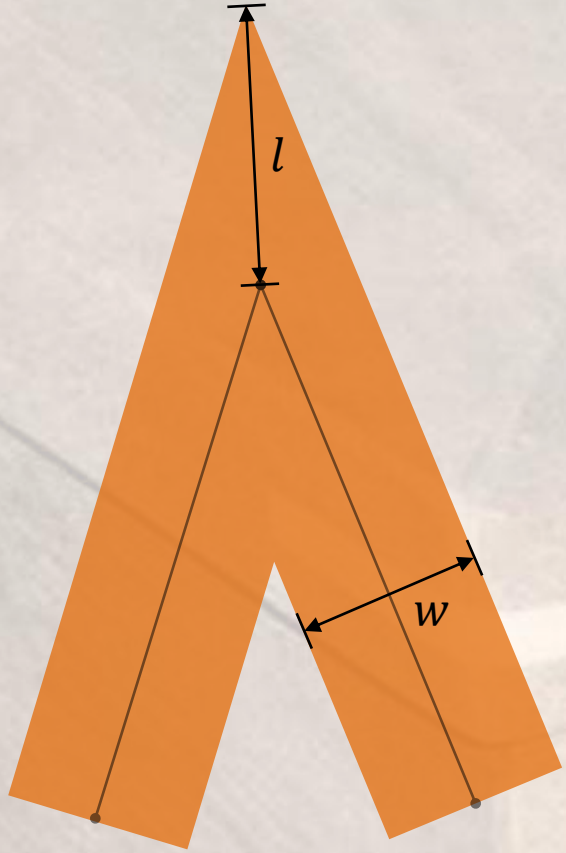
$M \leq 4$

Митровый лимит



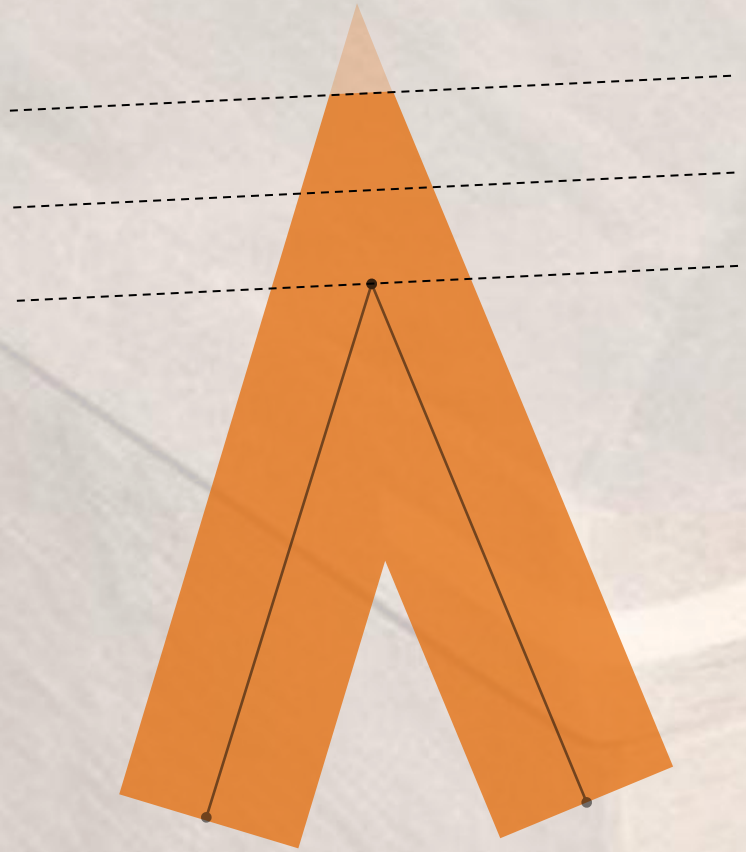
$M \leq 2$

Митровое соотношение (WPF)



$$M = \frac{2l}{w}$$

Митровый лимит (WPF)



$$M \leq 2$$

Толстые полилинии

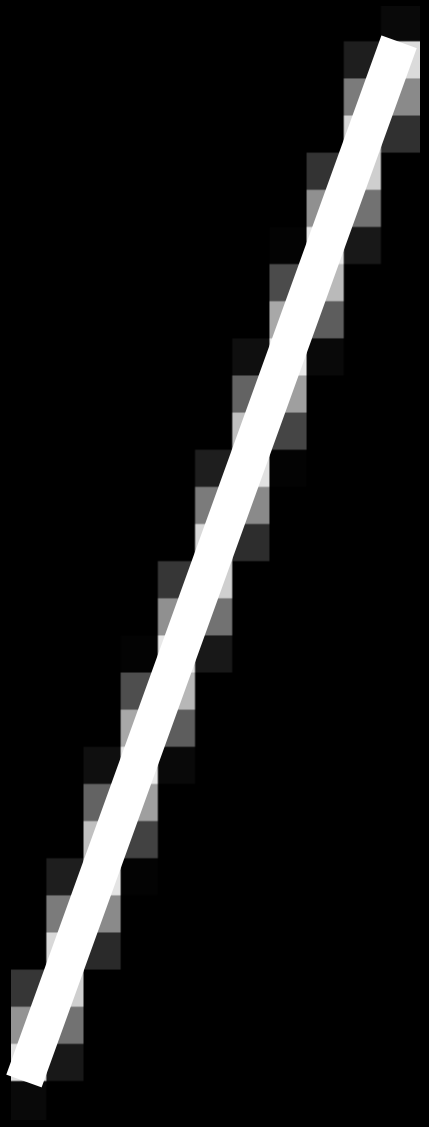
Толстые полилинии

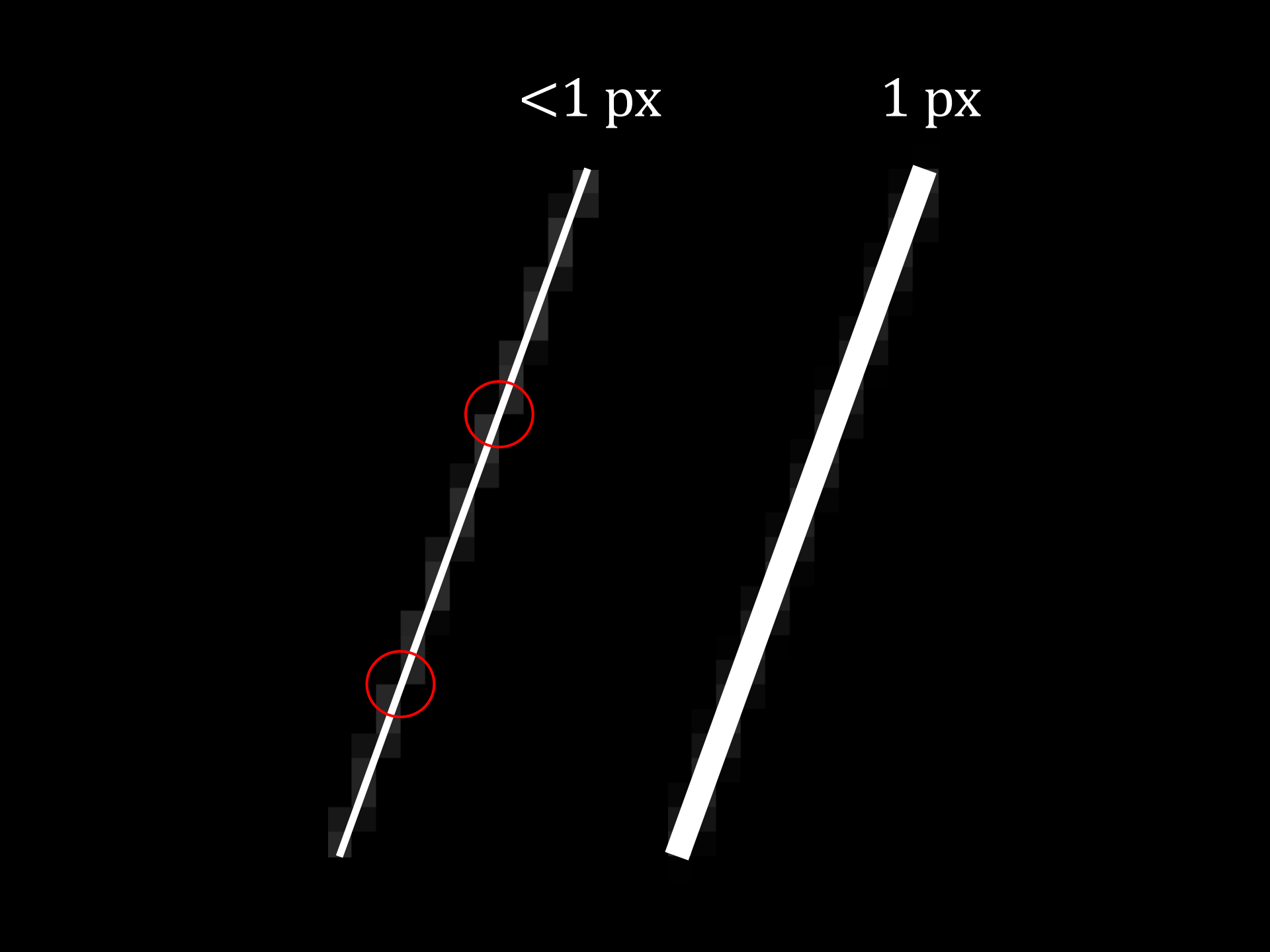
Толстые полилинии

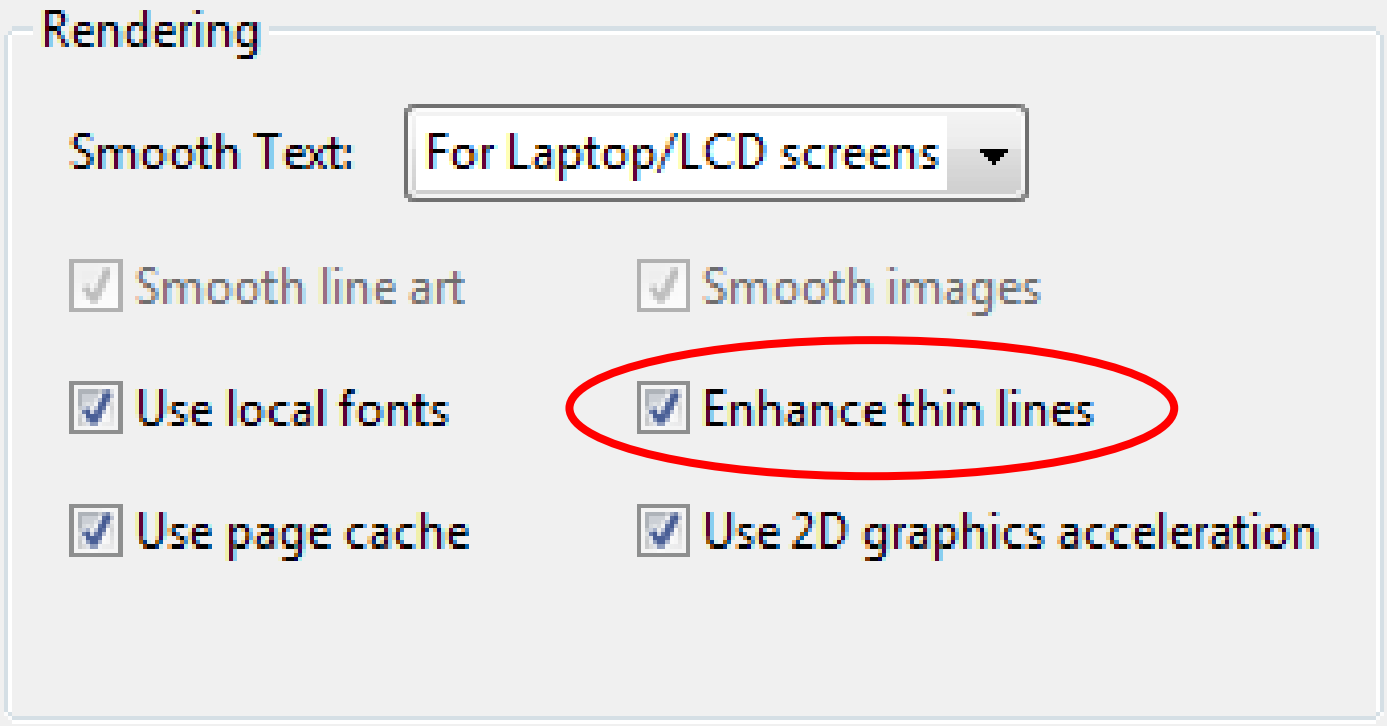
Толстые полилинии

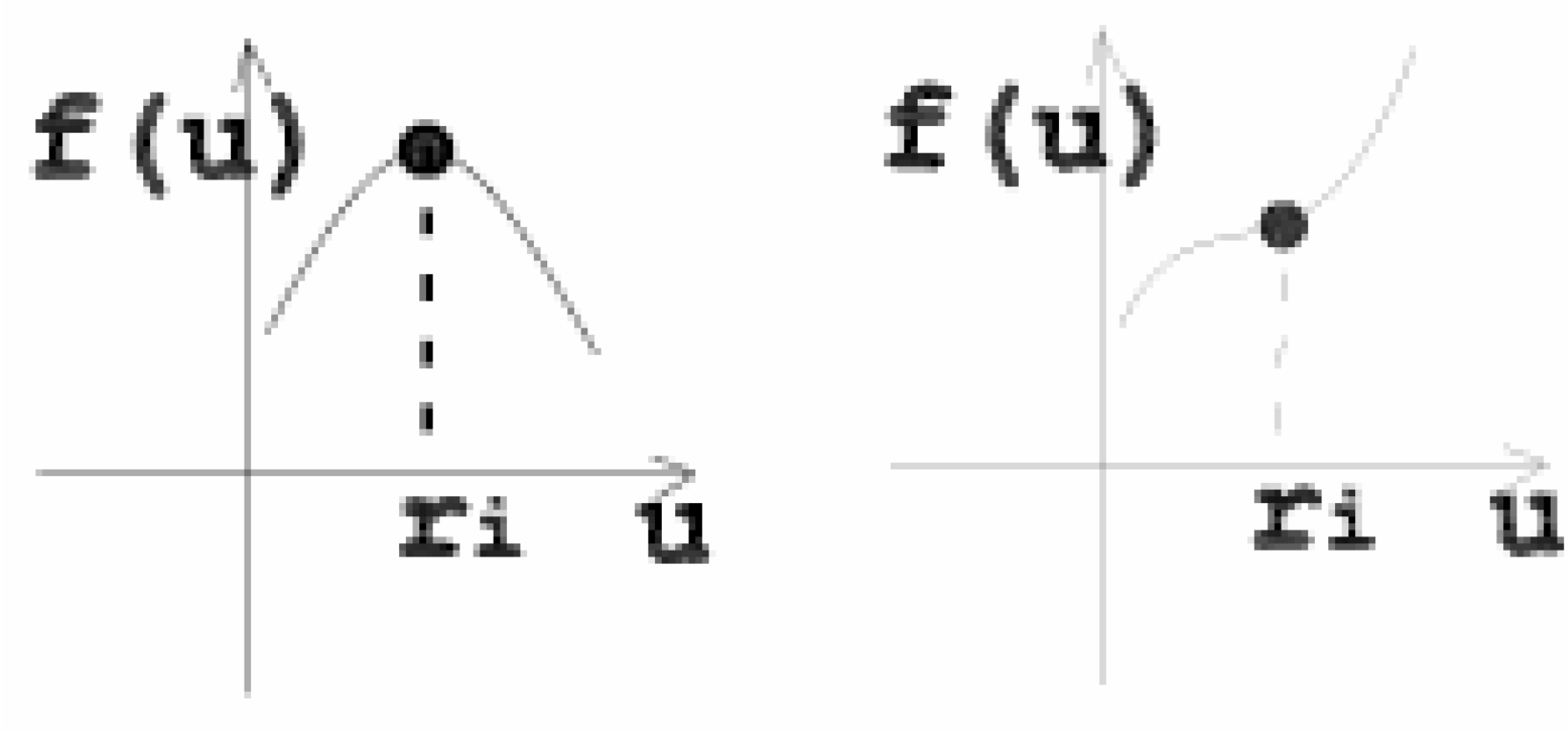
Тонкие линии

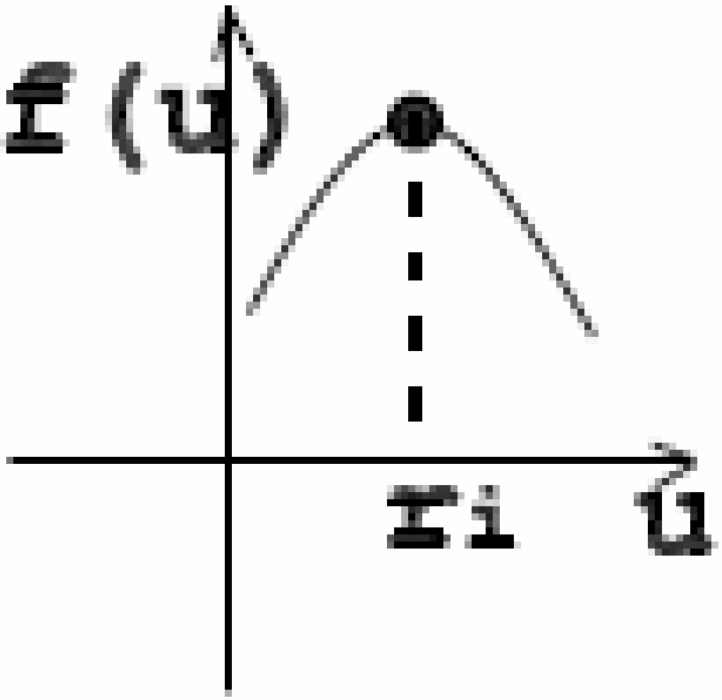
1 px

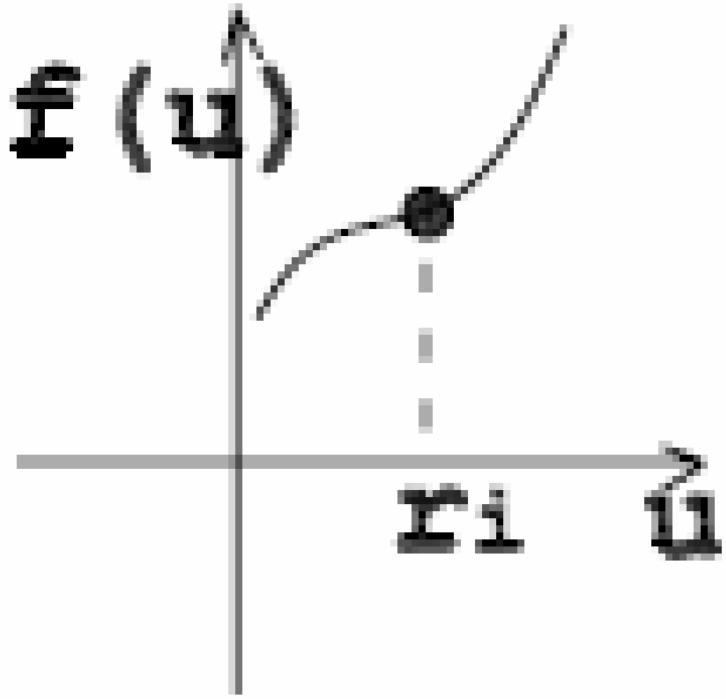



<1 px
1 px


Rendering
Smooth Text:
For Laptop/LCD screens
Smooth line art
Smooth images
Use local fonts
Enhance thin lines
Use page cache
Use 2D graphics acceleration


f(u)
xi
u
f(u)
xi
u

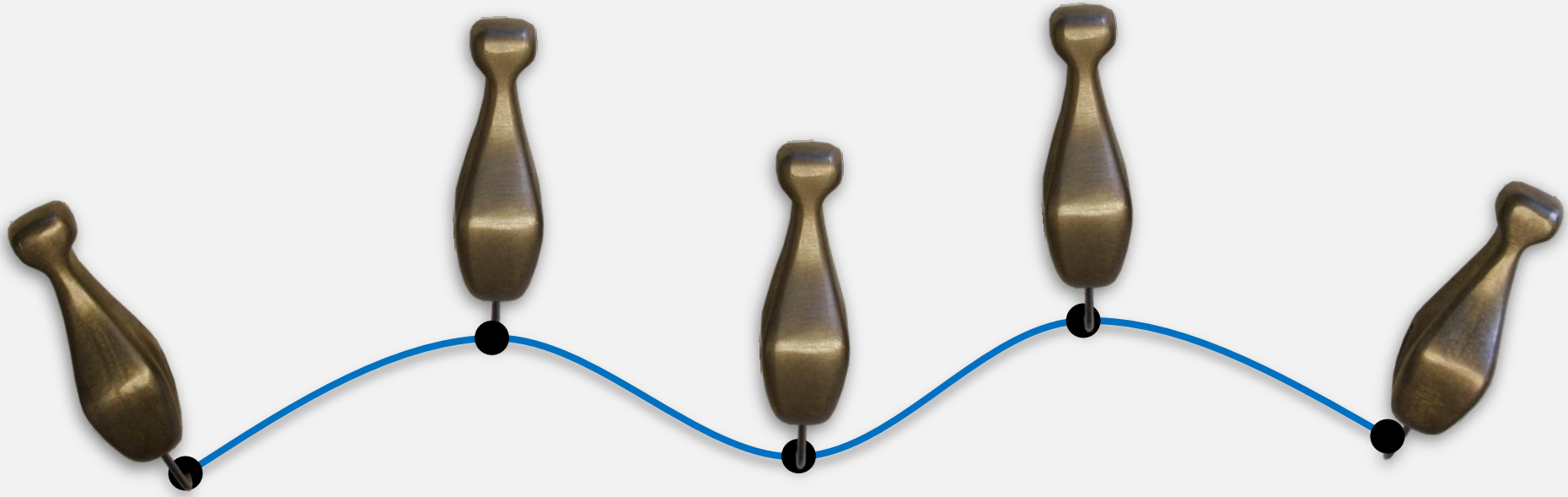

f(u)
xi
u

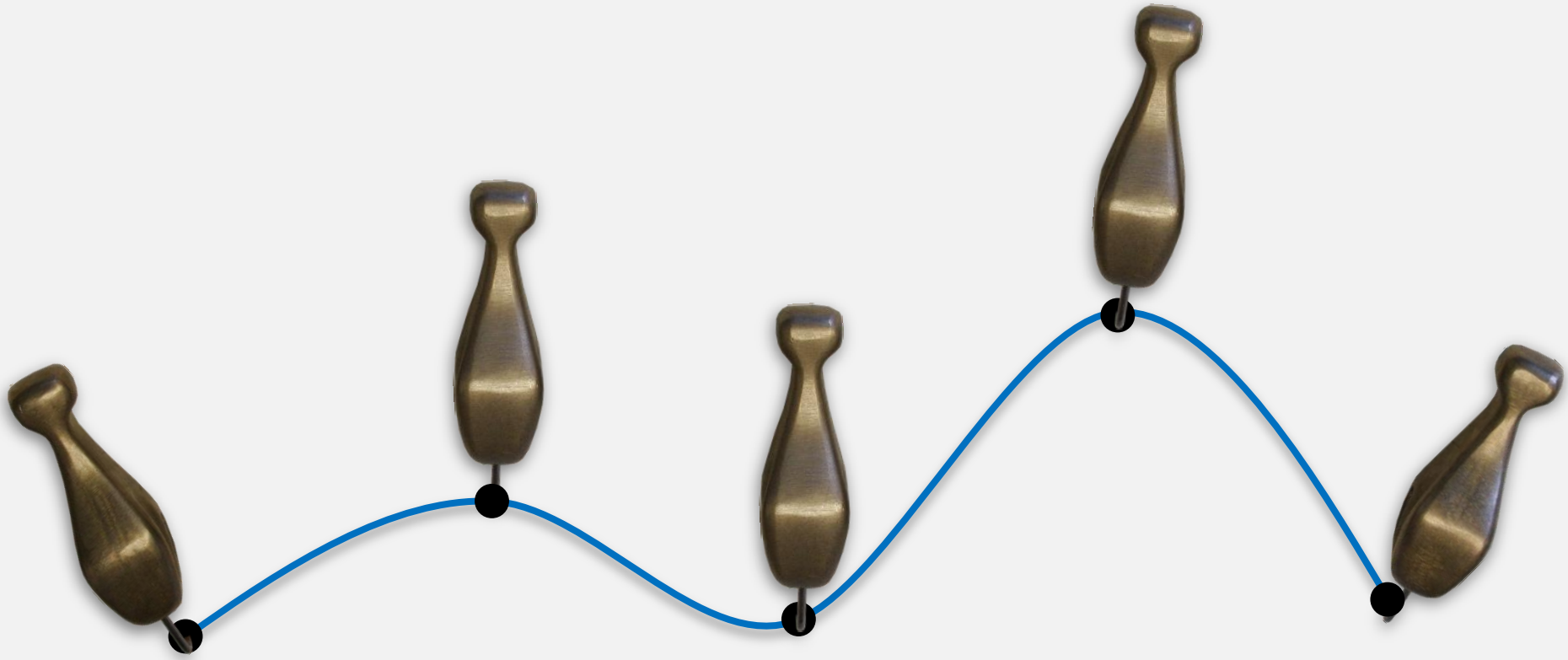

f(u)
xi
u

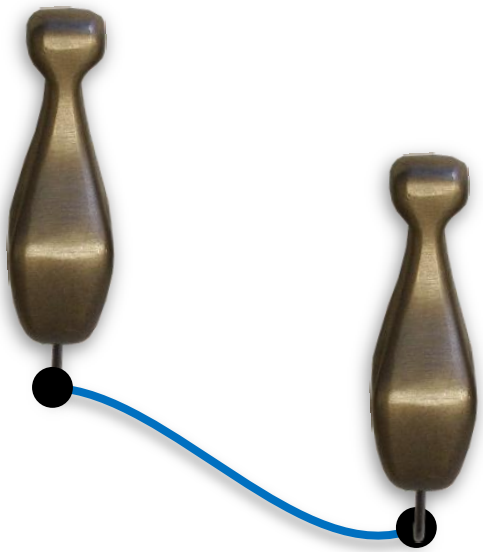
```
float _width = max(1, width);

float _opacity =
    width < 1
    ? opacity * width
    : opacity;

DrawLine(..., _width, _opacity);
```

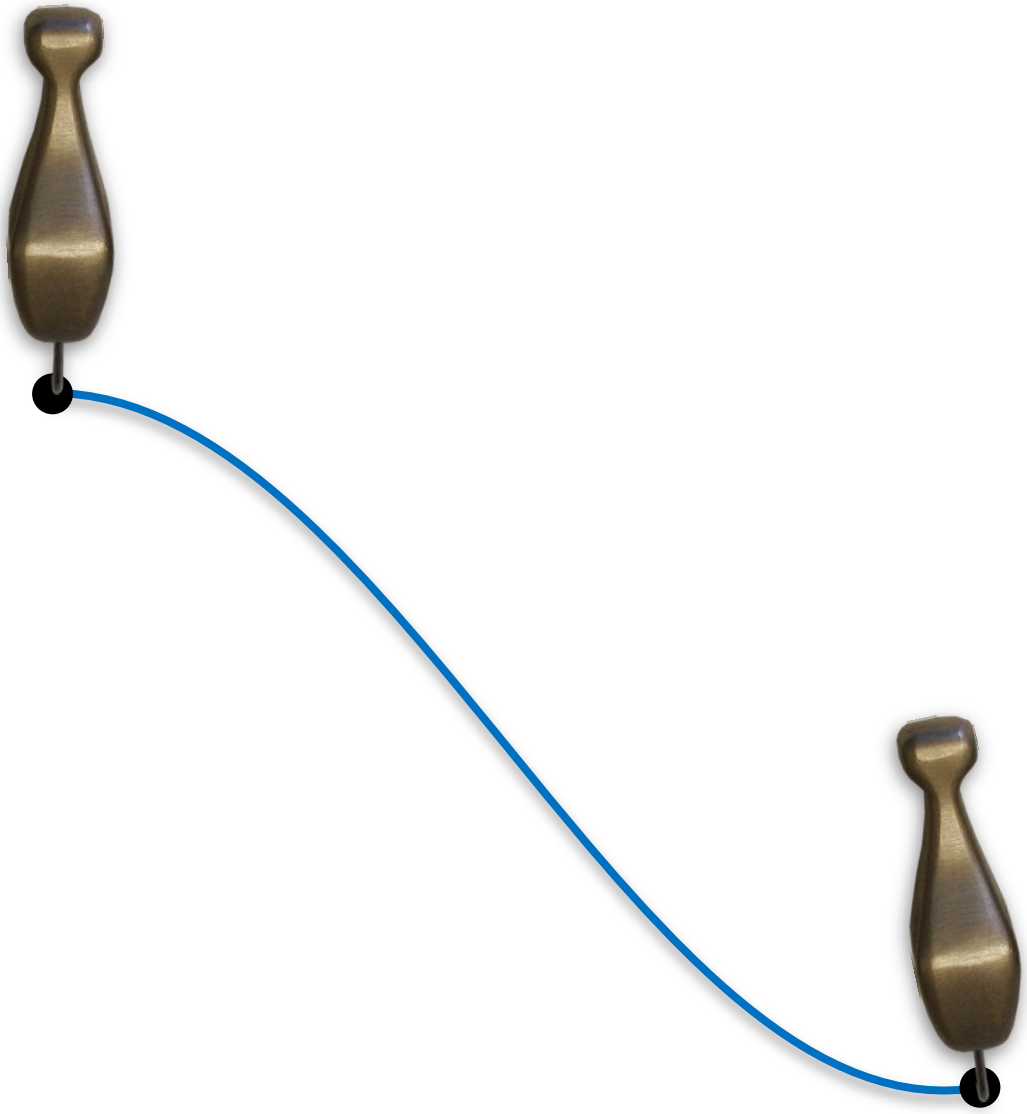

Сплайны

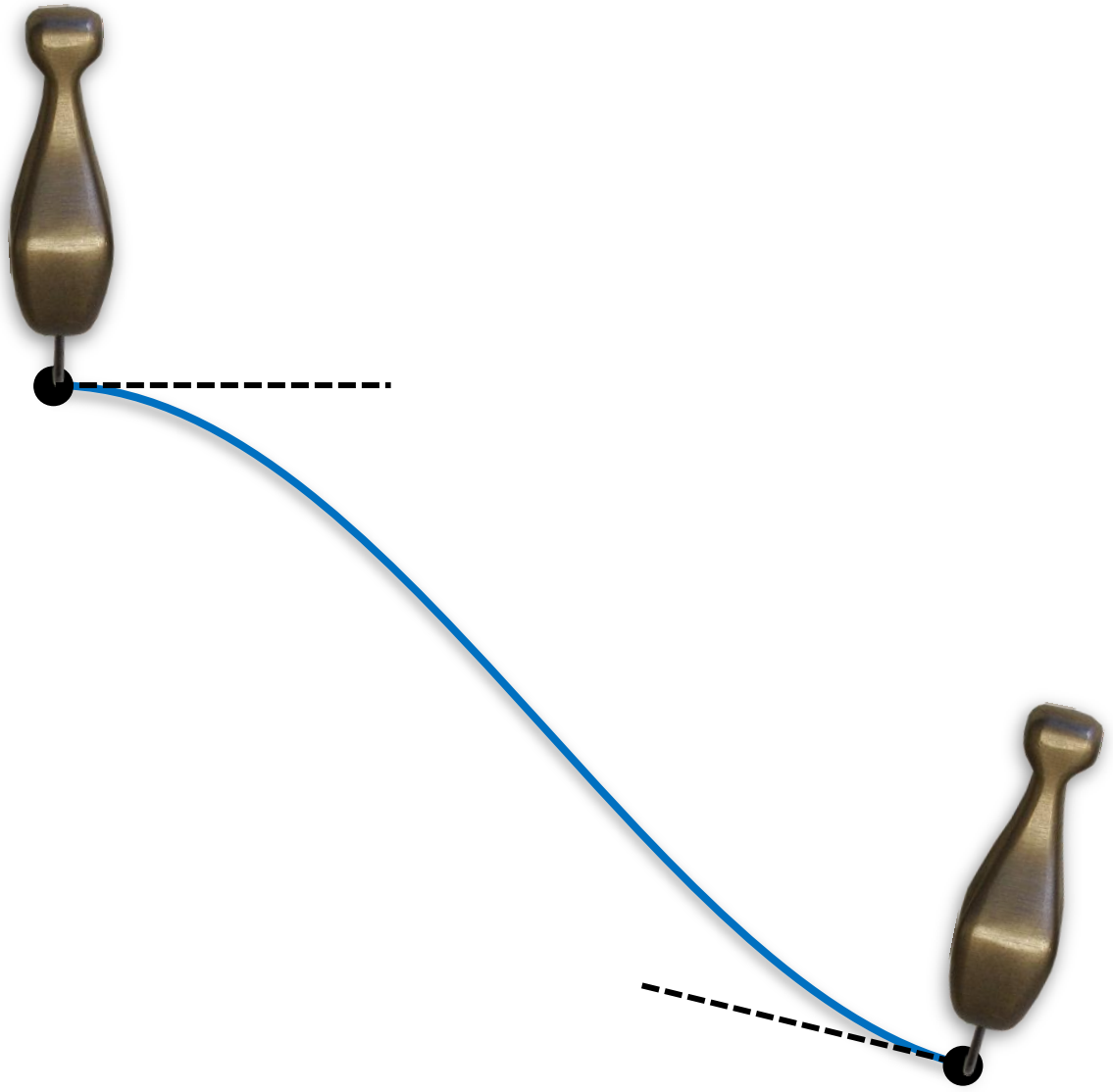

EDSON

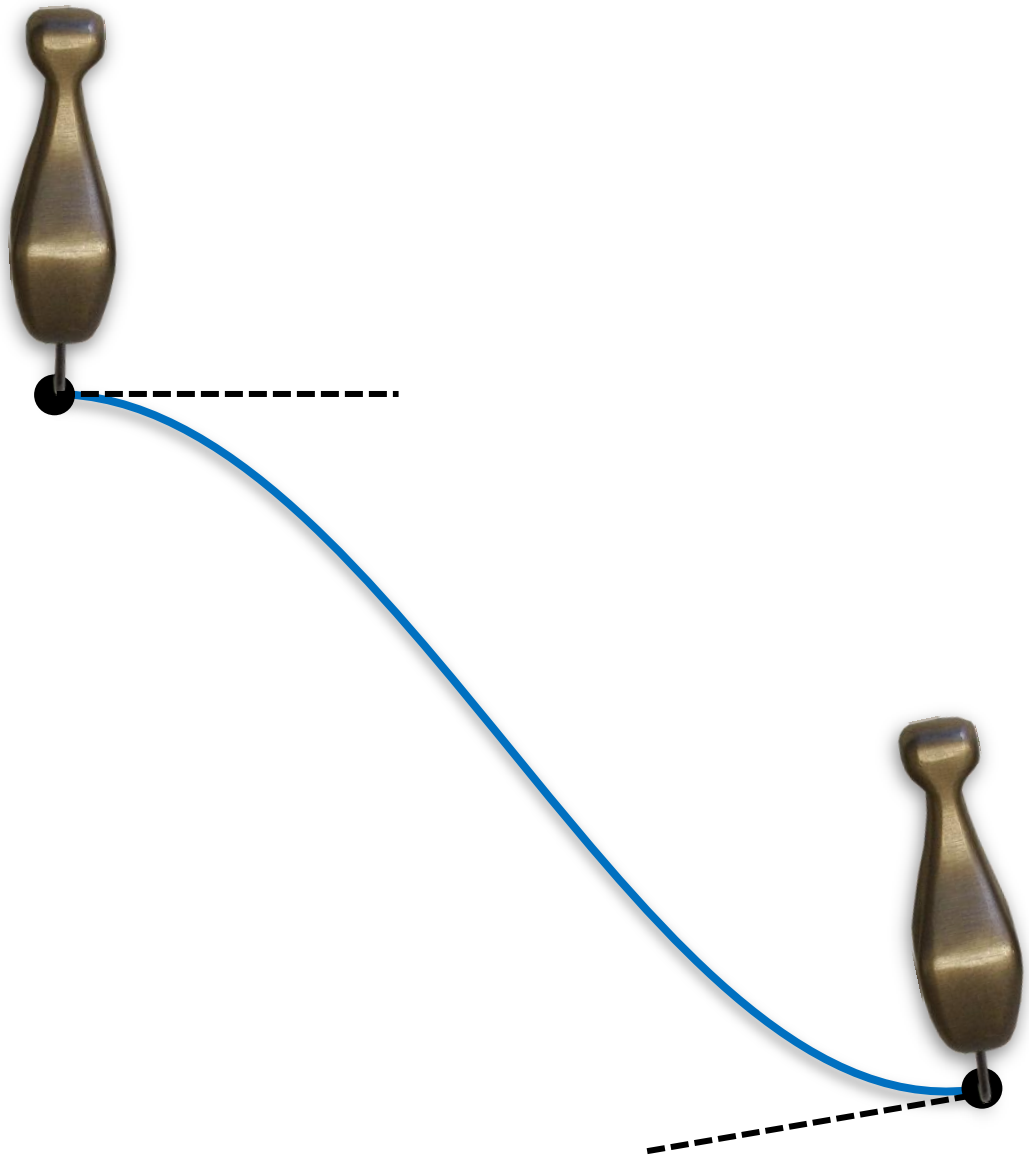


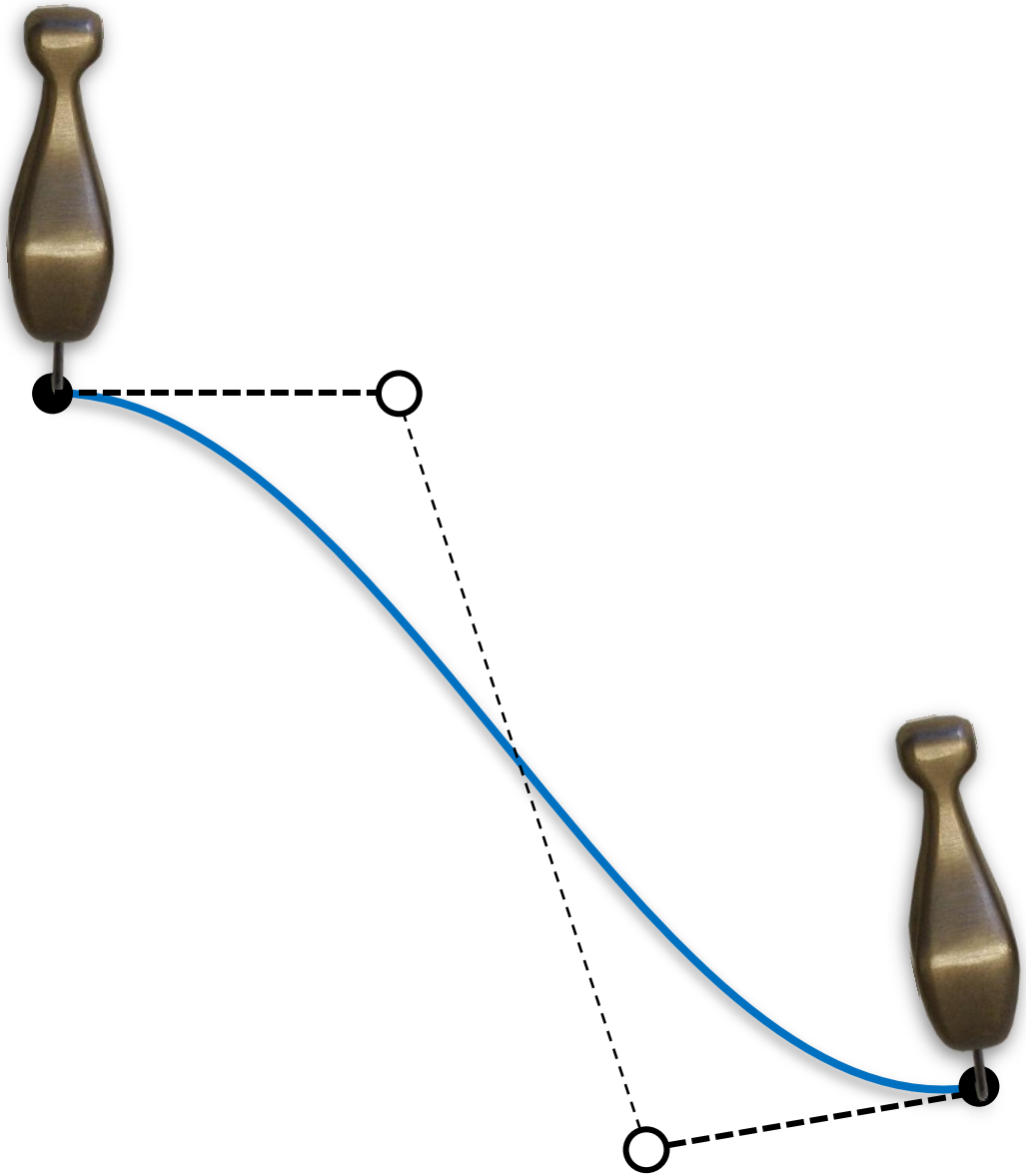


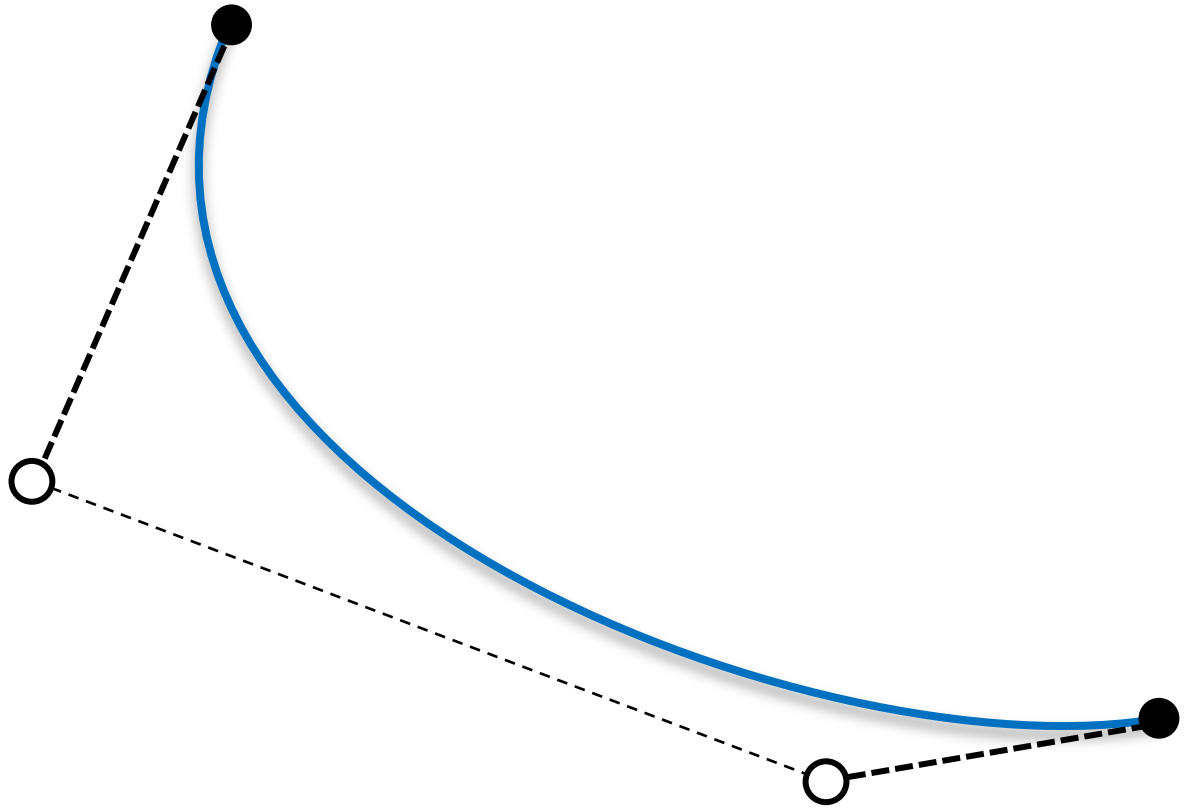


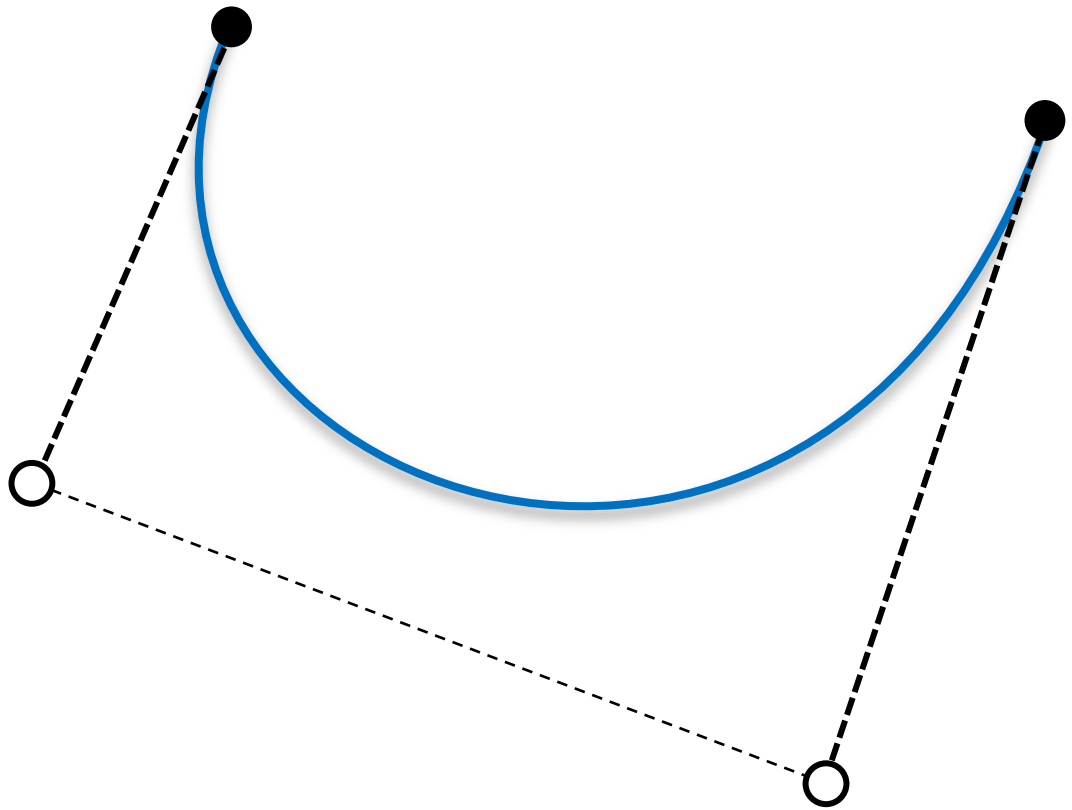




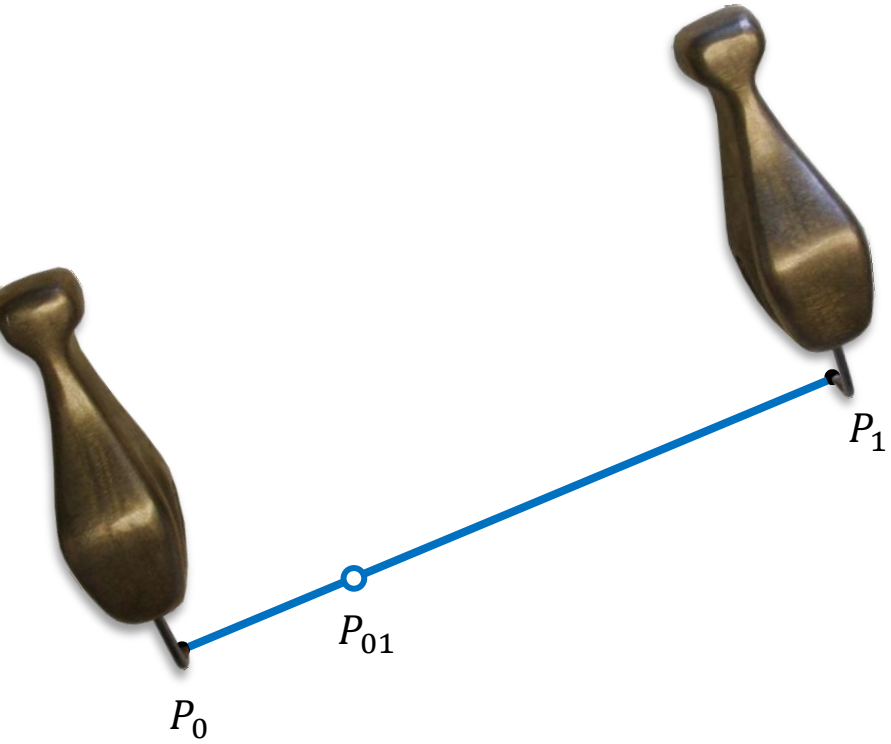








Кривые [1]



$$P_{01} = P_0 + t(P_1 - P_0)$$

$$P_{01} = P_0 + t(P_1 - P_0)$$
$$= P_0 + tP_1 - tP_0$$

$P_{01} = (1 - t)P_0 + tP_1$ ← знаем t, какая P_{01}?

$P_{01} = (P_1 - P_0)t + P_0$ ← знаем P_{01}, какой t?

$$P_{01} = (1 - t)P_0 + tP_1$$

```
// float t

float _1mt = 1 - t;

p01.X = _1mt * p0.X + t * p1.X;
p01.Y = _1mt * p0.Y + t * p1.Y;
```

$$P_{01} = (P_1 - P_0)t + P_0$$

$$P_{01} = (P_1 - P_0)t + P_0$$

$$at + b = 0, \qquad \begin{aligned} a &= P_1 - P_0 \\ b &= P_0 - P_{01} \end{aligned}$$

$$t = -\frac{b}{a}, \qquad a \neq 0$$

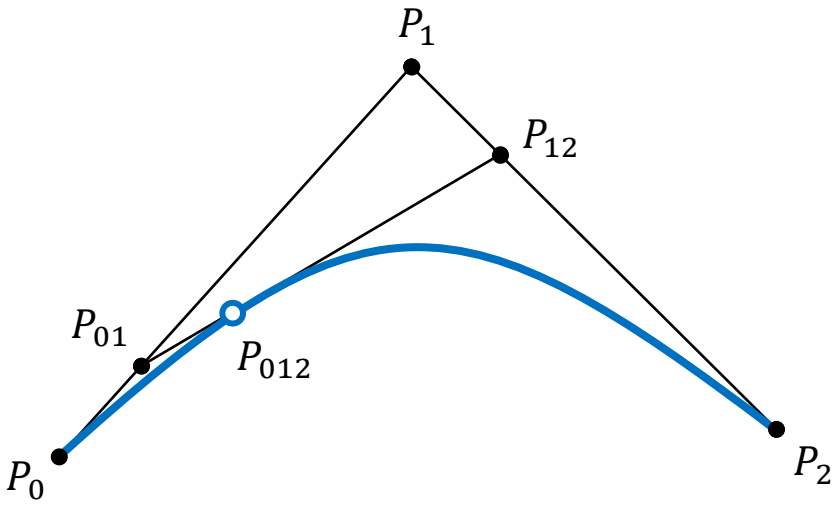
$$P_{01} = (P_1 - P_0)t + P_0$$

$$at + b = 0, \qquad a = P_1 - P_0$$

$$b = P_0 - P_{01}$$

$$t = -\frac{b}{a}, \qquad a \neq 0$$

Кривые 2



$$P_{01} = P_0 + t(P_1 - P_0)$$

$$P_{12} = P_1 + t(P_2 - P_1)$$

$$P_{012} = P_{01} + t(P_{12} - P_{01})$$

$$P_{01} = P_0 + t(P_1 - P_0)$$
$$P_{12} = P_1 + t(P_2 - P_1)$$

$$\begin{aligned} P_{012} &= P_{01} + t(P_{12} - P_{01}) \\ &= P_0 + t(P_1 - P_0) + t(P_1 + t(P_2 - P_1) - P_0 - t(P_1 - P_0)) \\ &= P_0 + tP_1 - tP_0 + tP_1 + t^2P_2 - t^2P_1 - tP_0 - t^2P_1 + t^2P_0 \end{aligned}$$

$$P_{012} = (t-1)^2P_0 - 2t(t-1)P_1 + t^2P_2$$

$$P_{012} = (P_2 - 2P_1 + P_0)t^2 - 2(P_1 - P_0)t + P_0$$

$$P_{012} = (t-1)^2 P_0 - 2t(t-1)P_1 + t^2 P_2$$

```
// float t

float  tm1   = t - 1;
float  t2    = t * t;
float  tm12  = tm1 * tm1;
float _2ttm1 = 2 * t * tm1;

p012.X = tm12 * p0.X - _2ttm1 * p1.X + t2 * p2.X;
p012.Y = tm12 * p0.Y - _2ttm1 * p1.Y + t2 * p2.Y;
```

$$P_{012} = (P_2 - 2P_1 + P_0)t^2 - 2(P_1 - P_0)t + P_0$$

$$P_{012} = (P_2 - 2P_1 + P_0)t^2 - 2(P_1 - P_0)t + P_0$$

$$at^2 + bt + c = 0,$$

$$a = P_2 - 2P_1 + P_0$$
$$b = 2(P_0 - P_1)$$
$$c = P_0 - P_{012}$$

$$t_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}, \qquad a \neq 0$$

$$P_{012} = (P_2 - 2P_1 + P_0)t^2 - 2(P_1 - P_0)t + P_0$$

$$at^2 + bt + c = 0,$$

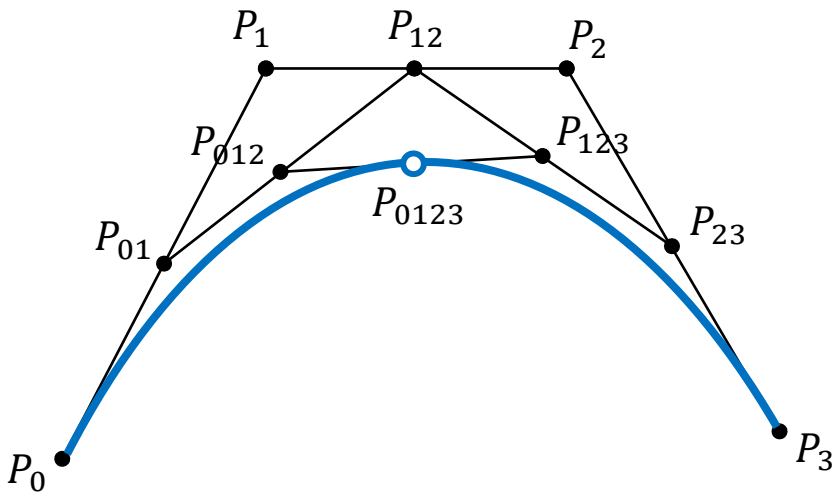
$$a = P_2 - 2P_1 + P_0$$

$$b = 2(P_0 - P_1)$$

$$c = P_0 - P_{012}$$

$$t_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}, \qquad a \neq 0$$

Кривые ^{3}



$$P_{01} = P_0 + t(P_1 - P_0)$$

$$P_{12} = P_1 + t(P_2 - P_1)$$

$$P_{23} = P_2 + t(P_3 - P_2)$$

$$P_{012} = P_{01} + t(P_{12} - P_{01})$$

$$P_{123} = P_{12} + t(P_{23} - P_{12})$$

$$P_{0123} = P_{012} + t(P_{123} - P_{012})$$

$$P_{01} = P_0 + t(P_1 - P_0)$$

$$P_{12} = P_1 + t(P_2 - P_1)$$

$$P_{23} = P_2 + t(P_3 - P_2)$$

$$P_{012} = P_{01} + t(P_{12} - P_{01})$$

$$P_{123} = P_{12} + t(P_{23} - P_{12})$$

$$\begin{aligned}
P_{0123} &= P_{012} + t(P_{123} - P_{012}) \\
&= P_{01} + t(P_{12} - P_{01}) + t(P_{12} + t(P_{23} - P_{12}) - P_{01} - t(P_{12} - P_{01})) \\
&= P_0 + t(P_1 - P_0) + t(P_1 + t(P_2 - P_1) - P_0 - t(P_1 - P_0)) \\
&\quad + t(P_1 + t(P_2 - P_1) + t(P_2 + t(P_3 - P_2) - P_1 - t(P_2 - P_1)) - P_0 - t(P_1 - P_0) \\
&\quad - t(P_1 + t(P_2 - P_1) - P_0 - t(P_1 - P_0))) \\
&= P_0 + tP_1 - tP_0 + tP_1 + t^2P_2 - t^2P_1 - tP_0 - t^2P_1 + t^2P_0 + tP_1 + t^2P_2 - t^2P_1 + t^2P_2 \\
&\quad + t^3P_3 - t^3P_2 - t^2P_1 - t^3P_2 + t^3P_1 - tP_0 - t^2P_1 + t^2P_0 - t^2P_1 - t^3P_2 + t^3P_1 \\
&\quad + t^2P_0 + t^3P_1 - t^3P_0
\end{aligned}$$

$$P_{0123} = (1-t)^3P_0 + 3t(1-t)^2P_1 + 3t^2(1-t)P_2 + t^3P_3$$

$$P_{0123} = (P_3 - 3P_2 + 3P_1 - P_0)t^3 + 3(P_2 - 2P_1 + P_0)t^2 + 3(P_1 - P_0)t + P_0$$

$$P_{0123} = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t) P_2 + t^3 P_3$$

```
// float t

float  t2   = t * 2;
float  t3   = t2 * t;
float _1mt  = 1 - t;
float _1mt2 = _1mt * _1mt;
float _1mt3 = _1mt2 * _1mt;
float _3t_1mt2 = 3 * t * _1mt2;
float _3t2_1mt = 3 * t2 * _1mt;

p0123.X = _1mt3 * p0.X - _3t_1mt2 * p1.X + _3t2_1mt * p2.X + t3 * p3.X;
p0123.Y = _1mt3 * p0.Y - _3t_1mt2 * p1.Y + _3t2_1mt * p2.Y + t3 * p3.Y;
```

$$P_{0123} = (P_3 - 3P_2 + 3P_1 - P_0)t^3 + 3(P_2 - 2P_1 + P_0)t^2 + 3(P_1 - P_0)t + P_0$$

$$at^3 + bt^2 + ct + d = 0,$$

$$a = P_3 - 3P_2 + 3P_1 + P_0$$
$$b = 3(P_2 - 2P_1 + P_0)$$
$$c = 3(P_0 - P_1)$$
$$d = P_0 - P_{0123}$$

$$a \neq 0$$

$$P_{0123} = (P_3 - 3P_2 + 3P_1 - P_0)t^3 + 3(P_2 - 2P_1 + P_0)t^2 + 3(P_1 - P_0)t + P_0$$

$$at^3 + bt^2 + ct + d = 0,$$

$$a = P_3 - 3P_2 + 3P_1 + P_0$$
$$b = 3(P_2 - 2P_1 + P_0)$$
$$c = 3(P_0 - P_1)$$
$$d = P_0 - P_{0123}$$

$$a \neq 0$$

$$b' = \frac{b}{a} \quad c' = \frac{c}{a} \quad d' = \frac{d}{a} \quad , \qquad a \neq 0$$

$$q = \frac{b'^2 - 3c'}{9} \qquad r = \frac{2b'^3 - 9b'c' + 27d'}{54}$$

$$\theta = \cos^{-1}\left(\frac{r}{\sqrt{q^3}}\right)$$

$$t_1 = -2\sqrt{q}\cos\left(\frac{\theta}{3}\right) - \frac{b'}{3}$$

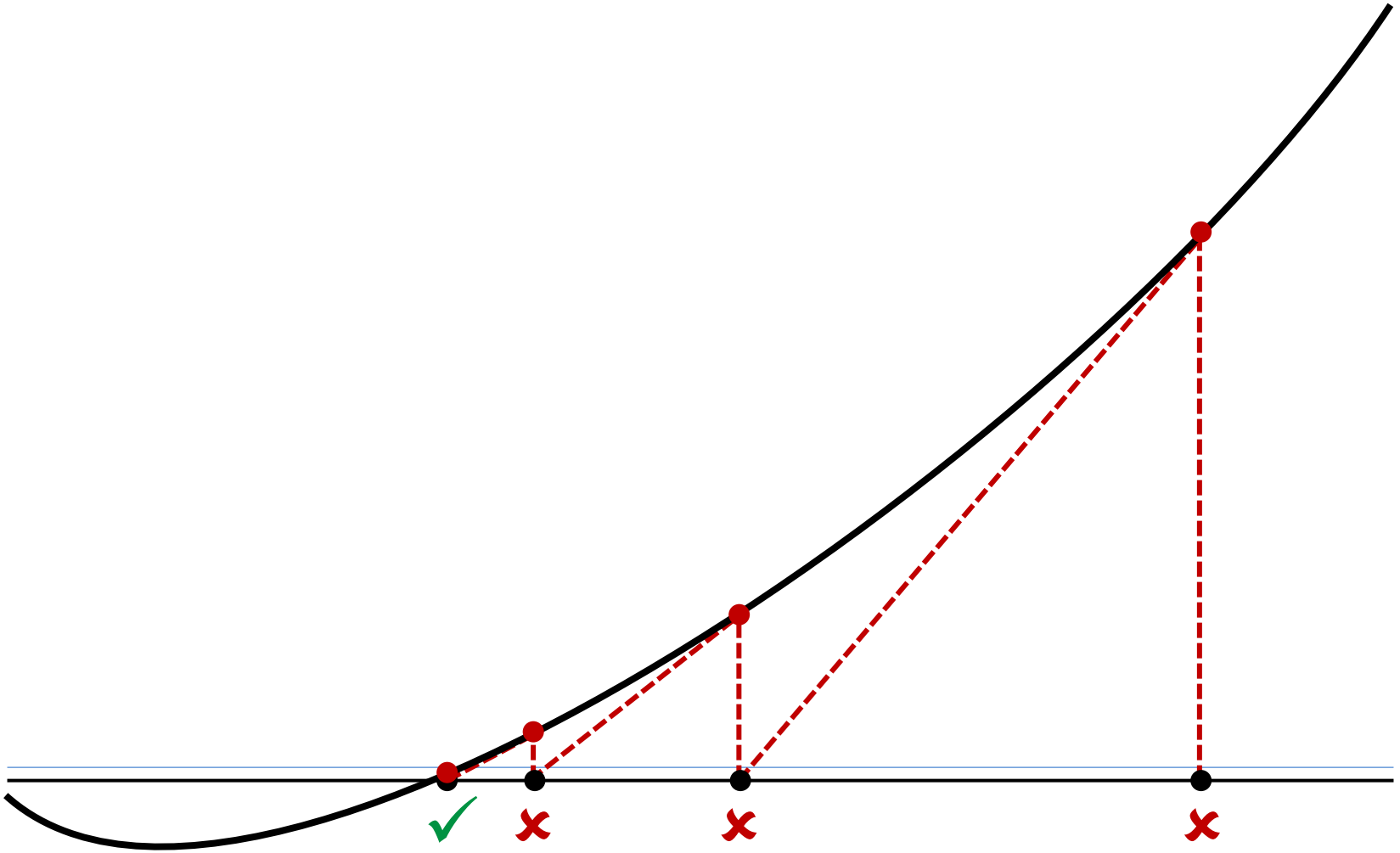
$$t_{2,3} = -2\sqrt{q}\cos\left(\frac{\theta \pm 2\pi}{3}\right) - \frac{b'}{3} \qquad , \qquad r^2 < q^3$$

$$s = -\frac{r}{|r|}\left|\sqrt[3]{|r| + \sqrt{r^2 - q^3}}\right| \qquad u = \begin{cases} 0 & \text{если } s = 0, \\ \dfrac{q}{s} & \text{если } s \neq 0. \end{cases}$$

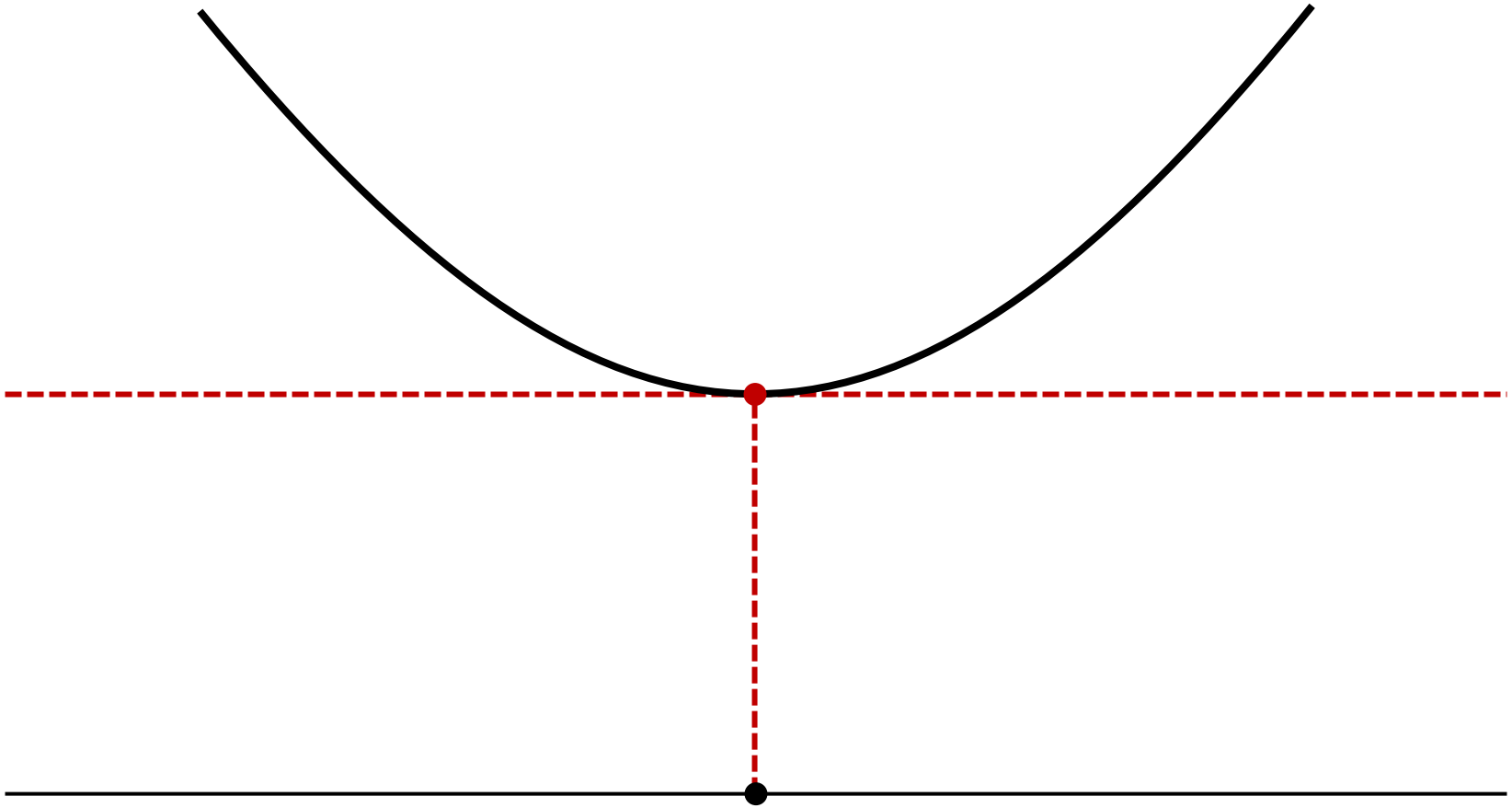
$$t_1 = s + u - \frac{b'}{3}$$

$$t_2 = -\frac{s+u}{2} - \frac{b'}{3}, \qquad s = u$$

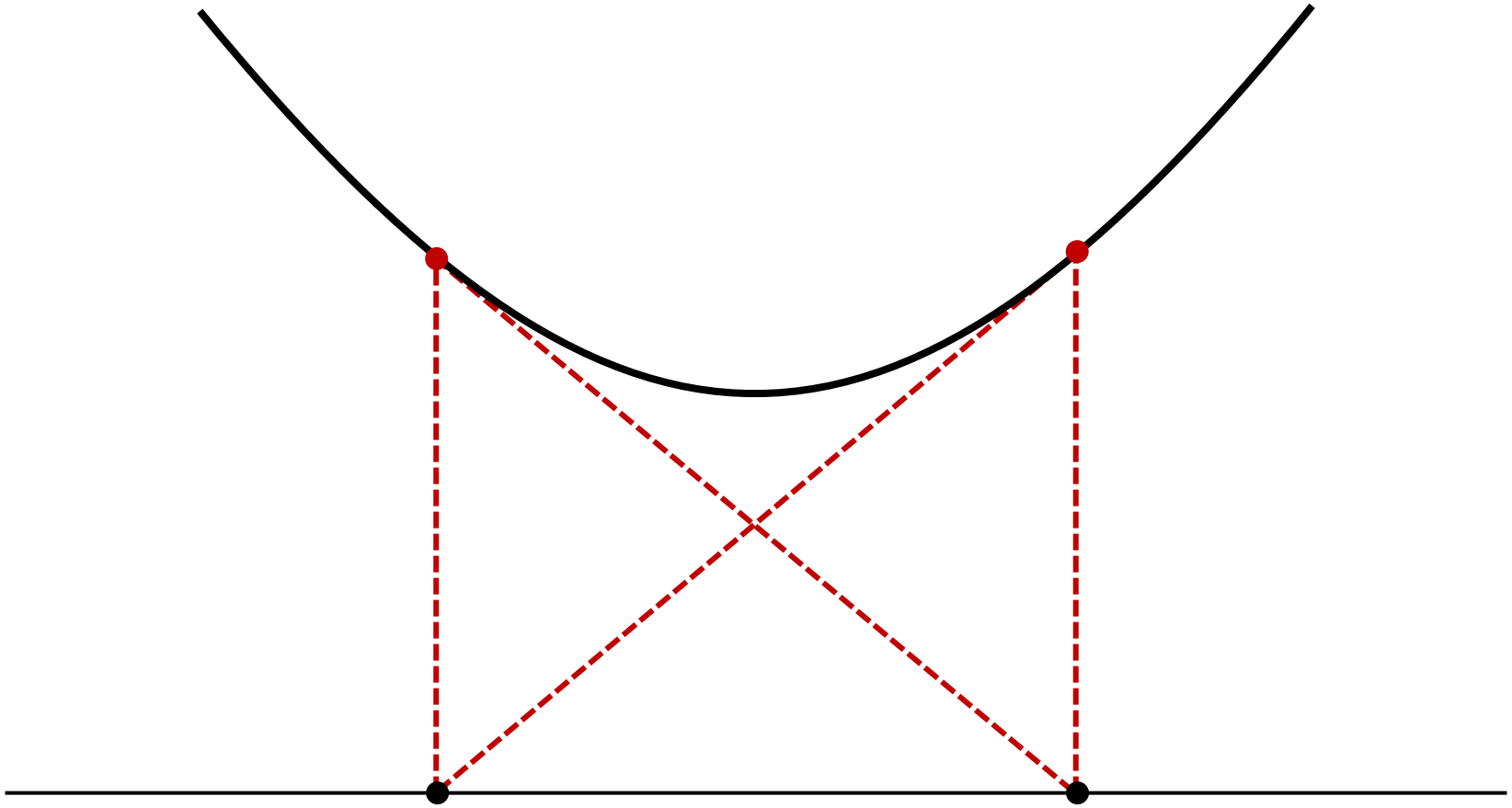
Метод Ньютона



Метод Ньютона

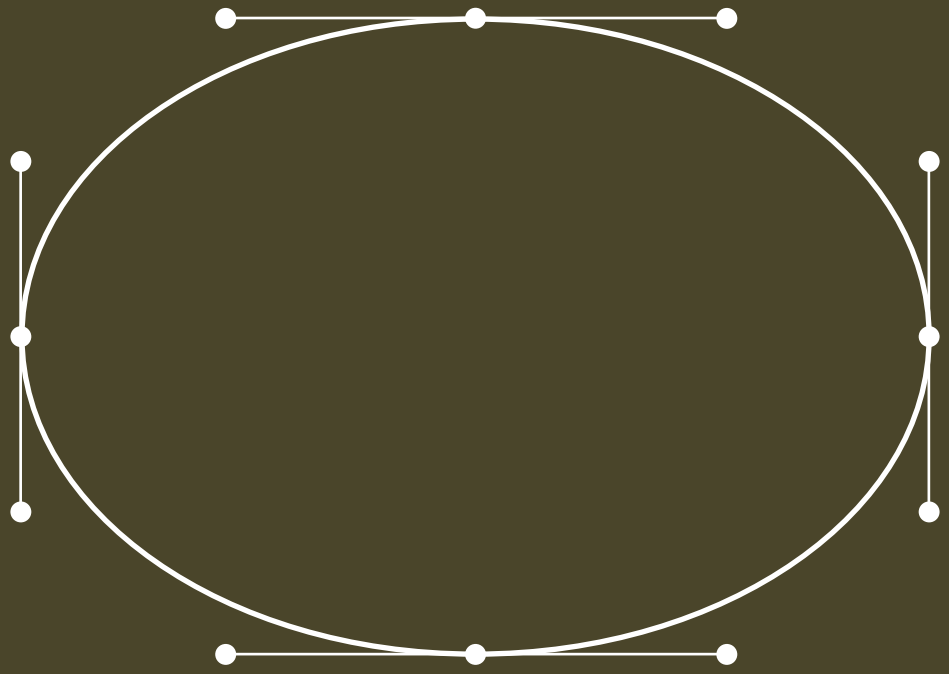


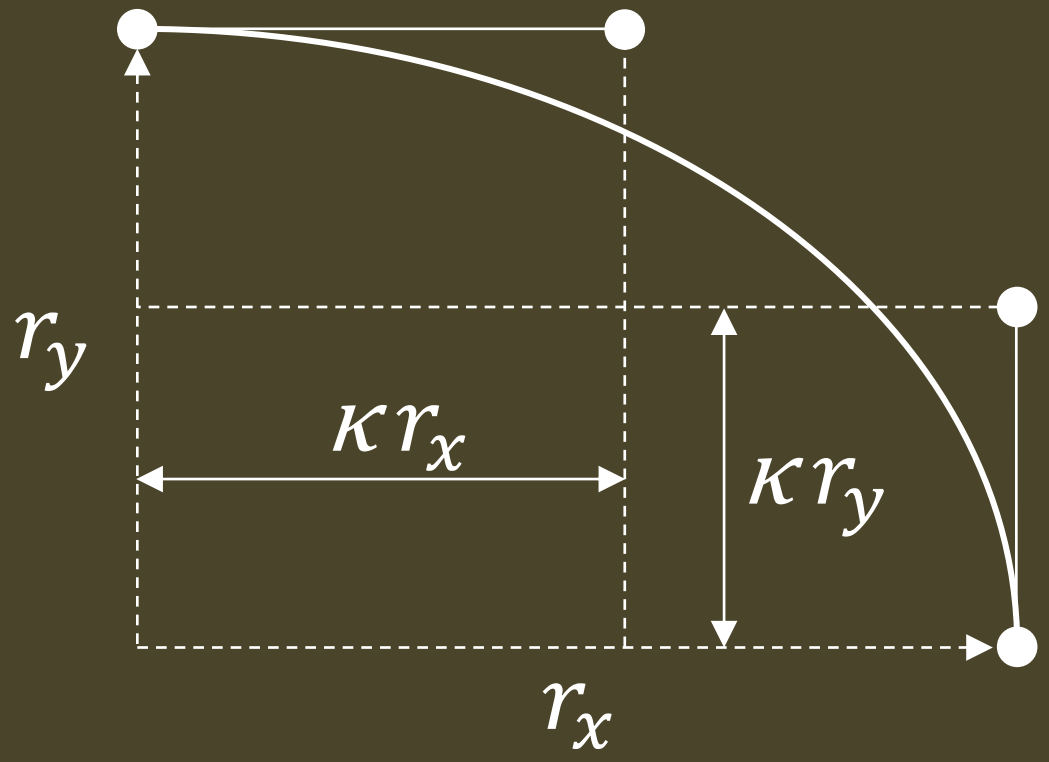
Метод Ньютона



Геометрические формы

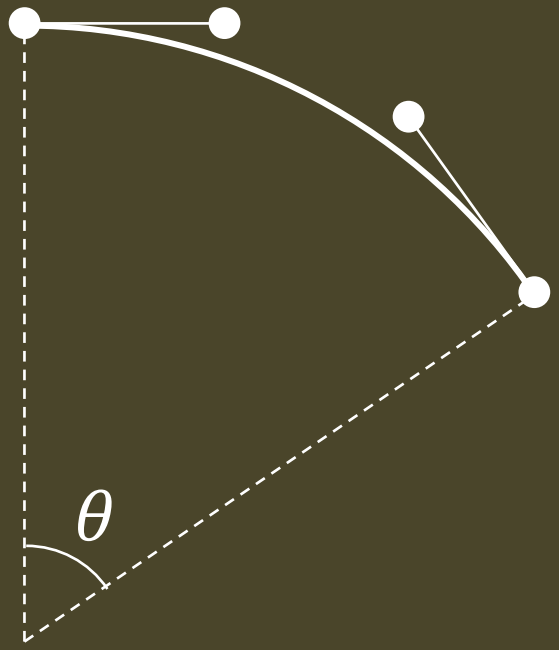
Круги/эллипсы



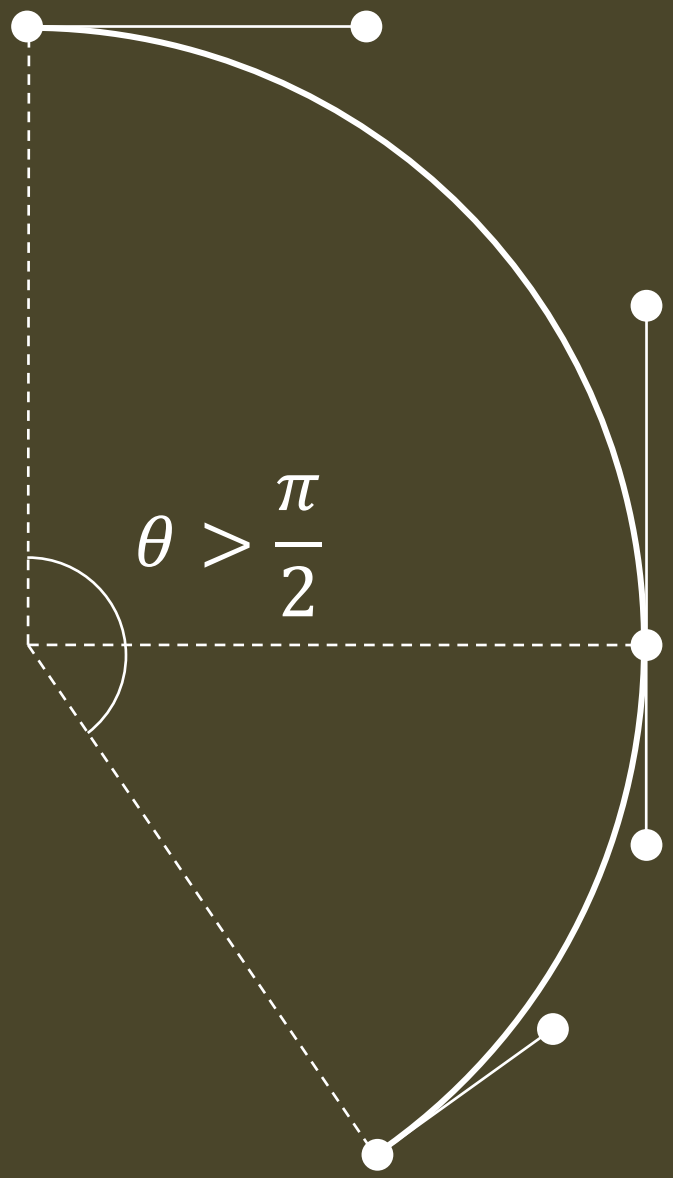


$$\kappa = \frac{4}{3}\left(\sqrt{2} - 1\right) \approx 0.5523$$

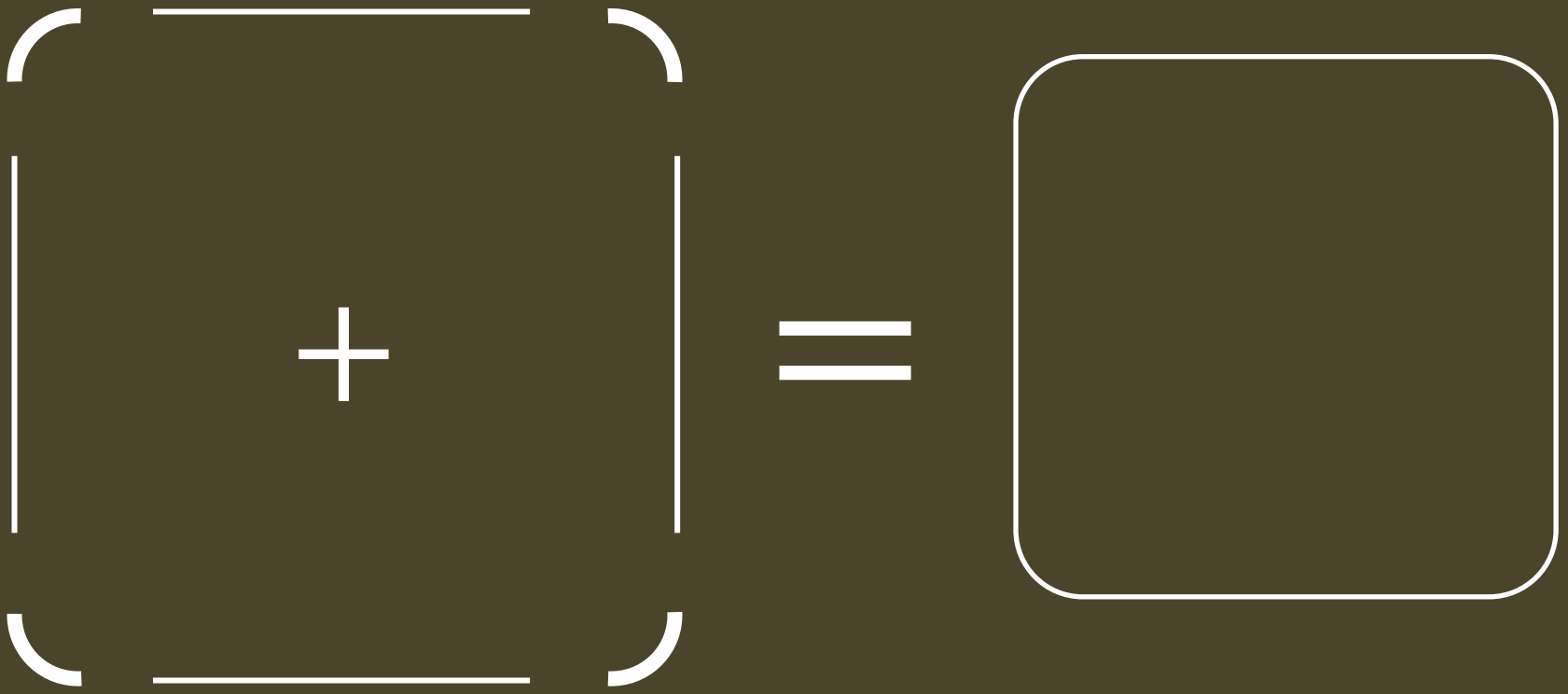
Арки



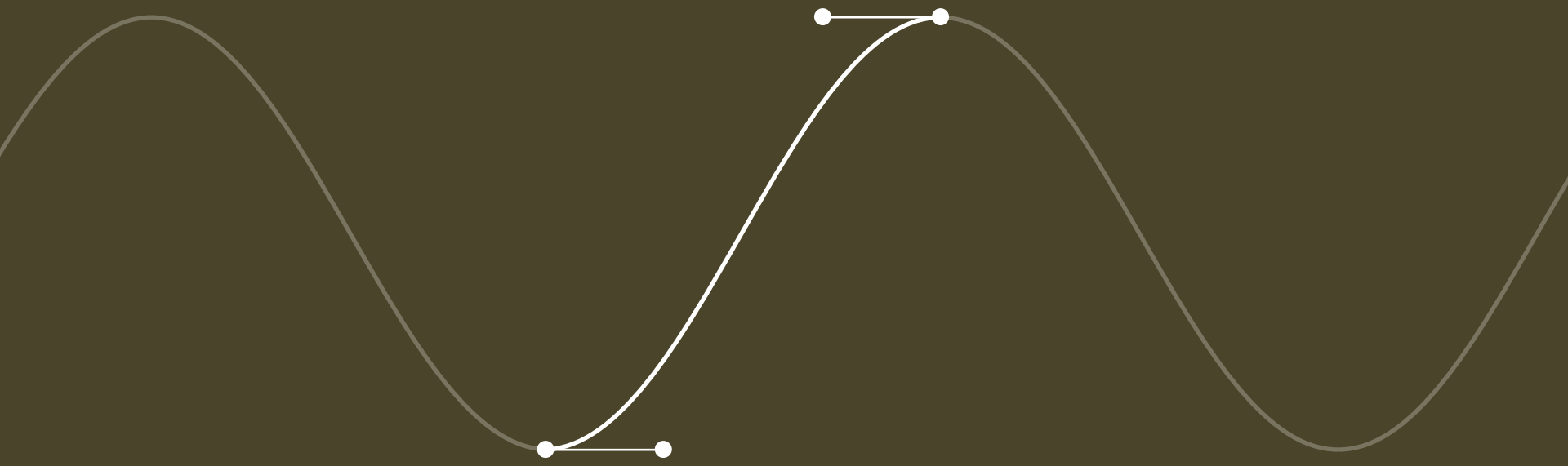
$$\kappa = \frac{4}{3}\tan\frac{\theta}{4}$$


$\theta > \frac{\pi}{2}$

Закруглённые углы



Синусоиды

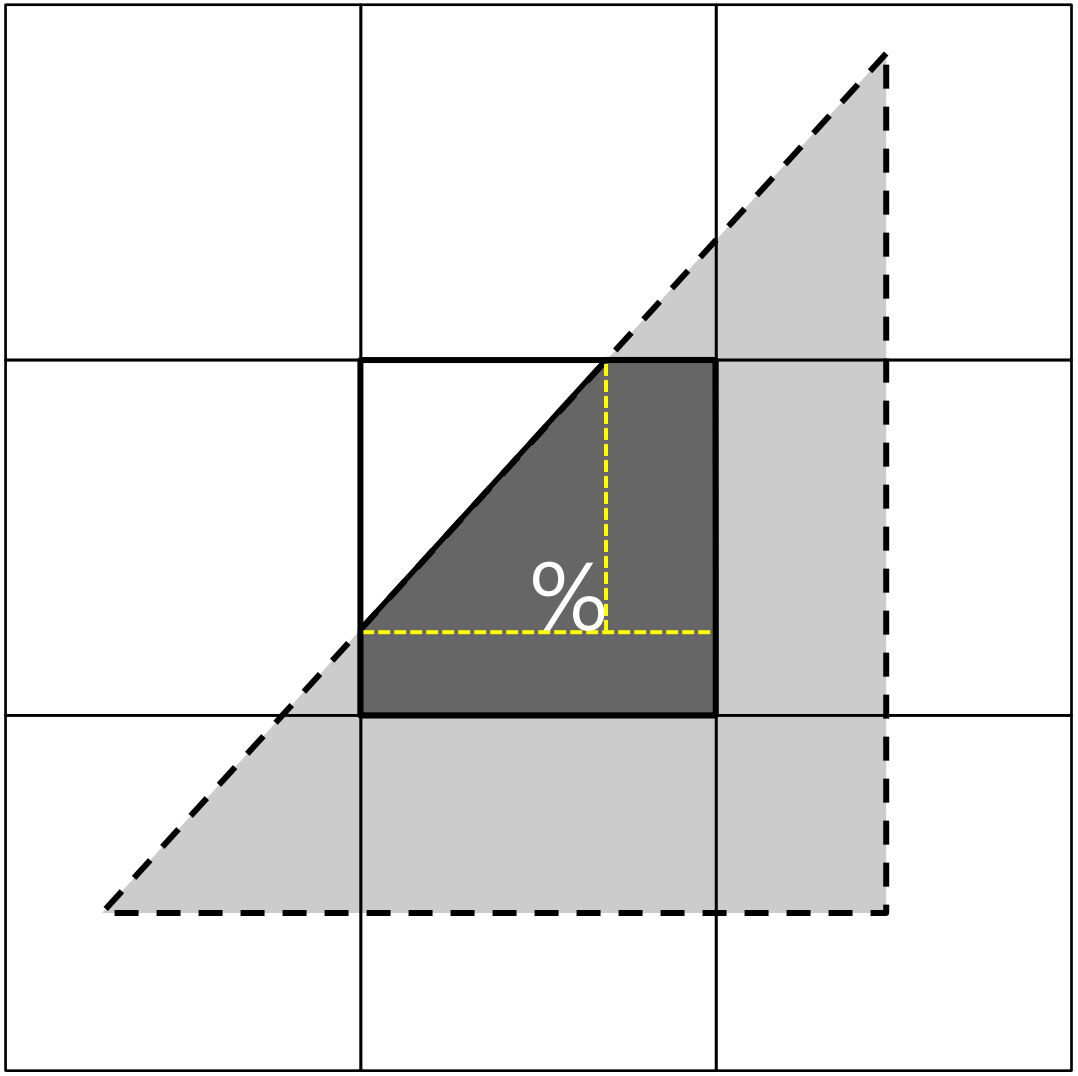


$$\kappa \approx \frac{1}{2.74} \approx 0.36496$$

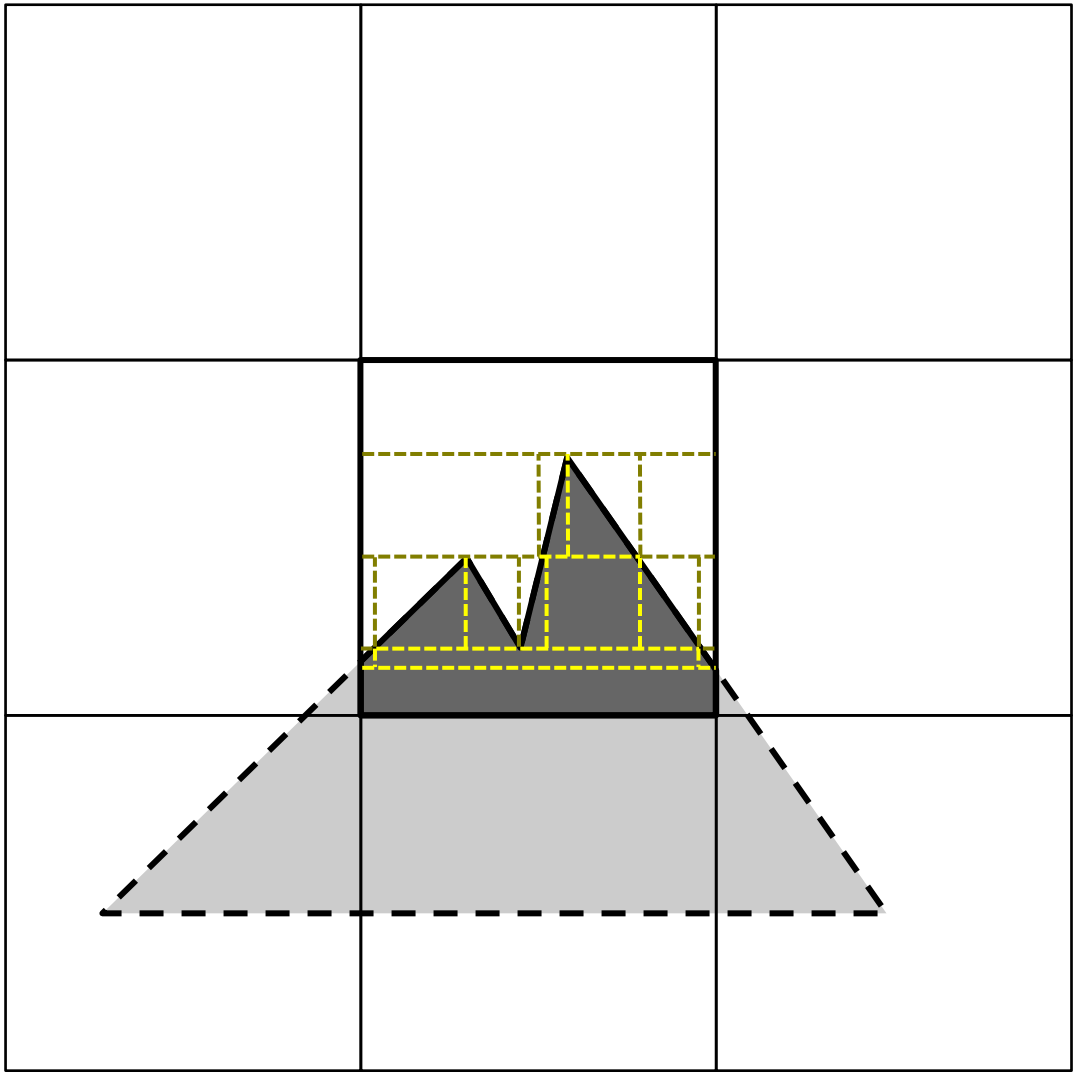
Сглаживание

(antialiasing)

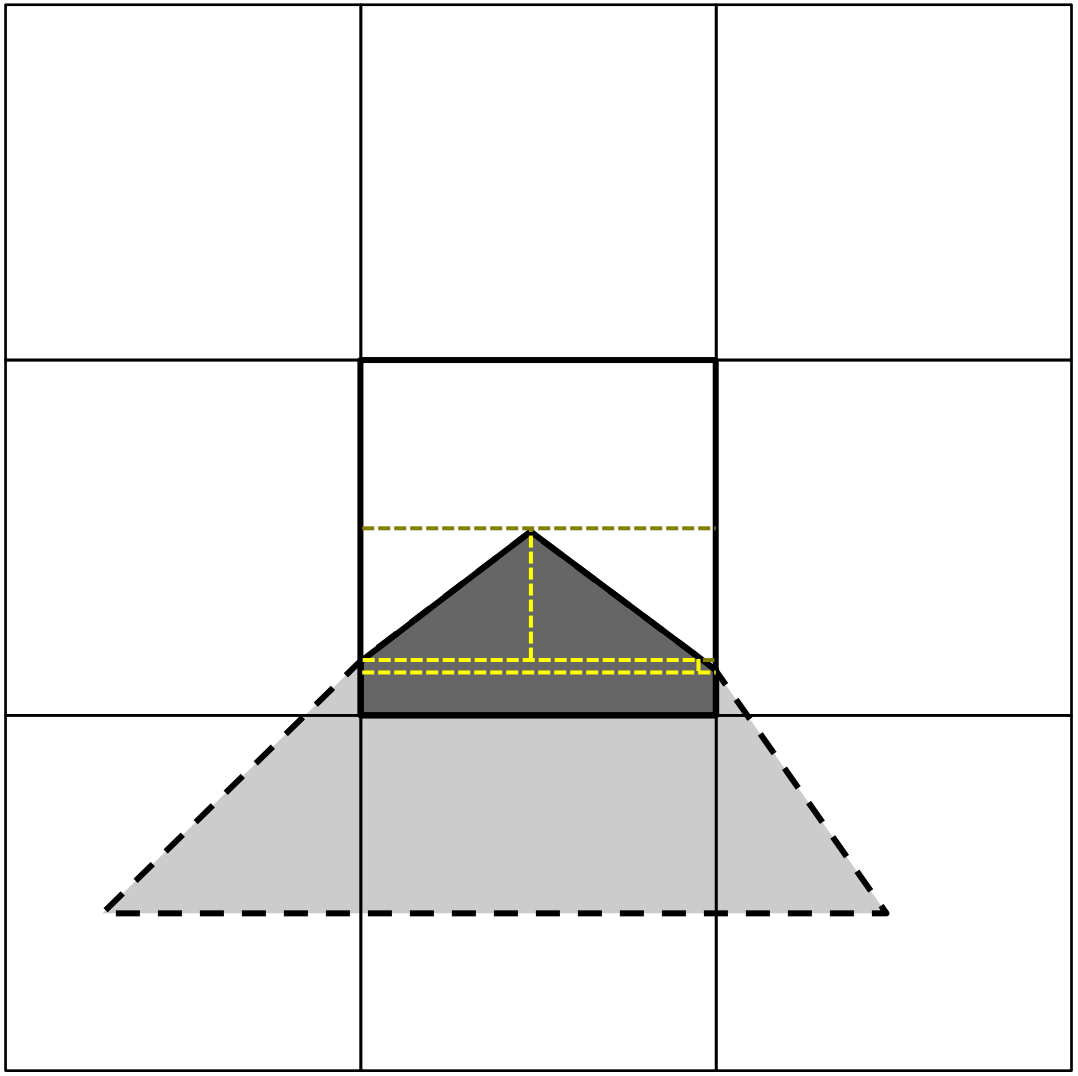
Аналитическое сглаживание

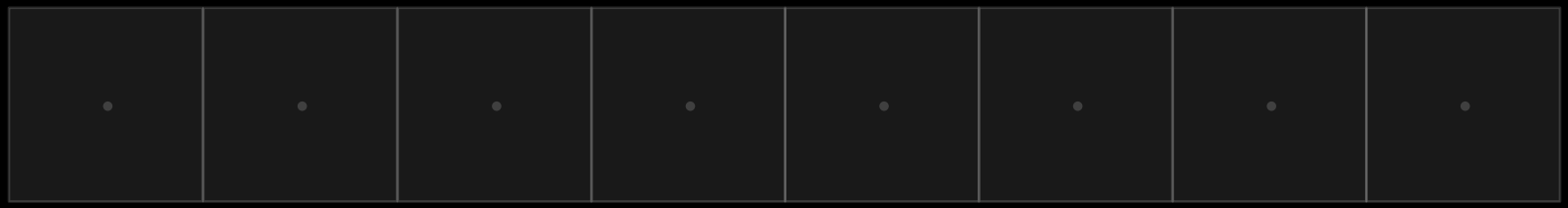


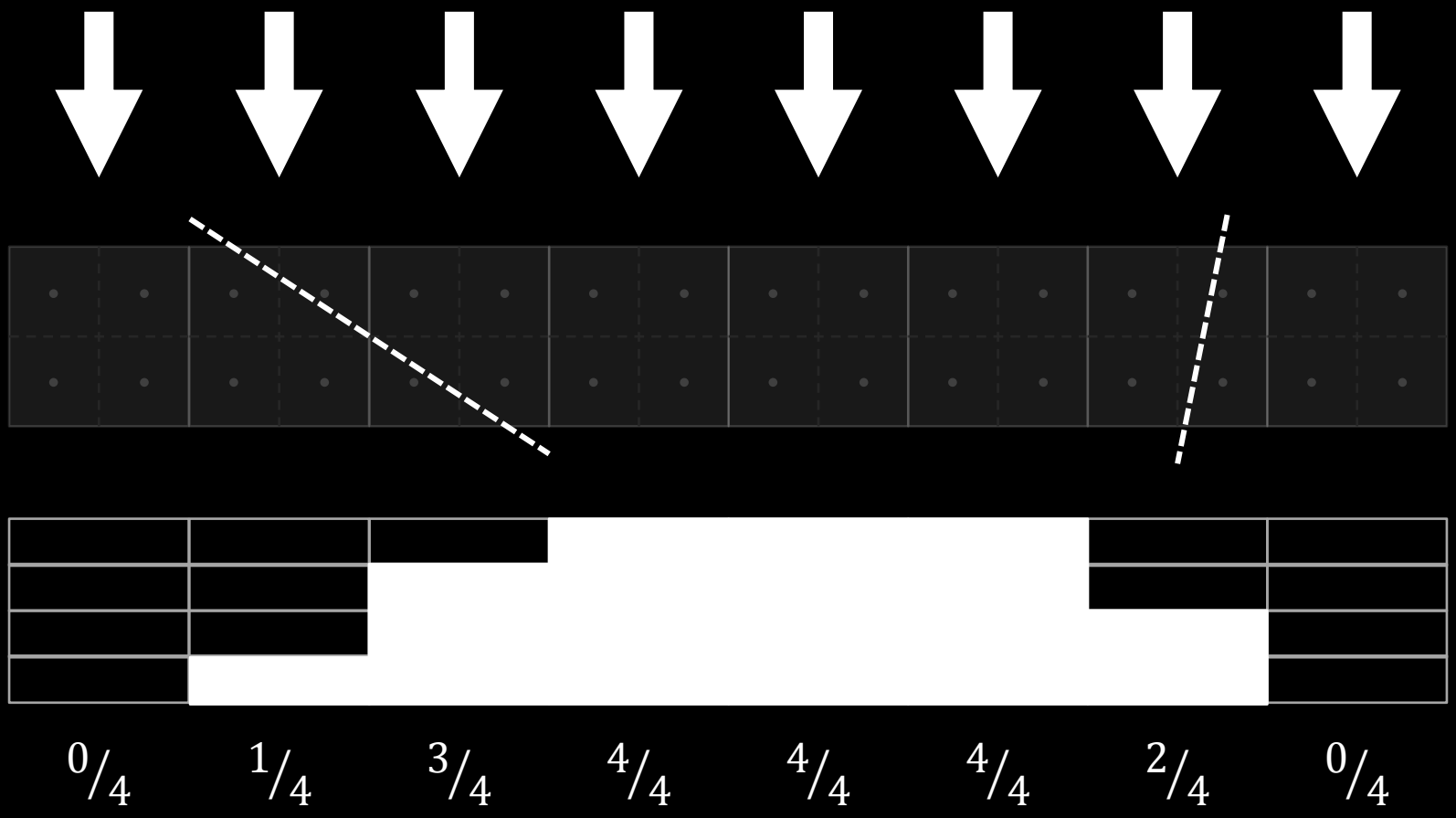
Аналитическое сглаживание

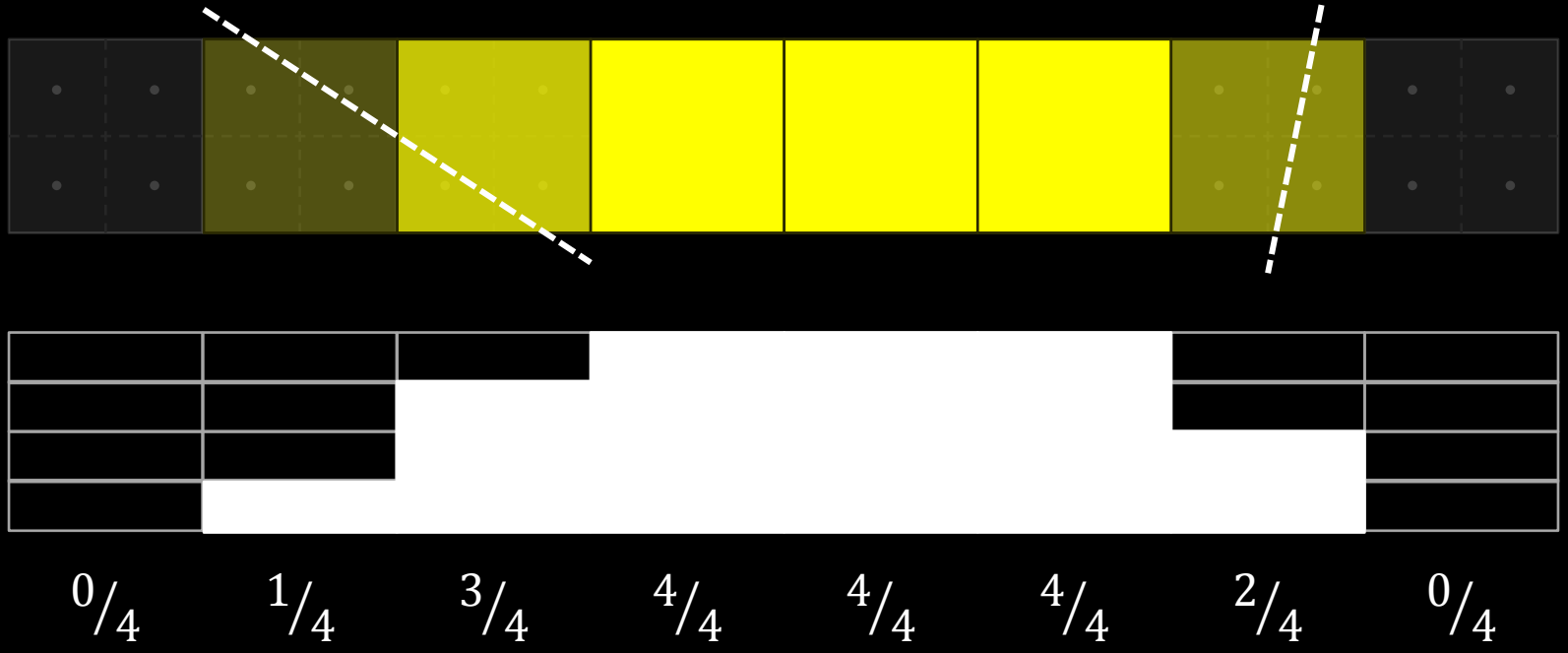


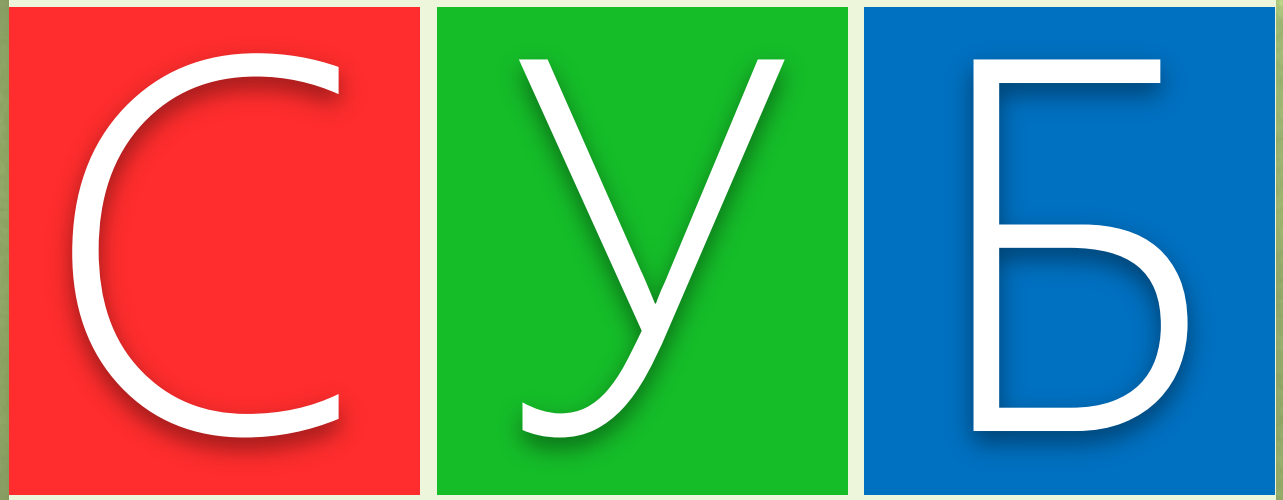
Аналитическое сглаживание



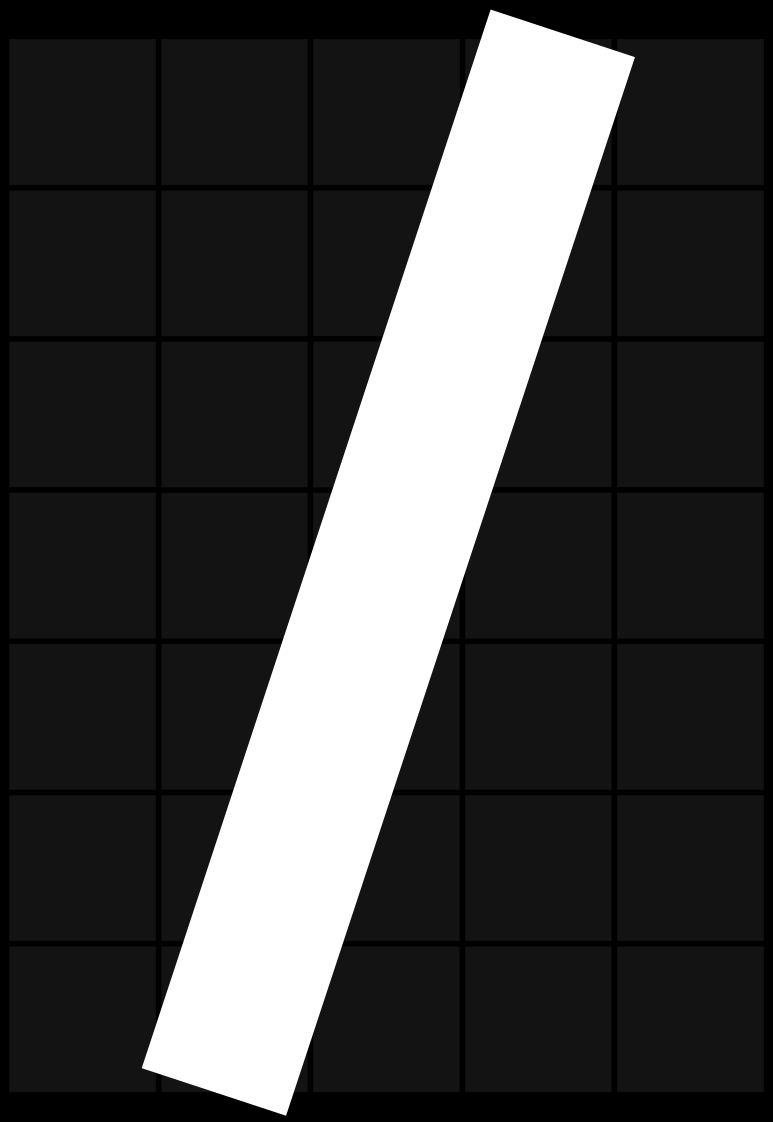


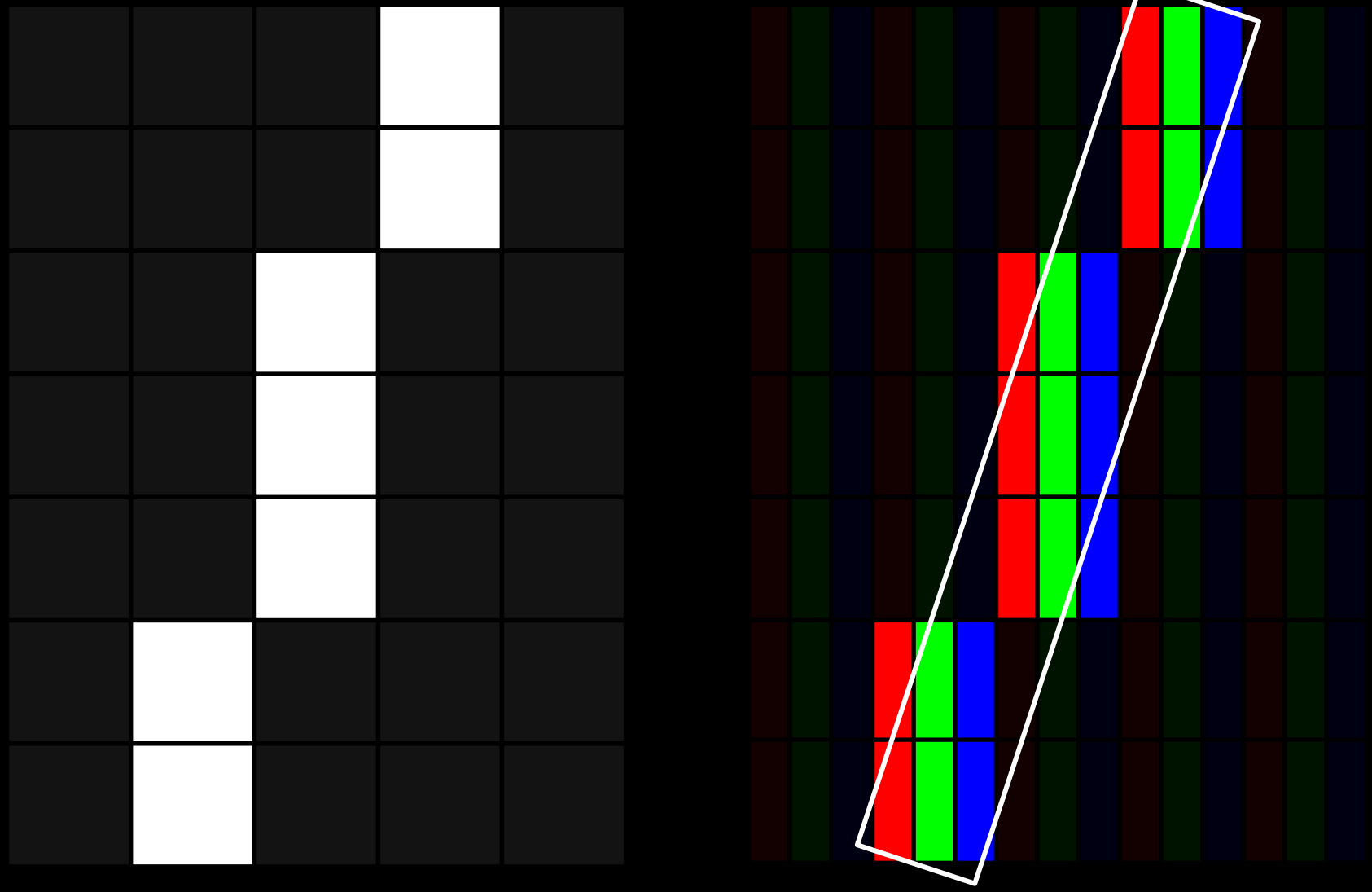

0/4
1/4
3/4
4/4
4/4
4/4
2/4
0/4

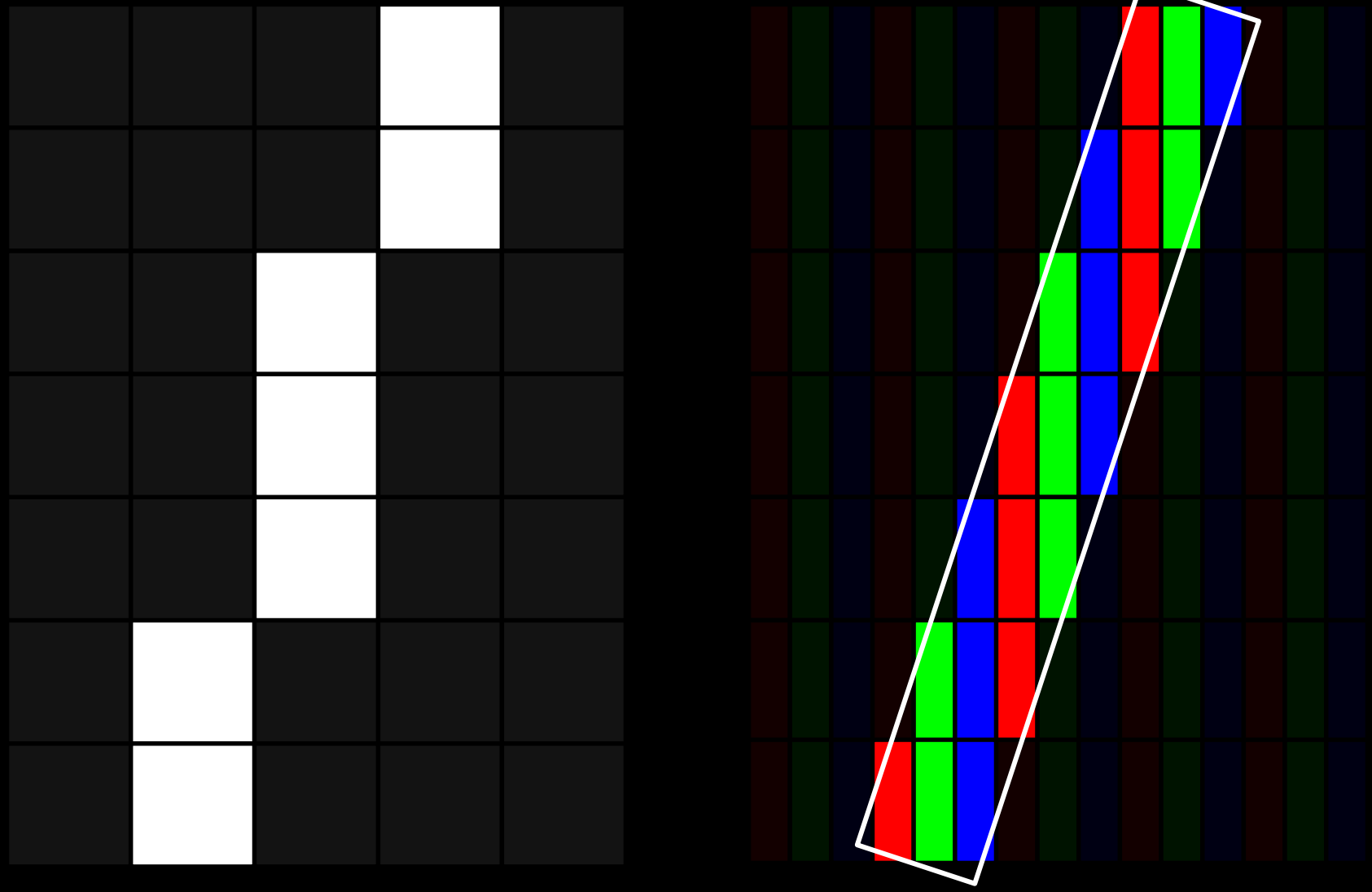

0/4
1/4
3/4
4/4
4/4
4/4
2/4
0/4

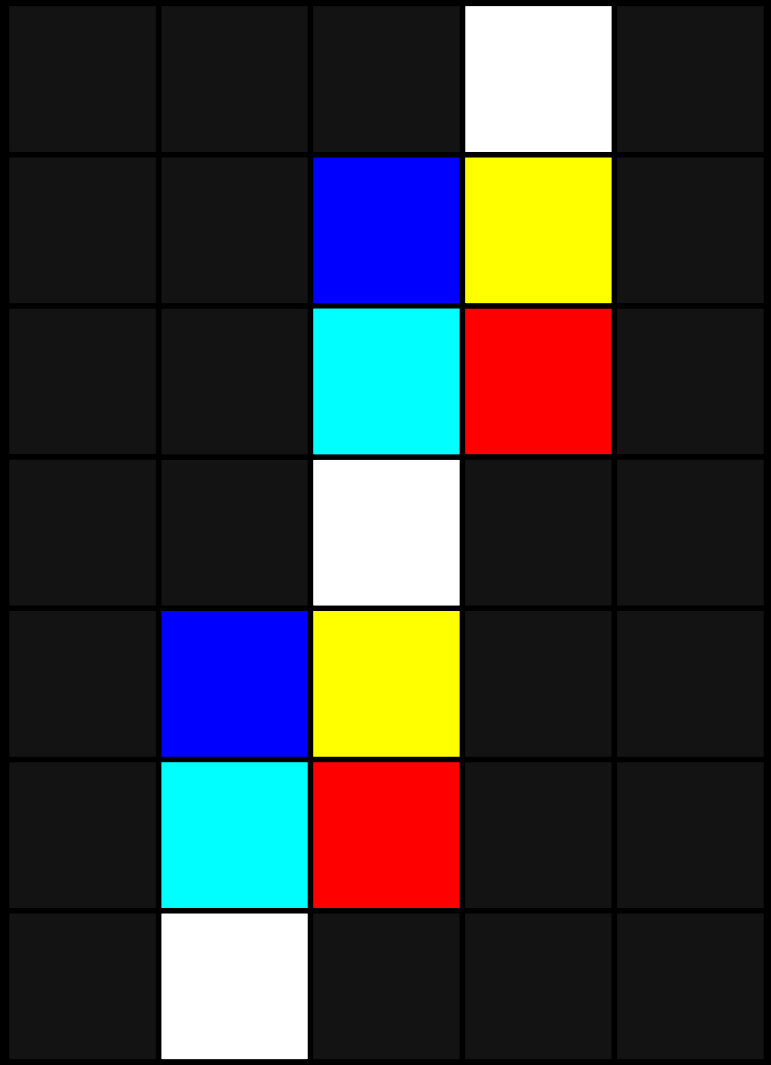


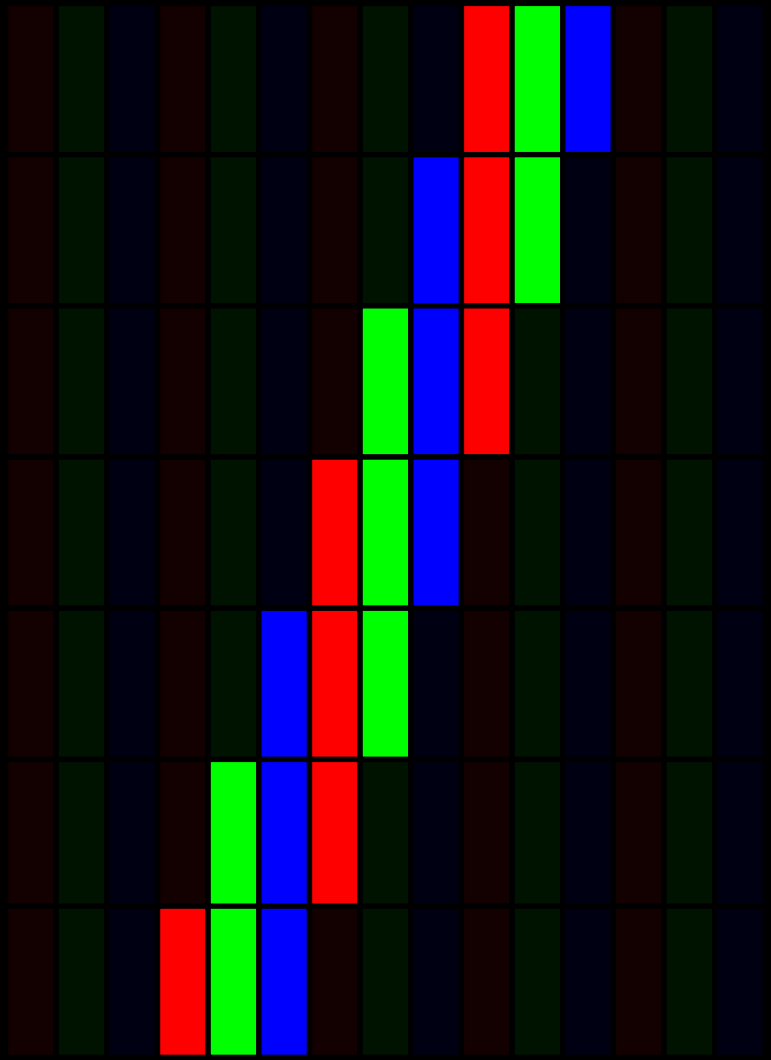
ПИКСЕЛИ

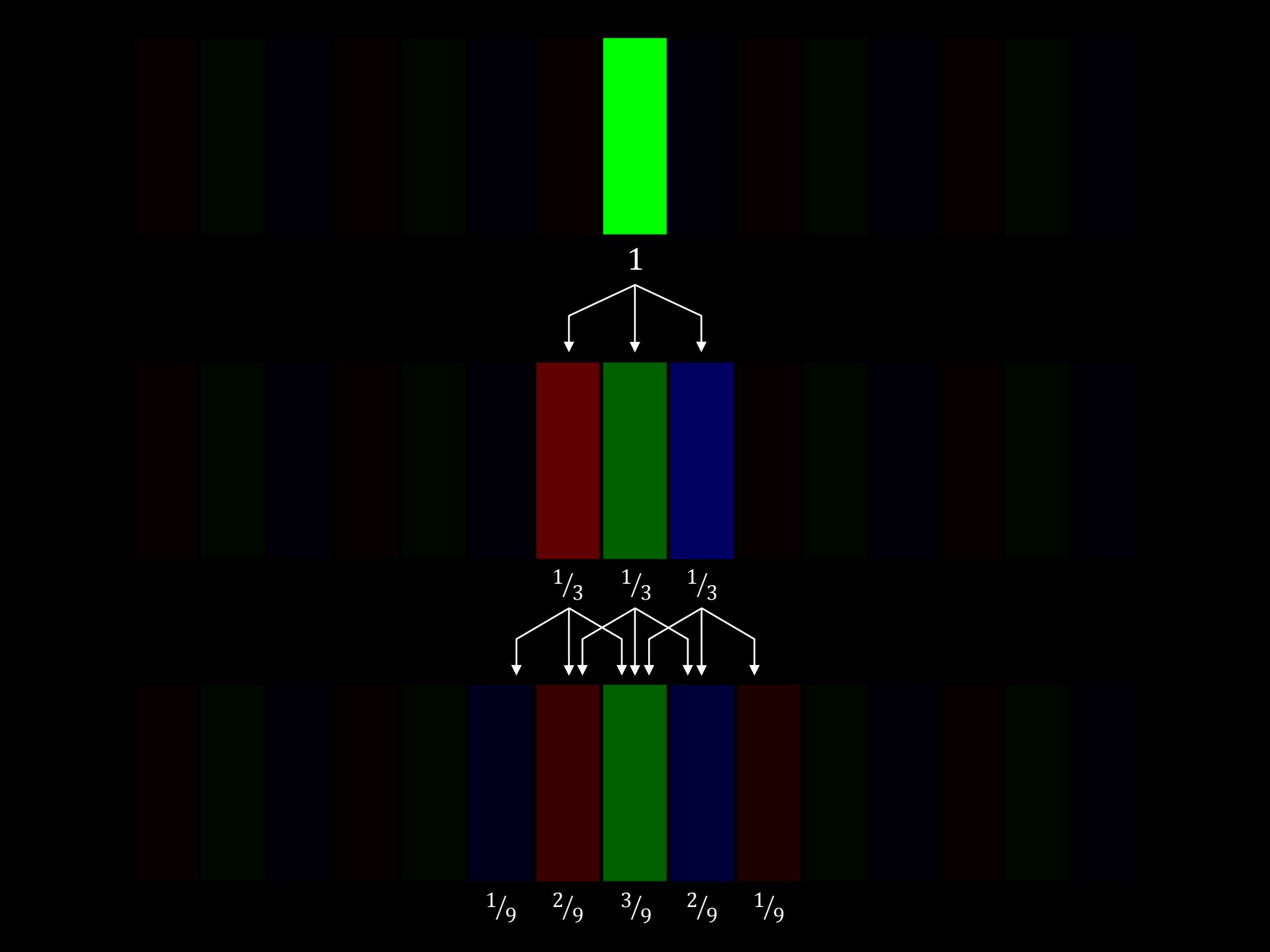


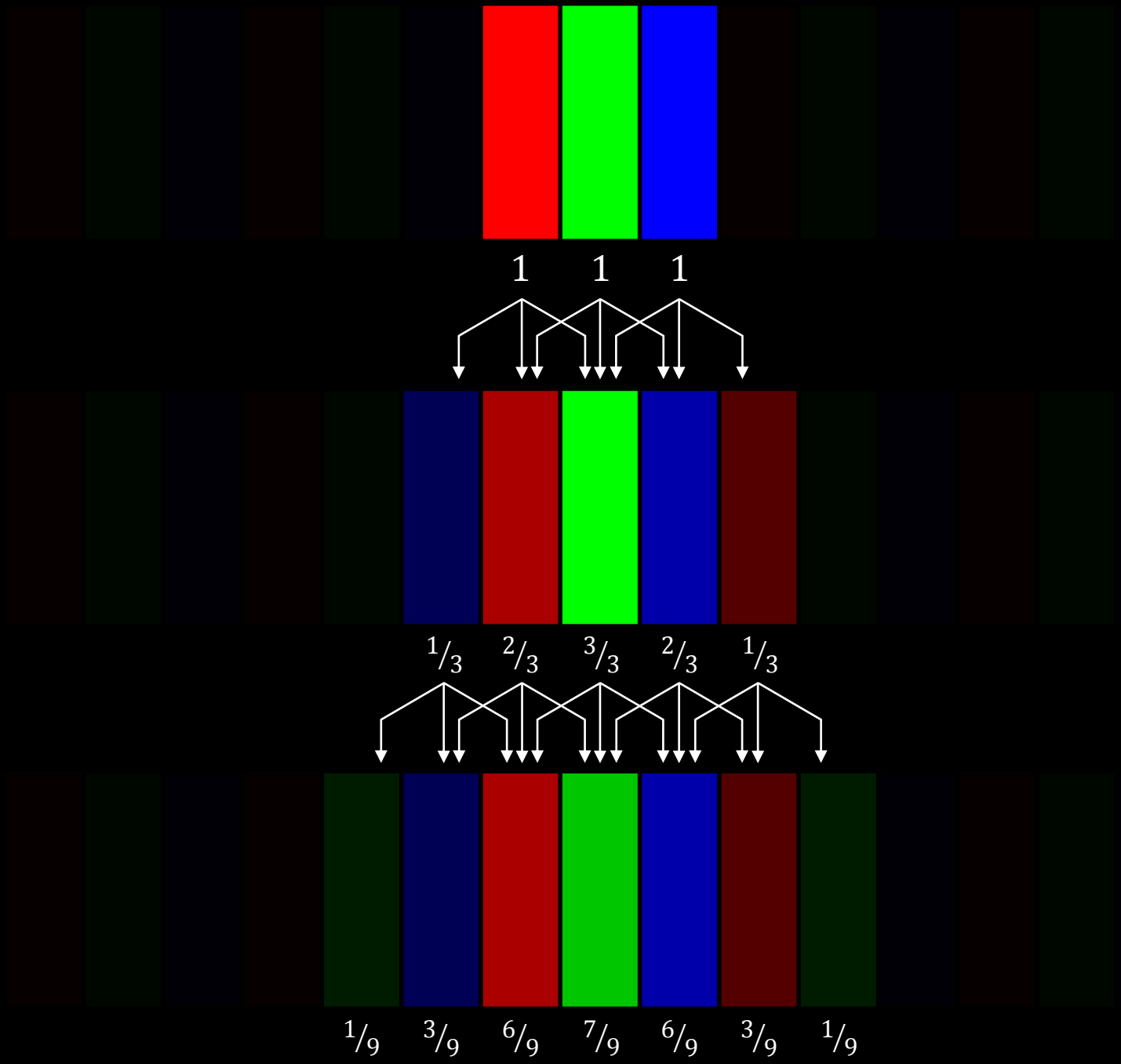


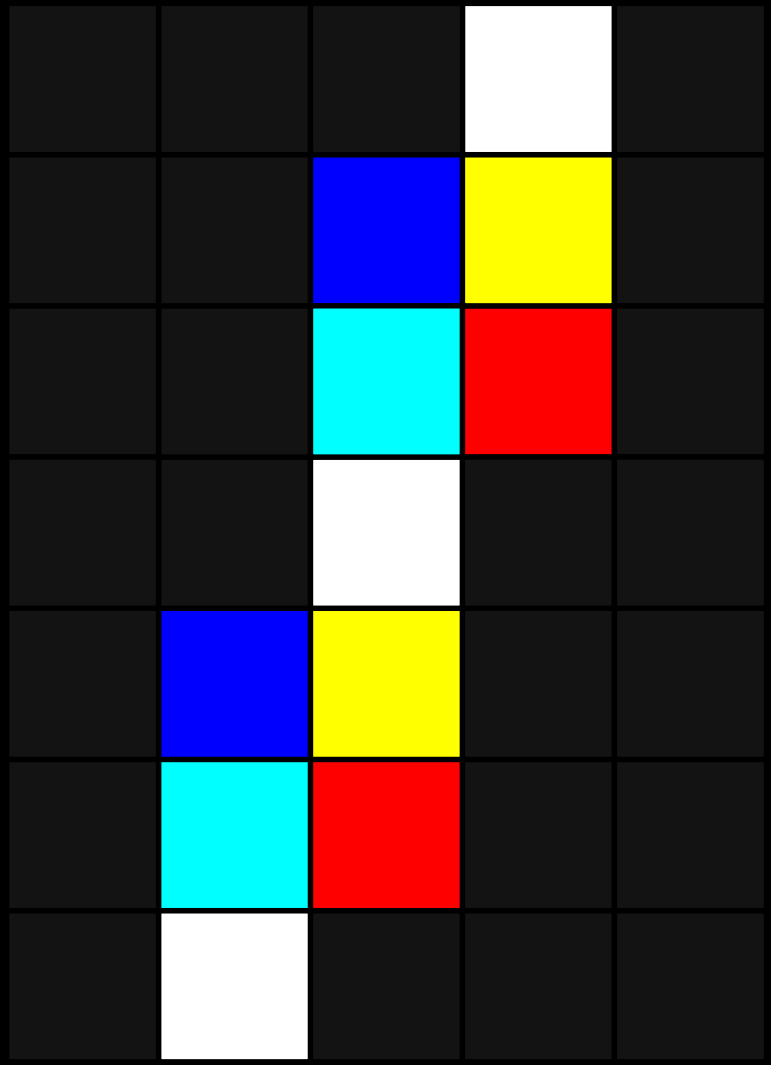


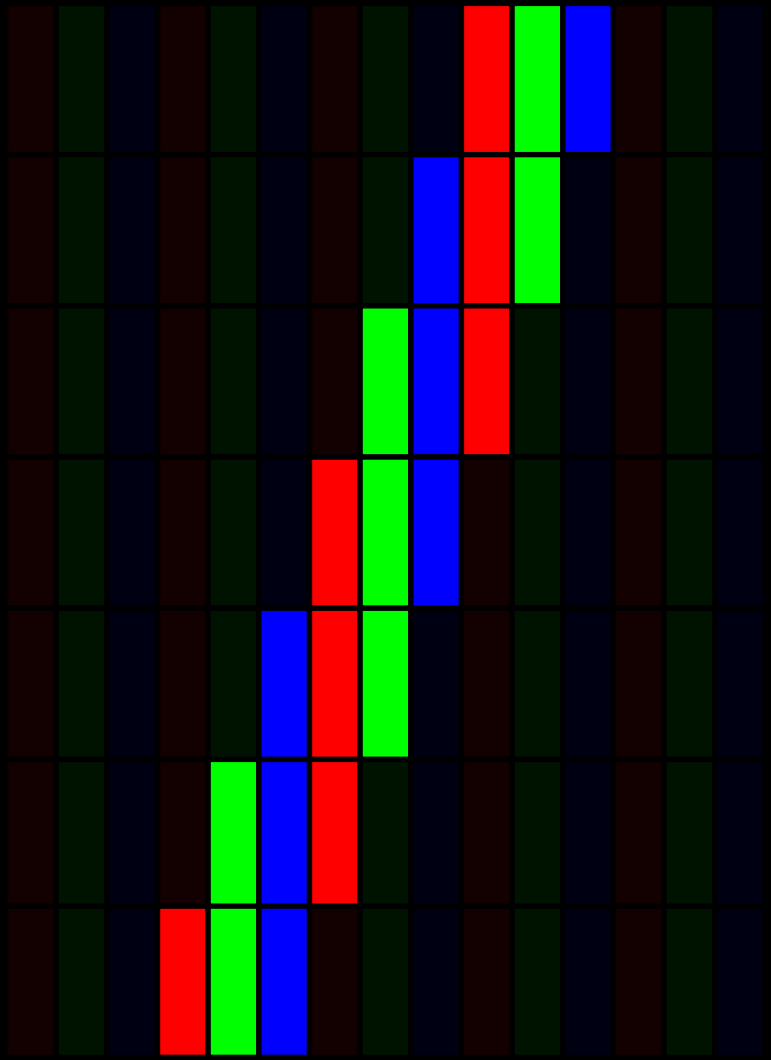


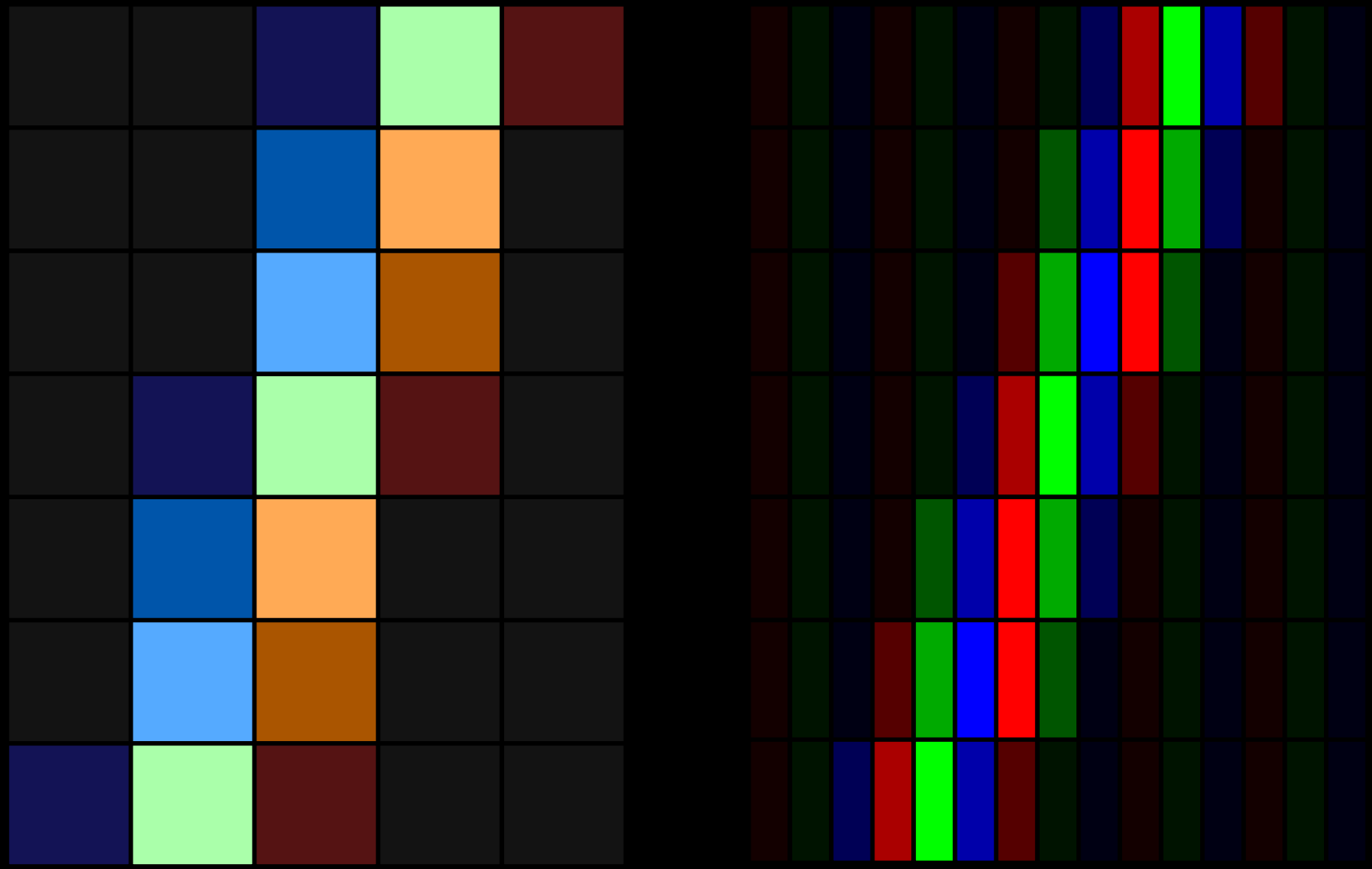



1
1/3
1/3
1/3
1/9
2/9
3/9
2/9
1/9


1 1 1
1/3 2/3 3/3 2/3 1/3
1/9 3/9 6/9 7/9 6/9 3/9 1/9



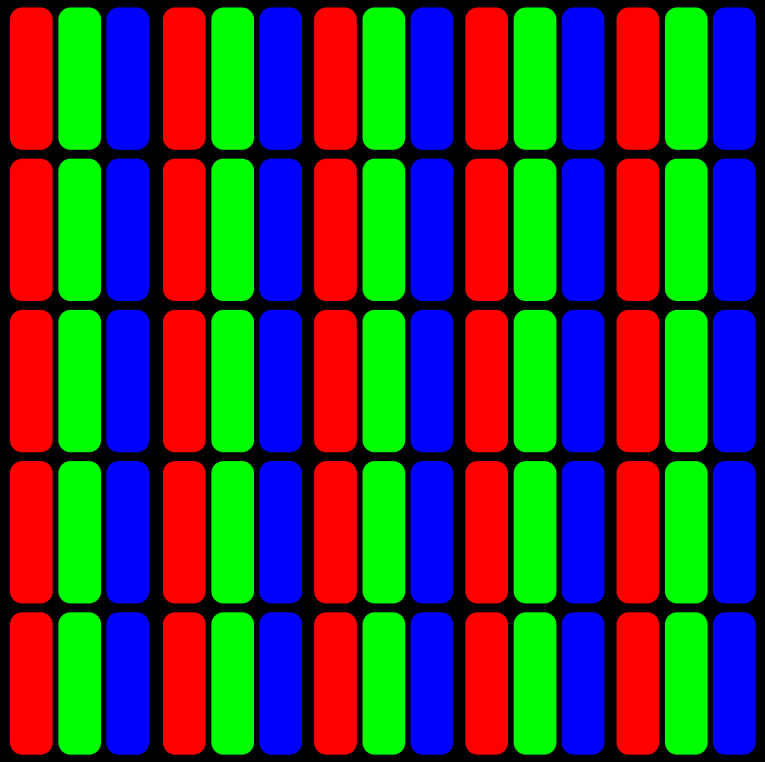




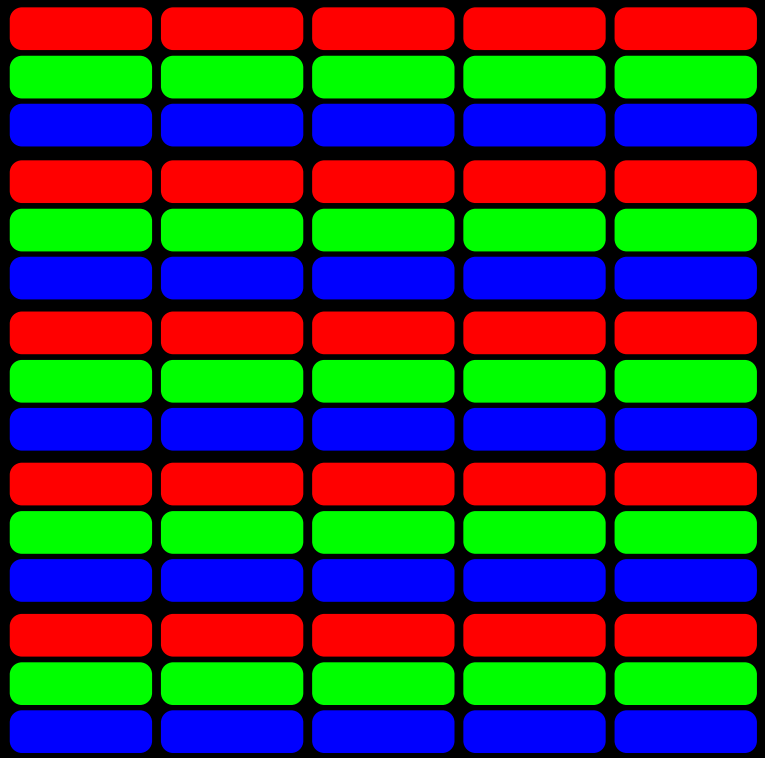
Облом с прозрачностью

Расположение субпикселей

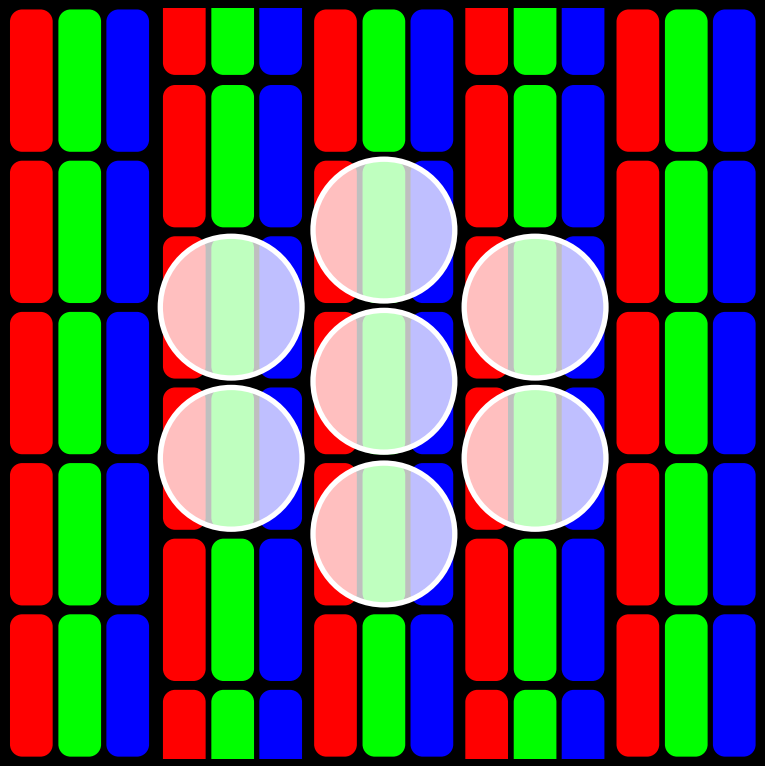
RGB (горизонтально)



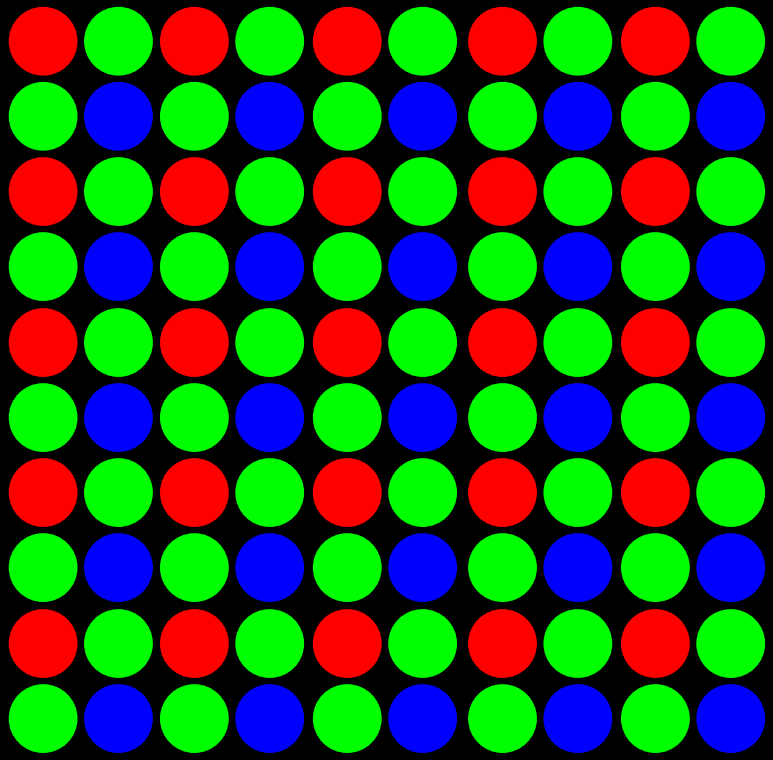
RGB (вертикально)



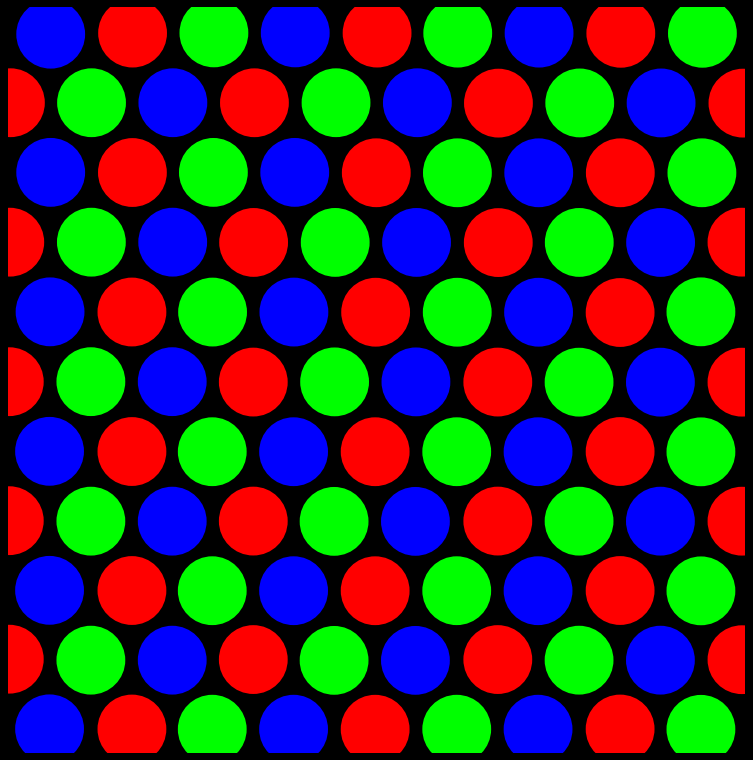
RGB (TV)



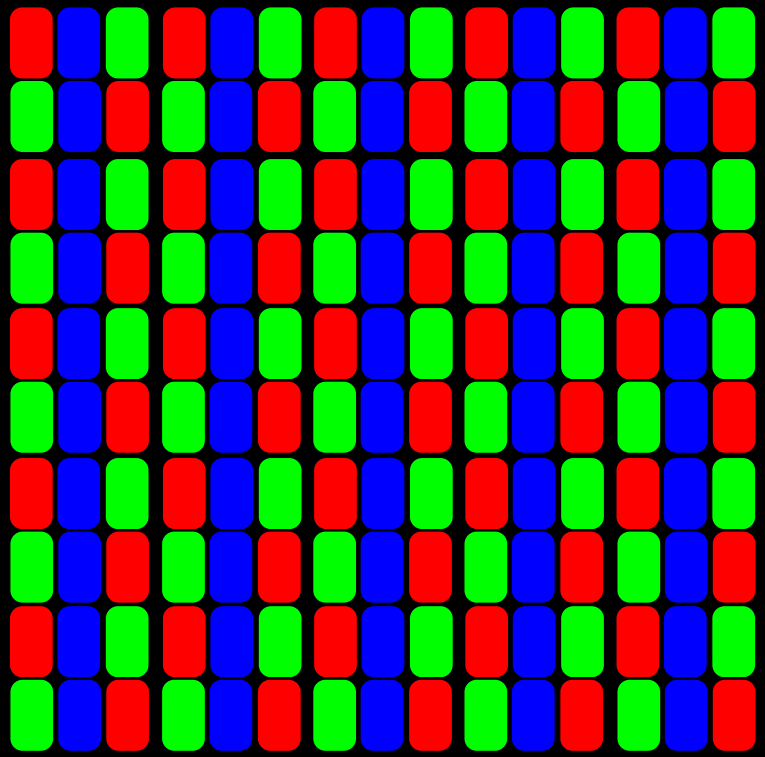
Bayer



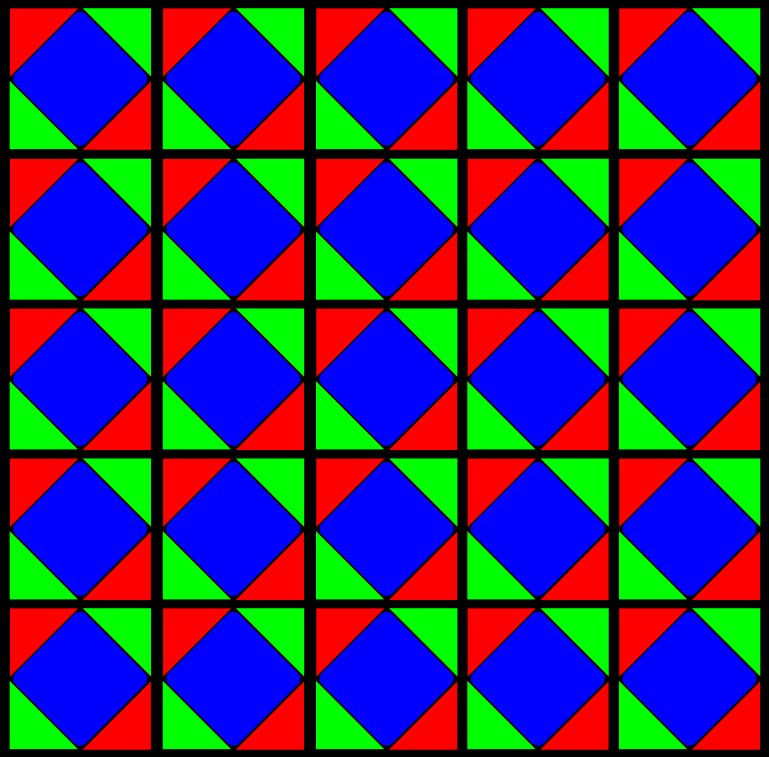
Hexagonal



PenTile RBG-GBR

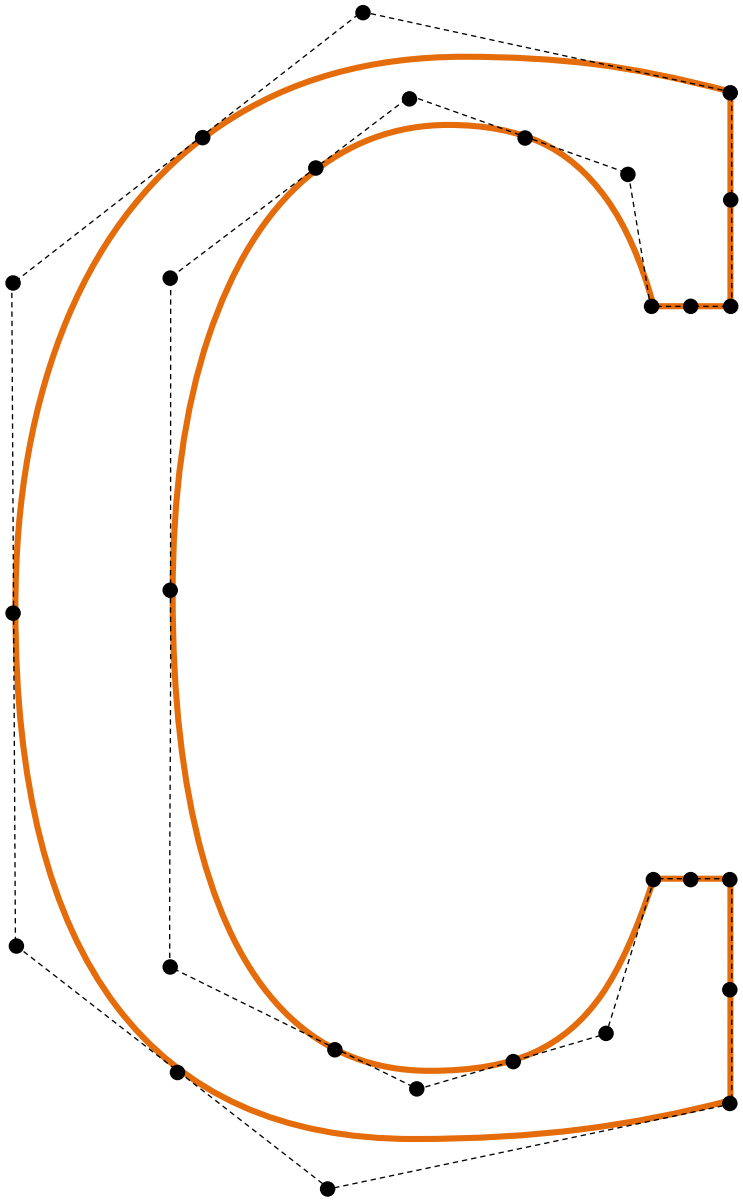


PenTile RG-B-GR



Текст

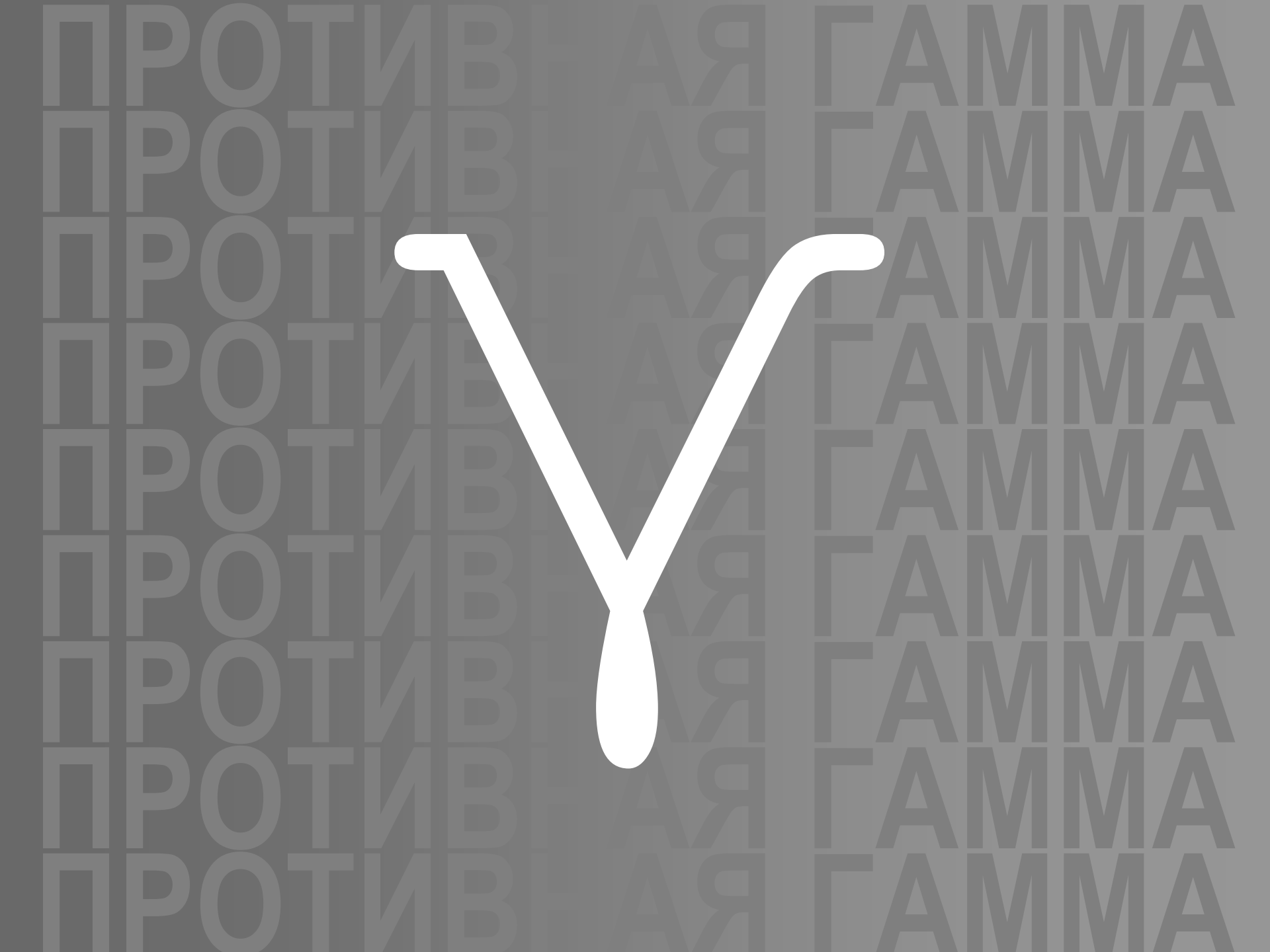

Шрифты



Faux italic

Faux italic

Faux italic

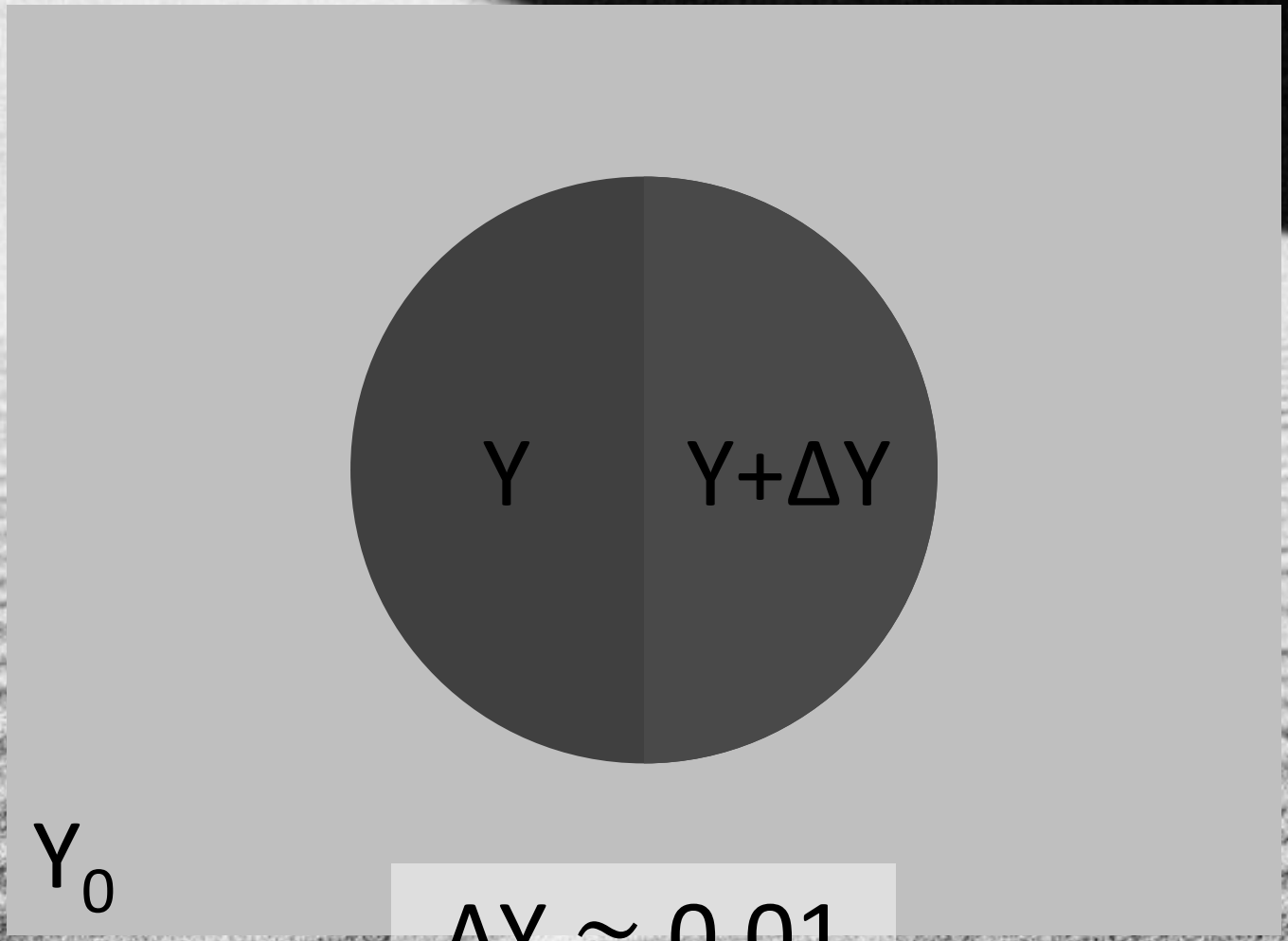

ПРОТИВНАЯ ГАММА

Контрастность зрения

100:1*

* Вообще-то 1000:1, но типичные условия не позволяют

Порог различения



Сколько нужно битов?

$$\frac{100}{0.01} = 10000$$

14 битов

$$\log_{1.01} 100 \approx 463$$

*9 битов**

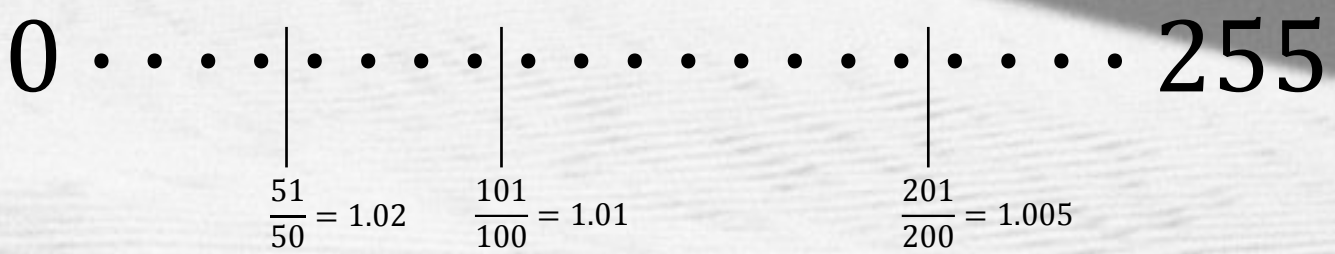
** то есть 8 :)*

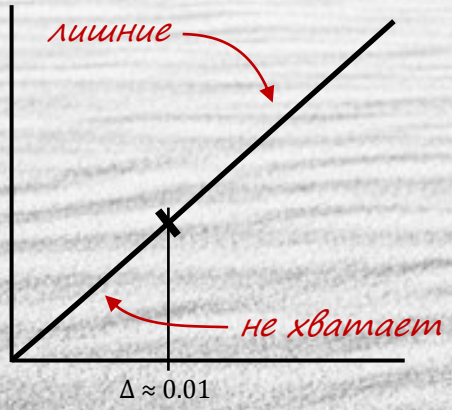
Сколько нужно битов?

256

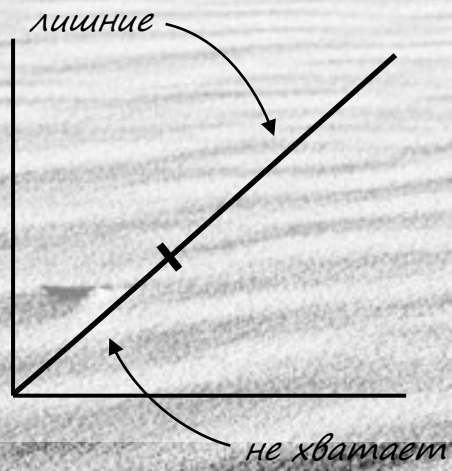
8 битов

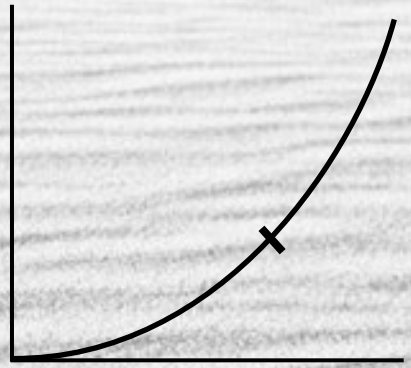
Проблема кода 100





Проблема кода 100



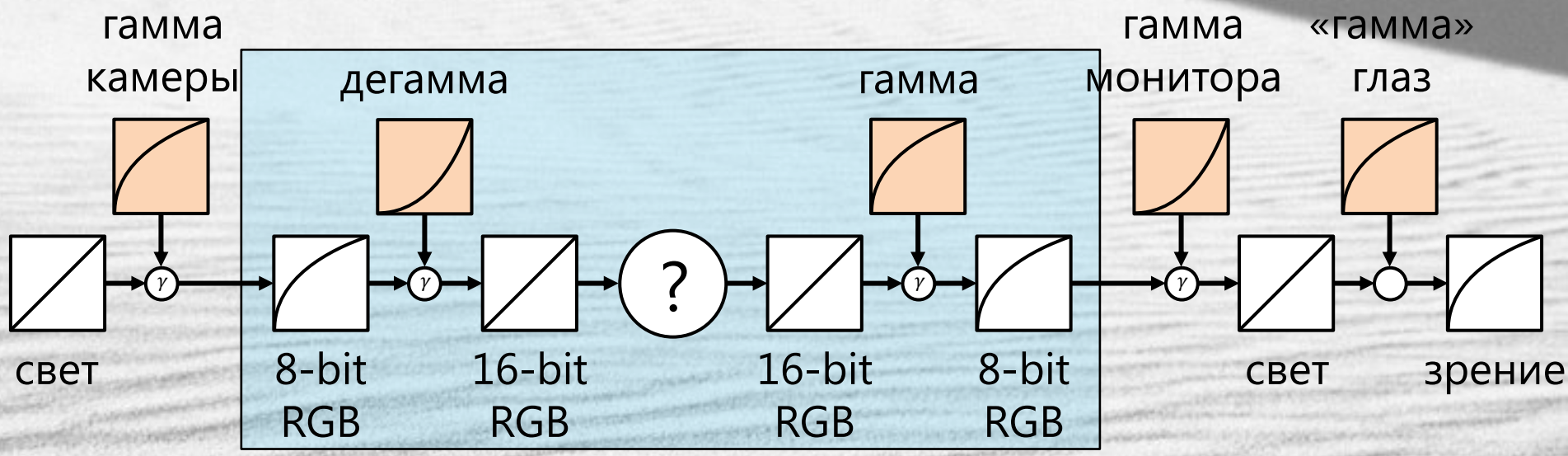


$$C = L^{\gamma}$$

$$L = C^{1/\gamma}$$

C – Corrected
L – Linear

Путь света


гамма
камеры
дегамма
гамма
гамма
монитора
«гамма»
глаз
γ
γ
γ
γ
?
свет
8-bit
RGB
16-bit
RGB
16-bit
RGB
8-bit
RGB
свет
зрение

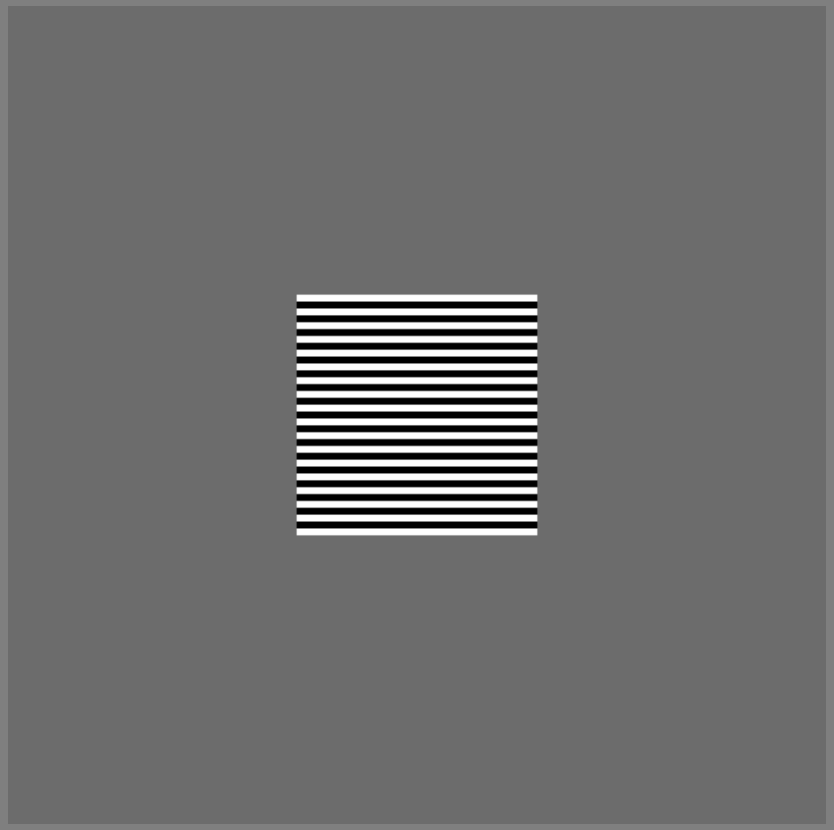
$$C = L^{\gamma}$$

$$L = C^{1/\gamma}$$

C – Corrected
L – Linear

для наших глаз
свет более светлый,
чем темнота тёмная

Калибрация

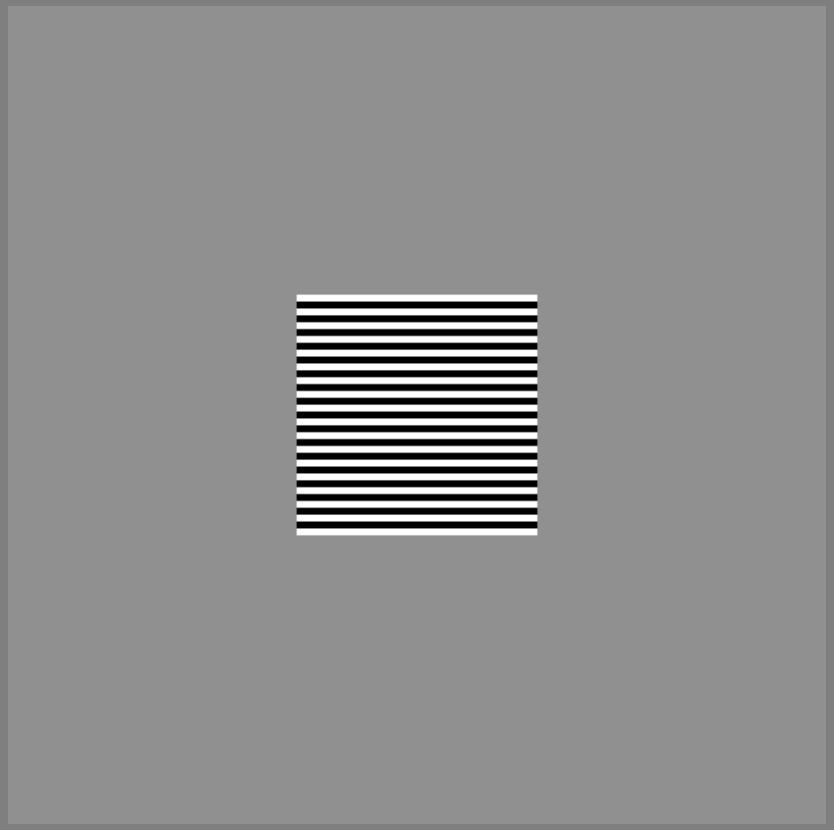


0.8

Калибрация

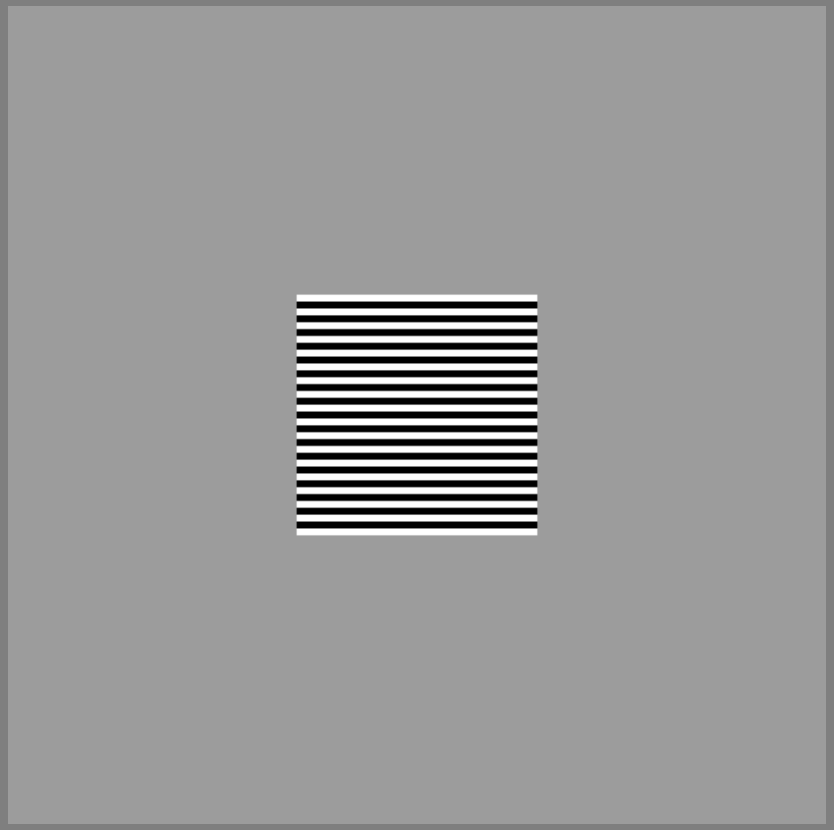
1.0

Калибрация



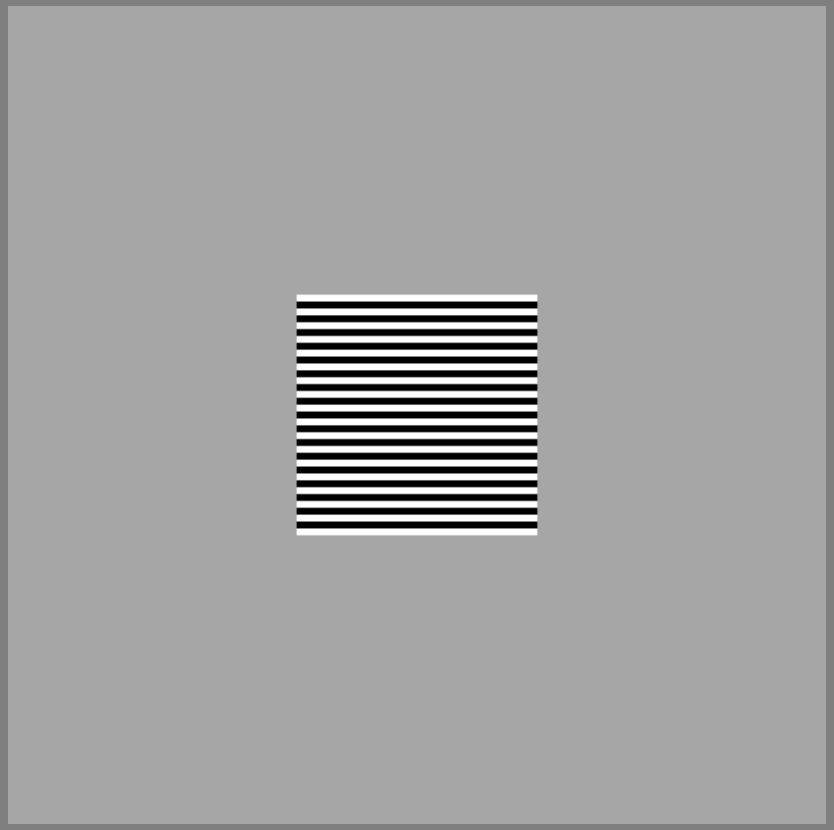
1.2

Калибрация



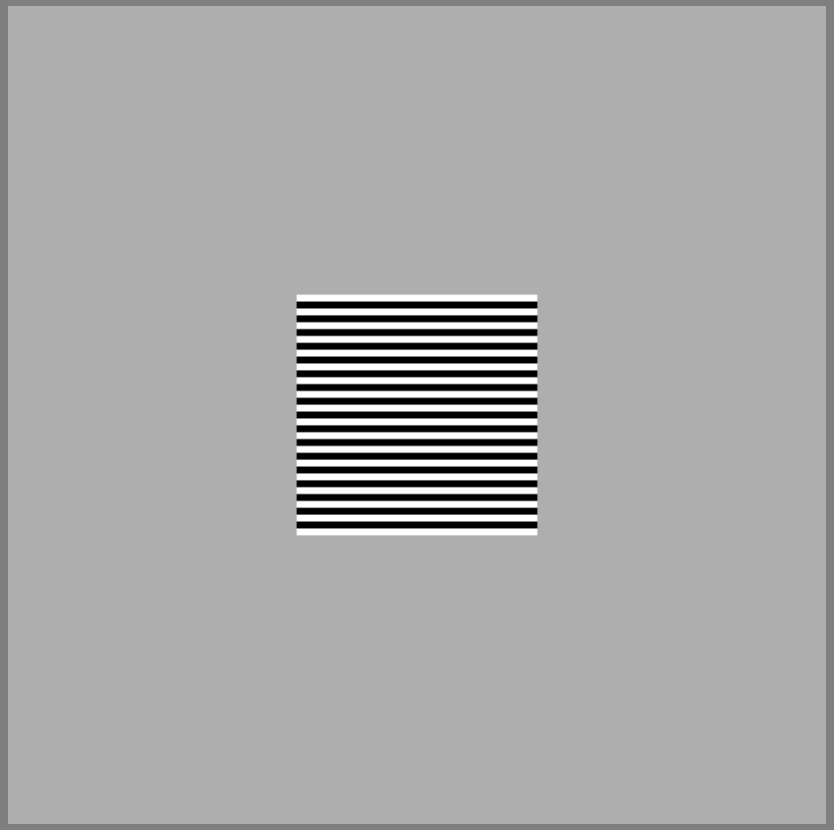
1.4

Калибрация



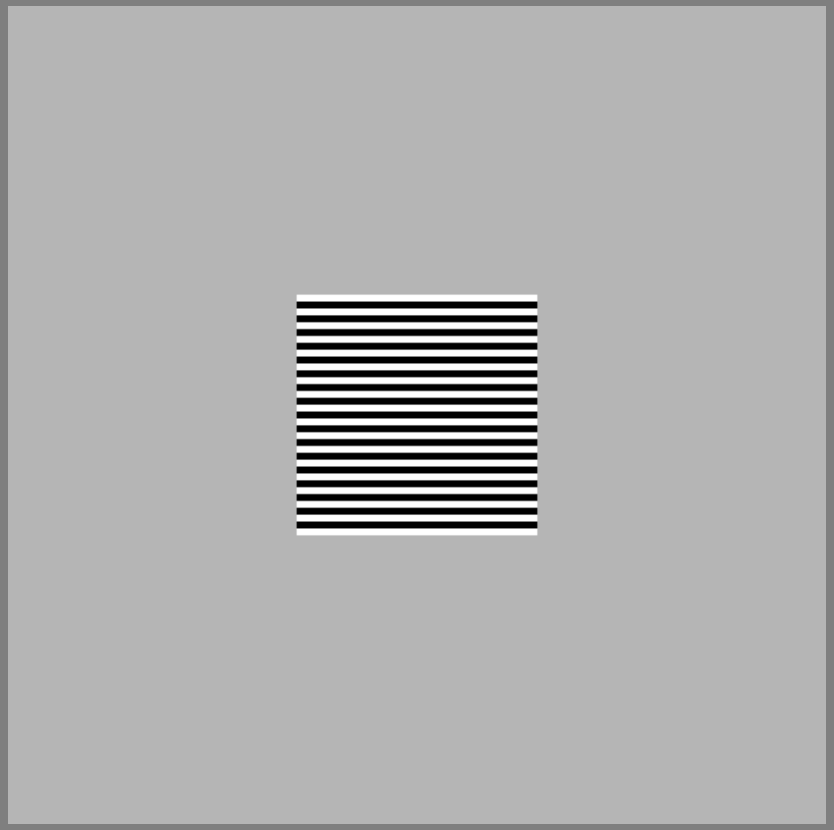
1.6

Калибрация



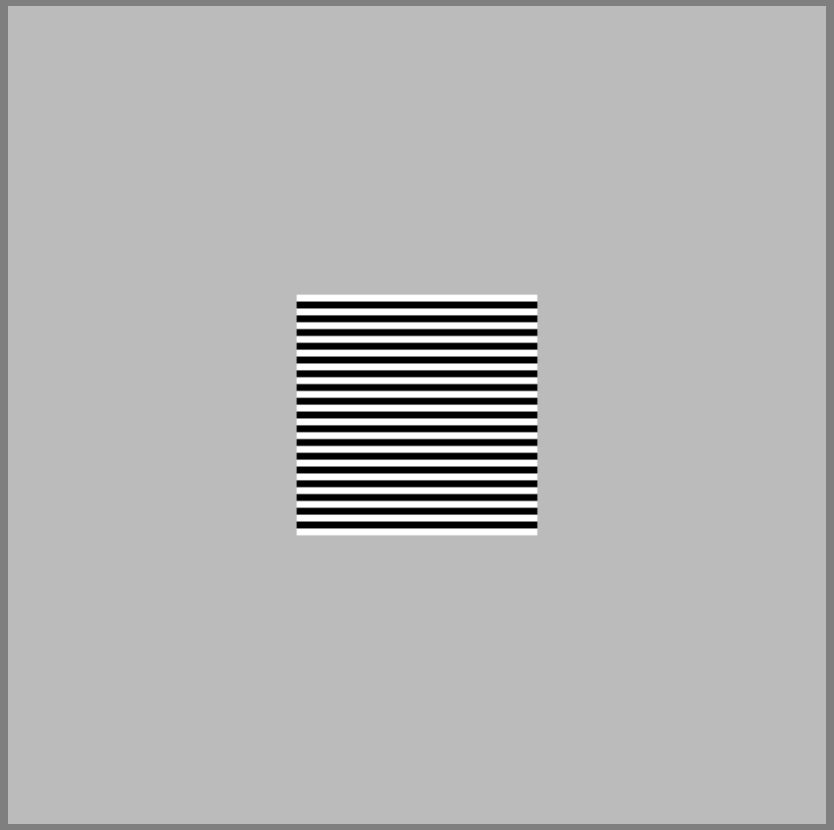
1.8

Калибрация

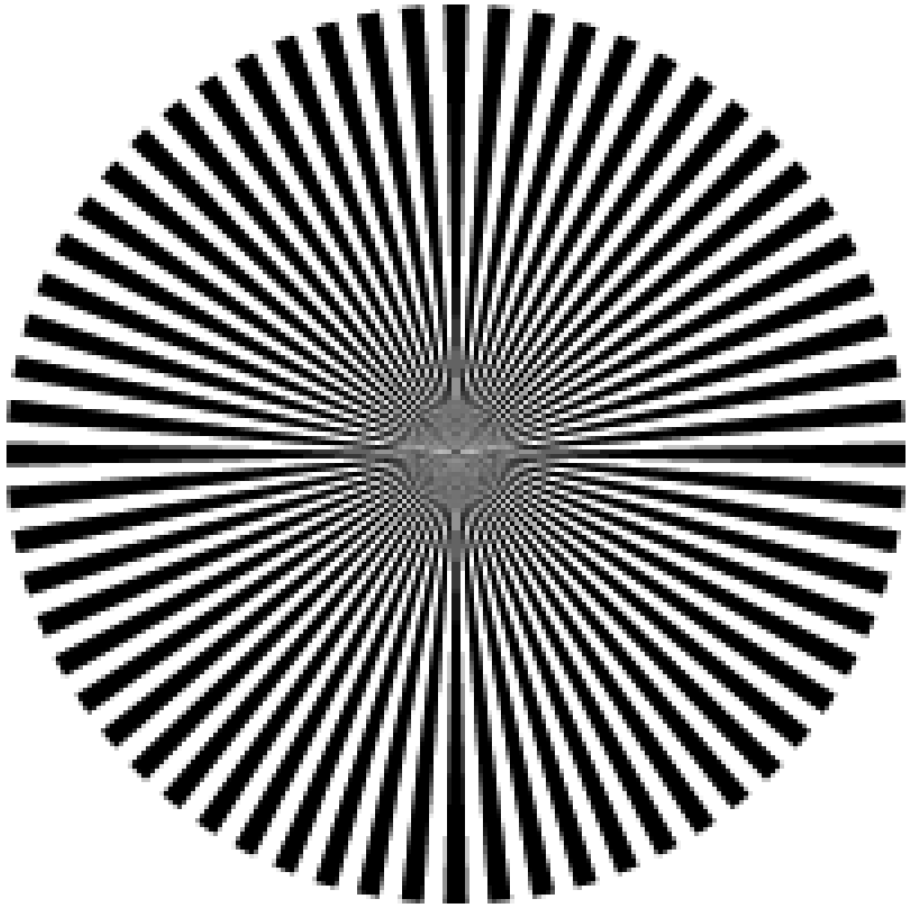


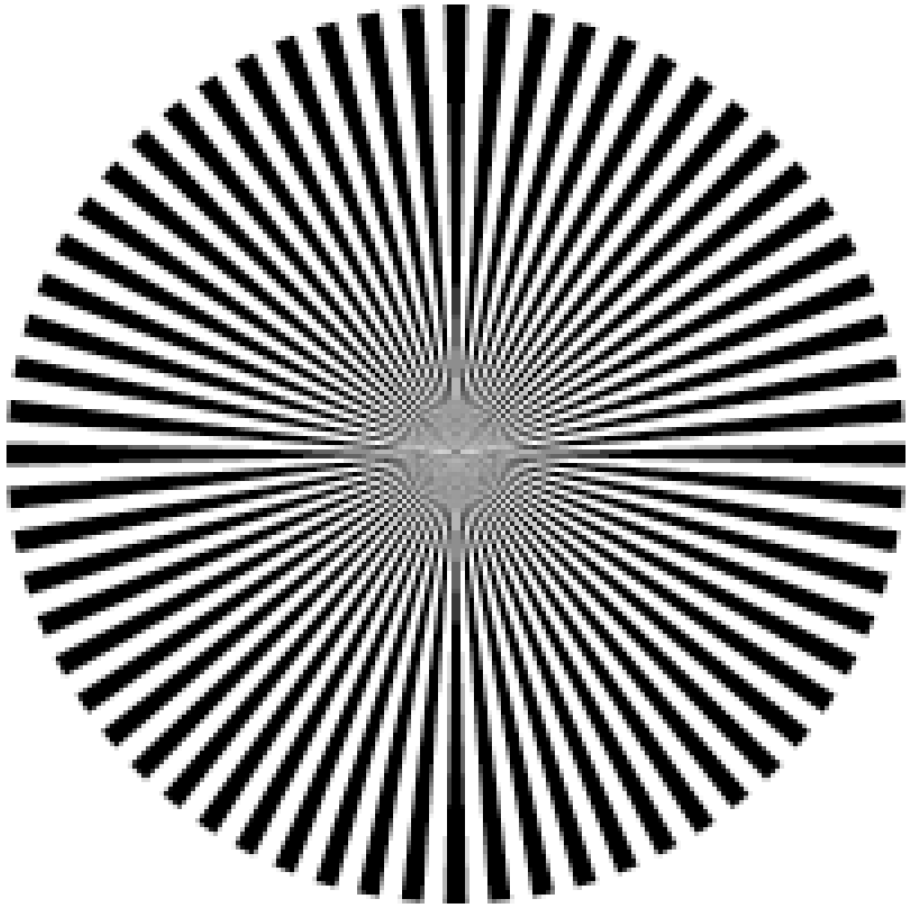
2.0

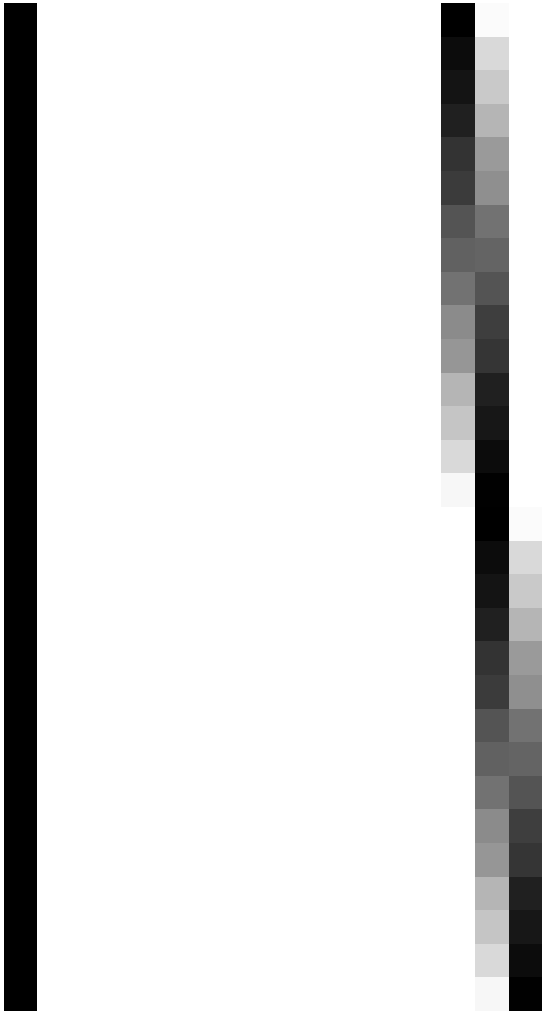
Калибрация

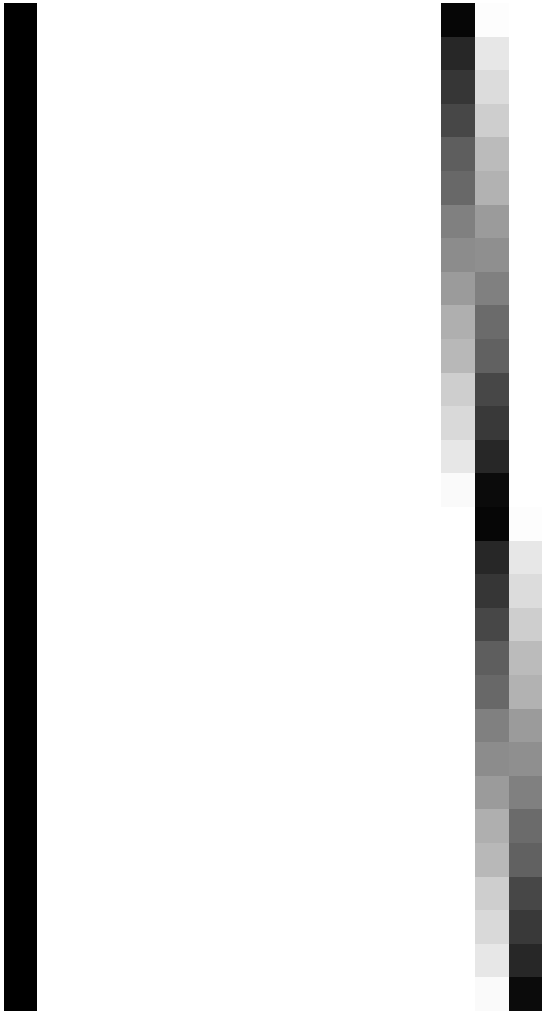


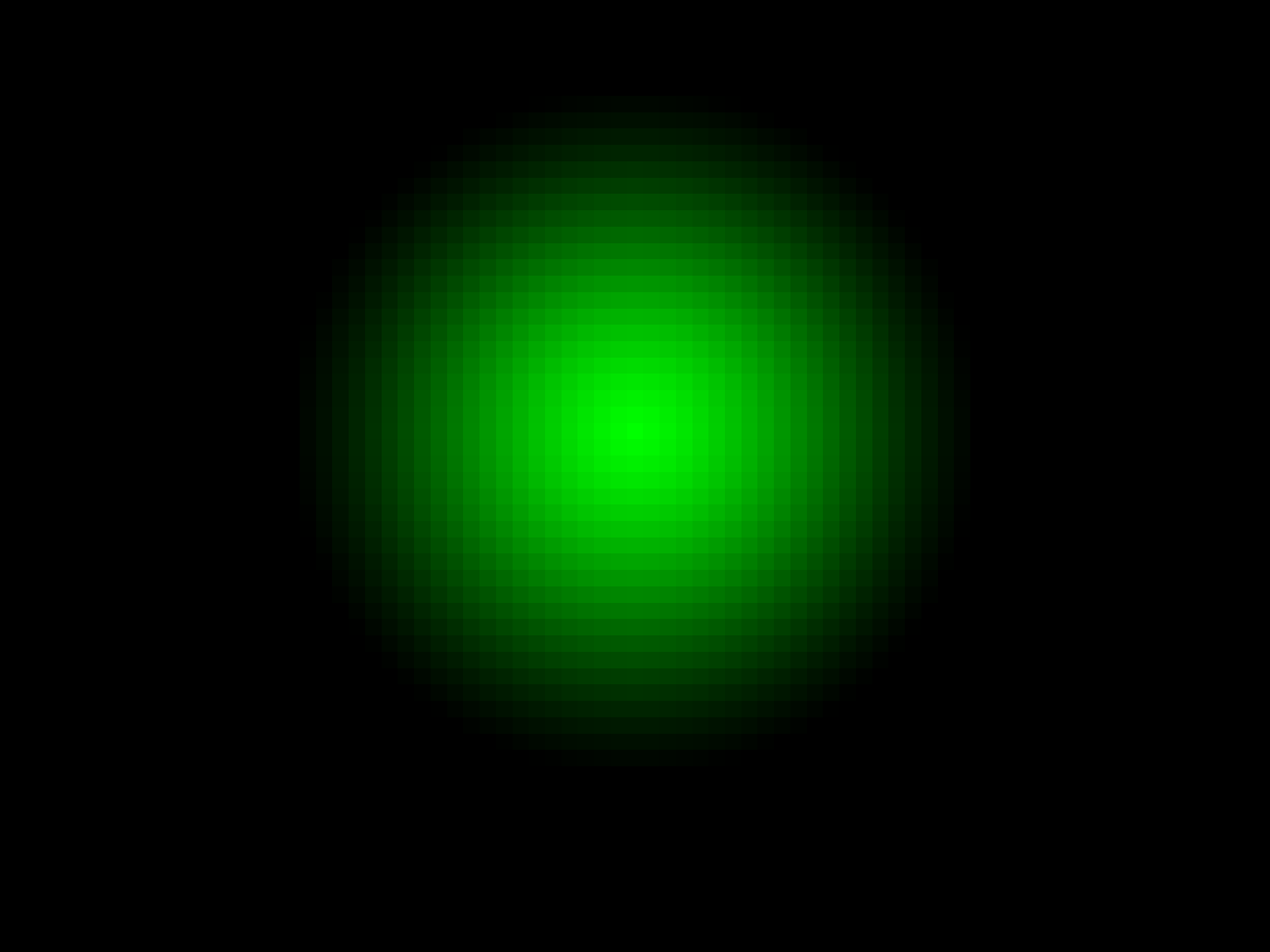
2.2

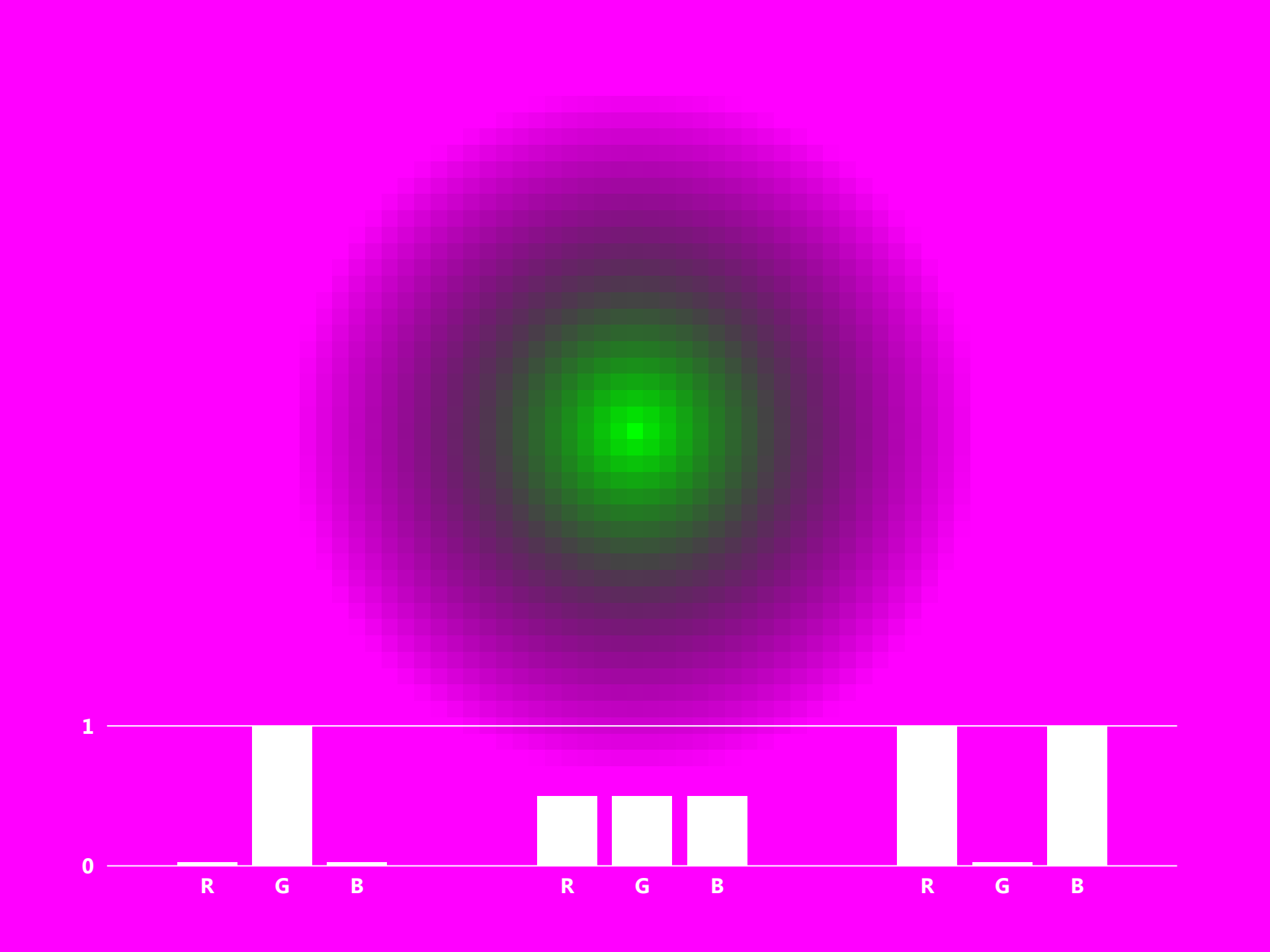


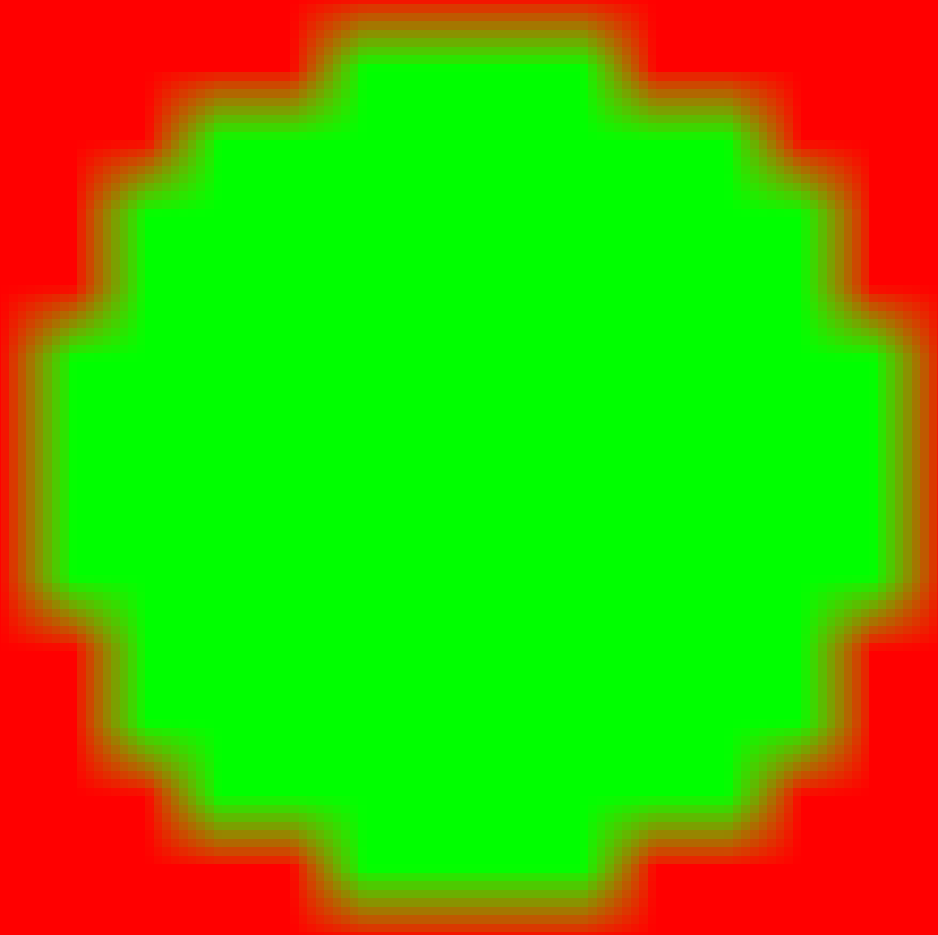


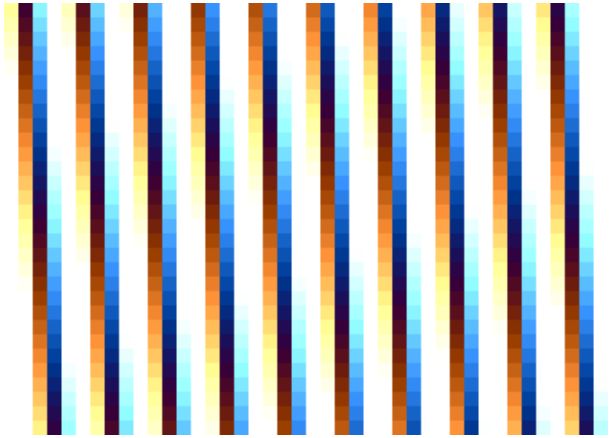


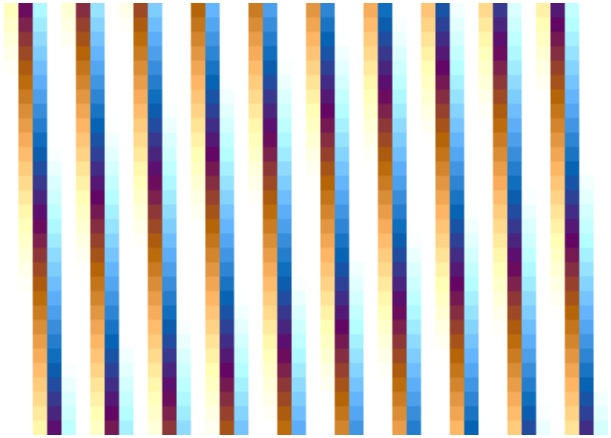




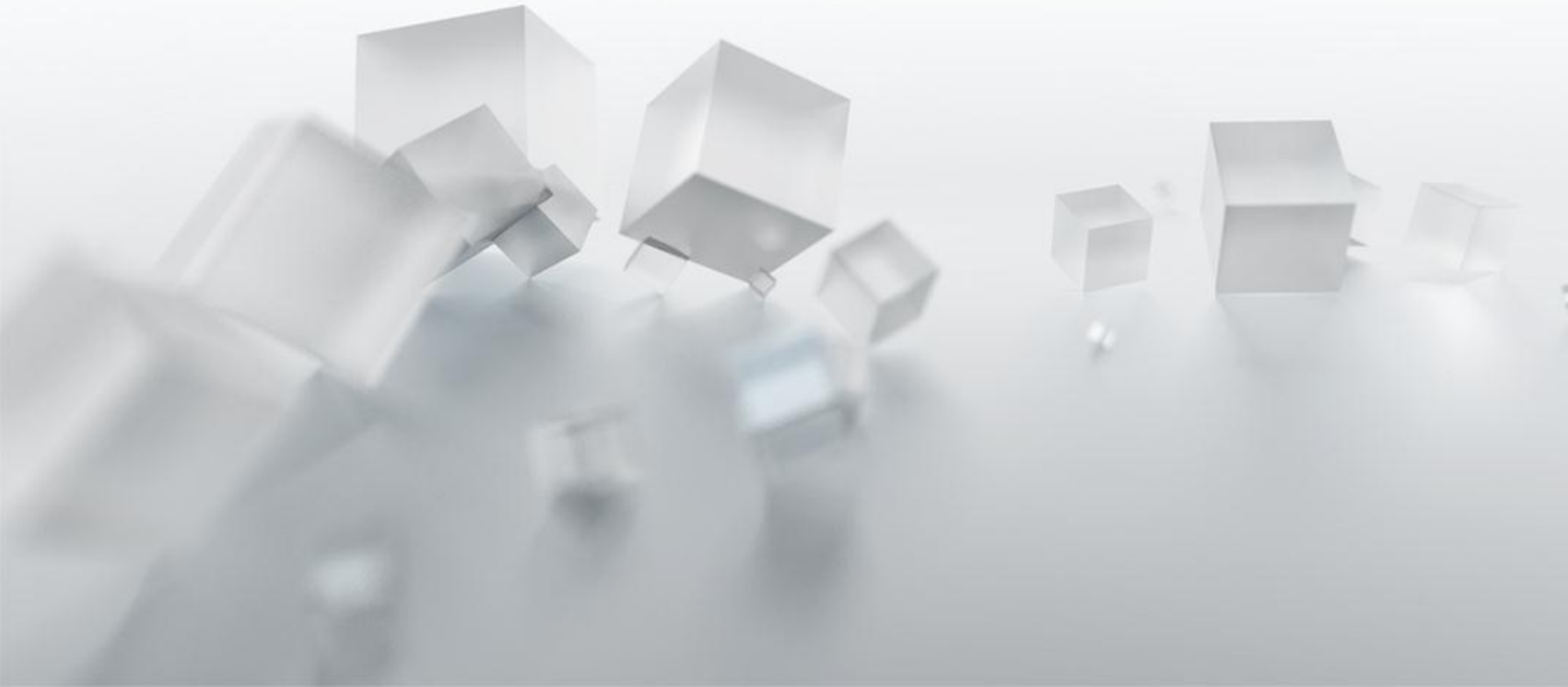

1
0
R
G
B
R
G
B
R
G
B



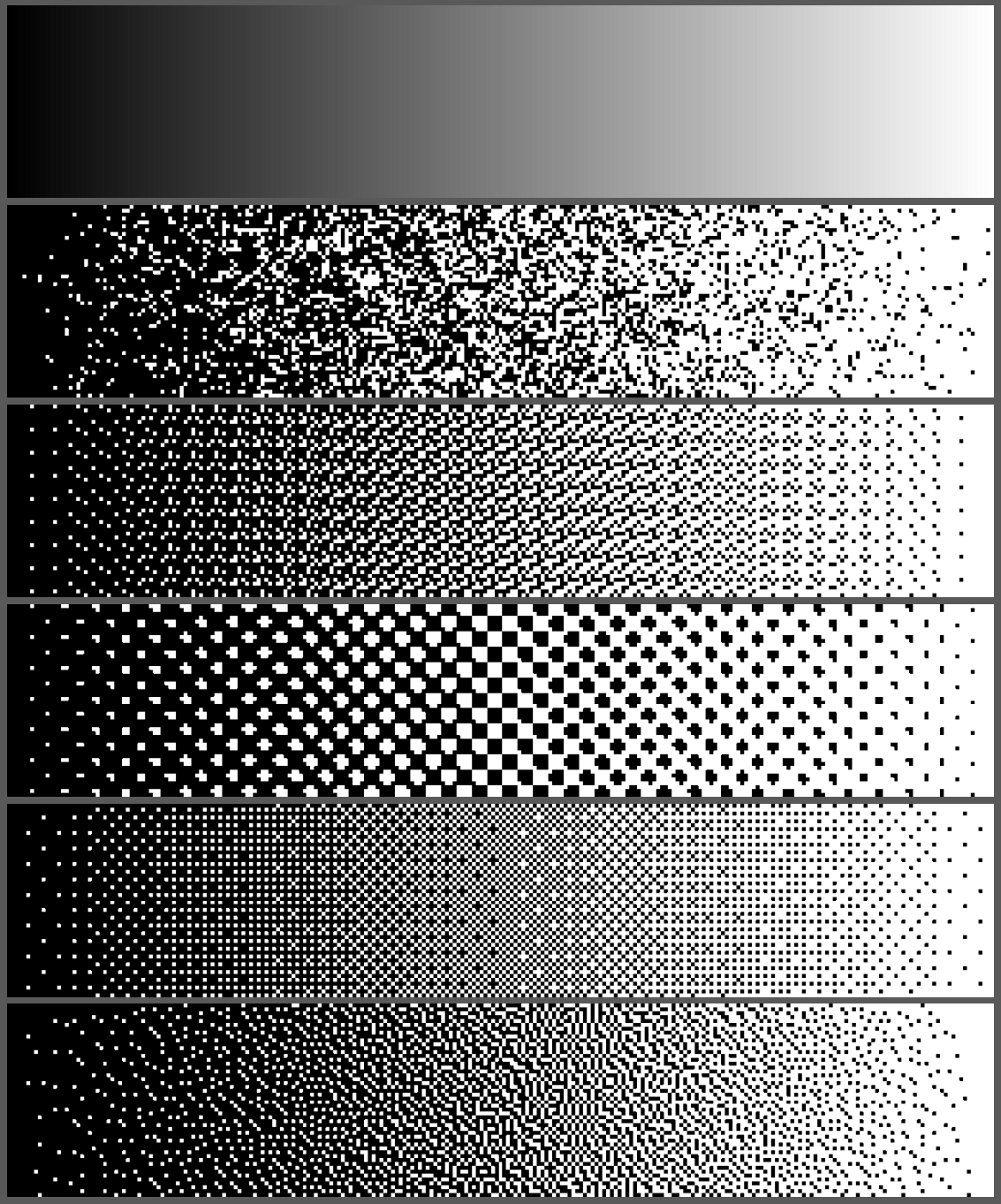




Разброс
(dithering)



10/9/11



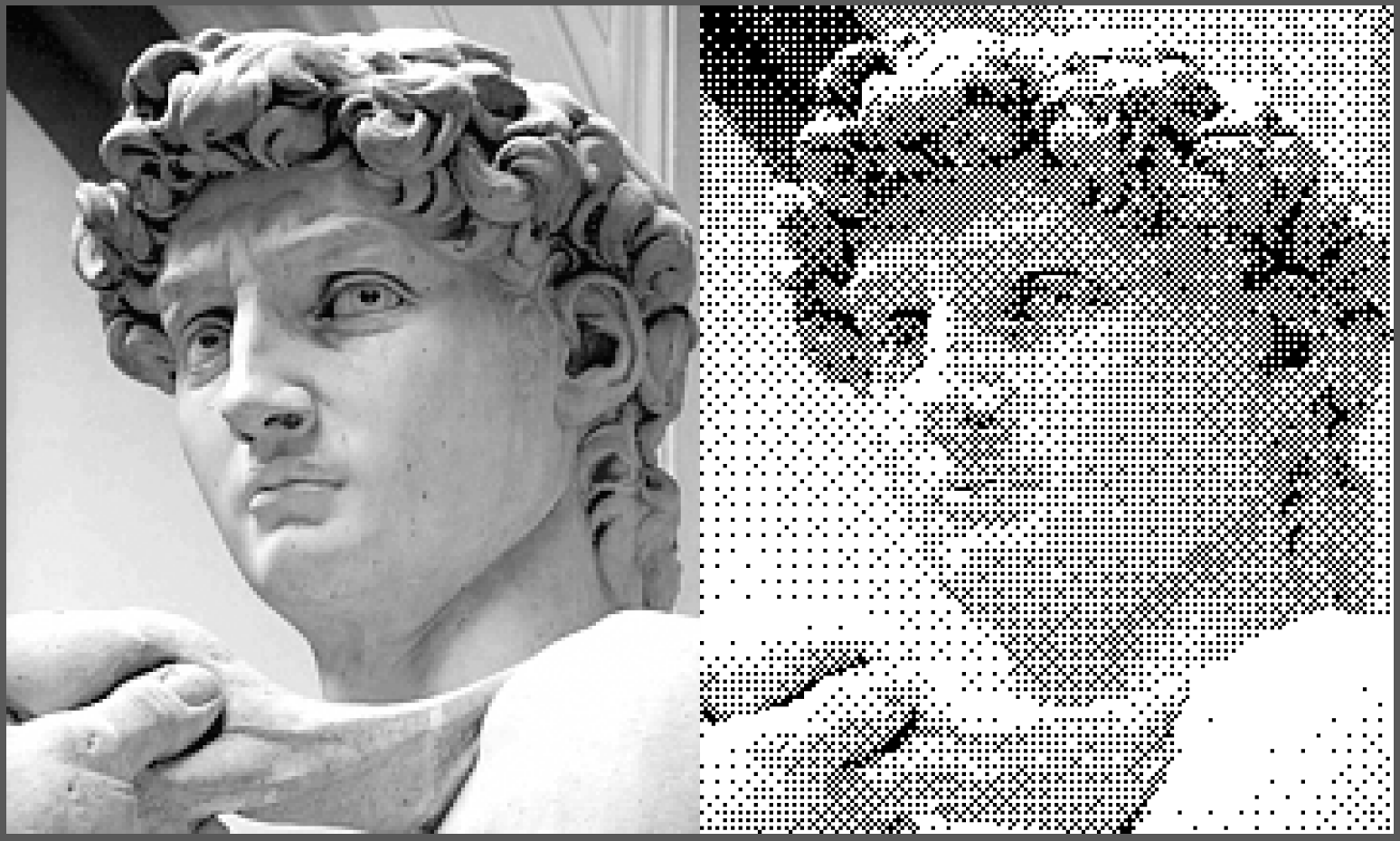
прозрачность

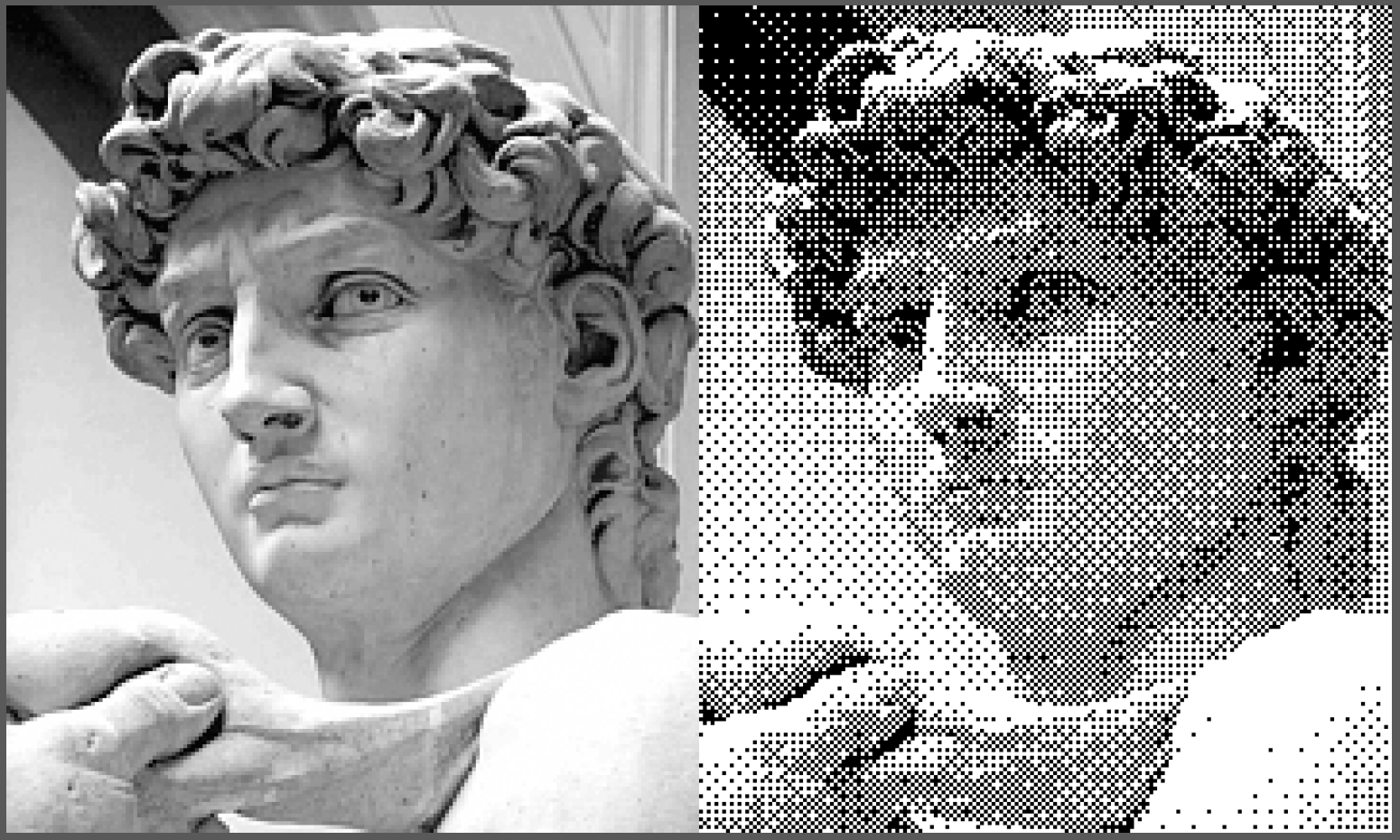
распыление

полутон

Bayer

Floyd-Steinberg






Оптимизация

Выбор языка

C (недоC++)

MMX

Исключение как правило

```
GetPixel(int x, int y);
SetPixel(int x, int y, int color);
```

```
pix8* buffer = image->GetBuffer();
int imageWidth = image->GetWidth();

...buffer[y * imageWidth + x]...
```

```
if (...)
{

}
else
{

}
```

~~`int c = min(a, b); // c = a < b ? a : b`~~

```
int c = a + (((b - a) >> 31) & (b - a));
```

```
int b = a / 3;
int b = (int)(((__int64)a * 0x55555556) >> 32);
```

Устали? ☺