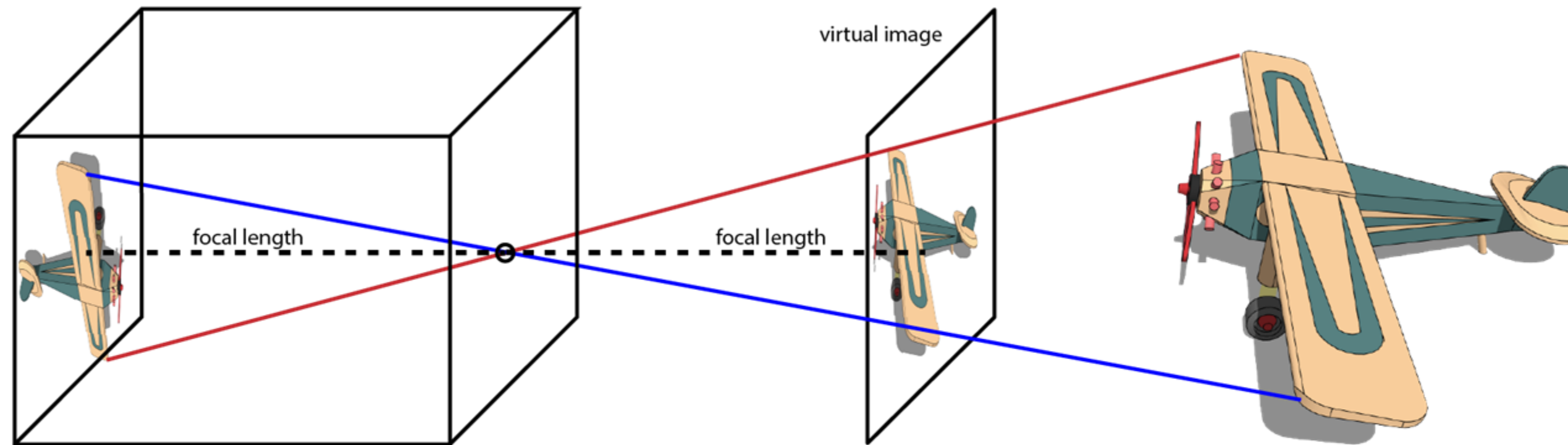
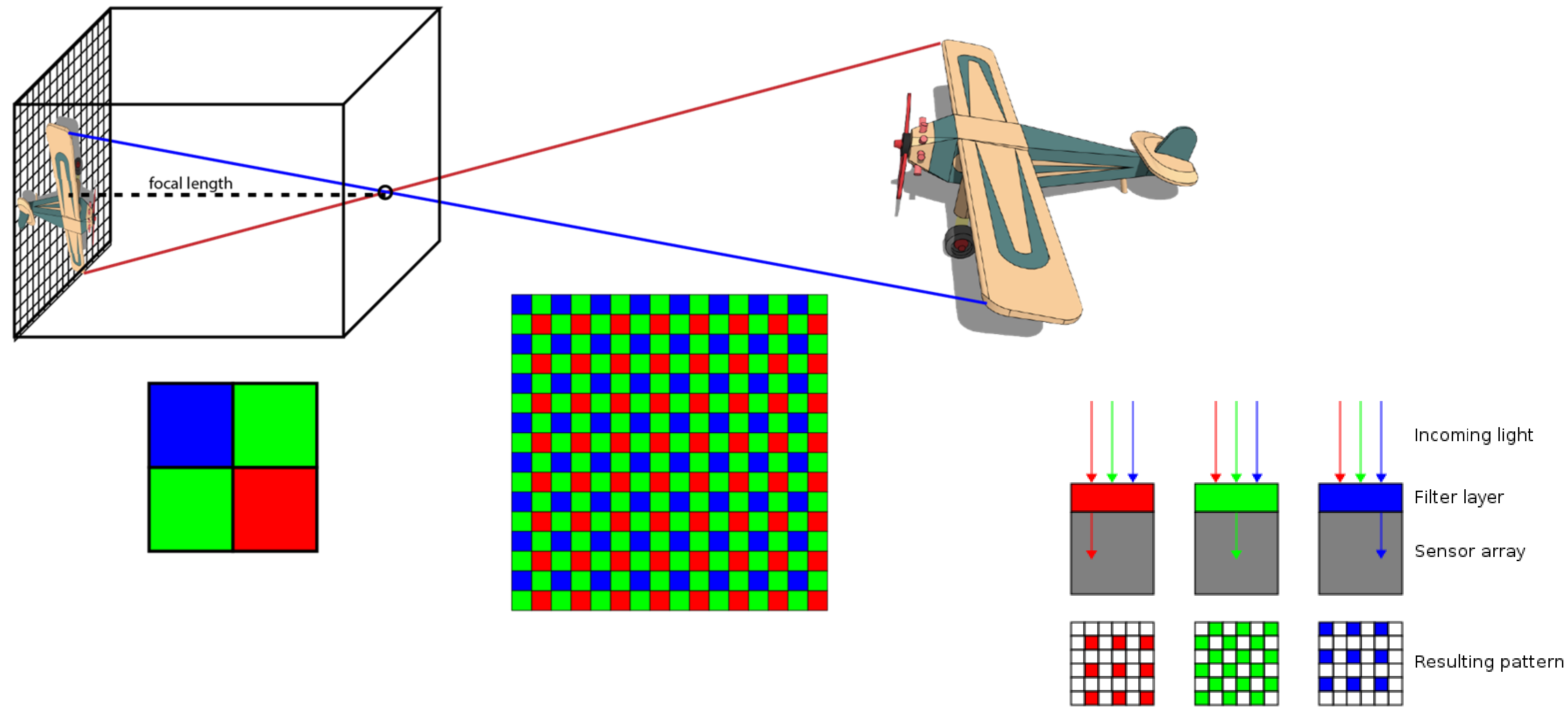


# Вспомогательные материалы

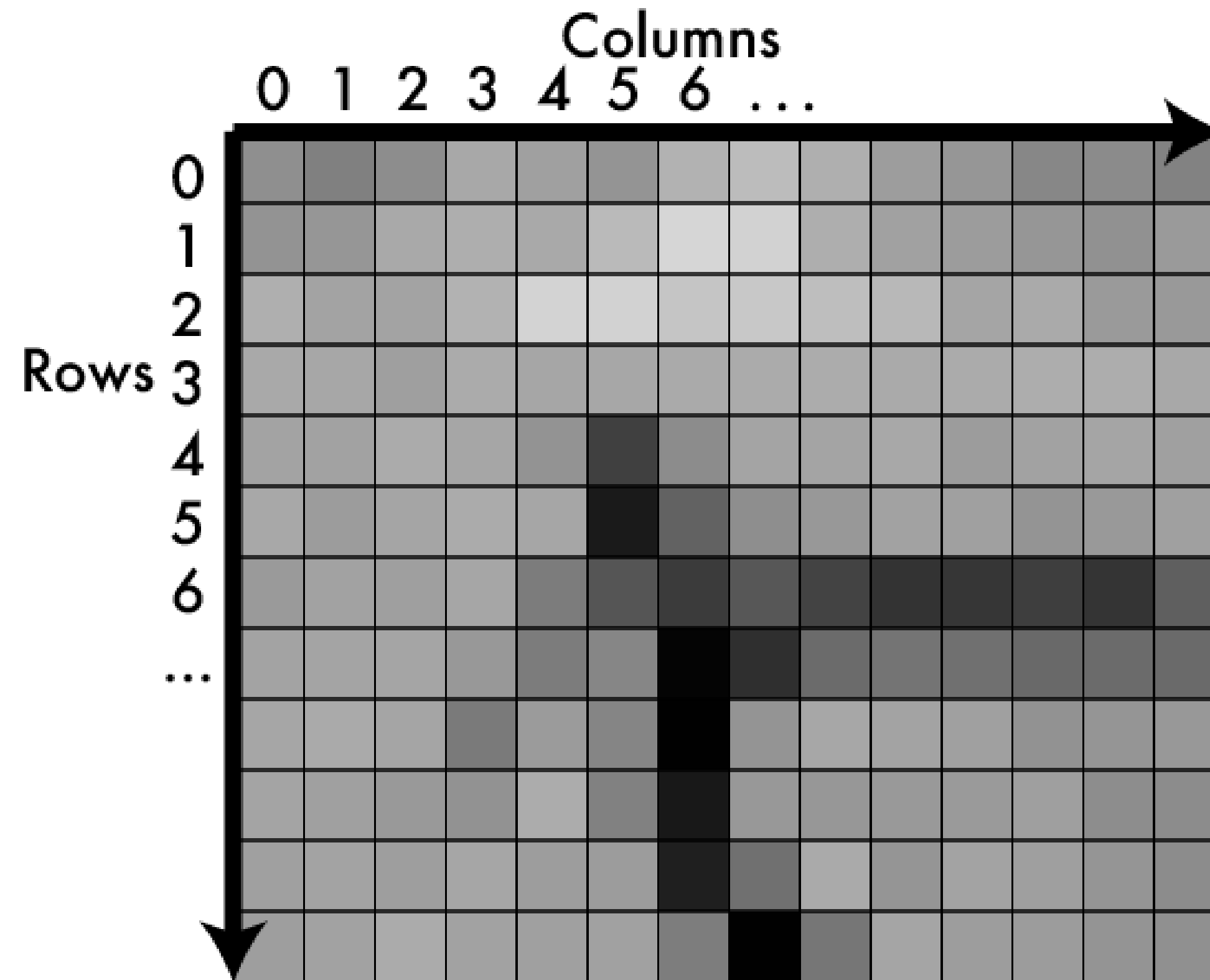
# Model: pinhole camera



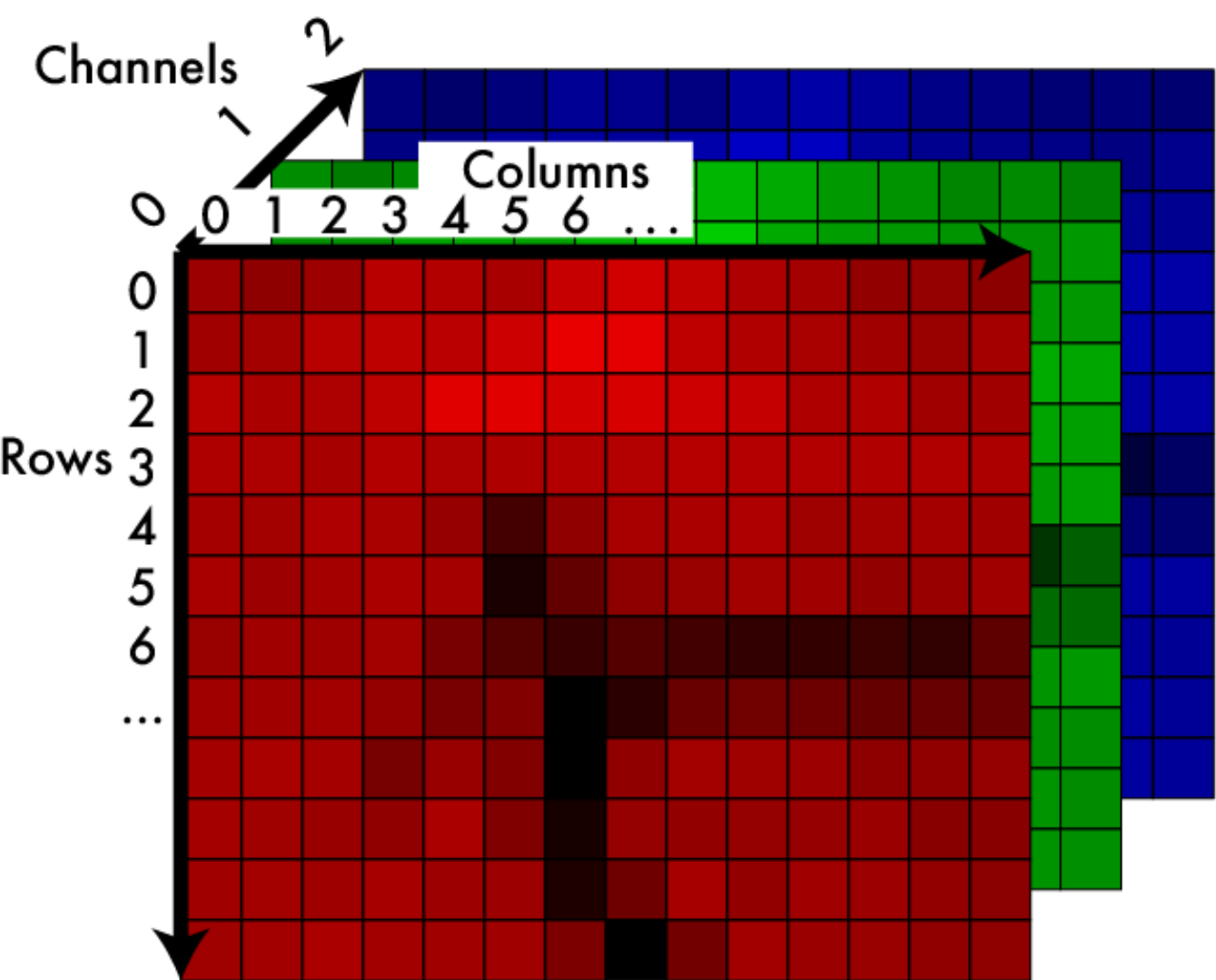
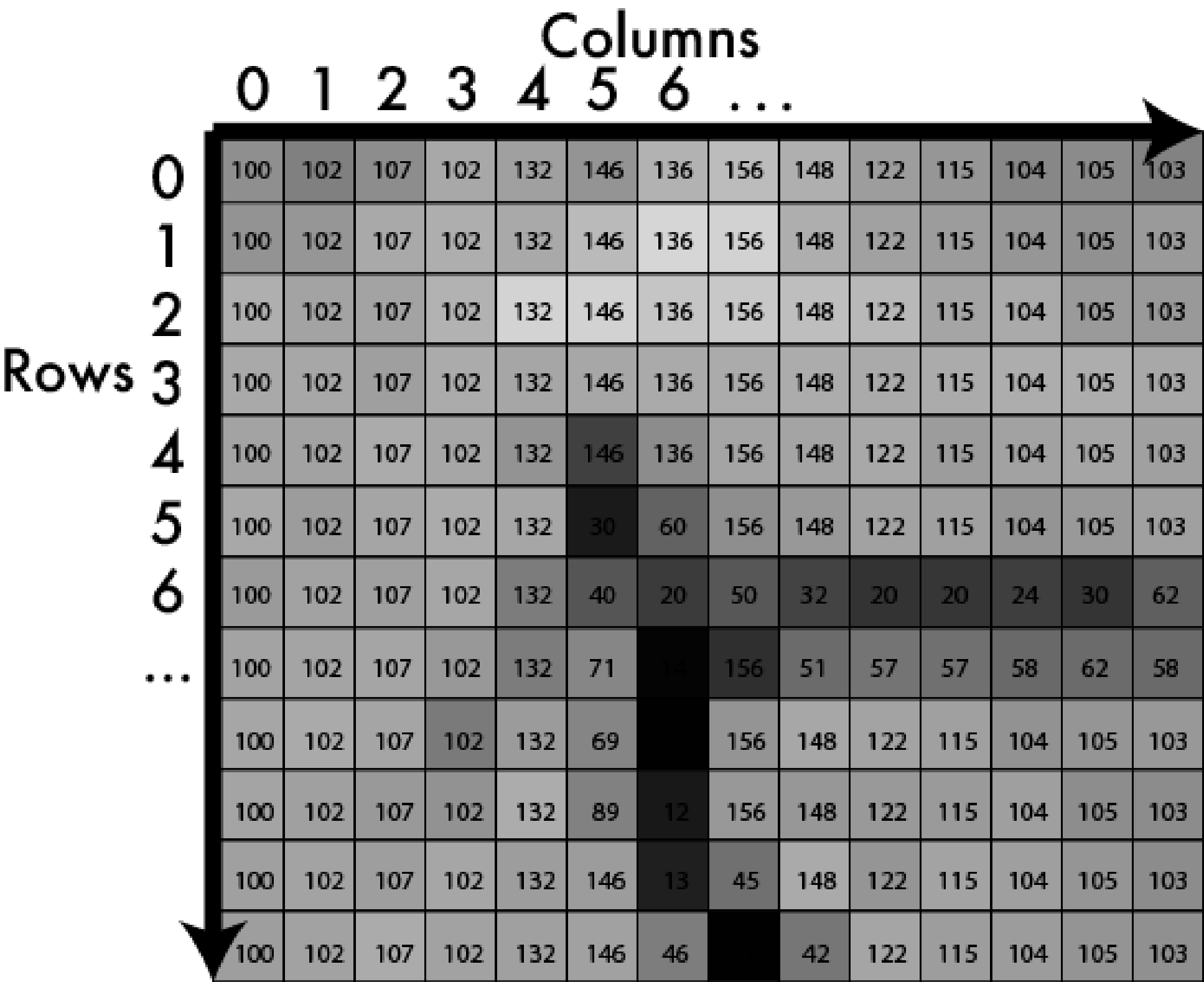
# CMOS sensors

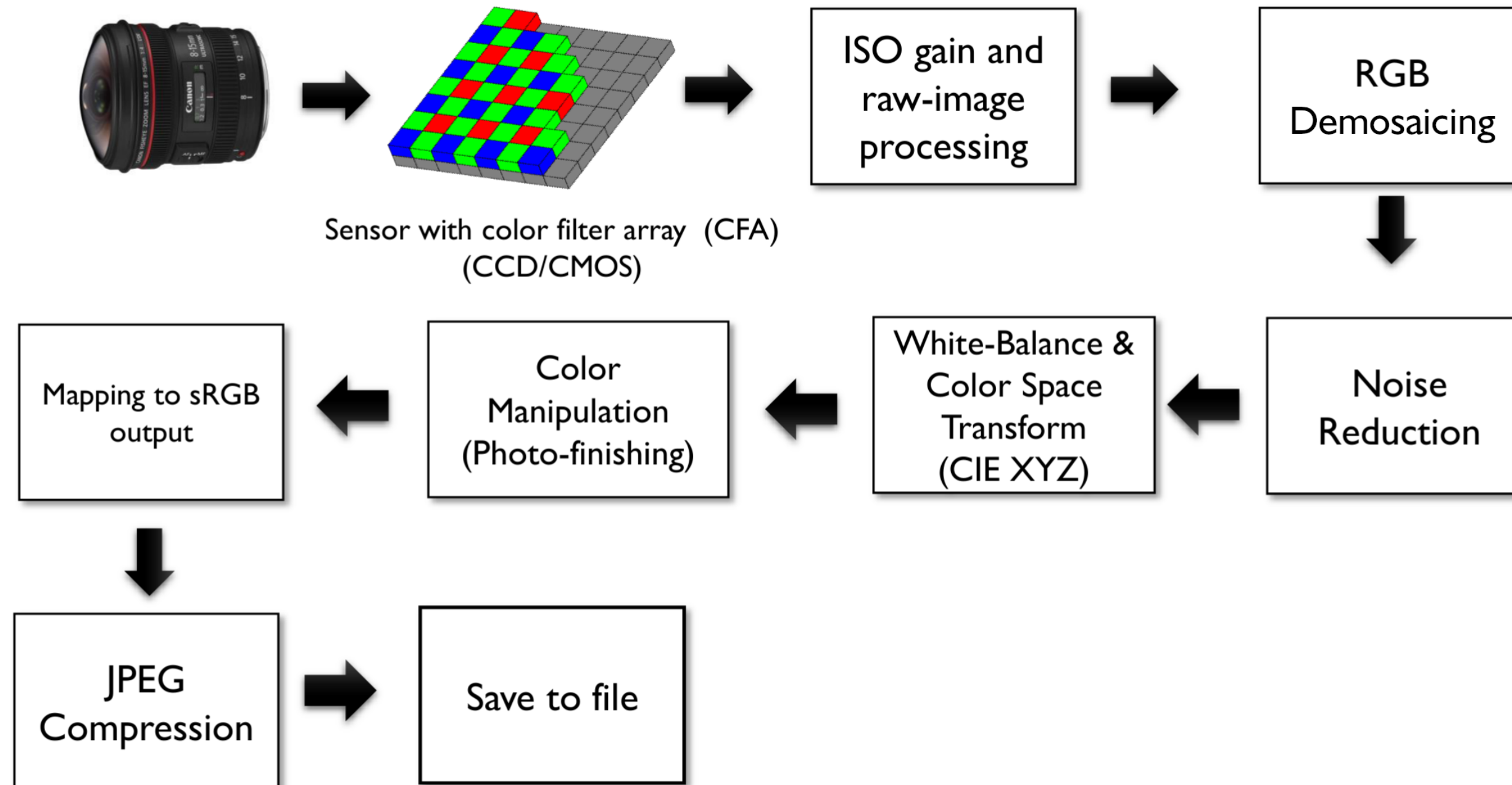


# An image is a matrix of light

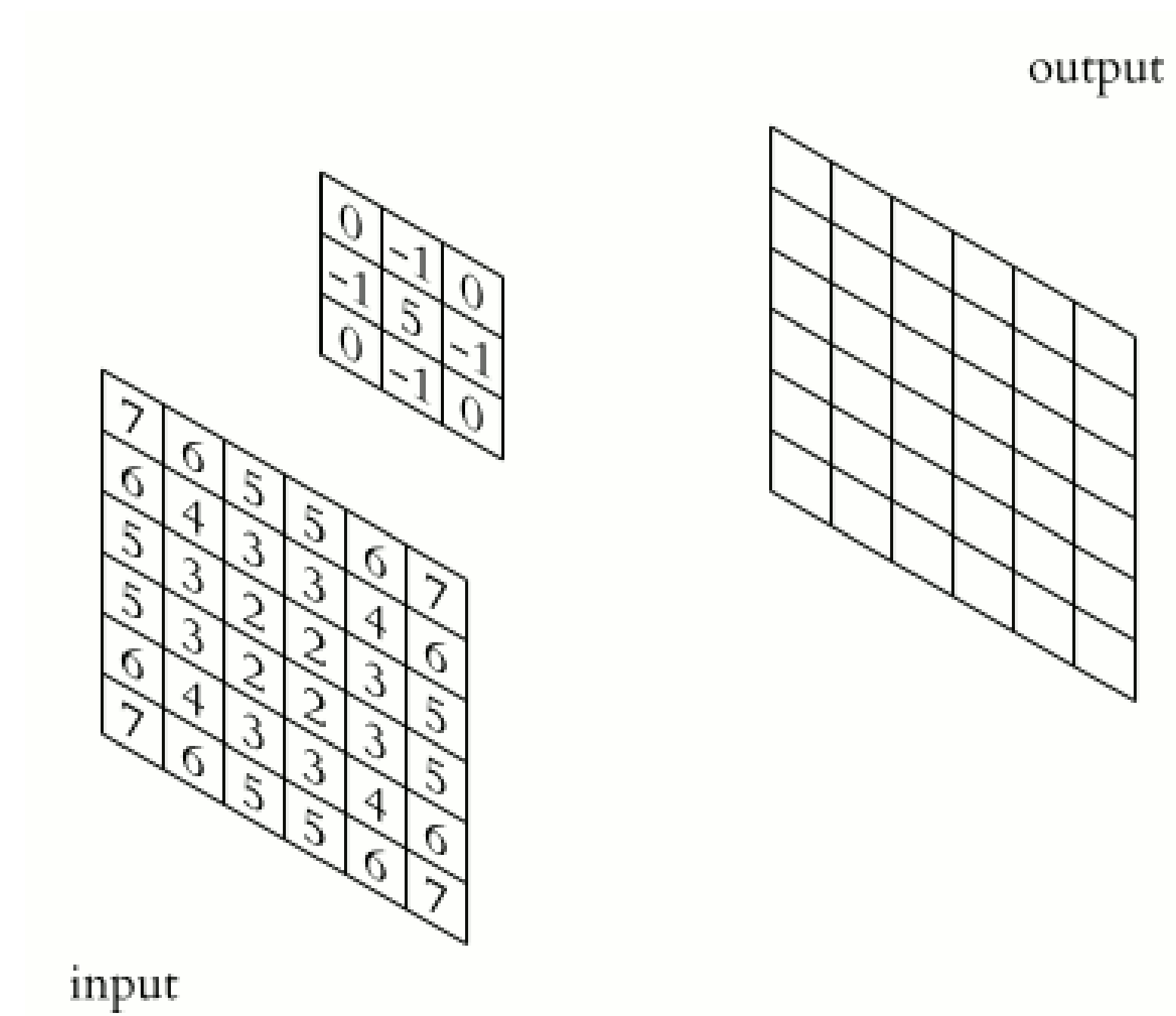


Values in matrix = how much light



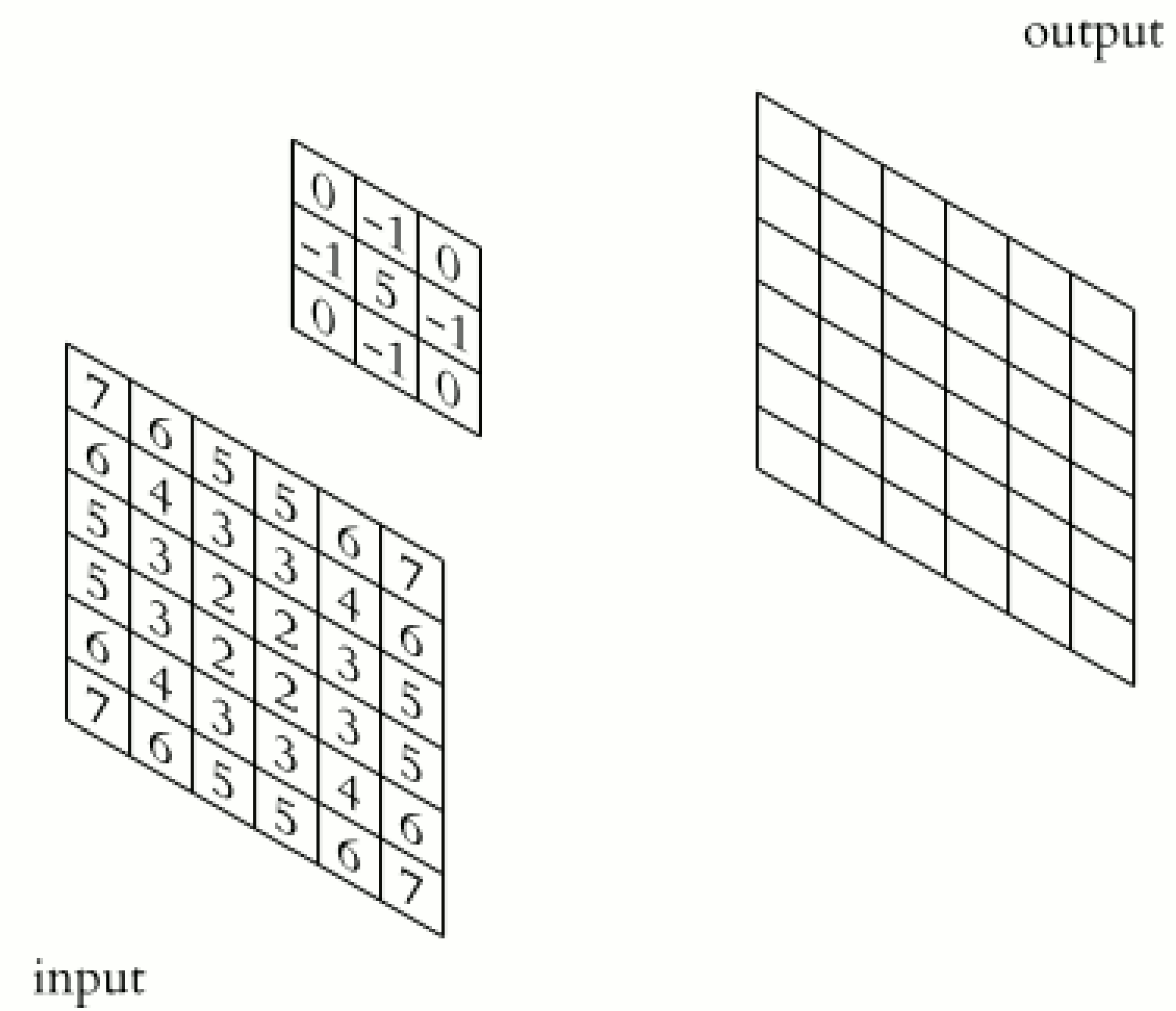


# Свертка

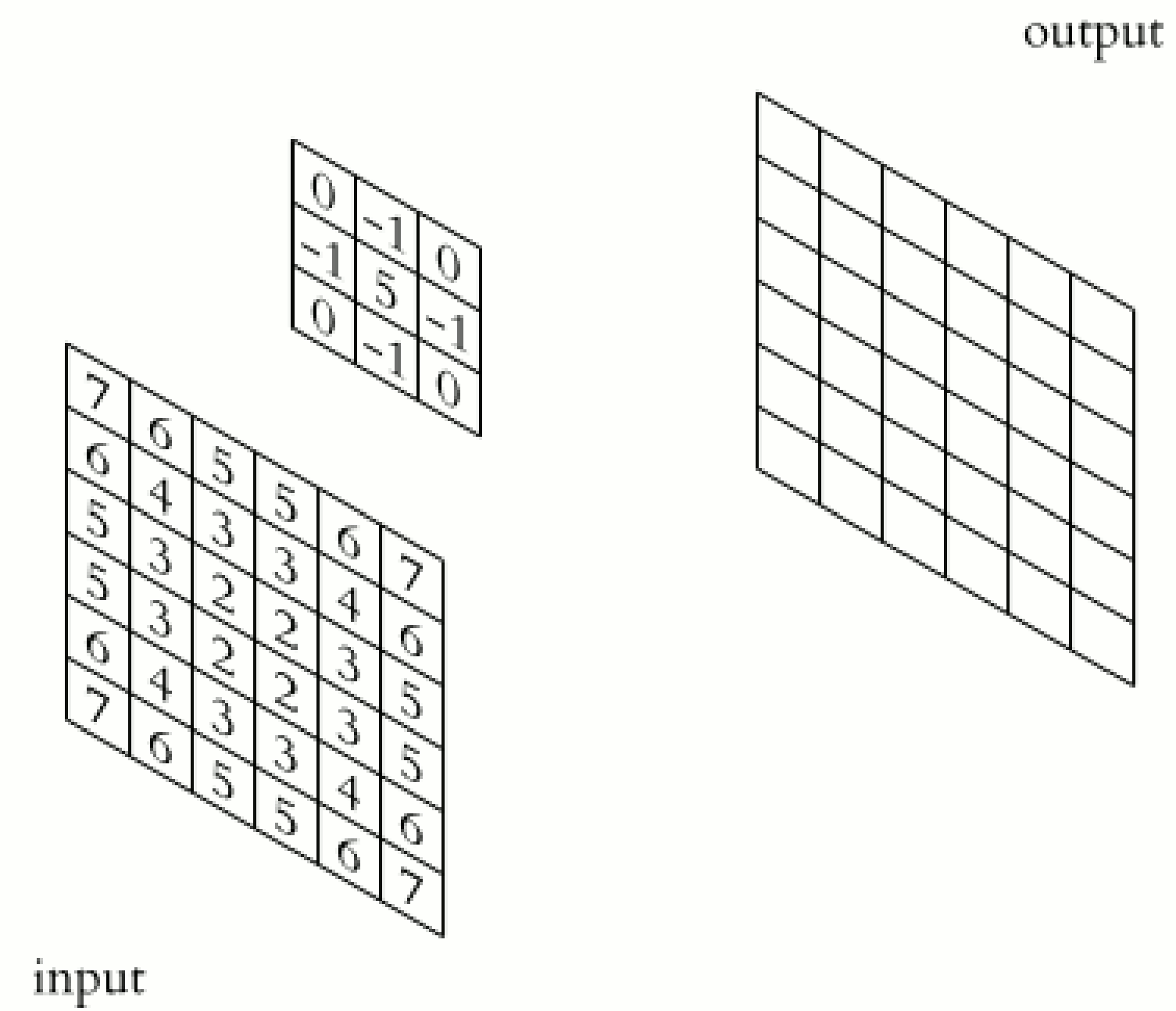


[https://upload.wikimedia.org/wikipedia/commons/1/19/2D\\_Convolution\\_Animation.gif](https://upload.wikimedia.org/wikipedia/commons/1/19/2D_Convolution_Animation.gif)

# Свертка

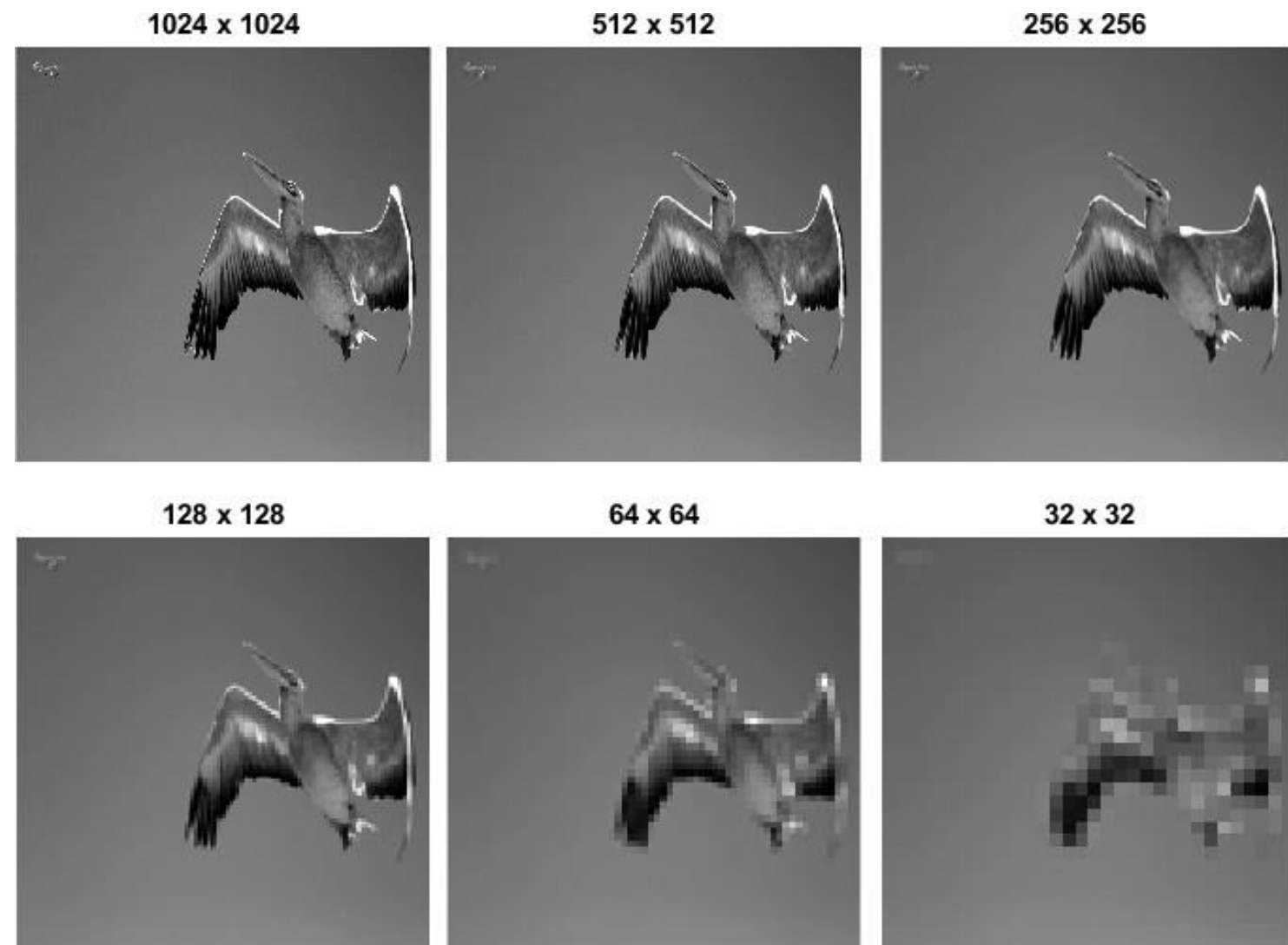


# Свертка

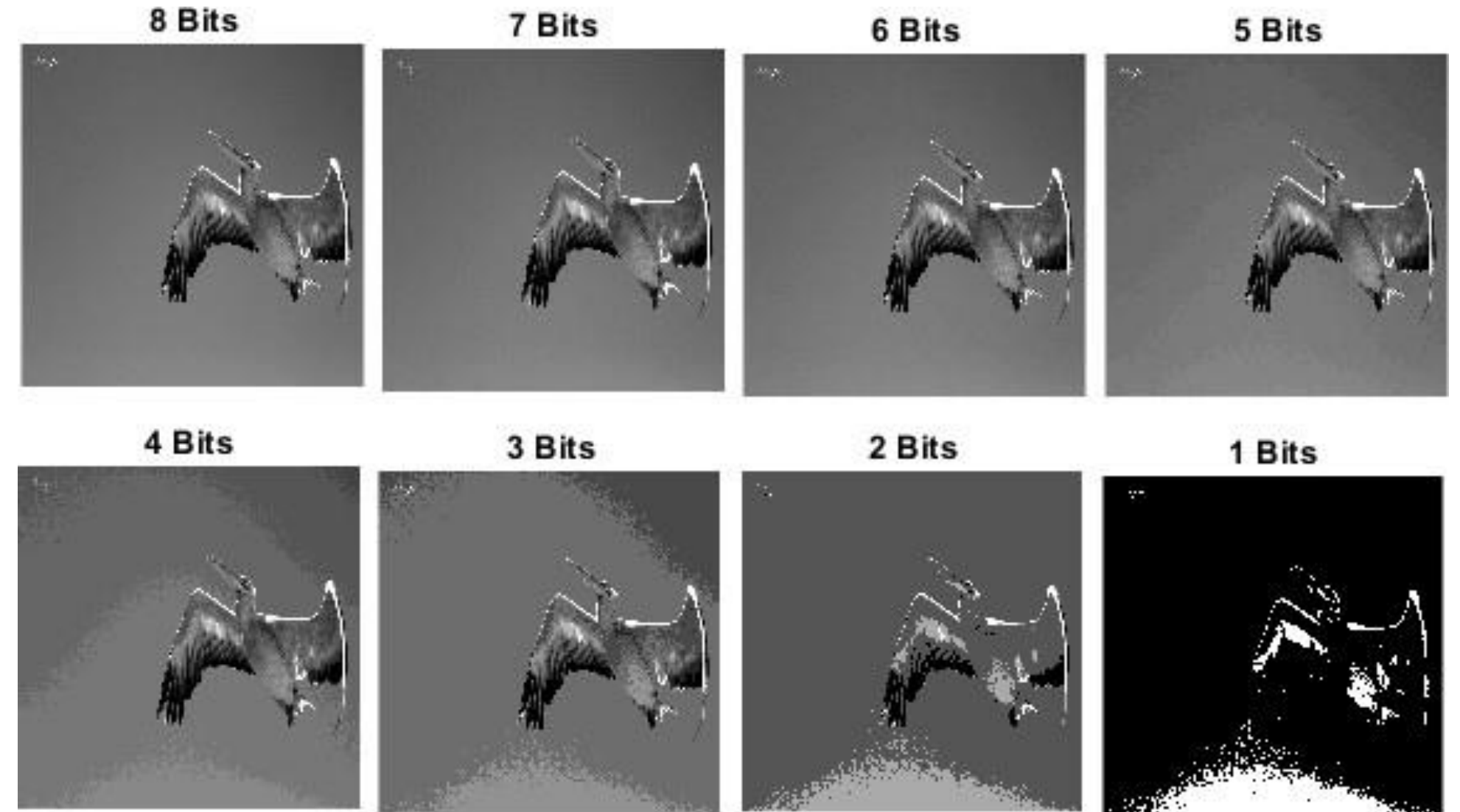


# Дискретизация и квантование

## Дискретизация



## Квантование



# Эквализация гистограммы

1	2	1	1	1
2	5	3	5	2
2	5	5	5	2
2	5	3	5	2
1	1	1	2	1

Input Image

Gray Level (rk) input Image (see ref image)	Frequency (nk)	$n_k/n$ (PDF= Probability Density Function)	CDF (Cumm Distributive Function)	Equalized level	Equalized level (round off)
0	0	0	0	0	0
1	8	0.32	0.32	2.24	2
2	8	0.32	0.64	4.48	4
3	2	0.08	0.72	5.04	5
4	0	0	0.72	5.04	5
5	7	0.28	1	7	7
6	0	0	1	7	7
7	0	0	1	7	7

1	2	1	1	1
2	5	3	5	2
2	5	5	5	2
2	5	3	5	2
1	1	1	2	1

Input Image

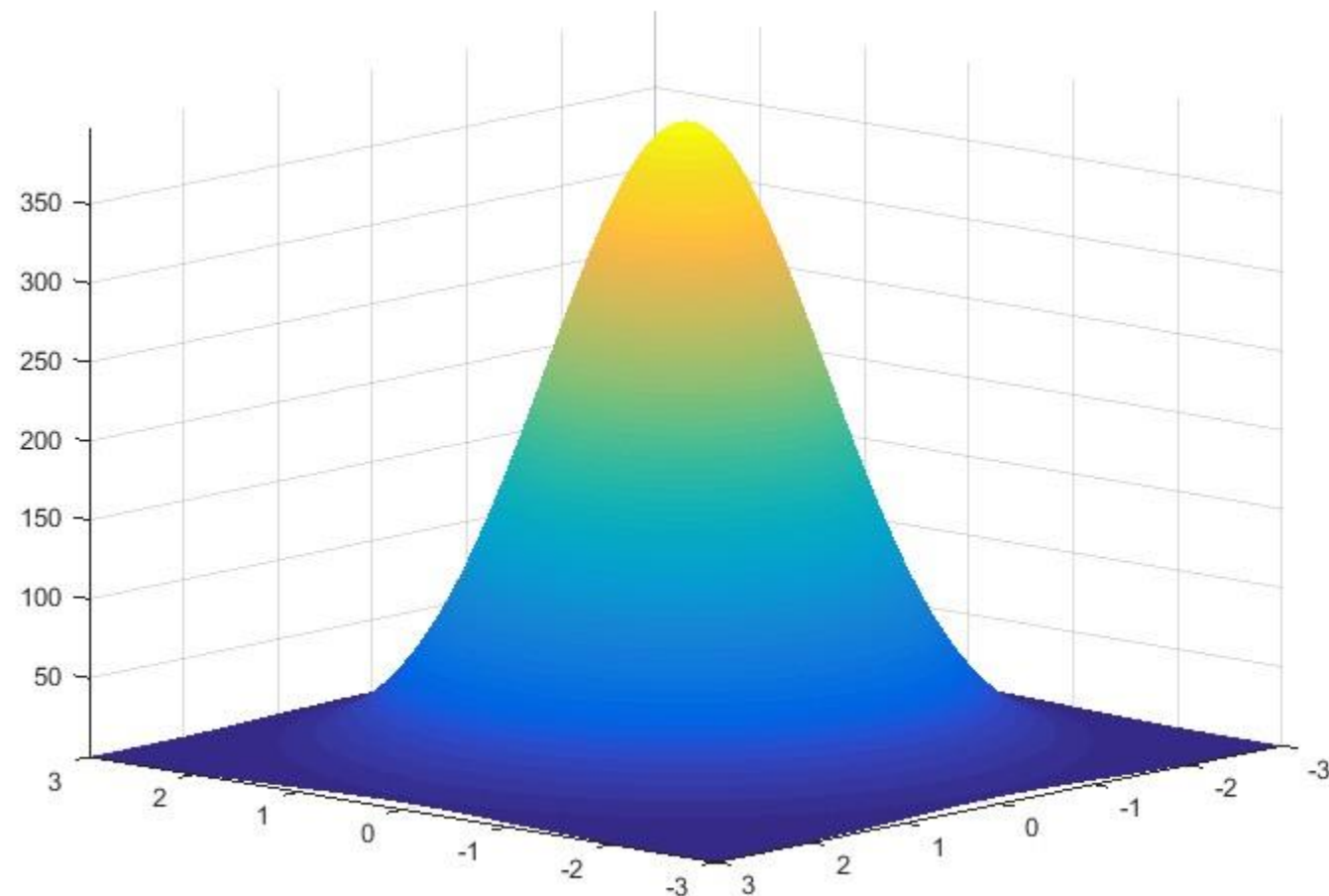


2	4	2	2	2
4	7	5	7	4
4	7	7	7	4
4	7	5	7	4
2	2	2	4	2

Output Image

# Фильтр Гаусса

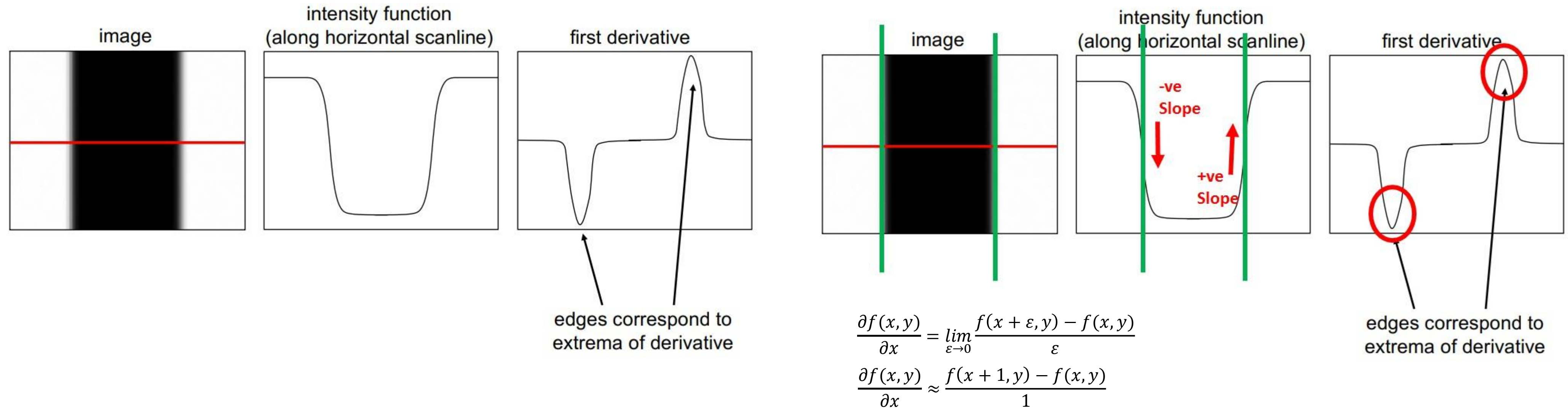
$$h(u, v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$



**1 / 16**

<b>1</b>	<b>2</b>	<b>1</b>
<b>2</b>	<b>4</b>	<b>2</b>
<b>1</b>	<b>2</b>	<b>1</b>

# Детекция границ



Исходное изображение



Градиент в направлении x



Градиент в направлении y

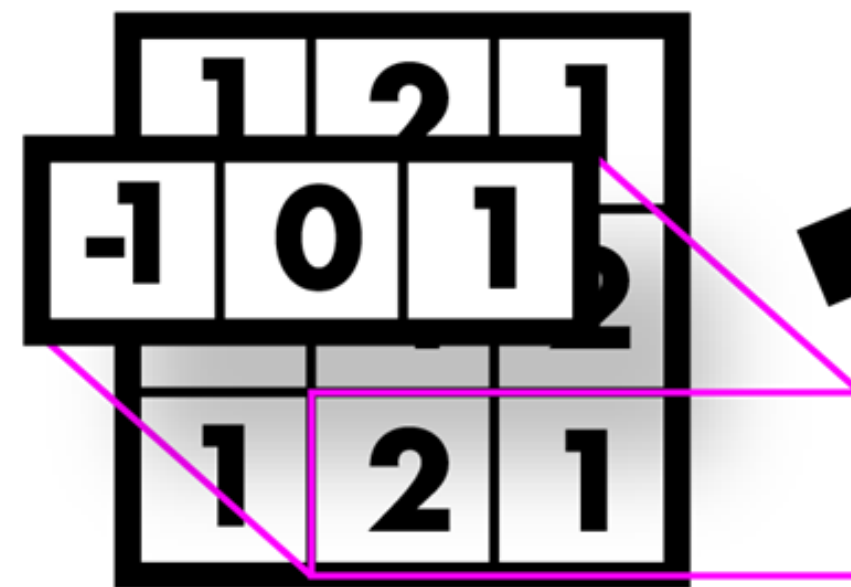
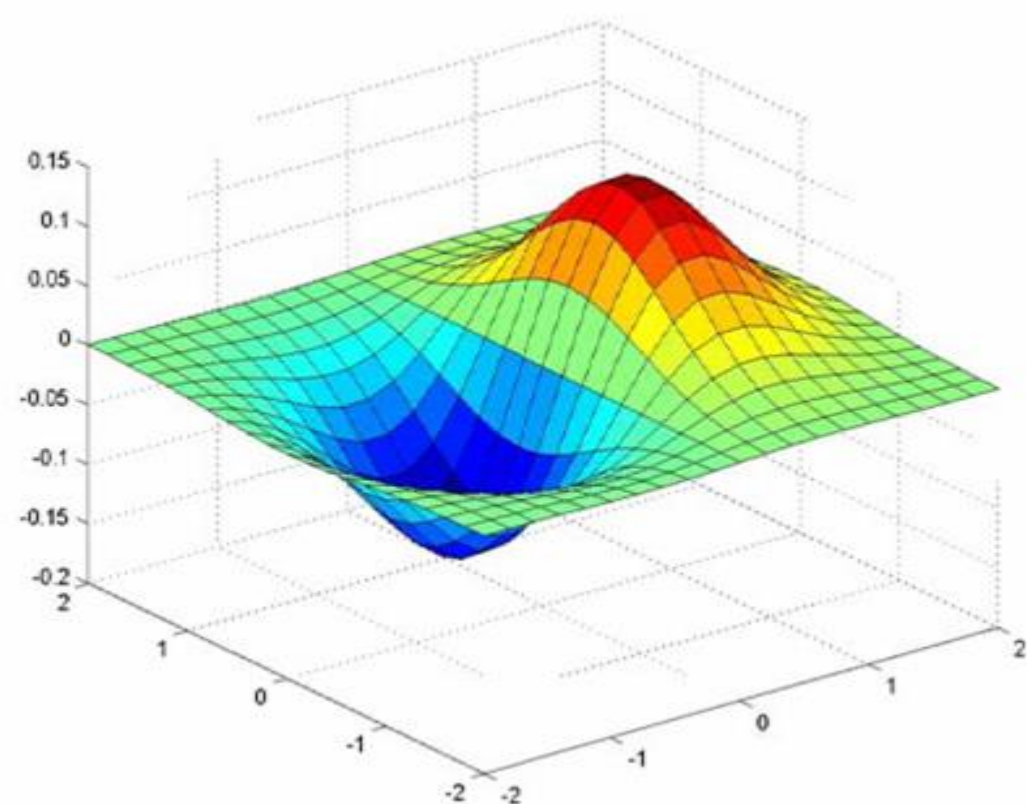


Амплитуда градиента



# Фильтр Собеля (сглаживание и производная)

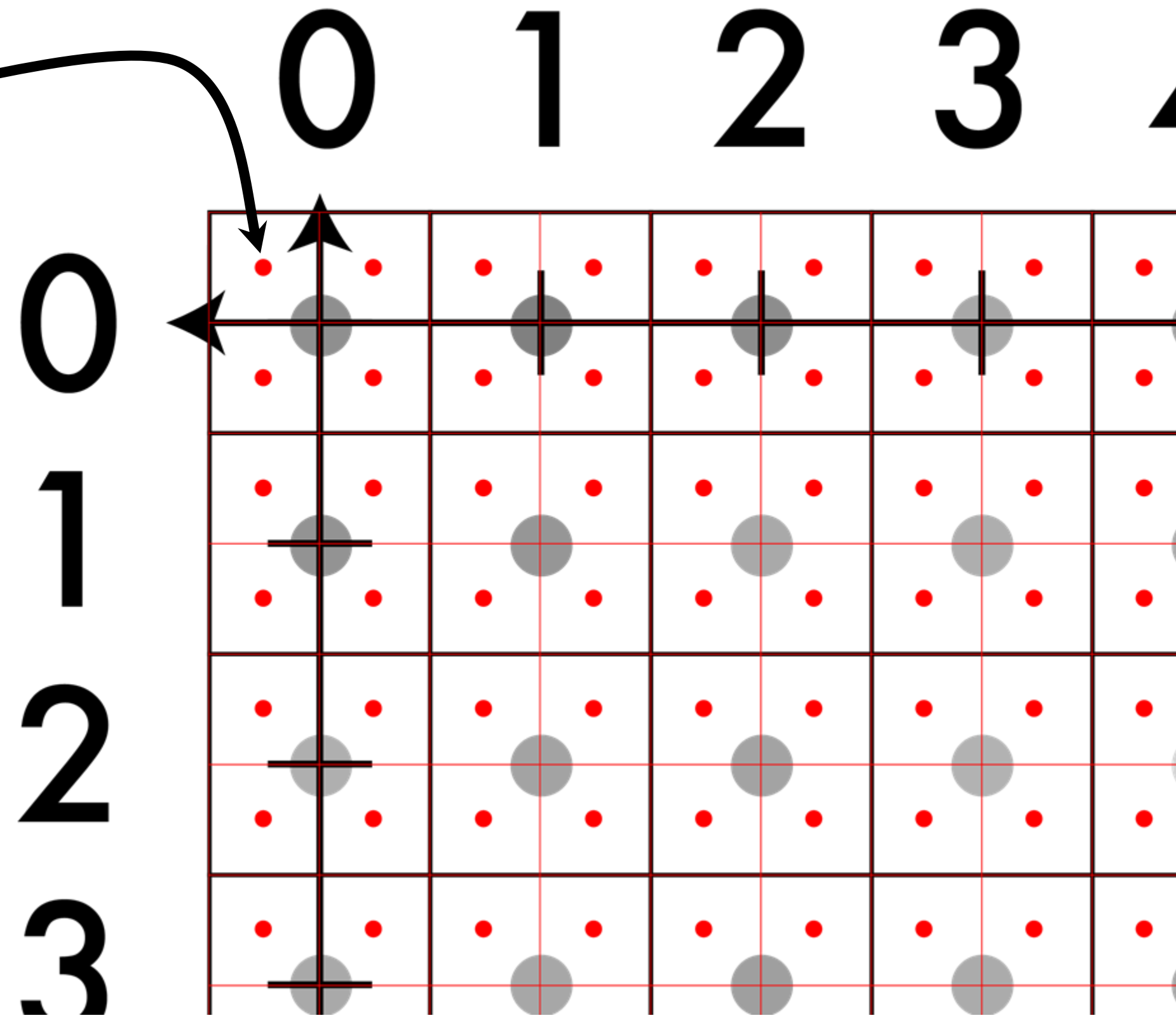
$$\frac{1}{2} \times \left( \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \right)$$



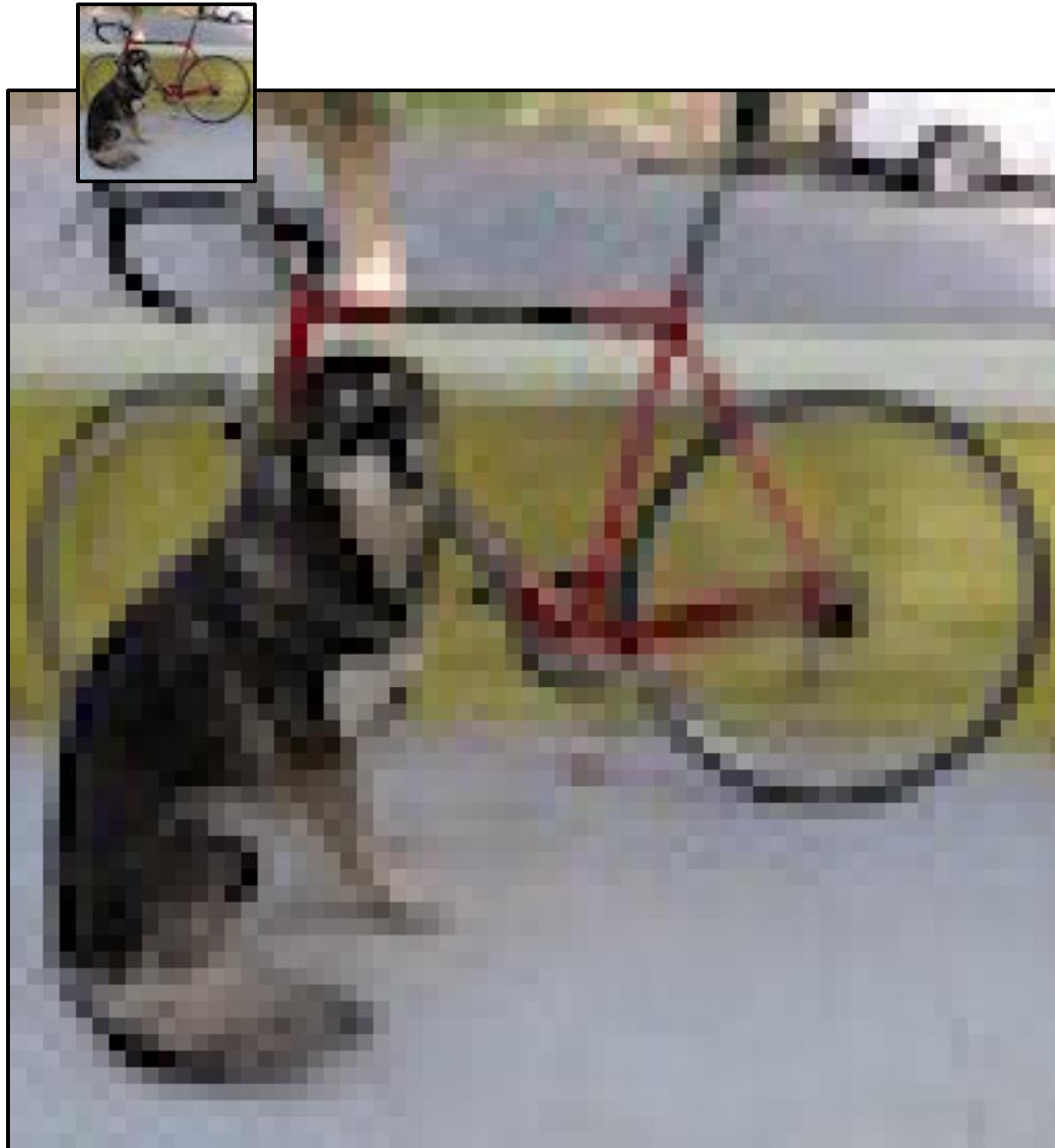
1	0	-1
2	0	-2
1	0	-1

# Проблема интерполяции

This point is:  
(-.25, -.25)



# Интерполяция по ближайшему соседу



# Билинейная интерполяция

$$q_1 = V_1 \cdot d_2 + V_2 \cdot d_1$$

$$q_2 = V_3 \cdot d_2 + V_4 \cdot d_1$$

$$q = q_1 \cdot d_4 + q_2 \cdot d_3$$

Equivalent:

$$q = q_1 \cdot d_4 + q_2 \cdot d_3$$

$$q = (V_1 \cdot d_2 + V_2 \cdot d_1) \cdot d_4 + (V_3 \cdot d_2 + V_4 \cdot d_1) \cdot d_3$$

(subst)

$$q = V_1 \cdot d_2 \cdot d_4 + V_2 \cdot d_1 \cdot d_4 + V_3 \cdot d_2 \cdot d_3 + V_4 \cdot d_1 \cdot d_3$$

(distribution)

Recall:

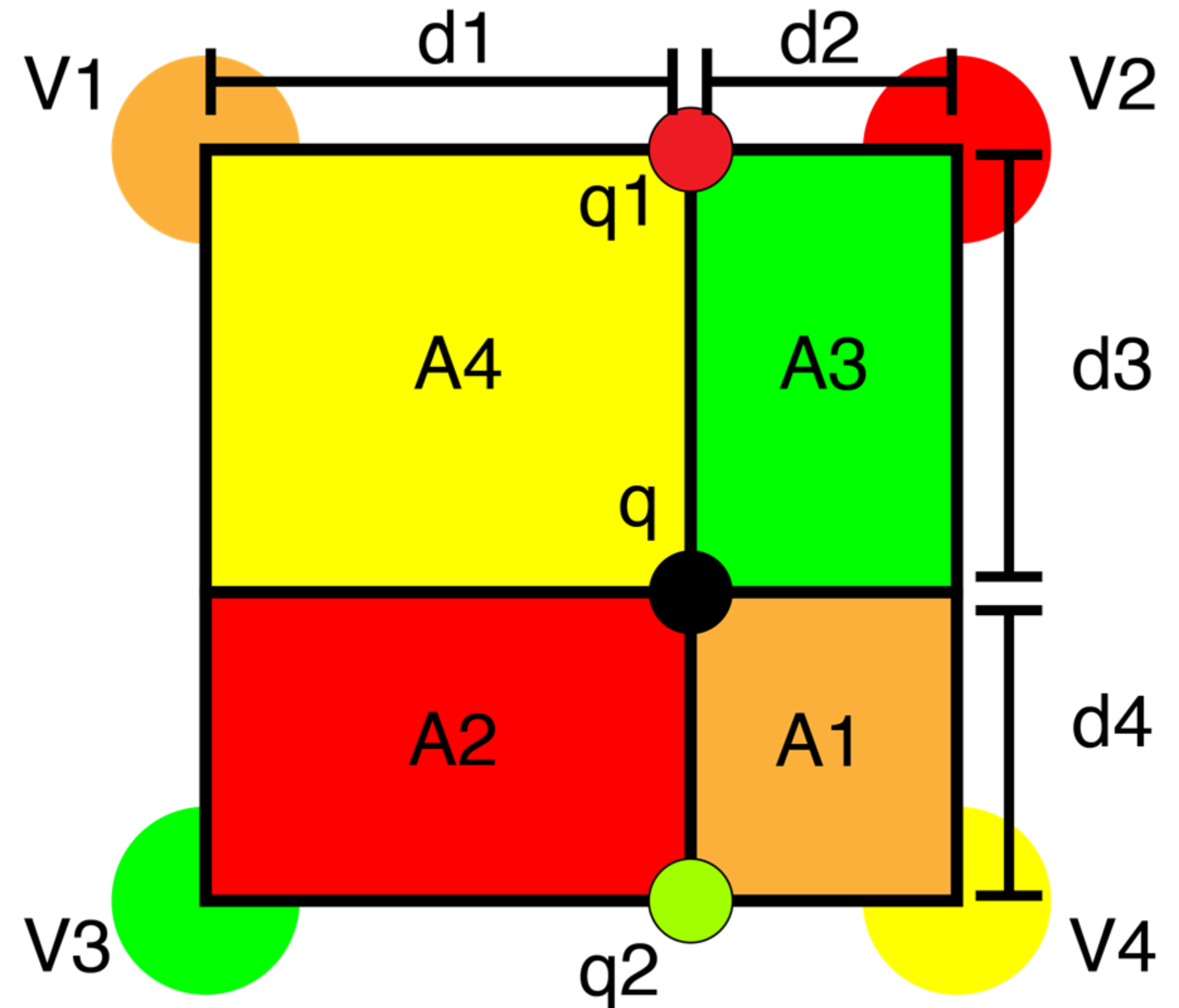
$$A_1 = d_2 \cdot d_4$$

$$A_2 = d_1 \cdot d_4$$

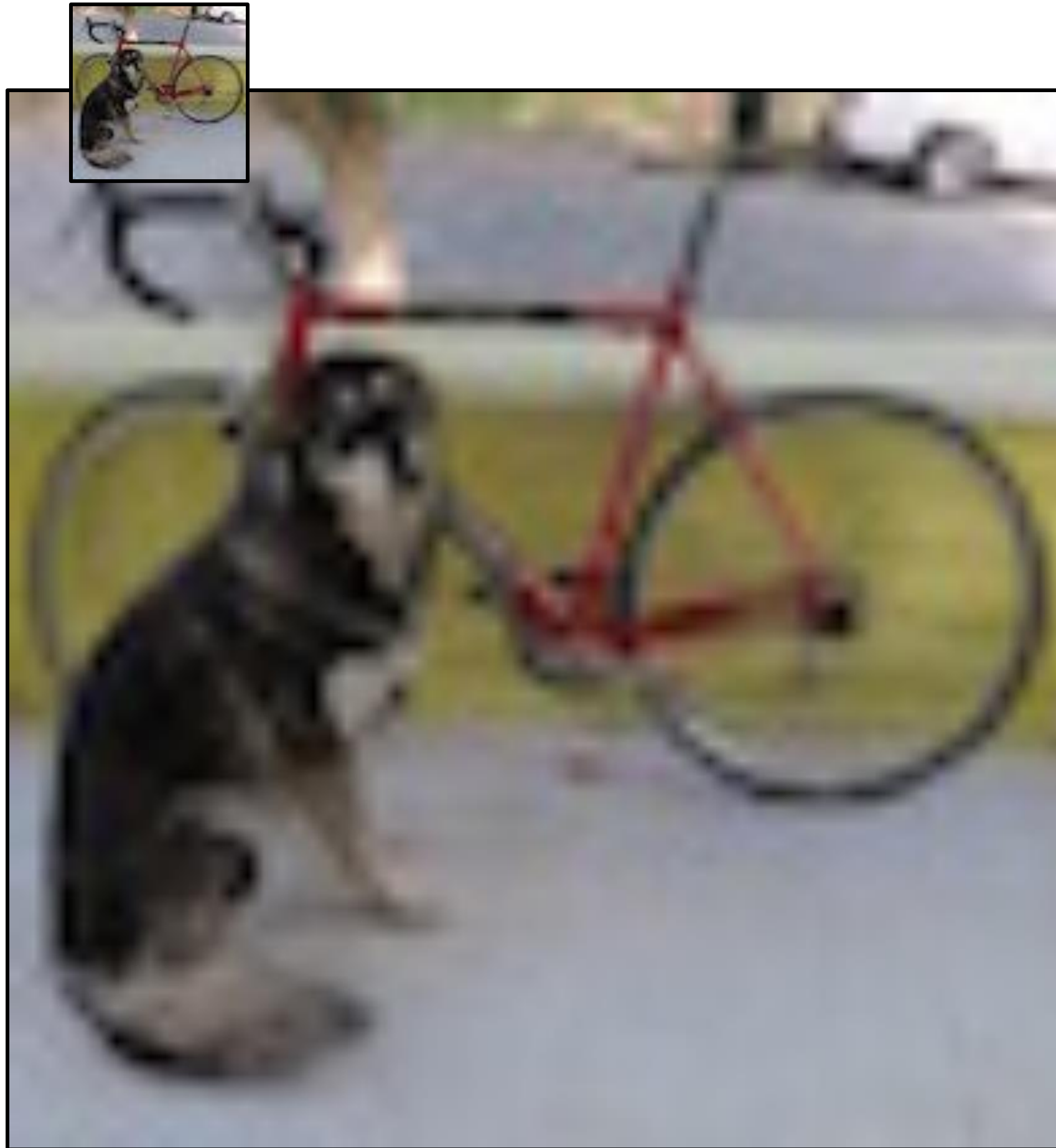
$$A_3 = d_2 \cdot d_3$$

$$A_4 = d_1 \cdot d_3$$

$$q = V_1 \cdot A_1 + V_2 \cdot A_2 + V_3 \cdot A_3 + V_4 \cdot A_4$$

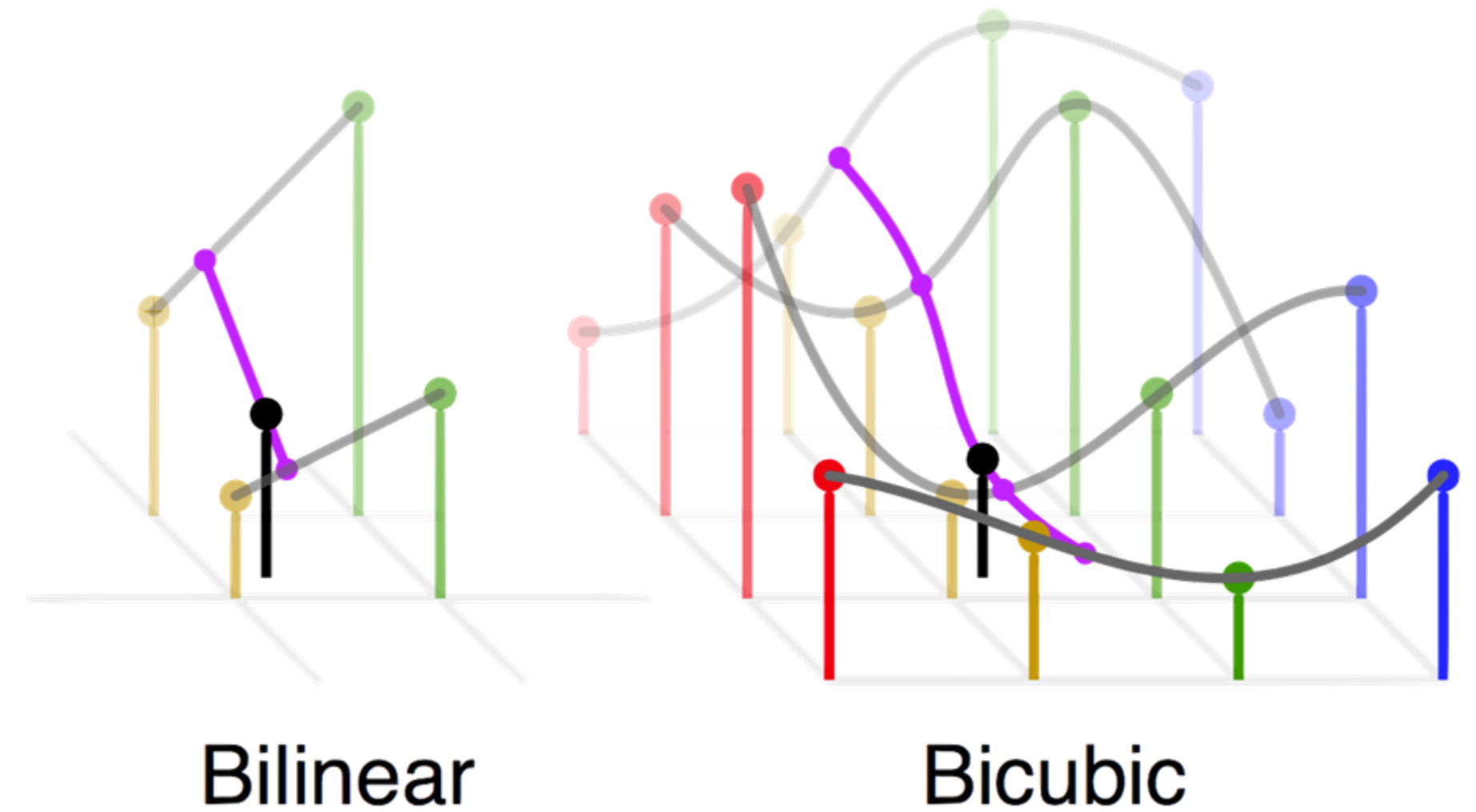


# Билинейная интерполяция

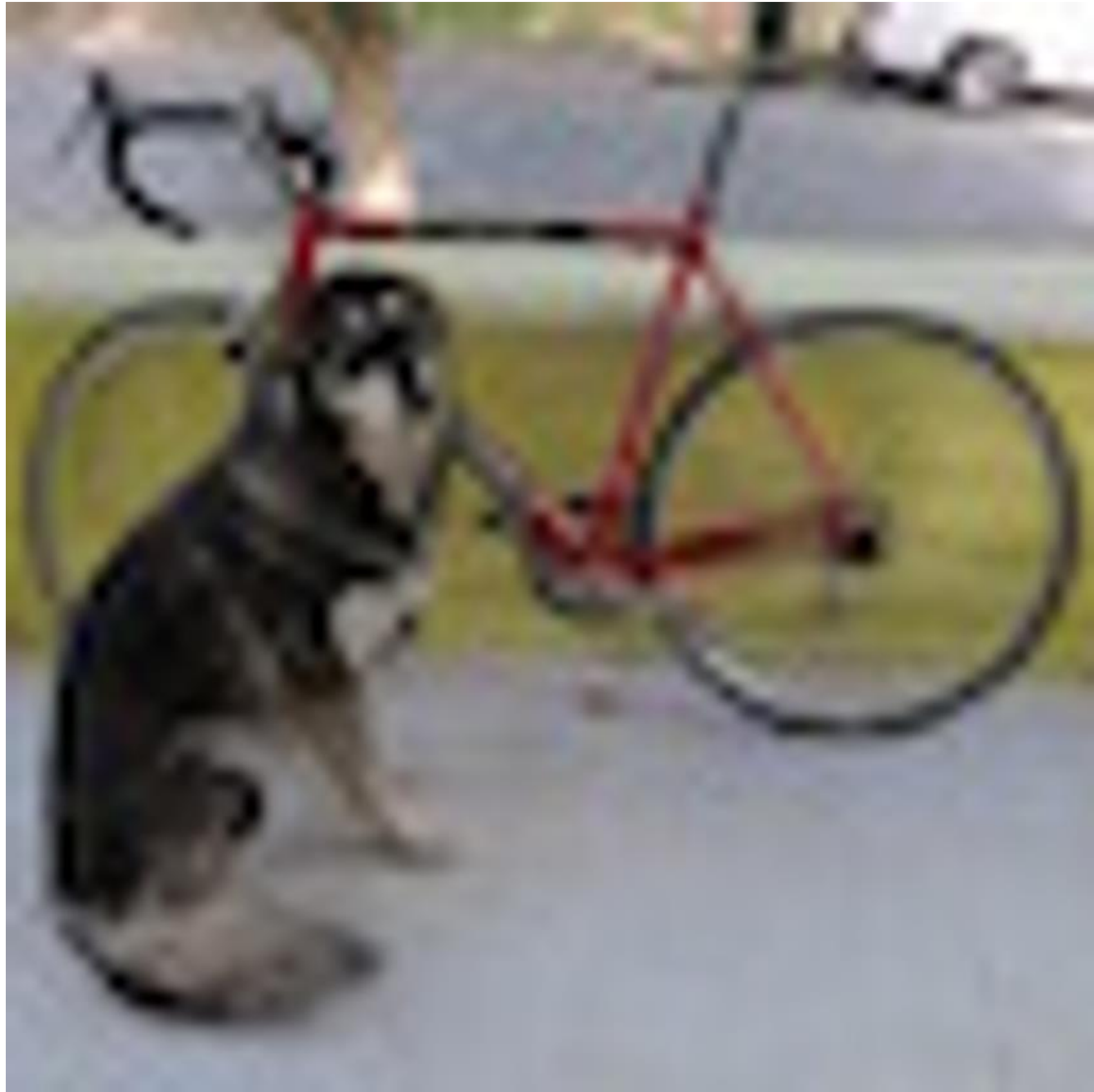


# Бикубическая интерполяция

- A cubic interpolation of 4 cubic interpolations
- Smoother than bilinear, no “star”
- 16 nearest neighbors
- Fit 3rd order poly:  
$$f(x) = a + bx + cx^2 + dx^3$$
- Interpolate along axis
- Fit another poly to interpolated values



# Бикубическая vs Билинейная



# Полезные ресурсы

- <https://pjreddie.com/courses/computer-vision/>
- <https://sites.google.com/site/drkhanrizwan17/computer-vision-course>
- [https://github.com/ml-dafe/cv\\_mipt\\_major](https://github.com/ml-dafe/cv_mipt_major)
- <https://code.mipt.ru/courses-public/cv/public/-/blob/fall-2023/README.md>