# Tutorial 2

Jiayu Yang

# Outline

- Image Filtering
  - Correlation and convolution filtering
  - Padding ⇨ Lab 1 Task 5
  - Type of kernels
  - Bilateral filtering
- Image Warping
  - Homogeneous Coordinate
  - Type of transformations
  - Forward warping and splatting
  - Inverse warping and interpolation ⇨ Lab 1 Task 6
  - Type of interpolation methods
    - Nearest-neighbor interpolation
    - Bilinear interpolation
    - Bicubic interpolation

*f(x)*: Original image
*g(x)*: New image
*h*: a mapping from image *f(x)* to new image *g(x)*

**range**: pixel intensity value, 0 (black) to 255 (white).
**domain**: pixel location (x,y).

Image Filtering

Change **range** of image *f*

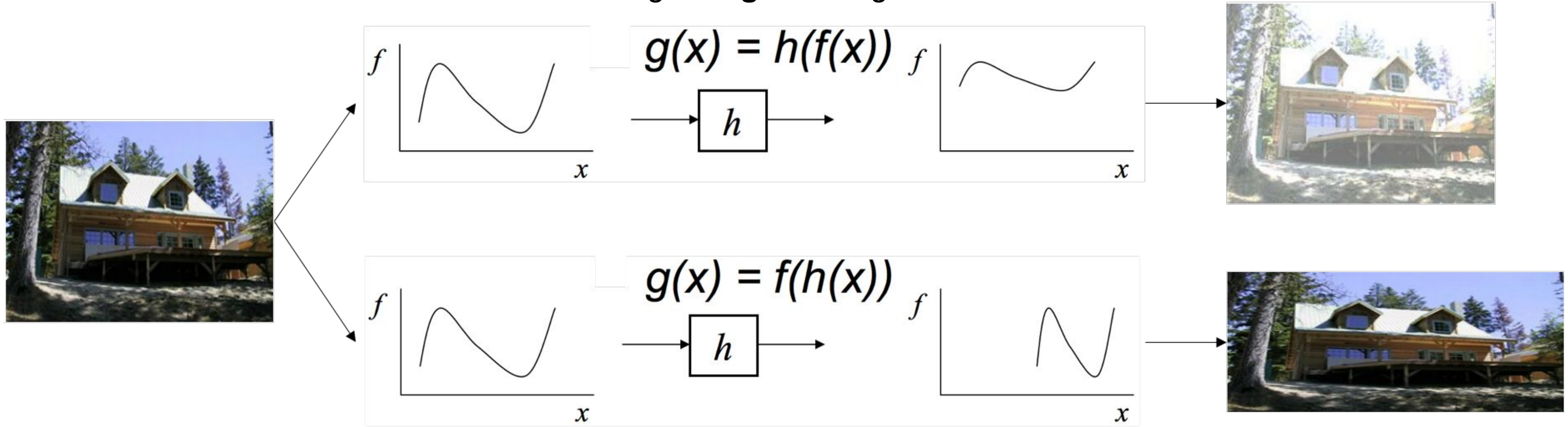



Image Warping

Change spatial **domain** of image *f*

*f(x)*: Original image
*g(x)*: New image
*h*: a mapping from image *f(x)* to new image *g(x)*

**range**: pixel intensity value, 0 (black) to 255 (white).
**domain**: pixel location (x,y).

Image Filtering

Change **range** of image *f*

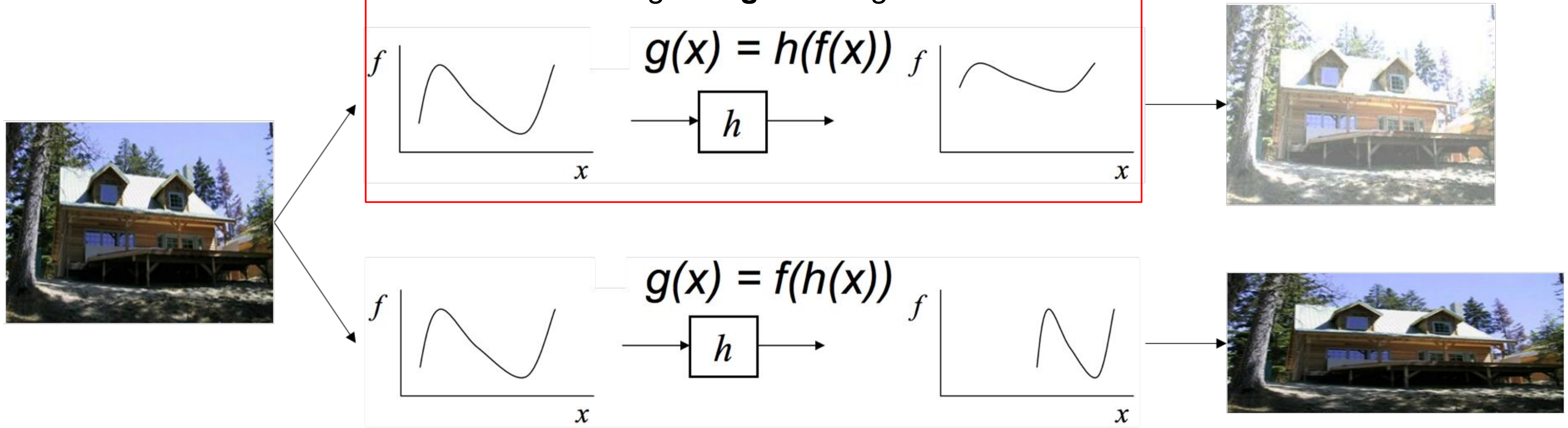



Image Warping

Change spatial **domain** of image *f*

# Image Filtering

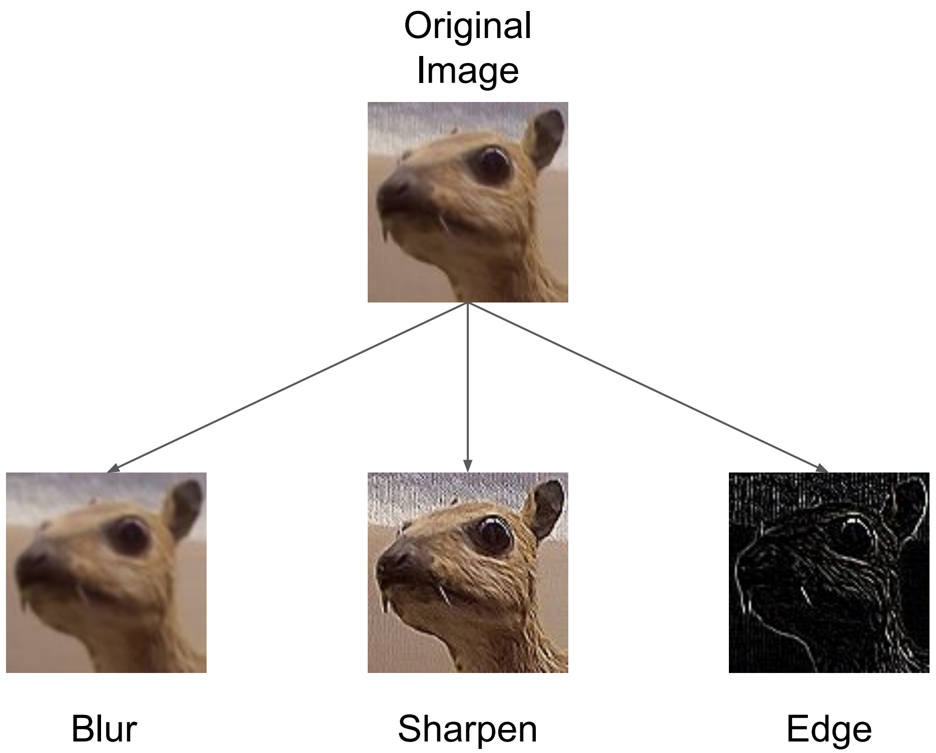
Change **range** of image *f*

$$g(x) = h(f(x))$$

**Enhance images**
- De-noise, increase contrast, etc.

**Extract information from images**
- Texture, edges, distinctive points, etc.

**Detect pattern**
- Template matching



## Image Filtering

$$g(x) = h(f(x))$$

**Correlation filtering**

$$G = H \otimes F$$

$$\Downarrow$$

$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v] F[i+u, j+v]$$

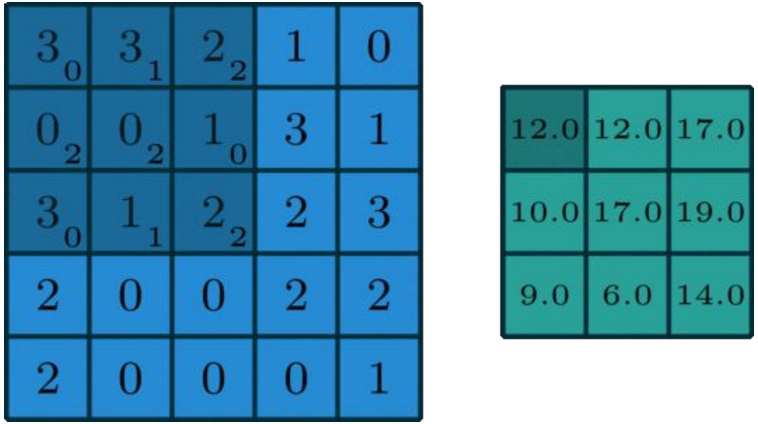
**Convolution filtering**

$$G = H \star F$$

$$\Downarrow$$

$$G[x,y] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v] F[x-u, y-v]$$

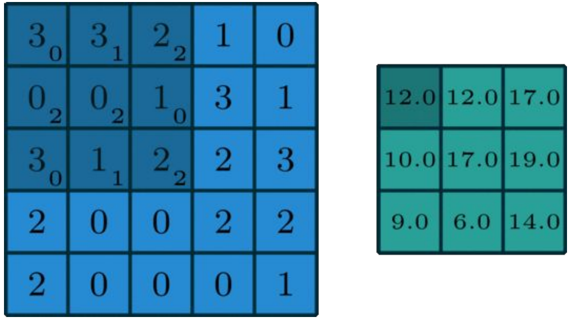
Slide kernel $H$ on image $F$




- The kernel is rotated by 180 degrees **or** flipped on both horizontal and vertical direction.
- Identical when the kernel is symmetrical.

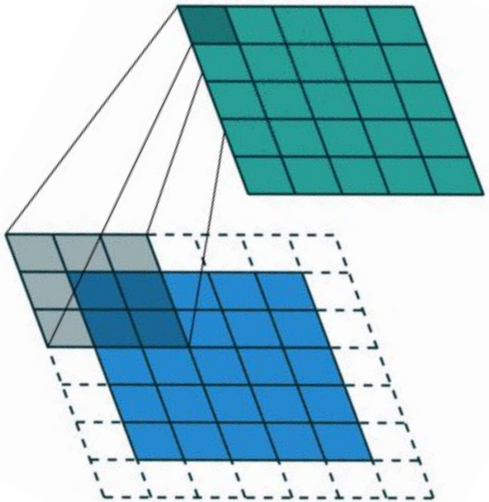
Image Filtering (Take 3x3 filter as example)

Slide kernel on image




Input:
H x W → Output:
H-2 x W-2

Padding



Input:
H+2 x W+2 → Output:
H x W

# Image Filtering

## Types of pedding

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 2 | 3 | 4 | 5 | 0 | 0 |
| 0 | 0 | 6 | 7 | 8 | 9 | 10 | 0 | 0 |
| 0 | 0 | 11 | 12 | 13 | 14 | 15 | 0 | 0 |
| 0 | 0 | 16 | 17 | 18 | 19 | 20 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Constant (zero)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 13 | 12 | 11 | 12 | 13 | 14 | 15 | 14 | 13 |
| 8 | 7 | 6 | 7 | 8 | 9 | 10 | 9 | 8 |
| 3 | 2 | 1 | 2 | 3 | 4 | 5 | 4 | 3 |
| 8 | 7 | 6 | 7 | 8 | 9 | 10 | 9 | 8 |
| 13 | 12 | 11 | 12 | 13 | 14 | 15 | 14 | 13 |
| 18 | 17 | 16 | 17 | 18 | 19 | 20 | 19 | 18 |
| 13 | 12 | 11 | 12 | 13 | 14 | 15 | 14 | 13 |
| 8 | 7 | 6 | 7 | 8 | 9 | 10 | 9 | 8 |

Mirror/Symmetric

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 2 | 3 | 4 | 5 | 5 | 5 |
| 1 | 1 | 1 | 2 | 3 | 4 | 5 | 5 | 5 |
| 1 | 1 | 1 | 2 | 3 | 4 | 5 | 5 | 5 |
| 6 | 6 | 6 | 7 | 8 | 9 | 10 | 10 | 10 |
| 11 | 11 | 11 | 12 | 13 | 14 | 15 | 15 | 15 |
| 16 | 16 | 16 | 17 | 18 | 19 | 20 | 20 | 20 |
| 16 | 16 | 16 | 17 | 18 | 19 | 20 | 20 | 20 |
| 16 | 16 | 16 | 17 | 18 | 19 | 20 | 20 | 20 |

Replicate

Original Image

Filtered Image with Black Border

Filtered Image with Border Replication

Different kinds of kernels

$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v]F[i+u,j+v]$$

$*$

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Laplacian

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Sharpen

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Mean blur

$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Gaussian 3x3

$$\frac{1}{256}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

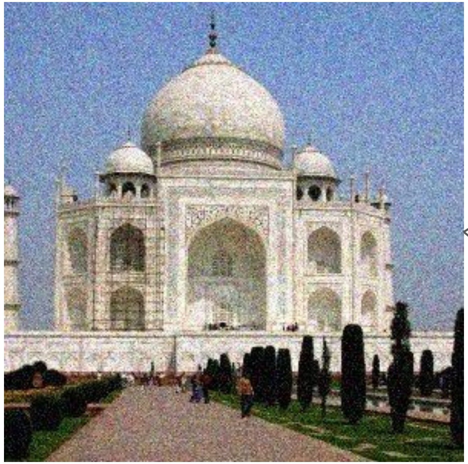
Gaussian 5x5

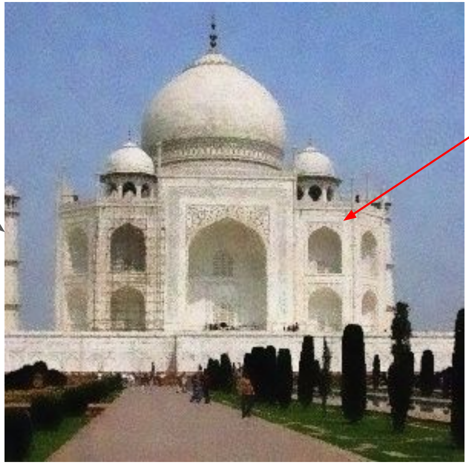
Check lecture04 for detail of each kernels

Bilateral filtering

Lab 1 Task 5
Challenge task

Gaussian





Edge preserved

Bilateral filtering

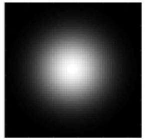
https://www.geeksforgeeks.org/python-bilateral-filtering/

# Bilateral filtering

## Fixed kernel

$$GB[I]_p = \sum_{q \epsilon S} G_\sigma(||p-q||) I_q$$

Normalized Gaussian Function



p: center pixel
S: 3x3 neighboring pixels
||p-q||: spatial distance

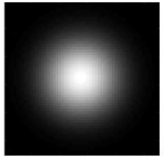
---

## Bilateral filtering

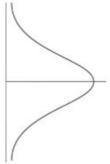
$$BF[I]_p = \frac{1}{W_p} \sum_{q \epsilon S} G_{\sigma_s}(||p-q||) G_{\sigma_r}(|I_p - I_q|) I_q$$

Normalization Factor

Space Weight



Range Weight



Lab 1 Task 5
Challenge task

Read and implement:

Tomasi, C; Manduchi, R (1998). Bilateral filtering for gray and color images. Sixth International Conference on Computer Vision. Bombay. pp. 839– 846.

https://www.geeksforgeeks.org/python-bilateral-filtering/

*f(x)*: Original image
*g(x)*: New image
*h*: a mapping from image *f(x)* to new image *g(x)*

**range**: pixel intensity value, 0 (black) to 255 (white).
**domain**: pixel location (x,y).

Image Filtering

Change **range** of image *f*

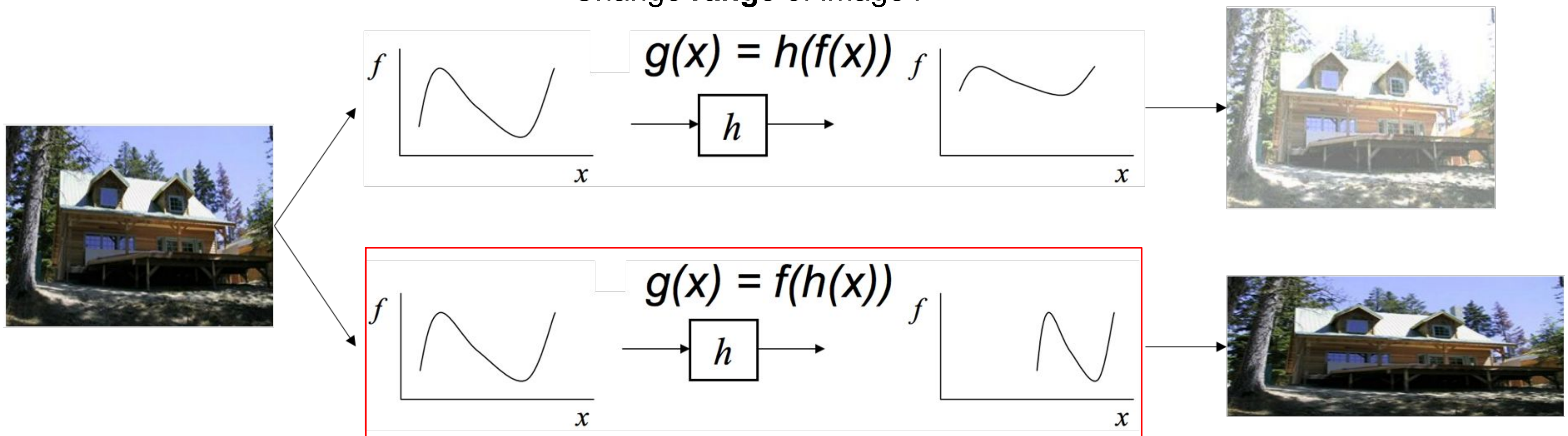



Image Warping

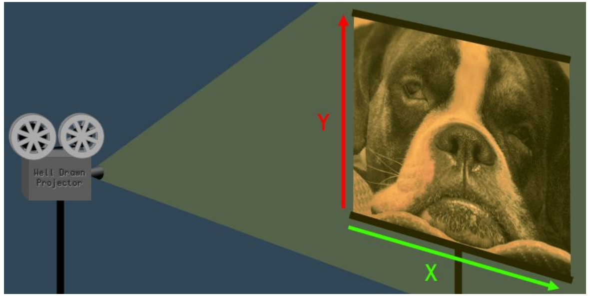
Change spatial **domain** of image *f*

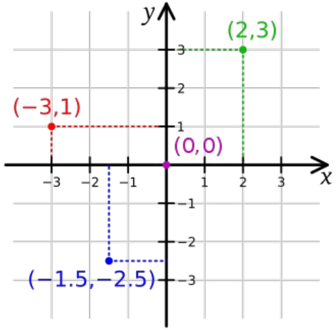
# Homogeneous Coordinate

## Cartesian coordinate

representing N-dimensional coordinates with **N** numbers

2D Point: $\begin{bmatrix} x \\ y \end{bmatrix}$



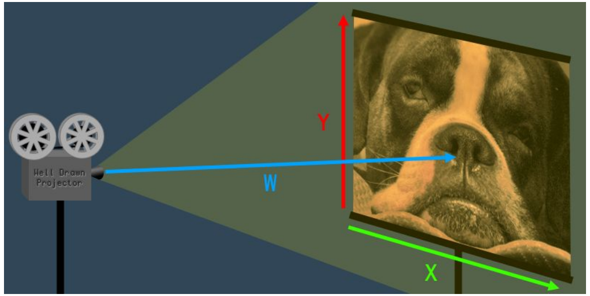



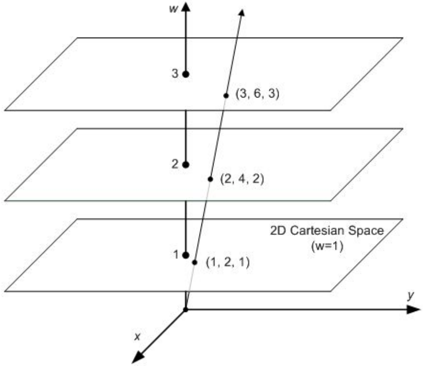

## Homogeneous Coordinate

representing N-dimensional coordinates with **N+1** numbers

**2D** Point: $\begin{bmatrix} x \\ y \\ w \end{bmatrix}$

Homogeneous coordinates have an extra dimension called $W$, which **scales** the $X$ and $Y$ dimensions



# Homogeneous Coordinate

Convert between cartesian coordinate and homogeneous coordinate

An example for 3D points:

**regular 3D point to homogeneous:**

$$\begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} \longrightarrow \begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix}$$

**homogeneous point to regular 3D:**

$$\begin{pmatrix} p_x \\ p_y \\ p_z \\ p_w \end{pmatrix} \longrightarrow \begin{pmatrix} p_x/p_w \\ p_y/p_w \\ p_z/p_w \end{pmatrix}$$



2D points follow the same way.

# Image warping

$$g(x) = f(h(x))$$

## Cartesian coordinate

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

Translate

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Scale

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Rotate

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & sh_x \\ sh_y & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Shear

---

## Homogeneous Coordinate

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Translate

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Scale

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Rotate

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & sh_x & 0 \\ sh_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Shear

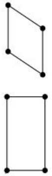
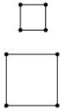
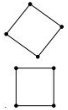
Lab 1 Task 6
1. Implement your own function my_translation()

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} & 3 \times 3 & \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

# Hierarchy of transformations

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} & & \\ & 3 \text{ x } 3 & \\ & & \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Detail in book
“Multi-view Geometry in Computer Vision” p44

| Group | Matrix | Distortion | Invariant properties |
|---|---|---|---|
| Projective<br>8 dof | $\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$ |  | Concurrency, collinearity, **order of contact**: intersection (1 pt contact); tangency (2 pt contact); inflections (3 pt contact with line); tangent discontinuities and cusps. cross ratio (ratio of ratio of lengths). |
| Affine<br>6 dof | $\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$ |  | Parallelism, ratio of areas, ratio of lengths on collinear or parallel lines (e.g. midpoints), linear combinations of vectors (e.g. centroids). The line at infinity, $\mathbf{l}_\infty$. |
| Similarity<br>4 dof | $\begin{bmatrix} sr_{11} & sr_{12} & t_x \\ sr_{21} & sr_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$ |  | Ratio of lengths, angle. The circular points, **I**, **J** (see section 2.7.3). |
| Euclidean<br>3 dof | $\begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$ |  | Length, area |

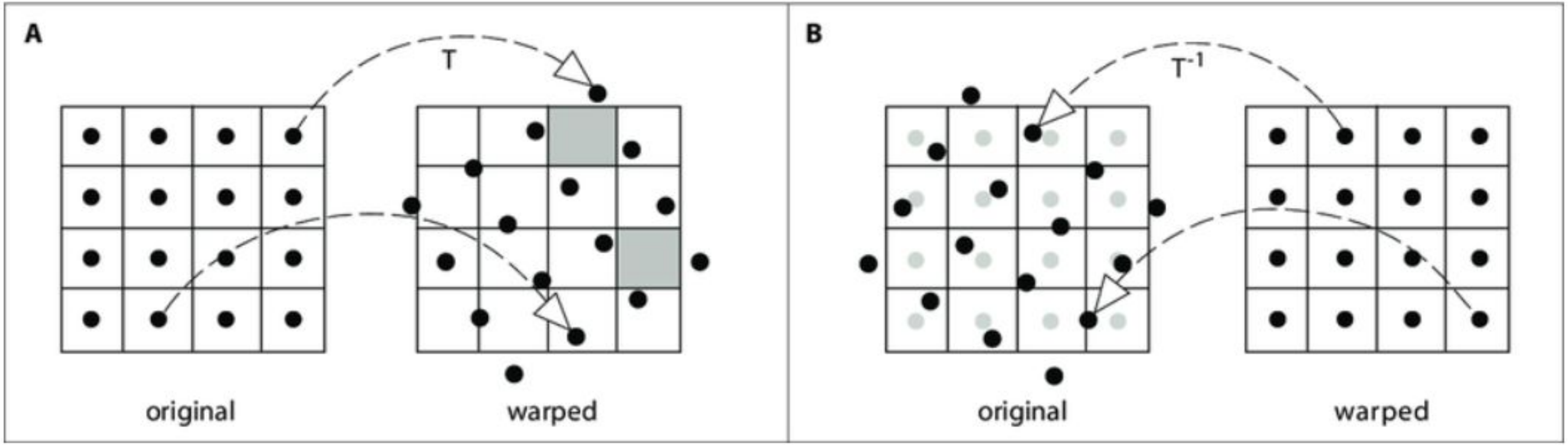
# Forward warping and inverse warping

Lab 1 Task 6
2. Compare forward and backward mapping and analyze their difference

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} & & \\ & T & \\ & & \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

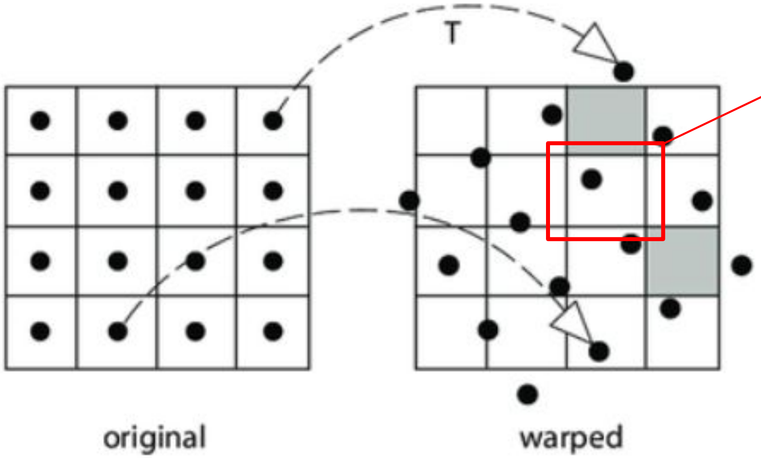
Forward warping

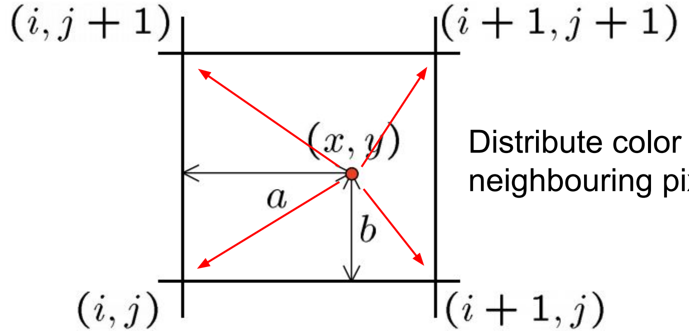
Inverse warping




(inverse)
2.: Forward and backward image warping. In the case of foward warping (A), holes can occur in the warped image, marked in gray. Backward warping (B) eliminates this problem since intensities at locations that do not coincide with pixel coordinates can be obtained from the original image using an interpolation scheme.

# Forward warping and splatting



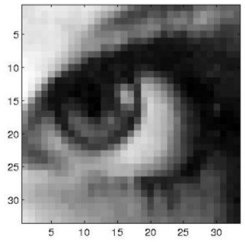



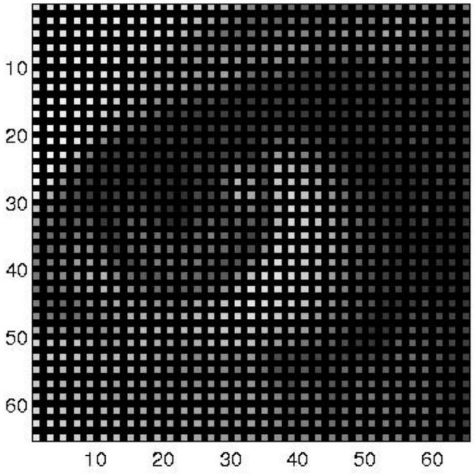

Distribute color among neighbouring pixels

Splatting

Given (x,y), calculate (x’,y’) by T

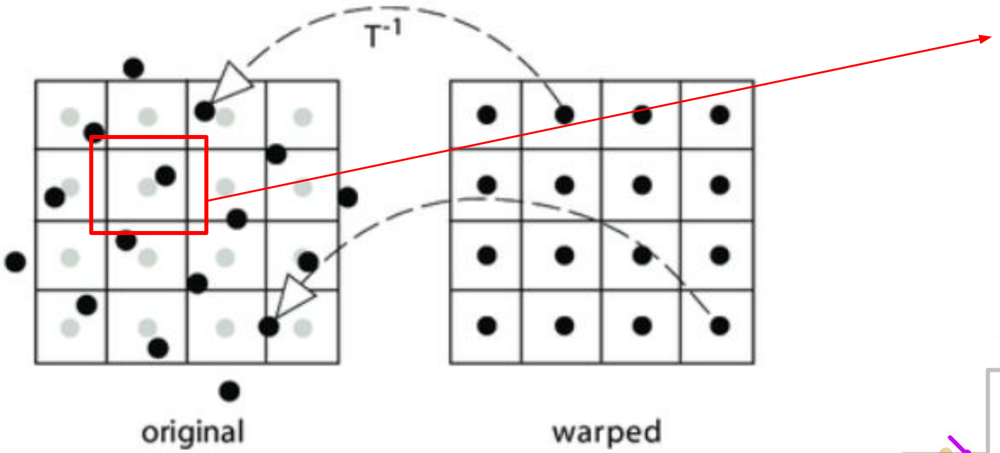
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = T \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



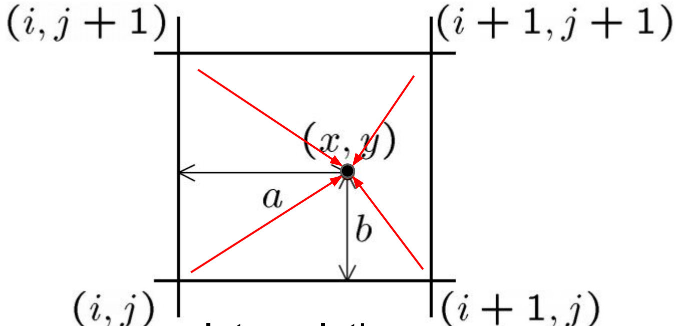

Holes in generated image

# Inverse warping and interpolation

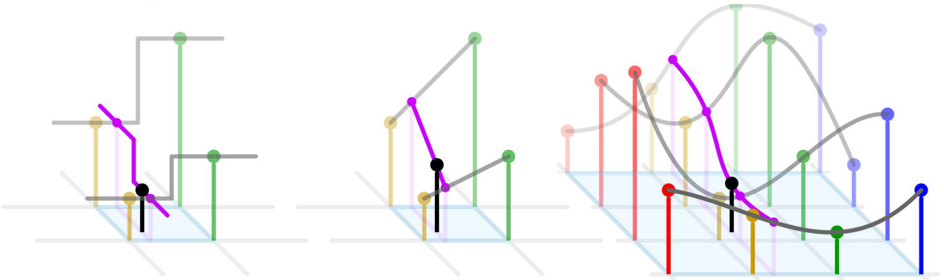



Given (x',y'), calculate (x,y) by $T^{-1}$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = T^{-1} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$




Interpolation

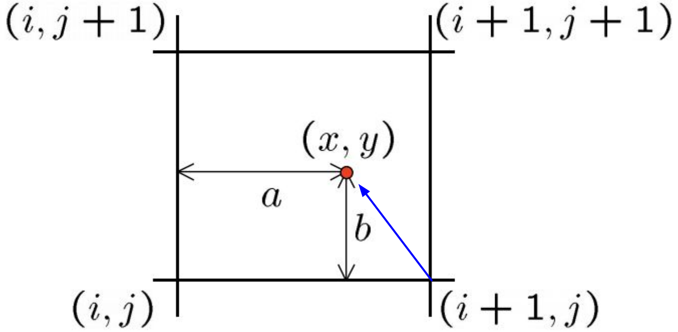


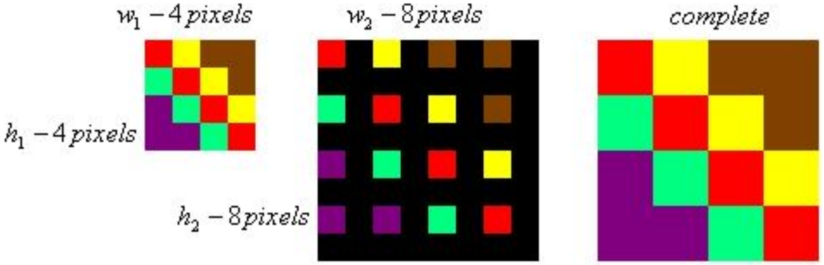
2D nearest-neighbour

Bilinear

Bicubic

# Nearest-neighbor interpolation




Selects the value of the nearest point

# Bilinear interpolation

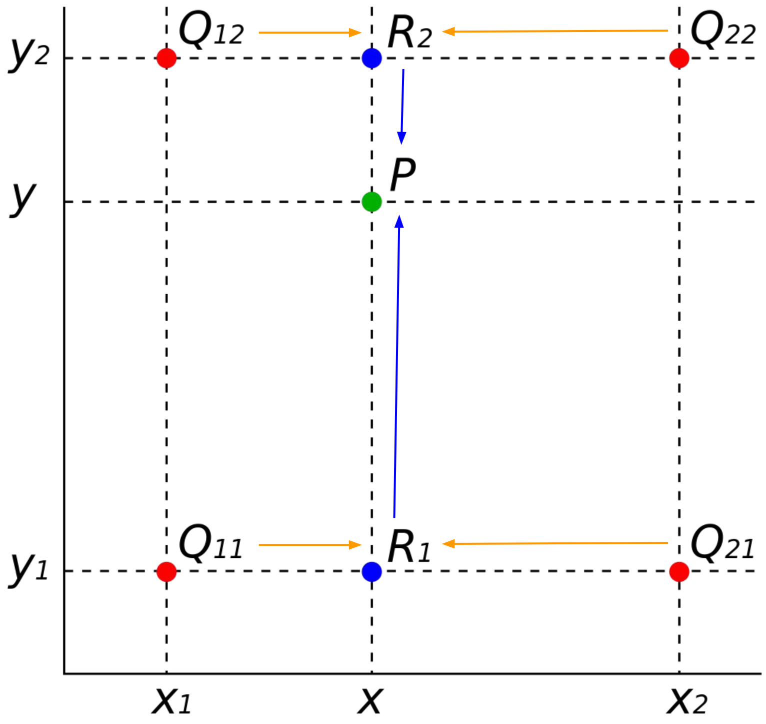
$$f(R_1) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21})$$

$$f(R_2) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22})$$

$$f(P) \approx \frac{y_2 - y}{y_2 - y_1} f(R_1) + \frac{y - y_1}{y_2 - y_1} f(R_2).$$

$$\Downarrow$$

$$\begin{aligned} f(x,y) \approx\ & \frac{f(Q_{11})}{(x_2-x_1)(y_2-y_1)}(x_2-x)(y_2-y) + \\ & \frac{f(Q_{21})}{(x_2-x_1)(y_2-y_1)}(x-x_1)(y_2-y) + \\ & \frac{f(Q_{12})}{(x_2-x_1)(y_2-y_1)}(x_2-x)(y-y_1) + \\ & \frac{f(Q_{22})}{(x_2-x_1)(y_2-y_1)}(x-x_1)(y-y_1) \\ =\ & \frac{1}{(x_2-x_1)(y_2-y_1)}\Big( f(Q_{11})(x_2-x)(y_2-y) + \\ & \quad f(Q_{21})(x-x_1)(y_2-y) + \\ & \quad f(Q_{12})(x_2-x)(y-y_1) + \\ & \quad f(Q_{22})(x-x_1)(y-y_1) \Big) \end{aligned}$$



# Bicubic interpolation

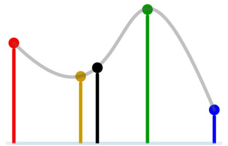
## A third degree polynomial and its derivative:

$$f(x) = ax^3 + bx^2 + cx + d \qquad f'(x) = 3ax^2 + 2bx + c$$

---

## Cubic interpolation:

$$f(p_0, p_1, p_2, p_3, x) = (-\tfrac{1}{2}p_0 + \tfrac{3}{2}p_1 - \tfrac{3}{2}p_2 + \tfrac{1}{2}p_3)x^3 + (p_0 - \tfrac{5}{2}p_1 + 2p_2 - \tfrac{1}{2}p_3)x^2 + (-\tfrac{1}{2}p_0 + \tfrac{1}{2}p_2)x + p_1$$



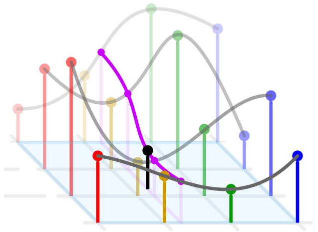
Cubic

---

## Bicubic interpolation:

$$p(x,y) = \begin{bmatrix} 1 & x & x^2 & x^3 \end{bmatrix} \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} 1 \\ y \\ y^2 \\ y^3 \end{bmatrix}$$

$$a_{00} = p_{11}$$
$$a_{01} = -\tfrac{1}{2}p_{10} + \tfrac{1}{2}p_{12}$$
$$a_{02} = p_{10} - \tfrac{5}{2}p_{11} + 2p_{12} - \tfrac{1}{2}p_{13}$$
$$a_{03} = -\tfrac{1}{2}p_{10} + \tfrac{3}{2}p_{11} - \tfrac{3}{2}p_{12} + \tfrac{1}{2}p_{13}$$
$$a_{10} = -\tfrac{1}{2}p_{01} + \tfrac{1}{2}p_{21}$$
$$a_{11} = \tfrac{1}{4}p_{00} - \tfrac{1}{4}p_{02} - \tfrac{1}{4}p_{20} + \tfrac{1}{4}p_{22}$$
$$a_{12} = -\tfrac{1}{2}p_{00} + \tfrac{5}{4}p_{01} - p_{02} + \tfrac{1}{4}p_{03} + \tfrac{1}{2}p_{20} - \tfrac{5}{4}p_{21} + p_{22} - \tfrac{1}{4}p_{23}$$
$$a_{13} = \tfrac{1}{4}p_{00} - \tfrac{3}{4}p_{01} + \tfrac{3}{4}p_{02} - \tfrac{1}{4}p_{03} - \tfrac{1}{4}p_{20} + \tfrac{3}{4}p_{21} - \tfrac{3}{4}p_{22} + \tfrac{1}{4}p_{23}$$
$$a_{20} = p_{01} - \tfrac{5}{2}p_{11} + 2p_{21} - \tfrac{1}{2}p_{31}$$
$$a_{21} = -\tfrac{1}{2}p_{00} + \tfrac{1}{2}p_{02} + \tfrac{5}{4}p_{10} - \tfrac{5}{4}p_{12} - p_{20} + p_{22} + \tfrac{1}{4}p_{30} - \tfrac{1}{4}p_{32}$$
$$a_{22} = p_{00} - \tfrac{5}{2}p_{01} + 2p_{02} - \tfrac{1}{2}p_{03} - \tfrac{5}{2}p_{10} + \tfrac{25}{4}p_{11} - 5p_{12} + \tfrac{5}{4}p_{13} + 2p_{20} - 5p_{21} + 4p_{22} - p_{23} - \tfrac{1}{2}p_{30} + \tfrac{5}{4}p_{31} - p_{32} + \tfrac{1}{4}p_{33}$$
$$a_{23} = -\tfrac{1}{2}p_{00} + \tfrac{3}{2}p_{01} - \tfrac{3}{2}p_{02} + \tfrac{1}{2}p_{03} + \tfrac{5}{4}p_{10} - \tfrac{15}{4}p_{11} + \tfrac{15}{4}p_{12} - \tfrac{5}{4}p_{13} - p_{20} + 3p_{21} - 3p_{22} + p_{23} + \tfrac{1}{4}p_{30} - \tfrac{3}{4}p_{31} + \tfrac{3}{4}p_{32} - \tfrac{1}{4}p_{33}$$
$$a_{30} = -\tfrac{1}{2}p_{01} + \tfrac{3}{2}p_{11} - \tfrac{3}{2}p_{21} + \tfrac{1}{2}p_{31}$$
$$a_{31} = \tfrac{1}{4}p_{00} - \tfrac{1}{4}p_{02} - \tfrac{3}{4}p_{10} + \tfrac{3}{4}p_{12} + \tfrac{3}{4}p_{20} - \tfrac{3}{4}p_{22} - \tfrac{1}{4}p_{30} + \tfrac{1}{4}p_{32}$$
$$a_{32} = -\tfrac{1}{2}p_{00} + \tfrac{5}{4}p_{01} - p_{02} + \tfrac{1}{4}p_{03} + \tfrac{3}{2}p_{10} - \tfrac{15}{4}p_{11} + 3p_{12} - \tfrac{3}{4}p_{13} - \tfrac{3}{2}p_{20} + \tfrac{15}{4}p_{21} - 3p_{22} + \tfrac{3}{4}p_{23} + \tfrac{1}{2}p_{30} - \tfrac{5}{4}p_{31} + p_{32} - \tfrac{1}{4}p_{33}$$
$$a_{33} = \tfrac{1}{4}p_{00} - \tfrac{3}{4}p_{01} + \tfrac{3}{4}p_{02} - \tfrac{1}{4}p_{03} - \tfrac{3}{4}p_{10} + \tfrac{9}{4}p_{11} - \tfrac{9}{4}p_{12} + \tfrac{3}{4}p_{13} + \tfrac{3}{4}p_{20} - \tfrac{9}{4}p_{21} + \tfrac{9}{4}p_{22} - \tfrac{3}{4}p_{23} - \tfrac{1}{4}p_{30} + \tfrac{3}{4}p_{31} - \tfrac{3}{4}p_{32} + \tfrac{1}{4}p_{33}$$



Bicubic

https://www.paulinternet.nl/?page=bicubic
https://en.wikipedia.org/wiki/Bicubic_interpolation

Original image:  x 10

Nearest-neighbor interpolation

Bilinear interpolation

Bicubic interpolation

Lab 1 Task 6

3. Compare different interpolation methods and analyze their difference