Università
Ca' Foscari
Venezia

Computer Vision

Projective geometry and 2D transformations

Filippo Bergamasco (filippo.bergamasco@unive.it)
http://www.dais.unive.it/~bergamasco
DAIS, Ca' Foscari University of Venice
Academic year 2018/2019

Geometric primitives

Geometric primitives form the basic building blocks used to describe 2D and 3D shapes.

We will study the basic geometric primitives (points, lines, conics) and the transformations that can be defined between them

The framework of projective geometry allow us to describe such transformations in a powerful generic way.

Università Ca' Foscari Venezia

Points

Points lying on an Euclidean 2D plane (like the image plane) are usually described as vectors:

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix} \in \mathbb{R}^2$$

This is a common way to reason about points but has some limitations. For example, we cannot describe points at infinity…

...Other alternatives are possible!

2D Projective space

Since the imaging apparatus usually behaves like a pinhole camera model, many of the transformations that may happen can be described as projective transformations.

This offer a general and powerful way to work with points, lines and conics

The **2D projective space** is simply defined as:

$$\mathbb{P}^2 = \mathbb{R}^3 - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Homogeneous coordinates

A point in a 2D Euclidean plane can be described in homogeneous coordinates (2D projective space) as follows:

$$\mathbf{x} = \begin{bmatrix} wx \\ wy \\ w \end{bmatrix} \in \mathbb{P}^2, \quad w \in \mathbb{R} - \{0\}$$

This implies that:

- There are infinitely many ways to describe a point **x**
- points in Euclidean space are represented by all the equivalence classes of three dimensional vectors where two elements are in the same class if they differ by a non zero scale factor.

Università Ca' Foscari Venezia

Homogeneous coordinates

From Euclidean to homogeneous:

Just add 1 to the last component

$$\begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

From homogeneous to Euclidean:

Divide by the last component (if not zero)

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \rightarrow \begin{bmatrix} x/w \\ y/w \end{bmatrix}$$

Homogeneous coordinates

All the points:

$$\begin{bmatrix} x \\ y \\ 0 \end{bmatrix} \in \mathbb{P}^2$$

are called **ideal points** or **points at infinity** and do not have an equivalent inhomogeneous representation (in Euclidean plane).

2D Lines

A line in the Euclidean plane can be described as the locus of points p=(x,y) so that:

$$ax + by + c = 0$$

In projective space, the same line can be represented as:

$$\begin{bmatrix} X & Y & w \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = 0$$

2D Lines

$$\begin{bmatrix} X & Y & w \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = 0$$

This represent each point on the line

This represent the line itself

in P^2 points and lines are described in the same way!

This leads to a simple expression to find the intersection **x** of two lines **u** and **u'** using the cross product:

$$\mathbf{x} = \mathbf{u} \times \mathbf{u}'$$

2D Lines intersection

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \times \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} u_2v_3 - u_3v_2 \\ u_3v_1 - u_1v_3 \\ u_1v_2 - u_2v_1 \end{bmatrix}$$

What happens if the lines are parallel?

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} \times \begin{bmatrix} a \\ b \\ c' \end{bmatrix} = \begin{bmatrix} bc' - cb \\ ca - ac' \\ 0 \end{bmatrix}$$

Università Ca' Foscari Venezia

Line and points

Similarly, the line $\mathbf{l}$ joining two points $\mathbf{p}_1$ and $\mathbf{p}_2$ can be expressed as:

$$\mathbf{l} = \mathbf{p}_1 \times \mathbf{p}_2$$

Università
Ca' Foscari
Venezia

2D Lines

We can normalize a line

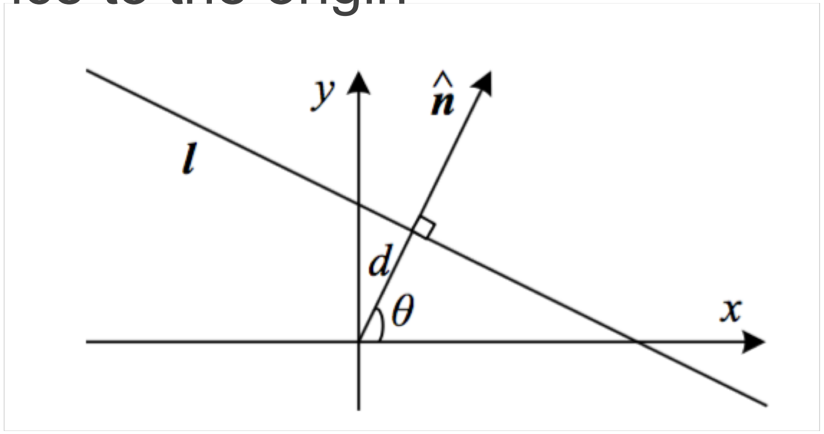
$$\mathbf{l} = \begin{bmatrix} n_1 \\ n_2 \\ d \end{bmatrix}$$

So that $\sqrt{n_1^2 + n_2^2} = 1$

2D Lines

With a normalized line,

- The vector $(n_1\ n_2)^T$ is the line normal and d is the line distance to the origin



- The point-line distance d from a line **l** and a point **x** can be computed as

$$d = |\mathbf{l}^T \mathbf{x}|$$

Duality principle

It is simple to note that the role of points and lines may be interchanged in statements containing lines and points.

Both lines and points in $\mathbb{P}^2$ are represented as vectors of 3 elements. This implies for example that the basic incidence equation $l^T x = 0$ can be swapped: $x^T l = 0$

Similarly we got:

Line-line intersection: $p = l \times l'$

Line passing through two points: $l = p \times p'$

Duality principle

«To any theorem of 2-dimensional projective geometry there corresponds a dual theorem, which may be derived by interchanging the roles of points and lines in the original theorem.»

Ideal points and the line at infinity

We have seen that all homogeneous vectors $x=(x_1,x_2,x_3)^T$ such that $x_3=0$ are called ideal points.

The set of all the ideal points lies on a single line, named line at infinity, denoted by the vector

$$l_\infty = (0,0,1)^T$$

Indeed: $(0,0,1)(x_1,x_2,0)^T = 0 \;\; \forall x_1, x_2$

Università Ca' Foscari Venezia

The line at infinity

Any line $l = (a, b, c)^T$ intersects l_∞ in the ideal point $l \times l_\infty = (b, -a, 0)^T$.

A line $l' = (a, b, c')^T$ (parallel to l) intersects l_∞ in the same ideal point $l' \times l_\infty = (b, -a, 0)^T$

The inhomogeneous vector $(b, -a)$ is a vector tangent to l and orthogonal to the line normal (a, b)

The line at infinity can be thought of as the set of directions of lines in a plane

Università Ca' Foscari Venezia

The projective plane

The 2D projective space $\mathbb{P}^2$ can be seen as a set of rays in $\mathbb{R}^3$:

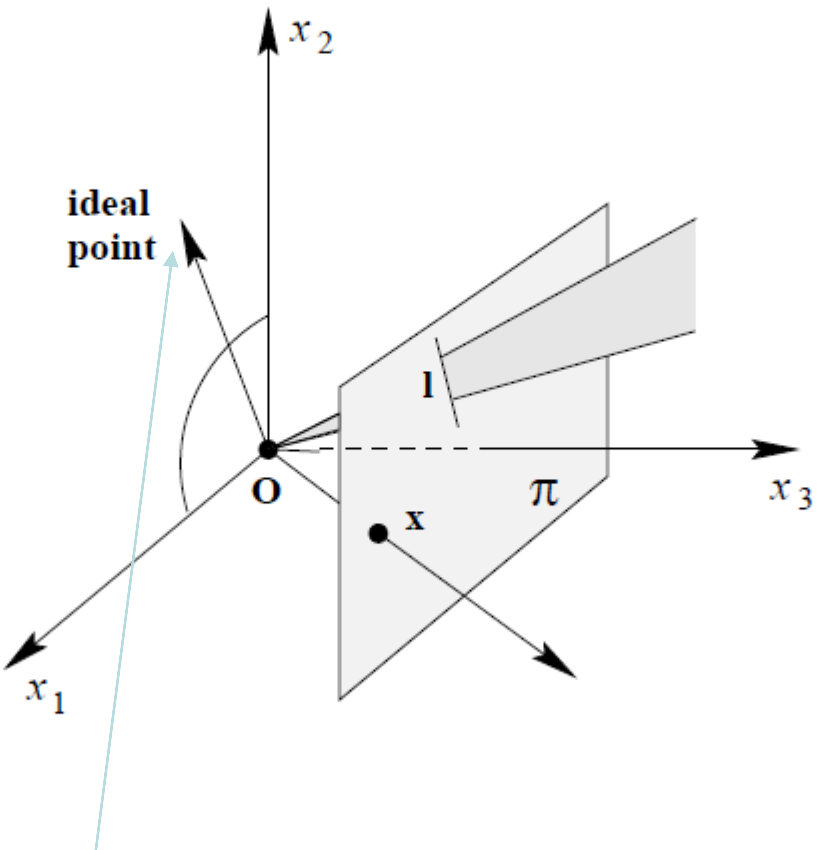
Indeed, the set of all vectors $k(x_1, x_2, x_3)^T$, varying k, forms a ray through the origin.

In this model we have:

A ray (through the origin) in $\mathbb{R}^3$ is a point in $\mathbb{P}^2$

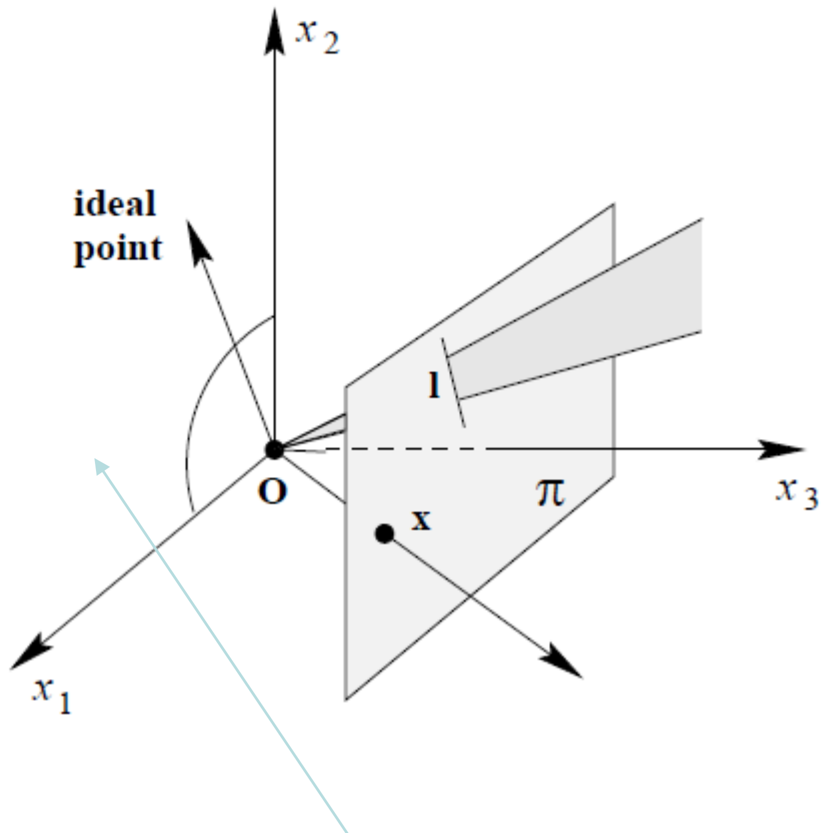
A line in $\mathbb{P}^2$ define a plane passing through the origin. In fact two non identical rays lie on exactly one plane and any two planes intersect in one ray.

The projective plane



All the lines lying in the plane spanned by x_1 and x_2 represent ideal points

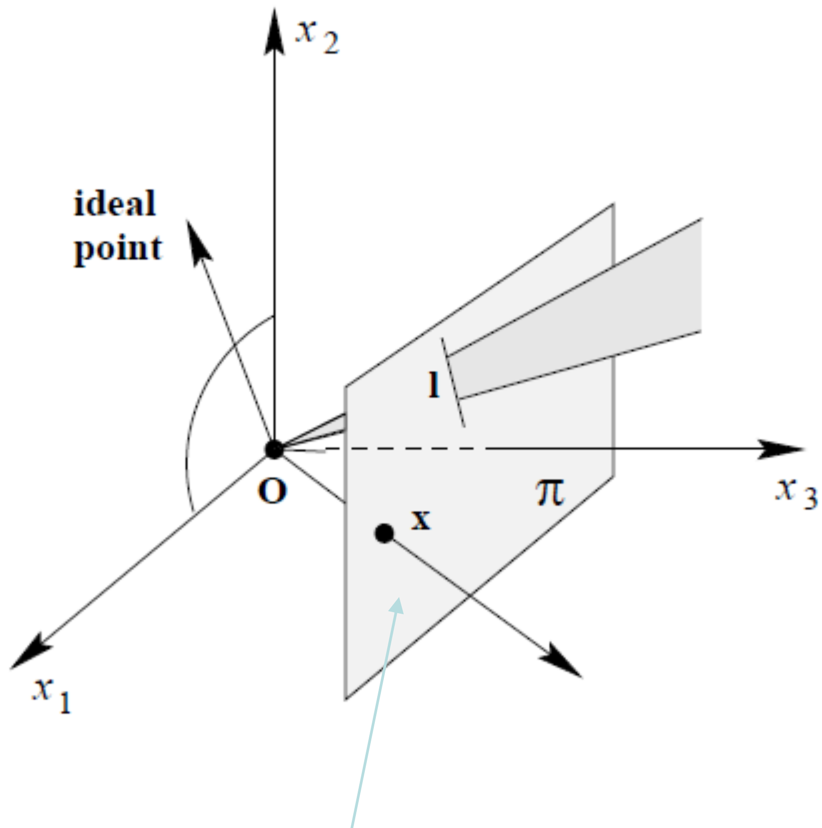
The projective plane



The plane spanned by x_1 and x_2 represents l_∞

Università
Ca' Foscari
Venezia

The projective plane



Points and lines may be obtained by intersecting rays and planes passing through the origin with the plane $x_3=1$

Università Ca' Foscari Venezia

Conics

Conics that can be represented by second degree polynomials in the form:

$$ax^2 + bxy + cy^2 + dx + ey + f = 0$$

In homogeneous coordinates the conic can be represented by a matrix

$$\mathtt{C} = \begin{bmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{bmatrix}$$

Such that a point $x \in \mathbb{P}^2$ belong to the conic if:

$$x^T C x = 0$$

Università Ca' Foscari Venezia

Conics

$$x^T C x = 0 \qquad \mathtt{C} = \begin{bmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{bmatrix}$$

Since multiplying C by any non-zero scalar does not affect the equation, **C is an homogeneous representation of the conic and has 5 degrees of freedom.**

Depending on the parameters, we can obtain different conics:

- Circles
- Ellipses
- Parabolas
- Hyperbolae

Dual Conics

$$x^T C x = 0$$

The definition above describe a point conic as it defines squation on points. For the duality principle there exist also a conic defining a similar equation for lines.

A dual conic is represented by a matrix $C^* = C^{-1}$ (if C is non singular) defined by the set of all the lines l tangent to the conic satisfying the equation:

$$l^T C^* l = 0$$

Università Ca' Foscari Venezia

Dual Conics

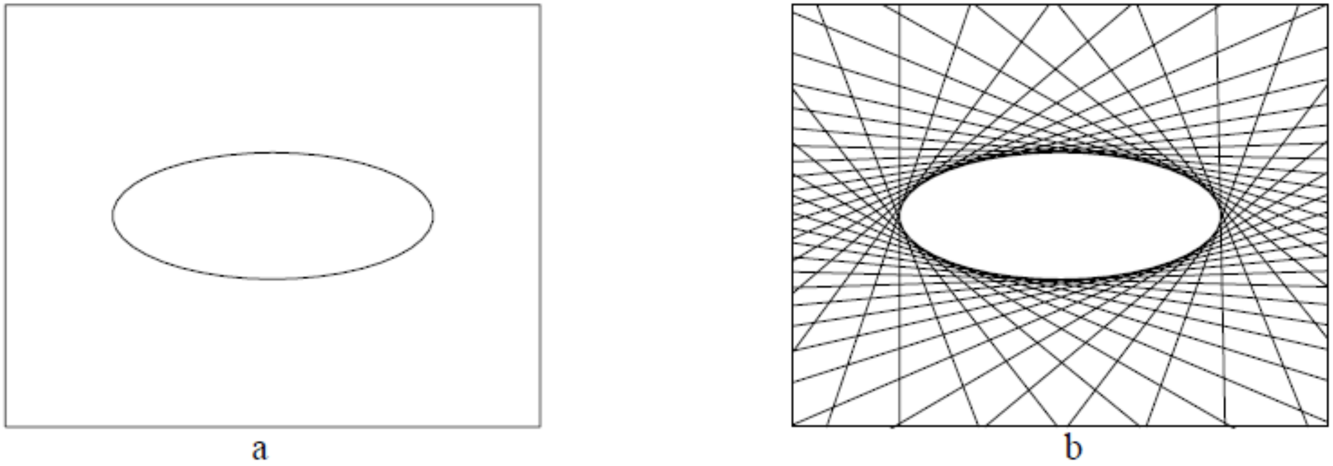


Fig. 2.2. *(a) Points* $\mathbf{x}$ *satisfying* $\mathbf{x}^\mathsf{T} \mathtt{C} \mathbf{x} = 0$ *lie on a point conic. (b) Lines* $\mathbf{l}$ *satisfying* $\mathbf{l}^\mathsf{T} \mathtt{C}^* \mathbf{l} = 0$ *are tangent to the point conic* $\mathtt{C}$*. The conic* $\mathtt{C}$ *is the envelope of the lines* $\mathbf{l}$.

2D projectivities

A planar projective transformation is a linear transformation in $\mathbb{P}^2$ that can be represented by any non-singular 3x3 matrix **H.**

Applied to points, the transformation is written as:

$$\begin{bmatrix} X' \\ Y' \\ W' \end{bmatrix} = \mathbf{Hx} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ W \end{bmatrix}$$

For the duality principle, the same transformation can be applied to a line as:

$$l' = H^{-T} l$$

Università
Ca' Foscari
Venezia

2D Translation

$$\begin{bmatrix} X' \\ Y' \\ W \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ W \end{bmatrix}$$

Dof: 2

Preserves: Orientation

Università
Ca' Foscari
Venezia

2D Rotation

$$\begin{bmatrix} X' \\ Y' \\ W \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ W \end{bmatrix}$$

Dof: 1

Preserves: Lengths

2D Rigid motion

$$\begin{bmatrix} X' \\ Y' \\ W \end{bmatrix} = \mathbf{TR} \begin{bmatrix} X \\ Y \\ W \end{bmatrix}$$

Combination of a rotation **R** and a translation **T**

Dof: 3

Preserves: Lengths

2D Similarity

$$\begin{bmatrix} X' \\ Y' \\ W \end{bmatrix} = \begin{bmatrix} s\cos\theta & -s\sin\theta & t_x \\ s\sin\theta & s\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ W \end{bmatrix}$$

Where s is a non-zero scale factor.

Composed by Rotation + Isotropic Scale + Transl.

Dof: 4

Preserves: Angles between lines

2D Affine transformation

Università Ca' Foscari Venezia

$$\begin{bmatrix} X' \\ Y' \\ W \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ W \end{bmatrix}$$

An affine transformation is a non-singular linear transformation followed by a translation

Dof: 6

Preserves: Parallellism of lines

Università
Ca' Foscari
Venezia

2D Projective transformation

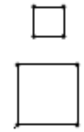
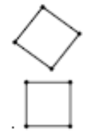
$$\begin{bmatrix} X' \\ Y' \\ W' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ W \end{bmatrix}$$

Projective transformation is also called **homography**.
Since we work in $\mathbb{P}^2$, **H** is defined up to scale.

Dof: 8 (not 9! Because any scale define the same transformation)

Preserves: Straight lines

Transformations table

Group	Matrix	Distortion	Invariant properties
Projective 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 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 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 3 dof	$\begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Length, area

Università
Ca' Foscari
Venezia

Projective transformations

The projective transformation (or homography) is a general non-singular linear transformation of homogeneous coordinates.

In block-form can be expressed as follows:

$$x' = Hx = \begin{pmatrix} A & t \\ \mathrm{v}^T & v \end{pmatrix} x$$

And can be decomposed into the chain of transformations:

$$\mathrm{H} = \mathrm{H_S}\,\mathrm{H_A}\,\mathrm{H_P} = \begin{bmatrix} s\mathrm{R} & \mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} \mathrm{K} & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} \mathrm{I} & \mathbf{0} \\ \mathbf{v}^\mathsf{T} & v \end{bmatrix} = \begin{bmatrix} \mathrm{A} & \mathbf{t} \\ \mathbf{v}^\mathsf{T} & v \end{bmatrix}$$

Projective transformations

$$\mathrm{H} = \mathrm{H}_S \, \mathrm{H}_A \, \mathrm{H}_P = \begin{bmatrix} s\mathrm{R} & \mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} \mathrm{K} & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} \mathrm{I} & \mathbf{0} \\ \mathbf{v}^\mathsf{T} & v \end{bmatrix} = \begin{bmatrix} \mathrm{A} & \mathbf{t} \\ \mathbf{v}^\mathsf{T} & v \end{bmatrix}$$

Similarity

Projective transformation

Affine transformation with K upper triangular matrix normalized so that det(K)=1

Note: the factorization is valid if $v \neq 0$

Projective vs. Affine

$$\mathbf{x}' = \mathrm{H_P}\mathbf{x} = \begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{v}^\mathsf{T} & v \end{bmatrix} \mathbf{x}$$

The main difference between projective and affine transformation is in the vector **v** (not null for a projectivity).

The vector is responsible for the non linear effects of the projectivity, and allows such transformation to model *vanishing points.*

Università Ca' Foscari Venezia

Projective vs. Affine

In a projective transformation ideal points are mapped to finite points:

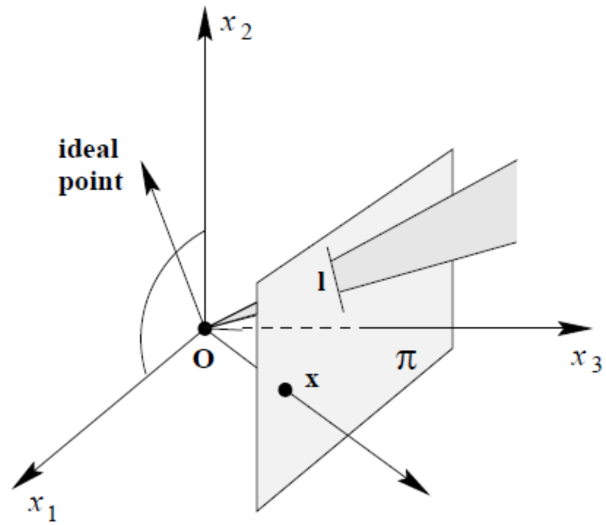
$$\begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{v}^\mathsf{T} & v \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 0 \end{pmatrix} = \begin{pmatrix} \mathtt{A} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ v_1 x_1 + v_2 x_2 \end{pmatrix}$$

In an affine transformation ideal points remain ideal:

$$\begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 0 \end{pmatrix} = \begin{pmatrix} \mathtt{A} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ 0 \end{pmatrix}$$

Projective transformations

Let's go back to the ray model discussed before:

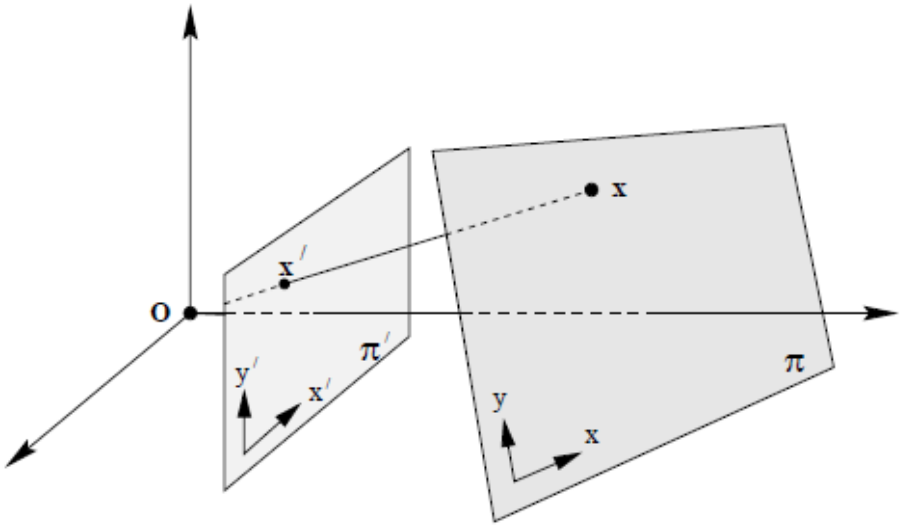


A projective transformation H is simply a linear transformation of $\mathbb{R}^3$.

Projective transformations

Università
Ca' Foscari
Venezia

A projective transformation H is simply a linear transformation of $\mathbb{R}^3$.

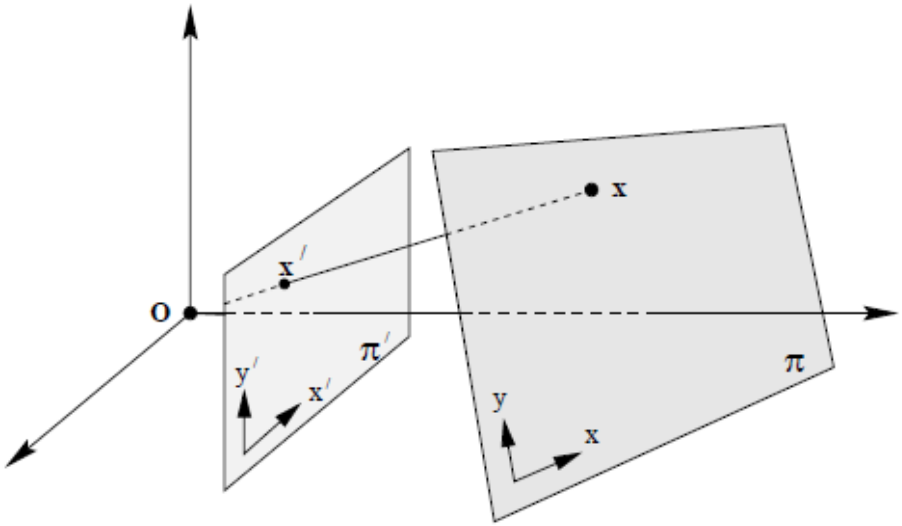


If a coordinate system is defined in each plane and points x and x' are represented in homogeneous coordinates (ie.rays), then the the central projection mapping x to x' can be expressed as x' = H x

Università Ca' Foscari Venezia

Projective transformations

A projective transformation H is simply a linear transformation of $\mathbb{R}^3$.



If the coordinate systems are both Euclidean (ie. Rectilinear) then the mapping defined by the central projection is called **perspectivity**.

We will see this for the pinhole camera model...

Spatial transformations on images

The projective transformations discussed so far can be applied to the image domain to **transform the geometry of the image plane**

Given an image $f(\mathbf{x}) = f(x, y)$

And a transformation function $s : \mathbb{R}^2 \rightarrow \mathbb{R}^2$

A spatial transformation changes the image as following:

$$g(\mathbf{x}) = f(s(\mathbf{x}))$$

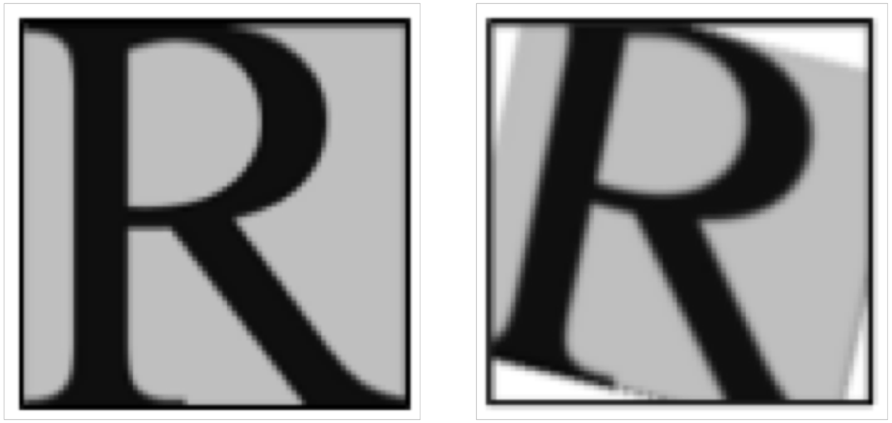
NOTE: Different from intensity transformations: $f(\mathbf{x}) = h(f(\mathbf{x}))$

Università Ca' Foscari Venezia

Spatial transformations on images

Example: image rotation

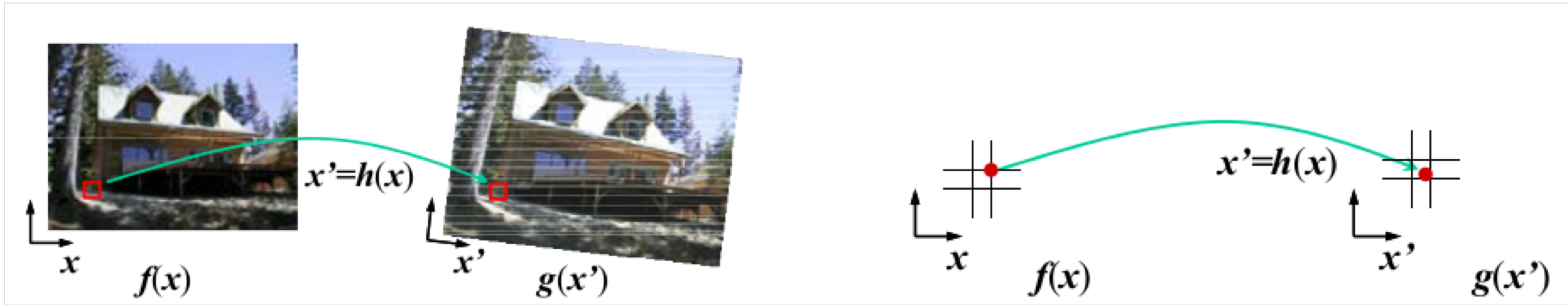
$$s(\mathbf{x}) = s(x, y) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



Università Ca' Foscari Venezia

Forward warp

How to perform the transformation? One simple way is via forward warping:



A pixel f(**x**) is copied to its corresponding location **x'**=s(**x**) in image g(**x'**).

Forward warp

Problem: x' usually has a non-integer value, and g(x') is not defined in that case

Possible solution1:

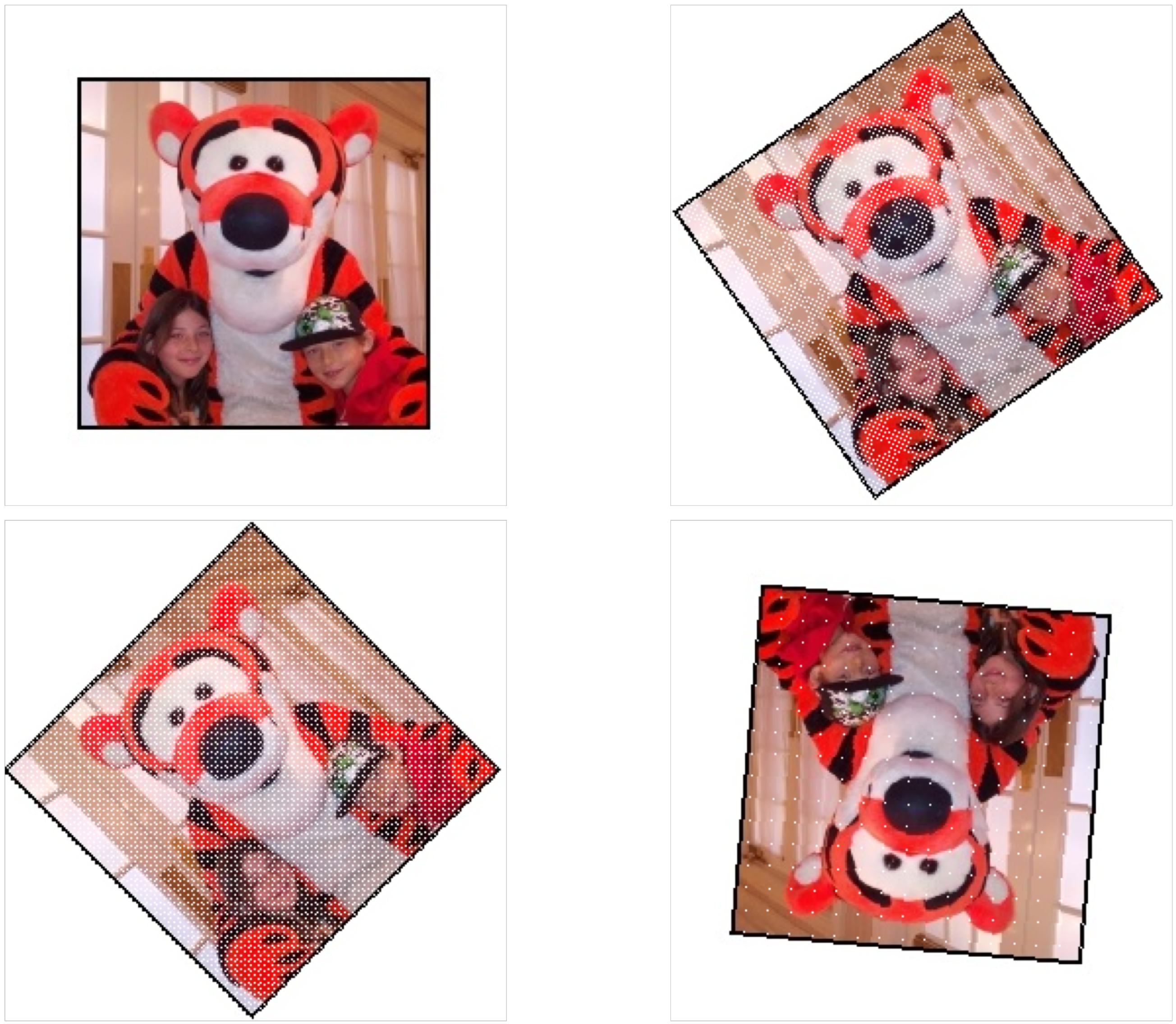
Round the value and copy the pixel there (Create cracks and holes)

Possible solution2:

Distribute the value among its n-neighbours in a weighted fashion (Cause aliasing and blur)

Università
Ca' Foscari
Venezia

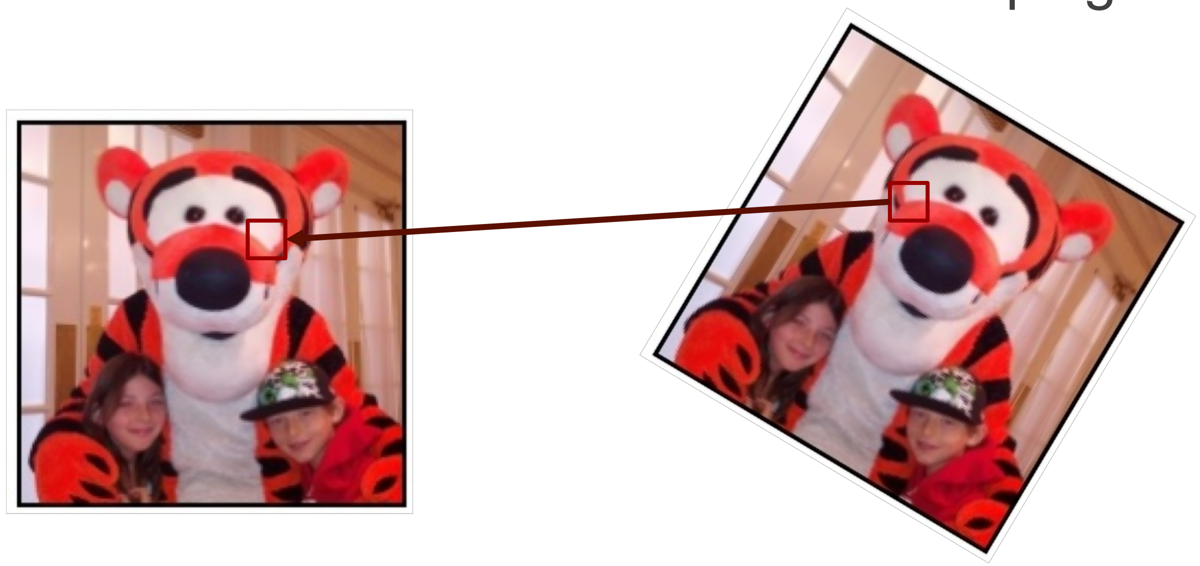
Forward warp



Università
Ca' Foscari
Venezia

Inverse warp

A preferable solution is to use inverse warping:



each pixel in the destination image g(**x'**) is sampled from the original image at **x** = s^{-1}(**x'**)

Inverse warp

Advantages:

- No holes since $s^{-1}(\mathbf{x}')$ is defined for all values of **x'** (in all the g(**x'**) domain)

Problems:

- The transformation function must be invertible (not really a problem since our transformation matrix is non-singular)
- Point sampling may still occur at non integer locations
  - This is a well studied problem: **Image interpolation**

Università Ca' Foscari Venezia

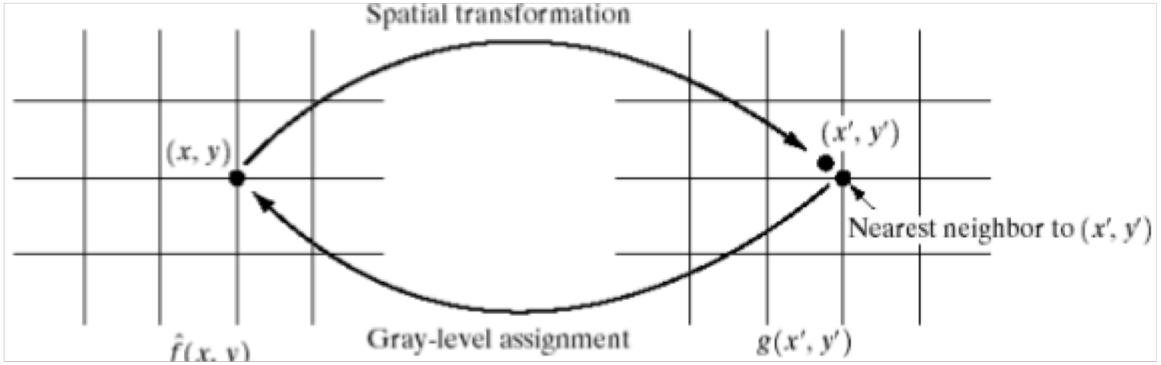
Nearest-neighbour interpolation

Interpolation: estimate the values using the information from nearby samples

Nearest-neighbour:

Use the image value at the closest integer location

$$g(\mathbf{x}') = f\left(\mathrm{round}(s^{-1}(\mathbf{x}'))\right)$$



Università Ca' Foscari Venezia

Nearest-neighbour interpolation

NN-interpolation is very fast but creates artefacts (blocks) especially if the transformation changes the scale (ie.zooming)

Università
Ca' Foscari
Venezia

Bilinear interpolation

Use the four points around s^{-1}(**x'**) to get a better estimation:

$$g(\mathbf{x}') = \alpha f(x', y') + \beta f(x'+1, y') + \gamma f(x', y'+1) + \delta f(x'+1, y'+1)$$

$$(x' \; y') = \text{floor}\left(s^{-1}(\mathbf{x}')\right)$$

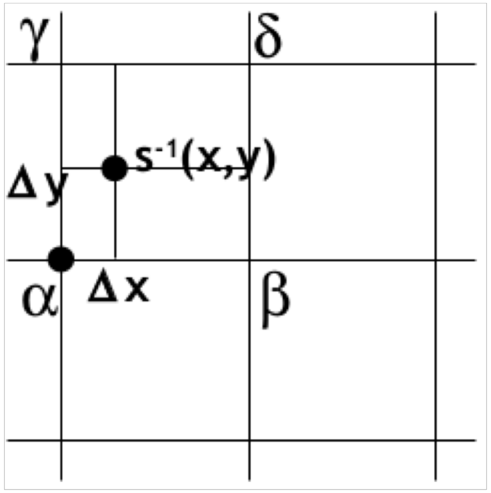
$$(\Delta x \; \Delta y) = s^{-1}(\mathbf{x}') - \mathbf{x}'$$

$$\alpha = (1 - \Delta x)(1 - \Delta y)$$

$$\beta = \Delta x(1 - \Delta y)$$

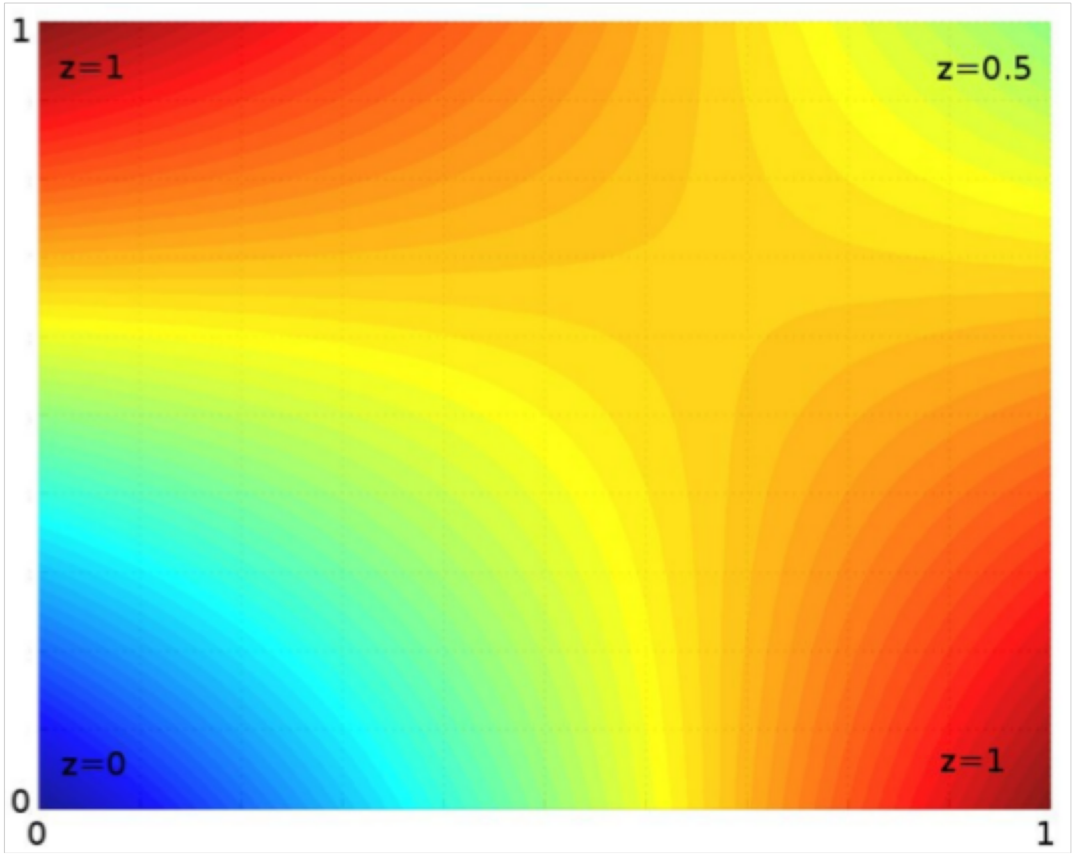
$$\gamma = (1 - \Delta x)\Delta y$$

$$\delta = \Delta x \Delta y$$



Università Ca' Foscari Venezia

Bilinear interpolation



Bilinear interpolation

NN-interpolation

Bilinear interpolation