Image Warping

Lecturer: Sang Hwa Lee

Image warping (1)

- What is image warping ?
  - Given a coordinate transformation $(x', y') = h(x, y)$ and a source image $f(x, y)$,
  - Generate a transformed and complete image $g(x', y') = f(h(x, y))$?

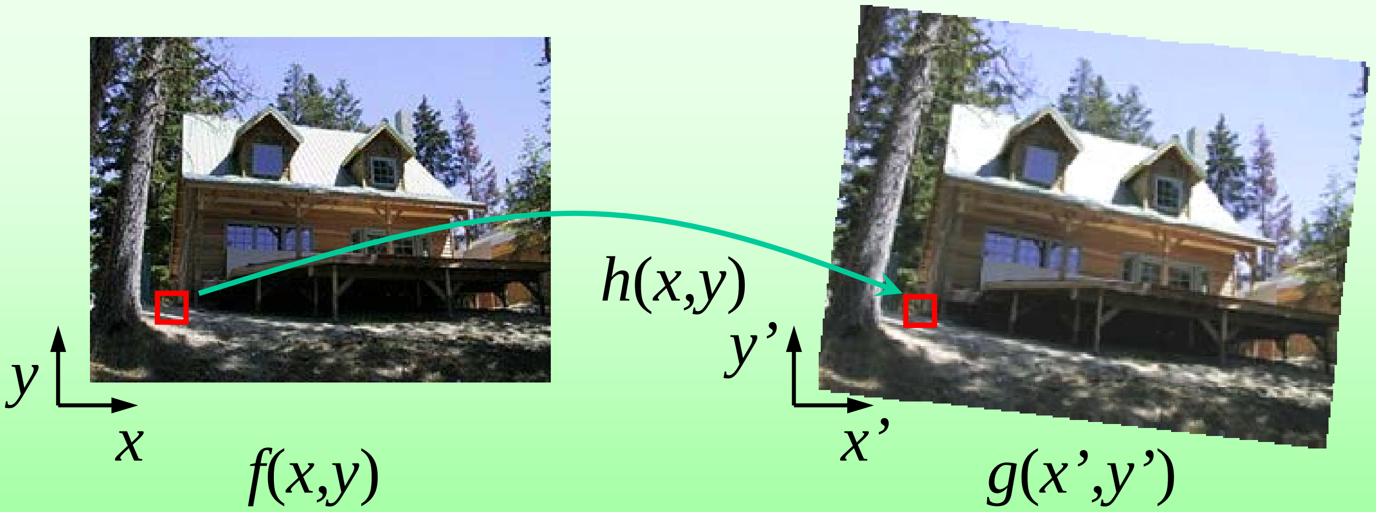


Image warping (2)

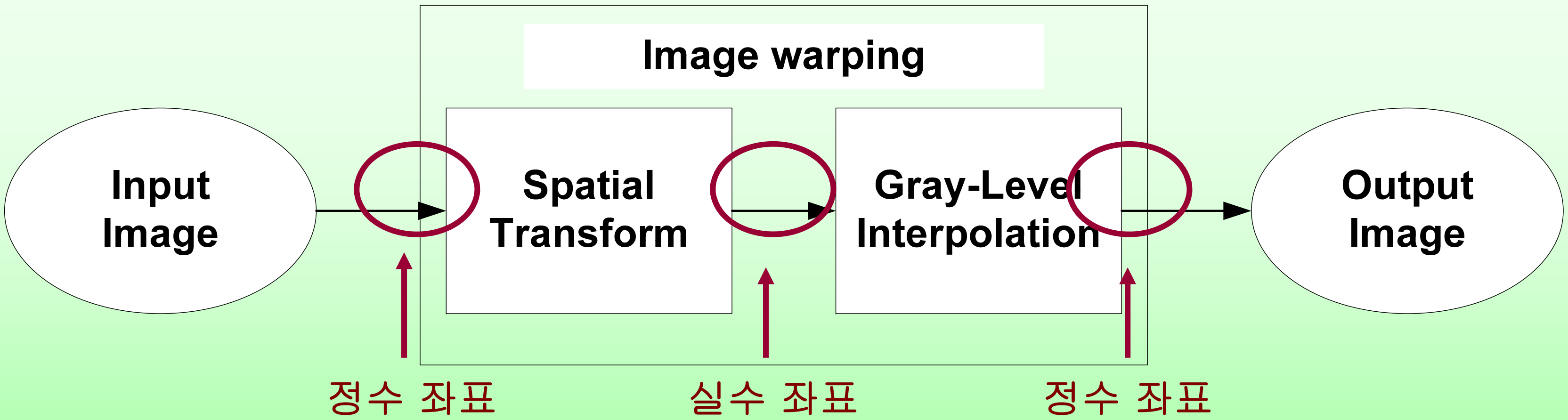
❑ Elements of image warping

- Spatial transformations
  - Affine, Projective transformation
  - Motion vector or disparity (depth) – compensation
- Error correction
  - Reducing aliasing
  - Hole filling due to non-integer spatial transformation
  - Interpolation/extrapolation
- Illumination conditions
  - Lighting direction
  - Shading
  - Reflection

Image warping (3)

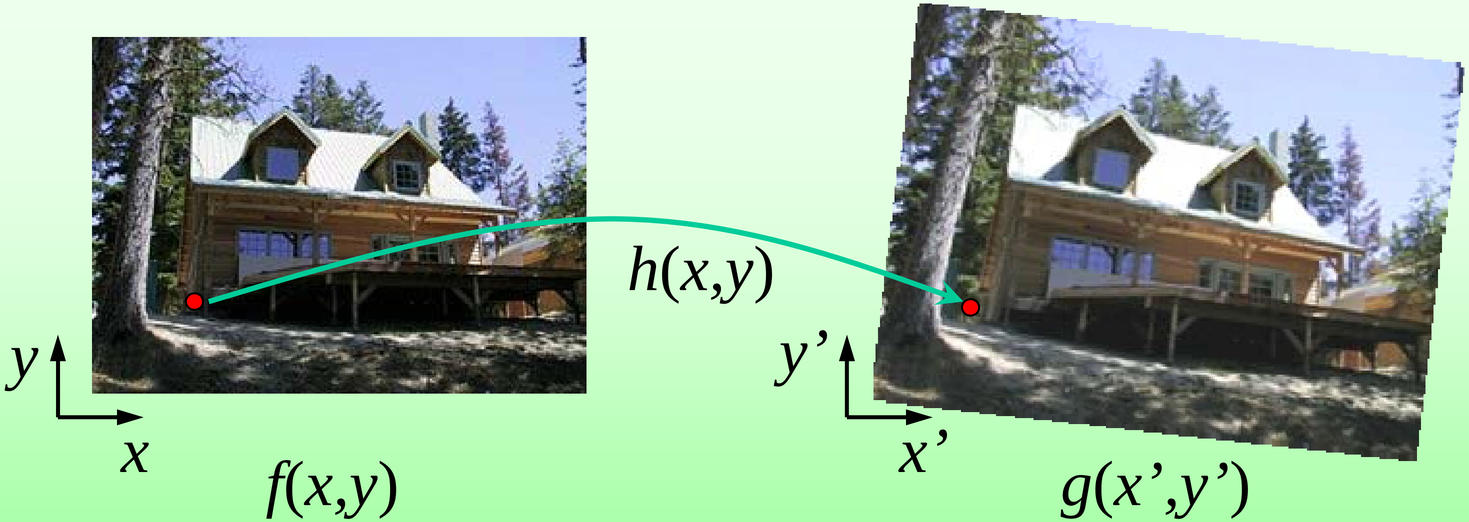
❑ Spatial transformations

- Transformation : from coordinate to coordinate.
- Reference : from value to value (intensity or color).
- Mapping is usually defined as matrix form.



Forward warping (1)

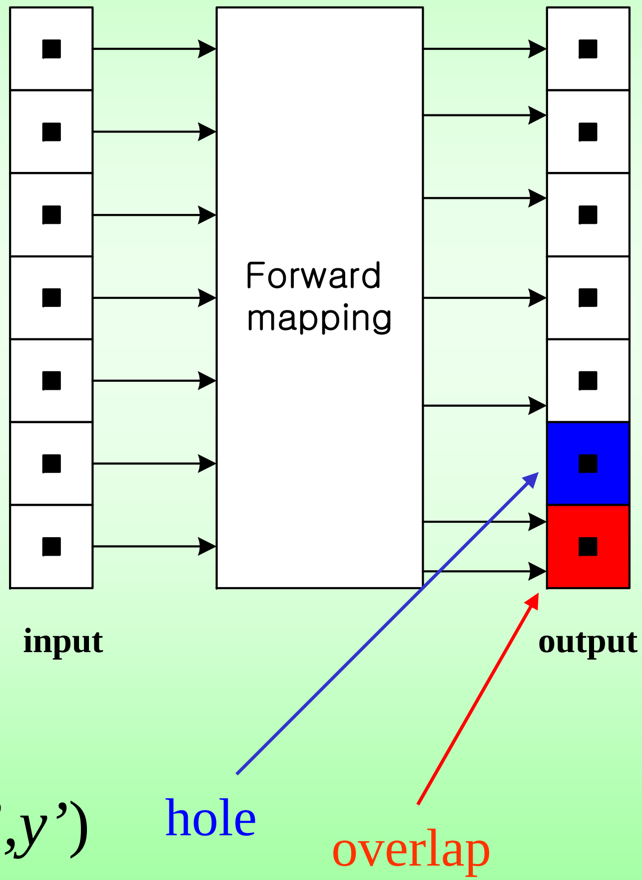
- Send each pixel $f(x,y)$ to its corresponding location $(x',y') = h(x,y)$ in the second image
- Splatting in computer graphics

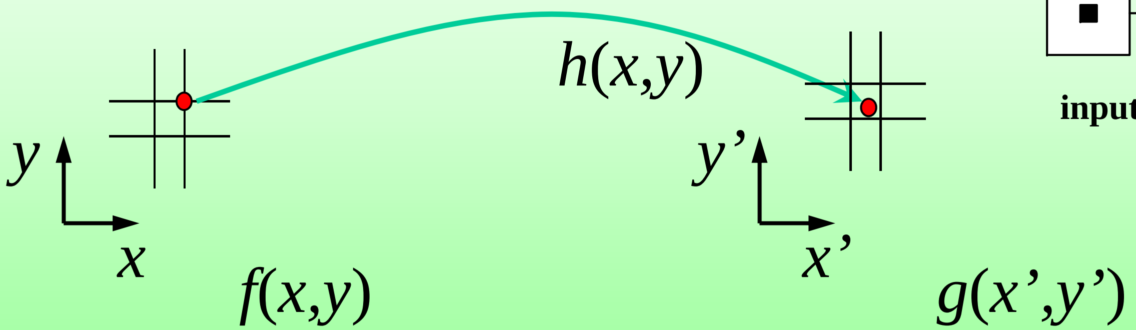


Forward warping (2)

❑ Forward mapping

- Integer input → real number output
- Problems : hole, overlap

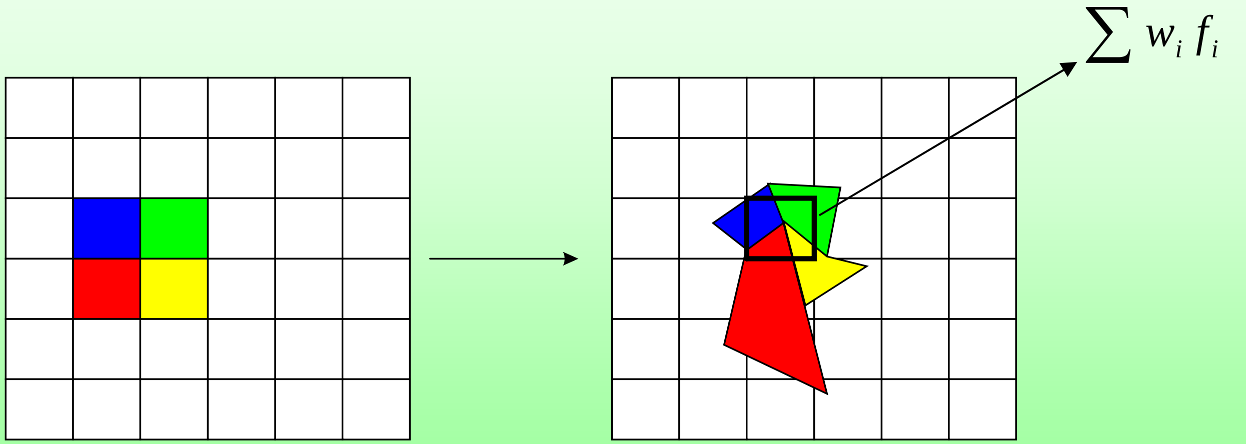




Forward warping (3)

Four-corner mapping

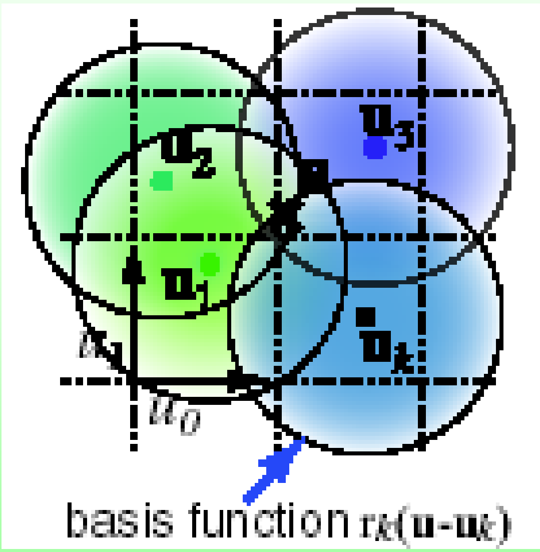
- Consider input pixel as square
- Squares → quadrilaterals
- contiguous pixels → contiguous pixels (removal of hole and overlap)
- Problems : costly intersection tests, magnification
- Solutions : adaptive sampling of input image



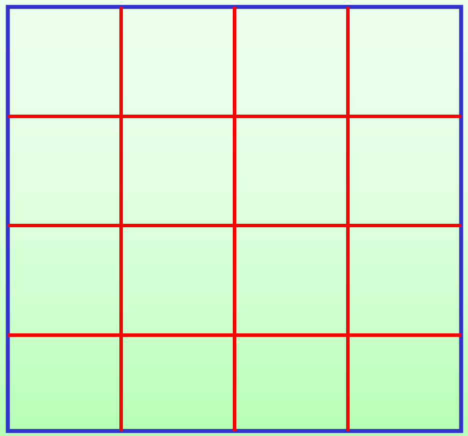
Forward warping (4)

❑ Recent solutions

- Distribution by continuous radial basis models
- Super resolution
  - ➢ Subpixels are generated by the interpolation
- Random resampling of input pixels

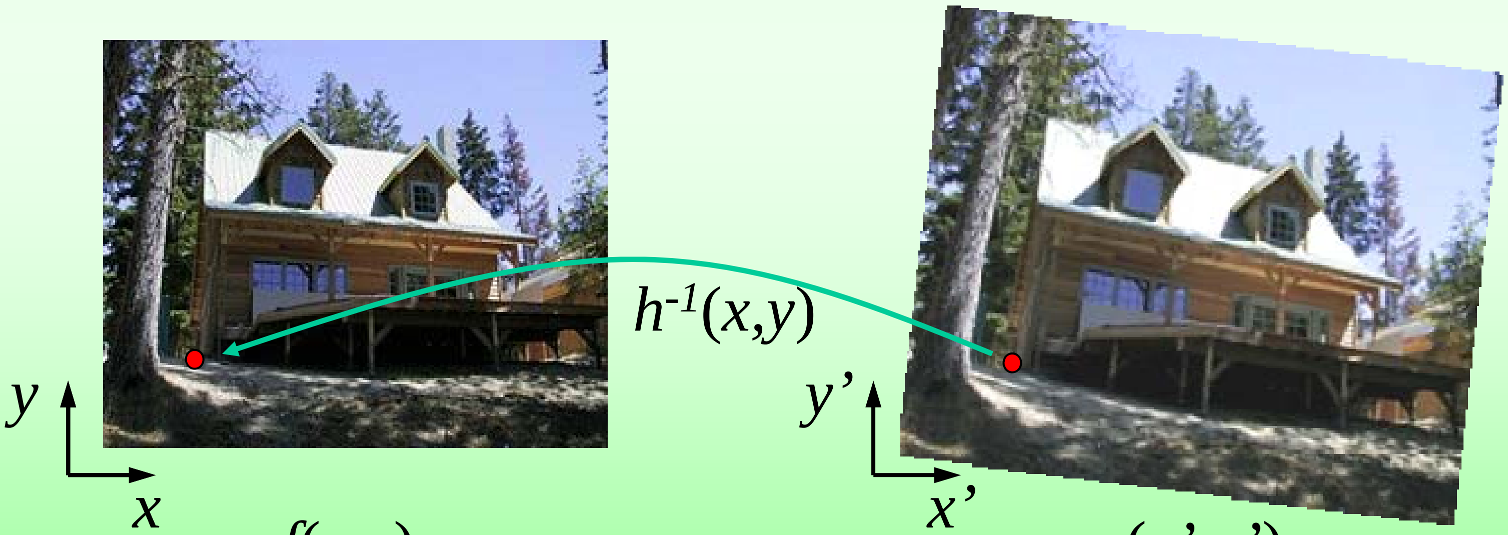


Radial basis modeling


Super resolution
(16 partitions of a pixel)

Inverse warping (1)

- Get each pixel $g(x',y')$ from its corresponding location $(x,y) = h^{-1}(x',y')$ in the first image

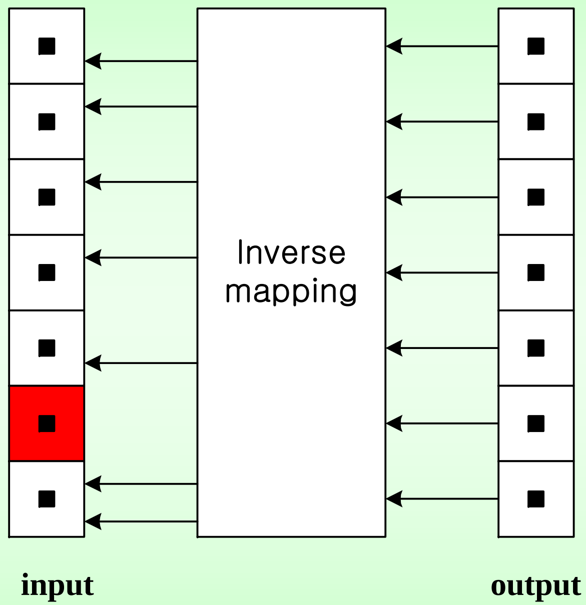


$f(x,y)$ $g(x',y')$

Inverse warping (2)

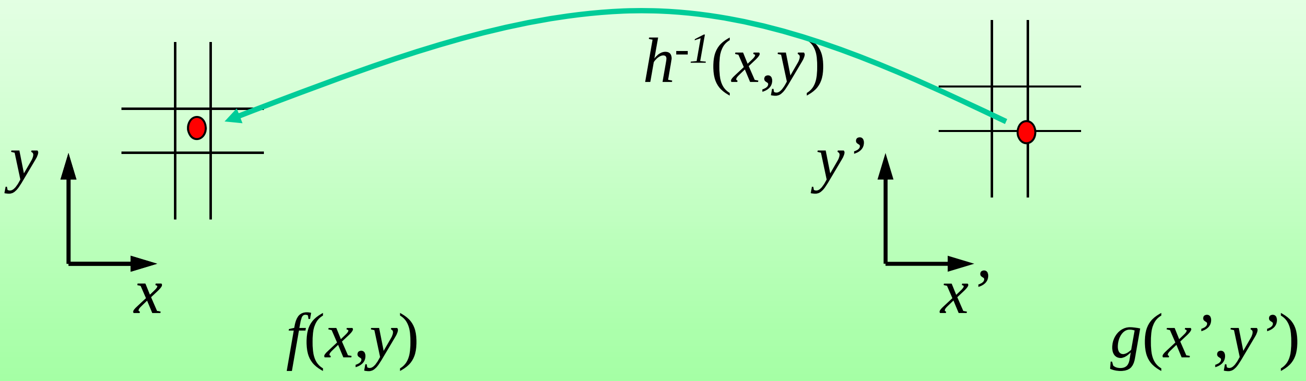
❑ Inverse mapping

- Integer number input → real number output.
- Guarantees that all output pixels are computed.
- Interpolation using surrounding pixels is needed.
- Filtering is needed.



Inverse warping (3)

- What if pixel comes from “between” four pixels?
  - resampling
    - nearest neighbor, bilinear
- Usually inverse warping is better
  - Elimination of spatial holes
  - Requirement of an invertible warp function



Forward Warping Examples (1)

원영상 (512x512)

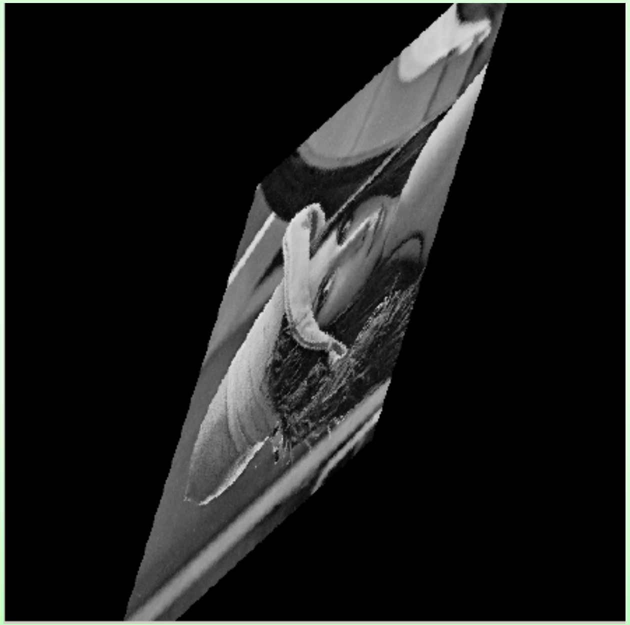
x축 60도 회전

y축 60도 회전

Forward Warping Examples (2)

z축 30도

z축 30도->y축 30도->x축 30도



z축 60도->y축 60도->x축 60도

Image Warping Comparison

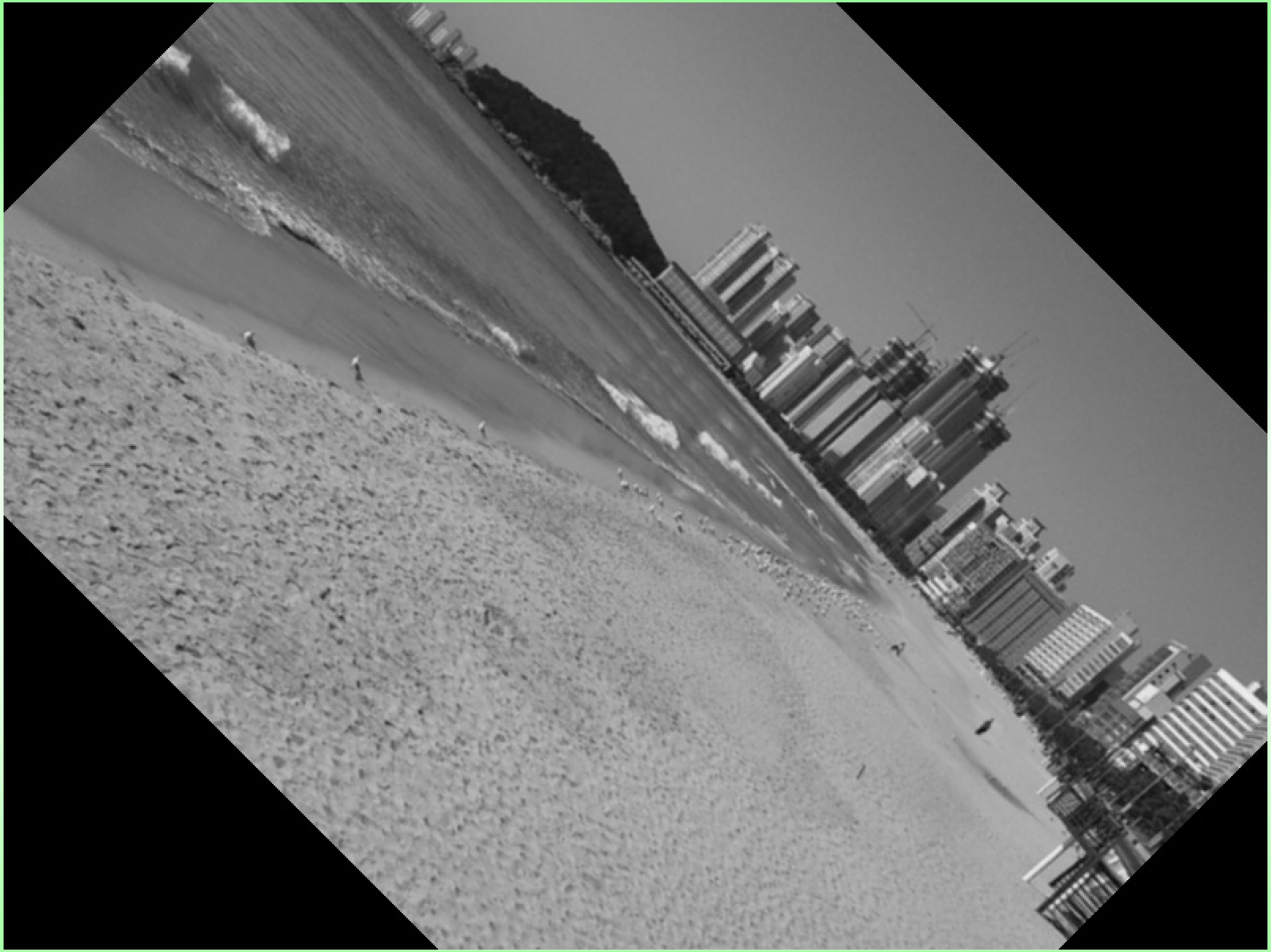
Forward mapping

Backward mapping

Homography:

z-axis rotation 10^{o}
Zoom (y-axis) 130%

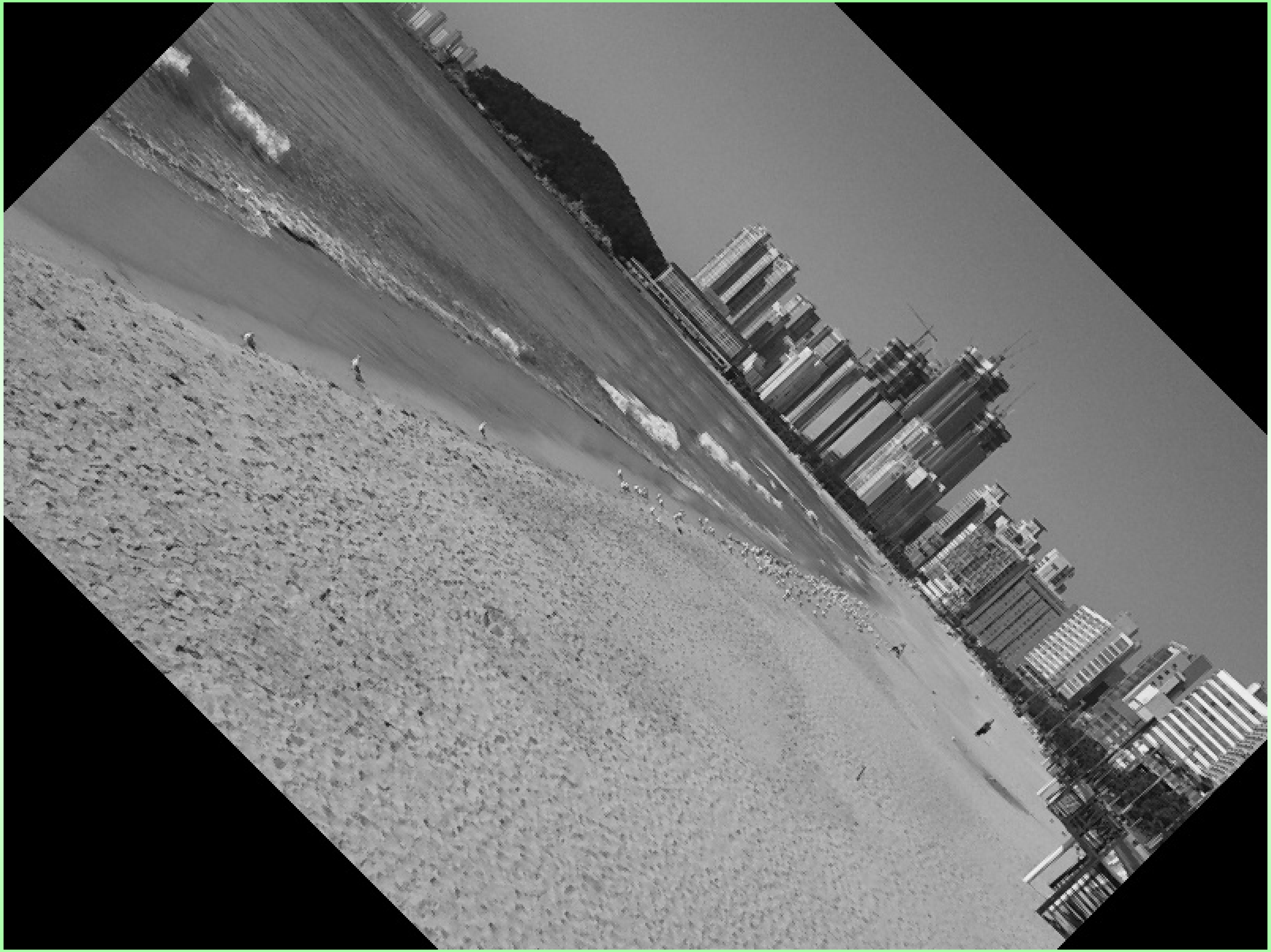
Homography: z-axis rotation 10(degree) + Scaling (y) 130% zoom



3) Forward Warping - RBF (scale 1.0, R^2=2, σ=1)

: **Gaussian 분포의 가중치로 모든 pixel을 연산하여 블러링 효과가 생기나 결함 제거됨**

: **최대반경 $\sqrt{2}$ pixel**



4) Backward Warping

: bilinear interpolation 하지 않아도 Forward Warping대비 높은 품질을 보여줌

Perspective transformation (1)

❑ Perspective transformation

- General representation (DOF = 8)

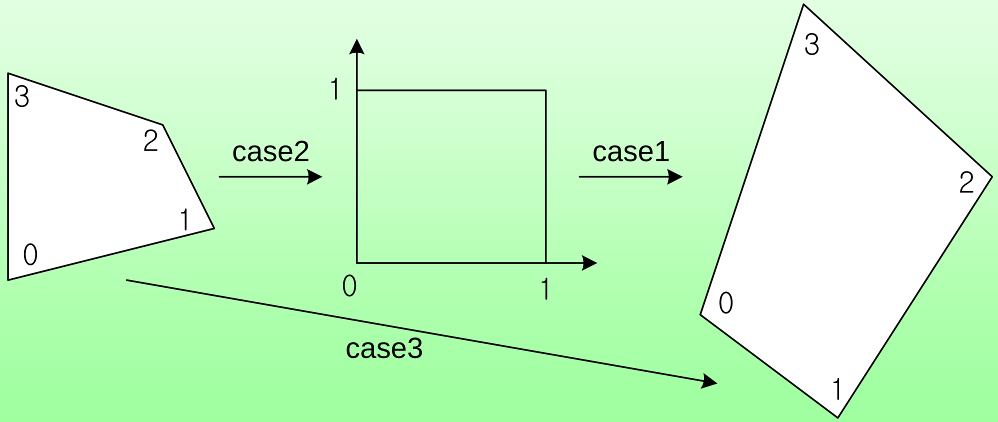
$$[x, y, w] = [u, v, w']\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

- Preserve parallel lines when are parallel to the projection plane.
- line → line
- quadrilateral→quadrilateral

Perspective transformation (2)

❑ Inferring perspective transformations

- Four points without colinearity of any three points (DOF=8)
- Fast computation
  - ➢ Case 1: square-to-quadrilateral → input (0,0), (0,1), (1,0), (1,1)
  - ➢ Case 2: quadrilateral-to-square
  - ➢ Case 3: quadrilateral-to-quadrilateral → cascade two cases



Perspective transformation (3)

- General computation by least squares psedoinverse

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \Leftrightarrow \mathbf{x}' = \mathbf{Hx}$$

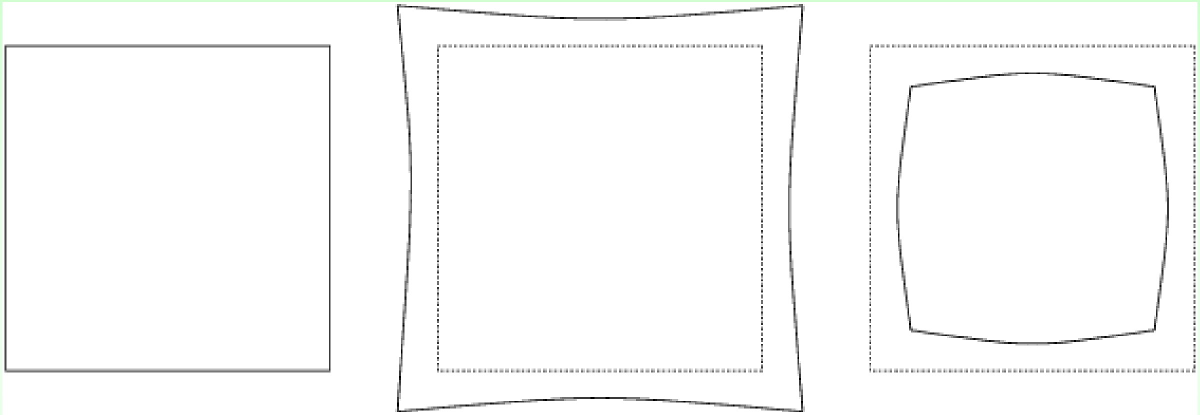
$$x' = \frac{x'_1}{x'_3} = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}}, \quad y' = \frac{x'_2}{x'_3} = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}$$

$$\begin{pmatrix} x & y & 1 & 0 & 0 & 0 & -xx' & -yx' \\ 0 & 0 & 0 & x & y & 1 & -xy' & -yy' \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix} \begin{pmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{pmatrix} = \begin{pmatrix} x' \\ y' \\ \vdots \end{pmatrix} \Leftrightarrow \mathbf{Ah} = \mathbf{b}$$

$$\mathbf{Ah} = \mathbf{b} \Leftrightarrow \mathbf{A}^{\mathrm{T}}\mathbf{Ah} = \mathbf{A}^{\mathrm{T}}\mathbf{b} \Leftrightarrow \mathbf{h} = (\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{b}$$

Polynomial transformation (1)

- Correction of non-linear geometric distortions



- Polynomial equation of spatial point (*x*,*y*)

$$u = \sum_{i=0}^{N} \sum_{j=0}^{N-i} a_{ij} x^i y^j, \quad v = \sum_{i=0}^{N} \sum_{j=0}^{N-i} b_{ij} x^i y^j$$

Polynomial transformation (2)

- N=1: Affine transformation
- N=2: second-order geometric transformation
  - 12 unknowns

$$u = \sum_{i=0}^{2} \sum_{j=0}^{2-i} a_{ij} x^i y^j, \quad v = \sum_{i=0}^{2} \sum_{j=0}^{2-i} b_{ij} x^i y^j$$

  - Sufficient for radial distortions
- How to find the coefficients
  - Least squares methods using corresponding points
  - Tie points or control grids
    - Spatial interpolation

Polynomial transformation (3)

- N=2: second-order geometric transformation
  - Need of 6 corresponding points
  - Least squares method
    - Pseudo inverse

$$\boxed{\mathbf{W}^T \mathbf{u} = \mathbf{W}^T \mathbf{W}\, \mathbf{a}}$$

$$\begin{bmatrix} \sum u \\ \sum xu \\ \sum yu \\ \sum xyu \\ \sum x^2u \\ \sum y^2u \end{bmatrix} = \begin{bmatrix} M & \sum x & \sum y & \sum xy & \sum x^2 & \sum y^2 \\ \sum x & \sum x^2 & \sum xy & \sum x^2y & \sum x^3 & \sum xy^2 \\ \sum y & \sum xy & \sum y^2 & \sum xy^2 & \sum x^2y & \sum y^3 \\ \sum xy & \sum x^2y & \sum xy^2 & \sum x^2y^2 & \sum x^3y & \sum xy^3 \\ \sum x^2 & \sum x^3 & \sum x^2y & \sum x^3y & \sum x^4 & \sum x^2y^2 \\ \sum y^2 & \sum xy^2 & \sum y^3 & \sum xy^3 & \sum x^2y^2 & \sum y^4 \end{bmatrix} \begin{bmatrix} a_{00} \\ a_{10} \\ a_{01} \\ a_{11} \\ a_{20} \\ a_{02} \end{bmatrix}$$