

## Image Zooming

### 1) Reducing

- new value is weighted sum of nearest neighbours
- new value equals nearest neighbour

### 2) Enlarging

- new value is weighted sum of nearest neighbours
- add noise to obscure pixelation

## Image resampling problem

When doing a rotation with a sine or cosine, an integer value is not always produced, so the output pixel may be non-integer. How do we solve this?

- apply the transformation in reverse, transform the image then take the nearest neighbour values of the source image (before transformation)
- **nearest neighbour interpolation**, simply round  $(x, y)$  to the nearest neighbour (nearest integer pixel) value and use this as the output
- **bilinear interpolation**, takes a weighted sum of the four nearest neighbours to  $(x, y)$ . The values of the four neighbours are weighted inversely by their distance from  $(x, y)$ . Needs more **computation** than simple **nearest neighbour interpolation** but results in a more visually appealing image where jagged effects are reduced.