Lab 3
CMPUT 206
Image Processing

Color and Geometric Operations

Bernal Manzanilla Saavedra
Prof. Nilanjan Ray

Color and Geometric operations

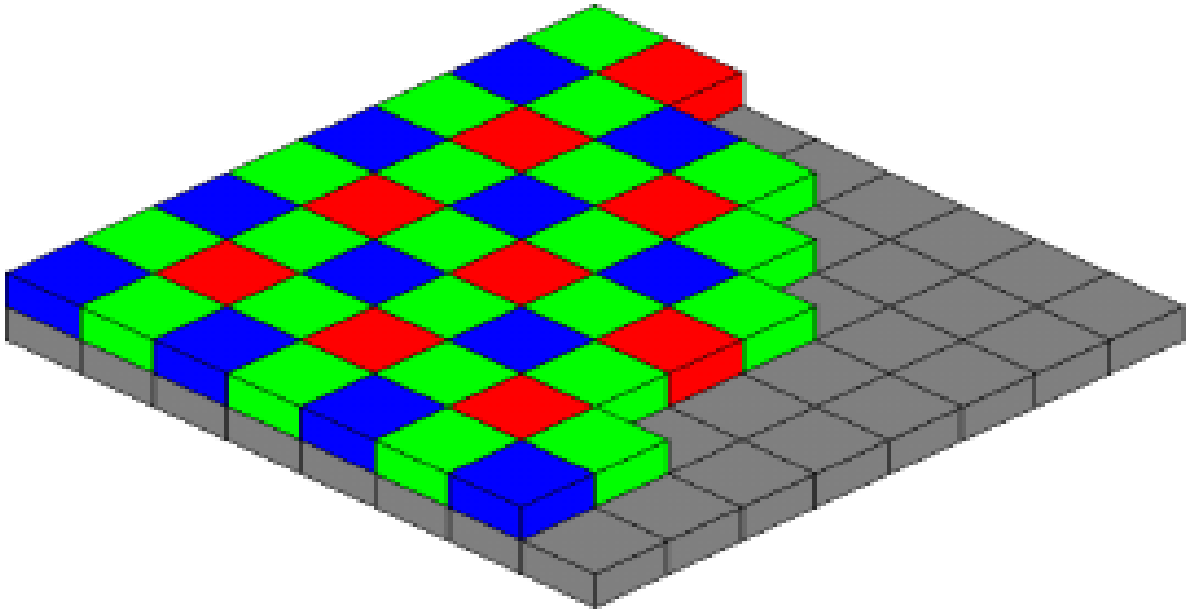
- Demosaicing of an image with the Bayer filter (25%)
- Apply the Floyd-Steinberg dithering algorithm (25%)
- Geometric Transformations (25%)
- Merge two images: registration(25%)

Geometric operations

- Demosaicing of an image with the Bayer filter (25%)
- Apply the Floyd-Steinberg dithering algorithm (25%)

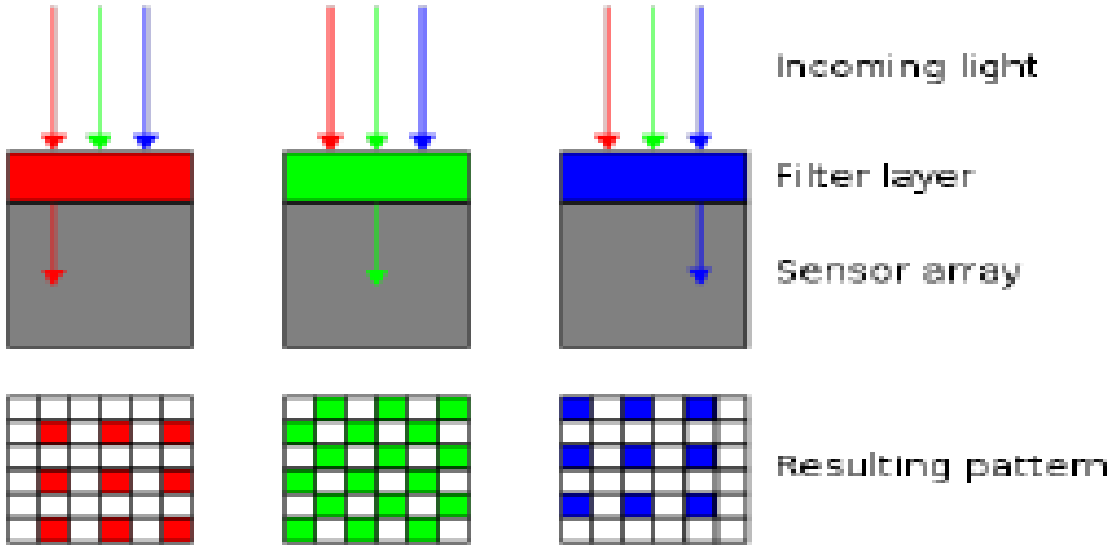
Bayer Filter Mosaic (cont'd)

Bayer arrangement of color filters on the pixel array of an image sensor



Gray boxes indicate individual sensor elements and the overlaid RGB boxes indicate the channel whose information each sensor element captures.

Profile/cross-section of sensor



Colored boxes in the resulting pattern indicate pixels where respective color information is captured; color information at white boxes needs to be reconstructed

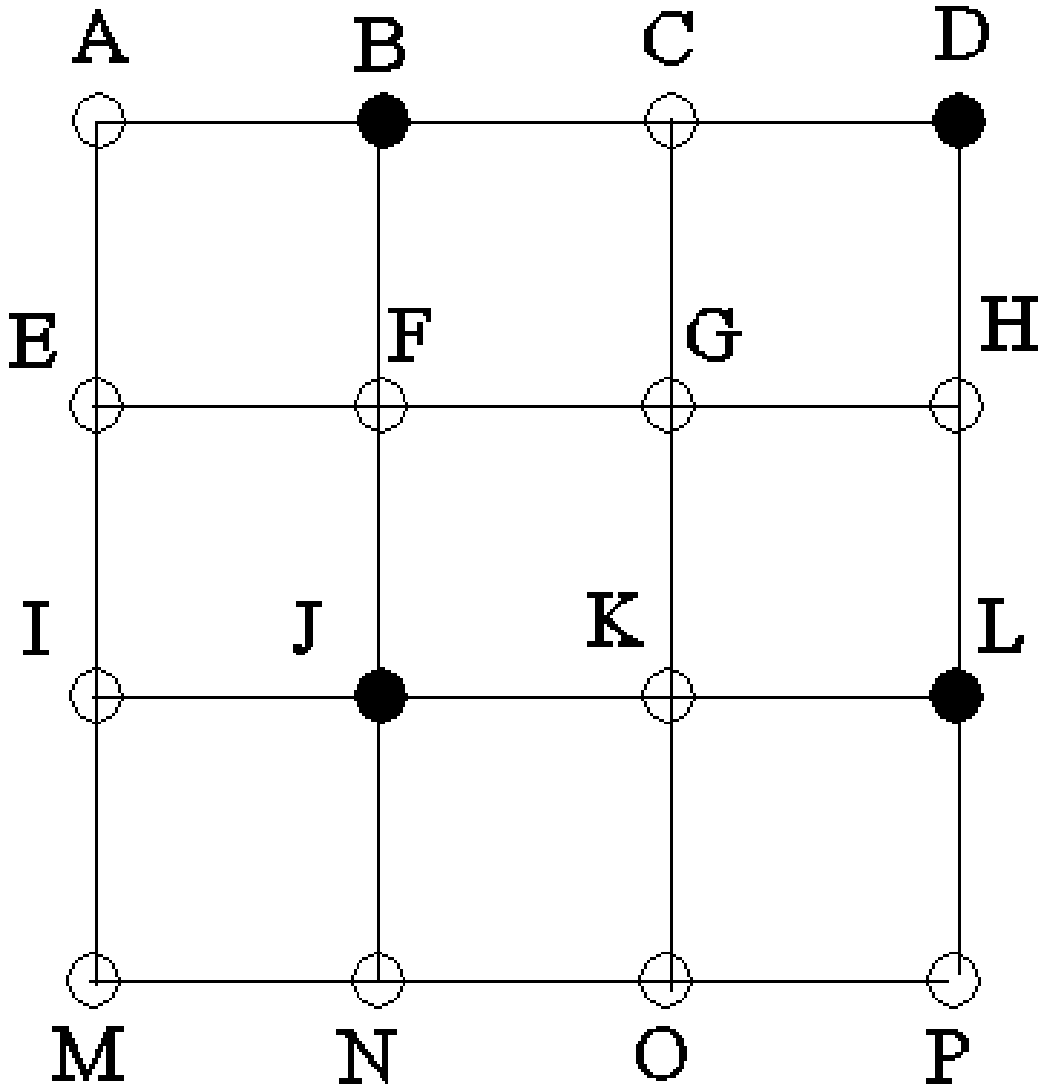
Demosaicing Rules

- The channel masks are moved over the Bayer pattern image like a sliding window but with no overlap
- Locations where the mask is 1 (shaded)
  - value is directly copied from the Bayer image.
- Locations where the mask is 0 (unshaded)
  - value is the average of neighboring existing pixels
  - no. of neighboring pixels to average may be 2 (above/below or right/left) or 4 (all diagonal corners)
  - if a row/column in the mask is entirely empty, values are copied from neighboring non empty row/column

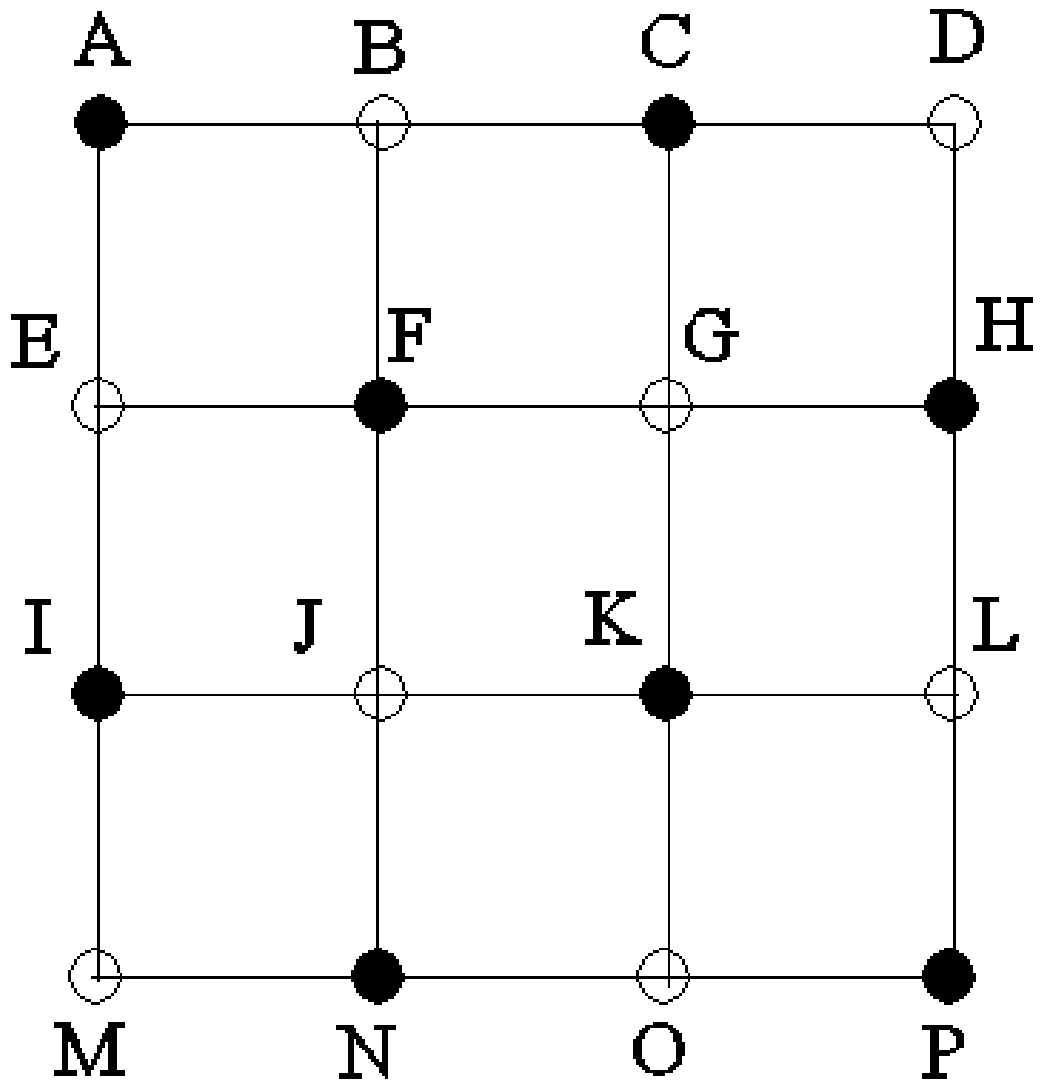
Bayer Filtering (INPUT)

Bayer Filtering (FINAL)

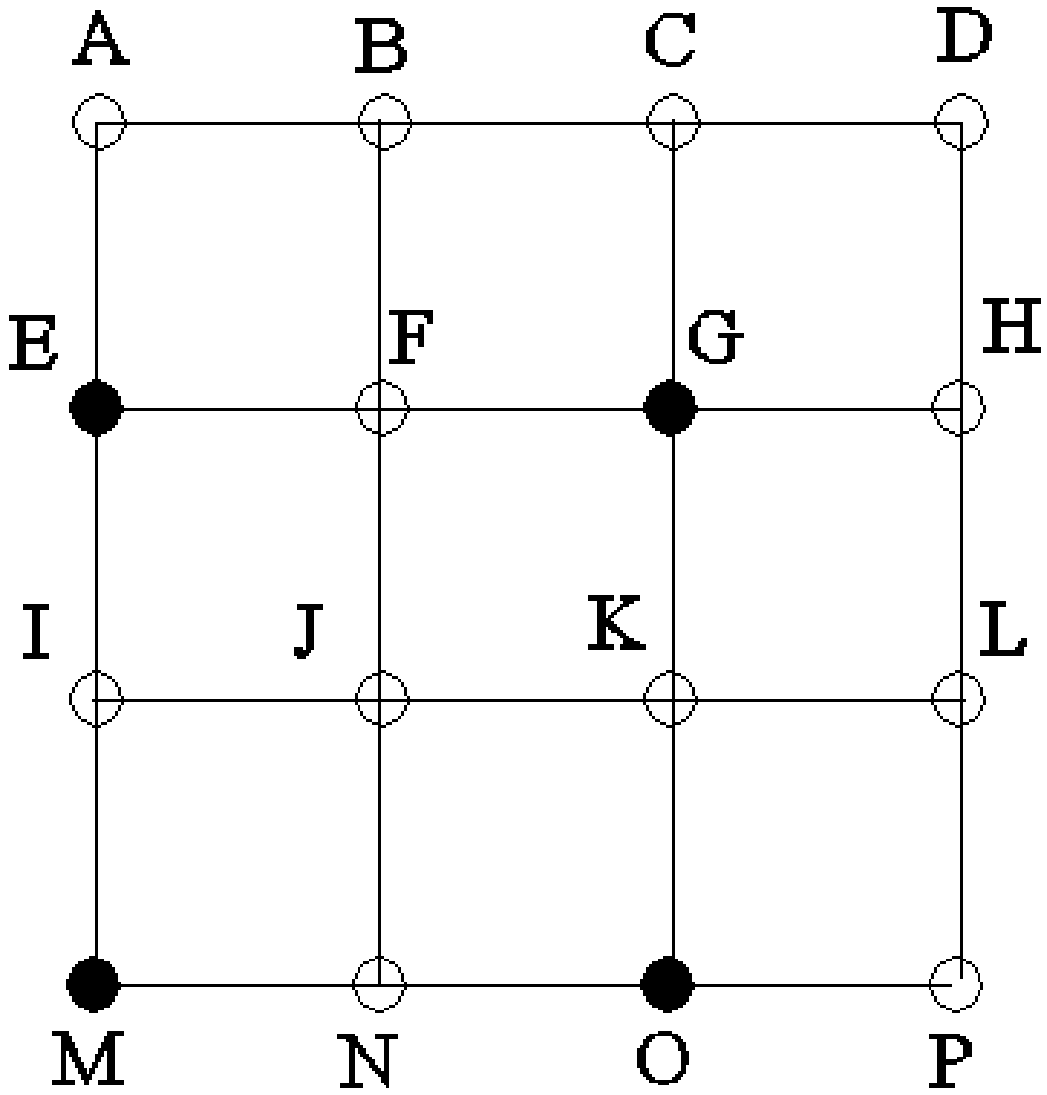
Bayer Filtering (RED MASK)



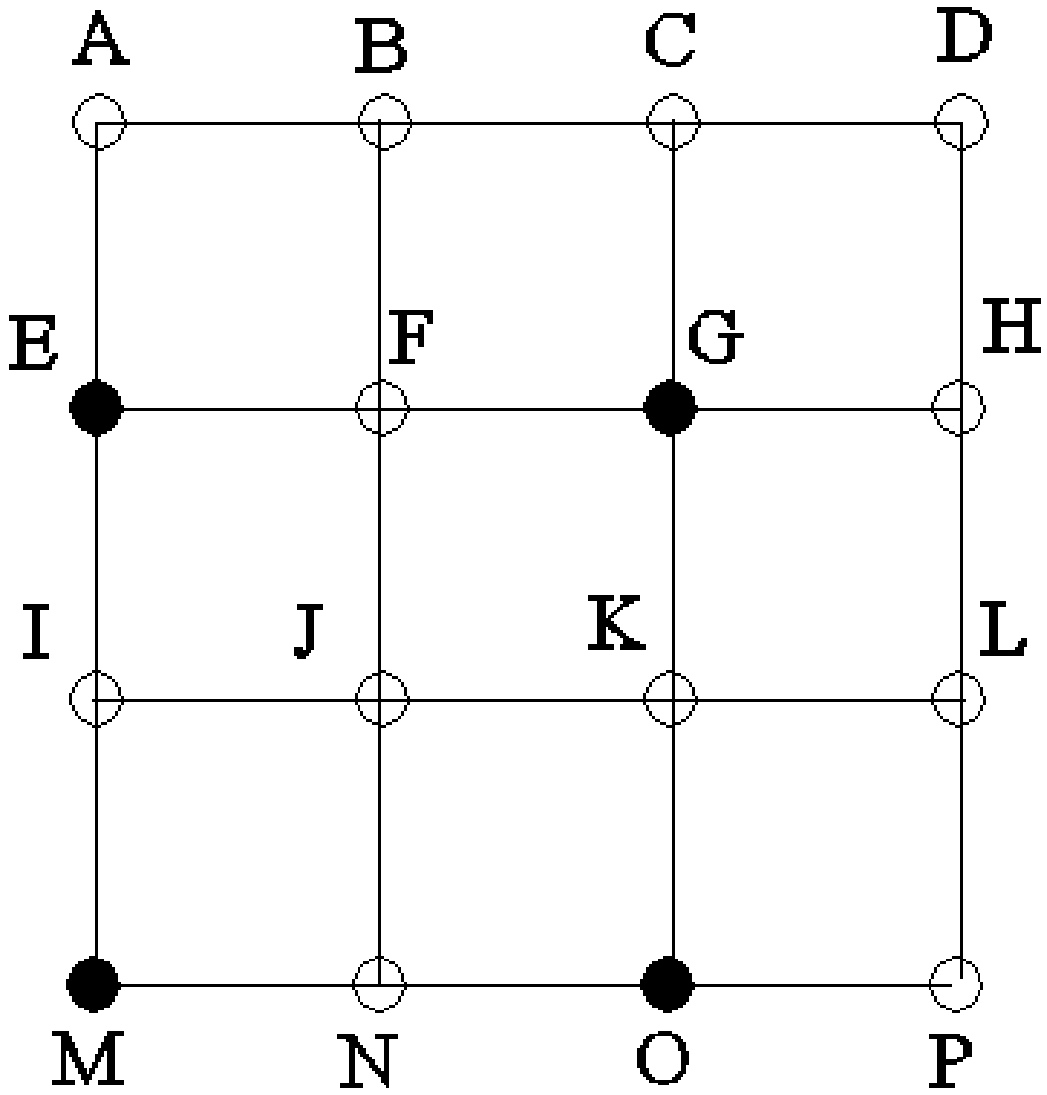
Bayer Filtering (GREEN MASK)



Bayer Filtering (BLUE MASK)

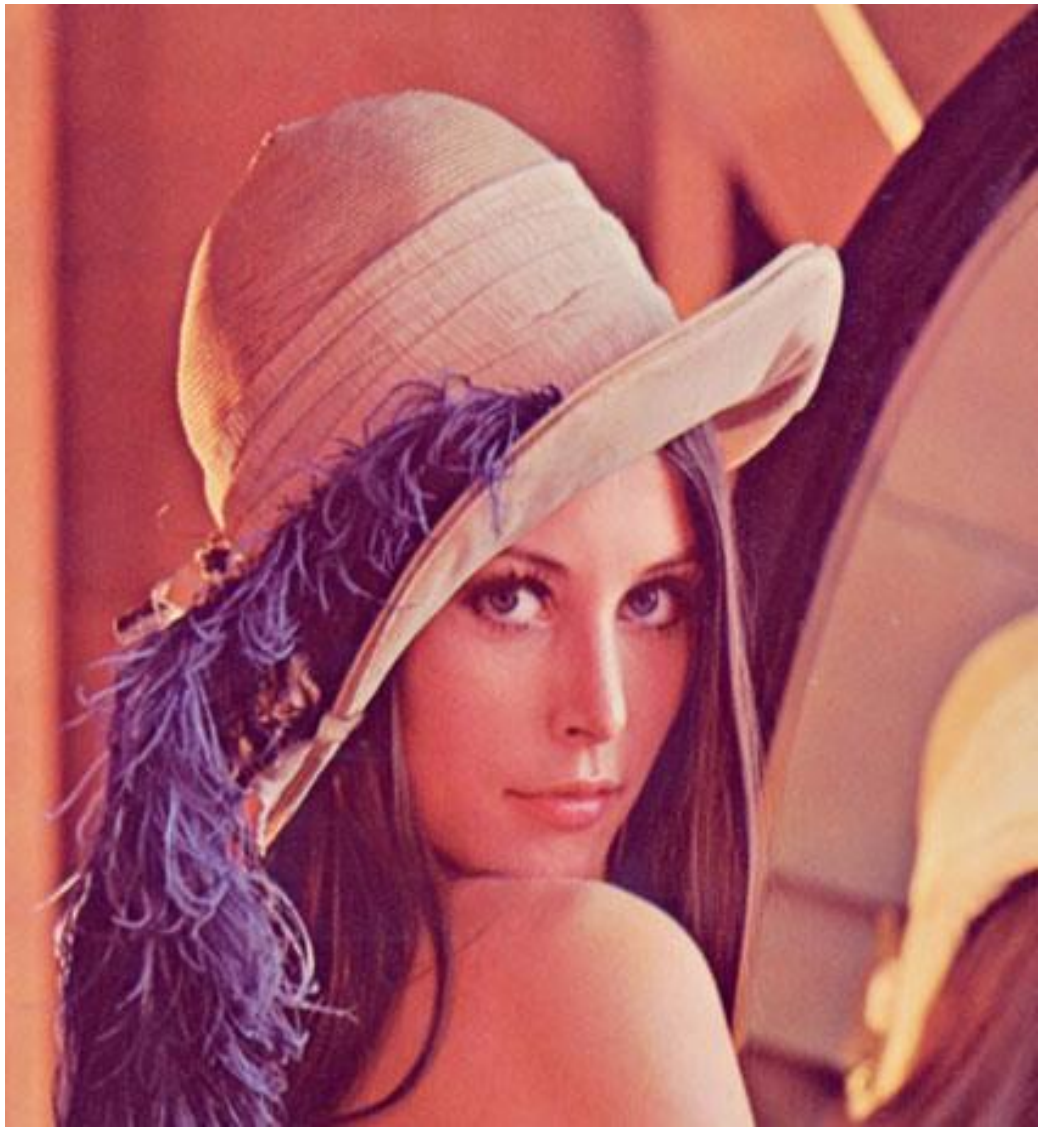


Bayer Filtering (BLUE MASK)



Dithering:

Dynamic range reduction and compression by pixel depth reduction



Dithering:

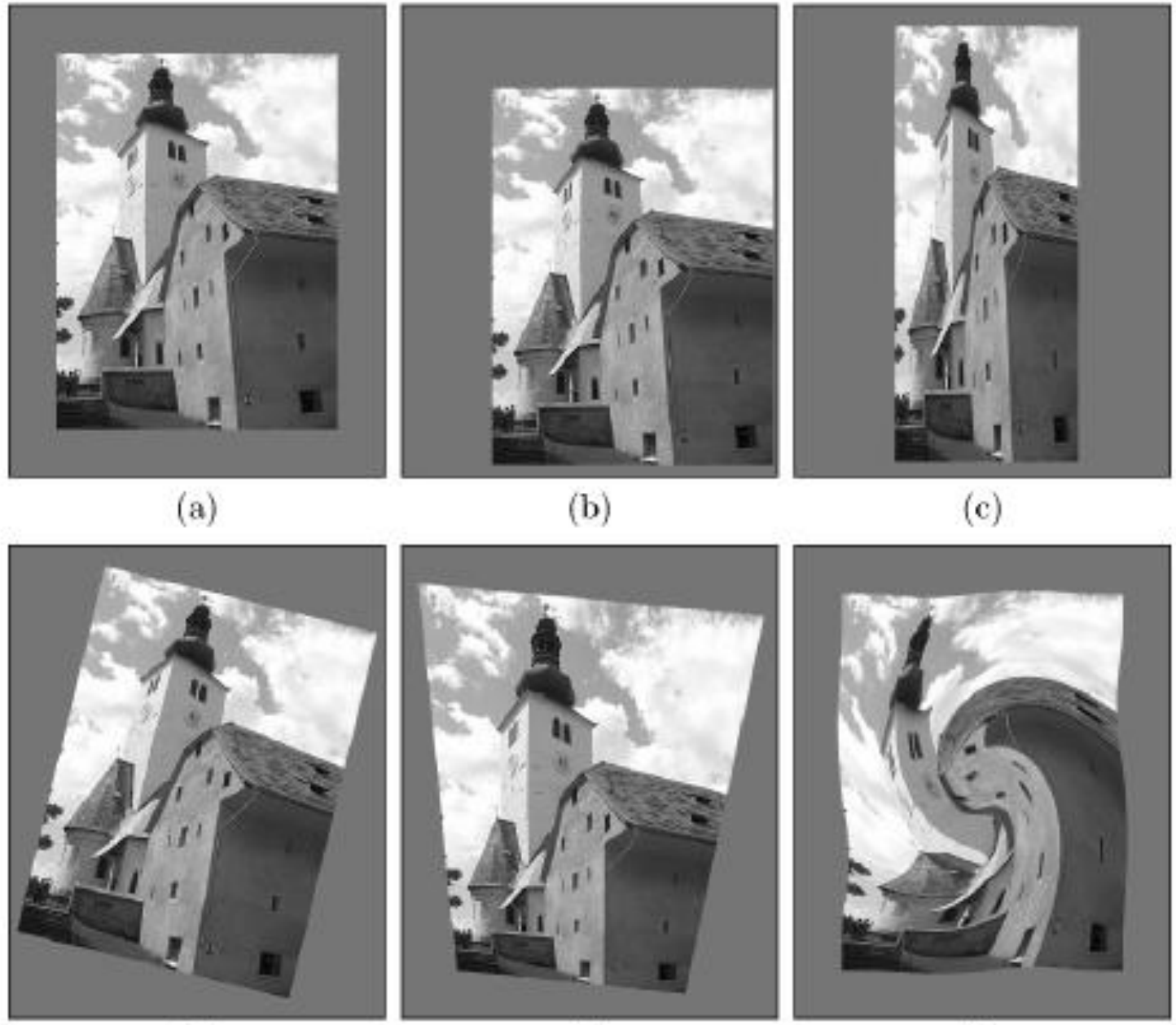
Dynamic range reduction and compression by pixel depth reduction

Floyd-Steinberg Dithering Algorithm

```
for each y from top to bottom do
    for each x from left to right do
        oldpixel := pixel[x][y]
        newpixel := find_closest_palette_color(oldpixel)
        pixel[x][y] := newpixel
        quant_error := oldpixel - newpixel
        pixel[x + 1][y    ] := pixel[x + 1][y    ] + quant_error × 7 / 16
        pixel[x - 1][y + 1] := pixel[x - 1][y + 1] + quant_error × 3 / 16
        pixel[x    ][y + 1] := pixel[x    ][y + 1] + quant_error × 5 / 16
        pixel[x + 1][y + 1] := pixel[x + 1][y + 1] + quant_error × 1 / 16
```

Geometric operations

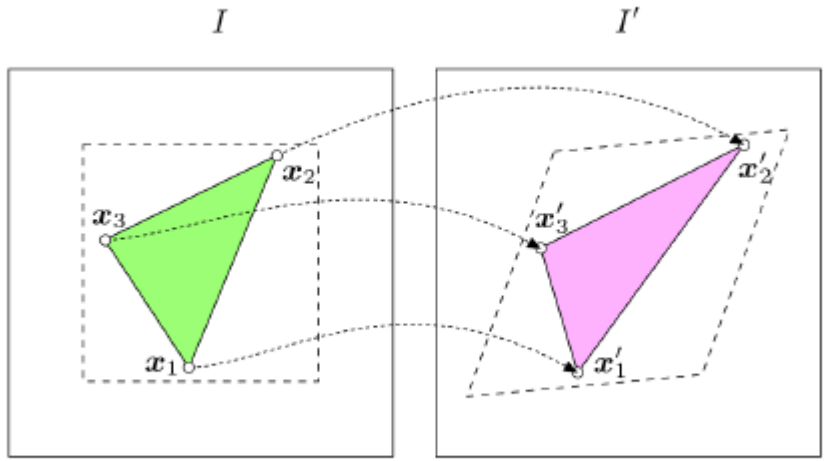
- Geometric Transformations (25%)
- Merge two images: registration(25%)


(a)
(b)
(c)
(d)
(e)
(f)

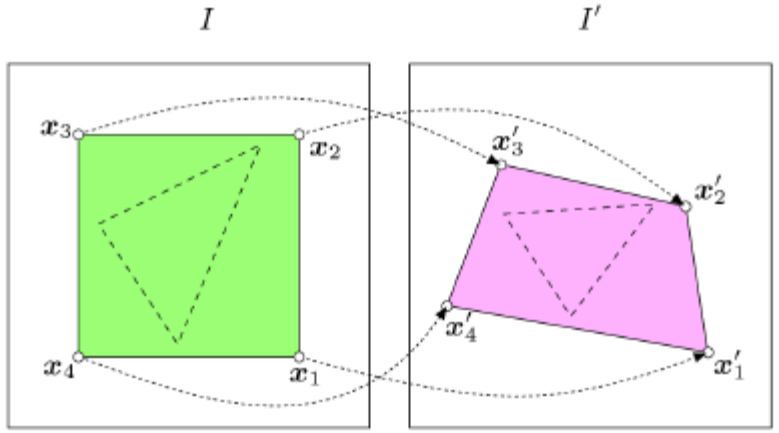
Transformations

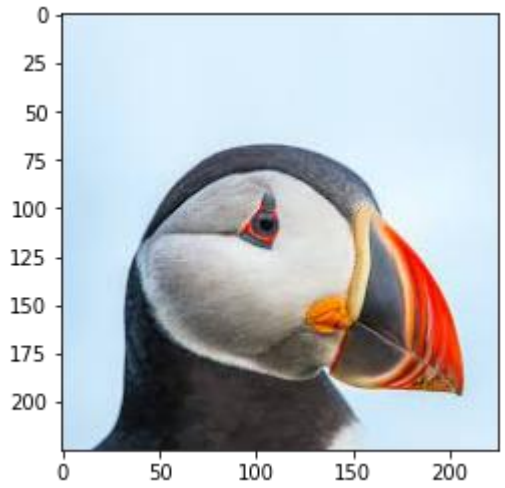
Scaling and Rotation: Writable as Matrix-vector product

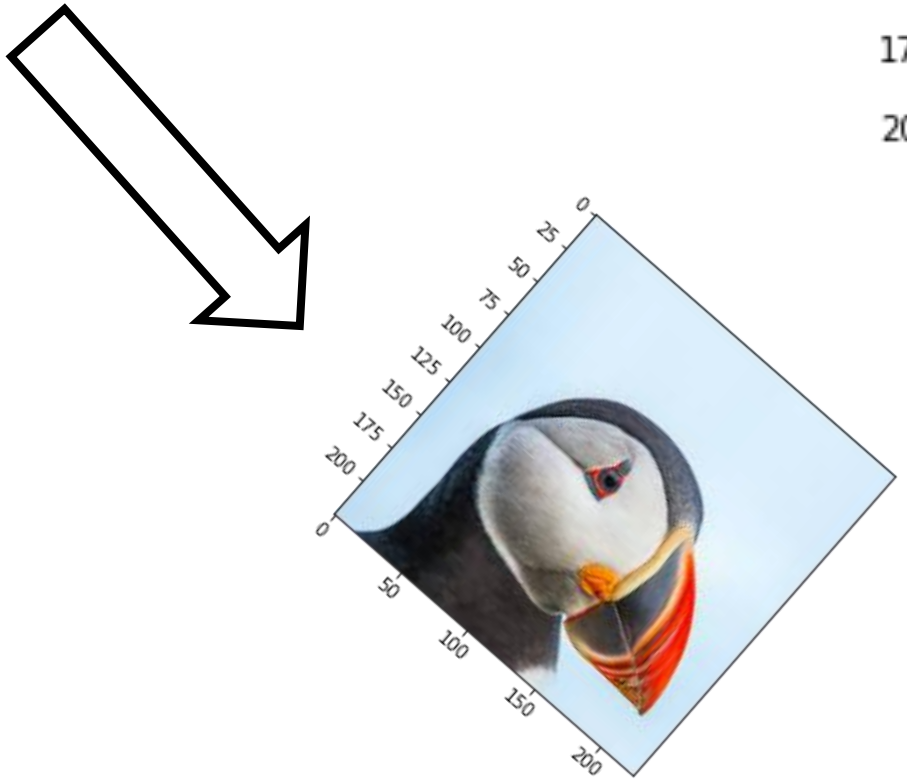
Affine (in the image plane)



Projective (perpendicular to the image plane)




0
25
50
75
100
125
150
175
200
0
50
100
150
200


0
25
50
75
100
125
150
175
200
0
50
100
150
200

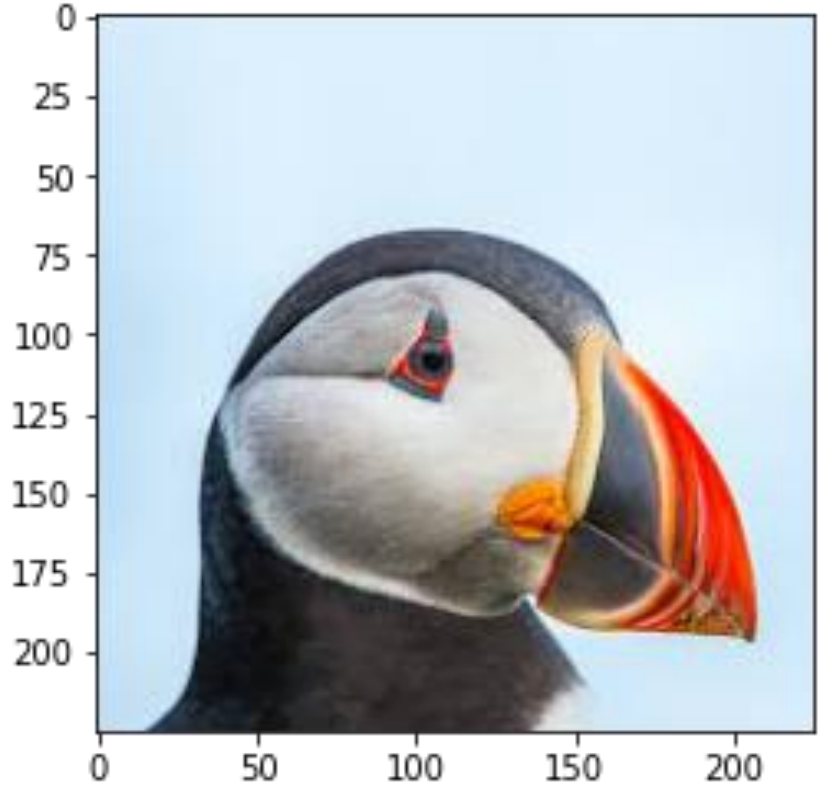
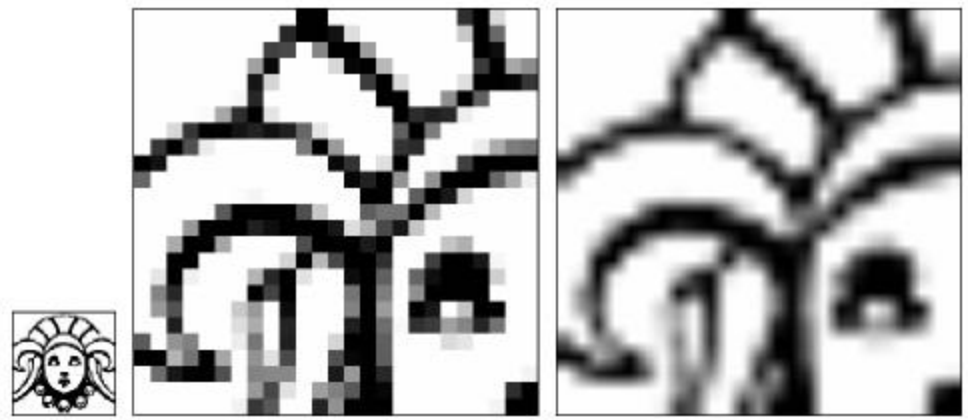

0
25
50
75
100
125
150
175
200
0
50
100
150
200

Image Enlargement



(a) Original image (b) 8x magnification by nearest neighbor method
(c) bilinear interpolation

Simple Mappings

- Translation (shift) by a vector (d_x, d_y):

$$\begin{aligned} T_x &: x' = x + d_x \\ T_y &: y' = y + d_y \end{aligned} \quad \text{or} \quad \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} d_x \\ d_y \end{pmatrix}$$

- Scaling along the *x* or *y* axis by the factor s_x or s_y, respectively:

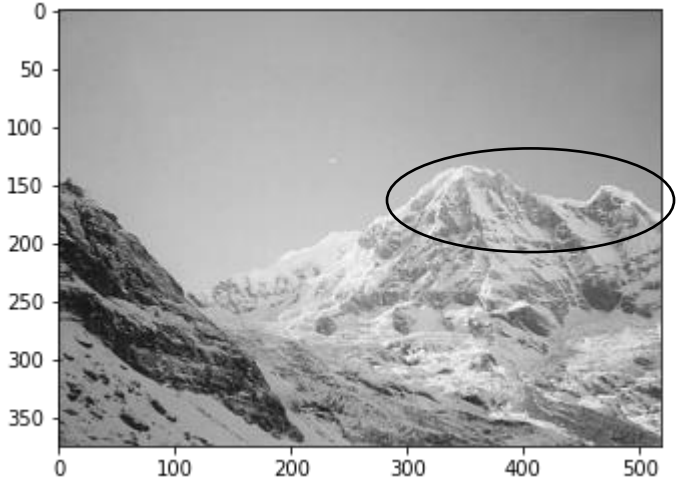
$$\begin{aligned} T_x &: x' = s_x \cdot x \\ T_y &: y' = s_y \cdot y \end{aligned} \quad \text{or} \quad \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

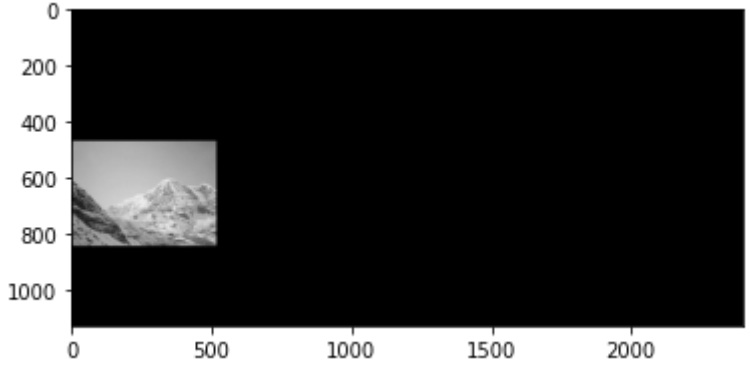
- Shearing along the *x* and *y* axis by the factor b_x and b_y, respectively:

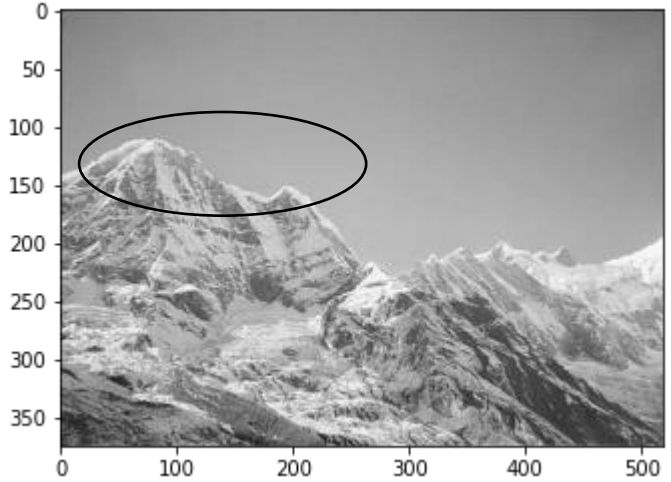
$$\begin{aligned} T_x &: x' = x + b_x \cdot y \\ T_y &: y' = y + b_y \cdot x \end{aligned} \quad \text{or} \quad \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & b_x \\ b_y & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

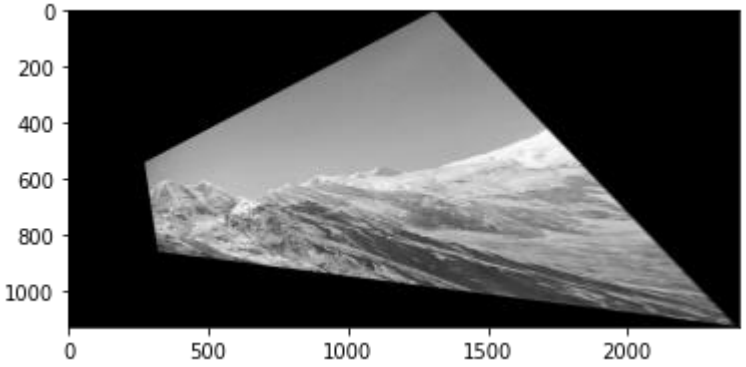
- Rotation by an angle α (coordinate origin is the center of rotation):

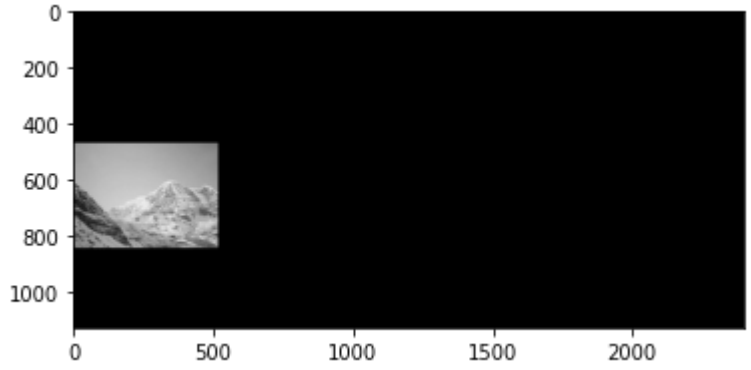
$$\begin{aligned} T_x &: x' = x \cdot \cos\alpha - y \cdot \sin\alpha \\ T_y &: y' = x \cdot \sin\alpha + y \cdot \cos\alpha \end{aligned} \qquad \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

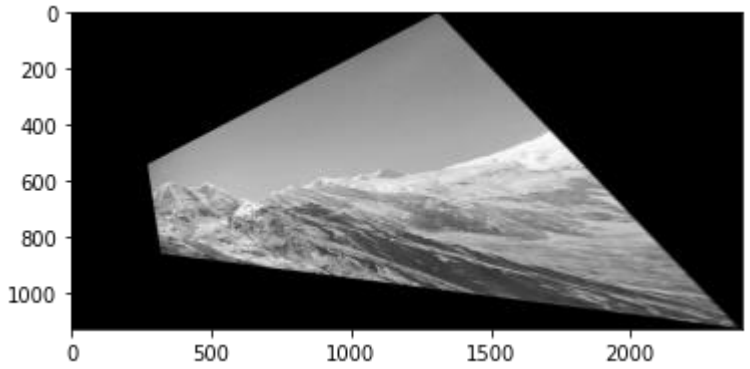

0
50
100
150
200
250
300
350
0
100
200
300
400
500

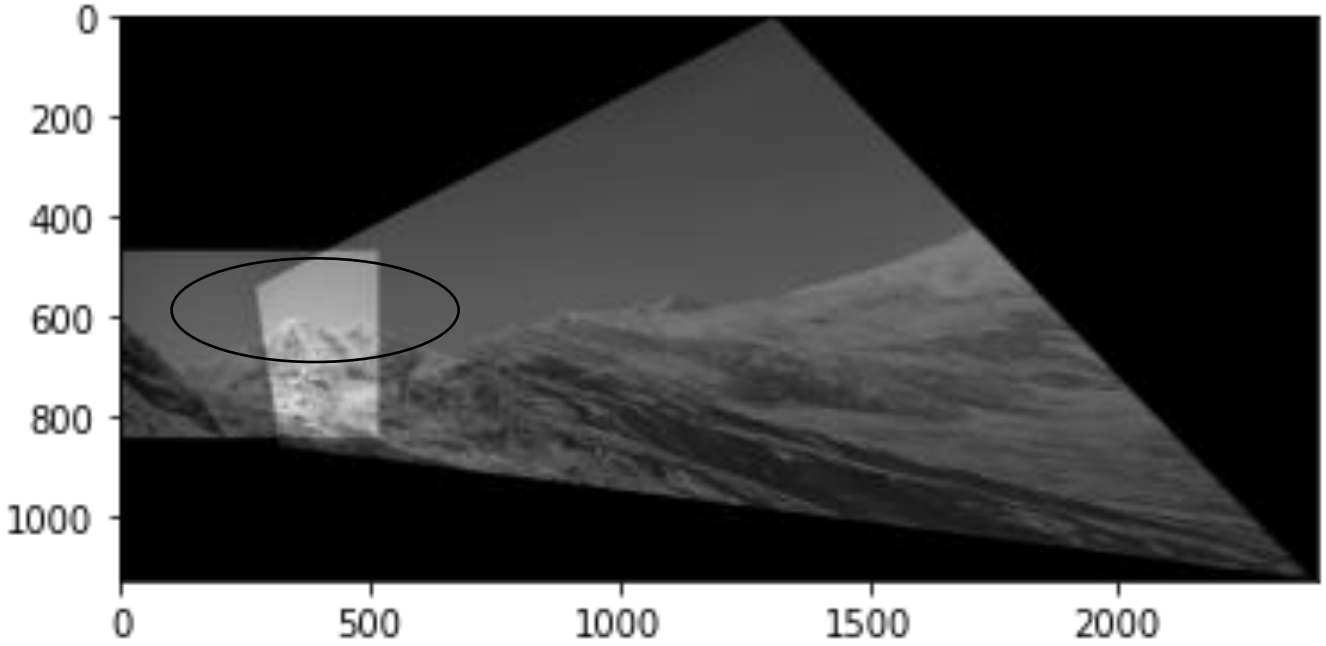

0
200
400
600
800
1000
0
500
1000
1500
2000


0
50
100
150
200
250
300
350
0
100
200
300
400
500


0
200
400
600
800
1000
0
500
1000
1500
2000


0
200
400
600
800
1000
0
500
1000
1500
2000


0
200
400
600
800
1000
0
500
1000
1500
2000


0
200
400
600
800
1000
0
500
1000
1500
2000