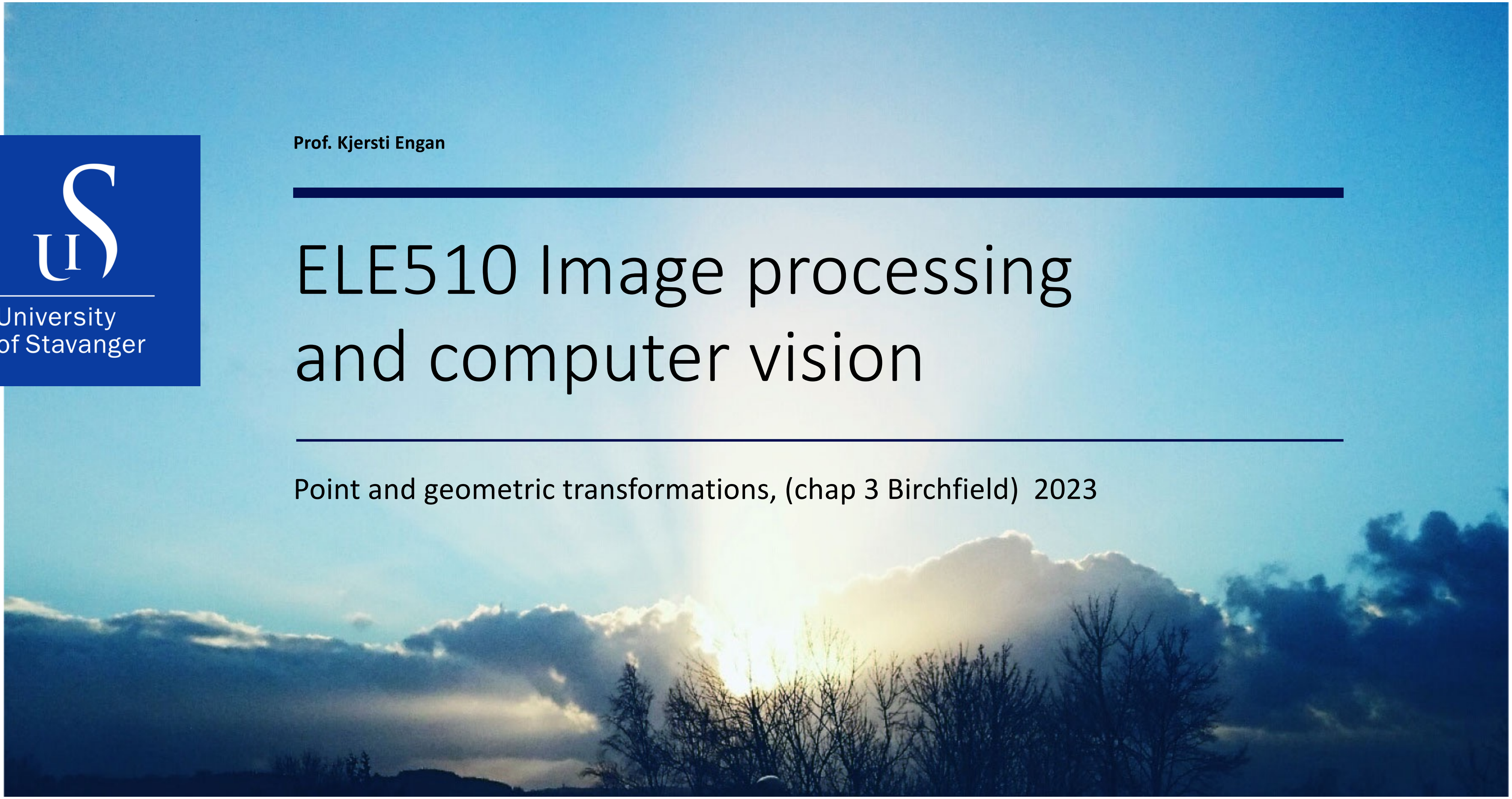


Prof. Kjersti Engan

ELE510 Image processing and computer vision

Point and geometric transformations, (chap 3 Birchfield) 2023

University of Stavanger

Geometric and point transformation (chap 3)

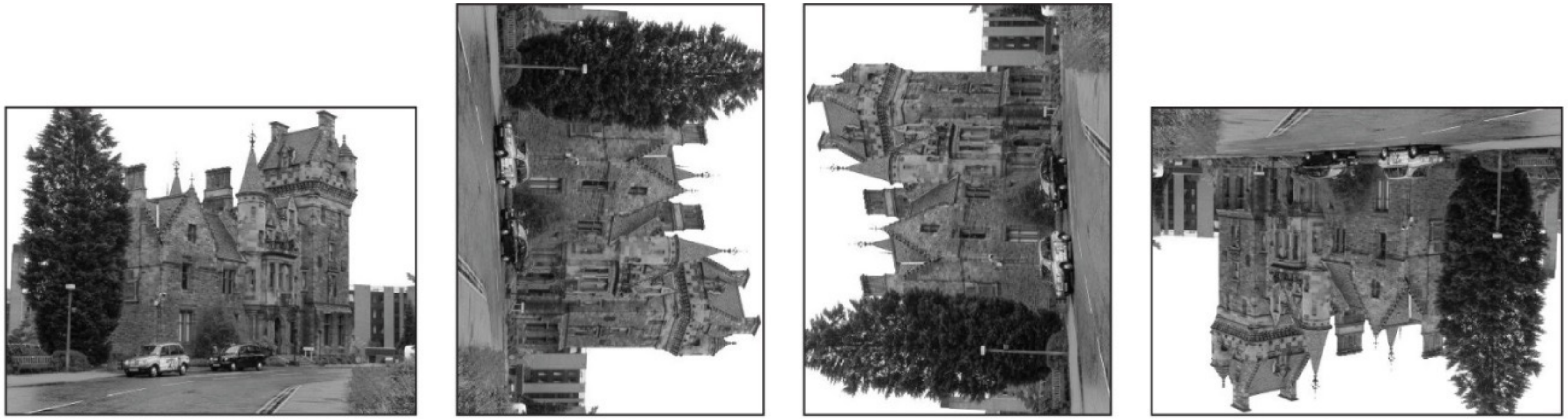
From last week:

- Geometric transformations:
  - changing a pixels location without changing its value

- Point transformations:
  - change a pixels value without changing the location
  - Independent of pixel coordinates and values of other pixels

Rotating by a multiple of 90 degrees

Figure 3.3 An image rotated by 0, +90, −90, and 180 degrees.

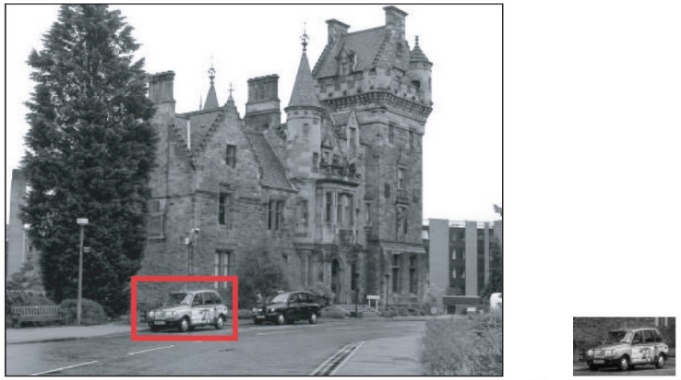


0°   +90°   −90°   180°

Stan Birchfield

Cropping an Image

Figure 3.5 An image and an automobile cropped out of the region of the image indicated by the red rectangle.



Image

Cropped region

Stan Birchfield

ALGORITHM 3.4 Crop an image

CropImage(I, $left$, top, $right$, $bottom$)

Input: image I, rectangle with corners $(left, top)$ and $(right-1, bottom-1)$
Output: cropped image I' of size $new\text{-}width \times new\text{-}height$

$new\text{-}width \leftarrow right - left$
$new\text{-}height \leftarrow bottom - top$
$I' \leftarrow$ AllocateImage($new\text{-}width$, $new\text{-}height$)
for $(x', y') \in I'$ **do**
  $I'(x', y') \leftarrow I(x' + left, y' + top)$
return I'

Downsampling and upsampling

- **Downsample** an image to produce a smaller image than the original (should be smoothed first):

$$I'(x, y) = I(2x, 2y) \qquad \text{(downsample by two)}$$

- **Upsample** an image to produce a larger image than the original (interpolation should be done):

$$I'(x, y) = I\left(\left\lfloor \frac{x}{2} \right\rfloor, \left\lfloor \frac{y}{2} \right\rfloor\right) \qquad \text{(upsample by two)}$$

Figure 3.6 LEFT: An image and the result of downsampling by a factor of 2 and 4, respectively, in each direction. RIGHT: A cropped region and the result of upsampling by a factor of 2 and 4, respectively, in each direction.

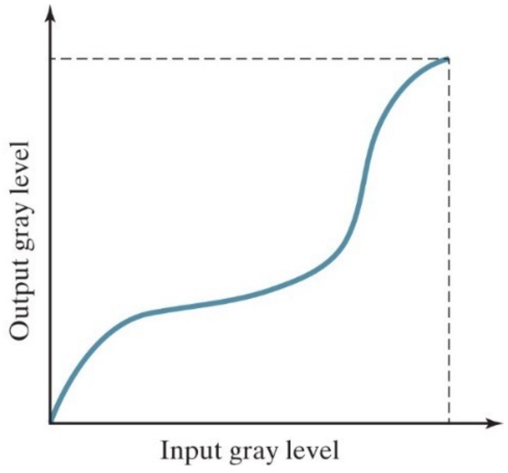
(3.2) Point transformations

- changes a pixel's value without changing its location.
- Independent of pixels coordinate and coordinates and values of other pixels
  - ✓Special case of spatial domain filtering where the constrain on values of neighbor pixels is removed.

- Gray level transformations
- Gray level histograms

$$I'(x, y) = f(I(x, y))$$

Figure 3.7 A graylevel transformation maps input gray levels to output gray levels. Based on http://www.unit.eu/cours/videocommunication/Point_Transformation_histogram.pdf

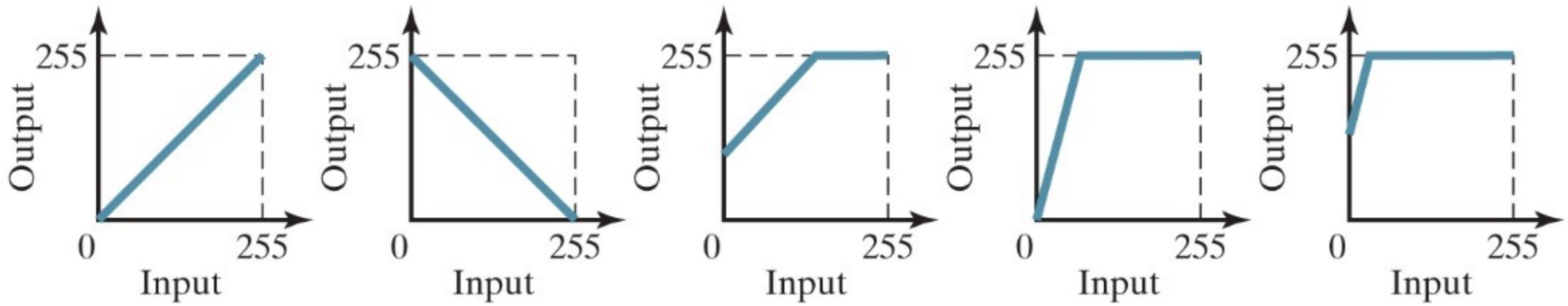


Arithmetic operations

- A useful class of gray level transformations is the set of arithmetic operations; inversion, addition (bias) , multiplication (gain)
- If we want a grayscale image out with $v \in [0,255]$ we need to use saturation arithmetic:

$I'(x,y) = min(I(x,y) + b, 255)$ example, addition (bias=b)

Figure 3.8 Arithmetic graylevel transformations. From left to right: identity, inversion, addition (bias), multiplication (gain), and gain-bias transformation, where saturation arithmetic prevents the output from exceeding the valid range. Note that the slope remains 1 under addition, while the mapping passes through the origin under multiplication.

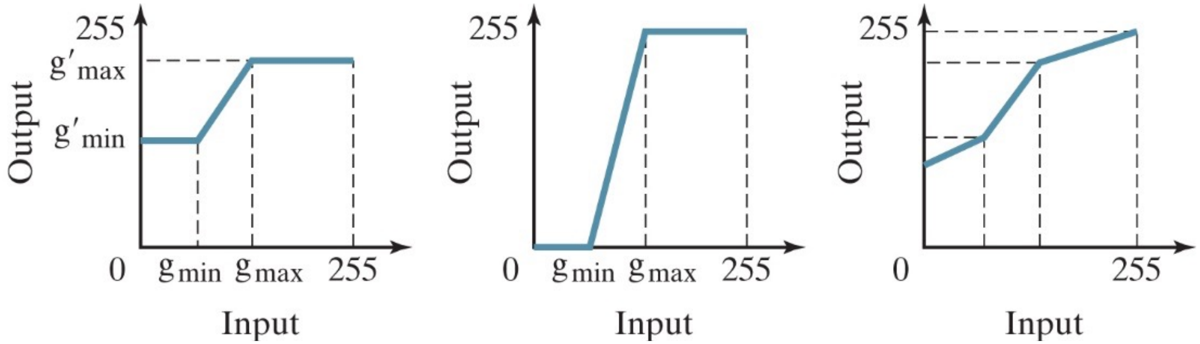


Linear contrast stretching

- **Linear contrast stretch:** A transformation that specifies a line segment that maps gray levels between g_{min} and g_{max} in the input image to the gray levels g'_{min} and g'_{max} in the output image according to a linear function:

$$I'(x,y) = \frac{g'_{max} - g'_{min}}{g_{max} - g_{min}}(I(x,y) - g_{min}) + g'_{min}$$

- Piecewise linear contrast stretch can model any gray level transformation given enough line segments



Analytic transformations

- Graylevel transformations can be specified using analytic functions such as the logarithm, exponential, or power functions:

$$I'(x,y) = \log(I(x,y))$$

$$I'(x,y) = \exp(I(x,y))$$

$$I'(x,y) = (I(x,y))^{\gamma}$$

- Rounding and clamping to [0,255] is needed to get a 8 bpp image. This is always the case but often left out for clarity. Might use images as matrices with real numbers, and perform multiple operations / filtering etc. Rounding/clamping is done on *final* image

Thresholding

- Takes a grayscale image and sets every output pixel to 1 if its input gray level is above a certain threshold, or to 0 otherwise:

$$I'(x,y) = \begin{cases} 1 & \text{if } I(x,y) > \tau \\ 0 & \text{otherwise} \end{cases}$$

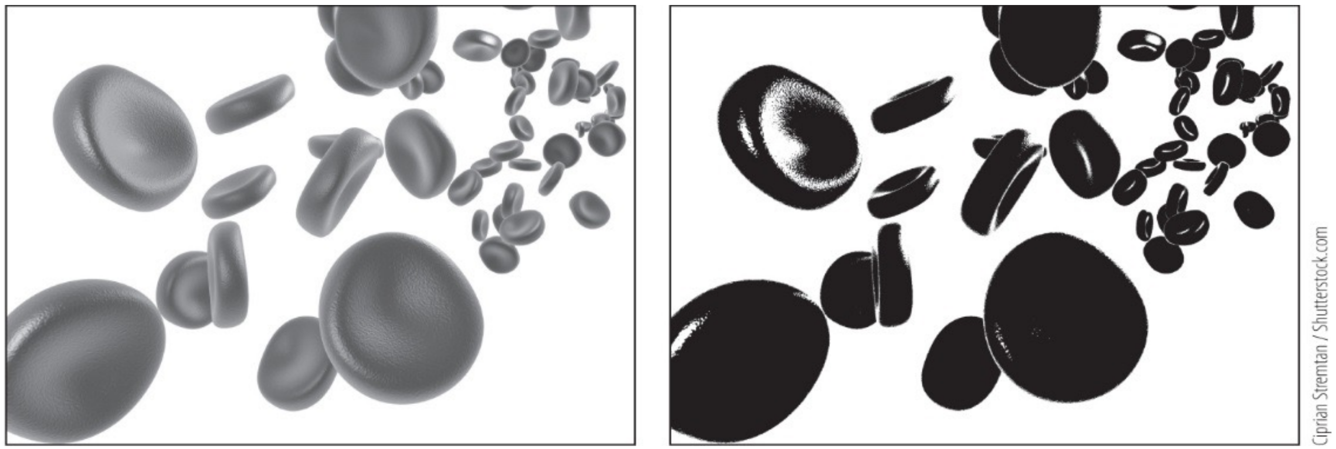
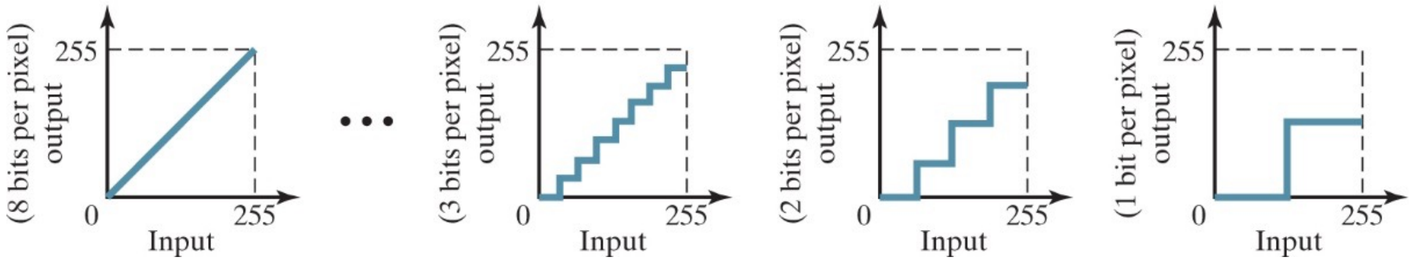


Figure 3.14 An 8-bit grayscale image (left), and the binarized result obtained by thresholding with $\tau = 150$ (right).

- **Density slicing**: assigns all gray levels within a certain range to a certain value. Others can be unaltered, set to zero etc.

- **Quantization**: discards one or more of the lower-order bits. How many bits per pixel (bpp)?

Figure 3.16 Quantization discards the lower-order bits via a staircase function, with the number of stairs determined by the number of bits retained. From right to left: Only 1 bit is retained, so the gray levels in the dark half (less than 128) map to 0, while the gray levels in the bright half (above 127) map to 128 (binary: 10000000); 2 bits are retained, so gray levels are mapped to either 0, 64, 128, or 192; 3 bits are retained, so all gray levels are mapped to either 0, 32, 64, 96, 128, 160, 192, or 224; all 8 bits are retained (no quantization, the identity function).



Geometric and point operations (3.1-3.2)

Topic in three points:

1. What defines geometric operations on digital images?
   - ✓ pixelsvalues are the same, position changes
2. What defines point operations on digital images?
   - ✓ Position is the same, but value change indepenent on other positions values.
3. Some examples of operators
   - ✓ Geometric: flop, flop, crop, .. Point: arithmetic, contrast stretching thresholding,