

Computer Vision I

Bjoern Andres

Machine Learning for Computer Vision
TU Dresden



<https://mlcv.cs.tu-dresden.de/courses/26-winter/cv1/>

Winter Term 2026/2027

Contents.

- ▶ This part of the course introduces the notion of finite and infinite digital images.
- ▶ It introduces some operators on digital images, including point operators, linear operators and non-linear operators.

Digital images

Definition 1. For any $n_0, n_1 \in \mathbb{N}$ and any $C \neq \emptyset$, a map $f \in C^{n_0 \times n_1}$ is called a **digital image**, and a map $f \in C^{\mathbb{Z} \times \mathbb{Z}}$ is called an **infinite digital image**.

In both cases, n_0, n_1 are called the **width** and **height** of the image, and C is called its **color set**. The elements of $n_0 \times n_1$ are called the **pixels** of the image.

The graph $G = (V, E)$ with $V = n_0 \times n_1$ and such that $\forall r, r' \in V: \{r, r'\} \in E \Leftrightarrow \|r - r'\| = 1$ is called its **pixel grid graph**.

Examples.

Gray levels	$C = \{0, \dots, 255\}$
RGB colors	$C = \{0, \dots, 255\}^3$
Real numbers	E.g. $C = \mathbb{R}$ or $C = [0, 1]$
Real tuples	E.g. $C = \mathbb{R}^n$ or $C = [0, 1]^n$

Point operator

Definition 2. For any $n_0, n_1 \in \mathbb{N}$ and any set $C \neq \emptyset$, a **point operator** on digital images of width n_0 , height n_1 and with color set C is a function

$$\varphi: C^{n_0 \times n_1} \rightarrow C^{n_0 \times n_1} \quad (1)$$

such that there exists a function

$$\chi: C \times n_0 \times n_1 \rightarrow C \quad (2)$$

such that for every digital image $f \in C^{n_0 \times n_1}$ and every pixel $(x, y) \in n_0 \times n_1$:

$$\varphi(f)(x, y) = \chi(f(x, y), x, y) . \quad (3)$$

Remark. The color $\varphi(f)(x, y)$ of the image $\varphi(f)$ at the pixel (x, y) depends only on the color $f(x, y)$ of the image f at that same pixel, and on the pixel coordinates, x and y .

Example. Every $\xi: C \rightarrow C$ defines a point operator, namely $\varphi_\xi: f \mapsto \xi \circ f$.

Gamma Operator

Definition 3. Let $C = [0, 1]$. For any $\gamma \in (0, \infty)$ and the function $\xi: C \rightarrow C: c \mapsto c^\gamma$, the point operator $\varphi_\xi: f \mapsto \xi \circ f$ is called the **gamma operator**.



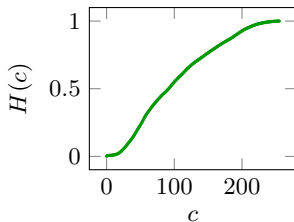
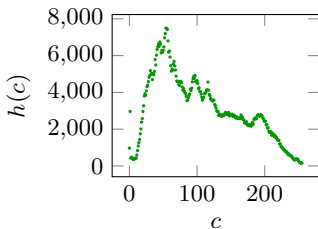
Histogram equilibration

Definition 4. The **histogram** of a digital image $f \in C^{n_0 \times n_1}$ with $C \subseteq \mathbb{R}$ is the $h: C \rightarrow \mathbb{N}_0$ such that for all $c \in C$:

$$h(c) = |\{r \in n_0 \times n_1 \mid f(r) = c\}| \quad (4)$$

The **cumulative distribution of colors** is the $H: C \rightarrow [0, 1]$ such that for all $c \in C$:

$$H(c) = \frac{1}{n_0 n_1} \sum_{\substack{c' \in f(n_0 \times n_1) \\ c' \leq c}} h(c') \quad (5)$$



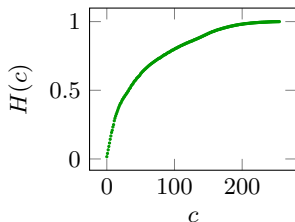
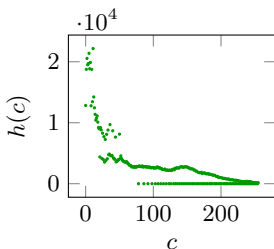
Histogram equilibration

Definition 4. The **histogram** of a digital image $f \in C^{n_0 \times n_1}$ with $C \subseteq \mathbb{R}$ is the $h: C \rightarrow \mathbb{N}_0$ such that for all $c \in C$:

$$h(c) = |\{r \in n_0 \times n_1 \mid f(r) = c\}| \quad (4)$$

The **cumulative distribution of colors** is the $H: C \rightarrow [0, 1]$ such that for all $c \in C$:

$$H(c) = \frac{1}{n_0 n_1} \sum_{\substack{c' \in f(n_0 \times n_1) \\ c' \leq c}} h(c') \quad (5)$$



Histogram equilibration

Definition 5. For any $C = [c^-, c^+] \subseteq \mathbb{R}$ and any increasing $H: C \rightarrow [0, 1]$ such that $H(c^+) = 1$, **H -equilibration** is the function

$$\begin{aligned}\xi_H: \quad [c^-, c^+] &\rightarrow [c^-, c^+] \\ c &\mapsto c^- + (c^+ - c^-) H(c)\end{aligned}$$

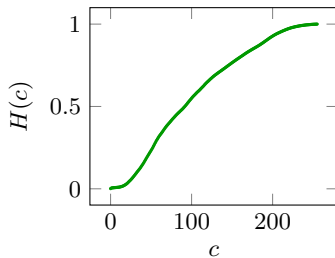
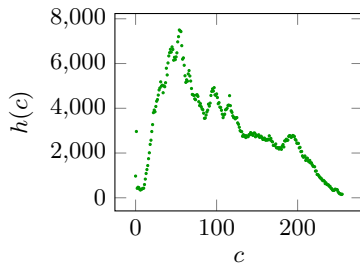
For fixed H and fixed $n_0, n_1 \in \mathbb{N}$, H -equilibration defines a point operator that we call the **H -equilibrator**:

$$\begin{aligned}\varphi_{\xi_H}: \quad C^{n_0 \times n_1} &\rightarrow C^{n_0 \times n_1} \\ f &\mapsto \xi_H \circ f\end{aligned}$$

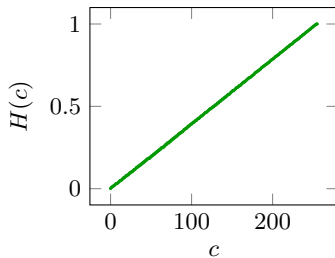
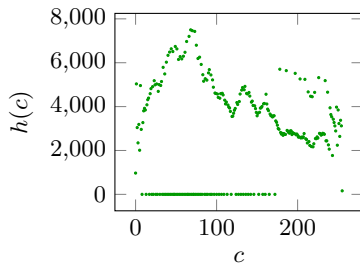
For any digital image f with the cumulative distribution H of colors C , we call the image $\varphi_{\xi_H}(f)$ the **self-equilibration of f** .

Question. Is self-equilibration a point operator?

Histogram equilibration



Histogram equilibration



Linear operators

An operator $\varphi: \mathbb{R}^{n_0 \times n_1} \rightarrow \mathbb{R}^{n_0 \times n_1}$ is **linear** if and only if there exists $a: (n_0 \times n_1)^2 \rightarrow \mathbb{R}$ such that for any (image) $f \in \mathbb{R}^{n_0 \times n_1}$ and any (pixel) $(x, y) \in n_0 \times n_1$:

$$\varphi(f)(x, y) = \sum_{j=0}^{n_0-1} \sum_{k=0}^{n_1-1} a_{xyjk} f(j, k) . \quad (6)$$

More restrictive is the typical setting in which we are given $m_0, m_1 \in \mathbb{N}$ and $g: m_0 \times m_1 \rightarrow \mathbb{R}$ and

$$\varphi(f)(x, y) = \sum_{j=0}^{m_0-1} \sum_{k=0}^{m_1-1} g(j, k) f\left(x + j - \left\lfloor \frac{m_0-1}{2} \right\rfloor, y + k - \left\lfloor \frac{m_1-1}{2} \right\rfloor\right) \quad (7)$$

Remark. In (7):

1. f needs to be extended in order for $\varphi(f)$ to be well-defined.
2. g uniquely defines the linear operator φ .
3. The application of the operator φ to images f defines a binary operation $f \otimes g := \varphi(f)$.
4. g itself is a digital image.

Definition 6. For any $C \neq \emptyset$, the map

$$X : \bigcup_{n_0, n_1 \in \mathbb{N}} C^{n_0 \times n_1} \rightarrow C^{\mathbb{Z} \times \mathbb{Z}} \quad (8)$$

such that for any $n_0, n_1 \in \mathbb{N}$, any $f \in C^{n_0 \times n_1}$ and any $(x, y) \in \mathbb{Z}^2$:

$$X(f)(x, y) = \begin{cases} f(x, y) & \text{if } (x, y) \in n_0 \times n_1 \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

is called the **infinite 0-extension** of digital images.

Definition 7. For any $j, k \in \mathbb{Z}$, the operator $S_{jk} : C^{\mathbb{Z} \times \mathbb{Z}} \rightarrow C^{\mathbb{Z} \times \mathbb{Z}}$ such that for any $x, y \in \mathbb{Z}$: $S_{jk}(f)(x, y) = f(x + j, y + k)$ is called the (x, y) -**shift** of infinite digital images.

Definition 8. The operator $L : C^{\mathbb{Z} \times \mathbb{Z}} \rightarrow C^{\mathbb{Z} \times \mathbb{Z}}$ such that for any $x, y \in \mathbb{Z}$: $L(f)(x, y) = f(-x, -y)$ is called the **reflection** of infinite digital images.

Definition 9. 1-dimensional discrete convolution is the operation $*$: $\mathbb{R}^{\mathbb{Z}} \times \mathbb{R}^{\mathbb{Z}} \rightarrow \mathbb{R}^{\mathbb{Z}}$ such that for any $f, g: \mathbb{Z} \rightarrow \mathbb{R}$ and any $t \in \mathbb{Z}$:

$$(f * g)(t) = \sum_{s=-\infty}^{\infty} f(t+s) g(-s) . \quad (10)$$

2-dimensional discrete convolution is the operation $*$: $\mathbb{R}^{\mathbb{Z} \times \mathbb{Z}} \times \mathbb{R}^{\mathbb{Z} \times \mathbb{Z}} \rightarrow \mathbb{R}^{\mathbb{Z} \times \mathbb{Z}}$ such that for any $f, g: \mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{R}$ and any $(x, y) \in \mathbb{Z} \times \mathbb{Z}$:

$$(f * g)(x, y) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} f(x+j, y+k) g(-j, -k) . \quad (11)$$

Remark. The minus (in $-s$ in (10), and in $-j, -k$ in (11)) makes the operator $*$ commutative.

Lemma 1. For any $f, g, h \in \mathbb{R}^{\mathbb{Z} \times \mathbb{Z}}$ and any $\alpha \in \mathbb{R}$, we have:

$$f * g = g * f \quad (\text{commutativity}) \quad (12)$$

$$f * (g * h) = (f * g) * h \quad (\text{associativity}) \quad (13)$$

$$f * (g + h) = (f * g) + (f * h) \quad (\text{distributivity}) \quad (14)$$

$$\alpha(f * g) = (\alpha f) * g \quad (\text{associativity with } \cdot) \quad (15)$$

Definition 10. For any $n_0, n_1, m_0, m_1 \in \mathbb{N}$, any $f \in C^{n_0 \times n_1}$, any $g \in C^{m_0 \times m_1}$, $d_0 = -\lfloor \frac{m_0-1}{2} \rfloor$ and $d_1 = -\lfloor \frac{m_1-1}{2} \rfloor$, the **convolution** of f and g is defined as

$$f * g := (X(f) * S_{d_0 d_1}(X(g)))|_{n_0 \times n_1} \quad (16)$$

Lemma 2. For any $f, g \in C^{\mathbb{Z} \times \mathbb{Z}}$:

$$f \otimes g = f * L(g) \quad . \quad (17)$$

Example 1. For any $\sigma \in \mathbb{R}^+$ and any $m \in \mathbb{N}_0$ (typically: $m \geq 3\sigma$), for the function

$$w: \mathbb{R} \rightarrow \mathbb{R}: \quad t \mapsto e^{-\frac{t^2}{2\sigma^2}} \quad (18)$$

and the number

$$N := \sum_{j=-m}^m w(j) \ , \quad (19)$$

the functions

$$g_0: (2m+1) \times 1 \rightarrow \mathbb{R}: \quad (x, 0) \mapsto \frac{w(j-m)}{N} \quad (20)$$

$$g_1: 1 \times (2m+1) \rightarrow \mathbb{R}: \quad (0, y) \mapsto \frac{w(j-m)}{N} \quad (21)$$

are called **Gaussian averaging filters**.

Linear operators

f



$f * g_0 * g_1$



$\sigma = 3.0$
 $m = 9$

$f * g_0 * g_1$



$\sigma = 10.0$
 $m = 30$

Linear operators

f



Linear operators

f



$2f - (f * g_0 * g_1)$



$\sigma = 1.0$

$m = 3$

Example 2. The **discrete derivatives** of an infinite digital image $f: \mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{R}$ are defined as

$$\partial_0 f := g * d_0 \quad (22)$$

$$\partial_1 f := g * d_1 \quad (23)$$

with

$$d_0 = \frac{1}{2}(1, 0, -1) \quad (24)$$

$$d_1 = \frac{1}{2} \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix} \quad (25)$$

The **discrete gradient** is defined as

$$\nabla f = \begin{pmatrix} \partial_0 f \\ \partial_1 f \end{pmatrix}, \quad (26)$$

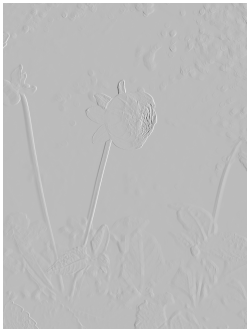
and $|\nabla f| = \sqrt{(\partial_0 f)^2 + (\partial_1 f)^2}$ is called its **magnitude**.

Linear operators

f



$f * d_0$



$f * d_1$



Linear operators

f



$\sqrt{(f * d_0)^2 + (f * d_1)^2}$



Definition 11. Let $n_0, n_1 \in \mathbb{N}$, let $V = n_0 \times n_1$ and let $C \subseteq \mathbb{R}$. Given

- ▶ a metric $d_s : V \times V \rightarrow \mathbb{R}_0^+$ and a decreasing $w_s : \mathbb{R}_0^+ \rightarrow [0, 1]$
- ▶ a metric $d_c : C \times C \rightarrow \mathbb{R}_0^+$ and a decreasing $w_c : \mathbb{R}_0^+ \rightarrow [0, 1]$
- ▶ a $N : V \rightarrow 2^V$ that defines for every pixel $v \in V$ a set $N(v) \subseteq V$ called the **spatial neighborhood** of v
- ▶ the $\nu : C^V \rightarrow \mathbb{R}^V$, called **normalization**, such that for any digital image $f : V \rightarrow C$ and any pixel $v \in V$:

$$\nu(f)(v) = \sum_{v' \in N(v)} w_s(d_s(v, v')) w_c(d_c(f(v), f(v'))) , \quad (27)$$

the **bilateral filter** wrt. d_s, w_s, d_c, w_c and N is the $\beta : C^V \rightarrow (\mathbb{R}C)^V$ such that for any digital image $f : V \rightarrow C$ and any pixel $v \in V$:

$$\beta(f)(v) = \frac{1}{\nu(f)(v)} \sum_{v' \in N(v)} w_s(d_s(v, v')) w_c(d_c(f(v), f(v'))) f(v') \quad (28)$$

Example.

- $d_s(v, v') = \|v - v'\|_2$ and, for a filter parameter $\sigma_s > 0$:

$$w_s(x) = \frac{1}{\sigma_s \sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma_s^2}\right) \quad (29)$$

- $d_c(g, g') = |g - g'|$ and, for a filter parameter $\sigma_c > 0$:

$$w_c(x) = \frac{1}{1 + \frac{x^2}{\sigma_c^2}} \quad (30)$$

- for a filter parameter $n \in \mathbb{R}_0^+$:

$$N(v) = \{v' \in V \mid d_s(v, v') \leq n\} \quad (31)$$

Exercise. Implementation and (recursive) application of the bilateral filter.

Definition 12. Let $n_0, n_1 \in \mathbb{N}$, let $V = n_0 \times n_1$, let $C \subseteq \mathbb{R}$ and let $N : V \rightarrow 2^V$. The **median operator** wrt. N is the function $M : C^V \rightarrow C^V$ such that for any $f : V \rightarrow C$ and any $v \in V$:

$$M(f)(v) = \text{median } f(N(v)) \quad (32)$$

Non-linear operators

Noisy image



f

Filtered image



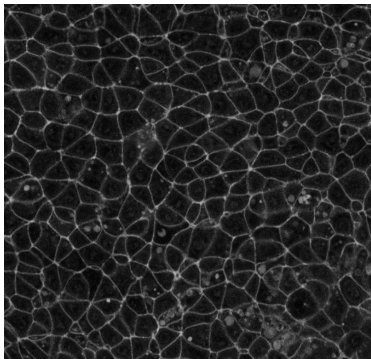
$M(f)$

Morphological operators

- ▶ We may identify any **binary** infinite digital image $f: \mathbb{Z}^2 \rightarrow \{0, 1\}$ with its support set $f^{-1}(1) = \{v \in \mathbb{Z}^2 \mid f(v) = 1\}$.
- ▶ This allows us to apply operations from the field of **binary mathematical morphology** to binary infinite digital images.

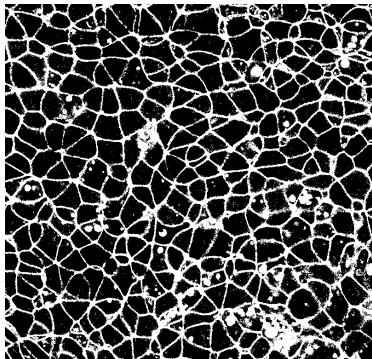
Non-linear operators

Image¹



f

Binary image



$f \geq 45$

¹By courtesy of Stephan Grill and his lab at the MPI of Molecular Cell Biology and Genetics.

Definition 13. A function $\otimes: 2^{\mathbb{Z}^2} \times 2^{\mathbb{Z}^2} \rightarrow 2^{\mathbb{Z}^2}$ is called a **morphological operation**.

Example 3. For any $A, B \subseteq \mathbb{Z}^2$:

$$A \ominus B := \{v \in \mathbb{Z}^2 \mid B + v \subseteq A\} \quad (\text{erosion}) \quad (33)$$

$$A \oplus B := \{v \in \mathbb{Z}^2 \mid -B + v \cap A \neq \emptyset\} \quad (\text{dilation}) \quad (34)$$

$$A \circ B := (A \ominus B) \oplus B \quad (\text{opening}) \quad (35)$$

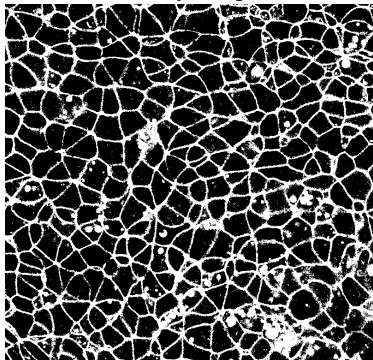
$$A \bullet B := (A \oplus B) \ominus B \quad (\text{closing}) \quad (36)$$

Definition 14. For any (typically finite and small) $B \subseteq \mathbb{Z}^2$ called a **structuring element** and any morphological operation $\otimes$, the **morphological operator** wrt. $\otimes$ and B is defined as the $\varphi_{\otimes B}: \{0, 1\}^{\mathbb{Z} \times \mathbb{Z}} \rightarrow \{0, 1\}^{\mathbb{Z} \times \mathbb{Z}}$ such that for any (infinite binary digital image) $f: \mathbb{Z}^2 \rightarrow \{0, 1\}$ and any (pixel) $v \in \mathbb{Z}^2$:

$$\varphi_{\otimes B}(f)(v) = 1 \quad \Leftrightarrow \quad v \in f^{-1}(1) \otimes B . \quad (37)$$

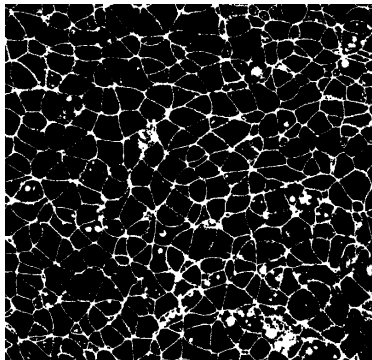
Non-linear operators

Binary image



f

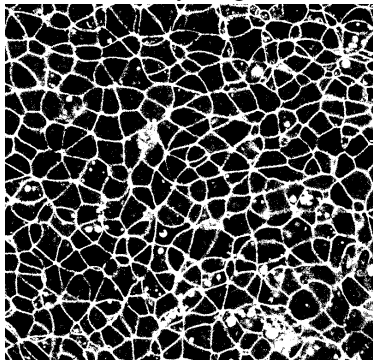
Erosion



$\varphi_{\ominus B}(f)$

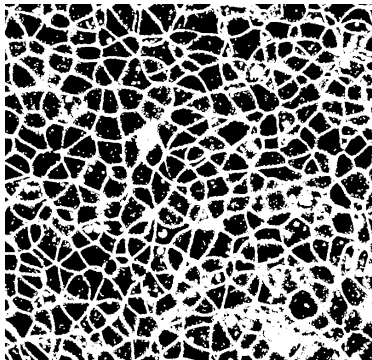
Non-linear operators

Binary image



f

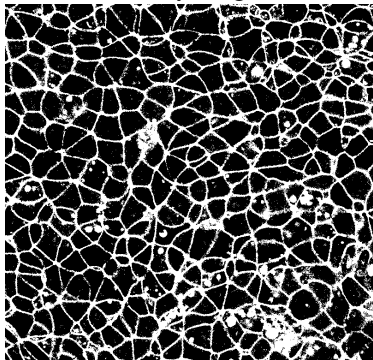
Dilation



$\varphi_{\oplus B}(f)$

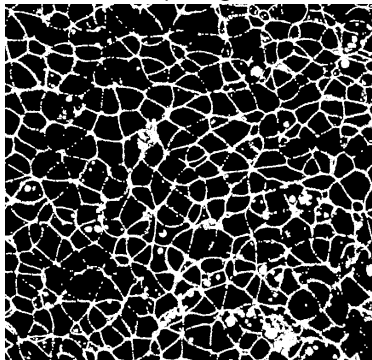
Non-linear operators

Binary image



f

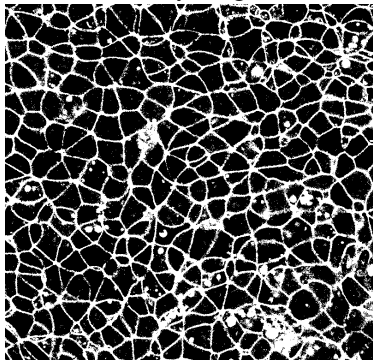
Opening



$\varphi_{\circ B}(f)$

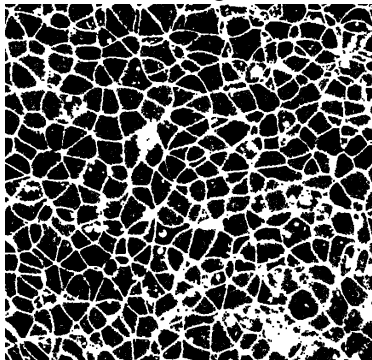
Non-linear operators

Binary image



f

Closing

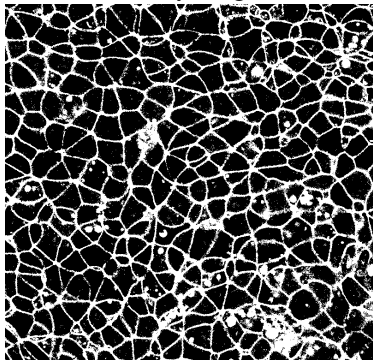


$\varphi_{\bullet B}(f)$

Definition 15. For any $n_0, n_1 \in \mathbb{N}$, the set $V = n_0 \times n_1$ and the pixel grid graph $G = (V, E)$, an operator $\varphi: \mathbb{N}_0^V \rightarrow \mathbb{N}_0^V$ is called a **(connected) components operator** if for any digital image $f: V \rightarrow \mathbb{N}_0$ and any pixels $v, w \in V$, we have $\varphi(f)(v) = \varphi(f)(w)$ iff there exists a vw -path in G along which all pixels have the color zero.

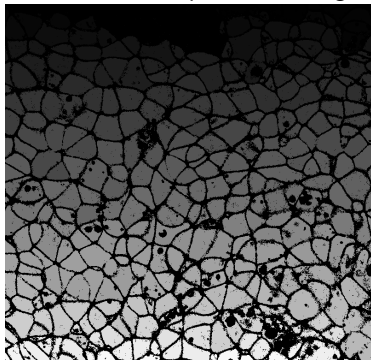
Non-linear operators

Binary image



f

Connected component labeling



$\varphi(f)$

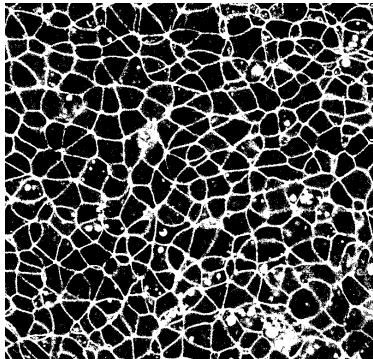
Non-linear operators

```
size_t componentsImage(
    Marray<size_t> const & image,
    Marray<size_t> & components
) {
    components.resize({image.shape(0), image.shape(1)});
    PixelGridGraph pixelGridGraph({image.shape(0), image.shape(1)});
    size_t component = 0;
    stack<size_t> stack;
    for(size_t v = 0; v < pixelGridGraph.numberOfVertices(); ++v) {
        Pixel pixel = pixelGridGraph.coordinate(v);
        if(image[pixel[0], pixel[1]] == 0
            && components[pixel[0], pixel[1]] == 0) {
            ++component;
            components[pixel[0], pixel[1]] = component;
            stack.push(v);
            while(!stack.empty()) {
                size_t const v = stack.top();
                stack.pop();
                for(auto it = pixelGridGraph.verticesFromVertexBegin(v);
                    it != pixelGridGraph.verticesFromVertexEnd(v); ++it) {
                    Pixel pixel = it.coordinate();
                    if(image[pixel[0], pixel[1]] == 0
                        && components[pixel[0], pixel[1]] == 0) {
                        components[pixel[0], pixel[1]] = component;
                        stack.push(*it);
                    }
                }
            }
        }
    }
    return component; // number of components
}
```

Definition 16. For any $n_0, n_1 \in \mathbb{N}$, the set $V = n_0 \times n_1$ and the pixel grid graph $G = (V, E)$, the **distance operator** $\varphi: \mathbb{N}_0^V \rightarrow \mathbb{N}_0^V$ is such that for any digital image $f: V \rightarrow \mathbb{N}_0$ and any pixel $v \in V$, the number $\varphi(f)(v)$ is the minimum distance in the pixel grid graph from v to a pixel w with $f(w) \neq 0$.

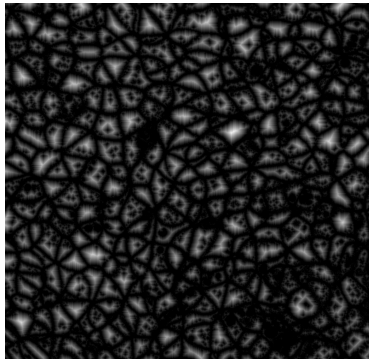
Non-linear operators

Binary image



f

Distance image



$\varphi(f)$

Non-linear operators

```
1 size_t distanceImage(  
2     Marray<size_t> const & image,  
3     Marray<size_t> & distances  
4 ) {  
5     distances.resize({image.shape(0), image.shape(1)}, 0);  
6     GridGraph pixelGridGraph({image.shape(0), image.shape(1)});  
7     size_t distance = 0;  
8     array<stack<size_t>, 2> stacks;  
9     for(size_t v = 0; v < pixelGridGraph.numberOfVertices(); ++v) {  
10         Pixel pixel = pixelGridGraph.coordinates(v);  
11         if(image[pixel[0], pixel[1]] != 0)  
12             stacks[0].push(v);  
13     }  
14     ++distance;  
15     for(;;) {  
16         auto & stack = stacks[(distance - 1) % 2];  
17         if(stack.empty())  
18             return distance - 1; // maximal distance  
19         while(!stack.empty()) {  
20             size_t const v = stack.top();  
21             stack.pop();  
22             for(auto it = pixelGridGraph.verticesFromVertexBegin(v);  
23                 it != pixelGridGraph.verticesFromVertexEnd(v); ++it) {  
24                 Pixel pixel = it.coordinate();  
25                 if(image[pixel[0], pixel[1]] == 0  
26                     && distances[pixel[0], pixel[1]] == 0) {  
27                     distances[pixel[0], pixel[1]] = distance;  
28                     stacks[distance % 2].push(*it);  
29                 }  
30             }  
31         }  
32         ++distance;  
33     }  
34 }
```

Non-linear operators

For any set V of pixels and neighborhood function $N: V \rightarrow 2^V$, **non-maximum suppression** is the operator $\varphi_{\text{NMS}}: \mathbb{R}^V \rightarrow \mathbb{R}^V$ such that for each digital image $f: V \rightarrow \mathbb{R}$ and all pixels $v \in V$:

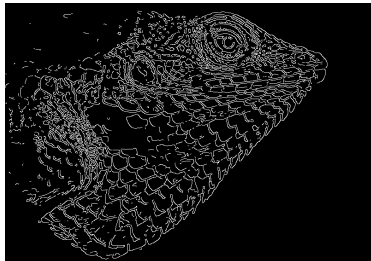
$$\varphi_{\text{NMS}}(f)(v) = \begin{cases} f(v) & \text{if } f(v) = \max f(N(v)) \\ 0 & \text{otherwise} \end{cases} \quad (38)$$

Edge and corner detection

Image¹



Edge detection¹



¹https://en.wikipedia.org/wiki/Canny_edge_detector

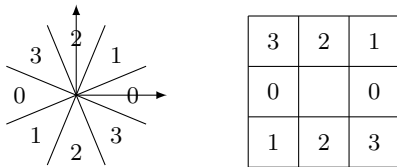
Canny's edge detection algorithm¹ has four steps

1. Gradient computation from digital image $f: V \rightarrow \mathbb{R}$:

$$g = \sqrt{\partial_0 f + \partial_1 f} \quad \text{std::hypot in C++} \quad (39)$$

$$\alpha = \text{atan2}(\partial_1 f, \partial_0 f) \quad \text{std::atan2 in C++} \quad (40)$$

2. Directional non-maximum suppression of g :

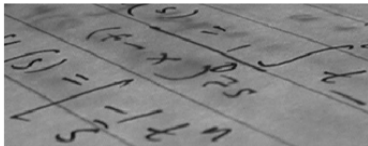


3. Double thresholding with $\theta_0, \theta_1 \in \mathbb{R}_0^+$ such that $\theta_0 \leq \theta_1$: A (any) pixel $v \in V$ is taken considered to be a **strong edge pixel** iff $\theta_1 \leq g(v)$ and is taken to be a **weak edge pixel** iff $\theta_0 \leq g(v) < \theta_1$.
4. Weak edge classification: A (any) pixel $v \in V$ is taken to be an **edge pixel** iff (i) v is a strong edge pixel, or (ii) v is a weak edge pixel and there is a strong edge pixel in the 8-neighborhood of v .

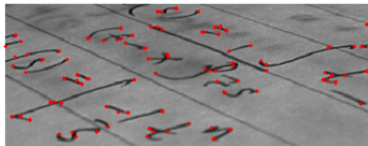
¹J. Canny. A Computational Approach To Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(6):679–698, 1986

Edge and corner detection

Image¹



Corner detection¹



¹https://en.wikipedia.org/wiki/Corner_detection

Definition 17. Let $n_0, n_1 \in \mathbb{N}$, let $V = n_0 \times n_1$, let $f: V \rightarrow \mathbb{R}$ a digital image, let ∂_0, ∂_1 be discrete derivative operators, and let $N: V \rightarrow 2^V$.

For each $v \in V$:

- Let $A(v)$ be the $N(v) \times 2$ -matrix such that for every $w \in N(v)$:

$$A_{w \cdot}(v) = ((\partial_0 f)(w), (\partial_1 f)(w)) \quad . \quad (41)$$

- Let $k_v: N(v) \rightarrow \mathbb{R}_0^+$ such that $\sum_{w \in N(v)} k_v(w) = 1$.
- Define the **structure tensor** of f at v wrt. k_v as the 2×2 -matrix

$$S_k(f)(v) := \sum_{w \in N(v)} k_v(w) A_{w \cdot}^T(v) A_{w \cdot}(v) \quad (42)$$

$$= \sum_{w \in N(v)} k_v(w) \begin{pmatrix} (\partial_0 f)^2(w) & (\partial_0 f)(w)(\partial_1 f)(w) \\ (\partial_0 f)(w)(\partial_1 f)(w) & (\partial_1 f)^2(w) \end{pmatrix} \quad . \quad (43)$$

Edge and corner detection

Remark 1. Fix a direction by choosing $r \in \mathbb{R}^2$ with $|r| = 1$ and consider the k_v -weighted squared projection of the gradient of the digital image:

$$P_r(v) = \sum_{w \in N(v)} k_v(w) |A_{w \cdot}(v) r|^2 \quad (44)$$

$$= \sum_{w \in N(v)} k_v(w) r^T A_{w \cdot}^T(v) A_{w \cdot}(v) r \quad (45)$$

$$= r^T \left(\sum_{w \in N(v)} k_v(w) A_{w \cdot}^T(v) A_{w \cdot}(v) \right) r \quad (46)$$

$$= r^T S(v) r \quad (47)$$

With the spectral decomposition

$$S(v) = \sigma_1(v) s_1(v) s_1^T(v) + \sigma_2(v) s_2(v) s_2^T(v) \quad (48)$$

we obtain

$$P_r(v) = r^T \left(\sigma_1(v) s_1(v) s_1^T(v) + \sigma_2(v) s_2(v) s_2^T(v) \right) r \quad (49)$$

$$= \sigma_1(v) |s_1(v) \cdot r|^2 + \sigma_2(v) |s_2(v) \cdot r|^2 . \quad (50)$$

Remark 2.

- ▶ If $\sigma_1 = \sigma_2 = 0$, we have $P_r(v) = 0$ for any direction r . I.e. the image is constant.
- ▶ If $\sigma_1 > 0$ and $\sigma_2 = 0$, we can choose a direction r such that $P_r(v) = 0$. I.e. the gradient of the image is non-zero and constant.
- ▶ If $\sigma_1, \sigma_2 > 0$, we cannot choose r such that $P_r(v) = 0$. I.e. the gradient of the image varies across $N(v)$.

Definition 18. Let V the set of pixels of a digital image, let $S: V \rightarrow \mathbb{R}^{2 \times 2}$ such that for any $v \in V$, $S(v)$ is the structure tensor of the image at pixel v , and let $\sigma_1(v) \geq \sigma_2(v) \geq 0$ be the eigenvalues of $S(v)$. **Harris' corner detector**² wrt. a neighborhood function $N: V \rightarrow 2^V$ refers to the function $\varphi_{\text{NMS}} \circ \sigma_2$.

²C. Harris and M. Stephens. A Combined Corner and Edge Detector. Alvey Vision Conference. Vol. 15. 1988

Summary.

- ▶ This part of the course has introduced operators on digital images.
- ▶ These include, in particular, point operators (like the gamma operator), linear operators (esp. convolution, Gaussian averaging and derivative operators) and non-linear operators (esp. self-equilibration, morphological operators, Canny edge detection and Harris corner detection).