Computer Vision I

Bjoern Andres, Holger Heidrich, Jannik Presberger

Machine Learning for Computer Vision
TU Dresden

Winter Term 2023/2024

For any $n \in \mathbb{N}$, let $[n] := \{0, \dots, n-1\}$.

Definition 1. A **digital image** of **width** $n_0 \in \mathbb{N}$ and **height** $n_1 \in \mathbb{N}$ with colors C is a map $f\colon [n_0] \times [n_1] \to C$.

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 $C = [0,1]^n$

Definition 2. For any digital image $f\colon [n_0] \times [n_1] \to C$, consider the graph $G = (V, E)$ with $V = [n_0] \times [n_1]$ and such that for any $u, v \in V$ we have $\{u, v\} \in E$ if and only if $|u - v| = 1$. It is called the **pixel grid graph** of the image. Its nodes are called the **pixels** of the image.

Definition 3. For any $n_0, n_1 \in \mathbb{N}$ and any set C, a **point operator** on digital images of width n_0, height n_1 and with colors C is a function

$$\varphi\colon C^{[n_0]\times[n_1]} \to C^{[n_0]\times[n_1]} \tag{1}$$

such that there exists a function

$$\chi\colon C \times [n_0] \times [n_1] \to C \tag{2}$$

such that for every digital image $f\colon [n_0] \times [n_1] \to C$ and every pixel $(x, y) \in [n_0] \times [n_1]$, we have

$$\varphi(f)(x, y) = \chi(f(x, y), x, y) \ . \tag{3}$$

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 location, and on the location (x, y) itself.

Example. Every $\xi\colon C \to C$ defines a point operator $\varphi_\xi\colon f \mapsto \xi \circ f$.

Definition 4. Let $C = [0, 1]$. For any $\gamma \in (0, \infty)$ and the function $\xi \colon C \to C : c \mapsto c^\gamma$, the point operator $\varphi_\xi \colon f \mapsto \xi \circ f$ is called the **gamma operator**.

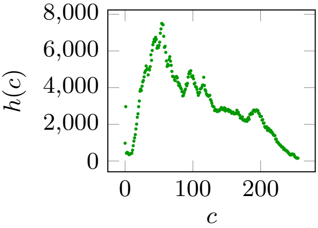
$\gamma = \frac{1}{4}$ $\quad$ $\gamma = \frac{1}{2}$ $\quad$ $\gamma = 1$ $\quad$ $\gamma = 2$ $\quad$ $\gamma = 4$

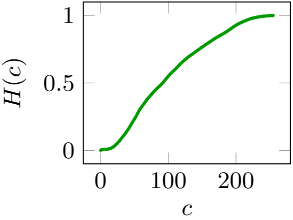
Definition 5. The **histogram** of a digital image $f\colon [n_0] \times [n_1] \to C \subseteq \mathbb{R}$ is the function $h\colon C \to \mathbb{N}_0$ such that for any $c \in C$ we have

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

The **cumulative distribution of colors** is the function $H\colon C \to [0,1]$ such that for any $c \in C$ we have

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



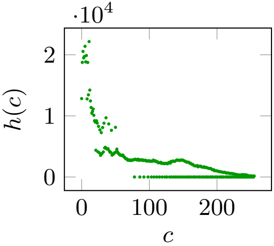


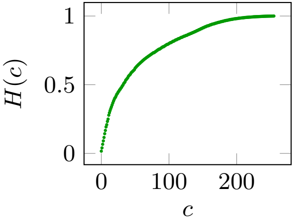
Definition 5. The **histogram** of a digital image $f\colon [n_0] \times [n_1] \to C \subseteq \mathbb{R}$ is the function $h\colon C \to \mathbb{N}_0$ such that for any $c \in C$ we have

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

The **cumulative distribution of colors** is the function $H\colon C \to [0,1]$ such that for any $c \in C$ we have

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





Definition 6. For any $C = [c^-, c^+] \subseteq \mathbb{R}$ and any monotonous function $H\colon C \to [0,1]$ such that $H(c^+) = 1$, H-**equilibration** is the function

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

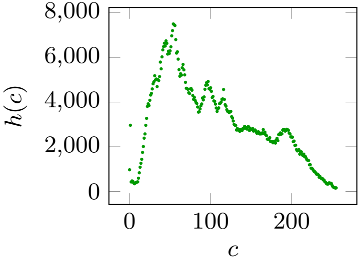
For fixed H and fixed $n_0, n_1 \in \mathbb{N}$, H-equilibration defines a point operator that we call the H-**equilibrator**:

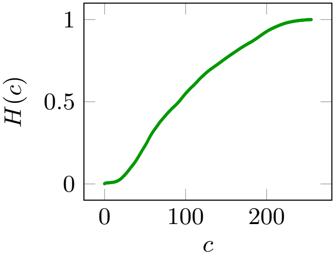
$$\begin{aligned} \varphi_{\xi_H}\colon \quad C^{[n_0]\times[n_1]} &\to C^{[n_0]\times[n_1]} \\ f &\mapsto \xi_H \circ f \end{aligned}$$

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.

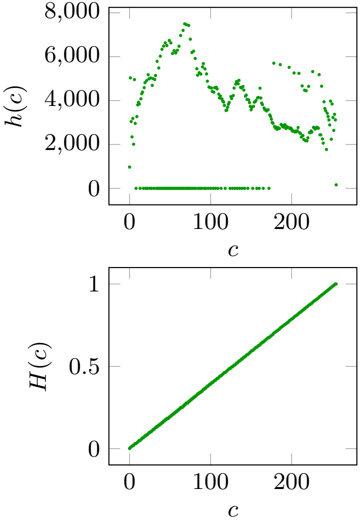
Question. Is self-equilibration a point operator?

Histogram equilibration





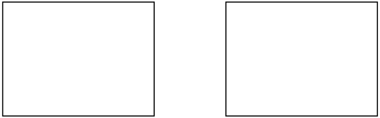
Histogram equilibration



Lemma 1. An operator $\varphi\colon \mathbb{R}^{[n_0]\times[n_1]} \to \mathbb{R}^{[n_0]\times[n_1]}$ is **linear** if and only if there exists $a\colon ([n_0]\times[n_1])^2 \to \mathbb{R}$ such that for any (image) $f \in \mathbb{R}^{[n_0]\times[n_1]}$ and any (pixel) $(x,y) \in [n_0]\times[n_1]$, we have

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

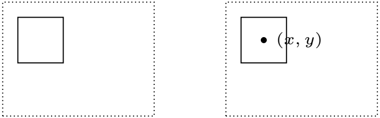
$\varphi(f)(x,y) \quad =$



$a_{xy\cdot\cdot} \qquad \cdot \qquad f$

More restrictive than such an operator with $(n_0 n_1)^2$ coefficients is:

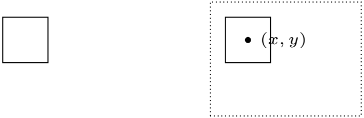
$\varphi(f)(x,y) \quad =$



$g_{xy} \qquad \cdot \qquad S_{xy} f$

Even more restrictive is the typical setting in which we are given $m_0, m_1 \in \mathbb{N}$ and $g\colon [m_0] \times [m_1] \to \mathbb{R}$ and

$\varphi(f)(x,y) \quad =$



$$g \qquad \cdot \qquad S_{xy} f$$

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

Remark 1.

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

Definition 7. For the set $\mathbb{R}^{\mathbb{Z}}$ of all functions from $\mathbb{Z}$ to $\mathbb{R}$, **convolution** is the operation $*\colon \mathbb{R}^{\mathbb{Z}} \times \mathbb{R}^{\mathbb{Z}} \to \mathbb{R}^{\mathbb{Z}}$ such that for any $f, g\colon \mathbb{Z} \to \mathbb{R}$ and any $t \in \mathbb{Z}$:

$$(f * g)(t) = \sum_{s=-\infty}^{\infty} f(t+s)\, g(-s) \ . \tag{7}$$

For the set $\mathbb{R}^{\mathbb{Z}\times\mathbb{Z}}$ of all functions from $\mathbb{Z} \times \mathbb{Z}$ to $\mathbb{R}$, **convolution** is the operation $*\colon \mathbb{R}^{\mathbb{Z}\times\mathbb{Z}} \times \mathbb{R}^{\mathbb{Z}\times\mathbb{Z}} \to \mathbb{R}^{\mathbb{Z}\times\mathbb{Z}}$ such that for any $f, g\colon \mathbb{Z} \times \mathbb{Z} \to \mathbb{R}$ and any $(x, y) \in \mathbb{Z} \times \mathbb{Z}$:

$$(f * g)(x, y) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} f(x+j, y+k)\, g(-j, -k) \ . \tag{8}$$

Lemma 2. For any $f, g, h \in \mathbb{R}^{\mathbb{Z}\times\mathbb{Z}}$ and any $\alpha \in \mathbb{R}$, we have:

$$f * g = g * f \qquad \text{(commutativity)} \qquad (9)$$

$$f * (g * h) = (f * g) * h \qquad \text{(associativity)} \qquad (10)$$

$$f * (g + h) = (f * g) + (f * h) \qquad \text{(distributivity)} \qquad (11)$$

$$\alpha(f * g) = (\alpha f) * g \qquad \text{(associativity with } \cdot\text{)} \qquad (12)$$

Definition 8. For any $C \neq \emptyset$, the operator $X : \bigcup_{n_0, n_1 \in \mathbb{N}} C^{[n_0] \times [n_1]} \to C^{\mathbb{Z} \times \mathbb{Z}}$ such that for any $n_0, n_1 \in \mathbb{N}$, any $f \colon [n_0] \times [n_1] \to C$ and any $(x, y) \in \mathbb{Z}^2$ we have

$$X(f)(x,y) = \begin{cases} f(x,y) & \text{if } (x,y) \in [n_0] \times [n_1] \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

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

Definition 9. For any $C \neq \emptyset$ and any $n_0, n_1 \in \mathbb{N}$, the map $R_{n_0, n_1} \colon C^{\mathbb{Z} \times \mathbb{Z}} \to C^{[n_0] \times [n_1]}$ such that for any $f \colon \mathbb{Z} \times \mathbb{Z} \to C$ and any $(x, y) \in [n_0] \times [n_1]$, we have $R_n(f)(x, y) = f(x, y)$ is called the (n_0, n_1)-**restriction** of infinite digital images.

Definition 10. For any $j, k \in \mathbb{Z}$, the operator $S_{jk}\colon C^{\mathbb{Z}\times\mathbb{Z}} \to C^{\mathbb{Z}\times\mathbb{Z}}$ such that for any $x, y \in \mathbb{Z}$, we have $S_{jk}(f)(x,y) = f(x+j, y+k)$ is called the (x,y)-**shift** of infinite digital images.

Definition 11. The operator $L\colon C^{\mathbb{Z}\times\mathbb{Z}} \to C^{\mathbb{Z}\times\mathbb{Z}}$ such that for any $x, y \in \mathbb{Z}$, we have $L(f)(x,y) = f(-x,-y)$ is called the **reflection** of infinite digital images.

Definition 12. For any $n_0, n_1, m_0, m_1 \in \mathbb{N}$, any $f \in C^{[n_0]\times[n_1]}$, any $g \in C^{[m_0]\times[m_1]}$, $d_0 = -\left\lfloor \frac{m_0-1}{2} \right\rfloor$ and $d_1 = -\left\lfloor \frac{m_1-1}{2} \right\rfloor$, the **convolution** of f and g is defined as

$$f * g := R_{n_0 n_1}(X(f) * S_{d_0 d_1}(X(g))) \tag{14}$$

Lemma 3. For any $n_0, n_1, m_0, m_1 \in \mathbb{N}$, any $f \in C^{[n_0]\times[n_1]}$ and any $g \in C^{[m_0]\times[m_1]}$:

$$f \otimes g = f * L(g) \tag{15}$$

Definition 13. For any $\sigma \in \mathbb{R}^+$ and any $m \in \mathbb{N}_0$ (typically: $m \geq 3\sigma$), for the function

$$w\colon \quad \mathbb{R} \to \mathbb{R}\colon \quad t \mapsto e^{-\frac{t^2}{2\sigma^2}} \tag{16}$$

and the number

$$N := \sum_{j=-m}^{m} w(j) \ , \tag{17}$$

the functions

$$g_0\colon \quad [2m+1] \times [1] \to \mathbb{R}\colon \quad (x, 0) \mapsto \frac{w(j-m)}{N} \tag{18}$$

$$g_1\colon \quad [1] \times [2m+1] \to \mathbb{R}\colon \quad (0, y) \mapsto \frac{w(j-m)}{N} \tag{19}$$

are called **Gaussian averaging filters**.

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

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

$\sigma = 1.0$
$m = 3$

Definition 14. The **discrete derivatives** of an infinite digital image $f\colon \mathbb{Z} \times \mathbb{Z} \to \mathbb{R}$ are defined as

$$\partial_0 f := g * d_0 \tag{20}$$

$$\partial_1 f := g * d_1 \tag{21}$$

with

$$d_0 = \frac{1}{2}(1, 0, -1) \tag{22}$$

$$d_1 = \frac{1}{2}\begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix} \tag{23}$$

The **discrete gradient** is defined as

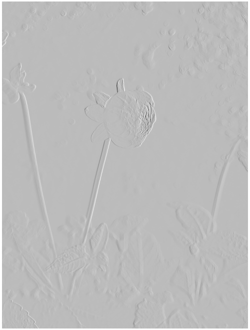
$$\nabla f = \begin{pmatrix} \partial_0 f \\ \partial_1 f \end{pmatrix} \ , \tag{24}$$

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

Linear operators

f

$f * d_0$



$f * d_1$

Linear operators

f

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

Definition 15. Let $n_0, n_1 \in \mathbb{N}$, let $V = [n_0] \times [n_1]$ and let $C \subseteq \mathbb{R}$. Given

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

$$\nu(f)(v) = \sum_{v' \in N(v)} w_s(d_s(v, v'))\, w_c(d_c(f(v), f(v'))) \ , \qquad (25)$$

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

$$\beta(f)(v) = \frac{1}{\nu(f)(v)} \sum_{v' \in N(v)} w_s(d_s(v, v'))\, w_c(d_c(f(v), f(v')))\, f(v') \qquad (26)$$

Example.

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

$$w_s(x) = \frac{1}{\sigma_s \sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma_s^2}\right) \tag{27}$$

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

$$w_c(x) = \frac{1}{1 + \frac{x^2}{\sigma_c^2}} \tag{28}$$

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

$$N(v) = \{v' \in V \mid d_s(v, v') \leq n\} \tag{29}$$

Definition 16. Let $n_0, n_1 \in \mathbb{N}$, let $V = [n_0] \times [n_1]$, let $C \subseteq \mathbb{R}$ and let $N : V \to 2^V$ define for every pixel $v \in V$ a set $N(v) \subseteq V$ called the **spatial neighborhood** of v. The **median operator** wrt. N is the function $M\colon C^V \to C^V$ such that for any $f\colon V \to C$ and any $v \in V$:

$$M(f)(v) = \text{median } f(N(v)) \tag{30}$$

Non-linear operators

Noisy image

f

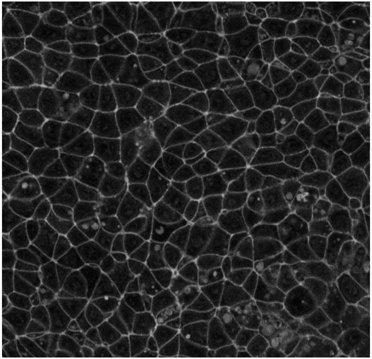
Filtered image

$M(f)$

Morphological operators

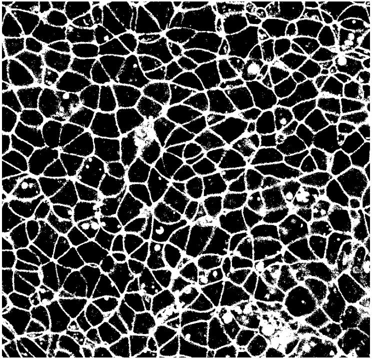
- We may identify any **binary** infinite digital image $f\colon \mathbb{Z}^2 \to \{0,1\}$ with its support set $f^{-1}(1) = \{v \in \mathbb{Z}^2 \mid f(v) = 1\}$.
- This allows us to apply operations from the field of **binary mathematical morphology** to binary infinite digital images.

Image[1]



f

Binary image



$f \geq 45$

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

Definition 17. For any $A, B \subseteq \mathbb{Z}^2$, we define

$$A \ominus B := \{v \in \mathbb{Z}^2 \mid B + v \subseteq A\} \tag{31}$$

$$A \oplus B := \{v \in \mathbb{Z}^2 \mid -B + v \cap A \neq \emptyset\} \tag{32}$$

and call these operations **erosion** and **dilation**. Moreover, we call the operations

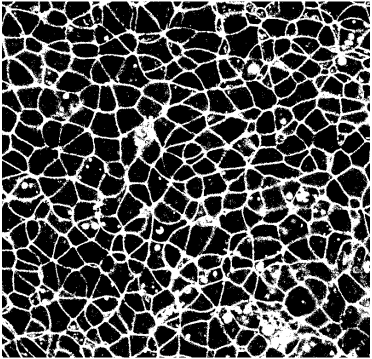
$$A \circ B := (A \ominus B) \oplus B \tag{33}$$

$$A \bullet B := (A \oplus B) \ominus B \tag{34}$$

opening and **closing**.

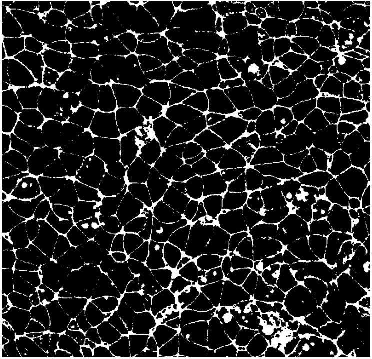
Definition 18. For any (typically small) support set B called a **structuring element** and any morphological operation $\otimes$, the operator $\varphi_{\otimes B} \colon \{0,1\}^{\mathbb{Z}\times\mathbb{Z}} \to \{0,1\}^{\mathbb{Z}\times\mathbb{Z}}$ such that for any (infinite binary digital image) $f \colon \mathbb{Z}^2 \to \{0,1\}$ and any (pixel) $v \in \mathbb{Z}^2$, we have $\varphi_{\otimes B}(f)(v) = 1$ if and only if $v \in f^{-1}(1) \otimes B$ is called the **morphological operator** wrt. $\otimes$ and B.

Binary image



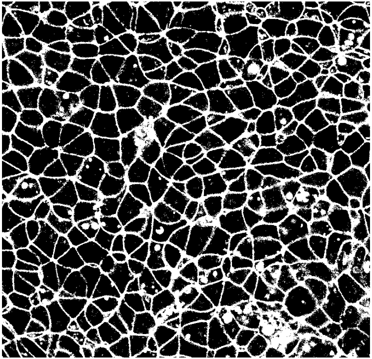
f

Erosion



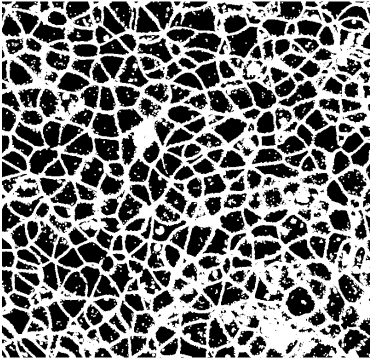
$\varphi_{\ominus B}(f)$

Binary image



f

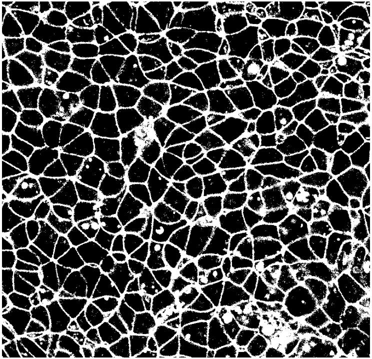
Dilation



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

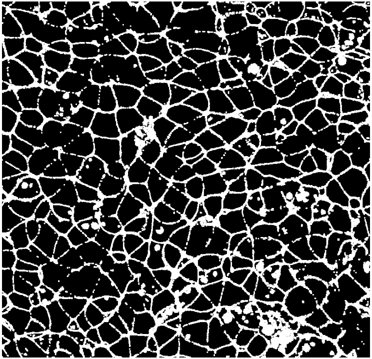
Non-linear operators

Binary image



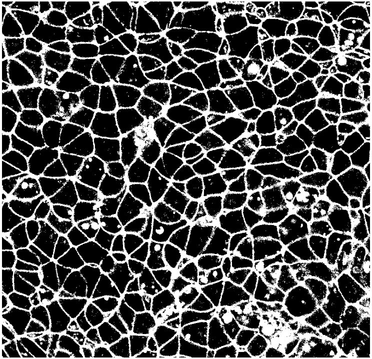
f

Opening



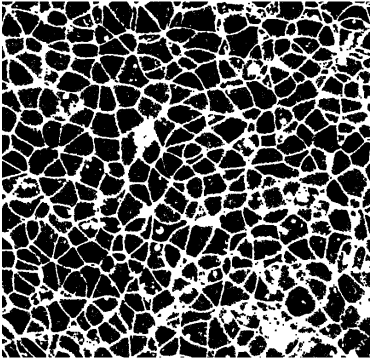
$\varphi_{\circ B}(f)$

Binary image



f

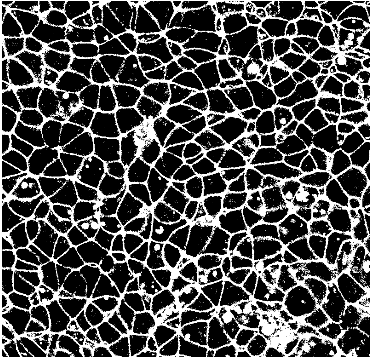
Closing



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

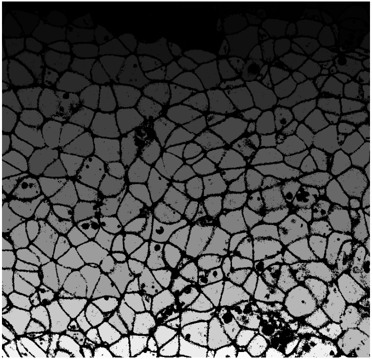
Definition 19. For any $n_0, n_1 \in \mathbb{N}$, the set $V = [n_0] \times [n_1]$ and the pixel grid graph $G = (V, E)$, an operator $\varphi \colon \mathbb{N}_0^V \to \mathbb{N}_0^V$ is called a **(connected) components operator** if for any digital image $f \colon V \to \mathbb{N}_0$ and any pixels $v, w \in V$, we have $\varphi(f)(v) = \varphi(f)(w)$ iff there exists a vw-path in G along which all pixels have the color zero.

Binary image



f

Connected component labeling



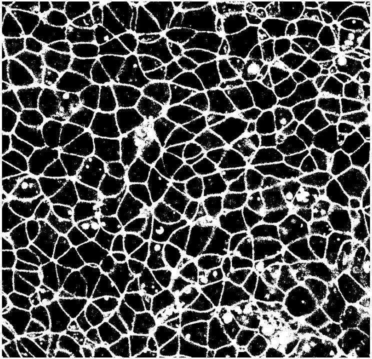
$\varphi(f)$

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
}
```

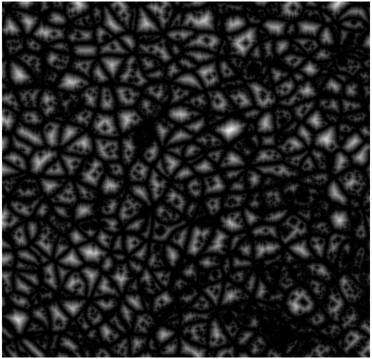
Definition 20. For any $n_0, n_1 \in \mathbb{N}$, the set $V = [n_0] \times [n_1]$ and the pixel grid graph $G = (V, E)$, the **distance operator** $\varphi \colon \mathbb{N}_0^V \to \mathbb{N}_0^V$ is such that for any digital image $f \colon V \to \mathbb{N}_0$ and any pixel $v \in V$, the number $\varphi(f)(v)$ is the minimum distance in the pixel grid graph from v to a pixel w with $f(w) = 1$.

Binary image



f

Distance image



$\varphi(f)$

```
size_t distanceImage(
    Marray<size_t> const & image,
    Marray<size_t> & distances
) {
    distances.resize({image.shape(0), image.shape(1)}, 0);
    GridGraph pixelGridGraph({image.shape(0), image.shape(1)});
    size_t distance = 0;
    array<stack<size_t>, 2> stacks;
    for(size_t v = 0; v < pixelGridGraph.numberOfVertices(); ++v) {
        Pixel pixel = pixelGridGraph.coordinates(v);
        if(image(pixel[0], pixel[1]) != 0)
            stacks[0].push(v);
    }
    ++distance;
    for(;;) {
        auto & stack = stacks[(distance - 1) % 2];
        if(stack.empty())
            return distance - 1; // maximal distance
        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
                && distances(pixel[0], pixel[1]) == 0) {
                    distances(pixel[0], pixel[1]) = distance;
                    stacks[distance % 2].push(*it);
                }
            }
        }
        ++distance;
    }
}
```

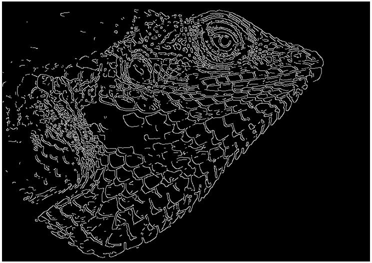
For any set V of pixels and neighborhood function $N\colon V \to 2^V$,
non-maximum suppression is the operator $\varphi_{\mathrm{NMS}}\colon \mathbb{R}^V \to \mathbb{R}^V$ such that for each digital image $f\colon V \to \mathbb{R}$ and all pixels $v \in V$:

$$\varphi_{\mathrm{NMS}}(f)(v) = \begin{cases} f(v) & \text{if } f(v) = \max f(N(v)) \\ 0 & \text{otherwise} \end{cases} \tag{35}$$

Edge and corner detection

Image[1]

Edge detection[1]



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

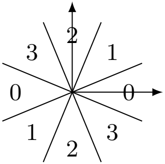
Canny's edge detection algorithm[1] has four steps

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

$$g = \sqrt{\partial_0 f + \partial_1 f} \qquad \texttt{std::hypot} \text{ in C++} \tag{36}$$

$$\alpha = \operatorname{atan2}(\partial_1 f, \partial_0 f) \qquad \texttt{std::atan2} \text{ in C++} \tag{37}$$

2. Directional non-maximum suppression of g:



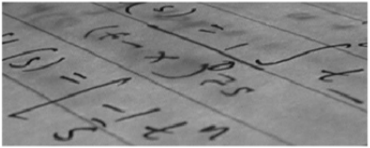
3	2	1
0		0
1	2	3

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.

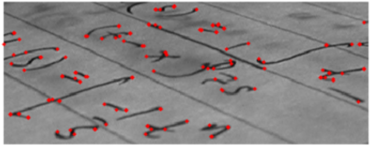
[1] J. Canny. A Computational Approach To Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(6):679–698, 1986

Edge and corner detection

Image[1]



Corner detection[1]



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

Definition 21. Let $n_0, n_1 \in \mathbb{N}$, let $V = [n_0] \times [n_1]$, let $f\colon V \to \mathbb{R}$ a digital image, let ∂_0, ∂_1 be discrete derivative operators, and let $N\colon V \to \mathbb{R}^V$.

For each $v \in V$:

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

$$A_{w\cdot}(v) = ((\partial_0 f)(w), (\partial_1 f)(w)) \ . \tag{38}$$

- Let $k_v\colon N(v) \to \mathbb{R}_0^+$ 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

$$S_k(f)(v) := \sum_{w \in N(v)} k_v(w) A^T_{w\cdot}(v) A_{w\cdot}(v) \tag{39}$$

$$= \sum_{w \in N(v)} k_v(w) \begin{pmatrix} (\partial_0 f)^2(w) & (\partial_0 f)(w)(\partial_1 f)(w) \\ (\partial_0 f)(w)(\partial_1 f)(w) & (\partial_1 f)^2(w) \end{pmatrix} \ . \tag{40}$$

Remark 2. 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:

$$
\begin{aligned}
P_r(v) &= \sum_{w \in N(v)} k_v(w) |A_{w\cdot}(v)\, r|^2 && (41)\\
&= \sum_{w \in N(v)} k_v(w)\, r^T A_{w\cdot}^T(v) A_{w\cdot}(v)\, r && (42)\\
&= r^T \left(\sum_{w \in N(v)} k_v(w) A_{w\cdot}^T(v) A_{w\cdot}(v) \right) r && (43)\\
&= r^T S(v)\, r && (44)
\end{aligned}
$$

With the spectral decomposition

$$
S(v) = \sigma_1(v) s_1(v) s_1^T(v) + \sigma_2(v) s_2(v) s_2^T(v) \qquad (45)
$$

we obtain

$$
\begin{aligned}
P_r(v) &= r^T \left(\sigma_1(v) s_1(v) s_1^T(v) + \sigma_2(v) s_2(v) s_2^T(v) \right) r && (46)\\
&= \sigma_1(v) |s_1(v) \cdot r|^2 + \sigma_2(v) |s_2(v) \cdot r|^2 \ . && (47)
\end{aligned}
$$

Remark 3.

- 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 22. Let V the set of pixels of a digital image, let $S\colon V \to \mathbb{R}^{2\times 2}$ 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 $N\colon V \to 2^V$ refers to the function $\varphi_{\mathrm{NMS}} \circ \sigma_2$.

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