Les Opérateurs de morphologie mathématique dans olena

Réda DEHAK

/oln/oln/morpho

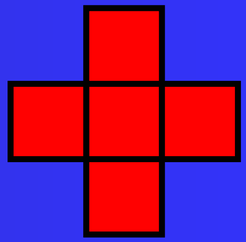
- dilation.hh : opérateur de dilatation.
- erosion.hh : opérateur d’érosion.
- closing.hh : opérateur de fermeture.
- opening.hh : opérateur d’ouverture.
- gradient.hh : opérateur du gradient
- top_hat.hh : chapeau haut-de-forme
- watershed.hh : ligne de partage des eaux
- hit_or_miss.hh.
- thinning.hh : amincissement
- thickening.hh : épaississement
- laplacian.hh.
- ….

élément structurant

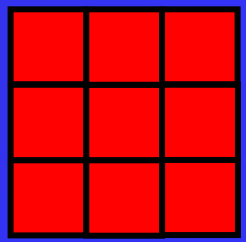
Les transformations morphologiques T sont exprimées à partir de la relation de l'image avec un ensemble de point Se appelé élément structurant. Cet ensemble est centré

oln/core/structelt.hh

win_c4p()   4 connexité



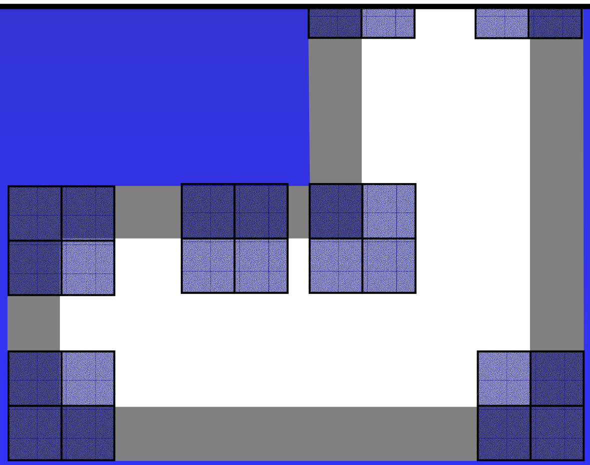
win_c8p()   8 connexité



dilatation

```
oln/morpho/dilation.hh
Concrete(I) morpho::dilation(const image<i> &input
```

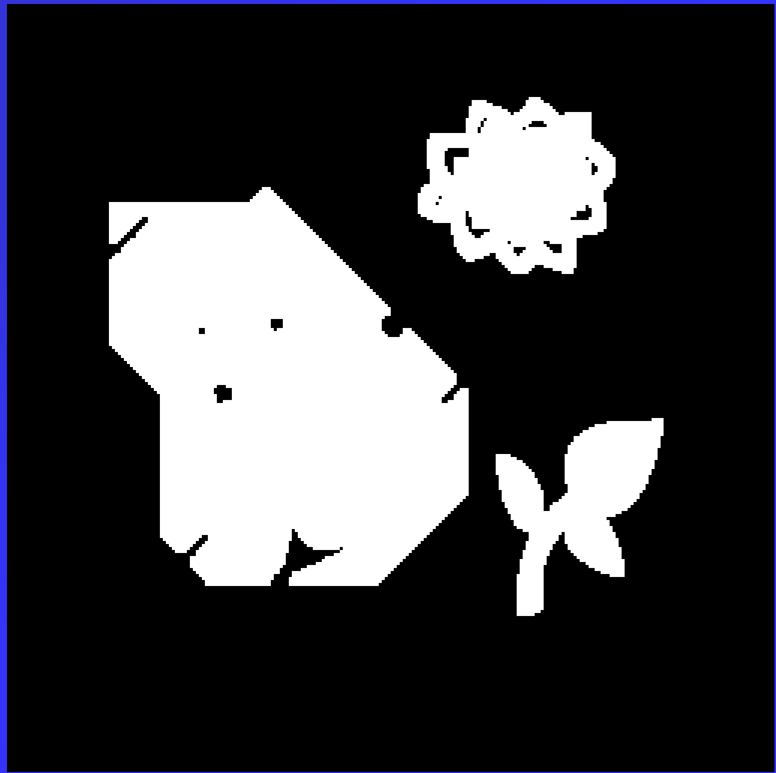
```
oln/morpho/dilation.hh
Concrete(I) morpho::fast::dilation(const image<i> &input,
                                   const struct_elt<E> &se);
Concrete(I) morpho::fast::n_dilation(const image<i> &input,
                                     const struct_elt<E> &se
                                     unsigned n);
```



$$x \in D(o) \Leftrightarrow \exists p \in o / x \in Se(p)$$

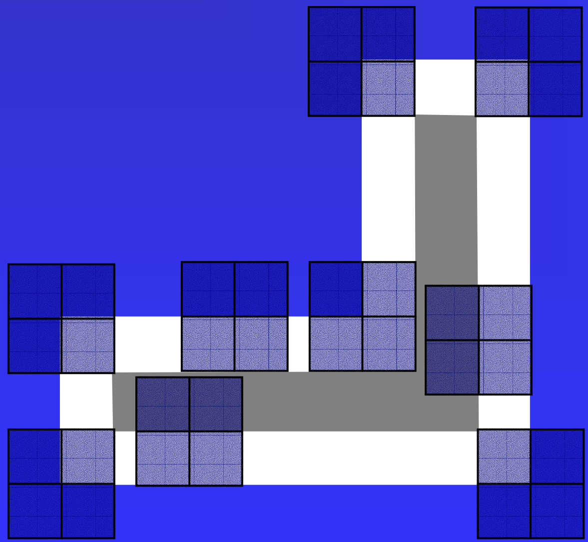
Exemple

```
image2d<bin> im = load("object.pbm");
save(morpho::dilation(im, win_c8p() ), "out.pbm");
```



Erosion

```
oln/morpho/erosion.hh
Concrete(I) morpho::erosion(const image<i> &input,
                            const struct_elt<E> &se);
Concrete(I) morpho::n_erosion(const image<i> &input,
                              const struct_elt<E> &se
                              unsigned n);
```



$$x \in E(o) \Leftrightarrow \forall p \in Se(x) / p \in o$$

Exemple

```
image2d<bin> im = load("object.pbm");
save(morpho::erosion(im, win_c8p() ), "out.pbm");
```

Répertoire tool/utilities/morpho

- black_top_hat_2d_int_u8*
- closing_2d_int_u8*
- dilation_2d_int_u8*
- erosion_2d_int_u8*
- fast_beucher_gradient_2d_int_u8*
- fast_black_top_hat_2d_int_u8*
- fast_closing_2d_int_u8*
- fast_dilation_2d_int_u8*
- fast_erosion_2d_int_u8*
- fast_external_gradient_2d_int_u8*
- ….

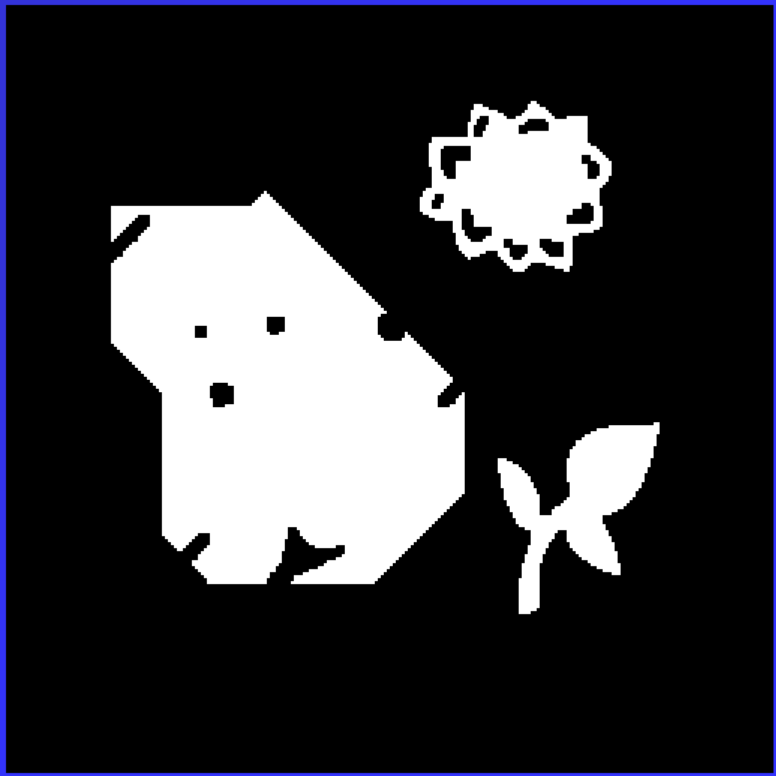
Fermeture

Une dilatation suivie d'une érosion en utilisant le même élément structurant.

```
oln/morpho/closing.hh
Concrete(I) morpho::closing(const image<i> &input,
                            const struct_elt<E> &se);
```

Exemple

```
image2d<bin> im = load("object.pbm");
save(morpho::closing(im, win_c8p() ), "out.pbm");
```



Ouverture

Une erosion suivie par une dilatation en utilisant le même élément structurant.

```
oln/morpho/opening.hh
Concrete(I) morpho::opening(const image<i> &input,
                            const struct_elt<E> &se);
```

Exemple

```
image2d<bin> im = load("object.pbm");
save(morpho::open(im, win_c8p() ), "out.pbm");
```

Gradient de Beucher

oln/morpho/gradient.hh :

```
template<class C, class I, class E>
typename mute<I, typename convoutput<C, Value(I)>::ret>::ret

     morpho::beucher_gradient (const image<i> &input,
                                const struct_elt<E> &se);
```

Beucher_gradient(I) = Dilatation(I) – Erosion(I)

Exemple

```
image2d<bin> im = load("object.pbm");
save(morpho::beucher_gradient(im, win_c8p() ), "out.pbm");
```

Exemple

```
Image2d<int_u8> im = load("lena.pgm");
save(morpho::beucher_gradient (im, win_c8p() ), "out.pgm");
```

chapeau haut-de-forme

```
oln/morpho/.hh
template<class C, class I, class E>
typename mute<I, typename convoutput<C, Value(I)>::ret>::ret

morpho::black_top_hat(const image<i> &input,
                                  const struct_elt<E> &se);

template<class C, class I, class E>
typename mute<I, typename convoutput<C, Value(I)>::ret>::ret

morpho::white_top_hat(const image<i> &input,
                                  const struct_elt<E> &se);
```

Black_top_hat(I) = Fermeture(I) – I

White_top_hat(I) = I – Ouverture(I)

Exemple

```
image2d<int_u8> im = load("lena.pgm");
save(morpho::black_top_hat(im, win_c8p() ), "out1.pgm");
save(morpho::white_top_hat(im, win_c8p() ), "out2.pgm");
```

Hit_or_missing

```
template<class C, class I_, class E1_, class E2_>
typename mute<I_, typename convoutput<C, Value(I_)>::ret>::ret
    morpho::hit_or_miss(const image<I_>& _input,
                        const struct_elt<E1_>& _se1,
                        const struct_elt<E2_>& _se2)
```

$$\text{Hit\_or\_miss}(I, se1, se2) = \{x \,/\, se1(x) \subset X \text{ et } se2(x) \subset X^c\}$$

$$= \text{Erosion}(I, se1) \cap \text{Erosion}(I^c, se2)$$

Amincissement & épaississement

```
oln/morpho/thinning.hh :

Concrete(I_) morpho::thinning( const image<I_>& _input,
                               const struct_elt<E1_>& _se1,
                               const struct_elt<E2_>& _se2)
```

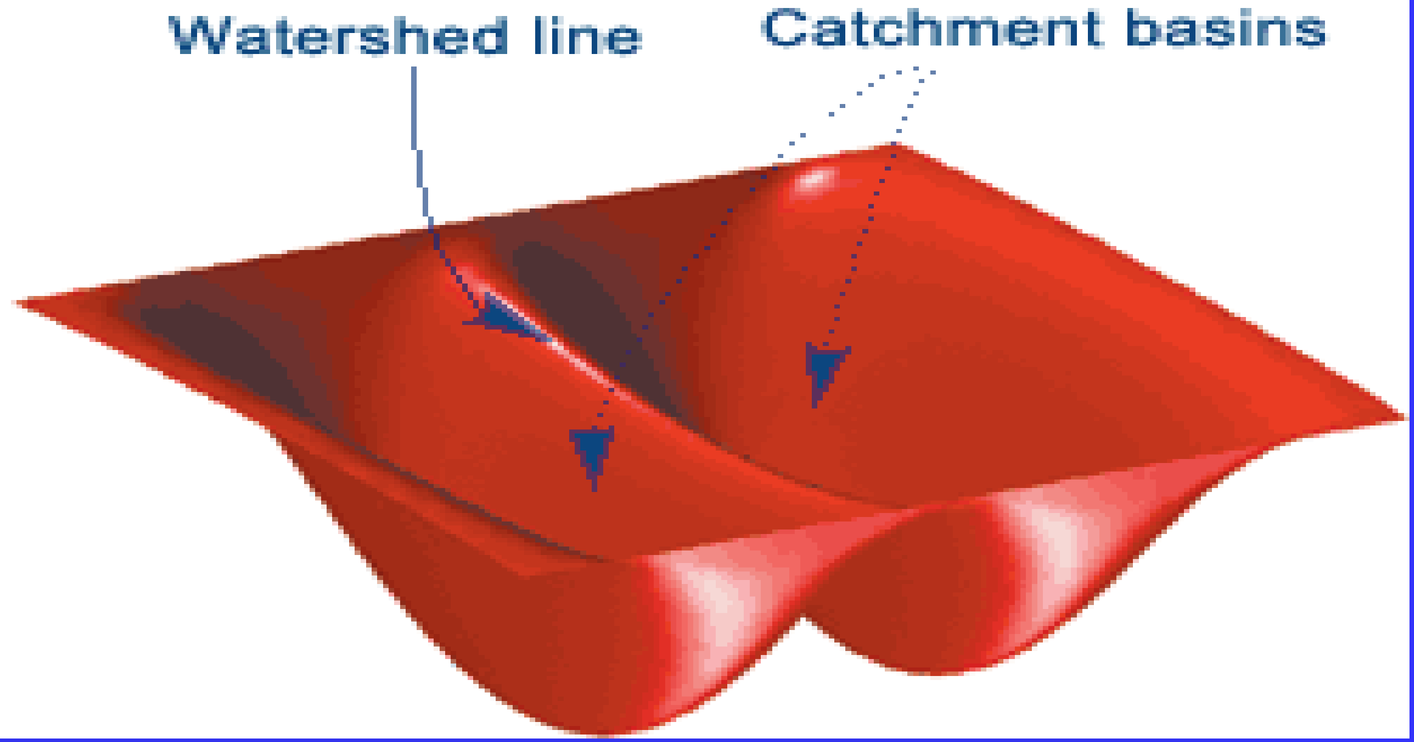
$$\text{Thinning}(I, se1, se2) = I \setminus \text{hit\_or\_missing}(I, se1, se2)$$

```
oln/morpho/thickening.hh :

Concrete(I_) morpho::thickening( const image<I_>& _input,
                                 const struct_elt<E1_>& _se1,
                                 const struct_elt<E2_>& _se2)
```

$$\text{Thickening}(I, se1, se2) = I \cup \text{hit\_or\_missing}(I, se1, se2)$$

Watershed : ligne de partage des eaux

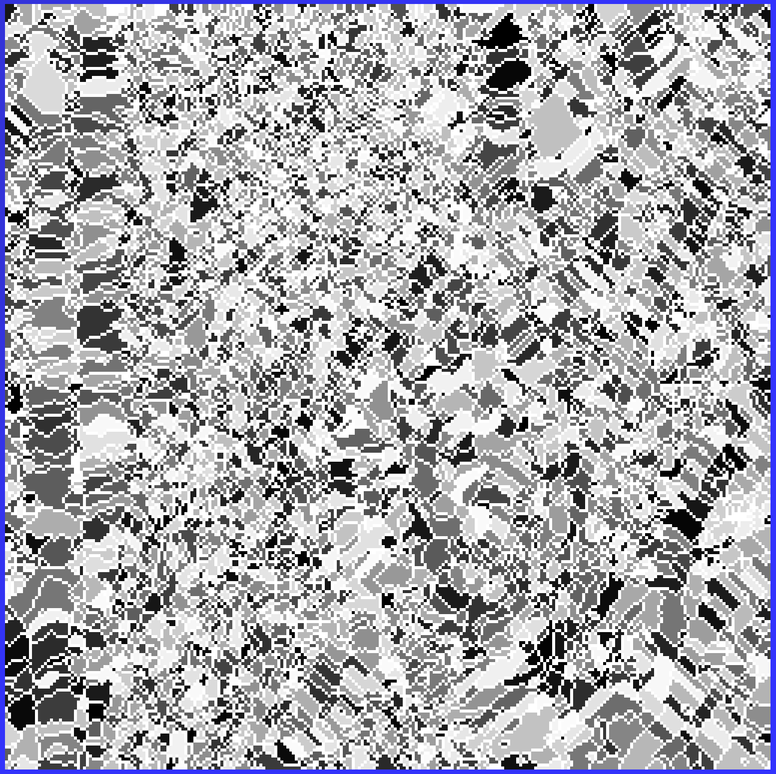


oln/morpho/watershed.hh :

```
template<class DestValue, class I, class N>
typename mute<I, DestValue>::ret
    morpho::watershed_seg(const image<I>& im_i,
                          const neighborhood<N>& Ng);
```

Exemple

```
Image2d<int_u8> im = load("lena.pgm");
save(morpho::watershed_seg(im, neighb_c4() ), "out.pgm");
```



Exemple

```
Image2d<int_u8> im = load("lena.pgm");
im = morpho::close(im, win_c8p() )
save(morpho::watershed_seg(im, neighb_c4() ), "out.pgm");
```