Appendix A

Distance Transforms

Distance measures are positive definite, symmetrical and satisfy the triangle inequality. Distances in the image can be approximated by the distances between neighbouring pixels. The well-known distance measures are City block and Chess board distance. The city block distance between two points $P = (x, y)$ and $Q = (u, v)$ is defined as: $d_4(P, Q) = |x - u| + |y - v|$. The chess board distance between $P, Q$ is $d_4(P, Q) = \max(|x - u|, |y - v|)$. The Euclidean distance between two points $P = (x, y)$ and $Q = (u, v)$ is defined as: $d_e(P, Q) = \sqrt{(x - u)^2 + (y - v)^2}$. The subscripts "4" and "8" indicate the 4-neighbour and 8-neighbour and "e" denotes Euclidean distance.

The chess board and city block methods are useful for distance transformation. They are very sensitive to the orientation of the object. Mathematical Morphological approach to distance transform is useful for the decomposition of global operations to local operations.
Properties of Lattices

1. In a lattice, the following properties hold:

Commutativity

\[ x \wedge y = \inf (x, y) = \inf (y, x) = y \wedge x. \]
\[ x \vee y = \sup (x, y) = \sup (y, x) = y \vee x. \]

Associativity

\[ x \wedge (y \wedge z) = \inf (x, (y \wedge z)) \]
\[ = \inf (x, \inf (y, z)) \]
\[ = \inf (x, y, z) \]
\[ = \inf (\inf (x, y), z) \]
\[ = \inf ((x \wedge y), z) \]
\[ = (x \wedge y) \wedge z. \]

Now, \[ x \vee (y \vee z) = \sup (x, (y \vee z)) \]
\[ = \sup (x, \sup (y, z)) \]
\[ = \sup (x, y, z) \]
\[ = \sup (\sup (x, y), z) \]
\[ = \sup ((x \vee y), z) \]
\[ = (x \vee y) \vee z. \]

Absorptivity

Now, \[ x \wedge (x \vee y) = \inf (x, x \vee y) \]
\[ = \inf (x, \sup (x, y)) \]
\[ = x \text{ [since } x \in \sup (x, y)] \]

and

\[ x \lor (x \land y) = \sup (x, x \land y) \]
\[ = \sup (x, \inf (x, y)) \]
\[ = x \text{ [since } \inf (x, y) \in x \}. \]

**Idempotency**

\[ x \land x = \inf (x, x) = x \text{ and } x \lor x = \sup (x, x) = x. \]

If a lattice \( L \) under the operations \( \land, \lor \) satisfies all the above properties then

\( (L, \land, \lor) \) is an algebraic lattice.

**Granulometry**

Convex structuring elements are scalable: Dilation / erosion with structuring

Element \( nB \) is equivalent to \( n \) dilations / erosions with structuring element \( B \).

Scalability is important for implementations in hardware. Rectangular structuring elements are separable. Multiple openings or closings with the same structuring element do not alter the image any more (idempotency).

Opening and closing are so-called sieve operations.
Using structuring elements of increasing size, one can remove small-, middle- and coarse-scale structures step by step. Such a morphological image decomposition into structures of different size is called granulometry.

**Pattern Spectrum**

Pattern Spectrum is known as granulometric size density. It is employed to measure the size distribution of an object.

Pattern spectrum $PS_{r_k}(F)$ of a set $F$ in terms of SE $r_k$ is defined as:

$$PS_{r_k}(F) = \begin{cases} Card((F \circ r_k) - (F \circ r_{k+1}K)), & i \geq 0 \\ Card((F \bullet r_k) - (F \bullet r_{k+1}K)), & i < 0 \end{cases}$$

where $Card(F)$ denotes the cardinality of set $F$

**Recursive Dilation**

Recursive Dilation is defined as: $F \oplus K = \begin{cases} F, & i = 0 \\ (F \oplus K) \oplus K, & i \geq 1 \end{cases}$ where $i$ is defined as scalar factor and $K$ as its base.

Recursive Dilation is employed to compose SE series in the same shape but different sizes.
Recursive Erosion

Recursive Erosion is also called successive erosion which is defined as:

\[ F \ominus_i K = \begin{cases} F, & i = 0 \\ (F \ominus (F \ominus K)) \ominus K, & i \geq 1 \end{cases} \]

When performing recursive erosions of an object, its components are progressively shrunk until completely disappeared. It is useful for distance transform and segmentation.

Fourier Transforms and Morphological Slope Transforms – A comparison

There exists a morphological system theory that resembles linear system theory.

The slope transform is the morphological analogue to the Fourier transform.

It transforms (tangential) dilation into addition. Parabolas / paraboloids as structuring functions are the morphological analogues to Gaussians in linear system theory. In linear system theory, Gaussian convolution plays a fundamental role: Gaussians remain Gaussians under the Fourier transform. Gaussians are the only separable and rotationally invariant convolution kernels.
Analog results for morphology:

Paraboloids remain paraboloids under the slope transform. Paraboloids are the only structuring functions that are separable and rotationally invariant. Morphological filters are invariant under monotonously increasing grey scale transformations. By replacing a grey value by its maximum or minimum within a neighbourhood, dilation and erosion are obtained. Dilation and erosion are used for shape analysis. Sequential combinations of erosion and dilation create openings and closings. They act as morphological low pass filters. Granulometries are examples for morphological band pass filters. Top hats result from computing differences between closing, original image, and opening. They act as morphological high pass fillers.
Operators In Software

Creating a structuring element

The following function creates structuring elements. It creates standard structuring element.

IplConvKernel* cvCreateStructuringElementEx(int cols,

    int rows,

    int anchor_x,

    int anchor_y,

    int shape,

    int* values=NULL)

where cols and rows is the number of columns and rows in the structuring element

anchor_x and anchor_y point to the anchor pixel. The pixel that is checked for when the transformation should be made or not.

shape - choose from three standard structuring elements.

CV_SHAPE_RECT
CV_SHAPE_CROSS

CV_SHAPE_ELLIPSE

CV_SHAPE_CUSTOM

Set `shape` to `CV_SHAPE_CUSTOM`, also supply the custom element. This is done using values. This parameter is used only if `shape` is set to custom.

`values` should be a 2D matrix, corresponding to the structuring element itself.

If `values` is NULL (and `shape` is custom), then all points in the structuring element will be considered nonzero (a rows*cols sized rectangle).

**Dilation**

This operation is the basic building block of morphology. The function is:

```c
void cvDilate(const CvArr* src,
    CvArr* dst,
    IplConvKernel* element=NULL,
    int iterations=-1);
```

The function takes four parameters:

`src`: The image to dilate

`dst`: This is where the dilated image is stored
element: (optional) The structuring element (use cvCreateStructuringElementEx to create one). If not specified, a 3×3 square is used.

iterations: (optional) Number of times to dilate src. If not specified, this is set to 1.

Use the same image as src and dst.

**Erosion**

Erosion is also a basic function of morphology. The function is:

```c
void cvErode(const CvArr* src,
             CvArr* dst,
             IplConvKernel* element=NULL,
             int iterations=1);
```

The parameters are the same as dilation. Perform erosion instead of dilation.

*src:* The image to erode

*dst:* This is where the eroded image is stored

*element:* (optional) The structuring element (use cvCreateStructuringElementEx to create one). If not specified, a 3×3 square is used.
\[ T \leq \varepsilon \]

*iterations:* (optional) Number of times to erode *src*. If not specified, this is set to 1.

This is also an in-place operation. *src* and *dst* can point to the same image.