2010 Spring (summer intensive) semester Special Lecture of Informatics
Topic 2. Mathematical morphology

Session 6. (3) Filter theorem / mathematical morphology and ordered set

This session presents two issues about mathematical morphology. Firstly, the *filter theorem* is explained; this theorem states that a wide range of image processing filters can be expressed by morphological and logical operations. The image processing filters will be explained again in the session on the relationship between image processing and neural networks in the next topic. Secondly, the concept of *ordered set* and its relationship to mathematical morphology are explained. The morphological operations for binary images, explained in the first session of this topic, are defined based on logical operations (AND and OR) of sets. The morphological operations for gray scale images are defined by expanding AND to infimum and OR to supremum. Defining morphological operations for color images requires the definition of infimum and supremum of vectors. The definitions of vector infimum and supremum are based on the concepts of ordered set and *lattice.*

Filter in the morphological sense

The *filter* in usual sense is defined as “an apparatus (containing, eg sand, charcoal, paper, cloth) for holding back solid substances in an impure liquid passed through it” (in *Oxford Advanced Learner’s Dictionary of Current English,* 1980). As an analogy, image processing filter is an algorithm that accepts a (corrupted) image and applies an operation, for example noise removal, to the image. The filter defined in the morphological sense is restricted to *translation-invariant* and *increasing* one. It is assumed that the “filter” hereafter satisfies these conditions.

An operation Ψ on a set (or an image) is *translation-invariant* if

$$[\Psi(X_b)]_{-b} = \Psi(X), \tag{1}$$

i. e. the effect of the operation is identical everywhere in the image. An operation Ψ is *increasing* if

$$X \subset Y \Rightarrow \Psi(X) \subset \Psi(Y), \tag{2}$$

i. e. the relationship of inclusion is preserved by the operation.

In the sense of mathematical morphology, *morphological filter* in broader sense is defined as all translationinvariant and increasing operations. Let us consider a noise removing filter for example; Since noise objects in an image should be removed wherever it is located, the translation-invariance is naturally required for noise removing filters. An increasing operation can express an operation that removes smaller objects and preserves larger objects, but cannot express an operation that removes larger and preserves smaller. Noise objects are, however, usually smaller than meaningful objects. Thus it is also natural to consider increasing operations only.

The morphological filter in narrower sense is all translation- invariant, increasing, and *idempotent* operations. An operation Ψ is defined idempotent if

$$\Psi\left[\Psi(X)\right] = \Psi(X), \tag{3}$$

i. e. iterative operations of Ψ is equivalent to one operation of Ψ. The opening and closing are the most basic morphological filters in narrower sense.

Filter theorem

The filter theorem states that all morphological filters can be expressed by OR of erosions. Let $\Psi(X)$ be a filter on the image X. The theorem states that the following holds for all $\Psi(X)$:

$$\Psi(X) = \bigcup_{B \in Ker[\Psi]} X \ominus \check{B}. \tag{4}$$

Here the set family $Ker[\Psi]$ is called *kernel* of filter Ψ, defined as follows:

$$Ker(\Psi) = \{X | \mathbf{0} \in \Psi(X)\}, \tag{5}$$

where “**0**” indicates the origin of X.

The following relationships hold for filters and their kernels:

$$Ker(\Psi_1) = Ker(\Psi_2) \Leftrightarrow \Psi_1 = \Psi_2. \tag{6}$$

Equation (6) indicates that a filter is expressed uniquely by its kernel.

The proof of the filter theorem in Eq. (4) is presented in the following. A more general proof is found in Chap. 4 of [2].

Let us consider an arbitrary vector (pixel) $h \in X \ominus \check{B}$ for a structuring element $B \in Ker[\Psi]$. From the definition of $X \ominus \check{B}$, $B_h \subseteq X$. Consequently, $B \subseteq X_{-h}$. Since Ψ is increasing, the relationship $B \subseteq X_{-h}$ is invariant by filter Ψ. Thus we get $0 \in \Psi(B) \Rightarrow 0 \in \Psi(X_{-h})$. Since Ψ is translation-invariant, we get $0 \in \Psi(B) \Rightarrow h \in \Psi(X)$ by translating $0 \in \Psi(X_{-h})$ by h. From the above discussion, $h \in X \ominus \check{B} \Rightarrow h \in \Psi(X)$ for all structuring element $B \in Ker[\Psi]$. Thus $\Psi(X) \supseteq \bigcup_{B \in Ker[\Psi]} X \ominus \check{B}$.

Let us consider an arbitrary vector $h \in \Psi(X)$. Since Ψ is translation-invariant, $h \in \Psi(X) \Rightarrow 0 \in \Psi(X_{-h})$. Thus we get $X_{-h} \in Ker[\Psi]$. Since $X \ominus \check{X}_{-h} = \{h' \mid (X_{-h})_{h'} \subseteq X\}$, and $\{(X_{-h})_{h'} \subseteq X\}$ is satisfied if $h' = h$, we get $h \in X \ominus \check{X}_{-h}$. By denoting X_{-h} by B, we get $h \in X \ominus \check{B}$.

Consequently, there exists a structuring element $B \in \mathrm{Ker}[\Psi]$ such that $h \in \Psi(X) \Rightarrow h \in X \ominus \check{B}$, i. e. any pixel in $\Psi(X)$ can be included in $X \ominus \check{B}$ by using a certain structuring element $B \in Ker[\Psi]$. Thus $\Psi(X) \subseteq \bigcup_{B \in Ker[\Psi]} X \ominus \check{B}$.

From the above discussion, it holds that $\Psi(X) = \bigcup_{B \in Ker[\Psi]} X \ominus \check{B}$.

The kernel is generally redundant and contains unnecessary sets for reconstruction of the filter. The family of necessary sets for the reconstruction of $\Psi(X)$ is called the basis.

Examples of morphological expressions of filters

We show some examples of morphological expressions of the median filter and the average filter, which are typical translation-invariant and increasing image processing filters. These filters calculate the median or the average of each pixel and its neighborhood pixels, and output this value as the pixel value at the same pixel position in the resultant image. The shape and size of the neighborhood are identical at every pixel in case of the translation-invariants filters, and the neighborhood is called *window*. The window is equivalent to the structuring element in morphological operations.

The median filter whose window size is n pixel is expressed as follows:

> ”the minimum of maximum (*or* of minimum) in every possible subwindow of $[n/2 + 1]$ pixels in the window.”

The operations deriving the maximum and minimum in each subwindow at every pixel are the set addition and set subtraction using the subwindow as the structuring element, respectively. Since the maximum and minimum operations are the fuzzy extensions of logical OR and AND operations, respectively, the median filter is expressed by the combination of morphological and logical operations, as shown in Figs. 1 and 2.

The simplest average filter operation, that is, the average of two pixel values f(x) and f(x+1), is expressed by the minimum and the supremum, as follows:

$$0.5[f(x) + f(x+1)] = \sup_{r \in \mathbb{R}}[\min\{f(x) - r, f(x+1) + r\}], \tag{7}$$

as shown in Fig. 3. This method of expressing the average of two values by the maximum and the minimum is equivalent to “the method to divide a cake into two even pieces,” found in some children’s quiz books. This method of serving a bar-type cake for two children P and Q is, if both of the children want to get the possibly largest piece, as follows:

1. P moves the knife slowly along the cake from an edge.
2. Q calls “stop” during the movement, and then P stops the knife.
3. P divides the cake at the current position of the knife, and P takes one of two pieces as P prefers.

Since P must take the larger piece of the two, Q calls “stop” so as to minimize the larger piece. Thus the cake is divided into two even pieces.

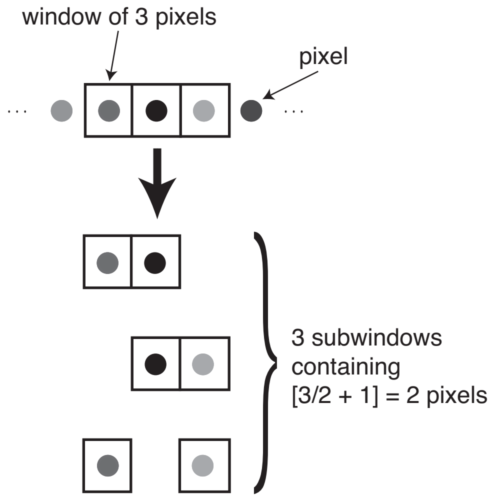

Fig. 1: Subwindows of $[n/2 + 1]$ pixels.

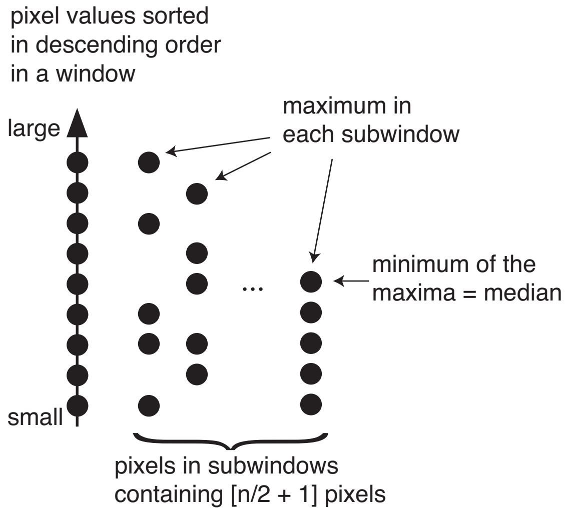
pixel values sorted in descending order in a window
large
small
maximum in each subwindow
...
minimum of the maxima = median
pixels in subwindows containing [n/2 + 1] pixels

Fig. 2: Median expressed by the maximum and minimum.

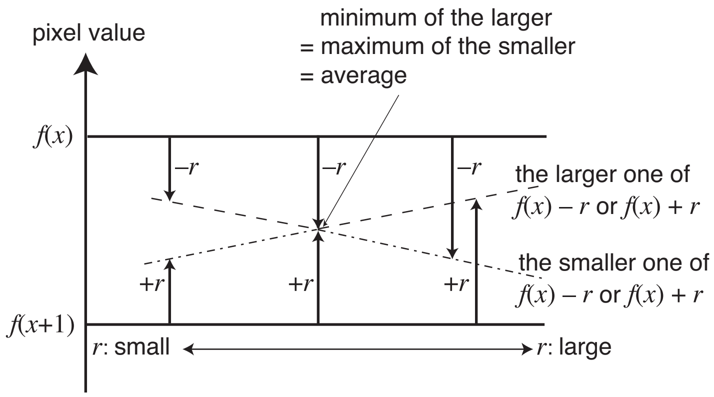
pixel value
minimum of the larger = maximum of the smaller = average
f(x)
f(x+1)
−r
−r
−r
+r
+r
+r
the larger one of f(x) − r or f(x) + r
the smaller one of f(x) − r or f(x) + r
r: small
r: large

Fig. 3: Average expressed by the maximum and minimum.

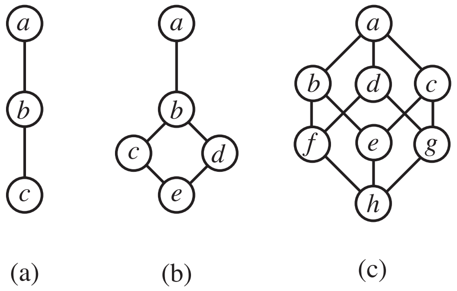

Fig. 4: Examples of lattice.

Color image and the concept of ordered set

Each pixel value of a color image is a vector whose elements indicate the brightness on a color basis like RGB, YIQ, etc. A definition of "maximum" and "minimum" of vectors is required for defining the morphological operations of color images.

A simple way of the definition is dividing the color image into each color plane, and applying the morphological operation to each plane separately using an identical structuring element. However, this method causes a color change of a specific object, since it is possible that an object is removed on a plane while the same object is preserved on another plane. This situation is out of "handling shape of objects," the essential concept of mathematical morphology.

There is a more sophisticated way that defines an "order" of vectors and the "maximum" and "minimum" of a set of vectors. Such set of vectors is called *ordered set.* The morphological operations on gray scale images are defined using the maximum and minimum of a subset of pixel values; Similarly, the morphological operations on color images can be defined if the supremum and infimum are defined for an arbitrary subset of vectors. Such algebraic system consists of an ordered set and the supremum and infimum operations is called *lattice.*

Definitions for ordered set and lattice [1]

A relationship "$\leq$" on a set X is called *ordering* or *partial ordering* if this relationship satisfies the following three properties:

1. *Reflectivity* — For all $x \in X$, $x \leq x$.
2. *Anti-symmetry* — For all $x, y \in X$, $x = y$ if $x \leq y$ and $y \leq x$.
3. *Transitivity* — For all $x, y, z \in X$, $x \leq z$ if $x \leq y$ and $y \leq z$.

X is called *semiordered set* (or *partially ordered set,* or *poset*) if an ordering is defined for some pairs of the elements of X. For an ordering "$\leq$," it is called that x is lower to y and y is *upper* to x when $x \leq y$ [1].

X is called *totally ordered set* if an ordering is defined for *all* pairs of the elements of X. All the elements of a totally ordered set can be arranged on a straight line in the defined order. For example, a subset of integers, for example possible values of gray scale pixels, is a totally ordered set with respect to the ordering $\leq$ (in ordinary sense).

For an element a of a subset A of an ordered set X,

> a is defined *maximal* of A if there is no element upper to a except a itself in A, and defined *minimal* if there is no element lower to a except a itself in A. In this case there can be an element whose ordering to a is not defined in A.
>
> a is defined *maximum* of A if a is *upper* to *all* elements of A, and defined *minimum* if a is *lower* to *all* elements of A.

The set of the elements of X that are upper [lower] to all the elements of A is defined *upper bound* [*lower bound*] of A. If the minimum [maximum] of the upper bound [lower bound] of A exists, it is defined *supremum* [*infimum*] of A. Note that the supremum and infimum are not always an element of A. If the supremum [infimum] is an element of A, it is equivalent to the maximum [minimum] of A.

If the upper bound and lower bound are defined for all combinations of two elements of X, and if the upper bounds and lower bounds are always in X, the algebraic system consisting of the set X and the operations to define the upper bound and lower bound is called *lattice.*

Figure 4 is called *Hasse diagram,* which illustrates orderings. Figures 4(a)(b) and (c) are examples of lattices. Figure 4(a) is a totally order set, and its Hasse diagram is a straight line. In Fig. 4(b), note that the upper bound of the elements c and d is not c or e, but b. Figure 4(c) shows a lattice composed by assigning an ordering on the vertices of a cube.

Lattice and morphology on color images

It follows from the above discussion that the morphological operations can be defined if the ordering of the vectors for color pixel values is defined to compose a lattice. There are several vector spaces to define color vectors , for example the RGB system as well as the YIQ system based on the brightness and color difference. Generally the ordering is defined by a linear combination of vector elements. However, researches of this area are in progress and various methods are being proposed. The lattice is also used for a general mathematical formulations of the mathematical morphology [2].

References

[1] 小倉久和，情報の基礎離散数学，近代科学社，ISBN4-7649-0276-1 (1999).

[2] H. J. A. M. Heijmans, *Morphological Image Operators,* Academic Press (1994). ISBN0-12-014599-5

[1] Although the symbol "$\leq$" is often used to express an ordering, it is not related to the inequality symbol in the ordinary mathematical sense. The terms *upper* and *lower* only have the meaning that y is defined *upper* if $x \leq y$.