Theoretical interpretation of morphological filters for the reconstruction of digital images

Dott.ssa Kusan Una
Centre of Excellence TeleGeomatics, University of Trieste, Italy

Keywords: filters, morphological operations, connected components, Find-Union

ABSTRACT: This paper has the aim to exam the morphological filters, in particular the filters for the elaboration of the images, under the point of view of morphological mathematics and to give a theoretical contribution in this sense. It introduces the representation of the classical linear and not linear filters in terms of morphological correlation. The morphological operations of base as erosion, expansion, opening and closing are used for the building of morphological filters of the type function-processing (FP). Is important to give an analytical and geometrical quantification to understand the similitudes and the differences between morphological filtering of the signals like as to give the definitions of the convulsion and the morphological correlation. The base concepts of elaboration of the images will be examined, and we will pass to the knowledges of mathematical morphology, explaining the classical working of the morphological "base" operators. We will see the classification of the filters under structural point of view. At the end we will show the morphological operations for the building of a filter called "Find-Union" and we will compared it with the other filters of the same kind, and we will finish with an extent of same filter for the treatment of the particular cases of morphological reconstruction of the images.

1 GENERAL IMAGE PROCESSING

A digital image $a[m,n]$ described in a 2D discrete space is derived from an analog image $a(x,y)$ in a 2D continuous space through a sampling process that is frequently referred to as digitization. The 2D continuous image $a(x,y)$ is divided into $N$ rows and $M$ columns. The intersection of a row and a column is termed a pixel. The value assigned to the integer coordinates $[m,n]$ with \( m=0,1,2,...,M-1 \) and \( n=0,1,2,...,N-1 \) is $a[m,n]$. In fact, in most cases $a(x,y)$--which we might consider to be the physical signal that impinges on the face of a 2D sensor--is actually a function of many variables including depth ($z$), colour ($\lambda$), and time ($t$). Unless otherwise stated, we will consider the case of 2D, monochromatic, static images.

The value assigned to every pixel is the average brightness in the pixel rounded to the nearest integer value. The process of representing the amplitude of the 2D signal at a given coordinate as an integer value with $L$ different gray levels is usually referred to as amplitude quantization or simply quantization.
The types of operations that can be applied to digital images to transform an input image \( a[m,n] \) into an output image \( b[m,n] \) (or another representation) can be classified into three categories as shown in Table 1.

### Table 1. Types of image operations. Image size = \( N \times N \), neighbourhood size = \( P \times P \).

<table>
<thead>
<tr>
<th>Operation</th>
<th>Characterization</th>
<th>Generic Complexity/Pixel</th>
</tr>
</thead>
<tbody>
<tr>
<td>* Point</td>
<td>the output value at a specific coordinate is dependent only on the input value at that same coordinate.</td>
<td>constant</td>
</tr>
<tr>
<td>* Local</td>
<td>the output value at a specific coordinate is dependent on the input values in the neighborhood of that same coordinate.</td>
<td>( P^2 )</td>
</tr>
<tr>
<td>* Global</td>
<td>the output value at a specific coordinate is dependent on all the values in the input image.</td>
<td>( N^2 )</td>
</tr>
</tbody>
</table>

Neighbourhood operations play a main role in modern digital image processing. It is therefore important to understand how images can be sampled:

- **Rectangular sampling** - In most cases, images are sampled by laying a rectangular grid over an image as illustrated in Figure 1a. This results in the type of sampling shown in Figure 3ab.
- **Hexagonal sampling** - An alternative sampling scheme is shown in Figure 1c and is termed hexagonal sampling.

Local operations produce an output pixel value \( b[m=m_0, n=n_0] \) based upon the pixel values in the neighbourhood of \( a[m=m_0, n=n_0] \). Some of the most common neighbourhoods are the 4-connected neighbourhood and the 8-connected neighbourhood in the case of rectangular sampling and the 6-connected neighbourhood in the case of hexagonal sampling illustrated in Figure 1c.

![Rectangular and Hexagonal Sampling](image)

**Figure 1.** Rectangular sampling, hexagonal sampling 4-connected 8-connected 6-connected

Certain tools are central to the processing of digital images. These include mathematical tools such as *convolution*, *Fourier analysis*, and *statistical* descriptions, and manipulative tools such as *chain codes* and *run codes*.
2 FILTER CLASSIFICATION

2.1 Linear filters

As digital images are two dimensional signals, two dimensional filtering operations are usually performed. For linear filters, the implementation is a 2-d convolution. A 2-d convolution is merely a weighted sum of pixel in the neighbourhood of the pixel to be processed. The weights are given by a small matrix called convolution kernel or mask. The following table shows some simple filter masks.

Table 2. Filter masks

<table>
<thead>
<tr>
<th>LPF</th>
<th>LPF</th>
<th>HPF</th>
<th>HPF</th>
<th>HPF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1 1</td>
<td>1 2 1</td>
<td>1 2 1</td>
<td>2 1 0</td>
<td>1 0 -1</td>
</tr>
<tr>
<td>1 1 1</td>
<td>2 4 2</td>
<td>0 0 0</td>
<td>1 0 1</td>
<td>2 0 -2</td>
</tr>
<tr>
<td>1 1 1</td>
<td>1 2 1</td>
<td>-1 -2 -1</td>
<td>0 -1 -2</td>
<td>1 0 -1</td>
</tr>
</tbody>
</table>

2.1.1 Non-linear filters

One important non-linear filter is the the median filter which is used to eliminate impulse noise. Using the median filter, the current pixel intensity is replaced by the median value of its neighbouring pixels. There are different ways for selecting neighbouring pixels. Some patterns are shown below:

Figure 2. Neighbourhood patterns

A notable property of the median filter is that it does not blur the edges of an image as much as a linear low pass filter does. It is also not effective as a linear low pass filter in filtering out Gaussian noise. A more general type of non-linear filter is called the rank filter which ranks the intensity of the neighbouring pixels and replaces the value of the current pixel with the kth-ranked value. Median filter is a special type of rank filter.

3 MORPHOLOGICAL OPERATORS

Erosion shrinks the boundary of a binary image to be smaller, while dilation expands the boundary to be larger. Usually the shrinkage or growth is by one pixel, but greater changes are possible with the recipe provided by the structuring element.
The structuring element M is a square matrix of binary values that is applied to the image I with the following rule for the eroded output. Erosion rule: replace the central point of the mask M with the smallest value of I covered by the mask. Assuming I is a binary image, then the mask centered at the boundary of I will output zero because the mask overlaps both 0s (outside boundary) and 1s (inside boundary).

Examples of eroded objects are given in Figure 3 for different structuring elements.

Large structuring elements tend to smooth boundary features, small ones tend to preserve boundary shape, and shaped structuring elements preserve similar appearing features in the boundary.

In a similar fashion, the rule for dilation is to replace the central point of the mask M with the largest value of I covered by the mask. If I is a binary image then this value just outside the boundary will be one because the mask centered one pixel outside the boundary overlaps both 0s (outside boundary) and 1s (inside boundary).

Dilation examples are given in Figure 4 for different structuring elements.

One of the uses for dilation and erosion is to eliminate isolated spots in an image. The opening algorithm is one erosion followed by one dilation:
\[ I \ast M = (I \ominus M) \oplus M \]

If the fundamental operations are applied in opposite order, i.e., dilation followed by erosion then holes in objects are filled; this is a closing procedure.

\[ I \bullet M = (I \oplus M) \ominus M \]

4 UNION-FIND

In mathematical morphology, connected set operators, most importantly preservation of shape. The earliest members of this class were openings and closing by reconstruction, for which an efficient algorithms were developed. An important development was the introduction of area openings and closing. The theory of area operators is given briefly here. We will first discuss binary area openings and closing and then the extension to the gray scale case. Binary area openings and closing are based on binary connected openings. Let the set \( X \subseteq M \) denote a binary image with domain \( M \). The binary connected opening \( G(X) \) of \( X \) at point \( x \in X \) and 0 otherwise. \( G \) extracts the connected component to which \( x \) belongs, discarding the others.

The binary area opening can be defined as:

**Definition:** Let \( X \subseteq M \) and \( \lambda \geq 0 \). The binary area opening of \( X \) with scale parameter \( \lambda \) is given by

\[ G(X) = \{ x \in X : A(G(X)) \geq \lambda \} \]

The binary area closing is defined by duality

\[ F(X) = C[G(C(X))] \]

The definition of an area opening of a gray-scale image \( f \) is usually derived from binary images \( T(f) \) obtained by thresholding \( f \) at \( h \).

\[ T(f) = \{ x \in M : f(x) \geq h \} \]

We have to define the flat zone \( L \) at level \( h \) of a gray-scale image \( f \) as a connected component of the set of pixels \( \{ p \in M : f(p) = h \} \). A regional maximum \( Mh \) at level \( h \) is a level component no members of which have neighbours larger than \( h \). A peak component \( Ph \) at level \( h \) is a connected component of the thresholded image \( T(f) \). At each level \( h \), there may be several such components, which will be indexed as \( Lh, Ph, Mh \), respectively. Any regional maximum \( Mh \) is also a peak component but the reverse is not true.

The Union-Find method is able to process multiple peak components simultaneously. Pixels are processed in gray level order. During this process, peak components are created and merged as needed, while keeping track of their areas. Once a peak component has an area of at least \( \lambda \), it ceases to grow.

For each set, an arbitrary member is chosen as representative for that set. The algorithm uses rooted trees to represent sets in which the root is chosen as the representative. Each non root node in a tree points to its parent, while the root points to itself.

There are four basic operations:

- **MakeSet** (x) Create a new singleton set \( \{ x \} \).
- **FindRoot** (x) Return the root of the tree containing x.
- **Union** (x, y) Form the Union of the two sets that contain x and y.
- **Criterion** (x, y) Determines the whether x and y belong to the same set.
The disjoint sets that we have to find are all flat zones which are not altered by the area opening, and for all other flat zones, the smallest peak component which has area $\lambda$ or more. The FindRoot routine also performs path compression. At the end of this part of algorithm we have found two kinds of disjoint sets: 1) those with constant gray level and ii) those with varying gray level. The most memory efficient approach is to store the output image in the parent array, because we can visit the pixels in reverse processing order. For area closing we have to change each test for gray level in the main loop of the routine to test for smaller than.

The algorithm can be extended also to compute other attribute opening and closing.

5 EXAMPLE

The image used to perform the algorithm is an orthophoto of Mestre (Venice). The software used is MatLab 6.1.

Input

- $f$: binary image (eight logical unit).
- $Bc$: Structural element (connected).

Output

- $y$: A binary image or gray level image (eight or sixteen logical unit).

The Matlab algorithm can have this structure:

```matlab
function y = mmlabeleq(f, Bc)
    faux=f;
    fmark=mmsubm(f,f);
    r=fmark;
    label=1;
    y=mmgray(f,uint16,0);
    while ~mmisequal(faux,logical(uint8(0)))
        x=find(faux);
        fmark(x(1))=1;
        r=mminfrec(fmark,faux,Bc);
        faux=mmsubm(faux,r);
        r=mmgray(r,'uint16',label);
        y=mmunion(y,r);
        label=label+1;
    end
```

D. Kusan Una
6 CONCLUSION

One the most important tasks in computer Vision and image processing applications is to find the various components in the captured image. The image can be represented in a number of ways like array, bintrees, octrees and so on. The problem of finding connected components can be treated as transforming the source image. The transformed image contains a unique label for each component in the source image.

The union-find algorithm for computation of area openings and closing has clear advantages in terms of computing time, especially using large values of $\lambda$. In the case of opening is also very efficient in terms of memory usage, especially if the original image may not be overwritten.

ACKNOWLEDGEMENT

I would like to acknowledge and thank the professor Francesco Gosen for his generous support of the research project.

REFERENCES


