

# Applied Research of Convolution and Correlation in Digital Image Processing

Xihu Zhi<sup>1, a</sup> and Aidi Zhi<sup>2</sup>

<sup>1</sup>College of Information Technology, Luoyang Normal University, Henan Luoyang, 471934, China

<sup>2</sup>College of Physics and Electronic Information, Luoyang Normal University, Henan Luoyang, 471934, China

<sup>a</sup>zhixihuedu@163.com

**Keywords:** Convolution; Correlation; Digital image processing; Realize through MATLAB

**Abstract.** Convolution and correlation operations are important tools to solve application problems in the field of signal processing, digital image filtering, earthquake detection, etc. As of today, however, there are no materials that can systematically and completely cover the concept and the specific application of convolution and correlation. From the perspective of mathematics and application with digital image processing, this article presents a detailed explanation on the concept and specific application of convolution and correlation, to serve as a reference for relevant study and applications.

## Introduction

Convolution and correlations are mathematical operations, which are important tools to solve application problems in the field of signal processing, digital image filtering, earthquake detection, etc. It is very important to master these mathematical operations in order to solve pertinent problems in application field.

## The Concept of Convolution and Correlation

### 1.1 Convolution

Convolution is a mathematical operation that combines two functions  $f$  and  $g$  to produce a third function. It is defined as the integral of the product of the two functions after  $g$  is reversed and shifted, it can be represented by the area overlap between the two functions [1]. The mathematical expression is as follows.

$$h(t) = \int_{-\infty}^{+\infty} f(\tau)g(x - \tau)d\tau \quad (1)$$

The function  $f(x)$  is known,  $g(x)$  is in general manually set depending on the application. The function  $h(x)$  is the convolution of the functions  $f(x)$  and  $g(x)$ . In real application, convolution can be used to represent the effect process on the input signal of detection system or observation equipment.

For scattering cases, the expression is:

$$h(t) = \sum_{-\infty}^{+\infty} A(j)B(x - j) \quad (2)$$

### 1.2 Correlation

Correlation is similar to convolution, except that  $g$  is not reversed. Correlation is normally used to measure the similarity of two functions relative to each other, common applications include signal detection. The mathematical expression is as follows.

$$h(t) = \int_{-\infty}^{+\infty} f(\tau)g(x + \tau)d\tau \quad (3)$$

For scattering cases, the expression is:

$$h(t) = \sum_{-\infty}^{+\infty} A(j)B(x+j) \quad (4)$$

### 1.3 One-Dimensional Convolution

Array  $A = \{\dots, a_1, a_2, a_3, \dots\}$  is known. Given array  $B = \{b_1, b_2, b_3\}$ , becomes  $\{b_3, b_2, b_1\}$  after reverse. The computational process of convolution  $C = A * B$  is shown in Fig. 1.

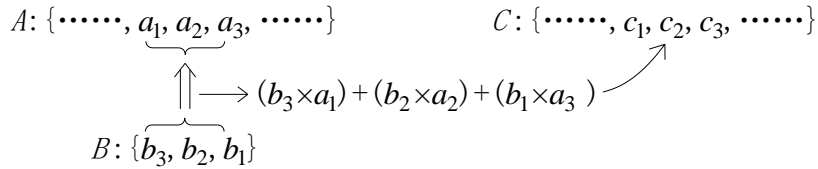


Figure 1. One-Dimensional Convolution Computation

Let  $A = \{0, 1, 2, 3, 4, 3, 2, 1, 0\}$  and  $B = \{1, 3, -1\}$ , and the computational process of  $A * B$  is as follows

1) Reverse B and align its center number with the first number of array A, result of the first step in convolution  $A * B$  is  $(0 \times (-1)) + (0 \times 3) + (1 \times 1) = 1$ .

where the empty point is assumed as 0.

2) Move array B one space to the right, result of the second step in convolution  $A * B$  is:

$$(0 \times (-1)) + (1 \times 3) + (2 \times 1) = 5$$

$$(0 \times (-1)) + (1 \times 3) + (2 \times 1) = 5$$

3) Move array B to the right one space after another, the final result of convolution  $A * B$

### 1.4 Two-Dimensional Convolution

Given input image A and the  $n \times n$  convolution template B ( $n = 3$  in this case), then the convolution  $A * B$  is shown in Fig. 2.

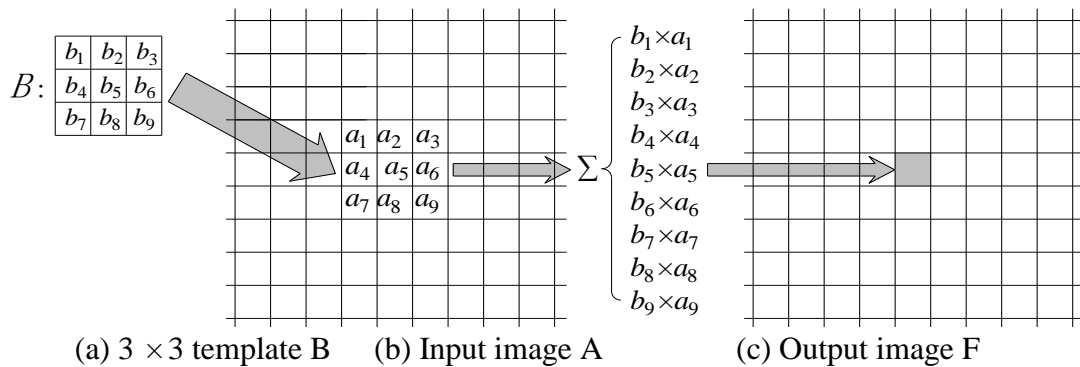


Figure 2. Two-Dimensional Convolution Computation

If the values of A and B are as follows:

$$A = \begin{pmatrix} 4 & 7 & 3 & 5 & 6 & 3 & 2 & 6 \\ 5 & 3 & 2 & 2 & 8 & 5 & 2 & 0 \\ 0 & 8 & 4 & 3 & 0 & 5 & 8 & 5 \\ 3 & 2 & 6 & 3 & 2 & 9 & 0 & 3 \\ 9 & 6 & 8 & 9 & 1 & 4 & 3 & 2 \\ 6 & 3 & 2 & 8 & 5 & 2 & 0 & 8 \\ 2 & 7 & 6 & 5 & 8 & 3 & 2 & 9 \\ 3 & 6 & 2 & 3 & 2 & 5 & 4 & 7 \end{pmatrix} \quad B = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 1 & -1 \\ 0 & -1 & -2 \end{pmatrix}$$

(5)

Then their convolution is:

$$A*B = \begin{pmatrix} 22 & 13 & 7 & 26 & 22 & 8 & 7 & 4 \\ 20 & 1 & -6 & 0 & 5 & 5 & 7 & -1 \\ 10 & 13 & 3 & 0 & 10 & 1 & 2 & -4 \\ 26 & 19 & 13 & -1 & 11 & 12 & -17 & -16 \\ 24 & 4 & 19 & 5 & -3 & -5 & -1 & 4 \\ 16 & -6 & 3 & 7 & -6 & -2 & 15 & 9 \\ 18 & 6 & 4 & 2 & -3 & -2 & 22 & 6 \\ 7 & -6 & -21 & -14 & -14 & -12 & -2 & -10 \end{pmatrix} \quad (6)$$

Note that before computing  $A*B$ , reversed  $B$  both horizontally and vertically (or rotate  $B$   $180^\circ$  clockwise about the center element), then carry out convolution computation.

Fig. 3 shows the method and result of how the upper left element is derived for  $A*B$ .

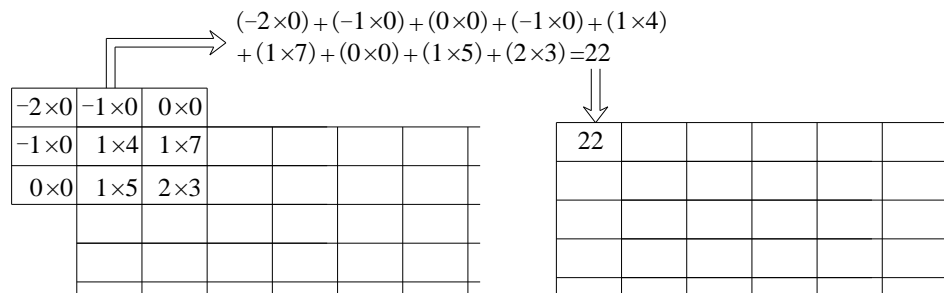


Figure 3. Example of Two-Dimensional Convolution Computation

## Processing Strategies for Boundary Points in Image Processing Using Convolution and Correlation

When processing images using convolution and correlation, the data overlay of template data on input data is divided to complete overlay and partial overlay, as shown in 错误!未找到引用源。 . The partially overlaid data are normally the boundary points of the input image, how these boundary points are processed, influences the visual effect of the resultant image, or affect the subsequent processing of the resultant image [2]. Therefore, comprehensive consideration is necessary based on specific application needs.

Depending on the actual application needs, the following strategies can be adopted for boundary points processing.

### 2.1 Neglect the Boundary Points

There are two ways to neglect the boundary points:

1) The values of the partially overlaid points are unchanged [3]. This processing method may produce image artifacts due to the difference between the output processed pixels and the unprocessed pixels.

2) The values of the partially overlaid points are substituted with assigned constant. If assigned constant is not zero, the resultant image may contain artifacts at the boundary points. If the assigned constant is zero, the resultant image becomes smaller than the original image, and this affects the subsequent processing.

### 2.2 Expand the Boundary Points

This method expands the data points at the boundary of the input image to allow partially overlaid data to become completely overlaid data [4]. The boundary point's expansion can be realized as follows.

1) Zero-value expansion. All values outside the image are assumed to be zero. This method causes the boundary of the processed image to darken. The darkened width is directly proportional to the convolution template size.

- 2) Equivalent value expansion. Values outside the image are assumed to be equivalent to the image boundary values. This method reduces the impact of image artifacts.
- 3) Mirror image expansion. Values outside the image are assumed to be the same as the row or column values near the boundary of the mirror image.
- 4) Periodical expansion. The image is assumed to be a two-dimensional periodical function, the pixel values appear periodically in horizontal and vertical directions.

### Image Filtering Method Using Convolution

In MATLAB, different filtering result for an input image can be achieved using different convolution template [5]. The designed template is called the filter. Linear filtering is a commonly used method, it is normally realized through the combination of functions `imfilter()` and `fspecial()`. The call format of the two functions is as follows.

1) Call format of function `imfilter()`

`g=imfilter(f,h,mode,boundary_options,size_options);`

Where:

f: input image

h: filter template

Mode: mode option, the value is 'conv' or 'corr', representing convolution or correlation (default)  
filter template

boundary options: defines how boundary points are processed, with 4 options

① X: input image boundary points are expanded using filler X, the default value of X is 0.

② 'symetric': input image boundary points are expanded with mirror image.

③ 'replicate': input image boundary points are expanded with equivalent values.

④ 'circular': input image boundary points are expanded in a periodical pattern.

size\_options: defines the resultant image dimension. The value is 'full' or 'same'.

① 'full': defines the output image dimension is the expanded image dimension.

② 'same': defines the output image dimension is the same as the input image dimension.

g: output image.

2) Call format of the function `fspecial()`

`h=fspecial(type,parameters);`

where:

h: the filter template derived from the function `fspecial()`, and serves as the parameter for the function `imfilter()`

type and parameter values and their correlations.

### Application Example of Image Filtering

Below is an application example of image filtering using convolution: a given image goes through motion fuzzy simulation using `fspecial()` and `imfilter()`, and Wiener filtering restoration using `deconvwnr()` [6]. MATLAB codes are as follows:

Clear all; close all; clc; % close all image windows, clear all work space

% parameters clear all command lines

I=imread('lenna.bmp'); % Read-in the image

I=rgb2gray(I); % Convert RGB image into grey scale

I=im2double(I); % Convert image data to double-precision

LEN=25; % Define motion displacement length as 25 pixels

THETA=20; % Define motion angle as 20 degrees

h=fspecial('motion',LEN,THETA); % Generate filter template

% generate motion fuzzy image using convolution filtering

J=imfilter(I,h,'conv','circular');

```
NSR=0;  
K=deconvwnr(J, h, NSR); % Restore motion fuzzy image using Wiener filter  
Figure;  
subplot(131); imshow(I); % Display original image  
subplot(132); imshow(J); % Display motion fuzzy image  
subplot(133); imshow(K); % Display restored image  
The program running result is shown in Fig. 4.
```



(a) Original image                      (b) Motion fuzzy image                      (c) Filter restored image

Figure 4. Application Example of Image Filtering

## Summary

Mathematics, as an important tool, its algorithm is abstract but practical in application. Being able to master the concept and be flexible in application are fundamental.

## References

- [1] R.C. Gonzalez, R.E. Woods, and S.L. Eddins. Digital Image Processing Using MATLAB, Pearson Prentice Hall, 2004.
- [2] R.C. Gonzalez and R.E. Woods, Digital Image Processing, 3rd edition, Prentice Hall, Upper Saddle River, NJ, 2008.
- [3] A.A. Goshtasby. 2-D and 3\_D Image Registration, Wiley, 2005.
- [4] E.B. Goldstein. Sensation and Perception, 7TH edition, Thomson Wadsworth, Belmont, CA, 2007.
- [5] T. Acharya and P.S. Tsai. JPGE2000 Standard for Image Compression: Concepts, Algorithms and VLSI Architectures, Wiley-Interscience, 2004.
- [6] R. Lukac. Guest editorial: special issue on applied color image processing: editorials. International Journal of Image Systems and Technology, 17(3):103-104, 2007.