Experimental Platform of SAR Image Processing Based on C#

Zeng-Guo SUN[1,a,*], Xiao-Peng YAN[2,b], Rui SHI[3,c], Ruo-Nan CHEN[4,d],and Cong WANG[5,e]

School of Computer Science, Shaanxi Normal University, Xi'an, China

[a]duffer2000@163.com, [b]yan14174@qq.com,
[c]1424022567@qq.com,[d]chenruonan@snnu.edu.cn,[e]wangcong373795225@vip.qq.com

Keywords: synthetic aperture radar image processing; summary design; detailed design; Visual C# development tools

Abstract; In order to help students to deepen the understanding of synthetic aperture radar (SAR) image processing algorithms, a SAR image processing experiment platform based on C# language is developed. Through this kind of simulation platform, the SAR image processing can be visualized and can choose different algorithms to enable students to vividly understand the SAR image processing algorithm differences and parameter adjustment corresponding to the effects, so as to help students better understand the algorithms. Then, the requirement analysis, summary design and detailed design of SAR image processing experiment platform based on C# are carried out, and then the system is coded and implemented by Visual C# development tool. The system can be used as an effective simulation platform and an auxiliary research tool for SAR image processing.

Introduction

With the development of modern technology, the processing of image has become very important for the image understanding and interpretation. There are many algorithms to solve such problems, and these algorithms are still being updated to meet different aspects of the demand. Synthetic aperture radar (SAR) is an active side-view radar system, and the imaging geometry belongs to the slanting projection type [1]. The information on the SAR image is the reflection of the radar of the target object to the radar beam, mainly the image information formed by the backscattering of the feature object. The SAR is based on the current resolution of the high resolution SAR system, which can reach the centimeter level. The SAR uses active microwave imaging, which has a certain penetrating effect and can effectively detect all kinds of camouflage targets. SAR has all-weather work characteristics, suitable for a variety of adverse battlefield conditions to complete the investigation task.

Requirement Analysis

As the SAR image processing algorithms are more abstract and more difficult to understand, many students cannot have a clear understanding of this. Through such a simulation platform, the SAR image processing algorithm can be visualized. Only by selecting a different button, you can intuitively simulate the SAR image processing such as despeckling and edge detection. You can also choose a different algorithm to demonstrate the performance of specific algorithms, which is helpful for students to vividly understand the SAR image processing algorithm differences and parameter adjustment corresponding to the effect and allows users to have a more intuitive understanding of the algorithm [2]. The system includes functions of image reduction and edge detection. The system is written by C# language, and the interface is clear and beautiful. According to the various types of image processing, this platform is divided into different operating windows, each of which contains the specific functions and the parameter panel required by the implementation. Such layout of interface is concise and clear. In addition, the operation window of this platform is composed of independent windows, and the images before and after the processing and the parameter setting panels occupy different windows respectively, which is convenient both for overall contrast operation and for independent observation. It can be seen that the C# SAR image processing experiment platform contains sufficient functions, has a good human-computer interaction, and meets the actual needs of users.

Summary Design

On the basis of the requirement analysis, the SAR image processing experiment platform is designed based on C#, and its purpose is to determine the structure frame and function module of the system. The structure of the SAR image processing experiment platform based on C# is shown in Fig. 1, including function modules of this system such as entering system module, system introduction module, algorithm description module, loading image module, processing image module and exiting function module. The algorithm description module contains descriptions of image denoising and edge detection, and the processing image module contains image denoising and edge detection. After entering the system module, click the icon, wait for the interface, and enter the main interface. Three functional labels appear, i.e., “system introduction”, “algorithm description”, and “start” from left to right. Button of system introduction means the detailed descriptions of this platform. Button of algorithm description denotes all kinds of algorithms used in this platform including noise reduction and edge detection. And button of start means the beginning of the platform implementation.

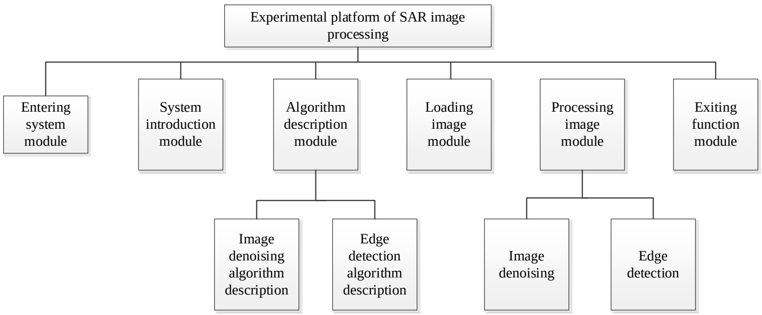

Experimental platform of SAR image processing
Entering system module
System introduction module
Algorithm description module
Loading image module
Processing image module
Exiting function module
Image denoising algorithm description
Edge detection algorithm description
Image denoising
Edge detection

Figure. 1. Block diagram of SAR image processing experiment platform

Detailed Design

The task of detailed design is to determine the specific implementation process of each function module. The basic operation flow chart of SAR image processing experiment platform is shown in Fig. 2. After entering the system main interface, first select the type of operation. If you choose the system description, the system displays the system description; if you select the algorithm description, enter the selected image reduction or edge detection algorithm, if you choose the image reduction, then enter the choice of traditional drop plaque algorithm or based on local statistical drop plaque algorithm. If you choose the traditional drop plaque algorithm, then enter the display of the traditional drop plaque algorithm, if you choose based on the local statistical drop plaque algorithm, then display based on local statistical drop plaque algorithm. If Edge Detection is selected, the Edge Detection algorithm is displayed. If you choose to start, enter the image reduction or edge processing, if you select the image reduction, then enter the image reduction feature interface, enter the loading image, enter the selection algorithm type, enter the specific algorithm and the corresponding parameters, enter the processing image The If you select edge processing, enter the edge processing interface, enter the load image, enter the selection algorithm type, enter the modified parameters, enter the processing image. The following image reduction function as an example, describes the basic operational procedures.

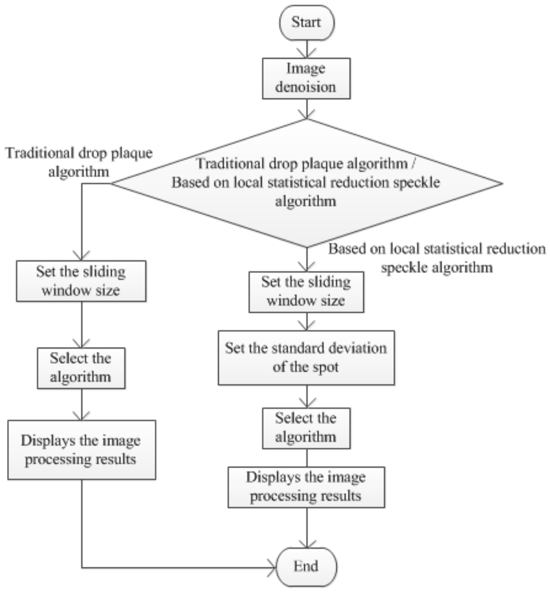

Start
Image denoision
Traditional drop plaque algorithm / Based on local statistical reduction speckle algorithm
Traditional drop plaque algorithm
Based on local statistical reduction speckle algorithm
Set the sliding window size
Select the algorithm
Displays the image processing results
Set the sliding window size
Set the standard deviation of the spot
Select the algorithm
Displays the image processing results
End

Figure. 2. Flow chart of image reduction function

Code Design

This platform is based on C# language written on Microsoft Visual Studio platform. C# is the mainstream development tool for Windows applications that can be exploited with rich resources. The system is written by C#, the interface is clear and beautiful. According to different processing classification, into the different operating window, each operation window of the specific function of the realization of the parameters through the panel on the independent label to distinguish, concise and clear. Among them, the operation window of the sorting processing is composed of independent windows, and the images before and after the processing and the parameter setting panels occupy different windows, respectively, and can be regarded as an overall comparison operation or independent observation. This section to Lee filter algorithm, for example, the user through the set of different parameters to achieve image down the class function, a detailed description of how to use Visual C# tool for coding [3,4,5], the code is as follows:

Lee filter:

```
public void LeeFilter(int windowSize,double canshu)
{
if (windowSize % 2 == 0) // Returns if the filter window length is even
{
return;
}
for (int i = 0; i < Height; i++)// The outer two layers cycle through all the points of the image and process them point by point
{
for (int j = 0; j < Width; j++)
{
//Find the local mean,ave=sum(Data[i,j])
  double sum = 0;
for (int g = -(windowSize - 1) / 2; g <= (windowSize - 1) / 2; g++)
// This two-layer loop traverses all the data in the window and sums it
{
     for (int k = -(windowSize - 1) / 2; k <= (windowSize - 1) / 2; k++)
    {
     int a = i + g, b = j + k;
     if (a < 0) a = 0;// Transboundary processing
       if (a > Height - 1) a = Height - 1;
   if (b < 0) b = 0;
   if (b > Width - 1) b = Width - 1;
     sum += Data[a, b];// Find the sum of the window data
   }
     }
   double  ave = (byte)(sum / (windowSize * windowSize));// Local mean
   // (Pow (Data [i, j] -ave, 2)) for the sum of the difference between the current data and the square of the
current window's local mean
   sum = 0;
   for (int g = -(windowSize - 1) / 2; g <= (windowSize - 1) / 2; g++)
   // The two layers loop through all the data in the window and ask the sum of Data [i, j] -ave                {
   for (int k = -(windowSize - 1) / 2; k <= (windowSize - 1) / 2; k++)
   {
   int a = i + g, b = j + k;// Transboundary processing
   if (a < 0) a = 0;
   if (a > Height - 1) a = Height - 1;
   if (b < 0) b = 0;
   if (b > Width - 1) b = Width - 1;
   sum += (Data[a, b] - ave) * (Data[a, b] - ave);
   // Data [i, j] -m [i, j] is the difference between the current data and the local mean value of the current
window
   }
   }
   //Seek standard deviation,bzc=sqrt(sum(pow(Data[i,j]-ave,2))/pow(windowSize))
   double bzc = Math.Sqrt( sum / (windowSize * windowSize));
   // sum/(windowSize * windowSize) is the variance,this step determines the current window standard
deviation
   double canshuCy=bzc / ave;
   if ( canshuCy<= canshu)// If the parameter Cy <= parameter k, the picture data is updated to the window
local mean
   Data[i, j] = (byte)ave;
   else
   {
   double canshuK=1-canshu*canshu/(canshuCy *canshuCy );// Find the parameters k[i,j]
   Data [i,j]=(byte)((1-canshuK )*ave+canshuK*Data [i,j]);
   // Calculate the results, update the picture data
     }
   }
   }
   }
```

Interface Design

Program Interface.

Program Main Interface.

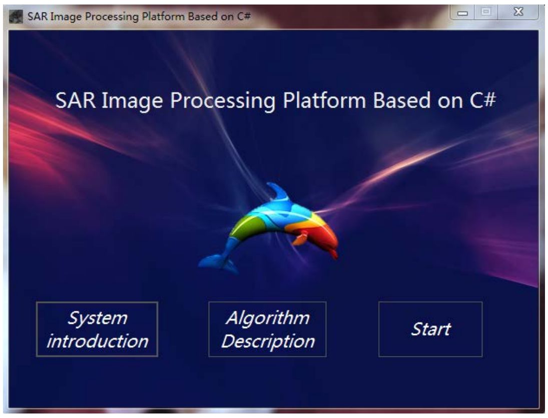
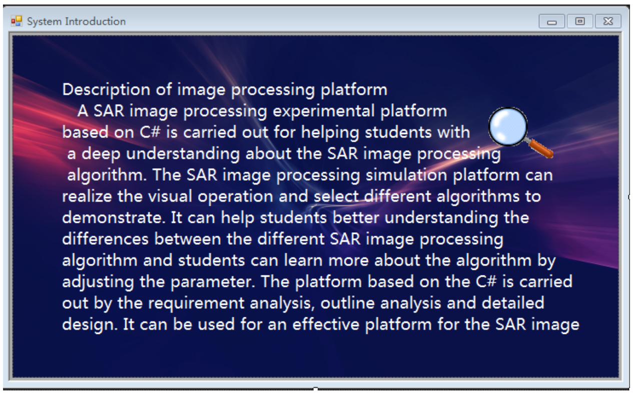

SAR Image Processing Platform Based on C#
SAR Image Processing Platform Based on C#
System introduction
Algorithm Description
Start

Figure. 3. Program main interface

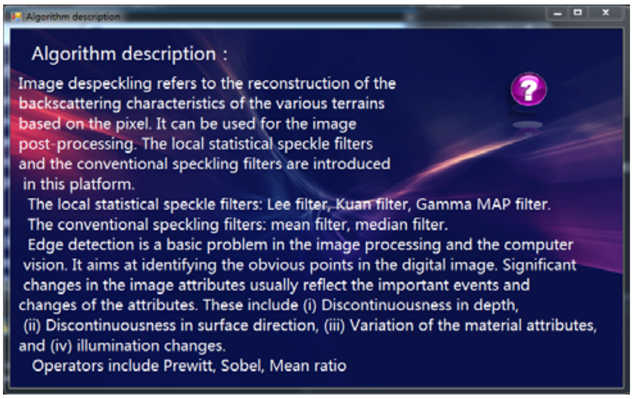
Program Sub-Interface One.



Description of image processing platform
A SAR image processing experimental platform based on C# is carried out for helping students with a deep understanding about the SAR image processing algorithm. The SAR image processing simulation platform can realize the visual operation and select different algorithms to demonstrate. It can help students better understanding the differences between the different SAR image processing algorithm and students can learn more about the algorithm by adjusting the parameter. The platform based on the C# is carried out by the requirement analysis, outline analysis and detailed design. It can be used for an effective platform for the SAR image

Figure. 4. Program sub-interface one

Program Sub-Interface Two.



Algorithm description :

Image despeckling refers to the reconstruction of the backscattering characteristics of the various terrains based on the pixel. It can be used for the image post-processing. The local statistical speckle filters and the conventional speckling filters are introduced in this platform.

The local statistical speckle filters: Lee filter, Kuan filter, Gamma MAP filter.

The conventional speckling filters: mean filter, median filter.

Edge detection is a basic problem in the image processing and the computer vision. It aims at identifying the obvious points in the digital image. Significant changes in the image attributes usually reflect the important events and changes of the attributes. These include (i) Discontinuousness in depth, (ii) Discontinuousness in surface direction, (iii) Variation of the material attributes, and (iv) illumination changes.

Operators include Prewitt, Sobel, Mean ratio

Figure. 5. Program sub-interface two

The Algorithm Starts

In the main interface, double-click "Start", that is, enter the selection algorithm interface, as shown in Fig. 6. The user can select what kind of processing the image wants to do in this interface.

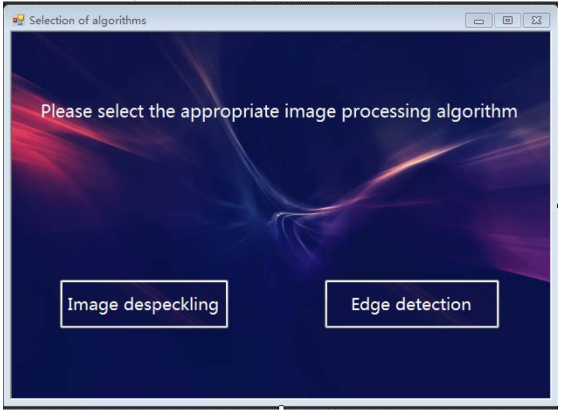


Figure. 6. Select the algorithm

Speckle Reduction

In the selection algorithm interface, double-click "image drop", you can enter the image reduction area. In this user interface, the user can load the SAR image that needs to be processed, freely select the algorithm type, selecting the specific algorithm in the type and the corresponding parameter [6,7].

Example 1. The user selects the SAR image to be processed in the local PC, selects the "mean filtering" algorithm in the traditional dropdown algorithm and adjusts the sliding window size to 9, and the processing result is shown in Figure. 7.

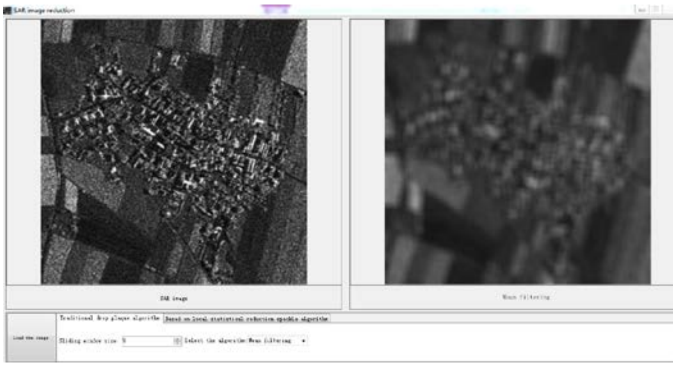
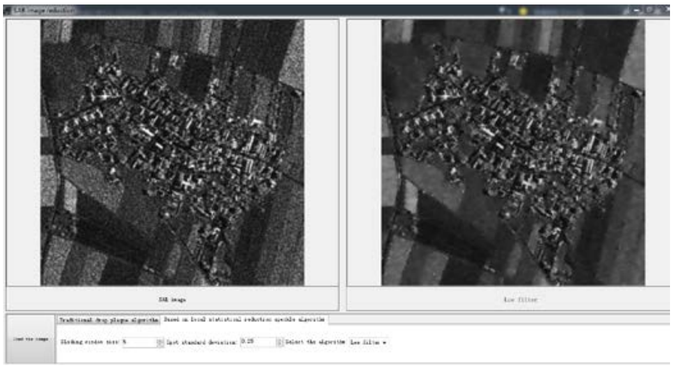


Figure. 7. Mean filter processing results

Example 2. We choose the "Lee filter" algorithm based on the local statistical reduction speckle algorithm and adjust the sliding window size to 5, and the standard deviation of the spot is 0.25, and the processing result is shown in Figure. 8. [8].


SAR image reduction
SAR image
Lee filter
Traditional drop plaque algorithm
Based on local statistical reduction speckle algorithm
Load the image

Sliding window size: 5

Input standard deviation: 0.25

Select the algorithm: Lee filter

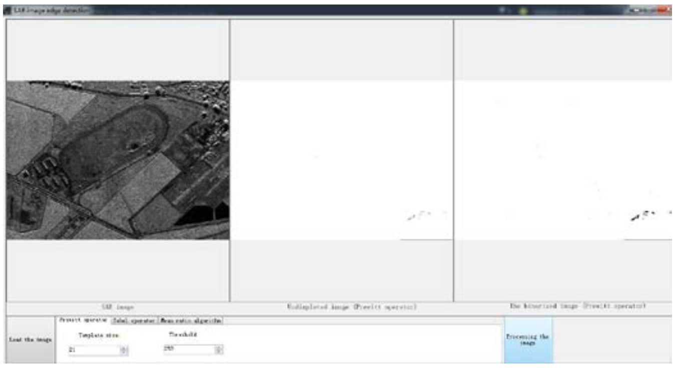
Figure. 8. Lee filter results

Therefore, we can see that in the image reduction speckle algorithm, Lee filter performance is better than the mean filter. Since the mean of the noise is not zero, the mean filter does not well remove the noise point.

Edge Detection

In the selection algorithm interface, double-click "edge detection", you can enter the edge detection interface. In this user interface, the user can load the SAR image that needs to be processed, choose the type of processing freely, and modify the corresponding parameter. Click "Process Image" to get the processed SAR image [9].

Example 3. The user selects the SAR image that needs to be processed in the local PC, selects "Prewitt operator", adjusts the template size to 21, the threshold is 253, and clicks on the "processed image" to obtain the intermediate binary edge image. The results are shown in Fig. 9.



Template size: 21

Threshold: 253

Figure. 9. Prewitt operator processing results

Example 4. The mean value ratio algorithm calculates the ratio of the mean of the different sub-windows in the sliding window, thereby recording and detecting the edge information. Select the "mean ratio algorithm", adjust the template size of 7, the threshold is 0.65, click on the "image" to get the middle of the binarized edge of the image, and the right of the binarized edge image, the results are shown in Fig. 10.

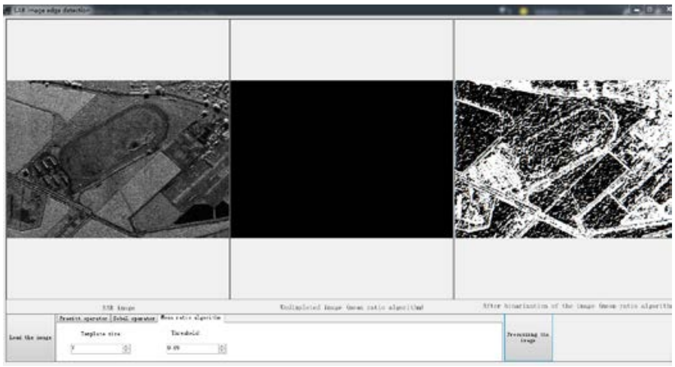


Figure. 10. Mean ratio algorithm processing results

Therefore, we can see that in the edge detection algorithm, the mean ratio algorithm is superior to the performance of the Prewitt operator.

Summary

In order to better understand the SAR image processing algorithm, this paper developed a SAR image processing experiment platform based on C#, which will be used to visualize the SAR image processing, select different algorithms to demonstrate, and help students to better understand the algorithm. The system interface is beautiful, easy to operate, practical and strong. The system can be used as an effective platform for SAR image processing research, but also as an auxiliary tool for SAR image processing classroom teaching.

Acknowledgements

This work is supported by National Natural Science Foundation of China (Grant No. 61102163), the fourth batch of Information Demonstration Course Construction Project of Shaanxi Normal University, “Teaching Model Innovation and Practice Research” Special Foundation of Shaanxi Normal University (Grant No. JSJX2016Y018) and All English Teaching Demonstration Course Construction Project of Shaanxi Normal University in 2017.

References

[1] Zhang Yan Ning, Li Ying . Key Technology of SAR Image Processing [M]. Beijing: Electronic Industry Press, 2014.

[2] Gonzalez, Ruan Qiuqi. Digital Image Processing [M]. Second Edition. Beijing: Electronic Industry Press, 2007.

[3] Ma Xiaobo, Practical Guide to C# Program Development, Beijing: Tsinghua University Press, 2013.

[4] Jiao Licheng, Intelligent SAR Image Processing and Interpretation [M]. Beijing: Science Press, 2008.

[5] Sun Zengguo. “Gamma-distributed maximum a posteriori despeckling algorithm of high-resolution synthetic aperture radar images [J]”. Acta Physica Sinica, 2013, 62 (18): 180701.

[6] Tison C, Nicolas JM, Tupin F. “A new statistical model for Markovian classification of urban areas in high-resolution SAR images[J]”. IEEE Transactions on Geoscience and Remote Sensing, 2004, 42(10): 2046-2057.

[7] Lee J S. “Refined filtering of image noise using local statistics[J]”. Computer Graphics and Image Processing. 1981, 15(3): 380-389.

[8] Kuan D.T, Sawchuk A.A and Strand T.C. “Adaptive noise smoothing filter for image with signal-dependent noise”. IEEE Trans. Pattern Anal. Mach. Intell 1985, 7(2): 165-177.

[9] Qiu Zhongfan, Wang Shuai, Sun Hexiong, Visual C# programming: 2012 edition, Beijing: Tsinghua University Press, 2013