Fast Processing of UAV Remote Sensing Image Based on DSP Hardware Simulation Platform

Fan Liu¹,³, Guixia Guan¹, Haimeng Zhao²,³*, Peng Sun²,³, Lei Yan³

¹Capital Normal University, Information Engineering College, Beijing 100048, China
²Guilin University of Aerospace Technology, Key Laboratory of Remote Sensing Surveying and Mapping for UAV in Universities of Guangxi, Guangxi GuiLin 541004, China
³Peking University, Space Information Integration and 3S Engineering Application Key Laboratory of Beijing, BeiJing 100871, China

Abstract. The traditional unmanned aerial vehicle (UAV) remote sensing image processing algorithm is difficult to guarantee both accuracy and speed when dealing with massive data of UAV network. This paper designs an on-the-spot processing system based on portable digital signal processor (DSP) hardware simulation platform. The bilinear interpolation algorithm is optimized and improved for the adaptability of the DSP platform with the reference of the bicubic interpolation algorithm, which is long time-consuming but high accuracy. The nearest neighbor interpolation is merged in the bilinear interpolation algorithm, which improves the edge blurring problem of bilinear interpolation algorithm. In the loop calculation of the multiplication and addition iteration calculation segment of the algorithm, the parallelism of the instruction level is increased by increasing the instruction level parallelism between the loop bodies. Experiments show that, on the basis of ensuring the accuracy of the bilinear interpolation algorithm, the execution speed of the processing algorithm is improved, which can meet the requirements of fast real-time processing of remote sensing images of complex unmanned aerial vehicles.

1. Introduction
With the characteristics of automation and intelligence, UAV low altitude remote sensing technology can quickly acquire space remote sensing information such as land resources and military reconnaissance[1]. In the observation for emergency or special needs[2], how to quickly interpolate images acquired from UAV for subsequent requirements is also one of the research priorities. Nowadays, the most common image interpolation algorithms include nearest neighbor interpolation, bilinear interpolation, bicubic interpolation, etc [3], thereinto bilinear interpolation is more practical while images interpolated using this method are prone to losing high-frequency information.

For now, the existed image data processing flow and algorithms are mostly based on traditional computers and server platforms which are large in size, poor in portability, and require long calculation time. Therefore, for the UAV aircraft network under emergency situations, it is hard to realize the processing of acquiring images in a short time. DSP (Digital Signal Processor) is a high-performance microprocessor with powerful information processing capabilities and high computing speeds[4], and actually, image interpolation processing is mainly about the operation of digital signals[5], so DSPs are widely used in real-time situations.
In this paper, the bilinear interpolation algorithm is optimized to solve the edge ambiguity problem, and the algorithm is implemented based on DSP real-time simulation platform. The optimized bilinear interpolation algorithm is compared with the existing bicubic interpolation algorithm in processing effect and processing time. After real-time verification by the simulation platform, and considering processing efficiency and processing accuracy, the optimized bilinear interpolation algorithm is suitable for the on-site fast drone image processing system.

2. Bilinear interpolation algorithm optimization

Bilinear interpolation is easy to implement, while may cause the lost of high frequency information of the image. In comparison, nearest neighbor interpolation, a nonlinear interpolation algorithm with high-pass filtering characteristic, can protect the edges [6]. In this paper, the bilinear interpolation algorithm is merged with the nearest neighbor interpolation. When calculating the gray value of the interpolation point, these two interpolation methods are weighted, which alleviates the bilinear interpolation edge blur problem.

For the pixel to be interpolated, the variance of the gray values of the four adjacent pixels is calculated first. The smaller the gray scale variance, the smoother the neighborhood of the image. Conversely, the larger the gray scale variance, the more likely the neighborhood is the edge region of the image. For smooth regions, the weight of bilinear interpolation increases, and for edge regions, the weight of the nearest neighbor is more significant. The formula for calculating the gray variance of the four neighborhood pixels around the interpolation point is shown in formula (1).

\[
\sigma = \sqrt{\frac{1}{4} \sum_{i=1}^{4} [F(x,y) - \bar{F}(x,y)]^2}
\]  

(1)

Where \(\bar{F}(x,y)\) is the average of the gray values of the four neighborhood pixels. The weight calculation here is based on the method in reference [6], as shown in formula (2).

\[
\begin{align*}
    w_1 &= \frac{1 + e^{-\lambda \sigma}}{2} \\
    w_2 &= 1 - w_1
\end{align*}
\]  

(2)

Next, the bilinear interpolation and the nearest neighbor interpolation are weighted and summed according to the method of formula (2), and the gray value calculation formula of the point to be interpolated is:

\[
F(x, y) = w_1 * F_i(x, y) + w_2 * F_n(x, y)
\]  

(3)

Where \(F_i\) is the result of the bilinear interpolation calculation and \(F_n\) is the result of the nearest neighbor interpolation.

3. Design of Image Interpolation Algorithm on DSP Simulation Platform

3.1 Hardware Simulation Environment Construction

TMS320C6678 DSP chip is used as the main processor of DSP hardware simulation platform, as shown in Figure 1. The chip is a multi-core digital signal processor based on KeyStone architecture, which is suitable for high-performance, low-power on-the-spot data calculation [7]. The test platform is connected to the XDS200mini module via a hardware B2B connector and the XDS200mini is connected to the computer using a USB cable.
3.2 CCS platform algorithm optimization design

C language is used to design algorithms in the DSP integrated development environment- Code Composer Studio 5.5. There are two working modes of CCS: Simulator and Emulator. In this paper, the Simulator mode is used to compare the optimized bilinear interpolation algorithm with the bicubic interpolation algorithm.

Based on the DSP simulation platform, the bilinear interpolation algorithm is further optimized. The block diagram of the coefficient generation unit of bilinear interpolation is shown in Figure 2, and $dx$ and $dy$ respectively represent the horizontal and vertical distances between the target point and the nearest neighbor among the four adjacent points. $W_0$, $W_1$, $W_2$, and $W_3$ are the corresponding weighted values of the four pixel points A, B, C, and D in the original image. Since both $dx$ and $dy$ are decimals, a large number of floating-point operations would be generated in subsequent calculation, which could affect processing speed. Therefore in this paper, $dx$ and $dy$ are enlarged into integers. Considering that if the magnification is too small, the calculation result may have a large error, and if the magnification is too large, it will exceed the range that the long integer can express. After experiments and comparison, the $dx$ and $dy$ are expanded by 2048 times. That is, $dx$ and $dy$ are shifted to the left by 11 bits, and the final result should be shifted to the right by 22 bits. Compared to the multiplier, the shift register occupies less logic resources, and the efficiency of multiplication and division by shifting is much more higher than by direct multiplication and division.

![Figure 2. Block diagram of the bilinear interpolation coefficient generation unit](image)

3.3 Optimization of the multiplication and iteration calculation segment

In this article, the system parallelism is considered to be increased by adding code loop body instructions and reducing the number of loops. The C66x DSP core structure is shown in Figure 3. According to the number configuration of 8 sets of hardware multipliers in C66x kernel, some calculation areas are tested in this paper. The test results show that the calculation speed is optimal when the number of groups to be calculated in the loop is the same as the number of hardware multipliers. The specific implementation method is to increase the number of loop bodies in the "for" loop to 8 loop bodies. The array subscript of the new loop body is incremented by 1 each time on the original basis, and the loop variable is changed from 1 increment each time to 8 increments each time.

![Figure 3. C66x DSP core structure](image)
3.4 CCS platform experimental simulation verification

Three sets of different resolution images taken with a UAV equipped with a SONY ILCE-QX1 v1.00 camera were used as standard test images. As shown in Figure 5, (a) and (b) are taken in a certain urban area of Tianjin, and the image resolutions are 1020*750 and 1160*853 respectively. The shooting location of Figure 4(c) is a certain urban area in Wuhan with an image resolution of 1228*818. The two interpolation methods were evaluated on the CCS platform from the perspectives of visualization and quantitative analysis (mean square error calculation).

In the simulation experiment, each test image is processed by two interpolation algorithms. Firstly, the test image is reduced by 2 times to obtain a low-resolution image, and then two interpolation methods are used in CCS to enlarge and restore the original image size. The experimental process is shown in Figure 5, and the debugging interface based on the CCS platform is shown in Figure 6.
Figure 6. CCS platform simulation interface

Taking the tennis court image as an example, the image interpolation processing results are analyzed. For comparison, the image is processed identically using an unoptimized bilinear interpolation algorithm. The processed image details are shown in Figure 7.

(a) Original grayscale image  (b) Bilinear interpolation algorithm

(c) Optimized bilinear interpolation algorithm  (d) Bicubic interpolation algorithm

Figure 7. DSP CCS simulation results

From the image display, among all images, the image (b) processed by the original bilinear interpolation algorithm is least ideal, and there is a serious edge blur problem. The image (d) processed by the bicubic interpolation is best, and the edges of the image are relatively clear. Although the image (c) processed using the method of this paper is not as ideal as image (d), it is significantly better than (b), and the edge blurring problem is somewhat relieved and closer to (d).

Next, the mean square error between the original image and the final image is calculated to estimate the degree of difference between the original image and the interpolated image. The mean square error can be calculated according to formula (4):

\[
MSE = \frac{1}{wh} \sum_{u,v} (I_s(u,v) - I_r(u,v))^2
\]

(4)

Where \( w \) and \( h \) are the length and width of the image, \( u \) and \( v \) are the coordinates of the pixel point, \( I_s(u,v) \) is the gray value of the pixel of the original image, and \( I_r(u,v) \) is the gray value of the pixel of the image after interpolation. Since serious errors are more likely to arise in the image boundary, only the gray value of the central region pixel of the image 9/10 is selected for calculation. Table 1 shows the MSE values between the processed image and the original image, as well as the processing time on the CCS platform. In this experiment, the clock length \( CLK \) is used to measure the length of time. The clock frequency \( T \) is 1.2 GHz, and the program running time can be calculated according to the formula (5).
Table 1 Comparison of image processing effects of different interpolation methods

| Sample image | Comparison category | Bilinear interpolation | Optimized bilinear interpolation | Bicubic interpolation |
|--------------|---------------------|------------------------|-----------------------------------|-----------------------|
| parking lot  | MSE                 | 5865.35592             | 5718.28255                        | 5468.58671            |
|              | Time                | 0.196s                 | 0.250s                            | 0.389s                |
| tennis court | MSE                 | 668.20966              | 634.06806                         | 537.99923             |
|              | Time                | 0.207s                 | 0.264s                            | 0.433s                |
| house        | MSE                 | 4151.217423            | 4050.95557                        | 4014.888927           |
|              | Time                | 0.221s                 | 0.298s                            | 0.548s                |

Taking the tennis court image as an example, part of the pixel gray value of the processed image is as shown in Figure 8.

In Figure 8, column A is the image pixel gray value of the original image, and columns B, C and D are the gray values of the image pixel points processed by the three interpolation algorithms, respectively. The three columns F, G, and H are the squared error values calculated by column A and columns B, C, and D, respectively. By summing the three columns B, C, and D, respectively, the squared error sum between the pixel points of the original image and the pixel points of the image processed by the three methods can be obtained. The calculated MSE values according to the squared error are 668.20966, 634.06806, and 537.99923, respectively.

It can be seen from Figure 8 that the MSE value of the image processed by the optimized bilinear interpolation algorithm is lower than that of the conventional bilinear interpolation algorithm, and is closer to the result of the bicubic interpolation. It can also be seen from Table 1, in terms of processing speed, the method in this paper is still superior to bicubic interpolation. Therefore, considering the time...
and image quality requirements, the bilinear interpolation optimization algorithm implemented in this paper is a method suitable for rapid processing of UAV images.

4. Summary
Aiming at the requirement of fast processing of remote sensing image of UAV, the bilinear interpolation algorithm is optimized and improved on DSP hardware simulation platform. The edge fuzzy problem existed in the traditional bilinear interpolation algorithm is improved by weighted fusion. The shift operation is used instead of multiplication and division operation, and the decimal is converted into integer to further improve the running speed. The results show that the improved bilinear interpolation algorithm can significantly improve the image interpolation quality, and the processing effect is equivalent to the bicubic interpolation, while the processing speed is obviously better than the bicubic interpolation algorithm. Therefore, the design of this paper can meet the requirements of fast real-time processing of remote sensing images of complex UAVs.

Acknowledgement
This research was financially supported by the National Key R&D Program (No.2017YFB0503000), National Natural Science Foundation of China (No.61841101), Guangxi Key R&D Program (Guike AB18050014).

References
[1] Xiao Chen, Yan Zhang. Simulation Analysis of Mathematical Model for UAV Channel Global Optimization Scheduling[J]. Microelectronics & Computer, 2017(4).
[2] Yu Wu, Tao Yu, Donghai Xie. Fast and automatic stitching of UAV images for emergency response[J]. Journal of Computer-Aided Design & Computer Graphics, 2013, 25(3): 410-416.
[3] Yuming Xu, Shuangchun Wen. Analysis and Implementation of Digital Image Interpolation Algorithm[J]. Fujian Computer, 2007(1).
[4] Sen Hou, Mantang Tan, Xingguo Luo. Research on application-driven multi-DSP processor array in high performance computing[J]. Journal of Computer Applications, 2011, 28(4): 1336-1338.
[5] Xiaoying Deng, Weifeng Chen. Image Interpolation Based on Improved Empirical Mode Decomposition Algorithm[J]. Modern Electronic Technology, 2011(14):18-20.
[6] Changlai Gong, Dongtao Yang. An improved bilinear interpolation image amplification algorithm[J]. LASER & Infrared, 2009(8).
[7] Zhenhuan Zhan. Design of image processing platform based on dual multi-core DSP [D]. Graduate School of Chinese Academy of Sciences (Xi'an Institute of Optics and Fine Mechanics), 2013.