

Spatio-Temporal Fusion Of UAV Remote Sensing Images Based on Pyramid Method

Chao Jiang^{1,*} and Yanfeng Yu²

¹Department of Artificial Intelligence, Zhengzhou Railway Vocational and Technical College, Zhengzhou 451460, China

²Department of Electronic Engineering, Zhengzhou Railway Vocational and Technical College, Zhengzhou 451460, China

The current methods used for the fusion of UAV remote sensing images ignore image interpolation, resulting in poor image fusion effect and quality, resulting in large image fusion registration errors. Hence, in this paper, a pyramid-based method is proposed for the spatio-temporal fusion of UAV remote sensing images. A median filter is used to denoise the image. The projection transformation model between the two images is described in the form of a matrix and homogeneous coordinate system. According to the respective gray levels between neighboring pixels, the corresponding integer coordinate gray levels in the neighborhood are calculated to perform image interpolation correction, image features are extracted and described, and matching strategies are used for feature matching. The corresponding relationship between features is established, the transformation model is estimated according to the parameters of feature-matching results, and the final transformation matrix is determined to complete the image registration. Through the establishment of a Gaussian pyramid, the Laplacian pyramid is constructed, and the image is reconstructed according to certain rules. By comparing the gray values of the corresponding Laplacian pyramid pixels, the pixels with large gray value are obtained to generate the Laplacian pyramid after fusion, so as to realize the fusion of UAV remote sensing images. The experimental results show that the image fusion effect and the quality of the results obtained by the proposed method are better, and it can effectively reduce the image fusion registration error.

Keywords: Gaussian Pyramid, Laplacian Pyramid, UAV Remote Sensing Image, Bilinear Interpolation, Fourier Transform.

1. INTRODUCTION

At present, aerial remote sensing technology has become a necessary scientific and technological means of obtaining information about the geographical environment and its changes. As the demand for remote sensing data in geographic surveying and mapping and related fields is increasingly rapidly, the utilization of UAVs as a remote sensing platform for aerial photography and earth observation has become a new solution (Logie and Coburn, 2018). UAV remote sensing has the characteristics of being fast, flexible, low cost and with high image resolution, enabling UAVs to fly at low altitudes under cloud, thereby addressing the defects of satellite optical remote sensing and ordinary aerial photography which cannot

obtain images under a cloud cover. The platform construction, maintenance, flight height and operation cost are relatively low, and large-scale, high-precision images can be obtained (Abugalia, 2019; Wang and Atkinson, 2018). Compared with other remote sensing platforms, the UAV's low altitude remote sensing system has obvious advantages in terms of mobility, flexibility and providing real-time data. Due to the wide application potential of UAV remote sensing technology, the corresponding technology for image processing is highly valued and of great interest. According to the characteristics of UAV remote sensing images, such as small image size, large number and multiple flight belts, image fusion becomes the key technology to solve the problem of obtaining a panoramic image of the region of interest. Image fusion includes image registration and image fusion. The goal of image registration is to discover the transformation relationship between two

*Corresponding Author Email: mailjc53240341@163.com.

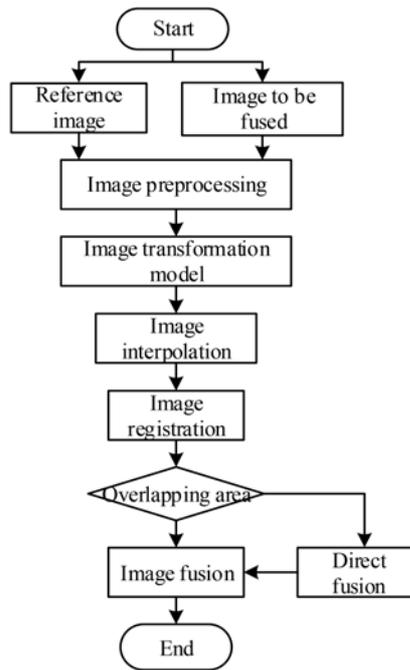


Figure 1 UAV remote sensing image fusion process.

or more images with overlapping areas, establish an image coordinate transformation model, and change multiple images into a unified coordinate system (Huang et al., 2019). Image fusion technology is used to eliminate the problems of brightness, color difference and fusion line between adjacent images after registration, and merge multiple images into a seamless, fused image.

In the fusion of remote sensing images, seams often appear in the fused region, which not only affects the visual effect of the fused image, but is not conducive to the subsequent processing and analysis of the image. Therefore, how to make the fused UAV remote sensing image information rich and transition naturally is a hot issue attracting research from scholars in this field. Han et al. (2020) use the optimized hyperspectral and multispectral images observed in the spectral domain, the spectral dictionary representing the generalized spectrum and its spectral sparseness coefficient. They construct the remaining high-frequency information model in the spatial domain, optimize the spatial dictionary of spatial information and its spatial sparse coefficient, and use the alternate direction method of multipliers to complete image fusion. This method has good fusion performance. He et al. (2019) use the proposed synchronization strategy, combined with non-subsampled transformation and sparse representation, to construct a novel dictionary (He et al., 2019; Gupta and Gupta, 2019). Combining the sparse properties of the learning dictionary with the multi-scale features of the non-subsampled contourlet transform, the same dictionary atoms are used to represent the sparse coefficients to achieve image fusion. This method has better image fusion anti-noise performance. However, the algorithm does not consider image interpolation processing, resulting in poor image fusion effect, and a large image fusion registration error.

In order to solve the above problems, a pyramid-based spatiotemporal fusion of UAV remote sensing images is proposed. Two images with overlapping regions are selected,

and median filtering is used for image denoising (Guo et al., 2016). According to the similarity between descriptors, all features of the two images are matched. This is done by building a Gaussian pyramid and a Laplacian pyramid, reconstructing the image according to certain rules, obtaining pixels with large gray values to generate a fused Laplacian pyramid, and finally obtaining a fused image. The use of a Laplacian pyramid for image fusion produces a good-quality fused image which can effectively reduce the image fusion registration error (Xi et al., 2021).

2. UAV REMOTE SENSING IMAGE FUSION TECHNOLOGY

UAV remote sensing image fusion is the process of transforming and synthesizing a wide-view image of multiple images with overlapping parts. The process of UAV remote sensing image fusion is depicted in Figure 1.

UAV remote sensing image fusion requires five steps. First, two images with overlapping areas are selected as the reference image and the image to be fused; secondly, due to the differences in the equipment, environment and operation mode of the two images, the image quality is different. Therefore, image preprocessing is needed to ensure the quality of image registration and fusion; Then, according to the similarity between descriptors, all the features of the two images are paired to find their corresponding points in the reference image, so that the common features of the two images can be accurately corresponding, and the non-overlapping regions are directly fused together; finally, due to different shooting angles or different lighting conditions, the difference in brightness of the two images after fusion is obvious, resulting in a fusion seam and ghosting. Further image fusion processing is needed to ensure the smoothness of coincidence areas, and finally the fused image is obtained.

2.1 Image Preprocessing

When UAVs acquire remote sensing images, there are obvious color differences in remote sensing images due to changes in illumination and shooting angle, which makes the brightness and saturation of images different, directly affecting the fusion quality of the image (Sharma et al., 2020; Paramitha et al., 2018). Therefore, it is necessary to preprocess the image before image fusion. In order to ensure the integrity of information and the accuracy of image fusion, a median filter is applied to achieve image denoising. Set the window Q with a certain size to filter the image, and the pixel points in the window are arranged according to the size of the gray value, and the median of the sequence is used to replace the gray value of the pixel at the center of the image. The calculation formula is expressed as:

$$U(x, y) = \text{median}\{f(o, p), (o, p) \in W\} \quad (1)$$

In Equation (1), U represents the filtered gray value of the center point, (o, p) represents the pixel coordinates in Q , and f represents the unprocessed sequence. The median filtering result should satisfy:

$$\text{median}\{e(a) + s(a)\} \neq \text{median}\{e(a)\} + \text{median}\{s(a)\} \quad (2)$$

In Equation (2), $e(a)$, $s(a)$ represents two images with area a . If the noise pixel is greater than half of the sum of the image pixels, the median filter is not suitable for this type of image denoising because it will easily cause image distortion. The window size of the median filter can be adjusted according to the complexity of the image information so as to control the speed and accuracy of the filter.

2.2 Image Transformation Model

In order to achieve the registration between images, it is necessary to determine the spatial correspondence between them; that is, to find the corresponding image transformation model. The projective transformation model between two images can be described by using the form of a matrix and homogeneous coordinate system, Assuming that (x, y) and (x_1, y_1) are the coordinates of a corresponding feature point of the image, the transformation formula is:

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} m_0 & m_1 & m_2 \\ m_3 & m_4 & m_5 \\ m_6 & m_7 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (3)$$

In Equation (3), m_0, m_1, m_3, m_4 represents scale and rotation, m_2 represents horizontal displacement, m_5 represents vertical displacement, and m_6, m_7 represents horizontal and vertical deformation (Tirupal et al., 2018). Different transformation models can be obtained according to the different values of model parameters. The commonly-used image transformation models are: rigid transformation, similar transformation, affine transformation, perspective transformation and nonlinear transformation.

- (1) Rigid body transformation: When the image is transformed, there is neither shape change nor size change.

Translation and rotation transformation is called 'rigid body transformation'. The relative distance between the points in the image will not change. The rigid body transformation formula is expressed as:

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} m_2 \\ m_5 \end{bmatrix} \quad (4)$$

In Equation (4), θ represents the rotation angle of the image. Rigid body transformation has 3 degrees of freedom, and at least 2 feature point pairs are required to find all variables.

- (2) Similarity transformation: When the image is similarly transformed, the shape does not change, although the size changes. The image has not only translation and rotation, but also scaling. The similar transformation formula is expressed as:

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = q \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} m_2 \\ m_5 \end{bmatrix} \quad (5)$$

In Equation (5), q represents the scaling factor. Similarity transformation has 4 degrees of freedom, and at least 2 feature point pairs are required to find all variables.

- (3) Affine transformation: Affine transformation not only produces shape changes, but also changes in size, although the parallel relationship does not change. It is a combination of linear transformation and translation transformation, and is widely used (Bi et al., 2020). The affine transformation formula is expressed as:

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} m_0 & m_1 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} m_2 \\ m_5 \end{bmatrix} \quad (6)$$

The affine transformation has 6 degrees of freedom, and at least 3 feature point pairs are required. Any 2 points pairs cannot be collinear to obtain all parameters.

- (4) Perspective transformation: Perspective transformation is a general transformation model. Shape, size, translation, rotation, and scaling changes can occur between images used for transformation, and this methods is widely used. The projection transformation formula is expressed as:

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} m_0 & m_1 \\ m_3 & m_4 \\ m_6 & m_7 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} m_2 \\ m_5 \\ 1 \end{bmatrix} \quad (7)$$

Perspective transformation has 8 degrees of freedom, so at least 4 feature point pairs are required. Any 3 points pairs cannot be collinear in order to solve all variables.

- (5) Nonlinear transformation: Nonlinear transformation is in the transformation model, the relationship between parameters is not linear, but flat or surface, sometimes the mathematical relationship may not be determined. Suppose the transformation coincidence mapping function is S , then the nonlinear transformation formula is expressed as:

$$(x_1, y_1) = S(x, y) \quad (8)$$

Nonlinear transformation is suitable not only for image transformation with global deformation, but also for rigid body transformation with local deformation.

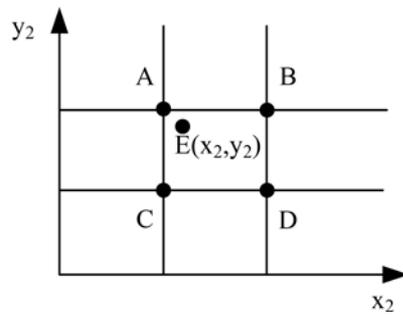


Figure 2 Nearest neighbor interpolation.

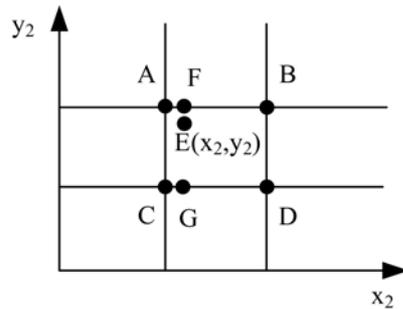


Figure 3 Bilinear interpolation.

2.3 Image Interpolation

After the parameters of the transformation model have been calculated, the two images can be transformed into the same coordinate system, although the transformed coordinates are generally not integers. Such pixel values are not defined and cannot fall on the grid points of the image, so image interpolation is needed for correction. The essence of image interpolation correction is to re-assign the gray level of the image, and the principle is to calculate the corresponding integer coordinate gray level of the adjacent pixels according to their respective gray levels (Chen et al., 2020). The commonly used image interpolation methods include nearest neighbor interpolation, bilinear interpolation, cubic convolution interpolation and so on.

- (1) Nearest neighbor interpolation: It uses the integer coordinates closest to the gray value after transformation to represent the pixel value of the original corresponding point. Assuming that there are four adjacent pixels near the pixel value after transformation, the nearest pixel gray value is selected as the interpolation value. The nearest neighbor interpolation is shown in Figure 2.

In Figure 2, point A is the closest point to point E, then the pixel value of point E is replaced by the pixel value of point A, which can be expressed as:

$$\begin{cases} x_2 = \lfloor \underline{y_2} + 0.5 \rfloor \\ y_2 = \lfloor \underline{x_2} + 0.5 \rfloor \end{cases} \quad (9)$$

In Equation (9), $\lfloor \cdot \rfloor$ represents the round-down symbol, the nearest neighbor interpolation operation is simple, and the amount of calculation is small. However, the approximate calculation produces a significant error, and

the image fusion is prone to the problems of jagged effect, image blur and discontinuity.

- (2) Bilinear interpolation: As an improved version of nearest neighbor interpolation, it uses four adjacent pixels to perform linear first-order interpolation in horizontal and vertical directions respectively. The computational complexity is moderate, the problem that the nearest neighbor interpolation error is too large is well handled, and the image quality and smoothness are improved. Bilinear interpolation is shown in Figure 3.

In Figure 3, let $A(i, j+1)$, $B(i+1, j+1)$, $C(i, j)$, $D(i+1, j)$ be the 4 adjacent pixels of the transformed non-integer point $E(x_2, y_2)$, and use R to represent the gray value.

Bilinear interpolation starts from the x_2 direction and calculates the gray values of F and G respectively. The formulas are:

$$\begin{aligned} R(F) &= (x_2 - i)[R(B) - R(A)] + R(A) \\ R(G) &= (x_2 - i)[R(D) - R(C)] + R(C) \end{aligned} \quad (10)$$

Interpolate from the y_2 direction, and finally get the gray level interpolation of point E expressed as:

$$R(E) = (y_2 - i)R(F) - R(G) \quad (11)$$

- (3) Cubic convolution interpolation: Bilinear interpolation uses four adjacent pixels for linear first-order interpolation, ignoring the influence of gray value changes between each pixel (Luo et al., 2019). Therefore, cubic convolution interpolation calculates the interpolation gray level according to 16 pixels near the interpolation point. The cubic convolution interpolation is shown in Figure 4.

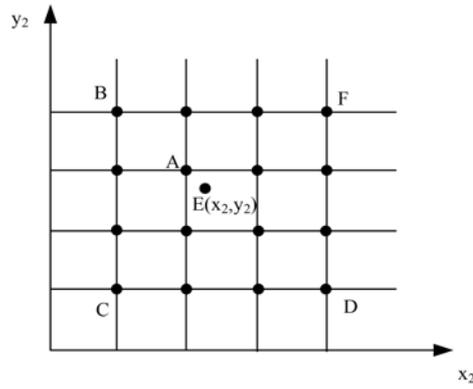


Figure 4 Cubic convolution interpolation.

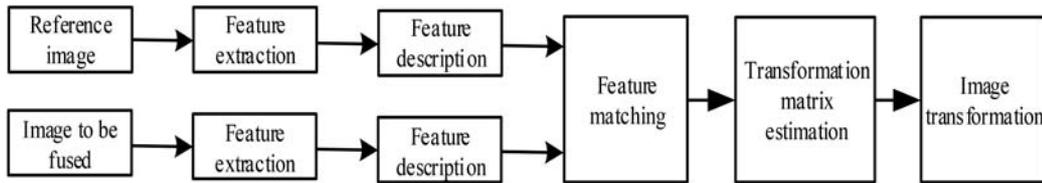


Figure 5 Image registration flow chart.

According to the continuous signal sampling theorem, if the sampling value is interpolated with the interpolation function $\text{sinc}(c)$, the function can be restored accurately. Then the cubic convolution interpolation method adopts the third approximation polynomial of $\text{sinc}(c)$ and can be expressed as:

$$Z(x_2) = \begin{cases} 1 - 2|x_2|^2 + |x_2|^3 & 0 \leq |x_2| < 1 \\ 4 - 8|x_2| + 5|x_2|^2 - |x_2|^3 & 1 \leq |x_2| < 2 \\ 0 & |x_2| \geq 2 \end{cases} \quad (12)$$

Suppose the coordinate of point A is (i, j) , then $B(i - 1, j + 1)$, $C(i - 1, j - 2)$, $D(i + 2, j - 2)$, $E(i + a, j + b)$, $F(i + 2, j + 1)$, then the gray value of point E can be expressed as:

$$R(E) = R(i + a, j + b) = M * K * J \quad (13)$$

The cubic convolution interpolation is the best of the three methods, but requires the greatest amount of calculation. The UAV remote sensing image fusion requires high real-time performance. Therefore, the cubic convolution interpolation is not applicable.

2.4 Image Registration

Image registration can be defined as the spatial correspondence between two images (Ding et al., 2018). For two given images $\alpha(x_\alpha, y_\alpha)$ and $\beta(x_\beta, y_\beta)$, assuming that image $\alpha(x_\alpha, y_\alpha)$ is the reference image and the other image $\beta(x_\beta, y_\beta)$ is the image to be registered, the image transformation formula can be expressed as:

$$\beta(x_\beta, y_\beta) = \delta\{\alpha[\gamma(x_\alpha, y_\alpha)]\} \quad (14)$$

In Equation (14), γ represents two-dimensional space coordinate transformation, and δ represents one-dimensional grayscale transformation. Image registration includes five steps: feature extraction, feature description, feature matching, transformation matrix estimation and image transformation. The image registration process is shown in Figure 5.

First, feature extraction is performed on the image to make the feature distinguishable, and then the feature is described. Its accuracy determines the accuracy of the matching. Then the matching strategy is used for feature matching, and the corresponding relationship between features is established. According to the result of feature matching, the parameters of transformation model are estimated, and the final transformation matrix is determined. Finally, coordinate transformation is completed by calculating the parameters of transformation model to realize image registration. The popular image registration methods are:

- (1) Registration method based on transform domain: Fourier transform is used to change the image from the space domain to the frequency domain. According to the translation property of Fourier transform, the translation amount between images is calculated. Assuming that two images α and β have a displacement (d_α, d_β) in the horizontal direction, it is satisfied:

$$\beta(x_\beta, y_\beta) = \alpha(x_\alpha - d_\alpha, y_\alpha - d_\beta) \quad (15)$$

After Fourier transform, we obtain:

$$F_2(\omega_\alpha, \omega_\beta) = e^{-j(\omega_\alpha d_\alpha + \omega_\beta d_\beta)} F_1(\omega_\alpha, \omega_\beta) \quad (16)$$

Then Fourier transform can be obtained with:

$$F_2 * (\omega_\alpha, \omega_\beta) = e^{-j(\omega_\alpha d_\alpha + \omega_\beta d_\beta)} F_1 * (\omega_\alpha, \omega_\beta) \quad (17)$$

It is found that the image with translation has the same amplitude after Fourier transform. However, the phase

has changed due to the translation relationship, and the phase difference is equal to the phase of the cross-power spectrum between images, which can be expressed as:

$$e^{-j(\omega_\alpha d_\alpha + \omega_\beta d_\beta)} = \frac{F_1(\omega_\alpha, \omega_\beta) F_2 * (\omega_\alpha, \omega_\beta)}{F_1(\omega_\alpha, \omega_\beta) F_2 * (\omega_\alpha, \omega_\beta)} \quad (18)$$

After the image is Fourier transformed, its translation and rotation can be reflected in the Fourier transform (Kinoshita and Kiya, 2019). The algorithm itself is simple to implement and has an anti-noise property, making it popular.

- (2) Gray-based registration method: The gray value is used to determine the transformation model, so as to get the corresponding pixel after transformation. Based on the gray value of the overlapped region, the corresponding cost function is established as the similarity measure. When the cost function gets the extreme value, it corresponds to the optimal parameter solution of the transformation model. Generally, SSD (sum of squared difference) is regarded as the cost function, which can be expressed as follows:

$$E = \sum [H'(x'_i, y'_i) - H(x_i, y_i)]^2 = \sum \sigma^2 \quad (19)$$

In Equation (19), $H'(x'_i, y'_i)$, $H(x_i, y_i)$ represents the grayscale of the pixels contained in the corresponding areas of the two images, and σ represents the pixel grayscale difference. Gray based registration is simple and accurate for searching all gray information. However, it requires a large amount of computation, relies too much on the gray level of the image, and is invariant only to translation changes. When scaling or rotation occurs, the registration effect is very poor.

- (3) Registration method based on image features: It has good robustness against changes of illumination. The image feature is a series of discontinuous points on the gray surface of the image, so the amount of calculation is greatly reduced, and the amount of information is large in the local area, so it has strong adaptability to image deformation and gray transformation. Usually, feature-based registration is manifested in points, edges and regions.
- (a) Method based on point feature: This is currently the most commonly used. Firstly, the image features are extracted, and then the obtained features are used to match the two images. This method has high registration efficiency and strong robustness.
- (b) Method based on feature edge: Its advantage lies in its fast speed. First, the edge part is detected because it can represent the general structure of the image and has strong robustness against regional grayscale distortion, and is often used for multi-mode image registration.
- (c) Method based on regional characteristics: Since moment invariants do not change with image rotation, translation, scaling and other transformations, the most commonly-used method of image

registration based on region features is moment invariant, which measures the similarity through Euclidean distance, although the description of the invariance of regional features still needs to be further explored.

3. UAV REMOTE SENSING IMAGE FUSION BASED ON THE PYRAMID METHOD

A pyramid algorithm is derived from the simulation of the human perception process in computer vision. The multi-resolution feature is formed by calculating the image from the bottom to the top. Each layer of image is formed by the former image filtered by template. After the inverse transformation process, the original image can be reconstructed accurately (Li, 2019). In this paper, the simplest Gaussian Laplacian pyramid is used to fuse UAV remote sensing images.

3.1 Construct a Gaussian Pyramid

Suppose the input image is $P_0(m, n)$ ($m \leq M, n \leq N$) where M, N represents the number of rows and columns of the image, and the input image P_0 is the bottom layer of the Gaussian pyramid, that is, the 0th layer; then, the first layer of the Gaussian pyramid can be expressed as:

$$P_1 = \sum_{m=-2}^2 \sum_{n=2}^2 t(m, n) P_0(2i + m, 2j + n) \quad (20)$$

In Equation (20), $t(m, n)$ represents the generating core, which is a window function and, in fact, is also a low-pass filter. Among them, $t(m, n)$ needs to meet the following conditions:

- (1) Separability, namely $t(m, n) = t(m)t(n)$;
- (2) Normalization, $t(m, n) = \sum t(m) = \sum t(n) = 1$;
- (3) Symmetry, $t(n) = -t(n)$;
- (4) Contributions of odd and even items, $t(-2) + t(2) + t(0) = t(-1) + t(1)$.

The above restriction on the generation of cores ensures the low-pass nature and maintains the smooth brightness of the image after shrinking and expansion without seam effects. A typical 5×5 sub-window that satisfies the above constraints is expressed as:

$$t(m, n) = \frac{1}{256} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix} \quad (21)$$

This is similar to the normalized Gaussian distribution, so the generated kernel is approximately a Gaussian low-pass filter. Generally, the $k - 1$ layer subgraph can be expressed as:

$$P_k = \sum_{m=-2}^2 \sum_{n=-2}^2 t(m, n) P_{k-1}(2i + m, 2j + n) \quad (22)$$

$$0 \leq k \leq N, 0 \leq i \leq R_k, 0 \leq j \leq L_k$$

In Equation (22), N represents the layer number of the top layer of the Gaussian pyramid, R_k represents the number of rows of the image of the k layer of the Gaussian pyramid, and L_k represents the number of columns of the image of the k layer of the Gaussian pyramid. It can be seen that the Gaussian pyramid of an image is generated by low-pass filtering of the next layer, and then down-sampling the interlaced and interlaced columns. This series of images, which are four times smaller than the next, are arranged from low to high to form the so-called Gaussian pyramid of images.

3.2 Constructing the Laplace Pyramid

The Laplacian pyramid comprises a set of band-pass filtering sequences, which defines the image scale reduction operator, Reduce, can be expressed as:

$$P_k = \text{Reduce}(P_{k-1}) \quad (23)$$

The generation of the Laplacian pyramid of an image needs to have an expansion process opposite to that of the image reduction, which is the process of expanding the image from the first layer to the same size as that of the $k - 1$ layer (Yang et al., 2018). The expansion operator, Expand, that defines the image can be expressed as:

$$P_{k*} = \text{Expand}(P_k) \quad (24)$$

Defining the expansion operator, Expand, is the inverse operation of the reduce operator Reduce, used to interpolate new sample values between the given values. The expansion of the Gaussian pyramid to the same size as the previous layer can be expressed as:

$$P_{k*} = 4 \sum_{m=-2}^2 \sum_{n=-2}^n t(m, n) P_k \left(\frac{i+m}{2}, \frac{j+n}{2} \right) \quad (25)$$

It can be seen that the Laplacian pyramid is actually the differential value between the Gaussian pyramid and its upper layer by interpolation, and the highest level is the Gaussian pyramid itself. Since the upper image is obtained by down sampling after low-pass filtering, the difference between the enlarged and original level indicates the difference in information between the two levels of the Gaussian pyramid; that is, the Laplacian pyramid is actually the high-frequency component of the same level Gaussian pyramid; that is, the detailed part of the image. The complete Laplace pyramid is defined with:

$$\begin{cases} LQ_k = P_k - \text{Expand}(P_{k+1}) \\ LQ_N = P_N \end{cases} \quad (26)$$

3.3 Reconstruction and Fusion of UAV Remote Sensing Images

The input image is reconstructed from the Laplacian pyramid, and the corresponding levels LQ_k and LQ_N of the Laplacian pyramid are reconstructed and fused according to certain rules, which can be expressed as:

$$\begin{cases} C_{Fk} = P_{Fk} + \text{Expand}(C_{Fk+1}) \\ C_{FN} = P_{FN} \end{cases} \quad (27)$$

The Laplacian pyramid shows the edge details of each level of image. Therefore, by comparing the Laplacian pyramids of the corresponding level of two images, it is possible to include the outstanding image details into the fusion image, so as to make the information content of the fusion image as rich as possible and achieve the purpose of fusion (Vargas et al., 2019). In this paper, by comparing the gray values of the pixels corresponding to the Laplacian pyramid, the pixels with large gray values are obtained to generate the fused Laplacian pyramid, and the l level Laplacian pyramid after fusion is set to LP_{lF} , and the two are to be merged. An image α and β are $LP_{l\alpha}$ and $LP_{l\beta}$ respectively, then the fusion rule is:

$$LP_{lF}(i, j) = \begin{cases} LP_{l\alpha}(i, j) & \text{if } |LP_{l\alpha}(i, j) - LP_{l\beta}(i, j)| > |LP_{l\beta}(i, j) - LP_{l\alpha}(i, j)| \\ LP_{l\beta}(i, j) & \text{other} \end{cases} \quad (28)$$

After obtaining the fused Laplacian pyramid, it is reconstructed using Equation (27) to obtain the fused image. Through the above steps, two images with overlapping areas are selected, and the images are preprocessed. According to the degree of similarity between the descriptors, all the features of the two images are matched in order to accurately correspond to their shared features. After establishing the Gaussian pyramid and Laplacian pyramid, the image is reconstructed according to certain rules, and the pixels with large gray value are obtained to generate the Laplacian pyramid after fusion to realize the fusion of UAV remote sensing image.

4. EXPERIMENTAL SIMULATION AND ANALYSIS

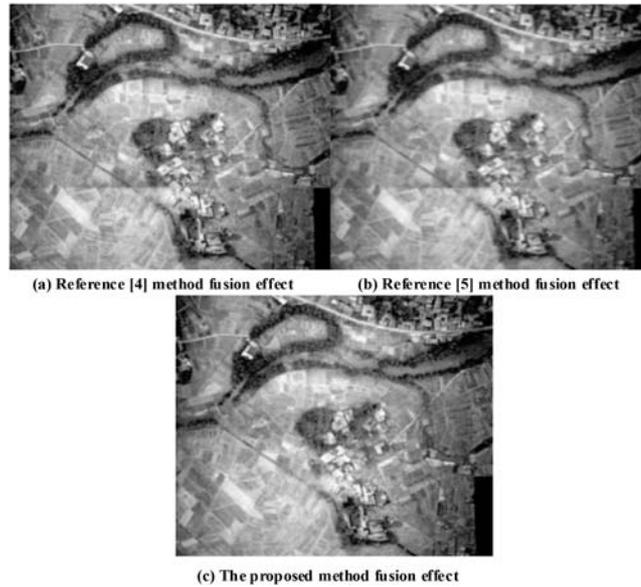
4.1 Set up the Experimental Environment

In order to confirm the effectiveness and performance of the spatio-temporal, pyramid-based fusion method, it is tested on a UAV remote sensing image. The experiment uses a computer with Inter Core i5-3470 processor, 8.00G memory, 600G hard disk, and 64-bit Windows7 operating system. The algorithm is implemented and tested in an experimental Matlab2014a environment. The data used in this experiment was acquired by the sensor mounted on the Quickeye-?? pull-in UAV, the model is Canon EOS 5D Mark ???. The specific parameters are shown in Table 1.

Canon EOS 5D Mark II belongs to a non-measurement camera, which focuses automatically during UAV flight shooting. Therefore, the focal length of each photo is

Table 1 Sensor performance parameters.

Project	Parameter
Sensor name	Canon EOS 5D Mark ??
Number of effective pixels	21 million 100 thousand
Lens focal length	24mm
Image size	5616 × 374
Image format	JPEG
CCD size	1/1.75inch


Figure 6 Different methods of UAV remote sensing image fusion effect.

different. However, camera calibration is outside the scope of this paper. In the experiment, it is assumed that the focal length of the camera is fixed when each image is taken, with basically the same height. The experiment uses the images acquired by the Canon EOS 5D Mark ?? sensor mounted on the drone to test and analyze the algorithm in terms of image fusion effect, anti-statistic performance, key sensitivity, and information entropy.

4.2 Comparison of Image Fusion Effects

Two UAV remote images with overlapping areas are selected, and the Ding et al (2018) method, the Gupta and Gupta (2019) method and the proposed method are used for the fusion process. The two UAV remote sensing images acquired under the same conditions are fused, and the fusion effects of different methods are compared as shown in Figure 6.

It can be seen from Figure 6 that there is a pixel width fusion suture in the image fusion region using the Ding et al. (2018) method, and the phenomenon of over fusion area is more obvious. The gray transition of pixel information in the image fusion area using the Gupta and Gupta (2019) method is more obvious, and there are phenomena such as blur and distortion. Compared with the Ding et al. (2018) method and the Gupta and Gupta (2019) method, the proposed method has no obvious fusion line in the processed fusion area, and achieves the purpose of removing the fusion gap

and smoothing the transition of the image fusion area. The actual fusion effect and practicability are significantly higher than those of the Ding et al. (2018) method and the Gupta and Gupta (2019) method. It can be seen that the image fusion effect of the proposed algorithm is better.

4.3 Image Fusion Registration Error Comparison

For multi-focus images, since the standard image is provided, the image fusion registration error can be directly measured by calculating the error between the fused image and the standard image. The image fusion registration error calculation formula is:

$$W_c = \frac{\sum_i^M \sum_j^N [F(i, j) - S(i, j)]^2}{M \cdot N} \quad (29)$$

where $F(i, j)$ represents the fusion image, and $S(i, j)$ represents the standard image. Using the Ding et al (2018) method, the Gupta and Gupta (2019) method and the proposed method respectively, the image fusion registration errors of different methods are calculated by Equation (29). The results of image fusion registration errors of different methods are shown in Table 2 for comparison.

It can be seen from Table 3 that the image fusion registration error of the Ding et al. (2018) method is 45.3%, the image fusion registration error of the Gupta and Gupta (2019) method

Table 2 Image fusion registration error of different methods.

Different methods	Image fusion registration error /%
The proposed method	38
Ding et al. (2018) method	45.3
Gupta and Gupta (2019) method	53.6

Table 3 Entropy and cross entropy of fused images with different methods.

Different methods	Entropy	Average cross entropy	Root mean square cross entropy
The proposed method	4.13	0.95	1.29
Ding et al. (2018) method	4.32	0.93	1.25
Gupta and Gupta (2019) method	4.45	0.90	1.21

is 53.6%, and the image fusion registration error of the proposed method is only 38%. It can be seen that, compared with the Ding et al. (2018) method and the Gupta and Gupta (2019) method, the image fusion registration error of the proposed method is smaller.

4.4 Comparison of Image Fusion Quality

The image fusion quality is judged by calculating the image information index, mainly calculating the entropy and cross entropy of the image.

Entropy indicates the amount of information contained in an image. The greater the entropy, the more information the image contains. It is generally believed that the greater the entropy in the fused image, the richer the information, and the better the fusion quality. Image entropy is obtained with:

$$\chi = - \sum_{i=1}^{\psi-1} r_i \log r_i \tag{30}$$

Where χ represents the entropy of the image, ψ represents the total number of gray levels of the image, and r_i represents the ratio of the number of pixels of the i gray level to the total number of pixels.

Cross entropy is used to indicate the difference between two images. The smaller the cross entropy, the smaller is the difference between the two images. Therefore, if the average or root mean square cross entropy is smaller, this means that the fused image obtains more information from the original image, so the fusion quality will be better. Assuming that the fused image is v and the original image is ρ, μ , the cross entropy of the two original images and the fused image are, respectively:

$$\begin{aligned} \chi_{\rho,v} &= - \sum_{i=1}^{\psi-1} r_{\rho i} \log \left(\frac{r_{\rho i}}{r_{v i}} \right) \\ \chi_{\mu,v} &= - \sum_{i=1}^{\psi-1} r_{\mu i} \log \left(\frac{r_{\mu i}}{r_{v i}} \right) \end{aligned} \tag{31}$$

In regard to comprehensiveness, the average cross entropy and root mean square cross entropy are:

$$\begin{aligned} MCE &= \frac{\chi_{\rho,v} + \chi_{\mu,v}}{2} \\ RCE &= \sqrt{\frac{\chi_{\rho,v}^2 + \chi_{\mu,v}^2}{2}} \end{aligned} \tag{32}$$

Using the Ding et al. (2018) method, the Gupta and Gupta (2019) method and the proposed method, respectively, the entropy and cross entropy of the fused image of different methods are calculated using Equation (32). The results of the entropy and cross entropy of the fused image of different methods are shown in Table 3 for comparison.

According to the data in Table 2, the entropy, average cross entropy and root mean square cross entropy of the fusion image in the Ding et al. (2018) method are 4.13, 0.95 and 1.29, respectively. Those of the Gupta and Gupta (2019) method are 4.32, 0.93 and 1.25 respectively, while those of the proposed method are 4.45, 0.90 and 1.21, respectively. Therefore, compared with the results obtained by the current methods, with the proposed method, the entropy of the fusion image is larger, the average cross entropy and root mean square cross entropy are smaller, the image contains more information, and the fusion quality is better.

5. CONCLUSIONS

In this paper, the research of UAV remote sensing image spatiotemporal fusion based on the pyramid method is proposed. The advantages of the Laplacian pyramid are brought into full play. The effect and quality of space-time fusion of UAV remote sensing image is good, which can effectively reduce the registration error in image fusion. However, during image fusion, as the number of fused images increases, a certain fusion gap will appear at the upper and lower boundaries of the image. A UAV remote sensing image has a large amount of data and high resolution. The speed of image fusion and the meeting of real-time requirements are two problems requiring research as a matter of urgency. Therefore, the next step is to find an adaptive image fusion method that can success fully address these two issues.

REFERENCES

1. Abugalia A. 2019. Effect of Corona on The Wave Propagation Along Overhead Transmission Lines. *Acta Electronica Malaysia*, 3(1), 06–09.

2. Bi, F., Lei, M., Wang, Y. 2020, Context-aware MDNet for target tracking in UAV remote sensing videos. *International Journal of Remote Sensing*, 41(10), 3784–3797.
3. Chen, S., Xiang, C., Kang, Q., Liu, K., Zhou, R. 2020, Accurate landslide detection leveraging UAV-based aerial remote sensing. *IET Communications*, (1), 1099–1115.
4. Ding, W.S., Bi, D.Y., He, L.Y., Fan, Z.L. 2018, Infrared and visible image fusion method based on sparse features. *Infrared Physics & Technology*, 92, 372–380.
5. Guo, L.Y., Luo, B.T., Zhang, G.Y., Wu, J.H., Tu, B., Ou, X.F., 2016. Edge detection of infrared image base on phase congruency. *Engineering Intelligent Systems*, 24(3–4), 125–130.
6. Gupta J.K., Gupta S.K. 2019. A Comparative Study of Crowd Counting and Profiling Through Visual and Non -Visual Sensors. *Acta Informatica Malaysia*, 3(1), 04–06.
7. Han, X.L., Yu, J., Xue, J.H., Sun, W. 2020, Hyperspectral and multispectral image fusion using optimized twin dictionaries. *IEEE Transactions on Image Processing*, 29, 4709–4720.
8. He, G.Q., Xing, S.Y., He, X.J., Wang, J., Fan, J.P. 2019, Image fusion method based on simultaneous sparse representation with non-subsampled contourlet transform. *IET Computer Vision*, 13(2), 240–248.
9. Huang, H., Song, J., Guo, L., Wang, H.F., Wang, P. 2019, Haze removal method based on a variation function and colour attenuation prior for UAV remote-sensing images. *Journal of Modern Optics*, 66(2), 1–14.
10. Kinoshita, Y., Kiyama, H. 2019, Scene segmentation-based luminance adjustment for multi-exposure image fusion. *IEEE Transactions on Image Processing*, 28(8), 4101–4116.
11. Li, K. 2019, Simulation analysis of feature extraction of degradation trend of landscape remote sensing images. *Computer Simulation*, 36(7), 195–198+313.
12. Logie, G., Coburn, C. 2018, An investigation of the spectral and radiometric characteristics of low-cost digital cameras for use in UAV remote sensing. *International Journal of Remote Sensing*, 39(15–16), 4891–4909.
13. Luo, W., Xu, X.L., Liu, W., Liu, M.X., Li, Z.W., Peng, T., Xu, C.H., Zhang, Y.H., Zhang, R.F. 2019, UAV-based soil moisture remote sensing in a Karst mountainous catchment. *Catena*, 174, 478–489.
14. Paramitha M., Agustia D., Soewarno N. 2018. Conceptual Framework of Good Governance, Organizational Culture, And Performance at Higher Education in Indonesia. *Information Management and Computer Science*, 1(1), 18–23.
15. Sharma, A.M., Dogra, A., Goyal, B., Vig, R., Agrawal, S. 2020, From pyramids to state-of-the-art: A study and comprehensive comparison of visible–infrared image fusion techniques. *IET Image Processing*, 14(9), 1671–1689.
16. Tirupal, T., Chandra Mohan, B., Srinivas Kumar, S. 2018, Multimodal medical image fusion based on yager’s intuitionistic fuzzy sets. *Etri Journal*, 39(2), 173–180.
17. Vargas, E., Espitia, O., Arguello, H., Tournet, G.Y. 2019, Spectral image fusion from compressive measurements. *IEEE Transactions on Image Processing*, 28(5), 2271–2282.
18. Wang, Q., Atkinson, P.M. 2018, Spatio-temporal fusion for daily Sentinel-2 images. *Remote Sensing of Environment*, 204, 31–42.
19. Xi, Y.P., Yang, S.D., Wang, P.L., Feng, Y.H., 2021. Human motion system model based on real-time image acquisition and data simulation. *Engineering Intelligent Systems*, 29(3), 175–181.
20. Yang, X., Wang, J., Zhu, R. 2018, Random walks for synthetic aperture radar image fusion in framelet domain. *IEEE Transactions on Image Processing*, 27(2), 851–865.