A Fast Image Template Matching Algorithm Based on Normalized Cross Correlation

Zhongjie Cui¹, Wenfa Qi²*, Yuxin Liu²

¹Beijing Institute of Information Application Technology, 100044, China
²Wangxuan Institute of Computer Technology, Peking University, Beijing 100080, China
qiwenfa@pku.edu.cn*

Abstract. Image matching algorithms based on image gray value are commonly used, which can achieve high matching accuracy. Among them, the Normalized Cross Correlation (NCC) method has high accuracy and strong adaptability, however it has the disadvantages of high computational complexity and slow calculation speed. By using dynamic programming strategy, a fast template matching algorithm based on NCC is proposed in this paper. At the same time, by calculating the cumulative sum of pixel values and the sum of squares of pixel values, and using the down sampling process, the calculation time and complexity are greatly reduced. Experimental results show that the proposed method has the same matching accuracy as the existing algorithms, and further improves the image matching efficiency.

1. Introduction

Image matching technology is often used in computer recognition, that is, to compare two or more images taken from the same scene by two different sensors or the same sensor at different times and conditions, and find the common image in group of images, or find the corresponding pattern in another image according to the given pattern. The classical image matching algorithms can basically be divided into two categories [1]: the feature-based method [2-5] and gray-based method [6-9].

The method based on image gray value is commonly used in image matching. It directly uses the pixel value of the original image for image matching, and makes full use of all the gray information in image to accurately distinguish different objects, so as to achieve high matching accuracy. However, because each pixel in benchmark image needs to be scanned in turn to get the difference between the real-time image and benchmark image, the amount of information to be processed is large and the computational complexity is high. The commonly used image matching algorithms based on image gray value are Absolute Balance Search algorithm [10], NCC [11-13], Sequential Similarity Detection Algorithm [14] and Sum of Squared Differences [15], et al. Among them, the NCC algorithm has high accuracy and adaptability, and the calculated NCC value is not affected by the linear transformation of gray value. However, the NCC algorithm also has disadvantage of large calculation amount and high computational complexity. Therefore, how to speed up the efficiency of image matching while ensuring the accuracy of image matching is a main subject of further research.

2. NCC Algorithm Principle

Let S be the matching image with the size of M*N pixels, and T be the template image with the size of m*n pixels. As shown in Fig.1, the template image T slides on the matching image S, and the gray
correlation value of images S and T is calculated by using the correlation function when the template image T slides to the position \((u,v)\). When the correlation value reaches the maximum value, the position of the search window determines the position of T in S.

\[
\text{Fig. 1 Schematic diagram of image matching principle}
\]

The NCC algorithm calculates the image matching degree between the two images through a normalized correlation measurement formula [11] as follows:

\[
R(u,v) = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} f(u+ (i+v) - f_{avg}) \cdot t(i,j) - \bar{t}}{\sqrt{\left(\sum_{i=1}^{m} \sum_{j=1}^{n} (f(u+ (i+v)) - f_{avg})\right)^2 \left(\sum_{i=1}^{m} \sum_{j=1}^{n} t(i,j) - \bar{t}\right)^2}}
\]

(1)

where \(f(i,j)\) is the pixel value of the matching image S at \((i,j)\), \(t(i,j)\) is the pixel value of the template image T at \((i,j)\), \(\bar{t}\) is the average pixel value of the template image T, and \(\bar{f}\) is the average pixel value of the corresponding image S under the current template image T.

Due to the high computational complexity of the above (1), a series of accelerating calculation methods can be adopted in practical application [16].

First, let \(t'(i,j) = t(i,j) - \bar{t}\), then the molecule of (1) can be expressed as follows:

\[
\sum I_{i,j} f(u + i, v + j) t'(i,j) - \bar{f} \sum i,j t'(i,j)
\]

(2)

Obviously, the last term of (2) is always equal to zero, and the first term of (2) can be expanded to obtain:

\[
\sum I_{i,j} f(u + i, v + j) t(i,j) - \bar{f} \sum I_{i,j} f(u + i, v + j)
\]

(3)

The first term of (3) can be regarded as the convolution of two signals in the spatial domain, and the convolution for the spatial domain is equivalent to the coefficient multiplication operation based on the frequency domain. Therefore, the first term of (3) is equivalent to the following form:

\[
F^{-1}[F(f) F^*(t)]
\]

(4)

where, \(F\) is the Fourier transform of the original signal, \(F^*\) is the conjugate complex operation of the transformed result, and \(F^{-1}\) is the inverse Fourier transform of the frequency domain signal.

Next, for the denominator of (1), the latter is the pixel value variance of the template image T. The first term can be simplified as follows:

\[
\sum I_{i,j} (f(u + i, v + j))^2 - \left[\sum I_{i,j} f(u + i, v + j)\right]^2
\]

(5)

If (5) is calculated, it is necessary to obtain the cumulative sum of all pixel values and the sum of squared values in the sliding window of the matching image S at \((u,v)\). It can be seen from (5) that direct calculation of correlation coefficient requires a lot of multiplications and additions. The larger the template image and the matching image are, the more additions and multiplications are needed, and the growth is rapid. When a new point is reached, parameters such as correlation value and sum of squares need to be recalculated, which is very wasteful. Therefore, combining the strategies of down sampling and dynamic programming, this paper further optimizes the above calculation process.
3. The Proposed Method

3.1. Calculate the matrix of cumulative sum and the sum of squares

In template image matching, it is necessary to calculate the sum of pixels and the sum of squares of any sub-matrix of matching image S in real-time. Therefore, in order to improve the matching efficiency, based on the dynamic programming strategy, the cumulative sum and sum of squares of sub-matrices are calculated in advance, and stored in the cache data table for use in the subsequent calculation process, so as to avoid repeated calculation. In the following, we will take the cumulative sum matrix of the image S as an example to describe the accelerated calculation method.

(1) Matrix extension. Firstly, the image area of the image S to be matched with the same size of T is expanded into a matrix K of 3M*3N. For convenience, the selected image area is still marked as S. The expansion method is to place S in the center of the matrix K, and set all the values of surrounding 8 regions be zero, as shown in Fig.2.

(2) The following calculations are performed on the expanded matrix K:

Step1. The calculation of matrix K1: Starting from the second row of K, the elements of each row are added to the elements of the same column in the next row to obtain:

$$K1(i, j) = K(i - 1, j) + K(i, j)$$

Step2. The calculation of matrix K2: From the first line of K1 to the end of line 2N, it calculates as follows:

$$K2(i, j) = K1(i + N, j) - K1(i, j)$$

Step3. The calculation of matrix K3: For K2, starting from the second column, it adds the elements of each column with the previous column of the same row to obtain:

$$K3(i, j) = K2(i, j - 1) + K2(i, j)$$

Step4. The calculation of matrix K4: From the first column of K3 to the end of 2M column, it calculates as follows:

$$K4(i, j) = K3(i, j + M) - K3(i, j)$$

(3) Matrix of cumulative sum SB: Let the matrix of cumulative sum SB be the sub-matrix with the size of 2M*2N in the upper left corner of K4, as shown in Fig.3.
Fig. 3 The composition of the cumulative sum matrix SB

As for any point $((i, j))$, the $S_1(i, j)$, $S_2(i, j)$, $S_3(i, j)$ and $S_4(i, j)$ represent the sum of some elements in the sub-matrix $S_1$, $S_2$, $S_3$ and $S_4$, respectively, just as shown in the shaded part of Fig. 4.

![Diagram of cumulative sum matrix](image)

Fig. 4 Diagram of cumulative sum matrix

3.2. Similarity calculation of image matching

For the matching image $S$, the cumulative sum matrix $SB$ can be obtained by using the method described in Section 3.1. At the same time, all the elements in $S$ are multiplied, and then the cumulative sum of squared value matrix $S^2B$ is obtained in the same way. Then, the calculation method of image matching similarity is as follows:

1) The matching coefficient $\gamma$ is calculated as follows:

$$\gamma = \sqrt{(m \times n - 1) \times \text{std}(T)}$$

(11)

where $\text{std}(T)$ is the variance of all elements in the template image $T$.

2) Calculate the denominator of the correlation measure of (1):

$$\alpha(i, j) = \gamma \sqrt{\max \left( S^2B(i, j) - \frac{SB(i, j)^2}{m \times n}, 0 \right)}$$

(12)

3) Calculate the molecular of the correlation measure of (1):

$$\beta(i, j) = r(i, j) - \frac{SB(i, j) \times \text{sum}(T)}{m \times n}$$

(13)

where $\text{sum}(T)$ is the sum of all elements in the template image $T$, and $r(i, j)$ is the value of the corresponding term in the convolution matrix $r$.

4) In the final template matching similarity matrix $C$, the value of each item is calculated as follows:
The similarity degree of two images is the maximum element value of matrix C, and if the maximum coordinate is \((i_m,j_m)\), then the relative offset after image matching is \((i_m - m, j_m - n)\).

3.3. Further optimization

Based on the template matching method proposed in Section 3.1 and 3.2, the following two methods are used to further improve the efficiency of template matching.

1) Reduce the number of sliding windows. In practical applications, there are a large number of sliding windows, which are of no computational value and will not be used in practice. For example, if the upper left corner of the image to be matched is taken as the coordinate origin, the upper left corner of the template image \(T\) can be slid from \((-m/2,-n/2)\) to \((M - m/2,N - n/2)\), and the the size of cumulative sum matrix is changed from \(2M*2N\) to \((M + m/2)*(N + n/2)\). When the size of template image is equal to the size of the matching image, the amount of data calculation is only half of the original.

2) Down sampling process. At present, with the continuous improvement of the equipment quality of mobile phones and digital cameras, the captured images have a large number of pixels, so the image precision is very high. But such a high precision will cost a lot of time in image matching. Therefore, the image to be matched and the template image can be down-sampled, which can further reduce the amount of calculation and storage space while ensuring the correct matching. For example, selecting the odd-numbered rows and odd-numbered columns of the original image to form a template image \(T\) with a size of \((m/2*n/2)\) and an image \(S\) to be matched with a size of \((M/2*N/2)\), the complexity is about 1/4 of the original algorithm.

4. Experimental Results and Analysis

For the purpose of verifying the accuracy and speed of the proposed method, two experiments are designed and performed:

1) Font type detection. The font type of the character image is determined by matching the character image of the specific font type with multiple standard images of the same character which are generated by the operation system.

2) Scene image matching. A part of the matching image \(S\) is intercepted as the template image \(T\), and the image matching experiment is done with the image \(S\) and \(T\).

4.1. Font type detection experiment

First of all, the English character image "S" is used for the font type detection experiment. The original standard character images automatically generated by the operation system are regarded as the template character images. The font size is 40 points, and the font types are Times New Roman, Arial and Verdana, respectively, as shown in Fig.5 (a). Fig.5 (b) shows the character image to be matched which is obtained by taking screenshot of the "S" in the Times New Roman style document, and then after binarization, it is scaled to the image size of Fig.5 (a). In the experiment, the template matching method proposed in this paper with reducing sliding window and the method in [16] were used to retrieve the character image with the highest similarity among Fig.5 (a), and the processing efficiency of two template matching methods was compared.

(a) The template images generated by the operation system    (b) the character image to be matched

Fig.5 Font type detection experiment images
The method in [16] and our proposed method are used to perform the experiment of font type detection. In our method, firstly, the character images in Fig.5 (a) are treated as template images and moved on the matching image Fig.5 (b), and the corresponding maximum correlation of character image is obtained by using the methods of Section 3.1 and 3.2. Due to the less details of the character image, the correlation values in the early moving process are small, and do not have much reference value. Therefore, the amount of computation can be further reduced by reducing the number of sliding windows, and the efficiency of image matching can be accelerated. The result of font type detection by using the proposed method is shown in Fig. 6. It can be seen that the character image of Times New Roman type in Fig.5 (a) has the highest matching degree with that in Fig.5 (b). Therefore, the detected font type is Times New Roman. The comparison results of character image matching efficiency and accuracy of the two methods are shown in Table.1.

From the results of Fig.6 and Table.1, it can be seen that both the method in [16] and our proposed method can search for the image that precisely matches the template image. There are some differences in the way of zero-padding for different number of sliding windows in the convolution operation, which leads to a small error in the similarity calculation between these two methods, but this error does not affect the final result. However, the calculation amount of the actual cumulative sum is only close to 1/2 of that in [16]. Therefore, without affecting the accuracy of image matching, the image matching efficiency of the proposed method is significantly improved.

### 4.2 Scene image matching experiment

In the scene image experiment, we select an original image with size of 1150*748 pixels, as shown in Fig.7 (a). It performs the color level transformation and filter processing on Fig.7 (a) to obtain Fig.7 (b) and (c), respectively. These three images are taken as the images to be matched, and the window area marked by rectangle are selected from Fig.7 (a) as the template image with the size of 69 * 89 pixels. The enlarged effect of template image is shown in Fig.7 (d). It slides the templated image on the images to be matched for calculating the correlation value by using the method in [16] and our proposed method. In our method, the template image and the image to be matched are both down-sampled. The result of template matching is shown in Fig.8, and the comparison result of scene image matching efficiency is shown in Table.2.
Fig. 7 Scene matching experiment images

Table 2 Comparison results of scene image matching efficiency

| Images          | Fig.5 (a) | Fig.5 (b) | Fig.5 (c) |
|-----------------|-----------|-----------|-----------|
| Method in [16]  | 24.656s   | 24.670s   | 24.735s   |
| Our proposed method | 6.002s   | 6.123s   | 6.330s    |
Due to the high resolution of the original image, the calculation complexity is high when directly matching the original image, so the images are down-sampled before the image matching. From the matching result shown in Fig.8, the main details of the image are still completely preserved after down-sampling, and the image matching accuracy is not affected. At the same time, it can be seen from Table.2 that, compared with the method in [16], the amount of image data after down sampling processing becomes 1/4 of the original one, and the efficiency of image matching is greatly improved.

5. Conclusion
This paper mainly discusses a fast template matching algorithm based on NCC. Firstly, the normalized template matching metric formula is equivalently transformed, and the original correlation calculation is replaced by the image convolution and the cumulative sum of calculation. Then, based on dynamic programming strategy, a lot of repeated calculations are reduced by calculating the accumulated sum and accumulated sum of squared value of any pixel, so as to improve the efficiency of image template matching. To reduce the meaningless calculation process in the image matching sliding process, it adopts the method of deleting the sliding window to reduce calculation times. Finally, the down-sampling process is performed on the high-resolution images obtained by the high-precision device, which further reduces the amount of calculation and storage space. In short, the fast template matching algorithm proposed in this paper not only ensures high matching accuracy, but also has high matching efficiency.

Acknowledgments
This work was supported by National R&D project of China under contract No. 2018YFB0803702.

References
[1] Chen Tianhua, Wang Fulong. Real-time robust feature-point matching algorithm[J]. Journal of Image and Graphics, 2016, 21(09):1213-1220(in Chinese).
[2] Liu Xiaohong, Zhu Yuquan, et al. Liver CT Image Feature Extraction Method Based on Improved Multi-scale LBP Algorithm[J]. Computer Science,2019,46(3):125-130 (in Chinese).
[3] Zhao Wu, Duan Yongxuan, et al. Analysis and Research on Image Feature Points Extraction by SIFT and Harris [J]. Computer Technology and Development, 2018,28(12):62-66 (in Chinese).
[4] Qiao Yuqing, Geng Zexun, et al. Speeded Up Robust Feature Descriptors and Matching Algorithm Based on Brightness Order[J]. Journal of Surveying and Mapping Science and Technology, 2017,34(06):617-621 (in Chinese).
[5] You Tingting, Tang Yan. Visual saliency detection based on adaptive fusion of color and texture features[C]. The 3rd IEEE International Conference on Computer and Communications (ICCC), Chengdu, 2017: 2034-2039.
[6] Dai Xiance, Liu Changjin. Fast Template-Matching Algorithm. Journal of Ordnance Equipment Engineering, 2015, 36(1):112-117(in Chinese).
[7] Han Bing, Wang Yongming. Research of Image Matching Based on a Fast Normalized Cross Correlation Algorithm[J]. ACTA ARMAMENTARII, 2010, 31(2):160-165(in Chinese).
[8] Wang Yongming. A consistent decision algorithm for N frames continuous scene matching [J]. Chinese Journal of Computer, 2005, 28(6):1032-1035. (in Chinese).

[9] Li Zhuo, Qiu Huijuan. Fast image matching based on correlation coefficient[J]. Journal of Beijing Institute of Technology, 2007, 27(11):998-1000(in Chinese).

[10] Zhu Mingbo, Zhang Dongxing, et al. Research on Adaptability to Noise of Image Matching Algorithms Based on Intensity Correlation[J]. Journal of Naval Aeronautical and Astronautical University, 2009(1):62-62(in Chinese).

[11] Jae ChernYoo, TaeHee Han. Fast Normalized Cross-Correlation[J]. Circuits, System and Signal Process, 2009,28:819–843.

[12] Tsai Du-Ming, Lin Chien-Ta. Fast normalized cross correlation for defect detection[J]. Pattern Recognit. Lett. 2003, 24(15), 2625–2631.

[13] J. P. Lewis. Fast Template Matching[J]. Vision Interface, 1995:120-123.

[14] Ma Guohong, Wang Cong, et al. Sequential Similarity Detection Algorithm Based on Image Edge Feature[J]. Journal of Shanghai Jiaotong University (Science), 2014(1):79-83(in Chinese).

[15] Hisham M B, Yaakob S N, et al. Template Matching using Sum of Squared Difference and Normalized Cross Correlation[C]. IEEE Student Conference on Research & Development. IEEE, 2016:100-104.

[16] Qi Wenfa, Guo Wei, et al. Robust Authentication for Paper-based Text Documents based on Text Watermarking Technology[J]. Mathematical Biosciences and Engineering, 2018, 16(4): 2233–2249.