Research and Comparison of Image Corner Detection Algorithm based on C#

JIN Xian-hua , MA xiao-jun

Computer Teaching Department, Anyang Normal University, Anyang Henan 455000, China

Abstract

Image matching is a basic problem in image processing. Image corner detection plays a very important role in the field of image matching and image understanding. The article describes three corner detection algorithm based on grayscale and uses C# to program and compare the extraction results. The test results show that the Harris algorithm has a good detection rate and good stability in most scenarios, has practical significance for further study.

Keywords: Image Matching, Corner Detection, Harris Algorithm

1. Introduction

The image matching is a process aligning two images in space and determining the relative offset between the two grayscale images caught by two different sensors from the same scene. The corner is a point whose 2D image brightness change violently [1-2]. The point is different from the around points and has curvature maximum value on the edges of the image curve. [3]

With the development of science and technology, image matching image information processing industry has become one of the key technology. The current most of the machine vision applications need image matching technology. Image matching is through the certain matching algorithm in two or more picture image recognition between the point of the process. Image matching is through the certain matching algorithm in two or more images of recognition between the same point, such as 2 d image matching by comparing the target area and search area of the same size of the window of the correlation coefficient, take the search area of the largest correlation coefficient of the corresponding window center as the same place. Its essence is in the primitive similarity conditions, using the best matching criterion search problem. Image matching is mainly can be divided into gray based matching and signature-based matching. Gray matching principle: the image as a two dimensional signal, through the corresponding statistical method for signal between the matching relation, and then through the two signal correlation function, analysis the similarity between them and get the point. Feature matching refers to two or more separately collection image features, the characteristic parameters, and then using description described parameters to match an algorithm.

Image matching common features edge, linear, interest points, etc. Based on the characteristics of the image matching for image distortion has some robustness, matching performance and feature extraction quality has strong relevance. Images of the characteristic information of the image processing is the foundation of the process. In the image characteristics of, the Angle point contains information data quantity minimum, also is the smaller data information storage the image gray scale fluctuation characteristic information, and the interference factor diagonal point feature extraction algorithm is less interference. Therefore corner detection algorithm has an important application significance. Image matching technology has a wide range of applications, such as military target tracking and orientation, the medical image forming, geological prospecting, etc. Image matching algorithm adaptability, reliability, timeliness and positioning accuracy is reflected matching algorithm of the main technical indicators.

Corner detection based on gray level of the assay, analysis pixels range of gray level change, calculated point curvature and gradient to detect image Angle point. Line feature is the image significant line features, such as buildings, road edge, etc. The traditional edge detection based on image line characteristics on the image of each pixel point for differential and for second order differential and then determine the edge pixels.

Common corner detection algorithm includes Harris algorithm, SUSAN algorithm and MIC algorithm. The Harris algorithm is based on a second-order directional derivative has good effect but
large amount of computation. SUSAN\cite{4} algorithm has higher efficiency and strong anti-noise ability. MIC algorithm has high precision and good stability.

2. Corner detection algorithms

2.1. Harris algorithm

Harris algorithm has a high stability and robustness. It can accurately extract the corner in the rotating images or gray changing images which is suitable for the extraction of multi-sensors image corner feature. The Moravec algorithm is the predecessor of the Harris algorithm. The following figures give the correlation window of four different positions.

![Figure 1](image1.png)

**Figure 1** The correlation window of four different positions

Fig.1 is four different positions: (a) represents that the current pixel is located inside the object on the assumption that the gray level is invariant in any case such as moving windows, (b) indicates that the pixel is on the edge of the object, the gray level change greatly if move associated window along the vertical edge, (c) represents the pixel is corresponding to a corner, the gray level change slightly if move along edges, (d) indicates the pixel is corresponding to a isolated point, the gray value change greatly no matter how to move the associated window.\cite{5,6}

The corner response function of Harris algorithm is defined as:

\[
R = Det(M) - kTr(M)^2
\]  

(1)

Corner response criterion \( R \) is positive in the corner region, negative on the edge area and very small in the unchanging area\cite{7,8}. In the practical application, the point is a corner point if \( R \) greater than a given threshold value.

The limitation of Harris algorithm is that it use the higher order partial derivative method, therefore it is sensitive to the noise and has high computational complexity when carry out Gaussian convolution operation. The gradient has anisotropic because of the only considering of the horizontal and vertical directions when calculated.

2.2. SUSAN algorithm

SUSAN algorithm is based on Unvalue Segment Assimilating Nucleus. SUSAN algorithm uses a circular template (also known as the window or nuclear) with isotropic feature for corner detection. The general template radius is 3-4 pixels. Fig.2 is a 7*7 template with a fixed threshold \( g = 37 \times 1/2 \).
The luminance values of the points in the template are compared with that of the nuclear point (current point) by the following similarity function. Here we use the template with 37 pixels.

\[ c(\vec{r}, \vec{r}_0) = \begin{cases} 1, & |I(\vec{r}) - I(\vec{r}_0)| \leq t \\ 0, & |I(\vec{r}) - I(\vec{r}_0)| > t \end{cases} \] (2)

\( c \) is an output function of comparison calculated by the pixels in the template. In the equation, \( \vec{r}_0 \) represents the location of the nuclear point of the 2D image, \( \vec{r} \) is the location of other points, \( I(\vec{r}_0) \) is the luminance value of a pixel in the template, \( t \) is a threshold of luminance difference. USAN can be calculated by the following equation:

\[ n(x_0, y_0) = \sum_{(x,y) \in (x_0, y_0)} c(x, y) \] (3)

\( n \) represents the number of pixels in area USAN. Comparing \( n \) with a special threshold value \( g \), SUSAN corner detection algorithm will set \( g \) as half of the \( n_{max} \) (the maximum area of \( n_{max} \)). The initial response function can be obtained through the following equation.

\[ R(\vec{r}_0) = \begin{cases} g - n(\vec{r}_0), & n(\vec{r}_0) < g \\ 0, & n(\vec{r}_0) \geq g \end{cases} \] (4)

That is a simple SUSAN algorithm formula, \( g \) defines the maximum value of USAN of the output corner point. Threshold \( t \) in formula (4) represents the minimal contrast ratio of the corner point detected, which is the maximum tolerance to noise. The value of \( t \) determines the quantity of the feature extracted. The smaller \( t \) is, the more feature extracted from the image with low contrast ratio. Therefore, for different contrast ratio and different noise image the \( t \) values are also different.

In actual applications, the stable comparison function shown as follow is often adopted:

\[ c(\vec{r}, \vec{r}_0) = \exp\left(-\left|\frac{I(\vec{r}) - I(\vec{r}_0)}{t}\right|^6\right) \] (5)

SUSAN corner detection algorithm can be described as: Along pixels be detected, in accordance with the order traversal each point on the image. When reach to a pixel, this pixel point is regarded as the core. Then scan along the arc with radius of R to identify the point on the arc where the gray-scale variation strongly.
2.3. MIC algorithm

MIC uses the definition of USAN. Suppose that any line cross the nuclear point has two intersections with the circle window \( P \) and \( P' \). Define the CRF (Corner Response Function) as follow:

\[
R_C = \min((f_p - f_C)^2 + (f_{p'} - f_C)^2)
\]  

(6)

\( C \) is the nuclear point, \( f_p \) grey level of point \( P \).

Figure 3. the linear difference diagram

The detection principle of MIC algorithm is to find the minimal brightness variation \( R \) in special circular window. If \( R \) larger than the threshold, it is the corner, else isn’t. We usually use a nearly circular discrete window in the calculation.

\[
R_C = \min_{P, P' \in S_n} ((f_p - f_C)^2 + (f_{p'} - f_C)^2)
\]  

(7)

\( C \) is the nuclear point. Point \( P \) and point \( P' \) are symmetrical on \( C \).

CRF calculation is divided into two steps. First the change of brightness along the horizontal and vertical should be calculated:

\[
\begin{align*}
r_A &= (f_a - f_C)^2 + (f_{a'} - f_C)^2 \\
r_B &= (f_b - f_C)^2 + (f_{b'} - f_C)^2
\end{align*}
\]  

(8)

The value of the corner response function is:

\[
R = \min(r_A, r_B)
\]  

(9)

A linear internal pixel interpolation method is used to calculate the change value of minimum luminance: \( CRF \)

\[
R = \min(r_1(x), r_2(x))
\]  

(10)

Parameter \( x \) is defined as the intersections of any line across \( N \) and the circle window, \( x \in (0, 1) \). Usually we take the position of the intersection of square side.
The reaction function is:

\[
\begin{align*}
    r_1(x) &= (f_p - f_C)^2 + (f_{p'} - f_C)^2 \\
    r_2(x) &= (f_Q - f_C)^2 + (f_{Q'} - f_C)^2
\end{align*}
\]  

(11)

Since the circular symmetry, the equation can be simplified as:

\[
\begin{align*}
    f_p &= (1 - x)f_A + x f_B \\
    f_{p'} &= (1 - x)f_{A'} + x f_{B'} \\
    f_Q &= (1 - x)f_{A'} + x f_B \\
    f_{Q'} &= (1 - x)f_A + x f_{B'}
\end{align*}
\]  

(12)

\[
\begin{align*}
    r_1(0) &= r_2(0) = r_A, r_1(1) &= r_2(1) = r_B. \text{ Take (12) into (11):}
\end{align*}
\]

\[
\begin{align*}
    r_1(x) &= A_1 x^2 + 2B_1 x + C \\
    r_2(x) &= A_2 x^2 + 2B_2 x + C
\end{align*}
\]  

(13)

In the equation:

\[
\begin{align*}
    A_1 &= r_B - r_A - 2B_1 \\
    A_2 &= r_B - r_A - 2B_2 \\
    B_1 &= (f_B - f_A)(f_A - f_N) + (f_{B'} - f_{A'})(f_{A'} - f_{N'}) \\
    B_2 &= (f_B - f_A)(f_A - f_N) + (f_{B'} - f_{A'})(f_{A'} - f_{N'}) \\
    C &= r_A
\end{align*}
\]  

(14)

To ensure the CRF value the largest in the minimum value, we get A>0. Corner response function has minimum value on the square, so A<1, B<0, A+B>0, then we obtain the value of CRF as:

\[
R = C - B^2 / A
\]  

(15)

If this value doesn’t meet the requirement, the value will be calculated by equation (9).

3. System implementation

1) Development environment
Here we use C#. C# is a development tool of .NET framework, a modern & object-oriented programming language.

2) System Framework
System interface is mainly divided into two parts: basic operation and corner detection.
System function module chart is shown as follow:
3) Harris algorithm realization:

① Suppose the gray level of pixel whose coordinate is \( (x,y) \) in the image \( I(x,y) \). Pointer is \( I[x, y] \). Each pixel of the image is filtered by horizontal and vertical difference operator to get \( I_x, I_y \):

\[
I \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} = I_x \quad I \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} = I_y
\]  

\( (16) \)

② \( I_x^2, I_y^2 \) and \( I_x \cdot I_y \) can be calculated.

③ The 5*5 2D Guass template with zero mean is set as

\[
gauss[i, j] = \exp\left(-\frac{(i - \frac{w}{2})^2 + (j - \frac{w}{2})^2}{2 \cdot \sigma^2}\right)
\]  

\( (17) \)

Normalize the template, final Guass template is obtained.

④ Gaussian filter the three elements, that is use Guass template convolute with \( I_x^2, I_y^2 \), \( I_x \cdot I_y \). The characteristic values for the correlation matrix \( M \) of each pixel can be calculated through these values.

\[
M = \begin{bmatrix} I_x^2 & I_x \cdot I_y \\ I_x \cdot I_y & I_y^2 \end{bmatrix}
\]

\( (18) \)

⑤ The corner response function \( cim \) is calculated:

\[
cim[i, j] = (I_x^2 \cdot I_y^2 - I_{xy}^2) - K(I_x^2 + I_y^2)^2
\]  

\( (19) \)

The value of \( K \) is between 0.04 and 0.06, here \( K \) equals to 0.05.

⑥ Seeking local maximum to determine the final corner.

In matrix \( cim \), if \( cim \) is the local maximum value and larger than \( \text{thresh} \), the point is considered the corner.
4) Code implementation

An array is defined to store binary data, and its code is shown below:

```csharp
public class Circle()
    
public int[,] im2bw(Bitmap image1)
    {
    int[,] bw=new int[image1.Height,image1.Width];
    Color C=new Color();
    int r=0;
    for(int a=0;a<image1.Height;a++)
        for(int b=0;b<image1.Width;b++)
        {
            C=image1.GetPixel(b,a);
            r=C.R;
            bw[a,b]=r;
        }
    return bw;
    }
```

4. Experiment analyses

Figure 5 is the standard image commonly used in the field of corner detection, which contains a large number of distinct geometry corner information while the transition regions on the left side contains unobvious corner information. Figure 6 is a plus-noise effect diagram used to detect the noise resistance of different algorithms. In order to compare different algorithms for different image processing, the experiment also used building blocks diagram Fig.7 and Housing diagram Fig.8 for comparison.

1) Compare the effect of different parameters to the three algorithms in Fig.4.
(1) Harris corner detection algorithm
The three parameters mentioned above have some effect in Harris corner detection.

![Harris corner detection](image)

(a) Threshold 50  
(b) Threshold 500

**Figure 9.** Harris changing threshold test

The other data is in Table 4.1:

### Table 1. Harris changing parameter test

<table>
<thead>
<tr>
<th></th>
<th>Threshold</th>
<th>50</th>
<th>500</th>
<th>5000</th>
<th>25000</th>
<th>50000</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Corner number</td>
<td>285</td>
<td>131</td>
<td>100</td>
<td>47</td>
<td>25</td>
</tr>
<tr>
<td>2</td>
<td>Guass window</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>9</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>Corner number</td>
<td>131</td>
<td>131</td>
<td>129</td>
<td>118</td>
<td>116</td>
</tr>
</tbody>
</table>

### Table 2. SUSAN changing parameter test

<table>
<thead>
<tr>
<th></th>
<th>Threshold t</th>
<th>10</th>
<th>50</th>
<th>100</th>
<th>150</th>
<th>250</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Corner number</td>
<td>71</td>
<td>37</td>
<td>33</td>
<td>18</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>Threshold g</td>
<td>0.1</td>
<td>0.3</td>
<td>0.5</td>
<td>0.7</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Corner number</td>
<td>2</td>
<td>14</td>
<td>71</td>
<td>1546</td>
<td>3358</td>
</tr>
</tbody>
</table>

### Table 3. MIC changing parameter test

<table>
<thead>
<tr>
<th>threshold</th>
<th>500</th>
<th>5000</th>
<th>25000</th>
<th>50000</th>
<th>80000</th>
<th>100000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corner number</td>
<td>1169</td>
<td>5000</td>
<td>1580</td>
<td>63</td>
<td>25</td>
<td>8</td>
</tr>
</tbody>
</table>

(2) SUSAN corner detection algorithm
The related data is in Table 2:

(3) MIC corner detection algorithm
The related data is in Table 3:

2) The comparison of detection time and corner number of three algorithms, the results are shown in Table 4.
### Table 4. Result statistics

<table>
<thead>
<tr>
<th>Figure number</th>
<th>Algorithm</th>
<th>Detection time (ms)</th>
<th>Average time (ms)</th>
<th>Corner number</th>
<th>Average number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fig. 4</td>
<td>Harris</td>
<td>1 8</td>
<td>8</td>
<td>131</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 7</td>
<td></td>
<td>131</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 9</td>
<td>8</td>
<td>131</td>
<td>131</td>
</tr>
<tr>
<td></td>
<td>SUSAN</td>
<td>1 32</td>
<td></td>
<td>71</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 31</td>
<td></td>
<td>71</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 33</td>
<td>32</td>
<td>71</td>
<td>71</td>
</tr>
<tr>
<td></td>
<td>MIC</td>
<td>1 0</td>
<td></td>
<td>1169</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 0</td>
<td></td>
<td>1169</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 0</td>
<td>0</td>
<td>1169</td>
<td>1169</td>
</tr>
<tr>
<td>Fig. 5</td>
<td>Harris</td>
<td>1 39</td>
<td></td>
<td>109</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 8</td>
<td></td>
<td>109</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 25</td>
<td>24</td>
<td>109</td>
<td>109</td>
</tr>
<tr>
<td></td>
<td>SUSAN</td>
<td>1 48</td>
<td></td>
<td>539</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 52</td>
<td></td>
<td>539</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 50</td>
<td>50</td>
<td>539</td>
<td>539</td>
</tr>
<tr>
<td></td>
<td>MIC</td>
<td>1 0</td>
<td></td>
<td>327</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 0</td>
<td></td>
<td>327</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 0</td>
<td>0</td>
<td>327</td>
<td>327</td>
</tr>
<tr>
<td>Fig. 6</td>
<td>Harris</td>
<td>1 4</td>
<td></td>
<td>252</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 8</td>
<td></td>
<td>252</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 5</td>
<td>6</td>
<td>252</td>
<td>252</td>
</tr>
<tr>
<td></td>
<td>SUSAN</td>
<td>1 57</td>
<td></td>
<td>1267</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 58</td>
<td></td>
<td>1267</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 58</td>
<td>58</td>
<td>1267</td>
<td>1267</td>
</tr>
<tr>
<td></td>
<td>MIC</td>
<td>1 0</td>
<td></td>
<td>566</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 0</td>
<td></td>
<td>566</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 0</td>
<td>0</td>
<td>566</td>
<td>566</td>
</tr>
</tbody>
</table>

With high computational complexity, Harris algorithm is sensitive to noise. The influence of noise can be reduced by increasing the Gauss window. Harris algorithm is anisotropic, whose performance is strongly reaction to the corners making it's easy to error detection.

SUSAN algorithm has strong anti noise capability and high accuracy which can be used to detect all types of corners. But the calculation of SUSAN algorithm is complex, when the contrast level is not the same in different area, the result under fixed threshold often do not meet the actual situation, and extraction effect is not evident in the gradient region.

MIC corner detection algorithm has high accuracy, good stability and fast operation. But it does not have rotation invariance and sensitivity to noise, the edge point can be erroneously detected as a corner, these factors affect the stability of the algorithm. When using a large template operator, a lot of corners will be missed and corner location is inaccurate.
5. Conclusions

Overall, Harris algorithm is not affected by light & rotation, although with low positioning accuracy and high computational complexity, it has a good detection rate & anti-noise performance and good stability in most cases, so it is the optimal algorithms in this article. For many applications, the positioning accuracy is not the most important, at the same time, due to the rapid development of computer hardware and increasing computational efficiency, the Harris algorithm has been used most widely.

6. References