IMPROVED HARRIS' ALGORITHM FOR CORNER AND EDGE DETECTIONS

Soo-Chang Pei, *Jian-Jiun Ding,*

Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, R.O.C
Email address: pei@cc.ee.ntu.edu.tw, djj@cc.ee.ntu.edu.tw

ABSTRACT

A more accurate algorithm for corner and edge detections that is the improved form of the well-known Harris' algorithm is introduced. First, instead of approximating $|L[m+x, n+y]-L[m, n]|^2$ just in terms of x^2, xy, and y^2, we will approximate $|L[m+x, n+y]-L[m, n]|(L[m+x, n+y]-L[m, n])$ by the linear combination of x^2, xy, y^2, x, y, and 1. With the modifications, we can observe the sign of variation. It can avoid misjudging the pixel at a dot or on a ridge as a corner and is also helpful for increasing the robustness to noise. Moreover, we also use orthogonal polynomial expansion and table looking up and define the cornity as the "integration" of the quadratic function to further improve the performance. From simulations, our algorithm can much reduce the probability of regarding a non-corner pixel as a corner. In addition, our algorithm is also effective for edge detection.

Index terms: — corner detection, edge detection, quadratic polynomial, ridge detection, noise immunity

1. INTRODUCTION

Corner detection is important for feature extraction and pattern recognition. In [1], Harris and Stephens proposed a corner detection algorithm. First, they used a quadratic polynomial to approximate the variation around $[m, n]$:

$$|L[m+x,n+y]-L[m,n]|^2$$
$$= A_{m,n}x^2 + 2\,C_{m,n}x\,y + B_{m,n}y^2 + \text{remained terms}, \quad (1)$$

where $A_{m,n}$, $B_{m,n}$, and $C_{m,n}$ were calculated from the correlations between the variations and a window function:

$$A_{m,n} = X^2 \otimes w\,,\ \ B_{m,n} = Y^2 \otimes w\,,\ \ C_{m,n} = XY \otimes w\,, \quad (2)$$

$$X = L[m,n]\otimes[-1,0,1], \qquad Y = L[m,n]\otimes[-1,0,1]^T,$$
$$w_{x,y} = \exp[-(x^2+y^2)/2\sigma^2]. \quad (3)$$

Then the variations along the principal axes can be calculated from the eigenvalues of the following 2×2 matrix:

$$\mathbf{H}_{m,n} = \begin{bmatrix} A_{m,n} & C_{m,n} \\ C_{m,n} & B_{m,n} \end{bmatrix}. \quad (4)$$

If both the two eigenvalues of $H_{m,n}$ are large, then we recognize the pixel $[m, n]$ as a corner. Harris and Stephens proposed a systematic and effective way to detect corners. However, the algorithm has some problems, such as the ability to distinguish the corner from the peak or the dip and the robustness to noise should be improved. In [5], an improved algorithm based on modifying the detection criterion was proposed. In this paper, we apply many new ideas listed in Section 2 to improve Harris' algorithm for corner detection.

2. IDEAS FOR IMPROVING PERFORMANCE

(A) Instead of approximating $|L[m+x, n+y]-L[m, n]|^2$, we use a quadratic polynomial to approximate $L_1[m, n, x, y]$:

$$L_1[m, n, x, y]$$
$$=|L[m+x,n+y]-L[m,n]|(L[m+x,n+y]-L[m,n])\,.(5)$$

This modification is helpful for improving the **robustness to noise**. Suppose that the image is interfered by a noise $\eta[m, n]$, i.e., $H[m, n] = L[m, n] + \eta[m, n]$, and

$$E[\eta[m, n]] = 0, \quad E[\eta[m, n]\eta[m+x, n+y]] = \tau_{x,y}, \quad (6)$$
$$\text{prob}\{\eta[m, n] = k\} = \text{prob}\{\eta[m, n] = -k\}, \quad (7)$$

then we can prove that

$$E\{|H[m+x,n+y]-H[m,n]|^2\}$$
$$= |L[m+x,n+y]-L[m,n]|^2 + 2(\tau_{0,0}-\tau_{x,y})\,, \quad (8)$$
$$E\{|H[m+x,n+y]-H[m,n]|(H[m+x,n+y]-H[m,n])\}$$
$$= |L[m+x,n+y]-L[m,n]|(L[m+x,n+y]-L[m,n])\,. \ (9)$$

Thus, the mean of (1) is affected by noise, but from (9) the mean of $L_1[m, n, x, y]$ will be not affected by noise.

The sign of variation is also important to distinguish a corner from a peak. For both a corner pixel and a peak, the variations along all the directions are large, as in Fig. 1. If we observe only the amplitude of variations, as Harris' algorithm, they are hard to distinguish. In contrast, if the sign of variations can be observed, we can distinguish them. For a peak, the signs of variations along all the directions are negative. For a corner, sometimes variation is positive and sometimes it is negative.

(B) We will approximate the variation by the linear combination of x^2, xy, y^2, x, y, and 1, see (16). For Harris' algorithm in (1), only the terms of x^2, xy, and y^2 are used. Note that x^2 and y^2 are **even-even** bases (i.e., even respect to both x and y) and xy is an **odd-odd** basis. Using them is hard to observe even-odd and odd-even variations. Thus, to observe all types of variations, we should use the terms of x (an **odd-even** basis) and y (an **even-odd** basis). With these terms, we can observe more styles of variations and classify the pixel into more types.

(C) Instead of (3), we use "orthogonal polynomial expansion" to find the coefficients. It can assure that the mean square error of approximation to be minimal [2].

(D) In [1], Harris classified the pixel into only three types: corners, edges, and plains. In this paper, with the help of Case table (see Table 1), we classify the pixel into corners, edges, ridges, valleys, peaks, saddles, and plains.

(E) From derivations and experiments, we find that, to achieve the best accuracy, it is better to define the **cornity** as the "integration" of the quadratic function. See Step 6.

3. ALGORITHM

(Step 1) First, find the orthonormal polynomial set that is orthogonal respect to a weighting function $w[m, n]$. That is,

$$\varphi_k[x, y] = a_{1,k} + a_{2,k}x + a_{3,k}y + a_{4,k}x^2 + a_{5,k}xy + a_{6,k}y^2,$$

$$\text{for } k = 1, 2, 3, 4, 5, 6, \tag{10}$$

and the parameters $a_{j,k}$ should be chosen properly such that

$$\sum_x \sum_y \varphi_k[x, y]\varphi_h[x, y]w[x, y] = \delta_{k,h}. \tag{11}$$

We can use the Gram-Schmidt algorithm to find $a_{j,k}$. For example, if we choose the weighting function as

$$w[x, y] = \exp[-(x^2 + y^2)/20] \quad \text{for } (x^2 + y^2)^{1/2} < 6.5, \tag{12}$$

$$w[x, y] = 0 \qquad \text{for } (x^2 + y^2)^{1/2} > 6.5, \tag{13}$$

then the orthogonal polynomials are

$$\varphi_1[x, y] = 0.1339, \ \varphi_2[x, y] = 0.0498x, \ \varphi_3[x, y] = 0.0498y,$$

$$\varphi_4[x, y] = 0.0157y - 0.1131, \quad \varphi_5[x, y] = 0.0206xy,$$

$$\varphi_6[x, y] = 0.0158y^2 + 0.00213x^2 - 0.1296. \tag{14}$$

(Step 2) Then, we do the inner product for $L_1[m, n, x, y]$ is (defined in (5)) and $\varphi_k[x, y]$, where $k = 1, 2, 3, 4, 5, 6$:

$$b_{k,m,n} = \sum_x \sum_y \varphi_k[x, y]L_1[m, n, x, y]w[x, y], \tag{15}$$

(Step 3) Then we express the variation around $[m, n]$ by:

$$Q_{m,n}[x, y] = \sum_{k=1}^{6} b_{k,m,n}\varphi_k[x, y]$$

$$= p_{1,m,n}x^2 + p_{2,m,n}xy + p_{3,m,n}y^2 + p_{4,m,n}x + p_{5,m,n}y + p_{6,m,n}. \tag{16}$$

Note that $Q_{m,n}[x, y] \approx L_1[m, n, x, y]$ (defined in (5)). Then, we expand $Q_{m,n}[x, y]$ as a function of principal axes

$$Q_{m,n}[x, y],$$

$$= c_{1,m,n}X_{m,n}^2 + c_{2,m,n}Y_{m,n}^2 + c_{3,m,n}X_{m,n} + c_{4,m,n}Y_{m,n} + p_{6,m,n}, \tag{17}$$

where $X_{m,n} = x\cos\theta_{m,n} + y\sin\theta_{m,n}$, $Y_{m,n} = -x\sin\theta_{m,n} + y\cos\theta_{m,n}$, and the **principal axes** $[\cos\theta_{m,n}, \sin\theta_{m,n}]^T$, $[-\sin\theta_{m,n}, \cos\theta_{m,n}]^T$ are the eigenvectors of

$$\mathbf{H}_{m,n} = \begin{bmatrix} p_{1,m,n} & 0.5p_{2,m,n} \\ 0.5p_{2,m,n} & p_{3,m,n} \end{bmatrix}, \tag{18}$$

$c_{1,m,n}$ and $c_{2,m,n}$ are the corresponding eigenvalues, and

$$c_{3m,n} = p_{4m,n}\cos\theta_{m,n} + p_{5m,n}\sin\theta_{m,n},$$

$$c_{4m,n} = -p_{4m,n}\sin\theta_{m,n} + p_{5m,n}\cos\theta_{m,n}. \tag{19}$$

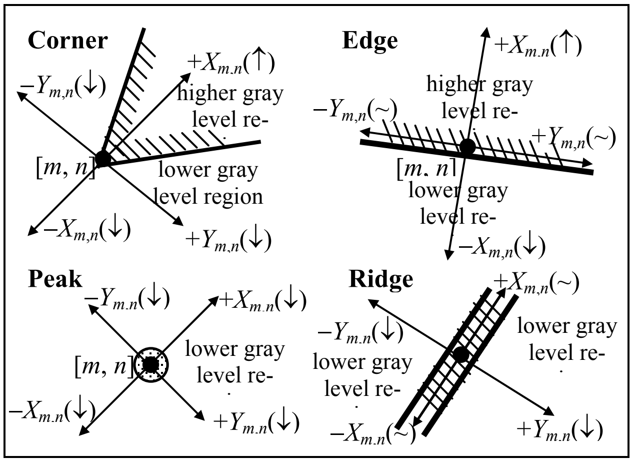


Fig. 1 Variations of gray levels along four principal directions for the pixel at a corner, on an edge, at a peak, and on a ridge.

Table 1 Case table.

v_1, v_2 / v_3, v_4	(↓, ↓)	(↓, ~) (~, ↓)	(↓, ↑) (↑, ↓)	(~, ~)	(~, ↑) (~, ↑)	(↑, ↑)
(↓, ↓)	Q_1	C	C	E_1	E_1	Q_3
(↓, ~) (~, ↓)	C	E	E	P	C_Y	E_2
(↓, ↑) (↑, ↓)	C	E	E	E	E	C
(~, ~)	E_1	P	E	P	P	E_2
(~, ↑) (↑, ~)	E_1	C_Y	E	P	E	C
(↑, ↑)	S	E_2	C	E_2	C	Q_2

C: corner, E: edge, E_1: ridge, E_2: valley,
Q_1: peak Q_2: dip, Q_3: saddle, C_Y: Y-corner, P: plain.

(Step 4) From (19), we can observe the variation along the two principal axes. For example, the variation along the positive part of $X_{m,n}$-axis (denoted by $+X_{m,n}$) is:

$$v_1 = \int_0^t (c_{1,m,n}X_{m,n}^2 + c_{3,m,n}X_{m,n})dX_{m,n}$$

$$= c_{1,m,n}t^3/3 + c_{3,m,n}t^2/2. \tag{20}$$

Similarly, the variations along the negative part of $X_{m,n}$-axis (denoted by $-X_{m,n}$) and along the positive and negative parts of $Y_{m,n}$-axis (denoted by $+Y_{m,n}$ and $-Y_{m,n}$) are:

$$\text{for } -X_{m,n}\text{:} \quad v_2 = c_{1,m,n}t^3/3 - c_{3,m,n}t^2/2. \tag{21}$$

$$\text{for } +Y_{m,n}\text{:} \quad v_3 = c_{2,m,n}t^3/3 + c_{4,m,n}t^2/2,$$

$$\text{for } -Y_{m,n}\text{:} \quad v_4 = c_{2,m,n}t^3/3 - c_{4,m,n}t^2/2. \tag{22}$$

We may choose t as 2~3. Then, we choose a threshold Δd

① $v_k > \Delta$ → denoted by ↑, ② $v_k < -\Delta$ → denoted by ↓,
③ $-\Delta \le v_k \le \Delta$ → denoted by ~. (23)

Harris' algorithm classifies the variation into only **3 types.** For our algorithm, there are four directions ($\pm X_{m,n}$, $\pm Y_{m,n}$) and for each direction there are 3 types of variations (↑, ↓, ~) . In sum, there are **3^4 = 81 types** of variations.

(Step 5) For the typical corner in Fig. 1, the variation along one axis is (↓, ↓) and along another axis is (↓, ↑). Thus, we conclude that if (v_1, v_2) = (↓,↓) and (v_3, v_4) = (↓,

↑), the pixel should be at the corner. If a pixel is on an edge, then the variation along one principal axis is (↓, ↑) and along another principal axis is (~, ~). We use Table 1 (Case table) to summarize the relations between the variation style and the most possible location of a pixel. With it, we can conclude the pixel is at a corner, on an edge, on a plane, on a ridge, in a valley, or at a peak.

(Step 6) In Step 5, we just obtain the **"corner candidates"**. Sometimes, the pixel that is not at a corner but near the corner will be recognized as a corner after Step 5. Thus we should use the **cornity** score to choose the corner from the corner candidates. We find that the most effective way is to define the cornity as **the integration of the quadratic polynomial $\boldsymbol{Q_{m,n}[x, y]}$** in a circular region:

$$\text{cornity} = \left|\int_0^R \int_0^{2\pi} Q_{m,n}[x, y] r dr d\theta\right|, \qquad (24)$$

where $X_{m,n} = r\cos\theta$, $Y_{m,n} = r\sin\theta$, and $Q_{m,n}[x, y]$ is defined in (16) and (17). Eq. (24) has an analytic solution:

$$\bullet\ \textbf{cornity} = \left|(c_{1,m,n} + c_{2,m,n})R^4\pi/4 + p_{6,m,n}R^2\pi\right|. \qquad (25)$$

Why is the cornity defined as the integration of $Q_{m,n}[x, y]$? Note that $Q_{m,n}[x,y]$ approximates the variation around $[m,n]$

$$Q_{m,n}[x, y] \approx |L[m+x, n+y] - L[m, n]|(L[m+x, n+y]-L[m, n]) \quad \text{when } x \text{ and } y \text{ are small.} \qquad (26)$$

If $[m, n]$ is on a plain ($L[m+x, n+y]-L[m, n]=0$) or an edge ($L[m+x, n+y]-L[m, n]=-(L[m-x, n-y]-L[m, n])$), then the integration of $Q_{m,n}(x, y)$ is near to zero. If $[m, n]$ is at a corner, then the integration of $Q_{m,n}[x, y]$ will be far from zero. Thus, $Q_{m,n}[x, y]$ is good for cornity measurement.

(Step 7) If a pixel $[m, n]$ is a corner candidate and

$$\text{cornity}[m, n] \geq \text{cornity}[m+\tau, n+\rho] \quad \text{for } \tau, \rho = -2\sim2. \qquad (27)$$

then $[m, n]$ is recognized as a corner.

4. EDGE DETECTION

Our algorithm can not only detect the corner but also detect the edge, the ridge, the valley, and the plain regions. To detect the edge, Steps 1~5 are the same as those of corner detection. In Step 6, we define the **edge score** as:

$$\textbf{E score} = \int_0^R \int_0^{2\pi} \left|T_{m,n}(x, y)\right| r dr d\theta, \qquad (28)$$

$$T_{m,n}[x, y] = \sum_{k=1}^{3} b_{k,m,n}\varphi_k[x, y] = t_{1,m,n}x + t_{2,m,n}y + t_{3,m,n}. \qquad (29)$$

E score also has an analytic solution. If $|\boldsymbol{t_{3,m,n}/a}| \leq R$,

$$\bullet\ \textbf{E score} = aR^3[\cos(3\varphi_0)/3 + \cos\varphi_0] + t_{3,m,n}R^2[-2\varphi_0 - \sin(2\varphi_0)], \qquad (30)$$

where $a^2 = t_{1,m,n}{}^2 + t_{2,m,n}{}^2$ and $\varphi_0 = \operatorname{asin}[-t_{3,m,n}/(aR)]$, $-\pi/2 \leq \varphi_0 \leq \pi/2$. If $|\boldsymbol{t_{3,m,n}/a}| > \boldsymbol{R}$, then

$$\bullet\ \textbf{E score} = |t_{3,m,n}|R^2\pi. \qquad (31)$$

For an "edge candidate" pixel $[m, n]$, if ① At least 17 adjective pixels satisfy:

$$\text{E score}[m+\tau, n+\rho] \leq \text{E score}[m, n] \quad \text{for } |\tau|, |\rho| \leq 2. \qquad (32)$$

② moreover, we classify the adjective pixels into four quadrants ($\tau>0$ & $\rho>0$, $\tau>0$ & $\rho<0$, $\tau<0$ & $\rho>0$, and $\tau<0$ & $\rho<0$) and for each quadrant, there should be at least one pixel that satisfies (33), then, we can treat $[m, n]$ as a pixel on an edge.

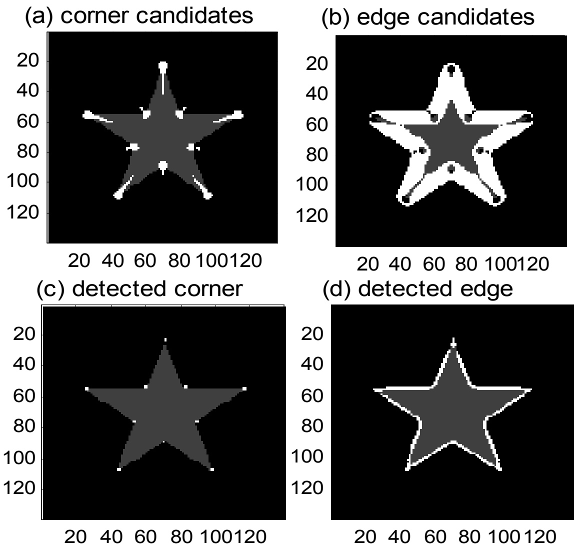


Fig. 2 Corner detection and edge detection for Star image.

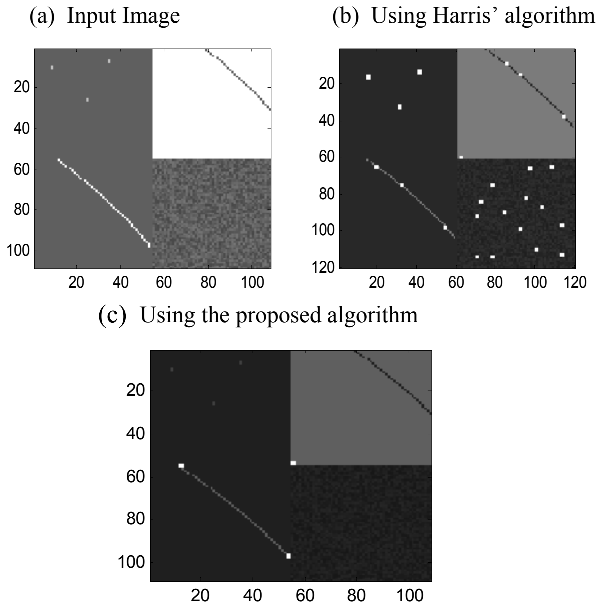
5. EXPERIMENTS

We first give an example in Fig. 2. We follow Steps 1~4 and use the Case table in Step 5 to find the corner and the edge candidates (shown in Figs. 2(a)(b)). Then, we use the "cornity" and "E score" and to select corners and edges from the corner and edge candidates. The results are shown in Fig. 2(c)(d), which show corners and edges are detected successfully.

Then we do some experiments to compare the performances of the proposed and the Harris' algorithm for the image in Fig. 3(a). When using Harris' algorithm, the results of corner detection is shown in Fig. 3(b). Note that some pixels at the dots, on the ridge, in the valley, or in a noise-interfered plain are treated as corners. From common senses, they should not be corners. However, since Harris' algorithm observes just the "amplitudes" of variations, and these pixels have large variations along two principle axes, they are misjudged as corners.

If we use the proposed algorithm, we can distinguish a corner from an isolated pixel, a belt valley, and a belt ridge according to the "sign" of variation. The result is shown in Fig. 3(c). Note that no non-corner pixel is misjudged as a corner. Especially, no pixel in the lower-right part (the noised-interfered plain) is regarded as a corner. It proves the theory in (9), i.e., the proposed algorithm is much more robust to noise.

In Fig. 4, we also do s experiment and show that the proposed algorithm can avoid regarding the Gaussian-like noise interfered region as a corner.


(a) Input Image

(b) Using Harris' algorithm

(c) Using the proposed algorithm

Fig. 3 (a) Image consists of three dots (upper-left), a valley (upper-right), a ridge (lower-left), and a noise-interfered region (lower-right). (b)(c) The results of corner detections.

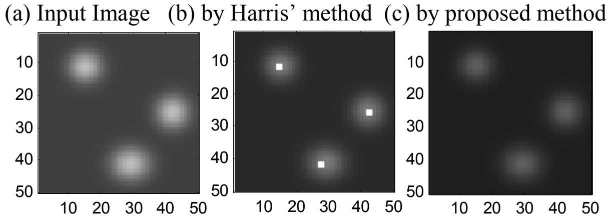

(a) Input Image (b) by Harris' method (c) by proposed method

Fig. 4 Corner detection results for the plain interfered by Gaussian-like noise.

We also do the experiment for a natural image in Figs. 5 and 6. It is obvious that, when using the proposed algorithm, the probability of misjudging the pixel in a valley, on a ridge, or at the isolated dots as a corner can be much reduced. Our algorithm can also detect most of the edges of Lena image successfully, as in Fig. 7.

6. CONCLUSIONS

We introduced an improved algorithm for corner and edge detections. For the proposed algorithm, since the "sign" of variations is considered, the difference among a corner, a belt ridge, and an isolated dot can be observed. Due to the same reason, our algorithm is more robust to noise, which was proven in Figs. 3, 4 and 6. We also apply orthogonal polynomial expansion, case table and the new definition of cornity to improve the performance.

7. REFERENCES

[1] C. Harris and M. Stephens, "A combined corner and edge detection", Proc. 4th Alvey Vision Conf., 1988, pp. 189-192.

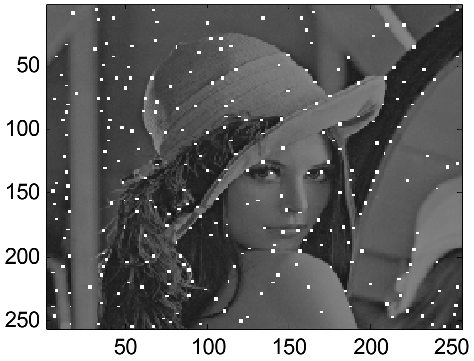


Fig. 5 Using Harris' algorithm to do corner detection

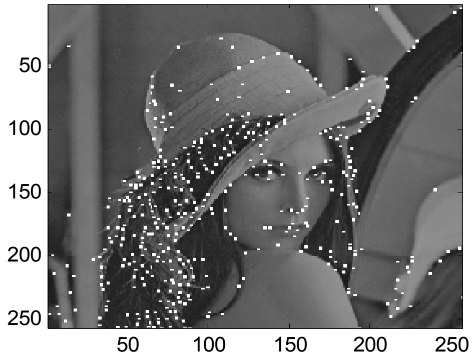


Fig. 6 Using the proposed algorithm to do corner detection.

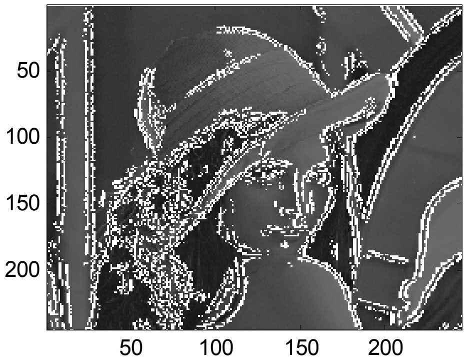


Fig. 7 Using the proposed algorithm to do edge detection.

[2] A. F. Nikiforov, S. K. Suslov, and V. B. Uvarov, *Classical Orthogonal Polynomials of a Discrete Variables*, Berlin; New York: Springer-Verlag, 1991.

[3] S. M. Smith and M. Brady, "SUSAN–a new approach to low level image processing", *Int. J. Computer Vision*, vol. 23, no. 1, 1997, pp. 45-78.

[4] H. Wang and M. Brady, "Real-time corner detection algorithm for motion estimation", *Image and Vision Computing*, vol. 13, no. 9, 1995, pp. 695-703.

[5] N. C. Overgaard, "On a Modification to the Harris Corner Detector", Symposium Svenska Sällskapet för Bildanalys, Stockholm, 2003.