Available online at www.ejournals.uofk.edu

UofKEJ Vol. 9 Issue 1 pp. 37- 46 (August 2019)

UNIVERSITY OF KHARTOUM ENGINEERING JOURNAL (UofKEJ)

Automatic Extraction of Corner Points from Aerial Images Using Point-Feature Operators and Hough Transform

Mohamed Eltahir Idris, Gamal H. Seedahmed

Department of Survey Engineering, Faculty of Engineering, University of Khartoum
Khartoum, Sudan (E-mail: Mohamed.idris@postgrad.uofk.edu, Gamal.Seedahmed@gmail.com)

Abstract: Low-level feature extraction such as lines and points (i.e. corners), forms a fundamental step in digital photogrammetry and other fields. They supply the inputs for the photogrammetric orientation procedures; and they serve as an intermediate input for other processes such as object recognition. With the accumulation of knowledge, the research community is in a better position to develop new generations of smart algorithms and solutions that possess a new level of maturity and understanding for the underlying challenges of automation. To this end, this paper presents an innovative approach for corner point extraction that combines the outputs from classical point feature operators with Hough Transform to generate a better hypothesis for a corner point that can be used for applications in urban areas. In particular, extracted point features were used to guide line extraction in a local neighbourhood by Hough Transform. Then the corner points that will be obtained from lines intersection in this local neighbourhood will be compared with their nearby ones that were extracted by point feature operators. Based on passing a set of criteria, the intersection points from lines will replace the point feature as a set of potential corner points. Experimental findings show promising results of the proposed approach that raises the confidence level of the extracted corners and eliminating outliers.

***Keywords**: Point-Feature Operators, Hough Transform, Corner points.*

1. INTRODUCTION

The main objective of digital photogrammetry is envisioned around automated processes [1]. An automatic process is typically defined as a process that involves little or no human intervention. Although higher order geometric primitives such as lines and conic sections are very key for the future progress in digital photogrammetry, point-feature, such as corner point, is very dominant in the current digital photogrammetric systems.

With the accumulation of knowledge, the research community is in a better position to develop new generations of smart algorithms and solutions that possess a new level of maturity and understanding for the underlying challenges of automation. As a demonstration of this maturity, this research is aiming to combine point-feature with Hough Transform (HT) to extract corner's point with high level of confidence. This level of confidence is trying to approximate the end results of human analysis and interpretation of a corner's point. In simple words, it delivers a corner point with semantic labelling using a data driven formulation. The proposed approach is fully automatic. In other words, it is an autonomous process since it does not require any manual intervention.

Literature is rich of researches that deal with point-like features detection and extraction [2]; and equally true there are abundant of research work in different aspects of Hough Transform (HT) to find straight lines in an image, and extending this usage of HT to find junctions of lines (intersections and corners) [3] [4] [5]. For example, [4] explains the finding of straight-lines in the image and then find their intersections. Also an application of the generalized Hough Transform in detecting corners in curved objects is proposed in [6]. Where in [7], a new approach that integrated a new vision-based image processing is proposed. In [8] extracted point features are used as matching entities to solve the correspondence problem in image matching.

In previous researches, corners were defined by the intersection of two lines [4]. It is a simple definition that includes a vast range of points in an image. Even the elimination of false corners was made with simple criteria; by assuming that most of false corners do not implement a corresponding edge point, which is not always the case. As shown in Figs. 1, 2, and 3, the corner point will be produced by the intersection of near-by object and not a true physical corner.

In aerial images of man-made features and urban areas, such as the images that were used in this research, different level of complexities will be encountered during the feature extraction process and the notion of cornerness will be greatly challenged.

As such, not every intersection in the image should be considered a corner point and the assumption that to eliminate any intersection with no edge point as used in [4] is not valid or general enough.

Harries operator [9] is used in the research since it is a simple and efficient operator that yields high quality point features, invariance to rotation, scale, illumination variation and image noise [7]. Upon these set of positive properties, Harries operator gains its popularity [10].

Harris operator adapts the basic idea of Moravec's operator [11], which states that feature points extraction are based on their intensity values. Points here can be defined as strong intensity values in a relatively small local neighbourhood.

Harries operator analyses the Eigenvalues of the autocorrelation function (structure tensor) to compute the differences between the intensity values locally in a moving window that moves in different directions. The existence of two strong Eigenvalues is an indicator of a corner point; otherwise there is no indication.

Fig. 1. Synthetic image

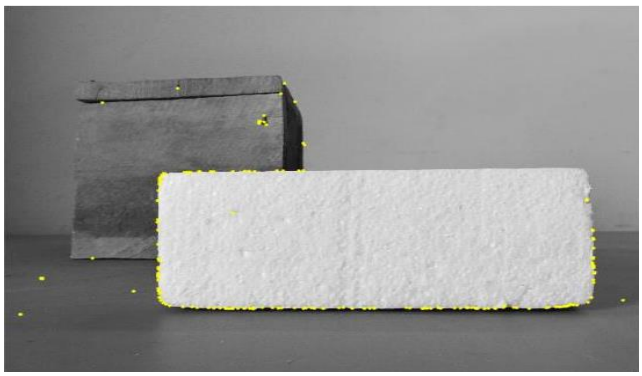

Fig. 2. Point features detected by Harris operator.

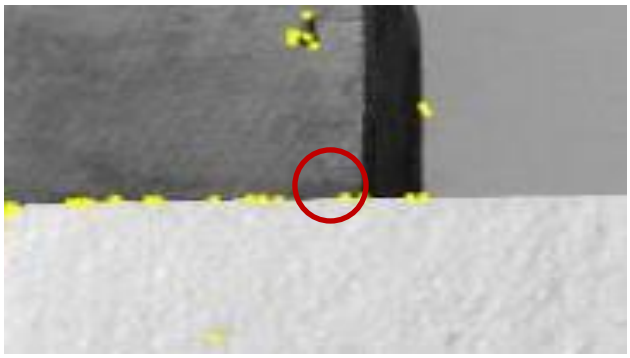

Fig. 3. Zoom in.

Generally, line intersections will probably fall into three cases, either a true corner intersection, or false intersection of one line intersect into another line and this is without being a location of a physical corner, or false intersection of two lines extension, which can be called virtual intersection. Fig. 4 shows all possible cases of lines intersection in a synthetic image. True corners were labelled with green colour and the false ones with red colour.

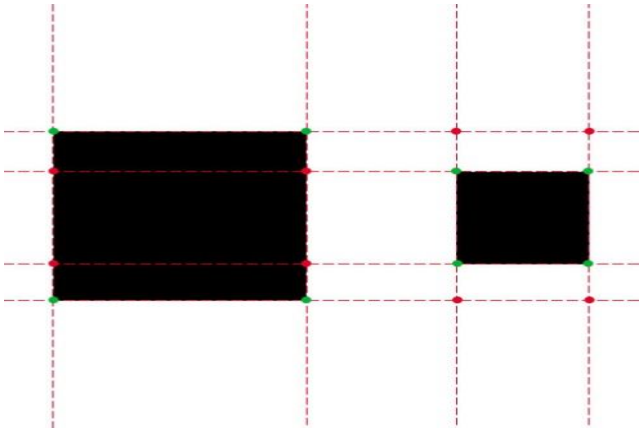

Fig. 4. Possible line intersections, Green dots are true corners, Red dots are false.

Those types of intersections could be considered corners using that simple definition mentioned earlier. But, adding more criteria to that definition as a corner is a feature point detected by one of the point feature operator and it falls within a specified threshold of two lines intersection detected by Hough Transform that will result in corners being detected with more confidence level.

In simple words, if a point is detected by a point operator and it's in a specified threshold distance from a two lines intersection; then there is a high possibility that this point is likely to be a true corner, which raises the confidence level of the extracted corners.

Although, the significant amount of research in corners extraction; the originality of this research is due to the fusion of two methods or techniques to increase the confidence level of the extracted corners.

Different algorithms set multiple criteria for corner detection, such as the mentioned by [4]:

a. The algorithm should detect all true corners. In fact, this criterion is hardily will be met in real images due to the huge complexities of the real world environment and its interaction with the imaging process. Therefore, new generation of algorithms are required to address these complexities and to approximate the reality.

b. The corner is well localized. This criteria calls for subpixel localization, which can be achieved by several approaches such as least squares solution.

c. Robust with respect to noise and outliers. Outliers in the context of feature extraction are typically refer to deficiency in the algorithms of feature extraction themselves.

d. Should be efficient. In other words, it should have an acceptable computational complexity.

e. Should not detect false corners. In reality, the goal will be to minimize the probability of false corners.

This set of criteria are well considered in the proposed algorithm.

Raising confidence level of detected corners is a process that starts with the stage of edge detection operations. It is an important task and is common in most of image processing and analysis algorithms [12]. It is applied as first stage to reduce the amount of information and the background noises before the use of Hough Transform or any other further processes.

It has been found that the robustness of Hough Transform (HT) performance and accordingly the proposed algorithm, is highly correlated with the edge detector operator that is being used. It has a great impact on Hough transform results. In [13] , a local edge detector is proposed to get better Hough transform results by finding edge points in a search technique of window-by-window locally.

This imperfection of edge detector operators cause the intersection point to lie in no edge point, also produce Geometrical Error (GE) higher than the specified threshold, causing some corner points to be neglected.

Using Hough Transform alone in detecting corners is not sufficient; one of the reasons is that it involves an extrapolating of the corner position [6]. In other words, virtual intersection will be encountered. Therefore, the proposed approach starts with point feature detection first, and then triggers a local Hough Transform around each detected point. The generated corner point should

resides in this local neighbourhood and within a specified distance tolerance. Hough Transform is a technique for detecting features in a particular shape in the image, which can be represented in parametric way [3]. Hough Transform forms the basis for most straight lines and corner detection techniques [6].

The computation starts by calculating the value of ρ at every possible θ (see Fig. 5), then accumulate all possible votes in Hough space using an "accumulator array".

The use of the polar representations:

$$\rho = Xcos\theta + Ysin\theta \quad (1)$$

ρ (the perpendicular distance of line from origin) and θ (the angle made by normal to this line with the positive direction about x-axis)

Although this method is time consuming [12] but it overcomes the issue of undefined slopes on vertical lines, and provide a constant resolution and bounded theoretical values for the angles, which is not the case for the use of m (slope) and b (y-intercept) in which the Hough space will have a large amount of possibilities [12]. And we can deal with the time consuming issue differently by using the zooming technique or local search for lines; which will be explained in more details later on.

A point in image space correspond a line (curve) in the parameter or Hough space. Thus, intersection points of lines in Hough space are lines in image space.

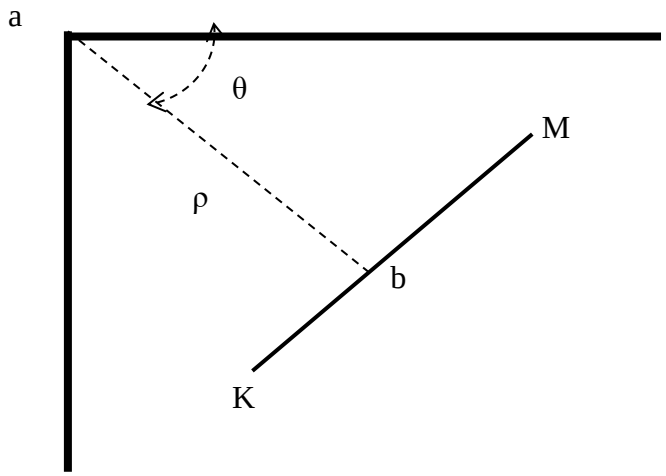

Fig. 5. ρ and θ the distance (magnitude) and direction of the normal (vector) a, b the normal to the line to be detected MK

This extraction of low-level features such as points, and corners detection has enormous applications in digital photogrammetry and mapping applications. As is well-known, feature extraction is the first step in image analysis and image matching [14]. In addition, it is very fundamental for object recognition and stereo vision [2]. Corners can be used in stereo-pairs matching for reconstruction of 3D scenes [5], in data compression, motion tracking [4], to estimate the position and rotation of objects in an image as it used in [3], to compute the rotation of a laser beam device, and in localizing the optic disc as used in [15] which is an example of applications in the medical field. In industrial field for safety and quality control where tests using algorithms can be accomplished with efficiently and low cost rates compared to hardware costs, as detecting corners can be used to detect defects and degree of bluntness in products [6].

This paper is organized as follows. Next section will present the material and methods, then followed by results and discussion in section 3. Conclusions and recommendations are presented in section 4.

2. MATERIALS AND METHODS

2.1 Materials

Comprehensive programing tasks using MATLAB programing language are conducted in order to develop an algorithm that works functionally with efficient results for detecting corners with high confidant levels.

The aerial images that are used are acquired from SUDAN EYE project, which is a low cost digital aerial camera system that's used in aerial mapping, the images cover the area of old Omdurman, Sudan. A permission of use from the copyrighter is granted. Experiments will be made on synthetics and the real images, to evaluate the algorithm

2.2 The Algorithm

The proposed approach or algorithm automatically detects corner points in an aerial image using a point feature operator and Hough transform. The detection process is a point-feature guided extraction. So, as a first step, point features in the image are extracted using a feature point operator (Harris), then a window (of known and pre-specified dimensions) is formed around each detected point feature (a zooming technique). Each detected point feature is the centre of each window.

After that, Hough Transform is used to detect all lines (group of lines) in each window; these lines are detected at a specific line length threshold (LLT).

The feature point of the window (centre of the window) is tested for satisfying each line equation of the Hough extracted group of lines, and Geometric Error (GE) is computed. A minimum set of two lines will be tested if they satisfy a threshold of GE. Theoretically the optimal value for the GE is zero, which implies that the tested point is an ideal corner. Mathematically, the GE is defined by the following equation:

$$GE = (Xcos\theta + Ysin\theta) - \rho \quad \textbf{(2)}$$

In case of only two lines are detected in the window space, they are solved simultaneously to find the corresponding intersection point (the mathematical representation of the corner point). If more than two lines are detected in the window, a combination is made to group all possible two lines together, and then each two lines are solved simultaneously as shown in equation 2 in the previous step. The simultaneous solution of two line equations in ρ and θ is formed by substituting each line's ρ and θ in the following equation, where X_C and Y_C are the corner coordinates:

$$\rho_1 = X_c cos\theta_1 + Y_c sin\theta_1 \quad \textbf{(3)}$$

$$\rho_2 = X_c cos\theta_2 + Y_c sin\theta_2 \quad \textbf{(4)}$$

It should be noted that these two eqautions 3, and 4 may not indicate a true physical corner as shown in Fig. 4. They just generate a hypothesis for a potential corner point that will be validated by a distance parameter (see equation 6). These equatios are expressed in the following set of matrices for easier solution:

$$\begin{bmatrix} X_c \\ Y_c \end{bmatrix} = \begin{bmatrix} cos\theta_1 & sin\theta_1 \\ cos\theta_2 & sin\theta_2 \end{bmatrix}^{-1} \begin{bmatrix} \rho_1 \\ \rho_2 \end{bmatrix} \quad \textbf{(5)}$$

In case of vertical lines where θ = 90°, an adjustment is made where x =ρ.

Parallelism is avoided by computing the angular difference between the two tested lines. This avoidance is implemented by a setting an angular range for the difference between the orientation of the two lines (10-135 degrees), and also by avoiding cases where $\theta_1 = \theta_2$. In other words, parallel lines are excluded from

the intersection process. It implies that there is no intersection point to be found in the selected neighbourhood.

Furthermore, the intersection point in the window is compared to the corresponding point feature detected by the operator (Harris). In the context of this research, the feature point can be called the image based physical point and the one that will be obtained by the intersection is called the mathematical point since it may not be accurately detected in the image due to several reasons such as the rounding or corner's effect [1]. In particular, the distance (***D***) from each intersection or mathematical point, which will be obtained from two lines or more, to the point feature or physical point that will be extracted by Harries operator, which will park on the centre of the window, is computed as follows:

$$D = \sqrt{\Delta X^2 + \Delta Y^2} \quad \textbf{(6)}$$

This distance parameter (***D***) is very critical in determining the final hypothesis for a corner point. Large distances undermine the corner's hypothesis; and shorter distances increase the likelihood for a corner point. In other words, the distance parameter is acting as a cost function for corner point identification. In the proposed algorithm, the computation efforts are confined to a small neighbourhood by performing Hough Transform in a relatively small window rather than the whole image where many lines will be extracted.

The zooming technique or local window search provides a high speed of Hough Transform and optimize the overall performance of the proposed approach. In a way that is fast and doesn't consume memory resources, which is a main concern while using Hough Transform. In other words, it combats the combinatorial nature of Hough Transform. A significant number of algorithm has been proposed to enhance and obtain fast results [12].

Fig. 6 shows the workflow for the major steps of the proposed approach. This approach can handle the intersection of more than two lines.

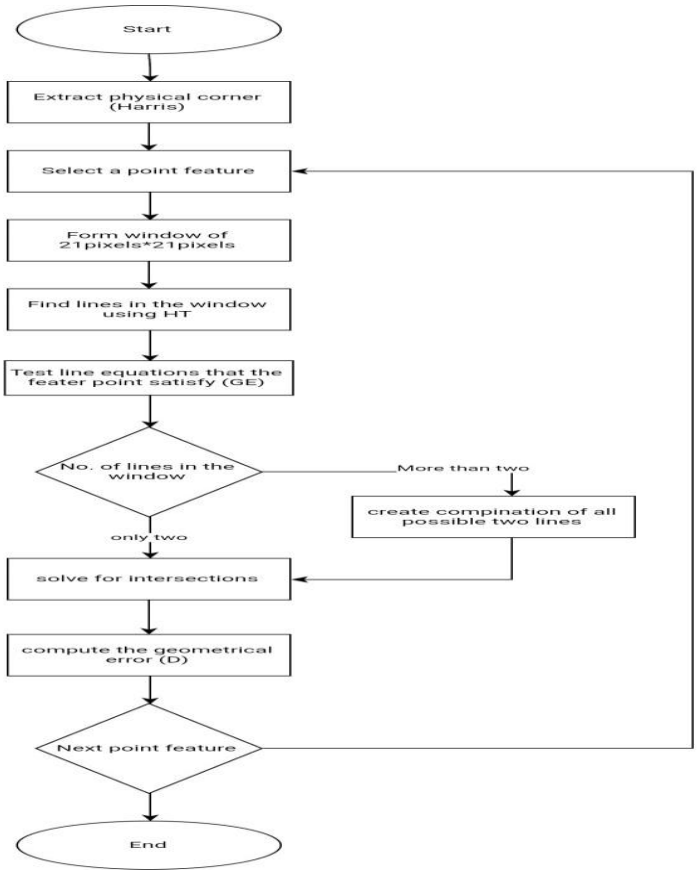

Fig. 6. Workflow of the proposed approach.

3. RESULTS AND DISCUSSION

3.1. Testing on Synthetic Image

A synthetic image with a size of 1000x1000 pixels is used as a controlled experiment to test the obvious performance of the proposed approach (see Fig. 7). This image contains 31 corner points. This number does not include the 5 corners on the boarder of the image. Fig. 8 shows the results of edge detection, which were used for lines extraction by Hough Transform.

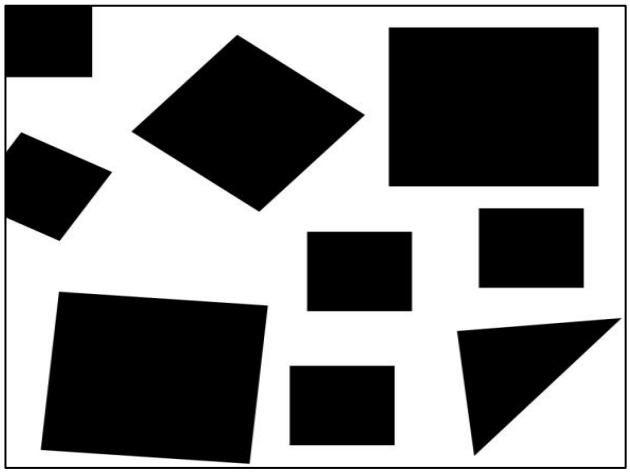

Fig. 7. The synthetic image.

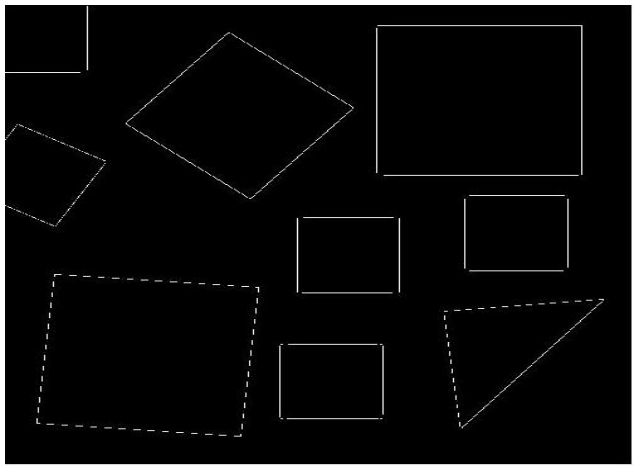

Fig. 8. Example of an edge image (Canny operator).

For the synthetic image presented in Fig. 7; the proposed approach starts with point feature extraction (Harries operator), then followed by the computation of the geometrical error (GE) with a threshold of 2 pixels. A line length threshold (LLT) of 5 pixels is used to define the minimum length, and a setting of a threshold for the distance (***D***) between Harris detected feature point and Hough Transform intersected point of 2 pixels. The local computation of the Hough Transform is performed on a window with a size of (21 x 21 pixels) for each point. The average time of 100 processes was 8.592 seconds (based on the specification of the used hardware).

The detectability in this image was 96.77%; and by physical inspections (see Fig. 9 and Table 1 and 2) all corners were detected and well localized and only one true corner point produced a distance (***D***) above the specified threshold of 2.115 (check the green circle in Fig. 9), which was eliminated. In fact, there are 15 points that have ***D*** values of greater than 1 pixel and less than 2 pixels. 10 of these points belong to rotated objects as shown in Fig. 9, which suggests that there are problems due to the discrete nature of Hough Transform and pixelization effects due to the discrete nature of the image space. The other 5 points that

have ***D*** values that are greater than 1 pixel belong to non-rotated objects, which suggests that there is a detection problem by the edge detection algorithm.

Table 1. Result of Harris operator

N	Harris	
	X	Y
1	596	45
2	921	45
3	469	464
4	898	578
5	604	901
6	134	146
7	604	739
8	736	416
9	736	578
10	898	416
11	921	370
12	596	370
13	442	901
14	631	464
15	442	739
16	469	626
17	631	626
18	197	258
19	557	224
20	394	421
21	360	61
22	25	261
23	164	342
24	83	481
25	729	921
26	56	910
27	84	588
28	406	616
29	378	938
30	703	668
31	956	641

Table 2. Result of the algorithm

N	HT		|D|
	X	Y	
1	596	44.910	0.090
2	921	44.854	0.146
3	469	463.910	0.090
4	898	576.904	1.096
5	604	899.904	1.096
6	133.109	145.158	1.226
7	603.837	738.398	0.623
8	736	416.105	0.105
9	736.868	578.008	0.868
10	898	415.634	0.366
11	920.109	369.158	1.226
12	596.868	370.008	0.868
13	442.621	900.837	0.642
14	631	463.080	0.920
15	441.306	738.306	0.981
16	470.021	625.587	1.101
17	631	624.904	1.096
18	195.954	256.954	1.479
19	556.637	222.576	1.470
20	394.594	421.107	0.604
21	359.372	59.408	1.711
22	24.191	259.855	1.403
23	164.935	340.504	1.764
24	83.597	481.635	0.871
25	728.594	922.429	1.486
26	55	910.726	1.236
27	407	614.533	1.775
28	378.412	938.047	0.415
29	702.158	667.109	1.227
30	956.611	641.054	0.613

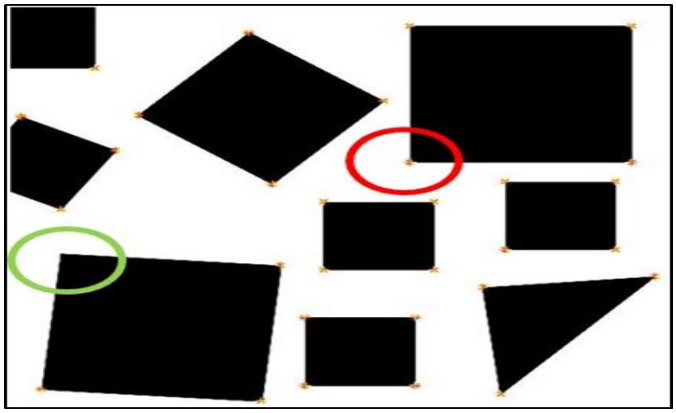

Fig. 9 **All detected corners.**

As we can see in the previous tables and figures, all corners are detected and there's a confirmation between the mathematical corner (Hough Transform detected feature) and the physical corner (Harris detected point feature). Fig. 10 shows a zoom-in around the red-circle shown in Fig. 9. The distance between the Hough Transform point and the Harries point is 0.868 pixel.

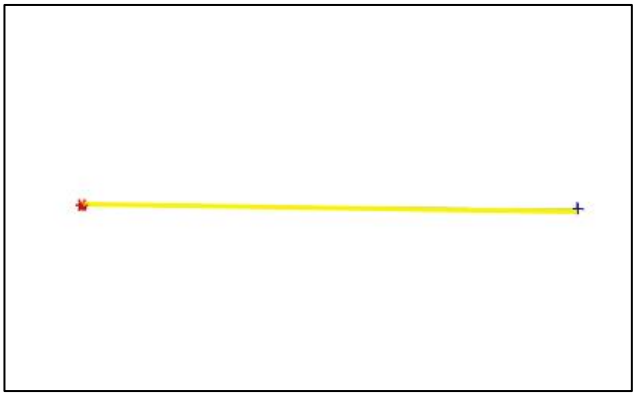

Fig. 10. Zoom in, Harris (Red), Hough (Blue)

As described earlier, a window is formed around each feature point detected by Harris operator, group of lines are detected and the feature point is tested for satisfying their line equations. -the following three results will be used to explain the internal computations of the proposed approach. .

For the first window shown in Fig. 11: two lines were detected in the Hough space (see Fig. 12), the two lines is perfectly vertical with an angular difference of 90º between the two lines, the two lines intersect near the center of the window (i.e. Harries detected point feature) which produced a very small distance (D=0.09),.

Fig. 11. Edge image with Hough detected lines and the intersection point

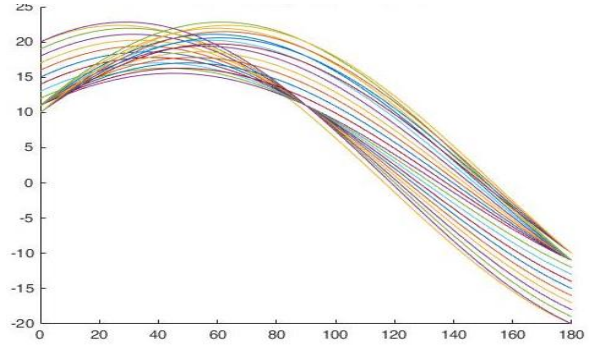

Fig. 12 Hough space for the first window

Fig. 12 shows two peaks that point to two lines in the image space.

In the second window shown in Fig. 13, three-lines were detected in the Hough space (see Fig. 14), but only two lines were within the specified line length threshold (5 pixels) were found. The third line is eliminated.

Fig. 13. Edge image with Hough detected lines and the intersection point

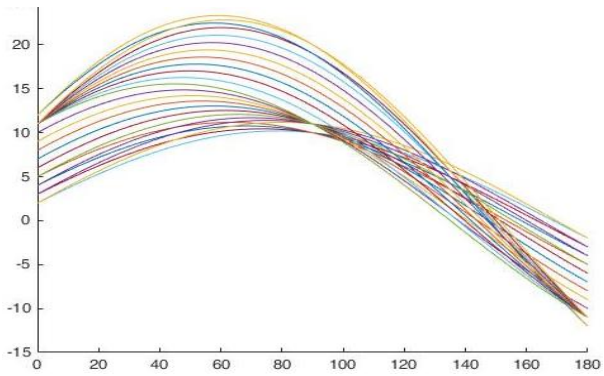

Fig. 14. Hough space for the second window

Only two lines were detected in the third window space (see Fig. 15) and the Hough space (see Fig. 16), the point feature detected by Harris operator perfectly fit both of the two lines.

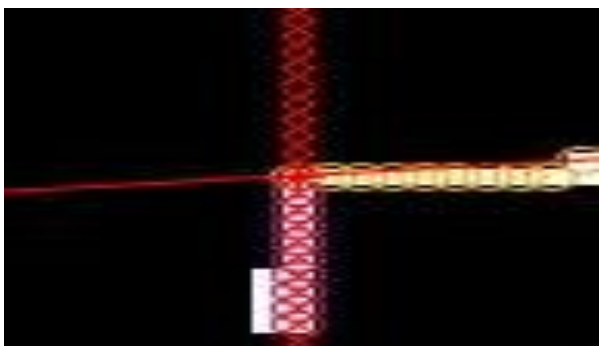

Fig. 15 E edge image with Hough detected lines and the intersection point

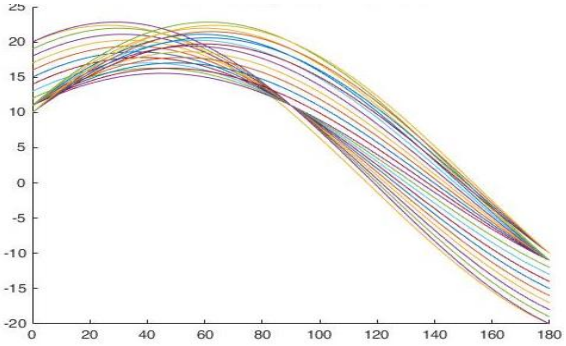

Fig. 16. Hough space for the third window

The rest of the windows were processed in the same manner, and produced similar results in the Hough space.

Another experiment using the same synthetic image only with added noise to test for the robustness of the algorithm.

The experiment was carried out based on the same settings used earlier except that a Gaussian noise with amount of 25% is added to the image.

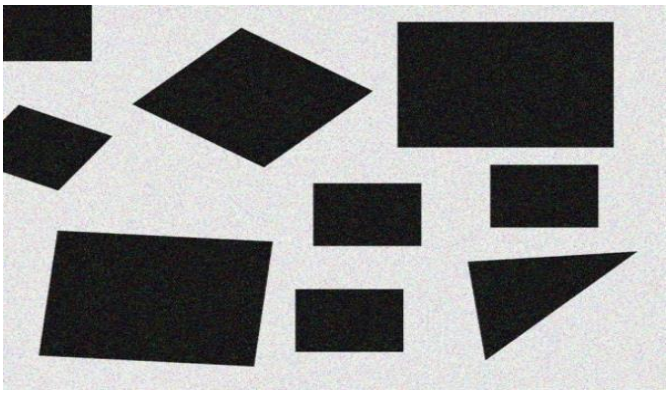

Fig. 17. The same synthatic image with added noise.

Harris operator detected 200 point features (see Fig. 18), while the proposed algorithm detected 31 point features (see Fig. 19), 27 of them were true corner points based on the first experiment, the other 4 feature points are false corners misguided by the Harris operator.

Table 3. detected corners by the proposed algorithm:

N	HT		|D|
	X	Y	
1	596.621	369.837	0.642
2	469	462.411	1.589
3	359.372	59.408	1.711
4	728.798	922.167	1.184
5	898	576.476	1.524
6	134	145.572	0.428
7	920.247	368.989	0.753
8	23.995	260.387	0.613
9	84.113	481.471	1.208
10	921.26	45.0232	0.26
11	164.935	340.504	1.764
12	394.377	421.427	0.569
13	956.614	639.657	1.477
14	737.592	415.028	1.138
15	631	626.879	1.879
16	595.302	44.369	0.941
17	195.621	257.365	1.518
18	631.490	464.040	1.078
19	557.403	223.369	0.750
20	603.144	901.664	0.680
21	468	626.171	1.015
22	55	910.078	0.078
23	702	667.086	0.914

24	898.155	415.222	0.793
25	407	614.533	1.775
26	378.768	937.829	0.787
27	373.116	611.433	0.579
28	871.474	578.042	0.476
29	513.474	463.038	1.072
30	201.875	262.570	1.204
31	463.529	900.959	0.472

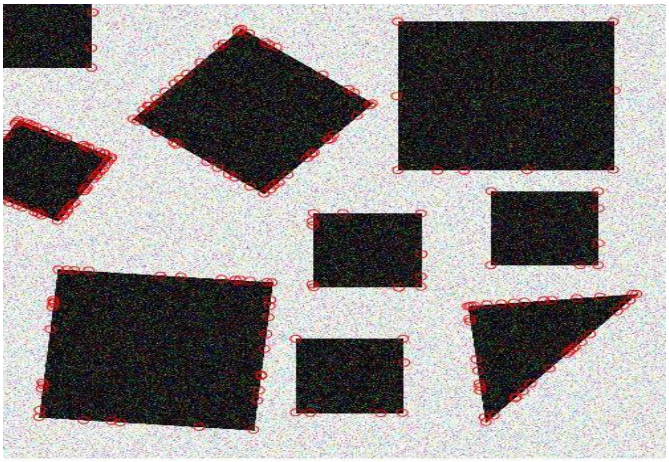

Fig. 18. The result of Harries operator.

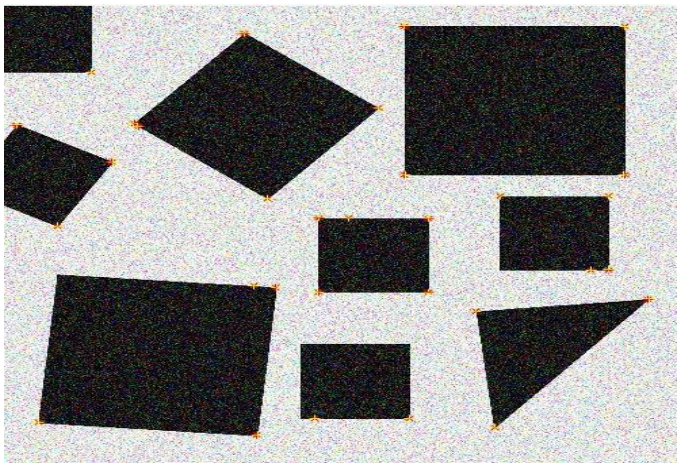

Fig. 19. Results of the proposed algorithm.

The result of Harris operator contains outliers due to the noise effect. As such, Harris operator is very sensitive to this amount of noise. On the other hand, the proposed approach performs very well with the same level of noise. In particular, the proposed algorithm increased the confidence level of the extracted corner points by eliminating outliers which are about 84.5% of Harris operator results. It detected 87.1% of the true corners extracted in the previous experiment.

This experiment showed the robustness of the proposed algorithm to the present amount noise, and raised the confidence level of the extracted corner points.

3.2. Tests on Real Aerial Images

The proposed approach will be tested on three image patches that were extracted from SUDANEYE imaging project.

The experiment was carefully carried with the following settings: the geometrical error (GE) is set to 2 pixels, a line length threshold (LLT) of 5 pixels, a threshold for the distance between Harris detected feature point and Hough intersection (D) of 2 pixels, and the window size of 21 x 21 pixels.

Harris operator detected 200 feature points (see Figs. 20 and 21 as well as Table 4), the proposed algorithm detected 105 corner points, which is about 52.5% of Harris's results that indicate 47.5% were eliminated.

Table 4. Result of the algorithm:

N	HT		|D|
	X	Y	
1	957.033	948.689	1.629
2	734.512	734.565	1.55
3	634.928	829.664	1.957
4	721.839	745.839	0.227
5	611.046	681.11	1.051
6	892.7	998.855	0.906
7	546.763	841.021	1.763
8	818.528	713.339	1.665
9	362.726	239.011	1.626
10	325.668	844.769	0.837
11	495.978	1000.3	1.201
12	703.038	978.248	0.993
13	699.616	978.126	1.621
14	509.736	882.031	1.064
15	637.334	760.2	1.556
16	883	985.873	1.008
17	882.58	989.475	1.665
18	700.755	727.982	1.047
19	514.023	894.257	0.258
20	574.084	843.076	1.079
21	607.484	902.858	1.24
22	447.159	1003.14	0.878
23	631.451	821.834	1.25
24	466.478	905.542	1.59
25	438.533	913.936	1.046
26	500.403	837.839	0.619
27	424.324	998.822	0.699
28	820.039	999.855	0.971
29	795.335	616.394	0.517
30	783.238	702.883	1.521
31	252.324	615.822	0.699
32	950.682	997.671	0.957
33	793.845	608.827	0.863
34	491.143	932.65	0.666
35	510.654	885.608	1.936
36	636.657	884.44	1.693
37	808.014	803.014	0.02
38	900.98	995.846	0.992
39	699.326	1012.136	1.982
40	513.403	909.12	1.657
41	859.326	838.326	0.46
42	626.357	187.38	0.893
43	313.582	818.095	1.584
44	369.885	910.918	1.554
45	512.476	1000.02	1.77
46	479.421	971.171	0.93
47	854.091	893.293	1.13
48	648.189	754.785	0.808
49	866.782	1016.174	0.801
50	704.922	1030.239	1.242
51	439.692	992.566	1.712
52	806.004	731.866	1.326
53	774.117	644.106	0.901
54	665.333	967.295	1.457
55	865.637	992.206	1.819
56	503.356	1002.828	1.337
57	511.075	902.547	1.455
58	390.561	900.253	1.621
59	439.261	1003.261	1.045
60	999.817	777.193	1.148
61	664.821	740.397	1.019
62	793.505	710.825	1.279
63	934.521	677.259	1.806
64	346.012	839.711	1.029
65	521.352	493.325	1.39
66	495.691	939.019	1.2
67	848.545	869.165	1.678
68	830.925	1029.473	1.475
69	494.85	843.325	1.196
70	815.19	723.437	0.477
71	873.414	999.945	1.207
72	236.077	561.747	1.556
73	473.142	911.978	1.983
74	625.678	867.728	1.442
75	448.109	988.286	1.146
76	764.632	894.641	1.753

77	596.736	805.152	1.742
78	426.237	871.877	1.891
79	804.75	799.968	1.062
80	569.538	835.768	0.517
81	806.538	805.261	1.638
82	612.476	909.152	0.499
83	148.368	875.181	0.898
84	736.765	31.854	0.779
85	118.287	894.378	0.807
86	308.36	777.264	0.975
87	562.859	812.756	1.252
88	436.289	879.054	0.989
89	71.17	1015.959	0.175
90	299.541	823.417	0.621
91	148.721	340.1	1.92
92	273.121	534.264	1.146
93	595.215	881.581	1.622
94	594.932	874.903	1.399
95	761.09	948.923	1.926
96	785.411	189.052	1.59
97	483.289	1016.054	0.989
98	285.86	559.86	0.199
99	511.529	852.338	1.666
100	823.711	719.813	1.221
101	336.826	841.39	1.925
102	417.318	966.17	0.889
103	792.122	734.662	1.1
104	679.339	943.219	1.265
105	847.829	870.098	0.918

Fig. 20. Image (KN_2) from SUDAN EYE.

Fig. 21. Results of the propoed algorithm

In the second experiment with a real aerial image (see Figs. 22 and 23 as well as Table 5), Harris operator detected 200 feature points, the proposed algorithm detected 94 corner points, which is about 47% of Harris's result that indicate 53% were eliminated.

Table 5 **Result of the algorithm**

N	HT		|D|
	X	Y	
1	33.577	499.356	1.704
2	275.611	2.371	0.74
3	60.842	499.593	0.935
4	420.227	343.255	1.275
5	547.465	56.756	1.354
6	454.88	499.626	1.379
7	558	463.883	1.117
8	132.051	239.252	0.257
9	69.458	498.317	0.822
10	320.276	87.767	0.361
11	600.249	198.306	1.286
12	509.896	44.522	0.489
13	428.894	499.662	1.289
14	191.243	449.825	1.941
15	390.616	454.686	0.922
16	162	272.623	0.377
17	140.556	261.06	1.197
18	357.726	375.585	0.498
19	547.015	500.078	0.922
20	117.555	266.073	1.161
21	127.324	90.283	1.505
22	1.169	499.652	0.901
23	74.663	107.68	0.95
24	256.146	347.987	1.325
25	86.041	3.092	1.045
26	539.027	206.622	0.379
27	585.362	434.574	1.972
28	598.183	468.652	1.045
29	42.754	19.705	1.318
30	403.187	96.833	1.843
31	39.462	234.091	0.545
32	28.197	286.832	2
33	143.532	324.762	1.324
34	499.729	454.086	1.933
35	285.201	483.206	1.219
36	106.693	288.73	1.864
37	309.816	64.048	1.52
38	43.265	27.46	1.346
39	506.88	499.28	0.924
40	494.795	455.551	0.494
41	116.995	11.088	1.088
42	518.732	430.546	0.527
43	120.735	61.502	1.668
44	531.801	176.426	1.586
45	132.951	312.845	0.163
46	432.024	170.463	1.08
47	577.699	126.992	1.227
48	117.424	92.589	1.524
49	107.514	248.448	1.626
50	367.038	100.944	0.068
51	304.677	62.677	0.457
52	151.476	342.024	1.77
53	456.654	165.547	1.593
54	518.222	470.416	1.599
55	329.596	289.646	0.762
56	301.318	28.905	1.14
57	73.617	113.917	1.105
58	493.579	485.556	1.64
59	366.487	461.149	1.713
60	428.66	292.67	1.79
61	55.757	147.685	1.337
62	337.346	111.186	1.359
63	192.074	207.416	1.095
64	261.625	344.959	1.376
65	246.711	37.681	0.984
66	152.135	47.469	1.228
67	124.104	239.217	0.79
68	594.127	465.975	0.129
69	620.757	0.175	1.842
70	323.509	2.962	0.493
71	205.052	479.592	0.595
72	77.753	2.577	0.49
73	142.839	322.839	1.186
74	340.438	327.477	1.624
75	368.289	372.46	1.567
76	358.296	103.265	1.856
77	166.438	375.775	1.578
78	142.163	269.852	1.859

79	566.808	464.318	1.373
80	238.478	423.754	0.917
81	343.305	392.914	0.316
82	545.385	459.959	0.616
83	232.218	41.218	0.308
84	357.446	386.926	1.801
85	310.317	271.046	1.005
86	400.782	3.069	0.785
87	291.259	481.838	1.512
88	108.329	13.329	0.466
89	237.87	41.758	0.903
90	198.449	266.873	1.032
91	144.3	253.769	1.79
92	196.866	449.203	0.89
93	206.764	474.833	1.772
94	239.791	35.436	1.976

Fig. 22. Image (BMO_1) from SUDAN EYE.

Fig. 23. Results of the propoed algorithm

In the third experiment, Harris operator detected 165 feature points and the proposed algorithm detected 51 corner points (see Figs. 24, 25, and 26 as well as Table 6), which is about 30.9% of Harris's results that indicate 69.1% were eliminated.

Table 6 Result of the algorithm:

N	HT		|D|
	X	Y	
1	506.87	227.89	1.585
2	72	243.44	0.559
3	470.19	82.451	1.561
4	383.58	306.2	1.848
5	519.95	228.6	1.599
6	573.79	396.34	1.678
7	492.66	371.51	1.737
8	76.311	481.89	0.332
9	393.65	294.59	1.699
10	564.75	379.08	1.938
11	573	392.93	1.468
12	257.93	343.66	1.957
13	304.08	324.84	1.928
14	54.134	252.08	1.563
15	335.03	375.77	1.981
16	303.41	353.24	1.302
17	503.21	214.2	1.129
18	553.80	345.59	0.455
19	396.09	302.54	1.183
20	223.67	476.03	1.182
21	221.64	469.59	1.699
22	297.02	336.23	0.77
23	438.58	285.28	1.914
24	583.63	295.43	1.479
25	572	300.99	0.993
26	449.77	27.68	1.341
27	595.39	280.06	1.122
28	396.56	303.24	1.316
29	259.879	350.927	1.387
30	395.22	299.729	1.424
31	30.732	259.59	1.831
32	64.689	506.278	0.998
33	45.525	282.525	0.743
34	572.648	311.979	0.649
35	578.855	281.191	1.403
36	586.7	319.785	0.369
37	444.257	23.365	1.781
38	1.121	254.14	0.869
39	461.009	270.073	1.012
40	543.809	405.058	1.519
41	583.665	277.665	1.887
42	592.751	286.537	1.359
43	28.754	457.348	1.294
44	408.798	343.773	1.957
45	561.009	292.009	1.402
46	587.648	367.359	1.687
47	243.702	54.926	0.973
48	212.751	499.047	1.968
49	114.673	239.662	1.885
50	58	483.43	1.43
51	369.833	325.664	1.343

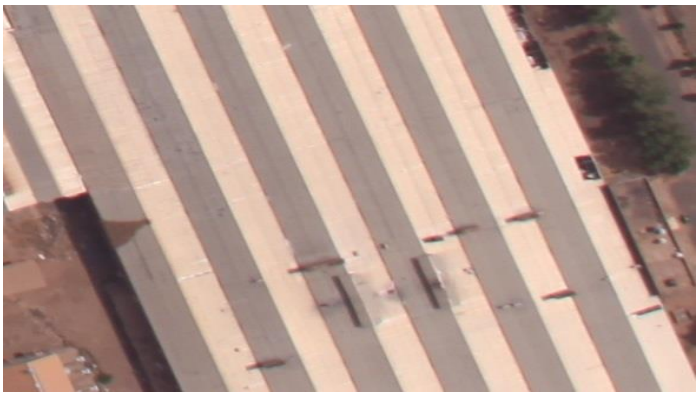

Fig. 24. Image (KN_1) from SUDAN EYE.

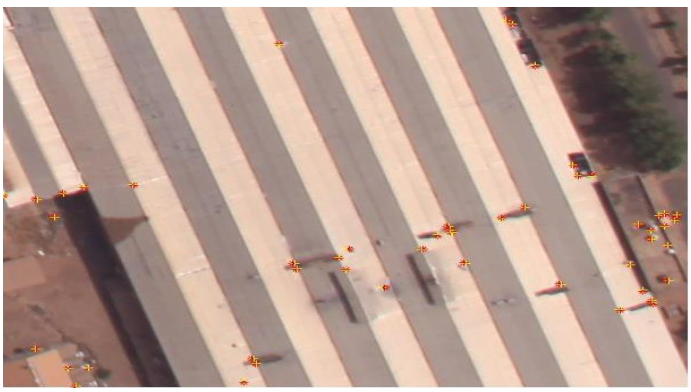

Fig. 25. Results of the propoed algorithm

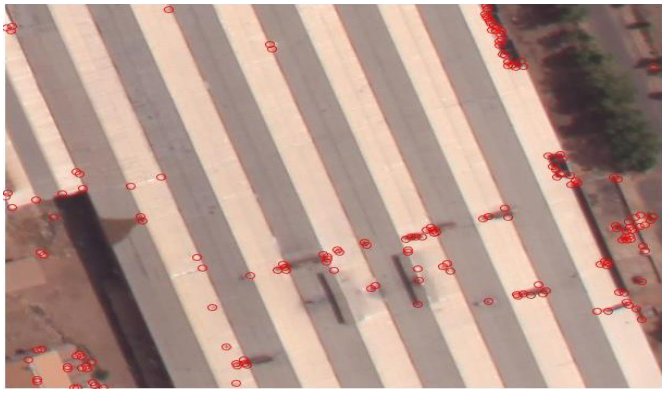

Fig. 26. Results of Harris operator

This image consist mostly of parallel lines so that most of Harris detected point features were eliminated as no intersections of Hough Transform detected lines were found. It should be noted that the Harris operator missed a great number of true corners and indeed they were not detected by the proposed approach since its operation depends on the feature point for widow selection. Future work should address this issue.

4. CONCLUSSIONS & RECOMMENDATIONS

An approach is proposed that raises the confidence level of the extracted corners from aerial images. Overall this approach is efficient, robust, reliable, and fast. It does not require high memory space, and can be adapted in enormous applications. In particular, it exploits the benefits of two worlds, namely, the world of point and the world of line features. Guided line extraction by point feature combats the combinatorial nature of the Hough Transform and minimizes the possibility of getting false corners that could be obtained from the intersections of lines extension and lines crossing.

The proposed approach can be extended in several directions. For example, least squares solution for line fitting can be applied to the set of points that were extracted by the Hough Transform to counteract the pixelization and the discretization effects. In addition, the extracted corners can be augmented by subpixel localization. Moreover, the proposed approach can be used to evaluate the property of cornerness of different point-feature extraction algorithms by comparing them to the intersection point that will be obtained from two lines. From shape recognition point of view, the proposed approach can be used to generate hypothesises for polygons from images and scanned cadastral maps.

.5. ACKNOWLEDGMENT

Great Thanks first to Allah, then to my parents, brothers and sisters, fiancé, colleagues, teachers, special thanks and gratitude to my supervisor, and surveying engineering department - University of Khartoum.

REFERENCES

[1] Schenk, T." Digital Photogrammetry". Terra Science. 1999.

[2] V. Rodehorst and A. F. Koschan, "Comparison and Evaluation of Feature Point Detectors," October 2006.

[3] A. Awwal, "Hough Transform Based Corner Detection For Laser Beam Positioning," 29 July 2005.

[4] S. Kang, Y. Choung and J.-A. Park, "Image Corner Detection Using Hough Transform," pp. 279-286, June 2005.

[5] W. Barrett and K. Petersen, "Houghing The Hough: Peak Collection For Detection of Corners, Junctions and Line Intersections.," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001.

[6] E. Davies, MA, DPhil, CPhys and FlnstP, "Application of the generalised Hough transform to corner detection," *IEE PROCEEDINGS, voi. 135, pt. E, NO. 1,*, pp. 49-54, January 1988.

[7] K. Patoommakesorn, F. Vignat and F. Villeneuve, "A new Straight Line Matching Technique by Integration of Vision-based image processing," *Procedia CIRP*, pp. 777-782, 2016.

[8] G. H. Seedahmed, "Automatic Matching of Close-Range Video Images Using Parameter Space Clustering," pp. Vol 3, No 2, Augsut 2013.

[9] C. Harris and M. Stephens, "A Combined Corner And Edge Detector," *In Proc. of Fourth Alvey Vision Conference*, pp. 147--151, 1988.

[10] J. Sanchez, N. Monzon and A. Salgado, "An Analysis and Implementation of the Harris Corner Detector," *Image Processing on Line*, pp. 305-328, 2018.

[11] H. P. Moravec, "Towards automatic visual obstacle avoidance," *Proceeding IJCAI'77 Proceedings of the 5th international joint conference on Artificial intelligence - Volume 2*, pp. 584 - 584, 22 - 25 August 1977.

[12] C. Singh and N. Bhatia, "A fast Decision Technique for Hierarchical Hough Transfrom For Line Detection," 2010.

[13] F. Wang, "A Local Edge Detector Used For finding Corners," 2001.

[14] W. Forstner and E. Gulch, "A fast Operation For Detection and Precise Location of distnct Points, Corners and Centers of Circular Features," *Proc. ISPRS intercommission conference on fast processing of photogrammetric data*, pp. 281-305, 2 June 1987.

[15] B. Gui, R.-J. Shuai and P. Chen, "Optic Disc Localization Algorithm Based on Improved Corner Detector," *Procedia Computer Science. 131*, pp. 311-319, 2018.