Comparison and contrast of various feature detection Algorithms

Phillip Garrad - S00193938

*Institute of Technology, Sligo*

Abstract

This paper will review the following different feature detection algorithms; Shi-Tomasi, Harris-Stephens and Triggs. It will then review the advantages and disadvantages of each method. This paper will show experimentation and highlight any issues faced.

**Keywords:** Computer Vision, Feature Detection, Imaging, Edge detection

# 1 Introduction

Feature detection is widely used in a growing number of areas such as automotive, medical and security. Depending on the industry there may be different optimal algorithms so for the purpose of this paper all applications will be considered as automotive based. The aim of feature detection is to identify patterns within the Regions of Interest. These patterns can then be further processed and identified against a database. This paper will review common feature detection algorithms, Harris-Stephens, Shi-Tomasi and Triggs in terms of feature detection performance.

# 2 State of the Art

Feature detection has developed a lot in recent years due to its importance in future technologies. Many computer vision applications rely on it such as object detection, structure-from-motion and image retrieval. [1]

## 2.1 Feature Detector Performance Evaluation

The need for having accurate methods of evaluation feature detection performance has grown with the number of new algorithms. The prime focus of evaluating feature detector performance has been based on first examining edge detection algorithms. [2] There are a few ways to measure performance of edge detection algorithm. One common method is called Quantitative Performance Evaluation [3]. A common method is to establish an image test set of input images. Make small variations by changing the image parameters. Next run each image through the algorithm whilst varying the algorithm parameters. This will establish operating curves which relate the probability of miss-detection and false alarm for each parameter of the algorithm. The caveat to this approach is the only one operating control can be used at a time depending on the algorithm parametrization. This methodology is common to any detection algorithm analysis. [3]

# 3 Experimentation

Experimentation was done using the Shi-Tomasi, Harris-Stephens and Triggs methods. Preparing the algorithms in Pycharm [4], results were obtained for features detected in the selected images. Varying the alpha value in each equation more features could be detected however it was clear the real-world examples were impacted by noise when alpha was modified, or when the number of detection points was too low. For this purpose, the comparison of the algorithms separately reviews the performance when used on computer-generated images as opposed to real-world images.

## 3.1 Image Selection

When comparing various algorithms on performance it is essential to have an image test set. The source of the input image can have a large effect on the test results of the algorithm so it’s essential to select a set which is not dissimilar to the end application, without being bias. [5] Depending on whether the image is a real-world example, or a computer-generated image will also have a strong impact as real-world examples will have noise in the background. To build a comprehensive test set of images several different images should be used. [6] For the purposes of this paper a real-world and computer-generated image were used for comparison. Both are similar pictures of road signs. These can be seen in Figure 1: Computer-Generated Road Sign and Figure 2: Real-World Road Sign below.

|  |  |
| --- | --- |
| A stop sign  Description automatically generated  Figure 1: Computer-Generated Road Sign | A stop sign in front of a building  Description automatically generated  Figure 2: Real-World Road Sign |

When experimenting on the real-world example the first issue struck is noise. As the computer-generated sign has no background this is not an issue there. There are a few ways of filtering out noise from the image such as correlation filtering or Gaussian filtering. [7] In experimentation of these algorithms no additional noise filtering was done.

## 3.2 Appling the algorithms

All three algorithms where applied using a similar system. It would read in the 2 images, convert the RGB image to greyscale, perform the detection algorithm and then colour the detected features blue or red. When reviewing the results in this paper successfully detected features will therefore be coloured blue. Controls where then applied to experiment in achieving optimal performance for each algorithm.

## 3.3 Harris-Stephens Algorithm

The Harris-Stephens main parameter is alpha. Varying alpha allows for less dominant features to be detected. With a standard alpha value of 0.06, primarily right angles were detected. With an alpha value of 0.01 many wider angles are detected. [8]

**Computer-Generated**

In Figure3, Harris-Stephens algorithm performed admirably on the computer-generated Stop sign. It Successfully detected the key features and cleared defined the corners in T and P. It did not detect and edges on O as there is no clustering of edges. Similarly, with S only the corners at the end of the tails is detected.

|  |  |
| --- | --- |
| A stop sign  Description automatically generated  **Figure 3: Harris-Stephens (Comp-Gen), Alpha = 0.06** | **Figure 4: Harris-Stephens (Comp-Gen), Alpha = 0.01** |

As seen in Figure4, additional features can be detected as the Alpha value is lowered however there is little benefit in this as the features detected are the border points.

**Real-World**

As can be seen from Figure5 below, noise is much more prominent in the real-world case. However, asides from the noise it is clear that S, T and P are detected as dominantly as they are in the computer-generated image.

|  |  |
| --- | --- |
| A stop sign  Description automatically generated  **Figure 5: Harris-Stephens (Real-World), Alpha = 0.06** | **Figure 6: Harris-Stephens (Real-World), Alpha = 0.01** |

Figure6 shows once again in lowering the alpha value the boarders are detected and more definition on the S, O and P is found however many new noise points are detected.

## 3.4 Shi-Tomasi Algorithm

The main control in the Shi-Tomasi algorithm is identifying how many points of interest are to be identified. If you take the smallest eigenvalues and select the 25 that are big enough you will have 25 points. Noise heavily effects the benefit in adding more points of interest. [9]

**Computer-Generated**

In Figure7, the Shi-Tomasi algorihim clearly detects the S, T, P and detects some points for O. For the S an additional reference points picked up on the inside curve of the S due to the difference in alogorithim it does notrely on a corner to detect. Similarly this is why the boundary is fully detected.

A close up of a logo

Description automatically generated

**Figure 7: Shi-Tomasi (Comp-Gen), Detection Points = 25**

**Real-World**

In Figure8, the real-world situation, this algorithm is seen to degrade as it fails to pick up as many points. It is heavily affected by noise in the sign over the Stop sign.

|  |  |
| --- | --- |
| A stop sign and a traffic light  Description automatically generated  **Figure 8: Shi-Tomasi (Real-Word), Detection Points = 25** | **Figure 9: Shi-Tomasi (Real-Word), Detection Points = 50** |

As seen in Figure9, when the number of points of detection is increased, a few more points of interest on the sign are found, but similarly there is an increase in noise detected.

## 3.5 Triggs’ Algorithm

Triggs algorithm works by taking an alpha value by the maximum eigenvalue away from the minimum eigenvalue. [10]

**Computer-Generated**

Using an alpha value of 0.05 there is excellent clarity of features in the computer-generated image

A close up of a logo

Description automatically generated

**Figure 10: Triggs (Comp-Gen), Detection Points = 25**

**Real-World**

In Figure 11 there are to many points of detection effected by noise, it is evident an increase is required.

|  |  |
| --- | --- |
| A red stop sign sitting on the side of the road  Description automatically generated  **Figure 11: Triggs (Real-Word), Detection Points = 25** | **Figure 12: Triggs (Real-Word), Detection Points = 50** |

As demonstrated in Figure12, by increasing the number of detection points the image picks up a lot more noise however proportionately more points of interest on the sign are detected too.

# 4 Results and comparison

Throughout experimentation it is clear to see there are benefits of each algorithm tested. The following table makes note of the performance of each algorithm in handling the computer-generated situation.

|  |  |  |  |
| --- | --- | --- | --- |
| Algorithm | Control Applied | Advantages | Disadvantages |
| Harris-Stephens | Alpha | Confident detection of important corners. With decreasing alpha value more important points are detected | Curves are not detected |
| Shi-Tomasi | Alpha, Number of Points | Strong clarity on curves, especially on P. | Low confidence on detected points due to variance |
| Triggs | Alpha, Number of Points | Strong detection on letters with many corners | Curves are not detected |

**Table1: Computer-Generated Algorithm Results**

From the Computer-Generated image it is clear a Harris-Stephens outperformed both the Shi-Tomasi and Triggs algorithms due to clarity and confidence. Trigs and Shi-Tomasi both need to allow a large number of detection points in the control to pick up the points of interest. The following table, Table2, details the performance of each algorithm in handling the real-world situation.

|  |  |  |  |
| --- | --- | --- | --- |
| Algorithm | Control Applied | Advantages | Disadvantages |
| Harris-Stephens | Alpha | Strong presence of feature detection in the sign. | Noise is densely detected. |
| Shi-Tomasi | Alpha, Number of Points | Can greatly increase number of detection points to get some sign detection | Background noise is more easily detected. Fails to detect features  Pattern does not correlate with Computer-Generated image. |
| Triggs | Alpha, Number of Points | Can greatly increase number of detection points to get some sign detection | Only T is correctly identified.  Noise is detected first. |

**Table2: Real-World Algorithm Results**

Similar as the computer-generated scenario, the real-world scenario demonstrates Stephen-Harris algorithm as the best performer. The pattern detected for “STOP” in both the computer-generated and real-world scenarios has many common points, showing there is a pattern present. A lot of noise is detected across all three algorithms however as Shi-Tomasi and Triggs rely on more involved control of the number of points to be detected. With a large number a lot of noise can be picked up however with few detection points the points of interest may be omitted.

# 5 Conclusions

From the results of testing done through this paper Harris-Stephens corner detector proved to be the most effective, however it is evident some curved letters may not be detected by this. As both patterns generated from the computer and real-world images appeared to correlate, this sets up well for following on with pattern recognition against a signpost database. Triggs algorithm similarly had common patterns between the real-world and computer-generated examples setting it up to be an effective means of feature detection too however it did have drawbacks of requiring a more intelligent control. For these common signposts, Shi-Tomasi did not perform well however it did have better performance on curved letters so depending on the sign being detected, Shi-Tomasi maybe the favoured algorithm in other cases.

Additionally, Harris-Stephens algorithm proved to be the easiest to implement with very little control required. Reducing alpha allows for a flood of data but with better feature detection.

# 6 References

|  |  |
| --- | --- |
| [1] | D. Tyagi, “Introduction To Feature Detection And Matching,” Medium,com, 3 January 2019. [Online]. Available: https://medium.com/analytics-vidhya/introduction-to-feature-detection-and-matching-65e27179885d. [Accessed 25 March 2020]. |
| [2] | P. Rockett, “Performance assessment of feature detection algorithms: a methodology and case study on corner detectors,” *IEEE Transactions on Image Processing IEEE Trans. on Image Process,* vol. 12, no. 12, pp. 1668-1676, 2003. |
| [3] | Tapas Kanungo, M.Y. Jaisimha, John Palmer, Rober M. Haralick, “A methodology for quantitative performance evaluation of detection algorithms,” *IEEE Transactions on Image Processing,* vol. 4, no. 12, pp. 1667-1675, 1995. |
| [4] | J. Brains, “PyCharm,” JetBrains, 17 March 2020. [Online]. Available: https://www.jetbrains.com/pycharm/download/#section=windows. [Accessed 28 March 2020]. |
| [5] | Benjamin Wilson, Judy Hoffman, Jamie Morgenstern, “Predictive Inequity in Object Detection,” 21 Febuary 2019. [Online]. Available: https://arxiv.org/abs/1902.11097. [Accessed 25 March 2020]. |
| [6] | Michael D. Heath, Sudeep Sarkar, Thomas Sanocki, Kevin W. Bowyer, “A Robust Visual Method for Assessing the Relative Performance of Edge-Detection Algorithims,” *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE,* vol. 19, no. 12, pp. 1338-1350, 1997. |
| [7] | A. Swain, “Noise filtering in Digital Image Processing,” Medium, 2 September 2018. [Online]. Available: https://medium.com/image-vision/noise-filtering-in-digital-image-processing-d12b5266847c. [Accessed 13 March 2020]. |
| [8] | Chris Harris, Mike Stephens, “A Combined Corner and Edge Detector,” in *4th Alvey Vision Conference*, Manchester, 1988. |
| [9] | J. Shi-Tomasi, “Good features to track,” in *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Computer Vision and Pattern Recognition*, Ithaca, New York, 1994. |
| [10] | B. Triggs, “Detecting Keypoints with Stable Position, Orientation, and Scale under Illumination Changes,” in *Computer Vision - ECCV 2004*, Prague, 2004. |