Features from Accelerated Segment Test (FAST)

Deepak Geetha Viswanathan

1. Introduction

FAST is an algorithm proposed originally by Rosten and Drummond [1] for identifying interest points in an image. An interest point in an image is a pixel which has a well-defined position and can be robustly detected. Interest points have high local information content and they should be ideally repeatable between different images [2]. Interest point detection has applications in image matching, object recognition, tracking etc.

The idea of interest point detection or corner detection (both are interchangeably used in literature), is not new. There are several well established algorithms like: Moravec corner detection algorithm, Harris & Stephens corner detection algorithm, SUSAN corner detector.

The reason behind the work of the FAST algorithm was to develop an interest point detector for use in real time frame rate applications like SLAM on a mobile robot, which have limited computational resources [3].

2. Feature Detection using FAST

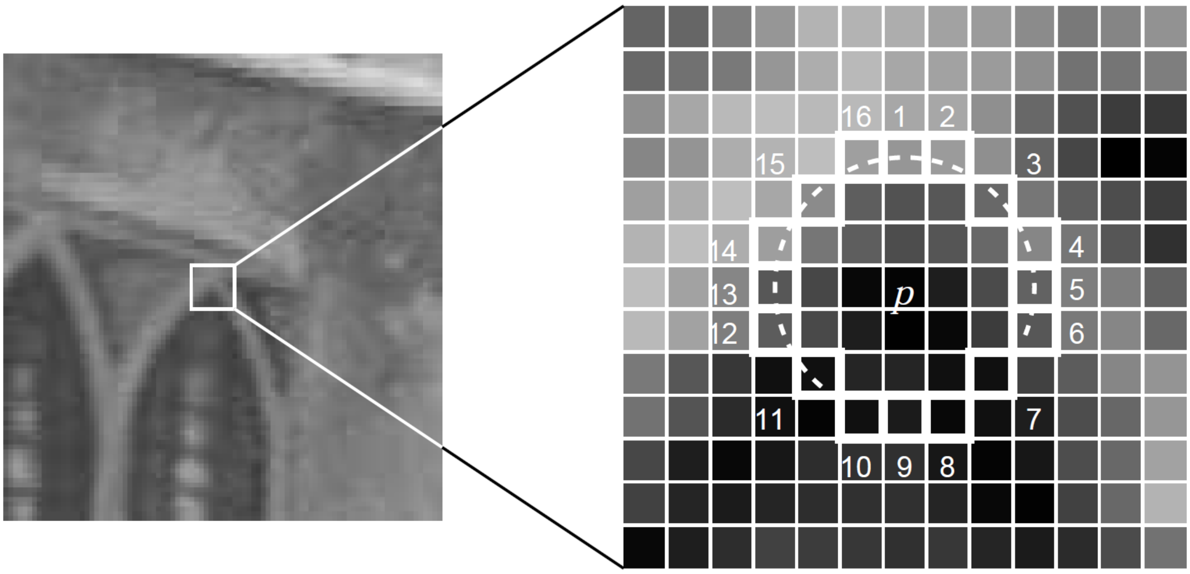

Figure 1. Image showing the interest point under test and the 16 pixels on the circle (image copied from [1]).

The algorithm is explained below:

1. Select a pixel „p" in the image. Assume the intensity of this pixel to be I_P. This is the pixel which is to be identified as an interest point or not. (Refer to fig.1)
2. Set a threshold intensity value T, (say 20% of the pixel under test).
3. Consider a circle of 16 pixels surrounding the pixel p. (This is a Bresenham circle [4] of radius 3.)
4. "N" contiguous pixels out of the 16 need to be either above or below I_P by the value T, if the pixel needs to be detected as an interest point. (The authors have used N = 12 in the first version of the algorithm)

5. To make the algorithm fast, first compare the intensity of pixels 1, 5, 9 and 13 of the circle with I_P. As evident from the figure above, at least three of these four pixels should satisfy the threshold criterion so that the interest point will exist.
6. If at least three of the four pixel values - I_1 ,I_5 ,I_9 I_{13} are **not** above or below I_P + T, then P is not an interest point (corner). In this case reject the pixel p as a possible interest point. Else if at least three of the pixels are above or below I_p + T, then check for all 16 pixels and check if 12 contiguous pixels fall in the criterion.
7. Repeat the procedure for all the pixels in the image.

There are a few limitations to the algorithm. First, for N<12, the algorithm does not work very well in all cases because when N<12 the number of interest points detected are very high. Second, the order in which the 16 pixels are queried determines the speed of the algorithm. [Refer to [5] section III (page 5) for further details.]

A machine learning approach has been added to the algorithm to deal with these issues [3] [5].

3. Machine Learning Approach

1. Select a set of images for training.
2. In every image run the FAST algorithm to detect the interest points by taking one pixel at a time and evaluating all the 16 pixels in the circle.
3. For every pixel „p", store the 16 pixels surrounding it, as a vector. (refer Fig. 2)

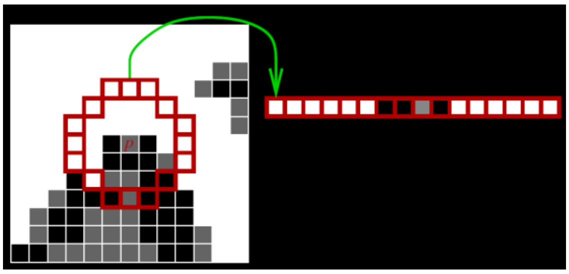

Figure 2. The 16 values surrounding pixel p stored in vector form
(image copied from [5])

4. Repeat this for all the pixels in all the images. This is the vector P which contains all the data for training. (Please note the difference between p-pixel under test and P- the vector)
5. Each value (one of the 16 pixels, say x) in the vector, can take three states. Darker than p, lighter than p or similar to p.
   Mathematically,

$$S_{p\to x} = \begin{cases} d, & I_{p\to x} \leq I_p - t \quad \text{(darker)} \\ s, & I_p - t < I_{p\to x} < I_p + t \quad \text{(similar)} \\ b, & I_p + t \leq I_{p\to x} \quad \text{(brighter)} \end{cases}$$

   $S_{p\to x}$ is the state, $I_{p\to x}$ is the intensity of the pixel x. and t is a threshold.

6. Depending on the states the entire vector P will be subdivided into three subsets, P_d, P_s, P_b.
7. Define a variable K_p which is true if p is an interest point and false if p is not an interest point.
8. Use the ID3 algorithm (decision tree classifier) to query each subset using the variable K_p for the knowledge about the true class.

9. The ID3 algorithm works on the principle of entropy minimization. Query the 16 pixels in such a way that the true class is found (interest point or not) with minimum number of queries. Or in other words, select the pixel x, which has the most information about the pixel p. The entropy for the set P can be mathematically represented as,

$$
\begin{aligned}
&H(P) = (c+\bar{c})\log_2(c+\bar{c}) - c\log_2 c - \bar{c}\log_2 \bar{c} \\
\text{where} \quad & c = \left|\{p|K_p \text{ is true}\}\right| \qquad \text{(number of corners)} \\
\text{and} \quad & \bar{c} = \left|\{p|K_p \text{ is false}\}\right| \qquad \text{(number of non corners)}
\end{aligned}
$$

10. Recursively apply this entropy minimization to all the three subsets.
11. Terminate the process when entropy of a subset is zero.
12. This order of querying which is learned by the decision tree can be used for faster detection in other images also.

4. Non Maximal Suppression for removing adjacent corners [4]

Detection of multiple interest points adjacent to one another is one of the other problems of the initial version of the algorithm. This can be dealt with by applying non maximal suppression after detecting the interest points.

The algorithm is described below:

1. Compute a score function V for each of the detected points. The score function is defined as: “The sum of the absolute difference between the pixels in the contiguous arc and the centre pixel”.
2. Consider two adjacent interest points, compare their V values.
3. Discard the one with the lower V value.

   The entire process can be summarised mathematically as follows:

$$
V = \max \begin{cases} \sum(\text{pixel values} - p) \text{ if } (\text{value} - p) > t \\ \sum(p - \text{pixel values}) \text{ if } (p - \text{value}) > t \end{cases}
$$

where, p is the centre pixel, t is the threshold for detection and pixel values correspond to the N contiguous pixels in the circle.

The score function can be defined in alternate ways also. The key point here is to define a heuristic function which can compare two adjacent detected corners and eliminate the comparatively insignificant one.

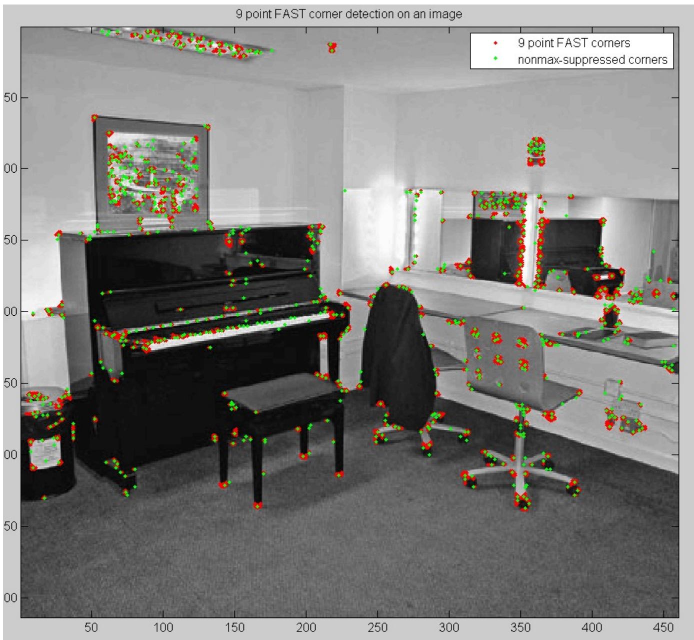

Figure 2. An image with interest points detected. The green dots show the Non-maximally suppressed corners. (Image source: Google. The interest points were generated with an open source MATLAB code for the FAST algorithm)

5. Spatio-Temporal Interest point detection using FAST

There are a few algorithms which extend the FAST algorithm to the spatio temporal domain, for interest point or interest region detection in video with dynamic objects, like moving people.

The fundamental idea is to consider three dimensions, X, Y and Time (T).

One of the algorithms V-FAST [6], considers three circles surrounding the interest point. In the XY plane, XT plane and in YT plane. An interest point is detected when the FAST algorithm criterion is satisfied in the XY plane and in one of XT and YT plane.

There are interest region detectors [7] also based on the FAST algorithm. Spatio temporal interest region detectors are used in video for event detection. Interest points can help in identifying local regions of interest which can be tracked over time.

6. References

1. E. Rosten and T. Drummond, “Machine learning for high speed corner detection,” in 9th Euproean Conference on Computer Vision, vol. 1, 2006, pp. 430–443.
2. Wikipedia, “Corner Detection”,< http://en.wikipedia.org/wiki/Corner_detection>, accessed 16 March 2011.
3. Edward Rosten and Tom Drummond, “Fusing points and lines for high performance tracking” in *Proc. Int. Conf. Computer Vision*, October 2005, vol. 2, pp. 1508–1511.
4. Wikipedia, “Mid-point algorithm”, <http://en.wikipedia.org/wiki/Midpoint_circle_algorithm>
5. Edward Rosten, Reid Porter and Tom Drummond, “FASTER and better: A machine learning approach to corner detection” in *IEEE Trans. Pattern Analysis and Machine Intelligence,2010,*vol 32, pp. 105-119.

6. Yu et al, “Real-time action recognition by spatiotemporal forests”, *BVMC*, *2010*, URL-<http://mi.eng.cam.ac.uk/~cipolla/publications/inproceedings/2010-BMVC-action.pdf>, accessed 16 March 2011.
7. Sander Koelstra, Ioannis Patras, “The fast-3D spatio-temporal interest region detector”, in 10th workshop on Image Analysis for Multimedia Interactive Services, 2009. WIAMIS '09, pp. 242 - 245.

7. Useful Links

The website by Edward Rosten : “FAST Corner Detection -- Edward Rosten” is an online repository of all the work done in this algorithm. It gives links to all the published papers and implementation of FAST algorithms in several programming languages.

Link: < http://mi.eng.cam.ac.uk/~er258/work/fast.html>, accessed 16 March 2011.