# A Machine Learning Approach to Corner Detection

## 1. PROBLEM

A common low-level operation in many computer vision systems is the detection of "interesting" parts, or *features*, within an image. Corners are point-like features in an image, which have a two-dimensional structure, such as where two edges come together, describing high levels of curvature of image gradients in a region. Corners are robust, stable and well-defined image features, and correspond to points in the both the world and image spaces, making them important to many applications.

Corner detection is used as the first step of many vision tasks such as tracking, localization, SLAM (simultaneous localization and mapping), image matching, image stitching and object recognition [Rosten et al. 2010].

In mobile robotics for example, since the camera moves with our robot, we can infer robot motion simply by tracking eight or more corners. Tracking multiple points across consecutive images allows the robot to estimate the relative rotation and translation of the camera, which can be then used for robot localization.

Consistency of image edge filtering is also of prime importance for 3D interpretation of images sequences using feature tracking algorithms [Harris and Stephens 1988], so care must be taken as not to sacrifice quality of the corners found.

However, despite the massive increase in computing power since the inception of corner detectors, it is still true that when processing live video streams at full frame rate, especially on a mobile robot, existing feature detectors leave little if any time for further processing [Rosten et al. 2010].

This paper proposes a unique approach to corner detection: combining a new heuristic for feature detection with machine learning techniques to provide a robust, high-speed corner detection algorithm to use in real-time image processing applications [Rosten et al. 2010].

## 2. RELATED WORK

Corner detection is used as the first step of many vision applications, and this need has driven the development of a large number of corner detectors. These vary widely in the kinds of feature detected, the computational complexity and the repeatability. Some related works for corner detection and feature extraction are as follows:

*SIFT*: Figure 1 shows a very common method for image feature generation is called the Scale Invariant Feature Transform (SIFT). This approach transforms an image into a large collection of local feature vectors, each of which is invariant to image translation, scaling, and rotation, and partially invariant to illumination changes and affine or 3D projection [Lowe 1999]. A Difference of Gaussian (DoG) kernel is computed at multiple scales (sizes) of the image to provide a stable scale-space technique [Rosten and Drummond 2006]. SIFT features are very stable and robust to changes in illumination, rotation and scale of the features, and is widely used for image matching [Lowe 1999]. A disadvantage to SIFT is that it not affine-invariant, such as arbitrary viewpoint changes [Rosten et al. 2010].

Fig. 1. SIFT features in action

*Harris*: The Harris corner detector functions by considering a local window in the image, and determining the average changes of image intensity that result from shifting the window by a small amount in various directions [Harris and Stephens 1988]. This is done by defining a corner to have large eigenvalues in a given image patch [Rosten and Drummond 2006]. One of the main advantages of the Harris operator is that it is very stable, and robust to noise [A and B ]. One of the main disadvantages is a manual, empirical parameter tuning for good performance [Rosten and Drummond 2006].

*SUSAN*: SUSAN is another related feature detector, and stands for "Similar Univalue Segment Assimilating Nucleus". Each image point has an associated local-area brightness, and is effectively region finding on a small scale. Each feature point in SUSAN contains the most important information about the circular region around any point, by computing a self-similarity of brightness around the center (nucleus) pixel. [Smith and Brady 1997] A low value of self similarity indicates a corner, since the patch is not self-similar [Rosten and Drummond 2006]. The advantages of the SUSAN operator are fast computation and outstanding results with unimpaired and unsmoothed images [A and B ]. Since it's based on brightness of pixels, one of the disadvantages of SUSAN corner detector is that it is too sensitive to contrast changes between images of the same object [A and B ].

## 3. APPROACH

This paper proposes an algorithm for high-speed corner detection, named FAST: Features from Accelerated Segment Test [Rosten et al. 2010]. FAST proposes using machine learning to help find optimal heuristics in feature detection technique, known as a segment test, while also striving for the goal of extremely fast processing time. It is desired that real-time image processing application, such as mobile robot navigation software, will benefit from using FAST.

*Segment Test.* The segment test criterion operates by considering a neighborhood of pixels around a corner candidate $p$. A simple base detector classifies $p$ as a corner if there exists a set of $n$ contiguous pixels in the region which are either all brighter than the candidate pixel's intensity $I_p$ plus a threshold $t$, or all darker than $I_p - t$ [Rosten and Drummond 2006]. To accelerate this test, only the neighboring compass pixels 1, 5, 9, and 13, must be examined, since for when $n = 12$, $p$ is a corner if at least three of these are darker than $I_p - t$ or brighter than $I_p + t$ [Rosten and Drummond 2006]. This can be repeated for all candidate in the rest of the image. The segment test in itself can exhibit high performance, but there are several weaknesses that remain [Rosten and Drummond 2006]:

—The high-speed test does not generalize well for $n < 12$
—The choice and ordering of the fast test pixels imposes assumptions about a feature's appearance
—Knowledge from the four tests is discarded between candidates
—A large corner may be detected as multiple adjacent features

*Machine Learning.* To address the issues above, this paper proposes an machine learning solution to help learn and tune better segment test heuristics automatically. An initial start to parametrization the segment test for machine learning is described here.

First, the segment test as described above is run, detecting corners from a set of images (preferably from the target application domain, but not essential), using a given $n$ and a convenient threshold (usually 3) [Rosten and Drummond 2006]. The slow version of the segment test is used here, where every pixel is fully tested and classified. For each pixel $x$ in the neighborhood of corner candidate $p$, a state $S_{x \to p}$ based on whether $x$ is darker, similar, or brighter relative to $p$ [Rosten and Drummond 2006]. Let this be done for the set of all pixels in the image, $p \in P$, so that $P$ is partitioned into three subsets, $P_d, P_s, P_b$ (darker,similar,brighter), based on a chosen $x$.

Next, the quality of a given corner candidate must be calculated. As a first step, the use of ID3 (Iterative Dichotomiser 3, related to information gain) is proposed for determining image entropy $K$, but other methods and heuristics shall be researched as well. For a set of corners $Q$, $K$ is calculated for each $x$ to find a partitioning of the darker, similar, or brighter sets ($P_d, P_s, P_b$) that maximizes the information for each corner candidate. $K$ can now represent the data in terms of the information gained from choosing each corner candidate as one of $P_d, P_s, P_b$ [Rosten et al. 2010].

Now, using a machine learning algorithm such as a *decision tree*, the correct formula for choosing corners based on $x$ belonging to $P_d, P_s, P_b$ can be automatically learned. The decision tree learned here can then be translated into a "hard-coded" fast classifier routine for determining when a pixel region (from the segment test) is a corner or not.

*Project Focus.* This idea of doing off-line machine learning on what makes a good corner as a preprocessing stage, is the basis for how an efficient and fast corner detector can be made. The process just described will serve as the starting point for this project. Some questions this research will try to address are: What parameters to tune? What other learning algorithms can be used? What comparisons other than entropy are useful? Is there any other preprocessing or post-processing that should take place? What are the performance trade-offs for more elaborate feature descriptors? All of these questions and more will help guide the research. A major emphasis will be on finding the fastest possible combination of all these approaches.

## 4. EVALUATION

The meaning of "corners" found in an image, however, depends on the context of the application and therefore do not necessarily correspond to physical corners in the scene [A and B ], but rather feature points of interest. So when evaluating, "corners" may also simply refer to the found salient image features and interest points.

It is necessary to define the requirements to an optimal interest operator. As criteria for a distinctive matching evaluation, the ideal candidate the characteristics proposed by [A and B ] provide good metrics for comparison:

—*Distinctness*: An interest point should stand out clearly against the background and be unique in its neighborhood.
—*Invariance*: The determination should be independent of the geometrical and radio-metrical distortions.
—*Stability*: The selection of interest points should be robust to noise and blunders.
—*Uniqueness*: Apart from local distinctiveness an interest point should also possess a global uniqueness, in order to improve the distinction of repetitive patterns.
—*Interpretability*: Interest values should have a significant meaning, so that they can be used for correspondence analysis and higher image interpretation.

Additionally, the performance of the detector under these criteria with respect to processing speed should be done as well, as one of the main goals of this project is to find the highest performing corner detection scheme without sacrificing the above mentioned qualities. It is believed FAST will "shine" in the areas of speed, invariance, and stability by using the segment test, while the machine learning should help FAST perform well in the areas of uniqueness, distinctness, and interpretability.

## 5. PLAN

As with all major projects, it it crucial to know the goal and direction of the research, and keep track of where time is spent. The major milestones that should progress consist of:

—Feature detection algorithms research and development
—Related machine learning techniques research
—Algorithm prototyping and feature detection approach fusion
—Software prototyping and evaluation
—Software implementation and tuning
—Final report

An example ideal time-line of how this project shall be completed is shown in Figure 2. Boxes represent focused parts of the research, and related areas (boxes) overlap. The groupings of topics are not fixed and merely serve as a guide on pacing the overall research flow.

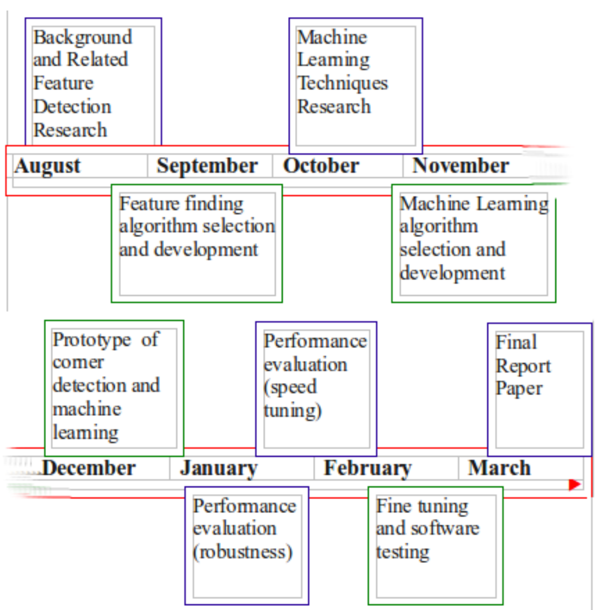


Fig. 2. Time projection of project milestones

Resources for this project include related academic papers pertaining to machine learning and computer image processing algorithms. While there is no requirements on programming language specifics, a final code that is fast and reliable should be produced. Knowledge of code optimization techniques may serve beneficial as well.

While the final deliverable will be a very fast and quality corner detection code, another useful artifact of this proposal will be an extensive write-up of the trials and errors that may be encountered along the way. Much research will be done here to test the possibilities of combining machine learning techniques and feature detection, and any insights along the way will be noted and also be put in the final report for future reference.

## REFERENCES

A, V. R. AND B, A. K. Comparison and evaluation of feature point detectors. *In Proceedings of 5th International Symposium Turkish-German Joint Geodetic Days*.

HARRIS, C. AND STEPHENS, M. 1988. *A combined corner and edge detector*. Vol. pages. 147–151.

LOWE, D. 1999. Object recognition from local scale-invariant features. 1150–1157.

ROSTEN, E. AND DRUMMOND, T. 2006. Machine learning for high-speed corner detection. In *In European Conference on Computer Vision*. 430–443.

ROSTEN, E., PORTER, R., AND DRUMMOND, T. 2010. Faster and better: A machine learning approach to corner detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 105–119.

SMITH, S. M. AND BRADY, J. M. 1997. Susana new approach to low level image processing. *International Journal of Computer Vision 23*, 45–78. 10.1023/A:1007963824710.