Intro to FAST, BRIEF, ORB

A Preprint

Algorithm FAST was presented in 2005 and was one of the first heuristic approaches to finding key points, which became popular due to it s efficiency. This approach focuses on pixel‘s brightness within circle with radius 3. By comparing brightnesses to brightness of center we get 3 values: brighter, darker, same. The point is counted is key point if there is 12 consecutive points which are darker or brighter. In order to speed up the process, the authors suggested first checking only four pixels numbered 1, 5, 9, 13. If there are 3 lighter or darker pixels among them, then a full check is performed on 16 points, otherwise the point is immediately marked as "not special". This greatly reduces the time of work, on average, it is enough to poll only about 4 points of the circle to make a decision. However, the original FAST has the following shortcomings:

Several adjacent pixels can be marked as singular points. Some measure of the "strength"of a feature is needed. One of the first measures proposed is the maximum value of t at which a point is still accepted as singular. The fast 4-point test does not generalize to n less than 12. For example, the visually best results of the method are achieved with n=9, not 12.

Instead of using a cascade of two tests of 4 and 16 points, it is proposed to do everything in one pass through the decision tree. The idea if pretty similar but authors achieve keypoint detection in $\tilde{2}$ steps.Each tree branching is defining if pixel is lighter, darker or the same as the center. The point is chosen if the entropy is maximum, the process stops when entropy equals zero. The whole idea means that decision making is optimized by tree. Algorithm is trained to make good decisions based on checking particular pixels. In that way decision making is way faster. The final step is NMS. For each point specific mere is calculated: $V = \max(\sum_{x \in S_{bright}} |I_x - I_p| - t, \sum_{x \in S_{dark}} |I_p - I_x| - t)$, where I_p is brightness of center, I is brightness of pixels in circles.

FAST-ER is an improved version of FAST, developed by the same authors. Like Tree FAST, it constructs a decision tree to detect corners efficiently, but instead of using greedy construction, the tree is optimized globally using simulated annealing.

The key idea is to treat the detector as a decision tree and define an objective function to evaluate its quality. The chosen criterion is repeatability, which measures how reliably keypoints are detected across different views of the same scene.

Given a pair of images, a keypoint is useful if it is visible in both views, and repeated if it is detected in both. Repeatability is then defined as:

$$R = \frac{N_{\text{repeated}}}{N_{\text{useful}}}$$

To avoid degenerate solutions (e.g., detecting everything or only one point), the authors propose a compound cost function:

$$k = \left(1 + \left(\frac{w_r}{R}\right)^2\right) \left(1 + \frac{1}{N}\sum_{i=1}^{N} \left(\frac{d_i}{w_n}\right)^2\right) \left(1 + \left(\frac{s}{w_s}\right)^2\right)$$

Here, R is repeatability, d_i is the number of detected keypoints in frame i, N is the total number of frames, s is the tree size (number of nodes), and w_r, w_n, w_s are tunable weights.

A solution (a detector tree) is modified iteratively. At each step, a random node is selected. If it is a leaf:

- Replace it with a random subtree of depth 1
- Or change its classification (corner/non-corner)

If it is an internal node:

- Change the tested point (from 0 to 47)
- Replace it with a leaf
- Swap its subtrees

The temperature schedule is defined as:

$$T(I) = \beta \exp\left(-\frac{\alpha I}{I_{\max}}\right)$$

with constants α, β, and $I_{\max}$ as the total number of iterations.

A modification is accepted with probability:

$$P = \exp\left(\frac{k(i-1) - k(i)}{T}\right)$$

This process allows both growth and shrinkage of the tree. Since it's stochastic, multiple runs are performed (e.g., 100 runs of 100,000 iterations), and the best solution is selected.

FAST-ER uses 47 test points instead of 16, but the core idea remains. The result is a detector that outperforms Tree FAST, especially on noisy images.

Brief is a fasts heuristic descriptor, build from 256 binary comparisons between pixel‘s brightness on blurred image. The original paper considered several methods for selecting points for binary comparisons. As it turned out, one of the best methods is to select points randomly with a Gaussian distribution around the central pixel. This random sequence of points is selected once and does not change. The size of the considered point neighborhood is 31x31 pixels, and the blur aperture is 5.

The resulting binary descriptor is robust to lighting changes, perspective distortion, is quickly calculated and compared, but is very unstable to rotation in the image plane. It is also important to note that original imagged is blurred using gaussian blur before using descriptor.

Oriented FAST and Rotated BRIEF is a combination of all previous ideas. Approach is robust and yet efficient.

ORB (Oriented FAST and Rotated BRIEF) extends the BRIEF descriptor to make it rotation-invariant and more discriminative. The algorithm works as follows:

1. Keypoints are detected using a fast tree-based FAST detector applied to the input image and across multiple levels of a scale pyramid.

2. A Harris corner score is computed for each keypoint; those with low response are discarded.

3. The orientation of each keypoint is computed using intensity moments over its neighborhood:

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y), \quad \theta = \operatorname{atan2}(m_{01}, m_{10})$$

This approach is referred to as "centroid orientation".

4. Given the orientation θ, all pairs of sampling points in the BRIEF descriptor are rotated:

$$\begin{pmatrix} x_i' \\ y_i' \end{pmatrix} = R(\theta) \cdot \begin{pmatrix} x_i \\ y_i \end{pmatrix}$$

This makes the descriptor rotation-invariant.

5. The BRIEF descriptor is then computed using these rotated coordinates.

However, rotating all keypoints to zero orientation introduces statistical artifacts: some binary comparisons become correlated, and the mean value of bits deviates from 0.5, reducing discriminative power.

To address this, ORB selects a set of binary tests based on training:

- Evaluate all possible tests across a training set of keypoints.

- Rank them by their deviation from the ideal mean of 0.5.
- Initialize an empty set R.
- Add the best test to R.
- For each next test, if its correlation with all tests in R is below a threshold, include it; otherwise discard.

This ensures selected tests are both uncorrelated and balanced, yielding a robust and discriminative descriptor.