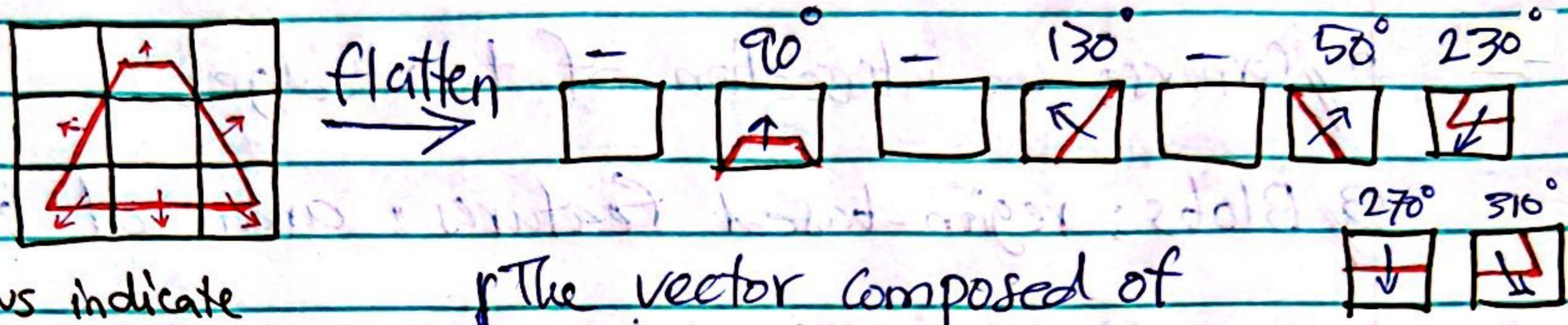
« Feature Vectors »

Gives us a robust representation of the shape of an object.



arrows indicate gradient in the image

The vector composed of the gradient direction of each cell is « feature vector »

→ But these vectors should allow for enough flexibility to detect (or accomodate) some variations in the shape while remaining distinct enough to be able to distinguish different shapes.

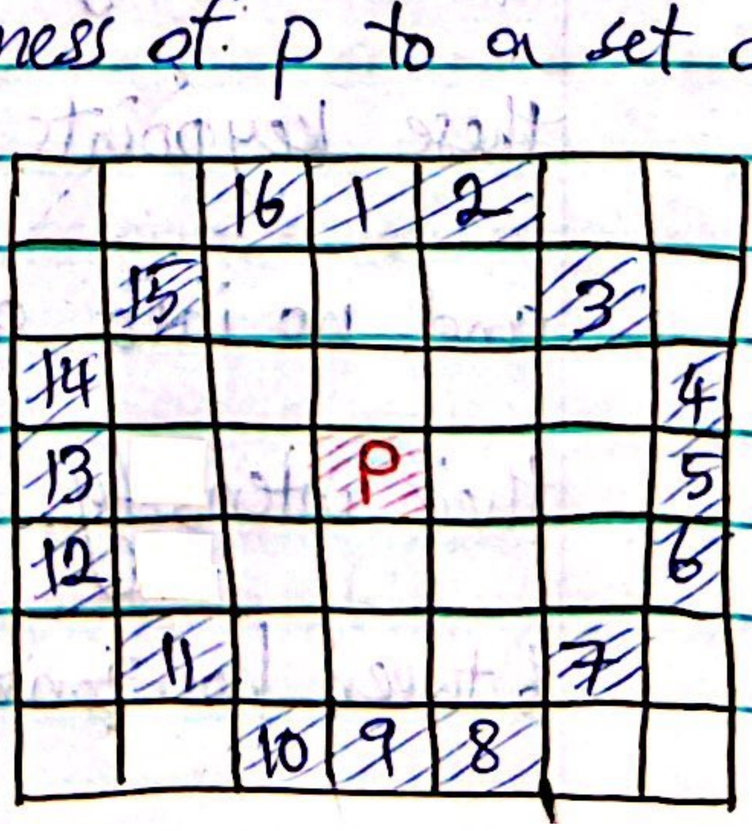
* Time is very critical in some applications such as robotics and self-driving cars: → We need to identify the objects in a fast and efficient manner. → This is where ORB comes in! → ORB creates feature vector of key points in an image → We use these features to identify objects in an image.

* ORB: Oriented fast and Rotated Brief

→ fast and Brief are feature detection in a vector creation algorithm. → ORB detects key points which are small regions in an image that are particularly distinctive, e.g., corners → Then ORB calculates corresponding feature vectors for each key point. (These vectors are binary vectors) → The vector stores patterns of intensity around a key point. → ORB is not only fast, but it's also impervious to noise, illumination, and image transformations such as rotations.

* The FAST algorithm: the first stage in ORB which detects key points which stands for Features from Accelerated Segments Test → for a pixel p in an image, FAST compares the brightness of p to a set of 16 surrounding pixels that are in a small circle around p.

→ Each pixel in this circle



is then sorted into 3 classes:

1) brighter than p $\geq I_{p+h}$ 2) darker than p $\leq I_{p-h}$

3) Similar to p $I_{p-h} < < I_{p+h}$

We refer brightness (intensity) of pixel p as I_p

→ We refer pixel p as keypoint if at least 9 out of 16 pixels are either darker or brighter than p.

→ The reason that FAST is so fast is that the same result will be achieved if we compare Ip to those of pixels 1, 5, 9, and 13 instead of all 16 pixels. → p is a key point if atleast a pair of these four points are either brighter or darker than p.

* Keypoints extracted by FAST give us info about the location of object defining edges in an image → But, these keypoints give us info about the location of edges and no info about their direction of the change in their intensity. e.g., we cannot distinguish between horizontal and vertical edges!

* The BRIEF algorithm in ORB, converts key points detected by FAST into feature vectors that together represent an object in the image. BRIEF stands for Binary Robust Independent Elementary features.

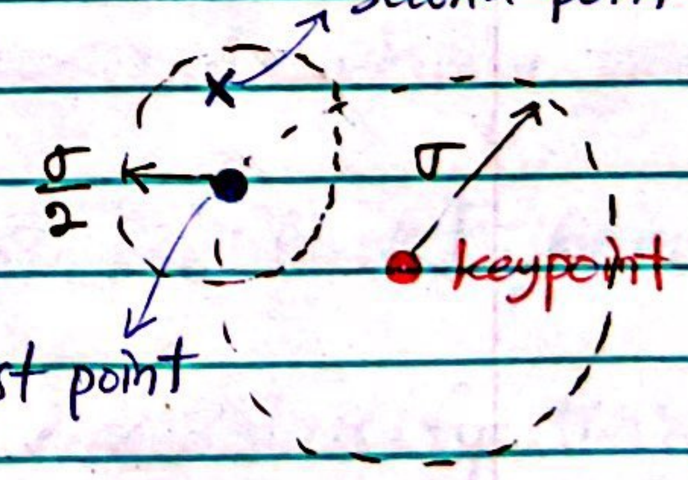
Binary Feature Vector: 128-512 bit string

$V_1 = [01011109 \ldots]$

↳ They can be stored efficiently and computed quickly.

1/ First, the BRIEF smoothen the image by Gaussian blur to prevent the descriptor from being too sensitive to high frequency noise

2/ In the next step, BRIEF samples two point via a Gaussian distribution. The first one, centered at the given key point with std of σ. the second point with a Gaussian distribution centered at the first point with std of $\frac{\sigma}{2}$.



3/ Then Brief starts to construct the binary descriptor

for the key point by comparing the brightness of two sampled pixels, If the first pixel is brighter than the second we assign the value of one to the corresponding bit in the descriptor; otherwise we assign zero.

4. We repeat stages 2 and 3 several times for the same key point before moving onto the next key point. (256)