Introduction to Computer Vision for Robotics

AE640A Autonomous Navigation

12th March, 2019

Lecture Outline

- Features
  - Motivation for feature points
  - Harris Detector
- SIFT

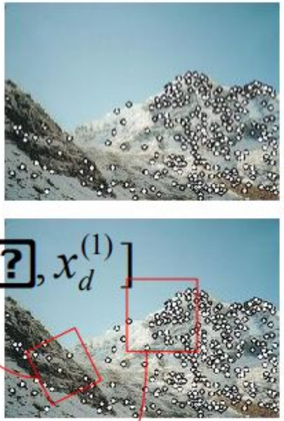
Features

Features

Credits: Vinay Namboodiri

Features

1) Detection: Identify the interest points



$$\mathbf{x}_1 = [x_1^{(1)}, \boxed{?}, x_d^{(1)}]$$

2) Description:Extract vector feature descriptor surrounding each interest point.

$$\mathbf{x}_2 = [x_1^{(2)}, \boxed{?}, x_d^{(2)}]$$

3) Matching: Determine correspondence between descriptors in two views

Credits: Vinay Namboodiri

Features

- Repeatability
  - The same feature can be found in several images despite geometric and photometric transformations
- Saliency
  - Each feature has a distinctive description
- Compactness and efficiency
  - Many fewer features than image pixels
- Locality
  - A feature occupies a relatively small area of the image; robust to clutter and occlusion

Credits: Vinay Namboodiri

Features

- We want to detect (at least some of) the same points in both images.

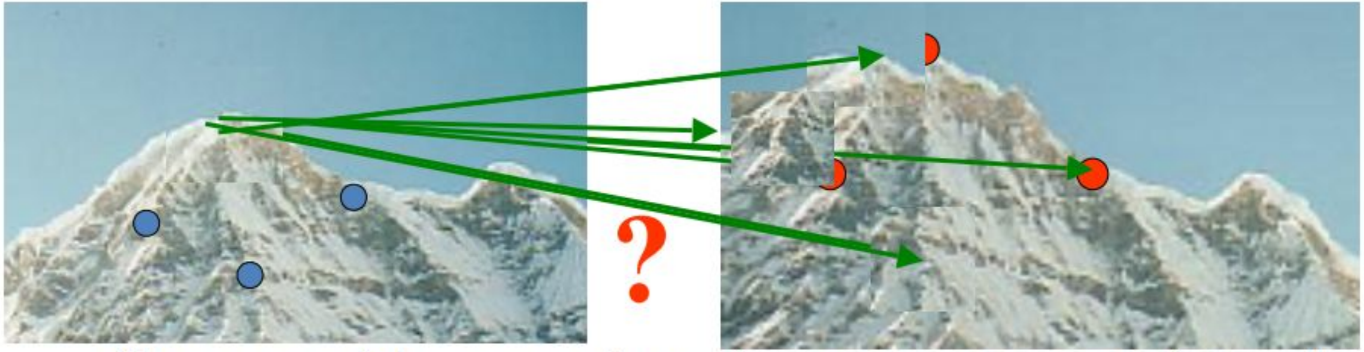
No chance to find true matches!

- Yet we have to be able to run the detection procedure *independently* per image.

Credits: Vinay Namboodiri

Features

- We want to be able to reliably determine which point goes with which.



- Must provide some invariance to geometric and photometric differences between the two views.

Credits: Vinay Namboodiri

Features

Look for image regions that are unusual

1) **Lead to unambiguous matches in other images**

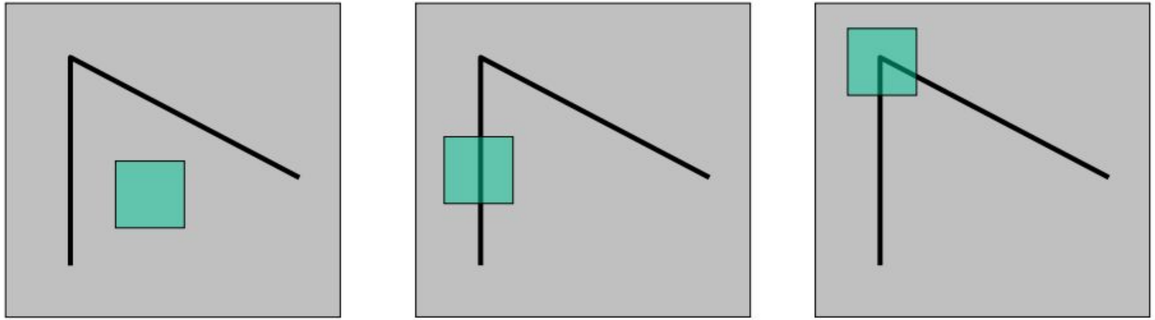
How to define "unusual"?

Credits: Vinay Namboodiri

Features

Suppose we only consider a small window of pixels

1) What defines whether a feature is a good or bad candidate?

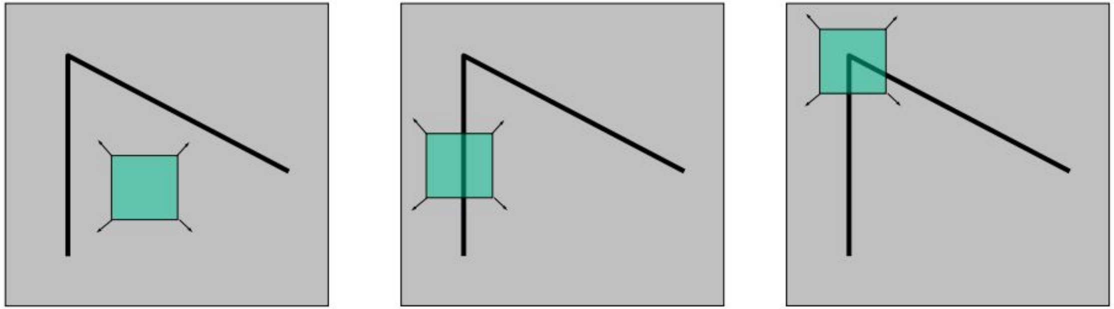


Credits: Vinay Namboodiri

Features

Local measure of feature uniqueness

1) How does the window change when you shift it?
2) Shifting the window in *any direction* causes a *big change*



"flat" region: no change in all directions

"edge": no change along the edge direction

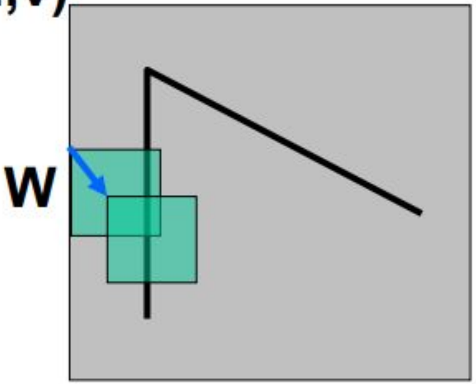
"corner": significant change in all directions

Credits: Vinay Namboodiri

Features

Consider shifting the window W by (u,v)

1) **how do the pixels in W change?**
2) **compare each pixel before and after by summing up the squared differences (SSD)**
3) **this defines an SSD "error" of *E(u,v)*:**



$$E(u,v) = \sum_{(x,y)\in W} \left[I(x+u, y+v) - I(x,y)\right]^2$$

Credits: Vinay Namboodiri

Features

Taylor Series expansion of I:

$$I(x+u, y+v) = I(x,y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \text{higher order terms}$$

If the motion (u,v) is small, then first order approx is good

$$I(x+u, y+v) \approx I(x,y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v$$

$$\approx I(x,y) + [I_x \;\; I_y] \begin{bmatrix} u \\ v \end{bmatrix}$$

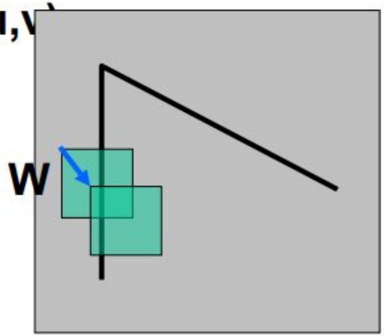
shorthand: $I_x = \frac{\partial I}{\partial x}$

Credits: Vinay Namboodiri

Features

Consider shifting the window W by (u,v)

1) how do the pixels in W change?
2) compare each pixel before and after by summing up the squared differences
3) this defines an "error" of E(u,v):



$$E(u,v) = \sum_{(x,y)\in W} [I(x+u,y+v) - I(x,y)]^2$$

$$\approx \sum_{(x,y)\in W} \left[I(x,y) + [I_x \; I_y]\begin{bmatrix} u \\ v \end{bmatrix} - I(x,y)\right]^2$$

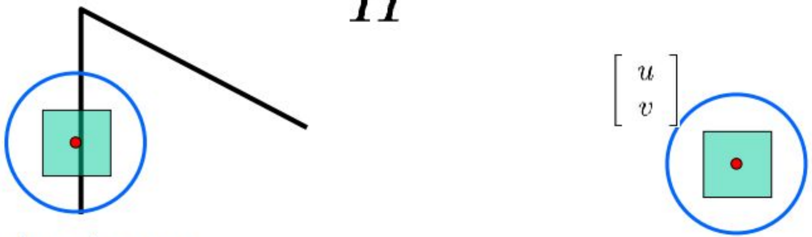
$$\approx \sum_{(x,y)\in W} \left[[I_x \; I_y]\begin{bmatrix} u \\ v \end{bmatrix}\right]^2$$

Credits: Vinay Namboodiri

Features

This can be rewritten:

$$E(u,v) = \sum_{(x,y)\in W} [u\ v] \underbrace{\begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix}}_{H} \begin{bmatrix} u \\ v \end{bmatrix}$$



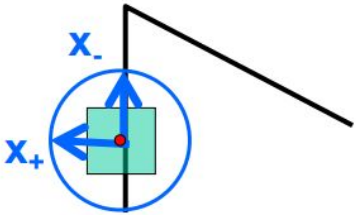
For the example above

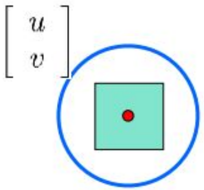
1) You can move the center of the green window to anywhere on the blue unit circle
2) Which directions will result in the largest and smallest E values?
3) We can find these directions by looking at the eigenvectors of ***H***

Credits: Vinay Namboodiri

Features

$$E(u,v) = \sum_{(x,y)\in W} [u\ v] \underbrace{\begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix}}_{H} \begin{bmatrix} u \\ v \end{bmatrix}$$





Eigenvalues and eigenvectors of H

1) **Define shifts with the smallest and largest change (E value)**
2) **x_+ = direction of largest increase in E.**
3) λ_+ **= amount of increase in direction x_+**
4) **x_- = direction of smallest increase in E.**
5) λ_- **= amount of increase in direction x_+**

$$Hx_+ = \lambda_+ x_+$$
$$Hx_- = \lambda_- x_-$$

Credits: Vinay Namboodiri

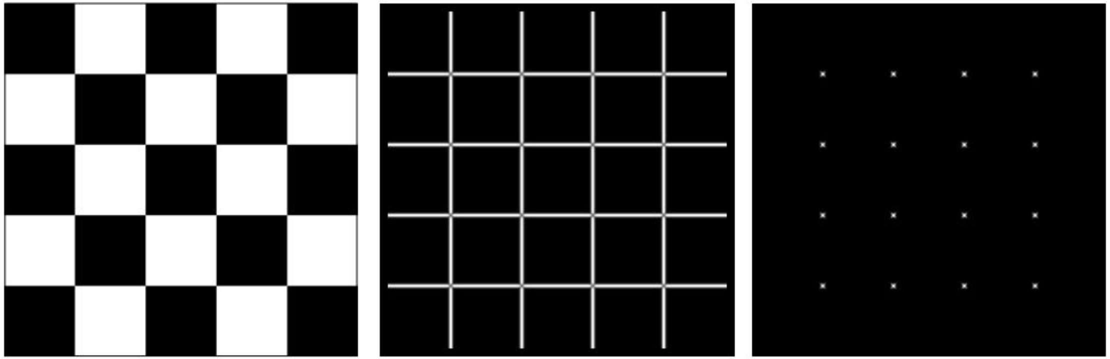
Features

How are λ_+, x_+, λ_-, and x_+ relevant for feature detection?

1) What's our feature scoring function?

Want *E(u,v)* to be *large* for small shifts in *all* directions

1) the *minimum* of *E(u,v)* should be large, over all unit vectors [u v]
2) this minimum is given by the smaller eigenvalue (λ_-) of *H*



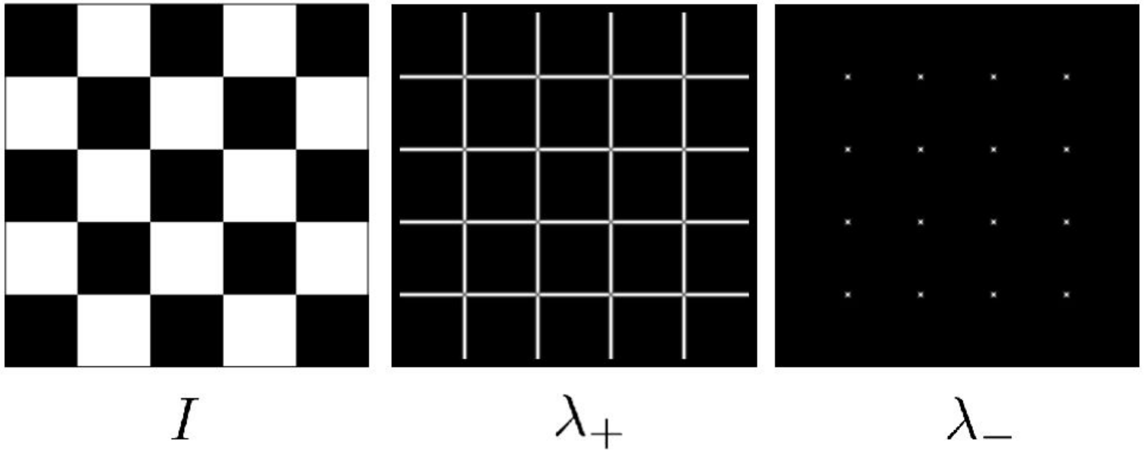
I $\qquad$ λ_+ $\qquad$ λ_-

Credits: Vinay Namboodiri

Features

Here's what you do

1) Compute the gradient at each point in the image
2) Create the *H* matrix from the entries in the gradient
3) Compute the eigenvalues.
4) Find points with large response (λ_- > threshold)
5) Choose those points where λ_- is a local maximum as features



Credits: Vinay Namboodiri

Features

λ_- is a variant of the "Harris operator" for feature detection

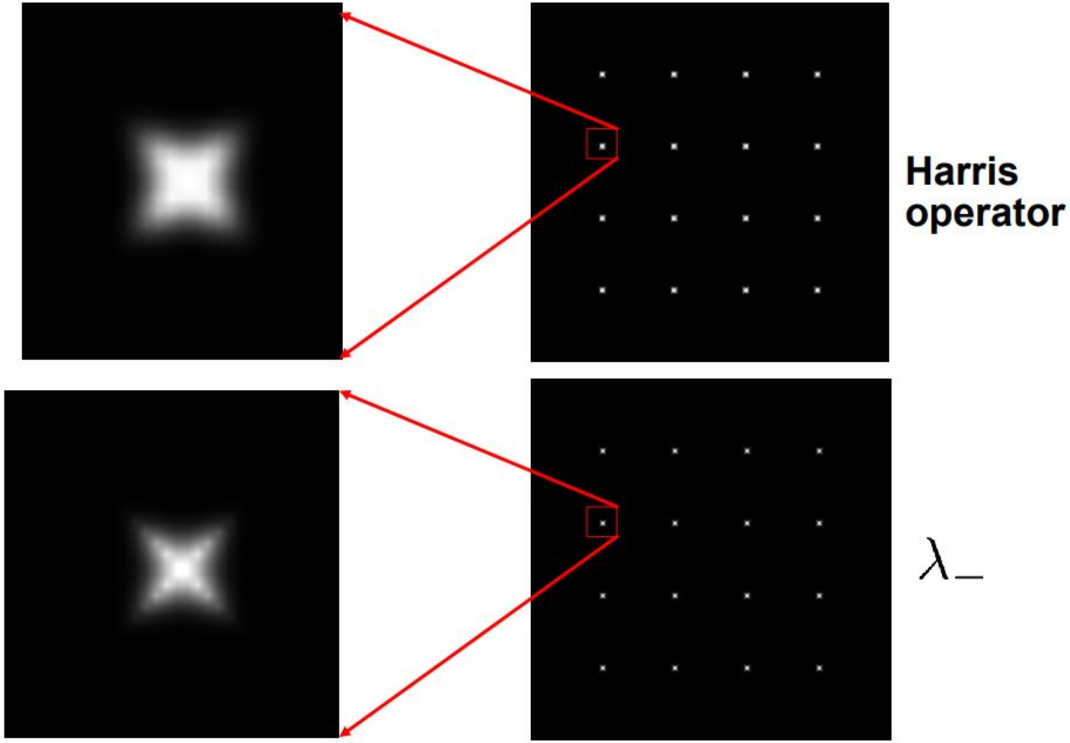
$$f = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2}$$

$$= \frac{determinant(H)}{trace(H)}$$

1) **The *trace* is the sum of the diagonals, i.e., *trace(H)* = $h_{11} + h_{22}$**
2) **Very similar to λ_- but less expensive (no square root)**
3) **Called the "Harris Corner Detector" or "Harris Operator"**
4) **Lots of other detectors, this is one of the most popular**

Credits: Vinay Namboodiri

Features

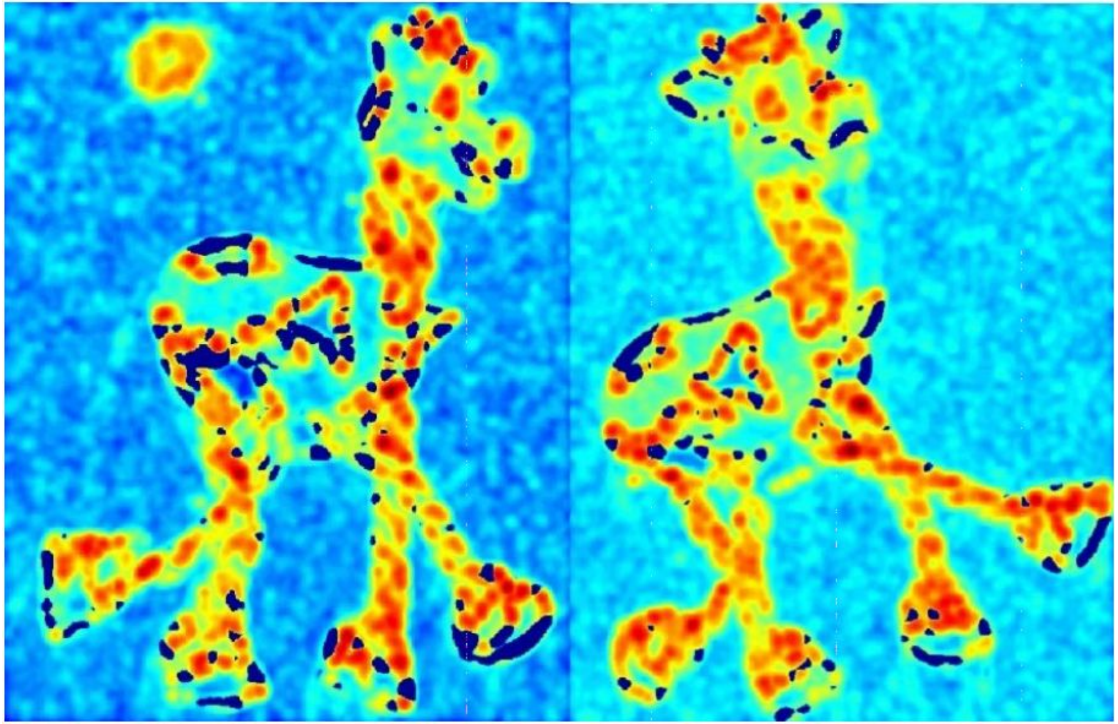


Credits: Vinay Namboodiri

Features

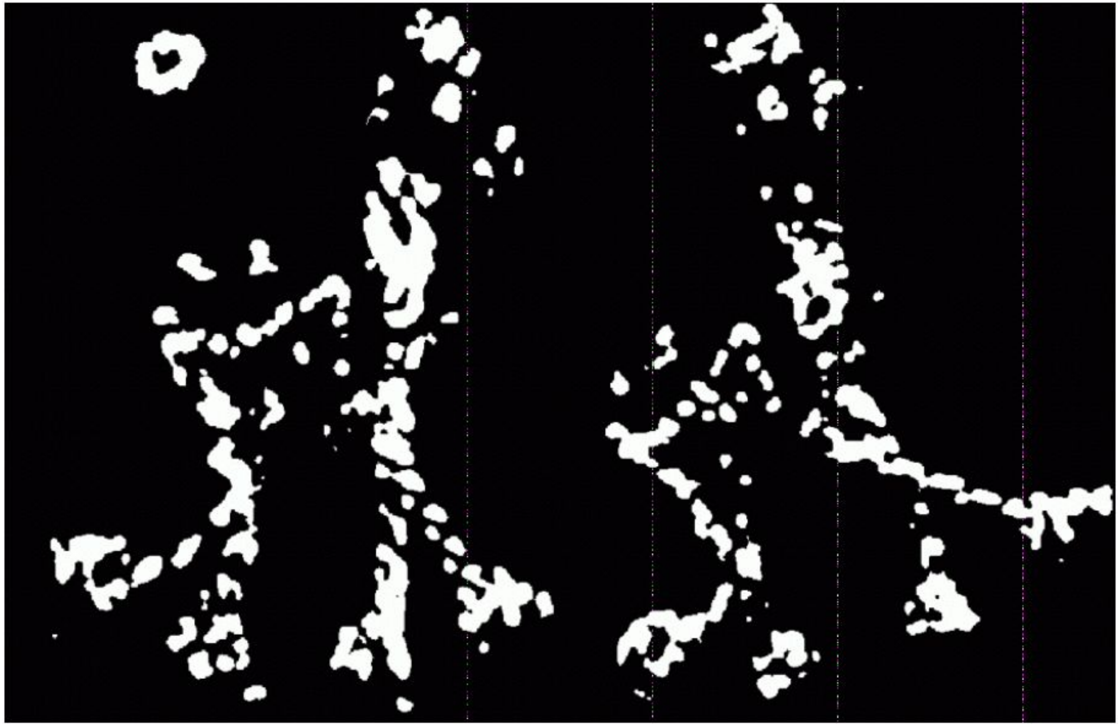
Credits: Vinay Namboodiri

Features



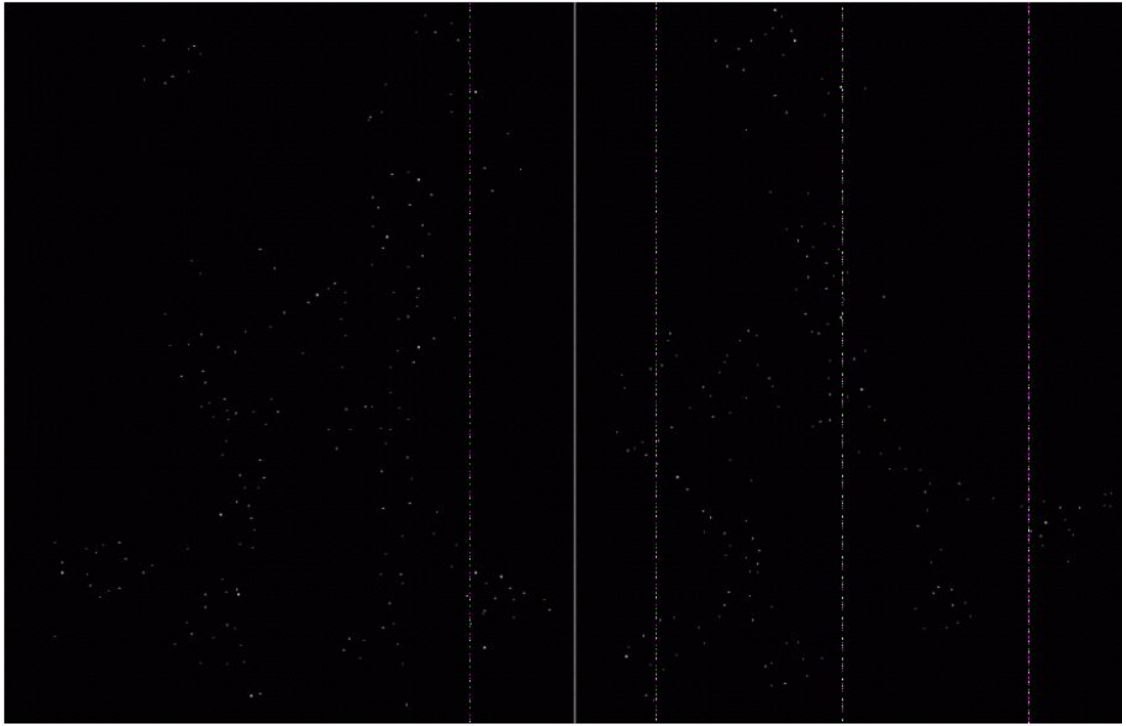
Credits: Vinay Namboodiri

Features



Credits: Vinay Namboodiri

Features



Credits: Vinay Namboodiri

Features

Credits: Vinay Namboodiri

SIFT