/ /

Section ⑥ Feature detector, descriptor

Image processing → detection → description

(Interest Point) → choose points from image → detector.
↳ + image patch → descriptor → pixels
↳ histogram (gradi...

[interest point] + descriptor → local feature
image Registration, 3d Const.

harris, Min Eigen, fast → detectors
* sift → detector, descriptor
(DoG) ↳ histogram of gradient orientation

SURF → fast of sift
Brisk → sift, SURF
detector: Corner detector
descriptor: binary string represent sign of difference between
Center pairs of pixels

Fast Algorithm Corner detector
harris, shi-tomasi غبي
① select pixel → p Intensity Ip
② select threshold (t)
③ 16 pixel around (P)

p: Corner → Ip + t brighter than
↳ Ip − t darker than

/ /

Machine learning Corner detector

1. Select images for training
2. Run fast Algo. to find feature point
3. Store 16 pixel around (P)
4. for each point → determine state → darker / simillar / bright
5. (KP) → all pixels are Corner
6. Algo → descioin tree
7. loop on all points

⇒ Non-maximal suppression

Multiple interest point → in adjacent location

نقرأ نبص نقرأ فيهم Features جنب بعض

1. Score function (V) → sum absolute diffrence between p, 16 surronding
2. 2 adjacent → Calculate (V)
3. discard lower (V) values

Code ↓

Iniate fast Algo → Fast Feature Detector_create : return object

kp = Fast.detect (img , None) → return key points

find of key points

Draw key points → drawkeypoint(img , kp , None , Color) (kp → key points)

(Return) object

1. threshold → fast.getThreshold()
2. nonMaxSuppression → fast.getnonMaxSuppression()
3. neighborhood → fast.getType()
4. total KP with no Max Suppression → len(kp)

/ /

Disable non maximum suppresion →
fast. setNonmaxSuppression (0)

SIft (SCale - invariant feature detector)

haris → rotation invariant   SCale → not invariant
sift → SCale invariant

خطوات
1. SCale Space extrema detection
2. key Point locolization
3. orientation Assignment
4. key descriptor → 16 X 16 window
   ↓
   16 X 16 Sub block → 4 X 4
   ↓
   8 bin
   = 128 bin value
5. key Point matching

Code

```
# create object from algo
sift = cv. SIFT_Create ( )
kp = sift . detect (img , none)  → detect key points
cv. drawkeypoint (img, kp, none)
```

كل حسب الـ key point → كبر الدائرة

```
new img = cv.drawkeypoint(img , kp, none,
    flag =
    cv.draw_matches
    _flags_ Flags_ Draw
    _ Rich
    _ keypoint)
```

Return keypoint , descriptor → object initiated

kp , des = (SIFT) . (detect and Compute (gray , none))