Python Applied to Machine Learning and Statistics

Lecture 03: Computer Vision with Python

Kelwin Fernandes   Ricardo Cruz   Pedro Costa

September 16, 2016

Object Recognition using SIFT (Scale-Invariant Feature Transform)

- SIFT and SURF are not part of OpenCV 3 by default.
- Why?

Object Recognition using SIFT (Scale-Invariant Feature Transform)

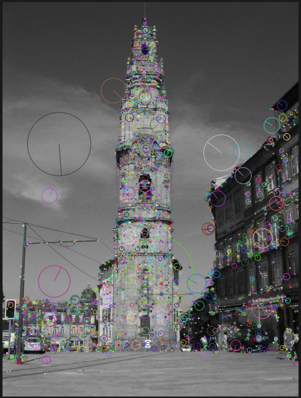
- SIFT and SURF are not part of OpenCV 3 by default.
- Why?

- SIFT and SURF are patented and cannot be used for commercial purposes without permission.

- `opencv_contrib`.
- or use alternative descriptors... BRISK.

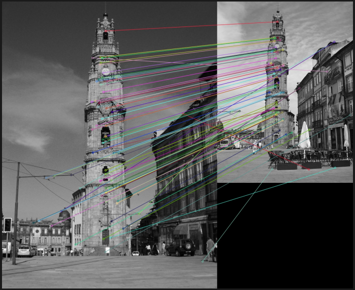
Object Recognition using SIFT (Scale-Invariant Feature Transform)

```python
sift = cv2.xfeatures2d.SIFT_create()
keypoints, descriptor = sift.detectAndCompute(img, None)

dst = cv2.drawKeypoints(img, keypoints, None,
           flags=cv2.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)
```



Object Recognition using SIFT (Scale-Invariant Feature Transform)

```
bf = cv2.BFMatcher()
matches = bf.knnMatch(descr1, descr2, k=2)

good = [[m] for m, n in matches
        if m.distance < thrs * n.distance]
dst = cv2.drawMatchesKnn(img1, kp1, img2, kp2, good,
                         None, flags=2)
```



Background Subtraction

- Loading a video.

```
camera = cv2.VideoCapture('video.avi')
```

- Background Subtraction using Mixture of Gaussians.

```
mog = cv2.createBackgroundSubtractorMOG2()
```

- Get next frame.

```
grabbed, frame = camera.read()
```

- Apply model.

```
fgmask = mog.apply(frame)
```

Background Subtraction

- Loading a video.

```
camera = cv2.VideoCapture('video.avi')
```

- Background Subtraction using Mixture of Gaussians.

```
mog = cv2.createBackgroundSubtractorMOG2()
```

- Get next frame.

```
grabbed, frame = camera.read()
```

- Apply model.

```
fgmask = mog.apply(frame)
```

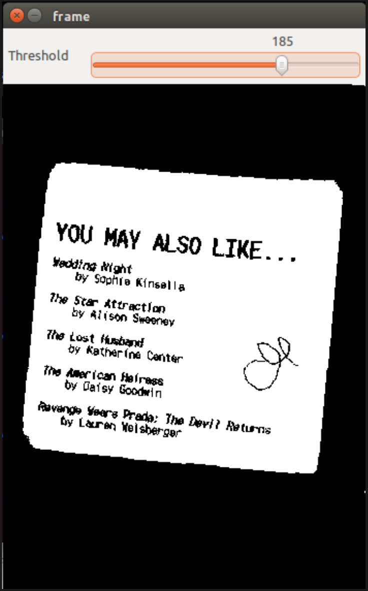
- For continuous model adaptation use:

```
fgmask = mog.apply(frame, learningRate=0.001)
```

Background Subtraction

```python
import numpy as np
import cv2

camera = cv2.VideoCapture('video.avi')
mog = cv2.createBackgroundSubtractorMOG2()

while True:
    grabbed, frame = camera.read()

    if not grabbed:
        break

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    gray = cv2.GaussianBlur(gray, (15, 15), 0)

    fgmask = mog.apply(frame, learningRate=0.001)

    fgmask = cv2.erode(fgmask, (7, 7), iterations=3)
    fgmask = cv2.dilate(fgmask, (7, 7), iterations=3)

    cv2.imshow("fgbg", np.hstack((gray, fgmask)))
    if cv2.waitKey(30) != -1:
        break
```

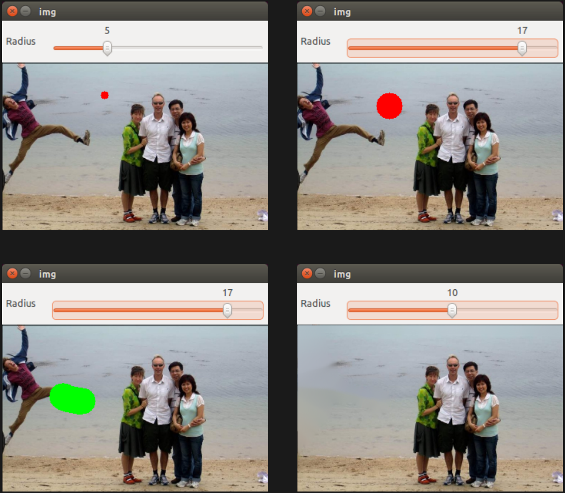
GUI with OpenCV: Trackbars



YOU MAY ALSO LIKE...
Wedding Night
by Sophie Kinsella
The Star Attraction
by Alison Sweeney
The Lost Husband
by Katherine Center
The American Heiress
by Daisy Goodwin
Revenge Wears Prada: The Devil Returns
by Lauren Weisberger

```python
import cv2

def foo(x):
    global img
    _, thrs = cv2.threshold(img, x,
                              255, 0)
    cv2.imshow('frame', thrs)


cv2.namedWindow('frame')
cv2.createTrackbar('Thrs', 'frame',
                    128, 255, foo)

img = cv2.imread('receipt.png', 0)
foo(128)
cv2.waitKey(0)
```

GUI with OpenCV: Mouse



```
cv2.setMouseCallback(wname, my_function)
my_function(event, x, y, flags, param)
cv2.EVENT_LBUTTONDOWN, cv2.EVENT_MOUSEMOVE, ...
```

scikit-image... skimage

scikit-image
image processing in python

- Open source project, free usage
- Seamless integration with OpenCV, numpy, scikit-learn, etc.

skimage: functionalities

- Color conversion.
- Exposure:
  - adjust gamme, histogram equalization, ...
- Feature extraction:
  - blobs, edges, corners, hog, lbp, ...
- Filters:
  - gabor, gaussian, sobel, thresholding, ...
- Graph:
  - graph cuts, hierarchical merging, shortest path, ...
- Restoration:
  - denoising, inpainting, ...
- Segmentation:
  - active contours, random walker, clustering, watershed, ...