

✓ 파이썬을 이용한 텔로 에듀 비디오 스트림 특징점 추출

✓ 텔로 에듀를 활용한 머신비전 분야의 하나인 특징점 추출에 대해 알아보기

- 텔로 에듀의 비디오 화상으로 수신되는 특정 대상물에 대하여 오픈소스 화상처리 패키지인 [OpenCV ORB 알고리즘을 이용하여 대상물의 특징점을 추출](#)

[+ 코드](#)[+ 텍스트](#)

✓ ORB(Oriented FAST and Rotated BRIEF)

- [특징점 검출 및 기술자\(descriptor\) 생성을 위한 알고리즘으로, 이미지에서 고유한 특징점을 찾고 이러한 특징점들을 설명하는 특징 기술자를 생성하는데 활용](#)
 - [FAST\(Features from Accelerated Segment Test\)와 BRIEF\(Binary Robust Independent Elementary Features\) 알고리즘을 기반으로 개발되었으며, 이 두 알고리즘의 장점을 조합하여 빠르고 강건한 특징점 검출 및 기술자 생성을 수행](#)
 - BRIEF는 특징점 검출을 지원하지 않는 디스크립터 추출기인데, BRIEF에 방향과 회전을 고려하도록 개선하고, 특징점 검출 알고리즘으로 FAST를 사용해 회전과 방향을 고려하도록 개선
 - [ORB라는 Feature를 사용](#)
 - ORB Feature는 계산하고 Matching하는데 매우 빠르다는 장점이 있음
 - 여러 Viewpoint에서 봤을 때 Invariance한 특성도 갖고 있음

- 객체 인식 및 이미지 매칭과 같은 컴퓨터 비전 애플리케이션에서 널리 사용

◦ 특징점을 기준으로 3차원 좌표 생성 등을 이용한 SLAM(Simulation Localization And Mapping)등에 활용할 수 있는 기본 자료로 사용

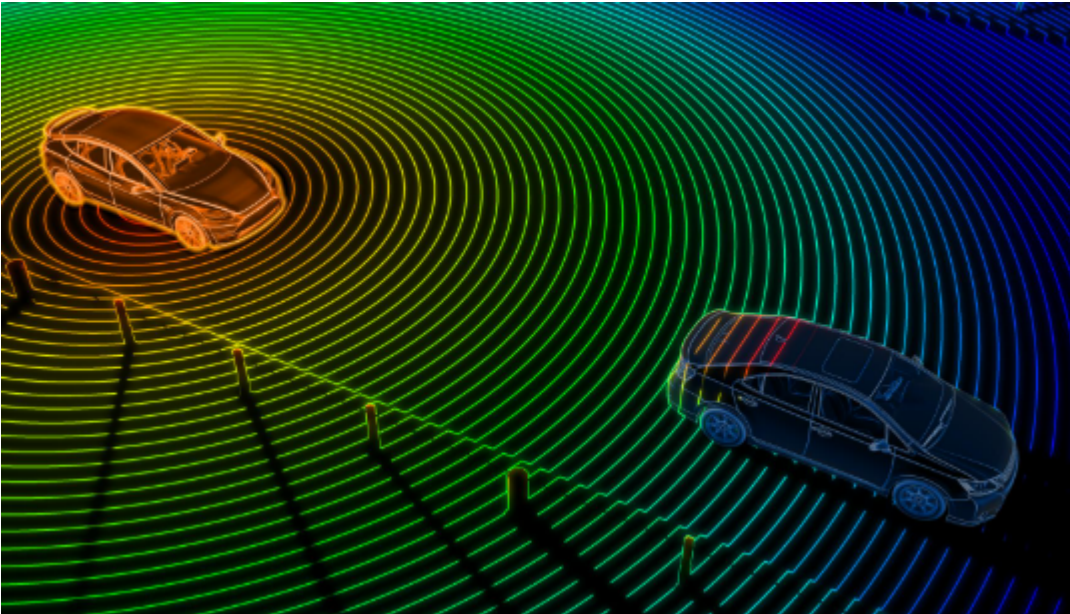
✓ OpenCV ORB (Oriented FAST and Rotated BRIEF)

```
1 from google.colab.patches import cv2_imshow
2 import numpy as np
3 import cv2
4
5 img = cv2.imread('/content/drive/MyDrive/드론코딩/드론코딩-15-파이썬을 이용한 텔로 에듀 비디오 스트림 특징점 추출/irinjpg.jpg')
6 img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
7 img2 = None
8
9 orb = cv2.ORB_create()
10
11 # ORB로 키포인트 / 디스크립터 찾기
12 kp, des = orb.detectAndCompute(img, None)
13
14 # 키포인트들의 위치만 나타낸다. 크기 / 방향 x
15 img2 = cv2.drawKeypoints(img, kp, img2, (0, 255, 0), flags=0)
16
17 cv2_imshow(img2)
18 cv2.waitKey(0)
19 cv2.destroyAllWindows()
```



✓ SLAM이란?

- 동시적 위치 추정 및 지도 작성을 의미 = 자신의 위치를 파악하며 맵 정보까지 만들겠다는 의미
 - 로봇이 주변 환경을 인식하고 스스로 위치를 파악하며 동시에 지도를 만들어가는 기술이라고 할 수 있음
 - 우리가 새로운 장소에 처음 방문했을 때, 주변을 둘러보며 어디에 있는지 파악하고 머릿속에 지도를 그리는 것과 비슷함
- SLAM 기술의 중요성
 - SLAM 기술은 자율주행 자동차, 드론, 로봇 청소기 등 다양한 분야에서 활용되고 있음
 - 이 기술이 없다면 로봇은 주변 환경을 정확하게 인식하지 못하고, 따라서 자율적인 움직임이 불가능할 것임
 - SLAM 기술은 로봇이 스스로 판단하고 행동하는 데 필수적인 기반 기술임
- <https://roytravel.tistory.com/405>



```
1 from IPython.display import YouTubeVideo, display
2 video = YouTubeVideo("mJRn41eIkLI", width=500)
3 display(video)
```



자율주행 로봇은 어떻게 주행하는가? SLAM편



✓ OpenCV 예제를 통한 ORB 알고리즘의 구현 과정 살펴보기

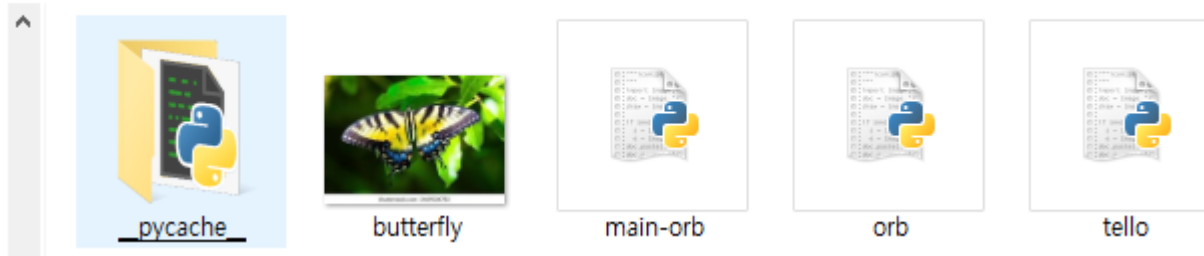
- C:\telloedu\Tello-ORB를 생성
- <http://www.gisdeveloper.co.kr/?p=6821>에서 예제를 복사하여 수정한 후 [orb.py](#)로 Tello-ORB 폴더에 저장
- 인터넷에서 나비 사진을 다운로드 후 butterfly.jpg로 Tello-ORB폴더에 저장

*orb - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
import numpy as np
import cv2
#filename = './data/butterfly.jpg'
filename = 'butterfly.jpg'
```

telloedu > Tello-ORB >



- 필요한 모듈 불러들이기 : import
- 그림 파일 입력 및 GRAY 변환 : cv2.imread
- OpenCV의 ORB 함수 적용 : orb.detectAndCompute
- 추출된 Keypoints 그리기 : cv2.drawKeypoints

```
1 import numpy as np
2 import cv2
3 from google.colab.patches import cv2_imshow
4 #filename = './data/butterfly.jpg'
5 filename='/content/drive/MyDrive/드론코딩/17-파이썬을 이용한 텔로 에듀 Objection Detection 컨트롤/butterfly.jpg'
6 #filename = 'butterfly.jpg'
7 img = cv2.imread(filename, cv2.COLOR_BGR2GRAY)
8 img2 = None
9 orb = cv2.ORB_create() # ORB 객체를 생성
```

```
10
11 kp, des = orb.detectAndCompute(img, None) #특징점 및 기술자를 계산
12 img2 = cv2.drawKeypoints(img, kp, None, (255,0,0), flags=0) #drawKeypoints() method in OpenCV to be able to draw the identified key points or
13
14 #cv2_imshow('img2', img2)
15 cv2.imshow('img2', img2)
16 cv2.waitKey(0)
17 cv2.destroyAllWindows()
```














SHUTTERSTOCK.COM - 2909039 / 03

✓ 텔로 에듀의 비디오 스트림을 수신하여 ORB 알고리즘을 이용하여 특징점 추출하기

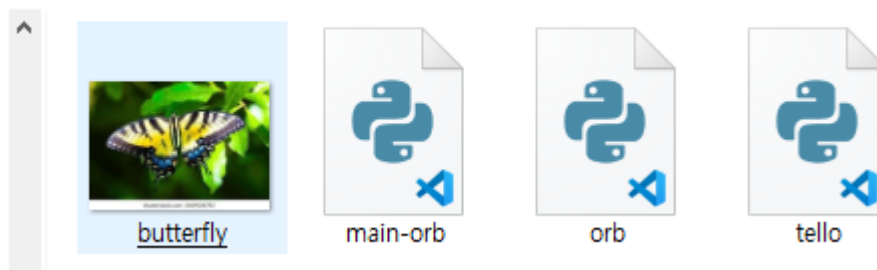
- <https://github.com/dji-sdk/Tello-Python>에서 Tello-Python-master.zip를 다운로드하여 C:\telloedu에 복사한 후 압축 해제
- C:\telloedu\Tello-Python-master\tello_video_dll(ForWin64)\tello_video_dll 폴더에 생성된 모든 파일을 C:\Python_All\Python27\Lib\site-packages에 복사

telloedu > Tello-Python-master > tello_video_dll(ForWin64) > tello_video_dll

이름	수정한 날짜
 avcodec-58.dll	2018-08-26 오전 4:36
 avdevice-58.dll	2018-08-26 오전 4:36
 avfilter-7.dll	2018-08-26 오전 4:36
 avformat-58.dll	2018-08-26 오전 4:36
 avutil-56.dll	2018-08-26 오전 4:36
 boost_python27-vc120-mt-x64-1_68.dll	2018-08-02 오후 7:53
 libh264decoder	2019-02-11 오후 9:03
 postproc-55.dll	2018-08-26 오전 4:36
 swresample-3.dll	2018-08-26 오전 4:36
 swscale-5.dll	2018-08-26 오전 4:36
 vcredist_x64	2019-02-13 오후 1:09

- 텔로 에듀의 비디오 스트림 수신 및 제어용 예제 프로그램을 https://github.com/hsgucci404/tello/tree/master/Tello_CV_core에서 `main.py`를 다운로드하여 `main-orb.py`로 변경한 후 `C:\telloedu\Tello-ORB`에 복사

telloedu > Tello-ORB



- 텔로 비디오 수신/제어용 `main-orb.py`를 수정
 - `time.sleep(0.5)` 하단에 다음 코드를 추가


```

drone = tello.Tello(' ', 8889, command_timeout=.01)

current_time = time.time() # 現在時刻の保存変数
pre_time = current_time # 5秒ごとの'command'送信のための時刻変数

time.sleep(0.5) # 通信が安定するまでちょっと待つ

#orb-1 initialization
orb = cv2.ORB_create()

#Ctrl+cが押されるまでループ
try:
    while True:

        # (A)画像取得
        frame = drone.read() # 映像を1フレーム取得
        if frame is None or frame.size == 0: # 中身がおかしかったら無視
            continue

        # (B)ここから画像処理
        image = cv2.cvtColor(frame, cv2.COLOR_RGB2BGR) # OpenCV用のカラー並びに変換
        small_image = cv2.resize(image, dsize=(480,360)) # 画像サイズを半分に変更

        #orb-2 convert to gray scale
        gray = cv2.cvtColor(small_image, cv2.COLOR_BGR2GRAY)
        gray = cv2.equalizeHist(gray)

        #orb-3 detect & compute orb
        kp, des = orb.detectAndCompute(gray, None)

        #orb-4 draw keypoints
        small_image = cv2.drawKeypoints(gray, kp, None, (255, 0, 0), flags=0)

```

```

1 #!/usr/bin/env python
2 # -*- coding: utf-8 -*-
3 import tello #tello.py 가져오기
4 import time #time.sleep을 사용
5 import cv2 # OpenCV를 사용

```

```
6
7 # 메인 함수
8 def main():
9     # Tello 클래스를 사용하여 drone이라는 인스턴스 (실체)를 만듦
10    drone = tello.Tello('', 8889, command_timeout=.01)
11
12    current_time = time.time() # 현재 시간의 저장 변수
13    pre_time = current_time    # 5초마다 'command' 전송을 위한 시간 변수
14
15    time.sleep(0.5)    # 통신이 안정 될 때까지 조금 기다리십시오.
16
17    #orb-1 초기화
18    orb = cv2.ORB_create()
19
20    #Ctrl+c를 누를 때까지 루프
21    try:
22        while True:
23
24            # (A) 이미지 획득
25            frame = drone.read()    # 영상을 1 프레임 획득
26            if frame is None or frame.size == 0:    # 내용이 이상하다면 무시
27                continue
28
29            # (B) 여기에서 이미지 처리
30            image = cv2.cvtColor(frame, cv2.COLOR_RGB2BGR)    # OpenCV 용 색상 및 변환
31            small_image = cv2.resize(image, dsize=(480,360) )    # 이미지 크기를 절반으로 변경
32
33            # orb-2 - gray scale 변환
34            gray = cv2.cvtColor(small_image, cv2.COLOR_BGR2GRAY)
35            gray = cv2.equalizeHist(gray)
36
37            # orb-3 - orb 감지 및 계산
38            kp, des = orb.detectAndCompute(gray, None)
39
40            # orb-4 - keypoints 그림
41            small_image = cv2.drawKeypoints(gray, kp, None, (255, 0, 0), flags=0)
42
```

```

43
44     # (X) 창에 표시
45     cv2.imshow('OpenCV Window', small_image)    # 윈도우에 표시하는 이미지를 바꾸면 여러가지 표시할 수 있음
46
47     # (Y) OpenCV 창에서 키 입력을 1ms 기다림
48     key = cv2.waitKey(1)
49     if key == 27:                                # k가 27 (ESC)이면 while 루프를 탈출하고 프로그램 종료
50         break
51     elif key == ord('t'):
52         drone.takeoff()                          # 이륙
53     elif key == ord('l'):
54         drone.land()                             # 착륙
55     elif key == ord('w'):
56         drone.move_forward(0.3)                  # 전진
57     elif key == ord('s'):
58         drone.move_backward(0.3)                 # 후진
59     elif key == ord('a'):
60         drone.move_left(0.3)                    # 왼쪽 이동
61     elif key == ord('d'):
62         drone.move_right(0.3)                   # 오른쪽 이동
63     elif key == ord('q'):
64         drone.rotate_ccw(20)                    # 왼쪽 소용돌이
65     elif key == ord('e'):
66         drone.rotate_cw(20)                     # 오른쪽 소용돌이
67     elif key == ord('r'):
68         drone.move_up(0.3)                       # 상승
69     elif key == ord('f'):
70         drone.move_down(0.3)                     # 하강
71
72     # (Z) 5 초마다 'command'를 보내고 사할 체크를 통과
73     current_time = time.time()# 현재 시간 얻기
74     if current_time - pre_time > 5.0 :           # 마지막 시간부터 5 초 이상 경과 했습니까?
75         drone.send_command('command')           # 'command' 제출
76         pre_time = current_time                  # 마지막 시간 업데이트
77
78 except( KeyboardInterrupt, SystemExit):        # Ctrl+c를 누르면 이탈
79     print( "SIGINTを検知" )

```

```
80
81     # tello 클래스 삭제
82     del drone
83
84
85 # "python main.py"로 실행되었을 때만 움직이는 처리
86 if __name__ == "__main__":      # import되면 "__main__"는 들어가지 않으므로 실행인지 import인지를 판단 할 수 있음
87     main()      # 메인 함수 실행
```

- C:\telloedu\Tello-Python-master\Tello_Video에 있는 [tello.py\(클래스 파일\)](#)를 C:\telloedu\Tello-ORB에 복사

tello - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
import socket
import threading
import time
import numpy as np
import libh264decoder
```

```
class Tello:
```

```
    """Wrapper class to interact with the Tello drone."""
```

```
    def __init__(self, local_ip, local_port, imperial=False, command_timeout=.3, tello_ip='192.168.10.1',
                  tello_port=8889):
```

```
        """
```

```
        Binds to the local IP/port and puts the Tello into command mode.
```

```
        :param local_ip (str): Local IP address to bind.
```

```
        :param local_port (int): Local port to bind.
```

```
        :param imperial (bool): If True, speed is MPH and distance is feet.
                                If False, speed is KPH and distance is meters.
```

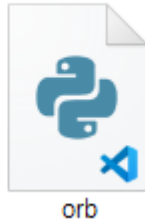
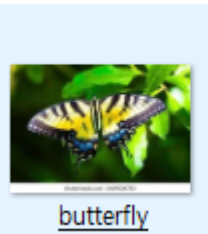
```
        :param command_timeout (int|float): Number of seconds to wait for a response to a command.
```

```
        :param tello_ip (str): Tello IP.
```

```
        :param tello_port (int): Tello port.
```

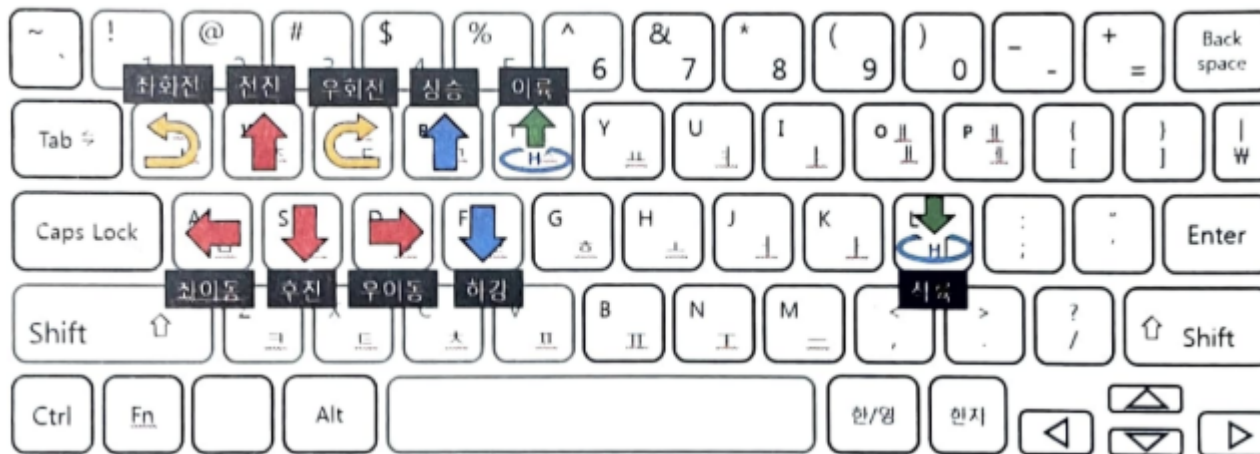
```
        """
```

telloedu > Tello-ORB



- 텔로 에듀를 [Wifi에 접속\(AP 모드\)하고 main-orb.py](#)를 실행
- 정상적으로 작동하면 상태표시창과 'OpenCV Window' 화면이 생성되어 [ORB 알고리즘 기반의 비디오 화면 화상처리에 의한 특징점이 추출됨](#)
 - [T 키를 입력하여](#) 텔로를 이륙시킴
 - 아래와 같이 수동으로 키 조작을 통해 적당한 거리와 고도로 이동
 - 기체의 이동 및 회전을 통해 다양한 장면에 대한 테스트를 마쳤으면 [L키를 입력하여](#) 착륙시킴
- 종료는 상태표시창을 닫거나 [ESC 키를 누름](#)

✓ [실행이 안될 경우 통신에 문제이니 지속적으로 반복 실행할 것!!!!](#)



C:\Python27\python.exe

```
[h264 @ 0000000035a9300] non-existing PPS 0 referenced
[h264 @ 0000000035a9300] non-existing PPS 0 referenced
[h264 @ 0000000035a9300] decode_slice_header error
[h264 @ 0000000035a9300] no frame!
[h264 @ 0000000035a9300] non-existing PPS 0 referenced
[h264 @ 0000000035a9300] non-existing PPS 0 referenced
[h264 @ 0000000035a9300] decode_slice_header error
[h264 @ 0000000035a9300] no frame!
[h264 @ 0000000035a9300] non-existing PPS 0 referenced
[h264 @ 0000000035a9300] non-existing PPS 0 referenced
[h264 @ 0000000035a9300] decode_slice_header error
[h264 @ 0000000035a9300] no frame!
[h264 @ 0000000035a9300] non-existing PPS 0 referenced
[h264 @ 0000000035a9300] non-existing PPS 0 referenced
[h264 @ 0000000035a9300] decode_slice_header error
[h264 @ 0000000035a9300] no frame!
[h264 @ 0000000035a9300] non-existing PPS 0 referenced
[h264 @ 0000000035a9300] non-existing PPS 0 referenced
[h264 @ 0000000035a9300] decode_slice_header error
[h264 @ 0000000035a9300] no frame!
[h264 @ 0000000035a9300] non-existing PPS 0 referenced
[h264 @ 0000000035a9300] non-existing PPS 0 referenced
[h264 @ 0000000035a9300] decode_slice_header error
```

OpenCV Window

