

Tellopy 및 opencv 간단한 주요함수 매뉴얼

1. 많이 쓰는 tello 라이브러리

Github에 많은 사람들이 만든 tello 라이브러리가 존재한다.

 hanyazou/TelloPy
DJI Tello drone controller python package
☆ 589 ● Python Updated on 6 Apr 2021

 dji-sdk/Tello-Python
This is a collection of python modules that interact with the Ryze Tello drone.
☆ 1.1k ● Python Updated on 8 Dec 2021

 damiafuentes/DJITelloPy
DJI Tello drone python interface using the official Tello SDK. Feel free to contribute!
☆ 735 ● Python MIT license Updated 16 days ago

위 그림은 대표적으로 3개를 캡처한 것이고 사용자들을 고려하여 가장 쉽게 사용할 수 있을 것 같은 TelloPy를 사용하게 되었다.

dji-sdk/Tello-Python은 정석적으로 tello sdk manual 처럼 쓸 수 있게 되어있고
send_command('takeoff')처럼 send_command 함수에 sdk 매뉴얼에 나와있는 command를 넣어주
게 돼있다.

damiafuentes/DJITelloPy는 TelloPy와 구조가 상당히 비슷하고, tello 객체를 선언하고 이 class에
선언된 함수를 쓰면 되는 부분이기 때문에 설명할 TelloPy와 상당히 비슷하다.

**참가팀 선발을 위해서 필자가 사용한 TelloPy 라이브러리에 대한 설명을 간단하게 제공하고 이에
대한 간단한 문제도 내려고 한다.** 후에 어떤 라이브러리를 사용해도 좋지만 이 매뉴얼을 읽고 진
행한다면 어떤 걸 쓰던 비슷한 구조이기 때문에 잘 알아볼 수 있을 것이라는 취지에 진행하는 것이기
에 오해 없으시길 바란다.

2. Tellopy

Tello drone controller를 제공하는 python 패키지인 tellopy를 소개하고자 한다. (많은 tello 관련
프로젝트에서 이를 사용하곤 한다.)

Github 주소는 아래와 같다.

<https://github.com/hanyazou/TelloPy>

- **pip install (additional)**

대부분 pip을 파이썬을 다운받을 때 같이 설치하였을 것이다. 혹시나 pip을 설치하지 않았다면 아래와 같은 방법으로 설치하길 바란다.

1. 터미널을 연다. (window에서는 검색창에 cmd를 치면 나오게 된다.)
2. curl <https://bootstrap.pypa.io/get-pip.py> -o get-pip.py 를 입력한다.

Python2 버전이나, python3 버전이나에 따라 다르게 되는데 pip은 python2 버전 사용자를 위한 패키지 설치 라이브러리이며 pip3는 python3 버전 사용자를 위한 패키지 설치 라이브러리이다. 파이썬 버전 확인은 터미널에 '**python -V**'라고 입력하면 나오게 된다. 버전이 2.XX 라면 python2 버전이며 3.xx라면 python3 버전이다.

3. Pip 설치

python2 get-pip.py → for python2 version

python3 get-pip.py → for python3 version

tellopy는 pip 설치가 돼 있다면 터미널에서 아래 명령을 쳐서 쉽게 받을 수 있다.

Pip(3) install tellopy

또는 github에서 clone 받아서 다운할 수도 있는데 이는 아래와 같다.

git clone https://github.com/hanyazou/TelloPy

cd TelloPy

python setup.py bdist_wheel

pip(3) install dist/tellopy-*.dev*.whl --upgrade

터미널 사용과 코드 작성을 동시에 하고 싶다면 **Visual studio code**를 사용하는 것을 추천한다.

설치 뒤에는 **import tellopy**를 통해 패키지를 쓸 수 있다.

내부 구성을 간단하게 알아보도록 한다.

develop-0.7.0	TelloPy / tellopy /	Go to file	Add file	...
Ryan Sprott Add comments. 1 c7e3949 on 2 Jul 2020 History				
..				
internal	Merge pull request #66 from apple-17/fastmode_control	3 years ago		
examples	Add comments.	2 years ago		
__init__.py	initial version	4 years ago		

TelloPy 파일 내에 소문자 (tellopy)라는 파일이 존재하고 여기에 들어가면 위 그림과 같이 뜨게 될 것이다. 여기서 '_internal' 파일 안에는 내부 함수가 정의되고 간단한 예제 코드는 'examples' 파일에 존재하게 된다.

- Examples file

develop-0.7.0	TelloPy / tellopy / examples /	Go to file	Add file	...
Ryan Sprott Add comments. 1 c7e3949 on 2 Jul 2020 History				
..				
__init__.py	initial version	4 years ago		
joystick_and_video.py	Add comments.	2 years ago		
keyboard_and_video.py	Delete -ffourcc avc1 args from mencoder command line (that does not w...	4 years ago		
record_log.py	Save log data into file	4 years ago		
simple_takeoff.py	Add internal state machine to recover the connection	4 years ago		
video_effect.py	Retry to open av stream in video_effect example	4 years ago		

Examples 파일에 들어가면 위 그림을 볼 수 있을 것이며 각 파이썬 파일은 아래와 같다.

joystick_and_video.py → 조이스틱 조종기와 비디오 예제

keyboard_and_video.py → 키보드 활용 조종기와 비디오 예제

record_log.py → log 데이터 저장 예제

simple_takeoff.py → 간단하게 이착륙 하는 예제

video_effect.py → tello가 촬영하고 있는 비디오를 pc에서 스트리밍 하는 예제

이번 매뉴얼에서 살펴볼 예제는 2개 (simple_takeoff.py, video_effect.py) 이며 이는 간단한 참가팀 선발 시험 범위에 들어가게 된다.

1. simple_takeoff.py

```
from time import sleep
```

```

import tellopy

def handler(event, sender, data, **args):
    drone = sender
    if event is drone.EVENT_FLIGHT_DATA:
        print(data)

def test():
    drone = tellopy.Tello()
    try:
        drone.subscribe(drone.EVENT_FLIGHT_DATA, handler)

        drone.connect()
        drone.wait_for_connection(60.0)
        drone.takeoff()
        sleep(5)
        drone.down(50)
        sleep(5)
        drone.land()
        sleep(5)
    except Exception as ex:
        print(ex)
    finally:
        drone.quit()

if __name__ == '__main__':
    test()

```

주요 코드는 진한 글씨로 된 부분이며 설명은 아래와 같다.

import tellopy → tellopy 라이브러리 호출한다.

drone = tellopy.Tello() → tellopy 내부에 정의 돼있는 Tello 객체를 drone이라는 변수로 선언한다.

drone.connect() → drone이라는 class에 선언 돼 있는 connect 함수를 통해 drone에 connection request를 보내게 된다.

drone.takeoff() → takeoff 함수를 통해 이륙 명령을 내린다.

drone.land() → land 함수를 통해 착륙 명령을 내린다.

sleep 함수는 몇 초동안 기다리게 하는 함수이다.

Subscribe는 FLIGHT_DATA를 받아와서 handler라는 함수를 거쳐 이 데이터를 터미널에 print 하겠다라고만 이해하면 된다. Publish/ subscribe 개념은 생략한다.

2. Video_effect.py

위 코드를 사용하기 위해서는 아래와 같이 터미널에서 pip 명령어를 사용하여 av, opencv, image 패키지를 다운 받아야 하며 이는 아래와 같다.

```
pip(3) install av
pip(3) install opencv-python
pip(3) install image
python(3) -m tellopy.examples.video_effect
```

- **code**

```
import sys
import traceback
import tellopy
import av
import cv2
import numpy
import time

def main():
    drone = tellopy.Tello()

    try:
        drone.connect()
        drone.wait_for_connection(60.0)

        retry = 3
        container = None
        while container is None and 0 < retry:
            retry -= 1
            try:
                container = av.open(drone.get_video_stream())
```

```

except av.AVError as ave:
    print(ave)
    print('retry...')

# red writing = 초반에 나오는 300 frame을 버리는 부분
frame_skip = 300
while True:
    for frame in container.decode(video=0):
        if 0 < frame_skip:
            frame_skip = frame_skip - 1
            continue
        start_time = time.time()
        image = cv2.cvtColor(numpy.array(frame.to_image()),
cv2.COLOR_RGB2BGR)
        cv2.imshow('Original', image)
        cv2.imshow('Canny', cv2.Canny(image, 100, 200))
        cv2.waitKey(1)
        if frame.time_base < 1.0/60:
            time_base = 1.0/60
        else:
            time_base = frame.time_base
        frame_skip = int((time.time() - start_time)/time_base)

except Exception as ex:
    exc_type, exc_value, exc_traceback = sys.exc_info()
    traceback.print_exception(exc_type, exc_value, exc_traceback)
    print(ex)
finally:
    drone.quit()
    cv2.destroyAllWindows()

if __name__ == '__main__':
    main()

```

마찬가지로 주요 코드는 진한 글씨로 나타내었다.

drone = tellopy.Tello() → Tello 객체 선언

drone.connect() → 드론 연결

container = av.open(drone.get_video_stream()) → get_video_stream 함수로 드론에서 제공하는 비디오 데이터를 받을 버퍼를 준비하고 이를 av.open 함수를 통해 열게 된다.

빨간색 글씨로 된 코드는 처음 300 프레임을 버리겠다는 의미이다.

그렇기 때문에 **동영상 나오는데 약간의 시간이 걸릴 수 있다**. 빨간색 코드를 없애도 큰 문제는 없을 것인데 본래 tello의 스트리밍 영상을 pc로 가져올 때 초반 frame은 안정적이지 않기 때문에 조금 기다리더라도 버리는 것도 좋은 방법이다.

왜 비디오 처리에 while True가 들어가는가?

우선 비디오는 정적 이미지의 연속된 파일이라고 생각하면 되기에 정적 이미지를 연속해서 계속 띄우기 위해서 while True를 넣는다고 생각하면 편하다.

image = cv2.cvtColor(numpy.array(frame.to_image()), cv2.COLOR_RGB2BGR) → frame이라는 변수는 container.decode(video=0)라는 것을 반복해서 담게되는 변수인데, 자세한 내용은 생략한다. **비디오 스트리밍을 위해 이 코드까지 수정할 내용은 없다! 이 image 변수를 영상처리 하면 되기 때문이다!**

위 코드를 간단하게 설명하자면 이미지 데이터를 numpy 라이브러리의 array 형으로 변환한 후에 기존에 RGB 순서대로 나왔던 데이터를 BGR 순서로 opencv의 cvtColor 함수로 바꿔주는 것이다.

Opencv는 기본적으로 RGB 순서가 아닌 BGR 순서를 사용한다.

cv2.imshow('Original', image) → 받아온 이미지를 화면에 띄우기 위한 함수이다.

'Original'은 영상을 띄울 창 이름이고 image는 띄울 이미지(numpy의 array 형)이다.

drone 객체는 위 말고 다양한 함수를 제공한다. 이제 앞에서 언급했던 '_internal' 파일에 들어가서 제공하는 함수를 살펴보도록 한다.

- **_internal 파일**

develop-0.7.0	TelloPy / tellopy / _internal /	Go to file	Add file	...
hanyazou Merge pull request #66 from appie-17/fastmode_control				
1 fb98e65 on 16 Nov 2019 History				
..				
init.py	initial version			4 years ago
crc.py	initial version			4 years ago
dispatcher.py	Fix import and except for python3			4 years ago
error.py	Add internal state machine to recover the connection			4 years ago
event.py	Add internal state machine to recover the connection			4 years ago
logger.py	Enable inheritance from 'Logger' class			3 years ago
protocol.py	Added support for drone attitude limit			3 years ago
state.py	Add internal state machine to recover the connection			4 years ago
tello.py	Merge pull request #66 from appie-17/fastmode_control			3 years ago
utils.py	Added support for drone attitude limit			3 years ago
video_stream.py	Fix: TypeError: unsupported operand type(s) for &: 'str' and 'int'			4 years ago

여기서 사용할 함수를 찾아보는 것은 tello.py 만 봐도 충분하다.

tello.py에는 다양한 함수가 정의돼 있다. 코드는 길기에 여기에는 첨부하지 않는다. 주요 함수만 간단히 설명하고 자세한 내용 또는 더 많은 함수는 tello.py 코드에서 참고하길 바란다.

get_video_stream() 함수 → Get_video_stream is used to prepare buffer object which receive video data from the drone.

connect 함수 → Connect is used to send the initial connection request to the drone.

Subscribe 함수 → Subscribe a event such as EVENT_CONNECTED, EVENT_FLIGHT_DATA, EVENT_VIDEO_FRAME and so on.

takeoff 함수 → Takeoff tells the drones to liftoff and start flying.

land 함수 → Land tells the drone to come in for landing.

quit 함수 → Quit stops the internal threads.

up 함수 → Up tells the drone to ascend. Pass in an int from 0-100.

down 함수 → Down tells the drone to descend. Pass in an int from 0-100.

forward 함수 → Forward tells the drone to go forward. Pass in an int from 0-100.

backward 함수 → Backward tells the drone to go in reverse. Pass in an int from 0-100.

right 함수 → Right tells the drone to go right. Pass in an int from 0-100.

left 함수 → Left tells the drone to go left. Pass in an int from 0-100.

clockwise 함수 → Clockwise tells the drone to rotate in a clockwise direction. Pass in an int from 0-100.

이 외에도 많은 함수들이 존재하게 된다.

```
drone = tellopy.Tello()
```

```
drone.connect()
```

함수를 선언했다면

drone.takeoff() 처럼 함수만 상속받아서 쓰면 된다. 함수에 받는 인자가 있는 경우에는 넣어줘야 한다!

기본적인 설명은 끝냈고 이번 대회에는 기본적인 영상처리를 해야하기 때문에 이에 대한 약간의 정보를 주고자 opencv에 대해 간단히 설명하고자 한다.

- **Opencv?**

Opencv는 컴퓨터 비전 task를 위해서 나온 오픈소스 라이브러리이다. 파이썬에서 간단하게 import cv2를 선언하여 쓸 수 있다. 이를 위해서는 opencv 패키지를 pip 명령어를 통해 다운받아야 하는데 이는 터미널에서 아래 명령을 쳐주면 된다.

```
pip or pip3 install opencv-python
```

```
pip or pip3 install opencv-contrib-python
```

contrib를 다운받게 되면 기본적인 opencv 함수 외에 더 많은 함수들도 사용할 수 있다.

기본적으로 이미지는 Height x width x channel 크기는 행렬로 표현이 되는데 이는 python에서 행렬연산을 위해 만들어진 numpy 라이브러리의 array 형으로 쉽게 표현할 수 있다.

Numpy의 array 형은 아래에서 자세한 정보를 얻을 수 있다.

<https://numpy.org/doc/stable/reference/generated/numpy.array.html?highlight=array#numpy.array>

다만 위에 video_effect.py에서 설명했듯이 image를 받아오는 것까지는 코드를 건드릴 필요가 없기 때문에 numpy를 깊게 공부할 필요는 없을 것이다.

- (참고) Opencv의 주요 모듈.

calib3d → 카메라 캘리브레이션 및 3차원 재구성 모듈

core → 행렬, 벡터 및 opencv 핵심 클래스와 연산 함수 모듈

dnn → 딥러닝 관련 모듈

features2d → 2차원 특징 추출 및 특징 벡터 및 매칭에 대한 모듈

flann → 다차원 공간에서 빠른 최근방 이웃 검색에 대한 모듈

highgui → 영상의 화면 출력 및 마우스 이벤트 처리 같은 사용자 인터페이스에 대한 모듈

imgcodecs → 영상 파일 입출력에 대한 모듈

imgproc → 필터링, 기하학적 변환, 색 공간 변환 등의 영상 처리 기능에 대한 모듈

ml → 머신 러닝 알고리즘에 대한 모듈

objdetect → 객체 검출에 대한 모듈

photo → 잡음 제거 같은 사진 처리 기능에 대한 모듈

stitching → 영상 이어 붙이기에 대한 모듈

video → optical flow 같은 동영상 처리에 대한 모듈

videoio → 동영상 파일 입출력에 대한 모듈

world → 통합 모듈

- Opencv의 기본적인 함수

cv2.imread(fileName, flag) : 이미지를 읽어오는 함수이다.

cv2.imshow(title, image) : 이미지를 화면에 출력하는 함수이다.

cv2.imwrite(fileName, image) : 이미지를 저장하는 함수이다.

cv2.waitKey(time) : 키보드 입력을 처리하는 함수이다.

cv2.destroyAllWindows() : 화면의 모든 창을 닫는 함수이다.

GaussianBlur : 가우시안 필터링 혹은 블러링 함수이다.

Sobel , Canny : 대표적인 에지 검출 알고리즘 함수이다.

HoughLines: 직선 검출 함수이다.

cvtColor: 색공간 변환 함수이다.

Threshold: 이진화 수행 함수이다.

등등이 있고 이미지 읽어오는 것들에 대한 기본적인 함수는 선발팀 테스트 시험에 나올 수 있다.

Opencv에 함수는 상당히 많고 각 예제에 대해 공식 홈 또는 구글에 많은 사람들이 수행한 경우가 많기 때문에 찾아보면서 응용하면 될 것이다.

문의 Instagram @2022falcon_challenger