Redirected Walking

231114 19101188 고은수

Camera Latency 줄이기

```python
cam1 = cv2.VideoCapture(cam_list[0])
cam2 = cv2.VideoCapture(cam_list[1])

while True:

    beforeRead = time.time()

    ret1, frame1 = cam1.read()
    ret2, frame2 = cam2.read()

    afterRead = time.time()

    print('read time : ' + str(afterRead - beforeRead))

    if not ret1:
        print("cam1 is not connected")
        break

    if not ret2:
        print("cam2 is not connected")
        break

    beforeShow = time.time()

    cv2.imshow("1", frame1)
    cv2.imshow("2", frame2)

    afterShow = time.time()

    print('show time : ' + str(afterShow - beforeShow))

    if cv2.waitKey(1) == 27:
        break
```

기존 방식 -> 약 1.9초정도 딜레이 발생

t2.py > ...

```python
import cv2
import threading

cam_list = ['rtsp://admin:dusrntlf123@192.168.0.133:554/Streaming/Channels/1', 'rts

cam_index = {
    'rtsp://admin:dusrntlf123@192.168.0.133:554/Streaming/Channels/1': 0,
    'rtsp://admin:dusrntlf123@192.168.0.138:554/Streaming/Channels/1': 1
}

cam1 = cv2.VideoCapture(cam_list[0])
cam2 = cv2.VideoCapture(cam_list[1])

def show_camera(cam, cam_name):
    while True:
        ret, frame = cam.read()
        if not ret:
            print(f"{cam_name} is not connected")
            break
        cv2.imshow(cam_name, frame)
        if cv2.waitKey(1) == 27:
            break

thread1 = threading.Thread(target=show_camera, args=(cam1, "Camera 1"))
thread2 = threading.Thread(target=show_camera, args=(cam2, "Camera 2"))
thread1.start()
thread2.start()

thread1.join()
thread2.join()

cv2.destroyAllWindows()
```

멀티 쓰레드 방식 -> 약 0.7초정도 딜레이 발생

Camera Latency 줄이기

```python
cam1 = cv2.VideoCapture(cam_list[0])
cam2 = cv2.VideoCapture(cam_list[1])

while True:

    beforeRead = time.time()

    ret1, frame1 = cam1.read()
    ret2, frame2 = cam2.read()

    afterRead = time.time()

    print('read time : ' + str(afterRead - beforeRead))

    if not ret1:
        print("cam1 is not connected")
        break

    if not ret2:
        print("cam2 is not connected")
        break

    beforeShow = time.time()

    cv2.imshow("1", frame1)
    cv2.imshow("2", frame2)

    afterShow = time.time()

    print('show time : ' + str(afterShow - beforeShow))

    if cv2.waitKey(1) == 27:
        break
```

버퍼 사이즈 줄이기

```python
cam1.set(cv2.CAP_PROP_BUFFERSIZE, 1)
cam2.set(cv2.CAP_PROP_BUFFERSIZE, 1)
```

1번 방식에서 약 1.9초 -> 약1.7초로 0.2초정도 딜레이 감소

Camera Latency 줄이기

-Bot-sort에 적용

```
def detection_thread(cam_num):
    detection(cam_num)
```

```
def detect(cam_num):
    global coordinates
    global x_average,y_average
    weights, view_img, save_txt, imgsz, trace = opt.weights, opt.view_img, opt.save_txt, opt.img_size, opt.trace
    if cam_num==1:
        source = cam_list[0]
    else:
        source = cam_list[1]
```

```
# parser.add_argument('--source', type=str, default='inference/images', help='source')  # file/folder, 0 for webcam
```

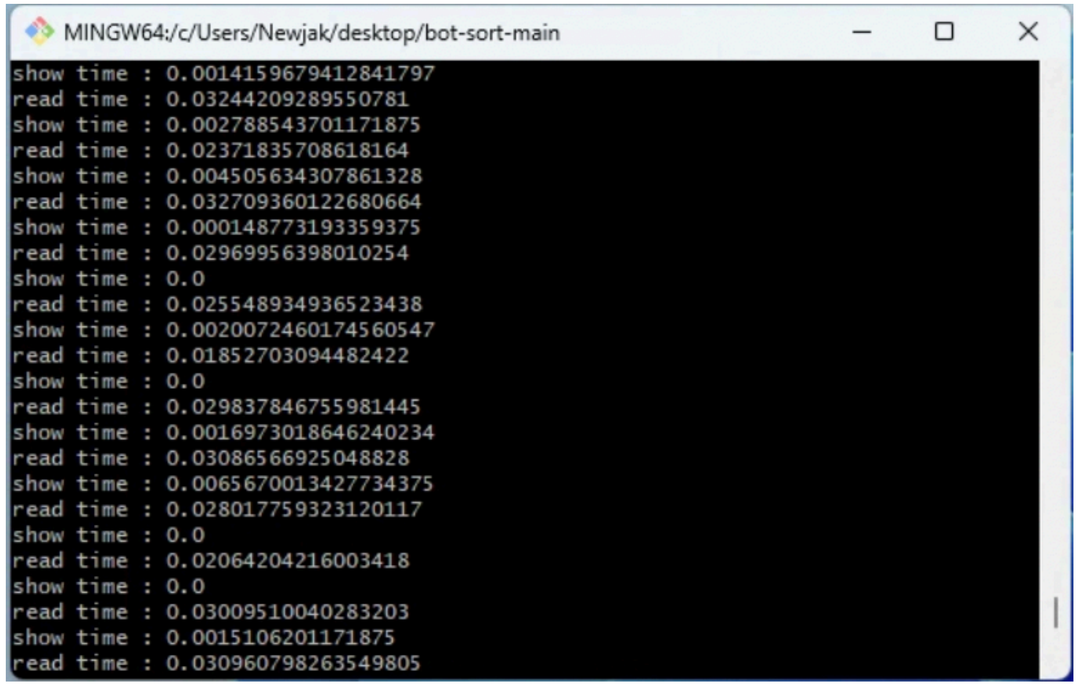
```
cam1 = threading.Thread(target=detection_thread, args=(1,))
cam2 = threading.Thread(target=detection_thread, args=(2,))

cam1.start()
cam2.start()
```

```
cap = cv2.VideoCapture(url)
cap.set(cv2.CAP_PROP_BUFFERSIZE, 1)
```

```
self.cap = cv2.VideoCapture(pipe)  # vide
self.cap.set(cv2.CAP_PROP_BUFFERSIZE, 1)
```

```
self.cap = cv2.VideoCapture(path)
self.cap.set(cv2.CAP_PROP_BUFFERSIZE, 1)
```

Camera Latency 줄이기



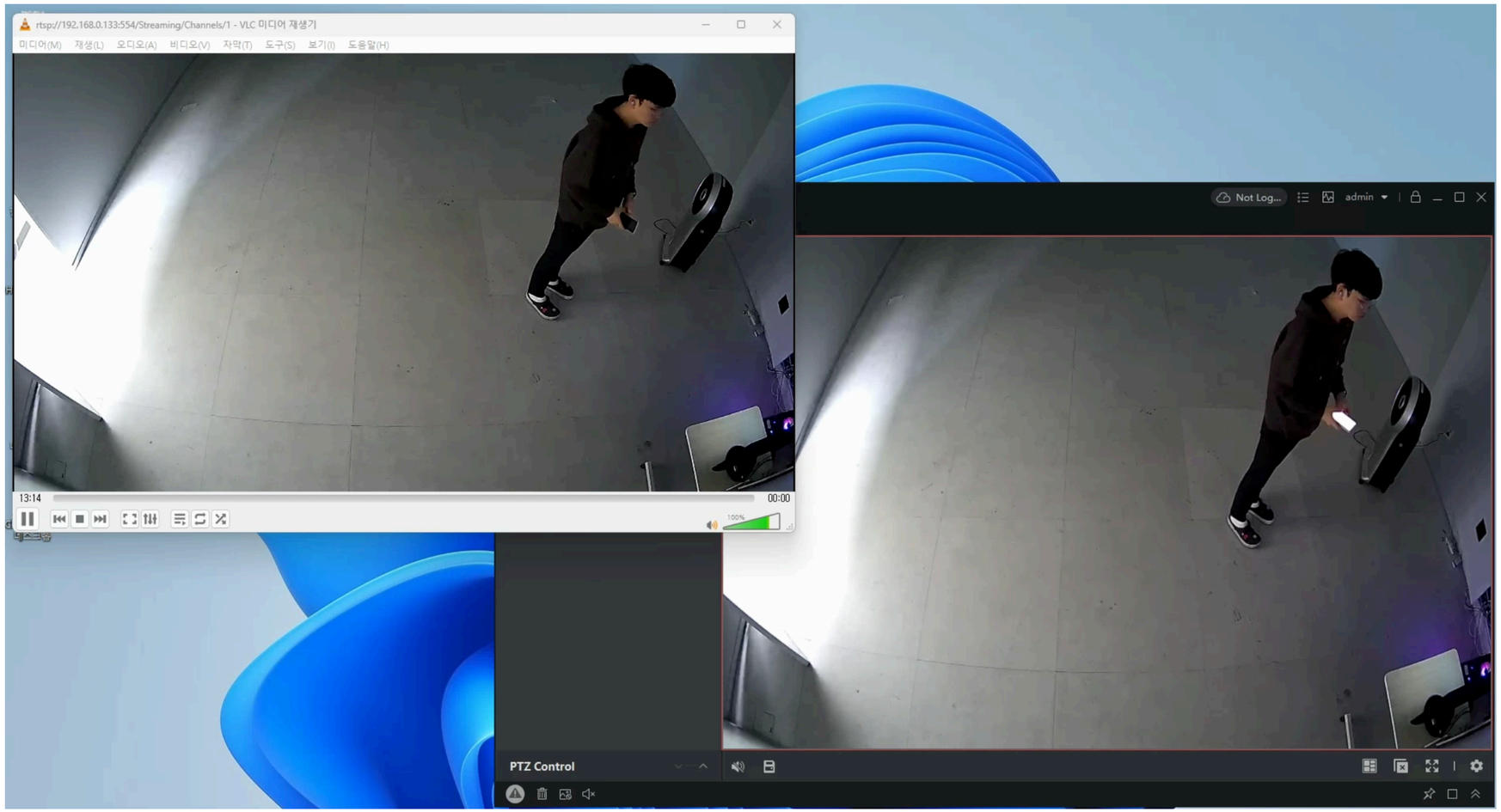
1번(기존) 방식에서 두 카메라의 프레임을 읽어오는 데 걸리는 시간 : 약 0.03초 내외

매 프레임을 화면에 출력하는 데 걸리는 시간 : 약 0.002초 미만

=> 네트워크 속도, 카메라 자체 레이턴시의 비중이 크다고 판단

Camera Latency 줄이기

-순수하게 카메라로부터 프레임 받아오는 지연이 얼마나 되는지?



VLC 플레이어 RTSP 스트리밍 - 약 1.2초

IVMS - 약 0.2초

감사합니다