USB Camera program

USB 카메라 구동 프로그램

```
import os
import sys
import time
import math
import getopt
import numpy as np
import cv2
import threading
import subprocess
from collections import deque
from lock_manager import Lock_Manager
from util import Util

class UsbWebCamera(threading.Thread):
  def __init__(self, fps=10, video_source=0, source=None):

    threading.Thread.__init__(self)

    self.name = self.__class__.__name__
    self.archive = os.path.join(os.path.dirname(os.path.realpath(__file__)), 'archive')

    self.current_frame = None

    self.codec = cv2.VideoWriter_fourcc('M','J', 'P', 'G')
    self.OBSERVER_LENGTH = 5 # Time in seconds to be observed for motion
    self.threshold = 15

    self.CAMERA_SOURCE = video_source
    self.REMAIN_RECORDING_FILES = 10 # 10이상 부터 삭제 후 저장

    self.source = cv2.VideoCapture(self.CAMERA_SOURCE) if source is not None else self.init_camera()

    #self.fps = self.find_fps(self.source)
    self.fps = fps
    Util.log(self.name, "Setting FPS to " + str(self.fps))
    self.source.set(cv2.CAP_PROP_FPS,self.fps)
    #self.height, self.width = self.get_dimensions(self.source)
    Util.log(self.name, "Initializing pi camera class with video_source=" + str(self.CAMERA_SOURCE))
    #Util.log(self.name,"width: {"+str(self.width)+"}, height : {"+str(self.height)+"}")
    self.lock_manager = Lock_Manager("motion")

  def __del__(self):
    self.source.release()
```

```
    cv2.destroyAllWindows()
    self.lock_manager.remove()

  def get_captureFrame(self):
    return self.current_frame if self.current_frame is not None else None

  def get_frame(self):
    return self.frame_to_jpg(self.current_frame) if self.current_frame is not None else None

  def frame_to_jpg(self, frame):
    ret, jpeg = cv2.imencode('.jpg', self.current_frame)
    return jpeg.tobytes()

  def get_dimensions(self, source):
    frame = cv2.cvtColor(source.read()[1],cv2.COLOR_RGB2GRAY)
    return frame.shape[0: 2]

  def find_fps(self, source):
    Util.log(self.name, "Determining FPS...")
    num_frames = 120
    start = time.time()
    for i in range(0, num_frames):
      ret, frame = source.read()
    end = time.time()
    fps = int(math.floor(num_frames / (end - start)))
    Util.log(self.name, "Setting FPS to " + str(fps))
    return fps

  def init_camera(self):
    camera = cv2.VideoCapture(self.CAMERA_SOURCE)
    time.sleep(0.5)

    return camera

  def run(self):
    while True:
      (grabbed, self.current_frame) = self.source.read()
      # End of feed
      if not grabbed:
        break
```

Revision #2
★Created 10 June 2023 12:52:46 by Hyeon Su Ryu
Updated 10 June 2023 13:24:51 by Hyeon Su Ryu