

CSI 4133 - Lab 03

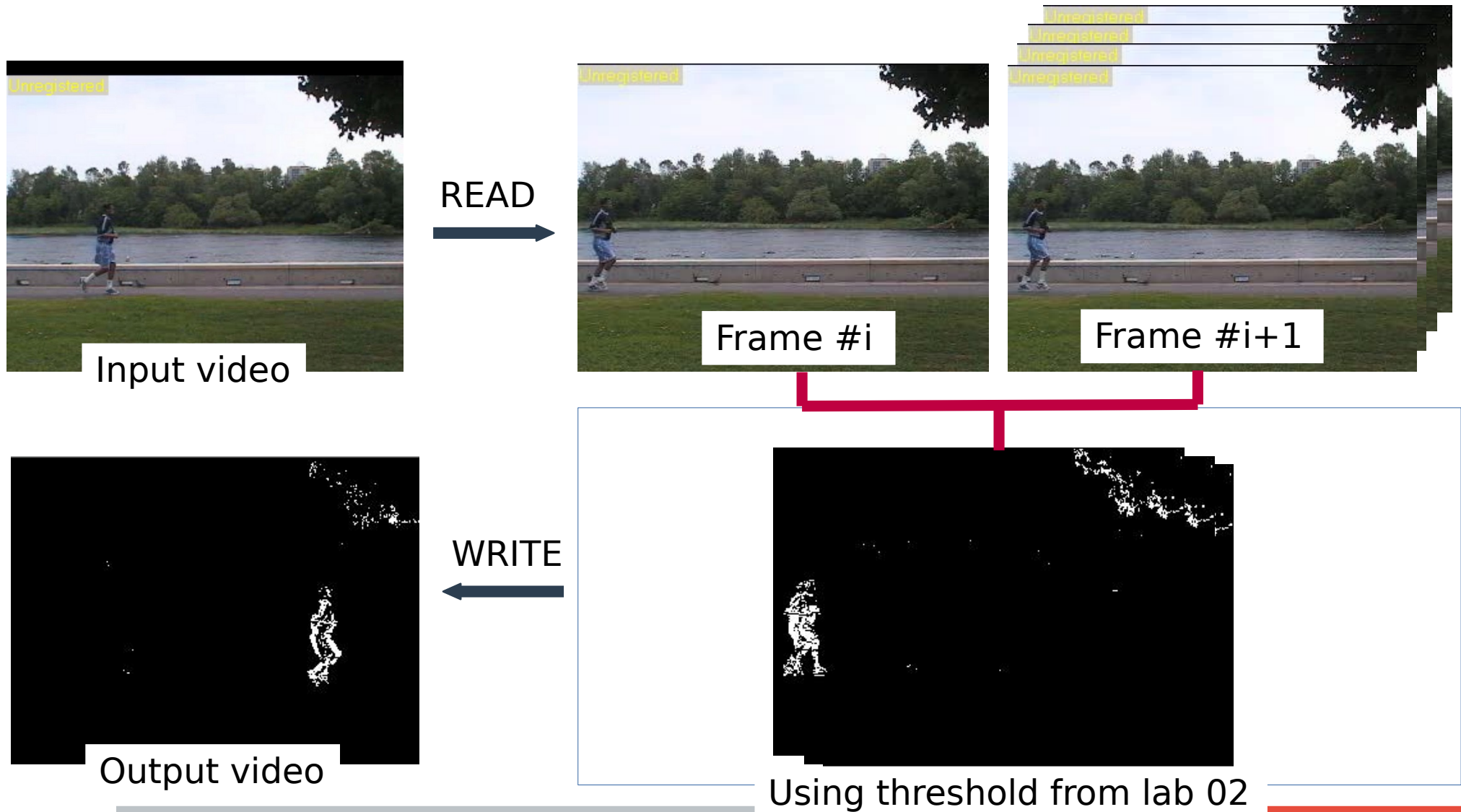
Calculating the difference between two consecutive video frames

Contents

Highlight moving objects from a video and save the resulting video to disk

- Read the input video
- Process the frames
- Write the output video

Procedure



Read video

- Initializing capture from a camera:
 - `cv2.VideoCapture(0); // open the default camera`
- Initializing capture from a video file:
 - `cv2.VideoCapture("video.avi"); // open the video file`

Get capture & frame properties

- `cv2.VideoCapture.get(propID)`

Property identifier. It can be one of the following:

- **CV_CAP_PROP_POS_MSEC** Current position of the video file in milliseconds or video capture timestamp.
- **CV_CAP_PROP_POS_FRAMES** 0-based index of the frame to be decoded/captured next.
- **CV_CAP_PROP_POS_AVI_RATIO** Relative position of the video file: 0 - start of the film, 1 - end of the film.
- **CV_CAP_PROP_FRAME_WIDTH** Width of the frames in the video stream.
- **CV_CAP_PROP_FRAME_HEIGHT** Height of the frames in the video stream.
- **CV_CAP_PROP_FPS** Frame rate.
- **CV_CAP_PROP_FOURCC** 4-character code of codec.
- **CV_CAP_PROP_FRAME_COUNT** Number of frames in the video file.
- **CV_CAP_PROP_FORMAT** Format of the Mat objects returned by `retrieve()`.
- **CV_CAP_PROP_MODE** Backend-specific value indicating the current capture mode.
- **CV_CAP_PROP_BRIGHTNESS** Brightness of the image (only for cameras).
- **CV_CAP_PROP_CONTRAST** Contrast of the image (only for cameras).
- **CV_CAP_PROP_SATURATION** Saturation of the image (only for cameras).
- **CV_CAP_PROP_HUE** Hue of the image (only for cameras).
- **CV_CAP_PROP_GAIN** Gain of the image (only for cameras).
- **CV_CAP_PROP_EXPOSURE** Exposure (only for cameras).
- **CV_CAP_PROP_CONVERT_RGB** Boolean flags indicating whether images should be converted to RGB.
- **CV_CAP_PROP_WHITE_BALANCE_U** The U value of the whitebalance setting (note: only supported by DC1394 v 2.x backend currently)
- **CV_CAP_PROP_WHITE_BALANCE_V** The V value of the whitebalance setting (note: only supported by DC1394 v 2.x backend currently)
- **CV_CAP_PROP_RECTIFICATION** Rectification flag for stereo cameras (note: only supported by DC1394 v 2.x backend currently)
- **CV_CAP_PROP_ISO_SPEED** The ISO speed of the camera (note: only supported by DC1394 v 2.x backend currently)
- **CV_CAP_PROP_BUFFERSIZE** Amount of frames stored in internal buffer memory (note: only supported by DC1394 v 2.x backend currently)

Set capture & frame properties

- `cv2.VideoCapture.set(propID, set)`

Property identifier. It can be one of the following:

- **CV_CAP_PROP_POS_MSEC** Current position of the video file in milliseconds.
 - **CV_CAP_PROP_POS_FRAMES** 0-based index of the frame to be decoded/captured next.
 - **CV_CAP_PROP_POS_AVI_RATIO** Relative position of the video file: 0 - start of the film, 1 - end of the film.
 - **CV_CAP_PROP_FRAME_WIDTH** Width of the frames in the video stream.
 - **CV_CAP_PROP_FRAME_HEIGHT** Height of the frames in the video stream.
 - **CV_CAP_PROP_FPS** Frame rate.
 - **CV_CAP_PROP_FOURCC** 4-character code of codec.
 - **CV_CAP_PROP_FRAME_COUNT** Number of frames in the video file.
 - **CV_CAP_PROP_FORMAT** Format of the Mat objects returned by `retrieve()`.
 - **CV_CAP_PROP_MODE** Backend-specific value indicating the current capture mode.
 - **CV_CAP_PROP_BRIGHTNESS** Brightness of the image (only for cameras).
 - **CV_CAP_PROP_CONTRAST** Contrast of the image (only for cameras).
 - **CV_CAP_PROP_SATURATION** Saturation of the image (only for cameras).
 - **CV_CAP_PROP_HUE** Hue of the image (only for cameras).
 - **CV_CAP_PROP_GAIN** Gain of the image (only for cameras).
 - **CV_CAP_PROP_EXPOSURE** Exposure (only for cameras).
 - **CV_CAP_PROP_CONVERT_RGB** Boolean flags indicating whether images should be converted to RGB.
 - **CV_CAP_PROP_WHITE_BALANCE_U** The U value of the whitebalance setting (note: only supported by DC1394 v 2.x backend currently)
 - **CV_CAP_PROP_WHITE_BALANCE_V** The V value of the whitebalance setting (note: only supported by DC1394 v 2.x backend currently)
 - **CV_CAP_PROP_RECTIFICATION** Rectification flag for stereo cameras (note: only supported by DC1394 v 2.x backend currently)
 - **CV_CAP_PROP_ISO_SPEED** The ISO speed of the camera (note: only supported by DC1394 v 2.x backend currently)
 - **CV_CAP_PROP_BUFFERSIZE** Amount of frames stored in internal buffer memory (note: only supported by DC1394 v 2.x backend currently)
- value** – Value of the property.

Write/Save video

- Initializing a video writer:
 - `cv2.VideoWriter([filename, fourcc, fps, frameSize[, isColor]]);`
 - Filename – Name of the output video file.
 - Fourcc – 4-character code of codec used to compress the frames
 - Example, `CV_FOURCC('P','I','M','1')` is a MPEG-1 codec, `CV_FOURCC('M','J','P','G')` is a motion-jpeg codec, etc.
 - Fps – Frame rate of the created video stream.
 - FrameSize – Size of the video frames.
 - IsColor – If it is not zero, the encoder will expect and encode color frames, otherwise it will work with grayscale frames (flag is currently only supported on windows)
- Write/Save video
 - Once you have a frame (frm) ready, you can write the output video file

```
Result = cv2.VideoWriter(para1,para2,para3,para4)
Result.write(frm)
```

Task

Goal: Highlight moving objects from a video and save the resulting video to disk.

Idea:

1. Load input video “park.avi” (in the file folder “video”)
2. Obtain two consecutive frames
3. Calculate the pixel intensity difference between the two consecutive frames
4. Perform thresholding on the difference image to get areas of movement in binary format
5. Save the resulting frame in a new video (eg: “result.avi”)
6. Repeat the step2-step5 until the last frame of the input video.

Task

Hints:

- Read the input video:
 - Open the input video.
 - Obtain the properties of the video, such as the frame size, the number of frames, etc.
 - Obtain the index of the frame (0-based index).
 - Obtain the data of each frame.
- Write the output video:
 - Create the output video.
 - Save processed frames into the output video.
- Thresholding:
 - Much of your code from Assignment 1 is reusable here
 - The threshold value from Assignment 1 is used here.

Task

Please submit a **lab report**, **source code**, **screenshots** of your results, and **output videos**.

Task

Examples



END

THANK YOU