



COMPUTER VISION AND IMAGE PROCESSING

LAB SESSION 3

WORKING WITH VIDEO STREAMS

DR. FEDERICO TOMBARI

Streaming video – openCv 1.0



- `CvCapture *stream = cvCaptureFromAVI (char *filename);`
- `CvCapture *stream = cvCaptureFromCAM (int camID);`

- CvCapture represents a video stream (from either a file or a device such as a webcam)
- camID usually «counts» the number of cameras plugged to the computer (starting from 0)

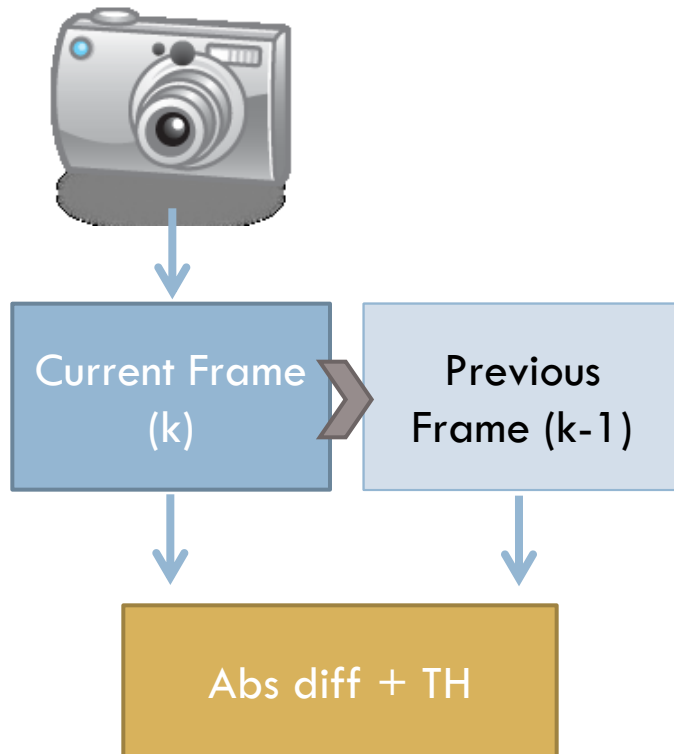
- To get the current frame:
 - `int res = cvGrabFrame(stream);`
 - `IplImage *frame= cvRetrieveFrame(stream);`
- Or
 - `IplImage *frame = cvQueryFrame(stream);`

Streaming video – openCv 2.2+



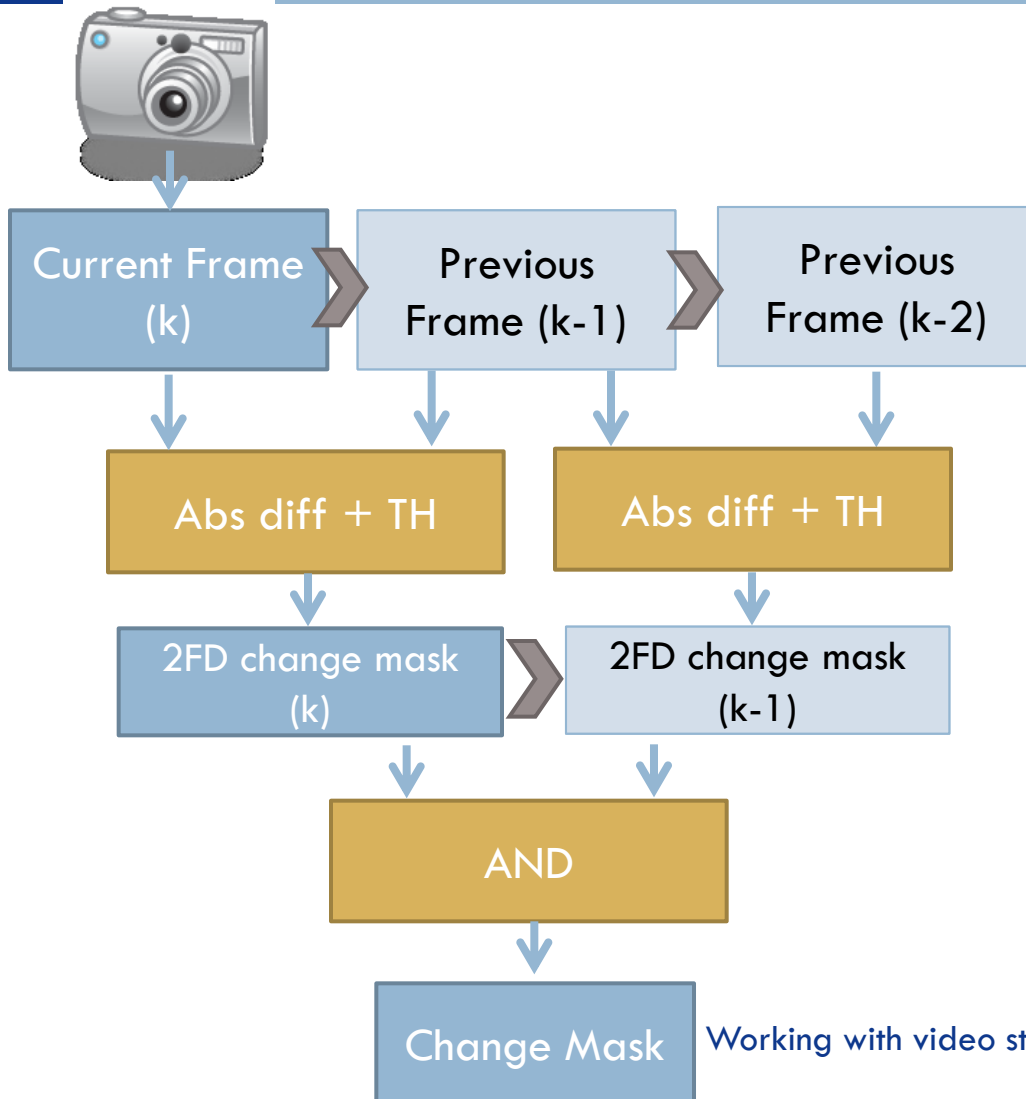
- C++ interface with class **cv::VideoCapture**
- The stream can be initialized directly on construction (alternatively using the **open()** method)
 - `cv::VideoCapture::VideoCapture (int camID);`
 - `cv::VideoCapture::VideoCapture (string &filename);`
 - Eg.: `VideoCapture stream(0);`
- To get the current frame:
 - `Cv::Mat frame;`
 - `stream >> frame;`
 - Alternatively, using *the read()* or the **grab()+retrieve()** methods
- **set()** and **get()** to read and set the properties of the device (when supported)

Two-frame difference



- Pixelwise absolute difference
- Thresholding
- At the end of each loop, the current frame becomes the previous one:
 - `cvCopyImage(previous, current);`
- **Alternatively:**
 - `Iplimage *temp = previous;`
 - `previous = current;`
 - `current = temp;`

Exercise – 3-frame difference



- Swapping pointers can be used for
 - updating the two stored «previous» frames
 - updating the stored two-frame-difference change mask