

IRIS DETECTION

November 21, 2016

ALGORITHM

The whole algorithm can be summarised into two basic steps

1. Hierarchical localization of the location of each eye (LOCALIZATION)
2. Detection of iris in the localized eyes (DETECTION)

LOCALIZATION

Main steps:

1. Detect face (if there is a full face in the video)
2. Detect pair of eyes (now enabled only for videos as the accuracy of eye pair detection using haar cascades is very less. As opposed to images, localization of eyes is critical in videos)
3. Detect left and right eyes separately

Other points:

1. Haar cascades used for face detection, eye-pair detection, and eye detection mainly because it is faster (sacrificing accuracy a bit especially for eye pair detection and individual eye detection steps)
2. Gaussian smoothing is employed in many stages especially to prevent false detections due to noise
3. Histogram equalization makes the algorithm invariant to lighting variations

DETECTION

1. Thesholding and median filtering used to extract region of extreme interest for blob detection
2. Biggest blob detected in the region of extreme interest
3. Size of the detected blob will be automatically adjusted according to the size of the largest keypoint
4. Hough transform based circle detection has also been tried out. For the current dataset, blob detection gives better results

FUTURE WORK POSSIBILIIES

1. For videos, implement interleaved detection and tracking. This will reduce the need for detection in each frame. Detection will only be triggered only when the tracker looses confidence beyond a limit
2. For images, get rid of haar cascades based localization and tryout something more efficient especially for localization step.
3. Better Iris detection by fitting an ellipse
4. Learn an average template of the eye from a database and try template matching based logic to improve eye detection accuracy

Readme

1. Detection result: left (on the screen) eye iris : blue circle, right eye iris : red circle
2. Current algorithm works for webcam video, offline videos (tested on .mp4,. mov) and images (tested on .png. .jpg, .raw)
3. Ubuntu 14.04, C++, Opencv 2.4.9
4. DEBUG/VERBOSE mode also available (settings.h)
5. The algorithm works very fast mainly because Haar cascades are used