**CVLab Walkthrough 1 – Failure Detection Methods in Correlation Tracking and Estimation with Kalman Filter**

The purpose of this walkthrough is to explore methods for detecting failure in correalation tracking. This can be observed in this scenario when the tracking rectangle erroneosuly tracks to an unexpected location in the window.

Failure detection is a problem in correlation tracking. In many cases manual input parameters are required to distinguish between a successful and unsucessful track. These manual input parameters can vary depending on noise, template size and other characteristics of images.

In additon, once failure is detected, this walkthough demonstrates how the Kalman estimator maintains tracking despite occlusions, based on inertia.

Requirements:

Connected and functioning web cam

CVLab executable for Windows (or the project built from github)

Your face

Note: Many of the windows in this software are managed by OpenCV and which does not provide a means of detecting when the window is closed. So clicking the top right x in some cases does not work, as the note on the application states, you must untoggle (Display Video) to close the window.

# Setting a Template

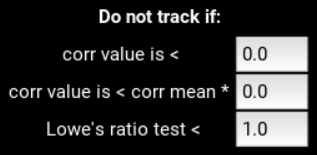
1. Make sure camera is selected (default settings) and your camera is plugged in and functioning.
2. **Click Display Video.** A video window depicting the camera output should appear.
3. **Press ‘t’ to set template – click and drag a box from your temple to your opposite cheek** (example below)



# Setting a Threshold

One of the problems in image recognition is detecting when a correlation match fails. The next few sections will explore ways in which we can detect and handle failures.

1. The next few steps will set a threshold value so the system will know when face detection is lost. This threshold value can vary depending on many conditions in the image space.
2. **Click the** **Max Corr Chart** toggle in the output column.
3. A chart will appear with populating values. Keep an eye on this chart while you **occlude your face with a paper**. The correlation values should drop significantly when your face is occluded from the camera and the green box may jump to another part of the video.
4. Choose a value just a bit higher than the lowest correlation value was when your face was occluded**, then enter this threshold value (corr value is <)** in the Algorithm column (depicted below), and **press enter**. Note: the value is comitted and changes are applied when enter is pressed.



Note that this process requires manual testing and entry. The following experimental approaches use ratios to become more general and therefore could remove this manual process.

1. **Untoggle the Max correlation and close the plot (top right x).**

# Setting a Correlation Ratio to Mean

1. **Set the (corr value is <) field back to zero and press enter**. The next field (corr value is < corr mean \*), allows the user to specify a ratio, so when the current correlation value falls below a ratio value of the mean of all previous max correlation values before it, tracking will not occur.
2. **Enter .85 in the (corr value < corr mean \*) field and press enter**. This process was better than before, but still susceptible to varying conditions in the image space. Since its based off the mean and considers all of the states before it, it doesn’t adapt to recent changes to the image space. .85 may not be ideal in all situations as well. Test different values to improve results.

# Lowe’s Ratio Test for correlation

1. **Set the (corr value < corr mean \*) back to zero and press enter**. The Lowe’s ratio test was originally designed for feature matching. The idea is simple. Calculate the ratio between 2nd best match to the highest best match and filter out matches whose values are close to 1. In matching features, this technique works well in filtering out bad matches. I have not seen this process attempted in correlation tracking before, so I tried it. Instead of match similarity, I use max correlation values. Because the characteristics of max correlation maps generally has a circular peak (turn on the correlation map in the Display column to see), a filled black circle is placed on the highest correlation value, then the second highest correlation value is found. This is to eliminate the common case where the 2nd highest correlation value may be very close to the first. See lowes\_ratio\_test.py for more details on the algorithm.
2. **Set (Lowe’s ratio test <) to .95 and press enter**. I found this value worked very well in preventing the system from bad matches, though still not perfect.

# Auto Detect Failure with an additional Lowe’s ratio test and Kalman filter (newly created method)

1. **Set (Lowe’s ratio test <) back to 1.0 and press enter.** Details on the implementation of this new method are posted in the blog: <https://garretmoore.wordpress.com/>
2. **Click on (Auto detect failure Lowes) toggle.**
3. Failure should now be automatically detected, without having to set a value. Failure is depicted by not drawing the rectangle on the screen. In general, this method detects failure when the peak correlation area falls just below the rest of the image.

# Estimation with the Kalman Filter

1. With any one of the failure detection methods turned on, turn on the (Kalman with Inertia) toggle and observe the behavior. When tracking is lost, the rectangle turns yellow until the face is detected again. The inertia of the last movement is maintained.
2. You can also turn on the **correlation map**, which is a map of all correlation values in the frame. It’s greyscaled where white represents the highest values and black represent the lowest correlation values.