

Computer Vision Project

1. Mark object/Give object position
 - a. Input: 1st frame
 - b. Use mouse clicks and the ginput function
 - c. Outputs the top-left and bottom-right corner of the bounding box of the ellipse
 - d. function [tl, br] = getObjectPosition(frame1)
2. Compute distribution of the object-window - (16 bins per channel - leave as a parameter)
 - a. Takes the n_h pixels inside the window (ellipse inside the rectangle) and computes a histogram
 - b. Input: the rectangular window - the array of pixel values inside the object-window
 - c. Use Epanechnikov kernel to compute the distribution
 - d. Output: The distribution - a 16x16x16 array -- leave the 16 as a parameter
 - e. function hist = computeDistribution(window)
3. Function to evaluate Bhattacharya coefficient
 - a. Input: 2 distributions
 - b. Compute the Bhattacharya coefficient using formula (17) from paper
 - c. function coeff = computeBhattacharyaCoefficient(window1, window2)
4. Function to compute mean-shifted position - (1 iteration of while)
 - a. Input: distributions q_u (from frame i) and p_u(y_0) (from frame i+1), frames i and i+1
 - b. Compute the mean-shifted position using formulae (26) and (25).
 - c. Output the mean-shifted position
 - d. function shiftedPos = computeMeanShiftPosition(frame1, q_u, frame2, p_u)
5. Function to mark the tracked object
 - a. Inputs: frame, boundingbox of ellipse
6. Main function
 - a. Get input, object positions
 - b. For each frame i
 - i. computeDistributionq_u for frame i at y0i with size = original size_i
 - ii. computeDistribution p_u for frame i+1 at y0i with size = 90%, 95%, 100%, 105%, 110% of original size
 - iii. Create an array to store the final BhattacharyaCoefficients for each object size and another to store the final positions
 - iv. Iterate over all p_u s
 1. While (convergence criteria)
 - a. computeBhattacharyaCoefficient
 - b. computeMeanShiftPosition update
 - c. Step 4 of the algorithm from paper
 - d. Update p_u for this new position
 2. Store the last computed Bhattacharya coefficient, and the final position in the respective arrays

- v. Choose that object size which gives the highest coefficient, and get that final position vector
 - vi. Display/Store frame $i+1$
 - vii. Save $y_0(i+1)$ and original $size_{i+1}$
- c. Close video display/Save video to file.