RIP CURRENT DETECTION USING OPTICAL FLOW

CMPS261 Project
Shweta Philip
• Assumptions made by optical flow algorithms:
  • Pixel intensities of an object do not change between consecutive frames.
  • Neighboring pixels have similar motion.

Optical flow Lucas-Kanade Algorithm, uses the following equation:

$$\frac{\partial I}{\partial x} \frac{\partial x}{\partial t} + \frac{\partial I}{\partial y} \frac{\partial y}{\partial t} + \frac{\partial I}{\partial t} = 0$$

$$\frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v + \frac{\partial I}{\partial t} = 0$$

Where $I(x,y,t)$ is a pixel intensity in a frame.
COLOR MAP FOR OPTICAL FLOW

• Velocity vector, output obtained by optical flow algorithm: \( \{u, v\} \)
• The direction of the velocity is represented as a different color.
• Each direction is represented by a hue, and to represent the magnitude, the saturation is adjusted. The more the magnitude, more the saturation.
• The magnitude is mapped along the radius of this color wheel.
• The result is then stored into three channels.
THE DIFFERENT CHANNELS OF THE OUTPUT FLOWMAP

Green Channel

Red Channel
THE BLUE CHANNEL VS THRESHOLDING

Blue Channel

Output obtained by thresholding w.r.t. direction
• A red mask was overlaid based on the information obtained by the blue channel.
• Red Area represents the motion in the frames that is going backwards.
• Though efficient to capture backward motion of the water, it comes with a lot of noise.
• Only a window of the blue channel is used.
FUTURE WORK/ IMPROVEMENTS

• Video Stabilization (Noise Reduction).
• The mask needs to represent the magnitude of the velocity. (map magnitude to transparency)
• Represent backward motion from surface color to arrows.
• Thresholding w.r.t magnitude of the velocity.
• Switch to Android Platform.
• Optimize the algorithm to get results in real time.
I COME IN PEACE
• Lucas Kanade Opencv
• Color Wheel mapping
• UCF Optical Flow algorithm
THANK YOU!