Optical Flow Report

Dimitri Montgomery
04/10/2025
CSC 340

Description

This assignment implements the Lucas-Kanade optical flow algorithm to detect motion between two image frames. Optical flow estimates the pixel motion vectors between frames, showing how objects move from scene to scene.

Key Equations

Gradient values:

```
emptyIMGIx[i, j] = float(image1o[i, j + 1]) - float(image1o[i, j - 1])
# Calculate y gradient using vertical neighbors
emptyIMGIy[i, j] = float(image1o[i + 1, j]) - float(image1o[i - 1, j])
```

Gradient products:

```
emptyIMGIxx[i, j] = emptyIMGIx[i, j] * emptyIMGIx[i, j]
emptyIMGIyy[i][j] = emptyIMGIy[i, j] * emptyIMGIy[i, j]
emptyIMGIxy[i][j] = emptyIMGIx[i, j] * emptyIMGIy[i, j]
```

Temporal Gradients:

```
emptyIMGIt[i, j] = float(image2o[i, j]) - float(image1o[i, j])
emptyIMGIxt[i, j] = emptyIMGIx[i, j] * emptyIMGIt[i, j]
emptyIMGIyt[i, j] = emptyIMGIy[i, j] * emptyIMGIt[i, j]
```

Sum of gradient values:

```
sumIxx += emptyIMGIxx[x, y]
sumIyy += emptyIMGIyy[x, y]
sumIxy += emptyIMGIxy[x, y]
sumIxt += emptyIMGIxt[x, y]
sumIyt += emptyIMGIyt[x, y]
```

Determinate

```
det_M = (sumIxx * sumIyy) - (sumIxy * sumIxy)
```

Solve for u & v:

```
u = ((-(sumIyy)*(sumIxt)) + ((sumIxy)*(sumIyt))) / det_M
v = (((sumIxy)*(sumIxt)) - ((sumIxx)*(sumIyt))) / det_M
```

Magnitude:

```
mag = math.sqrt(u**2 + v**2)
```

Convert flow into color:

```
theta = math.atan2(v, u)  # Direction of flow vector
blue = ((theta + math.pi) / (2 * math.pi))  # Map angle to blue channel [0,1]
green = 1 - blue  # Inverse mapping for green channel
colored_img[i][j][0] = blue * mag  # Scale blue by magnitude
colored_img[i][j][1] = green * mag  # Scale green by magnitude
```

Issues Encountered

Dimension Mismatch: Initially had a TypeError when trying to assign colors to the visualization array because it was initialized as a 2D array instead of a 3D array.
Incorrect
colored_img = np.zeros((rows1, cols1), np.float32)
Corrected
colored_img = np.zeros((rows1, cols1, 3), np.float32)

Index Out of Bounds: When making a larger window size (21×21), i had index errors because the window went beyond image boundaries. I was able to Fix this by adjusting loop ranges to ensure windows stayed within image boundaries.

Visualization Display: only one image window was showing at a time due to overlap.To fix this I created a menu-based interface to select between different visualizations.

Color Mapping: The color- addition for flow visualization was outside the main loop at first, resulting in only one pixel being colored. Fixed by moving the color calculation inside the loop.

Images