Daniel Herink
Jacquelin Rodriguez
ECPE 124
11/15/2016

Project Updates: Week 1

For the first week, one of our main objectives was to do more research on the Horn-Schunck Algorithm to get a better understanding of how it functions so we are better equipped when implementing it in our final project. Our programing objectives for this week were to start creating the functions that calculate the gradients Gx and Gy of the input images that we are analyzing as well as compute the displacement vector (u,v).

From the research we have done and from the past in class lectures that have focused on the optical flow constraint equation we were able to accomplish our objectives. The way we computed Gx and Gy was by creating a Gaussian and Gaussian derivative with a sigma value 0.6 and convolving both kernels over the input image to obtain the gradient Gx and Gy. To compute the displacement vector (u,v) it is a bit more complicated and less straightforward than computing the gradients. For this reason, we have to create various functions that compute specific aspects of the optical flow constraint equation. The optical flow constraint equation is the following:

$$\Delta I \bullet h + I_t = 0$$

h and ΔI represent the following:

$$h = (u, v) = (\tfrac{dx}{dt}, \tfrac{dy}{dt})$$
$$\Delta I = (I_x, I_y)$$

We have already calculated I_x *and* I_y which are the gradients Gx and Gy. To find what h is we have to rearrange the optical flow equation. The rearranged equation would look like the following:

$$h = -I_t \frac{\Delta I}{|\Delta I|^2}$$

However, this equation, known as *normal flow*, cannot determine the component of movement in the direction of the iso-brightness contours. Thus, additional constraints will be needed. Assuming constant movement, we will minimize the difficulties of the algorithm occluding edges by taking the square of the magnitude of the gradient of the optical flow velocity:

$$(\tfrac{\partial u}{\partial x})^2 + (\tfrac{\partial u}{\partial y})^2 \text{ and } (\tfrac{\partial v}{\partial x})^2 + (\tfrac{\partial v}{\partial y})^2$$

This is just the sum of the squares of the Laplacians of the x and y components, which are defined as

$$\nabla^2 u = \tfrac{\partial^2 u}{\partial x^2} + \tfrac{\partial^2 u}{\partial y^2} \text{ and } \nabla^2 v = \tfrac{\partial^2 v}{\partial x^2} + \tfrac{\partial^2 v}{\partial y^2}$$

From here, using the above as a measure of smoothness, we will estimate the partial derivatives of I_x, I_y, and I_t. This is done by averaging the first four differences taken over adjacent measurements in a chosen window. Estimating the Laplacians of the flow velocities comes next, followed by the minimization of errors in both estimations by the equations

$$(\alpha^2 + I_x{}^2 + I_y{}^2)(u^{n+1} - u^n) = -I_x[I_x u_t + I_y v_t + I_t]$$
$$(\alpha^2 + I_x{}^2 + I_y{}^2)(v^{n+1} - v^n) = -I_y[I_x u_t + I_y v_t + I_t]$$

where α^2 is a chosen weighting factor. Using the Gauss-Seidel method, we can compute the new set of velocity estimates, u^{n+1}, v^{n+1} from the previously estimated derivatives, u^n, v^n .

Finally, filling in the uniform regions with values found by the solution of the Laplace equation for the given boundary condition, filling in the velocity information from the boundary of a region of constant brightness. This will then give us the flow patterns of the images.