Jeremy Thomas

Optical Flow

The program takes a pair of images with a small amount of motion between them and implements the Luca-Kanade algorithm to compute the optical flow relating to the image pair. The program uses three functions responsible for collecting data, calculating the optical flow using the data collected, and a function that uses the optical flow calculation to generate the data flow visualizations.

Functions

The first function gathers data from the image by iterating over the pixels in the provided images. Information about the spatial gradient in the x direction, spatial gradient in the y direction, and temporal gradient are gathered and stored into arrays. This data is used to find the IxIx, IxIy, IyIy, IxIt, and IyIt values which are stored in arrays. The second function uses this data to calculate the motion in the images. It finds the summation of the previous values within a certain window. It also finds the determinate. These values are used to determine u, v, and the magnitude. The data used to visualize the flow angle, the optical flow in the x direction, and the optical flow in the y direction is found (fig. 1). The values of u and v are only calculated if the determinate is not 0. The third function visualizes the with colors and arrows using the data that has been gathered.

$$\text{determinate} = (\text{sum}\ (ixix) * \text{sum}\ (iyiy)) - (\text{sum}\ (ixiy)^2)$$

$$u = (-1 / \text{determinate}) * (\text{sum}\ (ixit) * \text{sum}\ (iyiy)) - (\text{sum}\ (ixiy) * \text{sum}\ (iyit))$$

$$v = (-1 / \text{determinate}) * ((-1 * (\text{sum}\ (ixit) * \text{sum}\ (ixiy)) + (\text{sum}\ (ixix) * \text{sum}\ (iyit))$$

$$\text{magnitude} = (u^2 + v^2)^{1/2}$$

$$\text{flow angle} = (\text{math.atan2}(v, u) + \text{math.pi}) / (2 * \text{math.pi})$$

fig. 1

Data

Fig. 2: Shows, from left to right, the flow in the x direction, the flow in the y direction, and the magnitude between sphere1 and sphere2.

Fig. 3: Shows, from left to right, the optical flow visualization and the optical flow angle visualization between sphere1 and sphere2.

Fig. 4: Shows, from top to bottom, the optical flow visualization and the optical flow angle visualization between house1 and house2.

Fig. 5: Shows, from top to bottom, the optical flow visualization and the optical flow angle visualization between mayaShapes1 and mayaShapes2.

Fig. 6: Shows, from top to bottom, the optical flow visualization and the optical flow angle visualization between mayaSolidShapes1 and mayaSolidShapes2.

Fig. 7: Shows, from top to bottom, the optical flow visualization and the optical flow angle visualization between tree1.jpg and tree2.jpg.

Fig. 2

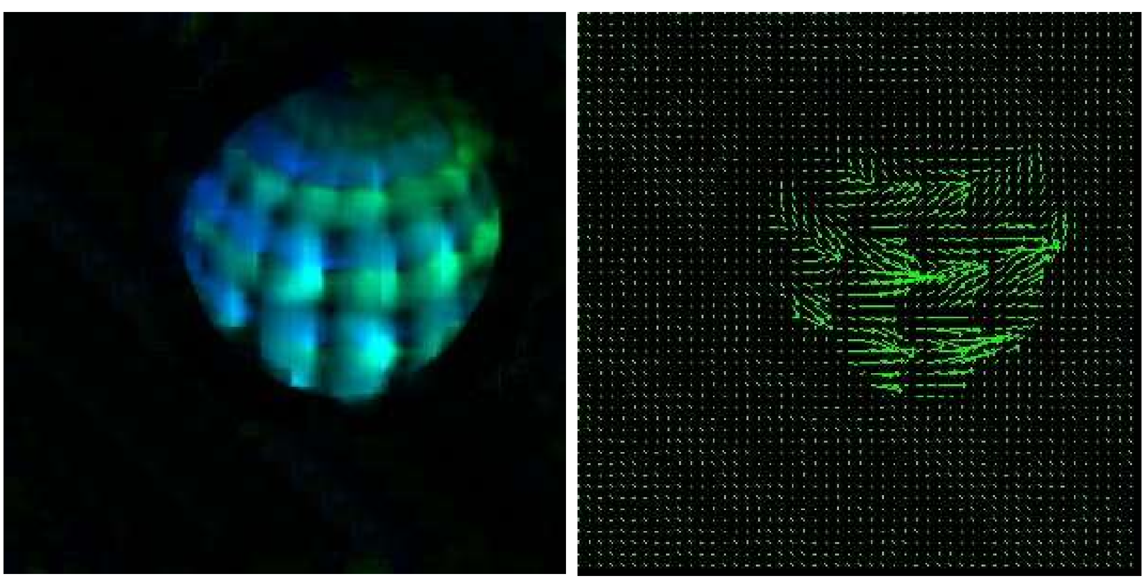

Fig. 3

Fig. 4

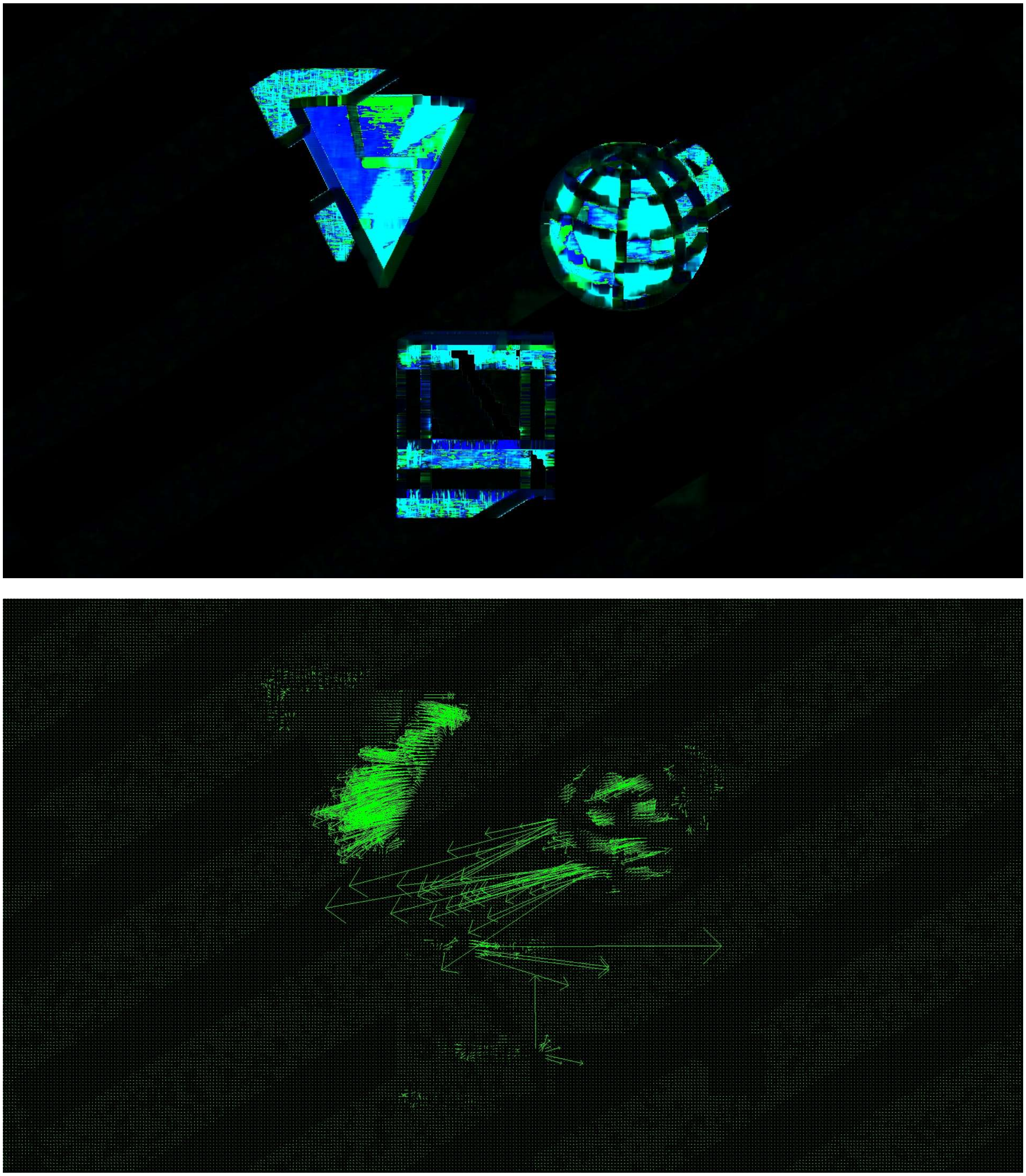

Fig. 5

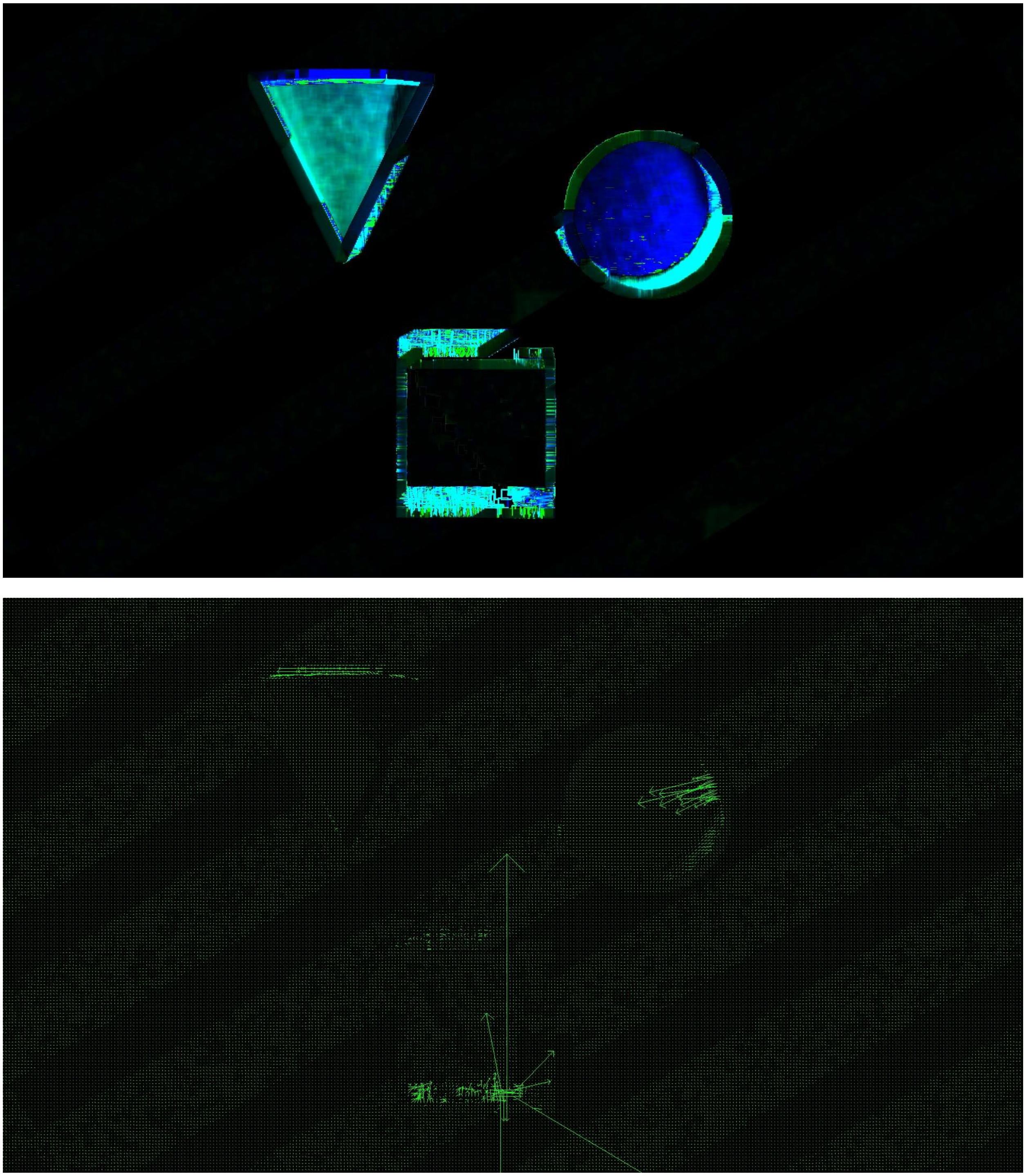

Fig. 6

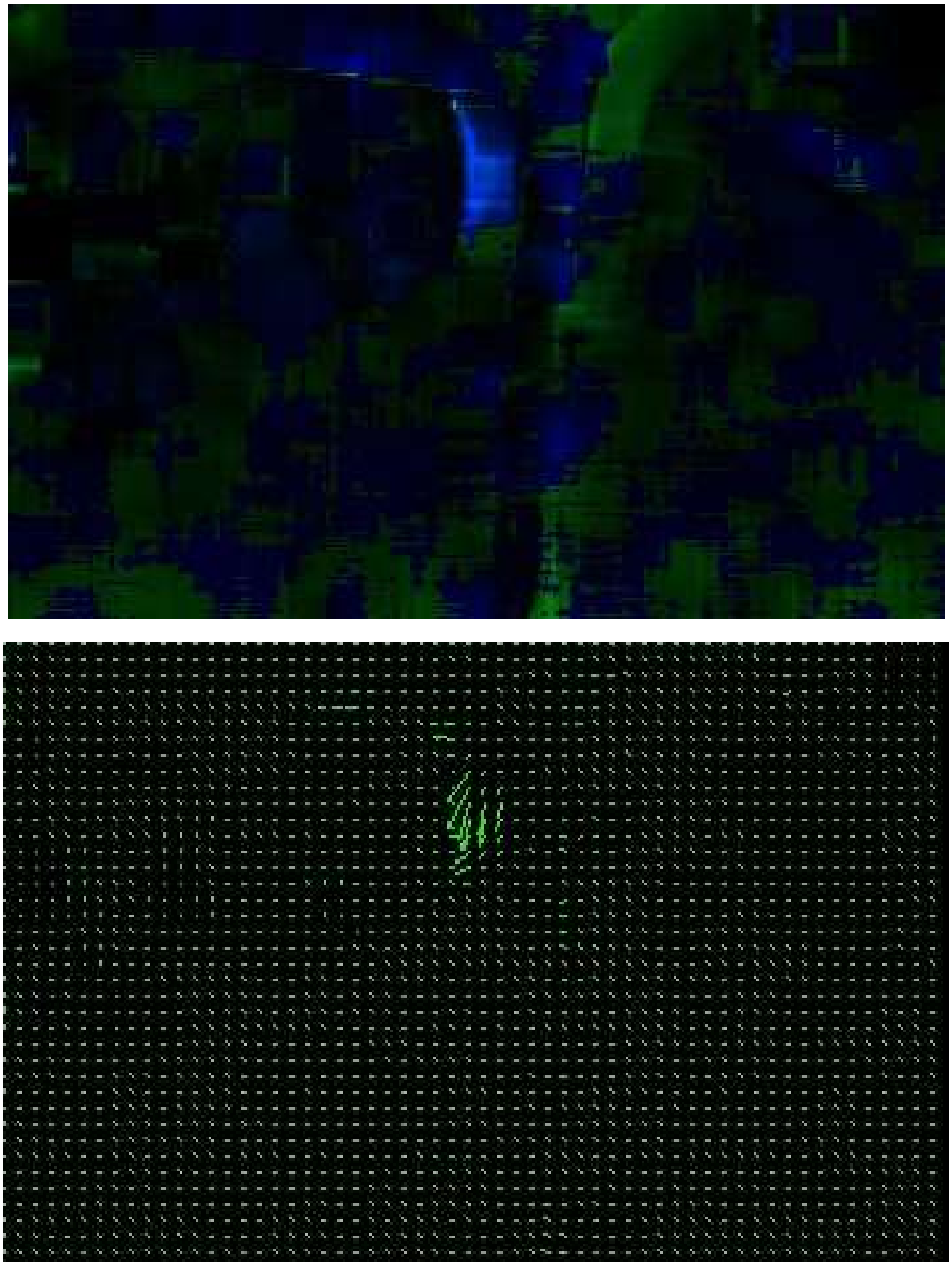

Fig. 7

Conclusion

It seems like the best results are from when we are tracking specific items. As the images get more complex, with moving backgrounds and moving foregrounds, the results seems to be less.

Items like the sphere seems to get better results. It may be because the item is moving on a static background.