Weekly Report

John Anglo

June 8, 2012

1 The Hough transform for circles

A Hough transform is a method by which shapes identified by a set of parameters, for example, lines and circles, are extracted from an image, through a voting procedure carried out in a space defined by those parameters. Circles in particular can be identified uniquely in 2D space by their radius and the coordinates of the center on the image.

Suppose we're looking for circles of a known radius on an image. Each point on the perimeter of any circle is at the same distance from the center of the circle, and that distance is the circle's radius. Meanwhile, all points that are that same distance away from a particular point on the perimeter fall on a circle having the same radius. So to locate the circle's center, we find the point shared by all circles centered on a point on the perimeter of the circle.

In terms of the transform procedure, for each nonzero pixel on the image, a circle is drawn having the known radius and centered at that pixel. Each pixel on that circle is a "vote" cast by the nonzero pixel, as each point on that circle is the right distance to be the center of the circle. Votes are collected in an array known as an accumulator, here having the same dimensions as the image. Circles are identified as being centered on local maxima in the accumulator.

If the radii of circles on an image are different, or not known, then a third dimension must be added to the accumulator, corresponding to the radius of a circle. Here each pixel must cast votes in three dimensions, drawing a circle in the accumulator at each possible radius. The set of votes cast by one pixel draws a cone in the three-dimensional space of the accumulator, with the radius of drawn circles increasing as one moves along the radius dimension.

Clearly this can be a very slow procedure; we're working with 256 × 256 images having circles of radius anywhere from 10 to 200 pixels. That means about 200 circles drawn for each of tens of thousands of pixels. So it would help to be able to remove as much non-circle information from an image as possible, and also spend the smallest possible amount of time drawing each circle.

For drawing circles I implemented the midpoint circle algorithm, as described on Wikipedia, which requires only addition and subtraction to draw a point on the circle, and seems to be as good as it gets, having to do at most six arithmetic operations to find each point. Additionally, the symmetry of the circle means that the operations only have to be performed for an octant of the circle, and mirrorred to draw the rest of the circle.

To remove unusable information, I have been looking at different methods of filtering and improving contrast, with the goal of making the arcs drawn by stars on the images stand out, without doing the same to what appears on the horizon. For example, taking the average of a stack of images darkens noise effectively, but doesn't make arcs bright enough. The most effective method I've found to draw out the arcs is to take the variance of the stack, but doing this also brightens the noise greatly.

So far it looks like image noise, combined with not having full circles to look at, has made searching for circles as three parameters ineffective. So I attempted to write a function that uses the same principle of voting, with the assumption that the circles in the image are concentric like the arcs, and centered around the zenith, which can be identified by locating Polaris. Here circles are drawn for a range of radii, all centered at the zenith, and each pixel that falls on a particular circle gets a vote weighted by the pixel value. The function still relies on the ability to draw out arcs and filter noise, but it is significantly faster, and with only one parameter to calculate I expect it to be more reliable.