Fiducial matching method

Jeren Suzuki

Last Edited July 10, 2013

The best way I came up with to find the fiducials is to make an NxN matrix with the rows/columns as N fiducial positions. For example, we have 4 fiducials that we call A, B, C, and D. The matrix looks like:

$$\begin{array}{ccccc} & A & B & C & D \\ A & & & & \\ B & & & & \\ C & & & & \\ D & & & & \end{array}$$

We then fill it in to be

$$\begin{array}{cccc} AA & AB & AC & AD \\ BA & BB & BC & BD \\ CA & CB & CC & CD \\ DA & DB & DC & DD \end{array}$$

These are the combinations of all possible chord lengths with the endpoints being the fiducial indexes. Now we rule out AA, BB, CC, and DD since the distance from A to A is 0, etc. We eliminate duplicate chords to get:

$$\begin{array}{cccc} 0 & AB & AC & AD \\ 0 & 0 & BC & BD \\ 0 & 0 & 0 & CD \\ 0 & 0 & 0 & 0 \end{array}$$

To get the actual chord lengths, we use the formula $D = \sqrt{x^2 + y^2}$ with:

$$x = \begin{array}{cccc} 0 & \overline{A_xB_x} & \overline{A_xC_x} & \overline{A_xD_x} \\ 0 & 0 & \overline{B_xC_x} & \overline{B_xD_x} \\ 0 & 0 & 0 & \overline{C_xD_x} \\ 0 & 0 & 0 & 0 \end{array}$$

$$y = \begin{array}{cccc} 0 & \overline{A_yB_y} & \overline{A_yC_y} & \overline{A_yD_y} \\ 0 & 0 & \overline{B_yC_y} & \overline{B_yD_y} \\ 0 & 0 & 0 & \overline{C_yD_y} \\ 0 & 0 & 0 & 0 \end{array}$$

Here is where I'm getting myself a little confused, do we have a list of chord lengths between fiducials or a list of fiducial positions? My thinking is that if they are chord lengths, then we can just match up the distances between fiducials but since each chord is a pair of fiducials, we don't know for a single chord which fiducial is on which end. We'd have to iterate through a bunch of fiducial chords to figure out which fiducial we ID is the fiducial in the table.

If we're working with a list of fiducial positions then I can just look up the distance of the fiducial from the origin. Any fiducial within, say, 1% of a fiducial position on our table is considered a match.

I don't see any clear benefits/cons between the two types of lists; is there one?