Elliptical Regression on FPGA

The equation, , is the standard equation for an ellipse. By solving for B, C, D, E, and F for a given set of data points, an ellipse can be fit to those data points. The method of least squares attempts to create this ellipse by reducing the average squared distance along the y-axis from the points to the ellipse to a mathematical minimum.

The traditional approach for finding the least squares solution is done using matrix multiplication and matrix transverse operations. For a set of data points, (x0, x1, … , xi), (y0, y1, … , yi), a few matrices are created.

A=

| y02 | x0y0 | x0 | y0 | 1 |
| --- | --- | --- | --- | --- |
| y12 | x1y1 | x1 | y1 | 1 |
| … | … | … | … | … |
| yi2 | xiyi | xi | yi | 1 |

AT=

| y02 | y12 | … | yi2 |
| --- | --- | --- | --- |
| x0y0 | x1y1 | … | xiyi |
| x0 | x1 | … | xi |
| y0 | y1 | … | yi |
| 1 | 1 | … | 1 |

b=

| -x02 |
| --- |
| -x12 |
| … |
| -xi2 |

From these matrices, two matrices are created using matrix multiplication.

ATA=

| Σy4 | Σxy3 | Σxy2 | Σy3 | Σy2 |
| --- | --- | --- | --- | --- |
| Σxy3 | Σx2y2 | Σx2y | Σxy2 | Σxy |
| Σxy2 | Σx2y | Σx2 | Σxy | Σx |
| Σy3 | Σxy2 | Σxy | Σy2 | Σy |
| Σy2 | Σxy | Σx | Σy | n |

ATb=

| Σ-(x2y2) |
| --- |
| Σ-(x3) |
| Σ-(x3y) |
| Σ-(x2y) |
| Σ-(x2) |

matrix=

| Σy4 | Σxy3 | Σxy2 | Σy3 | Σy2 | Σ-(x2y2) |
| --- | --- | --- | --- | --- | --- |
| Σxy3 | Σx2y2 | Σx2y | Σxy2 | Σxy | Σ-(x3) |
| Σxy2 | Σx2y | Σx2 | Σxy | Σx | Σ-(x3y) |
| Σy3 | Σxy2 | Σxy | Σy2 | Σy | Σ-(x2y) |
| Σy2 | Σxy | Σx | Σy | n | Σ-(x2) |

rref(matrix)=

| 1 | 0 | 0 | 0 | 0 | B |
| --- | --- | --- | --- | --- | --- |
| 0 | 1 | 0 | 0 | 0 | C |
| 0 | 0 | 1 | 0 | 0 | D |
| 0 | 0 | 0 | 1 | 0 | E |
| 0 | 0 | 0 | 0 | 1 | F |

These matrices together create a matrix that, row reduced, will yield the coefficients to the ellipse equation. The total number of values that need to be calculated for each data point is 13: y4, x3y, x2y2, xy3, x3, x2y, xy2, y3, x2, xy, y2, x, and y. These values can then each be summed for all of the data points and used to populate the same matrix that, when row reduced, will yield the solution of least squares.

Finding the required summations to calculate the coefficients can be accelerated in hardware. Multipliers can be used to calculate the 13 values for each data point, and adders can be used to sum the values together. A program to write the values to the FPGA, read the values from the FPGA, and perform the matrix row reduction will still be required.

In my system I use a dataInput of 8 bits, a dataOutput of 32 bits, a addressInput of 4 bits, and an inOrOutAddress and clk of 1 bit. The dataInput is in the form: bit(0)-sign bit, bit(1:7)-value bits; dataOutput: bit(0)-sign bit, bit(1:31)-value bits. The inOrOutAddress tells the system whether the addressInput describes an address where an input value should be stored or an address for the retrieval of an output value. A 0 indicates that the system should store the value at dataInput to the address indicated in addressInput on the next rising clk cycle and a 1 indicates that the system should write the value at the address indicated in addressInput to dataOutput on the next rising clk cycle. The addressInput, with the inOrOutAddress bit, is the input that directs the system to store or read values from the FPGA. When the inOrOutAddress is 0, even addresses represent the x-values for the data points while the following odd addresses will represent the y-values. When the inOrOutAddress bit is 1, the addresses point to: 0000-y4 ,0001-xy3, 0010-x3y 0011-x2y2, 0100-y3, 0101-x3, 0110-xy2, 0111-x2y, 1000-y2, 1001-x2, 1010-xy, 1011-y, and 1100-x. Finally, because the multiplexer and the registers are synchronous, a clk bit input is used. To write a value to a register, the value and register address must be written to the dataInput and addressInput respectively, and the inOrOutAddress needs to be set to 0. Then, when the clk input goes high, the value will be written to the register. To read a value from the system the address that corresponds to the desired value must be written to the addressInput and inOrOut Address must be set to 1. When the clk input then goes high, the value at the requested address will be written to the dataOutput.