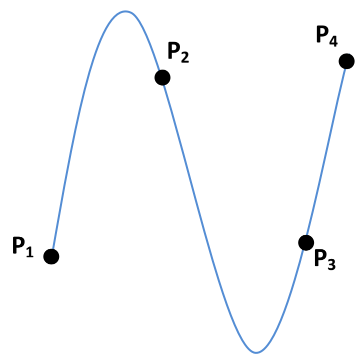
SECTION 3 DATA PREDICTION AND FILLING

3.1 General Curve Fitting

This section describes how to fit curves and surfaces to a set of data points. These methods require a certain number of points for each particular model. The resulting curve or surface will pass exactly through each of the points.

3.1.1 Polynomial Curve Fit

Description: Polynomial curve fit determines the equation of an n^{th} order polynomial that passes through n+1 points. Unlike polynomial regression, this method requires exactly n+1 data points and the resulting polynomial will pass through every point.



The figure above shows four points defining a cubic. The polynomial equation for a cubic is:

$$y = Ax^3 + Bx^2 + Cx + D$$

The variables A-D can be solved by a set of linear equations using the data from each of the points.

$$\begin{aligned} Ax_1^3 + Bx_1^2 + Cx_1 + D = y_1 \\ Ax_2^3 + Bx_2^2 + Cx_2 + D = y_2 \\ Ax_3^3 + Bx_3^2 + Cx_3 + D = y_3 \\ Ax_4^3 + Bx_4^2 + Cx_4 + D = y_4 \end{aligned}$$

The equations are then converted to matrix form:

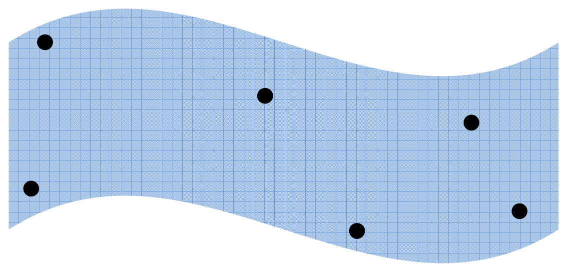
$$\begin{bmatrix} x_1^3 & x_1^2 & x_1 & 1 \\ x_2^3 & x_2^2 & x_2 & 1 \\ x_3^3 & x_3^2 & x_3 & 1 \\ x_4^3 & x_4^2 & x_4 & 1 \end{bmatrix} \cdot \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

Solve for A-D:

$$\begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} x_1^3 & x_1^2 & x_1 & 1 \\ x_2^3 & x_2^2 & x_2 & 1 \\ x_3^3 & x_3^2 & x_3 & 1 \\ x_4^3 & x_4^2 & x_4 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

3.1.2 Polynomial Surface Fit

Description: Polynomial surface fit determines the equation of an mth x nth order polynomial surface that passes through n*m+1 points. Unlike bi-polynomial regression, this method requires n*m+1 data points and the resulting polynomial surface will pass through every point.



The image above represents a 2nd order surface with the equation:

$$z = Ax^2 + Bxy + Cy^2 + Dx + Ey + F$$

To define this surface there must be 6 linearly independent points. To solve for the coefficients, the same process from the curve fitting is used. The surface is represented as 6 linear equations and solved simultaneously.

$$\begin{aligned}
Ax_1^2 + Bx_1y_1 + Cy_1^2 + Dx_1 + Ey_1 + F = z_1 \\
Ax_2^2 + Bx_2y_2 + Cy_2^2 + Dx_2 + Ey_2 + F = z_2 \\
Ax_3^2 + Bx_3y_3 + Cy_3^2 + Dx_3 + Ey_3 + F = z_3 \\
Ax_4^2 + Bx_4y_4 + Cy_4^2 + Dx_4 + Ey_4 + F = z_4 \\
Ax_5^2 + Bx_5y_5 + Cy_5^2 + Dx_5 + Ey_5 + F = z_5 \\
Ax_6^2 + Bx_6y_6 + Cy_6^2 + Dx_6 + Ey_6 + F = z_6
\end{aligned}$$

In matrix form:

$$\begin{bmatrix} x_1^2 & x_1y_1 & y_1^2 & x_1 & y_1 & 1 \\ x_2^2 & x_2y_2 & y_2^2 & x_2 & y_2 & 1 \\ x_3^2 & x_3y_3 & y_3^2 & x_3 & y_3 & 1 \\ x_4^2 & x_4y_4 & y_4^2 & x_4 & y_4 & 1 \\ x_5^2 & x_5y_5 & y_5^2 & x_5 & y_5 & 1 \\ x_6^2 & x_6y_6 & y_6^2 & x_6 & y_6 & 1 \end{bmatrix} \cdot \begin{bmatrix} A \\ B \\ C \\ D \\ E \\ F \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \end{bmatrix}$$

Solve for A-F by taking the inverse of the coefficient matrix.

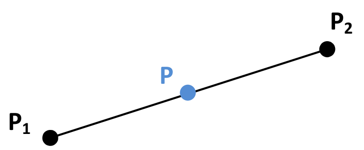
$$\begin{bmatrix} A \\ B \\ C \\ D \\ E \\ F \end{bmatrix} = \begin{bmatrix} x_1^2 & x_1y_1 & y_1^2 & x_1 & y_1 & 1 \\ x_2^2 & x_2y_2 & y_2^2 & x_2 & y_2 & 1 \\ x_3^2 & x_3y_3 & y_3^2 & x_3 & y_3 & 1 \\ x_4^2 & x_4y_4 & y_4^2 & x_4 & y_4 & 1 \\ x_5^2 & x_5y_5 & y_5^2 & x_5 & y_5 & 1 \\ x_6^2 & x_6y_6 & y_6^2 & x_6 & y_6 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \end{bmatrix}$$

3.2 Interpolation

Interpolation is the process of extracting information from a set of data between the data points. It involves determining a mathematical model to fit the collected data and then solving the prediction equation at a location where information is desired. Typically linear models are used, but there are countless others, polynomial, sinusoidal, spline, etc. Several are discussed in this section.

3.2.1 Linear Interpolation

Description: Linear interpolation determines the value of a given point by taking a weighted average of the two neighboring points based on the distance from each of those points.



$P \rightarrow (x, y)$
$P_1 \rightarrow (x_1, y_1)$
$P_2 \rightarrow (x_2, y_2)$

The distance from P_1 to P and the distance from P_1 to P_2 are proportional between the x and y directions for a linear system.

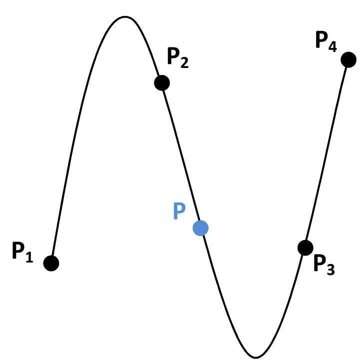
$$\frac{y - y_1}{y_2 - y_1} = \frac{x - x_1}{x_2 - x_1}$$

Given the value of x, the value of y can be solved for with some simple algebra.

$$y = \left(\frac{x - x_1}{x_2 - x_1}\right) \cdot (y_2 - y_1) + y_1$$

3.2.2 Polynomial Interpolation

Description: Polynomial interpolation determines the value of a given point by fitting an n^{th} order polynomial though n+1 points surrounding the given value. The example below shows a 3^{rd} order polynomial fit through 4 points.



$P \rightarrow (x, y)$
$P_n \rightarrow (x_n, y_n)$

The equation for a 3^{rd} order polynomial is:

$$y = Ax^3 + Bx^2 + Cx + D$$

The values of A, B, C, and D are solved for by solving a set of linear equations using the method described *Polynomial Curve Fit (Section 3.1.1)*.

$$Ax_1^3 + Bx_1^2 + Cx_1 + D = y_1$$

$$Ax_2^3 + Bx_2^2 + Cx_2 + D = y_2$$
$$Ax_3^3 + Bx_3^2 + Cx_3 + D = y_3$$
$$Ax_4^3 + Bx_4^2 + Cx_4 + D = y_4$$

Once the variables A, B, C, and D are calculated, y is calculated by plugging in x to the polynomial equation and solving.

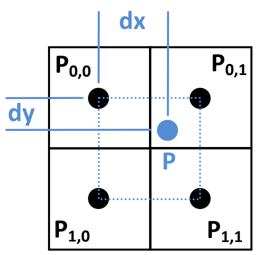
See also:
- *Polynomial Curve Fit (Section 3.1.1)*

3.2.3 Spline Interpolation

[TBD]

3.2.4 Bilinear Interpolation

Description: Bilinear interpolation determines the value at a given point by taking the average of the four closest neighbors weighted by distance from the new point. This particular algorithm requires that points be monotonic and regularly spaced (like pixels in an image). For data that is not evenly spaced, use *Irregular Bilinear Interpolation (Section 3.2.5) or Triangular Interpolation (Section 3.2.7).*



Location of Pixels:
$P \rightarrow (x, y)$
$P_{m,n} \rightarrow (x_m, y_n)$

Values of Pixels:
$P \rightarrow V$
$P_{m,n} \rightarrow V_{m,n}$

The figure above shows bilinear interpolation for an image. Point P is the location of a new pixel, while the values $P_{0,0} - P_{0,1}$ are the original pixels in the image that surround the new point. The values dx and dy represent the percent offsets from the top left pixel.

$$dx = \frac{x - x_0}{x_1 - x_0} \qquad \text{and} \qquad dy = \frac{y - y_0}{y_1 - y_0}$$

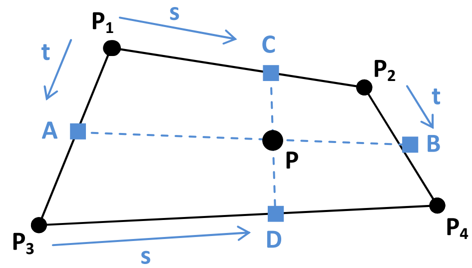
The value of the new pixel is calculated by taking weighted averages of the surrounding pixels using the offsets dx and dy.

$$V = V_{0,0} \cdot (1 - dx) \cdot (1 - dy) + V_{0,1} \cdot (dx) \cdot (1 - dy) + V_{1,0} \cdot (1 - dx) \cdot (dy) + V_{1,1} \cdot (dx) \cdot (dy)$$

3.2.5 Irregular Bilinear Interpolation

Description: Irregular bilinear interpolation determines the value at a given point by taking the weighted average of its four closest neighbors. This algorithm is nonlinear and more computationally intensive than standard bilinear interpolation. If the data being interpolated is regularly spaced use the standard version, *Bilinear Interpolation (Section 3.2.4)*

The figure below show an irregular point set. Points P represents the new point and points 1-4 are the bounding points. The distance s is the percentage along the vertical sides to point P, the distance t is the horizontal. Points A-D are intermediate points used for calculation.



The edges are represented as parametric lines using the equations:

$$x = x_0 + \alpha \cdot t$$
$$y = y_0 + \beta \cdot t$$

The intermediate points A - D can be described in terms of parametric lines through the corner points P_1 - P_4.

Point A

$$A_x = x_1 + x_{31} \cdot t$$
$$A_y = y_1 + y_{31} \cdot t$$

Point A

$$B_x = x_2 + x_{42} \cdot t$$
$$B_y = y_2 + y_{42} \cdot t$$

Point C

$$C_x = x_1 + x_{21} \cdot s$$
$$C_y = y_1 + y_{21} \cdot s$$

Point D

$$D_x = x_3 + x_{43} \cdot s$$
$$D_y = y_3 + y_{43} \cdot s$$

Where the intermediate variables are defined as:

$$x_{31} = x_3 - x_1$$
$$y_{31} = y_3 - y_1$$
$$x_{42} = x_4 - x_2$$
$$\dots$$

Using points A - D the point P can be solved for by two different parametric lines.

$$P_x = A_x + (B_x - A_x) \cdot s \qquad P_y = A_y + \left(B_y - A_y\right) \cdot s$$

and

$$P_x = C_x + (D_x - C_x) \cdot t \qquad P_y = C_y + \left(D_y - C_y\right) \cdot t$$

The value of P can be attained by solving either pair of these equations, but there are conditions that only one or the other set will give a viable answer.

Method 1

Using the first set of equations the value of s can be formulated by rearranging the equation for y_P.

$$P_y = A_y + \left(B_y - A_y\right) \cdot s \quad \rightarrow \quad s = \frac{P_y - A_y}{B_y - A_y}$$

The value of s can then be plugged into the equation for x_P.

$$P_x = A_x + (B_x - A_x) \cdot \left(\frac{P_y - A_y}{B_y - A_y}\right)$$

This equation is then rearranged:

$$0 = (B_x - A_x)(P_y - A_y) - (P_x - A_x)(B_y - A_y)$$

The values of A_x, B_x, A_y, and B_y are plugged in to place the solution in terms of t.

$$0 = \big((x_2 + x_{42} \cdot t) - (x_1 + x_{31} \cdot t)\big)\Big(P_y - (y_1 + y_{31} \cdot t)\Big) - \big(P_x - (x_1 + x_{31} \cdot t)\big)\big((y_2 + y_{42} \cdot t) - (y_1 + y_{31} \cdot t)\big)$$

Through a significant bit of algebra this equation can be condensed and factored into the terms of the second order polynomial:

$$At^2 + Bt + C = 0$$

Where the variables A - C are:

$$\begin{aligned}
A &= x_{31}y_{42} - y_{31}x_{42} \\
B &= P_y(x_{42} - x_{31}) - P_x(y_{42} - y_{31}) + x_{31}y_2 - y_{31}x_2 + x_1y_{42} - y_1x_{42} \\
C &= P_yx_{21} - P_xy_{21} + x_1y_2 - x_2y_1
\end{aligned}$$

The value of t can be solved for using the quadratic equation:

$$(t_1, t_2) = \frac{-B \pm \sqrt{B^2 - 4AC}}{2C}$$

Since this equation is quadratic, it is likely that two separate roots will be found when solving the equation. When the object of this method is interpolation (not extrapolation), determining which factor is real and which is false is simple. For any interpolation the value of t must be between 0 and 1.

Using the value of t the value of s can be solved for using the value of t.

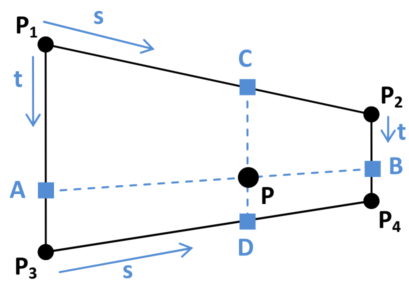
$$s = \frac{P_y - A_y}{B_y - A_y} \qquad \rightarrow \qquad s = \frac{P_y - y_1 - y_{31}t}{y_2 + y_{42}t - y_1 - y_{31}t}$$

After calculating the s and t values, the value at point P is calculated using the standard bilinear interpolation formula.

$$\begin{aligned}
P = P_1 &\cdot (1 - s) \cdot (1 - t) + \\
P_2 &\cdot (s) \cdot (1 - t) + \\
P_3 &\cdot (1 - s) \cdot (t) + \\
P_4 &\cdot (s) \cdot (t)
\end{aligned}$$

Method 2

If the vertical uprights of the interpolation area are parallel, the solution for the quadratic equation will return complex values. If this occurs a second method focusing on the horizontal sections of the viewing area can be used instead.



The same calculations from the first method are used but with the points C and D instead. Following the same calculations, the values of the quadratic coefficients become:

$$A = x_{21}y_{43} - y_{21}x_{43}$$
$$B = P_y(x_{43} - x_{21}) - P_x(y_{43} - y_{21}) + x_1y_{43} - y_1x_{43} + x_{21}y_3 - y_{21}x_3$$
$$C = P_yx_{31} - P_xy_{31} + x_1y_3 - x_3y_1$$

The value of s is then calculated using the quadratic equation.

$$(s_1, s_2) = \frac{-B \pm \sqrt{B^2 - 4AC}}{2C}$$

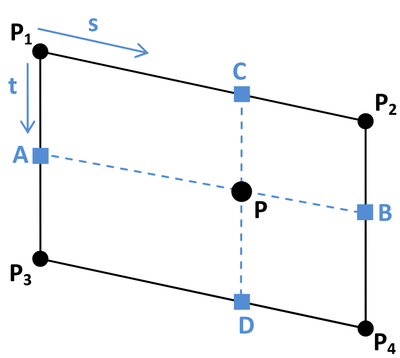
The value of t can be solved for

$$t = \frac{P_y - C_y}{D_y - C_y} \qquad \rightarrow \qquad t = \frac{P_y - y_1 - y_{21} \cdot s}{y_3 + y_{43} \cdot s - y_1 - y_{21} \cdot s}$$

And s and t can be used in the standard bilinear interpolation formula listed above.

Method 3

If the interpolation area becomes a parallelogram, a third method is required to solve for the value at P.



With both pairs of edges parallel, the solution for the point P becomes linear and only marginally more complicated than the standard bilinear interpolation method. Take the original equations for A and B:

$$A_x = x_1 + x_{31} \cdot t \qquad B_x = x_2 + x_{42} \cdot t$$
$$A_y = y_1 + y_{31} \cdot t \qquad B_y = y_2 + y_{42} \cdot t$$

And the solution for P given A and B:

$$P_x = A_x + (B_x - A_x) \cdot s$$
$$P_y = A_y + (B_y - A_y) \cdot s$$

Plugging in the values for A and B gives:

$$P_x = x_1 + x_{31}t + x_2 s + x_{31}st - x_1 s - x_{42}st$$
$$P_y = y_1 + y_{31}t + y_2 s + y_{31}st - y_1 s - y_{42}st$$

But since edges $\overline{P_1P_3}$ and $\overline{P_2P_4}$ are parallel, $x_{31} = x_{42}$ and $y_{31} = y_{42}$. This makes the cross terms cancel out and the remaining equations are:

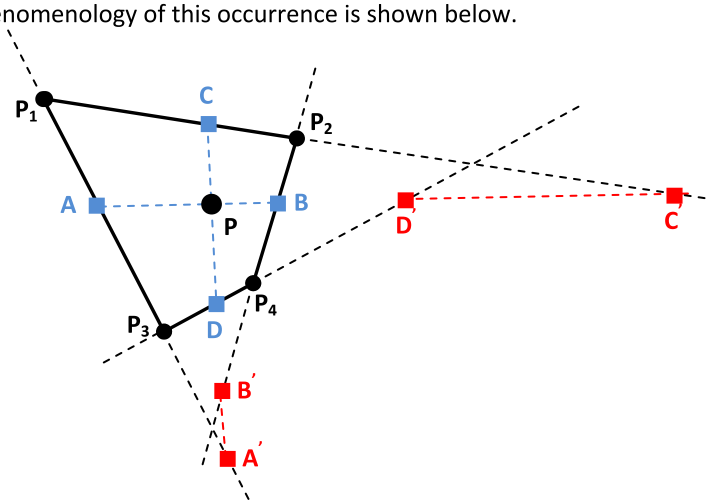
$$P_x = x_1 + x_{31}t + (x_2 - x_1)s$$
$$P_y = y_1 + y_{31}t + (y_2 - y_1)s$$

This system can now be solved as a set of linear equations.

$$\begin{bmatrix} x_{21} & x_{31} \\ y_{21} & y_{31} \end{bmatrix} \cdot \begin{bmatrix} s \\ t \end{bmatrix} = \begin{bmatrix} P_x - x_1 \\ P_y - y_1 \end{bmatrix}$$

<u>Error Conditions</u>

As mentioned earlier, the quadratic solution for the s and t variables presents the problem of solving for the incorrect roots. The phenomenology of this occurrence is shown below.



The blue dashed lines and boxes indicate the results from the correct roots. The point P is solved for using values of s and t that are between 0 and 1. The red boxes and dashed lines show the results from the other solution to this quadratic. At the values of A', B', C', and D' the lines formed still intersect at point P, but the values make less sense. In particular the horizontal intersection line $\overline{AB}$ has now become the vertical and vise versa.

Another problem that can arise is the condition of parallel lines in the interpolation region. This condition is what drives the need for multiple solution methods. Of course, there is the condition of a rectangular region, or more generally a parallelogram, which would cause both methods to fail. This condition must be checked for explicitly.

For the vertical edges of the search region, the parallelism condition can be checked for using the cross product of the two vertical edges.

$$\text{If} \quad (x_3 - x_1)(y_4 - y_2) - (x_4 - x_2)(y_3 - y_1) == 0 \quad \text{then} \quad VertLines = Parallel$$

Likewise for the horizontal edges the cross product can be used to determine parallelism.

$$\text{If} \quad (x_2 - x_1)(y_4 - y_3) - (x_4 - x_3)(y_2 - y_1) == 0 \quad \text{then} \quad HorzLines = Parallel$$

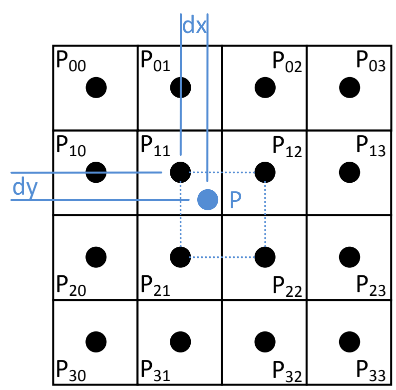
The processing structure uses this information to determine which method to use when performing the interpolation. The processing logic is as follows:

$$\begin{aligned}
&\text{if } (VertLines \neq Parallel) \;\&\; (HorzLines \neq Parallel) \;\rightarrow \text{use Method 1}\\
&\text{if } (VertLines \neq Parallel) \;\&\; (HorzLines = Parallel) \;\rightarrow \text{use Method 1}\\
&\text{if } (VertLines = Parallel) \;\&\; (HorzLines \neq Parallel) \;\rightarrow \text{use Method 2}\\
&\text{if } (VertLines = Parallel) \;\&\; (HorzLines = Parallel) \;\rightarrow \text{use Method 3}
\end{aligned}$$

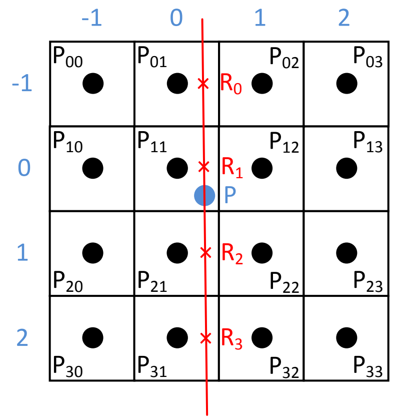
3.2.6 Bicubic Interpolation

Description: Bicubic interpolation determines the value at a given point by interpolating values from the 16 neighboring pixels using cubic polynomials in two directions. This algorithm requires that points be monotonic and evenly spaced (like pixels in an image). For data that is not evenly spaced, use *Irregular Bilinear Interpolation (Section 3.2.5) or Triangular Interpolation (Section 3.2.7).*

A typical pixel grid is show below. The black dots represent the centers of the pixels. The blue dot represents the location of the new pixel relative to the original pixel grid. The offsets from the top left pixel are shown by the values of dx and dy.



Bicubic interpolation creates a series of four cubics through the points of each row. These cubics are then solved at the points $R_0 - R_3$ shown in red below in the figure below. These four values then make up the vertical cubic which is solved at dy to get the value of the point P. The local pixel coordinates are shown on the edges of the grid in blue.



For an image, the spacing between pixels is assumed to be 1 unit so dx and dy can be solved for by subtracting the x and y components of point P_{11} from point P.

$$dx = P_x - P_{11x}$$
$$dy = P_y - P_{11y}$$

For grid spacing that is not unitized, dx and dy would be solved for using linear interpolation between P_{11} and P_{12} and P_{11} and P_{21}, respectively.

The first step of the algorithm is to solve for the pixel values of $R_0 - R_3$.

Starting with the equation of a cubic:

$$y = Ax^3 + Bx^2 + Cx + D$$

Solve for the coefficients $A - D$ of the cubic equation for each row. The x position across the image is the independent variable and the pixel intensity is the dependent. This is shown for the first row below:

$$P_{00} = A_0{x_0}^3 + B_0{x_0}^2 + C_0x_0 + D_0$$
$$P_{01} = A_0{x_1}^3 + B_0{x_1}^2 + C_0x_1 + D_0$$
$$P_{02} = A_0{x_2}^3 + B_0{x_2}^2 + C_0x_2 + D_0$$
$$P_{03} = A_0{x_3}^3 + B_0{x_3}^2 + C_0x_3 + D_0$$

Expanded to matrix form:

$$[A_0 \quad B_0 \quad C_0 \quad D_0] \cdot \begin{bmatrix} {x_0}^3 & {x_1}^3 & {x_2}^3 & {x_3}^3 \\ {x_0}^2 & {x_1}^2 & {x_2}^2 & {x_3}^2 \\ x_0 & x_1 & x_2 & x_3 \\ 1 & 1 & 1 & 1 \end{bmatrix} = [P_{00} \quad P_{01} \quad P_{02} \quad P_{03}]$$

The matrix equations can be further expanded to solve for all four coefficient rows simultaneously.

$$\begin{bmatrix} A_0 & B_0 & C_0 & D_0 \\ A_1 & B_1 & C_1 & D_1 \\ A_2 & B_2 & C_2 & D_2 \\ A_3 & B_3 & C_3 & D_3 \end{bmatrix} \cdot \begin{bmatrix} {x_0}^3 & {x_1}^3 & {x_2}^3 & {x_3}^3 \\ {x_0}^2 & {x_1}^2 & {x_2}^2 & {x_3}^2 \\ x_0 & x_1 & x_2 & x_3 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} P_{00} & P_{01} & P_{02} & P_{03} \\ P_{10} & P_{11} & P_{12} & P_{13} \\ P_{20} & P_{21} & P_{22} & P_{23} \\ P_{30} & P_{31} & P_{32} & P_{33} \end{bmatrix}$$

Solving for the cubic coefficients:

$$\begin{bmatrix} A_0 & B_0 & C_0 & D_0 \\ A_1 & B_1 & C_1 & D_1 \\ A_2 & B_2 & C_2 & D_2 \\ A_3 & B_3 & C_3 & D_3 \end{bmatrix} = \begin{bmatrix} P_{00} & P_{01} & P_{02} & P_{03} \\ P_{10} & P_{11} & P_{12} & P_{13} \\ P_{20} & P_{21} & P_{22} & P_{23} \\ P_{30} & P_{31} & P_{32} & P_{33} \end{bmatrix} \cdot \begin{bmatrix} {x_0}^3 & {x_1}^3 & {x_2}^3 & {x_3}^3 \\ {x_0}^2 & {x_1}^2 & {x_2}^2 & {x_3}^2 \\ x_0 & x_1 & x_2 & x_3 \\ 1 & 1 & 1 & 1 \end{bmatrix}^{-1}$$

The X values are constant for this region so the $[X]$ coefficient matrix can be specified explicitly.

$$[X] = \begin{bmatrix} {x_0}^3 & {x_1}^3 & {x_2}^3 & {x_3}^3 \\ {x_0}^2 & {x_1}^2 & {x_2}^2 & {x_3}^2 \\ x_0 & x_1 & x_2 & x_3 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 1 & 8 \\ 1 & 0 & 1 & 4 \\ -1 & 0 & 1 & 2 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

The inverse of $[X]$ becomes:

$$[X]^{-1} = \begin{bmatrix} -1/6 & 1/2 & -1/6 & 0 \\ 1/2 & -1 & -1/2 & 1 \\ -1/2 & 1/2 & 1 & 0 \\ 1/6 & 0 & -1/6 & 0 \end{bmatrix}$$

So the row cubic coefficients can be solved for by the values of the pixels and the $[X]$ inverse.

$$\begin{bmatrix} A_0 & B_0 & C_0 & D_0 \\ A_1 & B_1 & C_1 & D_1 \\ A_2 & B_2 & C_2 & D_2 \\ A_3 & B_3 & C_3 & D_3 \end{bmatrix} = \begin{bmatrix} P_{00} & P_{01} & P_{02} & P_{03} \\ P_{10} & P_{11} & P_{12} & P_{13} \\ P_{20} & P_{21} & P_{22} & P_{23} \\ P_{30} & P_{31} & P_{32} & P_{33} \end{bmatrix} \cdot \begin{bmatrix} -1/6 & 1/2 & -1/6 & 0 \\ 1/2 & -1 & -1/2 & 1 \\ -1/2 & 1/2 & 1 & 0 \\ 1/6 & 0 & -1/6 & 0 \end{bmatrix}$$

Using the coefficients, the values of point $R_0 - R_3$ can be solved for at position dx with the cubic equations:

$$R_0 = A_0 dx^3 + B_0 dx^2 + C_0 dx + D_0$$
$$R_1 = A_1 dx^3 + B_1 dx^2 + C_1 dx + D_1$$
$$R_2 = A_2 dx^3 + B_2 dx^2 + C_2 dx + D_2$$
$$R_3 = A_3 dx^3 + B_3 dx^2 + C_3 dx + D_3$$

Converted to matrix form:

$$\begin{bmatrix} R_0 \\ R_1 \\ R_2 \\ R_3 \end{bmatrix} = \begin{bmatrix} A_0 & B_0 & C_0 & D_0 \\ A_1 & B_1 & C_1 & D_1 \\ A_2 & B_2 & C_2 & D_2 \\ A_3 & B_3 & C_3 & D_3 \end{bmatrix} \cdot \begin{bmatrix} dx^3 \\ dx^2 \\ dx \\ 1 \end{bmatrix}$$

A vertical cubic must then be defined for the values of $R_0 - R_3$. Solving in the same fashion as before but with the vertical values:

$$\begin{bmatrix} R_0 \\ R_1 \\ R_2 \\ R_3 \end{bmatrix} = \begin{bmatrix} y_1^3 & y_1^2 & y_1 & 1 \\ y_2^3 & y_2^2 & y_2 & 1 \\ y_3^3 & y_3^2 & y_3 & 1 \\ y_4^3 & y_4^2 & y_4 & 1 \end{bmatrix} \cdot \begin{bmatrix} A_V \\ B_V \\ C_V \\ D_V \end{bmatrix}$$

Like the $[X]$ values, the $[Y]$ values are constant and can be define explicitly:

$$[Y] = \begin{bmatrix} -1 & 1 & -1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 8 & 4 & 2 & 1 \end{bmatrix}$$

So the inverse of $[Y]$ is:

$$[Y]^{-1} = \begin{bmatrix} -1/6 & 1/2 & -1/2 & 1/6 \\ 1/2 & -1 & 1/2 & 0 \\ -1/3 & -1/2 & 1 & -1/6 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

The final step is to solve for the pixel value at point P by evaluating the vertical cubic at point dy.

$$P = A_V dy^3 + B_V dy^2 + C_V dy + D$$

In matrix form:

$$P = [dy^3 \quad dy^2 \quad dy \quad 1] \begin{bmatrix} A_V \\ B_V \\ C_V \\ D_V \end{bmatrix}$$

Combining all of the matrix pieces together to form a single equation, P can be solved for by:

$$P = [dy^3 \quad dy^2 \quad dy \quad 1] \cdot [Y]^{-1} \cdot \begin{bmatrix} P_{00} & P_{01} & P_{02} & P_{03} \\ P_{10} & P_{11} & P_{12} & P_{13} \\ P_{20} & P_{21} & P_{22} & P_{23} \\ P_{30} & P_{31} & P_{32} & P_{33} \end{bmatrix} \cdot [X]^{-1} \cdot \begin{bmatrix} dx^3 \\ dx^2 \\ dx \\ 1 \end{bmatrix}$$

3.2.7 Triangular Interpolation

Description: Triangular interpolation determines the value at a given point by interpolating values from the three neighboring points. This method is particularly useful in interpolating irregularly spaced data. Method was developed to interpolate data in triangular mesh structures.

[TBD]

3.3 Regression and Least Squares Minimization

Regression is the process of using least squares minimization to statistically predict the mathematical model for a set of data. This process involves minimizing the error between the predicted equation and the sample data. The results of regression analysis are very typically used to determine an interpolation model to fill in missing data in a data set.

3.3.1 Least Squares Minimization

Least squares minimization is an approach for determining an equation for a set of observed data. The process involves determining a mathematical model (linear, polynomial, etc.) and adjusting the parameters of the equation to minimize the residuals (or errors) between the observed data and the math model. The sum of the residuals (ϵ) is calculated by taking the square of the difference between the equation and the observed data for every point and summing the results.

$$\epsilon = \sum_{i=1}^{n} (f(x_i) - y_i)^2$$

The minimum error value is found by setting the derivative of the residuals equation equal to zero.

$$\frac{d\epsilon}{dy} = 0$$

The equation $f(x)$ is nearly always multivariate so for the derivative of the residuals to be zero, the partial derivatives of all of its components must be zero as well. For instance in the case of a line:

$$f(x) = Ax + B$$

The partial derivatives of ϵ with respect to each of the coefficients must be equal to zero:

$$\frac{d\epsilon}{dA} = 0 \quad \text{and} \quad \frac{d\epsilon}{dB} = 0$$

Each of the partial derivatives becomes an equation in the least squares solution and then the coefficients are calculated by simply solving the system of equations. For the linear equation above:

$$\frac{d\epsilon}{dA} = \sum_{i=1}^{n} (Ax_i + B - y_i) \cdot x_i = 0 \quad \text{and} \quad \frac{d\epsilon}{dB} = \sum_{i=1}^{n} (Ax_i + B - y_i) = 0$$

The equations above reduce to a pair of equations with A and B as the unknowns.

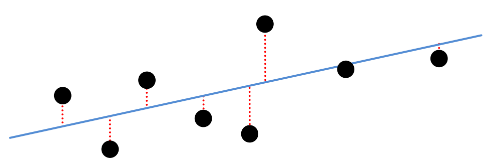
$$A\left(\sum_{i=1}^{n} {x_i}^2\right) + B\left(\sum_{i=1}^{n} x_i\right) = y_i x_i$$
$$A\left(\sum_{i=1}^{n} x_i\right) + B\left(\sum_{i=1}^{n} 1\right) = y_i$$

The solution of A and B represents the optimal line for the data set. For more detail on linear regression see *Linear Regression (Section 3.3.2).*

3.3.2 Linear Regression

<u>Description</u>: Linear regression calculates the equation of a line from a set of 2 or more data points. The line is determined by minimizing the residuals (or errors) between the line and the original points using least squares minimization.

This method of linear regression only works in two dimensions and non-vertical lines. For the 3D lines and vertical 2D lines, use *Parametric Linear Regression (Section 3.3.4)*.



The least squares minimization equation is:

$$\epsilon = \sum_{i=1}^{n} (f(x_i) - y_i)^2$$

Where y_i are the observed values and $f(x_i)$ is the y-value of the line at x_i. The equation of a 2D line is:

$$y = Ax + B$$

Plugging this value in to the regression equation gives

$$\epsilon = \sum_{i=1}^{n} (Ax + B - y_i)^2$$

To find the minimum residual error, the derivative of the residuals equation must be zero, which means all of the partial derivatives with respect to each coefficient must be equal to zero.

$$\frac{d\epsilon}{dA} = \sum_{i=1}^{n} (Ax_i + B - y_i) \cdot x_i = 0$$
$$\frac{d\epsilon}{dB} = \sum_{i=1}^{n} (Ax_i + B - y_i) = 0$$

These equations can be expressed in matrix form:

$$\begin{bmatrix} \sum x_i^2 & \sum x_i \\ \sum x_i & \sum 1 \end{bmatrix} \cdot \begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} \sum y_i x_i \\ \sum y_i \end{bmatrix}$$

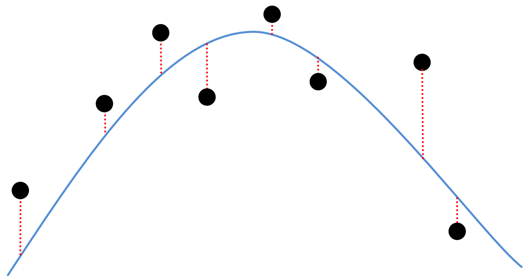
Solving for A and B:

$$\begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} \sum x_i^2 & \sum x_i \\ \sum x_i & \sum 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} \sum y_i x_i \\ \sum y_i \end{bmatrix}$$

See also:
- *Least Squares Minimization (Section 3.3.1)*

3.3.3 Polynomial Regression

Description: Polynomial regression calculates an n^th order polynomial from n+1 or more data points. The n^th order polygon is calculated by minimizing the residuals (or errors) between the polynomial and the original points using least squares minimization. The figure below shows the regression of a quadratic curve.



The least squares minimization equation is:

$$\epsilon = \sum_{i=1}^{n} (f(x_i) - y_i)^2$$

Where y_i are the observed values and $f(x_i)$ is the y-value of the polynomial at x_i. The equation of the quadratic is:

$$y = Ax^2 + Bx + C$$

Plugging this value in to the regression equation gives

$$\epsilon = \sum_{i=1}^{n} (Ax^2 + Bx + C - y_i)^2$$

To find the minimum residual error, the derivative of the residuals equation must be zero, which means all of the partial derivatives with respect to each coefficient must be equal to zero.

$$\frac{d\epsilon}{dA} = \sum_{i=1}^{n} (Ax^2 + Bx + C - y_i) \cdot {x_i}^2 = 0$$
$$\frac{d\epsilon}{dB} = \sum_{i=1}^{n} (Ax^2 + Bx + C - y_i) \cdot x_i = 0$$
$$\frac{d\epsilon}{dC} = \sum_{i=1}^{n} (Ax^2 + Bx + C - y_i) = 0$$

These equations can be expressed in matrix form:

$$\begin{bmatrix} \sum {x_i}^4 & \sum x_i^3 & \sum {x_i}^2 \\ \sum {x_i}^3 & \sum x_i^2 & \sum x_i \\ \sum {x_i}^2 & \sum x_i & \sum 1 \end{bmatrix} \cdot \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} \sum y_i {x_i}^2 \\ \sum y_i x_i \\ \sum y_i \end{bmatrix}$$

Solving for A, B, and C:

$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} \sum x_i^4 & \sum x_i^3 & \sum x_i^2 \\ \sum x_i^3 & \sum x_i^2 & \sum x_i \\ \sum x_i^2 & \sum x_i & \sum 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} \sum y_i x_i^2 \\ \sum y_i x_i \\ \sum y_i \end{bmatrix}$$

This form is consistent for any order polynomial. For instance a 5th order polynomial:

$$y = Ax^5 + Bx^4 + Cx^3 + Dx^2 + Ex + F$$

Works out to be:

$$\begin{bmatrix} A \\ B \\ C \\ D \\ E \\ F \end{bmatrix} = \begin{bmatrix} \sum x_i^{10} & \sum x_i^9 & \sum x_i^8 & \sum x_i^7 & \sum x_i^6 & \sum x_i^5 \\ \sum x_i^9 & \sum x_i^8 & \sum x_i^7 & \sum x_i^6 & \sum x_i^5 & \sum x_i^4 \\ \sum x_i^8 & \sum x_i^7 & \sum x_i^6 & \sum x_i^5 & \sum x_i^4 & \sum x_i^3 \\ \sum x_i^7 & \sum x_i^6 & \sum x_i^5 & \sum x_i^4 & \sum x_i^3 & \sum x_i^2 \\ \sum x_i^6 & \sum x_i^5 & \sum x_i^4 & \sum x_i^3 & \sum x_i^2 & \sum x_i \\ \sum x_i^5 & \sum x_i^4 & \sum x_i^3 & \sum x_i^2 & \sum x_i & \sum 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} \sum y_i x_i^5 \\ \sum y_i x_i^4 \\ \sum y_i x_i^3 \\ \sum y_i x_i^2 \\ \sum y_i x_i \\ \sum y_i \end{bmatrix}$$

The general form of the polynomial regression equation works out to be:

$$\begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_n \end{bmatrix} = \begin{bmatrix} \sum x_i^{2 \cdot n} & \cdots & \sum x_i^{n+1} & \sum x_i^n \\ \vdots & \ddots & \vdots & \vdots \\ \sum x_i^{n+1} & \cdots & \sum x_i^2 & \sum x_i \\ \sum x_i^n & \cdots & \sum x_i & n \end{bmatrix}^{-1} \cdot \begin{bmatrix} \sum y_i x_i^n \\ \vdots \\ \sum y_i x_i \\ \sum y_i \end{bmatrix}$$

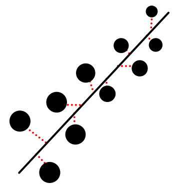
Where n is the order of the polynomial and $C_1 - C_n$ are its coefficients.

<u>*See also*</u>*:*
- *Least Squares Minimization (Section 3.3.1)*

3.3.4 Parametric Linear Regression

<u>Description</u>: Parametric linear regression is the process of determining a line using the parametric line equations. This method is applicable for any dimensionality of line and does not have the shortcomings that standard linear regression has. The line is determined by minimizing the residuals (or errors) between the line and the original points using least squares minimization.

The figure below represents a set of 3 dimensional points. The residuals measured with respect to the line direction, as opposed to a particular axis like the earlier regression techniques.



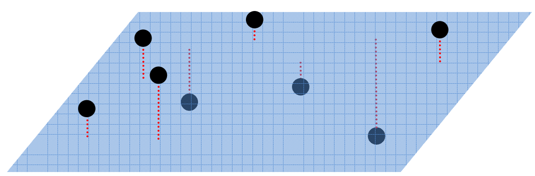
The parametric equations for a 3D line are:

$$x = x_0 + \alpha \cdot t$$
$$y = y_0 + \beta \cdot t$$
$$z = z_0 + \gamma \cdot t$$

[TBD]

3.3.5 Bilinear Regression (a.k.a Multilinear Regression)

Definition: Bilinear regression calculates the best fit plane through a group of 3 or more data points. The plane is calculated by minimizing the residuals (or errors) between the plane and the original points using least squares minimization.



The least squares minimization equation is:

$$\epsilon = \sum_{i=1}^{n} (f(x_i, y_i) - z_i)^2$$

Where z_i are the observed values and $f(x_i, y_i)$ is the y-value of the surface at x_i, y_i. The equation of the plane is:

$$z = Ax + By + C$$

Plugging this value in to the regression equation gives

$$\epsilon = \sum_{i=1}^{n} (Ax_i + By_i + C - z_i)^2$$

To find the minimum residual error, the derivative of the residuals equation must be zero, which means all of the partial derivatives with respect to each coefficient must be equal to zero.

$$\frac{d\epsilon}{dA} = \sum_{i=1}^{n} (Ax_i + By_i + C - z_i) \cdot x_i = 0$$
$$\frac{d\epsilon}{dB} = \sum_{i=1}^{n} (Ax_i + By_i + C - z_i) \cdot y_i = 0$$
$$\frac{d\epsilon}{dC} = \sum_{i=1}^{n} (Ax_i + By_i + C - z_i) = 0$$

These equations can be expressed in matrix form:

$$\begin{bmatrix} \sum x_i^2 & \sum x_i y_i & \sum x_i \\ \sum x_i y_i & \sum y_i^2 & \sum y_i \\ \sum x_i & \sum y_i & \sum 1 \end{bmatrix} \cdot \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} \sum x_i z_i \\ \sum y_i z_i \\ \sum z_i \end{bmatrix}$$

Solving for A, B, and C:

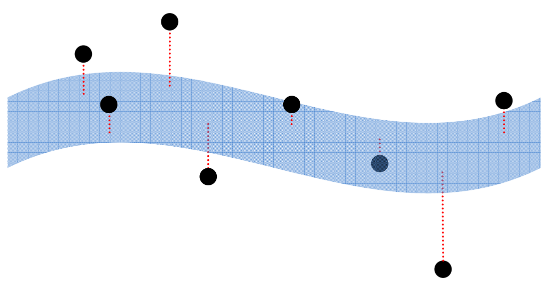
$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} \sum x_i^2 & \sum x_i y_i & \sum x_i \\ \sum x_i y_i & \sum y_i^2 & \sum y_i \\ \sum x_i & \sum y_i & \sum 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} \sum x_i z_i \\ \sum y_i z_i \\ \sum z_i \end{bmatrix}$$

See also:

- *Least Squares Minimization (Section 3.3.1)*

3.3.6 Bi-polynomial Regression

Definition: Bi-polynomial regression calculates a best fit polynomial surface through a group of n or more data points (where n represents the total number of variables in the polynomial surface equation). The surface is calculated by minimizing the residuals (or errors) between the surface and the original points using least squares minimization. The figure below represents a 2rd order surface.



The least squares minimization equation is:

$$\epsilon = \sum_{i=1}^{n} (f(x_i, y_i) - z_i)^2$$

Where z_i are the observed values and $f(x_i, y_i)$ is the y-value of the surface at x_i, y_i. The equation of the surface is:

$$z = Ax^2 + Bxy + Cy^2 + Dx + Ey + F$$

Plugging this value in to the regression equation gives

$$\epsilon = \sum_{i=1}^{n} (Ax_i^2 + Bx_i y_i + Cy_i^2 + Dx_i + Ey_i + F - z_i)^2$$

To find the minimum residual error, the derivative of the residuals equation must be zero, which means all of the partial derivatives with respect to each coefficient must be equal to zero.

$$\frac{d\epsilon}{dA} = \sum_{i=1}^{n}(Ax_i^2 + Bx_iy_i + Cy_i^2 + Dx_i + Ey_i + F - z_i) \cdot x_i^2 = 0$$
$$\frac{d\epsilon}{dB} = \sum_{i=1}^{n}(Ax_i^2 + Bx_iy_i + Cy_i^2 + Dx_i + Ey_i + F - z_i) \cdot x_iy_i = 0$$
$$\frac{d\epsilon}{dC} = \sum_{i=1}^{n}(Ax_i^2 + Bx_iy_i + Cy_i^2 + Dx_i + Ey_i + F - z_i) \cdot y_i^2 = 0$$
$$\frac{d\epsilon}{dD} = \sum_{i=1}^{n}(Ax_i^2 + Bx_iy_i + Cy_i^2 + Dx_i + Ey_i + F - z_i) \cdot x_i = 0$$
$$\frac{d\epsilon}{dE} = \sum_{i=1}^{n}(Ax_i^2 + Bx_iy_i + Cy_i^2 + Dx_i + Ey_i + F - z_i) \cdot y_i = 0$$
$$\frac{d\epsilon}{dF} = \sum_{i=1}^{n}(Ax_i^2 + Bx_iy_i + Cy_i^2 + Dx_i + Ey_i + F - z_i) = 0$$

These equations can be expressed in matrix form:

$$\begin{bmatrix} \sum x_i^4 & \sum x_i^3y_i & \sum x_i^2y_i^2 & \sum x_i^3 & \sum x_i^2y_i & \sum x_i^2 \\ \sum x_i^3y_i & \sum x_i^2y_i^2 & \sum x_iy_i^3 & \sum x_i^2y_i & \sum x_iy_i^2 & \sum x_iy_i \\ \sum x_i^2y_i^2 & \sum x_iy_i^3 & \sum y_i^4 & \sum x_iy_i^2 & \sum y_i^3 & \sum y_i^2 \\ \sum x_i^3 & \sum x_i^2y_i & \sum x_iy_i^2 & \sum x_i^2 & \sum x_iy_i & \sum x_i \\ \sum x_i^2y_i & \sum x_iy_i^2 & \sum y_i^3 & \sum x_iy_i & \sum y_i^2 & \sum y_i \\ \sum x_i^2 & \sum x_iy_i & \sum y_i^2 & \sum x_i & \sum y_i & n \end{bmatrix} \cdot \begin{bmatrix} A \\ B \\ C \\ D \\ E \\ F \end{bmatrix} = \begin{bmatrix} x_i^2z_i \\ x_iy_iz_i \\ y_i^2z_i \\ x_iz_i \\ y_iz_i \\ z_i \end{bmatrix}$$

Solving for A - F

$$\begin{bmatrix} A \\ B \\ C \\ D \\ E \\ F \end{bmatrix} = \begin{bmatrix} \sum x_i^4 & \sum x_i^3y_i & \sum x_i^2y_i^2 & \sum x_i^3 & \sum x_i^2y_i & \sum x_i^2 \\ \sum x_i^3y_i & \sum x_i^2y_i^2 & \sum x_iy_i^3 & \sum x_i^2y_i & \sum x_iy_i^2 & \sum x_iy_i \\ \sum x_i^2y_i^2 & \sum x_iy_i^3 & \sum y_i^4 & \sum x_iy_i^2 & \sum y_i^3 & \sum y_i^2 \\ \sum x_i^3 & \sum x_i^2y_i & \sum x_iy_i^2 & \sum x_i^2 & \sum x_iy_i & \sum x_i \\ \sum x_i^2y_i & \sum x_iy_i^2 & \sum y_i^3 & \sum x_iy_i & \sum y_i^2 & \sum y_i \\ \sum x_i^2 & \sum x_iy_i & \sum y_i^2 & \sum x_i & \sum y_i & n \end{bmatrix}^{-1} \cdot \begin{bmatrix} x_i^2z_i \\ x_iy_iz_i \\ y_i^2z_i \\ x_iz_i \\ y_iz_i \\ z_i \end{bmatrix}$$

<u>See also</u>:
- *Least Squares Minimization (Section 3.3.1)*