# Vector Function Fitting

## Necessary Knowledge

It is important to be informed about the least squares method of fitting functions over a Set of data points, to understand this particular version usage.  
Here are some sources:   
<http://mathworld.wolfram.com/LeastSquaresFitting.html>  
<https://en.wikipedia.org/wiki/Least_squares>

## Reasoning for Usage

The Problem with this “normal” kind of functions is apparent as soon as one tries to rotate them. This is not possible for polynomial functions, but necessary for the tracking and rotation of the lane that the vehicle is driving upon.  
For this the use of vector functions is the simplest choice to model the lane markings.

## Method

### Preparation

There is one condition for this to work: The points need to be ordered in the direction of the function that is desired. In this case the points need to be ordered in such a way, that the sequence of the points goes away from the car.

So: ym < yn , where m and n are indexes and m comes before n (if y is the axis, that points to front of and away from the car)

Disclaimer:   
This also works for other function types that are more complex, but it is important that the points are already ordered in the shape that is desired for the function to look like.

If the points are ordered, then the proceeding step is to create new pairs of data points in the following way:

* Split the data points in there x and y coordinates
* Merge them with the corresponding index

The end result are two sequences of points where one point has either the shape ( i | xi ) or ( i | yi ).  
“i” is the former index corresponding to the point inside the ordered sequence.

### Fitting

Fit two polynomial functions, each corresponding to one of the newly created sequences. For this we use the least square fit like so:

Where , describe the Error and , describe the functions that correspond to the two sequences.

### Merging the functions to one vector function

After calculating the two best fitting functions all that is left to do is the merging of these to functions into one vector function, by stacking them into one vector like so:

“i” does not directly correspond to an index anymore though and in later steps the original data points are not of any use, so this can be cleaned up into:

Inside the code “t” was chosen instead of “r”, but to clarify before any confusion ensues: this “t” has nothing to do with time but was chosen because of the physical standard use of t inside these sorts of functions