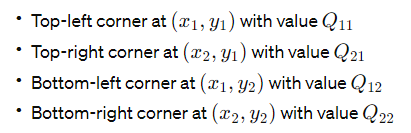
**Bilinear interpolation** is a straightforward method used to determine an interpolated value within a square (or rectangular) grid based on the values at the grid's four corner points. The main idea behind bilinear interpolation is to carry out linear interpolation first in one direction (e.g., horizontally) and then in the other direction (e.g., vertically).

**Explanation**

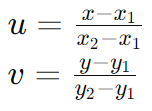
Suppose we have a rectangular cell defined by its four corner points with known values, and we want to find the value at a point ( *x*, *y* ) within this cell. Let's denote the corner points and their values as follows:



And the point where we want to interpolate the value is \((x, y)\).

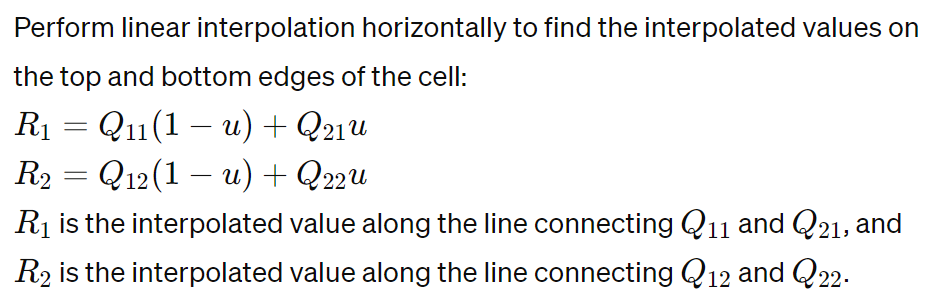
**1. Calculate Relative Position:**

The first step is to calculate the relative position of the point \((x, y)\) within the cell. This is done using the formulas:

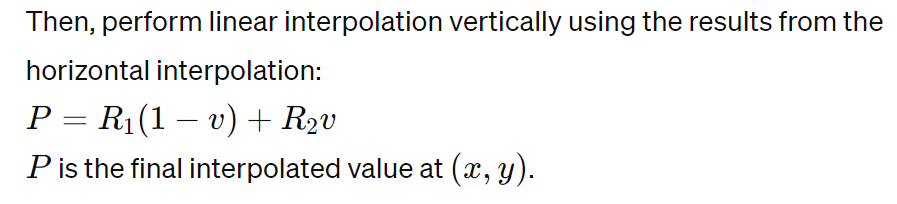


where (u) and (v) represent the normalized distances of ((x, y)) from the left and top edges of the cell, respectively.

**2. Horizontal Interpolation:**



**3. Vertical Interpolation:**



**Example**

In the context of the provided bilinearInterpolate function, Q\_{11}, Q\_{21}, Q\_{12}, and Q\_{22} correspond to the values mapped by `cornersMap` at the respective corners. The function calculates the relative positions (xRatio and yRatio, equivalent to (u) and (v) in the explanation) and performs the interpolations to find the interpolated value at **((x, y))**.

**Reference**

A good reference for understanding bilinear interpolation in more depth, including its applications and limitations, is the Wikipedia article on the subject:

- [Bilinear interpolation on Wikipedia](https://en.wikipedia.org/wiki/Bilinear\_interpolation)