

## HEATMAP INDEXING

We first define a **data matrix**  $M \in \mathbb{R}^{n \times m}$ . We then have a **tile size**  $\tau$  so that we have  $\lceil \frac{n}{\tau} \rceil = R$  row tiles and  $\lceil \frac{m}{\tau} \rceil = C$  column tiles. Because this data matrix cannot fit on the GPU it must be reshaped into a  $M^* \in \mathbb{R}^{p \times p}$  **reshaped matrix** where  $p$  is the max texture size on the computer (we assume that  $mn < p^2$  otherwise this doesn't work).

The data is thus reshaped into this new matrix in row major order under the mapping  $\phi: \{0, \dots, n-1\} \times \{0, \dots, m-1\} \rightarrow \{0, \dots, p-1\}^2$

$$\phi: (i, j) \rightarrow \left( \left\lfloor \frac{(i * m) + j}{p} \right\rfloor, ((i * m) + j) - p \left\lfloor \frac{(i * m) + j}{p} \right\rfloor \right).$$

Assume we have some fragment within a tile

$$F = ((a, b), (x, y)) \in [0, 1]^2 \times (\{0, \dots, C-1\} \times \{0, \dots, R-1\}) = \mathcal{F}$$

defined by **fragment coordinates**  $(a, b) \in [0, 1]^2$ , in normalized fragment space, and  $(x, y) \in \{0, \dots, C-1\} \times \{0, \dots, R-1\}$  the **tile coordinates**. Note that we are following the graphics convention, not the matrix convention - for matrices, the first coordinate represents row number (i.e going "up and down") but in graphics, the first coordinate represents the lateral coordinate, in this case for both the coordiante and the fragment coordinate.

In summary, every fragment in the total fragment space  $\mathcal{F}$  (over all tiles) is described by what tile it is in,  $(x, y)$ , and the coordinates within that tile in normalized fragment space  $(a, b)$ . We are then able to define the coordinates of the tile in terms of the original data matrix coordinates by the mapping  $\psi: \mathcal{F} \rightarrow \{0, \dots, n-1\} \times \{0, \dots, m-1\}$

$$\psi: ((a, b), (x, y) = T) \rightarrow (y * \tau + \lfloor b * \tau \rfloor, x * \tau + \lfloor a * \tau \rfloor).$$

This mapping takes the  $[0, 1]$  fragment, finds the coordinate within the tile that is correct independent of the underlying data (this is the floor term), and then offsets this coordinate by which tile of the data matrix we are in.

Thus the mapping  $\phi \circ \psi: \mathcal{F} \rightarrow \{0, \dots, p-1\}^2$  is a mapping from a fragment within a given tile to a spot on the reshaped data matrix, and this mapping only relies on knowing  $m, n, p, \tau$  and  $F$ , all of which are accessible for every fragment by binding these parameters as uniforms. By then normalizing the image of  $\phi \circ \psi$  i.e dividing both coordinates by  $p$ , we would get a final coordinate in  $[0, 1]^2$  which can be used to sample the reshaped data texture.