



MERIDIAN
Innovation

Mapping Temperature to a Colourmap

Application Note

Revision 0.2.0 – February 2020

1. MAPPING TEMPERATURE TO A COLOURMAP

The output temperature of the Bobcat needs to be mapped to a colourmap before it can be displayed on a screen.

A typical colourmap can be represented in the form of a lookup table that contains an RGB triplet, with each colour channel having the size of 256 and the entries within each channel bounded between 0 and 255. Therefore, the output temperature needs to be transformed into the corresponding index of the colourmap entry, ranging between 0-255 for zero-indexed and 1-256 for one-indexed.

The temperature to colourmap index transformation can be accomplished via the normalisation and casting of the temperature output through min-max feature scaling, which can restrict the range of the initial dataset between any arbitrary points. Below is the general form of the min-max feature scaling.

$$X' = a + \frac{(X - X_{min})(b - a)}{X_{max} - X_{min}} \quad (1)$$

Where a is the lower bound of the range, b is the upper bound of the range and X is the dataset to be normalised. With a and b equal to 0 and 255 respectively for the case of zero-indexed.

Other than the standard linear conversion, one can also change the dynamic range to compress or stretch the image colour for different temperature range by segmenting the temperature and choosing a and b appropriately.

A short pseudo code in C

```

/* Colormap */
const unsigned char red[256]={255,253,...,255};
const unsigned char green[256]={255,253,...,255};
const unsigned char blue[256]={255,253,...,24};

/* A function to colour a pixel on the display */
void draw(int row, int col, unsigned char r, unsigned char g, unsigned char b);
uint16_t gBuffer[80][62];

/* min-max feature scaling, gImageMax and gImageMin are the max and min value of gBuffer */
int imageRange = gImageMax - gImageMin;
for (int r = 0; r < IMAGE_ROWS; r++) {
    for (int c = 0; c < IMAGE_COLS; c++) {
        /* Equation 1 with a=0 and b=255 */
        value = (int)(gBuffer[r][c] - gImageMin) * 255 / imageRange;
        draw(r, c, red[value], green[value], blue[value]);
    }
}

```

2. REVISION HISTORY

Revision	Date	Notes
0.1.0	January 2019	Initial Version.
0.2.0	February 2020	Restyling of the document.

3. LEGAL INFORMATION

Meridian Innovation reserves the right to make changes to their products or to discontinue any product or service without further notice. SenXor™ logo is registered trademark of Meridian Innovation. All other trademarks are the property of their respective owners.

4. CONTACTS INFORMATION

For more information, please visit www.meridianinno.com

For sales inquiries, please email info@meridianinno.com

Headquarters: Meridian Innovation Pte. Ltd., 2 Vision Exchange, #11-08, Singapore

Company Registration Number: 201611173R