
camera_rotate

Lab Project

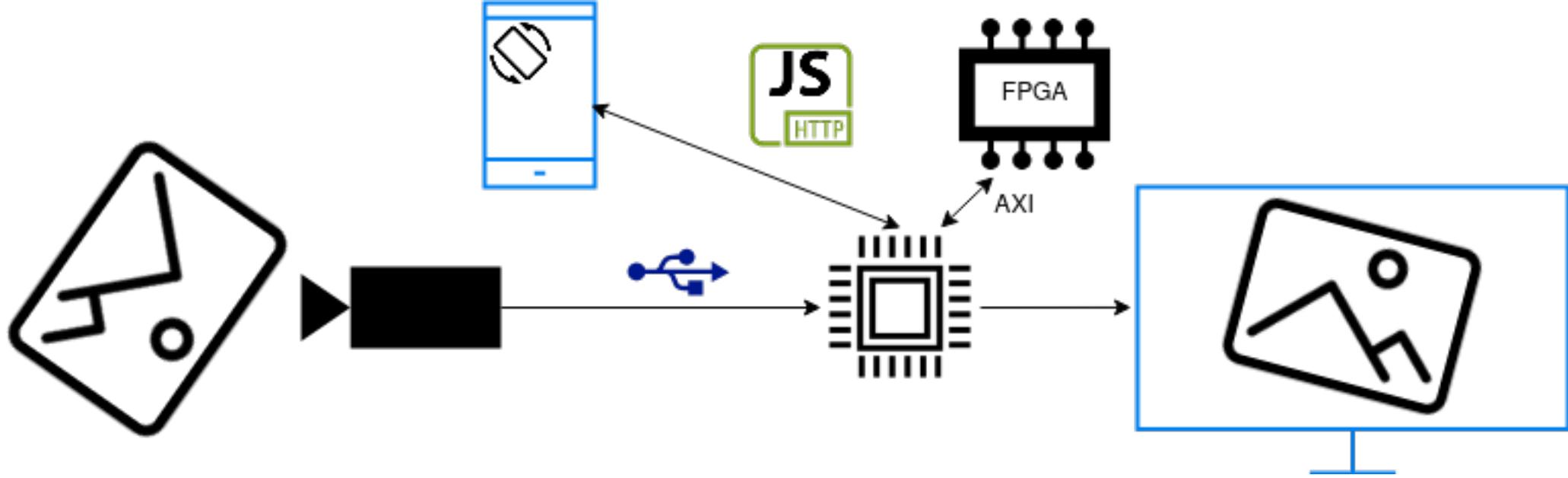
Erik Hennig

2023-11-29

Task

Automatically rotate live frames so they are always displayed upright, independent of the actual camera orientation.

Architecture



HLS function

- Only image rotation itself in HLS
- candidate function from Vitis Vision Library:
 - remap: swaps pixels using a relocation matrix
 - warpTransform: applies affine transformations to image
 - **rotate: rotates an image by 90, 180 or 270 degrees**

HLS function

```
void krnl_vadd(Pixel *src_ptr, Pixel *dst_ptr, uint16_t rows,
uint16_t cols, uint16_t direction) {
    #pragma HLS INTERFACE mode=s_axilite port=rows
    #pragma HLS INTERFACE mode=s_axilite port=cols
    #pragma HLS INTERFACE mode=s_axilite port=direction
    #pragma HLS INTERFACE mode=s_axilite port=return
    #pragma HLS INTERFACE mode=m_axi depth=__XF_DEPTH bundle=gmem0
port=src_ptr offset=slave
    #pragma HLS INTERFACE bundle=gmem1 port=dst_ptr // ...
```

```
const auto rotation = determine_rotation(direction);

if (rotation == None) {
    for (uint32_t i = 0; i < rows * cols; ++i) {
        dst_ptr[i] = src_ptr[i];
    }
} else {
    xf::cv::rotate<BITS_PER_PIXEL, BITS_PER_PIXEL, XF_8UC1, 32,
MAX_ROWS, MAX_COLS, XF_NPPC1>(src_ptr, dst_ptr, rows, cols,
rotation);
}}
```

Parameter

xf::cv::rotate<**IN_WIDTH,OUT_WIDTH,TYPE,** >

INPUT_PTR_WIDTH,OUTPUT_PTR_WIDTH,TYPE Tried different combinations,
only 8,8, XF_8UC1 worked

Parameter

`xf::cv::rotate<IN_WIDTH,OUT_WIDTH,TYPE,>` `ROWS,COLS,` `>`

`INPUT_PTR_WIDTH,OUTPUT_PTR_WIDTH,TYPE` Tried different combinations,
only 8,8, XF_8UC1 worked

`ROWS,COLS` $\leq 515 \times 515$ px (otherwise SEGFAULT in co-sim), careful with non-square images

Parameter

`xf::cv::rotate<IN_WIDTH,OUT_WIDTH,TYPE,TILE_SZ,ROWS,COLS,NPC>`

INPUT_PTR_WIDTH, OUTPUT_PTR_WIDTH, TYPE Tried different combinations,
only 8,8, XF_8UC1 worked

ROWS, COLS <= 515x515px (otherwise SEGFAULT in co-sim), careful with non-square images

TILE_SZ, NPC TODO

Ressource Usage and Performance

TODO

- Ressource usage (< 10%) independent from C0LS/R0WS

Problems

- Wrong documentation of Vitis Vision Library
 - “Direction of rotate, possible values are 90, 180 or 270”
 - `if (direction == 0) {} else if (direction == 1) {} else {}`

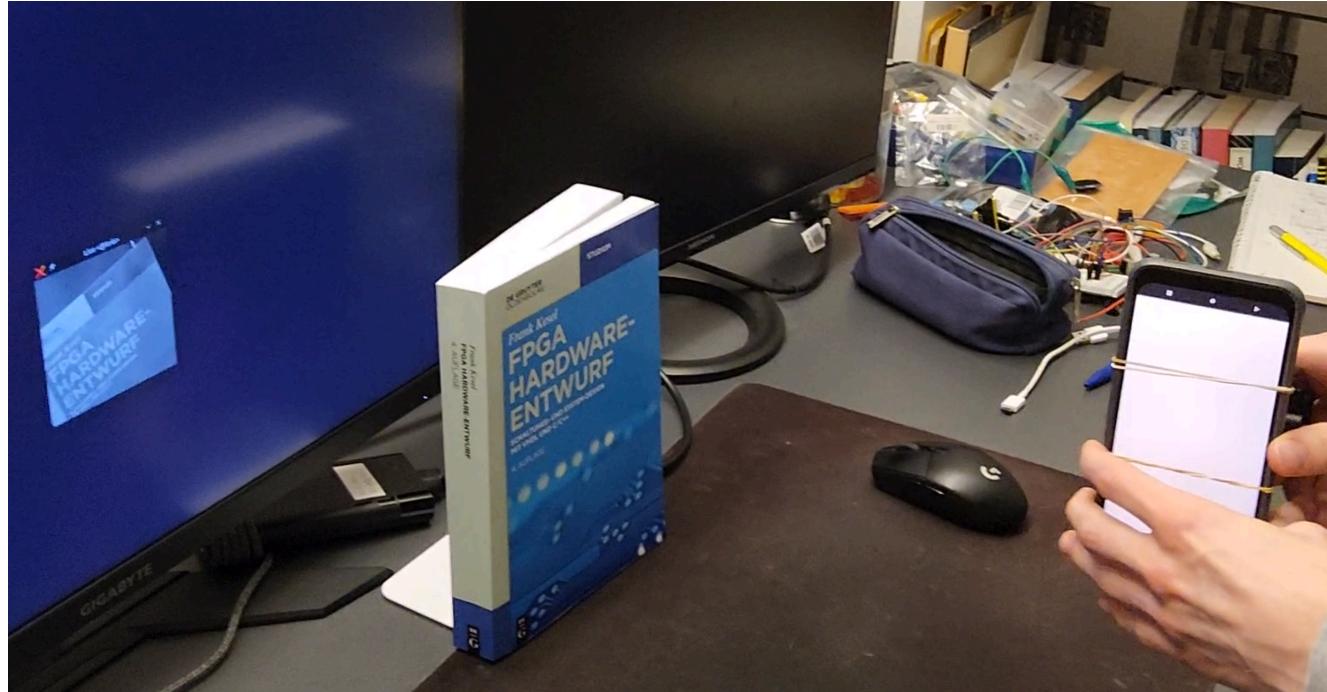
Problems

- Wrong documentation of Vitis Vision Library
 - “Direction of rotate, possible values are 90, 180 or 270”
 - `if (direction == 0) {} else if (direction == 1) {} else {}`
- Missing documentation of Vitis Vision Library
 - no include directive given for `xf::cv::rotate`

Problems

- Wrong documentation of Vitis Vision Library
 - “Direction of rotate, possible values are 90, 180 or 270”
 - `if (direction == 0) {} else if (direction == 1) {} else {}`
- Missing documentation of Vitis Vision Library
 - no include directive given for `xf::cv::rotate`
- Bad error messages
 - `INPUT_PTR_WIDTH` must be power of two → weird run-time errors
 - `SEGFAULT` in co-sim when `COLS/ROWS` too large

Demo



Thank you for your attention!
