LeapUVC Documentation

The Leap Motion Controller exposes a UVC interface for streaming image data. However, our use of UVC isn't quite standard. This document describes how to interact with the controller's UVC interface. Operating system interfaces such as DirectShow or Video4Linux have their own abstractions over UVC. How those interfaces correspond to this one is out of scope for this document. libuvc allows portable, direct UVC use, but may not be as efficient as the operating system interfaces.

NOTE: this document applies to firmware v ersion 1 .7.1 o r l ater. Previous verions of the firmware r equired a uthentication t o use.

Image data

The image encoding is reported as "YUY2". Any software which uses that format to decode will usually show a bright green/magenta distorted image. The image is actually an interleaving of two 8-bit monochrome images, one from each camera, starting with the left camera.

Here's some C for decoding a single pixel from a buffer:

```
struct pixel {
    uint8_t left;
    uint8_t right;
}

struct image {
    uint8_t *buffer;
    size_t height;
    size_t width;
}

struct pixel get_pixel_at(struct image image, size_t x, size_t y) {
    struct pixel pix;
    size_t left_idx = 2 * (y * image.width + x);
    pix.left = image.buffer[left_idx];
    pix.right = image.buffer[left_idx+1];
    return pix;
}
```

The supported resolutions are:

- 640 x 120
- 640 x 240

- 640 x 480
- 752 x 120
- 752 x 240
- 752 x 480

The maximum framerate at each resolution is fixed.

Image Metadata

At the bottom-right corner of every (pre-decoded) image are 12 special bytes that give metadata about the image, in particular the exposure (in microseconds), gain (in units of 1/16x), and whether the LEDs are on or off. Example C code for decoding these bytes is shown below.

```
const uint8_t* bytes = image.buffer + 2*image.width*image.height - 12;

int label1 = static_cast<int>(bytes[6] >> 4 & 0x1);
int label2 = m_DarkInterval ? static_cast<int>(((bytes[2] & 0xF) << 4) + (bytes[4] & 0xF))
int frameLabel = std::max(label1, label2);
int frameLabel &= 0x7F;

int exposure1 = static_cast<int>(bytes[6] & 0xF);
int exposure2 = static_cast<int>(bytes[8] & 0x1F);
int exposure = microseconds_i32((exposure1 << 5) + exposure2); // microseconds

int gain = static_cast<int>(bytes[10] & 0x1F);
```

Controlling the Camera

UVC supports "controls" for adjusting various properties of the image capture process. The LMC interprets the controls below. Setting any other controls is undefined behavior.

Zoom

The absolute zoom control (`libuvc_set_zoom_abs`) specifies the image exposure, in microseconds.

Gain

The gain control (`libuvc_set_gain`) is multiplexed, and controls analog gain, the FPS ratio, and the dark frame interval.

Unfortunately, due to this multiplexing, reading the gain control will only return the gain value.

```
#define GAIN_SELECTOR 0x4000
#define FPS_RATIO_SELECTOR 0x8000
#define DARK_FRAME_INTERVAL_SELECTOR 0xc000
#define PACK_VALUE(selector, value) ((selector) | ((value) & 0x3fff))
```

The gain specifies the *analog* gain as a scalar, in 1/16th increments.

The FPS ratio adjusts framerate as a proportion of the maximum for the configured resolution, in 1/1000 increments.

When the dark frame interval is N, the IR LEDs will be turned off for one frame, after N frames. When it is 0, the IR LEDs will always be on.

Brightness

The brightness control (`libuvc_set_brightness`) specifies the *digital* gain as a scalar, in 1/4th increments.

Gamma

The gamma control (`libuvc_set_gamma`) is a boolean and, if true (nonzero) the image will be in a non-linear color space approximating `sqrt(x)`. Otherwise, it will be linear.

White Balance

The white balance control is used in a non-standard way. It allows a temporal pattern for the infrared LEDs to be set where periodic frames have the LED off, so that the ambient-lit scene may be analyzed. Such frames are known as “dark” frames, and mainly consist of infrared light coming from other sources such as incandescent or halogen bulbs. In the case of daylight, such frames can still be very bright and/or overexpose. The value sets the total number of frames that should pass before one dark frame occurs. A value of 2 means that every other frame is dark, which is good for implementing software background subtraction algorithms. Other special values include 0 (no dark frames) and 1 (all dark frames).

Contrast

The contrast control (`libuvc_set_contrast`) is multiplexed, and controls HDR, rotation, the indicator LEDs, and vertical zoom/center.

Unfortunately, due to this multiplexing, reading the contrast control will only return the last value set, and cannot be configured to read the vcenter or vzoom.

The most significant 10 bits are a value, and the lowest 6 bits are a selector.

```
#define HDR_SELECTOR 0
#define ROTATE_SELECTOR 1
#define LED0_SELECTOR 2 // Left LED
#define LED1_SELECTOR 3 // Center LED
#define LED2_SELECTOR 4 // Right LED
#define VCENTER_SELECTOR 5
#define VZOOM_SELECTOR 6
#define PACK_VALUE(selector, value) ((selector) | ((value) << 6))
```

Nonzero values for HDR/rotate/LEDx are considered true, 0 false.

Hardware

The Leap Motion Controller is a stereo camera using a pair of global shutter Aptina MT9V024 sensors. The native resolution is 752x480, although our typical resolution is cropped to 640x480 (VGA) and then further downsampled vertically to 640x240 via hardware binning. The pixel size is 6 microns.

Three high-powered infrared LEDs provide a consistent narrow-band illumination at 850nm of the scene in all lighting conditions, and IR-pass filters both inside the lens and in the window of the device block out other wavelengths to aid object segmentation. To provide the maximum illumination during the short exposure, capacitors are used to constantly draw energy from the power source before discharging. The LEDs are physically wired to the sensor's exposure pin, so they exactly correspond to when the shutter is open. The dark interval and LED controls described above can override it and turn LEDs off however, using an AND gate.

Known Issues

When authentication is turned off, webcam and videoconferencing software may attempt to stream from the Leap Motion Controller. You may have to close such software in order to access the camera through UVC.