Biometric Systems
Capturing of 2D Hand Geometry With Line Camera

Bc. Petr Stehlík <xstehl14@stud.fit.vutbr.cz>
Bc. Marek Beňo <xbenom01@stud.fit.vutbr.cz>
Bc. Richard Wolfert <xwolfe00@stud.fit.vutbr.cz>

1 Assignment

The aim of the project is to assemble a mechanical device and create software for acquiring 2D hand geometry using a line camera. A solution for camera mount and hand placement is proposed in the following sections for best geometry acquisition and image reconstruction.

The document is structured as follows: in section 2 we describe the hardware part of the project, what hardware was used and we describe a solution for camera mount and hand placement. In section 3 we present the software needed for camera control and image reconstruction. In section 4 we detail created database of hand geometry data and method of acquisition. In last two sections 5 and 6 we sum up hardware and software solution and propose further improvements on our solution.

2 Hardware

In this section hardware used in the process of creating capture device is described, system schematic is laid out whilst taking into account the hardware limitations of chosen hardware, namely the focal length and depth of field of the chosen lens.

2.1 Hardware Setup

To be able to effectively scan 2D biometry of hand various hardware is necessary. Firstly we need framework to hold all other hardware together robust enought to hold weight and absorb vibrations generated by rail. This requirement is satisfied by using an aluminium framework.

Rail itself is mounted on this framework in a way to allow slide to move freely above the hand position. Camera and light are mounted on the slide moving along the rail to allow continuous capture of hand geometry with best lightning conditions. Control module consists of Raspberry Pi 3 and Arduino UNO running software necessary for capturing hand geometry and controlling the slide rail.

Camera

The selected camera is Basler raL6144-16gm. This camera provides us with the resolution of 6144×1 pixels with line rate up to 17 kHz and is controlled via ethernet connection and Pylon SDK. Captured pixels are in 8-bit grayscale color depth. The camera's shutter can be operated either via hardware or software trigger[1].

The camera is equipped with AF Nikon 50 mm f1.8D lens. The lens suits our needs for multiple reasons: It is relatively inexpensive, simple to use and has good depth-of-field control with aperture ranging from f/1.8 up to f/22.

Camera Mount

In order to acquire images with line camera either scanned object or camera have to move in smooth direct line to cover the whole scanned area. We decided to move the camera in order to minimize errors during scanning. The scanned hand is placed on stable platform and with only camera mount moving, human error is minimized and user experience is improved.

[1] The camera also enables "free-run mode".

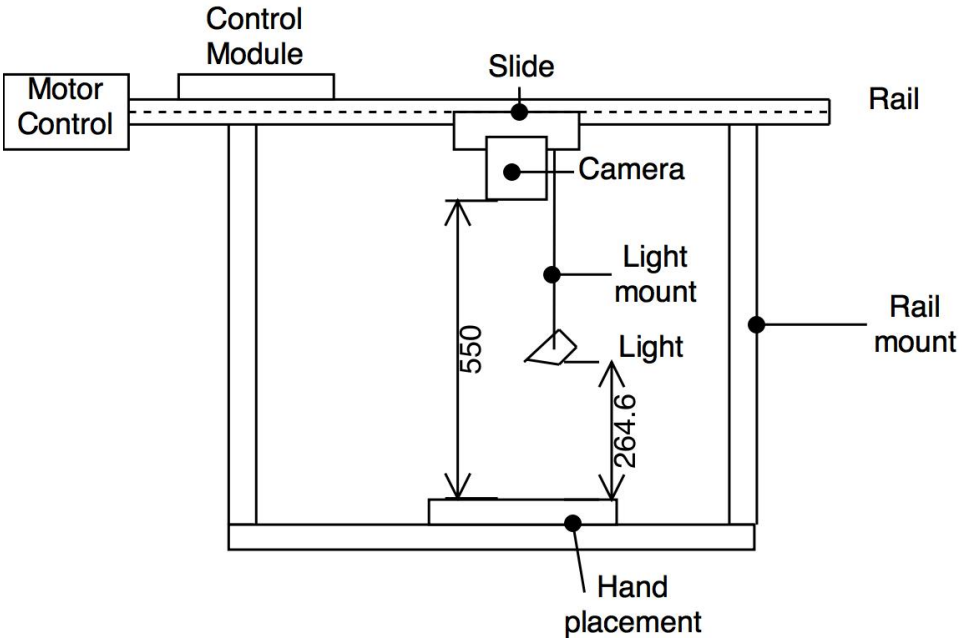

Figure 1: Schematic of hardware setup.

Slide moving along the rail houses the platform where the camera is mounted. This platform is facing down with camera and light mounted as seen in figure 2.1.

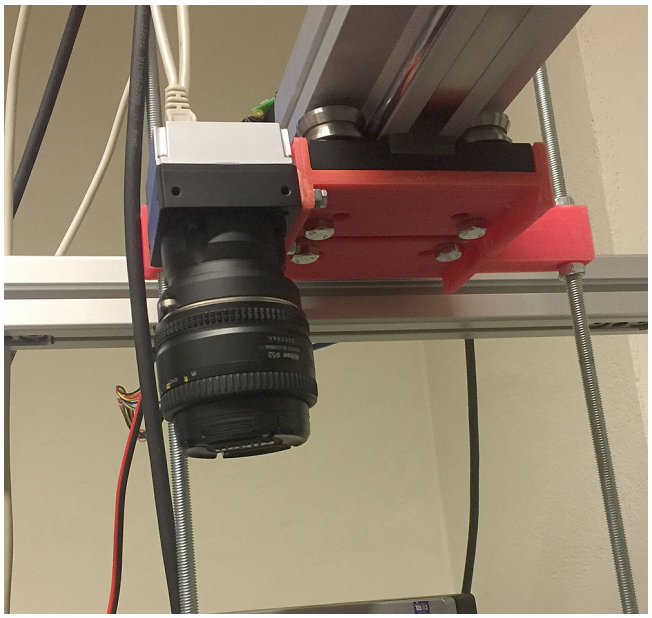

Figure 2: camera mount setup

Slide Rail & Stepper Motor

Slide moves on the rail in uniform motion along the rail axis. Slide movement is controlled by a stepper motor which controls direction and speed of the movement. Rail is mounted on the framework in the height of 60 cm. This allows us to a fine-tune the focus point of both the camera and the light.

The stepper motor itself is controlled by Leadshine EM705 Digital Stepper Drive which is controlled by pulses. We generate pulse width modulation (PWM) generated by Arduino UNO with our custom code generating 62 500 Hz PWM with 50 % duty cycle in Fast PWM mode.

The PWM can be calculated as follows:

$PWM_{Frequency} = 16000000/(Prescale\_Factor * 256)$

where $Prescale\_Factor$ can be set to 1, 8, 64, 256 or 1024. The default value is 64. Our code utilizes the Timer 2 available on Arduino UNO with prescale factor set to 1 which results in aforementioned 62 500 Hz PWM.

Apart from the PWM the Arduino UNO specifies direction in which the stepper motor should operate. The Leadshine EM705 Digital Stepper Drive also makes the `ENABLE` signal available which enables the output from it.

2.2 Slide Rail Control

The Arduino UNO itself is controlled via the serial interface where it actively listens for incoming messages. We defined a simple protocol:

```
<L|R><0 - 99999>
```

where the first characted specifies the direction of the movement. `L` stands for leftward movement which moves the slide towards the base of the rail. `R` standing for rightward movement move the slide away from the stepper motor.

After the direction specification calculation of duration stated in miliseconds of the movement follows. One milisecond of movement equals to 195.3125 mm as calculated by following formula:

$$v = \frac{PWM * duration[s]}{PPR} * Movement[mm]$$

where PPR denotes pulses per revolution. After substitution:

$$v = \frac{62500 * 10}{12800} * 4 = 195.3125mm$$

The Arduino UNO is connected to the Raspberry Pi 3 via USB which serves as a serial port and power source at once. The serial interface is set to the baud rate of 115 200 baud/s.

The Raspberry Pi 3 then sends messages in format specified in 2.2 and doesn't wait for any response since Arduino UNO serves as a standalone module and the Arduino UNO's output is only for debugging and informative purposes. Other options such as generating PWM directly on Raspberry Pi 3 or controlling Arduino UNO via GPIO pins were considered but during the development we stumbled upon problems with consistent and uniform signal generation caused by Raspberry Pi 3 itself which is not suited for such applications.

2.3 Light

To acquire high quality images of hand it is necessary to have proper light source. The line light chosen for our assignment is Chromasens Corona II type C with LED-control unit XLC4-1 which can produce lightning intesity up to 300 000 lux. The light is mounted directly below the slide platform angled 10 ° towards camera's line of sight.

LED-Control unit is controlled through the telnet interface via which one can set light intensity, set channels on or off and query various metrics and information. The LED-Control unit has a default IP address `192.168.87.234` and default credentials with username `admin` and password `chromasens`. We provide a simple interface for interacting with the control unit via a script `rail-control/tel_commander.py`.

2.4 Hand Placement

The hand is placed in the centre of the framework on a prepared platform which aligns and spreads the fingers in order to capture the same hand geometry during different scans. The hand is stable during image capture procedure which minimizes human error. Figure 2.4 shows placement of hand during scan.

2.5 Control

The whole system is connected using standard ethernet cables using switch and a DHCP server, which is setup up to assign IP addresses from `192.168.87.0/28` because of the default configuration of the

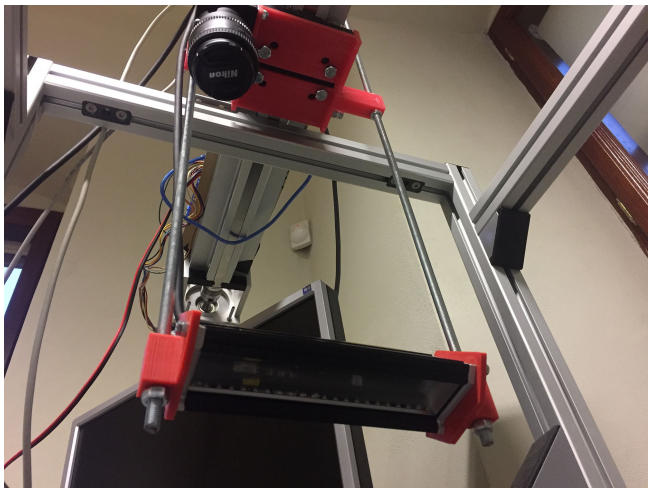

Figure 3: light mount setup

Figure 4: Hand placement during scan

LED-control unit. User enters the system through the Raspberry Pi 3 and issues commands on it with a prepared script `rail-control/capture.py`.

3 Software

Hardware solution for proposed device consists from three main parts: moving slide on rail, camera and light. Each of these parts needs to be controlled in order to scan images of hand geometry. All of the software as well as webserver storing results runs on control module - Raspberry Pi.

3.1 Used Software

Control software is implemented in python language for its ease of use and comprehension. With all necessary software running on Raspbery Pi minimum setup is necessary. When acessing system through Raspberry Pi control module user script `rail-control/capture.py` handles all neccessary communication with hardware parts while `src/capture` handles image capture using basler pylon 5 library and image reconstruction using OpenCV library. During testing, these `C++` libraries proved to be more reliable then their python counterparts and so they were interfaced with rest of the codebase.

3.2 Acquiring Images

To acquire images simple setup is necessary. Assembly and connection of all hardware parts is necessary with individual parts connected to Raspberry Pi control module according to schematics. Raspberry control module needs to be setup and software installed. When Raspberry is connected to internet image acquisition can be done through remote access. Initially, hand should be placed on designated platform and then software is run. Firstly lights are switched on, then camera starts recording and slide moves along the rail till it completes the scan. Image is then processed and saved in non-volatile memory using ".jpg" format. The light is then turned off, which indicates that image was successfully processed and saved. Finally, scan slide moves to initial position and device is immediately ready for another scan.

4 Hand Geometry Database

Scanned images are stored directly on Raspberry Pi immedietly after capture. Images are exposed through webserver and are availible for direct download through web browser.

Image naming convention allows to distinguish between different takes just by storing time of capture with filename as follows: result_<timestamp> where timestamp means standart Unix timestamp. High resolution images enable user to detect pores of sweat gland ducts in skin, whilst maintaining up to 2 scans per minute in proposed solution.

5 Achieved Results

Solution described in this report was assembled and implemented with excellent results. Integration of all hardware parts and software control results in images being scanned quickly in resolution exceeding 60 MPx. During testing it was proved to be effective to not only scan images of hand geometry but also fingerprints which leads to general purpose sensoric solution for capturing multiple biometric data. Furthermore, the solution is able to function with very limited resources necessary and provides instant access to scanned images via a web serber. Complete scan of both of the subject's hands can be done under 1 minute.

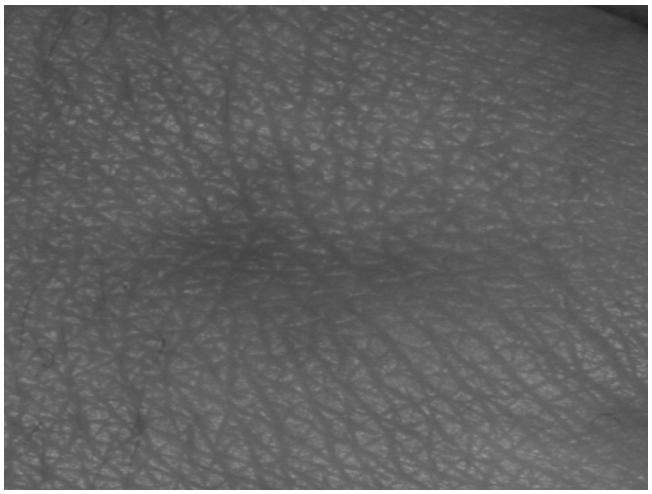

Figure 5: Dorsal detail of scanned hand

6 Summary

As previously dicussed in 5 proposed solution is able to quickly and accurately provide 2D hand geometry in great resolution. Solution was used to create initial image database with emphasisis on

Figure 6: Acquired image of a prostetic hand (left) and a cutout (right) from the image showing the detail

ease of use and precision so this database can be quickly expanded.

Solution proved to be effective to not only capture 2D hand geometry but also capture other biometric data such as fingertip scans. This results in ability to make multiple models from data and design efficient classification and identification algorhitms.

Improvements to this solution in the future could mean even better precision in acquiring image scans such as use of antivibration materials on slide platform to remove slight dillatations in images, enhancement of hand positioning using pillar to increase similarities between images. Further work should focus on solidifying the whole structure while making it more compact. With a second line camera the system can be expanded to capture 3D hand geometry. With acquired data the system be expanded for biometric identification and verification.