QUY TRÌNH ĐỒNG BỘ HỆ THỐNG CAMERA 3D CHỤP CƠ THỂ NGƯỜI

Mục tiêu:

- Tạo point cloud toàn bộ cơ thể người

Thiết kế hệ thống.

- 4 cameras, mỗi cặp 2 camera (stereo) tương đương với 1 camera 3D.

- Mỗi cặp stereo chụp 2 lần liên tiếp (khoảng cách thời gian giữa 2 lần chụp gần nhau nhất có thể):

+ Lần 1: Ảnh trái, phải có pattern Kinect để áp dụng thuật toán SGM, BM tạo ra depth/point cloud.

+ Lần 2: Camera trái/phải chụp ảnh IR, camera phải/trái chụp ảnh RGB.

- Hệ thống chuyển động thay đổi vị trí của 2 cặp stereo từ trên xuống dưới để chụp được toàn bộ cơ thể người theo 3 mức.

+ Ở mỗi mức có một công tắc hành trình để xác định vị trí 2 cặp stereo dừng lại để chụp 2 lần liên tiếp tạo ra point cloud ở 1 vị trí trên cơ thể người.

+ Sau khi chụp xong ở 3 vị trí, hệ thống tự động quay trở về vị trí cao nhất để chuẩn bị cho lần chụp tiếp theo.

+ 3 point cloud tại từng vị trí chụp sẽ được ghép lại tạo thành point cloud toàn bộ cơ thể.

Trong hệ thống này, bài toán đồng bộ bao gồm:

(1) Đồng bộ thời điểm chụp của 2 camera trong hệ thống stereo.

(2) Đồng bộ thời điểm chụp, thời điểm đèn sáng và thời điểm đóng mở bandpass-Ircut, bandpass-allpass: lần chụp thứ nhất, Kinect pattern sáng và cả 2 camera trong stereo chặn bandpass, lần chụp thứ 2, đèn IR sáng camera trái đóng IR-cut, camera phải ở chế độ allpass.

(3) Đồng bộ thời điểm hoạt động hệ thống ở bước 3 với thời điểm hệ thống dừng tại mỗi lần chụp trong quá trình trượt dọc theo cơ thể.

Cài đặt chương trình:

Source code: <Link>

Build project:

- Dependences: opencv, curl

Install opencv:

$ sudo apt-get install -y build-essential cmake git libgtk2.0-dev pkg-config libavcodec-dev libavformat-dev libswscale-dev python-dev python-numpy libtbb2 libtbb-dev libjpeg-dev libpng-dev libtiff-dev libjasper-dev libdc1394-22-dev

$ git clone <http://github.com/opencv/opencv.git>

$ cd opencv; mkdir release; cd release

$ cmake -D CMAKE\_BUILD\_TYPE=RELEASE -D CMAKE\_INSTALL\_PREFIX=/usr/local ..

$ make && sudo make install

$ cd ../../; rm -rf opencv

Install curl

$ sudo apt install curl libcurl4-openssl-dev

- Build project

$ git clone <http://github.com/ntddungbk/xxxxxxx.git>

$ cd <>

$ mkdir build; cd build; cmake ..; make

Kết nối camera và kit arduino vào cùng một mạng.

\* Cách tìm địa chỉ IP của camera và kit.

Dependences: fping

Cài đặt fping:

$ sudo apt install fping

Tìm địa chỉ IP

$ fping -c 1 -g <Dải địa chỉ IP>

$ arp -n | grep <Địa chỉ MAC>

Dải địa chỉ IP: là địa chỉ mạng của mạng mà thiết bị kết nối: 10.11.11.0/24.

Địa chỉ MAC của thiết bị hoặc một phần địa chỉ MAC

**1. Đồng bộ thời điểm chụp của hệ thống camera (1fps)**

Thiết lập: thực hiện lần lượt với từng camera trong hệ thống stereo.

+ Chuyển camera sang chế độ allpass hoặc IRcut để camera thu được hình ảnh (ánh sáng nhìn thấy) các con số trên stopwatch.

$ cd <path to synchronize directory>/build

$ ./ircut <IP of camera> on

+ Đặt stopwatch trước camera và ghi hình liên tục.

$ ./showCam <IP of camera> <makedir>

<IP of camera> là địa chỉ IP của camera, ví dụ: “10.11.11.120”

<makedir> = 1 nếu cần tạo thư mục “Image” để lưu lại các frame, = giá trị khác nếu không cần lưu lại các frame

+ Tinh chỉnh thời gian ghi hình của camera: đưa số lẻ giây của stopwatch thu được từ camera về vị trí chẵn lần giây bằng cách gửi liên tiếp một số lần lệnh thay đổi GOP, số lần gửi được xác định bằng số lẻ giây của stopwatch ở thời điểm trước khi gửi lệnh thay đổi GOP.

Mỗi lần thay đổi lệnh GOP thì thời điểm ghi hình frame tiếp theo cách frame trước đó khoảng 80-100ms

Quan sát số lẻ giây trên stopwatch qua các frame thu được. Chạy chương trình sau để đưa số lẻ giây thu được về xấp xỉ 0

$ ./synSecond <IP> <number>

<number> phần lẻ của giây

**2. Đồng bộ thời điểm chụp, đèn và đóng mở bandpass/allpass/Ircut**

Mục đích: Chụp 2 frame liên tiếp với khoảng cách thời gian gần nhau nhất có thể sao cho frame thứ nhất của cả 2 cam trong hệ thống stereo chứa pattern, frame thứ 2 của cam trái thu được ảnh RGB, cam phải thu được ảnh IR.

Thời điểm đóng mở bandpass-IRcut và bandpass-allpass đồng thời với bật-tắt đèn

Bài toán đồng bộ bao gồm:

(1) Đồng bộ thời điểm bắt đầu chụp và thời điểm bật-tắt đèn, xác định độ trễ ghi hình.

(2) Đồng bộ thời điểm bắt đầu chụp và thời điểm đóng mở bandpass-allpass bandpass-Ircut, xác định độ trễ ghi hình.

(3) Kết hợp (1) và (2) để thu được 2 frame gần nhau nhất có thể sao cho frame thứ nhất của cả 2 cam trong hệ thống stereo chứa pattern, frame thứ 2 của cam trái thu được ảnh RGB, cam phải thu được ảnh IR.

Setup:

- Kit arduino kết nối cùng mạng với Camera và PC (Laptop), kết nối đèn IR và pattern

<Hình ảnh minh họa kết nối>

**2.1 Đồng bộ thời điểm bắt đầu chụp và thời điểm bật đènIR-pattern.**

- Điều khiển đèn sử dụng Socket TCP/IP gửi tín hiệu điều khiển cho arduino bật/tắt đèn IR/Pattern.

- Xác định thời điểm gửi (Sau khi thu được frame thứ ***i***).

Điều chỉnh delay sao cho thu được 2 frames liên tiếp có sự chuyển đổi IR→ Pattern mà không bị mix (frame ***i+x+n*** và frame ***i+x+n+1***)

=> Xác định ***n***

Chạy chương trình xác định sự chuyển đổi

$ ./capPattern <IP> <Offset>

Chương trình xác định thời điểm bật đèn là sau khi thu được frame thứ 10 + <Offset>milliseconds.

Mở thư mục CapIR để xác định thời điểm frame xuất hiện đèn IR. Nếu có frame bị mix (frame IR có lần pattern hoặc ngược lại) thì chạy lại chương trình và điều chỉnh <Offset> đến khi nào không có frame bị mix thì lưu lại giá trị <offset> **x** và thứ tự frame chuyển đổi IR->Pattern **n**

**2.2. Đồng bộ thời điểm bắt đầu chụp và thời điểm đóng mở banpass-IRcut bandpass-allpass**

- Điều khiển đóng mở IR-cut sử dụng ONVIF

- Xác định thời điểm gửi (Sau khi thu được frame thứ ***j***)

Điều chỉnh delay sao cho thu được 2 frames liên tiếp có sự chuyển đổi đóng IR-cut→ mở IR-cut mà không bị mix (frame ***j+y+m*** và frame ***j+y+m+1***)

=> Xác định ***m***

Chạy chương trình xác định sự chuyển đổi

$ ./capIRcut <IP> <Offset>

Chương trình xác định thời điểm bật đèn là sau khi thu được frame thứ 10 + <Offset>milliseconds.

Mở thư mục CapIRcut để xác định thời điểm frame chuyển đổi chế độ bandpass-IRcut hoặc bandpass-allpass. Nếu có frame bị mix (frame IR có lần pattern hoặc ngược lại) thì chạy lại chương trình và điều chỉnh <Offset> đến khi nào không có frame bị mix thì lưu lại giá trị <offset> **x** và thứ tự frame chuyển đổi chế độ bandpass-IRcut (hoặc bandpass-allpass) **m**

**2.3. Đồng bộ ghi hình + đèn + IR-cut**

Xác định thời điểm gửi lệnh điều khiển đèn và thời điểm điều khiển đóng mở IR-cut sao cho thu được 2 frames liên tiếp đóng mở IR-cut trùng với bật tắt đèn

Gọi thời điểm gửi tín hiệu bật tắt đèn là sau khi thu được frame thứ **X**.

Gọi thời điểm gửi tín hiệu đóng mở IRcut là sau khi thu được frame thứ ***Y***.

Xác định mối liên hệ ***X,Y*** để thời điểm thu được 2 frames liên tiếp

**X + m = Y +n**

Với m, n xác định ở **mục (2.1) và (2.2)**.

Từ (2.2) và (2.3) xác định được sau ***m*** frame với offset ***x*** thì thu được frame bật đèn và sau ***n*** frame với offset ***y*** thì thu được frame có sự chuyển đổi chế độ bandpass-allpass, bandpass-IRcut.

Chạy chương trình tổng hợp lại kết quả

$ ./capPatIRcut <IP> <m> <x> <n> <y>

Quan sát frame thu được, điều chỉnh <x>, <y> để không bị mix giữa 2 frame liên tiếp có sự chuyển đổi chế độ.

**3. Đồng bộ hệ thống (2) và hệ thống chuyển động để chụp body**

- Xác định thời điểm hệ thống chuyển động dừng lại để gửi tín hiệu chụp ảnh cho hệ thống đã đồng bộ ở mục (2) bằng công tắc hành trình kết nối với arduino.

- Sau khi chụp xong hệ thống (2) gửi tiếp tín hiệu cho arduino điều khiển hệ thống chuyển động sang mức tiếp theo để tiếp tục chụp.

- Sau khi đã chụp đủ 3 vị trí, hệ thống (2) gửi tín hiệu điều khiển hệ thống về vị trí ban đầu để chuẩn bị lượt chụp tiếp theo.

- Mỗi mức dừng chụp có công tắc hành trình, khi công tắc được gạt sẽ trả về tín hiệu cho arduino.