**Trường đại học Bách Khoa Hà Nội**

**Viện Công Nghệ Thông Tin Và Truyền Thông**



**Báo Cáo Môn Đồ án I**

**Đề tài:**

Nhận dạng bàn tay và đếm số ngón tay

Giáo viên hướng dẫn: Th.s Nguyễn Đức Tiến

Sinh viên thực hiện: Nguyễn Trọng Hảo

Hà nội ngày 7 tháng 4 năm 2018

Mục lục

[**A.** **Nội dung** 3](#_Toc510897001)

[Phần 1: Tổng quan về đề tài và một số khái niệm 3](#_Toc510897002)

[**1.** **Ý tưởng** 3](#_Toc510897003)

[**2.** **Một số khái niệm** 4](#_Toc510897005)

[Phần 2: Tiến hành chương trình 8](#_Toc510897006)

[**1.** **Phân tích kết quả** 8](#_Toc510897007)

[**2.** **Một số hàm,control sử dụng trong chương trình** 9](#_Toc510897008)

[**B.** **Kết luận** 9](#_Toc510897009)

1. **Nội dung**

## Phần 1: Tổng quan về đề tài và một số khái niệm

1. **Ý tưởng** 
   * + - sử dụng phân đoạn màu da của con người để phát hiện đường viền của bàn tay kết hợp với sự phát hiện lồi tay để nhận ra cử chỉ và đếm số lượng ngón tay. Gồm 2 phần
       - Phần thứ nhất là 1 chương trình nhận biết da sử dụng 2 khung tham chiếu hình ảnh khác nhau .

* khung tham chiếu thực từ file nguồn(video file) với class YCrCbSkinDetector ().

Ảnh có chứa tường, trong nhà, người

Mô tả được tạo với mức tin cậy rất cao

* khung tham chiếu mặt na nhị phân với class HsvSkinDetector().



* Phần thứ 2 là 1 thư viện gồm 1 số lớp để đếm số ngón tay dựa vào các khoảng không gian màu khác nhau . YrcCb và Hsv là 2 không gian màu khác nhau được sử dụng và được xác định bằng các giới hạn trên và dưới.( YrcCb\_min, YrcCb\_max,Hsv\_min,Hsv\_max trong source code) trong đó
* Convex hull : để xác định vỏ lồi bao bọc xung quanh bàn tay.Sử dụng phân đoạn màu da để xác định vỏ lồi này.
* Convexity defect : để nhận dạng chuyển động khi ta di chuyển bàn tay và đếm số ngon tay sử dụng 3 điểm : StartPoint,EndPoint và DepthPoint.

1. **Một số khái niệm**

Ảnh có chứa bản đồ, văn bản

Mô tả được tạo với mức tin cậy rất cao

* Khởi tạo 3 điểm StartPoint,EndPoint,DepthPoint

+ CircleF startCircle = new CircleF(startPoint, 5f);

+ CircleF depthCircle = new CircleF(depthPoint, 5f);

+ CircleF endCircle = new CircleF(endPoint, 5f);

* Các câu lệnh vẽ lên các điểm StartPoint,EndPoint,DepthPoint

+ currentFrame.Draw(startCircle, new Bgr(màu), size);

+ currentFrame.Draw(depthCircle, new Bgr(màu), size);

+ currentFrame.Draw(endCircle, new Bgr(màu), size);

* Đường màu da cam là đường nối giữa startPoint và depthPoint để biểu hiện cho số ngón tay.Có bao nhiêu đường màu da cam thì sẽ có bấy nhiêu ngon tay được đếm.

Ảnh có chứa bản đồ

Mô tả được tạo với mức tin cậy rất cao

* câu lệnh để vẽ đường nối giữa startPoint và depthPoint

+ currentFrame.Draw(startDepthLine, new Bgr(Color.Orange), 2);

* Đường màu canh lá cây là biggest contour biểu diễn cho đường viền bao bọc bàn tay.

Ảnh có chứa kẹp giấy, văn phòng phẩm

Mô tả được tạo với mức tin cậy rất cao

* Sử dụng class private void ExtractContourAndHull(Image<Gray, byte> skin) để xác định đường này.
* Sử dụng câu lệnh currentFrame.Draw(currentContour, new Bgr(Color.LimeGreen), 2) để vẽ nên đường này.

## Phần 2: Tiến hành chương trình

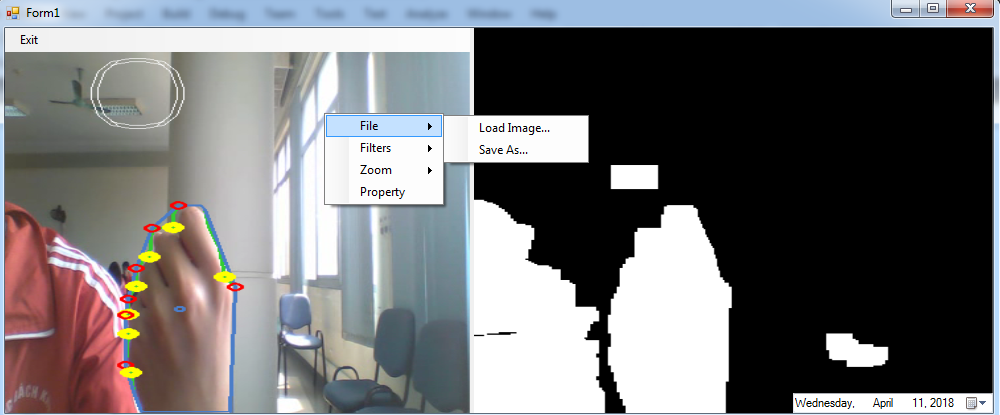
1. **Phân tích kết quả**

* Khi thực hiện chạy chương trình thì trên màn hình sẽ xuất hiện giao diện sau

Ảnh có chứa trong nhà

Mô tả được tạo với mức tin cậy cao

* Đây là 2 khung tham chiếu của bàn tay,phía bên tay trái là khung tham chiếu có được từ camera hoặc là video file, còn khung bên phải là mặt nạ nhị phân của nó.Chúng ta sử dụng các hình thái như làm xói mòn và xóa để loại bỏ một số tiếng ồn từ video file nguồn để có được một mặt nạ mượt mà và sạch sẽ hơn .
* Trên hình thì các điểm startpoint được biểu diễn bằng hình tròn màu đỏ, depthPoint biểu diễn bằng chấm màu đỏ và không có điểm endPoint.Đường màu xanh da trời bao bọc quanh bàn tay là convex Hull, đường màu xanh là cây đậm hơn là đường nối giữa StartPoint và DepthPoint và đường màu xanh lá cây nhạt là contour.
* Khi chúng ta di chuyển tay làm cho số ngón tay ít hơn hoặc nhiều hơn thì ngay lập tức đường màu xanh lá cây đậm sẽ thay đổi tương ứng và số lượng ngón tay sẽ được hiển thị ở bên góc trái phía trên của màn hình(trong trường hợp này là 4).Đồng thời Convex Hull cũng thay đổi kích thước của nó so với ban đầu để phù hợp với trạng thái hiện tại của bàn tay.
* Một số chăng năng khác để xem các thuộc tính của kết quả



1. **Một số hàm,control sử dụng trong chương trình**

* PointF(single,single) để khởi tạo 1 điểm với các tọa độ cụ thể
* LineSegment2D(Point2D<T>, Point2D<T>) để tạo 1 đoạn thẳng với điểm đầu và điểm cuối cụ thể
* CircleF(PointF, Single) để tạo 1 vòng tròn với bán kính cụ thể
* MCvFont(FONT, Double, Double) để tạo phông chữ (hiển thị số lượng ngón tay) với kích thước tự tạo.
* Một số control sử dụng : MenuStrip , imageBox , DateTimePicker , Button ,ToolTip , ContextMenuStrip ,Label

1. **Kết luận**

Chương trình đã hoàn thành một số tính năng chính nhưng vẫn còn nhiều điểm khuyết thiếu :

* + - * Chất lượng video file và camera không thật sự tốt,còn phụ thuộc nhiều vào chất lượng ánh sáng,góc quay.
      * Giao diện chưa được tốt.
      * Khi chaỵ chương trình camera trước thì khi chạy chương trình sử dụng video chậm hơn so với bình thường.
      * số lượng control vẫn đang còn ít.

Các giải pháp :

* + - * Tìm hiểu về xử lí ảnh trong C# để có thể thay đổi toạ độ theo mong muốn
      * Tìm hiểu thêm về giao diện và cách thiết kế để hoàn thiện giao diện