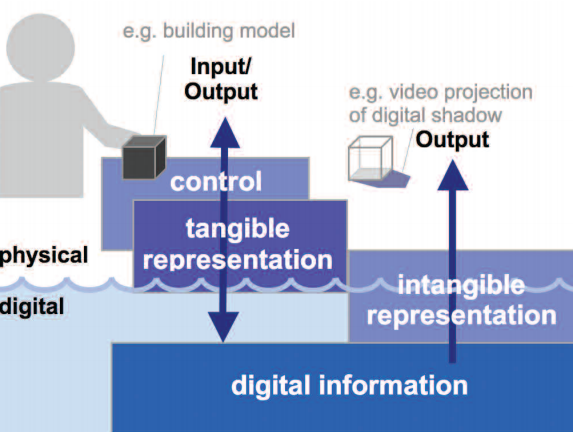
## 1 Giới thiệu

## 2 Các nghiên cứu liên quan

### 2.1 Tangible là gì?

- Định nghĩa tangible và môi trường hỗ trợ tangible là gì?.

Tangible user interface (TUI) là giao diện người dùng giúp con người có thể tương tác với thông tin kĩ thuật số thông qua môi trường vật lí. Mục đích của sự phát triển TUI là để tăng cường tính cộng tác, học hỏi, thiết kế bằng cách vật lí hóa những thông tin kĩ thuật số ví dụ như cầm nắm đồ vật, vật liệu [1]

Hình 1. Tương tác có tính chất tangible và không tangible

Theo như Hình 1, môi trường hỗ trợ tương tác Tangible là đưa yếu tố tương tác vật lý vào một môi trường chỉ hỗ trợ trình chiếu đơn thuần, đồng hộ hóa hai yếu tố này lại với nhau. Môi trường tương tác có yếu tố tangible giúp người dùng nắm bắt thông tin một cách trực tiếp, hiệu quả hơn do sự phản hồi xúc giác trong quá trình tương tác mang lại [2].

- Một số ứng dụng tiêu biểu:

- Giáo dục: [][] làm về ứng dụng….

- Urban plannning.

- [][]

### 2.2 Các kỹ thuật tương tác trong môi trường hỗ trợ tương tác tangible

Hệ thống sử dụng ….. như ởHình 2**.**

### 2.3 Hiển thị thông tin trong môi trường hỗ trợ tương tác tangible

## 3 Hệ thống xây dựng

### 3.1 Cấu hình hệ thống

### 3.2 Cách hệ thống làm việc

#### 3.2.1 Nhận dạng vật thể tương tác

#### 3.2.2 Nhận dạng vật thể tương tác

Việc nhận dạng chính xác vật thể tương tác đóng vai trò quan trọng trong ứng dụng vì nó giúp người dùng giao tiếp được với hệ thống và nâng cao trải nghiệm của người dùng. Để có thể triển khai nhanh ứng dụng với chi phí rẻ, chúng em quyết định dùng một camera màu hỗ trợ FullHD (webcam) hiệu …. để thực hiện lấy ảnh màu cho việc nhận dạng.

Chúng em áp dụng các phương pháp xử lý ảnh cơ bản để nhận dạng vật theo màu sắc theo quy trình sau:

Lấy mẫu ảnh

Tọa độ vật thể

Chuyển sang hệ HSV

Tìm vùng chứa vật thể

Lọc nhiễu

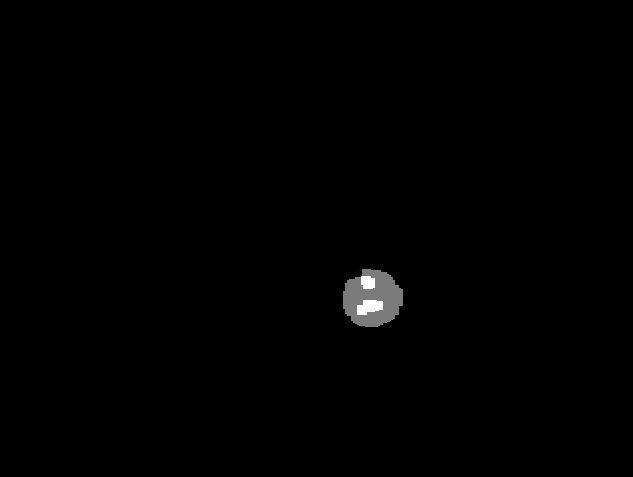
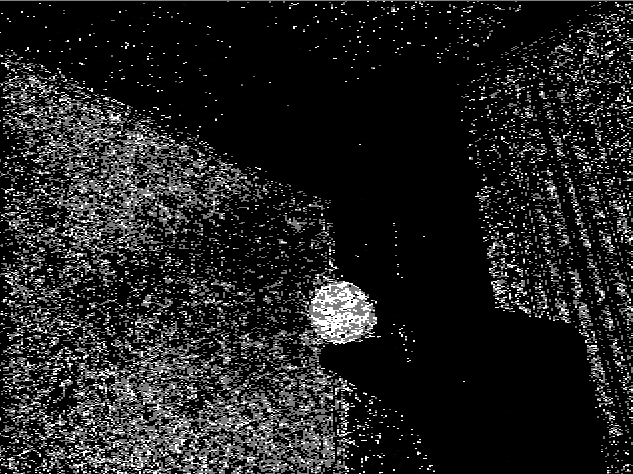
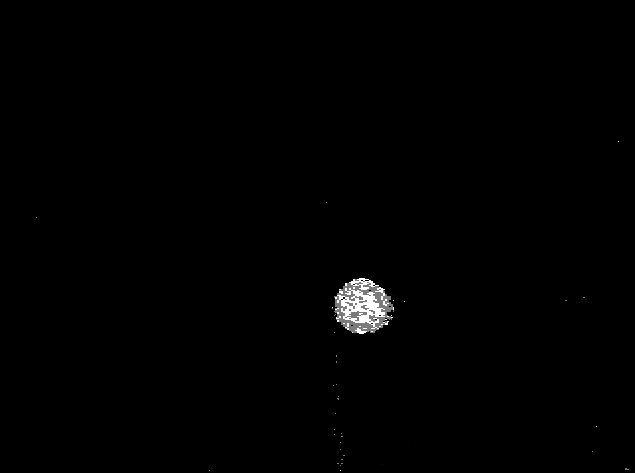
Tracking bằng Camshift

Video

Sơ đồ a

Sơ đồ a mô tả quy trình nhận dạng một vật thể theo kênh màu. Bước đầu tiên ta sẽ lấy vùng ảnh chứa vật thể cần nhận dạng, sau đó chuyển ảnh sang hệ màu HSV rồi tiến hành tính toán biểu đồ Histogram của vùng màu được chọn. Ta bắt đầu tìm vùng ảnh chứa vật thể cần nhận dạng. Dựa vào biểu đồ histogram, ta áp dụng phương pháp Back Projection để lấy được ảnh xác suất màu sắc có thể xuất hiện trong mỗi điểm ảnh như Hình (b). Sau đó đem chồng lên 1 lớp ảnh Mask (Hình (a)) đã lọc theo ngưỡng màu từ ảnh gốc. Kết quả thu được ta sẽ thu được một ảnh trắng đen như Hình (c) với vùng màu trắng là vùng có kênh màu cần nhận dạng. Tiến hành thí nghiệm nhận dạng một nắp chai màu xanh, ta thấy ảnh thu được sau khi tìm vùng chứa vật thể còn nhiễu khá nhiều, nguyên nhân là do các vùng lân cận có thể có kênh màu gần giống với vật thể. Để thu được kết quả chính xác hơn, ta tiến hành bước lọc nhiễu.

Để lọc các giá trị nhiễu này, ta áp dụng phương pháp lọc nhiễu hình thái (morphology) với 2 thao tác cơ bản là co ảnh (erosion) và giãn nở ảnh (dilation). Ý tưởng chính của phương pháp này là dùng phép toán erosion làm giảm kích thước của đối tượng, tách rời các đối tượng gần nhau, từ đó làm tiêu biến các điểm nhiễu nhỏ. Sau đó sẽ dùng phép toán dilation làm cho đối tượng ban đầu trong ảnh tăng lên về kích thước để vùng biên của vật thể tròn trịa, mịn hơn.



Hình a

Hình b

Hình c

Hình d

Toán tử erosion :

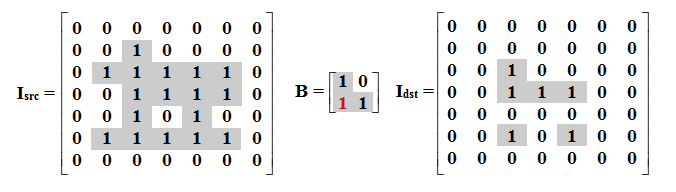


Trong đó:

* A: Ma trận điểm ảnh của ảnh nhị phân.
* B: Là phần tử cấu trúc.

Phép co ảnh erosion sẽ cho ra một tập điểm ảnh c thuộc A, nếu ta đi chuyển phần tử cấu trúc B theo c, thì B nằm trong đối tượng A.

Một ví dụ trực quan hơn, ta có ma trận điểm ảnh Isrc, ma trận điểm ảnh sau phép co Idst và cấu trúc phần tử B. Ứng với công thức ở trên, ta lần lượt đặt phần tử cấu trúc vào các điểm ảnh có giá trị 1 của ma trận điểm ảnh Isrc. Kết quả thu được là ma trận điểm ảnh Idst.



Toán tử dilation : với phương pháp tương tự như toán tử erosion, nhưng cấu trúc phần tử B sẽ mở rộng vùng chứa ô có giá trị

Sau quá trình lọc nhiễu, ta thu được Hình (d) để tiến hành việc truy vết ảnh. Ở đây chúng em dùng thuật toán Camshift (Continuously Adaptive Meanshift) trong OpenCV. Thuật toán này bản chất là áp dụng thuật toán Meanshift lên mỗi frame ảnh của nguồn video thu được từ webcame.

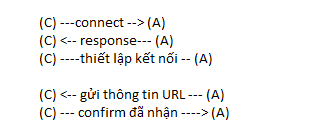
#### 3.2.2 Giao tiếp qua giữa các thiết bị tương tác

Theo như kiến trúc ứng dụng trên được trình như ở …., ứng dụng chia ra thành các module độc lập với nhau, mỗi module chịu một trách nhiệm riêng biệt như: module xử lý dữ liệu và render hình ảnh (A), module nhận dạng vật thể (B), module tương tác nội dung với người dùng như điện thoại, máy tính bảng (C). Thêm vào đó, các module cần một phương thức để giao tiếp, truyền đạt dữ liệu qua lại lẫn nhau một cách hiệu quả.

Trong giới hạn luận văn này, ứng dụng được triển khai trên môi trường mạng cục bộ LAN, nên chúng em quyết định dùng phương thức Socket để giao tiếp giữa các module. Socket là một phương pháp để thiết lập kết nối truyền thông giữa một chương trình yêu cầu dịch vụ (client) và một chương trình cung cấp dịch vụ (server) trên mạng LAN, WAN hay Internet.

Module (A) và (B) chạy cùng trên một máy tính và (B) đóng vai trò là client, chỉ có nhiệm vụ gửi tọa độ truy vết được cho (A) xử lý. (B) được viết bằng ngôn ngữ Python và dùng thư viện OpenCV để hỗ trợ việc truy vết. (A) đóng vai trò là server với (B) và chỉ tiếp nhận dữ liệu từ (B) nếu có và không có phản hồi lại. (A) sẽ lấy dữ liệu tọa độ truy vết được từ (B) để tiến hành căn chỉnh tọa độ phục vụ cho việc render hình ảnh

Module (A) và (C) chạy trên 2 thiết bị khác nhau và (A) đóng vai trò là server, (C) là client, giao tiếp với nhau theo kịch bản



Đối với hệ điều hành Android, để sử dụng SocketIO, ta thêm thư viện vào file gradle như ở Mã 1.

|  |
| --- |
| import java.net.Socket |

Mã 1

Để mở một kết nối Socket, ta thực hiện đoạn lệnh sau :

**Socket** mSocket = **new** **Socket** (mIPAddress, mPort);

Để gửi gói tin từ module C về module A, ta sử dụng đoạn lệnh

**PrintWrite** mSender = **new** **PrintWriter** (socket.getOutputStream(), true);

mSender.print(“Hello server”);

Để nhận c

Đối với Module B, trong ngôn ngữ Python, để sử dụng SocketIO, ta chỉ cần thêm thư viện socket và khởi tạo một đối tượng socket

import socket

socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)

và thực hiện việc gửi dữ liệu như sau

sock.sendto(data, (IP, PORT))

Ở đây, module (B) chỉ có nhiệm vụ gửi vị trí tọa độ vật thể truy vết được 1 cách liên tục mà không cần quan tâm việc phản hồi, nên chúng em đã dùng phương thức gửi UDP, cụ thể là dùng tham số socket.SOCK\_DGRAM để thiết lập việc gửi theo hình thức UDP. Phương thức gửi UDP tuy không thể xác nhận việc gửi có thành công hay không, và giới hạn kích thước gói tin gửi bị giới hạn, tuy nhiên tốc độ gửi sẽ nhanh hơn nhiều so với TCP. Ở ngữ cảnh này, việc dùng UDP là hợp lý

Đối với môi trường Unity, để sử dụng SocketIO, ta làm theo các bước sau

### 3.3 Các chức năng hỗ trợ

#### 3.3.1 Cho xem camera tại giao lộ

#### 3.3.2

**4 Đánh giá hệ thống**

**5 Kết luận**

## Kỹ thuật giao tiếp qua socket

* Đặt vấn đề

Theo như kiến trúc ứng dụng trên, ứng dụng chia ra thành các module độc lập với nhau, mỗi module chịu một trách nhiệm riêng biệt như: module xử lý dữ liệu và render hình ảnh (A), module nhận dạng vật thể (B), module tương tác nội dung với người dùng như điện thoại, máy tính bảng (C). Và các module cần một phương thức để giao tiếp, truyền đạt dữ liệu qua lại lẫn nhau một cách hiệu quả.

* Giải pháp

Trong giới hạn luận văn này, ứng dụng

TÀI LIỆU THAM KHẢO

[1] E. Sharlin, B. Watson, Y. Kitamura, F. Kishino, and Y. Itoh, “On tangible user interfaces, humans and spatiality,” *Pers. Ubiquitous Comput.*, vol. 8, no. 5, pp. 338–346, Sep. 2004.

[2] H. Ishii and Hiroshi, “The tangible user interface and its evolution,” *Commun. ACM*, vol. 51, no. 6, p. 32, Jun. 2008.