

TRƯỜNG ĐẠI HỌC LẠC HỒNG  
KHOA CÔNG NGHỆ THÔNG TIN

## BÁO CÁO NGHIÊN CỨU KHOA HỌC

Đề tài:

# XÁC ĐỊNH VẬT CẢN VÀ ỨNG DỤNG DÒ ĐƯỜNG ĐI CHO NGƯỜI MÙ

NGUYỄN THỊ KIM LOAN  
NGUYỄN THỊ NGỌC HUYỀN

BIÊN HÒA, THÁNG 12 / 2012



**TRƯỜNG ĐẠI HỌC LẠC HỒNG**  
**KHOA CÔNG NGHỆ THÔNG TIN**

## **BÁO CÁO NGHIÊN CỨU KHOA HỌC**

**Đề tài:**

# **XÁC ĐỊNH VẬT CẢN VÀ ỨNG DỤNG DÒ ĐƯỜNG ĐI CHO NGƯỜI MÙ**

**SVTH: NGUYỄN THỊ KIM LOAN**

**NGUYỄN THỊ NGỌC HUYỀN**

**GVHD: Ths. NGUYỄN PHÁT NHỰT**

**BIÊN HÒA, THÁNG 12 / 2012**



## LỜI CẢM ƠN

Lời đầu tiên, chúng em xin chân thành cảm ơn khoa Công Nghệ Thông Tin, trường Đại Học Lạc Hồng đã tạo điều kiện tốt cho chúng em học tập và thực hiện đề tài này trong suốt thời gian học tập tại trường.

Chúng em xin gửi lời biết ơn chân thành nhất đến thầy Ths.Nguyễn Phát Nhựt là người đã tận tình hướng dẫn, chỉ bảo chúng em trong suốt thời gian thực hiện đồ án. Và nhờ vào những lời nhận xét, góp ý của thầy đã giúp chúng em có một định hướng đúng đắn và nhìn ra được những ưu khuyết điểm của đề tài, để từ đó từng bước hoàn thiện hơn.

Đồng thời, chúng em cũng xin trân trọng gửi lời cảm ơn sâu sắc đến quý Thầy Cô trong Khoa đã tận tình giảng dạy, trang bị cho chúng em những kiến thức quý báu trong những năm học vừa qua. Và tất cả các bạn đã giúp đỡ chúng em trong suốt quá trình bắt đầu thực hiện đồ án.

Cuối cùng, chúng con xin gửi lòng biết ơn sâu sắc đến ba, mẹ, các anh chị và bạn bè đã ủng hộ, giúp đỡ và động viên chúng em trong những lúc khó khăn cũng như trong suốt thời gian học tập và nghiên cứu này. Đặc biệt là sự quan tâm lo lắng và hy sinh lớn lao của cha mẹ luôn là nguồn động lực cho chúng con phấn đấu trên con đường học tập của mình. Một lần nữa chúng con xin gửi đến những bậc sinh thành lời biết ơn sâu sắc nhất.

Mặc dù chúng em rất đã cố gắng hoàn thành luận văn trong phạm vi và khả năng cho phép, nhưng chắc chắn sẽ không tránh khỏi những thiếu sót, kính mong sự cảm thông và tận tình chỉ bảo của quý thầy cô và các bạn.

Biên Hòa, ngày....tháng....năm 2012

Sinh viên thực hiện

Nguyễn Thị Kim Loan - Nguyễn Thị Ngọc Huyền

## MỤC LỤC

<b>CHƯƠNG I. MỞ ĐẦU.....</b>	<b>1</b>
1.1. Dẫn nhập .....	1
1.2. Tổng quan tình hình nghiên cứu trong và ngoài nước liên quan đến đề tài .....	1
1.3. Tính cấp thiết của đề tài.....	6
1.4. Mục tiêu thực hiện đề tài .....	7
1.5. Phương pháp thực hiện .....	8
1.6. Đối tượng và phạm vi nghiên cứu .....	15
1.7. Sơ lược nội dung nghiên cứu.....	15
<b>CHƯƠNG II. NỘI DUNG THỰC HIỆN .....</b>	<b>17</b>
2.1. Tìm hiểu về Kinect .....	17
2.2. Các thư viện hỗ trợ hệ thống dẫn đường dành cho người mù .....	20
2.3. Phân tích và thiết kế hệ thống hỗ trợ dò đường bằng kinect .....	24
2.3.1. Các tiêu chí để xây dựng hệ thống .....	24
2.3.2. Phân tích .....	24
2.3.3. Thiết kế .....	25
2.4. Xây dựng chương trình điều khiển thiết bị dò đường bằng kinect.....	28
2.5. Tính toán góc quay tối ưu để tránh vật.....	34
2.5.1. Vật cản nằm phía bên trái người mù .....	34
2.5.2. Vật cản nằm phía bên phải người mù .....	35
2.5.3. Vật cản nằm phía chính giữa người mù.....	36
2.5.4. Tính toán đi thêm một đoạn an toàn sau khi tránh được vật cản.....	37
2.6. Kết quả thử nghiệm .....	38
2.6.1. Kết quả thử nghiệm trong nhà .....	38
2.6.2. Kết quả thử nghiệm ngoài trời.....	40

2.7.	Đánh giá chương trình .....	40
2.8.	Hướng phát triển của đề tài .....	41
2.8.1.	Về chương trình .....	41
2.8.2.	Về thiết bị .....	41
<b>CHƯƠNG III. KẾT LUẬN .....</b>		<b>45</b>
3.1	So sánh kết quả nghiên cứu với mục tiêu đặt ra ban đầu .....	45
3.2	Kết luận.....	45

## DANH MỤC HÌNH

Hình 1.1: “Chiếc nón kỳ diệu” của TS. Nguyễn Bá Hải.[6] .....	2
Hình 1.2: Thiết bị “PBNT” của nhóm sinh viên trường ĐH Khoa Học Tự Nhiên .....	3
Hình 1.3: Chó robot đang được thử nghiệm dẫn đường[3] .....	4
Hình 1.4: Sản phẩm sáng tạo NAVI.....	5
Hình 1.5: Người mù khi gặp vật cản phía trước, sẽ được cảnh báo bởi âm thanh gọi ý rẽ sang trái và đã tránh được vật cản an toàn. ....	7
Hình 1.6: Cảm biến siêu âm .....	8
Hình 1.7: Sự phản xạ của sóng siêu âm trên bề mặt vật liệu[15].....	9
Hình 1.8: Hiện tượng Forecasting[15] .....	10
Hình 1.9: Hiện tượng Crosstalk[15] .....	10
Hình 1.10: Phương pháp Optical Flow .....	11
Hình 1.11: Ảnh gốc và ảnh sau khi tách biên[19] .....	12
Hình 1.12: Phương pháp dò nền (Floor Finder)[18] .....	13
Hình 1.13: Phương pháp Stereo Vision.....	14
Hình 2.1: Camera Kinect của hãng Microsoft.....	17
Hình 2.2: Cấu tạo cơ bản của Kinect.....	18
Hình 2.3: Bên trong Kinect gồm RGB, IR Camera và IR Projector .....	19
Hình 2.4: Ảnh được thu độ sâu bởi Kinect[10] .....	19
Hình 2.5: Trình bày tổng thể mô hình 3 lớp[8]. ....	23
Hình 2.6: Sơ đồ tránh vật cản .....	25
Hình 2.7: Tính tọa độ Kinect trên người mù .....	26
Hình 2.8: Hệ trục có các chiều X, Y, Z tương ứng với các màu đỏ, xanh lá, xanh đương. ....	27
Hình 2.9: Phân ngưỡng khoảng cách từ người đến các chương ngại vật phía trước ....	27
Hình 2.10: Sơ đồ xử lý phát hiện và tách vật .....	28



Hình 2.11: Ảnh RGB và bản đồ độ sâu .....	28
Hình 2.12: Point Cloud.....	29
Hình 2.13: Pass Through .....	29
Hình 2.14: Voxel Grid .....	30
Hình 2.15: Minh họa cho RANSAC cho việc tìm đường thẳng trong mặt phẳng .....	31
Hình 2.16: Trước và sau khi sử dụng thuật toán RANSAC .....	33
Hình 2.17: Point Cloud sau khi thực hiện xong bước lọc và phân đoạn.....	33
Hình 2.18: Object Clusters .....	34
Hình 2.19: Vật cản nằm bên trái người mù, tính góc xoay sang phải. ....	35
Hình 2.20: Vật cản nằm bên phải người mù, tính góc xoay sang trái .....	36
Hình 2.21: Vật cản nằm ở giữa đường đi của người mù .....	37
Hình 2.22: Tính toán đi một khoảng an toàn.....	37
Hình 2.23: Hình ảnh được chụp vào ban đêm.....	38
Hình 2.24: Xác định vật cản, tường và sàn trong môi trường ánh sáng trong nhà.....	39
Hình 2.25: Khoảng cách vừa đủ cho người mù đi qua. ....	39
Hình 2.26: Dây nguồn của thiết bị Kinect.....	42
Hình 2.27: Hình ảnh cho Pin Varta .....	42
Hình 2.28: Cảm biến Kinect cho Windows.....	43
Hình 2.29: Ảnh minh họa hướng phát triển sản phẩm .....	43

## DANH MỤC TỪ VIẾT TẮT

<b>A</b>	<b>API</b> : Application programming interface
<b>C</b>	<b>CL</b> : Code Laboratories <b>CMOS</b> : Complementary metal – Oxide - Semiconductor
<b>F</b>	<b>FLANN</b> : Fast Library For Approximate Nearest Neighbors
<b>H</b>	<b>HDD</b> : Hard Disk Drive
<b>I</b>	<b>IR</b> : Infrared
<b>N</b>	<b>NI</b> : Natural Interaction <b>NUI</b> : Natural User Interface <b>NAVI</b> : Navigational Aids for the Visually Impaired
<b>O</b>	<b>OPENNI</b> : Open Natural Interaction
<b>Q</b>	<b>QVGA</b> : Quater Video Graphic Array
<b>R</b>	<b>RAM</b> : Random Access Memory <b>RANSAC</b> : Random Sample Consensus
<b>S</b>	<b>SDK</b> : Software Development Kit
<b>T</b>	<b>TOF</b> : Time Of Light
<b>U</b>	<b>USB</b> : Universal Serial Bus
<b>V</b>	<b>VTK</b> : Visualization Toolkit

## **CHƯƠNG I. MỞ ĐẦU**

### **1.1. Dẫn nhập**

Ngày nay, với những tiến bộ mới trong khoa học kỹ thuật và công nghệ đã giúp ích rất nhiều cho đời sống của con người. Mọi công việc hầu hết đều dần được tự động hóa và hiệu suất ngày càng được nâng cao với sự hỗ trợ của máy móc, thiết bị hiện đại. Một trong những công nghệ tiên tiến đang được áp dụng rộng rãi trong đời sống chính là công nghệ nhận dạng và xử lý ảnh.

Đây cũng là một trong những chủ đề được các nhà nghiên cứu trong và ngoài nước quan tâm và đang khai phá nhiều tính năng mạnh mẽ mà công nghệ mới này mang lại để xây dựng các ứng dụng hỗ trợ cho y tế, giáo dục, quốc phòng... Và đạt được những thành công đáng kể trong thời gian qua. Trong đó, không thể không nhắc đến các công nghệ được nghiên cứu, phát minh giành riêng cho người mù, những người không có những món tiền lớn để chữa trị hoặc không còn khả năng thấy ánh sáng nữa.

Trước đây, các thiết bị dùng để hỗ trợ cho người mù thường là cây gậy bằng cảm biến siêu âm, dùng để dò tìm đường đi và tránh được vật cản. Nhưng hiện nay một công nghệ hoàn toàn mới đã xuất hiện, thiết bị camera kinect với những tính năng vượt trội có thể khôi phục môi trường phía trước người mù dưới dạng 3D, cho ra những thông tin chính xác về khoảng cách vật cản phía trước, giúp người mù có thể nhận biết một cách dễ dàng và tránh được vật cản an toàn nhất. Và đây cũng là mục tiêu nhóm tác giả hướng đến với một ý nghĩa nhân văn, là đưa công nghệ hiện đại vào cuộc sống giúp ích được phần nào đó cho những người khiếm khuyết hòa nhập với cuộc sống số.

### **1.2. Tổng quan tình hình nghiên cứu trong và ngoài nước liên quan đến đề tài**

#### **1.2.1. Trong nước**

##### **Thiết bị “SPKT EYE” với cảm biến Laser[5]**

TS. Nguyễn Bá Hải, giảng viên trường ĐH Sư Phạm Kỹ Thuật TPHCM đã cùng với các học trò chế tạo chiếc nón “SPKT EYE” giúp cho người mù di chuyển vừa đưa vào dùng thử.

Bằng công nghệ chuyển thông tin thị giác thành thông tin xúc giác, khi cảm biến quét ngang vật, hệ thống sẽ xử lý và chuyển thành tín hiệu

rung nhẹ trên nón ở vị trí ngay giữa trán người dùng. Khi người mù đến gần vật cản thì tín hiệu rung sẽ mạnh hơn, và nếu không có vật cản tín hiệu rung sẽ tắt.



Hình 1.1: “Chiếc nón kỳ diệu” của TS. Nguyễn Bá Hải.

#### **Ưu điểm**

- Người mù có thể cảm nhận và tránh được vật cản trên đường đi bộ nhờ vào cảm biến laser gắn phía trước nón trong khoảng 0.5m - 3m.

#### **Nhược điểm:**

- Khâu xử lý chương trình phức tạp, việc nhận dạng vật cản để chuyển thành tín hiệu rung còn khá chậm, chưa đáp ứng đủ thời gian tránh vật.
- Còn một vấn đề nữa là góc mở tia laser không đủ lớn để quan sát được các vật nhỏ hay nằm dưới mặt đất, chỉ có thể quan sát được các vật phía trên, còn từ đầu gối xuống không thể nhìn thấy được, nên phải kèm thêm một cây gậy để dò tìm đường đi an toàn, gây thêm nhiều vướng víu trong quá trình di chuyển.

### Thiết bị dò đường “PBNT” với cảm biến siêu âm



Hình 1.2: Thiết bị “PBNT” của nhóm sinh viên trường ĐH Khoa Học Tự Nhiên

Thiết bị có tên “PBNT” sử dụng cảm biến siêu âm được gắn trên mô bàn tay, giúp người đi có thể dò tìm đường đi phía trước và tránh được vật cản nhờ vào tiếng “bíp” phát ra. Do nhóm sinh viên trường đại học khoa học tự nhiên đang phát triển.

#### **Ưu điểm:**

- Thiết bị nhỏ gọn được gắn trên mô bàn tay tiện lợi hơn cho việc dò đường.
- Phát hiện được những vật cản ở khoảng cách xa.

#### **Nhược điểm:**

- Dùng cảm biến siêu âm còn phải phụ thuộc vào hình dạng kích thước vật thể và điều kiện môi trường cụ thể, với những vật quá bé nằm sát mặt đất thì từ trên mô bàn tay sóng siêu âm sẽ không phát hiện được vì góc mở không đủ lớn nên việc va phải vào vật cản là điều không thể tránh.
- Trường hợp nữa là gặp nhiều vật cản máy sẽ phát ra tiếng kêu liên tục, mà không định được hướng đi nào an toàn để tránh

được vật để đưa ra các thông báo gợi ý, đó cũng là một bất lợi với sóng siêu âm.

- Thiết bị gắn trên mô bàn tay lâu sẽ gây mỏi cho người sử dụng khi thường xuyên đưa tay lên dò tìm đường đi.

### 1.2.2. Ngoài nước

#### Chó robot dẫn đường[3]

Công ty NSK đã phối hợp với trường đại học truyền thông Electro của Nhật, chế tạo ra một con chó robot dẫn đường cho người mù dựa trên bộ cảm biến kinect của Microsoft.

Đề tài sử dụng bánh xe thay cho chân khi chuyển động trên mặt phẳng, dưới chân robot cũng sẽ được gắn thêm cảm biến để cảnh báo vật chính xác nhất cho chủ nhân về môi trường xung quanh họ.



Hình 1.3: Chó robot đang được thử nghiệm dẫn đường

Ưu điểm:

- Thiết bị được tích hợp nhiều cảm biến nên việc dò tìm đường đi cho người mù chính xác hơn.

Nhược điểm:

- Kích thước chó robot quá lớn và cồng kềnh gây bất tiện cho người sử dụng.
- Do di chuyển bằng bánh xe nên sẽ khó khăn khi đi trên các địa hình khác nhau.

- Do tích hợp nhiều cảm biến nên việc tiêu thụ điện năng rất nhiều, và yêu cầu xử lý phức tạp.

### **Hệ thống NAVI giành cho người mù [12]**

NAVI (Navigational Aids for the visually Impaired) là sản phẩm sáng tạo của hai thạc sĩ Michael Zollner và Stephan Huber thuộc đại học Konstanz của Đức.

Thiết bị được gắn trên đỉnh đầu của người mù thông qua một chiếc mũ cứng và băng keo dán, hệ thống NAVI có thể giúp người mù xác định được các vật cản phía trước họ bằng cách phản hồi âm thanh và rung động trong phạm vi từ 0.5m-5m, để họ có thể tránh vật cản trong khi di chuyển.



Hình 1.4: Sản phẩm sáng tạo NAVI

Ưu điểm:

- Thiết bị có thể phát hiện được các vật thể nằm bên ngoài phạm vi kiểm soát nhỏ hẹp của cây gậy, dựa vào góc mở lớn.

Nhược điểm:

- Thiết bị kinect được đặt trên đỉnh đầu, thì tầm nhìn của thiết bị sẽ bị hạn chế bởi góc mở, vì vậy phải kèm theo một cây gậy dò đường bên dưới nên các thiết bị còn cồng kềnh khó khăn cho việc đi lại.
- Khi đặt ở đỉnh đầu sẽ làm hạn chế khả năng xử lý do các cử động của đầu.

- Việc không giới hạn khoảng cách từ kinect đến vật trong tầm nhìn 0.5 – 5m, nên ở một không gian rộng lớn sẽ xuất hiện nhiều chướng ngại vật ở xa không cần thiết, dẫn đến tốc độ xử lý chậm.

Tiếp nối những thành công đó, nhóm tác giả muốn đóng góp một phần sức nhỏ cho sự nghiệp tin học hóa vào đời sống nhất là xây dựng các ứng dụng cho người khiếm khuyết có thể sử dụng. vì vậy nhóm tác giả đi tìm hiểu và xây dựng ứng dụng trên thiết bị cảm biến kinect mới với đề tài “Xác định vật cản và ứng dụng tìm đường đi cho người mù”. Với mong muốn là khai thác được những tính năng mạnh mẽ mà thiết bị kinect mang lại và ứng dụng cho thực tiễn. Tìm ra những giải pháp tốt hơn thay thế các loại cảm biến còn hạn chế trước đây.

### **1.3. Tính cấp thiết của đề tài**

#### **Khảo sát thực trạng**

Hội nghị thường niên về phòng chống mù lòa và khoa học kỹ thuật ngành mắt Việt Nam năm 2012, đang diễn ra tại trung tâm hội nghị quốc tế Hà Nội. Vừa cho biết hiện nay cả nước có khoảng 400.000 [4] người mù hai mắt và nếu tính mù một mắt thì cả nước có tới 2 triệu người, chưa kể hàng năm số người mù mới lên đến hàng trăm nghìn người và tỷ lệ mù tồn đọng mỗi năm lên tới 150.000 người. Bệnh đục thủy tinh thể chiếm 66% nguyên nhân chính gây mù. Gần 1/3 số người mù không có tiền để chữa trị nên đành chấp nhận số phận và vô tình họ trở thành gánh nặng của gia đình và xã hội.

#### **Tính cấp thiết**

Khoa học kỹ thuật ngày càng phát triển các ứng dụng của nó không chỉ tiếp cận người bình thường mà còn phải tiếp cận và hỗ trợ hơn nữa những người khuyết tật, những người vốn đã gặp rất nhiều thiệt thòi và khó khăn trong cuộc sống.

Các ứng dụng được nghiên cứu ở trong nước sử dụng cảm biến laser hay siêu âm để dò đường cho người mù thì không quá mới mẻ, nó đã được đưa vào nghiên cứu và sử dụng rất nhiều trong và ngoài nước. Tuy không phủ nhận hiệu quả của nó mang lại, nhưng tính khả thi chưa được cao, khi sử dụng thiết bị còn bị phụ thuộc vào hình dạng kích thước vật thể và điều kiện môi trường cụ thể,



với những vật quá bé nằm sát mặt đất thì sẽ không phát hiện được vì góc mở không đủ lớn nên việc va phải vào vật cản là điều khó tránh.

Còn các ứng dụng mà các nhà nghiên cứu ngoài nước đang phát triển, dựa trên công nghệ mới của thiết bị Kinect thực sự mang đến những kết quả đáng kể. Song vẫn còn những mặt hạn chế ở khâu xử lý ảnh, hay thiết kế. Cần được khắc phục để mang lại hiệu quả tốt hơn.

Do nhu cầu người sử dụng đang cần một loại thiết bị tốt hơn và khắc phục được những nhược điểm trên nên tập thể giảng viên và sinh viên đã quyết định chọn đề tài này để nghiên cứu và cố gắng khắc phục những vấn đề còn tồn đọng chưa được giải quyết của những ứng dụng trên với mong muốn sẽ góp phần đáp ứng những nhu cầu cấp thiết đó.

#### 1.4. Mục tiêu thực hiện đề tài

Một trong những khó khăn thách thức mà người mù đang gặp phải là họ hoàn toàn không có khả năng phát hiện được các chướng ngại vật trên đường đi bộ. Vì thế họ sẽ gặp rất nhiều khó khăn cho những sinh hoạt hằng ngày của mình.



Hình 1.5: Người mù khi gặp vật cản phía trước, sẽ được cảnh báo bởi âm thanh gọi ý rẽ sang phải và đã tránh được vật cản an toàn.

Trong khuôn khổ của đề tài thực hiện, nhóm tác giả sẽ xây dựng phần mềm chạy trên máy tính để điều khiển thiết bị Kinect (giả lập thiết bị hỗ trợ dẫn đường) giúp người mù xác định được vật cản phía trước và tránh các chướng ngại vật trên quãng đường di chuyển của họ. Đề tài tập trung vào 2 mục tiêu chính sau:

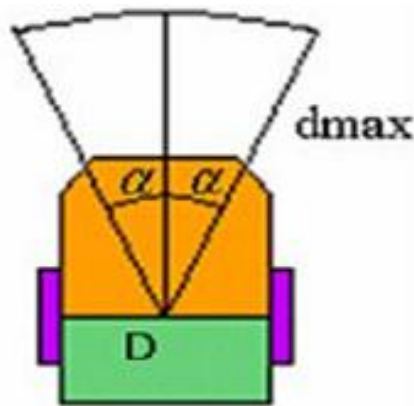
- Tập trung vào việc làm thế nào để phát hiện được vật cản và tính toán được khoảng cách đến vật cản phía trước để giúp người mang thiết bị tránh vật cản và tìm được đường đi an toàn nhất cho mình với môi trường trong nhà.
- Phát ra các tín hiệu âm thanh cảnh báo cho người sử dụng biết các thông tin phía trước.

### 1.5. Phương pháp thực hiện

Sau khi nhóm tác giả đã tìm hiểu qua các phương pháp phát hiện vật cản bằng nhiều loại cảm biến, thì thấy mỗi phương pháp có những ưu nhược điểm riêng như:

#### 1.5.1. Phương pháp phát hiện vật cản dùng cảm biến siêu âm

Cảm biến sẽ phát ra sóng siêu âm với góc mở nhất định. Khi đó nếu trong tầm quét của nó phát hiện chướng ngại vật thì sóng siêu âm sẽ phản hồi lại. Ta có thể đo khoảng cách bằng cách tính thời gian từ lúc sóng siêu âm phát ra đến lúc thu sóng về, sau đó kết hợp với vận tốc sóng siêu âm (khoảng 343 m/s) để biết quãng đường mà sóng đã đi.



Hình 1.6: Cảm biến siêu âm

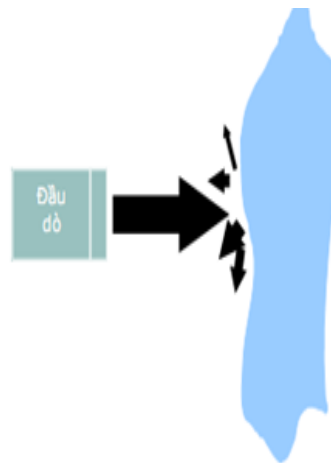
$D_{max}$  và  $\alpha$  phải đảm bảo sao cho cảm biến có vùng kiểm tra đủ rộng để khi tiến thẳng vẫn có thể nhận biết được vật cản.

Ưu điểm:

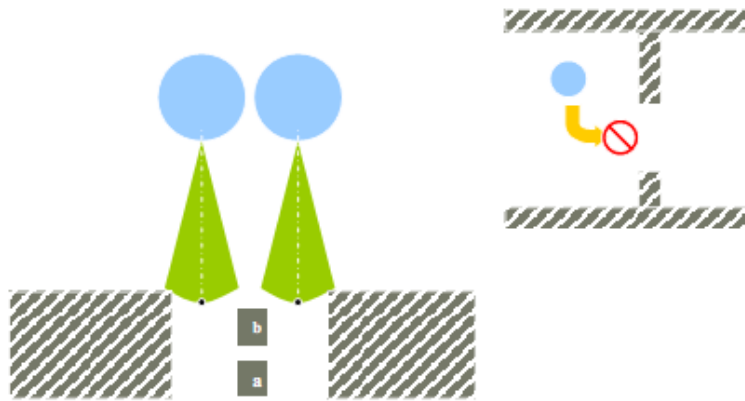
- Xử lý nhanh, kết quả tương đối chính xác.

Khuyết điểm:

- Cảm biến siêu âm chỉ nhận biết được vật cản khi mặt phẳng quét của cảm biến cắt ngang vật cản, do đó nó sẽ không phát hiện ra những vật cản nhỏ, thấp và nằm sát mặt đất.
- Do sử dụng sóng siêu âm và sự phản xạ của nó để tính khoảng cách và phát hiện vật cản nên giải thuật điều khiển khá phức tạp và phát sinh một số trường hợp sai khó khắc phục như: sai số lặp, hiện tượng Forecasting, hiện tượng đọc chéo Crosstalk.
  - Sai số lặp: là sai số luôn xảy ra với tất cả các thiết bị đo lường, trong đó có cả cảm biến siêu âm.
  - Hiện tượng Forecasting: là hiện tượng phản xạ góc sai lệch của cảm biến. Theo nguyên lý TOF để có khoảng cách đúng, cảm biến siêu âm phải hướng vuông góc với bề mặt chướng ngại vật cần đo. Nhưng một điều khó khăn ở đây là các chướng ngại vật không phải lúc nào cũng phẳng mịn, nên tia phản xạ có thể không tương ứng với góc tới. Các chùm tia phản xạ này có năng lượng phản xạ thấp hơn. Tuy vậy ở một khoảng cách nào đó, cảm biến siêu âm vẫn có thể ghi nhận được những tín hiệu phản xạ này. Kết quả thông số đọc về của cảm biến siêu âm bị lệch do góc mở của cảm biến siêu âm lớn.

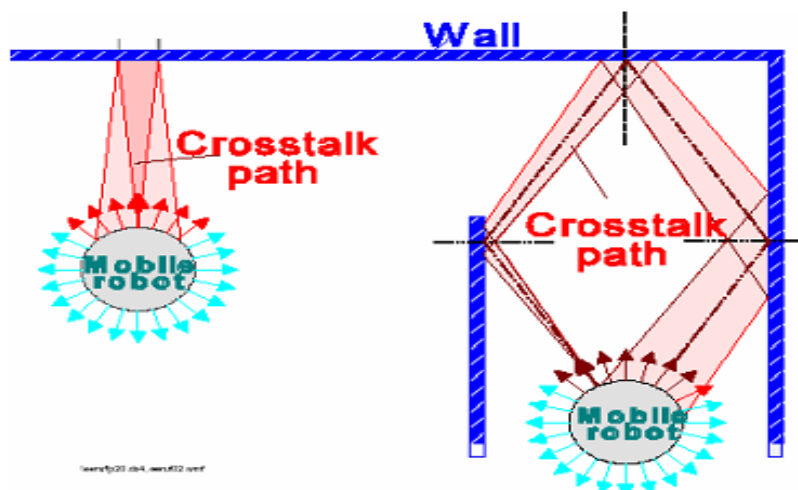


Hình 1.7: Sự phản xạ của sóng siêu âm trên bề mặt vật liệu[15]



Hình 1.8: Hiện tượng Forecasting[15]

- Hiện tượng Crosstalk: là hiện tượng mà cảm biến siêu âm này ghi nhận tín hiệu phản xạ hoặc trực tiếp từ cảm biến siêu âm khác, hoặc sau quá trình sóng siêu âm truyền đi và phản xạ qua các bề mặt quay lại cảm biến một cách không mong muốn.



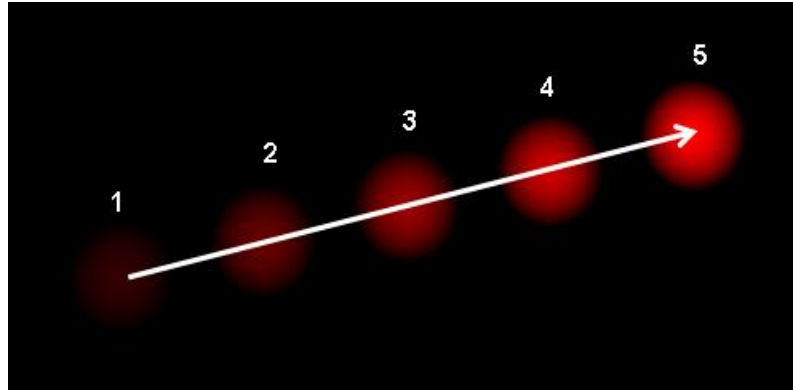
Hình 1.9: Hiện tượng Crosstalk[15]

### 1.5.2. Phương pháp phát hiện vật cản dùng một camera

Phương pháp này chủ yếu dựa trên các phân tích màu sắc hay sự thay đổi do chuyển động của các khung hình liên tiếp nhau, kỹ thuật xử lý ảnh đơn giản nhưng chỉ hiệu quả trong một số môi trường nhất định. Sau đây là một số phương pháp thường được sử dụng như: optical flow, edge detection, floor finder technique.

## Optical Flow

Là phương pháp tránh chướng ngại vật nhờ vào quan sát sự di chuyển của phần tử ảnh. Giải thuật sẽ tìm những phần tử ảnh đặc biệt trong ảnh tại một frame nào đó và quan sát độ dịch chuyển của nó ở frame tiếp theo. Vật càng gần thì độ dịch chuyển càng lớn.



Hình 1.10: Phương pháp Optical Flow

Ưu điểm:

- Phát hiện chướng ngại vật không phụ thuộc nhiều vào hình dạng vật.

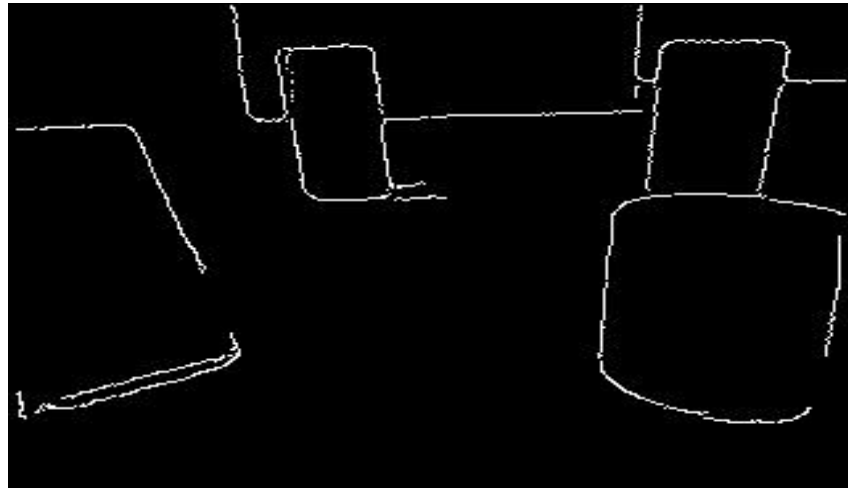
Khuyết điểm:

- Chỉ tối ưu với những vật có góc cạnh, có nhiều điểm đặc biệt. Trường hợp nếu gặp một bức tường màu trắng, thì sẽ không thể phân biệt được phần tử nào là phần tử tương ứng khi xét từ frame này sang frame khác hoặc nếu nền có hoa văn hay đường viền sẽ gây nhiễu do camera sẽ bám theo các phần tử đặc biệt trên nền.
- Ngoài ra vật cản nằm trong tầm nhìn của camera phải đứng yên, nếu chuyển động thì sẽ làm tang Optical Flow dẫn đến khó xác định được vật cản phía trước môi trường.
- Tốc độ xử lý chậm do yêu cầu tính toán nặng.

## Edge Detection

Phương pháp này dùng những kỹ thuật tách biên, cho ta ảnh chỉ hiển thị đường biên của vật thể như hình 1.11. Từ đó giúp ta phân biệt được

nền và các vật cản. Vật cản sẽ là những vật có viền bao quanh, còn nền là vùng không gian còn lại.



Hình 1.11: Ảnh gốc và ảnh sau khi tách biên[19]

Ưu điểm:

- Xử lý nhanh, dò tìm ra vật cản tốt, chính xác.

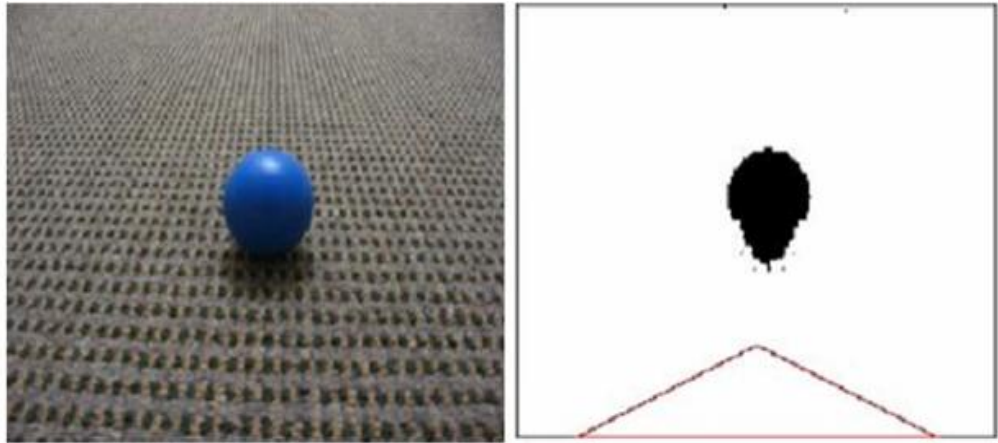
Khuyết điểm:

- Chỉ hoạt động tốt trong điều kiện nền đơn sắc và không có hoa văn họa tiết.

### **Floor Finder Technique**

Phương pháp này dựa trên màu sắc của các điểm ảnh, những điểm ảnh không trùng màu với nền thì được xem là vật cản.

Giả sử rằng vùng không gian nhỏ phía trước là không có vật cản. Ta sẽ lấy giá trị màu sắc các điểm ảnh trong vùng này thành một tập mẫu màu sắc của nền, rồi đem tập mẫu đó đi so sánh giá trị màu của từng điểm ảnh còn lại trong hình, lúc này ta sẽ xác định được điểm nào thuộc về nền, điểm nào thuộc về vật cản.



Hình 1.12: Phương pháp dò nền (Floor Finder)[18]

Ưu điểm:

- Đơn giản, hiệu quả, phát hiện vật cản chính xác, không phụ thuộc vào hình dạng và kích thước của vật cản.
- Tốc độ xử lý nhanh do yêu cầu tính không nhiều.

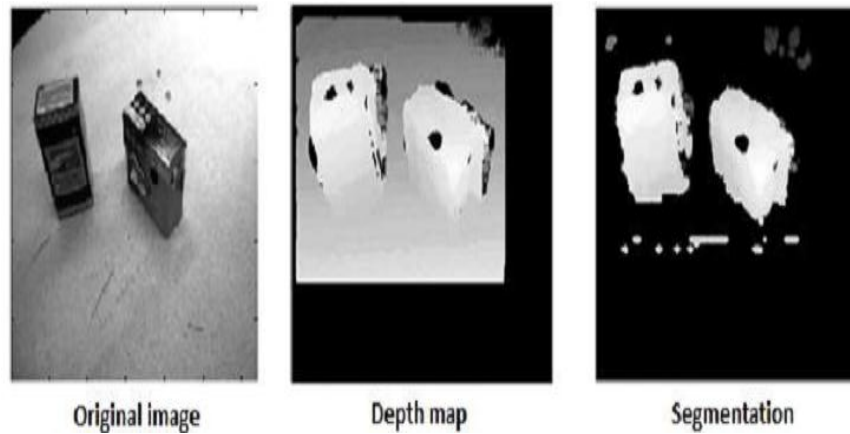
Khuyết điểm:

- Như các phương pháp dùng xử lý ảnh khác, do sử dụng màu sắc để nhận biết nên dễ nhầm lẫn giữa bóng đổ trên sàn và vật cản do bóng đổ có màu khác với nền. Ngoài ra, nếu vật có cùng màu với màu nền thì phương pháp này không đạt hiệu quả cao.

### 1.5.3. Phương pháp xử lý trên hai camera

Với phương pháp sử dụng một camera chỉ thu được ảnh trong không gian 2D và phát hiện vật cản dựa vào màu sắc, nên sẽ rất khó khăn trong việc xác định và tránh vật cản.

Vì vậy việc sử dụng hai camera trở lên sẽ giúp ta thu được đầy đủ thông tin từ môi trường phía trước hơn, và hạn chế được những nhược điểm từ một camera. Nhưng công việc xử lý ảnh sẽ phức tạp hơn do phải làm việc trong không gian 3D.



Hình 1.13: Phương pháp Stereo Vision

#### Phương pháp Stereo Vision

Ảnh màu được thu về thông qua hai camera, sau đó qua khâu hiệu chỉnh, khắc phục trước khi cho ra bản đồ độ sâu thông qua các giải thuật đặc biệt cùng với sự hỗ trợ của thư viện OpenCV. Ta thu được đầy đủ thông tin của vật cản trong môi trường như chiều cao, bề rộng hay khoảng cách từ camera tới vật sau khâu khôi phục 3D.

#### Ưu điểm:

- Phát hiện vật cản chính xác, hiệu quả, không phụ thuộc hình dạng, kích thước hay màu sắc vật, có thể phát hiện các vật thể trong không gian.

#### Khuyết điểm:

- Xử lý phức tạp, đòi hỏi sự chính xác cao trong khâu hiệu chỉnh. Tính toán khá nặng nên yêu cầu bộ vi xử lý cao.

#### 1.5.4. Các đặc điểm nổi trội của thiết bị Kinect

Với sự phát triển của công nghệ hiện nay thì tốc độ xử lý không còn là vấn đề khó khăn nữa. Qua tìm hiểu các phương pháp dùng để phát hiện vật cản, nhóm tác giả nhận thấy rằng “Camera Stereo Vision” nổi trội hơn cả. Nhưng kết quả lại phụ thuộc nhiều vào khâu hiệu chỉnh ban đầu, đòi hỏi xử lý phức tạp và tính toán khá nặng.



Vì thế, tác giả tìm đến thiết bị có khả năng hỗ trợ việc lấy bản đồ độ sâu trực tiếp mà không cần bận tâm nhiều đến việc hiệu chỉnh cho camera. Đó là thiết bị cảm biến Kinect.

- Kinect có khả năng thu về bản đồ độ sâu một cách chính xác mà không cần hiệu chỉnh nhiều.
- Nhờ sử dụng chip PS1080 của Prime Sense trên Kinect tính toán trực tiếp độ sâu nên nhanh, chính xác và ổn định hơn.
- Kết quả thu về trên Kinect chính xác, ổn định, tiêu tốn tài nguyên máy tính ít hơn nhiều so với việc sử dụng hai camera trong phương pháp Stereo Vision.
- Các bộ thư viện hỗ trợ mạnh mẽ như Kinect SDK, OpenNI, PCL...giúp lấy và xử lý thông tin dễ dàng, nhanh chóng.
- Kinect cùng mục đích với phương pháp Stereo Vision là thu về bản đồ độ sâu nhưng cách thực hiện lại có chút khác biệt, công nghệ mà Kinect sử dụng được xem là sự kết hợp giữa Stereo Vision và Range Finder (phương pháp đo đặc khoảng cách bằng sóng như laser, hồng ngoại hay sóng siêu âm), dù sử dụng giải thuật tính toán tương tự nhau.

#### **1.6. Đối tượng và phạm vi nghiên cứu**

- Môi trường thực nghiệm: Giới hạn với môi trường trong nhà, phát hiện vật cản tĩnh trên sàn, và nơi có ánh sáng yếu.
- Đối tượng nghiên cứu: người mù nhưng vẫn có thể đi lại bình thường.

#### **1.7. Sơ lược nội dung nghiên cứu**

##### **Chương I: Mở đầu**

- Tìm hiểu các đề tài trong và ngoài nước để thấy được tính cấp thiết để từ đó đưa ra mục tiêu thực hiện đề tài và đưa ra phương pháp phù hợp

##### **Chương II: Nội dung thực hiện**

- Tìm hiểu các thành phần của Kinect và các thư viện hỗ trợ cho người cho người mù nhằm xác định vật cản.
- Thực hiện phương pháp phát hiện vật cản với thiết bị cảm biến Kinect và tính toán các hướng quay an toàn giúp cho người mù có thể tránh vật.

- Thử nghiệm và đánh giá kết quả đạt được.
- Hướng phát triển của đề tài

### **Chương III: Kết luận**

- So sánh kết quả đã thực hiện với mục tiêu đã đặt ra
- Kết luận chương trình.

## CHƯƠNG II. NỘI DUNG THỰC HIỆN

### 2.1. Tìm hiểu về Kinect



Hình 2.1: Camera Kinect của hãng Microsoft

Kinect lần đầu tiên được giới thiệu vào ngày **01/06/2009** với tên gọi là Project Natal ở hội nghị thường niên E3. Kinect là một thiết bị mà Microsoft đã phát triển dành riêng cho hệ máy Xbox 360 mang đến trải nghiệm game không cần đến bất kì thiết bị điều khiển nào[11].

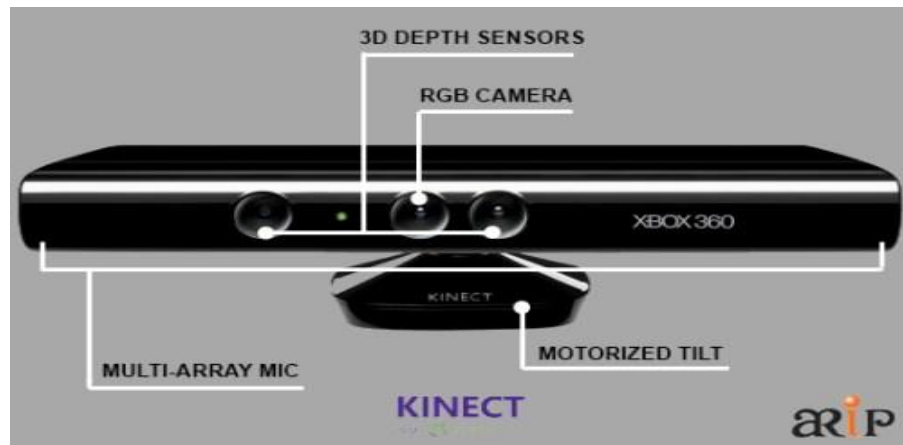
Bằng cách sử dụng hệ thống cảm biến và camera hồng ngoại, Kinect cho phép mọi người chơi game dựa trên các chuyển động tự nhiên của cơ thể như cử chỉ, hành động đem lại cảm giác thật cho người chơi game trên Xbox. Trong khi Wii hay PlayStation 3 Move phải theo dõi chuyển động của từng bộ phận thì thiết bị của Microsoft lại bao quát toàn bộ cơ thể, có thể xác định vị trí game thủ đang đứng, phân tích chuyển động và diễn giải từng ý nghĩa trong đó.

Tuy là một dòng sản phẩm thương mại với mục đích giải trí chơi game 3D trên Xbox, song hiện nay Kinect còn có thể được sử dụng nghiên cứu xử lý ảnh 3D, phát hiện hành vi cử chỉ, bám theo cơ thể người và tạo nên nhiều ứng dụng, các sản phẩm mang tính khoa học cao phục vụ cho đời sống thông tin hóa như ngày nay người ta dùng Kinect cho các ứng dụng trong giáo dục, y tế, quân sự và nhiều lĩnh vực khác...

#### 2.1.1. Những thành phần chính của Kinect

Thiết bị cảm biến Kinect là một thanh ngang kết nối với một trụ nhỏ bằng một trục cơ đứng. Thiết bị bao gồm hai camera, một cảm biến chiếu IR, microphone nhận diện giọng nói và một số bộ phận phụ được vận hành bởi một phần mềm độc quyền của Microsoft và công nghệ 3D camera của hãng

PrimeSense giúp thiết bị có thể bắt chuyển động 3D toàn cơ thể, nhận diện khuôn mặt và nhận diện giọng nói.



Hình 2.2: Cấu tạo cơ bản của Kinect

### 2.1.2. Thông số kỹ thuật cơ bản của Kinect

Thiết bị cảm biến ảnh màu của Kinect có thể ghi nhận ảnh màu RGB (8bit) với tần số 30Hz, kích thước khung hình 640 x 480 điểm ảnh.

Thiết bị ghi nhận độ sâu cũng có kích thước khung hình là 640 x 480 điểm, mỗi điểm độ sâu có 11 bit, do đó có thể phân biệt được 2048 độ sâu khác nhau trong cùng 1 ảnh độ sâu.

Góc mở theo phương ngang là  $57^{\circ}$ , theo phương dọc là  $43^{\circ}$ , với khoảng cách ngắn nhất có thể đạt tỉ lệ 1.3mm tương đương 1 điểm ảnh.

Hỗ trợ cổng USB nên hiện nay nó có thể kết nối với nhiều thiết bị, trong đó có cả máy vi tính [2].

### 2.1.3. Tính toán độ sâu trong Kinect

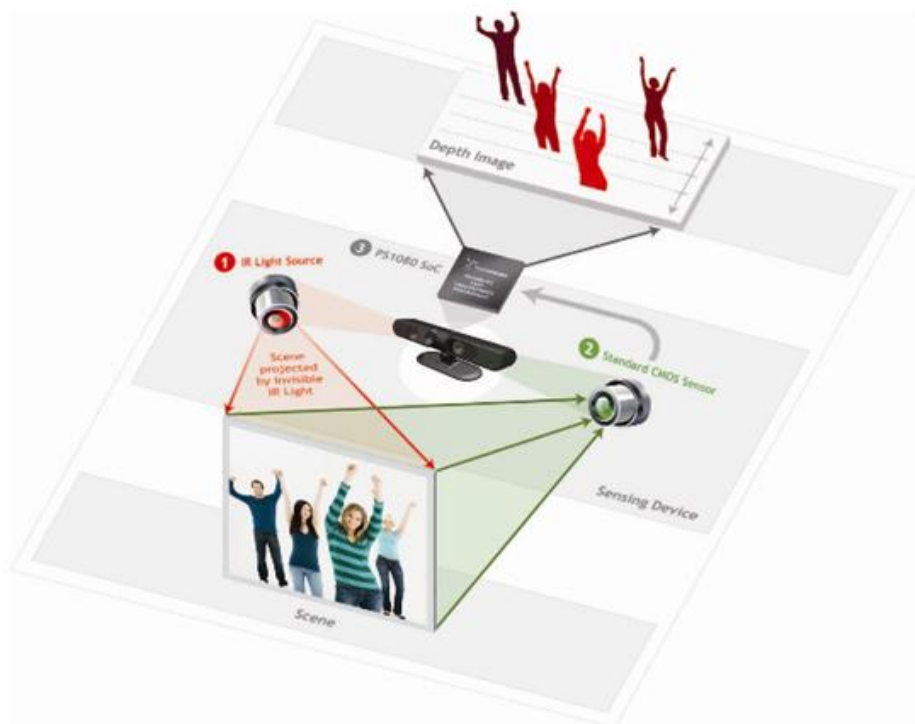
Để thu nhận độ sâu từng điểm ảnh là nhờ vào sự phối hợp giữa cặp cảm biến IR Camera và IR Projector dựa trên công nghệ Light Coding.

Công nghệ này hoạt động dựa trên việc phát ra chùm tia hồng ngoại, đặc trưng riêng của từng tia là không thấy được dưới mắt thường. Công nghệ này khác với công nghệ Time Of Flight (TOF) thường dùng Stereo Camera định nghĩa khoảng cách bằng ước lượng thời gian di chuyển của tia sáng đi và về trong không gian.



Hình 2.3: Bên trong Kinect gồm RGB, IR Camera và IR Projector

Kinect dùng một bộ cảm biến CMOS chuẩn, để ghi nhận lại các tia hồng ngoại bị phản xạ lại khi tiếp xúc với môi trường, dựa vào các đặc trưng mà xác định cụ thể vị trí tia hồng ngoại trong chùm tia và độ sâu của tia đo được. Tập hợp các chùm sáng này được IR Camera chụp lại được, thông qua giải thuật đặc biệt được tích hợp trong chip PS1080 SoC của PrimeSense [10] tính toán song song để xác định độ sâu của toàn bộ chùm tia phản xạ và xuất ra độ sâu của tất cả điểm ảnh.



Hình 2.4: Ảnh được thu độ sâu bởi Kinect

Phương pháp này được cho là công nghệ đáp ứng chính xác hơn với môi trường trong nhà hoặc nơi có ánh sáng yếu, đặc biệt có một ưu điểm là Kinect có thể nhìn thấy được vào thời điểm chiều tối. Ở môi trường ngoài trời nơi có ánh sáng mạnh, thì kết quả thu về bản đồ độ sâu không chính xác.

## 2.2. Các thư viện hỗ trợ hệ thống dẫn đường dành cho người mù

Để lấy các thông tin cần thiết từ Kinect, ta khó có thể tương tác trực tiếp với driver của thiết bị, việc này đòi hỏi kỹ thuật cao, khả năng lập trình tốt và am hiểu cấu tạo thiết bị. Hướng giải quyết là dùng một thư viện được cộng đồng mã nguồn mở hỗ trợ, để lấy dữ liệu một cách dễ dàng và được chuẩn hóa.

Thư viện hỗ trợ Kinect: Ngay khi mới ra đời, Kinect đã được quan tâm bởi rất nhiều nhà phát triển phần mềm, không chỉ trên mảng phát triển game cho Xbox mà còn trên mảng xử lý ảnh ứng dụng trong y học, robot, mapping... Do đó mà nhiều thư viện được viết cho Kinect ra đời. Cho đến thời điểm hiện tại, các thư viện đáng chú ý là Libfreenect, Code Laboratories Kinect, OpenNI và Kinect SDK.

**Thư viện Libfreenect:** Là thư viện được phát triển bởi OpenKinect, do một cộng đồng những người quan tâm đến phần cứng kinect viết ra và chia sẻ. Cộng đồng OpenKinect làm việc hoàn toàn tự nguyện và không vì mục đích lợi nhuận, họ phát triển Libfreenect thành một mã nguồn mở cho các hệ điều hành khác nhau Windows, Linux và OS X. Hiện tại, Libfreenect được đóng gói cho việc sử dụng trên Python, C, C++, C#, Java JNI, Java JNA, Javascript.

**Thư viện Code Laboratories Kinect:** Code Laboratories (CL) là một công ty phần mềm chuyên hỗ trợ các nhà phát triển, lập trình viên khai thác các tính năng của các thiết bị xử lý ảnh. Trong số đó Kinect không phải là ngoại lệ, CL cung cấp cho người sử dụng những tính năng cơ bản nhất của Kinect về camera, audio và motor.

**Thư viện OpenNI:** Thư viện OpenNI được xem là thư viện mạnh nhất trước sự có mặt của Kinect SDK 1.0, thư viện này hỗ trợ đa ngôn ngữ trên nhiều platform khác nhau, giúp cho các lập trình viên có thể viết các ứng dụng trên Kinect rất dễ dàng tương tác tự nhiên Natural Interaction (NI). Mục đích chính của OpenNI là xây dựng các hàm API chuẩn, cho phép thư viện có khả năng kết hợp với các middleware nhằm làm tăng sức mạnh cho Kinect.

**Thư viện Kinect SDK:** SDK cung cấp cho nhà phát triển những bộ cảm biến có chiều sâu, camera cảm biến màu sắc và xác định nguồn âm thanh theo chùm tia. Các nhà phát triển cũng có thể theo dõi ảnh ở dạng khung sườn của một hay hai người đang di chuyển trong tầm ngắm Kinect, từ đó các nhà phát triển có thể tạo ra những ứng dụng hướng theo cử chỉ (gesture). SDK gồm tính năng âm thanh như giảm tiếng ồn và hạn chế tiếng vang. Các nhà phát triển có thể truy cập vào công nghệ hình thành tia để nhận diện ra nguồn âm thanh hay có thể truy cập vào giao diện trình ứng dụng (API) để nhận diện giọng nói trong hệ thống Windows. Các nhà phát triển có thể viết ứng dụng bằng ngôn ngữ C#, C++ và bất cứ ngôn ngữ .Net nào.

Tới thời điểm hiện tại, hai thư viện OpenNI và Kinect SDK là sự lựa chọn sáng suốt nhất cho việc lập trình trên Kinect bởi tính năng hỗ trợ mạnh mẽ của hai thư viện này.

OpenNI cho phép thu về bản đồ độ sâu trong giới hạn 0.5 mét trở về trước, nên việc quan sát đối tượng vật cản phía trước sẽ thuận lợi hơn so với giới hạn thu của Kinect SDK là 0.8 mét mặc dù chất lượng bản đồ độ sâu là như nhau.

Còn một đặc điểm nữa là OpenNI ra đời trước thư viện SDK beta nên gần như được tích hợp thêm các thư viện xử lý ảnh của Point Cloud Library hỗ trợ cho việc xử lý ảnh đầu vào, xác định được các đối tượng vật cản và mặt phẳng trong không gian 3D. Trong khi thư viện SDK beta vừa mới ra mắt nên cũng chưa được hỗ trợ gì nhiều trong xử lý ảnh. Đó cũng là lý do sinh viên lựa chọn thư viện OpenNI kết hợp với Point Cloud Library trong khi thực hiện đề tài này.

### 2.2.1. Thư viện OpenNI

OpenNI (Open Natural Interaction): Là thư viện hỗ trợ đa ngôn ngữ. Mục đích chính của OpenNI là xây dựng các hàm API chuẩn, cho phép thư viện có khả năng kết hợp với các middleware làm tăng sức mạnh cho Kinect. Cũng cho phép các nhà phát triển viết các thuật toán trên định dạng dữ liệu thô, bất kể là thiết bị cảm biến đã sản xuất ra chúng [13]

Chức năng nổi trội mà OpenNI hỗ trợ lập trình viên thao tác với Kinect

**Alternative View:** Do cảm biến ghi nhận thông tin ảnh và thông tin IR ở 2 vị trí khác nhau trên Kinect, nên khi lấy dữ liệu ảnh và độ sâu sẽ không

khớp với nhau về góc nhìn, OpenNI dựa trên vị trí cố định giữa Projector và Webcam trên Kinect để ánh xạ dữ liệu ảnh, độ sâu IR vào cùng một hệ trục tọa độ.

**Cropping:** Hỗ trợ cắt bớt dữ liệu xuất ra thay vì phải lấy toàn bộ khung hình 640 x 480, thì chỉ lấy giới hạn kích thước để kết quả dữ liệu nhỏ gọn hơn, phù hợp cho những ứng dụng đòi hỏi chạy realtime.

**Frame Sync:** Khi lấy một thông tin ảnh hoặc độ sâu, thì không cần quan tâm đến vấn đề đồng bộ dữ liệu giữa thành phần này, nhưng khi cần lấy hai thông tin này cùng lúc thì đòi hỏi một cơ chế giúp đổi bộ quá trình ghi nhận dữ liệu ảnh RGB và độ sâu.

**Mirror:** Cơ chế ánh xạ từ trái sang phải và ngược lại để hình ảnh thu được không bị ngược chiều so với thực tế.

**Pose and User Detection:** Giúp xác định vị trí của người khi di chuyển vào góc nhìn của Kinect.

**Skeleton:** Hỗ trợ xuất ra thông tin khung xương của đối tượng (người).

**Error State:** Cung cấp tình trạng dữ liệu được lấy ra hoặc kiểm tra các node có tồn tại hay không.

**LockAware:** Chia sẻ Kinect giữa các phần mềm.

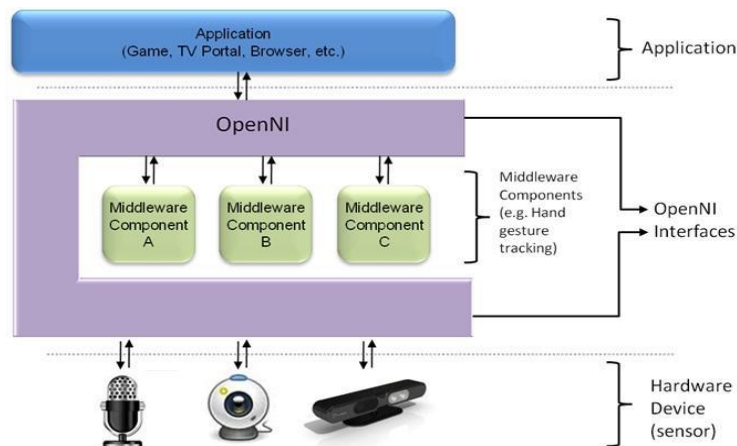
**Recording and Playing:** Ghi nhận thông tin trực tiếp từ Kinect vào File.ONI (định dạng riêng của OpenNI), hỗ trợ replay bằng cách đọc từ file.ONI mà không cần chỉnh sửa lại cấu hình các thành phần bên trong.

**Middleware** gồm (xem hình 2.5)

- **Full body analysis middleware:** là một thành phần phần mềm xử lý dữ liệu cảm giác và tạo ra thông tin liên quan đến cơ thể (hỗ trợ bám từng phần cơ thể người).
- **Hand point analysis middleware:** là một thành phần phần mềm xử lý dữ liệu cảm giác và tạo ra vị trí của một điểm tay.



- **Gesture detection middleware:** là một thành phần phần mềm nhận dạng cử chỉ được xác định trước (ví dụ: một bàn tay vẫy chào và cảnh báo các ứng dụng).
- **Scene analyzer middleware:** là một thành phần phần mềm phân tích hình ảnh của khung cảnh để xuất ra các thông tin như:
  - Tách biệt giữa mặt trước và sau của khung cảnh.
  - Tọa độ của mặt phẳng sàn.
  - Việc xác định cá nhân của các nhân vật trong cảnh.



Hình 2.5: Trình bày tổng thể mô hình 3 lớp[8]

### 2.2.2. Thư viện Point Cloud

Là bộ thư viện gồm các hàm, được định nghĩa nhằm mục đích xử lý những hình ảnh và tập hợp những điểm thể hiện vật thể [9]. Tức là dựa vào tập hợp những điểm đó để lập ra một mô hình của vật thể, từ đó có thể tính toán được các kích thước, khoảng cách.

Tập hợp điểm là những điểm mà bao quanh vật, từ những điểm đó có thể tái hiện được vật thể và có thể xác định được độ sâu, kích thước, khoảng cách đến vật.

Có thể nói PCL là sự kết hợp của nhiều module nhỏ, những module này thực chất cũng là các thư viện thực hiện các chức năng riêng lẻ trước khi được PCL đóng gói.

- `libpcl_features`: có nhiều tính năng 3D.
- `libpcl_surface`: kỹ thuật xây dựng lại bề mặt.
- `libpcl_filters`: bộ lọc dữ liệu tập hợp điểm.

- libpcl\_io: hoạt động I/O.
- libpcl\_segmentation: hoạt động phân chia một tập hợp điểm thành các nhóm riêng biệt.

Các thư viện cơ bản trong PCL[9]

- Eigen: là thư viện hỗ trợ cho phép toán tuyến tính, được dùng trong hầu hết các tính toán toán học của PCL.
- Flann: hỗ trợ cho việc tìm kiếm nhanh các điểm lân cận trong không gian 3D.
- Boost: giúp cho việc chia sẻ con trỏ trên tất cả các module và thuật toán trong PCL để tránh việc sao chép trùng lặp dữ liệu đã được lấy về trong hệ thống.
- VTK: hỗ trợ cho nhiều platform trong việc thu về dữ liệu 3D, hỗ trợ việc hiển thị và ước lượng thể tích vật thể.

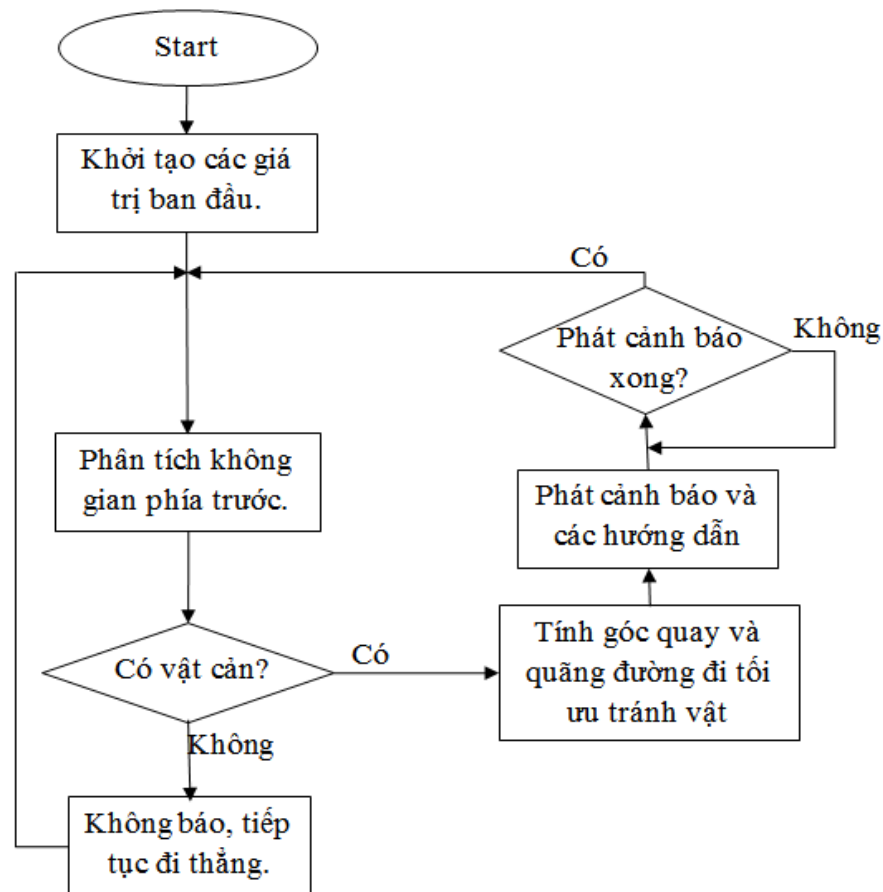
## **2.3. Phân tích và thiết kế hệ thống hỗ trợ dò đường bằng kinect**

### **2.3.1. Các tiêu chí để xây dựng hệ thống**

- Có thể nhìn bao quát được khoảng không gian môi trường phía trước.
- Tránh được các vật cản trong nhà.
- Phát âm thanh cảnh báo và hỗ trợ gợi ý hướng đi an toàn
- Tiện lợi trong việc sử dụng đi lại, không gây vướng víu cho người sử dụng.

### **2.3.2. Phân tích**

- Khi người dùng bắt đầu đi, đồng thời kinect sẽ khởi tạo các giá trị ban đầu thu về bản đồ độ sâu và phân tích các không gian phía trước, nếu không có vật cản phía trước thì tiếp tục đi thẳng, ngược lại nếu gặp vật cản thì chương trình sẽ xử lý tính góc quay và quãng đường đi tối ưu để tránh vật.
- Sau khi tính toán xong chương trình sẽ phát các gợi ý cảnh báo bằng âm thanh với các hướng dẫn về hướng đi cụ thể để tránh được các chướng ngại vật phía trước.
- Sau khi tránh vật thành công kinect sẽ quay lại khởi tạo lại giá trị mới.



Hình 2.6: Sơ đồ tránh vật cản

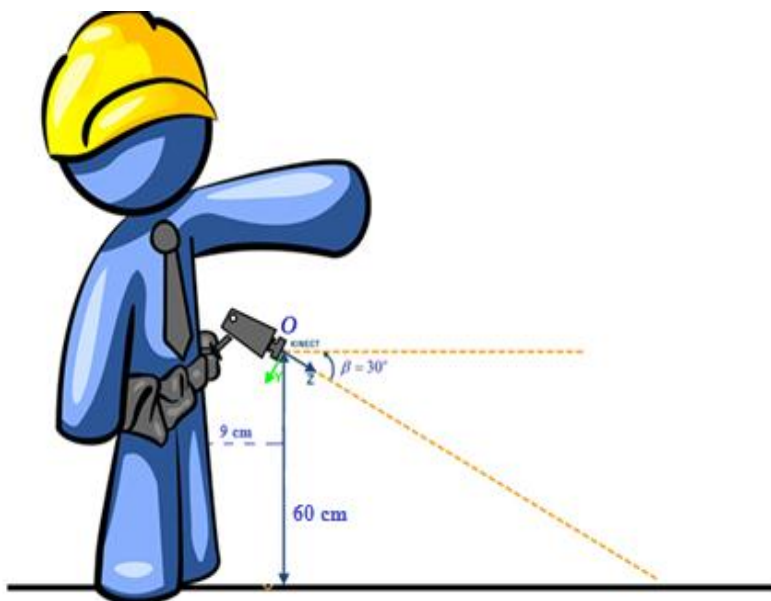
### 2.3.3. Thiết kế

#### Chuyển hệ trục tọa độ người và kinect

Thông tin môi trường trong quá trình di chuyển của người mù được thiết bị Kinect đưa về xử lý ở dạng 3D. Nhờ sự hỗ trợ mạnh mẽ của thư viện Point Cloud, cho ta những thông tin chính xác về tọa độ các điểm hay vật thể trong môi trường phía trước.

Một nhược điểm lớn của thiết bị Kinect đó là vùng mù, nó nằm trong khoảng 0 - 0.5 mét, là một hạn chế lớn đối với chức năng tránh vật cản của người mù. Chính vì lý do đó nên ta đặt Kinect ở vị trí thắt lưng và hướng xuống với một góc nghiêng nhất định, đồng thời thắt lưng là nơi ít bị ảnh hưởng đến việc thu hình ảnh như khi đặt trên đầu, vị trí thắt lưng sẽ đảm bảo cho Kinect luôn hướng về phía trước và không gây áp lực, mệt mỏi như khi đặt trên đầu.

Xem hình 2.7 mục đích của việc dời hệ trục tọa độ Kinect OXZ về trùng với tọa độ của người mù OXY là để người mù nhìn được một cách tổng quát môi trường phía trước.



Hình 2.7: Tính tọa độ Kinect trên người mù

Ta lần lượt thực hiện các công việc sau đối với hệ trục Kinect:

- Xoay mặt phẳng OYZ quanh trục X ngược chiều kim đồng hồ một góc  $30^0$ .

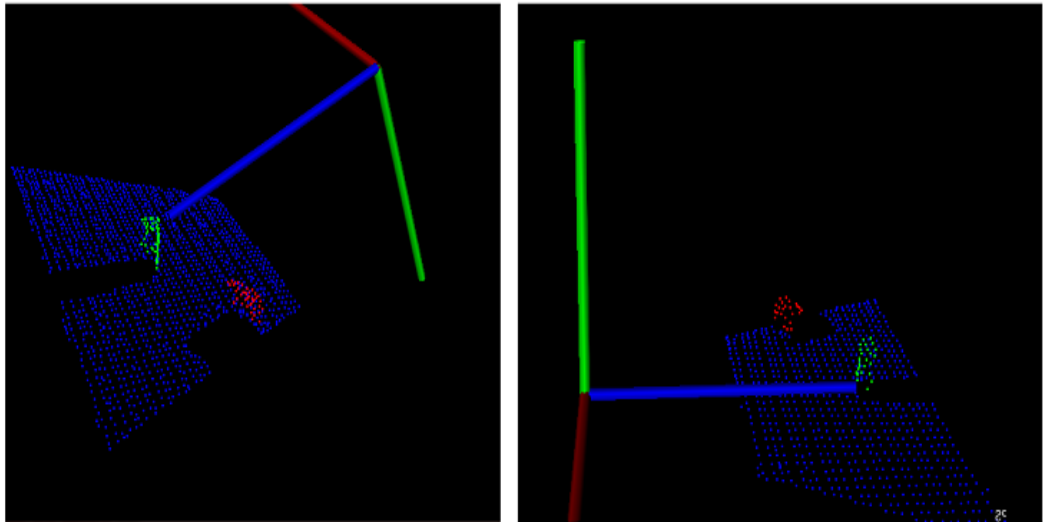
$$\begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} = [X \ Y \ Z] \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\beta & \sin\beta \\ 0 & -\sin\beta & \cos\beta \end{bmatrix} = \begin{bmatrix} X \\ Y\cos\beta - Z\sin\beta \\ Y\sin\beta + Z\cos\beta \end{bmatrix}$$

- Tịnh tiến trục Y xuống một đoạn 80cm và tịnh tiến trục Z về sau một đoạn 9cm.

$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \end{bmatrix} = \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} + \begin{bmatrix} 0 \\ -60 \\ 9 \end{bmatrix} = \begin{bmatrix} X \\ Y\cos\beta - Z\sin\beta - 60 \\ Y\sin\beta + Z\cos\beta + 9 \end{bmatrix}$$

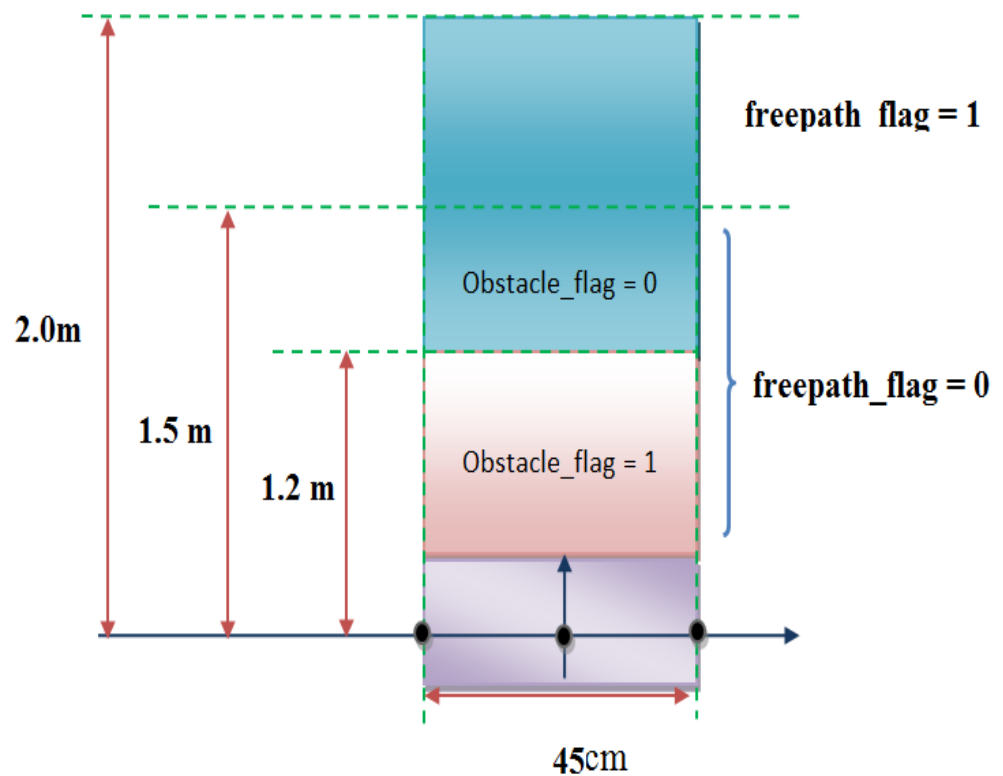
- Đảo chiều Y.

$$\begin{bmatrix} X_3 \\ Y_3 \\ Z_3 \end{bmatrix} = \begin{bmatrix} X_2 \\ -Y_2 \\ Z_2 \end{bmatrix} = \begin{bmatrix} X \\ -Y\cos\beta + Z\sin\beta + 60 \\ Y\sin\beta + Z\cos\beta + 9 \end{bmatrix}$$



Hình 2.8: Hệ trục có các chiều X, Y, Z tương ứng với các màu đỏ, xanh lá, xanh dương.

### Phân ngưỡng khoảng cách xác định vật cản

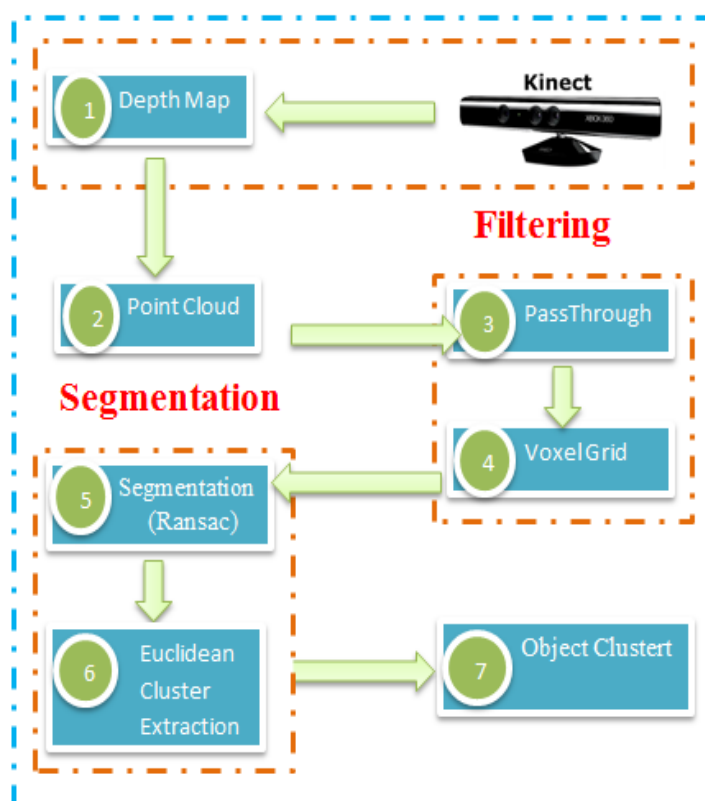


Hình 2.9: Phân ngưỡng khoảng cách từ người đến các chướng ngại vật phía trước

- Khoảng cách mà người có thể phát hiện thấy vật cản nằm trong khoảng từ 0.5m trở về trước 1.2m, khoảng cách này giúp cho người phát hiện và tránh vật một cách kịp thời nhất.

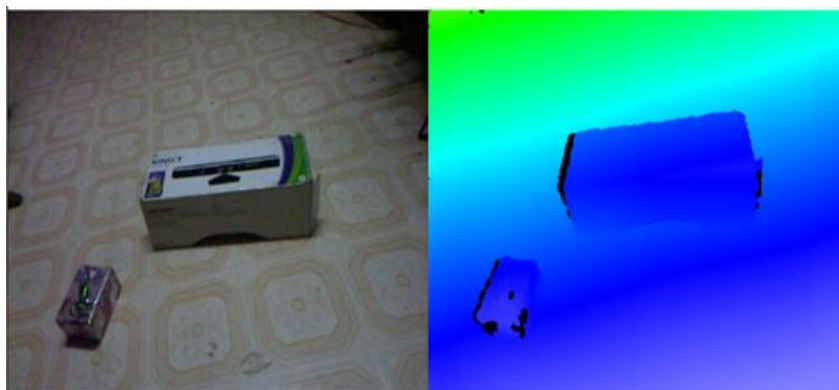
- Khoảng cách 1.2m – 1.5m là những vật cản nằm ở xa chưa gây cản trở việc di chuyển nên còn nằm trong phạm vi an toàn.
- Cuối cùng là khoảng cách từ 1.5 – 2.0m là khoảng cách an toàn, không có vật cản phía trước nên người mù có thể đi tự do.
- Việc phân ngưỡng các khoảng cách như thế sẽ giúp người mù di chuyển an toàn, và định hướng tốt hơn khi gặp các chướng ngại vật.

#### 2.4. Xây dựng chương trình điều khiển thiết bị dò đường bằng kinect



Hình 2.10: Sơ đồ xử lý phát hiện và tách vật

##### Bước 1: Lấy bản đồ độ sâu bằng Depth Map

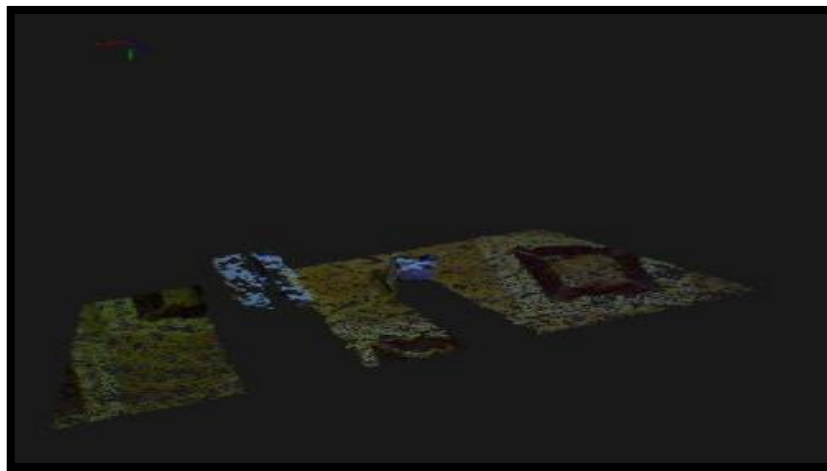


Hình 2.11: Ảnh RGB và bản đồ độ sâu

Depth map hay còn gọi là bản đồ độ sâu chứa thông tin vị trí của vật trong không gian phía trước Kinect. Độ phân giải càng nhỏ thì tốc độ xử lý khi qua thư viện Point Cloud càng được tăng lên (xem hình 2.11)

## **Bước 2: Thuật toán Point Cloud**

Sử dụng thuật toán Point Cloud tập hợp các đỉnh trong một hệ thống tọa độ ba chiều. Những đỉnh đó được định nghĩa bởi tọa độ X, Y và Z, và thường được coi là những đại diện bề mặt bên ngoài của một đối tượng (xem hình 2.12).



Hình 2.12: Point Cloud

Point Cloud  $P$  là một tập hợp các phần tử  $p_i$ , mỗi phần tử này sẽ chứa các giá trị biểu diễn trong không gian  $nD$ .

## **Bước 3: Thuật toán Pass Through**



Hình 2.13: Pass Through

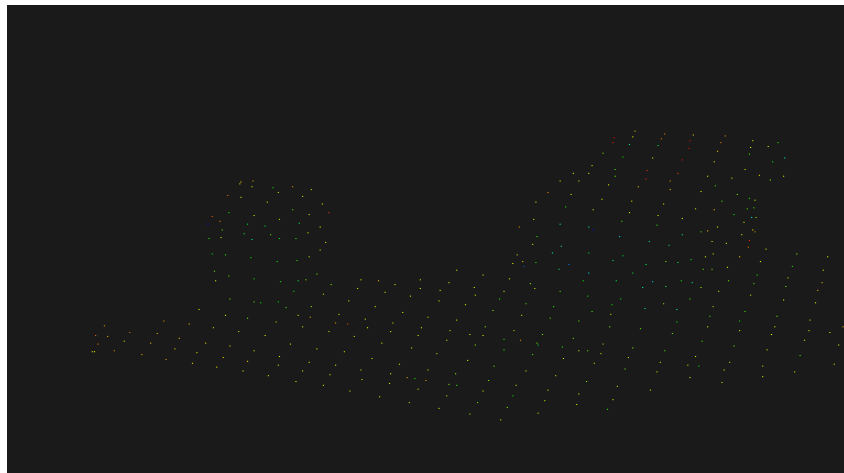
Thuật toán có tác dụng lọc qua tập hợp những điểm hạn chế không cần thiết, nhằm đáp ứng cho người dùng các giá trị nên được thiết lập (xem hình 2.13).

Giá trị khoảng cách theo chiều z mà Kinect có thể thấy là 0.5 - 5.0 mét, việc giới hạn lại sẽ làm PCL xử lý nhanh hơn vì thế ta chỉ cần cho Kinect nhìn trong khoảng 0.5-2.0 mét là đủ cho người mù tránh được vật cản, nên ta dùng PassThrough để cắt bỏ những điểm nằm ngoài khoảng cách này.

Hàm hỗ trợ:

```
setFilterFieldName ("z") ;  
setFilterLimits (0.5,2.0) ;
```

#### **Bước 4: Thuật toán Voxel Grid**



Hình 2.14: Voxel Grid

Voxel Grid có chức năng làm giảm mật độ số điểm xuống, là một cấu trúc đơn giản để giữ lại một điểm trọng tâm nhằm xác định được khoảng cách đến vật chính xác hơn (xem hình 2.14).

Ta chọn giá trị mật độ phù hợp mà vẫn đảm bảo quan sát rõ hình dạng vật thể, mật độ ở đây ta chọn là 3cm theo ba chiều x, y, z.

Hàm hỗ trợ:

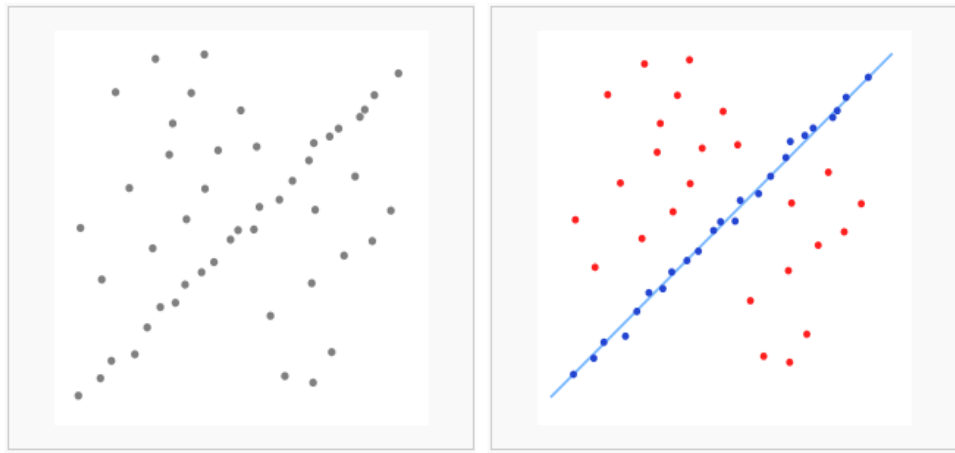
```
Grid_setLeafSize (0.03, 0.03, 0.03) ;
```



### Bước 5: Thuật toán Segmentation

Thuật toán thích hợp nhất để xử lý một tập hợp điểm có cấu trúc phẳng, sau đó tách tập điểm có tổng số điểm lớn nhất bằng thuật giải RANSAC. Đây là thuật toán lần đầu tiên công bố vào năm 1981 bởi Fischler và Bolles [16]

Thuật toán RANSAC được đánh giá là một thuật toán mạnh mẽ, do thay vì dùng số lượng điểm nhiều nhất có thể thì RANSAC lại chọn số điểm ít nhất mà vẫn đạt hiệu quả tương đương.



Hình 2.15: Minh họa cho RANSAC cho việc tìm đường thẳng trong mặt phẳng[16]

Hình 2.15 cho ta thấy ứng dụng của thuật toán RANSAC là tìm mô hình đường thẳng có dạng  $ax+by+c=0$  trong một tập hợp các điểm có chưa nhiều.

Giải thuật tìm đường thẳng bằng thuật toán RANSAC được mô tả như sau:

Đầu vào:

Data: tập hợp các điểm

K: số lần lặp

T: ngưỡng sai số để xác định điểm nào đó có khớp với mô hình không

N: số lượng tối thiểu của dữ liệu cần thiết để phù hợp với mô hình

Đầu ra:

Best\_model: mô hình tốt nhất

Best\_consensus\_set: tập hợp điểm khớp với Best\_model

Best\_model = null

Best\_consensus\_set = null

Best\_num\_point = 0

Lặp K lần:

Consensus\_set = tập hợp 2 điểm ngẫu nhiên thuộc data

Model = mô hình đường thẳng suy diễn từ 2 điểm trên

Với mỗi điểm thuộc data nhưng không thuộc consensus\_set

Ta xét distance = khoảng cách từ điểm đến đường thẳng

If distance < t (điểm thuộc mô hình sai số nhỏ hơn mức ngưỡng đặt trước)

Thêm điểm đó vào consensus\_set

Num\_point = số lượng phần tử trong consensus

If Num\_point > Best\_num\_points

Best\_model = Model

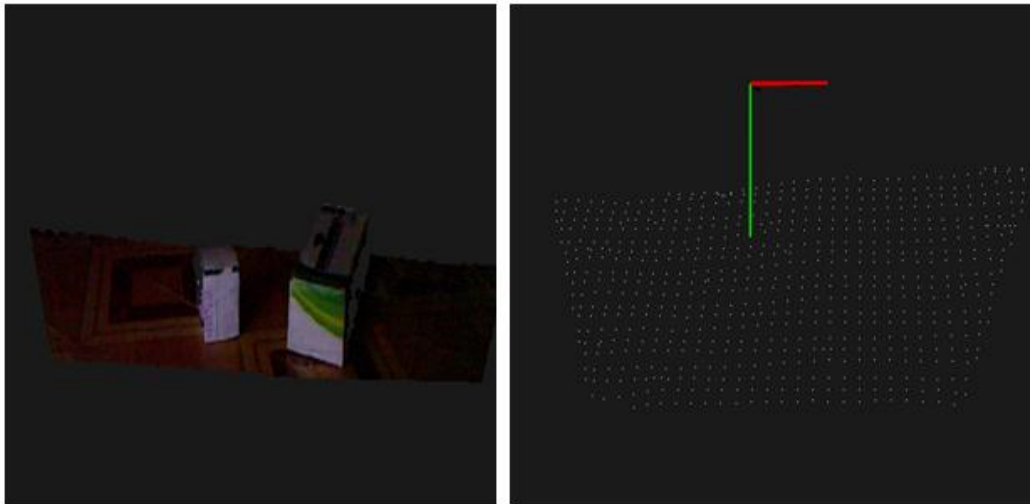
Best\_consensus\_set = Consensus\_set

Best\_num\_points = Num\_points

⇒ Trả về giá trị: Best\_model và Best\_consensus\_set

Với mô hình mặt phẳng  $\alpha$  có dạng  $ax+by+cz+d=0$  thì thuật toán RANSAC cũng có thể làm được điều tương tự.

Thay vì chọn hai điểm bất kì để tìm mô hình đường thẳng, thì ta sẽ chọn 3 điểm để tìm ra mô hình mặt phẳng. Quan sát hình 2.16 ta thấy các tập hợp điểm màu xanh sẽ là mặt phẳng nền nhà được tìm ra nhờ thuật toán RANSAC.

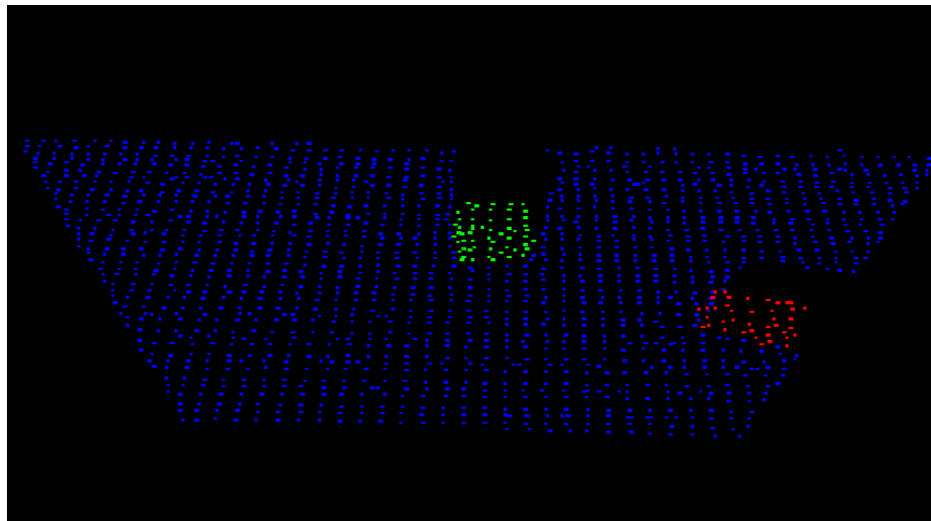


Hình 2.16: Trước và sau khi sử dụng thuật toán RANSAC

Hàm hỗ trợ:

```
seg_setMethodType(pcl::SAC_RANSAC);
seg_setMaxIterations(1000);
seg_setDistanceThreshold(0.035);
```

#### Bước 6: Thuật toán EuclideanClusterExtraction



Hình 2.17: Point Cloud sau khi thực hiện xong bước lọc và phân đoạn

Thuật toán làm công việc tách các tập hợp điểm có mặt trên nền nhà, tập hợp các điểm gần nhau sẽ được nhóm lại thành một tập hợp điểm hoặc một cụm, mỗi cụm thể hiện cho một vật thể.

Hình 2.17 có hai cụm : cụm màu đỏ là cụm gần Kinect và cụm màu xanh lá là cụm xa Kinect, mặt phẳng nền nhà là màu xanh dương.

Hàm hỗ trợ :

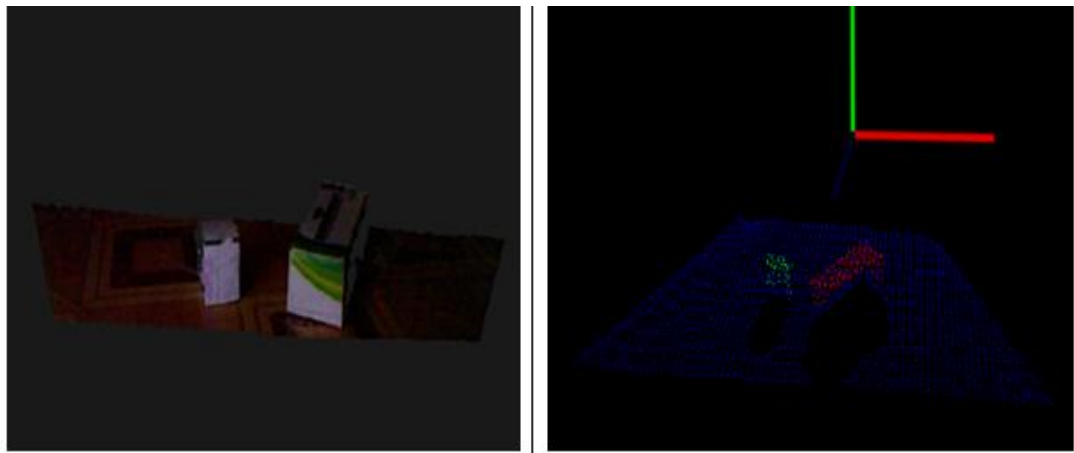
```
SetInputCloud(cloud_filtered);

SetIndices(indiers);

Filter(*cloud_cluster);
```

### Bước 7: Thuật toán Object Clusters

Bước này là bước làm nhiệm vụ phân tích các đặc tính của vật cản về kích thước cũng như vị trí của vật trong môi trường phía trước mà Kinect thấy được, đây là cũng những thông tin cần thiết cho việc xác định và tránh vật cản cho người mù (xem hình 2.18).



Hình 2.18: Object Clusters

Hàm hỗ trợ :

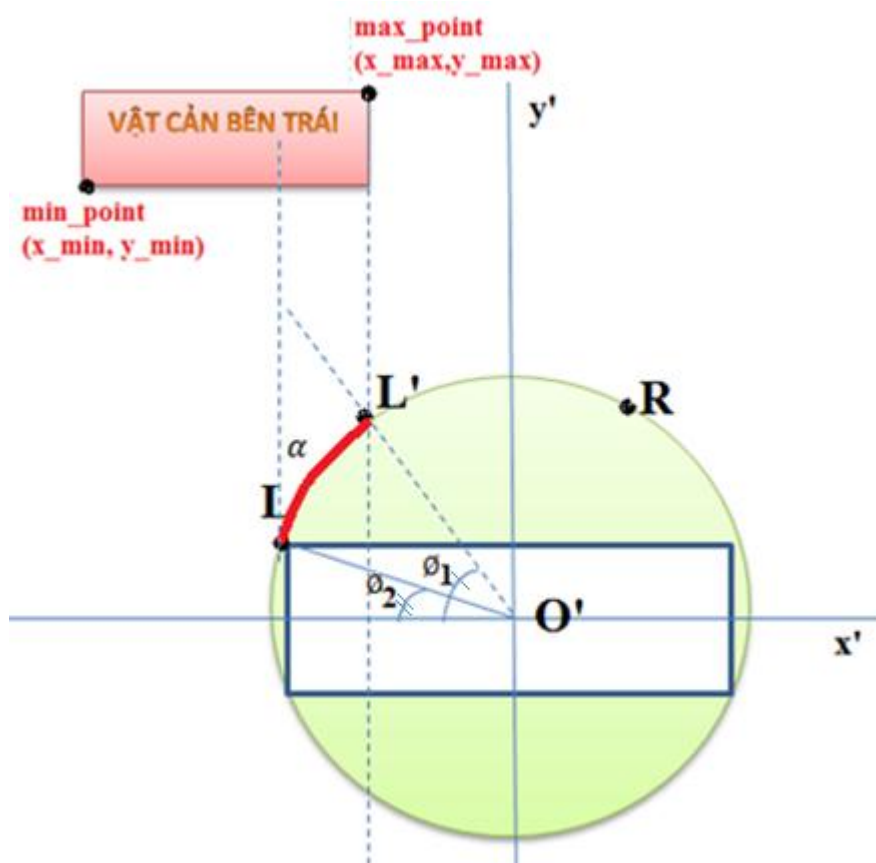
```
GetMinMax3D(&cloud_cluster, min_point, max_point);
```

## 2.5. Tính toán góc quay tối ưu để tránh vật

### 2.5.1. Vật cản nằm phía bên trái người mù

Quan sát trên hình 2.19 ta thấy rằng góc cần thiết để thoát khỏi vật cản là góc  $\alpha$  ( $\alpha$ ). Khi người ở vị trí cách vật một đoạn.

Góc  $\alpha$  có thể tính được nếu ta biết được tọa độ  $\text{max\_point}$ , góc  $\theta$  và độ lớn  $O'L$ .



Hình 2.19: Vật cản nằm bên trái người mù, tính góc xoay sang phải.

Công thức tính góc quay sang trái

$$\alpha = \theta_1 - \theta_2$$

$$\alpha = \arccos\left(\frac{-\text{max\_point}.x}{O'L'}\right) - \arccos\left(\frac{\frac{\text{bodywidth}}{2}}{O'L}\right)$$

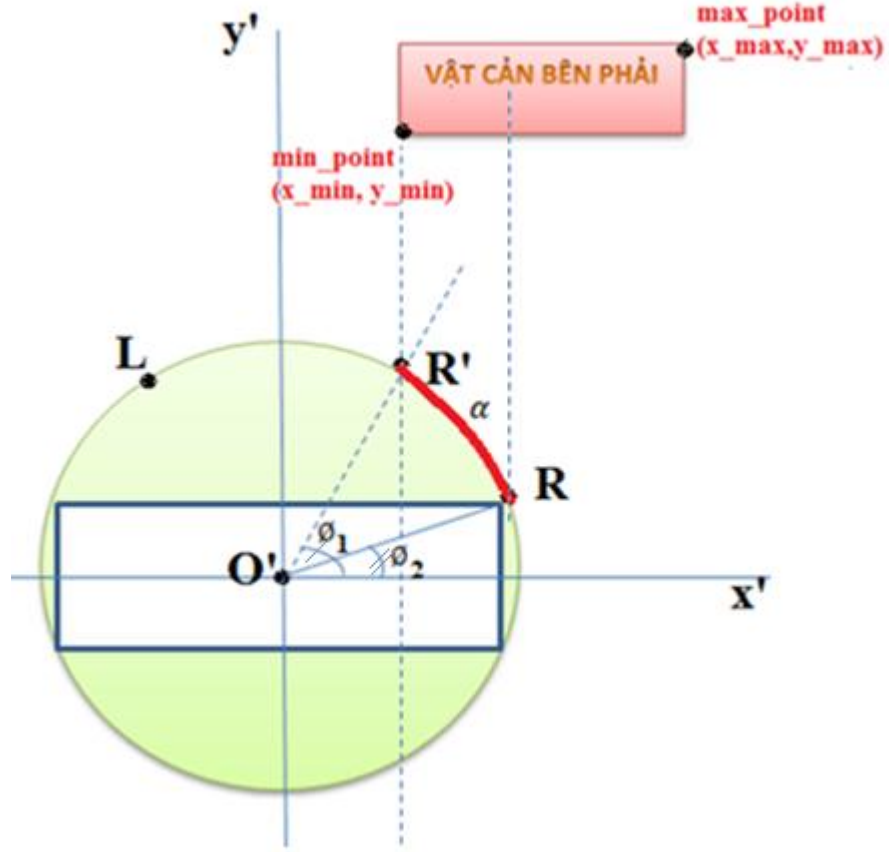
### 2.5.2. Vật cản nằm phía bên phải người mù

Tương tự nếu vật cản nằm bên phải người mù thì góc anpha có thể tính được nếu ta biết được tọa độ min\_point, góc  $\theta$  và độ lớn  $O'R$  (xem hình 2.20).

Công thức tính góc quay sang phải

$$\alpha = \theta_1 - \theta_2$$

$$\alpha = \arccos\left(\frac{\text{min\_point}.x}{O'R'}\right) - \arccos\left(\frac{\frac{\text{bodywidth}}{2}}{O'R}\right)$$



Hình 2.20: Vật cản nằm bên phải người mù, tính góc xoay sang trái

### 2.5.3. Vật cản nằm phía chính giữa người mù

Trường hợp vật cản nằm ở giữa đường đi của người mù, ta sẽ tính góc né vật sao cho anpha nhỏ nhất. Lúc đó nếu độ lớn vật cản thiên nhiều hơn về phía trái nhiều hơn thì tính toán cho người mù quay về bên phải và ngược lại (xem hình 2.21).

Vật cản nằm bên trái nhiều hơn nên tính góc quay về bên phải

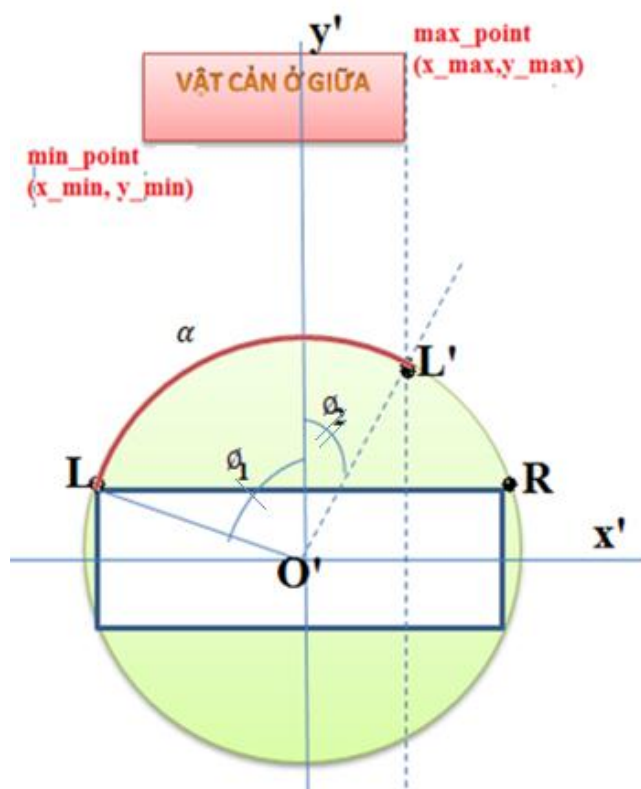
$$\alpha = \theta_1 + \theta_2$$

$$\alpha = \arcsin\left(\frac{-\text{max\_point}.x}{O'L}\right) + \arcsin\left(\frac{\frac{\text{bodywidth}}{2}}{O'L'}\right)$$

Vật cản nằm bên phải nhiều hơn nên tính góc quay về bên trái

$$\alpha = \theta_1 + \theta_2$$

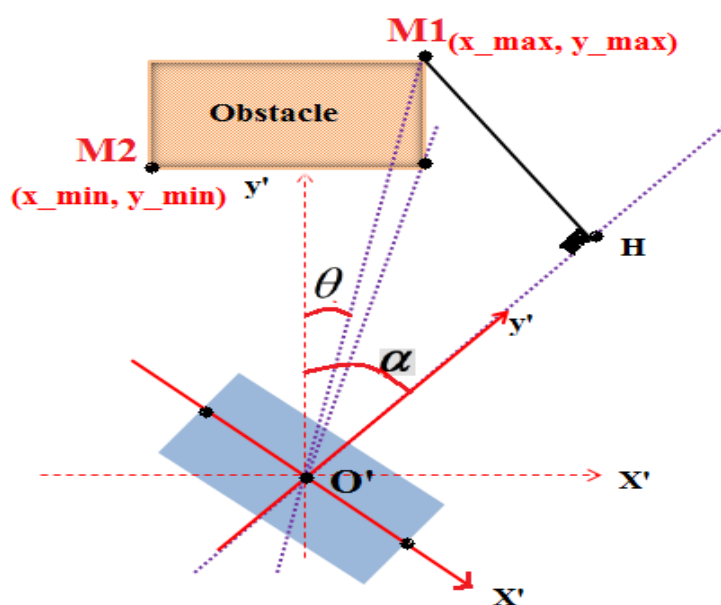
$$\alpha = \arcsin\left(\frac{\text{min\_point}.x}{O'L}\right) + \arcsin\left(\frac{\frac{\text{bodywidth}}{2}}{O'L'}\right)$$



Hình 2.21: Vật cản nằm ở giữa đường đi của người mù

#### 2.5.4. Tính toán đi thêm một đoạn an toàn sau khi tránh được vật cản

Sau khi đã tính toán góc anpha tránh được vật cản rồi, ta sẽ tính tiếp một đoạn đủ để người mù có thể di chuyển tiếp một đoạn nữa để đảm bảo thoát hoàn toàn khỏi chương ngại vật đó (Xem hình 2.22).



Hình 2.22: Tính toán đi một khoảng an toàn

Công thức tính khoảng cách đi an toàn

$$O'H = O'M_1 * \cos(\alpha - \theta)$$

$$O'H = 100 * \sqrt{(x_{max})^2 + (y_{max})^2} * \cos(\alpha - \tan\left(\left|\frac{x_{max}}{y_{max}}\right|\right))$$

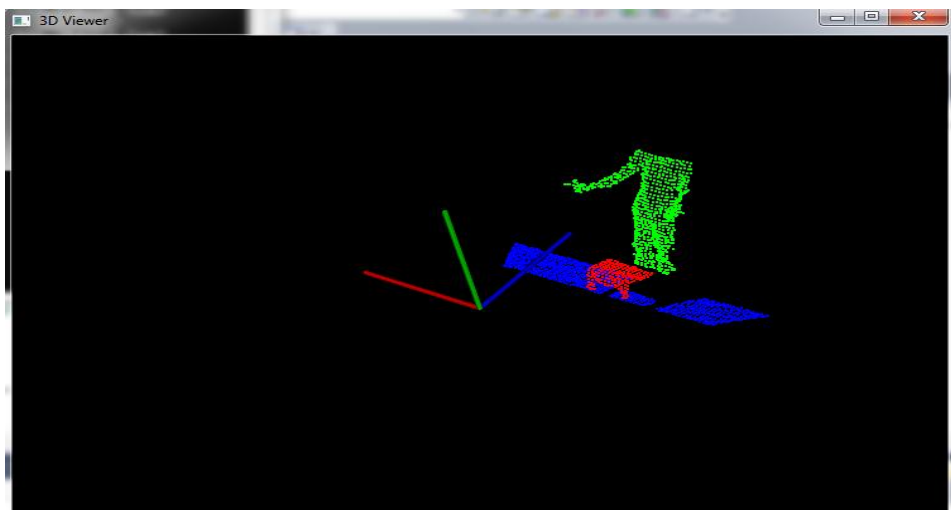
## 2.6. Kết quả thử nghiệm

### 2.6.1. Kết quả thử nghiệm trong nhà

#### Ban đêm

Về cơ bản sóng hồng ngoại cũng như sóng ánh sáng, nhưng sóng hồng ngoại lại có những bước sóng nằm ngoài dải nhìn thấy của mắt người vì thế chúng ta không thể cảm nhận các tia hồng ngoại chiếu đến vật thể. Nhưng các loại camera kỹ thuật số ngày nay đang rất phổ biến trên các thiết bị như điện thoại, hay máy ảnh kỹ thuật số lại có khả năng nhìn thấy các bước sóng hồng ngoại này. Người ta thường sử dụng hồng ngoại cho các loại thấu kính nhìn đêm, để phục vụ cho các lĩnh vực quân sự, an ninh, theo dõi, giám sát...

Trên camera Kinect cũng vậy, nhờ vào tia hồng ngoại này mà ta có thể nhìn mọi thứ trong bóng tối, khi IR projector phát chùm ánh sáng hồng ngoại vào vật thể, sẽ được IR camera thu về và camera này có thể nhìn thấy được các vật cản trong bóng đêm. Đây cũng là một lợi thế cho thiết bị cảm biến này, người mù có thể sử dụng khi trời tối mà camera vẫn có thể xử lý được môi trường vật cản phía trước họ, mà các thiết bị cảm biến khác không làm được.



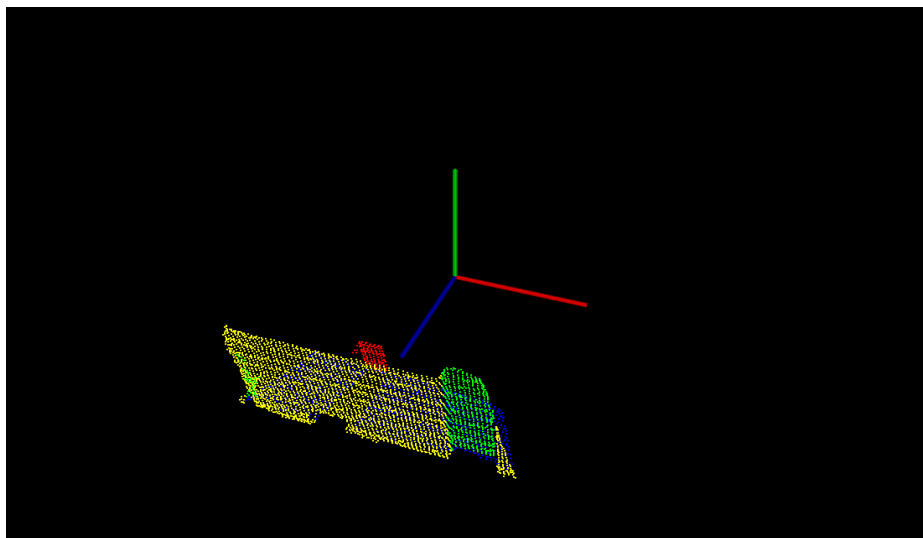
Hình 2.23: Hình ảnh được chụp vào ban đêm



### Ban ngày

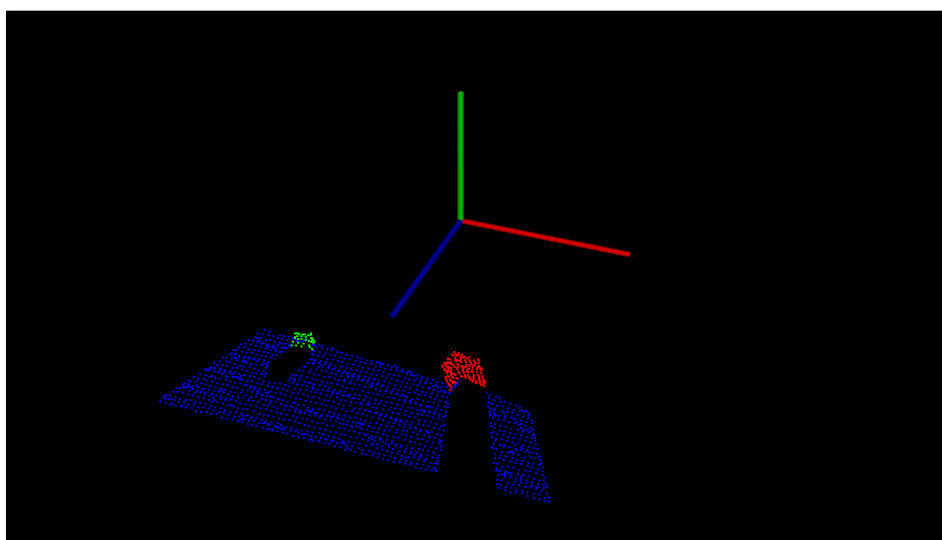
Camera Kinect sẽ ổn định hơn khi ở môi trường trong nhà, với ánh sáng vừa và ánh sáng yếu, dưới đây là các hình ảnh thử nghiệm.

Hình 2.24 cho ta thấy thông tin vị trí vật cản trước không gian 3 chiều như đã phát hiện được vật cản nguy hiểm là màu đỏ, vật cản an toàn là màu xanh lá cây, phân biệt được mặt phẳng của tường màu vàng và sàn màu xanh dương.



Hình 2.24: Xác định vật cản, tường và sàn trong môi trường ánh sáng trong nhà.

Khoảng cách vừa đủ cho người mù đi qua được, ở đây đề tài giới hạn khoảng cách có thể lọt qua được là bằng kích thước trung bình cơ thể người. Còn ngược lại sẽ không đi được (xem hình 2.25).



Hình 2.25: Khoảng cách vừa đủ cho người mù đi qua.

### 2.6.2. Kết quả thử nghiệm ngoài trời

Ánh sáng mặt trời có khoảng 50% năng lượng thuộc về tia hồng ngoại. Chính vì thế khi đưa thiết bị Kinect ra môi trường bên ngoài thì chùm tia hồng ngoại của Kinect sẽ bị bão hòa bởi một lượng tia hồng ngoại quá lớn của mặt trời, sẽ dẫn đến khó khăn cho việc xác định vật cản không được chính xác và ổn định như môi trường trong nhà.

### 2.7. Đánh giá chương trình

Nhóm tác giả đã nhận rất nhiều ý kiến đánh giá của các cô chú tại hội người mù, họ cho biết là rất xúc động và vui mừng khi biết được có một ứng dụng hữu ích đang được nghiên cứu giành cho những hoàn cảnh kém may mắn như mình. Và họ cũng lần lượt đi thử nghiệm và kết quả cho thấy khá tốt có thể tránh được vật cản an toàn.

#### Ưu điểm :

- Đề tài đã đi đúng mục tiêu mà nhóm đề ra là xác định và tránh được vật cản với sai số có thể chấp nhận được với môi trường trong nhà.
- Đặc biệt hơn là thiết bị cảm biến này có thể nhìn được vật thể trong bóng tối nhờ vào chùm sáng phát ra từ tia hồng ngoại, đó là một đặc điểm khá hay được tích hợp trên camera kinect mà chưa thiết bị cảm biến nào có thể làm được.
- Dựa vào lợi thế về góc mở của Kinect khá lớn, được đo thực nghiệm như sau góc mở kinect có thể thấy được theo chiều dọc là 43 độ xấp xỉ bằng 1.6 mét và chiều ngang 57 độ xấp xỉ 1.2 mét, nên có thể nhìn mọi vật xung quanh một cách tổng thể hơn, không chỉ nhìn thấy vật nằm sát mặt đất như các cảm biến siêu âm, laser hay phụ thuộc vào hình dạng kích thước màu sắc của vật cản như của các cảm biến một camera, hai camera... mà kinect còn có thể nhìn được bất kì các chướng ngại vật nào ở các vị trí khác nhau trong tầm nhìn của nó, đáp ứng độ tin cậy cao hơn so với nhiều loại cảm biến khác.
- Phát hiện được các chướng ngại vật phía trước ở các phạm vi khác nhau như vật cản ở gần hay còn gọi là vật cản nguy hiểm, vật cản ở xa nằm trong phạm vi an toàn, phân biệt được mặt phẳng tường và sàn.

- Phát ra âm thanh cảnh báo và gợi ý hướng đi, giúp người mù tránh được vật cản trong khoảng thời gian ngắn nhất.
- Khắc phục được phần nào những hạn chế về kỹ thuật, cũng như tốc độ xử lý của những đề tài trước.

### **Hạn chế:**

- Chương trình viết trên môi trường Windows với sự hỗ trợ từ thư viện xử lý ảnh Point Cloud còn gặp nhiều hạn chế, nhất là tốc độ xử lý.
- Có thể gắn thêm tai nghe Bluetooth hoặc làm động cơ rung trên dây đai thắt lưng, như vậy người dùng sẽ dễ dàng sử dụng hơn.

## **2.8. Hướng phát triển của đề tài**

### **2.8.1. Về chương trình**

Như đã trình bày ở phần trước, chương trình viết trên môi trường Windows nên còn nhiều hạn chế, nhất là hạn chế về tốc độ xử lý, điều này sẽ được khắc phục phần nào khi kết hợp thư viện Point Cloud trên môi trường Linux, và thiết kế thêm động cơ rung trên đai lưng để người sử dụng dễ dàng nhận tín hiệu đúng lúc.

Tăng cường các giải thuật xử lý hình ảnh, xử lý point cloud để không chỉ xác định được vật cản và đưa ra cảnh báo, mà còn có thể nhận diện vật thể phía trước, phân biệt được các vật dụng cần thiết và các vật cản thực sự để đưa ra những gợi ý thông minh hơn.

Hướng đến việc phát triển khả năng xác định đích dựa trên yêu cầu của người mù, kết hợp xây dựng hành trình để không còn là bị động tránh vật cản nữa mà chủ động đi theo hướng không có vật cản.

### **2.8.2. Về thiết bị**

Để đề tài có thể phát triển và đem lại lợi ích thực sự cho người mù thì việc cải tiến hệ thống thiết bị là rất quan trọng. Do trong thời gian nghiên cứu và thử nghiệm nên các hệ thống chưa được tối ưu hóa, thiết bị kinect cắm với nguồn điện, kết nối với chương trình thông qua cổng USB trên máy tính, nên còn khá cồng kềnh.

Vì Kinect cần nhiều điện năng để hoạt động nên cổng USB của Xbox-360 không thể đáp ứng mà phải qua một cổng chia để chia thành 2 kết nối riêng

là USB và kết nối nguồn, giúp cho thiết bị kết nối với Xbox-360 bằng cổng USB trong khi nguồn điện cần cho Kinect là 12VDC được lấy từ adapter. Phiên bản Xbox-360 mới sẽ không cần adapter vì nó có các AUX port đặc biệt để cung cấp cho cổng kết nối. Với kết nối USB ta hoàn toàn có thể cho Kinect giao tiếp với máy tính. Bằng cách thay adapter bằng nguồn pin 12V (xem hình 2.26).



Hình 2.26: Dây nguồn của thiết bị Kinect

Hướng phát triển tiếp theo sẽ thay thế nguồn điện của kinect chuyển sang sử dụng thành các loại pin sạc, sẽ gọn nhẹ và dễ dàng di chuyển hơn. (Xem hình 2.27)



Hình 2.27: Hình ảnh cho Pin Varta

Và chương trình được lập trình trên máy tính sẽ phát triển theo hệ thống nhỏ gọn hơn, có thể tích hợp trên các thiết bị nhỏ gọn hơn như máy tính bảng hoặc một số thiết bị cài đặt hệ điều hành của Windows. (Xem hình 2.29)

Tìm hiểu thông số kỹ thuật của Kinect và tìm ra một giải pháp tối ưu hóa cho mô hình sản phẩm được nhỏ gọn tiện lợi hơn cho người sử dụng.



Hình 2.28: Cảm biến Kinect cho Windows

Bộ cảm biến kinect chỉ sẽ hoạt động với các ứng dụng được phát triển Kinect cho windows. Và xây dựng các ứng dụng trên các ngôn ngữ lập trình C++, C# hoặc Visual Studio 2010 (Xem hình 2.28).

Yêu cầu cho phần cứng như sau :

- 32 bit (x86) hoặc 64 bit (x64) Processor
- Dual-core 2.66 GHz hoặc bộ xử lý nhanh hơn
- Ram 2GB trở lên.
- USB 2.0 bus
- Kinect chạy trên hệ điều hành Windows 7 và Windows 8



Hình 2.29: Ảnh minh họa hướng phát triển sản phẩm

Tương lai thiết bị cảm biến kinect còn có thể vươn xa hơn, chứ không đơn thuần là việc xác định vật cản và đưa ra thông tin cảnh báo nữa, ngoài việc xử lý tốc độ nhanh hơn, kinect còn có thể nhận dạng được tất cả các loại vật cản chứ không phụ thuộc vào hình dạng kích thước, màu sắc, vị trí của các cảm biến khác từ trước đến giờ chưa làm được.

Đề tài tạo nền tảng hướng đến việc xây dựng bản đồ dạng 3D để biết được thông tin vị trí tất cả các vật thể trong nhà, nhận biết được các đồ vật, hay có thể giúp người mù đọc chữ làm tất cả mọi thứ mà không cần đến sự trợ giúp của người khác nữa.

## **CHƯƠNG III. KẾT LUẬN**

### **3.1 So sánh kết quả nghiên cứu với mục tiêu đặt ra ban đầu**

Sản phẩm nghiên cứu là phần mềm “xác định vật cản và tìm đường đi cho người mù” đã thực hiện được các mục tiêu ban đầu đề ra là xác định được vật cản và phát ra âm thanh cảnh báo gợi ý hướng đi an toàn cho người mù.

Do đây mới chỉ là phần mềm được lập trình trên thiết bị, để điều khiển chương trình sử dụng cho người mù và tương lai nó sẽ được các nhà sản xuất tích hợp lên các thiết bị nhỏ gọn trên các mạch điện tử, giá thành sẽ phù hợp với túi tiền người sử dụng.

### **3.2 Kết luận**

Sau quá trình nghiên cứu và tìm hiểu về đề tài “Xác định vật cản và ứng dụng dò tìm đường đi cho người mù”, sinh viên đã có dịp tiếp xúc và chia sẻ về cuộc sống cũng như những khó khăn mà người mù đang gặp phải và đây cũng là nguồn động lực cho nhóm tập trung nghiên cứu.

Nhóm sinh viên đã tìm hiểu tổng quan về các loại cảm biến hỗ trợ cho người mù cũng như các phương pháp xác định vật cản, nhận ra mỗi phương pháp có các ưu khuyết điểm riêng và sử dụng tùy vào từng trường hợp mục đích thích hợp.

Ở đề tài này, nhóm chỉ tập trung nghiên cứu và khai thác những tính năng mạnh mẽ của thiết bị cảm biến Kinect cho khả năng xác định vật cản giành riêng cho người mù, là sự kết hợp giữa camera kinect và các thư viện hỗ trợ xử lý ảnh của PCL. Và kết quả đạt được là đã hoàn thành những mục tiêu mà đề tài đặt ra là xác định được các chướng ngại vật phía trước người mù, và xuất âm thanh cảnh báo gợi ý hướng đi an toàn cho họ.

Cảm biến Kinect thật sự vượt trội hơn hẳn các cảm biến thông thường khác trước đây. Ví dụ như cảm biến siêu âm, laser thì chỉ phát hiện vật cản nhanh nhưng còn dựa vào hình dạng kích thước vật, phương pháp một camera thì chỉ thu về ảnh 2D và xác định vật cản dựa trên màu sắc đường viền vật thể nên còn nhiều hạn chế. Còn phương pháp sử dụng 2 camera thì thu về ảnh 3D cho hiệu quả cao hơn nhưng khâu điều chỉnh phức tạp, chưa ổn định nên chưa được tối ưu. Vì thế mà nhóm tập trung vào phương pháp xử lý ảnh trong không gian 3D với cảm biến Kinect là sự kết hợp của tất cả các

công nghệ cảm biến trên, đưa ra các thông số chính xác nhất về vật cản phía trước người mù. Không những vậy, thiết bị cảm biến mới này thật sự là con lóc của công nghệ xử lý ảnh. Chỉ trong một thời gian ngắn, Kinect từ một thiết bị giải trí đã trở thành một vũ khí mạnh cho tất cả các ứng dụng mà các nhà nghiên cứu trong và ngoài nước quan tâm từ giáo dục, y tế, kinh doanh thương mại cho đến quốc phòng...

Đề tài mong muốn tạo nền tảng ban đầu để xây dựng những ứng dụng cho người mù, và tương lai sẽ hướng đến việc xây dựng bản đồ dạng 3D để biết được thông tin vị trí tất cả các vật thể trong nhà, nhận biết được các đồ vật, hay có thể giúp người mù đọc chữ, làm tất cả mọi việc như người bình thường, mà không cần đến sự trợ giúp của người khác nữa, đó cũng là những vấn đề mà xã hội rất quan tâm.



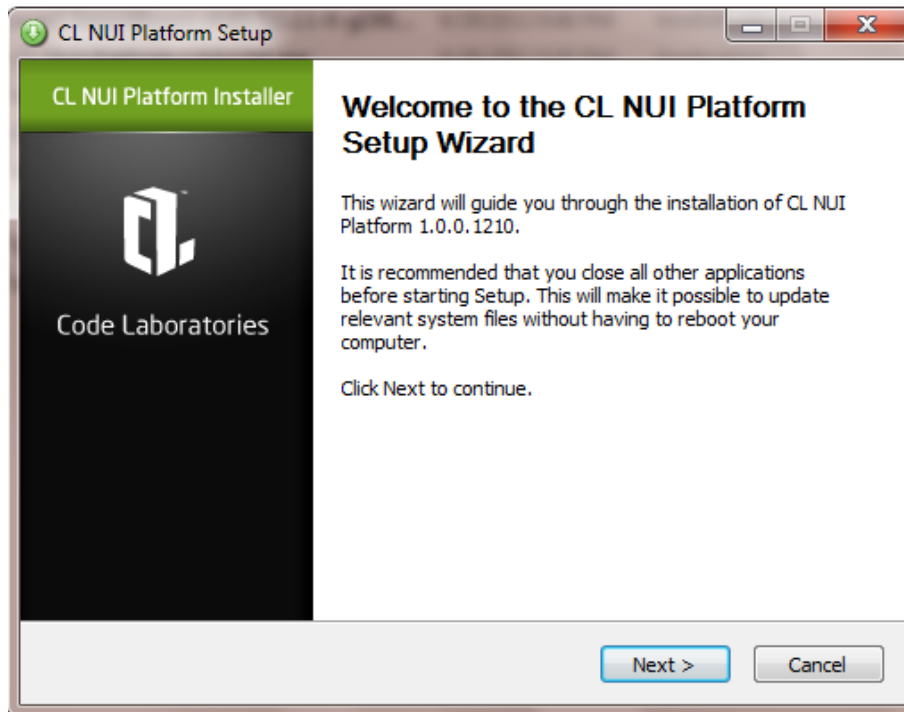
## PHỤ LỤC

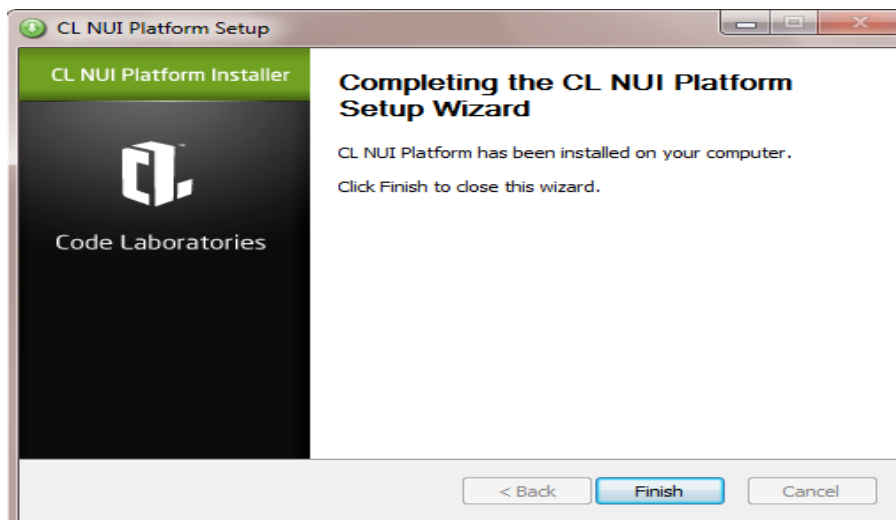
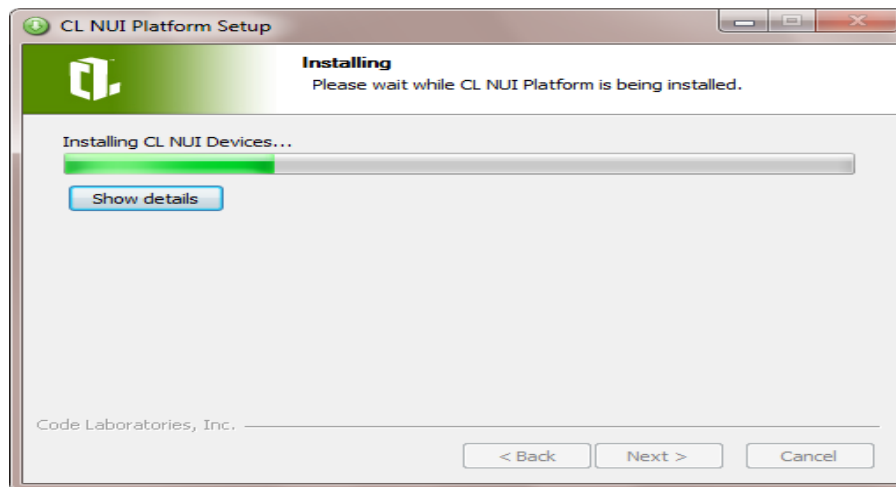
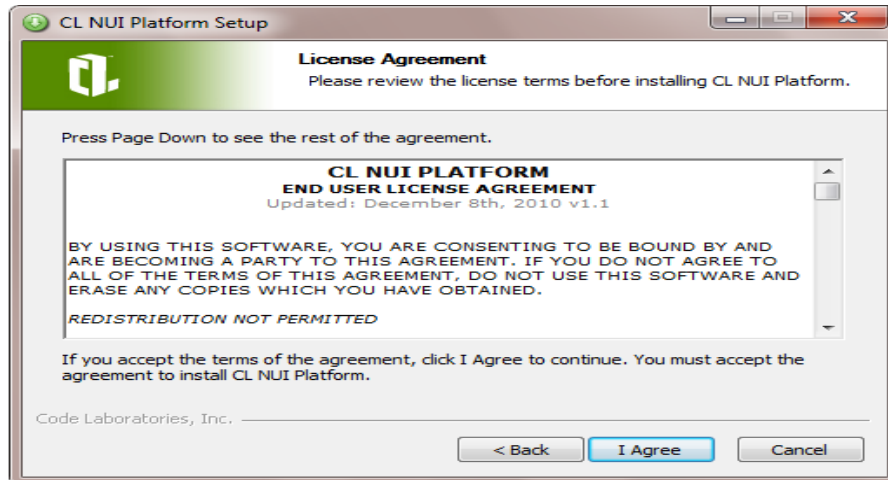
**Kết hợp thư viện OpenNI và Code Laboratories Kinect (CL) để sử dụng chức năng điều khiển động cơ Kinect.**

Thư viện OpenNI không hỗ trợ cho việc điều khiển động cơ của Kinect, nên việc cài đặt cần chút thủ thuật với việc cài đặt thêm CL. Cả OpenNI và CL đều đóng vai trò như driver truy xuất phần cứng Kinect nên sẽ xung đột nếu cài đặt đồng thời hai thư viện này.

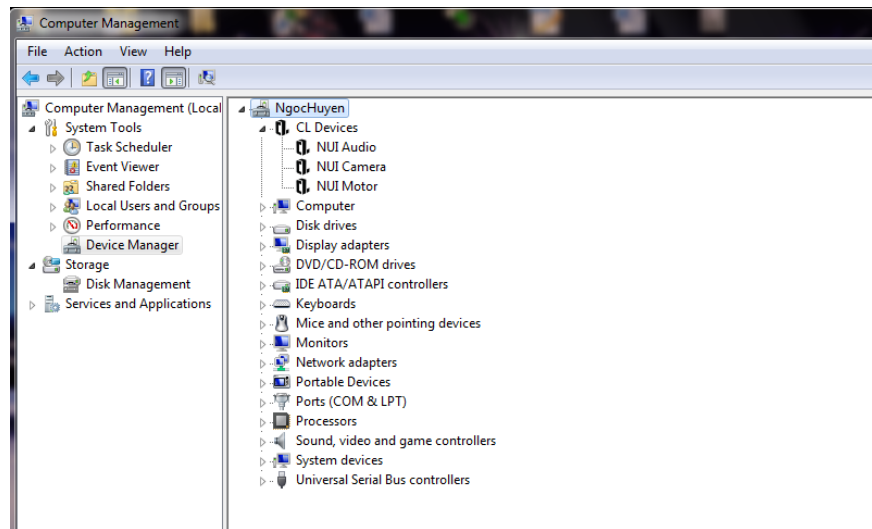
**Bước 1: Cài đặt CL bình thường và đảm bảo chưa cài OpenNi và các chương trình đi kèm trc đó.**

Tải CL từ: <http://codelaboratories.com/nui>

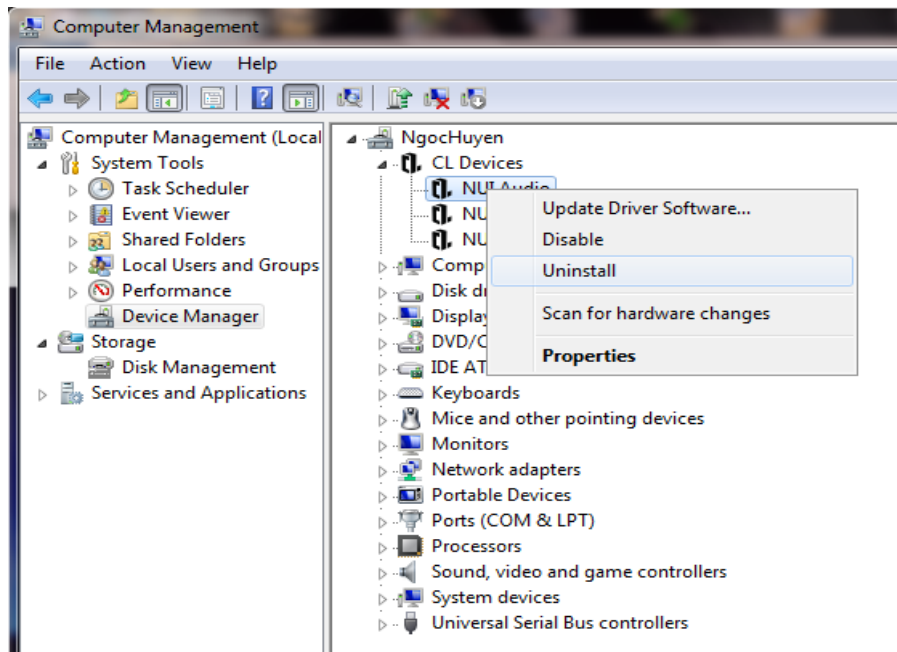




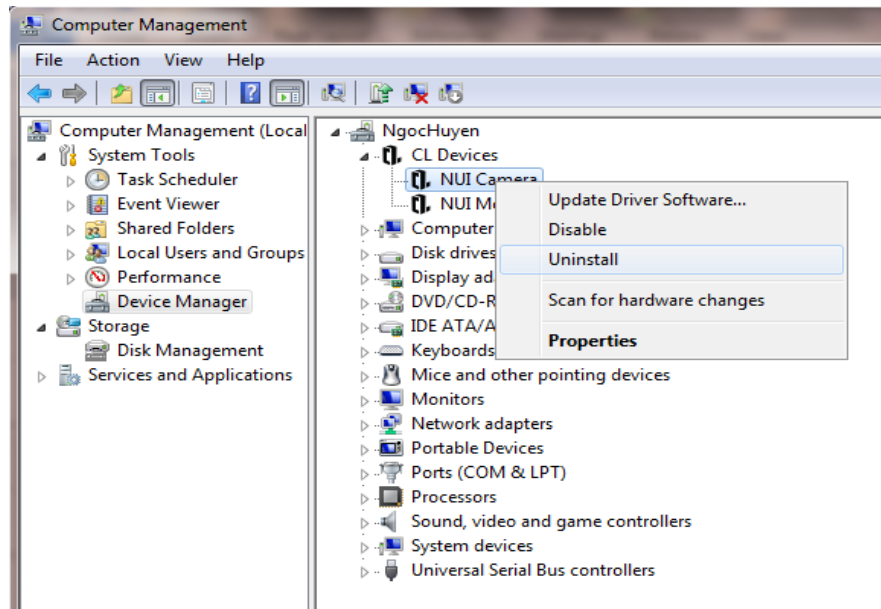
**Bước 2: Mở cửa sổ Computer Management, chọn Uninstall các mục dưới CL devices:**



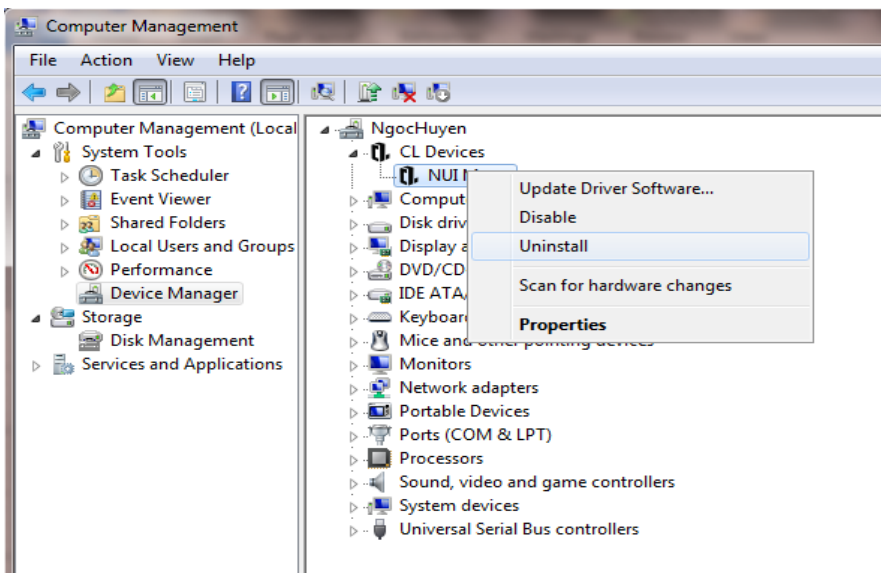
Chọn Uninstall mục NUI audio dưới CL device



Chọn Uninstall mục NUI camera dưới CL device

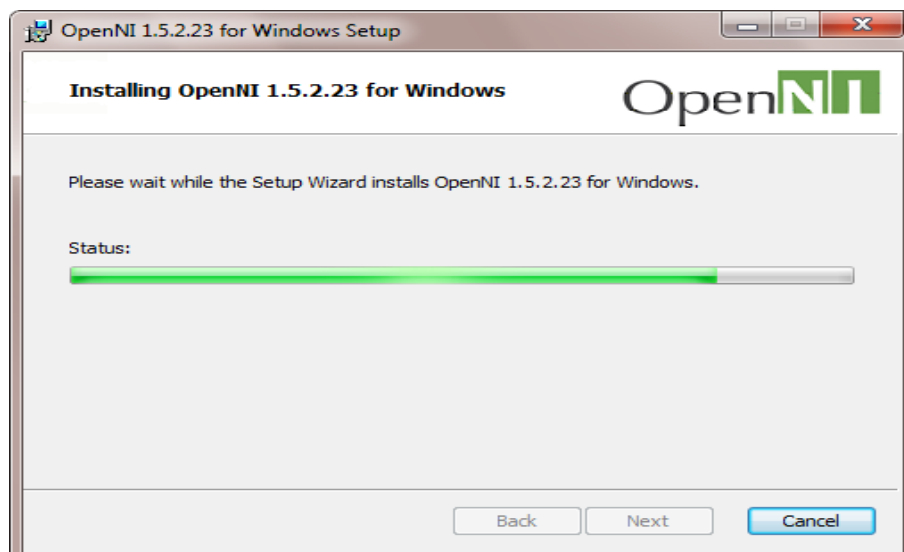
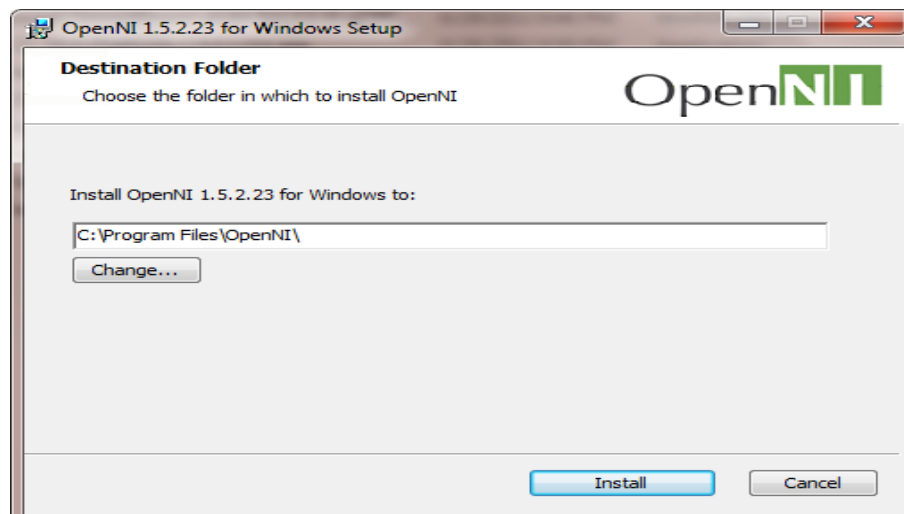
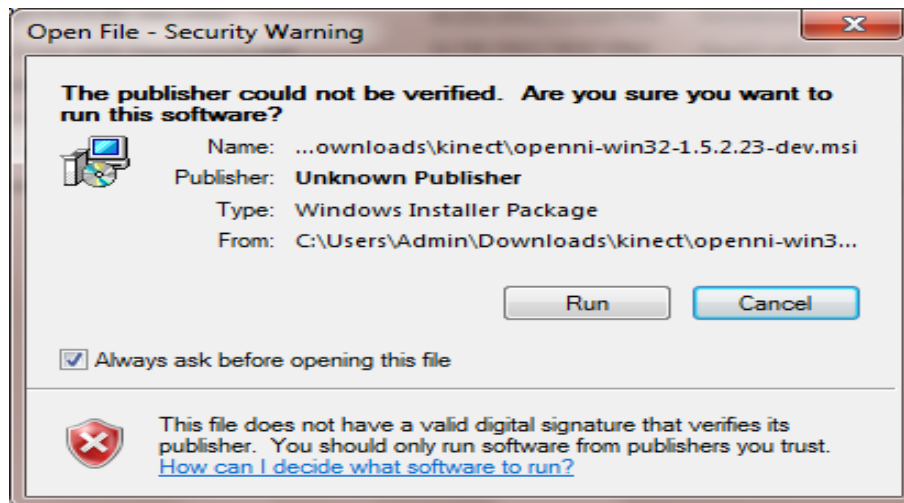


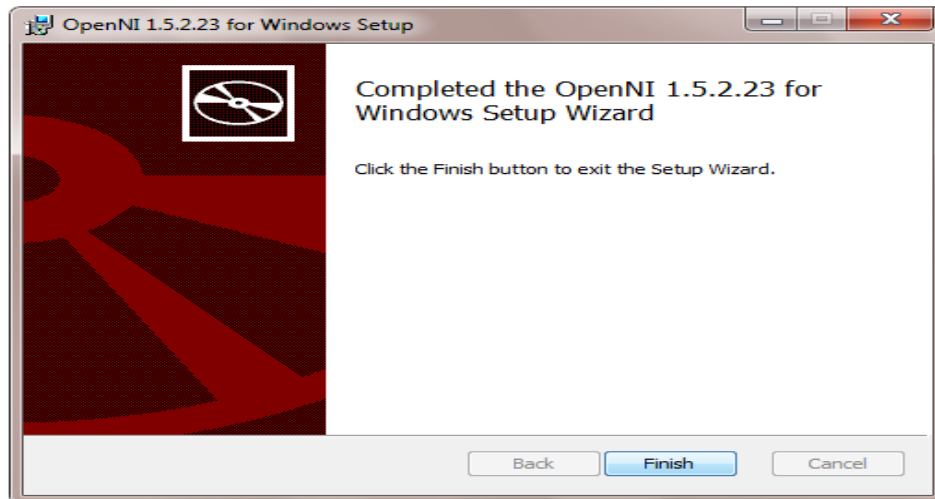
Chọn Uninstall mục NUI motor dưới CL device



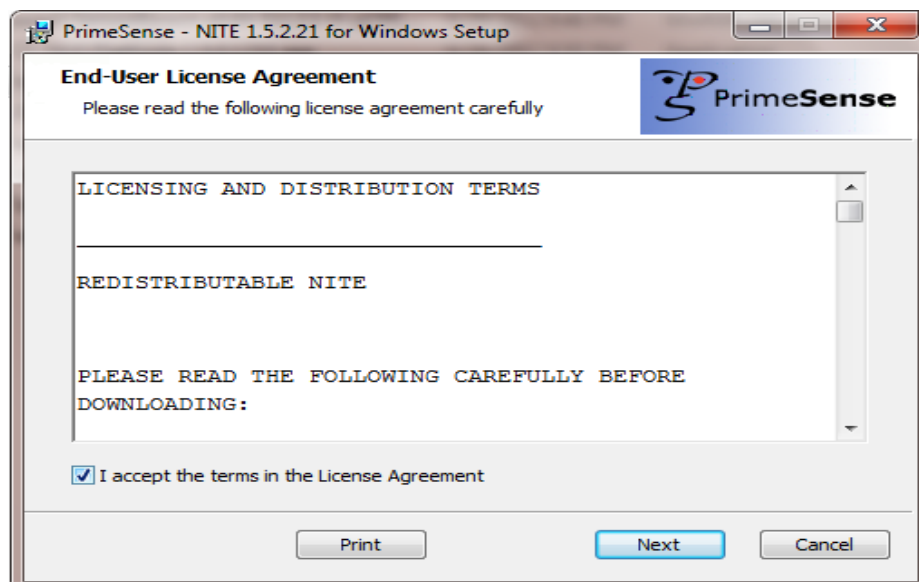
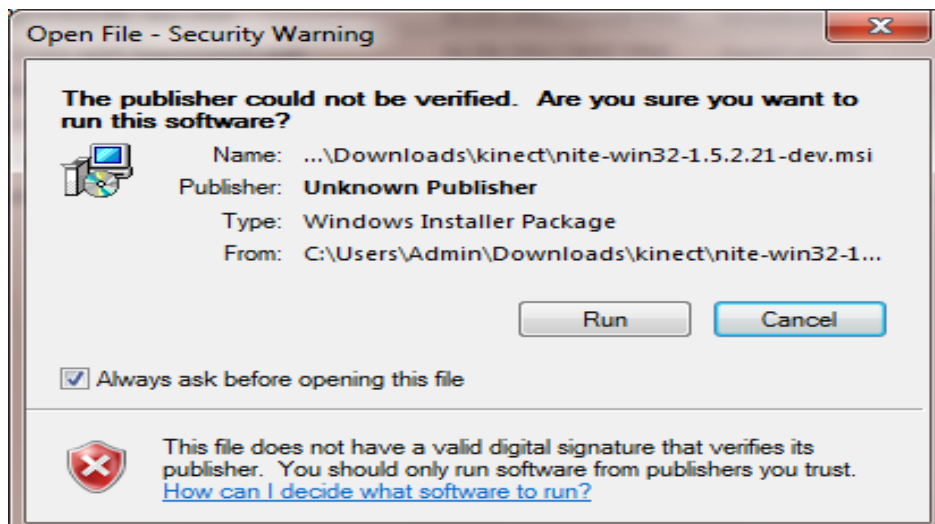
**Bước 3: Cài đặt OpenNI và các chương trình kèm theo**

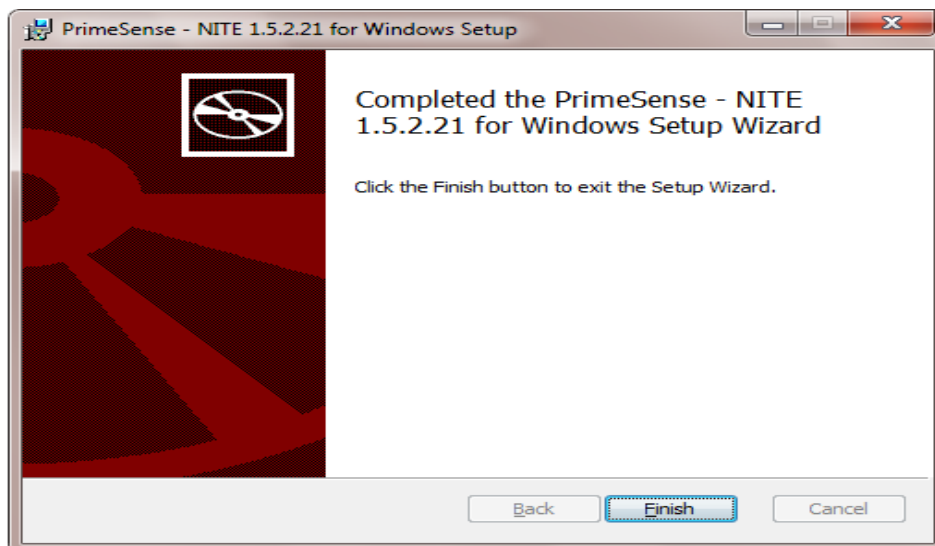
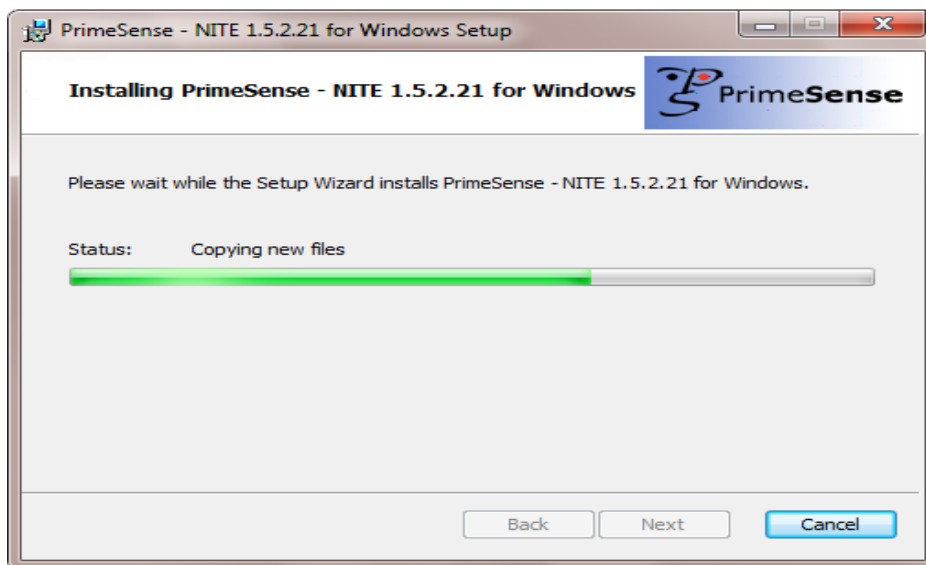
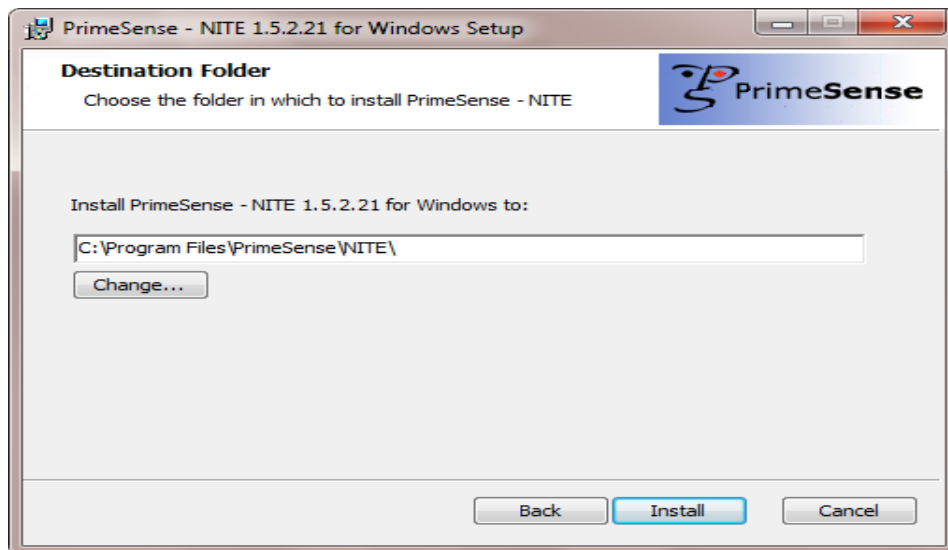
Cài đặt Openni-win32-1.5.2.23-dev.msi



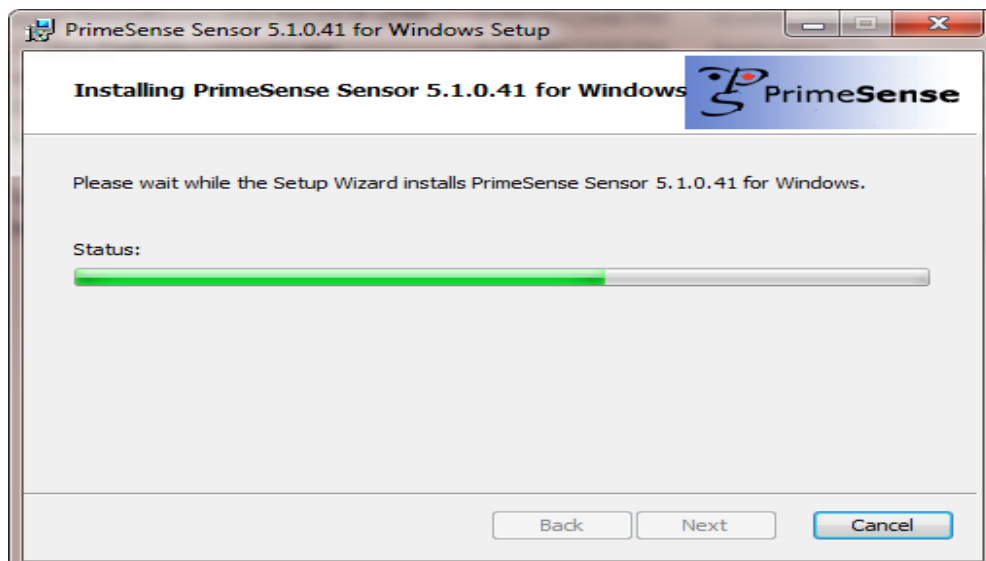
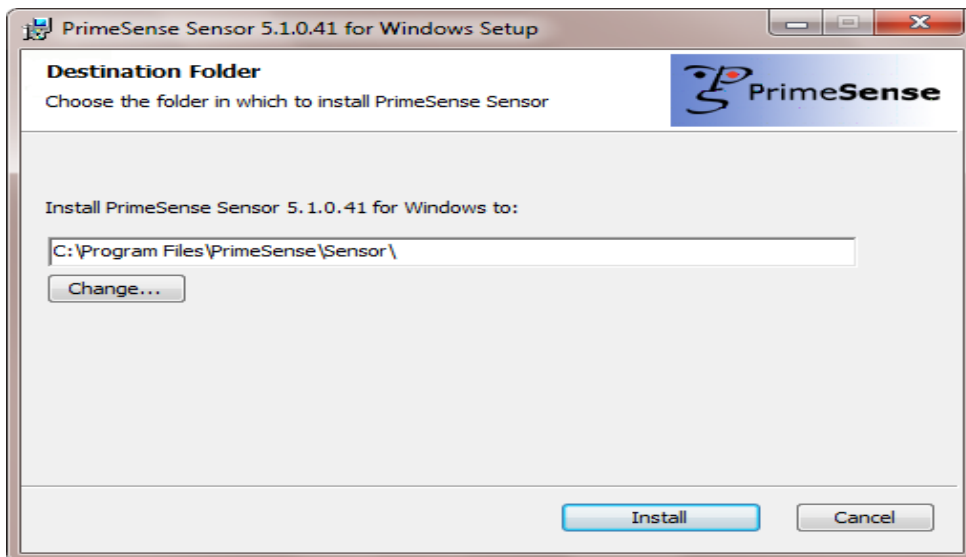


Cài đặt nite-win32-1.5.2.21-dev.msi

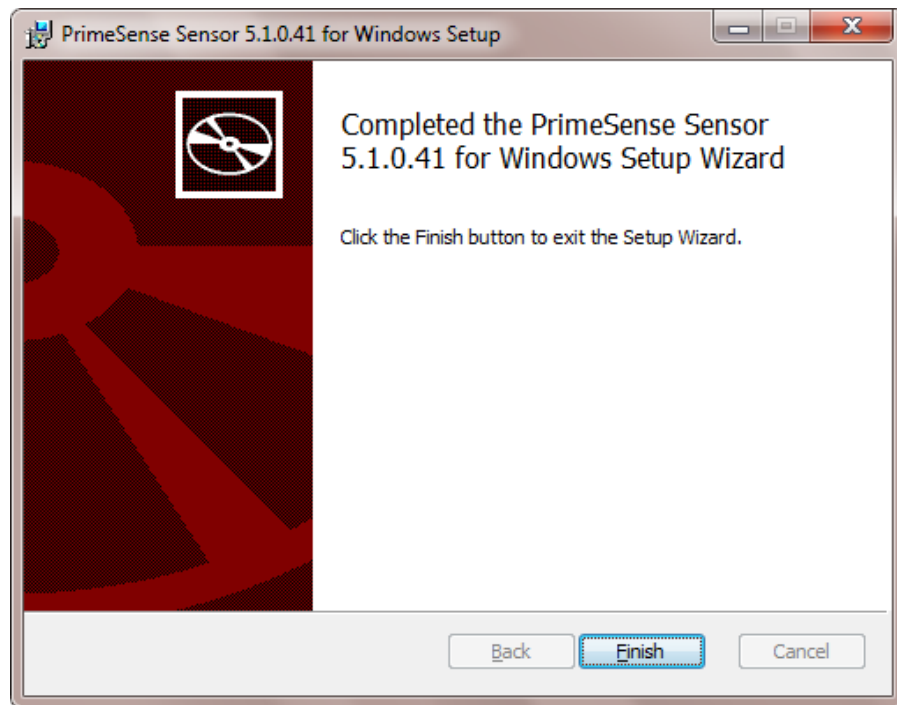




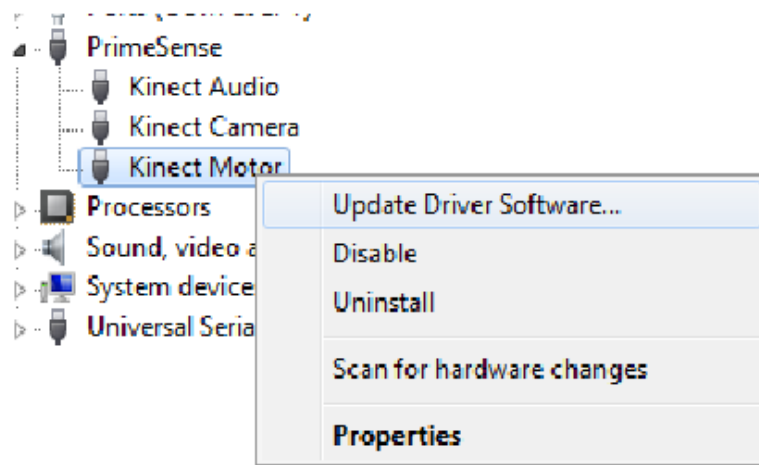
Cài đặt sensor-win32-5.1.0.41-redist.msi







**Bước 4:** Vào cửa sổ như ở bước 1, lúc này Kinect được nhận bởi PrimeSense. Ta nhấn phải chuột trên **Kinect Motor** và chọn **Update Driver Software**.



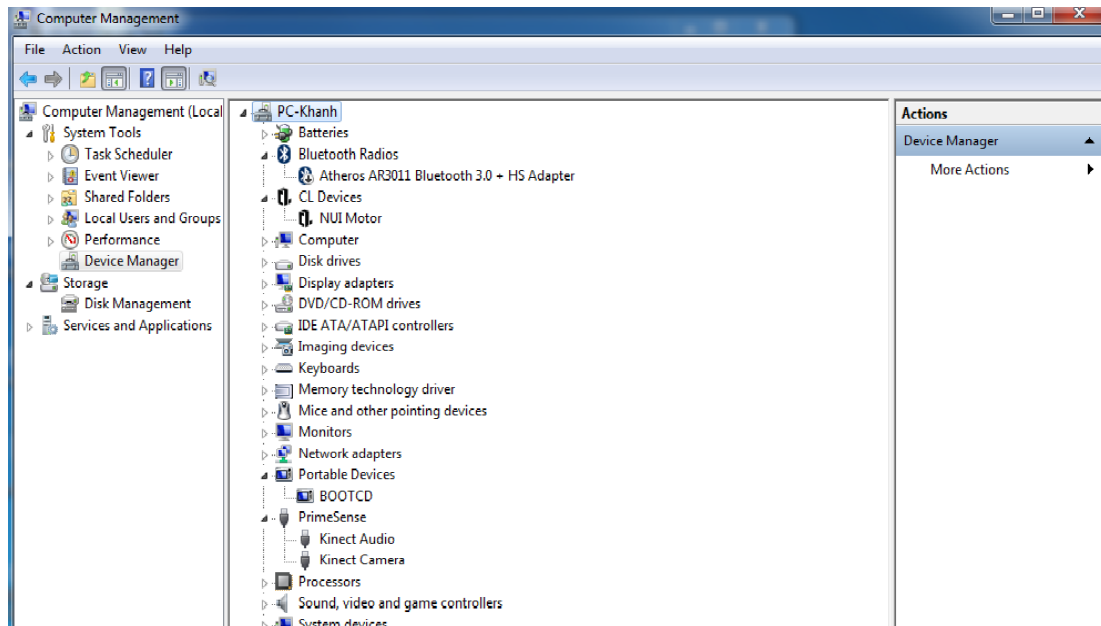
Và duyệt đến thư mục driver của CL:

Search for driver software in this location:

ROGRAM FILES\CODE LABORATORIES\CL NUI PLATFORM\DRIVER ▼ Browse...

☒ Include subfolders

Lúc này ta có thể sử dụng chức năng điều khiển động cơ Kinect trên thư viện OpenNi kết hợp CL.



## TÀI LIỆU THAM KHẢO

### Tài liệu tham khảo Tiếng Anh

- [1] A.M. Fischler and C.R.Bolles, *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography*, Communications of the ACM, vol.24, no.6, pp.381-395, June 1981.
- [2] Rasoul Mojtahedzadeh, *Robot Obstacle Avoidance using the Kinect*, Master of Science Thesis Stockholm, Sweden 2011.

### Tài liệu tham khảo Internet

- [3] Các ứng dụng với mắt thần Kinect, <http://sohoa.vnexpress.net/tin-tuc/doi-song-so/cac-ung-dung-dot-pha-voi-mat-than-kinect-1513399.html>
- [4] Cả nước có khoảng 400 ngàn người mù, <http://www.baomoi.com/Home/DoiNoi-DoiNgoai/www.ktdt.com.vn/Ca-nuoc-co-khoang-400000-nguoi-mu-hai-mat/9526278.epi>
- [5] Chiếc nón kỳ diệu, <http://tuoitre.vn/Giao-duc/Khoa-hoc/Phat-minh-moi/489775/Chiec-non-ky-dieu-cho-nguoi-khiem-thi.html>
- [6] Các phép chuyển hệ trục tọa độ cơ bản, <http://forever.oni.cc/dohoamaytinh/Htm/Chuong6.htm>
- [7] CL NUI Platform installer, <http://codelaboratories.com/nui>
- [8] Documentation of OpenNI, <http://openni.org/Documentation/ProgrammerGuide.html>
- [9] Documentation of PCL, <http://pointclouds.org/documentation/>
- [10] Depth sensing, <http://www.primesense.com/en/technology/115-the-primesense-3d-sensing-solution>
- [11] Main page of OpenKinect, [http://openkinect.org/wiki/Main\\_Page](http://openkinect.org/wiki/Main_Page)
- [12] NAVI Project turns Kinect into a set of eyes for the visually impaired, <http://www.gizmag.com/kinect-as-a-set-of-eyes/18179/>
- [13] History Develop, <http://en.wikipedia.org/wiki/Kinect>

[14] Introduction to Kinect, <http://www.i-programmer.info/ebooks/practical-windows-kinect-in-c/3738-introduction-to-kinect.html?start=1>

[15] Phương pháp xác định vật cản, <http://www.searchdocument.com/pdf/2/3/phuong-phap-duong-cheo.html>

[16] RANSAC, <http://en.wikipedia.org/wiki/RANSAC>

[17] Sensor setup and support, [http://www.microsoft.com/en-us/kinectforwindows/purchase/sensor\\_setup.aspx](http://www.microsoft.com/en-us/kinectforwindows/purchase/sensor_setup.aspx)

[18] Floor Finder Technique,  
[http://www.roborealm.com/tutorial/Obstacle\\_Avoidance/slide040.php](http://www.roborealm.com/tutorial/Obstacle_Avoidance/slide040.php)

[19] Edge Detection,  
[http://www.roborealm.com/tutorial/Obstacle\\_Avoidance/slide020.php](http://www.roborealm.com/tutorial/Obstacle_Avoidance/slide020.php)