**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA CÔNG NGHỆ THÔNG TIN**

**LỚP CỬ NHÂN TÀI NĂNG**

**VŨ TUẤN HƯNG – NGUYỄN VINH TIỆP – HUỲNH QUỐC TRÍ**

**NGHIÊN CỨU VÀ PHÁT TRIỂN THỬ NGHIỆM MỘT SỐ PHƯƠNG PHÁP TƯƠNG TÁC VỚI MÁY TÍNH SỬ DỤNG THỊ GIÁC MÁY TÍNH**

**KHÓA LUẬN TỐT NGHIỆP CỬ NHÂN CNTT**

**TP.HCM, 20****10**

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA CÔNG NGHỆ THÔNG TIN**

**LỚP CỬ NHÂN TÀI NĂNG**

**VŨ TUẤN HƯNG 0612151**

**NGUYỄN VINH TIỆP 0612450**

**HUỲNH QUỐC TRÍ 0612483**

**NGHIÊN CỨU VÀ PHÁT TRIỂN THỬ NGHIỆM MỘT SỐ PHƯƠNG PHÁP TƯƠNG TÁC VỚI MÁY TÍNH SỬ DỤNG THỊ GIÁC MÁY TÍNH**

**KHÓA LUẬN TỐT NGHIỆP CỬ NHÂN TIN HỌC**

**GIÁO VIÊN HƯỚNG DẪN**

**TS.Trần Minh Triết**

**NIÊN KHÓA 2006 – 2010**

|  |
| --- |
| **NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  Khóa luận đáp ứng yêu cầu của LV cử nhân tin học.  TpHCM, ngày …… tháng …… năm 2010  Giáo viên hướng dẫn |
| **NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN**  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  Khóa luận đáp ứng yêu cầu của LV cử nhân tin học.  TpHCM, ngày …… tháng …… năm 2010  Giáo viên phản biện |

**LỜI CÁM ƠN**

Chúng em xin chân thành cảm ơn Khoa Công Nghệ Thông Tin, trường Đại Học Khoa Học Tự Nhiên, Tp.HCM đã tạo điều kiện tốt cho chúng em thực hiện đề tài này.

Chúng em xin chân thành cảm ơn Thầy Trần Minh Triết là người đã tận tình hướng dẫn, chỉ bảo chúng em trong suốt thời gian thực hiện đề tài.

Chúng em cũng xin 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 quí báu trong những năm học vừa qua.

Chúng em 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.

Xin chân thành cám ơn Thầy Nguyễn Khắc Huy, bạn Nguyễn Đức Hoàng, các em khóa 2008 trong nhóm Smart Digital Content (Lê Hoàng Ân, Phạm Trường An, Mạc Cự Khôi Nguyên, Vũ Đức Quang Minh, Nguyễn Hưng, Đoàn Minh Thông) đã hỗ trợ cho nhóm trong quá trình thực hiện và thử nghiệm đề tài.

Mặc dù chúng em 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.

Nhóm thực hiện

Vũ Tuấn Hưng – Nguyễn Vinh Tiệp & Huỳnh Quốc Trí

**ĐỀ CƯƠNG CHI TIẾT**

|  |  |
| --- | --- |
| **Tên Đề Tài**: Nghiên cứu và phát triển thử nghiệm một số phương pháp tương tác với máy tính sử dụng thị giác máy tính | |
| **Giáo viên hướng dẫn:** TS.Trần Minh Triết | |
| **Thời gian thực hiện**: từ ngày 15/12/2009 đến ngày 15/07/2010 | |
| **Sinh viên thực hiện:**  0612450 - Nguyễn Vinh Tiệp  0612151 - Vũ Tuấn Hưng  0612483 - Huỳnh Quốc Trí | |
| **Loại đề tài**: Nghiên cứu lý thuyết về thị giác máy tính trong lĩnh vực tương tác người - máy, tìm hiểu công nghệ và xây dựng ứng dụng thử nghiệm | |
| **Nội Dung Đề Tài:** Khảo sát, nghiên cứu, phân tích một số phương pháp tương tác người – máy sử dụng thị giác máy tính; đề xuất giải pháp tương tác máy tính để sử dụng một và hai camera; từ đó xây dựng thử nghiệm một số ứng dụng cho phép tương tác giữa người – máy.  Nội dung chi tiết của đề tài bao gồm:   * Nghiên cứu, khảo sát các kỹ thuật HCI, kiến trúc hệ thống HCI * Một số vấn đề về camera: mô hình và các xác định các tham số của camera * Hai bài toán quan tâm: Dựng đối tượng 3D ảo dựa trên đối tượng thật sử dụng một camera, tương tác máy tính dựa vào thông tin 3D của đối tượng sử dụng 2 camera.   + ***Thực tại ảo tăng cường sử dụng một camera để dựng một thành phố 3D ảo***: mục tiêu là nhận diện và tái tạo thông tin 3D của đối tượng (các lá bài) và đặt trên đó những kiến trúc ứng với từng loại lá bài. Người dùng có thể tùy ý thay đổi sắp xếp lại vị trí qua lại lên xuống các kiến trúc. Ý nghĩa thực tế của ứng dụng là xây dựng một thành phố 3D ảo với sự tương tác rất đơn giản từ phía người dùng, tạo sự tiện lợi cho việc thiết kế và triển khai những dự án xây dựng, các công trình kiến trúc.   + ***Hệ thống tương tác với máy tính bằng chuột ảo, sử dụng thông 3D tái tạo từ bàn tay của 2 camera***: mục tiêu của ứng dụng là giả lập việc sử dụng chuột bằng các chuyển động tương ứng của bàn tay và ngón tay. Hệ thống sẽ chỉ sử dụng 2 camera với chi phí thấp để thực hiện rút trích thông tin của tay. Ý nghĩa thực tế của ứng dụng này là tìm một cách thay thế việc sử dụng chuột thông thường bằng một phương thức thuận tiện hơn với người dùng. * Ứng dụng thử nghiệm:   + Trò chơi “Lá bài ma thuật” dựa trên trò chơi “Eye of Judgement” của hãng Sony   + Ứng dụng chuột ảo từ thông tin 3D rút trích từ 2 camera,   + Ứng dụng Surface tương tác trực tiếp với màn hình máy tính sử dụng 2 camera | |
| **Kế Hoạch Thực Hiện**:   * 15/12/2009- 14/01/2010: Khảo sát về các hướng tiếp cận trong lĩnh vực tương tác người – máy, tập trung vào các nghiên cứu sử dụng Thị giác máy tính. * 15/01/2009- 14/02/2010: Tìm hiểu mô hình camera và cách xác định các tham số (trong và ngoài) của camera * 15/02/2009- 14/03/2010: Nghiên cứu và xây dựng giải pháp để lấy thông tin vị trí trong không gian 3 chiều của đối tượng đã được đăng ký trước trong hệ thống từ 1 camera * 15/03/2009- 14/04/2010: Nghiên cứu và xây dựng giải pháp để xác định vị trí trong không gian 3 chiều của bàn tay và các hành động của bàn tay với hình ảnh ghi nhận từ 2 camera * 15/04/2009- 14/05/2010: Phát triển thử nghiệm ứng dụng minh họa “Lá bài ma thuật” và phát triển thử nghiệm hệ thống framework hỗ trợ tương tác với máy tính sử dụng thông tin ghi nhận từ 2 camera * 15/05/2009- 14/06/2010: Phát triển thử nghiệm ứng dụng chuột ảo và surface * 15/06/2009- 14/07/2010: Hoàn thiện luận văn | |
| **Xác nhận của GVHD** | **Ngày 15 tháng 7 năm 2010 SV Thực hiện** |

**Mục lục**

🙢🕮🙠

[Chương 1 Giới thiệu 1](#_Toc297369920)

[1.1 Đặt vấn đề 1](#_Toc297369921)

[1.2 Mục tiêu của đề tài 4](#_Toc297369922)

[1.3 Nội dung luận văn 5](#_Toc297369923)

[Chương 2 Cấu hình kineck để thu nhận dữ liệu và xử lý nhiễu 6](#_Toc297369924)

[2.1 Kĩ thuật lấy thông tin ảnh màu và ảnh độ sâu từ kineck 6](#_Toc297369925)

[2.2 khử nhiễu ảnh màu 6](#_Toc297369926)

[Tài liệu tham khảo 9](#_Toc297369927)

**Danh sách hình**

🙢🕮🙠

[Hình 1.1 Những thành tựu trong HCI từ những năm đầu [55] 2](#_Toc297369928)

[Hình 2.1 mặt nạ màu thông dụng của bộ lọc Bayer[ 91][ 93] 7](#_Toc297369929)

[Hình 2.2 Ảnh màu (được phóng to 4 lần) trước và sau khi sử dụng bộ lọc Bayer (thuật toán nội suy EdgeAware) 8](#_Toc297369930)

Danh sách bảng

🙢🕮🙠

**No table of figures entries found.**

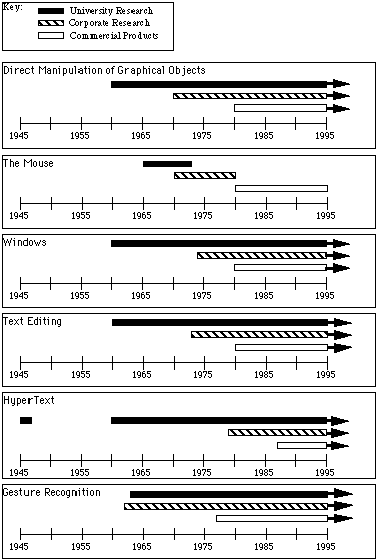
# Giới thiệu

**Tóm tắt chương:**

*✍ Nội dung chương 1 trình bày tổng quan về luận án, mục tiêu của luận án.* *Nội dung tóm tắt của từng chương trong luận án được trình bày ở cuối phần này.*

## Đặt vấn đề

Từ những năm đầu xuất hiện của máy tính, vấn đề giao tiếp giữa con người với máy tính (Human-Computer Interaction, viết tắt là HCI) nổi lên thành một trong những lĩnh vực được quan tâm rất nhiều. Nếu các nghiên cứu trước kia tập trung vào việc thiết kế các mô hình nhập liệu hiệu quả, các nghiên cứu gần đây về HCI tập trung vào việc đơn giản và tự nhiên hóa cho quá trình tương tác giữa người và máy. Với sự đột phá ngày càng mạnh mẽ của các thiết bị phần cứng, HCI ngày nay đã đạt được nhiều bước tiến quan trọng, khiến cho việc sử dụng máy tính trở nên tiện dụng và thân thuộc với con người hơn. Xét quá trình đi lên của HCI, từ những ngày đầu của máy tính, khi dữ liệu được đưa vào bằng việc bật tắt các công tắc hoặc nhét các thẻ bấm lỗ, đến khi con người có thể sử dụng cử chỉ, giọng nói để làm việc với máy tính như trong những năm gần đây, ta có thể thấy được những tiến bộ đạt được trong lĩnh vực HCI là rất lớn. Năm 1995, Brad A. Myers trong bài báo “*A Brief History of Human Computer Interaction Technology*” **Error! Reference source not found.**], đã đưa ra một bảng tóm tắt về những thành tựu HCI từ những năm 1960, được thể hiện ở Hình 1.1. Có thế nói, những thành tựu trên đều đã làm thay đổi hoàn toàn cách thức con người sử dụng máy tính, từng bước mang máy tính lại “gần” con người hơn. Ở **Error! Reference source not found.**, chúng tôi sẽ đi sâu tìm hiểu kỹ hơn về HCI, bao gồm định nghĩa, kiến trúc của các hệ thống HCI cũng như trình bày về những hướng tiếp cận đang được quan tâm nhiều trong HCI.



Hình . Những thành tựu trong HCI từ những năm đầu Error! Reference source not found.]

Trong những thập kỷ gần đây, lĩnh vực thị giác máy tính (***Computer Vison)*** đã đạt được những bước tiến đáng kể, với sự cải thiện lớn về hiệu năng cũng như tính mạnh mẽ (***robustness*)** của các thuật toán phát hiện, nhận dạng, theo vết và mô hình hóa đối tượng, đặc biệt là với yêu cầu thực thi thời gian thực **Error! Reference source not found.**]. Đồng thời với sự tăng tốc của thuật toán, các thiết bị phần cứng sử dụng trong thị giác máy tính, điển hình như camera, ngày càng có nhiều tính năng mới và độ phân giải cao với giá thành hợp lý. Với những lý do trên đó, thị giác máy tính ngày nay đã trở thảnh một thể thức nhập liệu khả thi cho HCI. Hàng loạt những nghiên cứu, dự án về HCI theo hướng tiếp cận thị giác máy tính xuất hiện. Hệ thống Sixth Sense của MIT Media Lab **Error! Reference source not found.**] hay dự án Natal của Microsoft **Error! Reference source not found.**] đều là những ví dụ cho xu hướng sử dụng thị giác máy tính vào HCI. Đặc biệt, khi xét về khía cạnh công nghiệp, phản ứng của cộng đồng đối với những dự án trên đều rất tích cực, là một tín hiệu khả quan cho tương lai không xa của các hệ thống HCI dựa trên nền tảng thị giác máy tính.

Nhiệm vụ chính của thị giác máy tính trong các hệ thống HCI là phát hiện, nhận dạng và mô hình những thông tin thị giác bổ sung (có thể có được từ camera) và trả về những thông tin liên quan như vị trí, biểu lộ nét mặt, cử chỉ của tay, ngón tay,… Những công việc trên cũng có thể được thực hiện bằng những thiết bị chuyên dụng như các thiết bị cảm ứng ảnh sáng, cảm ứng lực; tuy nhiên, chúng sẽ không thể làm cho người dùng cảm thấy tự nhiên khi sử dụng và thao tác. Một số công việc chủ yếu của thị giác máy tính trong HCI là làm sao để có thể xác định được: **Error! Reference source not found.**]

• Vị trí, sự xuất hiện của đối tượng

• Định danh của đối tượng

• Biểu lộ cảm xúc của đối tượng

• Hướng chú ý của đối tượng

• Các cử động cơ thể và sự chuyển động

• Điệu bộ, cử chỉ của đối tượng

• Hoạt động của đối tượng

Những hệ thống HCI tận dụng những thông tin thị giác rút trích từ phía người dùng như vậy sẽ mang tính “đời thực” hơn, tạo ra cảm giác tự nhiên, thoải mái và thuận tiện khi thao tác. Việc sử dụng máy tính, do đó, cũng sẽ được mở rộng với nhiều đối tượng hơn, bao gồm cả những người có những khiếm khuyết cơ thể, gặp khó khăn trong việc thao tác với các thiết bị tương tác phổ biến như chuột, bàn phím.

Xuất phát từ những phân tích trên về việc áp dụng thị giác máy tính vào HCI, có thể thấy được tiềm năng rất lớn của hướng nghiên cứu này cũng như các ứng dụng liên quan, đứng trên cả phương diện công nghiệp lẫn xã hội. Điều này đã thúc đẩy nhóm chúng tôi chọn đề tài “Nghiên cứu và phát triển thử nghiệm một số phương pháp tương tác với máy tính sử dụng thị giác máy tính”.

## Mục tiêu của đề tài

Mục tiêu của đề tài nhằm khảo sát, nghiên cứu, phân tích một số phương pháp tương tác người – máy sử dụng thị giác máy tính; đề xuất giải pháp tương tác máy tính để sử dụng một và hai camera; từ đó xây dựng thử nghiệm một số ứng dụng cho phép tương tác giữa người – máy.

Chúng tôi sẽ trình bày hai ứng dụng mà nhóm đã triển khai xây dựng bao gồm:

* ***Thực tại ảo tăng cường sử dụng một camera để dựng một thành phố 3D ảo***: mục tiêu là nhận diện và tái tạo thông tin 3D của đối tượng (các lá bài) và đặt trên đó những kiến trúc ứng với từng loại lá bài. Người dùng có thể tùy ý thay đổi sắp xếp lại vị trí qua lại lên xuống các kiến trúc. Ý nghĩa thực tế của ứng dụng là xây dựng một thành phố 3D ảo với sự tương tác rất đơn giản từ phía người dùng, tạo sự tiện lợi cho việc thiết kế và triển khai những dự án xây dựng, các công trình kiến trúc.
* ***Hệ thống tương tác với máy tính bằng chuột ảo, sử dụng thông 3D tái tạo từ bàn tay của 2 camera***: mục tiêu của ứng dụng là giả lập việc sử dụng chuột bằng các chuyển động tương ứng của bàn tay và ngón tay. Hệ thống sẽ chỉ sử dụng 2 camera với chi phí thấp để thực hiện rút trích thông tin của tay. Ý nghĩa thực tế của ứng dụng này là tìm một cách thay thế việc sử dụng chuột thông thường bằng một phương thức thuận tiện hơn với người dùng.

Trong mỗi ứng dụng, chúng tôi sẽ lần lượt liệt kê những bài toán gặp phải cũng như tìm hiểu vả đề xuất phương án để giải quyết cho từng bài toán. Bên cạnh đó, những kết quả thực nghiệm và nhận xét đánh giá cũng được đưa vào cuối mỗi phần trình bày.

## Nội dung luận văn

Luận văn sẽ bao gồm 8 chương:

* : trong chương này, chúng tôi sẽ trình bày tổng quan về luận văn cũng như nêu rõ mục tiêu của đề tài mà chúng tôi hướng đến. Bên cạnh đó một số nét khái quát về những ứng dụng mà chúng tôi xây dựng cũng được đề cập đến ở phần cuối chương
* Error! Reference source not found.: nội dung chương 2 sẽ trình bày về HCI, bao gồm các định nghĩa, thuật ngữ, các hướng nghiên cứu gần đây và một số kỹ thuật tiên tiến cũng như kiến trúc của hệ thống HCI. Chương sẽ cung cấp cái nhìn tổng quan về HCI đồng thời làm rõ ý nghĩa thực tế trong việc ứng dụng những lĩnh vực như xử lý âm thanh, xử lý ảnh… vào các hệ thống HCI.
* Error! Reference source not found.: trong chương này, chúng tôi sẽ trình bày về nền tảng lý thuyết của mô hình pinhole camera và hiệu chỉnh thông số camera (calibration)
* Error! Reference source not found.: trình bày về lý thuyết và thực nghiệm kỹ thuật tương tác mày tính dựa vào thông tin 3D rút trích từ 1 camera. Chương sẽ trình bày sâu về lý thuyết và các thực nghiệm thống kê liên quan. Việc xây dựng phần mềm ứng dụng dựa trên cơ sở lý thuyết ở **Error! Reference source not found.** sẽ được trình bày ở **Error! Reference source not found.**
* Error! Reference source not found.: xây dựng phần mềm ứng dụng dựa trên cơ sở lý thuyết về rút trích thông tin 3D của đối tượng từ 1 camera.
* Error! Reference source not found.: Lý thuyết và thực nghiệm về các kỹ thuật rút trích thông tin bàn tay, sử dụng 2 camera. Việc xây dựng phần mềm ứng dụng hỗ trợ tương tác người-máy được trình bày trong **Error! Reference source not found.**.
* Error! Reference source not found.: đề xuất framework của một hệ thống HCI dựa trên việc rút trích thông tin bàn tay sử dụng 2 camera. Trình bày một số ứng dụng đã được cái đặt trên cơ sở sử dụng framework đề xuất
* **Chương 8**: trình bày kết luận và hướng phát triển của đề tài

# Cấu hình kineck để thu nhận dữ liệu và xử lý nhiễu

**Tóm tắt chương:**

*✍ Nội dung chương này trình bày kĩ thuật lấy thông tin từ kineck, bao gồm thông tin ảnh màu, ảnh độ sâu, và tình trạng bị nhiễu của ảnh đầu vào.* Đ*ể giải quyết ảnh bị nhiễu, chúng tôi áp dụng kĩ thuật xử lý ảnh bằng bộ lọc Bayer, với thuật toán nội suy màu EdgeAware.*

## Kĩ thuật lấy thông tin ảnh màu và ảnh độ sâu từ kineck

Tóm tắt…

Trình bày tất cả thông tin có thể lấy được từ kineck, recontruction thì chỉ cần 2 thông tin là ảnh RGB và ảnh độ sâu,…

Cài đặt driver

Thiết lập lấy thông tin dùng thư viện mã nguồn mở OpenNI

Thiết lập file cấu hình xuất ra depth +image

Đọc file cấu hình, kiểm tra Node depth + image

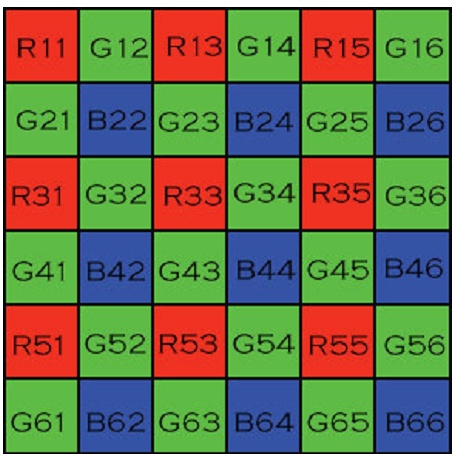
Cấu hình để xuất thông tin depth dựa trên image

Lấy thông tin depth + image, thiết lập chế độ lấy ảnh depth + image đồng bộ, để tránh tình trạng thông tin depth không khớp với image

## khử nhiễu ảnh màu

Trong quá trình lấy dữ liệu ảnh màu RGB và ảnh độ sâu, chúng tôi nhận thấy ảnh màu thu được không tốt như các webcam thông thường (cùng kích thước và cùng cảnh chụp), ảnh RGB bị hiện tượng nhiễu (có sọc đỏ tương tự nhau), rõ nét nhất là ở các vị trí đường biên của các đối tượng trong ảnh RGB. Vấn đề này xuất hiện hầu hết trong các webcam hoặc máy chụp hình kĩ thuật số hiện nay do sử dụng bộ cảm biến đơn sắc (chỉ nhận 1 trong 3 màu), 3 bộ cảm biến liên tiếp ghi nhận thông tin màu của 1 ví trí trên đối tượng [ 1]. Điều này ảnh hưởng đến ảnh kết quả, độ chính xác của các điểm đặc trưng trong ảnh RGB, được tiếp tục dùng trong thuật toán tính so khớp giữa các ảnh. Để giải quyết vấn đề này, một kĩ thuật thường được sử dụng là áp dụng 1 hay vài bộ lọc phù hợp để làm giảm hiện tượng nhiễu. Bộ lọc phù hợp trong trường hợp này là bộ lọc Bayer [ 3].

Nguyên lý hoạt động của Bayer rất đơn giản, áp dụng 1 mặt nạ màu (Bayer Pattern) phù hợp lên từng điểm ảnh, đổng thời thực hiện thuật toán nội suy màu căn cứ trên các điểm lân [ 2][ 4]. Nhiệm vụ chính của Bayer Pattern chính là đảm bảo thuật toán tính nội suy màu, ứng với từng kênh màu (channel) chỉ quan tâm những màu tương ứng trong các điểm ảnh lân cận. Như vậy, ta sẽ có 4 vị trí khác nhau, mà thuật toán nội suy màu cần quan tâm. Quy ước, R là màu đỏ, G là màu xanh lá, B là xanh dương, ta có các vị trí cụ thể như sau: vị trí R, vị trí G dòng R, vị trí G dòng B, vị trí B.



Hình . mặt nạ màu thông dụng của bộ lọc Bayer[ 2][ 4]

Trong bộ lọc Bayer, để tính toán màu cho từng điểm ảnh, có thể sử dụng các thuật toán nội suy phổ biến như Pixel Doubling, Bilinear, Gradient Based, High Quality Linear [ 2]. Nhiệm vụ chính lúc này là giảm sọc ngang trên ảnh, chúng tôi chọn thuật toán nội suy EdgeAware.

Các bước áp dụng bộ lọc Bayer - nội suy EdgeAware như sau:

Bước 1: Với mỗi điểm ảnh, xác định vị trí ứng với mặt nạ màu (1 trong 4 vị trí đã nêu ở phần trên).

Bước 2: Với từng vị trí, từng kênh màu (channel) cụ thể R, G, hay B, ta tính nội suy màu theo EdgeAware như sau:

Vị trí G dòng R, Vị trí G dòng B: Với từng kênh màu, tính trung bình của các điểm lân cận có mặt nạ màu tương ứng.

Vị trí R, Vị trí B: Với kênh màu R, B, tính trung bình của các điểm lân cận có mặt nạ màu tương ứng. Với kênh màu G, xác định 2 điểm G theo chiều dọc hay chiều ngang có độ chênh lệch G ít nhất, tính trung bình cộng 2 điểm G đó. Nếu độ chênh lệch bằng nhau, tính trung bình cộng 4 điểm G[ 5]. Đây chính là nguyên nhân chúng tôi chọn thuật toán nội suy EdgeAware thay vì chọn các thuật toán khác. Đối với các sọc ngang độ biến màu theo chiều ngang cao, do đó, nội suy màu sẽ thường sử dụng thông tin của các điểm màu lân cận theo chiều dọc



Hình . Ảnh màu (được phóng to 4 lần) trước và sau khi sử dụng bộ lọc Bayer (thuật toán nội suy EdgeAware)

Kết quả đạt được sau khi áp dụng bộ lọc Bayer, ảnh thu được từ kineck giảm hẳn hiện tượng sọc ngang. Lưu ý, thu ật to án này cần xử lý riêng các điểm màu ở mép ảnh (ở cạnh và góc ảnh) vì tại các vị trí này thiếu thông tin các điểm màu lân cận.

# Mô hình hóa thế giới thực từ kineck

## Tổng quan

## Kĩ thuật xác định vị trí 3D (Pose) tương đối giữa 2 đám mây.

Trong hầu hết các ứng dụng, thư viện ghép nối đám mây hiện nay (chương trình meshlab – plugins align, thư viện pcl – code align, rgbdemo của Ph.D Burrus Nicolas), qua phân tích code của các chương trình/ thư viện này, kĩ thuật thường được sử dụng là kết hợp kĩ thuật chọn cặp điểm ngẫu nhiên (ransac) và thuật toán tối ưu hoá hàm biến đổi giữa 2 đám mây (Levenberg–Marquardt). Vị trí tương đối giữa 2 đám mây, về bản chất gồm 2 thành phần cơ bản là ma trận xoay và ma trận tịnh tiến. Do đó, nhiệm vụ chính trong phần này là xác định được ma trận xoay và ma trận tịnh tiến để biến đổi đám mây này sang đám mây khác, mà độ lỗi (khoảng cách giữa cặp điểm tương đồng giữa 2 đám mây sau biến đổi) là tối ưu nhất.

### Tối ưu hóa ma trận biến đổi giữa 2 đám mây điểm

Thuật toán Levenberg–Marquardt tương tự như các thuật toán tối ưu khác, là 1 thuật toán lặp, bắt đầu từ tham số khởi tạo với độ lỗi tính toán được cao, sau mỗi lần lặp đỗ lỗi giảm dần dựa trên việc chọn 1 tham số ước lượng phù hợp hơn (damping parameter) [ 6]. Các thông tin cần cung cấp cho thuật toán bao gồm: các cặp điểm tương đồng (xi, yi), hàm biến đổi f từ x sang y, tham số và cách xác định độ lỗi S sau khi biến đổi để công thức  đạt tối thiểu [ 7].

Áp dụng trong trường hợp cần tính vị trí tương đối giữa 2 đám mây, thì các thông tin này được ánh xạ như sau:

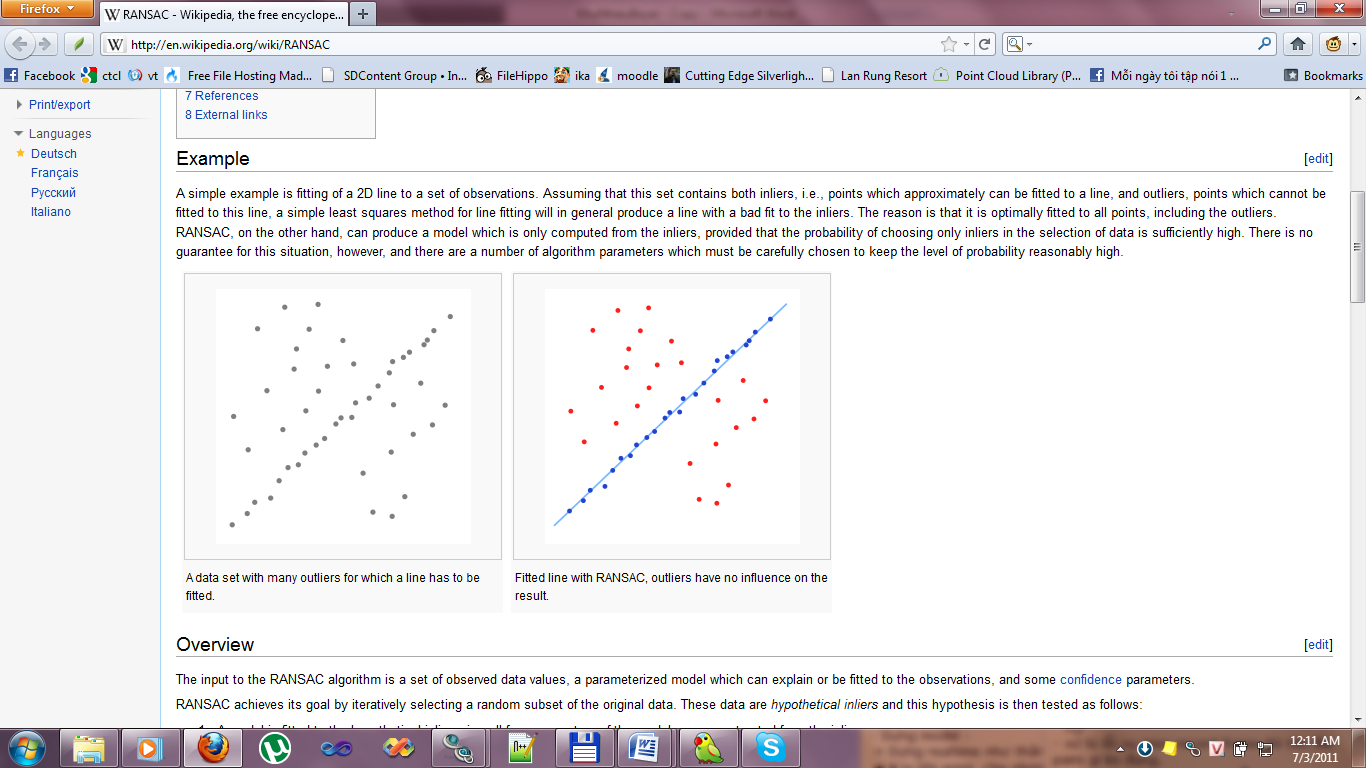
* Cặp điểm tương đồng (xi, yi): là các cặp điểm tương đồng trên ảnh RGB, được xác định bằng thuật toán tìm đặc trưng ảnh (xem mục xx) và ánh xạ thành điểm trong không gian 3D, đã được lọc bỏ bớt các điểm không xác định được chiều sâu hoặc không chính xác (xem mục xx).
* Hàm biến đổi f từ điềm x sang điểm y’: Là phép chiếu (tịnh tiến, xoay) trong không gian 3D, từ điểm x sang y. Đối số là ma trận xoay, ma trận tịnh tiến và điểm x. Kết quả trả ra là điểm y’ tương ứng.
* Tham số : chính là ma trận xoay và tịnh tiến trong không gian 3D, cũng chính là biến số cần tối ưu trong thuật toán này.
* Cách xác định độ lỗi S: là hàm tính bình phương khoảng cách trong không gian 3D giữa 2 điểm y và điểm y’ (x sau biến đổi).

Diễn giải thuật toán 1 cách đơn giản là tìm 1 ma trận biến đổi (tịnh tiến, xoay) sao cho có thể biến đổi đám mây điểm (chứa điểm xi) sang đám mây điểm (chứa điểm yi) một cách chính xác nhất có thể được.

### Xác định ma trận biến đổi tốt nhất

Trong số các cặp điểm tương đồng 3D tính được, không thể tránh khỏi việc có một số cặp điểm không chính xác (sau khi xác định cặp điểm tương đồng trên ảnh RGB, ánh xạ lên không gian 3D, 1 số điểm có hiện tượng ở đám mây A thì nằm trên đường biên đối tượng, ở đám mây B thì điểm tương ứng nằm trên tường) hoặc có độ sai số nhất định so với vật thật, dẫn đến khoảng cách, hướng trong không gian 3D của các cặp này, ảnh hưởng đến kết quả tính ma trận biến đổi trong thuật toán ở phần trước. Giải pháp thường xuyên được áp dụng để giải quyết vấn đề này là dùng thuật toán Ransac.

Thuật toán Ransac (RANdom SAmple Consensus), được công bố bởi 2 nhà thống kê đại tài là Fischler and Bolles, là thuật toán lặp lấy tập mẫu ngẫu nhiên từ tập nguồn có chứa nhiều phần tử lỗi (outlier), để việc tính toán trên tập nguồn tránh những phần tử lỗi (outlier)[ 8]. Thuật toán Ransac được đánh giá là 1 thuật toán mạnh mẽ, do thay vì dùng số lượng điểm nhiều nhất có thể, Ransac lại chọn số điểm ít nhất mà vẫn đạt hiệu quả tương đương[ 9]. Hai vấn đề cần quan tâm khi sử dụng Ransac là: tập mẫu tối thiểu bao nhiêu là phù hợp và lặp bao nhiêu lần thì kết quả đạt được là đủ tốt. Đối với mỗi lần lặp, số lượng phần tử tối thiểu và số lần lặp cần thiết sẽ được tính lại dựa trên tỉ lệ các phần tử phù hợp (inlier) và các phần tử lỗi (outlier) của những vòng lặp trước đó. Công thức chi tiết tham khảo thêm trong tài liệu gốc [ 10].



Hình . Minh họa việc dùng Ransac áp dụng cho việc tìm đường thẳng trong mặt phẳng[ 8].

Áp dụng thuật toán Ransac cho bài toán 1 số cặp điểm tương đồng trong không gian bị lỗi, ta thực hiện các bước như sau[ 9]:

Bước 1: Chọn ngẫu nhiên 1 số lượng phần tử theo công thức [ 10].

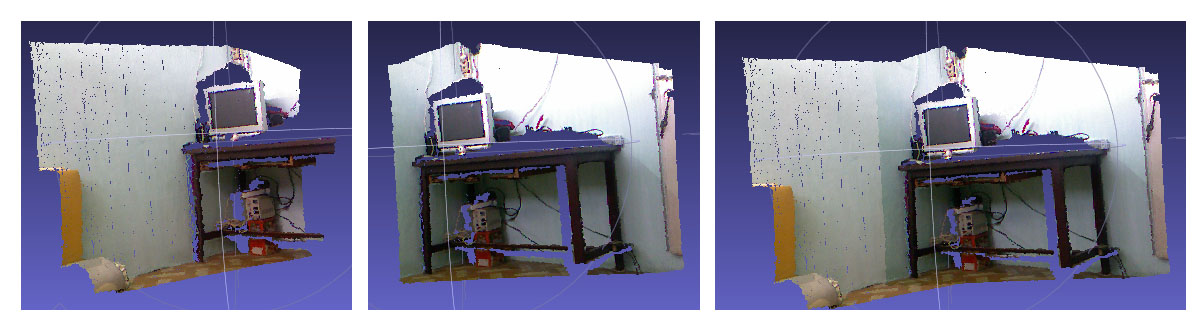
Bước 2: Ước lượng ma trận biến đổi dựa trên thuật toán Levenberg–Marquardt, xác định.

Bước 3: Loại bỏ nếu độ chính xác không chấp nhận được.

Bước 4: Nếu kết quả ở bước 2 chấp nhận được, xác định tỉ lệ các cặp điểm bị lỗi (outlier) khi áp dụng kết quả cho toàn bộ tập nguồn.

Bước 5: Chọn kết quả nàu nếu như tỉ lệ lỗi thấp nhất so với các lần lặp trước đó.

Bước 6: Thực hiện lặp lại bước 1 đến khi hết số vòng lặp cần thiết.



Hình . Minh họa kết quả ghép 2 đám mây sau khi áp dụng ransac và Levenberg–Marquardt để xác định ma trận biến đổi.

Tài liệu tham khảo

[ ] Dipl. Ing. (FH) Uwe Furtner (2001), “Color processing with Bayer Mosaic sensors”, *p.1-2, Image-Processing Products Manager, MATRIX Vision GmbH.*

[ ] Rémi Jean, “Demosaicing with The Bayer Pattern”, *p.1-5, Department of Computer Science, University of North Carolina.*

[ ] Paul M. Hubel, John Liu and Rudolph J. Guttosch , “Spatial Frequency Response of Color Image Sensors: Bayer Color Filters and Foveon X3”, *p.1-2, Foveon, Inc. Santa Clara, California.*

[ ] Bradley Atcheson, “Subpixel Rendering of Bayer-Patterned Images”, *p.1-1, Department of Computer Science, The University of British Columbia.*

[ ] <https://code.ros.org/trac/opencv/ticket/1093>

[ ] K. Madsen, H.B. Nielsen, O. Tingleff (2004), “Methods for Non-Linear problems”, *p.10-11, Informatics and Mathematical Modelling, Technical University of Denmark.*

[ ] <http://en.wikipedia.org/wiki/Levenberg_Marquardt_algorithm>

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

[ ] Konstantinos G. Derpanis (2010), Overview of the RANSAC Algorithm.

[ ] M.A. Fischler and R.C. Bolles, “Random sample consensus: A paradigm for model ﬁtting with applications to image analysis and automated cartography”. Communications of the ACM, 24(6):381–395, 1981.