



BÁO CÁO ĐỒ ÁN

ỨNG DỤNG SỬ LÝ ẢNH SỐ VÀ VIDEO SỐ



FEBRUARY 7, 2022

KFC
K19CLC

Nội dung

Nội dung.....	1
Lời cảm ơn.....	2
I. Thông tin nhóm:.....	2
II. Ý nghĩa – động lực nghiên cứu:.....	2
III. Phát biểu bài toán:	3
IV. Related works.....	7
V. Nguyên lý - Phương pháp:.....	13
VI. Cấu trúc ba tầng ứng dụng:.....	29
VII. Tổng kết	33
VIII. Một số thuật ngữ - từ viết tắt:.....	33
IX. Tham khảo:.....	34

Lời cảm ơn

Chúng em xin gửi lời cảm ơn chân thành đến Khoa Công nghệ thông tin, Trường đại học Khoa học tự nhiên đã tạo điều kiện thuận lợi cho chúng em học tập và hoàn thành đề tài nghiên cứu này. Đặc biệt, chúng em xin bày tỏ lòng biết ơn sâu sắc đến thầy Lý Quốc Ngọc đã dày công truyền đạt kiến thức và hướng dẫn chúng em trong quá trình làm bài.

Em đã cố gắng vận dụng những kiến thức đã học được trong học kỳ qua để hoàn thành bài tiểu luận. Nhưng do kiến thức hạn chế và không có nhiều kinh nghiệm thực tiễn nên khó tránh khỏi những thiếu sót trong quá trình nghiên cứu và trình bày. Rất kính mong sự góp ý của quý thầy cô để bài tiểu luận của em được hoàn thiện hơn.

Một lần nữa, em xin trân trọng cảm ơn sự quan tâm giúp đỡ của các thầy đã giúp đỡ em trong quá trình thực hiện bài tiểu luận này.

Xin trân trọng cảm ơn!

I. Thông tin nhóm:

1. Tên đồ án: Định vị và tái tạo môi trường xung quanh. vSLAM (Visual Simultaneous Localization and Mapping).
2. Tên nhóm: KFC
3. Các thành viên:

MSSV	Họ và tên
19127618	Nguyễn Thanh Tùng
19127517	Hồ Thiên Phước
19127506	La Trường Phi

II. Ý nghĩa – động lực nghiên cứu:

1. Trong khoa học:
 - Việc thu thập và trích xuất thông tin hình ảnh 3 chiều là một trong những vấn đề mà các nhà khoa học thị giác máy tính đã và đang hướng tới, nó là một xu hướng trong tương lai. Khi chúng ta có được thông tin dạng 3 chiều chúng ta có thể trích xuất các thông tin như kích thước, tọa độ, ... của thế giới thực qua những bức hình một cách dễ dàng.
 - Các nhà khoa học và nhà phát minh đã phát minh ra rất nhiều thiết bị chuyên dụng để làm điều này tiêu biểu là LiDar. Đây là những thiết bị chuyên dụng tương tự với độ chính xác cao, tuy nhiên nó lại rất đắt đỏ để sở hữu và nghiên cứu cá nhân.
 - Từ đó chúng ta có giải pháp là tái tạo lại môi trường 3D từ môi trường 2D cụ thể phương pháp này là vSLAM. Tuy nhiên việc tái tạo lại chiều không gian thứ 3 chưa bao giờ là việc dễ dàng.
 - Với bước phát triển của vSLAM gần đây đã đóng góp rất nhiều vào sự phát triển của các thuật toán và công nghệ khác. Nó phục vụ tốt hơn cho công việc phát hiện và phân lớp khuôn mặt 3D thay vì 2D đồng thời ràng buộc độ khớp của class ID,.. hay trong các hệ thống tạo dựng bản đồ của các robot tự hành.
 - Bài báo cáo này sẽ tập trung vào nghiên cứu ưu nhược điểm, cách thức hoạt động của vSLAM và cách khắc phục nếu có.
2. Trong thực tiễn:

- Việc định vị và tái tạo môi trường xung quanh đóng vai trò rất lớn trong việc vận hành của các hệ thống tự hành, những hệ thống tự hành yêu cầu rất nhiều về việc kiểm soát được vị trí trong môi trường như xe tự hành, robot tự hành. Bên cạnh đó SLAM còn được ứng dụng trong các robot tái tạo bản đồ của môi trường xung quanh, hay việc sử dụng việc tái tạo 3D để tái tạo cơ thể con người để đưa ra các đề xuất trang phục phù hợp với mỗi thân hình khác nhau.
- Thiết bị tự hành ứng dụng cho những đơn vị cứu trợ, bảo hành cầu cống, đường ống ngầm/đập nước. Những nơi con người hạn chế lui tới được như địa hình dốc, hiểm trở, drone kèm camera có thể thay con người quan sát tìm kiếm nạn nhân; hoặc những nơi như bên trong các thiết bị, vật tư cần được bảo hành sửa chữa như cầu cống,.. drone với máy quét có thể quét được cấu trúc bên trong để con người đưa ra phán đoán rằng vật có cần được bảo hành hay không. Những nơi rò rỉ phóng xạ như nhà máy hạt nhân, những nơi rò rỉ khí amoniac như xưởng nước đá, con người sẽ hạn chế lui tới được, và khi đó máy móc sẽ làm thay để phục vụ công tác cứu trợ cứu nạn, nó sẽ giúp hạn chế phần nào thiệt hại. Hoặc chỉ đơn giản là đường cống hôi thối mà con người hạn chế lui tới, camera và máy quét sẽ thay con người lấy những thông tin những nơi như vậy để con người đưa ra phán đoán về việc bảo trì,..

III. Phát biểu bài toán:

1. Bài toán:

- Chúng ta có một thiết bị tự hành (robot, xe, drone, ...), chúng ta muốn thiết bị đó di chuyển từ điểm A tới điểm B và trong quá trình di chuyển thiết bị có khả năng tự tránh né vật cản và lưu lại bản đồ.
- Với 2 điểm A và B chúng ta phải cung cấp tọa độ hoặc khoảng cách và phương hướng cho thiết bị xác định được điểm đi và tới. Chúng ta có thể cung cấp cho thiết bị một bản đồ đã có sẵn để thiết bị xác định điểm tới chính xác hơn.

2. Đầu vào của ứng dụng:

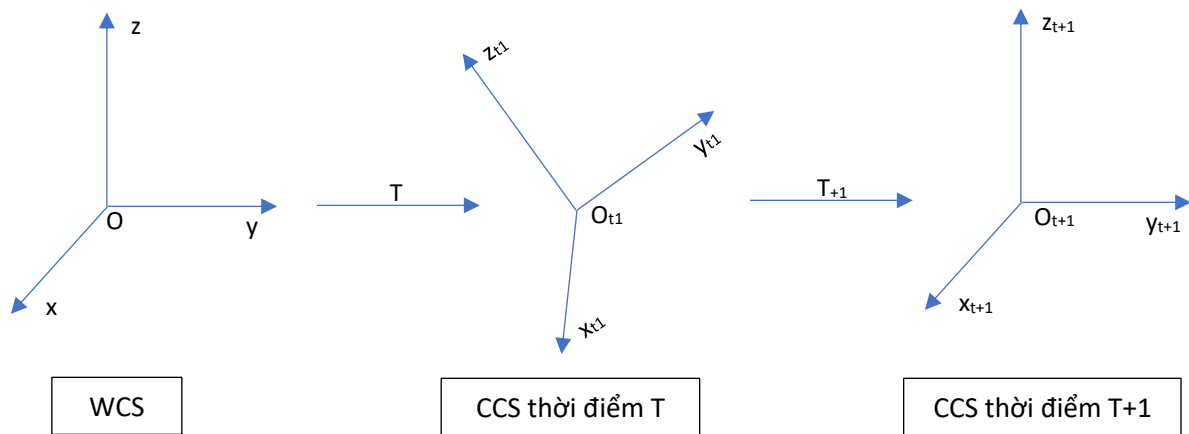
- Thông tin dạng hình ảnh: tập ảnh hoặc video RGB-D (có thể là có sẵn từ trước hoặc qua trong thời gian thực) chứa cảnh vật môi trường để phần mềm tái tạo lại vị trí và quang cảnh. Các tấm ảnh phải có mối quan hệ về không gian và thời gian.
 - Thông tin dạng tín hiệu: dữ liệu được thu thập từ các cảm biến (cảm biến hồng ngoại, cảm biến siêu âm...).
 - Thông tin dạng đám mây điểm 3 chiều: dữ liệu thu từ các thiết bị quét như Lidar...
- ❖ Trong đồ án này, nhóm quyết định nghiên cứu kiểu được đầu vào bằng hình ảnh RGB-D.

3. Đầu ra:

- Tọa độ và bản đồ của thiết bị trong môi trường.

4. Hệ tọa độ:

- Chúng ta có 2 hệ toạ độ: hệ toạ độ gốc ban đầu (tại điểm xuất phát) (world coordinate system) và hệ toạ độ camera (camera coordinate system).



- T : Phép biến đổi toạ độ
- WCS: World coordinate system
- CCS: Camera coordinate system
- Khi bắt đầu xuất phát hệ toạ độ của thiết bị sẽ là WCS, sau khi thiết bị di chuyển vị trí camera hệ toạ độ của thiết bị lúc này sẽ là CCS. Với mỗi khung hình khác nhau chúng ta sẽ có một CCS khác nhau.
- Tựu chung lại việc chúng ta cần làm là tìm được phép biến đổi toạ độ camera từ thời điểm T sang thời điểm $T+1$. Có n phép biến đổi toạ độ như vậy. Và từ đó suy ngược lại vị trí từ thời điểm $T+1$ sang thời điểm T và từ thời điểm T về lại WCS để xây dựng một hệ toạ độ đồng nhất, xây dựng lên bản đồ cho thiết bị.

5. Framework:

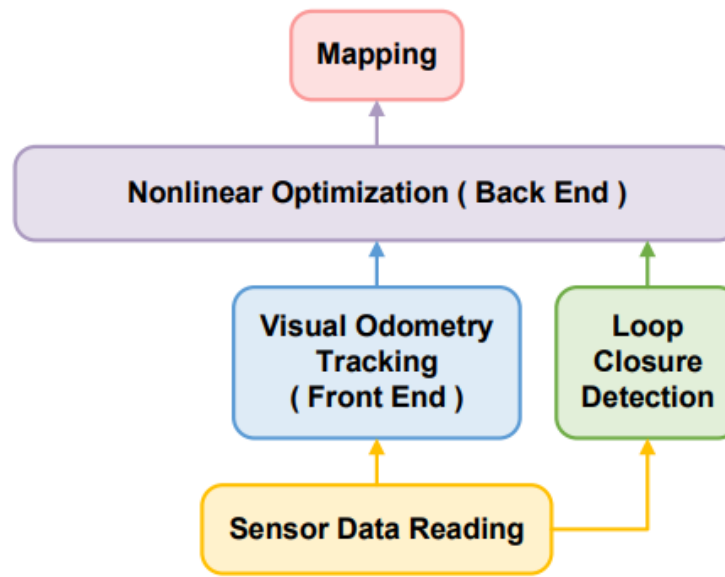


Figure 1.1: Classic Visual SLAM Framework

- **Sensor Data Reading:** bước này tập trung vào việc thu thập và tiền xử lý ảnh đầu vào từ camera của điện thoại.
- **Visual Odometry Tracking:** công việc của bước này là tính toán và ước tính chuyển động của camera và tìm cấu trúc của các hình gần kề nhau, liên kết những điểm trùng nhau lại và tạo thành một bản đồ cục bộ. Bước này còn có một tên gọi khác là front end.
- **Loop Closure Detection:** trong trường hợp camera quay lại vị trí cũ mà nó đã từng quét function này sẽ được kích hoạt và dừng vòng lặp, sau đó mọi thông tin camera sẽ được chuyển xuống back end.
- **Nonlinear Optimization:** ở bước này các hình dáng mà camera thu lại ở những thời điểm khác nhau sẽ được tập trung lại, kết hợp với thông tin về phát hiện đóng vòng lặp nó sẽ tối ưu hoá những thông tin trên. Sau đó xây dựng những toạ độ thống nhất trên phạm vi toàn cục. Vì nó được kết nối ngay sau bước front end nên nó còn được gọi là bước back end.
- **Mapping:** dựa vào những toạ độ đã được tính toán ở bước back end mà nó sẽ tạo ra một bản đồ hoàn chỉnh.

6. Thách thức của bài toán:

Với vSLAM có những thách thức như sau

- Vì giả sử ảnh input là RGB-D nên bài toán đã trở nên đơn giản hơn. Nhưng so với thực tế, trong một vài trường hợp, ảnh input là một dãy ảnh RGB không bao gồm tọa độ 3D (depth), ta cần phải bổ sung thêm công đoạn tái tạo độ sâu của những điểm ảnh.
- Lỗi định vị tích tụ, gây ra độ lệch đáng kể so với giá trị thực tế:
 - SLAM ước tính chuyển động tuần tự, gồm cả những sai số. Những sai số này tích tụ theo thời gian sẽ gây ra độ lệch đáng kể so với thực tế. Điều này sẽ khiến cho bản đồ bị sai khác và méo mó, không giống với thực tế, gây khó khăn trong vận hành và định vị.



Giải pháp khắc phục vấn đề này là chúng ta cần ghi nhớ những đặc điểm và dùng nó như một điểm mốc. Việc tối ưu này được gọi là điều chỉnh gói trực quan trong SLAM (bundle adjustment in visual SLAM).



- Lỗi định vị không thành công và vị trí trên bản đồ bị mất:
 - Lỗi này thường xảy ra khi thiết bị bị di chuyển đột ngột từ vị trí này sang vị trí khác hoặc khi các cảm biến hoạt động không chính xác dẫn đến các lỗi về mặt tính toán. Thiết bị sẽ không thể định vị các vị trí đã đánh dấu trước đó.
 - Lỗi này có thể khắc phục bằng cách kết hợp mô hình chuyển động với nhiều cảm biến để thực hiện các tính năng tính toán dựa trên dữ liệu của cảm biến. Gần đây mạng học sâu cũng đang được áp dụng để xử lý những lỗi về này.
- Chi phí tính toán cao để xử lý hình ảnh, xử lý đám mây điểm và tối ưu hoá:
 - Thông thường các robot sẽ có một bộ xử lý không quá mạnh như trên phòng lab hay máy tính mà nó thường sẽ là những hệ thống nhúng. Nhưng để có thể tạo ra một

bản đồ chính xác, điều cần thiết là phải thực hiện xử lý ảnh và đối sánh đám mây một cách liên tục.

- Một biện pháp giải quyết là chạy song song các tác vụ khác nhau. Chúng ta có thể sử dụng những CPU có nhiều lõi để xử lý tính toán nhiều dữ liệu theo lệnh đơn (SIMD) và GPU nhưng có thể cải thiện tốc độ hơn nữa trong một số trường hợp.
- Hạn chế về sự thích ứng với những thay đổi về môi trường:
 - Vì môi trường xung quanh luôn thay đổi, ví dụ thay đổi vị trí của những đồ nội thất hoặc khi con người di chuyển cũng sẽ là những sự thay đổi khiến cho một cỗ máy phải đau đầu xử lý. Những thay đổi này sẽ làm thay đổi những điểm mốc mà hệ thống đã đánh dấu và sử dụng như những “biển chỉ dẫn”. Vậy nên khi thay đổi môi trường sẽ khiến thiết bị hoạt động sai lệch hoặc phải xây dựng lại bản đồ.
 - Khắc phục: cách khắc phục dễ nhất là từ phía người dùng, chỉ cần họ hạn chế di chuyển đồ đạc, mọi thứ sẽ hoạt động dễ dàng hơn với thiết bị của chúng ta. Hoặc khi họ thay đổi thì chạy lệnh để thiết bị quét lại bản đồ. Còn về phía sự di chuyển của động vật, con người hoặc các thiết bị có khả năng di chuyển khác (đây là những di chuyển liên tục hoặc không), chúng ta cần có sự trợ giúp của các thiết bị cảm biến khác như cảm biến hồng ngoại, cảm biến gia tốc và cảm biến chuyển động để xác định đây không là vị trí cố định của những vật thể đó và loại bỏ nó ra khỏi danh sách những điểm mốc.
- Hạn chế về ánh sáng: Việc thiếu ánh sáng sẽ ảnh hưởng tới camera ở 3 khía cạnh. Thứ nhất, ảnh sẽ có nhiều noise. Thứ 2, tốc độ chụp của camera sẽ giảm. Và cuối cùng tối quá sẽ không thể thấy được vật thể. Tuy nhiên nhược điểm về ánh sáng có thể xử lý bằng cách dùng camera hồng ngoại có thể nhìn trong đêm hoặc dùng các loại cảm biến để thu thập dữ liệu. Tuy nhiên đây vẫn là nhược điểm lớn với những thiết bị cầm tay nhỏ gọn.
- Hạn chế về quyền riêng tư: việc sử dụng camera và thu thập dữ liệu về hình ảnh luôn tồn tại vấn đề về quyền riêng tư và thu thập dữ liệu người dùng trái phép. Không ai có thể hoàn toàn đảm bảo một cái camera sẽ không bị hacker tấn công hay cài mã độc. Thậm chí là bị điều khiển từ xa.

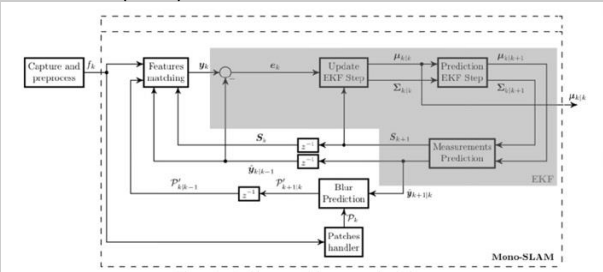
IV. Related works

1. Người ta đã làm gì?

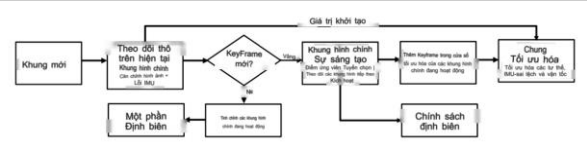
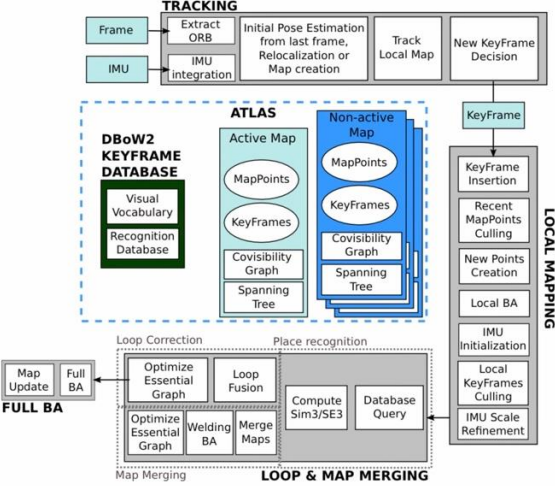
Bảng 1

<i>Stt</i>	Phương pháp (năm)	Đặc trưng	Input	Thiết bị (sensor)	Định vị (camera pose)	Tái tạo
<i>1</i>	MonoSLAM-Monocular (2003)	line segments, đặc trưng góc của mỗi khung hình	Ảnh RGB	Camera đơn	Extended EKF	Khởi tạo bản đồ using known object Extended EKF-based mapping
<i>2</i>	CNN-SLAM-convolutional neural networks (2017)	pixels có cường độ sáng intensities cao nhất		RGB-D camera	Cực tiểu lỗi photometric khung hình hiện tại và các khung hình gần đây nhất	-
<i>3</i>	MMS- mapping system (2014)	blob-like hình dạng tròn hoặc elip	Ảnh RGB-D	RGB-D camera	Giả sử RGB-D camera lấy ảnh RGB-D. Dùng SIFT tìm những đặc trưng tương đồng. Lấy depth của những đặc trưng tương đồng. Đưa các điểm đó (x,y,z) vào $P'=AP$ ra được 6 ẩn số (chính là phép biến đổi A) bằng phương pháp Hồi qui hoặc Ransac.	Dựa trên hệ tọa độ camera tại thời điểm t, capture dữ liệu môi trường xung quanh (đám mây điểm). Mỗi lần di chuyển sẽ phát sinh 1 đám mây điểm 3D. Qui tất cả các đám mây điểm 3D tại mỗi hệ tọa độ camera rồi rạc về 1 hệ qui chiếu chung. Kết quả nhận được 1 đám mây điểm chung so với hệ qui chiếu ban đầu.
<i>4</i>	PTAM- Parallel Tracking and Mapping (2007)	Thông tin độ sâu của một điểm	-	-	Qua mỗi keyframe mới, ước tính initial pose để đối sánh/ chiếu điểm đặc trưng của đám mây điểm map points của ảnh input.	Khởi tạo bản đồ bằng thuật toán five-point, khởi tạo bản đồ độ sâu xài stereo measurement. 3D positions của điểm đặc trưng dùng triangulation và tối ưu vị trí ấy bằng BA-based mapping
<i>5</i>	DTAM- dense tracking and mapping (2011)	pixels có cường độ sáng intensities cao nhất	-	virtual camera	Synthetic view generation từ bản đồ đã tái tạo. Cực tiểu lỗi photometric	Khởi tạo bản đồ using stereo measurement Thông tin depth xài multi-baseline stereo và tối ưu bằng considering space continuity
<i>6</i>	LSD-SLAM-Large-Scale Direct Monocular SLAM (từ semi-dense visual odometry phát triển thành) (2014)	pixels có cường độ sáng intensities cao nhất	Ảnh phân giải thấp	Camera đơn, Cam kép (stereo camera) và omni-directional cameras, 3D laser scanner	Synthetic view generation từ bản đồ đã tái tạo, cực tiểu lỗi photometric	Khởi tạo depth là giá trị ngẫu nhiên Reconstructed areas are limited to high-intensitygradient areas. 7 DoF pose-graph optimization is employed to obtain geometrically consistent map.
<i>7</i>	Kinect Fusion (2011)	pixels có cường độ sáng intensities cao nhất	Ảnh RGB	RGB-D sensor	Thuật toán ICP, cực tiểu lỗi photometric	-
<i>8</i>	ORB-SLAM2 (2017)	points		Camera đơn, cam kép và RGB-D	Tracking thread- tìm các đặc trưng tương đồng và cực tiểu lỗi reprojection.	Local mapping thread
<i>9</i>	DSO- direct sparse odometry (2018)	các điểm có cường độ intensity cao nhất.	-	Stereo-cameras IMU	cực tiểu lỗi photometric	-
<i>10</i>	VIORB/ ORB-SLAM VI - Visual-Inertial Monocular ORB-SLAM (2017)	points	-	Camera đơn, cam kép (Stereo camera) IMU	Tracking thread- tìm các đặc trưng tương đồng và cực tiểu lỗi reprojection.	-
<i>11</i>	VINS Mono-Monocular Visual-Inertial System (2018)	Point và line hoặc plan	-	binocular và cam kép (stereo camera)	-	-

12	VI-DSO- Visual-Inertial Direct Sparse Odometry (2018)	Point	-	Cam đơn, cam kép (Stereo camera) IMU	-
13	ORB-SLAM3 (2020) = ORB-SLAM +VIO RB	points in the local window is over a threshold	-	Camera đơn, cam kép (stereo), và RGB-D cameras, Mono Inertial, Stereo Inertial	Tracking thread- tìm các đặc trưng tương đồng và cực tiểu lỗi reprojection.

STT	Phương pháp (năm) + Framework	Lỗi	Tập dữ liệu	Ưu	Nhược
1	MonoSLAM (2013) 	...	Rawseeds	Chạy với thời gian thực. Truy xuất quỹ đạo máy ảnh và xây dựng bản đồ điểm từ môi trường xung quanh 3D với qui mô vừa và nhỏ.	Vấn đề của phương pháp này là chi phí tính toán tăng tỷ lệ với kích thước của môi trường. Ở trong môi trường lớn, kích thước của một vectơ trạng thái trở thành lớn vì số điểm vi phạm lớn. Trường hợp này rất khó để đạt được tính toán thời gian thực.
2	PTAM (2007) <ol style="list-style-type: none"> 1. Khởi tạo bản đồ được thực hiện bởi thuật toán five-point. 2. Vị trí máy ảnh được ước tính từ các điểm đặc trưng phù hợp giữa các điểm trên bản đồ và hình ảnh đầu vào. 3. Vị trí 3D của các điểm đặc trưng được ước tính bằng phương pháp đo và vị trí 3D ước tính được áp dụng bởi BA 4. Quá trình theo dõi được phục hồi bằng randomized tree-based searching 	Để cải thiện độ chính xác của vSLAM, điều quan trọng là phải tăng số điểm đặc trưng trong bản đồ. Phương pháp dựa trên BA tốt hơn phương pháp dựa trên EKF vì nó có thể xử lý các điểm có số lượng lớn.	PTAM đã được sử dụng để theo dõi. Do đó, theo dõi là phương pháp dựa trên tính năng và không phải là phương pháp hoàn toàn chính xác
3	DTAM (2011) <ol style="list-style-type: none"> 1. Quá trình khởi tạo bản đồ được thực hiện bằng phép đo âm thanh nổi. 2. Chuyển động của máy ảnh được ước tính bằng cách tạo khung nhìn tổng hợp từ bản đồ được tái tạo. 3. Thông tin về độ sâu được ước tính cho mọi pixel bằng cách sử dụng âm thanh nổi đa cơ sở và sau đó, nó được tối ưu hóa bằng cách xem xét tính liên tục của không gian. 	Thuật toán DTAM được tối ưu hóa để đạt được xử lý thời gian thực trên điện thoại di động. Về cơ bản, các phương pháp này được thiết kế để tạo mô hình 3D trực tuyến và nhanh chóng.	NIL
4	LSD-SLAM (2014) <ol style="list-style-type: none"> 1. Các giá trị ngẫu nhiên được đặt làm giá trị độ sâu ban đầu cho mỗi pixel. 2. Chuyển động của máy ảnh được ước tính bằng cách tạo khung nhìn tổng hợp từ bản đồ được tái tạo. 3. Các khu vực được tái tạo bị giới hạn ở các khu vực có cường độ cao. 4. 7 DoF pose-graph optimization được sử dụng cho bản đồ nhất quán về mặt đo lường 	Lỗi phép tính tiến 1.14%.	KITTI	NIL	Không thể tái tạo lại cấu trúc cảnh ngay cả khi chuyển động của máy ảnh quay, kết quả của LSD-SLAM bị nhiễu đáng kể, vì đường cơ sở âm thanh nổi yêu cầu để ước tính độ sâu đối với hầu hết các khung hình là không đủ. Mức độ tiếng ồn cao. Độ chính xác thấp hơn PTAM và ORB
5	KinectFusion (2011) <ol style="list-style-type: none"> 1. Cấu trúc 3D của môi trường được tái tạo bằng cách kết hợp các bản đồ ảnh thu được trong không gian voxel 2. Chuyển động của máy ảnh được xác định bằng thuật toán ICP sử dụng cấu trúc 3D ước tính và bản đồ độ sâu đầu vào dựa trên độ sâu vSLAM 	Orig. 0.665 IMU 0.220 +noise 0.309 (RMSE ATE)	Freiburg	Có thể tái tạo môi trường trong những phòng có kích thước trung bình lớn.	Tích lũy lỗi drift vì không đóng loop.

6	ORB-SLAM2 (2017)		Lỗi phép tính tiến KITTI, EuroC Mức độ noise thấp.	Vấn đề duy nhất với ORB SLAM với máy ảnh một mắt có tốc độ chậm lúc khởi tạo và nó cũng mất điểm trong quá trình ánh xạ.
7	CNN-SLAM (2017)		Bản đồ độ sâu 66.18% ICL-NUIM, TUM	Có thể tái tạo lại cấu trúc cảnh ngay cả khi dịch chuyển camera hay độ dốc hình ảnh cao. giải quyết ước tính tỷ lệ tuyệt đối, thu được độ sâu dày đặc dọc theo các vùng không có kết cấu và xử lý các chuyển động quay. Dùng mô hình end-to-end vừa rút trích đặc trưng vừa học các điểm đặc trưng tương đồng.
8	DSO (2018)		Lỗi phép tính tiến KITTI, Cityscapes, EuRoC Monocular 0.84% 0.601 (RMS ATE)	Đóng loop hạn chế lỗi drift. Ngay cả khi không đóng loop, DSO cung cấp kết quả chính xác hơn ORB-SLAM2.
9	VIORB / ORBSLAM-VI (2017)		Monocular inertial 0.075 (RMS ATE) EuRoC	Mức độ noise thấp. Thời gian khởi tạo IMU lâu 10s-15s
10	VINS-Mono (2018)		Monocular inertial 0.110 (RMS ATE) EuRoC	Độ chính xác cao. Mức sử dụng bộ nhớ lớn dù chỉ xem pose và vận tốc từ trạng thái IMU..

11	VI-DSO (2018)		Monocular inertial 0.089 (RMS ATE)	EuRoC	Độ chính xác cao, mạnh mẽ hơn DSO.	Quá trình khởi tạo chậm vì khởi tạo dựa vào tinh chỉnh chiếu phối cảnh (bundle adjustment)
12	ORB-SLAM3		(2020) Stereo inertial 0.035 Monocular inertial 0.043 Stereo 0.084 Monocular 0.041 (RMS ATE)	EuRoC	Mức độ noise thấp. Thời gian khởi tạo IMU được cải thiện.	không thể tái tạo lại cấu trúc cảnh ngay cả khi chuyển động của máy ảnh quay, do thiếu đường cơ sở cần thiết để khởi tạo thuật toán.
13	MMS (2014)	<ol style="list-style-type: none"> App mobile để thu thập dữ liệu ảnh, GPS, cảm biến chuyển động. Khởi tạo giá trị IOP, EOP bằng cách dùng phép đo từ bộ thu GPS và cảm biến chuyển động. Từ video input, xếp chồng ảnh để tái tạo môi trường xung quanh với một tỉ lệ chồng chéo nhất định. Phát hiện điểm đặc trưng, đối sánh các điểm, phát hiện sai lầm và loại bỏ. Loại bỏ lỗi từ tham số khởi tạo ban đầu bằng đối sánh ảnh và ràng buộc hình học epipolar. Tinh chỉnh chiếu phối cảnh (bundle adjustment) để tính toán tọa độ bản đồ 3D của các điểm đặc trưng. 	Lỗi đo đặc sau khi scale 4.92%	-	Không thủ tục quét ảnh rườm rà, đơn giản hoá rút trích đặc trưng tự động, lưu trữ linh hoạt, tham chiếu trực tiếp tọa độ cảm biến	Cảm biến trên mobile có độ chính xác kém.

Bảng 2

		Deep learning	MMS (đề xuất)
Khác nhau	Rút trích đặc trưng và Học các đặc trưng tương đồng	Mô hình end-to-end vừa rút trích đặc trưng đồng thời học những đặc trưng tương đồng nên tốc độ sẽ nhanh hơn.	Công đoạn rút trích đặc trưng và học những đặc trưng tương đồng là 2 công đoạn tách biệt nên tốc độ sẽ chậm hơn

Bảng 3

2. Mình cần làm gì?
- Tìm hiểu nguyên lí của phương pháp đề xuất.
 - Tìm hiểu để chọn input đầu vào phù hợp với khả năng và thích hợp với phương pháp đề xuất.
 - Tìm hiểu các khái niệm đặc trưng để lựa chọn đặc trưng phù hợp.

- Tìm hiểu các nguyên lý, phương pháp tìm cặp điểm đặc trưng tương đồng trong các cặp khung hình.
- Thu thập dữ liệu hình ảnh từ camera điện thoại và các cảm biến chuyển động.
- Với việc tái tạo môi trường, ta sử dụng mạng DISN cho việc tái tạo môi trường 3D.
- Sau khi tái tạo được môi trường 3D, mobile robot tự định vị vị trí trong không gian thông qua việc so sánh các hình ảnh môi trường nhận được, sau khi qua các bước xử lý như lọc nhiễu, phát hiện các điểm đặc trưng, ... tiến hành so sánh với bản đồ gốc trong bộ nhớ và tính toán vị trí và góc hướng thực tế.

V. Nguyên lý - Phương pháp:

NGUYÊN LÝ

ĐỊNH VỊ:

Giả sử RGB-D camera lấy ảnh input loại RGB-D (có depth).

Dùng SIFT tìm những đặc trưng tương đồng.

Đưa các điểm đặc trưng tương đồng (x,y,z) vào $P'=AP$, giải ra được phép biến đổi A (có 6 ẩn số) bằng phương pháp Hồi quy tuyến tính hoặc phương pháp Ransac (chọn 2 phương pháp này vì thích hợp cho bài toán có số phương trình nhiều hơn số ẩn số).

TÁI TẠO MÔI TRƯỜNG XUNG QUANH:

Dựa trên hệ tọa độ camera tại thời điểm t , thiết bị capture dữ liệu môi trường xung quanh (đám mây điểm). Mỗi lần di chuyển sẽ phát sinh 1 đám mây điểm 3D.

Qui "tất cả đám mây điểm 3D tại mỗi hệ tọa độ camera rời rạc" về 1 hệ qui chiếu chung (dùng registration).

Kết quả nhận được 1 đám mây điểm chung so với hệ qui chiếu ban đầu.

PHƯƠNG PHÁP

1. Một số khái niệm cơ bản

- Đặc trưng: Có hai loại đặc trưng đã học gồm đặc trưng góc và blob (hình dạng tròn, elip). Đặc trưng blob bất biến đối với phép biến đổi Scale.
- Trích xuất đặc trưng

- Phát hiện biên cạnh

- Một số khái niệm cơ bản





Biên cạnh được hiểu là tại mỗi điểm pixel có **biến thiên độ xám** (ví dụ tại đó giá trị ≈ 255) thì ta sẽ **vẽ biên cạnh**. Ngược lại không có biến thiên độ xám thì ta không vẽ biên cạnh.

Vẽ biên cạnh là vẽ biên độ của **vector** pháp tuyến gradient tại mỗi vị trí đó (vector là gồm 2 thành phần x và y). Nhưng khi lưu thì người ta thường lưu vector tiếp tuyến (edge direction) hơn

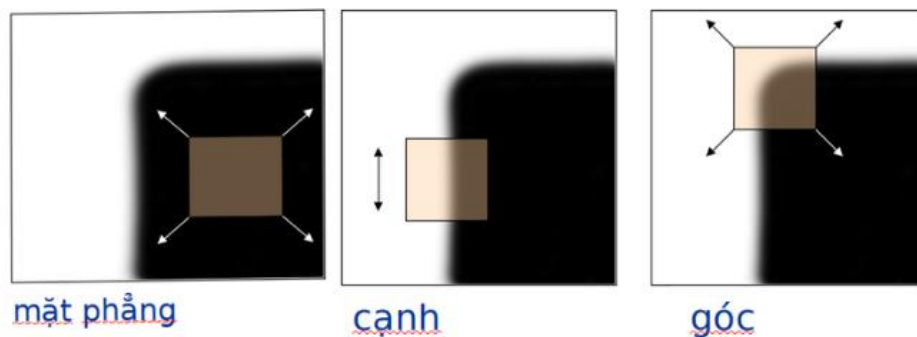
Ảnh sau khi vẽ biên cạnh là ảnh phân ngưỡng (trắng đen, chỉ giữ lại 0 với 255). Số nét biên cạnh nhiều hay ít phụ thuộc vào ngưỡng nhỏ hay lớn (dùng đạo hàm bậc 1).

Đối với miền biến thiên độ xám trải rộng như ảnh y khoa, người ta dùng đạo hàm bậc 2 thay vì chỉ đạo hàm bậc 1. Đạo hàm bậc 2 không phụ thuộc vào **biên độ**/chiều dài vì nó chỉ chú ý tới sự chuyển dấu + sang – hoặc – sang +.

Mục tiêu: Rời rạc hoá bằng cách khảo sát sơ đồ lưới biến thiên toạ độ nguyên để xem hình dạng vì mỗi hình dạng ứng với một edge nhất định.

Differencing $h=b-a$  <pre> a a a a a b b b b b a a a a a b b b b b a a a a a b b b b b a a a a a b b b b b a a a a a b b b b b Vertical step edge 0 0 0 0 h 0 0 0 0 </pre>	Differencing $h=b-a$  <pre> a a a a c b b b b b a a a a c b b b b b a a a a c b b b b b a a a a c b b b b b a a a a c b b b b b Vertical ramp edge 0 0 0 0 h/2 h/2 0 0 0 </pre>	Differencing (localize edge center of ramp edge) $h=b-a$  <pre> a a a a c b b b b b a a a a c b b b b b a a a a c b b b b b a a a a c b b b b b a a a a c b b b b b Vertical ramp edge 0 0 h/2 h h/2 0 0 </pre>
Prewitt Sobel Frei-Chen $h=b-a$  <pre> a a a a c b b b b b a a a a c b b b b b a a a a c b b b b b a a a a c b b b b b a a a a c b b b b b Vertical ramp edge 0 0 h/2 h h/2 0 0 </pre>	Prewitt Sobel Frei-Chen $h=b-a$ Diagonal ramp edge $0 \quad \frac{h}{\sqrt{2}(2+k)} \quad \frac{h}{\sqrt{2}} \quad \frac{\sqrt{2}(1+k)h}{(2+k)} \quad \frac{h}{\sqrt{2}} \quad \frac{h}{\sqrt{2}(2+k)} \quad 0$	

- Các thuật toán như Gradient, Laplace,...
- Phát hiện góc
 - Các thuật toán như Harris corner detection.



Để hiểu hơn về góc trong ảnh, ta thấy rằng đối với vùng mặt phẳng khi ta di chuyển mặt nạ bên trong vùng mặt phẳng sẽ không có bất cứ thay đổi nào về cường độ của các pixel. Đối với vùng cạnh khi ta di chuyển mặt nạ theo chiều của mép cạnh cũng sẽ không có bất cứ sự thay đổi nào về cường độ. Nhưng với góc, khi ta di chuyển mặt nạ theo bất cứ hướng nào thì cường độ đều thay đổi.

Do vùng góc di chuyển của nó theo hướng nào thì cũng có sự thay đổi về cường độ nên để phát hiện góc ta sử dụng công thức :

$$E(u,v) = \sum_{x,y} w(x,y) [I(x+u,y+v) - I(x,y)]^2$$

Trong đó :

- $w(x, y)$: cửa sổ trượt tại tọa độ (x, y)
- $I(x + u, y + v)$: cường độ tại tọa độ đã dịch chuyển một khoảng (u, v)
- $I(x, y)$: cường độ tại tọa độ điểm hiện tại
- $E(u, v)$: sự thay đổi cường độ với cửa sổ (x, y) so với cường độ tại $(x + u, y + v)$

Với công thức trên ta sẽ có một biểu thức tương đương như sau:

$$E(u, v) \approx [u \ v] M \begin{bmatrix} u \\ v \end{bmatrix}$$

Trong đó: I_x và I_y là đạo hàm theo hướng x và y tương ứng của ảnh ta sẽ có biểu thức của M như sau:

$$M = \sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

Với công thức trên ta có thể tính được sự thay đổi về cường độ, từ đó có thể chọn ra vùng có khả năng là góc. Sau khi ước lượng được các vùng là góc, ta cần xác định được rằng vùng đó có thật sự là góc hay không với thuật toán Harris Conner Detection. Thuật toán Harris Conner Detection sử dụng một confidence score để đánh giá với biểu thức như sau:

$$M_c = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2 = \det(A) - k(\text{trace}^2(A))$$

Trong đó λ_1 , λ_2 lần lượt là giá trị eigen của ma trận M

Nếu giá trị của λ_1 với λ_2 đều nhỏ, tức là sự tác động của λ_1 và λ_2 tới biểu thức M không nhiều, nên M gần như không có sự thay đổi theo bất kì hướng nào. Khi đó vùng ảnh đang nằm trong cửa sổ không phải là điểm góc.

Nếu giá trị λ_1 lớn và λ_2 nhỏ hoặc ngược lại, tức là biểu thức M có sự thay đổi nếu trượt cửa sổ theo một hướng nào đó, khi đó nếu ta trượt theo hướng trục giao thì M không thay đổi đáng kể, lúc này vùng ta đang xét không phải là điểm góc mà là vùng cạnh

Nếu giá trị λ_1 và λ_2 đều lớn, tức là nếu ta trượt cửa sổ theo bất kì hướng nào thì biểu thức M đều thay đổi giá trị đáng kể về cường độ xám. Như vậy ta sẽ xác định được đây là điểm góc.

- Phát hiện đặc trưng giữa 2 khung hình

- Các thuật toán như SIFT, SURF, ...

SIFT

Khái niệm: Thuật toán đi tìm đặc trưng blob tương đồng trong giữa các khung hình và các đặc trưng này bất biến với phép. Kết thúc thuật toán, ta nhận được các cặp điểm tương đồng mỗi cặp khung hình.

Nguyên lí: Dùng đạo hàm bậc 2 của hàm Gauss như là một mask/filter/ma-trận-kết-cấu áp lên từng pixel của ảnh, điểm nào thỏa tính chất cực trị (lớn hơn hẳn 9 pixel ở tầng sigma trên, 9 pixel tầng sigma dưới và 8 pixel láng giềng) sẽ được cho là blob. Chọn lọc lại blob thỏa điều kiện “blob nào xung quanh có 1 corner thì đó là điểm ổn định” và lấy descriptor (histogram của vector tiếp tuyến tại vùng đó).

Mục tiêu: Chú trọng giải quyết 2 vấn đề của thuật toán Blob:

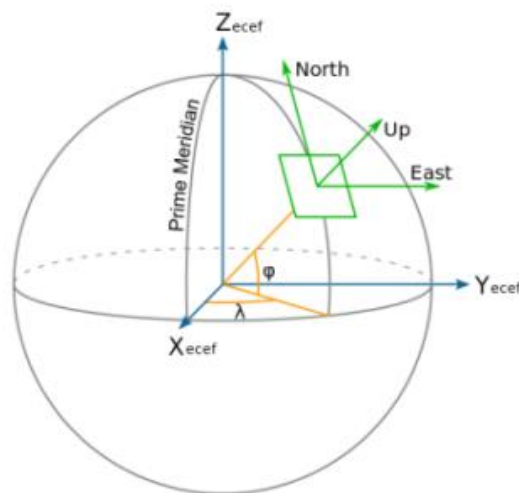
Một, điểm không có hình dáng nhưng thỏa tính chất cực trị (blob giả).

Hai, phụ thuộc ngưỡng của bộ lọc.

Điều kiện ràng buộc: Ứng viên Blob có độ tin cậy cao phải lớn hơn 10% DOG hoặc nhỏ hơn 10% DOG. Điểm Blob lớn đáng kể sẽ có độ tương phản lớn hơn 0.03, nên những điểm không có hình dáng nhưng thỏa tính chất cực trị và có độ tương phản bé hơn 0.03 sẽ bị loại bỏ. Sự tương phản được thể hiện bằng độ lệch màu của ứng viên Blob đó với những điểm lân cận.

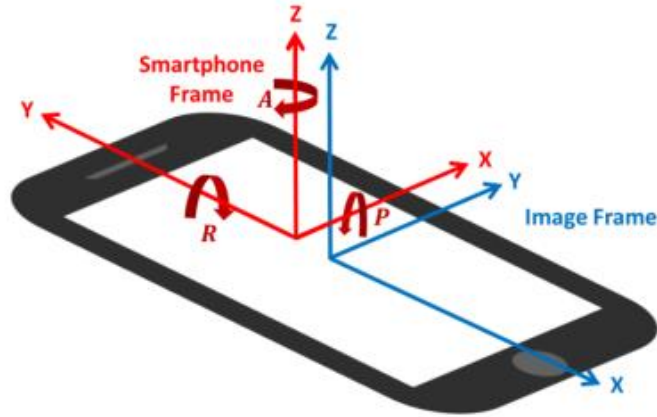
c. Khung tọa độ

- Các hệ tọa độ ánh xạ khác nhau có thể được sử dụng để thu được các giải pháp ánh xạ cuối cùng của các hệ thống ánh xạ. Earth Centered Earth Fixed (ECEF) và Local Level Frame (ENU) là hai ví dụ về các khung lập bản đồ thường được sử dụng.
- Trong phần này, một khung ánh xạ tọa độ Local Level Frame sẽ được sử dụng. Ma trận quay giữa các khung tọa độ khác nhau rất quan trọng để chuyển vector điểm quan sát từ khung tọa độ ảnh sang khung ánh xạ mong muốn. Phần này giới thiệu các khung được sử dụng trong hệ thống đã phát triển và các ma trận xoay khác nhau giữa chúng.



- Khung tọa độ của ảnh và của điện thoại
 - Hệ tọa độ này được xác định bởi Android và giống nhau đối với tất cả các thiết bị hỗ trợ Android. Phương trình dưới là ma trận xoay từ hình ảnh đến các khung tọa độ của điện thoại. Hình phía dưới cho thấy hình ảnh và các hệ thống tọa độ của điện thoại thông minh và hệ tọa độ của ảnh. [13]

$$R_I^{SP} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



Trong đó:

x, y, z là các trục tọa độ

R, P, A lần lượt là Roll, Pitch, Azimuth rotation angles

- Khung tọa độ cấp local (ENU)
 - Các cảm biến định hướng trong thiết bị hỗ trợ Android sử dụng khung này để cung cấp góc xoay cao độ, góc cuộn và góc phương vị của thiết bị.
 - Chúng ta có thể sử dụng ma trận sau để chuyển đổi hệ tọa độ từ điện thoại sang ENU:

$$R_{SP}^{ENU} = \begin{bmatrix} \cos(R) \cos(-a) & -\cos(r) \sin(-a) & \sin(r) \\ \cos(-p) \sin(-a) + \sin(-p) \sin(r) \cos(-a) & \cos(-p) \cos(-a) - \sin(-p) \sin(r) \sin(-a) & -\sin(-p) \cos(r) \\ \sin(-p) \sin(-a) - \cos(-p) \sin(r) \cos(-a) & \sin(-p) \cos(-a) + \cos(-p) \sin(r) \sin(-a) & \cos(-p) \cos(r) \end{bmatrix}$$

- Ma trận xoay giữa hệ tọa độ ảnh và ENU:

$$R_{Image}^{ENU} = R_{SP}^{ENU} R_{Image}^{SP}$$

d. Hiệu chỉnh máy ảnh

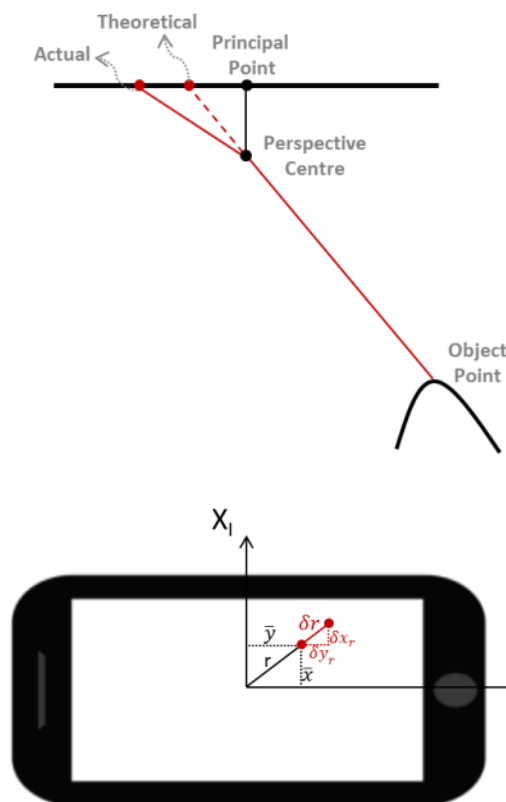
- Chúng ta gặp một khó khăn khi sử dụng hệ thống camera của điện thoại là tiêu cự, các thông số biến dạng và principal point offset [20] là các thông số không biết trước và thay đổi trên mỗi thiết bị. Nói chung, mối quan hệ giữa độ dài tiêu cự và IOP của máy ảnh được sử dụng có thể được mô tả bằng một mô hình nhất quán.

- Tính năng zoom trên camera cũng là một tính năng quan trọng nhưng nó sẽ làm méo ảnh và thay đổi tiêu cự nên chúng ta cũng sẽ phải khoá nó lại.
- Chúng ta có 8 thông số được xem là quan trọng trong hiệu chỉnh camera là: tiêu cự của camera, principal point offset (x_p, y_p) , 3 tham số biểu thị độ méo xuyên tâm (k_1, k_2, k_3) và 2 thông số biến dạng $(p_1$ và $p_2)$. Nguồn méo chính của máy ảnh là méo xuyên tâm δr , thường được biểu thị bằng một đa thức bậc lẻ (công thức ở dưới). Hầu hết lỗi méo xuyên tâm được tính bằng cách tính số hạng $k_1 r^3$

$$\delta r = k_1 r^3 + k_2 r^5 + k_3 r^7$$

Trong đó r là khoảng cách xuyên tâm từ điểm chính

k_1, k_2, k_3 là các hệ số méo xuyên tâm khác nhau.



- Ta có công thức để tính những độ méo dựa trên 2 trục x và y là

$$\delta x_r = \delta r \frac{\bar{x}}{r}$$

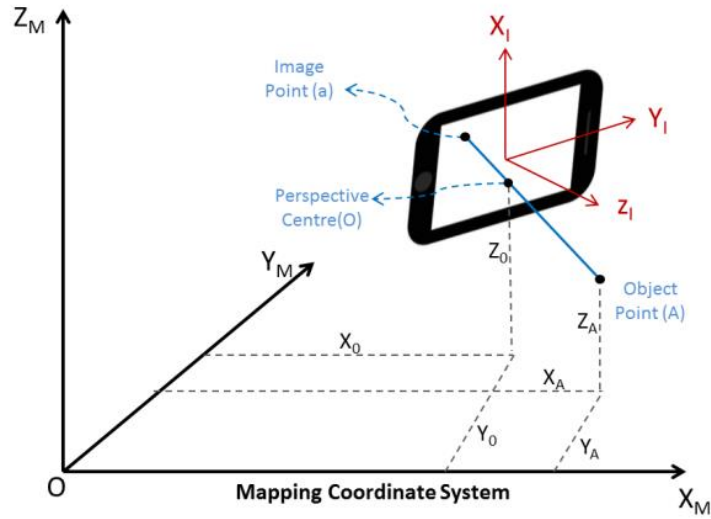
$$\delta y_r = \delta r \frac{\bar{y}}{r}$$

Trong đó \bar{x} và \bar{y} là khoảng cách từ điểm chính theo hướng x và y.

e. Extended Kalman Filter (EKF)

- Là heuristic cho bộ lọc phi tuyến.

- Thường hoạt động rất tốt nếu nó được điều chỉnh đúng cách.
- Được dùng rộng rãi trong thực tế.
- Cơ sở:
 - Tuyến tính động và các hàm ước lượng trả về tức thì.
 - Ước tính gần đúng của kì vọng có điều kiện và ma trận hiệp phương sai
- Các bước tính và ví dụ có thể tham khảo tại slide của stanford [30]
- f. Phương trình cộng tuyến (Extended Collinearity Equations)
 - Phương trình thẳng hàng bao gồm mô hình toán học đại diện cho mối quan hệ chung giữa hình ảnh và các hệ tọa độ thực tế. Các điểm thẳng hàng là các đối tượng trong thực tế, điểm ảnh và quang tâm.



- Như có thể nhận thấy từ hình trên, cả hai vector \overrightarrow{OA} và \overrightarrow{Oa} đều thẳng hàng. Dựa trên điều này, chúng tôi có thể nói:

$$\overrightarrow{Oa} = \mu_{ENU}^{Image} R_{ENU}^{Image} \overrightarrow{OA} \quad (1.e.1)$$

Trong đó:

\overrightarrow{OA} là vector nối từ perspective centre tới object point

\overrightarrow{Oa} là vector nối từ perspective centre tới image point

μ_{ENU}^{Image} là hệ số tỉ lệ giữa hệ tọa độ thực tế và hệ tọa độ ảnh

R_{ENU}^{Image} là ma trận xoay giữa hệ tọa độ thực tế và hệ tọa độ ảnh.

- $\overrightarrow{OA}, \overrightarrow{Oa}$ có thể được khai triển như sau:

$$\overrightarrow{Oa} = \begin{bmatrix} x_a - x_p \\ y_a - y_p \\ -C \end{bmatrix} \quad (1.e.2)$$

(1.e.3)

$$\overrightarrow{OA} = \begin{bmatrix} X_A - X_0 \\ Y_A - Y_0 \\ Z_A - Z_0 \end{bmatrix}$$

Trong đó:

- C là khoảng cách tiêu cự máy ảnh
- x_p, y_p là tọa độ của điểm chính trong không gian ảnh
- x_a, y_a là tọa độ của ảnh của vật trong không gian
- X_0, Y_0, Z_0 là tọa độ của quang tâm trong thế giới thực
- X_A, Y_A, Z_A là tọa độ của đối tượng trong thế giới thực

- Thay (1.e.2) và (1.e.3) vào (1.e.1) ta được

$$x_a = x_p - c \frac{r_{11}(X_A - X_0) + r_{12}(Y_A - Y_0) + r_{13}(Z_A - Z_0)}{r_{31}(X_A - X_0) + r_{32}(Y_A - Y_0) + r_{33}(Z_A - Z_0)} \quad (1.e.4)$$

$$y_a = y_p - c \frac{r_{21}(X_A - X_0) + r_{22}(Y_A - Y_0) + r_{23}(Z_A - Z_0)}{r_{31}(X_A - X_0) + r_{32}(Y_A - Y_0) + r_{33}(Z_A - Z_0)} \quad (1.e.5)$$

Trong đó r_{ij} là phần tử hàng thứ i và cột thứ j của ma trận xoay R_{ENU}^{Image} .

- Xét thêm độ méo do camera gây ra, phương trình cộng tuyến lúc này xuất hiện thêm độ lỗi trên trục x và y của mặt phẳng ảnh:

$$x_a + \delta x_r = x_p - c \frac{r_{11}(X_A - X_0) + r_{12}(Y_A - Y_0) + r_{13}(Z_A - Z_0)}{r_{31}(X_A - X_0) + r_{32}(Y_A - Y_0) + r_{33}(Z_A - Z_0)} \quad (1.e.6)$$

$$y_a + \delta y_r = y_p - c \frac{r_{21}(X_A - X_0) + r_{22}(Y_A - Y_0) + r_{23}(Z_A - Z_0)}{r_{31}(X_A - X_0) + r_{32}(Y_A - Y_0) + r_{33}(Z_A - Z_0)} \quad (1.e.7)$$

g. Bundle Adjustment

- Bundle adjustment là một ước tính bình phương tối thiểu phụ thuộc vào các phương trình thẳng hàng để tính toán hàm chi phí cho quá trình ánh xạ.
- Sử dụng Bundle adjustment, tọa độ hình ảnh của các điểm đang được chú ý tới, tọa độ thực tế tương ứng của chúng, IOP của máy ảnh và EOP của hình ảnh đã chụp có thể liên quan với nhau.
- Điều chỉnh bình phương tối thiểu được sử dụng để tính toán vectơ trạng thái chưa biết từ một tập hợp các quan sát khi số lượng quan sát lớn hơn số ẩn số bằng cách tối thiểu hóa một hàm chi phí nhất định.
- Trong bình phương tối thiểu, hàm chi phí là hiệu số giữa các quan sát đo được và các quan sát được tính toán. Phương trình (1.f.1) cho thấy phương trình mô hình quan sát Gauss Markov cho điều chỉnh bình phương tối thiểu.

$$(1.f.1)$$

$$z = Ax + e$$

Trong đó:

z là vector quan sát

x là vector chưa biết

A là ma trận biểu diễn mối tương quan giữa vector ẩn và vector quan sát

e là độ lỗi

- Công thức tính vector ẩn:

$$\hat{x} = (A^T R^{-1} A)^{-1} A^T R^{-1} z \quad (1.f.2)$$

Trong đó \hat{x} là giá trị kỳ vọng của vector ẩn số và R là ma trận hiệp phương sai của vector quan sát

- Đối với mô hình phi tuyến tính, chuỗi Taylor được sử dụng để tuyến tính hóa trong đó các số hạng cao hơn của nó bị loại bỏ để đơn giản hóa. Kết quả ước lượng phi tuyến tính có thể được biểu thị bằng công thức:

$$\delta \hat{x} = (A^T R^{-1} A)^{-1} A^T R^{-1} \delta z \quad (1.f.3)$$

Trong đó:

$\delta \hat{x}$ là các hiệu chỉnh đối với vector ẩn số mong đợi

A là đạo hàm riêng của ma trận thiết kế đối với các ẩn số khác nhau

δz là hiệu số giữa các quan sát đo được và các quan sát dự kiến sử dụng vector chưa biết mong đợi

- δz được tính bằng

$$\delta z = \begin{bmatrix} x_a - x'_a \\ y_a - y'_a \end{bmatrix} \quad (1.f.4)$$

Trong đó:

x_a, y_a là tọa độ điểm ảnh

x'_a, y'_a là tọa độ được tính từ phương trình cộng tuyến

- Các giá trị chưa biết ban đầu tốt là rất quan trọng trong việc mở rộng chuỗi Taylor. Các giá trị ban đầu không hợp lệ có thể khiến kết quả cuối cùng khác nhau. Ngoài ra, việc lặp lại quá trình tìm lời giải giúp thu được kết quả có độ chính xác cao hơn. Sau mỗi lần lặp, các giá trị của vector chưa biết được cập nhật bằng công thức:

$$\hat{x} = \hat{x}_{n-1} + \delta \hat{x}_n \quad (1.f.5)$$

Trong đó: \hat{x}_n, \hat{x}_{n-1} lần lượt là vector ẩn tại vòng lặp thứ n và $n - 1$.

- Trong Bundle adjustment, số hàng và số cột của ma trận đại diện cho số biến quan sát và biến ẩn số, tương ứng. Mỗi cột trong ma trận là đạo hàm riêng của các phương trình thẳng hàng

đối với một ẩn số nhất định. Ma trận phương sai / hiệp phương sai của vector ẩn số là một ma trận đối xứng mà từ đó có thể thu được một số giá trị. Giả sử rằng các ẩn số của phương pháp là IOP của máy ảnh, EOP của mỗi hình ảnh và tọa độ thực tế của các điểm đang được chú ý, hình dạng của ma trận phương sai / hiệp phương sai có dạng:

	EOPs	IOPs	TPs
EOPs	P_{11}	P_{12}	P_{13}
IOPs	P_{21}	P_{22}	P_{23}
TPs	P_{31}	P_{32}	P_{33}

Giải thích:

P_{11} số lượng image

P_{22} số lượng camera

P_{33} số lượng điểm chú ý

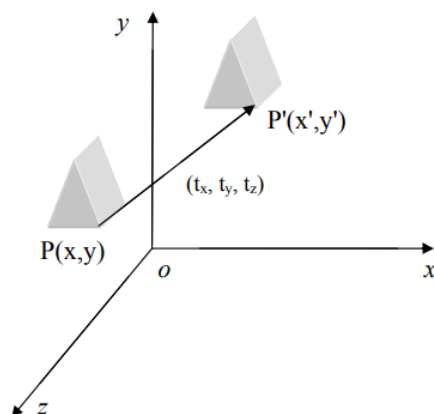
P_{12}, P_{21} camera được sử dụng để chụp một hình ảnh nhất định

P_{13}, P_{31} khả năng hiển thị của các điểm buộc trong hình ảnh

P_{23}, P_{32} Khả năng hiển thị của các điểm ràng buộc trong hình ảnh được chụp bởi một máy ảnh nhất định

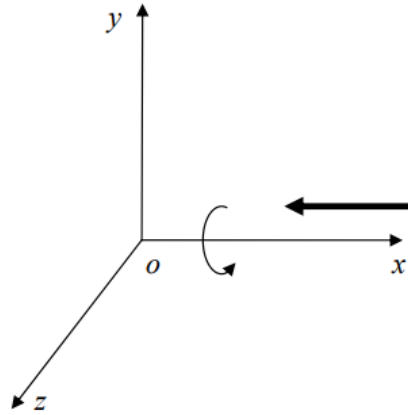
h. Các phép biến đổi trong không gian 3 chiều

- Phép tịnh tiến



$$P' = AP = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

- Phép xoay từng trục



- Quanh Ox

$$P' = AP = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

- Quanh Oy

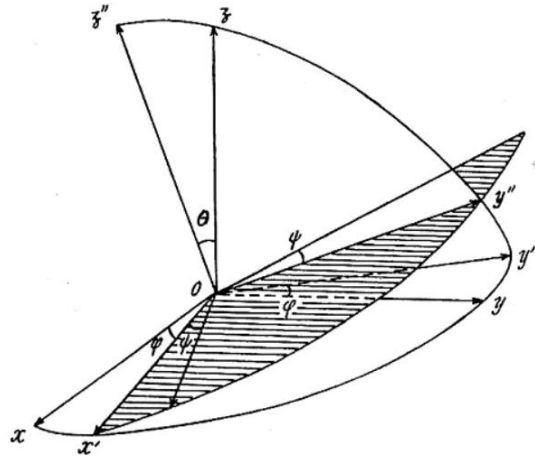
$$P' = AP = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

- Quanh Oz

$$P' = AP = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Với θ là góc xoay

- Phép xoay tổng quát cả 3 trục:



$$\begin{pmatrix} \cos\psi \cos\varphi - \sin\psi \cos\theta \sin\varphi & -\cos\psi \sin\varphi - \sin\psi \cos\theta \cos\varphi & \sin\psi \sin\theta \\ \sin\psi \cos\varphi + \cos\psi \cos\theta \sin\varphi & -\sin\psi \sin\varphi + \cos\psi \cos\theta \cos\varphi & -\cos\psi \sin\theta \\ \sin\theta \sin\varphi & \sin\theta \cos\varphi & \cos\theta \end{pmatrix}$$

Với:

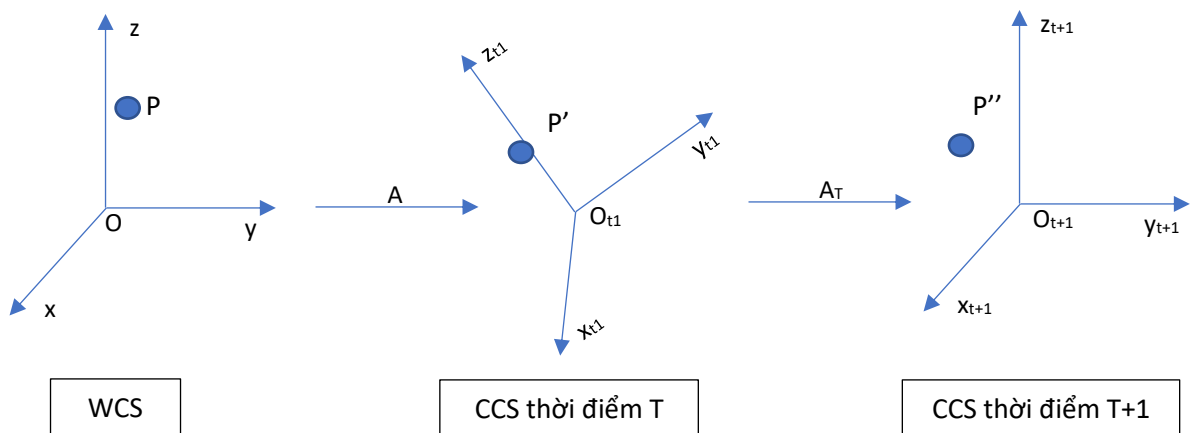
φ là góc quay trục Oz

ψ là góc quay trục Oz

θ là góc quay trục Oy

2. So sánh các đặc điểm

a. Framework



- T: Phép biến đổi tọa độ
- WCS: World coordinate system
- CCS: Camera coordinate system

- Ta có điểm $P' = A.P$, $P = (x, y, z, 1)$, $P' = (x', y', z', 1)$

Trong đó:

P và P' là điểm trong hệ tọa độ thực tế và tọa độ trong hệ tọa độ camera

A là ma trận biến đổi P thành P' và ngược lại

- Vậy công việc chúng ta cần làm là tìm được ma trận biến đổi A . A sẽ là phương trình gồm 6 ẩn tự do, 1 của phép xoay $R(x_R, y_R, z_R)$ và 1 của phép tịnh tiến $T(x_T, y_T, z_T)$.

- Công thức biến đổi tổng quát
$$\begin{cases} x' = a_1x + b_1y + c_1z + m \\ y' = a_2x + b_2y + c_2z + n \\ z' = a_3x + b_3y + c_3z + p \end{cases}$$

Với a_i, b_i, c_i, m, n, p là hằng số

- Ta có ma trận biến đổi $A = \begin{bmatrix} a_1 & a_2 & a_3 & 0 \\ b_1 & b_2 & b_3 & 0 \\ c_1 & c_2 & c_3 & 0 \\ m & n & p & 1 \end{bmatrix}$

b. Tìm các điểm đặc trưng trong từng ảnh

- Ở bước này công việc của chúng ta cần làm là tìm các điểm đặc trưng trong mỗi tấm ảnh 2D.
- Các đặc trưng của ảnh có thể là các góc, blob,... và các phương pháp này đã được nhóm giới thiệu ở mục V.1.a phía trên
- Để tìm các đặc trưng của một bức ảnh ta có thể sử dụng một trong các thuật toán SIFT, SURF,... và các thuật toán này cũng đã được nhóm giới thiệu ở phần phía trên.



Phát hiện điểm đặc trưng bằng SURF

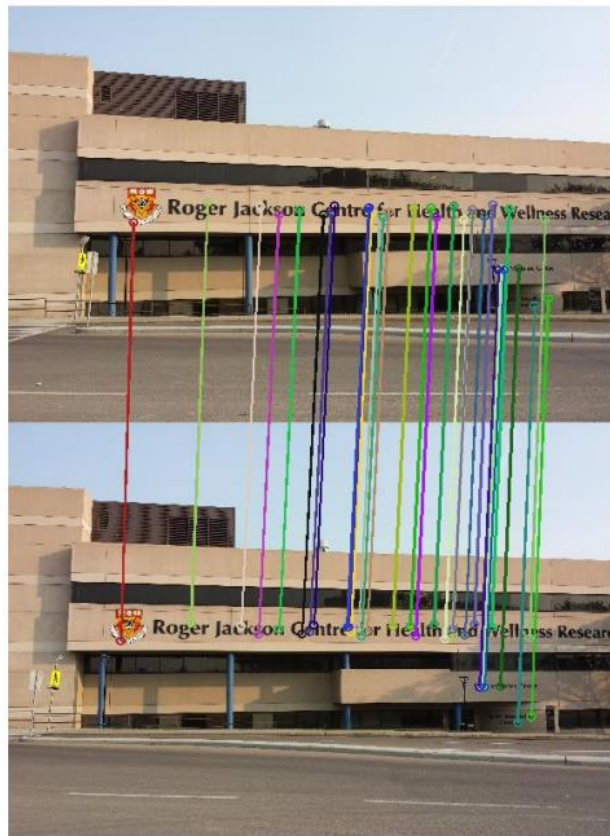
- Ví dụ như tấm ảnh phía trên chúng ta có thể thấy những điểm khoanh tròn, đây chính là các điểm đặc trưng của tấm ảnh này.
- c. Tìm các điểm đặc trưng tương đồng của 2 ảnh liên quan nhau
- Sau khi tìm được đặc trưng của từng ảnh, chúng ta bắt đầu so khớp các đặc trưng của 2 ảnh với nhau, tìm ra những đặc trưng chung giữa 2 ảnh. Ở bước này sẽ dùng thuật toán KNN

search để tiến hành so khớp. Ở trong bài báo cáo của tác giả Ebrahim Karami và các cộng sự [21] có đề cập tới vấn đề này một cách tổng quát, mọi người có thể tham khảo thêm. Chúng ta cũng có thể áp dụng những hàm đã xây dựng sẵn của thư viện OpenCV [22] để triển khai.

- Vậy tại sao lại phải sử dụng SIFT hay SURF để so khớp các đặc điểm tương đồng? Bởi vì chúng ta cần tìm nhiều hơn 3 cặp điểm tương đồng để giải hệ phương trình 6 ẩn A (chứa phép quay và tịnh tiến ma trận).
- Một điều đáng lưu ý như tiêu đề của mục này đã ghi là 2 ảnh phải liên quan nhau. Tức là chúng phải có mối quan hệ về cả mặt thời gian và không gian.
- Sau bước này chúng ta sẽ thu được vector các cặp điểm tương đồng

$$M = \begin{bmatrix} P_1 & P'_1 \\ \vdots & \vdots \\ P_n & P'_n \end{bmatrix}$$

- Dù chỉ cần $n=3$ chúng ta có thể giải hệ phương trình 6 ẩn tuy nhiên nó sẽ gây ra sai số rất lớn. Với n lớn chúng ta sẽ áp dụng phương pháp hồi quy tuyến tính để tính chính xác hơn.

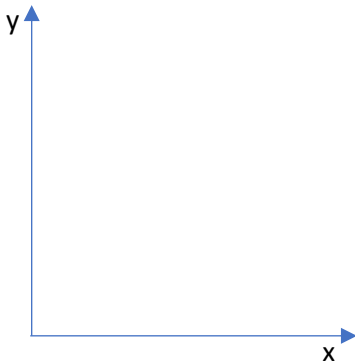


So khớp đặc trưng giữa 2 ảnh bằng SURF

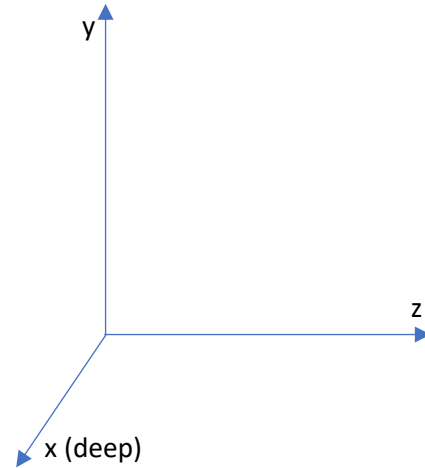
d. Chuyển hoá 3 chiều

- Chúng ta cần có một hệ toạ độ 3 chiều làm chuẩn để dễ thao tác.

$$\begin{cases} x: \text{độ sâu, hướng trước mặt} \\ y: \text{độ cao, hướng lên trên} \\ z: \text{rộng, hướng sang bên phải} \end{cases}$$



màn hình



thực tế

- Giả sử kích thước ảnh là $m \times n$ pixel. (u, v) là tọa độ điểm. Chúng ta có

$$P(x, y, z) \begin{cases} x = \text{độ sâu} \\ y = -\left(v - \frac{n}{2}\right) * \frac{\text{độ sâu}}{\text{tiêu cự}} \\ z = \left(u - \frac{m}{2}\right) * \frac{\text{độ sâu}}{\text{tiêu cự}} \end{cases}$$

- Tiếp theo là tìm một ma trận biến đổi A. Dùng bình phương tối thiểu có thể giải quyết vấn đề này. Chúng ta có thể dùng RANSAC hoặc mô hình đường hồi quy tuyến tính để giải hệ phương trình này.
- Áp dụng hồi quy tuyến tính vào phương trình $P' = AP$ và phần các phép biến đổi trong không gian 3 chiều (1.h) chúng ta có thể tìm được mối quan hệ giữa P' và P (tức là A)
- Tham khảo thêm về phương pháp đường hồi quy tuyến tính của nhóm ở môn PTTKDLNB [35].
- Sau khi tìm được A chúng ta có thể chuyển hệ tọa độ của tất cả các camera về cùng một hệ tọa độ đồng nhất.

3. Tạo bản đồ và tự định vị

Việc định vị và tái tạo môi trường là 2 bước thực hiện liên tiếp nhau. Ở thời điểm đầu tiên khi robot được kích hoạt, đây là vị trí T_0 và hệ tọa độ thực tế. Với mỗi tấm ảnh mới mà robot chụp chúng ta đã có một hệ tọa độ camera mới. Từ đây, chúng ta thấy rằng với mỗi lần camera chụp việc chúng ta cần làm là xác định vị trí của robot rồi mới tiếp tục tái tạo môi trường quanh robot. Cụ thể sẽ được trình bày ở dưới

a. Xác định tư thế

- Sau mỗi lần thay đổi hệ toạ độ việc chúng ta cần làm là tìm vị trí mới của hệ xem nó ở đâu so với hệ cũ. Xét một dãy ảnh hữu hạn N ảnh, đặt P_k là vị trí tại hạng $k \in [0, N]$, ma trận A của chúng ta có thể được biểu diễn bằng ma trận 4×4 với R có kích thước 3×3 và T là một vector.

$$P_k = \begin{bmatrix} & R & \begin{matrix} t_x \\ t_y \\ t_z \end{matrix} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.a.1)$$

- Giả sử chúng ta có ma trận xoay là R_0 và vector tịnh tiến là $t_0 = (x_0, y_0, z_0)^T$. Ma trận (4.a.1) trở thành:

$$P_0 = \begin{bmatrix} & R_0 & \begin{matrix} x_0 \\ y_0 \\ z_0 \end{matrix} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Chúng ta có thể chọn luôn vị trí P_0 làm gốc toạ độ cho dễ tính, lúc này $P_0 = I_4$
- Từ công thức $P' = AP$ với nhiều giá trị P từ 0 tới k, ta luôn có một phép biến đổi P thành P'. Thừa kế những tính chất của P_k phía trên, A ở dưới này cũng có thể biểu diễn bằng ma trận 4×4 . Nếu P_0 xác định vị trí đầu tiên, thì chúng ta có thể suy ra P_i bằng cách:

$$P_k = \prod_{i=0}^{k-1} A_{i+1} P_0 \quad (4.a.2)$$

b. Khởi tạo mạng

- Ngắn gọn mà nói thì bước này là bước giảm dữ liệu (các dáng ở phần trên) đầu vào. Lý do là với FPS cao các tấm ảnh được đưa vào liên tục, nhiều tấm có những dáng không khác gì nhiều so với tấm ảnh trước nó. Điều này làm giảm hiệu suất của hệ thống. Vậy giải pháp là cắt giảm hết nhưng dáng trùng nhau, chỉ xử lý những tấm ảnh có dáng khác với tấm ảnh trước nó những tấm ảnh này có tên là keyposes.
- Chúng ta có 2 cách để giải quyết vấn đề này. Một là chỉ lấy ảnh mới sau khi thiết bị đã di chuyển một ngưỡng nhất định. Hai là dựa vào các điểm đặc trưng, hệ thống sẽ cập nhật khi mà số điểm đặc trưng của những tấm hình trước đó với những tấm hình mà thiết bị thu được ít hơn một ngưỡng nào đó.
 - Ví dụ cho cách 1, thiết bị tự hành sẽ tự di chuyển và cứ sau khi đi được 0.1m hoặc quay 10 độ thì thiết bị sẽ lại cập nhật dữ liệu mới một lần.
 - Để hình dung cách 2 dễ hiểu, giả sử ảnh cái tủ lạnh có 10 đặc trưng, nhưng sau khi camera quay một góc alpha nào đó, mà ảnh chỉ còn thu được một nửa cái tủ lạnh với 5 điểm đặc trưng, nếu ngưỡng của chúng ta đặt là 6 thì ảnh sau đã thấp hơn ngưỡng -> tiến hành xử lý ảnh mới. Còn giả sử ngưỡng là 4 thì ảnh vẫn còn cao hơn ngưỡng -> không cần xử lý ảnh mới.
- Sau khi lấy được những keyposes cần thiết, mỗi keyposes đấy sẽ đóng góp vào mạng một hoặc nhiều những node (đặc trưng). Keypose mới sẽ được kết nối với những keyposes cũ bằng những đường cạnh.

c. Loop closures

- Ở phần khởi tạo mạng, có nhắc tới việc các node của keypose mới sẽ liên kết với keyposes cũ. Nhưng vấn đề là có rất nhiều node trong hệ thống của chúng ta, vậy thì những node mới sẽ kết nối với những node cũ như thế nào.
- Cách đơn giản nhất là vét cạn, chúng ta sẽ so sánh khung hình đang xét với toàn bộ những khung hình trước đó. Tuy nhiên cách này lại tốn nhiều thời gian và chi phí. Để cải tiến cách này, người ta đã sử dụng RGB histogram để lọc bước đầu trước. Những hình không liên quan sẽ bị loại bỏ trước khi chúng ta vét cạn.
- Giả sử có một điểm P_k và có các điểm $C1$ $C2$ $C3$ với các C và P_k là những điểm gần nhau. Lúc này chúng ta cần tìm ra mức độ tương đồng giữa các hình trong C và P_k . Giữa P_k với $C1$, $C2$, $C3$ lần lượt là 55%, 53% và 90%. Nhìn vào kết quả này, dĩ nhiên $C3$ là điểm được chọn.

d. Tái tạo

- Sau bao cố gắng phía trên thì đây chính là bước được mong chờ nhất, đây chính là lúc đám mây điểm 3 chiều được tạo ra từ dãy ảnh. Thực tế mà nói, trong mỗi lần thiết bị tiến hành quét môi trường xung quanh, nó đã thu được một đám mây điểm, tuy nhiên đám mây này lại chưa hoàn thiện mà nó rời rạc với những đám mây điểm trước đó. Nguyên nhân là vì sau mỗi lần quét thứ thu được là đám mây điểm của hệ tọa độ camera. Tuy nhiên thứ chúng ta cần là một đám mây điểm trên một hệ tọa độ đồng nhất. Điều cần làm là xét mọi keyposes với hệ tọa độ của keypose đầu tiên, các đám mây điểm sau khi biến đổi sẽ được nối với nhau để tạo thành một không gian hoàn chỉnh.
- Việc ghép nối các điểm thành một object hoàn chỉnh là điều có thể thực hiện được, tuy nhiên trong nhiều trường hợp điều này lại không thực sự cần thiết. Ví dụ như trong trường hợp của chúng ta, hệ thống chỉ cần thu thập các điểm trong không gian để có thể né tránh vật cản. Những trường hợp cần xây dựng một object hoàn chỉnh thường là những trường hợp cần cho người dùng xem hoặc để in ra thành mô hình thật. Nếu điều này là cần thiết, hãy tham khảo thư viện PCL [36].

VI. Cấu trúc ba tầng ứng dụng:

1. Tầng 1: Mô hình xử lý dữ liệu TGMT

- Mô hình học:

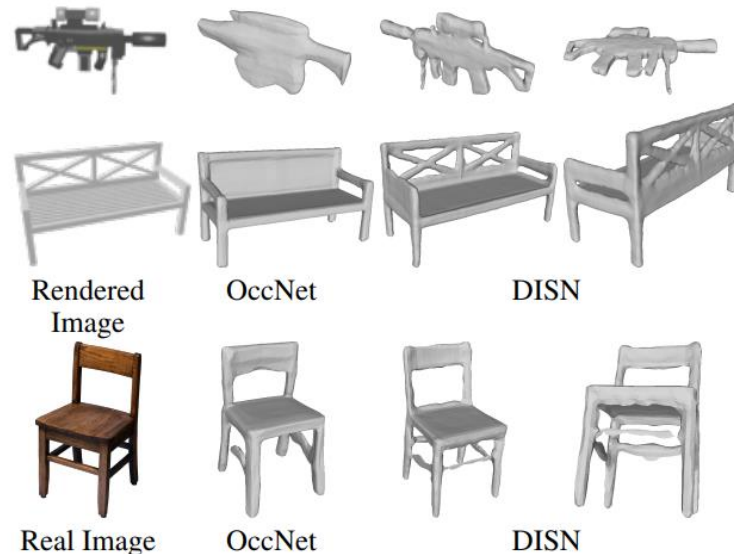
Học có giám sát – là thuật toán tiên đoán nhãn cho dữ liệu mới dựa trên tập huấn luyện (các mẫu trong tập này đều đã được gán nhãn). Thông qua quá trình huấn luyện, một mô hình sẽ được xây dựng cho các tiên đoán và khi các tiên đoán bị sai thì mô hình này sẽ được tinh chỉnh lại. Việc huấn luyện sẽ dừng lại khi mô hình đạt đến độ chính xác mong muốn dựa trên dữ liệu huấn luyện.

- Rút trích đặc trưng (points)
 1. Đầu tiên, khởi tạo ngẫu nhiên tham số cho những bộ lọc.
 2. Khi cho ảnh qua bộ lọc, ta được một số giá trị đầu ra output.
 3. Sau đó tiếp tục cho output qua các lớp tiếp theo trong mạng nơ-ron và mạng nơ-ron tiên đoán nhãn cho tấm này.
 4. Máy tính sẽ so sánh nhãn này với nhãn đúng của tập ảnh trong tập huấn luyện và điều chỉnh tham số của bộ lọc để kết quả gán nhãn giống hơn với nhãn đúng.
 5. Quá trình lặp đi lặp lại và mạng nơ-ron sẽ học được cách gán nhãn tốt nhất có thể.

2. Tầng 2:

- Tầng 2 tập trung vào các tác vụ:
 - Tái tạo ba chiều: Việc tái tạo 3 chiều đóng vai trò rất quan trọng trong các hệ thống tự hành. Với phương pháp DISN (Deep Implicit Surface Network) bao gồm 2 phần:
 - Ước tính tư thế máy ảnh
 - Tính SDF (signed point feature)

với các thông số trên máy ảnh của mobile robot, ta sẽ chiếu từng điểm truy vấn 3D lên mặt phẳng hình ảnh và thu thập các tính năng multi_scale CNN cho hình ảnh tương ứng. Sau đó DISN sẽ giải mã mã điểm không gian đã cho thành giá trị SDF bằng cách sử dụng cả các tính năng multi-scale local image và the global image features.



- Tự định vị:

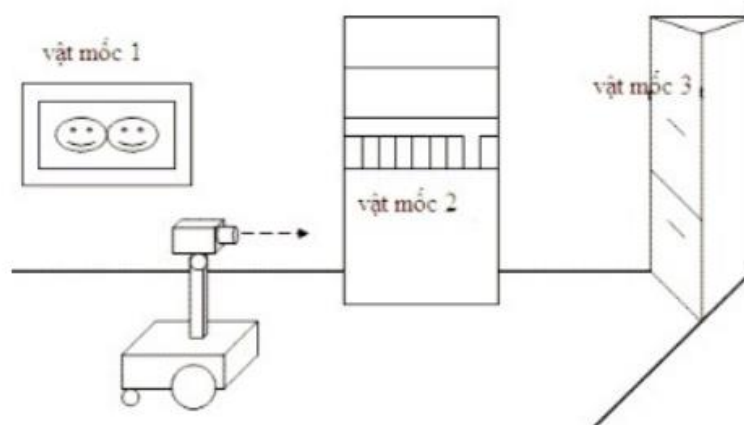
Có rất nhiều phương pháp để robot có thể tự định vị được vị trí trong môi trường:

1. Phương pháp Dead-reckoning

Dead-reckoning là phương pháp dẫn đường được sử dụng rộng rãi nhất đối với rô bốt di động. Phương pháp này cho độ chính xác trong thời gian ngắn, giá thành thấp và tốc độ lấy mẫu rất cao. Tuy nhiên do nguyên tắc cơ bản của phương pháp dead-reckoning là tích lũy thông tin về gia tốc chuyển động theo thời gian do đó dẫn tới sự tích lũy sai số. Sự tích lũy sai số theo hướng sẽ dẫn đến sai số vị trí lớn tăng tỉ lệ với khoảng cách chuyển động của robot. Dead-reckoning dựa trên nguyên tắc là chuyển đổi số vòng quay bánh xe thành độ dịch tuyến tính tương ứng của rô bốt. Nguyên tắc này chỉ đúng với giá trị giới hạn. Có một vài lý do dẫn đến sự không chính xác trong việc chuyển từ số gia vòng quay bánh xe sang chuyển động tuyến tính. Tất cả các nguồn sai số này được chia thành 2 nhóm: sai số hệ thống và sai số không hệ thống. Để giảm sai số dead-reckoning cần phải tăng độ chính xác động học cũng như kích thước tới hạn.

2. Phương pháp dẫn đường bằng cột mốc

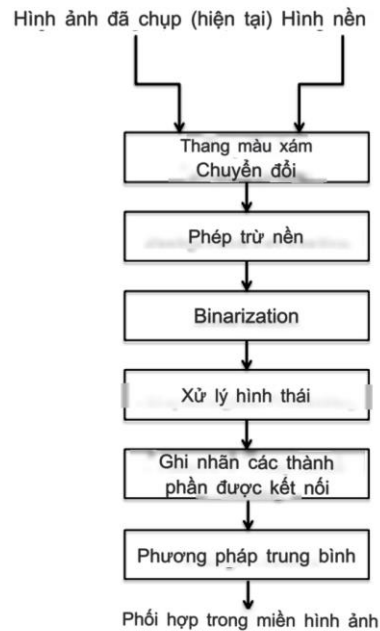
Các cột mốc được đặt tại các vị trí chính xác, có thể là các cột mốc nhân tạo hoặc tự nhiên, cho phép xác định chính xác tọa độ của vật thể. Có hai phương pháp đo dùng trong hệ thống cột mốc, đó là phép đo ba cạnh tam giác và phép đo ba góc tam giác. Phép đo 3 cạnh tam giác xác định vị trí vật thể dựa trên khoảng cách đo được tới cột mốc biết trước. Trong hệ thống dẫn đường sử dụng phép đo này thông thường có ít nhất là 3 trạm phát đặt tại các vị trí biết trước ngoài môi trường và 1 trạm nhận đặt trên **rô bốt**. Hoặc ngược lại có 1 trạm phát đặt trên rô bốt và các trạm nhận đặt ngoài môi trường. Sử dụng thông tin về thời gian truyền của chùm tia hệ thống sẽ tính toán khoảng cách giữa các trạm phát cố định và trạm nhận đặt trên **robot**



3. Phương pháp định vị sử dụng bản đồ

Rô bốt sử dụng các cảm biến được trang bị để tạo ra một bản đồ cục bộ môi trường xung quanh. Bản đồ này sau đó so sánh với bản đồ toàn cục lưu trữ sẵn trong bộ nhớ. Nếu tương ứng, rô bốt sẽ tính toán vị trí và góc hướng thực tế của nó trong môi trường.

- Trước khi xác định vị trí rô bốt bằng kỹ thuật bản đồ hóa, các hình ảnh thu được phải trải qua quá trình xử lý ảnh để loại bỏ nhiễu và thông tin không cần thiết. Các kỹ thuật này bao gồm thang màu xám, trừ nền, xử lý hình ảnh hình thái học và kỹ thuật gắn nhãn các thành phần được kết nối.



3. Tầng 3:

- Ứng dụng vào đâu:
 - Các robot, thiết bị tự hành hoạt động trong nhà, trong môi trường hẹp, gps không có sẵn hoặc không khả thi, ...
 - Các thiết bị dùng để vẽ lại bản đồ chi tiết của một khu vực.
 - Các thiết bị tự hành tránh né vật cản và vẽ bản đồ thời gian thực.
 - ...
- Sản phẩm nhóm hướng tới:
 - **Robot phục vụ bàn (waiter).**
 - Nhiệm vụ: order, bưng bê đồ ăn, vật phẩm lên cho khách hàng trong quán ăn, nhà hàng, khách sạn...
 - Nơi hoạt động: trong nhà, nơi có nhiều người di chuyển và có nhiều vật cản.
 - Lợi ích nếu dùng robot:
 - Tăng sự độc đáo cho nhà hàng.
 - Cắt giảm chi phí nhân sự.
 - Tăng độ chính xác, giảm độ trễ trong khâu di chuyển (robot sẽ nhanh hơn và ít sai sót hơn con người).
 - Giải quyết vấn đề thái độ phục vụ của nhân viên và vấn đề thái độ của khách tới nhân viên.
 - Tập khách hàng hướng tới:
 - Các nhà hàng khách sạn.
 - Tiềm năng:

- Theo thống kê năm 2020-2021 cả nước hiện có 550.000 cơ sở kinh doanh dịch vụ nhà hàng cà phê. Trong đó có 430.000 cơ sở kinh doanh truyền thống. 82.000 nhà hàng đồ ăn nhanh. 22.000 cửa hàng cà phê. Và các dịch vụ ăn uống khác khoảng 16.000 cơ sở [34].
- Các mô hình dịch vụ ăn uống đang ngày một nhiều và phát triển. Hầu hết các hàng quán hiện tại đều dùng nhân viên là con người để phục vụ, số lượng nhân viên có thể từ 1 tới hàng trăm nhân viên trên 1 cửa hàng.
- Các mô hình này cũng đang dần đổi mới, tìm điểm mới lạ.
- Ý tưởng này không mới, thậm chí ở Trung Quốc và các nước lớn nó đã được phổ biến rộng rãi dưới nhiều hình thức như: robot vận chuyển thay các shipper hay những robot trợ giúp ở sân bay. Nhưng ở Việt Nam ý tưởng này chưa thực sự có nhiều người làm.
- Vậy nên việc tạo ra một “bạn nhân viên” là robot hoàn toàn là một ý tưởng có thể ứng dụng và bán rộng rãi trên toàn lãnh thổ Việt Nam cũng như trên toàn thế giới nếu chúng ta có sự đầu tư thích hợp.
- Bán như thế nào:
 - Các công nghệ và tiện ích trên sản phẩm: sản phẩm dự tính sẽ có những chức năng như: tự động đưa đồ ăn, lau sàn trong quá trình di chuyển, tự động né vật cản, tự vẽ và ghi nhớ bản đồ...
 - Mức giá ước chừng: với thị trường là Việt Nam thì mức giá khoảng dưới 100 triệu sẽ là một mức giá hợp lý.
 - Kênh gọi vốn: các kênh online như các trang mạng xã hội, các công ty công nghệ lớn, các nhà hàng, khách sạn lớn. Ban đầu sẽ là gọi vốn đầu tư với những khuyến mãi, sau khi quy mô sản xuất đã đạt đủ lớn sẽ bán và doanh thu chính sẽ là tiền sản phẩm

VII. Tổng kết

Qua bài đồ án này, chúng em có thêm kiến thức về việc áp dụng các bài học lý thuyết trên lớp. Chúng em biết thêm ...

Xung quanh còn rất nhiều đồ án khác (cũng có bài thi cuối kỳ) nên thời gian dành cho đồ án này không đủ để phát triển thêm. Trong phạm vi đồ án, em cũng đã thử các tổng hợp nhiều kiến thức nhất có thể từ bài giảng và tham khảo thêm nhiều tư liệu khác.

Để hoàn thành được bài thực hành này, chúng em xin cảm ơn các giảng viên đã hỗ trợ nhiệt tình, tận tâm hết lòng vì chúng em trong môn học này, kính chúc thầy cô sức khỏe và niềm vui trong công việc giảng dạy tại HCMUS. Lời nói cuối cùng, mong sao ta sẽ được gặp lại.

VIII. Một số thuật ngữ - từ viết tắt:

- EKF: Extended Kalman Filter
- ECEF: Earth Centered Earth Fixed
- ENU: Local Level Frame
- GPS: Global Positioning System
- POI: Point of Interest

- SIFT: Scale-Invariant Feature Transform [31][32]
- SURF: Speeded Up Robust Features [33]
- VO: Visual Odometry
- LOS: line of sight
- IOP: Interior Orientation Parameters
- IMU: Inertial measurement unit
- DoG: Difference of Gaussian
- DISN: Deep Implicit Surface Network
- TGMT: Thị giác máy tính
- PCL: Point Cloud Library

IX. Tham khảo:

1. Haoyue Zheng, B.Eng. *Monitoring-Camera-Assisted SLAM for Indoor Positioning and Navigation* (2021).
https://macsphere.mcmaster.ca/bitstream/11375/26641/2/Zheng_Haoyue_202106_MASc.pdf
2. Rodrigo Munguia, Antoni Grau. *Monocular SLAM for Visual Odometry: A Full Approach to the Delayed Inverse-Depth Feature Initialization Method.* (2012).
<https://www.hindawi.com/journals/mpe/2012/676385/>
3. Sven Albrecht. *An Analysis of VISUAL MONO-SLAM* (2009).
<https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.403.489&rep=rep1&type=pdf>
4. Nirmal K A. *Visual SLAM: Possibilities, Challenges and the Future* (2020).
<https://ignitarium.com/visual-slam-possibilities-challenges-and-the-future>
5. Hong Liu, Zhi Wang, Pengjin Chen. *Feature points selection with flocks of features constraint for visual simultaneous localization and mapping* (2017).
https://www.researchgate.net/publication/314250363_Feature_points_selection_with_flocks_of_features_constraint_for_visual_simultaneous_localization_and_mapping
6. *SLAM (Simultaneous Localization and Mapping)*. <https://www.mathworks.com/discovery/slam.html>
7. Kim Uyên. *Giới thiệu tổng quát SLAM* (2020). <https://www.stdio.vn/dien-tu-ung-dung/gioi-thieu-tong-quat-ve-slam-xHu14>
8. Shaowu Yang, Andreas Zell. *Visual SLAM for autonomous MAVs with dual cameras* (2014).
https://www.researchgate.net/publication/260134138_Visual_SLAM_for_autonomous_MAVs_with_dual_cameras
9. Yi Yang, Di Tang, Dongsheng Wang, Wenjie Song, Junbo Wang, Mengyin Fu. *Multi-camera visual SLAM for off-road navigation.*
<https://www.sciencedirect.com/science/article/pii/S0921889019308711>
10. Erqun Dong, Jingao Xu, Chenshu Wu, Yunhao Liu, Zheng Yang. *Pair-Navi: Peer-to-Peer Indoor Navigation with Mobile Visual SLAM.* https://cswu.me/papers/infocom19_pairnavi.pdf
11. Sylwester Bala. *Introducing SLAM* (2018). <https://community.arm.com/arm-community-blogs/b/graphics-gaming-and-vr-blog/posts/introducing-slam-technology>
12. C.M. Ellum, Dr. N. El-Sheimy. *A MOBILE MAPPING SYSTEM FOR THE SURVEY COMMUNITY.*
https://www.researchgate.net/profile/Naser-El-Sheimy/publication/267244184_A_MOBILE_MAPPING_SYSTEM_FOR_THE_SURVEY_COMMUNITY/links/54cfbf690cf24601c0958d09/A-MOBILE-MAPPING-SYSTEM-FOR-THE-SURVEY-COMMUNITY.pdf

13. Al-Hamad, Amr. *Mobile Mapping Using Smartphones* (2014).
https://prism.ucalgary.ca/bitstream/handle/11023/1958/ucalgary_2014_al-hamad_amr.pdf?sequence=2&isAllowed=y
14. H. Salgues , H. Macher , T. Landes. *EVALUATION OF MOBILE MAPPING SYSTEMS FOR INDOOR SURVEYS* (2020).
https://www.researchgate.net/publication/344292038_EVALUATION_OF_MOBILE_MAPPING_SYSTEMS_FOR_INDOOR_SURVEYS
15. Mario Sabatino Riontino. *What is Mobile Mapping* (2020). <https://www.celantur.com/blog/getting-started-with-mobile-mapping/>
16. Kai Wei Chiang, Guang-Je Tsai, Jhih Cing Zeng. *Mobile Mapping Technologies* (2021).
https://link.springer.com/chapter/10.1007/978-981-15-8983-6_25
17. Rui Wang, Martin Schworer, Daniel Cremers. *Large-Scale Direct Sparse Visual Odometry with Stereo Cameras* (2017).
https://openaccess.thecvf.com/content_ICCV_2017/papers/Wang_Stereo_DSO_Large-Scale_ICCV_2017_paper.pdf
18. Carlos Campos, Richard Elvira, Juan J. Gomez Rodriguez, Jose M.M. Montiel, Juan D. Tardos. *An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM* (2021).
<https://arxiv.org/pdf/2007.11898v2.pdf>
19. video: https://www.youtube.com/watch?v=mQQL8pmztb4&ab_channel=DanielDeTone
20. Abdulrahman Saleh Alturki. *PRINCIPAL POINT DETERMINATION FOR CAMERA CALIBRATION*. (2017).
https://etd.ohiolink.edu/apexprod/rws_etd/send_file/send?accession=dayton1500326474390507&disposition=inline
21. Ebrahim Karami, Siva Prasad, and Mohamed Shehata. *Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images*.
<https://arxiv.org/ftp/arxiv/papers/1710/1710.02726.pdf>
22. Opencv. *Tutorial Python matcher*. https://docs.opencv.org/4.x/dc/dc3/tutorial_py_matcher.html
23. *Pinhole Camera Model*. <https://hedivision.github.io/Pinhole.html>
24. PCL. *Fast triangulation of unordered point clouds*.
https://pointclouds.org/documentation/tutorials/greedy_projection.html
25. S.Tiendee, C.Sinthanayothin. *Method of 3D Mesh Reconstruction from Point Cloud Using Elementary Vector and Geometry Analysis*.
https://www.researchgate.net/publication/277301611_Method_of_3D_mesh_reconstruction_from_point_cloud_using_elementary_vector_and_geometry_analysis
26. Sebastian Ochmanna, Richard Vocka, Reinhard Kleina. *Automatic reconstruction of fully volumetric 3D building models from point clouds*. (2019). <https://arxiv.org/pdf/1907.00631.pdf>
27. *Camera Models and Parameters*. <https://ftp.cs.toronto.edu/pub/psala/VM/camera-parameters.pdf>
28. Marcus A. Brubaker, Andreas Geiger, and Raquel Urtasun. *Map-Based Probabilistic Visual Self-Localization*. (2015). <https://www.cs.toronto.edu/~mbrubake/projects/PAMI15-Lost.pdf>
29. Ranjit Ray, Virendra Kumar, Debojyoti Banerjee, Debanjali Bhattacharya. *MAP-BUILDING AND MAP-BASED SELF-LOCALIZATION OF A MOBILE ROBOT USING SCAN-CORRELATION TECHNIQUE – A NEW APPROACH*. (2010). https://www.researchgate.net/publication/290753009_Map-building_and_map-based_self-localization_of_a_mobile_robot_using_scan-correlation_technique_-_A_new_approach
30. *The Extended Kalman filter*. <https://stanford.edu/class/ee363/lectures/ekf.pdf>

31. Ngọc Dao. *Giới thiệu giải thuật SIFT để nhận dạng ảnh*. <https://kipalog.com/posts/Gioi-thieu-giai-thuat-SIFT-de-nhan-dang-anh>
32. *Scale-Invariant Feature Transform*. Trong thư mục reference
33. Deepanshu Tyagi. *Introduction to SURF (Speeded-Up Robust Features)*. (2019). <https://medium.com/data-breach/introduction-to-surf-speeded-up-robust-features-c7396d6e7c4e>
34. Kathy Trần. *Tổng quan thị trường nhà hàng/cafe Việt Nam 2020 và Q1/2021*. (2021). <https://babuki.vn/tong-quan-thi-truong-nha-hang-cafe-viet-nam-2020-va-q1-2021/>
35. La Trường Phi. *Báo cáo mô hình hồi quy tuyến tính nhiều biến*. (2022). https://studenthcmusedu-my.sharepoint.com/:p:/g/personal/19127618_student_hcmus_edu_vn/Edwgmplg-dltd2QfXfgKrsBaCkrIcYWORAwlGidjkAo2g?e=0wy4BJ
36. *Point cloud library*. <https://pointclouds.org/>

☆ HẾT ☆