

# BÁO CÁO BÀI TẬP LỚN

## Môn Computer Vision

Xây dựng bản đồ bằng thuật toán SLAM

## Thành viên

Lê Trọng Đức

Phạm Văn Thành

Trần Hữu Trí

Phạm Minh Tiến

Giảng viên hướng dẫn : PGS. TS Vũ Hải

# Mục lục

<b>1 Giới thiệu bài toán</b>	<b>4</b>
1.1 Phát biểu bài toán . . . . .	4
1.2 Lý do chọn bài toán . . . . .	4
1.3 Mục tiêu mong muốn . . . . .	5
<b>2 Cơ sở lý thuyết</b>	<b>6</b>
2.1 Phương pháp calibration . . . . .	6
2.1.1 Các tham số calibration của camera . . . . .	7
2.2 Phương pháp trích chọn đặc trưng . . . . .	8
2.2.1 Phương pháp ORB (Oriented FAST and Rotated BRIEF) . . . . .	8
2.3 Monocular Visual Odometry (VO) . . . . .	10
2.3.1 Lý thuyết . . . . .	10
2.3.2 Ứng dụng của Monocular Visual Odometry . . . . .	11
<b>3 Triển khai giải quyết bài toán</b>	<b>14</b>
3.1 Thu thập dữ liệu . . . . .	14
3.2 Xử lý dữ liệu . . . . .	15
3.3 Lấy thông tin phần cứng . . . . .	15
3.4 Tìm hiểu, triển khai thử nghiệm thuật toán . . . . .	15
3.5 Kết quả thực hiện . . . . .	16

# Lời mở đầu

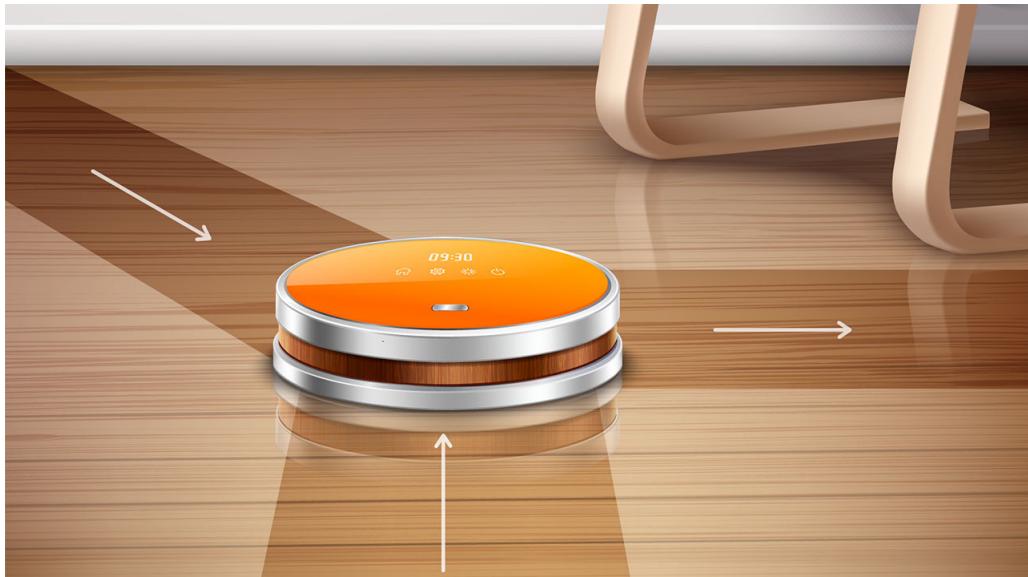
Khoa học càng ngày càng phát triển, nhu cầu xây dựng các hệ thống khoa học kỹ thuật để phục vụ cho cuộc sống con người ngày càng tăng cao. Xử lý hình ảnh là điển hình cho sự đóng góp của khoa học vào cuộc sống con người. Một trong những ứng dụng của xử lý hình ảnh là tái tạo môi trường vào map. SLAM - Simultaneous Localization And Mapping - Là hệ thống sử dụng thông tin ảnh thu được từ camera để tái tạo môi trường bên ngoài bằng cách đưa thông tin môi trường vào một map (2D hoặc 3D), từ đó thiết bị (robot, camera, xe) có thể định vị (localization) đang ở đâu, trạng thái, tư thế của nó trong map để tự động thiết lập đường đi (path planning) trong môi trường hiện tại. Ứng dụng của việc tái tạo môi trường xung quanh đã đưa công nghệ phát triển robots, xây dựng, ... đạt được những đột phá rất lớn. Trong đề tài này, nhóm chúng em đề xuất xây dựng và cài đặt SLAM với mong muốn nâng cao hiểu biết và áp dụng vào bài toán cụ thể trong cuộc sống.

# Chương 1: Giới thiệu bài toán

Ở chương này, chúng em sẽ trình bày sơ lược về nội dung bài toán mà nhóm sẽ giải quyết. Cùng với đó là mục đích và mục tiêu của nhóm khi lựa chọn bài toán này.

## 1.1 Phát biểu bài toán

SLAM là hệ thống sử dụng thông tin ảnh thu được từ camera để tái tạo môi trường bên ngoài bằng cách đưa thông tin môi trường vào một map (2D hoặc 3D), từ đó thiết bị (robot, camera, xe) có thể định vị (localization) đang ở đâu, trạng thái, tư thế của nó trong map để tự động thiết lập đường đi (path planning) trong môi trường hiện tại.



Hình 1.1: Hình ảnh ví dụ robot sử dụng SLAM

## 1.2 Lý do chọn bài toán

Giải quyết bài toán này sẽ giúp các thành viên trong nhóm có cơ hội tìm hiểu về xây dựng hệ thống SLAM, hiểu được cách vận hành hệ thống và trực tiếp xây dựng hệ thống. Đây là 1 bài toán tương đối nhiều ứng dụng nhưng cũng khá thử thách. Việc tìm hiểu bài toán cũng giúp nhóm có được 1 trải nghiệm về cách

các hệ thống xử lý hình ảnh được áp dụng vào trong cuộc sống như thế nào.

Đề tài được lấy ý tưởng từ bài học trong khóa học, cùng sự hướng dẫn của thầy giáo, nhóm đã nỗ lực hoàn thành và đã có những kết quả tương đối. Qua đó thu được những cái nhìn sâu hơn về hệ thống SLAM, cách các hình ảnh được sử lý, tổng hợp để xây dựng lên hệ thống.

### **1.3 Mục tiêu mong muốn**

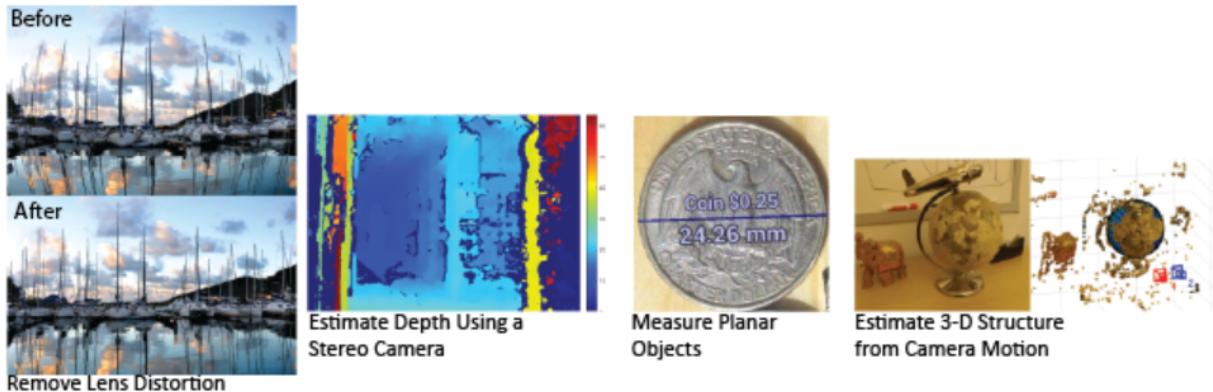
Mục tiêu cuối cùng của nhóm là xây dựng được hệ thống SLAM để giải quyết được bài toán xây dựng map của môi trường xung quanh. Để xây dựng được mô hình tốt nhất có thể, nhóm chúng em sẽ phải tìm hiểu và nắm chắc được về cơ sở lý thuyết SLAM, đánh giá mô hình. Ngoài ra, nhóm còn thu thập và xây dựng tập dữ liệu hình ảnh để áp dụng làm đầu vào cho bài toán. Các kết quả thu được trong các phần sau đều từ tập hình ảnh mà nhóm đã thu thập được

# Chương 2: Cơ sở lý thuyết

Trong chương 2, chúng em sẽ trình bày về các mô hình chính sẽ áp dụng để giải quyết bài toán xây dựng hệ thống SLAM.

## 2.1 Phương pháp calibration

Hiệu chỉnh máy ảnh hình học, còn được gọi là thay đổi kích thước máy ảnh, ước tính các thông số của ống kính và cảm biến hình ảnh của máy ảnh hoặc máy quay video. Bạn có thể sử dụng các thông số này để sửa độ méo ống kính, đo kích thước của vật thể theo đơn vị thế giới hoặc xác định vị trí của máy ảnh trong cảnh. Các tác vụ này được sử dụng trong các ứng dụng như thị giác máy để phát hiện và đo lường các đối tượng. Chúng cũng được sử dụng trong chế tạo người máy, hệ thống định vị và tái tạo cảnh 3-D.

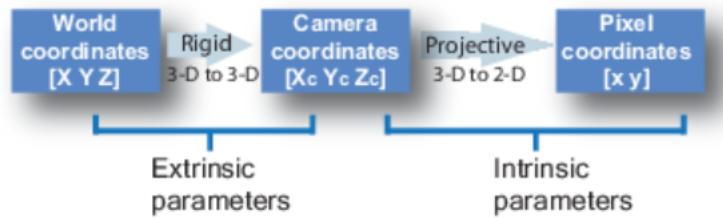


Hình 2.1: Ứng dụng của calibration

Các thông số của máy ảnh bao gồm hệ số nội hàm, ngoại lai và hệ số biến dạng. Để ước tính thông số máy ảnh, ta cần có điểm thế giới 3-D và điểm hình ảnh 2-D tương ứng của chúng. Ta có thể nhận được những tương ứng này bằng cách sử dụng nhiều hình ảnh của một mẫu hiệu chuẩn, chẳng hạn như bàn cờ. Sử dụng các thư từ, bạn có thể giải quyết các thông số máy ảnh. Sau khi hiệu chỉnh máy ảnh, để đánh giá độ chính xác của các thông số ước tính, bao gồm:

- Vẽ các vị trí tương đối của máy ảnh và mẫu hiệu chuẩn

- Tính toán các lối chiếu.
- Tính toán các sai số ước lượng tham số.



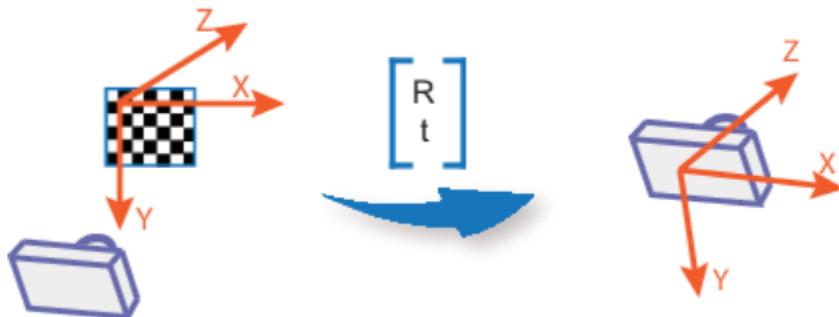
Hình 2.2: Tham số camera

### 2.1.1 Các tham số calibration của camera

Thuật toán hiệu chuẩn tính toán ma trận camera bằng cách sử dụng các tham số bên ngoài và bên trong. Các thông số bên ngoài thể hiện sự chuyển đổi cứng nhắc từ hệ tọa độ thế giới 3-D sang hệ tọa độ của máy ảnh 3-D. Các thông số nội tại thể hiện sự chuyển đổi xạ ảnh từ tọa độ của máy ảnh 3-D thành tọa độ hình ảnh 2-D.

#### Extrinsic Parameters

Các tham số bên ngoài bao gồm một phép quay, R và một phép tịnh tiến, t. Điểm gốc của hệ tọa độ của máy ảnh là tại trung tâm quang học của nó và trực x và y của nó xác định mặt phẳng hình ảnh.



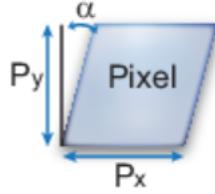
Hình 2.3: Tham số extrinsic

#### Intrinsic Parameters

Các thông số nội tại bao gồm độ dài tiêu cự, trung tâm quang học, còn được gọi là điểm chính và hệ số xiên. Ma trận nội tại của máy ảnh, K, được định nghĩa là:

$$\begin{bmatrix} f_x & 0 & 0 \\ s & f_y & 0 \\ c_x & c_y & 1 \end{bmatrix} \quad (2.1)$$

Dộ lệch pixel được định nghĩa là:



$[c_x \ c_y]$  — Optical center (the principal point), in pixels.

$(f_x, f_y)$  — Focal length in pixels.

$$f_x = F/p_x$$

$$f_y = F/p_y$$

$F$  — Focal length in world units, typically expressed in millimeters.

$(p_x, p_y)$  — Size of the pixel in world units.

$s$  — Skew coefficient, which is non-zero if the image axes are not perpendicular.

$$s = f_x \tan \alpha$$

## 2.2 Phương pháp trích chọn đặc trưng

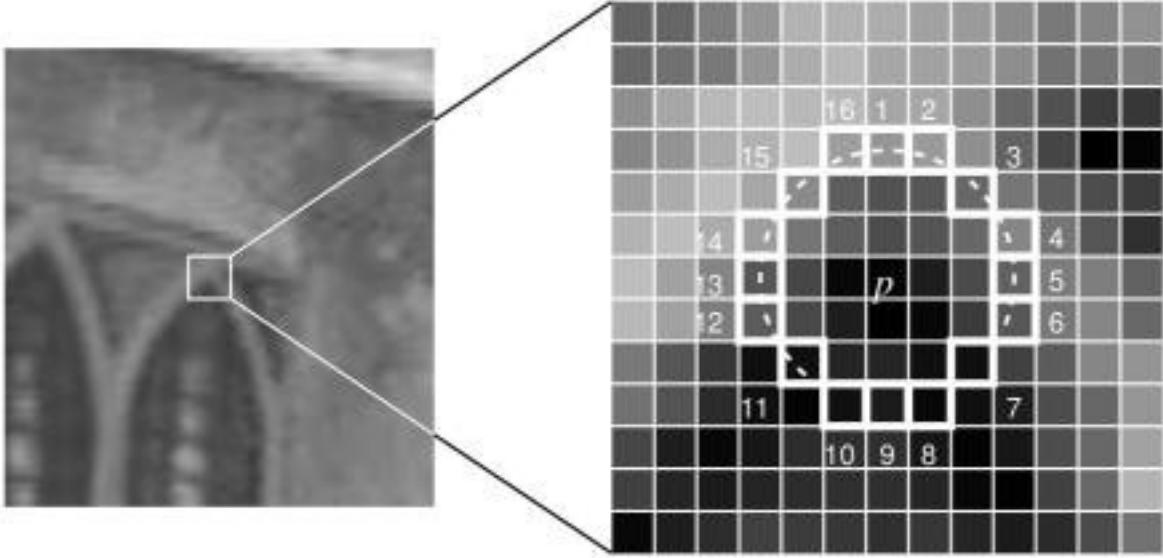
Trong thị giác máy tính và tính năng xử lý hình ảnh, tính năng phát hiện tính năng xử lý hình ảnh bao gồm các phương pháp tính toán sự trùng tương của thông tin hình ảnh và đưa ra quyết định cục bộ tại mọi điểm hình ảnh cho dù có đặc điểm hình ảnh của một loại nhất định tại điểm đó hay không. Các đặc trưng thu được sẽ là các tập con của miền ảnh, thường ở dạng các điểm biệt lập, các đường cong liên tục hoặc các vùng kết nối.

### 2.2.1 Phương pháp ORB (Oriented FAST and Rotated BRIEF)

Thuật toán này được đưa ra bởi Ethan Rublee, Vincent Rabaud, Kurt Konolige và Gary R. Bradski trong bài báo của họ ORB: Một giải pháp thay thế hiệu quả cho SIFT hoặc SURF vào năm 2011. Như tiêu đề đã nói, nó là một giải pháp thay thế tốt cho SIFT và SURF trong tính toán chi phí, hiệu suất phù hợp và chủ yếu là bằng sáng chế. Có, SIFT và SURF được cấp bằng sáng chế và bạn phải trả tiền cho họ để sử dụng nó.

ORB về cơ bản là sự kết hợp giữa FAST keypoint detector và BRIEF descriptor với nhiều sửa đổi để nâng cao hiệu suất. Đầu tiên, nó sử dụng FAST để tìm các điểm chính, sau đó áp dụng phép đo góc Harris để tìm N điểm cao nhất trong số đó. Nó cũng sử dụng kim tự tháp để tạo ra các tính năng đa cấp.

ORB tính toán trọng tâm cường độ của bản vá với góc nằm ở trung tâm. Hướng của vectơ từ điểm gốc này đến tâm là định hướng. Để cải thiện tính bất biến của vòng quay, mô men được tính với x và y phải nằm trong một vùng hình tròn bán kính  $r$ , trong đó  $r$  là kích thước của miếng vá.



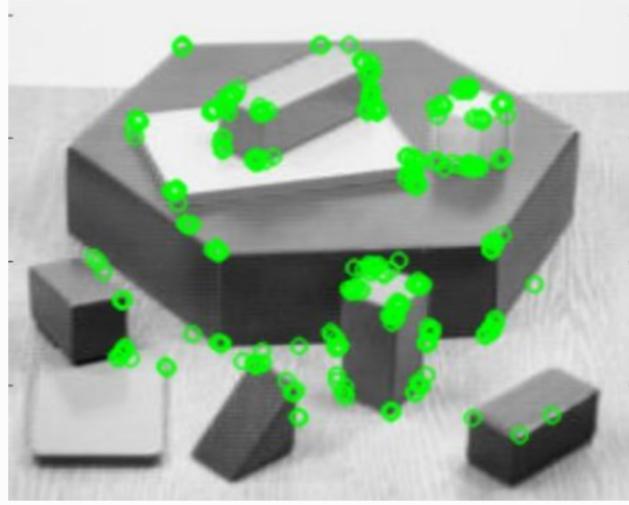
Hình 2.4: Phương pháp ORB

Bây giờ đối với bộ mô tả, ORB sử dụng bộ mô tả BRIEF. Nhưng chúng ta đã thấy rằng BRIEF hoạt động kém với việc xoay vòng. Vì vậy, những gì ORB làm là “điều khiển” SƠ LUỘC theo định hướng của các điểm chính. Đối với bất kỳ tập hợp tính năng nào gồm  $n$  phép thử nhị phân tại vị trí  $(x_i, y_i)$ , hãy xác định ma trận  $2 \times n$ ,  $S$  chứa tọa độ của các pixel này. Sau đó, bằng cách sử dụng định hướng của bản vá,  $\theta$ , ma trận xoay của nó được tìm thấy và xoay  $S$  để có được phiên bản được điều khiển (xoay)  $S_\theta$ .

ORB tùy chỉnh góc để tăng  $2\pi/30$  (12 độ) và xây dựng bảng tra cứu các mẫu BRIEF được tính toán trước. Miễn là hướng keypoint  $\theta$  nhất quán giữa các khung nhìn, tập hợp các điểm  $S_\theta$  chính xác sẽ được sử dụng để tính toán bộ mô tả của nó.

BRIEF có một đặc tính quan trọng là mỗi đặc trưng bit có phương sai lớn và giá trị trung bình gần 0,5. Nhưng một khi nó được định hướng theo hướng keypoint, nó sẽ mất đặc tính này và trở nên phân tán hơn. Phương sai cao làm cho một đối tượng địa lý dễ phân biệt hơn, vì nó phản ứng khác biệt với các đầu vào. Một tính chất mong muốn khác là không có các thử nghiệm tương quan với nhau, vì vậy mỗi thử nghiệm sẽ đóng góp vào kết quả. Để giải quyết tất cả những điều này, ORB chạy một tìm kiếm tham lam trong số tất cả các thử nghiệm nhị phân có thể có để tìm những thử nghiệm có cả phương sai cao và có nghĩa là gần 0,5, cũng như không có liên quan. Kết quả được gọi là rBRIEF.

Để đổi sánh bộ mô tả, LSH đa đầu dò cải tiến trên LSH truyền thống, được sử dụng. Bài báo cho biết ORB nhanh hơn nhiều so với SURF và SIFT và bộ mô tả ORB hoạt động tốt hơn SURF. ORB là một lựa chọn tốt trong các thiết bị tiêu thụ điện năng thấp để ghép ảnh toàn cảnh, v.v.



Hình 2.5: Kết quả của thuật toán ORB

### 2.3 Monocular Visual Odometry (VO)

Monocular Visual Odometry (VO) là quá trình xác định vị trí và hướng của camera bằng cách phân tích một chuỗi hình ảnh. Monocular Visual Odometry được sử dụng trong nhiều ứng dụng, chẳng hạn như rõ bót di động, ô tô tự lái và phương tiện bay không người lái.

Vấn đề nghiên cứu chính của VO là làm thế nào để ước tính quỹ đạo của máy ảnh theo hình ảnh. Đối với hình ảnh bản địa hóa và ánh xạ đồng thời (vSLAM), có một vấn đề nghiên cứu chính, đó là tạo ra một vòng lặp và tích hợp hiệu quả các ràng buộc mới vào bản đồ hiện tại dựa trên VO.

#### 2.3.1 Lý thuyết

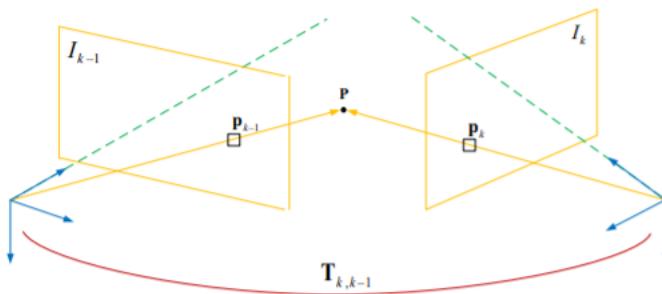
Đối với monocular VO, tại thời điểm  $k$ , bộ ảnh được chỉ định  $I_{0:k} = I_0, \dots, I_k$ , được thu bởi máy ảnh của một vật cứng người máy. Giả sử rằng tọa độ máy ảnh là tọa độ của người máy. Tuy nhiên, trong hệ thống thị giác âm thanh nổi, camera bên trái thường là bản gốc.

Tuy nhiên, việc sử dụng VO trong ống nhòm dẫn đến sự suy giảm mạnh về độ chính xác của phép đo tam giác, như khoảng cách giữa tâm của hai máy ảnh bị ảnh hưởng bởi các điều kiện của độ chính xác của phép đo và sự thay đổi khí hậu.

Một phép biến đổi  $T_{k,k-1} \in R^{4 \times 4}$  được hình thành bởi hai máy ảnh hàng xóm đặt ra từ thời điểm  $k-1$  và  $k$ , được hiển thị như sau:

$$T_{k,k-1} = \begin{bmatrix} R_{k,k-1} & t_{k,k-1} \\ 0 & 1 \end{bmatrix} \quad (2.2)$$

với  $R_{k,k-1}$  là ma trận xoay,  $t_{k,k-1}$  là ma trận dịch. Đặt  $T_{1:k} = \{T_{1,0}, T_{2,1}, \dots, T_{k,k-1}\}$  bao gồm các chuỗi



Hình 2.6: Bài toán onocular VO

chuyển động. Cuối cùng, thiết lập tư thế máy ảnh là  $C_{0:k} = C_0, C_1, \dots, C_k$   $C_k$  có thể tính từ connection giữa phép chuyển  $T_k (k = 1, 2, \dots, n)$  là tọa độ ban đầu tại thời điểm .

Mục tiêu chính của VO là tính  $T_k$  từ ảnh  $I_k$  sang ảnh  $I_{k-1}$  và sau đó tích hợp tất cả các phép biến đổi thành khôi phục toàn bộ đường dẫn  $C_{0:k}$  của camera. VO là một quỹ đạo tái thiết tăng dần. Tối ưu hóa lặp đi lặp lại dựa trên  $m$  tư thế trước đó có thể được thực hiện. Và sau đó, ước tính quỹ đạo cục bộ chính xác hơn có thể thu được. Tối ưu hóa lặp đi lặp lại giảm thiểu lỗi từ chối của Các điểm 3D trong bản đồ địa phương dựa trên m khung hình trước đó (ví dụ: điều chỉnh gói dựa trên cửa sổ trượt, vì nó thực thi trên cửa sổ m-frame). Độ sâu của các điểm 3D trong không gian bản đồ địa phương được ước tính bằng phương pháp tam giác. Do đó, một vấn đề tối ưu hóa có thể được xây dựng, điều chỉnh R và t để cho tất cả các điểm tính năng  $z^j$ , lỗi tích lũy của hai định mức là tối thiểu, và kết quả như sau:

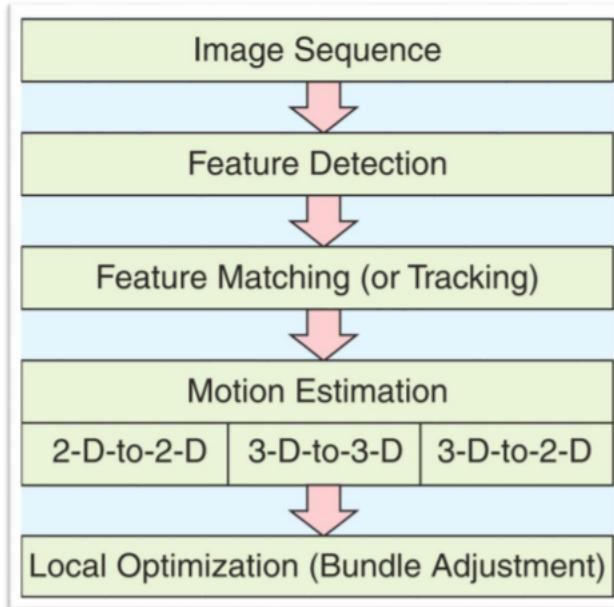
$$\min_{X, R, t} \sum_{j=1}^N \left\| \frac{1}{\lambda_1} CX^j - [z_1^j, 1]^T \right\|^2 + \left\| \frac{1}{\lambda_2} C(RX^j + t) - [z_2^j, 1]^T \right\|^2.$$

### 2.3.2 Ứng dụng của Monocular Visual Odometry

Monocular vision là một bài toán khó nhưng rất thú vị, đặc biệt là ứng dụng của nó vào vấn đề điều hướng chung.

Mục đích của Monocular Visual Odometry (VO) là ước tính tư thế của robot dựa trên một số phép đo từ (các) hình ảnh. Thật khó để xác định một nguyên lý cốt lõi duy nhất - Lý thuyết Xác suất Bayes có thể là nguyên lý cốt lõi, nhưng hình học biểu thức chắc chắn quan trọng.

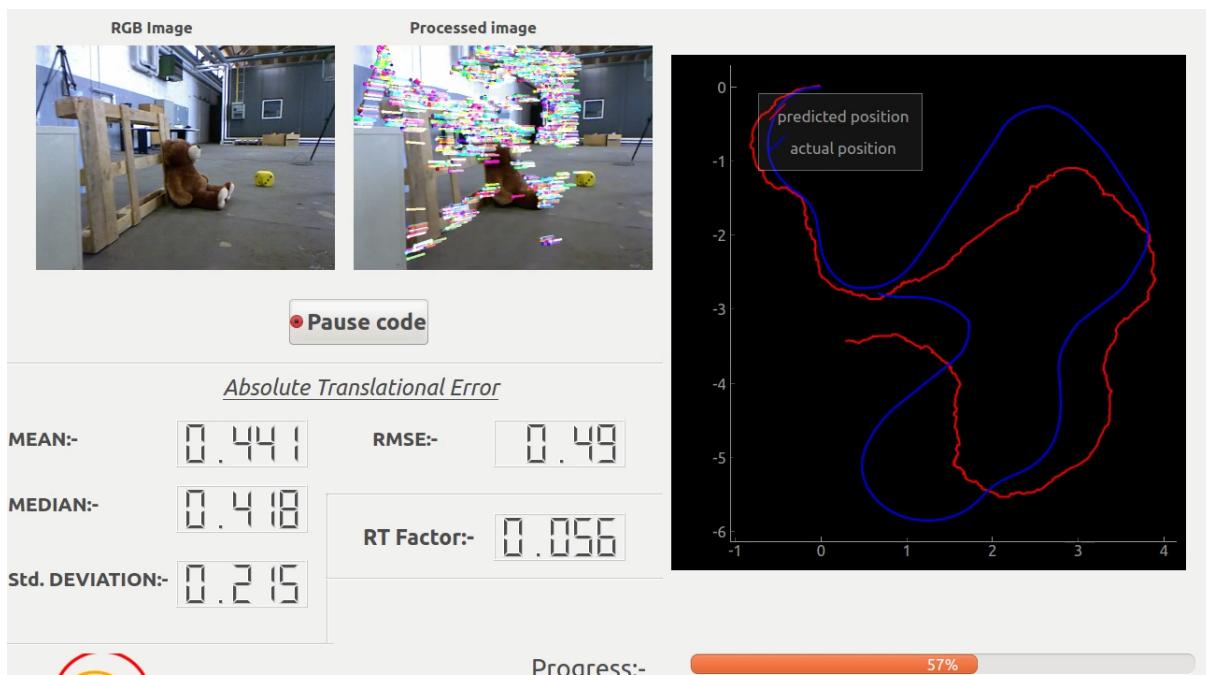
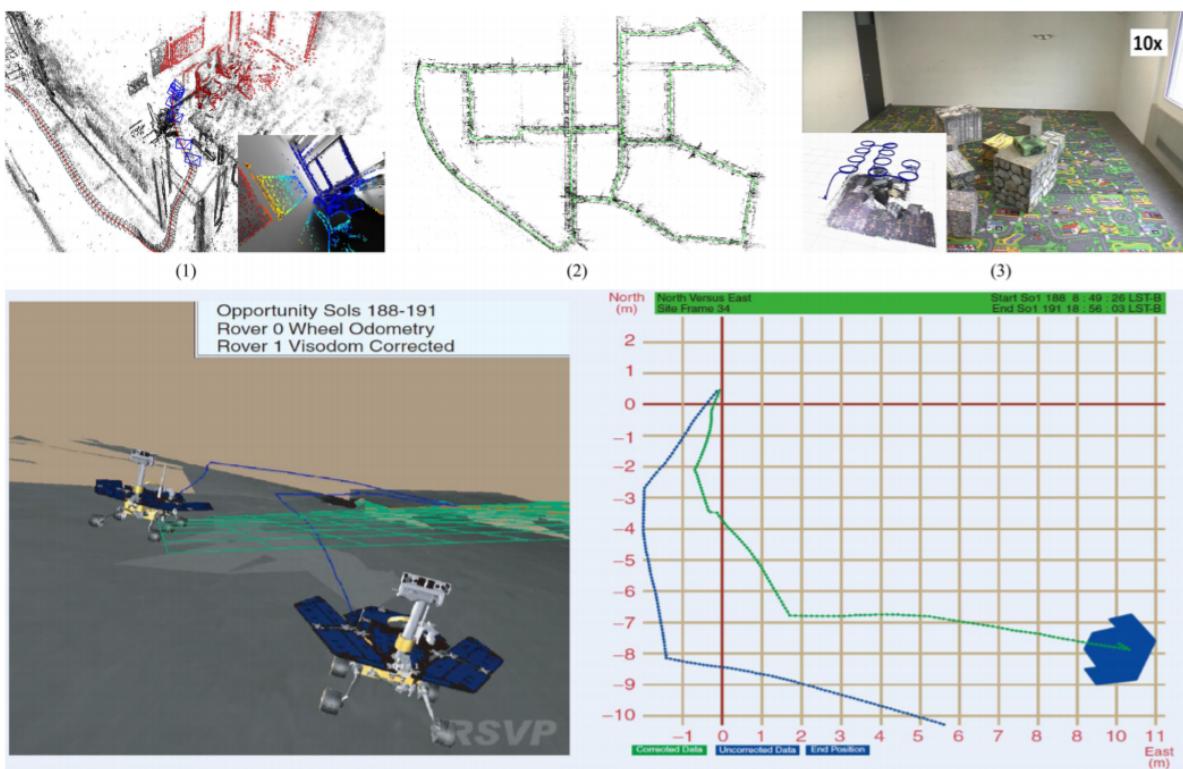
Đối với âm thanh nỗi, ý tưởng chung là nếu bạn biết các thông số máy ảnh của mình và bạn biết rằng mối quan hệ giữa máy ảnh của bạn là cố định, thì một điểm p được xem từ cả hai máy ảnh sẽ chiếu lên mỗi mặt phẳng hình ảnh một đoạn đường biểu diễn. Sử dụng mặt phẳng cực hình được xác định bởi các đoạn đường thẳng này và hình học của máy ảnh, có thể ước tính khoảng cách đến điểm, có thể được sử dụng để ước tính tư thế của robot.



Hình 2.7: Pipeline of VO

Dối với thị giác một mắt, nhiệm vụ này phức tạp hơn vì chỉ có một camera. Để có một cặp ảnh, một ảnh được chụp tại thời điểm  $t$ ; rõ bốt được kích hoạt và một hình ảnh khác được chụp tại thời điểm  $t + 1$  (ví dụ: luồng video). Cặp hình ảnh này có thể được sử dụng giống như bất kỳ hình ảnh âm thanh nổi nào khác, nhưng vì chúng tôi không biết chắc chắn mối quan hệ giữa các máy ảnh (do chuyển động sai lệch so với những gì rõ bốt đã làm và những gì nó được hướng dẫn), để ước tính tư thế của robot, mô hình chuyển động phải được tính đến. Cuối cùng, phép đo thị giác một mắt khó hơn vì có nhiều ẩn số hơn phải được tính toán một cách xác suất.

Mục tiêu chính là đối sánh hình ảnh. Trong khi có nhiều cách tiếp cận cho vấn đề này, một số cách tiếp cận tìm ra điểm đặc trưng trong cảnh và so khớp giữa các hình ảnh để khớp với chính hình ảnh. Một số thuật toán có thể được sử dụng để tìm điểm đặc trưng; SIFT là một tiêu chuẩn tốt mặc dù các phương pháp khác đã được chứng minh là tốt hơn nó. Với các tính năng được mô tả, một thuật toán phù hợp với các điểm nên được sử dụng; RANSAC là một trong những thuật toán như vậy. Tận dụng các mối quan hệ đã biết (chẳng hạn như mô hình chuyển động chính xác hoặc máy ảnh / máy chiếu cố định) có thể cải thiện khả năng khớp hình ảnh.



Hình 2.8: Minh họa kết quả của VO

# Chương 3: Triển khai giải quyết bài toán

Ở chương này, chúng em sẽ trình bày sơ lược về những công việc mà nhóm thực hiện triển khai giải quyết bài toán xây dựng map theo thuật toán SLAM. Bài toán được triển khai toàn bộ các bước từ đầu cuối gồm: Thu thập dữ liệu; Xử lý dữ liệu; Lấy thông tin phần cứng; Tìm hiểu, triển khai thử nghiệm thuật toán.

## 3.1 Thu thập dữ liệu

Dầu tiên, với dữ liệu thử nghiệm nhóm sử dụng bộ dữ liệu có sẵn là KITTI DATASET để có thể xem cấu trúc của các bài toán đã triển khai trước đó.

Với dữ liệu thực tế cần thu thập, nhóm sử dụng thiết bị là camera GO PRO HERO 7 rồi thực hiện gắn camera vào xe tự lái, cho xe tự lái chạy quanh tầng 8 để lưu lại video quang đường xe chạy. Tuy nhiên, xe tự lái nhóm nhận được hoạt động không được tốt nên nhóm đã tìm cách thay thế xe tự lái bằng một thiết bị bán tự lái do nhóm thiết kế để thực hiện thu thập video. Nhóm sẽ thực hiện đẩy xe bán tự lái này



Hình 3.1: Hình ảnh xe bán tự lái do nhóm thiết kế

đi quanh tầng 8 để quay video làm dữ liệu xây dựng bản đồ thực tế.

## 3.2 Xử lý dữ liệu

Với dữ liệu lấy được từ camera GO PRO thì video sẽ có các khung hình mặc định với kích thước 1920x1080 và FPS là 30. Việc đầu tiên để xử lý dữ liệu này là chuyển video về tập các ảnh với tỷ lệ 3 frame từ video sẽ lưu lại 1 frame. Việc thứ hai nhóm thực hiện là giảm kích thước ảnh xuống 60% so với ban đầu để cho việc chạy thuật toán ở các bước sau được dễ dàng hơn.

## 3.3 Lấy thông tin phần cứng

Với việc triển khai thuật toán Visual Odometry thì số liệu quan trọng cần sử dụng là intrinsic parameter của camera. Để lấy được cái tham số này, nhóm sử dụng các hàm của OpenCV để thực hiện tìm như sau:

- Quay video với ảnh có hình bàn cờ bằng camera
- Thực hiện calibrate từ hình ảnh trong video để thực hiện tính toán tham số



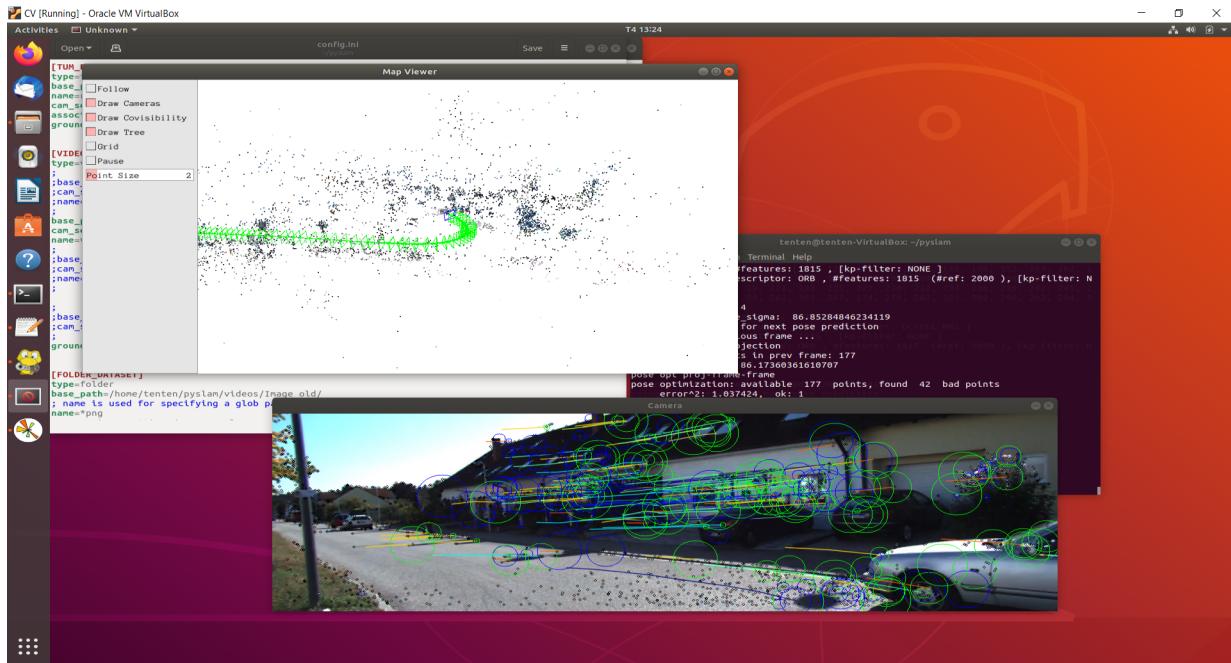
Hình 3.2: Hình ảnh thực hiện calibrate để thu thập tham số camera

## 3.4 Tìm hiểu, triển khai thử nghiệm thuật toán

Nhóm thực hiện tìm hiểu thuật toán SLAM với monocular camera đã được trình bày ở trên. Việc triển khai các thuật toán này vì nhóm chỉ có kinh nghiệm chính với ngôn ngữ Python nên việc lựa chọn các giải pháp thực hiện cũng bị hạn chế theo. Việc triển khai cũng gặp khó khăn do tốc độ chạy của ngôn ngữ Python cũng chậm hơn C++ và tài nguyên triển khai của nhóm bị hạn chế nên không thử nghiệm được nhiều thuật toán SLAM, thay vào đó nhóm đã thực hiện các thử nghiệm với 1 phần của thuật toán là phần trích xuất đặc trưng của các khung ảnh liên tiếp để xem kết quả triển khai trên dữ liệu thực tế. Nhóm

thực hiện thử nghiệm với ORB-SLAM, SHI-TOMASHI, FAST, BRISK, ORB, AKAZE, SIFT, ROOT-SIFT, SUPERPOINT, CONTEXTDESC.

### 3.5 Kết quả thực hiện



Hình 3.3: Hình ảnh kết quả vẽ bản đồ với bộ dữ liệu KITTI với SHI-TOMASI-ORB

# Tổng kết

Sau khi hoàn thành bài tập lớn này, các thành viên trong nhóm đã hiểu về bài toán sinh mặt người tự động. Nhóm hiểu rằng đây là một bài toán khó, còn nhiều thách thức tuy nhiên tính ứng dụng của nó là vô cùng lớn. Đặc biệt là khi bây giờ, sự phát triển về phần cứng đã tạo ra các nhu cầu về nguồn dữ liệu lớn cho các nhà nghiên cứu. Do vậy, việc phát triển các thuật toán SLAM đã đạt được những thay đổi lớn và có nhiều đóng góp cho công nghiệp robot, xây dựng và rất nhiều các ngành liên quan.

Trong quá trình thực hiện bài tập lớn nhóm chúng em đã có thêm nhiều kiến thức về các mô hình SLAM, các kiến thức về xử lý hình ảnh. Hơn nữa, mọi người cũng có thêm các kỹ năng về nghiên cứu, hoạt động nhóm, xây dựng hệ thống SLAM.

Chúng em xin cảm ơn thầy Vũ Hải đã tham gia hướng dẫn, giúp đỡ nhóm hoàn thành được bài tập lớn này. Nhóm nhận thấy kết quả vẫn cần phải cải thiện nên các thành viên sẽ tiếp tục nghiên cứu, tìm hiểu nhiều phương pháp để có thể nâng cao được hiệu năng của mô hình hơn.

# Tài liệu tham khảo

- *AI Academy* - CV course slides [1]
- *Visual Odometry (VO)* [http://www.cs.toronto.edu/~urtasun/courses/CSC2541/03\\_odometry.pdf](http://www.cs.toronto.edu/~urtasun/courses/CSC2541/03_odometry.pdf) [2]
- *Ming He, Chaozheng Zhu, Qian Huang, Baosen Ren Jintao Liu* - A review of monocular visual odometry [3]
- <https://dattadebrup.github.io/monocular/inertial/odometry/2018/07/23/Monocular-Visual-and-Inertial-Odometry.html> - Monocular Visual and Inertial Odometry [4]
- *Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, Sanja Fidler* - Skip-Thought Vectors [5]
- *Richard Szeliski* - Computer Vision: Algorithms and Applications, 2nd ed. [6]
- *Ethan Rublee; Vincent Rabaud; Kurt Konolige; Gary Bradski* - ORB: An efficient alternative to SIFT or SURF [7]
- <https://www.mathworks.com/help/vision/ug/camera-calibration.html> [8]