

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN

Nhóm CKT

**Xây dựng hệ thống phát hiện và tránh
vật cản cho hệ tự hành**

Ứng dụng xử lý ảnh số và video số 20_23

Giảng viên – TS. Lý Quốc Ngọc

Thành phố Hồ Chí Minh – 2023

MỤC LỤC

THÔNG TIN CÁC THÀNH VIÊN TRONG NHÓM	3
Lời Cảm Ơn	4
I. Giới thiệu	5
1. Bối cảnh chung	5
2. Động lực nghiên cứu	6
<i>a) Về mặt khoa học</i>	6
<i>b) Về mặt ứng dụng</i>	6
3. Phát biểu bài toán	7
II. Các công trình liên quan	8
1. "Real-time obstacle avoidance for autonomous vehicles in unstructured environments" của N. B. Priyantha và cộng sự (2006)	8
2. "Obstacle detection and avoidance system for unmanned ground vehicles" của H. Ali và cộng sự (2016)	10
3. "Vision-based obstacle detection and avoidance for unmanned aerial vehicles" của A. Bazzoli và cộng sự (2019)	11
4. "Obstacle detection and avoidance for mobile robots using multiple sensors and neural networks" của J. Zhang và cộng sự (2017)	13
III. Phương pháp	14
1. Ước định vị trí và tái tạo bản đồ	14
2. Tránh vật cản tĩnh	29
3. Tránh vật cản động	40
References	45

THÔNG TIN CÁC THÀNH VIÊN TRONG NHÓM

MSSV	Họ Tên	Email	Ghi chú
20120201	Phạm Gia Thông	20120201@student.hcmus.edu.vn	
20120259	Nguyễn Văn Chung	20120259@student.hcmus.edu.vn	Nhóm trưởng

Lời Cảm Ơn

Chúng em xin chân thành cảm ơn thầy Lý Quốc Ngọc – giảng viên lý thuyết và đồng thời cũng là giảng viên thực hành của môn học “Ứng dụng xử lý ảnh số và video số” đã trang bị cho nhóm chúng em những kiến thức, tài liệu và kỹ năng cơ bản cần có để có thể hoàn thành đồ án của môn học.

Trong quá trình làm đồ án, do kiến thức về môn học còn hạn chế nên nhóm không tránh khỏi nhiều thiếu sót khi tìm hiểu và thực hiện đồ án. Rất mong nhận được sự quan tâm, góp ý và đánh giá của các thầy để đồ án của chúng em được đầy đủ và hoàn thiện hơn.

Chúng em xin chân thành cảm ơn.

I. Giới thiệu

1. Bối cảnh chung

Lí do ra đời của việc nghiên cứu hệ thống này:

- An toàn giao thông: Trong lĩnh vực ô tô, xe tự lái và xe hơi tự động, việc phát hiện và tránh né vật cản là một yếu tố quan trọng để đảm bảo an toàn cho người lái và người đi đường. Hệ thống này có thể giúp phát hiện các vật cản như xe khác, người đi bộ, xe đạp, hay chướng ngại vật trên đường và đưa ra các biện pháp tránh né để tránh va chạm.
- Công nghiệp và robot hợp tác: Trong môi trường công nghiệp hoặc khi sử dụng robot hợp tác, hệ thống phát hiện và tránh né vật cản có thể giúp máy móc tránh va chạm với các vật thể không mong muốn, bảo vệ công nhân và ngăn chặn thiệt hại cho thiết bị và môi trường xung quanh.
- Robot gia đình và dịch vụ: Trong ngành công nghệ gia đình và dịch vụ, hệ thống phát hiện và tránh né vật cản có thể được áp dụng cho robot hút bụi tự động, robot trợ giúp trong việc chăm sóc người cao tuổi hoặc người khuyết tật. Điều này giúp tránh các va chạm không mong muốn và bảo vệ các vật thể nhạy cảm, như đồ vật trong nhà hoặc con người.
- Khám phá không gian và robot khám phá: Trong ngành robot khám phá và khám phá không gian, hệ thống phát hiện và tránh né vật cản có thể giúp các robot di chuyển qua môi trường khó khăn như môi trường ngoài trái đất, mặt trăng, hoặc dưới nước. Điều này cho phép robot tự động tránh các vật thể, như đá, hố sâu, hoặc vật thể nguy hiểm khác, và đảm bảo việc điều hướng an toàn và hiệu quả.

→ Có nhu cầu cấp thiết trong việc xây dựng hệ thống phát hiện và tránh né vật cản đối với những lĩnh vực trên bởi những lí do sau:

- An toàn: Hệ thống này đảm bảo an toàn cho con người và các thiết bị bằng cách ngăn chặn các va chạm không mong muốn với vật cản, giảm thiểu nguy cơ tai nạn và thương tích.
- Hiệu suất: Đối với các ứng dụng công nghiệp và robot hợp tác, việc tránh né vật cản giúp tăng hiệu suất và giảm thiểu thời gian dừng lại do va chạm hoặc xung đột.
- Tự động hóa: Hệ thống này cho phép các thiết bị tự động hoạt động và di chuyển một cách an toàn và đáng tin cậy, mở ra tiềm năng tự động hóa cao hơn trong các lĩnh vực khác nhau.
- Tiện ích: Tránh né vật cản giúp bảo vệ các vật thể quan trọng và giảm thiểu thiệt hại vật chất, cung cấp lợi ích kinh tế và bảo vệ tài sản.

2. Động lực nghiên cứu

a) Về mặt khoa học

Việc xây dựng hệ thống Phát hiện và Tránh vật cản cho hệ tự hành có ý nghĩa khoa học rất lớn trong lĩnh vực robot học và trí tuệ nhân tạo. Dưới đây là một số ý nghĩa khoa học của việc xây dựng hệ thống này:

- Bài toán detection: Tăng độ chính xác cho việc nhận dạng các vật thể lớn bé khác nhau thay đổi liên tục, đồng thời nhận dạng ảnh có độ mờ hay nhiều do môi trường, nhận diện các đối tượng bị che khuất, nhận dạng được đối tượng ở không gian 3D, tăng tốc độ nhận dạng và xử lý, tạo thêm dữ liệu để huấn luyện nhận dạng.
- Bài toán tracking: Trong quá trình theo dõi, có những lúc vật biến đổi các hướng trên không gian 3D nên khi giải được bài toán ta có theo vết vật tốt hơn về mặt biến đổi hình dạng và màu sắc, theo vết được nhiều đối tượng cùng lúc, theo dõi được các đối tượng bị mất mát thông tin do không có trong khung hình hoặc bị khuất.
- Bài toán classification: Giúp việc phân loại chính xác hơn với các đối tượng như nhau nhưng hành động với chúng là khác nhau vì trong bài toán sẽ phải phân loại đối tượng cần tránh né và đối tượng vật cản không cần né.

b) Về mặt ứng dụng

Hệ thống Phát hiện và Tránh vật cản (Obstacle Detection and Avoidance System) là một thành phần quan trọng trong hệ thống tự hành, giúp xe tự động phát hiện và tránh các vật cản trong quá trình di chuyển. Đây là một công nghệ đang được nghiên cứu và phát triển để đưa vào sử dụng trong xe tự động, robot di động và các thiết bị tự hành khác. Dưới đây là một số ứng dụng của việc xây dựng hệ thống Phát hiện và Tránh vật cản cho hệ tự hành:

- Tăng cường an toàn giao thông: giúp hoàn thiện xe tự hành trong khâu tránh né đưa ra quyết định tăng giảm tốc hay bẻ lái tránh xảy ra tai nạn, hỗ trợ quản lý giao thông tự động tránh vật cản có thể tăng hay giảm tốc hoặc chọn vị trí thuận lợi để không bị kẹt từ đó sẽ không gây ùn tắc giao thông.
- Tăng hiệu quả sản xuất trong các nhà máy tự động: Hệ thống này cũng có thể được áp dụng trong các nhà máy tự động để giúp robot di chuyển trong không gian hạn chế, tránh các vật cản trong quá trình di chuyển và tối ưu hóa quá trình sản xuất, giảm thiểu các lỗi do chậm trễ và đi không tới nơi của robot.
- Hỗ trợ điều khiển máy bay không người lái (Drone): Hệ thống Phát hiện và Tránh vật cản có thể được áp dụng để hỗ trợ điều khiển máy bay không người lái (Drone) có thể tránh các vật cản trong không gian và giúp nó thực hiện các nhiệm vụ khác nhau một cách an toàn và hiệu quả hơn.

- Hỗ trợ trong công nghiệp địa chất: Hệ thống Phát hiện và Tránh vật cản có thể được sử dụng để hỗ trợ trong các hoạt động khai thác và khai thác tài nguyên thiên nhiên, giúp xe tự động tránh các vật cản trong môi trường địa chất phức tạp và giảm nguy cơ tai nạn.
- Hỗ trợ trong y tế: Hệ thống này cũng có thể được sử dụng trong y tế để giúp robot di động di chuyển trong bệnh viện, tránh các vật cản trong quá trình di chuyển và hỗ trợ cho việc điều trị và chăm sóc bệnh nhân.
- Hỗ trợ trong nhà hàng: Hệ thống này giúp cho các robot giao đồ ăn và phục vụ né tránh các bàn, ghế, khách hàng, các đầu bếp và dụng cụ trong quán ăn thuận lợi và tăng tốc quá trình phục vụ và vận chuyển hàng hóa trong nhà hàng

Như vậy, hệ thống Phát hiện và Tránh vật cản là một công nghệ quan trọng trong hệ tự hành, có nhiều ứng dụng trong đời sống

3. Phát biểu bài toán

Bài toán xây dựng hệ thống phát hiện và tránh vật cản cho hệ tự hành là một bài toán quan trọng trong lĩnh vực trí tuệ nhân tạo và robotica. Mục tiêu của bài toán là thiết kế một hệ thống nhận diện và phản ứng với các vật cản trong môi trường để giúp hệ tự hành di chuyển an toàn và hiệu quả.

Hệ thống phát hiện và tránh vật cản cần có khả năng xử lý dữ liệu từ các cảm biến, bao gồm máy quét laser, camera và cảm biến siêu âm, để phát hiện và định vị vật cản trong không gian xung quanh. Sau đó, hệ thống cần có khả năng phân tích và đánh giá thông tin này để quyết định các hành động phản ứng như di chuyển, tránh vật cản hoặc dừng lại.

Để xây dựng một hệ thống phát hiện và tránh vật cản hiệu quả, cần phải sử dụng các phương pháp và thuật toán như phân tích hình ảnh, xử lý tín hiệu, phân loại và lọc thông tin. Đồng thời, cần có khả năng học máy và điều khiển tự động để cải thiện khả năng dự đoán và phản ứng của hệ thống.

Bài toán phát hiện và tránh vật cản cho hệ tự hành là một trong những thách thức quan trọng trong việc phát triển các hệ thống tự động. Việc giải quyết bài toán này sẽ mang lại nhiều lợi ích cho các lĩnh vực như vận tải, dịch vụ, nông nghiệp, y tế và nhiều lĩnh vực khác.

Mục tiêu của bài toán Xây dựng hệ thống Phát hiện và Tránh vật cản cho hệ tự hành là xây dựng một hệ thống máy tính thông minh, có khả năng phát hiện và tránh các vật cản trong quá trình di chuyển của một xe tự động.

Sơ đồ: Bộ phận cảm biến -> Bộ phận phân tích-> Né tránh vật cản

Input:

- Thông tin từ các cảm biến, bao gồm camera, lidar, radar, GPS để hệ thống có thể nhận biết và định vị vật cản trong môi trường.
- Camera: dãy ảnh f capture liên tục $f(x,y,t)$ với $t=1 \dots n$, $f_i(x,y)$ với $i=1 \dots n$
- Lidar: đám mây điểm 3D (PCL) so với lidar $\{x_i(t), y_i(t), z_i(t)\}$ với $i=1 \dots n$, $t=1 \dots n$
- GPS: vị trí của đối tượng so với hệ quy chiếu của nó trong không gian, so với vị trí của đối tượng và vật cản.
- Radar: quét qua đối tượng động/tĩnh trả về chuyển động của đối tượng

Output:

- Các bounding box 3D của đối tượng (vật cản/không phải vật cản)
- Vectơ chuyển động của xe (độ lớn, hướng di chuyển)
- Phương hướng chuyển động của đối tượng \Rightarrow có phải vật cản không

Giới hạn của bài toán Xây dựng hệ thống Phát hiện và Tránh vật cản cho hệ tự hành là tập trung vào việc phát hiện và tránh các vật cản có kích thước lớn trong môi trường đường phố, bao gồm cả các vật cản động và tĩnh như xe cộ, người đi bộ, đường vật cản, tường, cột điện và cây cối, vật cản phải có đủ các yếu tố ánh sáng và không bị che lấp

Thách thức:

- Phát hiện vật cản: ở camera ta thiếu xác định chiều thứ 3, lidar thiếu tập điểm thuộc đối tượng
- Vật nào là vật cản vật nào không: yếu tố động của vật cản gây cản trở vì mỗi lúc vật trả về 1 kết quả khác nhau là cản hay không cản tránh, không đủ dữ liệu về các yếu tố như vị trí và tốc độ hay hướng di chuyển của vật cản so với hệ thống
- Tránh vật cản: vật cản đang di chuyển gây nhiều việc tránh, tránh như nào để về đích sớm đôi lúc sẽ đi xa khỏi đích

II. Các công trình liên quan

1. "Real-time obstacle avoidance for autonomous vehicles in unstructured environments" của N. B. Priyantha và cộng sự (2006)

Bài báo này trình bày một phương pháp phát hiện và tránh vật cản trên xe tự động sử dụng các cảm biến và thuật toán điều khiển, và thực hiện tránh vật cản một cách thời gian thực trong môi trường không có cấu trúc.

Trong công trình, tác giả đã giới thiệu một thuật toán điều khiển hợp tác cho phép xe tự hành tránh được các vật cản trong môi trường không cấu trúc.

Thuật toán này bao gồm ba bước chính:

- Xác định các vật cản: Hệ thống sử dụng các cảm biến như lidar và radar để xác định các vật cản trong phạm vi hoạt động.
- Dự đoán hướng di chuyển của các vật cản: Hệ thống sử dụng một mô hình dự đoán để ước tính hướng di chuyển của các vật cản, dựa trên thông tin về tốc độ và hướng di chuyển trước đó của chúng.
- Lập kế hoạch tránh vật cản: Hệ thống sử dụng một mô hình tối ưu để tính toán hành động tối ưu để tránh vật cản, bao gồm tốc độ, hướng di chuyển và quỹ đạo di chuyển.

Tác giả đã chứng minh rằng thuật toán này có thể hoạt động hiệu quả trong môi trường không cấu trúc, nơi các vật cản xuất hiện ngẫu nhiên và có thể di chuyển một cách không đều. Tuy nhiên, thuật toán này cần được cải tiến để đáp ứng yêu cầu của các ứng dụng thực tế, như là cải thiện tốc độ tính toán và độ chính xác của dự đoán vật cản.

Nguyên lý: tránh vật cản bằng cách tính toán khoảng cách dự toán chuyển động và hướng của vật cản rồi len vào khoảng trống giữa các vật cản nếu có đủ diện tích

Độ đo: Không dùng mạng học và dùng tập dữ liệu tự tạo của tác giả (không được cung cấp trong công trình).

Thành tựu và khuyết điểm:

- Giải quyết vấn đề camera xác định thiếu chiều thứ ba: Công trình đã giải quyết vấn đề thiếu chiều thứ ba trong việc xác định vật cản bằng cách sử dụng các kỹ thuật và cảm biến khác nhau, không chỉ dựa vào thông tin từ camera. Việc sử dụng nhiều cảm biến và kỹ thuật khác nhau như LIDAR, RADAR, hoặc các phương pháp xử lý hình ảnh khác đã cho phép hệ thống có thông tin đầy đủ về môi trường xung quanh và tránh vật cản hiệu quả hơn.
- Linh động trong việc xác định tập điểm thuộc đối tượng: Công trình đã tìm cách giải quyết vấn đề lidar (thiếu tập điểm thuộc đối tượng) bằng cách sử dụng các phương pháp xử lý dữ liệu và thuật toán phân đoạn (segmentation) để phân biệt vật cản và nền. Việc này giúp hệ thống có thể xác định rõ ràng các đối tượng trong môi trường và tránh va chạm với chúng.
- Chưa giải quyết được yếu tố động của vật cản: Mặc dù công trình đã đạt được kết quả tốt trong việc tránh vật cản tĩnh, nhưng vẫn còn tồn đọng một số thách thức khi đối mặt với vật cản động. Các phương pháp trong công trình chưa thực hiện phân tích và

dự đoán hành vi của các vật cản động, dẫn đến khả năng tránh va chạm chưa linh hoạt và hiệu quả trong những tình huống đó.

- Chưa phân biệt vật cần tránh và vật không cần tránh: Công trình chưa đưa ra giải pháp cụ thể để phân biệt giữa các vật cản mà phương tiện tự hành cần tránh và những vật cản không gây nguy hiểm. Điều này có thể dẫn đến việc hệ thống tránh vật cản không cần thiết hoặc không đáng tin cậy, gây ra sự cản trở và giảm hiệu suất của phương tiện tự hành.

→ Tóm lại, công trình đã giải quyết một số vấn đề quan trọng liên quan đến tránh vật cản trong môi trường không cấu trúc. Tuy nhiên, vẫn còn những thách thức và vấn đề cần được nghiên cứu và cải thiện, như yếu tố động của vật cản và khả năng phân biệt vật cần tránh và vật không cần tránh.

2. "Obstacle detection and avoidance system for unmanned ground vehicles" của H. Ali và cộng sự (2016)

Bài báo này trình bày một hệ thống phát hiện và tránh vật cản cho các xe không người lái trên đất, sử dụng các cảm biến và thuật toán điều khiển để đảm bảo an toàn và hiệu quả.

Công trình sử dụng một thuật toán phát hiện và tránh vật cản dựa trên các cảm biến hồng ngoại và siêu âm.

Thuật toán của công trình bao gồm các bước sau:

- Thu thập dữ liệu từ các cảm biến hồng ngoại và siêu âm.
- Sử dụng các thuật toán xử lý tín hiệu để loại bỏ nhiễu và đối sánh các dữ liệu từ các cảm biến khác nhau để xác định vị trí và khoảng cách của các vật cản.
- Dựa trên thông tin về vị trí và khoảng cách của các vật cản, đưa ra quyết định tránh vật cản bằng cách tính toán đường đi an toàn và điều khiển UAV di chuyển tránh vật cản.

Các bước của thuật toán này được thực hiện liên tục để đảm bảo rằng UAV có thể phát hiện và tránh vật cản trong thời gian thực. Tuy nhiên, công trình này chưa đề cập đến việc xử lý các trường hợp phức tạp khi có nhiều vật cản xuất hiện cùng một lúc, hoặc khi vật cản di chuyển nhanh hoặc không đều.

Nguyên lý: tránh vật cản bằng cách tính toán khoảng cách vật cản rồi len vào nơi to nhất có độ lớn lớn hơn diện tích của hệ thống

Độ đo: Công trình được dùng tập dữ liệu tự tạo không được đề cập

Thành tựu và khuyết điểm:

- Giải quyết vấn đề camera xác định thiếu chiều thứ ba: Công trình đã sử dụng các cảm biến và kỹ thuật khác nhau, không chỉ dựa vào camera, để đạt được thông tin đầy đủ về môi trường xung quanh xe. Bằng cách kết hợp các cảm biến như camera, LIDAR, RADAR và ultrasonic, công trình đã giải quyết được vấn đề thiếu chiều thứ ba trong việc xác định vật cản.
- Linh động trong việc xác định tập điểm thuộc đối tượng: Công trình đã sử dụng các thuật toán và phương pháp xử lý dữ liệu để phân tích thông tin từ các cảm biến và xác định tập điểm thuộc đối tượng. Bằng cách áp dụng các thuật toán như phân đoạn (segmentation) và phân loại (classification), công trình đã giải quyết vấn đề lidar và có khả năng phân biệt vật cản và nền.
- Chưa giải quyết được yếu tố động của vật cản: Công trình chưa đưa ra giải pháp cụ thể để xử lý yếu tố động của vật cản. Việc dự đoán và ứng phó với hành vi động của vật cản vẫn là một thách thức đối với hệ thống tránh vật cản của công trình.
- Chưa phân biệt vật cần tránh và vật không cần tránh: Công trình chưa đề cập đến cách phân biệt giữa các vật cần tránh và vật không cần tránh. Điều này có thể dẫn đến sự tránh vật cản không cần thiết hoặc không đáng tin cậy, gây ra sự cản trở và giảm hiệu suất của hệ thống.

→ Tóm lại, công trình đã giải quyết một số vấn đề quan trọng liên quan đến phát hiện và tránh vật cản cho xe không người lái trên mặt đất. Tuy nhiên, vẫn còn những thách thức và vấn đề cần được nghiên cứu và cải thiện, như yếu tố động của vật cản và khả năng phân biệt vật cần tránh và vật không cần tránh.

3. "Vision-based obstacle detection and avoidance for unmanned aerial vehicles" của A. Bazzoli và cộng sự (2019)

Bài báo này trình bày một phương pháp phát hiện và tránh vật cản cho các máy bay không người lái, sử dụng các cảm biến hình ảnh và thuật toán xử lý hình ảnh để đảm bảo an toàn trong quá trình di chuyển.

Công trình sử dụng một thuật toán phát hiện và tránh vật cản dựa trên hình ảnh và các thuật toán xử lý hình ảnh.

Thuật toán của công trình bao gồm các bước sau:

- Thu thập hình ảnh từ các camera được gắn trên UAV.
- Sử dụng các thuật toán xử lý hình ảnh như bộ lọc Canny, phân vùng hình ảnh và phân loại đối tượng để phát hiện các vật cản xung quanh UAV.
- Xác định vị trí và khoảng cách của các vật cản bằng cách sử dụng các thuật toán xử lý hình ảnh khác nhau, bao gồm tính toán độ sâu và áp dụng giải thuật mô hình đồ thị.

- Dựa trên thông tin về vị trí và khoảng cách của các vật cản, đưa ra quyết định tránh vật cản bằng cách tính toán đường đi an toàn và điều khiển UAV di chuyển tránh vật cản.

Các bước của thuật toán này được thực hiện liên tục để đảm bảo rằng UAV có thể phát hiện và tránh vật cản trong thời gian thực.

Nguyên lý: tránh vật cản bằng cách tính toán khoảng cách vật cản rồi len vào nơi to nhất có độ lớn lớn hơn diện tích của hệ thống

Độ đo: Công trình được dùng tập dữ liệu UZH-FPV Drone Racing Dataset. Tập dữ liệu UZH-FPV Drone Racing Dataset cung cấp các hình ảnh và video chất lượng cao với độ phân giải và tốc độ khung hình cao. Nó cung cấp một tập hợp đa dạng về môi trường, bao gồm các vật cản đa dạng như cột, tường, cây cối, và các vật thể di động khác.

Thành tựu và khuyết điểm:

- Giải quyết vấn đề camera xác định thiếu chiều thứ ba: Công trình đã sử dụng các phương pháp và thuật toán xử lý hình ảnh để khắc phục vấn đề thiếu chiều thứ ba trong việc xác định vật cản. Bằng cách sử dụng các thuật toán stereo vision hoặc structure from motion (SFM), công trình đã tái tạo được một phần thông tin về chiều sâu hoặc cấu trúc 3D của môi trường xung quanh UAVs từ các hình ảnh 2D thu thập được từ camera.
- Linh động trong việc xác định tập điểm thuộc đối tượng: Công trình đã sử dụng các thuật toán xử lý hình ảnh và máy học để phân đoạn (segmentation) vật cản và xác định tập điểm thuộc đối tượng. Các phương pháp như semantic segmentation, instance segmentation, hoặc object detection đã được áp dụng để xác định rõ ràng các vật cản trong môi trường và tránh va chạm với chúng.
- Chưa giải quyết được yếu tố động của vật cản: Mặc dù công trình đã đạt được kết quả tốt trong việc tránh vật cản tĩnh, nhưng vẫn còn tồn đọng một số thách thức khi đối mặt với vật cản động. Các phương pháp trong công trình chưa thực hiện phân tích và dự đoán hành vi của các vật cản động, dẫn đến khả năng tránh va chạm chưa linh hoạt và hiệu quả trong những tình huống đó.
- Chưa phân biệt vật cần tránh và vật không cần tránh: Công trình chưa đưa ra giải pháp cụ thể để phân biệt giữa các vật cản mà UAVs cần tránh và những vật cản không gây nguy hiểm. Điều này có thể dẫn đến sự tránh vật cản không cần thiết hoặc không đáng tin cậy, gây ra sự cản trở và giảm hiệu suất của hệ thống.

→ Tóm lại, công trình đã giải quyết một số vấn đề quan trọng liên quan đến phát hiện và tránh vật cản cho UAVs dựa trên hình ảnh camera. Tuy nhiên, vẫn còn những thách thức và

vấn đề cần được nghiên cứu và cải thiện, như yếu tố động của vật cản và khả năng phân biệt vật cần tránh và vật không cần tránh.

4. "Obstacle detection and avoidance for mobile robots using multiple sensors and neural networks" của J. Zhang và cộng sự (2017)

Bài báo này trình bày một phương pháp phát hiện và tránh vật cản cho các robot di động, sử dụng nhiều cảm biến và mạng nơ-ron để đạt được hiệu suất cao và độ tin cậy trong việc tránh vật cản.

Công trình sử dụng một thuật toán phát hiện và tránh vật cản dựa trên một mạng nơ-ron nhân tạo và nhiều cảm biến khác nhau.

Thuật toán của công trình bao gồm các bước sau:

- Thu thập dữ liệu từ các cảm biến siêu âm, cảm biến hồng ngoại, cảm biến tầm nhìn và cảm biến lực.
- Xử lý dữ liệu từ các cảm biến bằng một mạng nơ-ron nhân tạo để xác định vị trí và khoảng cách của các vật cản xung quanh robot.
- Dựa trên dữ liệu từ mạng nơ-ron, đưa ra quyết định tránh vật cản bằng cách tính toán đường đi an toàn và điều khiển robot di chuyển tránh vật cản.

Các bước của thuật toán này được thực hiện liên tục để đảm bảo rằng robot có thể phát hiện và tránh vật cản trong thời gian thực.

Nguyên lý: tránh vật cản bằng cách dựa vào dữ liệu huấn luyện mạng học đưa ra kết quả nên tránh trong từng trường hợp

Độ đo: Công trình được dùng tập dữ liệu KITTI. Tập dữ liệu KITTI bao gồm hình ảnh, dữ liệu 3D và các thông tin liên quan khác như lưu lượng giao thông, định vị GPS và đo tốc độ. Các tác giả đã sử dụng tập dữ liệu này để đánh giá hiệu suất của hệ thống phát hiện và tránh vật cản của họ.

Thành tựu và khuyết điểm:

- Giải quyết vấn đề camera xác định thiếu chiều thứ ba: Công trình đã sử dụng nhiều cảm biến khác nhau, bao gồm cả camera và các cảm biến khác như lidar, ultrasonic, hoặc radar, để thu thập thông tin về môi trường xung quanh robot. Việc sử dụng nhiều cảm biến khác nhau giúp bổ sung thông tin chiều sâu và khắc phục vấn đề thiếu chiều thứ ba trong hình ảnh từ camera.
- Linh động trong việc xác định tập điểm thuộc đối tượng: Công trình đã sử dụng mạng neural và các thuật toán xử lý dữ liệu để phân loại và xác định tập điểm thuộc đối

tượng từ thông tin thu thập được từ các cảm biến. Việc sử dụng mạng neural giúp robot nhận biết và phân đoạn vật cản từ dữ liệu đa dạng thu thập được từ nhiều cảm biến khác nhau.

- Chưa giải quyết được yếu tố động của vật cản: Mặc dù công trình sử dụng nhiều cảm biến và mạng neural để xử lý dữ liệu và phát hiện vật cản, nhưng vẫn còn một số thách thức trong việc đối phó với yếu tố động của vật cản. Các phương pháp trong công trình chưa thực hiện phân tích và dự đoán hành vi của các vật cản động, dẫn đến khả năng tránh va chạm chưa linh hoạt và hiệu quả trong những tình huống đó.
- Chưa phân biệt vật cản tránh và vật không cần tránh: Công trình chưa đưa ra giải pháp cụ thể để phân biệt giữa các vật cản mà robot cần tránh và những vật cản không gây nguy hiểm. Điều này có thể dẫn đến sự tránh vật cản không cần thiết hoặc không đáng tin cậy, gây ra sự cản trở và giảm hiệu suất của hệ thống.

→ Tóm lại, công trình đã đạt được một số tiến bộ quan trọng trong việc phát hiện và tránh vật cản cho robot di động bằng cách sử dụng nhiều cảm biến và mạng neural. Tuy nhiên, vẫn còn những thách thức và vấn đề cần được nghiên cứu và cải thiện, như yếu tố động của vật cản và khả năng phân biệt vật cản tránh và vật không cần tránh.

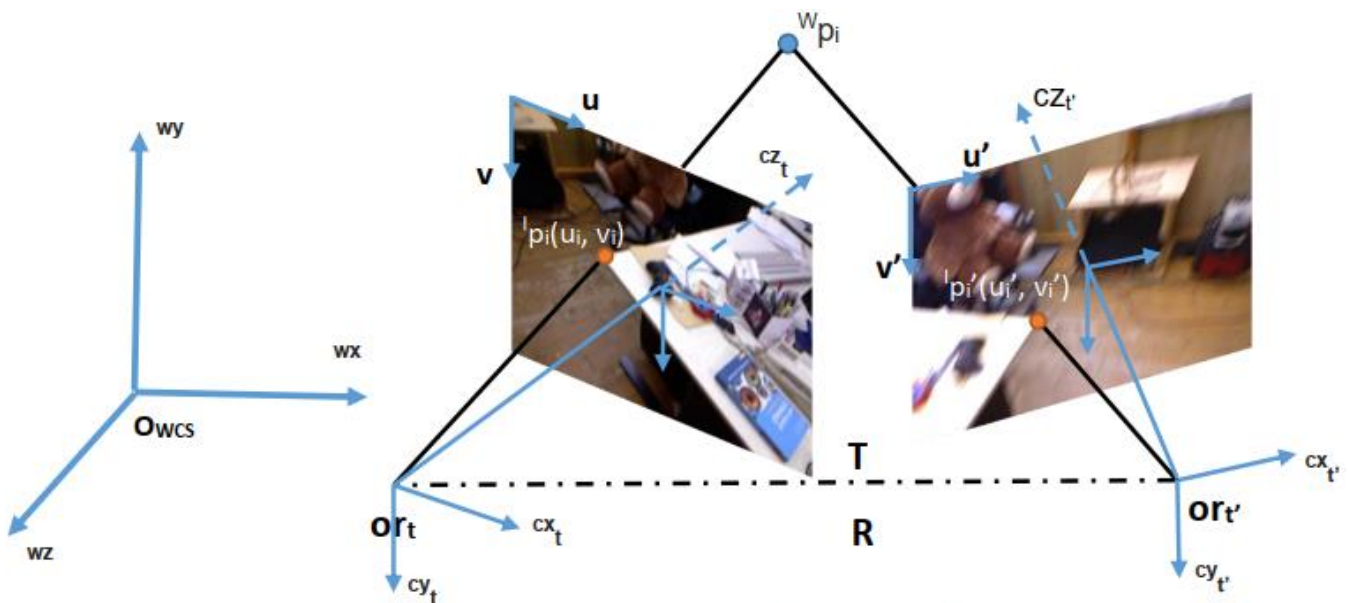
III. Phương pháp

Nguyên lý cơ bản: Tái tạo bản đồ rồi tìm khoảng trống lớn nhất để len vào sao với điều kiện khoảng trống gần đích nhất, có xem xét thêm yếu tố về đích.

- Thu thập thông tin về môi trường xung quanh: Sử dụng các cảm biến như lidar, radar hoặc camera để nhận biết và đo khoảng cách đến các vật cản trong phạm vi phát hiện.
- Xác định khoảng trống lớn nhất: Phân tích dữ liệu từ cảm biến để tìm khoảng trống lớn nhất giữa các vật cản. Khoảng trống này thường là một không gian trống rộng và không có vật cản lớn nào.
- Theo dõi và định vị khoảng trống: Theo dõi vị trí của khoảng trống trong quá trình di chuyển. Điều này có thể được thực hiện bằng cách theo dõi sự thay đổi của các vật cản và tính toán lại vị trí của khoảng trống.
- Điều chỉnh hướng di chuyển: Dựa trên vị trí của khoảng trống, điều chỉnh hướng di chuyển của robot hoặc xe tự hành để tránh va chạm với các vật cản. Thông thường, robot sẽ cố gắng di chuyển qua khoảng trống lớn nhất mà không gặp phải vật cản.

1. Ước định vị trí và tái tạo bản đồ

Để có vị trí hiện tại, robot cần biết được nó đã di chuyển theo hướng nào, với khoảng cách là bao nhiêu với thời điểm trước đó. Thông tin này có thể thu được từ thiết bị đo lường quán tính (Inertial Measurement Unit-IMU). Tuy nhiên, thông tin này không chính xác theo thời gian, rất dễ bị trôi (drift) khiến cho ước lượng vị trí robot sai. Vậy nên để ước lượng chuyển động của robot kết hợp thông tin hình ảnh (chính xác hơn) và thông tin từ IMU (nhanh hơn) nhằm khiến hệ thống hoạt động chính xác và hiệu quả trong thời gian thực: Chuyển động đến r_t sẽ được ký hiệu là W_t bao gồm một phép tịnh tiến T và một phép xoay R . Tại hai vị trí r_t và $r_{t'}$ robot cùng quan sát được cùng một điểm trong môi trường thực tế (gọi là điểm tương ứng), tuy nhiên do sự khác nhau về vị trí quan sát và phép chiếu mà ta chỉ biết được điểm đó có vị trí (u, v) trong ảnh tại vị trí r_t và vị trí (u', v') trong ảnh tại vị trí $r_{t'}$, hai vị trí quan sát cùng một điểm không thay đổi trong không gian ba chiều $w_p(x, y, z)$. Nhiệm vụ của thuật toán ước lượng chuyển động là tìm một tập điểm tương ứng và vị trí ba chiều của tập điểm, sau đó ước lượng vị trí (phép tịnh tiến T) và hướng (phép xoay R) thể hiện chuyển động đến r_t của robot theo hệ tọa độ thế giới (WCS).



Hình 1 Minh họa cho bài toán ước lượng chuyển động bằng thông tin thị giác. Điểm màu xanh là điểm trong không gian ba chiều trên thực tế. Điểm màu đỏ là tọa độ trên ảnh của điểm thực tế khi camera quan sát được tại hai thời điểm khác nhau t, t' .

Với sự xuất hiện của dữ liệu IMU, ta có thể biết được chính xác phép quay R trong phép biến đổi W_t , và với thông tin hình ảnh từ hai camera để ước lượng phép dịch chuyển T khi đã biết phép xoay R từ IMU. Thông tin hình ảnh đến từ hai camera chính (25Hz) và camera phụ (1Hz) có tần số khác nhau nên thông tin hình ảnh sẽ được chia thành hai hệ thống gồm có camera đơn (camera chính 25Hz) (monocular) và hệ thống camera đồng bộ (hai camera

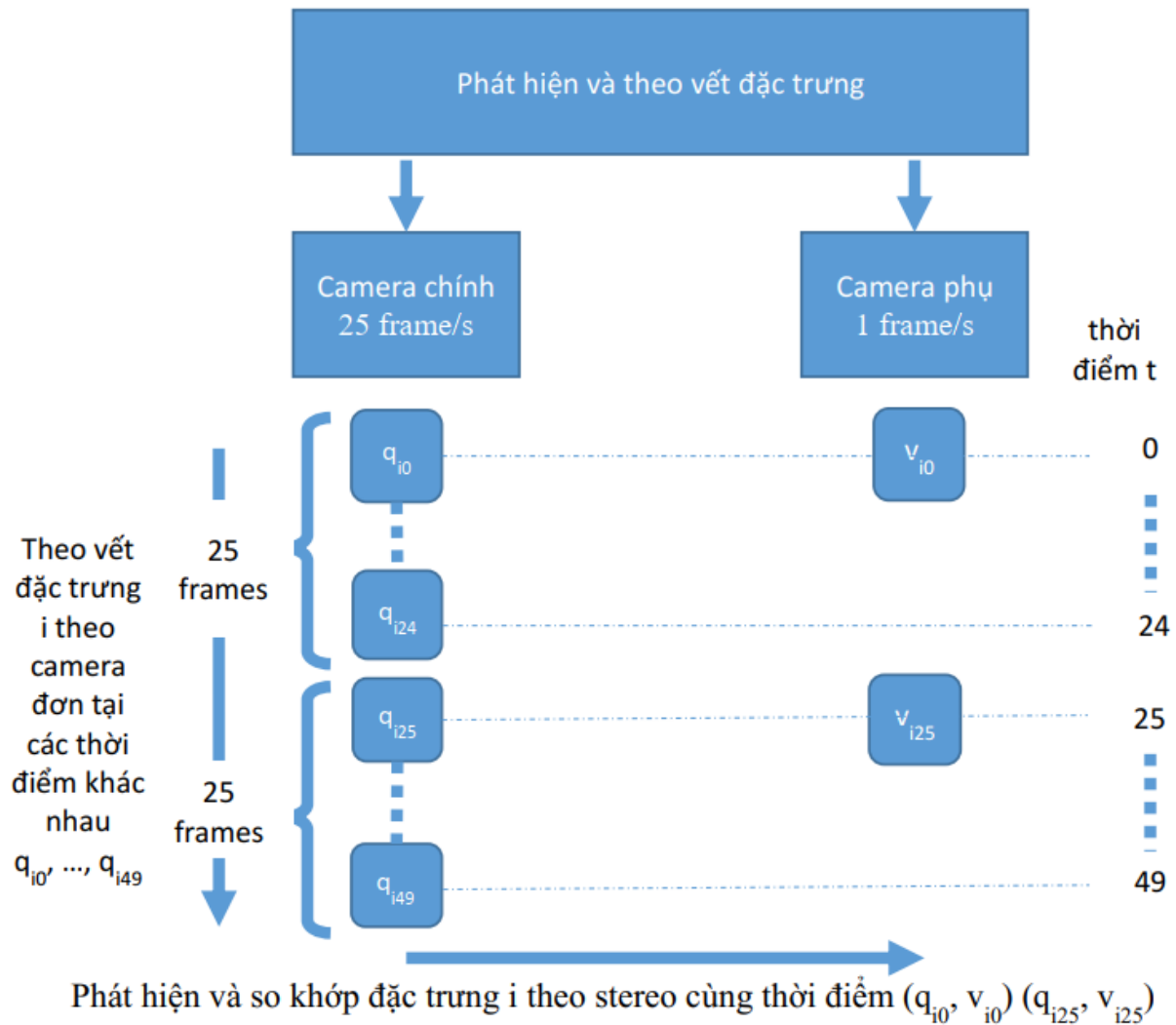
1Hz) (stereo). Vì hệ thống chủ yếu sử dụng đặc trưng để tính toán, hai hệ thống trên sẽ có cách tìm đặc trưng khác nhau. Với hệ thống camera đồng bộ là so khớp hai ảnh cùng thời điểm camera chính và phụ để tìm cặp đặc trưng tương ứng, hệ thống camera đơn là theo vết các đặc trưng theo từng thời điểm ở camera chính.

Hệ thống định vị và tái tạo bản đồ (Simultaneous Localization and Mapping - SLAM) gồm hai phần chính là định vị (Localization) và tái tạo bản đồ (Mapping) được thực hiện đồng thời tại mỗi thời điểm t . Tại thời điểm t , việc định vị được thực hiện trước dựa trên việc tạo bản đồ thời điểm $t-1$ trước đó. Sau khi có vị trí Robot, việc tái tạo bản đồ cho thời điểm hiện tại t được thực hiện. Vì việc định vị được thực hiện trước, nên sẽ có một pha dùng để tạo bản đồ khởi động cho hệ thống định vị và tái tạo bản đồ. Hệ thống gồm các bước chính: Phát hiện và theo vết đặc trưng, khởi động bản đồ, định vị vị trí, cập nhật bản đồ lân cận, phát hiện và phục hồi lỗi, kết hợp định vị bằng IMU.

→ Tóm lại, để biết vị trí hiện tại để biết vị trí hiện tại của vật ta cần biết phép biến đổi từ điểm p tại hệ tọa độ t sang p' tại hệ tọa độ t' .

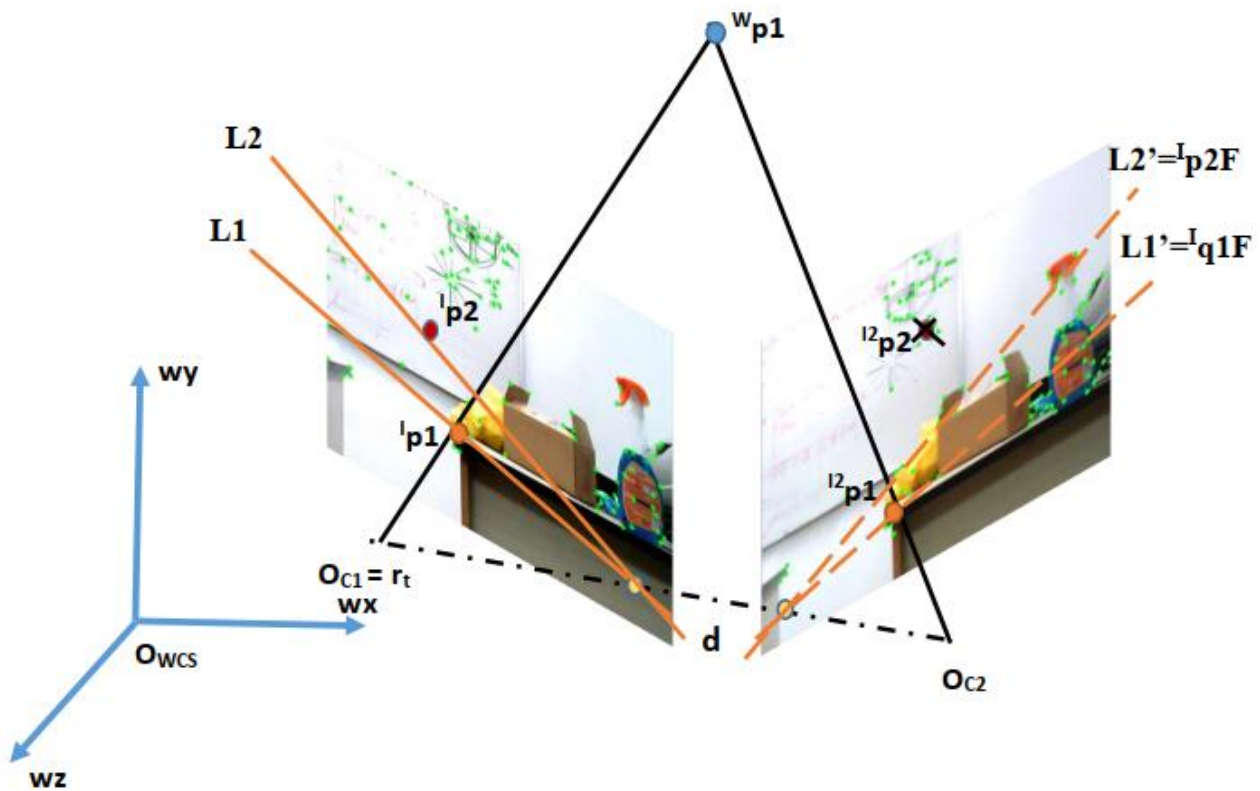
Để làm được điều đó ta cần đổi chiều hệ tọa độ của 2 frame ảnh tại 2 thời điểm trên để tìm ra 6 ẩn số để có 6 phương trình (3 cho phép quay và 3 cho phép tịnh tiến) để giải ra các tập điểm trên. Nhưng với IMU ta đã biết được phép quay nên ta chỉ cần tính cho phép tịnh tiến.

Ở phần này ta dùng 2 camera 1 phụ 1 frame/s và 1 chính 25 frame/s như hình:



Hình 2: Phát hiện và theo vết đặc trưng của 2 camera từng thời điểm

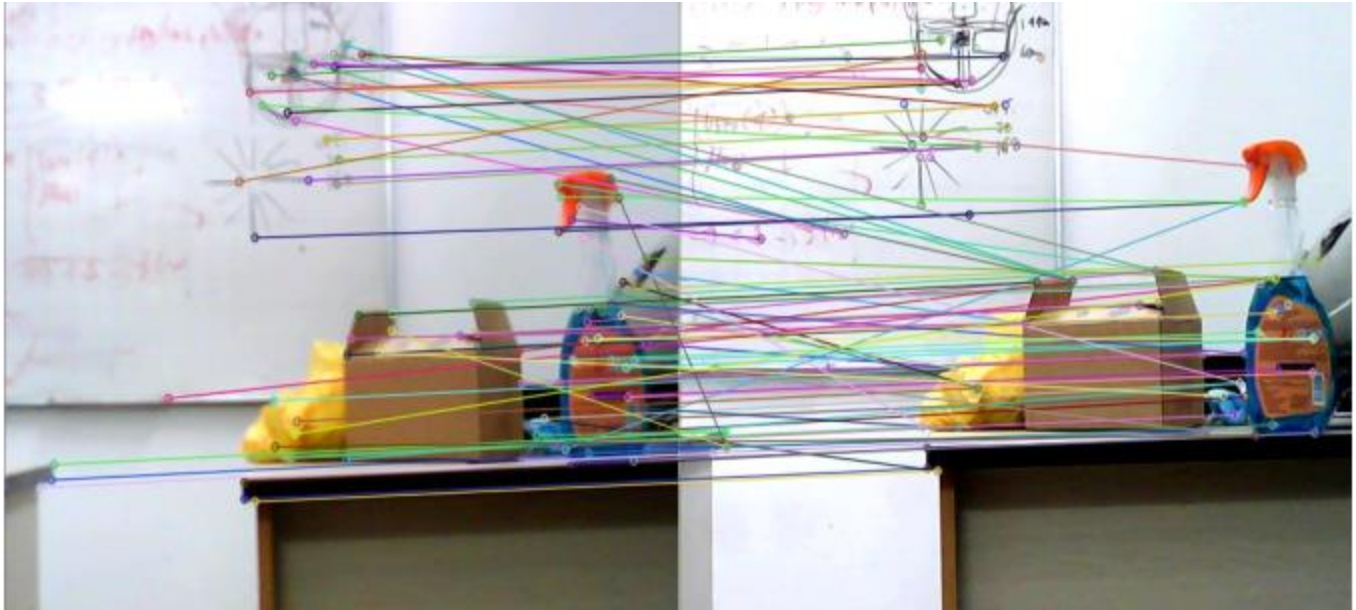
Để phát hiện đặc trưng ta sẽ dùng đặc trưng góc Shi-Tomasi để tiết kiệm chi phí hơn so với các đặc trưng SIFT, SURF... mà vẫn đảm bảo tính bất biến với một số phép affine. Và sau đó ta sẽ mô tả vùng xung quanh đặc trưng bằng các chuỗi bit (BRIEF escription) và BRIEF cũng bất biến với các phép affine.



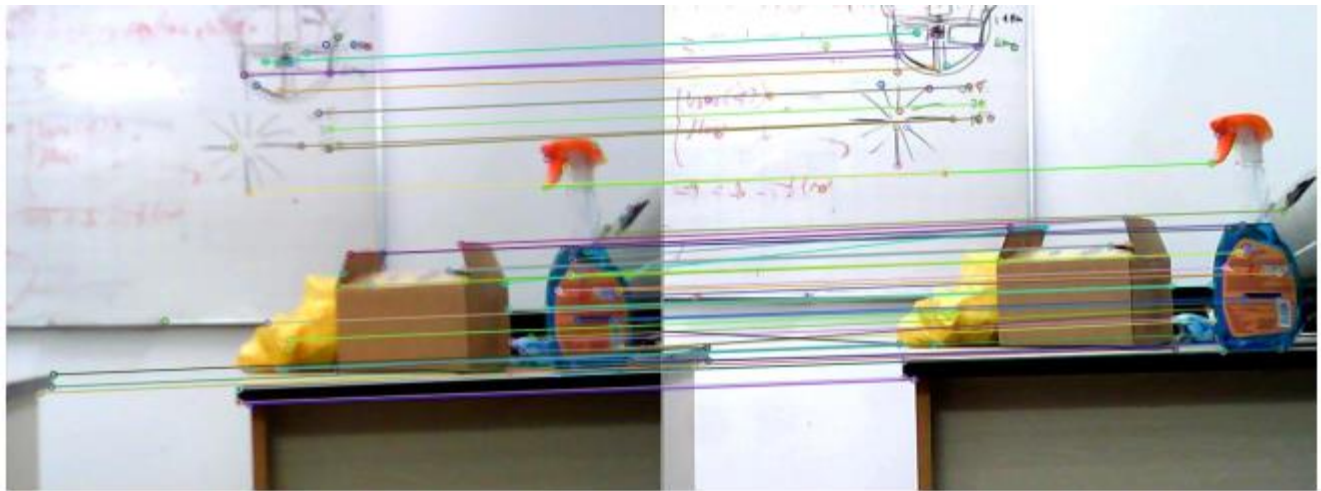
Hình 3 Các điểm màu xanh là đặc trưng góc Shi-Tomasi tại camera chính và phụ cùng thời điểm t . Điều kiện ràng buộc để thỏa mãn cặp tương đồng epipolar là một điểm phải nằm trên đường thẳng epipolar line tương ứng với điểm tương đồng. I_{2p1} nằm trên đường thẳng epipolar $L1'$ (màu cam) tương ứng với I_{p1} nên I_{p1} và I_{2p1} thỏa mãn điều kiện cặp tương đồng. Trong khi đó I_{2p2} không nằm trên $L2'$ là epipolar tương ứng của I_{p2} nên không phải là cặp tương đồng.

Sau khi tìm được đặc trưng thì ta sẽ đối sánh đặc trưng thông qua tìm kiếm láng giềng gần nhất dựa trên khoảng cách Hamming của các vector đặc trưng.

Hai đặc trưng của cùng một điểm 3D lên mặt phẳng ảnh thì thỏa mãn giới hạn Epipolar: $(I_{p1}) F(I_{2p1}) = 0$. Với F là fundamental matrix, I_{p1} và I_{2p1} tương ứng vị trí pixel của đặc trưng tại ảnh từ hai camera, với ý nghĩa $(I_{p1}) F$ là epipolar line $L1'$ trên mặt phẳng ảnh camera phụ mà I_{2p1} sẽ nằm trên đường thẳng này, nếu I_{2p1} không thuộc $L1'$ tích $(I_{p1}) F(I_{2p1})$ sẽ lớn và I_{p1} , I_{2p1} được xác định là nhiễu. Việc loại bỏ các điểm nhiễu (outlier) có thể được thực hiện bằng so sánh tích trên với một ngưỡng, ở đây là 0.1, nếu lớn hơn ngưỡng thì sẽ xác định là outlier và loại cặp điểm đó ra. Ý nghĩa của việc này là nhằm loại bỏ những cặp điểm bị nhầm lẫn trong quá trình so khớp do sai số bởi tính đồng nghĩa và đa nghĩa của đặc trưng thị giác.

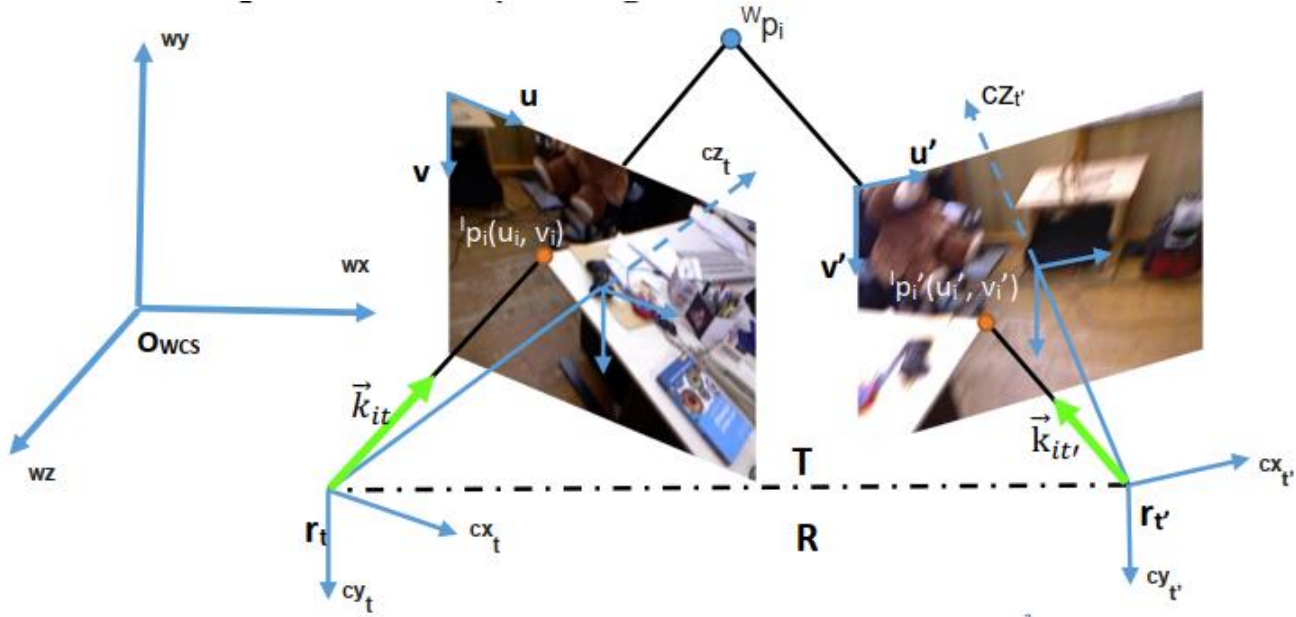


Hình 4: So khớp ảnh từ camera chính (phải) và camera phụ (trái) cùng 1 thời điểm



Hình 5: Cặp đặc trưng sau khi được lọc nhiễu trong việc so khớp

Với camera chính ta sử dụng phương pháp theo vết Kanade-Lucas-Tomasi Feature Tracker (KLT) để theo vết đặc trưng qua từng khung ảnh. Phương pháp dựa trên tìm kiếm một vùng (patch) xung quanh đặc trưng theo đạo hàm của nó. Do độ dịch chuyển pixel của đặc trưng giữa các khung hình là rất bé, nên KLT tốn rất ít chi phí.



Hình 6 Minh họa cho quan sát của đặc trưng W_{p_i} tại hai thời điểm t, t' trên mặt phẳng ảnh camera chính. Vector màu xanh vector k_{it} và vector, vector $k_{it'}$ lần lượt là quan sát của p_i tại t, t' . Các vector trên có hướng từ vị trí robot từng thời điểm ($r_t = O(C1)$ t) chỉ đến vị trí đặc trưng p_i trong tọa độ thế giới. Vector quan sát của đặc trưng có thể tính từ vị trí đặc trưng trên mặt phẳng ảnh camera chính

Các quan sát đặc trưng đóng vai trò rất quan trọng với hệ thống, mỗi đặc trưng sẽ cung cấp thông tin về hướng từ vị trí camera đến vị trí đặc trưng thông qua quan sát tại mặt phẳng ảnh từng thời điểm. Thông tin đó được dùng tìm lại vị trí camera. Cách tính được vector quan sát (unit feature observation) gồm các bước:

Quan sát của đặc trưng i từ hệ tọa độ mặt phẳng ảnh thuần nhất $I_{p_i}(u_i, v_i, 1)$ sẽ được chuyển thành vector quan sát ở hệ tọa độ Camera (C) k_{it}^c :

$$\vec{k}_{it}^c = K^{-1} I_{p_i}$$

Sau đó đưa về dạng vector đơn vị

$$\vec{k}_{it}^c = \frac{\vec{k}_{it}^c}{\|\vec{k}_{it}^c\|} \quad (3.1)$$

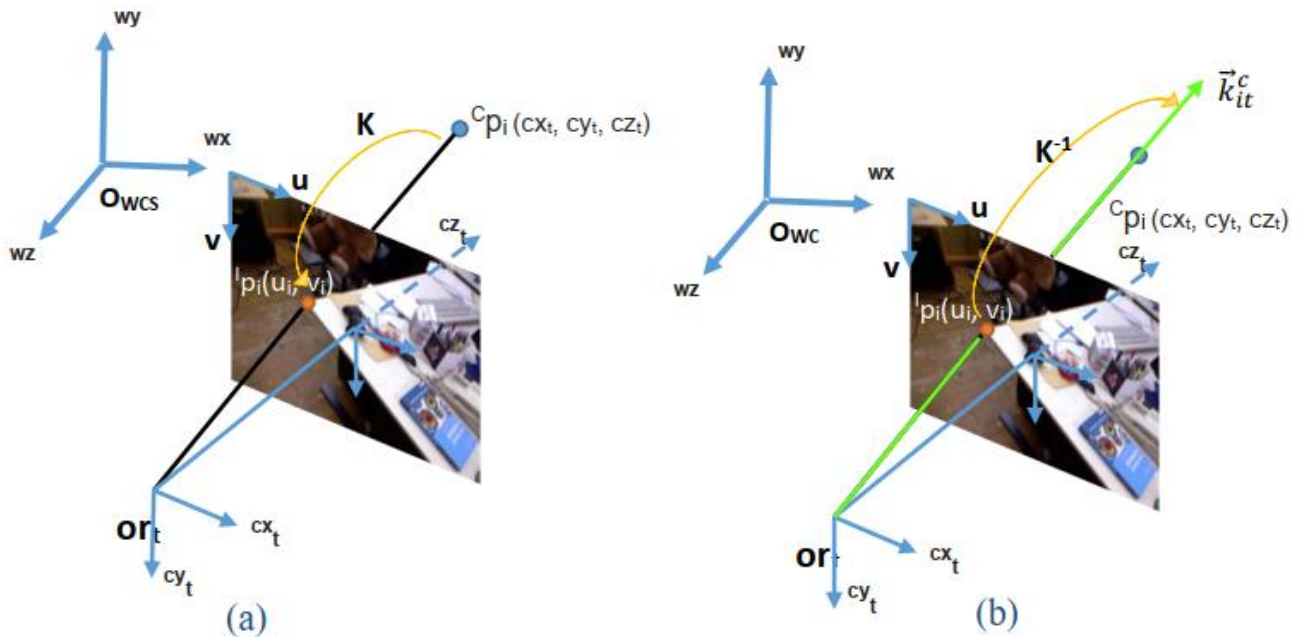
Với K là phép biến đổi từ hệ tọa độ camera (C) vào hệ tọa độ mặt phẳng ảnh:

với:

$$\mathbf{K} = \begin{pmatrix} f_u & 0 & u_c \\ 0 & f_v & v_c \\ 0 & 0 & 1 \end{pmatrix}$$

(u_c, v_c) : tâm chiếu $f_u = f_v = f$: tiêu cự theo chiều u, v

\mathbf{K} là ma trận camera chiếu điểm C_{pi} (c_x^t, c_y^t, c_z^t) thuộc tọa độ Camera xuống tọa độ mặt phẳng ảnh là $I_{pi}(u_i, v_i)$. Ma trận nghịch đảo của \mathbf{K} là \mathbf{K}^{-1} sẽ chiếu một điểm $p_i(u_i, v_i)$ trên mặt phẳng ảnh thành vector chỉ hướng đến C_{pi} trong hệ tọa độ camera (C). Sau đó vector chỉ hướng sẽ được chuyển thành vector đơn vị \vec{k}_{itC} .



Hình 7 a) Minh họa cho phép chiếu \mathbf{K} chiếu một điểm từ hệ tọa độ camera (C) C_{pi} xuống mặt phẳng ảnh. b) Minh họa cho phép chiếu \mathbf{K}^{-1} chiếu một điểm trên mặt phẳng ảnh thành vector có hướng từ tâm camera $O(C1)$ tới điểm 3D của nó ở tọa độ camera C_{pi} .

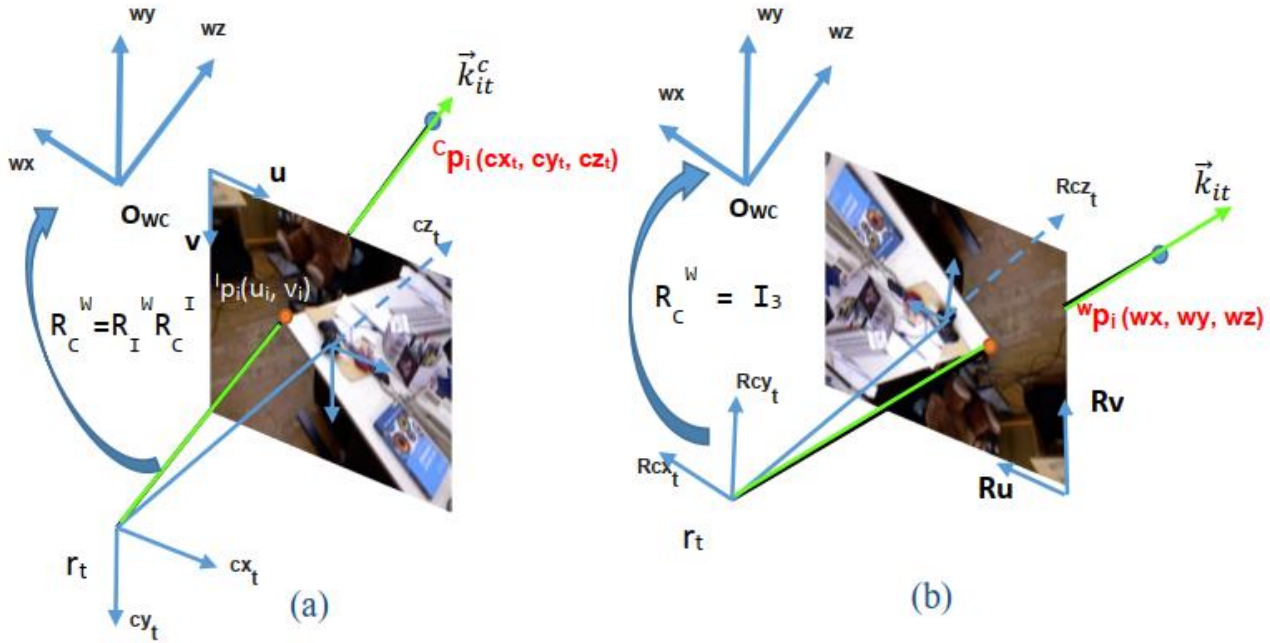
Chuyển hệ trục cho vector hướng \vec{k}_{itC} từ hệ tọa độ camera (C) sang tọa độ thế giới (W) \vec{k}_{it} :

$$\vec{k}_{it} = R_I^W R_C^I \vec{k}_{itC} \quad (3.2)$$

với:

- R_C^I : phép xoay từ camera \rightarrow IMU (có được khi cân chỉnh camera)
- R_I^W : phép xoay từ IMU \rightarrow world (W) (có được từ IMU tại mỗi thời điểm)

Ở đây, ta quy ước phép xoay $R_c^I = I_3$, tức là hệ trục camera đối với hệ trục IMU không chứa phép xoay. Nên công thức trên có thể viết thành: Vì \vec{k}_{itC} là vector hướng, nên việc chuyển sang hệ tọa độ thế giới chỉ cần phép xoay giữa hai hệ tọa độ. Nếu thêm phép dịch chuyển vào cũng sẽ cho ra vector có hướng cùng với việc chỉ thực hiện phép xoay. Ý nghĩa chuyển \vec{k}_{itC} sang hệ tọa độ thế giới: điều chỉnh hướng của vector từ tâm camera đến điểm 3D ở tọa độ camera C_{pi} sang từ tâm camera đến điểm 3D ở tọa độ thế giới W_{pi} . Lý do của việc này là vì cần tính tọa độ 3D của đặc trưng tại một hệ tọa độ chung là WCS.



Hình 8 a) Minh họa phép xoay từ hệ trục camera sang hệ trục thế giới. b) Khi áp dụng phép xoay vào, vector quan sát \vec{k}_{itC} trở thành \vec{k}_{it} và có hướng từ tâm camera qua vị trí 3D của đặc trưng thứ i trong hệ tọa độ thế giới.

Sau bước đối sánh đặc trưng, hệ thống đã có được tập đặc trưng qit ở ảnh camera chính và tập tương ứng vit ảnh từ camera phụ. Việc tính toán vị trí 3D được thực hiện theo Linear Triangulation Methods, đây là một cách tính đơn giản được Hartley và Zisserman đề xuất.

Trong quá trình cân chỉnh camera, ta đã biết được phép chiếu từ không gian 3D theo hệ tọa độ camera chính xuống ảnh ở hai camera chính và phụ là P và P' : $q_{it} = P C_{pit}$, $v_{it} = P' C_{pit}$ với C_{pit} là tọa độ 3D theo hệ trục camera chính, q_{it} , v_{it} là hình chiếu xuống mặt phẳng ảnh camera chính và phụ của X . Từ hai phương trình trên, có thể viết lại thành dạng tuyến tính theo X : $AX = 0$ và giải theo phương pháp cực tiểu bình phương (least square) SVD để tìm X .

$$A^{C_{p_{it}}} = \begin{bmatrix} uP^{3T} - P^{1T} \\ vP^{3T} - P^{2T} \\ u'P'^{3T} - P'^{1T} \\ v'P'^{3T} - P'^{2T} \end{bmatrix} C_{p_{it}} = 0 \quad (3.3)$$

Sau khi tìm được vị trí 3D $C_{p_{it}}$ của các cặp đặc trưng tương ứng qit, vit với hệ thống camera đồng bộ (stereo), ta sẽ chuyển sang hệ trục tọa độ thế giới WCS. Vì định nghĩa vị trí camera chính thời điểm thứ $t = 0$ cũng là vị trí robot thời điểm $t = 0$ là $r_0 = (0, 0, 0)$. Nên ta chỉ cần áp dụng phép xoay có được từ IMU để chuyển hệ tọa độ các điểm trên $C_{p_{it}}$ từ hệ trục tọa độ camera chính (C) sang hệ trục tọa độ thế giới WCS:

$${}^W p_i = R_I^W R_C^I C_{p_{it}} \quad (3.4)$$

Như đã nói ở trên, $R_C^I = I3$ nên ta chỉ cần áp dụng phép xoay từ IMU sang hệ tọa độ thế giới có được ở mỗi thời điểm. Các vị trí này sẽ được thêm vào bản đồ lân cận cùng với quan sát của nó như là khởi tạo.

Vị trí của robot sẽ được tính bằng cực tiểu hóa độ lỗi bình phương phép reprojection của tất cả feature trong bản đồ 3D đặc trưng lân cận:

$$r_t = \underset{r_t}{\operatorname{argmin}} \sum_{i \in \mathfrak{S}} \left\| \frac{r_t - {}^W p_i}{\|r_t - {}^W p_i\|} \times \vec{k}_{it} \right\|^2 \quad (3.5)$$

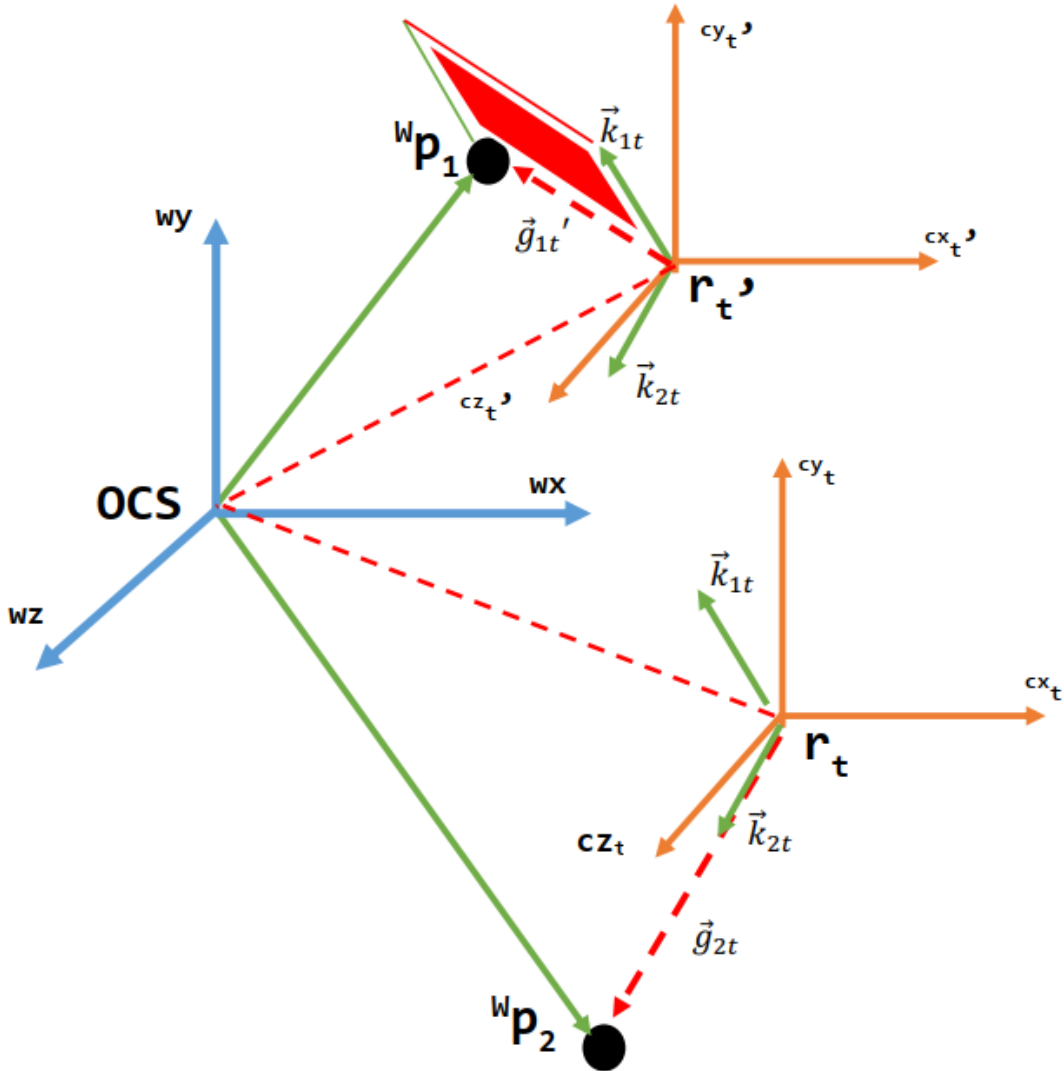
Diễn giải:

- \vec{k}_{it} là vector quan sát của đặc trưng thứ i thời điểm t , cũng vector nối từ đặc trưng i đến r_t .
- $\frac{r_t - {}^W p_i}{\|r_t - {}^W p_i\|}$ ký hiệu \vec{g}_{it} là vector đơn vị thật sự khi có vị trí chính xác của r_t .
- \mathfrak{S} là tập các đặc trưng quan sát được tại ảnh camera chính thời điểm t .

Tích có hướng giữa $\vec{g}_{it} \times \vec{k}_{it}$ tỷ lệ với diện tích hình bình hành tạo ra từ 2 vector. Do đó, khi góc giữa hai vector là bé thì tích này sẽ bé và ngược lại. Mục đích ở đây là cực tiểu sai số giữa các quan sát (\vec{k}_{it}) và vector thật sự (\vec{g}_{it}) khi tìm được r_t bằng cách tìm vị trí r_t trong không gian để có \vec{g}_{it} có hướng gần nhất với \vec{k}_{it} đã biết.

Lưu ý rằng, do độ dịch chuyển camera giữa các khung ảnh là rất bé, nên ta có thể tính xấp xỉ:

$$d_i = \|r_t - {}^w p_i\| \approx \|r_{t-1} - {}^w p_i\| \quad (3.6)$$



Hình 9 Minh họa các vị trí camera thời điểm t và sai số so với vector quan sát k_{it} . Vị trí có sai số nhỏ nhất (tổng diện tích hình bình hành nhỏ nhất) sẽ là vị trí robot thời điểm t . Ở đây là vị trí r_t có sai số nhỏ nhất so với vector quan sát được.

Có r_t và r'_t là các điểm trong vô số điểm của camera có thể trong không gian tại thời điểm t , ta có W_{p1} và W_{p2} là các vị trí 3D của đặc trưng và k_{1t} , k_{2t} là các quan sát của đặc trưng tại thời điểm t . Mục đích là tìm r_t sao cho tất cả vector $= g_{it}$ có hướng gần với vector k_{it} tương ứng nhất.

$$\cdot \frac{r_t - {}^w p_i}{\|r_t - {}^w p_i\|}$$

Khi chọn camera ở vị trí r_t' (sai) thì ta sẽ tính được vector g_{1t}' (mũi tên nét đứt đỏ) và tích vô hướng với k_{1t}' sẽ ra giá trị độ lớn tương ứng phần hình bình hành màu đỏ lớn (diện tích càng lớn \rightarrow độ lỗi càng nhiều).

Khi chọn camera ở vị trí r_t (đúng) thì vector g_{2t} (mũi tên nét đứt đỏ) nhân tích vô hướng với k_{2t} sẽ ra giá trị độ lỗi gần bằng 0 (do hướng đúng, SBìnhHành = 0). Do đó sai số tại vị trí r_t là bé nhất nên vị trí cần tìm sẽ là r_t .

Từ (3.6) chuyển về dạng tuyến tính tìm r_t :

$$\left(\sum_{i \in \mathfrak{S}} \frac{\mathbb{I}_3 - \vec{k}_{it} \vec{k}_{it}^T}{d_i} \right) r_t = \sum_{i \in \mathfrak{S}} \frac{\mathbb{I}_3 - \vec{k}_{it} \vec{k}_{it}^T}{d_i} w p_i \quad (3.7)$$

Với:

- $d_i \approx \|r_{t-1} - w p_i\|$: giá trị đã biết
- \mathfrak{S} : là tập các đặc trưng quan sát được tại ảnh camera chính thời điểm t .
- $w p_i$: vị trí 3D của đặc trưng thứ i hệ trục tọa độ thế giới WCS.
- r_t : giá trị cần tìm, vị trí robot thời điểm t .

Để tìm vị trí robot thời điểm hiện tại t , ta sử dụng công thức tuyến tính (3.7) tính toán với tập \mathfrak{S} gồm các đặc trưng dùng để tính:

$$\left(\sum_{i \in \mathfrak{S}} \frac{\mathbb{I}_3 - \vec{k}_{it} \vec{k}_{it}^T}{d_i} \right) r_t = \sum_{i \in \mathfrak{S}} \frac{\mathbb{I}_3 - \vec{k}_{it} \vec{k}_{it}^T}{d_i} w p_i \quad (3.8)$$

Để đánh giá chất lượng của vị trí vừa ước lượng được, ta sẽ dựa vào hai tiêu chí sau:

- Số lượng các đặc trưng thỏa vị trí robot hiện tại (càng lớn càng tốt)
- Căn bậc hai trung bình bình phương sai số (root mean squared error) khi lấy độ lớn tích vô hướng của vector quan sát k_{it} và vector thực sự g_{it} sử dụng vị trí robot này (càng nhỏ càng tốt).

Việc sử dụng cả hai tiêu chí thay vì chỉ xét đến sai số giữa hai tập điểm là cần thiết. Vì quá trình tính sai số chỉ xét đến các điểm thỏa phép biến đổi nên sai số ít 34 không chắc chắn rằng đây sẽ là một phép biến đổi tốt. Một phép biến đổi được xem là đủ tốt nếu số lượng cặp

điểm thỏa phép biến đổi lớn hơn 20 và trung bình sai số ít hơn 5 cm. Việc so sánh giữa hai phép biến đổi sẽ dựa vào cả hai tiêu chí này. Phép biến đổi nào tốt hơn ở cả hai tiêu chí thì mới được xem là tốt hơn.

Sau đó ta sẽ cập nhật lại bản đồ lân cận (bản đồ vị trí 3D của các đặc trưng vẫn còn quan sát được trên ảnh camera chính, tức là các đặc trưng chưa bị mất dấu trong quá trình theo vết) để hoàn thành, bước này gồm: thêm đặc trưng, xóa đặc trưng

*Với thêm đặc trưng

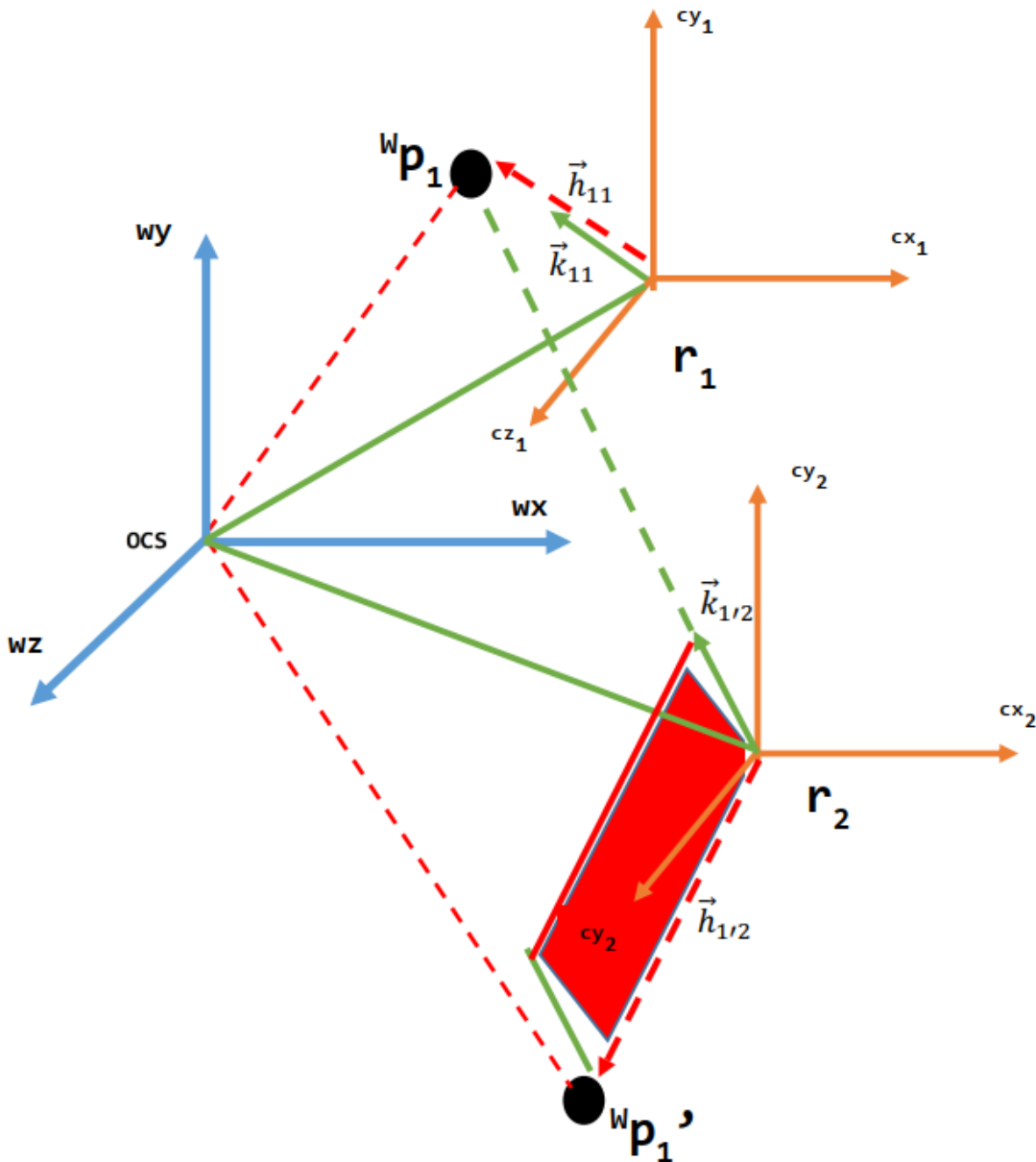
Với hệ thống camera đồng bộ (stereo):

- Với mỗi thời điểm có hai ảnh từ hai camera chính và phụ, hệ thống camera đồng bộ sẽ hoạt động, tìm và xác định vị trí 3D của các đặc trưng vừa tìm được bằng giải thuật Linear Triangulation Methods. Sau khi đã có vị trí 3D của các đặc trưng từ hệ thống stereo, cập nhật bản đồ lân cận với các điểm trong bản đồ là vị trí 3D của các đặc trưng tìm được sau khi đã biến đổi sang hệ tọa độ thế giới bởi vị trí robot thời điểm hiện tại vừa tính được. Các đặc trưng này được lưu với quan sát của nó nhằm tính toán cho các khâu tiếp theo.
- +Với hệ thống monocular: Khi một đặc trưng có quan sát tương ứng tại cả hai camera (hệ thống camera đồng bộ) thì dễ dàng tính được vị trí và thêm vào bản đồ lân cận. Nhưng nhiều đặc trưng chỉ có quan sát tại camera chính (hệ thống camera đơn), việc cập nhật bản đồ lân cận cho hệ thống camera đơn chính là đi tìm vị trí 3D cho các đặc trưng như vậy dựa vào vị trí robot và quan sát của các đặc trưng các thời điểm tương ứng.

$${}^w p_i = \underset{{}^w p_i}{\operatorname{argmin}} \sum_{t \in \tau} \|({}^w p_i - r_t) \times \vec{k}_{it}\|^2 \quad (3.9)$$

Với τ là tập hợp các thời điểm t mà đặc trưng thứ i quan sát được trên camera chính.

- $W_{pi} - rt$ ký hiệu là \vec{h}_{it} là vector tính được khi có vị trí 3D của đặc trưng. Có hướng đi từ tâm camera thời điểm t đến vị trí 3D của đặc trưng i



Hình 10: Minh họa các vị trí đặc trưng i thời điểm 1, 2 và sai số so với vector quan sát $\vec{k}_{i1}, \vec{k}_{i2}$. Vị trí có sai số nhỏ nhất (tổng diện tích hình bình hành đỏ nhỏ nhất) sẽ là vị trí đặc trưng ở tọa độ thể giới. Ở đây là vị trí W_{p1} có sai số nhỏ nhất so với vector quan sát được

Đặc trưng thứ 1 có quan sát ở camera chính tại 2 thời điểm r_1 và r_2 :

+ Nếu chọn vị trí đặc trưng 1 là W_{p1} (đúng) thì vector quan sát được (\vec{k}_{11} và \vec{k}_{12} (mũi tên đứt màu xanh) sẽ cùng phương với vector thực tế ($W_{p1} - r_1 = \vec{h}_{11}$ và $W_{p1} - r_2 = \vec{h}_{12}$ (mũi tên đứt màu đỏ) \rightarrow tích vô hướng gần bằng 0 (ShìnhBinhHành ≈ 0))

+ Nếu chọn đặc trưng 1 vị trí là W_{p1} (sai) thì vector quan sát ($k_{1/2}$ quan sát mũi tên liên màu xanh) và dự đoán ($W_{p1} - r_2 = h_{1/2}$ quan sát mũi tên liên màu) không cùng hướng, sẽ tạo ra tích vô hướng lớn (thể hiện qua diện tích hình bình hành đỏ).

Chọn vị trí W_{pi} vì tổng tích vô hướng các quan sát được nhỏ nhất (tổng diện tích độ lỗi (ShìnhBìnhHành) bé nhất).

Chuyển về dạng tuyến tính W_{pi} :

$$\left(\sum_{t \in \tau} \mathbb{I}_3 - \vec{k}_{it} \vec{k}_{it}^T \right) W_{pi} = \sum_{t \in \tau} (\mathbb{I}_3 - \vec{k}_{it} \vec{k}_{it}^T) r_t \quad (3.10)$$

Với:

- τ : tập hợp các thời điểm quan sát được đặc trưng thứ i trên mặt phẳng ảnh camera chính.
- W_{pi} : vị trí 3D đặc trưng i theo hệ tọa độ thế giới (W).
- r_t : vị trí camera thời điểm t theo (W).

Việc cập nhật như vậy khiến hệ thống SLAM ít phụ thuộc vào đặc trưng từ hệ thống Stereo (1Hz), giúp hệ thống Stereo có thể chạy với tần số thấp hơn mà vẫn đảm bảo tính liên tục của hệ thống đồng thời khắc phục khuyết điểm của Stereo là có tầm nhìn hạn chế (chỉ tính được vị trí 3D của các điểm trong khoảng cách gần). Đồng thời vì kết hợp hệ thống Stereo, hệ thống SLAM còn giúp khắc phục khuyết điểm của Monocular (chỉ biết tỷ lệ khoảng cách, không biết khoảng cách chính xác).

Do các đặc trưng theo vết chưa hẳn sẽ có điểm đối sánh tại ảnh camera phụ, nên các đặc trưng có thể sẽ không có vị trí 3D tính toán được từ hệ thống camera đồng bộ. Việc cập nhật sẽ phân thành:

- Với các đặc trưng đã có vị trí 3D từ hệ thống camera đồng bộ: thêm vào vị trí 3D từ hệ thống monocular.
- Với các đặc trưng chưa có vị trí 3D từ camera đồng bộ: chỉ thêm các đặc trưng vào bản đồ nếu số lượng các đặc trưng trong bản đồ xuống dưới một ngưỡng cho phép

*Với xóa đặc trưng: Nếu hệ thống theo vết trên chuỗi ảnh từ camera chính bị mất dấu đặc trưng (lost tracking), những đặc trưng như thế sẽ không còn có quan sát (từ ảnh) vì vậy sẽ

được xóa khỏi bản đồ lân cận và thêm vào những đặc trưng mới được phát hiện nhằm bảo đảm tính liên tục cho hệ thống.

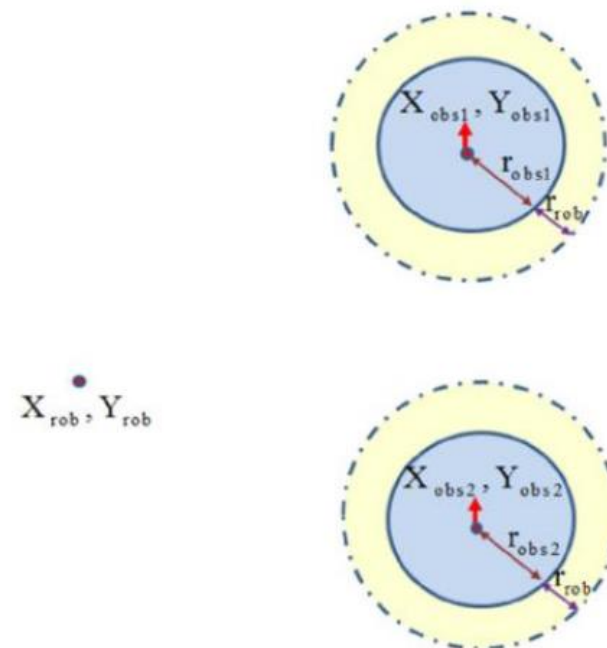
2. Tránh vật cản tĩnh

Để tránh vật cản thì chúng em sẽ sử dụng phương pháp follow the gap. Phương pháp Follow the Gap dựa trên việc xây dựng một vòng tròn bán kính bao xung quanh vật thể và tính toán góc hướng tốt nhất để đưa robot vào khoảng trống cực đại của các vật cản xung quanh, đồng thời xem xét điểm mục tiêu.

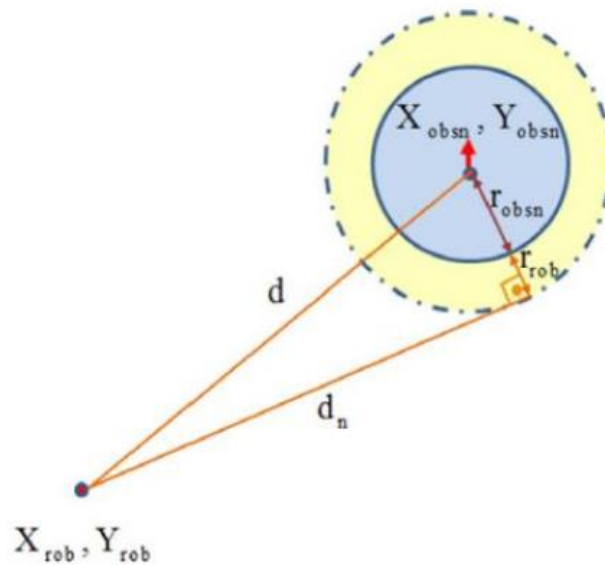
Thuật toán chia làm 3 bước chính:



Robot được coi là một điểm và mở rộng bán kính của vật cản ra nhằm đảm bảo tính chính xác:

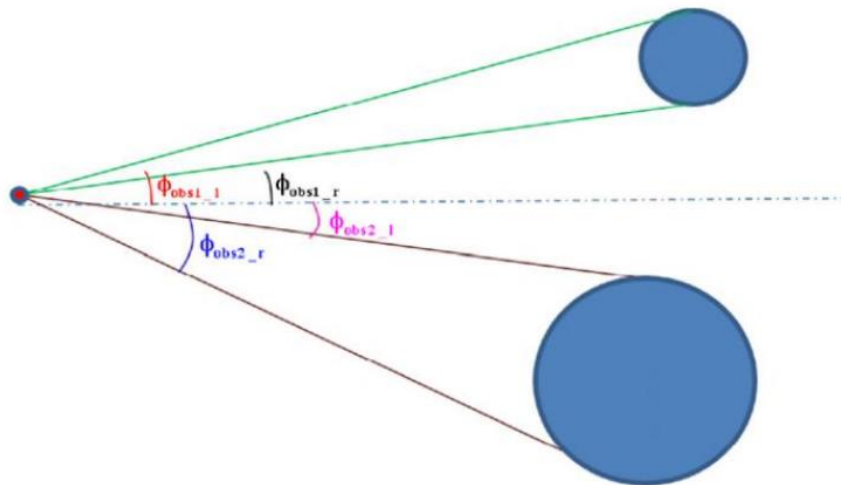


Khoảng cách từ robot đến vật cản:

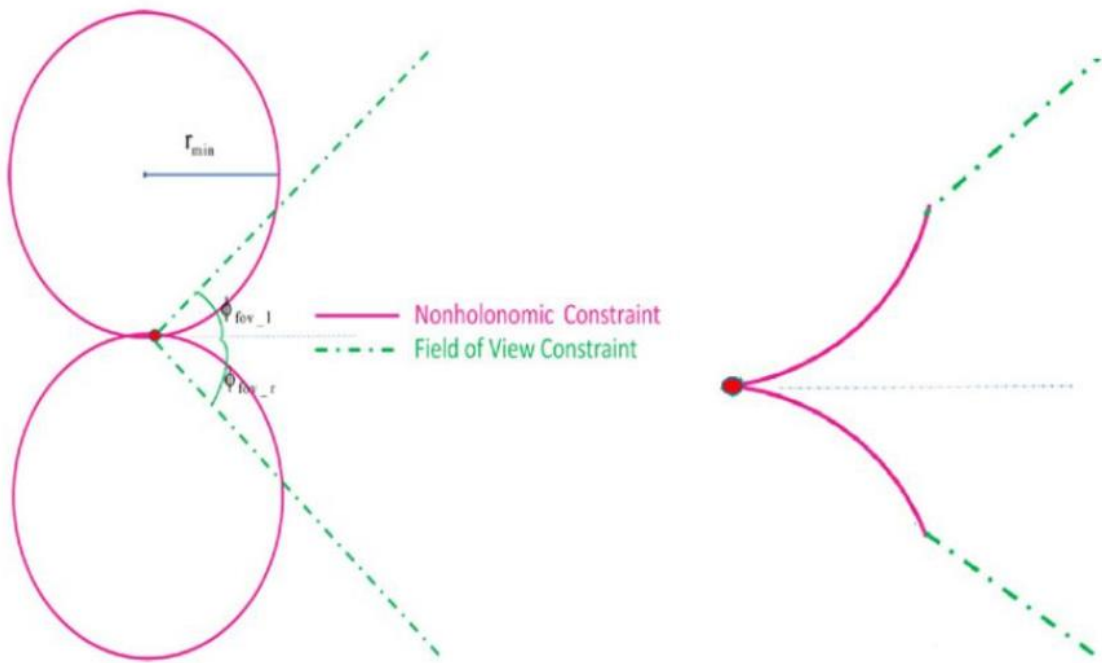


Bước 1: Tính toán mảng khoảng cách và tìm khoảng cách tối đa

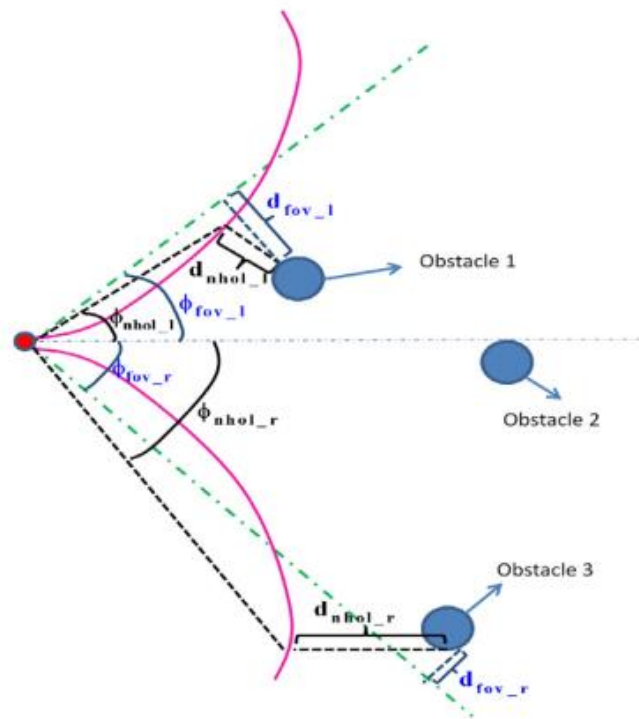
Để tính được thì ta cần các giá trị góc đường biên (ϕ_{obs1_l} và ϕ_{obs1_r} lần lượt biểu thị các giá trị góc biên trái và phải của chướng ngại vật thứ i) và hai giá trị đường viền ngoài đường viền của chướng ngại vật.



Hình góc đường biên



Hình đánh giá biên khoảng trống



Hình tham số đường viền khe hở

ϕ_{nhol} : Góc biên đến từ ràng buộc không đơn sắc. (ϕ_{nhol_l} : Dành cho bên trái xe. ϕ_{nhol_r} : Dành cho bên phải xe.)

ϕ_{fov} : Góc biên đến từ trường nhìn. (ϕ_{fov_l} : Đối với phía bên trái của xe. ϕ_{fov_r} : Đối với phía bên phải của phương tiện.)

d_{nhol} : Khoảng cách gần nhất giữa cung giới hạn không độc lập và biên giới chương ngại vật. (d_{nhol_l} : Dành cho bên trái xe. d_{nhol_r} : Dành cho bên phải xe.)

d_{fov} : Khoảng cách gần nhất giữa trường nhìn và đường viền chương ngại vật. (d_{fov_l} : Dành cho bên trái xe. d_{fov_r} : Dành cho bên phải xe.)

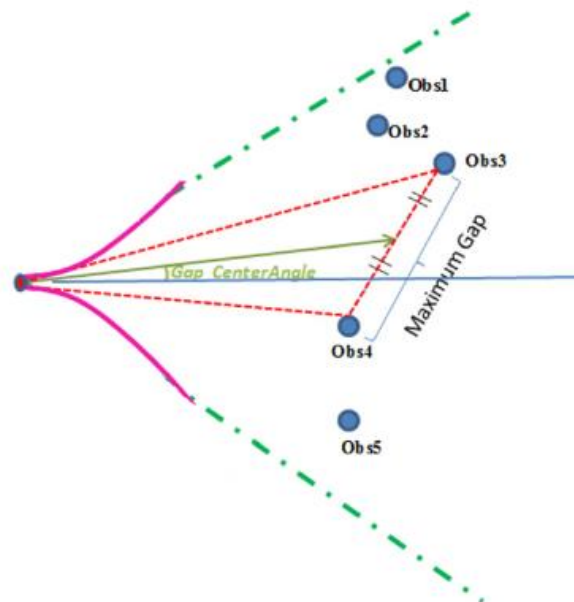
Sau khi biểu diễn đường viền khoảng cách và đường viền chương ngại vật, mảng khoảng cách có thể được tạo. Phải có $N + 1$ khoảng trống cho N chương ngại vật. $N+1$ phần tử của mảng khoảng cách được minh họa bên dưới theo các định nghĩa trước đó. Với công thức:

$$Gap[N + 1] = [(\phi_{lim_l} - \phi_{obs1_l})(\phi_{obs1_r} - \phi_{obs2_l}) \cdots (\phi_{obs(n-1)_r} - \phi_{obs(n)_l})(\phi_{obs(n)_r} - \phi_{lim_r})]. \quad ($$

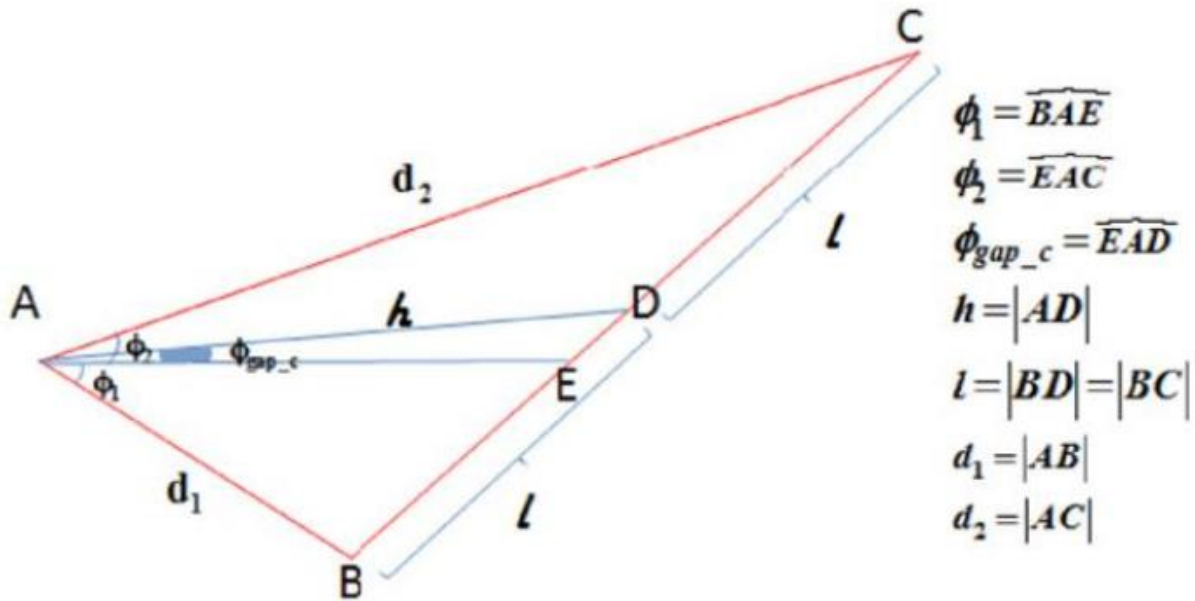
Khoảng cách cực đại là mức cực đại của các phần tử mảng khoảng cách và được chọn cho bước thứ hai của thuật toán. Nếu tồn tại nhiều hơn một khoảng cách cực đại có cùng giá trị, thuật toán sẽ chọn giá trị được tính toán đầu tiên.

Bước 2: Tính toán góc tâm khe hở

Ở bước này ta cần tìm góc ở tâm khe hở (ϕ_{gap_c}) là góc của vectơ trung tuyến từ robot đến đường giữa các chương ngại vật tạo/gây ra khoảng cách cực đại



Hình góc ở tâm khe hở



Hình tham số hóa góc tâm khe hở

Mục đích là để tìm ϕ_{gap_c} theo các tham số d_1, d_2, ϕ_1, ϕ_2 có thể đo được như trong hình trên. (d_1, d_2 là khoảng cách đến chướng ngại vật tính từ khe hở tối đa. ϕ_1, ϕ_2 là các góc của chướng ngại vật so với khoảng cách cực đại.)

Các bước:

Cosin cho tam giác ABC

$$(2l)^2 = d_1^2 + d_2^2 - 2d_1d_2 \cos(\phi_1 + \phi_2)$$

$$l^2 = \frac{d_1^2 + d_2^2 - 2d_1d_2 \cos(\phi_1 + \phi_2)}{4} \quad (1)$$

Apollonius cho tam giác ABC

$$d_1^2 + d_2^2 = 2l^2 + 2h^2 \quad (2)$$

Thay l vào (1)

$$h^2 = \frac{d_1^2 + d_2^2 + 2d_1d_2 \cos(\phi_1 + \phi_2)}{4} \quad (3)$$

Cosin cho tam giác ABD

$$l^2 = d_1^2 + h^2 - 2d_1h \cos(\phi_1 + \phi_{gap_c}). \quad (4)$$

Thay l, h với các phương trình (1), (3):

$$\begin{aligned} & \frac{d_1^2 + d_2^2 - 2d_1d_2 \cos(\phi_1 + \phi_2)}{4} \\ &= d_1^2 + \frac{d_1^2 + d_2^2 + 2d_1d_2 \cos(\phi_1 + \phi_2)}{4} \\ & \quad - 2d_1 \frac{\sqrt{d_1^2 + d_2^2 + 2d_1d_2 \cos(\phi_1 + \phi_2)}}{2} \cos(\phi_1 + \phi_{gap_c}) \\ \phi_{gap_c} &= \arccos \left(\frac{d_1 + d_2 \cos(\phi_1 + \phi_2)}{\sqrt{d_1^2 + d_2^2 + 2d_1d_2 \cos(\phi_1 + \phi_2)}} \right) - \phi_1. \end{aligned} \quad (5)$$

Góc trung tâm khoảng cách có thể được tìm thấy bằng cách tính trung bình của các góc ϕ_1 và ϕ_2 bỏ qua các giá trị khoảng cách d_1 và d_2 :

$$\phi_{gap_c_basic} = \frac{\phi_1 + \phi_2}{2}. \quad (6)$$

Bước 3: Tính toán góc tiêu đề cuối cùng

Góc cuối cùng là sự kết hợp của góc giữa khoảng trống và góc đích. Cấu trúc tổ hợp phụ thuộc vào khoảng cách cực tiểu đến các chướng ngại vật xung quanh và hệ số trọng lượng. Nếu các chướng ngại vật ở gần robot, nó nên xem xét sự an toàn trước tiên; tức là góc giữa khoảng trống, chứ không phải là góc mục tiêu và ngược lại. Hàm hợp nhất bên dưới mô tả phép tính góc cuối cùng:

$$\phi_{final} = \frac{\frac{\alpha}{d_{min}} \phi_{gap_c} + \beta \phi_{goal}}{\frac{\alpha}{d_{min}} + \beta} \quad \text{where } d_{min} = \min_{i=1:n}(d_n). \quad (7)$$

ϕ_{gap_c} : Góc tâm khoảng trống.

ϕ_{goal} : Góc khung thành.

α : Hệ số trọng lượng cho khoảng cách.

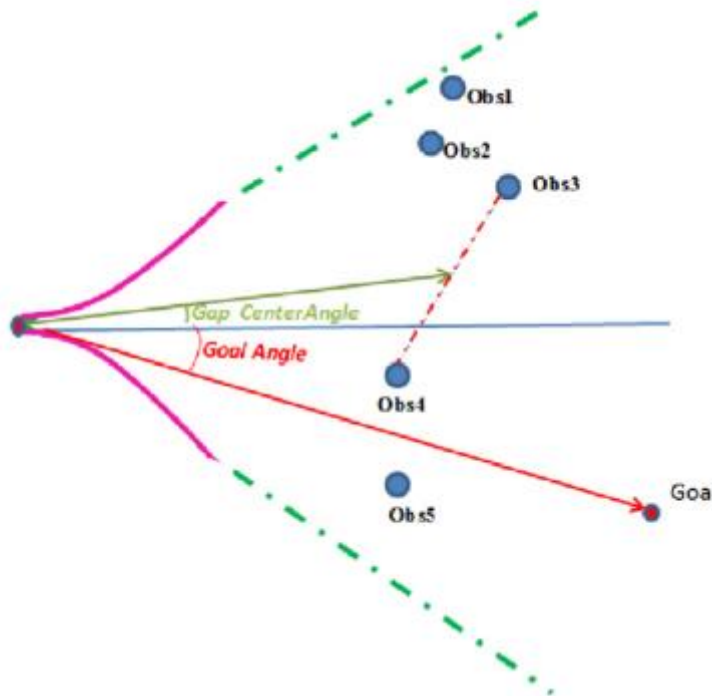
β : Hệ số trọng lượng cho mục tiêu.

n: Số chướng ngại vật. d_n : Khoảng cách đến chướng ngại vật thứ n.

d_{\min} : Giá trị khoảng cách cực tiểu của d_n

Để đơn giản β được lấy là "1" và hệ số α được sử dụng làm hệ số trọng số giữa góc giữa khoảng cách và góc khung thành. Thì (7) thành:

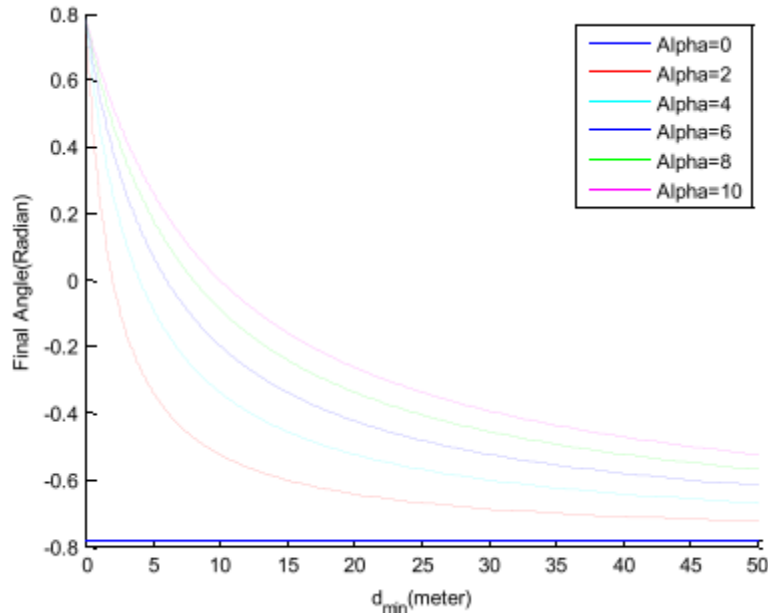
$$\varphi \text{ thức} = \frac{\frac{\alpha}{d_{\min}} \varphi_{\text{gap_c}} + \varphi_{\text{mục tiêu}}}{\frac{\alpha}{d_{\min}} + 1} \quad \text{trong đó } d_{\min} = \min_{i=1:n} (d_n).$$



Hình vật cản và đích

Góc cuối cùng tiến đến góc khoảng cách khi khoảng cách đến chướng ngại vật gần bằng không

$$\begin{aligned} \lim_{d_{\min} \rightarrow 0} \phi_{\text{final}} &\cong \frac{\infty}{\infty} \Rightarrow \lim_{d_{\min} \rightarrow 0} \phi_{\text{final}} \\ &\cong \frac{\frac{\partial \left(\frac{\alpha}{d_{\min}} \phi_{\text{gap_c}} + \beta \phi_{\text{goal}} \right)}{\partial d_{\min}}}{\frac{\partial \left(\frac{\alpha}{d_{\min}} + \beta \right)}{\partial d_{\min}}} = \frac{\frac{-\alpha \phi_{\text{gap_c}}}{d_{\min}^2}}{\frac{-\alpha}{d_{\min}^2}} = \phi_{\text{gap_c}}. \end{aligned}$$

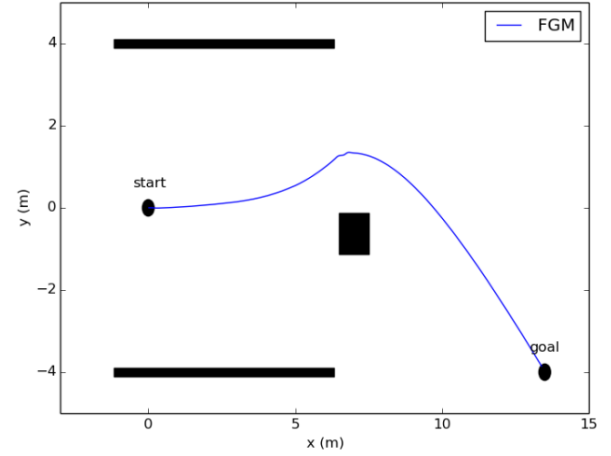
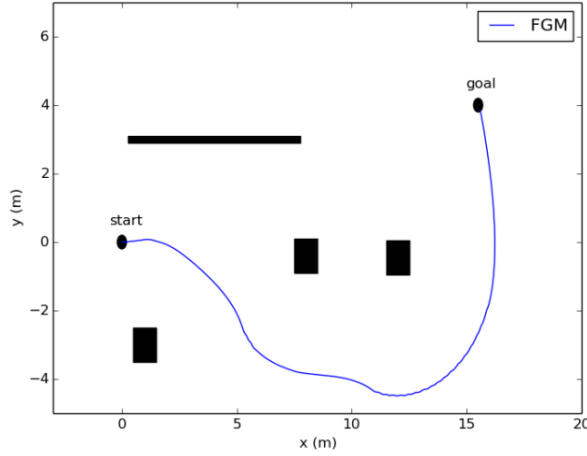


Hình góc cuối cùng đối với khoảng cách cực tiểu và hệ số α ($\varphi_{gap_c} = \pi/4$ radian và $\varphi_{goal} = \pi/4$ radian).

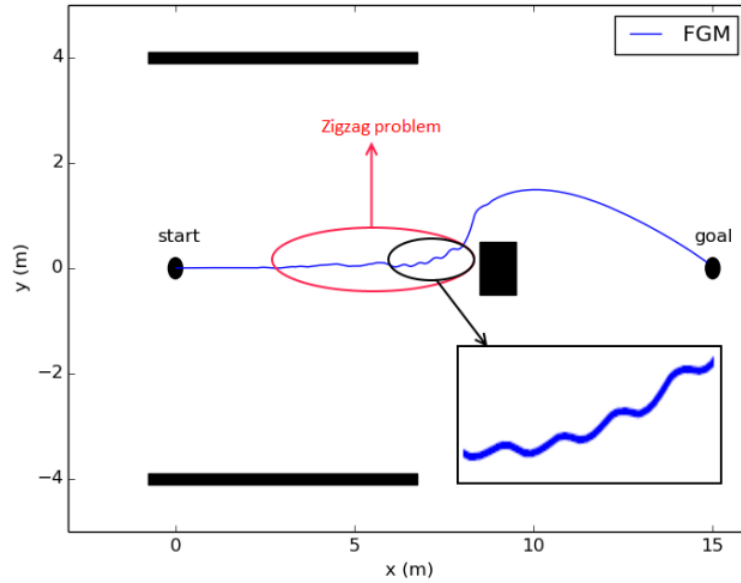
Đối với hình này, $\varphi_{gap_c} = \pi/4$ và $\varphi_{goal} = \pi/4$ có nghĩa là tâm của khoảng cách và mục tiêu nằm ở các hướng khác nhau. φ_{final} hội tụ đến $\varphi_{gap_c}(\pi/4)$ để giảm giá trị của d_{min} (chướng ngại vật đang tiến đến gần rô-bốt) và φ_{final} hội tụ đến $\varphi_{goal}(-\pi/4)$ để tăng giá trị của d_{min} (chướng ngại vật đang di chuyển ra xa). Giá trị α xác định mức độ định hướng mục tiêu hoặc định hướng khoảng cách của rô-bốt. Đối với $\alpha = 0$, φ_{final} bằng $\varphi_{mục\ tiêu}(-\pi/4)$ và việc tăng giá trị của alpha sẽ đưa φ_{final} đến gần $\varphi_{gap_c}(\pi/4)$

Ưu điểm: FGM chỉ có một tham số điều chỉnh, đó là hệ số alpha trong biểu thức. Vì vậy, điều chỉnh thuật toán là rất dễ dàng. FGM không gặp vấn đề tối thiểu cục bộ như ở APF và VFF. Khác với các thuật toán tránh né trước đây, cách tiếp cận mới này có thể xem xét các ràng buộc phi đối xứng của rô-bốt và chỉ tạo ra các quỹ đạo khả thi. Một ưu điểm khác là trường quan sát của rô-bốt được tính đến và rô-bốt không bị buộc phải di chuyển về các hướng không đo được. Cuối cùng, FGM điều khiển rô-bốt di chuyển về phía tâm điểm cực đại. Và khoảng cách càng nhiều càng tốt. Điều này dẫn đến các quỹ đạo an toàn hơn so với các phương pháp trước đây

Khuyết điểm: Tuy nhiên trong một số trường hợp thì FGM lại không đáp ứng tốt ở khoản về đích do luôn chọn vị trí trống lớn nhất không cần biết có về đích sớm hay không. Ngoài ra, vì khoảng cách lớn nhất rất gần nhau nên đôi khi có một số sai sót sẽ dẫn đến thay đổi thất thường trong quỹ đạo di chuyển của robot gây không an toàn. Vì vậy, chúng em sẽ đề nghị phương án cải tiến của FGM để giải quyết vấn đề trên.



Hình mô tả vấn đề về độ dài của việc di chuyển đến đích



Hình mô tả vấn đề zigzag của FGM

Ta có thể dùng hàm tiện ích thay đổi khoảng cách. Nó bao gồm hai biến: thứ nhất là kích thước của khoảng trống d_{gap-n} trong khi cái thứ hai là góc giữa tâm khoảng trống và điểm cầu môn $\phi_{gap-n-to-goal}$. Bằng cách này, việc lựa chọn khoảng cách không chỉ được thực hiện bởi kích thước như trong FGM mà còn hướng đến tọa độ mục tiêu.

$$(10) \quad U_n = k_1 d_{gap-n} + k_2 (\pi - \phi_{gap-n-to-goal})$$

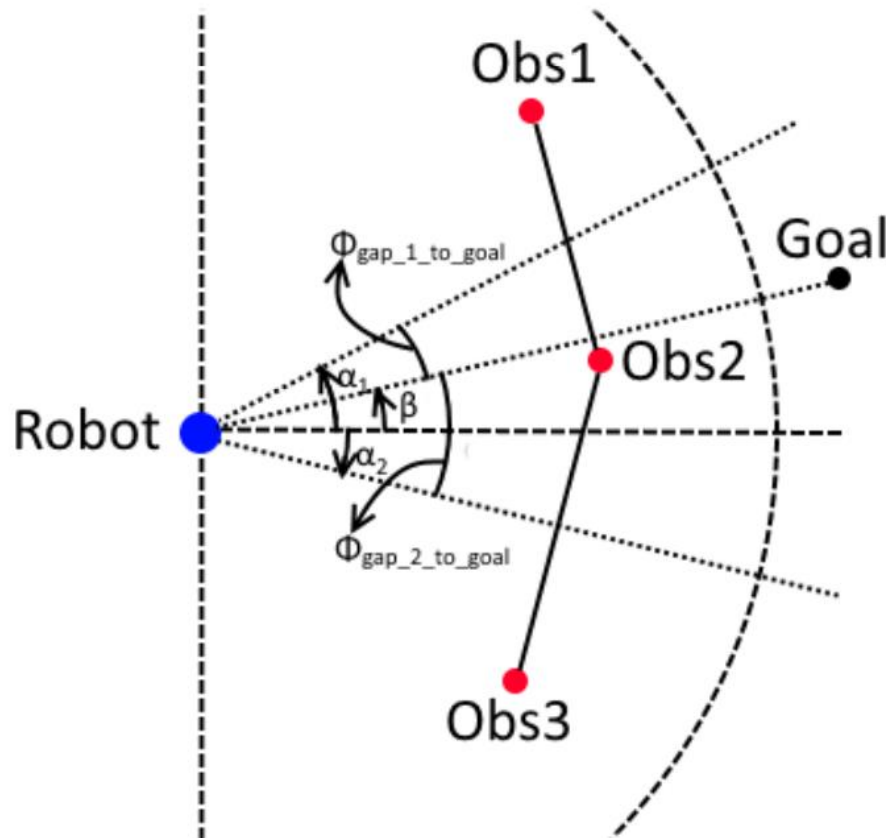
d_{gap_n} : Kích thước của khoảng cách thứ n.

k_1 : Hệ số trọng số cho

kích thước khoảng cách.

$\Phi_{\text{gap}_n \text{ to goal}}$: Góc giữa tâm khoảng cách thứ n và goal.

k_2 : Hệ số trọng lượng cho góc.

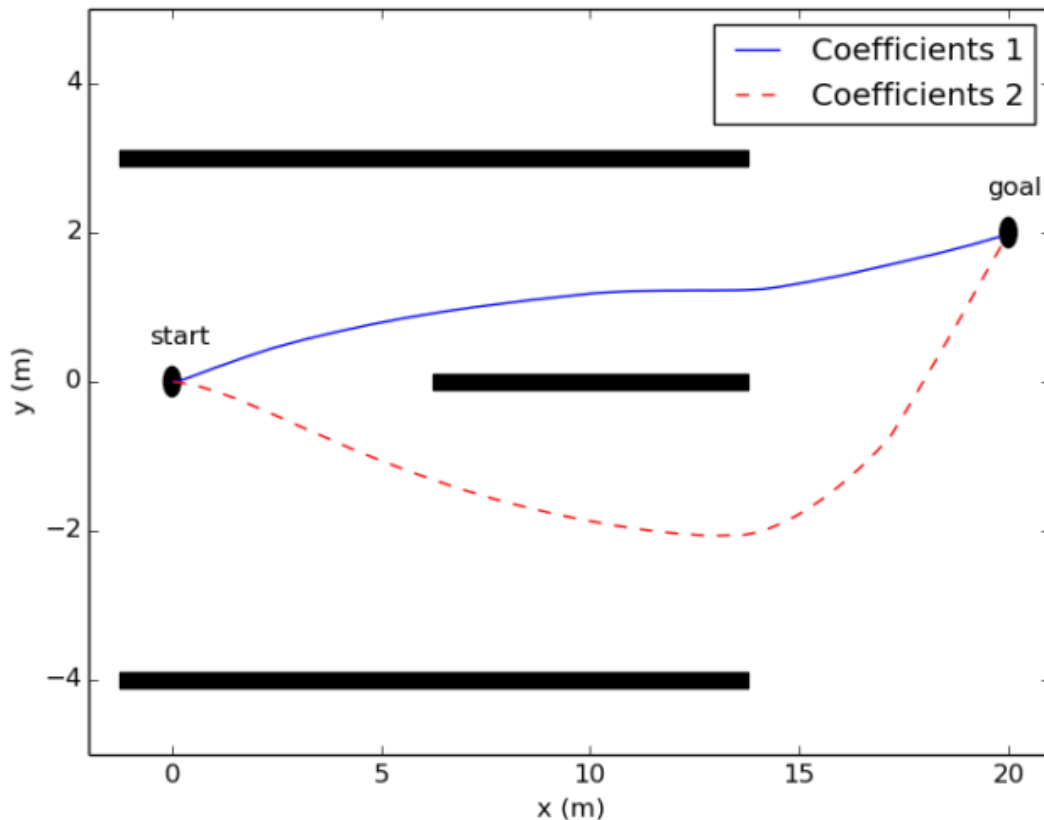


Hình biểu diễn các góc được bao gồm trong tham số thứ hai của hàm tiện ích

$$\phi_{\text{gap}_n \text{ to goal}} = |\alpha_n - \beta|$$

Việc tính toán $\Phi_{\text{gap}_n \text{ to goal}}$ dẫn đến việc lựa chọn khe hở để chui qua trong FGM truyền thống chỉ phụ thuộc vào độ lớn khe hở nhưng bây giờ qua FGM_I nó còn phụ thuộc vào góc giữa trung tâm khoảng cách và điểm đích. Điều này làm cho việc lựa chọn khoảng cách để đến đích được ưu tiên hơn => giải quyết được vấn đề độ dài quãng đường.

Và ngoài ra nếu ta muốn tăng độ an toàn cho quá trình thì ta có thể tăng k_1 hoặc giảm k_2 còn nếu độ dài quãng đường được ưu tiên hơn thì đảo ngược lại quá trình trên



Hình ví dụ ảnh hưởng của các hệ số lên quỹ đạo. Hệ số hiệu quả 1: $k_1=0,2$, $k_2=0,8$. Hệ số 2: $k_1=0,6$, $k_2=0,4$.

Để giải quyết vấn đề zigzag của FMG thì ta cần tìm giá trị hữu dụng của khe hở được chọn là giá trị tốt nhất có nghĩa là nó có giá trị lớn nhất, U_{\max} sau đó tính khoảng cách tối đa là hằng số R_{\max} .

Đối với phép tính khoảng cách tiếp theo, giá trị tiện ích của khoảng cách phải lớn hơn R_{\max} tối đa được tính cho quyết định thay đổi khoảng cách. Điều này mang sự cân nhắc thay đổi khoảng cách đã chọn.

$$R_{\max} = U_{\max} + R$$

R_{\max} : Phần thưởng tối đa.

U_{\max} : Giá trị tiện ích cho khoảng cách tốt nhất.

R : Hằng số phần thưởng.

Hằng số R ảnh hưởng đến cường độ của chuyển động zigzag. Nếu nó được chọn là giá trị lớn, robot có thể mất khả năng thay đổi khoảng cách. Nếu hằng số R được chọn là giá trị nhỏ, vấn đề zigzag có thể tiếp tục. Do đó, hằng số R được chọn là 0,1 tùy thuộc vào kết quả của các thử nghiệm. Có thể nhận thấy rằng sau khi thêm giá trị không đổi, trong FGM-I, góc nghiêng cuối cùng và hướng của robot ổn định hơn và quỹ đạo mượt mà hơn so với FGM ban đầu.

3. Tránh vật cản động

Vì trong thực tế các vật cản có thể sẽ không đứng yên mà chúng sẽ chuyển động dẫn đến các phương pháp tránh vật cản tĩnh sẽ gặp trục trặc và không an toàn. Vì vậy nên chúng ta cần phải có một phương pháp cho việc tránh vật cản động và ở đây chúng em sử dụng FGM-DW là sự kết hợp của FGM (Follow The Gap) và DWA (Dynamic Window Approach). Phương pháp này truy xuất các góc nghiêng cuối cùng từ phương pháp FGM, và sau đó sử dụng giai đoạn tính toán thiết lập vận tốc cho phép được sử dụng như trong phương pháp DWA ban đầu. Cuối cùng, tín hiệu điều khiển thích hợp nhất được tính toán từ hàm mục tiêu mới của nó. Phương pháp FGM-DW được đề xuất hướng rô-bốt đến các khu vực an toàn hơn với các cặp tốc độ góc và tốc độ quay có thể chấp nhận được tính toán bằng cách xem xét động lực học của robot.

Để biết được FGM-DW thì ta sẽ tìm hiểu về DWA và FGM. Tuy nhiên ở phần trên ta đã phân tích về FGM rồi nên phần này ta sẽ phân tích DWA và tiến đến giải pháp FGM-DW.

DWA là một phương pháp tìm kiếm không gian vận tốc có tính đến động lực học của robot. Phương pháp này gồm 3 giai đoạn.

Giai đoạn đầu tiên loại bỏ vận tốc không thể đạt được đến từ các giới hạn gia tốc của robot, trong lần thứ hai giai đoạn này, tất cả các cặp vận tốc không thể dừng lại trước khi va chạm với chướng ngại vật đều bị loại bỏ. Trong giai đoạn thứ ba, DWA đánh giá một tập vận tốc chấp nhận được bằng cách tối đa hóa hàm mục tiêu của nó được thể hiện trong biểu thức (1). DWA dự đoán kết quả của từng ứng cử viên cặp vận tốc về góc nghiêng cuối cùng, khoảng cách tối thiểu đến chướng ngại vật và giá trị vận tốc tuyến tính, đồng thời chọn cặp vận tốc tối ưu bằng cách tối đa hóa hàm mục tiêu

$$G(V, w) = \sigma[\alpha head(V, w) + \beta dist(V, w) + \gamma vel(V, w)] \quad (1)$$

$\text{head}(V, w)$ biểu thị giá trị gần đúng của góc mục tiêu và giá trị của nó tăng lên khi tiêu đề của robot tiếp cận vị trí mục tiêu

Mục đích của hàm khoảng cách $\text{dist}(V, w)$ là thúc đẩy điều hướng an toàn. Nó tính toán các giá trị khoảng cách tối thiểu đến các chướng ngại vật trên quỹ đạo thu được từ cặp vận tốc.

Hàm vận tốc $\text{vel}(V, w)$ tính toán các giá trị vận tốc tuyến tính trong tập hợp vận tốc. Hệ số α , β và γ là trọng số của các hàm này và σ là toán tử làm mịn. Tối đa hóa chức năng mục tiêu này dẫn đến các quỹ đạo an toàn để tiếp cận mục tiêu càng nhanh càng tốt

DWA chọn đệ quy các tín hiệu hướng đến mục tiêu nhanh hơn, an toàn hơn và nhiều mục tiêu hơn từ không gian vận tốc của rô-bốt. Phương pháp này có một số nhược điểm (ví dụ cực tiểu cục bộ, chỉ xét chuyển động tịnh tiến) vậy nên ta sẽ kết hợp 2 phương pháp lại và dùng FGM-DW để khắc phục việc di chuyển đến vùng trống nhưng không xem xét quá nhiều đến sự xê dịch của vật cản của FGM bằng việc lựa chọn vận tốc thích hợp và ước tính va chạm sớm của DWA.

Phương pháp sẽ gồm 3 bước: Calculation guide angle with FGM à Calculating admissible velocities à Objective function calculation.

- Calculation guide angle with FGM:

Góc dẫn hướng là một thành phần của góc mục tiêu và góc khoảng cách và nó dần dần hướng robot đến vị trí mục tiêu.

Như đã đề cập trong 2, góc dẫn hướng có thể được suy ra từ biểu thức (8). Người ta cũng chỉ ra rằng trong phương trình này, các giá trị α và d_{min} xác định góc dẫn hướng gần đúng với góc khe hở tối đa. d_{min} là khoảng cách của rô-bốt tính từ chướng ngại vật gần nhất. Nếu rô-bốt ở gần chướng ngại vật, d_{min} tiến dần đến 0, nghĩa là rô-bốt chủ yếu hướng đến tâm khoảng cách an toàn nhất còn nếu không, nó chủ yếu hướng đến mục tiêu.

- Calculating admissible velocities:

DWA sử dụng gia tốc góc và tuyến tính (V, w) để tính vận tốc có thể đạt được từ vận tốc hiện tại (V_n, w_n) trong một khoảng thời gian ngắn (t). Việc tính toán vận tốc có thể đạt được (V_r) được hiển thị trong biểu thức:

$$V_r = \{(V, w) | V \in [V_n - \dot{V}t, V_n + \dot{V}t] \wedge w \in [w_n - \dot{w}t, w_n + \dot{w}t]\}$$

Vận tốc cho phép là các cặp vận tốc có thể đạt được, có khả năng dừng rô-bốt trước khi va chạm. Việc tính toán các cặp này phụ thuộc vào độ giảm tốc (V' break, w' break) của robot và khoảng cách tối thiểu giữa các chướng ngại vật và quỹ đạo tạo ra với cặp vận tốc (tâm (V, w)). Tính toán vận tốc cho phép (V_a) được hiển thị trong biểu thức (3). Phương trình này xuất phát từ tính toán chuyển vị của đối tượng giảm tốc. Trong bài báo gốc của DWA, một cặp vận tốc có thể chấp nhận được được mô tả như sau: “Một cặp (V, w) được coi là có thể chấp nhận được, nếu robot có thể dừng lại trước khi nó chạm tới chướng ngại vật gần nhất trên đường cong tương ứng”

$$V_a = \{(V, w) | V \leq \sqrt{2 \min_d(V, w) \dot{V}_{break}} \wedge w \leq \sqrt{2 \min_d(V, w) \dot{w}_{break}}\}$$

(3)

- Objective function calculation:

Như đã lưu ý trước đây, DWA sử dụng ba thành phần để đánh giá các cặp vận tốc. Một trong số đó là chức năng khoảng cách xử phạt vận tốc dẫn đến quỹ đạo nguy hiểm gần chướng ngại vật. Mặt khác, FGM đã đảm bảo an toàn cho rô bốt do hướng robot đến trung tâm khoảng trống lớn nhất hiện có. Do đó, chức năng khoảng cách của DWA trở nên dư thừa vì FGM rất hiệu quả trong việc tính toán hướng di chuyển an toàn.

Chức năng vận tốc đảm bảo rằng robot hoạt động ở tốc độ tuyến tính tối đa cho phép. Hàm đánh giá này được chuẩn hóa và tính toán như trong biểu thức (4). Chức năng này cung cấp hiệu ứng tối đa hóa tốc độ cho hàm mục tiêu ngoại trừ nó ở gần vị trí mục tiêu.

$$vel(V, w) = \begin{cases} \frac{V}{V_{max}} & , \text{ if robot is far from goal} \\ 1 - \frac{V}{V_{max}} & , \text{ if robot is near to goal} \end{cases}$$

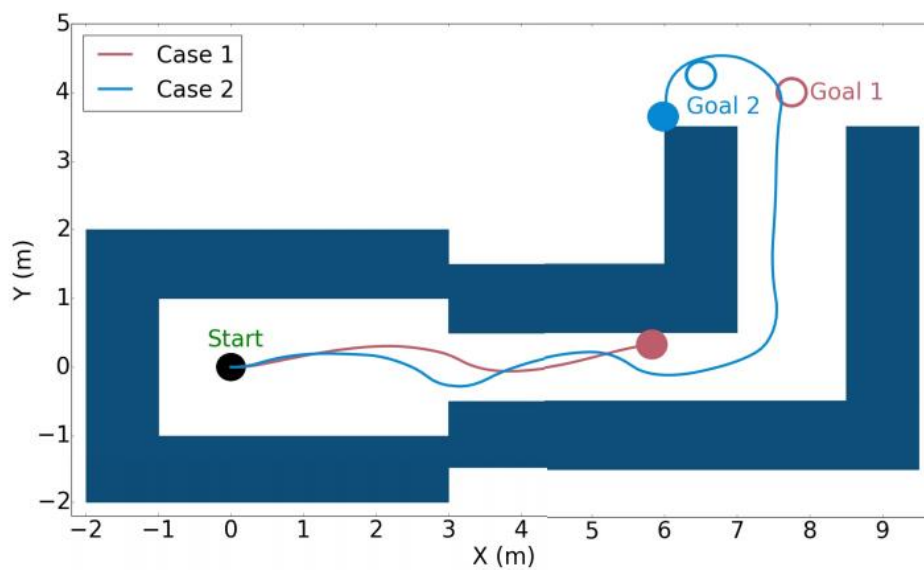
(4)

Chức năng tiêu đề so sánh góc tiêu đề cuối cùng của cặp vận tốc và góc dẫn hướng thu được từ FGM. Sự khác biệt về góc này được sử dụng để tạo ra điểm tiêu đề. Hàm này được chuẩn hóa và tính toán như trong biểu thức (5). Trong phương trình này $\Delta\theta$ biểu thị góc chênh lệch giữa các hướng này.

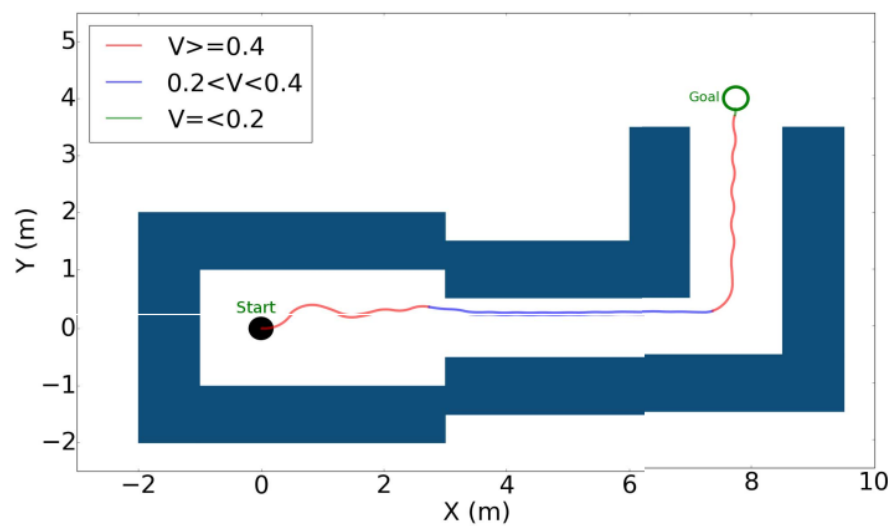
$$(5) \quad head(V, w) = 1 - \frac{|\Delta\theta|}{\pi}$$

Hàm mục tiêu của FGM-DW là một thành phần của các chức năng vận tốc và tiêu đề được minh họa trong biểu thức (6).

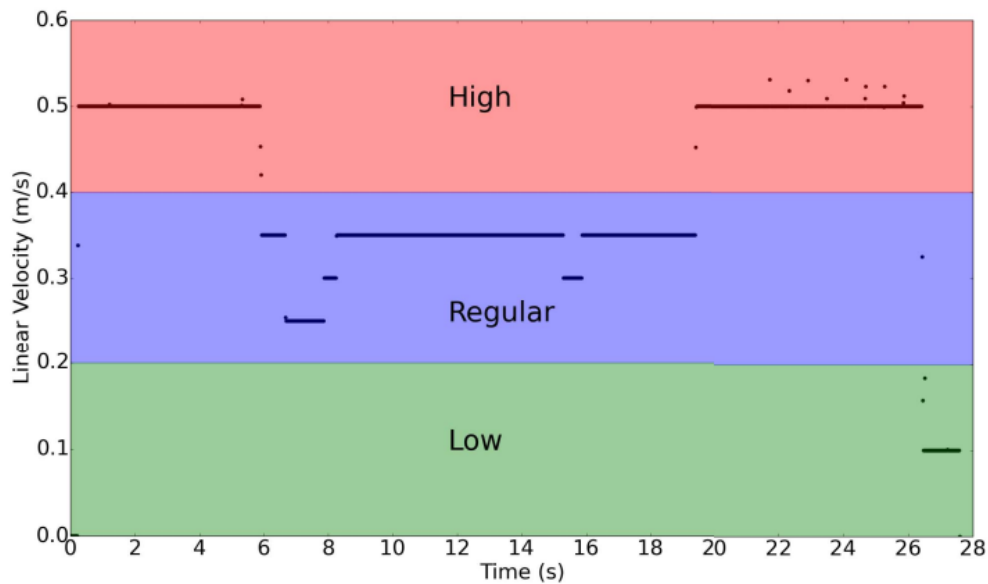
$$(6) \quad G_{new}(V, w) = \beta head(V, w) + \gamma vel(V, w)$$



Hình 1



Hình 2



Hình 3

Phương pháp FGM-DW được triển khai trong môi trường được minh họa trước đây trong Hình 1 để cho thấy sự khác biệt giữa các phương pháp FGM và FGM-DW thông thường. Các đường đi của FGM-DW và vận tốc tuyến tính được áp dụng lần lượt được hiển thị trong Hình 2 và Hình 3. Phương pháp FGM-DW cho kết quả phù hợp khi robot di chuyển trong các đoạn hẹp.

Phương pháp FGM-DW được đề xuất làm giảm tốc độ rô-bốt trong các đoạn hẹp và tăng tốc độ ở các vùng rộng nhờ sự trợ giúp của bộ phận DW. Nếu rô-bốt tiếp cận khu vực mục tiêu nhỏ hơn khoảng cách giới hạn (0,5 mét trong mô phỏng), phương pháp này thường cho tốc độ thấp nhiều hơn tốc độ cao hơn. Công thức này ngăn robot vượt qua khu vực mục tiêu.

→ Kết luận: vậy phương pháp FGM-DW đã có thể giải quyết được cho chúng ta tránh vật cản động bằng việc sử dụng lợi thế của FGM và việc tính vận tốc đến vật cản tối đa và dự đoán sớm vận tốc của DWA

References

- [1] A new gap-based obstacle avoidance approach: follow the obstacle circle method
- [2]<https://www.mdpi.com/2079-9292/11/18/2801>
- [3]<https://www.sciencedirect.com/science/article/abs/pii/S0921889089900419>
- [4]<https://www.mdpi.com/1424-8220/19/22/4963>
- [5]<https://www.mdpi.com/1424-8220/17/5/1061>
- [6] ĐIỀU HƯỚNG ROBOT DỰA VÀO CAMERA ĐỒNG BỘ VÀ THIẾT BỊ ĐO LƯỜNG QUÁN TÍNH CÓ XÉT YẾU TỐ VẬT CẢN
- [7] Improved Follow the Gap Method for Obstacle Avoidance
- [8] A novel obstacle avoidance algorithm: “Follow the Gap Method”