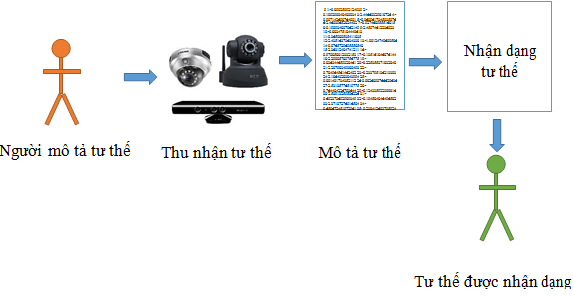
# BÀI TOÁN NHẬN DẠNG CỬ CHỈ ĐỘNG CỦA TAY SỬ DỤNG KINECT

Nhận dạng đối tượng trong lĩnh vực thị giác máy tính có thể hướng đến nhiều đối tượng khác nhau như: lá cây, logo, vân tay hay nhận dạng cử chỉ động của tay … Trong khuôn khổ của đề tài, tôi hướng đến giải quyết bài toán nhận dạng cử chỉ động của tay người.

*Nhận dạng cử chỉ của tay là xác định xem cử chỉ mà người đó thực hiện thuộc lớp cử* *chỉ nào trong số tập cử chỉ đã định nghĩa từ trước.*

Ví dụ: Định nghĩa 20 loại cử chỉ động của tay từ trước theo tên của mỗi loại cử chỉ. Nhiệm vụ của hệ thống nhận dạng là với một cử chỉ thu nhận được và hệ thống có thể nhận dạng được đó là cử chỉ thuộc lớp nào trong 20 lớp đã định nghĩa sẵn.



Hình 1. Nhận dạng tư thế người

**Các ứng dụng của nhận dạng cử chỉ động của tay người**

Bài toán nhận dạng cử chỉ có nhiều ứng dụng như:

• Phát triển các công cụ trợ giúp nói chuyện bằng tay

• Giúp trẻ em có thể thao tác với máy tính

• Chuẩn đoán các cảm xúc của bệnh nhân, đo mức độ trầm cảm

• Phát hiện nói dối

• Thực hiện di chuyển vật thể trong môi trường ảo

• Trợ giúp dạy học từ xa

• v.v.

**Phân loại các hệ thống nhận dạng**

Để nhận dạng được cử chỉ, một số cảm biến được sử dụng để thu thập dữ liệu về cử chỉ đó. Hệ thống nhận dạng cử chỉ có thể được tạm phân thành hai loại.

1) Dựa trên cảm biến gắn trên người: Theo loại này, người sẽ phải đeo một số cảm biến (vận tốc, gia tốc, cảm biến quang, v.v) ở một số vị trí thực hiện cử chỉ (cánh tay, ngón tay). Điển hình của phương pháp này là các hệ thống games, đồ họa sử dụng găng tay chuyên dụng để đo sự chuyển động của tay và các ngón tay .

2) Dựa trên các cảm biến độc lập với người: Theo loại này, hệ thống cảm biến gắn bên ngoài môi trường. Người không phải đeo bất kỳ loại cảm biến nào cả. Điển hình của phương pháp này sử dụng cảm biến camera gắn trong môi trường. Ưu điểm chính của phương pháp này so với phương pháp thuộc lớp thứ nhất là chi phí giá thành cho thiết bị thông dụng, rẻ. Người không phải mang vác bất kỳ thiết bị nào nên thực hiện cử chỉ một cách tự nhiên. Trong vài năm trở lại đây, với sự ra đời của cảm biến Kinect của Microsoft với giá thành rẻ nhưng lại cung cấp dữ liệu đa thể thức về môi trường như: âm thanh, hình ảnh RGB, ảnh độ sâu, v.v. Việc sử dụng kết hợp các loại dữ liệu khác nhau cho phép nâng cao hiệu quả của nhận dạng.

Trong khuôn khổ của đề tài, tôi đi theo hướng tiếp cận thứ 2, sử dụng cảm biến ngoài, cụ thể là cảm biến Kinect để thu lại dữ liệu về cử chỉ thực hiện.

**Mô hình nhận dạng**

Bài toán nhận dạng thông thường được thực hiện thông qua 2 pha chính:

• **Pha huấn luyện:** hệ thống sẽ học các tham số mô hình biểu diễn các lớp cử chỉ từ dữ liệu thu thập được từ trước.

• **Pha nhận dạng:** hệ thống đưa ra kết quả nhận dạng với một dữ liệu vào mới



Hình 2. Thiết bị Kinect

Kinect (hay còn biết với mã là Project Natal) là 1 thiết bị thu nhận các cảm biến chuyển động, được phát triển bởi Microsoft.

Các mốc thời gian ra đời, phát triển của kinect và các thành phần liên quan:

- **5/30/2007:** Microsoft nung nấu ý tưởng về 1 thiết bị dùng camera ghi nhận cử động điều khiển thay cho các thiết bị truyền thống.

- **6/1/2009:** Microsoft công bố “Project Natal” ở hội nghị thường niên E3.

- **6/13/2010:** Trong suốt hội nghị E3, đổi tên “Project Natal” thành Kinect, chính thức là 1 thiết bị hỗ trợ cho Xbox 360.

- **11/4/2010:** Microsoft chính thức tung ra thị trường Kinect, cũng từ đây, nhưng kế hoạch phát triển Driver nguồn mở cho Kinect của các tổ chức/ hacker cũng bắt đầu thực hiện.

- **11/10/2010:** hacker trẻ tuổi Hector đã phát triển thành công Driver cho Kinect.

Kinect sử dụng webcam, thiết bị thu phát sóng hồng ngoại, và thiết bị thu nhận âm thanh để ghi nhận tín hiệu chuyển động của người dùng và nhận dạng các lệnh điều khiển thông qua giọng nói, giúp cho họ tương tác với thiết bị mà không cần chạm vào bất kì thiết bị điều khiển nào.



Hình 3. Nhận diện hành động người dùng và tương tác lại với chương trình.

Kinect, hỗ trợ các chức năng tương tác sau:

- Ghi nhận chuyển động tay (hand gesture), bao gổm các hành động xoay vòng(circle), di chuyển tay (wave gesture), push,...

- Ghi nhận chuyển động toàn cơ thể (full body skeleton), xác định các vị trí chính của cơ thể như đầu, vai, cẳng tay, chân,…

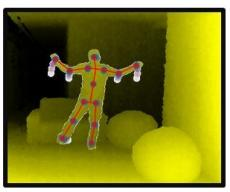
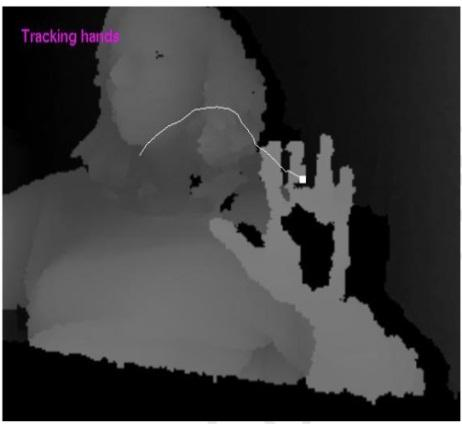
- Điều khiển bằng giọng nói.



Hình 4. Kinect có thể điều khiển bằng giọng nói người chơi

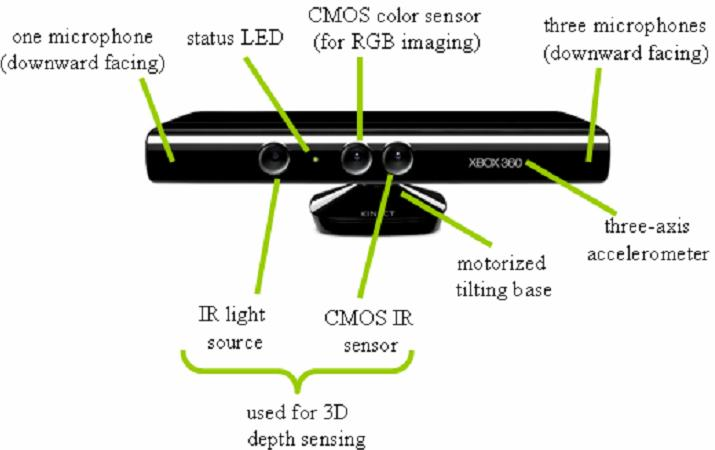
- Nhận dạng số người đang chơi

- Phân biệt đối tượng dựa vào độ sâu.



Hình 5. Các chức năng thường dùng của Kinect

Kinect có cấu tạo thon, dài, nằm ngang bao gồm 1 webcam có thể ghi nhận được hình ảnh; 1 thiết bị phát tia hồng ngoại, 1 thiết bị thu nhận tín hiện hồng ngoại 1 thiết bị ghi nhận âm thanh. Kinect dùng công nghệ xác định khoảng cách dùng camera kết hợp thiết bị thu phát tia hồng ngoại, phát triển bởi các lập trình viên Israel thuộc PrimeSense, có khả năng ghi nhận thông tin 3D của đối tượng dựa trên các thông tin, cấu trúc của các tia hồng ngoại nhận được. Có thể xem như Kinect là một 3D scanner, xây dựng cấu trúc vật thể 3D dựa trên thông tin ảnh và khoảng cách đến từng điểm của ảnh.



Hình 6. Những thành phần chính của Kinect

Các thành phần bên trong Kinect gồm có: bộ nhớ RAM, bộ cảm ứng Prime Sense PS1080-A2, quạt tản nhiệt, động cơ điều khiển gốc ngẩng (Motorized Tilt), bộ gia tốc 3 trục, 4 microphone (Multi – Array Mic) và 3 camera: RGB camera, bộ cảm biến độ sâu (3D Depth Sensors).

**Thư viện hỗ trợ Kinect**

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

***Thư viện Libfreenect***

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

***Thư viện Code Laboratories Kinect***

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

***Thư viện OpenNI***

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

### *Thư viện Kinect SDK*

SDK cung cấp cho nhà phát triển những bộ cảm biến có chiều sâu, camera cảm biến màu sắc và xác định nguồn âm thanh theo chùm tia. Các nhà phát triển cũng có thể theo dõi ảnh ở dạng khung sườn của một hay hai người đang di

chuyển trong tầm ngắm Kinect, từ đó các nhà phát triển có thể tạo ra những ứng dụng hướng theo cử chỉ (gesture).

SDK gồm tính năng âm thanh như giảm tiếng ồn và hạn chế tiếng vang. Các nhà phát triển có thể truy cập vào công nghệ hình thành tia để nhận diện ra nguồn âm thanh hay có thể truy cập vào giao diện trình ứng dụng (AP ) để nhận diện giọng nói trong hệ thống Windows. Các nhà phát triển có thể viết ứng dụng bằng ngôn ngữ C#, C++ và bất cứ ngôn ngữ .Net nào.

**Các tính năng của Kinect SDK bao gồm :**

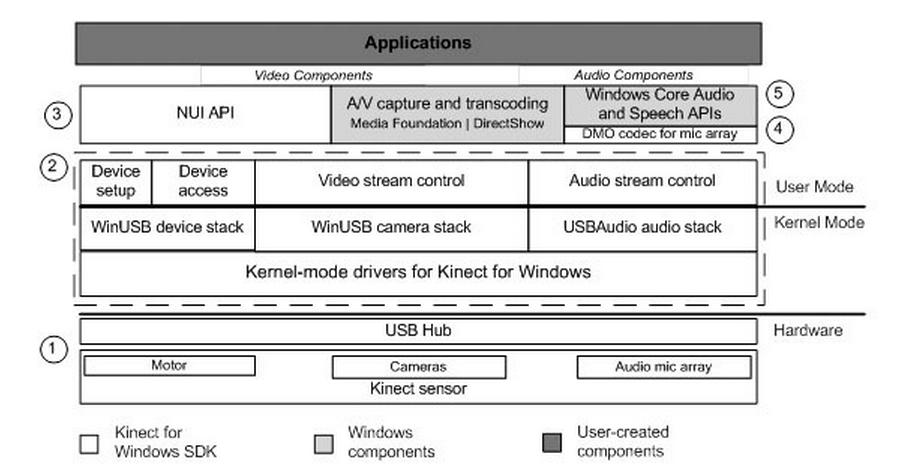
**Raw Sensor Streams - Bộ cảm biến độ sâu**: Các nhà phát triển có thể truy cập dữ liệu thô từ bộ cảm biến sâu, camera cảm biến mầu sắc và dải 4 yếu tố microphone. Điều đó cho phép họ xây dựng các cảm biến cấp độ thấp hơn thông qua cảm biến Kinect.

**Theo dõi cấu trúc**: SDK có khẳ năng theo dõi kiến trúc xương của một hoặc hai người đang cử động trong tầm nhìn của Kinect giúp điều chỉnh các ứng dụng về cử động. Bên cạnh đó, thư viện này hỗ trợ theo dõi cùng lúc 20 khớp xương trên cơ thể.

**Khả năng xử lý âm thanh:**bao gồm loại bỏ tiếng ồn và tiếng vang, hình thành chùm tia để xác định nguồn âm thanh, và tích hợp với các hàm API nhận dạng giọng nói của Windows.

**Dễ dàng cài đặt**: SDK nhanh chóng cài đặt theo tiêu chuẩn cho Windows 7 mà không đòi hỏi các cấu hình phức tạp với dung lượng nhỏ hơn 100 MB. Các nhà phát triển có thể cài đặt và chạy chỉ trong một vài phút với một cảm biến tiêu chuẩn Kinect được phổ biến rộng rãi tại kênh bán lẻ.

**Tài liệu hướng dẫn**: SDK bao gồm hơn 100 trang tài liệu công nghệ chất lượng cao. Ngoài việc có các thông tin trợ giúp tổ hợp sẵn, tài liệu còn bao gồm những hướng dẫn chi tiết cho hầu hết các mẫu tham khảo được cung cấp với SDK.



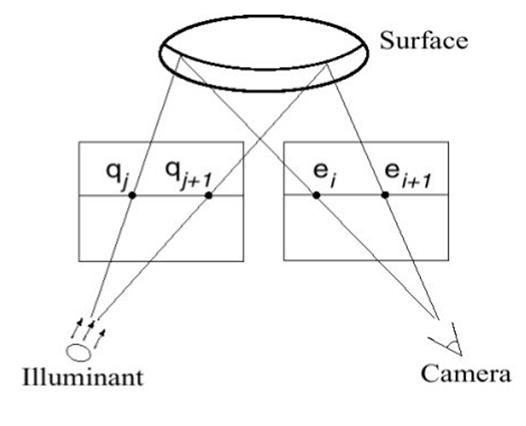
Hình 7. Kiến Trúc Kinect .Net SD

## Nguyên lý hoạt động cảm biến

### *Giải pháp PrimeSensor*

Giải pháp PrimeSensor của hãng PrimeSense có khả năng xử lý các cử chỉ dạng 3 chiều (3D), nhưng dữ liệu hình ảnh lại được từ cảm biến ảnh CMOS. Giải pháp này, có khả năng ghi nhận độ sâu, màu sắc và cả âm thanh ở chế độ thời gian thực của không gian cần kiểm soát mà không cần yêu cầu người dùng (người điều khiển hệ thống) phải mặc hay cầm bắt cứ công cụ nào khác, cũng như không bắt buột phải cân chỉnh hay truy xuất dữ liệu đã được số hóa từ máy tính trung tâm.

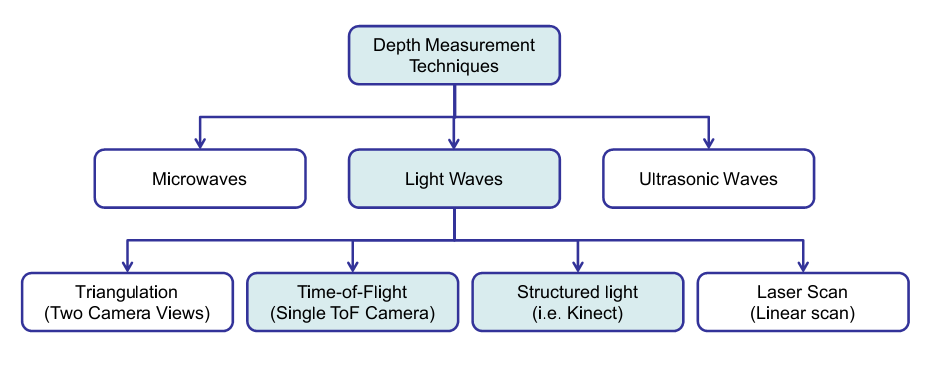
Giải pháp System on a chip (**SoC**) của PrimeSense có khả năng tính toán giá trị độ sâu của mỗi điểm ảnh (pixel), dựa trên tính hiệu trả về của cảm biến CMOS. Ngoài ra, **SoC** này cũng trang bị cảm biến CMOS để nhận dạng màu sắc của điểm ảnh, xử lý nguồn sáng trả về dựa trên công nghệ phân mã nguồn sáng Light Coding. Về nguyên tắc, Light Coding làm việc bằng cách mã hóa vùng quan sát thông qua đèn R – đèn này “vô hình” đối với mắt người.



Hình 7. Giải pháp PrimeSensor

Giải pháp dựa trên sự mã hóa ánh sáng từ tia hồng ngoại của camera IR light. Với việc hỗ trợ một cảm biến hồng ngoại bằng việc tạo ra 2 camera phát và nhận tia laser thì PrimeSense đã áp dụng vào rất nhiều thiết bị 3D của mình – trong đó Kinect Sensor là một điển hình.

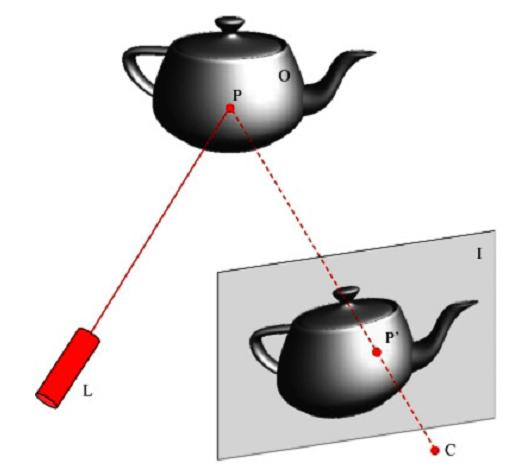
### *Phân tích kỹ thuật đo lường chiều sâu*



Hình 8. Các kỹ thuật đo lường chiều sâu

**Kỹ Thuật Triangulation**

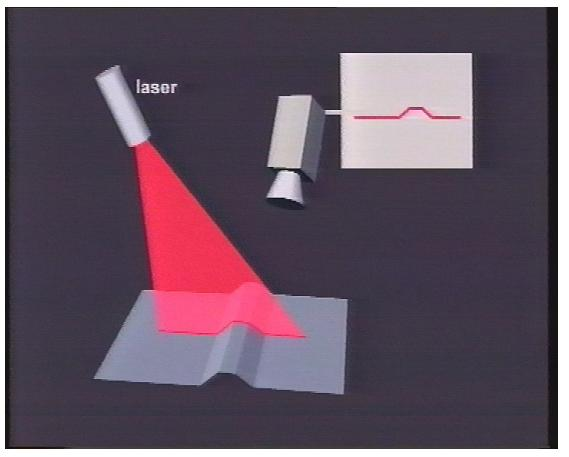
Kỹ thuật này bao gồm 2 thành phần: 1 thành phần phát sóng tia laser, 1 thành phần nhận thông tin ảnh, nhận tia laser tương ứng. Thành phần phát nhận tia laser giúp xác định vị trí của điểm laser trên vật tương ứng với điểm nào trên ảnh màu ghi nhận được từ thành phần còn lại. Như vậy, với vị trí của điểm laser trên vật, vị trí của thành phần phát tia laser, thành phần nhận thông tin ảnh màu, laser, tạo thành tam giác. Các thông tin cơ bản của tam giác này đã có sẵn như vị trí tương đối của thành phần thu/phát tia laser và thành phần nhận ảnh, góc hợp bởi 2 thành phần này, từ đó tính được khoảng cách thực tế của điểm trên vật, và có cả màu sắc của điểm đó đựa trên ảnh màu.



Hình 1.8– Minh họa vị trí 2 thành phần chính trong Triangulation

**Kỹ Thuật Structure light**

Kỹ thuật này phát ra chùm tia có cấu trúc lặp (pattern), và ghi nhận lại kết quả. Ảnh kết quả có thể áp dụng nhiều phương pháp phân tích khác nhau để xác định được độ lồi lõm của vật được chiếu. Mẫu chiếu lên đối tượng có thể là mẫu 1 chiều hoặc 2 chiều. Phương pháp phân tích do dựa vào mẫu được phát ra và kết quả thu nhận lại nên hiện nay có rất nhiều cách phân tích khác nhau, và liên tục được phát triển.



Hình 9. Mô phỏng kỹ thuật Structured Light

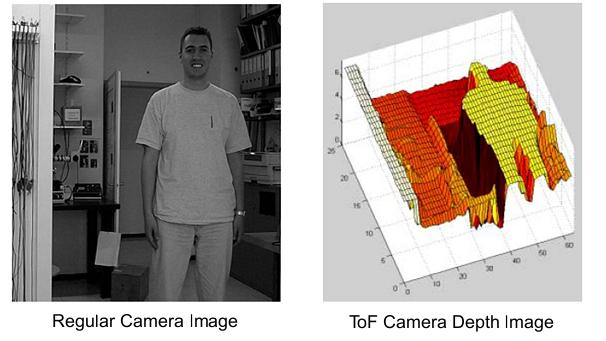
Bộ phát tia laser phát chùm tia laser theo 1 chiều, camera nghiêng 1 góc xác định với đối tượng, dựa vào độ lồi lõm của đường thằng ghi nhận được mà tính toán được cấu trúc 3D của vật được quét. Chỉ 1lần chụp hình có thể tính được độ sâu của toàn bộ khuôn mặt dựa trên mẫu bị biến dạng như thế nào so với mẫu ban đầu. Cách này thường áp dụng cho các chương trình đòi hỏi tốc độ cao hoặc ứng dụng tương tác trong thời gian thực. Phụ thuộc nhiều vào thuật toán nội suy và độ nhiễu của ảnh kết quả.

**Kỹ Thuật Time – Of – Flight**

Kỹ thuật này phát ra các tia laser để cảm nhận bề mặt của đồ vật, để tính toán được khoảng cách từ thiết bị đến từng điểm của đối tượng, người ta dùng cách tính thời gian di chuyển của chùm tia laser, tên gọi Time of flight cũng xuất phát từ đó. Thiết bị thuộc nhóm này bao gồm 1 bộ phận phát tia laser và bộ cảm biến nhận tia laser. Bộ phận phát tia laser sau khi phát chùm tia laser, ghi nhận lại thời điểm phát sóng, bộ phận nhận tia khi nhận đựa tia laser nào thì ghi nhận thời điểm nhận tia đó. Khi đó, với mội điểm, tương ứng 1 tia trong chùm tia phát ra, ta có khoảng thời gian lan truyền t của tia laser đó.Thời gian lan truyền bao gồm thời gian chạm đối tượng, và thời gian phản chiếu lại vào thiết bị nhận tia. Ta đã biết thời gian di chuyển của ánh sáng là hằng số c, đã biết thời gian lan truyền, ta tính khoảng cách

d = (c \* t) / 2.

Độ chính xác của thiết bị càng lớn khi khoảng thời gian giữa các lần bắt tia laser trong thiết bị nhận tia càng nhỏ. Để có thể đạt đến độ chính xác là milimet thì khoảng thời gian giữa 2 lần nhận tia phải là 3.3 picosecond. Đây là khoảng thời gian mà chưa có thiết bị nào hiện nay có thể đạt được. Khuyết điểm của kĩ thuật này là độ chính xác không cao, phụ thuộc vào thiết bị nhận. Tuy nhiên khoảng cách mà thiết bị này có thể ghi nhận có thế đến con số kilomet, do đó, phù hợp cho việc quét các đối tượng có kích thước lớn.



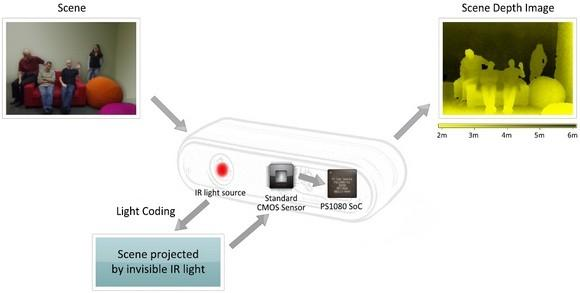
Hình 10. Mô phỏng kỹ thuật Time – Of – Flight

### *Tính toán độ sâu*

Cặp cảm biến IR camera và IR projector sẽ phối hợp với nhau cho ra giá trị độ sâu ảnh bằng công nghệ Light Coding của PrimeSense. Để lấy các thông tin cần thiết từ kinect, ta không thể tương tác trực tiếp với driver của thiết bị, việc này đòi hỏi kĩ thuật cao và khả năng lập trình rất tốt và am hiểu cấu tạo thiết bị. Hướng giải quyết là dùng một thư viện hỗ trợ để lấy dữ liệu một cách dễ dàng và được chuẩn hóa.

Để hiện thực hóa 1 đối tượng 3D chúng ta cần xác định ít nhất phải có các thông tin về màu sắc độ sâu từ nhiều điểm trên đối tượng ở nhiều góc nhìn khác nhau. Phần này trình bày cách thức hoạt động của kinect, những thông tin thu nhận được, để có thể trả ra các thông tin về ảnh và độ sâu như mong muốn.

Để thu nhận hình ảnh, sử dụng thiết bị thu nhận hình ảnh là 1 webcam bình thường. Để thu nhận độ sâu từng điểm ảnh, Kinect hoạt động dựa trên công nghệ Light Coding. Công nghệ Light Coding hoạt động dựa trên việc phát ra chùm tia hồng ngoại đặc trưng riêng từng tia (không thấy được dưới mắt thường). Kinect dùng 1 bộ cảm biến CMOS chuẩn, để ghi nhận lại các tia hồng ngoại bị phản xạ lại khi tiếp xúc với môi trường, dựa vào các đặc trưng mà xác định cụ thể vị trí tia hồng ngoại trong chùm tia và độ sâu của tia đo được. Sử dụng **PS1080 SoC** chip tính toán song song để xác định độ sâu của toàn bộ chùm tia phản xạ và xuất ra độ sâu của tất cả điểm ảnh.



Hình 11. Quá trình thu về bản đồ độ sâu ảnh

Khác với kỹ thuật Stereo Camera với việc dùng cặp camera giống nhau để xây dựng nên bản đồ độ sâu, hay kỹ thuật **Time – Of – Flight** (TOF) định nghĩa khoảng cách bằng ước lượng thời gian di chuyển của tia sáng đi và về trong không gian. Công việc tính toán này được thực hiện bên trong Kinect bằng chip **PS1080 SoC** của PrimeSense. Công nghệ mới này được cho là đáp ứng chính xác hơn, giá cả rẻ hơn cho việc sử dụng ở môi trường trong nhà.

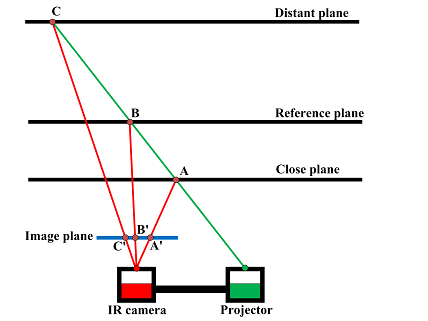
Project sẽ chiếu một chùm sáng hồng ngoại, tạo nên những đốm sáng ở không gian phía trước Kinect, tập hợp đốm sáng được phát ra này là cố định. Những đốm sáng này được tạo ra nhờ một nguồn sáng truyền qua lưới nhiễu xạ (diffraction grating). Tập hợp các điểm sáng này được IR camera chụp lại, thông qua giải thuật đặc biệt được tích hợp trong chip **PS1080 SoC** cho ra bản đồ độ sâu. Bản chất của giải thuật này là các phép toán hình học dựa trên quan hệ hai cảm biến IR camera và Projector mà ta sẽ tìm hiểu ở phần tiếp theo.



Hình 12. Mẫu hình chiếu từ IR camera và Project

Để hiểu cách thức Kinect ước lượng khoảng cách tới vật thể trong môi trường, ta cần phân tích một số điểm sau .

Ta giả sử Project phát đi một tia sáng dọc đường màu xanh lá, nó sẽ được chụp lại dưới dạng một đốm sáng bởi IR camera khi chạm vào bề mặt vật thể trong không gian.



Hình 13. Tính toán khoảng cách tới một điểm chiếu từ Project

Ta sẽ bắt đầu xét ba mặt phẳng ở ba khoảng cách khác nhau:

- Mặt phẳng gần Kinect (close plane).

- Mặt phẳng ở xa Kinect (Distant plane).

- Mặt phẳng tham chiếu (Reference plane) ở giữa hai mặt phẳng trên.

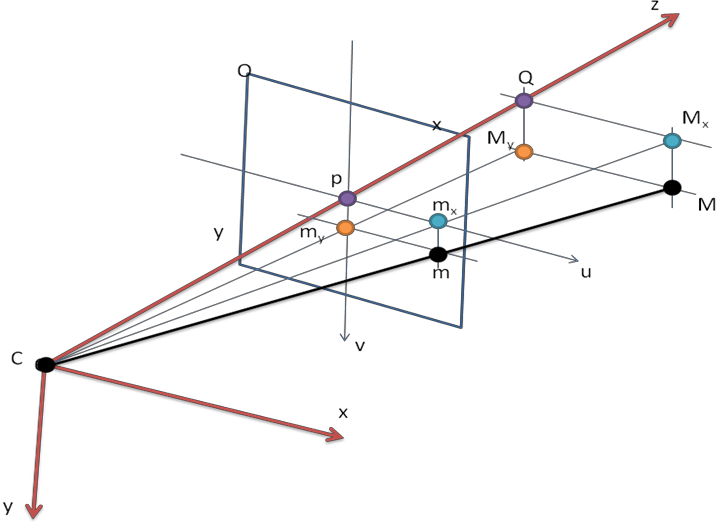
Trong đó, mặt phẳng tham chiếu ngầm được biết trước bên trong Kinect với đầy đủ thông tin về khoảng cách. Ngoài ra, ta cũng sẽ đề cập thêm đến mặt phẳng ảnh (image plane) của IR camera, là mặt phẳng hình chiếu của các điểm trong không gian thu về bởi IR camera.

Ta xét ba trường hợp: Khi tia sáng màu xanh lá chạm vào 3 điểm trên 3 mặt phẳng lần lượt là A, B, C. Ba điểm này được chiếu lên mặt phẳng ảnh tương ứng là A’, B’, C’. Quan sát vị trí A’, B’và C’, ta có nhận xét: Điểm A càng gần Kinect (hay close plane càng gần Kinect) thì A’ càng xa B’ về phía bên phải. Ngược lại, điểm C càng xa Kinect (hay distant plane càng xa Kinect) thì C’ càng xa B’ về phía bên trái.

Từ đó, khi ta biết trước hướng, điểm xuất phát của tia sáng từ Project và vị trí B’ là hình chiếu của điểm B trên mặt phẳng tham chiếu lên mặt phẳng ảnh, ta hoàn toàn có thể tính toán được độ sâu ảnh hay khoảng cách tới vật thể.

### *Xác định vị trí trong không gian*

Dựa trên hình 2.10, ta đã có thông tin Cp, vị trí điểm trong mặt phẳng ảnh theo hệ trục Oxy, tương ứng là độ sâu CQ của điểm đó trong không gian 3D lấy từ kinect. Chuyển tư điểm m(x, y) sang M(X, Y, Z) xem như là đã có thông tin Z cần xác định thêm thông tin X và Y. Do đó, phân tích điểm M thành 2 điểm Mx, My bằng cách chiếu 2 điểm này lên mặt phẳng Czx, và Czy.



Hình 14. Phân tích điểm M trong không gian.

Để tính Y, Áp dụng công thức của 2 tam giác đồng dạng là Cpmy và CQMy ta có:

Thay QMy= Y, biến đổi công thức và chuyển đổi hệ trục tọa độ từ Oxy sang puv (m(x, y) thành m(x – 320, y - 2 0)) ta được:

Đây là công thức tỉ lệ nên có thể áp dụng được với các đại lượng khác nhau (mét của độ sâu và pixel của ảnh). Tương tự áp dụng cho tính X, ta cũng có được công thức:

Như vậy từ 2 thông tin ảnh (màu sắc red, green, blue; vị trí trong mặt phẳng ảnh Oxy dòng, cột tính theo đơn vị pixel) và thông tin độ sâu tính theo đơn vị mét, ta tính được vị trí (X, Y, Z) của điểm đó trong không gian và màu sắc tương ứng.