

ĐẠI HỌC THÁI NGUYÊN  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

**HOÀNG THỊ BÍCH LỆ**

**PHÁT HIỆN LỖI SẢN PHẨM TRÊN DÂY CHUYỀN  
ĐÓNG CHAI NƯỚC BẰNG XỬ LÝ ẢNH**

**LUẬN VĂN THẠC SĨ KHOA HỌC MÁY TÍNH**

**Chuyên ngành: Khoa học máy tính**

**Mã số: 60.48.01.01**

**Người hướng dẫn khoa học: TS. Phạm Đức Long**

**Thái Nguyên - 2015**

## LỜI CAM ĐOAN

Tôi xin cam đoan những nội dung được trình bày trong bản luận văn này là công trình nghiên cứu của riêng tôi, trong quá trình nghiên cứu luận văn “Phát hiện lỗi sản phẩm trên dây chuyền đóng chai nước bằng xử lý ảnh”, các kết quả và dữ liệu được nêu ra hoàn toàn trung thực dưới sự hướng dẫn của TS Phạm Đức Long. Mọi thông tin trích dẫn trong luận văn đã được ghi rõ nguồn gốc và có liệt kê các tài liệu tham khảo.

Tôi xin chịu trách nhiệm về những lời cam đoan trên.

*Thái Nguyên, ngày      tháng      năm 2015*

**HỌC VIÊN**

**Hoàng Thị Bích Lệ**

## LỜI CẢM ƠN

Để hoàn thành được luận văn này em xin chân thành cảm ơn sự giúp đỡ nhiệt tình và tạo điều kiện của trường Đại học Công nghệ Thông tin & Truyền thông và Tiến sĩ Phạm Đức Long đã hướng dẫn và động viên em rất nhiều trong suốt quá trình em làm luận văn.

Em xin được gửi lời cảm ơn sâu sắc tới các Thầy, Cô giáo trong trường Đại học Công nghệ Thông tin & Truyền thông, đồng nghiệp và các bạn những người luôn sát cánh và sẻ chia cùng mình.

Cuối cùng em xin gửi lời cảm ơn sâu sắc và chân thành nhất đến những người thân trong gia đình luôn tận tình cảm thông và chia sẻ những niềm vui và nỗi buồn cùng em trong suốt thời gian làm luận văn.

*Thái Nguyên, ngày tháng năm 2015*

## MỤC LỤC

LỜI CAM ĐOAN .....	i
LỜI CẢM ƠN .....	iii
MỤC LỤC.....	iv
DANH MỤC CÁC HÌNH.....	vi
DANH MỤC CÁC BẢNG.....	viii
MỞ ĐẦU.....	1
Chương 1 .....	2
TỔNG QUAN VỀ XỬ LÝ ẢNH VÀ ỨNG DỤNG XỬ LÝ ẢNH .....	2
TRONG CÔNG NGHIỆP .....	2
1.1 Tổng quan về lý thuyết xử lý ảnh.....	2
1.1.1 Giới thiệu một hệ thống xử lý ảnh .....	2
1.1.2 Các vấn đề cơ bản trong xử lý ảnh.....	6
1.1.2.1 Khái niệm .....	6
1.1.2.2 Các vấn đề cơ bản trong xử lý ảnh.....	9
1.1.3 Một số công việc thông dụng trong xử lý ảnh .....	13
1.2. Ứng dụng xử lý ảnh trong công nghiệp .....	19
1.2.1 Chiết xuất thông tin dạng số liệu từ ảnh .....	19
1.2.2 Nhận dạng đối tượng.....	22
1.2.2.1 Nhận dạng ảnh dựa trên phân hoạch không gian .....	22
1.2.2.2 Nhận dạng ảnh dựa trên cấu trúc.....	28
Chương 2.....	31
MỘT SỐ KỸ THUẬT PHÁT HIỆN LỖI SẢN PHẨM .....	31
2.1. Giới thiệu bài toán kiểm tra sản phẩm .....	31

2.1.1 Dây chuyền sản xuất nước đóng chai.....	31
2.1.2 Bài toán kiểm tra sản phẩm bị lỗi bằng camera .....	32
2.2 Các thuật toán dùng xử lý ảnh để kiểm tra sản phẩm .....	37
2.2.1 Thuật toán dùng xử lý ảnh để kiểm tra nắp của chai .....	37
2.2.2 Thuật toán dùng xử lý ảnh để kiểm tra thể tích của chai .....	41
2.2.2.1 Phân tích nhiệm vụ.....	41
2.2.2.2 Thuật toán thực hiện.....	42
2.2.3 Thuật toán dùng xử lý ảnh để kiểm tra nhãn của chai .....	43
2.2.3.1 Phân tích nhiệm vụ.....	43
2.2.3.2 Thuật toán thực hiện.....	43
Chương 3. THỰC NGHIỆM.....	44
3.1 Thiết bị thu ảnh công nghiệp camera Eye-RIS .....	44
3.3.1 Phần cứng.....	44
3.3.2 Phần mềm.....	46
3.3.2.1 Phần mềm điều khiển Eye-RIS ADK 10.2 .....	46
3.3.2.2 Một số hàm thông dụng của Eye-RIS ADK 10.2 .....	54
3.2 Mô tả dây chuyền nước đóng chai .....	60
3.2.1 Mô tả thực nghiệm dây chuyền nước đóng chai .....	60
3.2.2 . Bố trí camera, chiếu sáng.....	61
3.3 Các kết quả thực nghiệm.....	62
3.4 Nhận xét, đánh giá.....	65
KẾT LUẬN .....	66
TÀI LIỆU THAM KHẢO.....	67

## DANH MỤC CÁC HÌNH

Hình 1. 1. Các bước cơ bản trong xử lý ảnh .....	3
Hình 1. 2. Biểu diễn ảnh với độ phân giải khác nhau .....	7
Hình 1. 3. Ảnh biến dạng do nhiễu .....	10
Hình 1. 4. Lấy tổ hợp các điểm ảnh lân cận.....	16
Hình 1. 5. Lược đồ mức xám của ảnh.....	18
Hình 1. 6. Mô hình camera pinhole .....	19
Hình 1. 7. Quan hệ giữa vật thực và ảnh.....	20
Hình 1. 8. Các hệ tọa độ trên một hệ thống có camera để xác định vị trí đối tượng.....	21
Hình 2. 1. Máy thổi khí .....	31
Hình 2. 2. Nhân viên kiểm tra thủ công sản phẩm.....	32
Hình 2. 3. Kiểm tra chai nước bằng xử lý ảnh .....	33
Hình 2. 4. Phân ngưỡng để có ảnh nhị phân và các vị trí kiểm tra trên ảnh...	34
Hình 2. 5. Ảnh của chai nước trên dây chuyền không phải lúc nào cũng thu được đầy đủ: a) thu đầy đủ; b) thu không đầy đủ; c), d), e), f): các trường hợp trong cửa sổ tính toán. g), h), i), k): Ảnh nhị phân của c), d), e), f).....	36
Hình 2. 6. Phân ngưỡng ảnh nhị phân để kiểm tra nắp của sản phẩm.....	38
Hình 2. 7. Thu mảng 2 chiều có kích thước 6x7 .....	38
Hình 2. 8. Thuật toán 1 kiểm tra tồn tại của nút chai.....	39
Hình 2. 9. Thuật toán 2 kiểm tra tồn tại của nút chai.....	40
Hình 2. 10. Ảnh nhị phân thu được để kiểm tra thể tích của chai .....	41
Hình 2. 11. Thuật toán kiểm tra thể tích nước ngọt trong chai.....	42
Hình 2. 12. Kiểm tra có nhãn trên vỏ chai hay không .....	43
Hình 3. 1. Camera tốc độ cao Eye- RIS .....	44
Hình 3. 2. Các cổng vào/ ra (I/ O port) trên Eye- RIS V2.1 .....	45

Hình 3. 3. Đầu nối các chân vào/ra của Eye-RIS V2.1.....	46
Hình 3. 4. Cài đặt phần mềm ứng dụng.....	47
Hình 3. 5. Chọn thư mục cài đặt .....	47
Hình 3. 6. Hoàn thành quá trình cài đặt .....	48
Hình 3. 7. Giao diện chính của phần mềm Eye-RIS ADK 10.2 .....	48
Hình 3. 8. Cấu trúc phần mềm lập trình cho Eye-RIS .....	49
Hình 3. 9. Chương trình Eye-RIS ADK có hai file Main.cpp và CFPPCode.fpp.....	50
Hình 3. 10. Thực hiện mã trong Eye-RIS .....	53
Hình 3. 11. Chọn kiểu chạy chương trình.....	54
Hình 3. 12. Thực nghiệm mô hình dây chuyên đóng chai nước.....	60
Hình 3. 13. Chai coca cola trên băng tải .....	61
Hình 3. 14. Chiếu sáng cho đối tượng .....	61
.Hình 3. 15. Ảnh chương trình kiểm tra thể tích: a) chai có nút và b) chai không có nút.....	62
Hình 3. 16. Kết quả chương trình kiểm tra nhãn .....	63
Hình 3. 17. Tính Histogram trung bình trong vùng tính toán VTT3 .....	63
Hình 3. 18. Kiểm tra liên tục trên dây chuyền .....	64

## DANH MỤC CÁC BẢNG

Bảng 3. 1. Chức năng các chân vào/ra.....	46
Bảng 3. 2. Kiểm tra thể tích .....	64
Bảng 3. 3. Tính Histogram và so sánh giá trị trung bình khi không có nhãn .....	65



## MỞ ĐẦU

Ngày nay, trong bối cảnh toàn cầu hóa, hội nhập kinh tế đang diễn ra mạnh mẽ và sự phát triển nhanh chóng của công nghệ thông tin và truyền thông, xu hướng phổ cập Internet, phát triển thương mại điện tử, Chính phủ điện tử... dẫn đến sự bùng nổ về công nghệ thông tin. Việc ứng dụng công nghệ thông tin và truyền thông vào thực tế đang phát triển mạnh mẽ, đặc biệt xử lý ảnh đã được nghiên cứu mạnh mẽ và được ứng dụng rất mạnh mẽ vào thực tế. Như trong y học, xử lý ảnh số đã được dùng để phát hiện và nhận dạng khối u, chụp cắt lớp, nhận dạng đường biên mạch máu từ những ảnh chụp mạch bằng tia X. Trong giao thông, dùng xử lý ảnh trong việc cải tiến hệ thống điều khiển đèn tín hiệu giao thông, giám sát xử phạt trật tự an toàn giao thông, kiểm tra biển số.... Trong Khoa học kỹ thuật, xử lý ảnh đang và đã có những đóng góp rất quan trọng.

Ngoài phần mở đầu và kết luận, nội dung của luận văn được trình bày trong 03 chương dưới đây:

Chương 1: Tổng quan về xử lý ảnh và ứng dụng xử lý ảnh trong công nghiệp

Chương 2: Một số kỹ thuật phát hiện lỗi sản phẩm

Chương 3: Thực nghiệm và kết quả

## Chương 1

# TỔNG QUAN VỀ XỬ LÝ ẢNH VÀ ỨNG DỤNG XỬ LÝ ẢNH TRONG CÔNG NGHIỆP

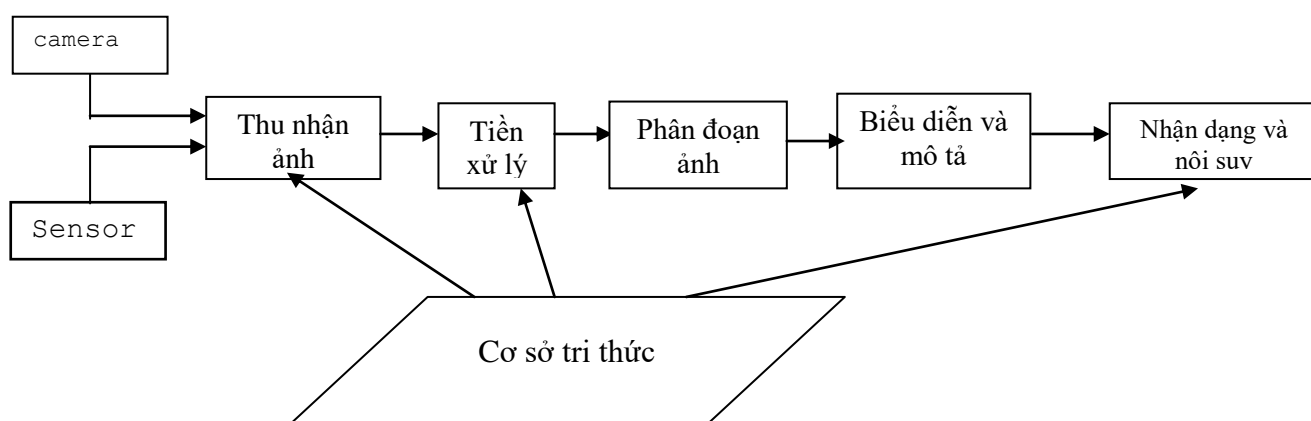
### 1.1 Tổng quan về lý thuyết xử lý ảnh

#### *1.1.1 Giới thiệu một hệ thống xử lý ảnh*

Xử lý ảnh [2], [3], [4], [5], [6] là một lĩnh vực mang tính khoa học và công nghệ cao. Nó có tốc độ phát triển nhanh và có tiềm năng ứng dụng rộng rãi trong nhiều lĩnh vực của cuộc sống như: trong y học, thiên văn, quân sự, công nghiệp [1], ...

Các phương pháp xử lý ảnh bắt đầu từ các ứng dụng chính như: nâng cao chất lượng ảnh và phân tích ảnh. Ứng dụng đầu tiên được biết đến là nâng cao chất lượng ảnh báo được truyền qua cáp từ Luân Đôn đến New York từ những năm 1920. Vấn đề nâng cao chất lượng ảnh có liên quan tới phân bố mức sáng và độ phân giải của ảnh. Việc nâng cao chất lượng ảnh được phát triển vào khoảng những năm 1955. Điều này có thể giải thích được vì sau thế chiến thứ hai, máy tính phát triển nhanh tạo điều kiện cho quá trình xử lý ảnh số thuận lợi. Năm 1964, máy tính đã có khả năng xử lý và nâng cao chất lượng ảnh từ mặt trăng và vệ tinh Ranger 7 của Mỹ bao gồm: làm nổi đường biên, lưu ảnh. Từ năm 1964 đến nay, các phương tiện xử lý, nâng cao chất lượng, nhận dạng ảnh và phát triển không ngừng. Các phương pháp tri thức nhân tạo như mạng nơ ron nhân tạo, các thuật toán xử lý hiện đại và cải tiến, các công cụ nén ảnh ngày càng được áp dụng rộng rãi và thu được nhiều kết quả khả quan.

Để có thể hình dung cấu hình của một hệ thống xử lý ảnh chuyên dụng hay một hệ thống xử lý ảnh dùng trong nghiên cứu, đào tạo, dưới đây là các bước cần thiết trong xử lý ảnh:



**Hình 1. 1. Các bước cơ bản trong xử lý ảnh**

Sơ đồ này bao gồm các thành phần chủ yếu sau đây:

*\* Phân thu nhận ảnh (Image Acquisition)*

Ảnh có thể nhận qua camera màu hoặc đen trắng. Thường ảnh nhận qua camera là ảnh tương tự (loại camera ống chuẩn CCIR với tần số 1/25, mỗi ảnh 25 dòng), cũng có loại camera đã được hiệu số hóa (loại CCD - Charge Coupled Device) là loại photodiode tạo cường độ sáng tại mỗi điểm ảnh. Ảnh cũng có thể thu nhận được từ vệ tinh qua các bộ cảm ứng (Sensor) hay ảnh, tranh được quét trên Scanner.

Camera thường dùng là loại quét dòng; ảnh tạo ra có dạng hai chiều. Chất lượng một ảnh thu được phụ thuộc vào thiết bị thu, vào môi trường (ánh sáng, phong cảnh).

*\* Tiền xử lý (Image Pre-processing)*

Sau bộ thu nhận, ảnh có thể nhiều độ tương phản thấp nên cần đưa vào bộ tiền xử lý để nâng cao chất lượng. Chức năng chính của bộ tiền xử lý là lọc nhiễu, nâng độ tương phản để làm ảnh rõ hơn, nét hơn.

*\* Phân đoạn (Segmentation) hay phân vùng ảnh*

Phân vùng ảnh là tách một ảnh đầu vào thành các vùng thành phần để biểu diễn phân tích, nhận dạng ảnh. Ví dụ: để nhận dạng chữ (hoặc mã vạch) trên phong bì thư cho mục đích phân loại bưu phẩm, cần chia các câu, chữ về

địa chỉ hoặc tên người thành các từ, các chữ, các số (hoặc các vạch) riêng biệt để nhận dạng. Đây là phần phức tạp khó khăn nhất trong xử lý ảnh và cũng dễ gây lỗi, làm mất độ chính xác của ảnh. Kết quả nhận dạng ảnh phụ thuộc rất nhiều vào công đoạn này.

*\* Biểu diễn ảnh (Image Representation)*

Đầu ra ảnh sau phân đoạn chứa các điểm ảnh của vùng ảnh (ảnh đã phân đoạn) cộng với mã liên kết với các vùng lân cận. Việc biến đổi các số liệu này thành dạng thích hợp là cần thiết cho xử lý tiếp theo bằng máy tính. Việc chọn các tính chất để thể hiện ảnh gọi là trích chọn đặc trưng (Feature Selection) gắn với việc tách các đặc tính của ảnh dưới dạng các thông tin định lượng hoặc làm cơ sở để phân biệt lớp đối tượng này với đối tượng khác trong phạm vi ảnh nhận được. Ví dụ: trong nhận dạng ký tự trên phong bì thư, chúng ta miêu tả các đặc trưng của từng ký tự giúp phân biệt ký tự này với ký tự khác.

*\* Nhận dạng và nội suy ảnh (Image Recognition and Interpretation)*

Nhận dạng ảnh là quá trình xác định ảnh. Quá trình này thường thu được bằng cách so sánh với mẫu chuẩn đã được học (hoặc lưu) từ trước. Nội suy là phán đoán theo ý nghĩa trên cơ sở nhận dạng. Ví dụ: một loạt chữ số và nét gạch ngang trên phong bì thư có thể được nội suy thành mã điện thoại. Theo lý thuyết về nhận dạng nói chung và nhận dạng ảnh nói riêng có 03 cách tiếp cận khác nhau như sau :

- Nhận dạng theo tham số (Nhận dạng dựa vào phân hoạch không gian).
- Nhận dạng theo cấu trúc.
- Nhận dạng dựa vào kỹ thuật mạng Noron.

Một số đối tượng nhận dạng khá phổ biến hiện nay đang được áp dụng trong khoa học và công nghệ là: nhận dạng ký tự (chữ in, chữ viết tay, chữ ký

điện tử), nhận dạng văn bản (Text), nhận dạng vân tay, nhận dạng mã vạch, nhận dạng mặt người...

*\* Cơ sở tri thức (Knowledge Base)*

Như đã nói ở trên, ảnh là một đối tượng khá phức tạp về đường nét, độ sáng tối, dung lượng điểm ảnh, môi trường để thu ảnh phong phú kéo theo nhiều. Trong nhiều khâu xử lý và phân tích ảnh ngoài việc đơn giản hóa các phương pháp toán học đảm bảo tiện lợi cho xử lý, người ta mong muốn bắt chước quy trình tiếp nhận và xử lý ảnh theo cách của con người. Trong các bước xử lý đó, nhiều khâu hiện nay đã xử lý theo các phương pháp trí tuệ con người. Vì vậy, ở đây các cơ sở tri thức được phát huy.

*\* Mô tả (biểu diễn ảnh)*

Từ Hình 1.1, ảnh sau khi số hoá sẽ được lưu vào bộ nhớ, hoặc chuyển sang các khâu tiếp theo để phân tích. Nếu lưu trữ ảnh trực tiếp từ các ảnh thô, đòi hỏi dung lượng bộ nhớ cực lớn và không hiệu quả theo quan điểm ứng dụng và công nghệ. Thông thường, các ảnh thô đó được đặc tả (biểu diễn) lại (hay đơn giản là mã hoá) theo các đặc điểm của ảnh được gọi là các đặc trưng ảnh (Image Features) như: biên ảnh (Boundary), vùng ảnh (Region). Một số phương pháp biểu diễn thường dùng:

• Biểu diễn bằng mã chạy (Run-Length Code)

Phương pháp này thường biểu diễn cho vùng ảnh và áp dụng cho ảnh nhị phân. Một vùng ảnh R có thể mã hoá đơn giản nhờ một ma trận nhị phân:

$$U(m, n) = 1 \text{ nếu } (m, n) \text{ thuộc } R$$

$$U(m, n) = 0 \text{ nếu } (m, n) \text{ không thuộc } R$$

Trong đó:  $U(m, n)$  là hàm mô tả mức xám ảnh tại tọa độ  $(m, n)$ . Với cách biểu diễn trên, một vùng ảnh được mô tả bằng một tập các chuỗi số 0 hoặc 1. Giả sử chúng ta mô tả ảnh nhị phân của một vùng ảnh được thể hiện theo tọa độ  $(x, y)$  theo các chiều và đặc tả chỉ đối với giá trị “1” khi đó dạng

mô tả có thể là:  $(x, y)r$ ; trong đó  $(x, y)$  là toạ độ,  $r$  là số lượng các bit có giá trị “1” liên tục theo chiều ngang hoặc dọc.

- Biểu diễn bằng mã xích (Chain-Code) .

Phương pháp này thường dùng để biểu diễn đường biên ảnh. Một đường bất kỳ được chia thành các đoạn nhỏ. Nội các điểm chia, ta có các đoạn thẳng kế tiếp được gán hướng cho đoạn thẳng đó tạo thành một dây xích gồm các đoạn. Các hướng có thể chọn 4, 8, 12, 24,... mỗi hướng được mã hoá theo số thập phân hoặc số nhị phân thành mã của hướng.

- Biểu diễn bằng mã tứ phân (Quad-Tree Code)

Phương pháp mã tứ phân được dùng để mã hoá cho vùng ảnh. Vùng ảnh đầu tiên được chia làm bốn phần thường là bằng nhau. Nếu mỗi vùng đã đồng nhất (chứa toàn điểm đen (1) hay trắng (0)), thì gán cho vùng đó một mã và không chia tiếp. Các vùng không đồng nhất được chia tiếp làm bốn phần theo thủ tục trên cho đến khi tất cả các vùng đều đồng nhất. Các mã phân chia thành các vùng con tạo thành một cây phân chia các vùng đồng nhất.

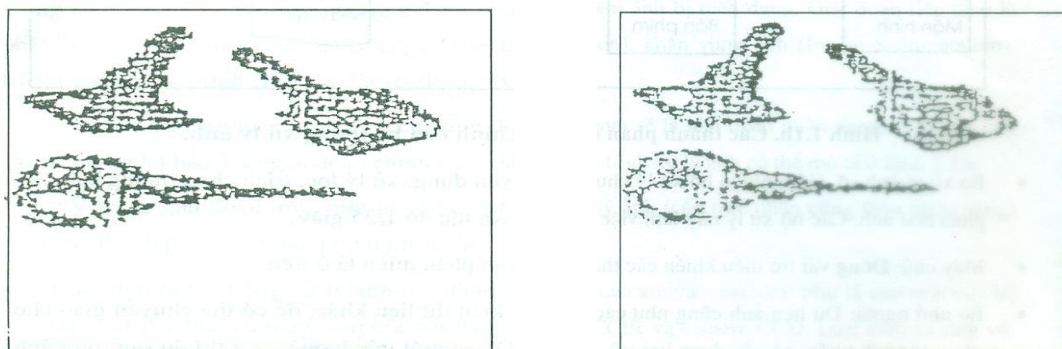
### **1.1.2 Các vấn đề cơ bản trong xử lý ảnh**

#### **1.1.2.1 Khái niệm**

\* *Pixel (Picture Element) : Phần tử ảnh hay điểm ảnh*

Ảnh trong thực tế là một ảnh liên tục về không gian và về giá trị độ sáng. Để có thể xử lý ảnh bằng máy tính cần thiết phải tiến hành số hoá ảnh. Trong quá trình số hoá, người ta biến đổi tín hiệu liên tục sang tín hiệu rời rạc thông qua quá trình lấy mẫu (rời rạc hóa về không gian) và lượng hoá thành phần giá trị mà về nguyên tắc bằng mắt thường không thể phân biệt được hai điểm kề nhau. Trong quá trình này, người ta sử dụng khái niệm Picture element mà ta quen gọi hay viết là Pixel - phần tử ảnh. Ở đây cũng cần phân biệt khái niệm pixel hay đề cập đến trong các hệ thống đồ hoạ máy tính. Để tránh nhầm lẫn ta tạm gọi khái niệm pixel này là pixel thiết bị. Khái niệm

pixel thiết bị có thể xem xét như sau: khi ta quan sát màn hình (trong chế độ đồ hoạ), màn hình không liên tục mà gồm nhiều điểm nhỏ, gọi là pixel. Mỗi pixel gồm một cặp tọa độ x, y và màu.



a) ảnh với độ phân giải 128 x 128

b) ảnh với độ phân giải 64 x 64

### Hình 1. 2. Biểu diễn ảnh với độ phân giải khác nhau

Cặp tọa độ x, y tạo nên *độ phân giải* (resolution). Như màn hình máy tính có nhiều loại với độ phân giải khác nhau: màn hình CGA có độ phân giải là 320 x 200; màn hình VGA là 640 x 350,...

Như vậy, một ảnh là một tập hợp các điểm ảnh. Khi được số hoá, nó thường được biểu diễn bởi bảng hai chiều  $I(n,p)$ : n dòng và p cột. Ta nói ảnh gồm n x p pixels. Người ta thường kí hiệu  $I(x,y)$  để chỉ một pixel. Thường giá trị của n chọn bằng p và bằng 256. Hình 1.2 cho ta thấy việc biểu diễn một ảnh với độ phân giải khác nhau. Một pixel có thể lưu trữ trên 1, 4, 8 hay 24 bit.

#### \* *Mức xám (Grey level)*

Mức xám là kết quả sự mã hóa tương ứng một cường độ sáng của mỗi điểm ảnh với giá trị số - kết quả của quá trình lượng hóa. Cách mã hóa kinh điển thường dùng 16,32 hay 64 mức. Mã hóa 256 mức là phổ dụng nhất do lý do kỹ thuật. Vì  $2^8 = 256$  (0,1, ..., 255), nên với 256 mức, mỗi điểm ảnh sẽ được mã hóa bởi 8 bit.

Ảnh có hai mức xám được gọi là ảnh nhị phân. Mỗi điểm ảnh của ảnh nhị phân chỉ có thể là 0 hoặc 1. Ảnh mức xám lớn hơn 2 được gọi là ảnh đa cấp xám hay ảnh màu.

Ảnh đen trắng là ảnh chỉ có hai màu đen và trắng, mức xám ở các điểm ảnh có thể khác nhau.

Với ảnh màu, có nhiều cách tổ hợp màu khác nhau. Theo lý thuyết màu do Thomas đưa ra từ năm 1802, mọi màu đều có thể tổ hợp từ 3 màu cơ bản: Red (đỏ), Green (lục) và Blue (Lam). Mỗi điểm ảnh của ảnh màu lưu trữ trong 3 bytes và do đó ta có  $2^{8 \times 3} = 2^{24}$  màu (cỡ 16,7 triệu màu).

Ảnh xám là ảnh chỉ có các mức xám. Thực chất màu xám là màu có các thành phần R, G, B trong hệ thống màu RGB có cùng cường độ. Tương ứng với mỗi điểm ảnh sẽ có một mức xám xác định.

#### *\* Độ phân giải của ảnh*

Độ phân giải ảnh là số điểm ảnh (pixel) có trên 1 đơn vị chiều dài của hình ảnh đó. Độ phân giải ảnh được tính bằng đơn vị ppi (pixels per inch) hoặc dpi (dots per inch).

Ví dụ: một hình ảnh có kích thước 1 inch x 1 inch và có độ phân giải 72 ppi sẽ chứa tổng cộng  $72 \times 72 = 5.184$  pixels. Hình ảnh có kích thước tương tự nhưng với độ phân giải 300 ppi sẽ chứa tổng cộng  $300 \times 300 = 90.000$  pixels. Hình bên trái có độ phân giải 72 ppi, hình bên phải 300 ppi.

Hình ảnh có độ phân giải càng cao thì càng sắc nét và màu sắc càng chính xác. Và khi đó, dung lượng file cũng sẽ tăng theo, đòi hỏi nhiều bộ nhớ và đĩa cứng hơn.

Hình ảnh sử dụng cho thiết kế web chỉ cần có độ phân giải 72 ppi. Trường hợp hình ảnh dùng cho thiết kế đồ họa in ấn thì bạn cần nhớ hai quy tắc sau: Để rửa ảnh kỹ thuật số thì hình ảnh cần có độ phân giải 300 ppi. Nếu là ảnh nét (line art) hoặc đơn sắc (monochrome) thì ảnh nên có độ phân giải là



1.200 ppi. Nếu là ảnh chụp màu (color photograph) hoặc ảnh chụp đen trắng (black and white photograph) thì ảnh nên có độ phân giải 300 ppi. Nếu in ảnh hi-flex với kích thước lớn (để quảng cáo ngoài trời chẳng hạn) thì hình ảnh cần có độ phân giải khoảng 72 ppi đến 100 ppi.

### **1.1.2.2 Các vấn đề cơ bản trong xử lý ảnh**

#### *\* Biểu diễn ảnh và mô hình hóa*

##### - Biểu diễn ảnh

Ảnh có thể xem là một hàm 2 biến chứa các thông tin như biểu diễn của một ảnh. Các mô hình biểu diễn ảnh cho ta một mô tả logic hay định lượng của hàm này. Dựa vào phần tử đặc trưng của ảnh cho ta một mô tả logic hay định lượng của hàm này. Dựa vào phần tử đặc trưng của ảnh là pixel. Giá trị Pixel có thể có giá trị vô hướng, hoặc là 1 vector (3 thành phần trong trường hợp ảnh màu RGB).

Một số mô hình thường được sử dụng trong biểu diễn ảnh: Mô hình toán học, mô hình thống kê.

+ Với mô hình thống kê: một ảnh được coi như một phần tử của một tập hợp đặc trưng bởi các đại lượng như: kỳ vọng toán học, hiệp biến, phương sai, moment.

+ Với mô hình biểu diễn ảnh bằng hàm toán học, hoặc các ma trận điểm. Trong mô hình toán học, ảnh hai chiều được biểu diễn nhờ các hàm hai biến.

##### - Mô hình hóa ảnh

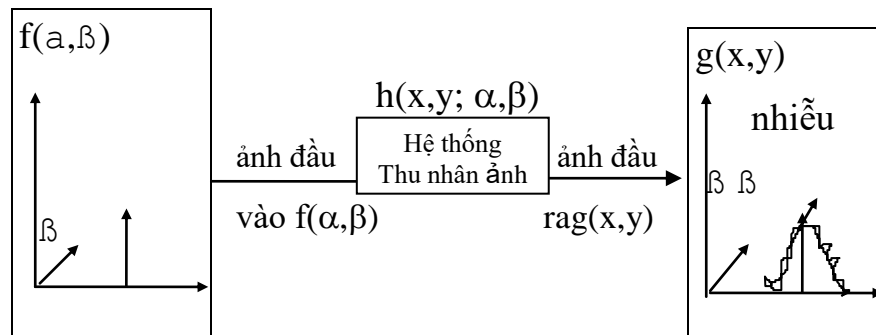
+ Mô hình cảm nhận ảnh: Là mô hình biểu diễn thông qua các thuộc tính cảm nhận ảnh (màu sắc, cường độ sáng), các thuộc tính về thời gian, các cảm nhận về phối cảnh, bố cục.

+ Mô hình cục bộ: Là mô hình biểu diễn thể hiện mối tương quan cục bộ của các phần tử ảnh (ứng dụng cho các bài toán xử lý ảnh và nâng cao chất lượng ảnh).

+ Mô hình tổng thể: Là mô hình biểu diễn ảnh xem ảnh như là một tập hợp các đối tượng và các đối tượng này có mối quan hệ không gian với nhau (ứng dụng cho các bài toán phân nhóm và nhận dạng ảnh).

\* *Tăng cường ảnh - khôi phục ảnh*

Tăng cường ảnh là bước quan trọng, tạo tiền đề cho xử lý ảnh. Nó gồm một loạt các kỹ thuật như: lọc độ tương phản, khử nhiễu, nổi màu, v...v.



**Hình 1.3. Ảnh biến dạng do nhiễu**

Hình 1.3 ở trên cho ta thí dụ về sự biến dạng của ảnh do nhiễu.

Khôi phục ảnh là nhằm loại bỏ các suy giảm (degradation) trong ảnh. Với một hệ thống tuyến tính, ảnh của một đối tượng có thể biểu diễn bởi:

$$g(x,y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} h(x,y;\alpha,\beta) f(\alpha,\beta) d\alpha d(\beta + \eta(x,y))$$

Trong đó:

- $\eta(x,y)$  là hàm biểu diễn nhiễu cộng.
- $f(a,\beta)$  là hàm biểu diễn đối tượng.
- $g(x,y)$  là ảnh thu nhận.
- $h(x,y; a,\beta)$  là hàm tán xạ điểm (Point Spread Function - PSF).

Một vấn đề khôi phục ảnh tiêu biểu là tìm một xấp xỉ của  $f(a,\beta)$  khi PSF của nó có thể đo lường hay quan sát được, ảnh mờ và các tính chất sắc xuất của quá trình nhiễu.

*\* Biến đổi ảnh*

Thuật ngữ biến đổi ảnh (Image Transform) thường dùng để nói tới một lớp các ma trận đơn vị và các kỹ thuật dùng để biến đổi ảnh. Cũng như các tín hiệu một chiều được biểu diễn bởi một chuỗi các hàm cơ sở, ảnh cũng có thể được biểu diễn bởi một chuỗi rời rạc các ma trận cơ sở gọi là *ảnh cơ sở*. Phương trình ảnh cơ sở có dạng:

$A^*_{k,l} = a_k a_l^{*T}$ , với  $a_k$  là cột thứ  $k$  của ma trận  $A$ .  $A$  là ma trận đơn vị. Có nghĩa là  $A A^{*T} = I$ . Các  $A^*_{k,l}$  định nghĩa ở trên với  $k,l = 0,1, \dots, N-1$  là ảnh cơ sở. Có nhiều loại biến đổi được dùng như :

- Biến đổi Fourier, Sin, Cosin, Hadamard, . . .
- Tích Kronecker (\*)
- Biến đổi KL (Karhunen Loeve): biến đổi này có nguồn gốc từ khai triển của các quá trình ngẫu nhiên gọi là phương pháp trích chọn các thành phần chính.

Do phải xử lý nhiều thông tin, các phép toán nhân và cộng trong khai triển là khá lớn. Do vậy, các biến đổi trên nhằm làm giảm thứ nguyên của ảnh để việc xử lý ảnh được hiệu quả hơn.

(\*) Trong xử lý ảnh, việc phân tích có thể được đơn giản hơn khá nhiều do làm việc với ma trận khối gọi là tích Kronecker.

- Ma trận khối là ma trận mà các phần tử của nó lại là một ma trận.

$$\begin{bmatrix} A_{11} & A_{12} & \dots & A_{1,n} \\ \dots & \dots & \dots & \dots \\ A_{m1} & A_{m2} & \dots & A_{mn} \end{bmatrix}$$

**Ma trận A**

với  $A_{i,j}$  là ma trận  $m \times n$ ;  $i = 1, 2, \dots, m$  và  $j = 1, 2, \dots, n$ .

- Tích Kronecker

Cho  $A$  là ma trận kích thước  $M_1 \times M_2$  và  $B$  ma trận kích thước  $N_1 \times N_2$ .

Tích Kronecker của  $A$  và  $B$  ký hiệu là  $A \otimes B$  là ma trận khối được định nghĩa:

$$A \otimes B = \begin{bmatrix} a_{1,1}B & a_{1,2}B & \dots & a_{1,M_2}B \\ \dots & \dots & \dots & \dots \\ a_{M_1,1}B & a_{M_1,2}B & \dots & a_{M_1,M_2}B \end{bmatrix}$$

với  $a_{ij}$  là các phần tử của ma trận  $A$ .

Thí dụ

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

**ma trận A**

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

**ma trận B**

$$\text{thì } A \otimes B = \begin{bmatrix} 1 & 2 & 1 & 2 \\ 3 & 4 & 3 & 4 \\ 1 & 2 & -1 & -2 \\ 3 & 4 & -3 & -4 \end{bmatrix}$$

*\* Phân tích ảnh*

Phân tích ảnh liên quan đến việc xác định các độ đo định lượng của một ảnh để đưa ra một mô tả đầy đủ về ảnh. Các kỹ thuật được sử dụng ở đây nhằm mục đích xác định biên của ảnh. Có nhiều kỹ thuật khác nhau như lọc vi phân hay dò theo quy hoạch động.

Người ta cũng dùng các kỹ thuật để phân vùng ảnh. Từ ảnh thu được, người ta tiến hành kỹ thuật tách (split) hay hợp (fusion) dựa theo các tiêu chuẩn đánh giá như: màu sắc, cường độ, v...v. Các phương pháp được biết đến như Quad-Tree, mảnh hoá biên, nhị phân hoá đường biên. Cuối cùng, phải kể đến các kỹ thuật phân lớp dựa theo cấu trúc.

*\* Nhận dạng ảnh*

Nhận dạng ảnh là quá trình liên quan đến các mô tả đối tượng mà người ta muốn đặc tả nó. Quá trình nhận dạng thường đi sau quá trình trích chọn các đặc tính chủ yếu của đối tượng. Có hai kiểu mô tả đối tượng:

- Mô tả tham số (nhận dạng theo tham số).
- Mô tả theo cấu trúc (nhận dạng theo cấu trúc).

Trên thực tế, người ta đã áp dụng kỹ thuật nhận dạng khá thành công với nhiều đối tượng khác nhau như: nhận dạng ảnh vân tay, nhận dạng chữ (chữ cái, chữ số, chữ có dấu).

Nhận dạng chữ in hoặc đánh máy phục vụ cho việc tự động hoá quá trình đọc tài liệu, tăng nhanh tốc độ và chất lượng thu nhận thông tin từ máy tính.

Nhận dạng chữ viết tay (với mức độ ràng buộc khác nhau về cách viết, kiểu chữ, v...,v ) phục vụ cho nhiều lĩnh vực.

Ngoài 2 kỹ thuật nhận dạng trên, hiện nay một kỹ thuật nhận dạng mới dựa vào kỹ thuật mạng nơ ron đang được áp dụng và cho kết quả khả quan.

*\* Nén ảnh*

Dữ liệu ảnh cũng như các dữ liệu khác cần phải lưu trữ hay truyền đi trên mạng. Như đã nói ở trên, lượng thông tin để biểu diễn cho một ảnh là rất lớn. Trong phần 1.1 chúng ta đã thấy một ảnh đen trắng cỡ 512 x 512 với 256 mức xám chiếm 256K bytes. Do đó làm giảm lượng thông tin hay nén dữ liệu là một nhu cầu cần thiết. Nhiều phương pháp nén dữ liệu đã được nghiên cứu và áp dụng cho loại dữ liệu đặc biệt này.

**1.1.3 Một số công việc thông dụng trong xử lý ảnh**

**\* Tích chập cuộn:**

- Xếp chồng tại biên

$$Y(m,n) = \sum_{k=0}^{L-1} \sum_{l=0}^{L-1} H(k,l) * X(m-k,n-l) \quad (1.1)$$

Theo công thức này, nếu  $K=L=3$ , nhân chập  $H$  có thể viết:

$$H(k,l) = \begin{bmatrix} H_{00} & H_{01} & H_{02} \\ H_{10} & H_{11} & H_{12} \\ H_{20} & H_{21} & H_{22} \end{bmatrix}$$

- Xếp chồng tại trung tâm

$$Y(m,n) = \sum_{k=1}^L \sum_{l=1}^L H(k,l) * X(m-k+Lc, n-l+Lc) \text{ với } Lc = \frac{L+1}{2} \quad (1.2)$$

Thực tế, công thức này có thể áp dụng cho cả hai trường hợp. Nếu áp dụng để tính cho điểm ở biên, ta coi các điểm ngoài biên có giá trị 0.

Thí dụ cho ảnh số  $I$  sau:

$$I = \begin{bmatrix} 4 & 7 & 2 & 7 & 1 \\ 5 & 7 & 1 & 7 & 1 \\ 6 & 6 & 1 & 8 & 3 \\ 5 & 7 & 5 & 7 & 1 \\ 5 & 7 & 6 & 1 & 2 \end{bmatrix}$$

Và nhân chập  $H$ :

$$H = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Tích chập  $H \otimes I$  tính theo công thức 1.2 ta được như sau:

$$H \otimes I = \begin{bmatrix} 23 & 26 & 31 & 19 & 16 \\ 35 & 39 & 46 & 31 & 27 \\ 36 & 43 & 49 & 34 & 27 \\ 36 & 43 & 48 & 34 & 12 \\ 24 & 35 & 33 & 22 & 11 \end{bmatrix}$$

Tích chập là một khái niệm rất quan trọng trong xử lý ảnh, đặc biệt là tính chất của nó có liên quan đến biến đổi Fourier: biến đổi Fourier của một tích chập bằng tích đơn giản các biến đổi Fourier của tín hiệu đó:

$$F[H(x,y) \otimes I(x,y)] = F[H(x,y)]. F[I(x,y)] \quad (1.3)$$

Trong kỹ thuật, người ta gọi H là nhân chập hay nhân cuộn và cũng còn là mặt nạ (mask); I [x,y] trong công thức trên là ảnh đối tượng.

Dưới đây, đưa ra một thuật toán tổng quát để tính nhân chập dùng cho mọi trường hợp. Để sử dụng thuật toán này chỉ cần thay đổi 2 thông số: ma trận biểu diễn ảnh số cần xử lý và ma trận biểu diễn nhân chập.

Thuật toán được mô tả dưới dạng Pascal như sau:

NhanChap(ImagIn,ImagOut: ảnh; H: Nhân chập; N: kích thước ảnh ;  
w:kích thước nhân chập)

/\* Vao: ImagIn

Nhân chập H

Ra: ImagOut \*/

Begin

For i:=1 to N do

For j:=1 to N do

Begin Sum :=0; Lc:=(w+1) div 2;

For k:=1 to w do

For l:=1 to w do

Begin Col:=i-k+Lc;Row:=j+l+Lc

If (Col<>0)and (Col <=N) then

If (Row<>0)and (Row <=N) then

Sum:= Sum + ImagIn[Col,Row] \* H[k,l];

End;

ImagOut[i,j]:=Sum

End;

End;

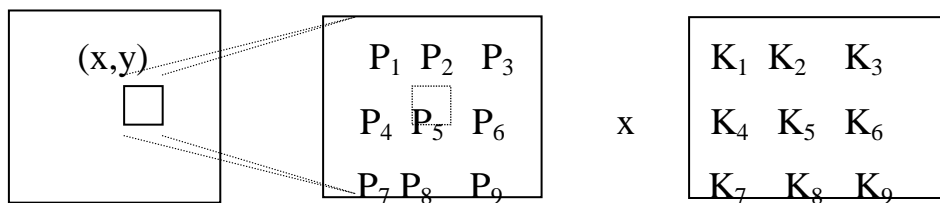
### \* Thực hiện bộ lọc

Cơ sở lý thuyết của kỹ thuật lọc số là dựa trên tính dư thừa thông tin không gian: các Pixel lân cận có thể có cùng hoặc gần cùng một số đặc tính. Hơn nữa, nhiều có thể coi như sự đột biến của một điểm ảnh so với các điểm lân cận. Ta có thể sử dụng một trong 2 loại lọc như: lọc tuyến tính và lọc phi tuyến.

- Lọc tuyến tính: ảnh thu được sẽ là tổng trọng số hay là trung bình trọng số các điểm lân cận với nhân chập cuộn hay mặt nạ. Nguyên tắc lọc theo tổng trọng số được minh họa qua hình 1.4.

Thí dụ: Tâm mặt nạ là điểm  $P_5$ , thì điểm  $P_5$  mới sẽ được tính theo công thức sau:

$$P_5 = P_1K_1 + P_2K_2 + P_3K_3 + P_4K_4 + P_5K_5 + P_6K_6 + P_7K_7 + P_8K_8 + P_9K_9$$



8 lân cận của  $P_5$  Nhân cuộn  $3 * 3$

**Hình 1. 4. Lấy tổ hợp các điểm ảnh lân cận**

Nói chung, người ta sử dụng nhiều kiểu mặt nạ khác nhau:

$$H_1 = \frac{1}{9} \begin{vmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{vmatrix} \quad H_2 = \frac{1}{10} \begin{vmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{vmatrix} \quad H_3 = \frac{1}{16} \begin{vmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{vmatrix}$$

Mặt nạ  $H_1$  là mặt nạ dùng để tính trung bình không trọng số (không ưu tiên theo hướng nào cả). Mặt nạ  $H_2$  cho trọng số lớn nhất với điểm ở tâm. Còn mặt nạ  $H_3$  ưu tiên cho hướng x,y.



Giả sử  $I_i$  là ảnh đang xét và  $I_f$  là ảnh thu được và cả 2 ảnh đều có cùng kích thước  $p \times p$ . Với mặt nạ trên, mỗi điểm ảnh thu được  $I_f(x,y)$  sẽ được tính bởi:

$$\begin{aligned} I_f &= \frac{1}{9} \{ I_i(x-1,y-1) + I_i(x-1,y) + I_i(x-1,y+1) + I_i(x,y-1) + I_i(x,y) + I_i(x,y+1) \\ &\quad + I_i(x+1,y-1) + I_i(x,y) + I_i(x+1,y+1) \} \\ &= \frac{1}{9} \sum_{i=-1}^1 \sum_{j=-1}^1 H_1(i+1,j+1) I_i(x+i,y+j) \end{aligned} \quad (1.4)$$

Nếu  $H$  là bộ lọc kích thước  $(n+1) \times (n+1)$ ,  $n$  chẵn và tổng các hệ số là  $K$ ,  $I_f$  sẽ được tính bởi:

$$I_f = \frac{1}{K} \sum_{i=-n/2}^{n/2} \sum_{j=-n/2}^{n/2} H_1(i+n/2,j+n/2) I_i(x+i,y+j) \quad (1.5)$$

Công thức trên chính là tích chập giữa mặt nạ  $H$  và ảnh gốc  $I$ :  $I_f = H \otimes I_i$ .

- Lọc phi tuyến: Khác với lọc tuyến tính, kỹ thuật lọc phi tuyến coi một điểm ảnh kết quả không phải là tổ hợp tuyến tính của các điểm lân cận. Bộ lọc phi tuyến thường dùng là lọc trung vị mang tên Tuckey. Trong trường hợp một chiều, trung vị  $x$  của một chuỗi phần tử  $\{x_n\}$  được định nghĩa:

Nếu  $n$  lẻ: có  $(n-1)/2$  phần tử  $x_a$  và  $(n-1)/2$  nhỏ hơn hay bằng  $x_a$ .

Nếu  $n$  chẵn:  $x_a$  là trung bình cộng của 2 phần tử  $x_i$  và  $x_j \in \{x_n\}$  sao cho có  $(n-2)/2$  phần tử nhỏ hơn hay bằng  $x_i$  và  $(n-2)/2$  phần tử lớn hơn hay bằng  $x_j$

*Một cách tổng quát ta có thuật toán tìm lọc phi tuyến như sau:*

B1. Lấy các phần tử trong cửa sổ ra mảng một chiều (  $L$  phần tử).

B2. Tìm Min của lần lượt các chuỗi con rồi lấy max: gọi  $m_1$  là giá trị này.

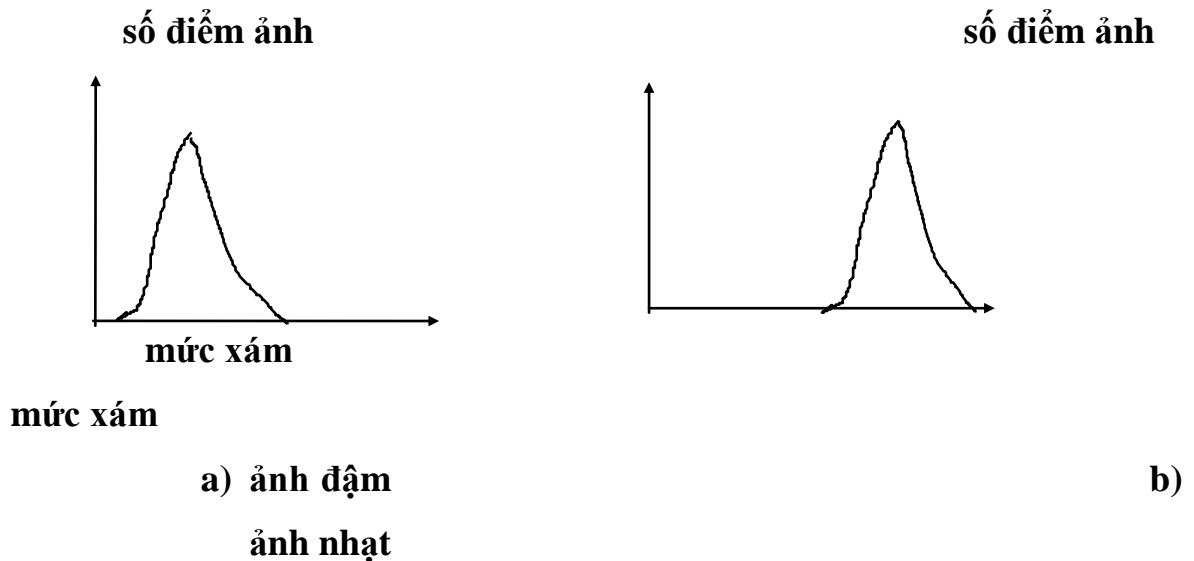
B3: Tìm Max của lần lượt các chuỗi con rồi lấy min: gọi  $m_2$  là giá trị tìm được.

B4. Gán giá trị điểm đang xét là trung bình cộng của  $m_1$  và  $m_2$ .

#### \* Lược đồ mức xám

Lược đồ mức xám của một ảnh ta quy ước là lược đồ mức xám, là một hàm cung cấp tần suất xuất hiện của mỗi mức xám.

Lược đồ xám được biểu diễn trong một hệ tọa độ vuông góc  $x,y$ . Trong hệ tọa độ này, trục hoành biểu diễn số mức xám từ 0 đến  $N$ ,  $N$  là số mức xám (256 mức trong trường hợp đang xét). Trục tung biểu diễn số điểm ảnh cho một mức xám (số điểm ảnh có cùng mức xám). Cũng có thể biểu diễn khác một chút: trục tung là tỷ lệ số điểm ảnh có cùng mức xám trên tổng số điểm ảnh.



**Hình 1. 5. Lược đồ mức xám của ảnh**

Thuật toán xây dựng lược đồ mức xám có thể được mô tả như sau :

Bắt đầu

$H$  là bảng chứa lược đồ xám ( là vectơ có  $N$  phần tử)

a. Khởi tạo bảng

Đặt tất cả các phần tử của bảng là 0.

b. Tạo bảng

Với mỗi điểm ảnh  $I(x,y)$  tính  $H[I(x,y)] = H[I(x,y)] + 1$

c. Tính giá trị Max của bảng  $H$ . Sau đó hiện bảng trong khoảng 0 đến Max.

Kết thúc

Lược đồ là một công cụ hữu hiệu dùng trong nhiều công đoạn của xử lý ảnh như tăng cường ảnh.

## 1.2. Ứng dụng xử lý ảnh trong công nghiệp

### 1.2.1 Chiết xuất thông tin dạng số liệu từ ảnh

Để có thể thu được các số liệu và chuyển thành dữ liệu ảnh không có một phương thức chung cho tất cả các trường hợp bởi vì tùy theo vị trí tương đối của thiết bị thu ảnh (camera) với đối tượng thu, tùy theo độ méo hình học của thiết bị thu, số camera sử dụng mà sẽ có những công thức quan hệ tính toán khác nhau. Nhưng các bước chung nhất thường sử dụng là:

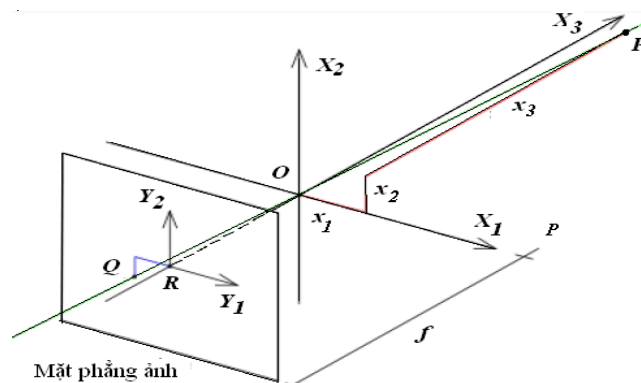
- Hiệu chỉnh các thông số của camera (calibration) để có được ảnh đúng
- Dùng các công thức quan hệ hình học và số liệu kỹ thuật kích thước tấm sensor của camera để tính ra các số liệu kích thước về độ dài, rộng, cao, về góc lệch, góc nghiêng thực sự.

Ví dụ: Sử dụng mô hình camera Pinhole để xác định khoảng cách từ vị trí đặt camera đến đối tượng. Trong ví dụ này để đơn giản sử dụng các giả thiết: Mặt phẳng chứa tấm sensor của camera và mặt phẳng chứa ảnh của đối tượng song song với nhau.

Trục quang của ống kính camera trùng với pháp tuyến từ tâm hình học của đối tượng trong mặt phẳng chứa ảnh 2D của nó.

Độ phân giải cả camera và kích thước tấm sensor cảm quang của camera đã biết.

Mô hình camera pinhole tiêu chuẩn:

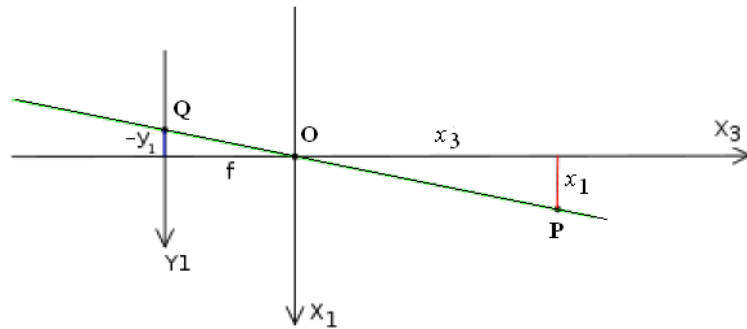


**Hình 1. 6. Mô hình camera pinhole**

Trong mô hình này O là điểm cửa chắn sáng của ống kính được chọn làm gốc tọa độ. Trục  $X_3$  là trục quang học (hoặc trục chính). Mặt phẳng ảnh cách mặt phẳng  $X_1X_2$  một khoảng  $f$  là tiêu cự. Trục  $X_3$  giao với mặt phẳng ảnh tại R. R là gốc tọa độ trên mặt phẳng ảnh. Một điểm P trong không gian 3D được chiếu lên mặt phẳng ảnh, cắt mặt phẳng này tại Q. Nó có các tọa độ  $(y_1, y_2)$  trong hệ tọa độ mặt phẳng ảnh. Như vậy một điểm trong không gian 3D đã được ánh xạ thành một ảnh 2D. Các tọa độ  $(y_1, y_2)$  phụ thuộc vào các tọa độ  $(x_1, x_2, x_3)$  như sau.

Theo hình vẽ xem xét quan hệ giữa  $x_1, x_3$  chúng ta có:

$$\frac{-y_1}{f} = \frac{x_1}{x_3} \text{ hoặc } y_1 = -f \frac{x_1}{x_3} \quad (1.6)$$



**Hình 1. 7. Quan hệ giữa vật thực và ảnh**

Tương tự xem xét quan hệ  $x_1, x_2$  có

$$\frac{-y_2}{f} = \frac{x_2}{x_3} \text{ hoặc } y_2 = -f \frac{x_2}{x_3} \quad (1.7)$$

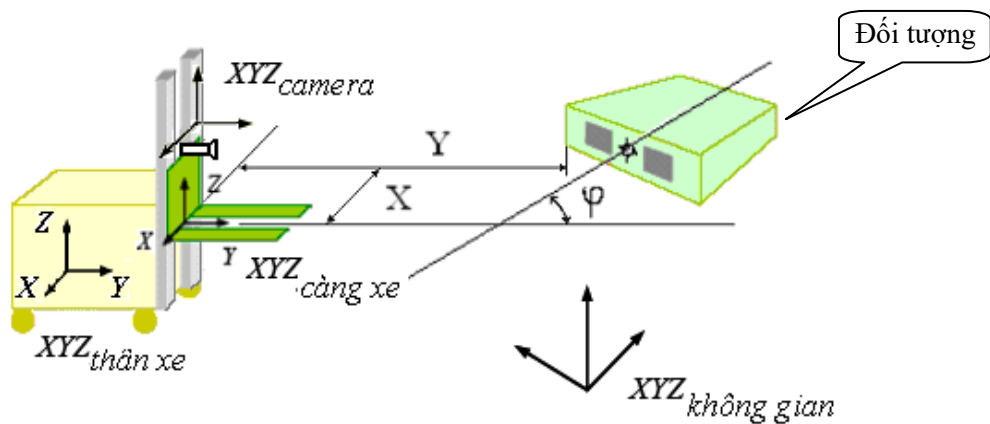
Trong thực tế các ảnh của camera được đảo lại để người sử dụng thuận tiện hơn và dấu trừ trong các công thức (1.6) và (1.7) trở thành dương. Chúng ta có thể viết tổng quát thành

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \frac{f}{x_3} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \quad (1.8)$$

Như vậy muốn tìm khoảng cách từ camera đến vật thực trên ảnh có thể dùng công thức (1.8) khi đã biết  $f$  và các kích thước thực  $x_1$  hoặc  $x_2$  của ảnh. Tuy nhiên khi sử dụng camera trên xe tự động thì còn cần phải có các phép biến đổi hệ tọa độ nữa.

Ví dụ: Một xe tự động nâng hàng (autonomous forklift) dùng cang móc vào hai lỗ của một đối tượng là một giá đỡ hàng (pallet) khi đó trong cặp các đối tượng này có các hệ tọa độ

**Các hệ tọa độ:**  $XYZ_{camera}$  là hệ tọa độ camera.  $XYZ_{cang xe}$  là hệ tọa độ của cang xe móc vào giá đỡ.  $XYZ_{thân xe}$  hệ tọa độ của thân xe.  $XYZ_{không gian}$  hệ tọa độ không gian chung. Giả sử giá đỡ hàng nằm trên mặt đất. Do trục Y của  $XYZ_{cang xe}$  và  $XYZ_{thân xe}$  song song nên góc  $\varphi$  để điều chỉnh xe và cang là như nhau. Các trục của camera pentil  $XYZ_{camera}$  có thể thay đổi so với các trục XYZ tương ứng của 3 hệ tọa độ kia.



**Hình 1. 8. Các hệ tọa độ trên một hệ thống có camera để xác định vị trí đối tượng**

Trong trường hợp mô hình tiêu chuẩn, mặt phẳng đối tượng và mặt phẳng sensor ảnh song song với nhau. Chúng ta có:

$$\frac{H}{D} = \frac{h}{f}$$

Từ đó chúng ta có:

$$D = f \frac{H}{h} = f \frac{H}{n.k} \quad (1.9)$$

$n$ : số pixel của ảnh theo hướng  $h$ .

$k$ : hệ số thể hiện số milimet/pixel.

$$k = \frac{W}{N} \quad (1.10)$$

$W$ : Kích thước của tấm sensor ảnh

$N$ : Độ phân giải của sensor ảnh.

Trong thực tế khi sử dụng máy tính để xử lý ảnh; chúng ta sử dụng màn hình để hiển thị ảnh, khi đó chúng ta đã ánh xạ tấm sensor lên màn hình máy tính,  $W$  là kích thước của màn hình theo hướng tính toán.

### **1.2.2 Nhận dạng đối tượng**

Trong lý thuyết nhận dạng nói chung và nhận dạng ảnh nói riêng có 3 cách tiếp cận khác nhau:

- Nhận dạng ảnh dựa vào phân hoạch không gian.
- Nhận dạng ảnh dựa vào cấu trúc.
- Nhận dạng ảnh dựa vào kỹ thuật mạng nơ ron.

#### **1.2.2.1 Nhận dạng ảnh dựa trên phân hoạch không gian**

Trong kỹ thuật này, các đối tượng nhận dạng là các đối tượng định lượng. Mỗi đối tượng được biểu diễn bởi một vectơ nhiều chiều. Trước tiên, ta xem xét một số khái niệm như: phân hoạch không gian, hàm phân biệt sau đó sẽ đi vào một số kỹ thuật cụ thể.

##### *\* Phân hoạch không gian*

Giả sử không gian đối tượng  $X$  được định nghĩa :  $X = \{X_i, i=1, 2, \dots, m\}$ ,  $X_i$  là một vectơ. Người ta nói  $p$  là một phân hoạch của không gian  $X$  thành các lớp  $C_i$ ,  $C_i \subset X$  nếu:

$$C_i \cap C_j = \emptyset \text{ với } i \neq j \text{ và } \cup C_i = X$$

Nói chung, đây là trường hợp lý tưởng: tập  $X$  tách được hoàn toàn. Trong thực tế, thường gặp không gian biểu diễn tách được từng phần. Như

vậy phân loại là dựa vào việc xây dựng một ánh xạ  $f: X \rightarrow p$ . Công cụ xây dựng ánh xạ này là các hàm phân biệt (Discriminant functions).

\* *Hàm phân lớp hay hàm ra quyết định*

Để phân đôi tượng vào các lớp, ta phải xác định số lớp và ranh giới giữa các lớp đó. Hàm phân lớp hay hàm phân biệt là một công cụ rất quan trọng. Gọi  $\{g_i\}$  là lớp các hàm phân lớp. Lớp hàm này được định nghĩa như sau:

nếu  $\forall i \neq k, g_k(X) > g_i(X)$  thì ta quyết định  $X \in$  lớp  $k$ .

Như vậy để phân biệt  $k$  lớp, ta cần  $k-1$  hàm phân biệt. Hàm phân biệt  $g$  của một lớp nào đó thường dùng là hàm tuyến tính, có nghĩa là:

$$g(X) = W_0 + W_1 X_1 + W_2 X_2 + \dots + W_k X_k \quad (1.11)$$

trong đó:

- $W_i$  là các trọng số gán cho các thành phần  $X_i$ .
- $W_0$  là trọng số để viết cho gọn.

Trong trường hợp  $g$  là tuyến tính, người ta nói là việc phân lớp là tuyến tính hay siêu phẳng (hyperplan).

Các hàm phân biệt thường được xây dựng dựa trên khái niệm khoảng cách hay dựa vào xác suất có điều kiện.

Lẽ tự nhiên, khoảng cách là một công cụ rất tốt để xác định xem đối tượng có "gần nhau" hay không. Nếu khoảng cách nhỏ hơn một ngưỡng  $\tau$  nào đấy ta coi 2 đối tượng là giống nhau và gộp chúng vào một lớp. Ngược lại, nếu khoảng cách lớn hơn ngưỡng, có nghĩa là chúng khác nhau và ta tách thành 2 lớp.

Trong một số trường hợp, người ta dựa vào xác suất có điều kiện để phân lớp cho đối tượng. Lý thuyết xác suất có điều kiện được Bayes nghiên cứu khá kỹ và chúng ta có thể áp dụng lý thuyết này để phân biệt đối tượng.

Gọi :  $P(X/C_i)$  là xác suất để có  $X$  biết rằng có xuất hiện lớp  $C_i$

$P(C_i/X)$  là xác suất có điều kiện để  $X$  thuộc lớp  $C_i$ .

với  $X$  là đối tượng nhận dạng,  $C_i$  là các lớp đối tượng.

Quá trình học cho phép ta xác định  $P(X/C_i)$  và nhờ công thức Bayes về xác suất có điều kiện áp dụng trong điều kiện nhiều biến, chúng ta sẽ tính được  $P(C_i/X)$  theo công thức:

$$P(C_i/X) = \frac{P(X/C_i)P(C_i)}{\sum_{i=1}^n P(C_i/X)P(C_i)} = \frac{P(X/C_i)P(C_i)}{P(X)} \quad (1.12)$$

Nếu  $P(C_i/X) > P(C_k/X)$  với  $\forall i \neq k$  thì  $X \in C_i$ . Tùy theo các phương pháp nhận dạng khác nhau, hàm phân biệt sẽ có các dạng khác nhau.

*\* Nhận dạng thống kê*

Nếu các đối tượng nhận dạng tuân theo luật phân bố Gauss, mà hàm mật độ xác suất cho bởi:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-m)^2}{2\pi\sigma^2}\right) \quad (1.13)$$

Người ta có dùng phương pháp ra quyết định dựa vào lý thuyết Bayes. Lý thuyết Bayes thuộc loại lý thuyết thống kê nên phương pháp nhận dạng dựa trên lý thuyết Bayes có tên là phương pháp thống kê.

**Quy tắc Bayes**

- Cho không gian đối tượng  $X = \{X_l, l=1, 2, \dots, L\}$ , với  $X_l = \{x_1, x_2, \dots, x_p\}$
- Cho không gian diễn dịch  $\Omega = \{C_1, C_2, \dots, C_r\}$ ,  $r$  là số lớp

Quy tắc Bayes phát biểu như sau:

$\varepsilon: X \rightarrow \Omega$  sao cho  $X \in C_k$  nếu  $P(C_k/X) > P(C_l/X) \forall l \neq k, l=1, 2, \dots, r$ .

Trường hợp lý tưởng là nhận dạng luôn đúng, có nghĩa là không có sai số. Thực tế, luôn tồn tại sai số  $\varepsilon$  trong quá trình nhận dạng. Vấn đề ở đây là xây dựng quy tắc nhận dạng với sai số  $\varepsilon$  là nhỏ nhất.



Phương pháp ra quyết định với  $\varepsilon$  tối thiểu

Ta xác định  $X \in C_k$  nhờ xác suất  $P(C_k/X)$ . Vậy nếu có sai số, sai số sẽ được tính bởi  $1 - P(C_k/X)$ . Để đánh giá sai số trung bình, người ta xây dựng một ma trận  $L(r,r)$  giả thiết là có  $n$  lớp.

Ma trận  $L$  được định nghĩa như sau:

$$L_{k,j} = \begin{cases} l_{k,j} > 0 \text{ nếu } k \neq j \text{ (tồn tại sai số)} \\ l_{k,j} \leq 0 \text{ nếu } k = j \text{ (không có sai số)} \end{cases} \quad (1.14)$$

Như vậy, sai số trung bình của sự phân lớp sẽ là:

$$rk(X) = \sum_{j=1}^r l_{k,j} P(C_j / X) \quad (1.15)$$

Để sai số là nhỏ nhất ta cần có  $rk$  là min. Từ công thức 1.11 và 1.14 ta có:

$$r_k(X) = \frac{\sum_{j=1}^r l_{k,j} P(X / C_j)}{P(C_j)} \quad (1.16)$$

Vậy, quy tắc ra quyết định dựa trên lý thuyết Bayes có tính đến sai số được phát biểu như sau:

$$X \in C_k \text{ nếu } \rho_k < \rho_p \text{ với } p \neq k, p=1, 2, \dots, r. \quad (1.17)$$

với  $\rho_k$  là  $r_k(X)$ .

Trường hợp đặc biệt với 2 lớp  $C_1$  và  $C_2$ , ta dễ dàng có:

$$X \in C_1 \text{ nếu } P(X/C_1) > \frac{l_{12} - l_{22}}{l_{11} - l_{21}} \frac{P(C_2)}{P(C_1)} P(X / C_2) \quad (1.18)$$

Giả sử thêm rằng xác suất phân bố là đều ( $P(C_1) = P(C_2)$ ), sai số là như nhau ta có:

$$X \in C_1 \text{ nếu } P(X/C_1) > P(X/C_2) \quad (1.19)$$

\* Một số thuật toán nhận dạng tiêu biểu trong tự học

Thực tế có nhiều thuật toán nhận dạng học không có thầy. Ở đây, chúng ta xem xét 3 thuật toán hay được sử dụng: Thuật toán nhận dạng dựa vào khoảng cách lớn nhất, thuật toán K - trung bình (K mean) và thuật toán

ISODATA. Dưới đây chỉ xét các thuật toán này vì chúng có bước tiếp nối, cải tiến từ thuật toán này qua thuật toán khác.

\* *Thuật toán dựa vào khoảng cách lớn nhất*

- Nguyên tắc

Cho một tập gồm  $m$  đối tượng. Ta xác định khoảng cách giữa các đối tượng và khoảng cách lớn nhất ứng với phần tử xa nhất tạo nên lớp mới. Sự phân lớp được hình thành dần dần dựa vào việc xác định khoảng cách giữa các đối tượng và các lớp.

- Thuật toán

Bước 1: + Chọn hạt nhân ban đầu: giả sử  $X_1 \in C_1$  gọi là lớp  $g_1$ . Gọi  $Z_1$  là phần tử trung tâm của  $g_1$ .

+ Tính tất cả các khoảng cách  $D_{j1} = D(X_j, Z_1)$  với  $j = 1, 2, \dots, m$

+ Tìm  $D_{k1} = \max_j D_{j1}$ .  $X_k$  là phần tử xa nhất của nhóm  $g_1$ . Như vậy  $X_k$  là phần tử trung tâm của lớp mới  $g_2$ , kí hiệu  $Z_2$ .

+ Tính  $d_1 = D_{12} = D(Z_1, Z_2)$ .

Bước 2:

+ Tính các khoảng cách  $D_{j1}, D_{j2}$ .

+  $D_{j1} = D(X_j, Z_1), D_{j2} = D(X_j, Z_2)$ . Đặt  $D_k^{(2)} = \max_j D_j$

Nguyên tắc chọn

+ Nếu  $D_k^{(2)} < \theta d_1$  kết thúc thuật toán. Phân lớp xong.

+ Nếu không, sẽ tạo nên nhóm thứ ba. Gọi  $X_k$  là phần tử trung tâm của  $g_3$ , kí hiệu  $Z_3$ .

+ Tính  $d_3 = (D_{12} + D_{13} + D_{23})/3$

với  $\theta$  là ngưỡng cho trước và  $D_{13} = D(Z_1, Z_3), D_{23} = D(Z_2, Z_3)$ .

Quá trình cứ lặp lại như vậy cho đến khi phân xong. Kết quả là ta thu được các lớp với các đại diện là  $Z_1, Z_2, \dots, Z_m$ .

\* Thuật toán K trung bình ( giả sử có K lớp)

- Nguyên tắc

Khác với thuật toán trên, ta xét K phần tử đầu tiên trong không gian đối tượng, hay nói một cách khác ta cố định K lớp. Hàm để đánh giá là hàm khoảng cách Euclide:

$$J_k = \sum_{X \in G_k} D(X, Z_k) = \sum_{j=1}^k D_2(X_j, Z_k) \quad (1.20)$$

$J_k$  là hàm chỉ tiêu với lớp  $C_k$ . Việc phân vùng cho k hạt nhân đầu tiên được tiến hành theo nguyên tắc khoảng cách cực tiểu. Ở đây, ta dùng phương pháp đạo hàm để tính cực tiểu.

Xét  $\frac{\partial J_k}{\partial Z_k} = 0$  với  $Z_k$  là biến. Ta dễ dàng có (1.20) min khi:

$$\sum_{i=1}^N (X_i - Z_k) = 0 \implies Z_k = \frac{1}{N_c} \sum_{j=1}^{N_c} Z_j \quad (1.21)$$

Công thức (1.21) là giá trị trung bình của lớp  $C_k$  và điều này lý giải tên của phương pháp.

- Thuật toán

Chọn  $N_c$  phần tử (giả thiết có  $N_c$  lớp) của tập T. Gọi các phần tử trung tâm của các lớp đó là:  $X_1, X_2, \dots, X_{N_c}$  và ký hiệu là  $Z_1, Z_2, \dots, Z_{N_c}$ .

Thực hiện phân lớp

$X \in C_k$  nếu  $D(X, Z_k) = \text{Min } D(X, Z_j)^{(1)}$ ,  $j=1, \dots, N_c$ . (1) là lần lặp thứ nhất.

Tính tất cả  $Z_k$  theo công thức (1.21).

Tiếp tục như vậy cho đến bước q.

$$X \in G_k(q-1) \text{ nếu } D(X, Z_k(q-1)) = \min_1 D(X, Z_l(q-1)).$$

Nếu  $Z_k(q-1) = Z_k(q)$  thuật toán kết thúc, nếu không ta tiếp tục thực hiện phân lớp.

*\* Thuật toán ISODATA*

ISODATA là viết tắt của từ Interactive Self Organizing Data Analysis. Nó là thuật toán khá mềm dẻo, không cần cố định các lớp trước. Các bước của thuật toán được mô tả như sau:

- Lựa chọn một phân hoạch ban đầu dựa trên các tâm bất kỳ. Thực nghiệm đã chứng minh kết quả nhận dạng không phụ thuộc vào phân lớp ban đầu.
  - Phân vùng bằng cách sắp các điểm vào tâm gần nhất dựa vào khoảng cách Euclide.
  - Tách đôi lớp ban đầu nếu khoảng cách lớn hơn ngưỡng  $t_1$ .
  - Xác định phân hoạch mới trên cơ sở các tâm vừa xác định lại và tiếp tục xác định tâm mới.
  - Tính tất cả các khoảng cách đến tâm mới.
  - Nhóm các vùng với tâm theo ngưỡng  $t_2$ .
- Lặp các thao tác trên cho đến khi thỏa tiêu chuẩn phân hoạch.

**1.2.2.2 Nhận dạng ảnh dựa trên cấu trúc**

*\* Biểu diễn định tính*

Ngoài cách biểu diễn theo định lượng như đã mô tả ở trên, tồn tại nhiều kiểu đối tượng mang tính định tính. Trong cách biểu diễn này, người ta quan tâm đến các dạng và mối quan hệ giữa chúng. Giả thiết rằng mỗi đối tượng được biểu diễn bởi một dãy ký tự. Các đặc tính biểu diễn bởi cùng một số ký tự. Phương pháp nhận dạng ở đây là nhận dạng lôgic, dựa vào hàm phân biệt là hàm Bool. Cách nhận dạng là nhận dạng các từ có cùng độ dài.

Giả sử hàm phân biệt cho mọi ký hiệu là  $g_a(x)$ ,  $g_b(x)$ , ..., tương ứng với các ký hiệu a, b, .... Để dễ dàng hình dung, ta giả sử có từ "abc" được biểu diễn bởi một dãy ký tự  $X = \{x_1, x_2, x_3, x_4\}$ . Tính các hàm tương ứng với 4 ký tự và có:

$$g_a(x_1) + g_b(x_2) + g_c(x_3) + g_c(x_4)$$

Các phép cộng ở đây chỉ phép toán OR. Trên cơ sở tính giá trị cực đại của hàm phân biệt, ta Quyết định X có thuộc lớp các từ "abc" hay không. Trong cách tiếp cận này, đối tượng tương đương với câu.

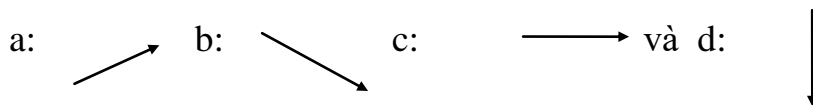
*\* Phương pháp ra quyết định dựa vào cấu trúc*

Thủ tục phân loại và nhận dạng ở đây gồm 2 giai đoạn: Giai đoạn đầu là giai đoạn xác định các quy tắc xây dựng, tương đương với việc nghiên cứu một văn phạm trong một ngôn ngữ chính thống. Giai đoạn tiếp theo khi đã có văn phạm là xem xét tập các dạng có được sinh ra từ các dạng đó không? Nếu nó thuộc tập đó coi như ta đã phân loại xong. Tuy nhiên, văn phạm là một vấn đề lớn. Trong nhận dạng cấu trúc, ta mới chỉ sử dụng được một phần rất nhỏ mà thôi.

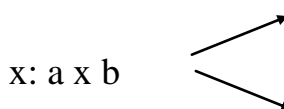
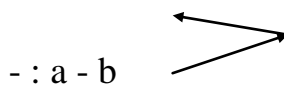
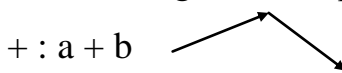
Như trên đã nói, mô hình cấu trúc tương đương một văn phạm  $G : G = \{V_n, V_t, P, S\}$ . Có rất nhiều kiểu văn phạm khác nhau từ chính tắc, phi ngữ cảnh,... một ngôn ngữ có thể được áp dụng trong nhận dạng cấu trúc: đó là ngôn ngữ PLD (Picture Language Description).

Ví dụ: Ngôn ngữ PLD

Trong ngôn ngữ này, các từ vựng là các vạch có hướng. Có 4 từ vựng cơ bản:



Các từ vựng trên các quan hệ được định nghĩa như sau:



$*$  :  $a * b$



Văn phạm sinh ra các mô tả trong ngôn ngữ được định nghĩa bởi:

$$GA = \{V_n, V_T, P, S\}$$

với  $V_n = \{A, B, C, D, E\}$  và  $V_T = \{a, b, c, d\}$ . S là ký hiệu bắt đầu và P là tập luật sản xuất.

Ngôn ngữ này thường dùng nhận dạng các mạch điện.

*\* Phương pháp nhận dạng*

Các đối tượng cần nhận dạng theo phương pháp này được biểu diễn bởi một câu trong ngôn ngữ  $L(G)$ . Khi đó thao tác phân lớp chính là xem xét một đối tượng có thuộc văn phạm  $L(G)$  không? Nói cách khác nó có được sinh ra bởi các luật của văn phạm  $G$  không? Như vậy sự phân lớp là theo cách tiếp cận cấu trúc đòi hỏi phải xác định:

- Tập  $V_t$  chung cho mọi đối tượng.
- Các quy tắc sinh P để sản sinh ra một câu và chúng khác nhau đối với mỗi lớp.
- Quá trình học với các câu biểu diễn các đối tượng mẫu l nhằm xác định văn phạm G.
- Quá trình ra quyết định: xác định một đối tượng X được biểu diễn bởi một câu  $l_x$ . Nếu  $l_x$  nhận biết bởi ngôn ngữ  $L(G_x)$  thì ta nói rằng  $X \in C_k$ .

Nói cách khác, việc ra quyết định phân lớp là dựa vào phân tích  $G_k$  biểu diễn lớp  $C_k$ . pháp của văn phạm. Cũng như trong phân tích cú pháp ngôn ngữ, có phân tích trên xuống, dưới lên, việc nhận dạng theo cấu trúc cũng có thể thực hiện theo cách tương tự.

## Chương 2

### MỘT SỐ KỸ THUẬT PHÁT HIỆN LỖI SẢN PHẨM

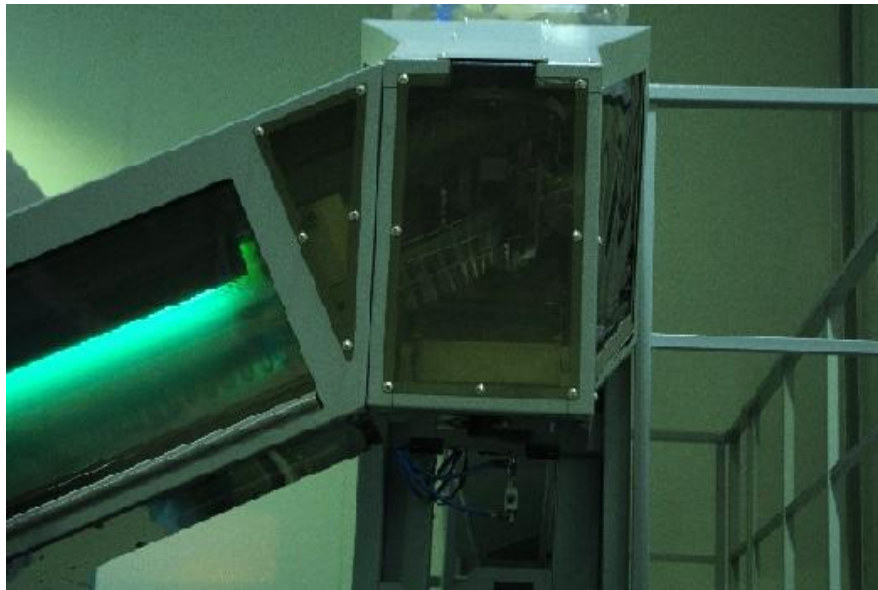
#### 2.1. Giới thiệu bài toán kiểm tra sản phẩm

##### 2.1.1 Dây chuyền sản xuất nước đóng chai

Hệ thống dây chuyền sản xuất nước đóng chai trải qua các giai đoạn: từ khâu thổi chai, chiết rót, đóng nắp, dán nhãn đến đóng thùng.

##### \* *Khâu thổi chai*

Phôi chai được đưa vào máy thổi qua các công đoạn gia nhiệt ở 120 độ C và thổi hình bằng khí nén. Ngay cả không khí đưa vào bên trong buồng chiết vô trùng phải qua ba cấp lọc: lọc thô, lọc tinh và lọc vô trùng để loại bỏ toàn bộ vi sinh vật và nhằm đảm bảo không có bụi bẩn, côn trùng hay bất kỳ vật thể nào có thể lọt vào trong được.



Hình 2. 1. Máy thổi khí

\* *Khâu chiết rót:* Khi vào máy chiết, chai được dốc ngược và cùng với nắp chay qua thiết bị súc rửa hoàn toàn bằng nước được tiệt trùng ở nhiệt độ 135 độ C.

- Máy chiết rót chất lỏng hay còn gọi là máy định lượng sản phẩm lỏng đó chính là việc chiết một thể tích nhất định của sản phẩm lỏng và rót vào trong chai, bình, lọ, v.v.. Máy chiết rót sản phẩm lỏng được sử dụng rộng rãi trong nhiều ngành sản xuất thực phẩm.

- Máy định lượng chiết rót sản phẩm lỏng thường được áp dụng cho những trường hợp yêu cầu năng suất cao, hoặc các yêu cầu nghiêm ngặt về vệ sinh thực phẩm. Tùy theo tính chất của chất lỏng, các máy chiết rót sẽ khác nhau ở các bộ phận làm việc chính, các cơ cấu rót.

**\* Khâu kiểm tra sản phẩm: đóng nắp, dán nhãn và đóng thùng sản phẩm...**

Sau khi chai được chiết rót và đóng nắp sẽ có *nhân viên kiểm tra để loại ra tất cả sản phẩm bị lỗi* không đạt yêu cầu như: Chưa đóng nắp, chưa dán nhãn, chưa đủ thể tích...

### **2.1.2 Bài toán kiểm tra sản phẩm bị lỗi bằng camera**

Tại khâu kiểm tra sản phẩm đối tượng kiểm tra đang di chuyển trên một băng tải của dây chuyền, tốc độ chuyển động của chai trên dây chuyền với công suất 6-8 nghìn/giờ, nhân viên khó có thể kiểm tra được tất cả sản phẩm lỗi bằng mắt thường.



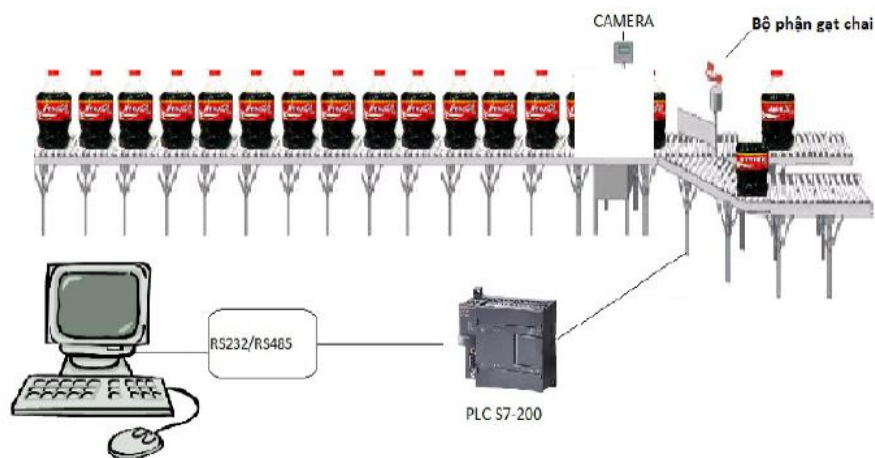
**Hình 2. 2. Nhân viên kiểm tra thủ công sản phẩm**



Việc ứng dụng công nghệ thông tin trong việc kiểm tra các sản phẩm bị lỗi như: có đủ thể tích không? chai có đóng nắp? có dán nhãn không? Chất lượng sản phẩm?... là vấn đề hết sức quan trọng và cần thiết. Tránh những sản phẩm không đủ tiêu chuẩn, chất lượng được lưu hành trên thị trường và ảnh hưởng đến uy tín của công ty.

Dưới đây bài toán chỉ giải quyết các chai bị lỗi như: có đủ mức nước, đã dán nhãn chưa và đã đóng nắp hay chưa? Do vậy việc ứng dụng công nghệ thông tin vào kiểm tra đồng thời trọng lượng, tình trạng nút và dán nhãn như sau:

Ở hình 2.3 chúng ta đặt camera bắt và xử lý ảnh tốc độ cao để thu ảnh của các chai nước. Dùng chương trình xử lý ảnh để xác định phát hiện các lỗi về nắp chai, thể tích chai và dán nhãn. Nếu sản phẩm nào có lỗi chương trình sẽ gửi tín hiệu báo lỗi đến PLC S7 200 (programmable logic controller là thiết bị điều khiển logic lập trình được, hay thiết bị logic khả trình cho phép thực hiện linh hoạt các thuật toán điều khiển logic thông qua một ngôn ngữ lập trình) để điều khiển hệ thống cơ khí gạt chai có lỗi đó ra để sửa lỗi thủ công (thêm mức nước, dán thêm nhãn, thêm nút),...



**Hình 2. 3. Kiểm tra chai nước bằng xử lý ảnh**

- Các bước kiểm tra đồng tình trạng nắp, nhãn và mức nước của chai như sau:

**Bước 1:** Thu ảnh xám, tăng cường ảnh, sử dụng hàm phân ngưỡng để chuyển thành ảnh Nhị phân.

**Bước 2:** Kiểm tra có nắp, mức nước, có nhãn hay không

- Kiểm tra có nắp hay không nếu không có nắp, đếm số lỗi không nắp để gạt phế phẩm (đếm số lỗi không nút tức là nhớ vị trí sản phẩm không có nắp tính từ sản phẩm đầu dây chuyền đến khi chạy đến sản phẩm đó sẽ gạt ra).

- Ngược lại, kiểm tra sản phẩm đó có đủ mức nước hay không nếu không đủ đếm số lỗi không đủ mức nước, gạt phế phẩm.

- Ngược lại kiểm tra sản phẩm đó có dán nhãn hay chưa nếu chưa dán nhãn thì đếm số lỗi không nhãn, gạt phế phẩm.

- Ngược lại chuyển sang bước 3

**Bước 3:** Đếm số sản phẩm đạt đã đủ trọng lượng, đã đóng nắp và có nhãn. Nếu sản phẩm không đạt thì kết thúc, ngược lại thì chuyển sang sản phẩm khác để kiểm tra tiếp, quay lại bước 2.

#### \* Phân tích bài toán

Sau khi đã thu được ảnh xám sử dụng hàm phân ngưỡng ta sẽ chuyển ảnh xám thành ảnh đen trắng.



**Hình 2. 4. Phân ngưỡng để có ảnh nhị phân và các vị trí kiểm tra trên ảnh**

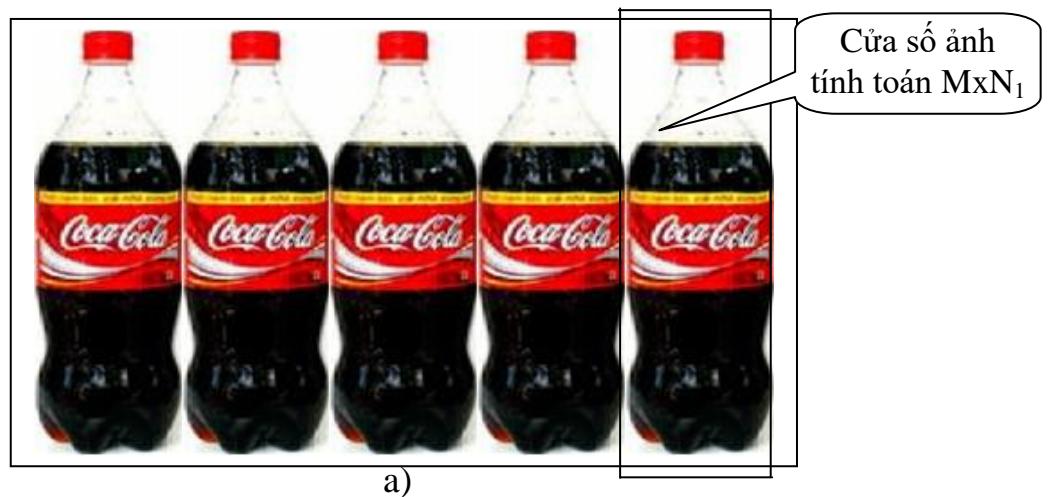
Đây là bài toán ứng dụng công nghệ thông tin trong sản xuất công nghiệp trên dây chuyền sản xuất tự động.

**Phân tích:**

Hệ thống camera đặt cố định chính diện với chai coca-cola chạy trên băng tải đảm bảo quan sát được nắp chai, mức nước và nhãn của chai. Khung nhìn của camera bao quát trong tầm nhìn các chai trên nền trắng. Quá trình thu và xử lý ảnh được thực hiện liên tục. Như vậy các ảnh thu được không phải lúc nào cũng đảm bảo là một ảnh đầy đủ của các chai coca-cola (hình 2.5a) mà có thể như hình 2.5b). Ta cần xem xét ảnh để xử lý và cho kết quả chính xác với chai đang được kiểm tra.

Giả sử ảnh thu được đầy đủ, rõ nét (camera chỉnh đúng tiêu cự, độ mở ống kính, ngưỡng chuyển ảnh từ gray sang binary hợp lý) như hình 2.5. Ảnh thu được sẽ có kích thước  $M \times N$  theo thông số kỹ thuật của camera. Vì từng chai nước phải được kiểm tra nên từ ảnh  $M \times N$  ta lấy ra một cửa sổ ảnh để tính toán  $M \times N_1$  với  $N_1 < N$  như hình 2.5a)

Trên cửa sổ tính toán ảnh chai nước thu được có thể có các trường hợp





b)

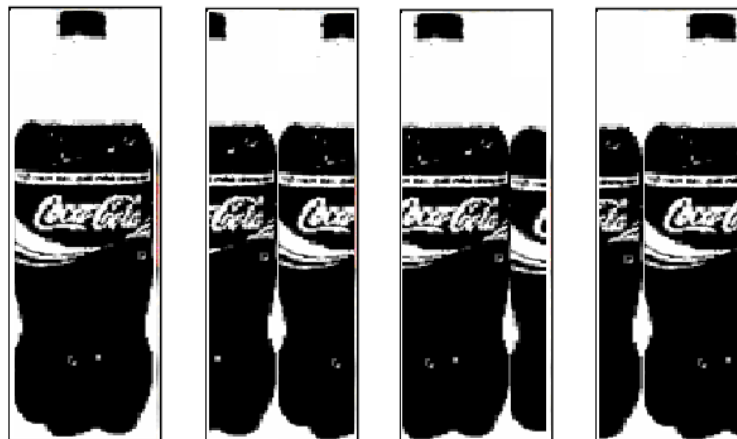


c)

d)

e)

f)



g)

h)

i)

k)

**Hình 2. 5.** Ảnh của chai nước trên dây chuyền không phải lúc nào cũng thu được đầy đủ: a) thu đầy đủ; b) thu không đầy đủ; c), d), e), f): các trường hợp trong cửa sổ tính toán. g), h), i), k): Ảnh nhị phân của c), d), e), f)

**Nhận xét:**

Ta thấy trong tất cả các trường hợp trong cửa sổ tính toán c), d), e), f) thì ta luôn thấy rằng sự có mặt của vùng ảnh thể hiện nút chai là luôn chắc chắn trong vùng tọa độ từ  $y = 0$  đến  $\frac{M}{H} * h + c$

Với  $h$ : là chiều cao thực của nút chai

và  $H$ : Chiều cao thực của toàn bộ chai.

$c$ : là một số nguyên  $> 0$ ;

Tuy nhiên do chai đang chuyển động nên ảnh của cái nút chai đó có thể là của cái chai trước ta đã kiểm tra rồi mà trong khi cái chai tiếp theo phần nút của nó còn chưa đi đến. Những sự phân tích này sẽ rất có ích cho chúng ta khi giải quyết bằng xử lý ảnh nhiệm vụ xác định: có nút hay không và có đủ thể tích hay không?

Một nhận xét quan trọng nữa rằng: Do các chai trên dây chuyền chuyển động sát nhau không có khoảng cách nên tổng số điểm đen của vùng tính toán ngoài khu vực dán nhãn là một con số tương đối ổn định. Với nhận xét này chúng ta có thể xác định Histogram của vùng dán nhãn thực tế để so sánh với Histogram của vùng này khi không có nhãn để rút ra kết luận rằng nhãn có được dán hay không trên vỏ chai nhựa.

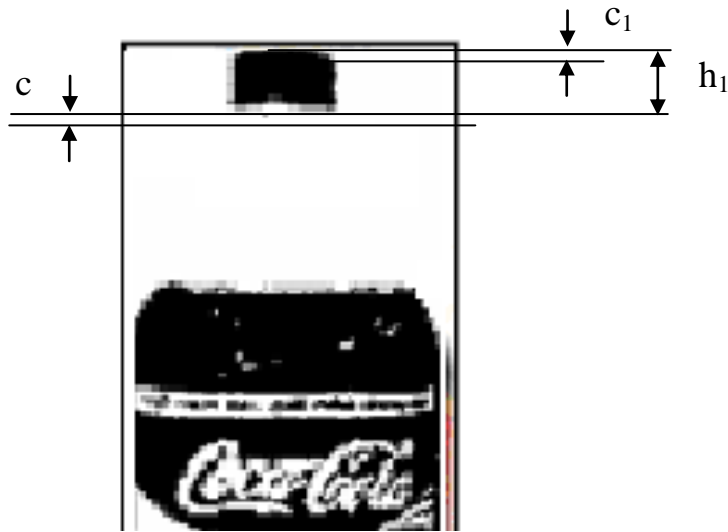
**2.2 Các thuật toán dùng xử lý ảnh để kiểm tra sản phẩm****2.2.1 Thuật toán dùng xử lý ảnh để kiểm tra nắp của chai**

**\*Thuật toán 1:** Như đã thấy trong phần phân tích, trong cửa sổ tính toán chúng ta thu được có thể là:

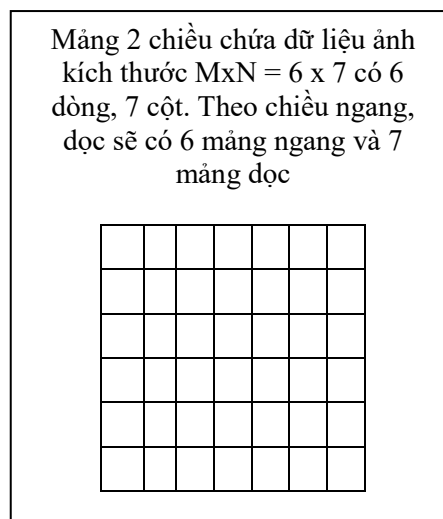
Ảnh toàn bộ của chai (hình 2.5g), hoặc Ảnh chai bắt đầu đi vào vùng cửa sổ tính toán (hình 2.5h), hoặc Ảnh của chai đi vào gần trọn vẹn vùng tính toán (hình 2.5i), hoặc Chai gần ra khỏi vùng tính toán (hình 2.5k), thì dù ở trường hợp nào trong vùng tọa độ  $y = 0$  đến  $\frac{M}{H} * h + c$  khi ta lấy  $\frac{M}{H} * h + c$  các

mảng ảnh ngang thì trong các mảng này khi khảo sát từ đầu đến cuối mỗi mảng luôn xuất hiện ít nhất một điểm đen (trường hợp h) khi có một chai đang vào vùng tính toán và đó là các pixel đầu tiên của nắp chai bắt đầu tiến vào).

Để bớt quá trình tính toán chúng ta có thể chỉ lấy 1 mảng ngang  $N_1$  phân tử liên nhau tại vị trí giữa nắp. Tức là các vị trí  $y = c + \frac{h_1 - c_1}{2}$



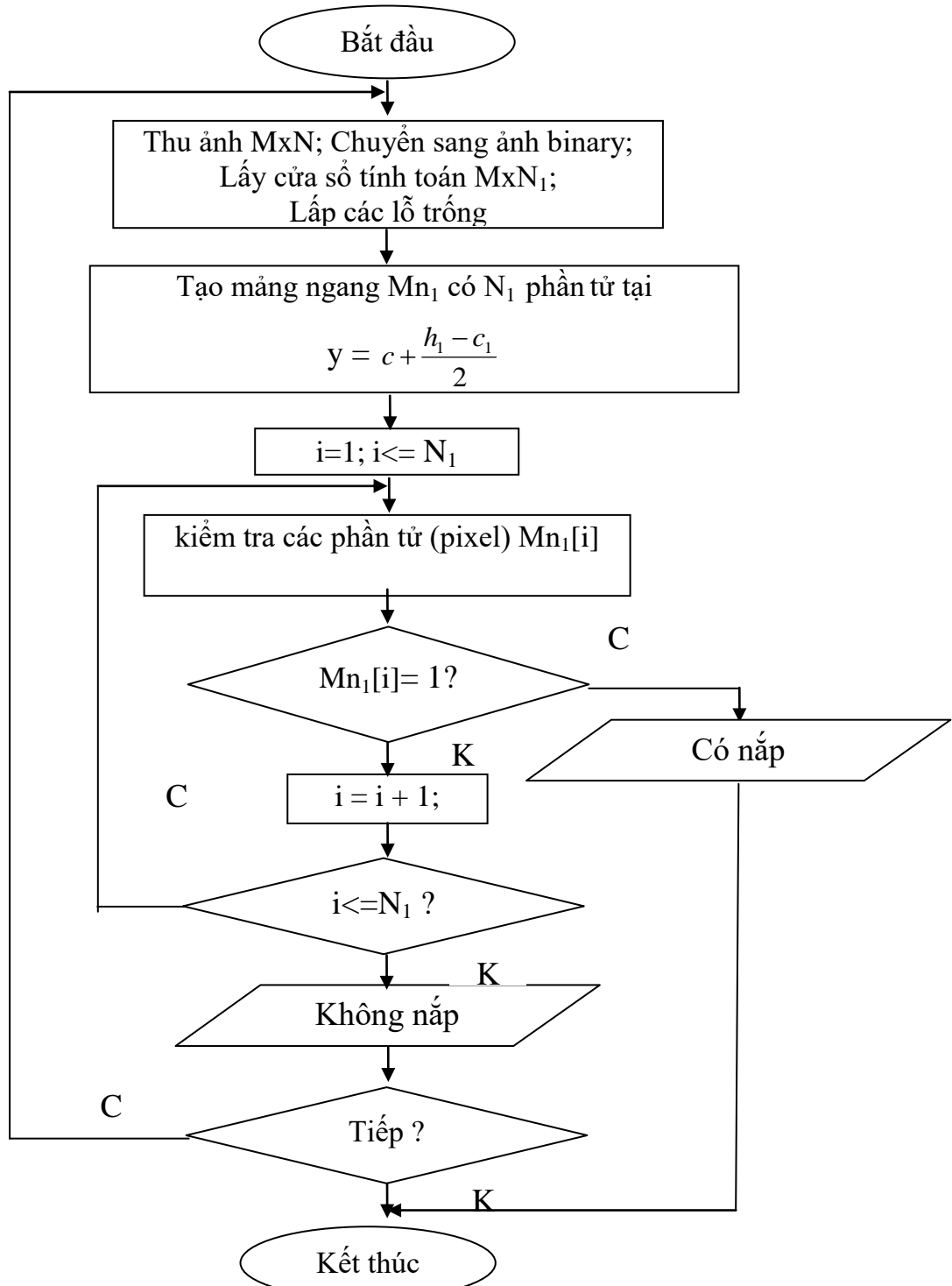
**Hình 2. 6. Phân ngưỡng ảnh nhị phân để kiểm tra nắp của sản phẩm**



**Hình 2. 7. Thu mảng 2 chiều có kích thước  $6 \times 7$**

Phương pháp này có thể chấp nhận được với sai số rất nhỏ khi ảnh được lấy ngưỡng sang nhị phân và lấp lỗ chưa tốt và chẳng may vùng ảnh đó rơi vào vùng tọa độ  $y = 0$  đến  $\frac{M}{H} * h + c$ .

**\* Sơ đồ thuật toán kiểm tra nắp của sản phẩm**

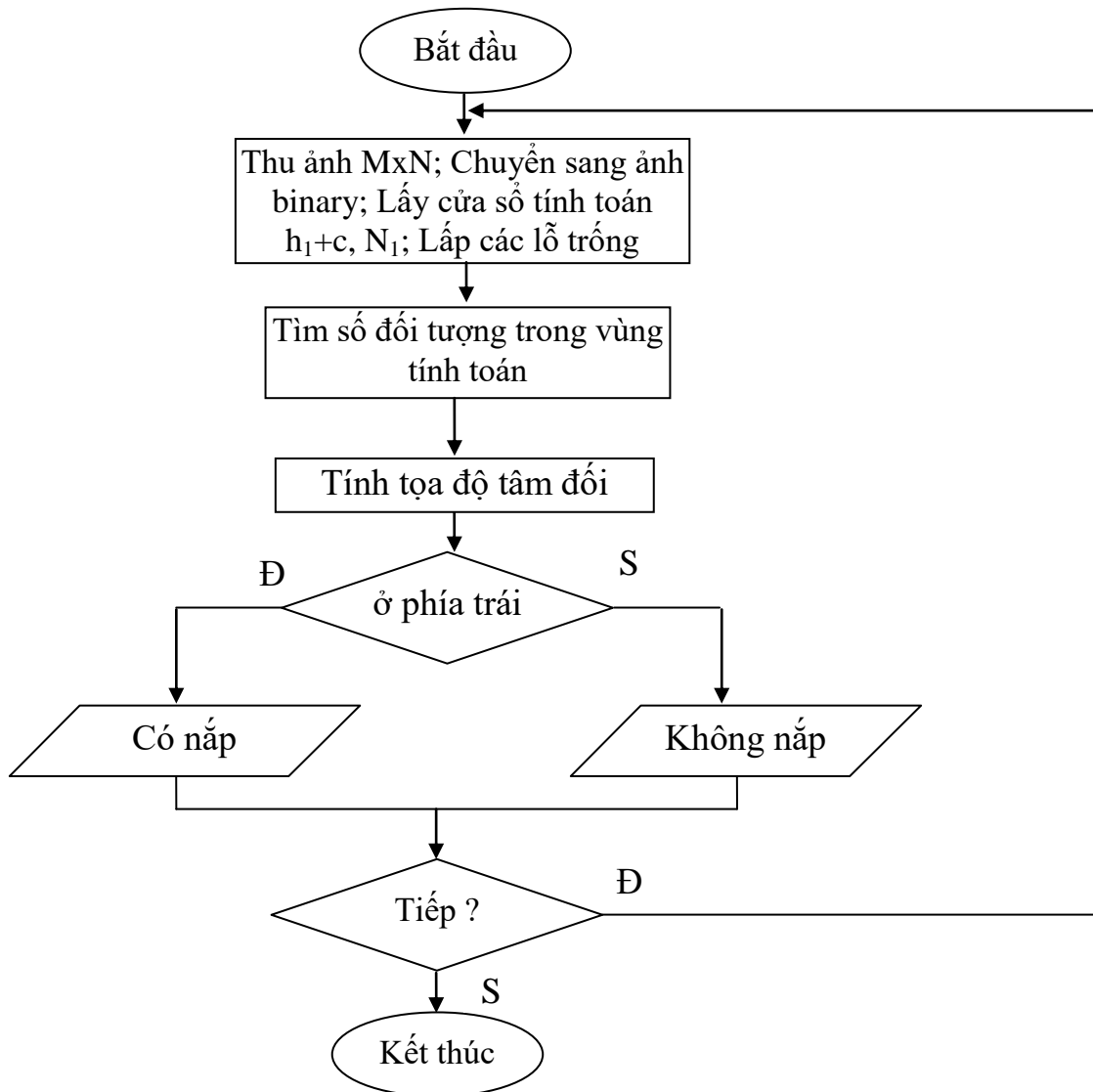


Số hóa bởi Trung tâm Học liệu – ĐHTN <http://www.lrc.tnu.edu.vn>

**Hình 2. 8. Thuật toán 1 kiểm tra tồn tại của nắp chai**

\* **Thuật toán 2:** Để có một phương pháp tốt hơn với độ chính xác tuyệt đối chúng ta nhận xét rằng: trong vùng tính toán với cửa sổ  $(h_1 + c, N_1)$  chúng ta luôn có ít nhất một đối tượng là ảnh của nắp chai. Như vậy trong cửa sổ này chúng ta kiểm tra số đối tượng và kiểm tra tọa độ tâm của chúng là chúng ta có thể rút ra kết luận về sự tồn tại của nắp chai.

\* **Sơ đồ thuật toán như sau:**



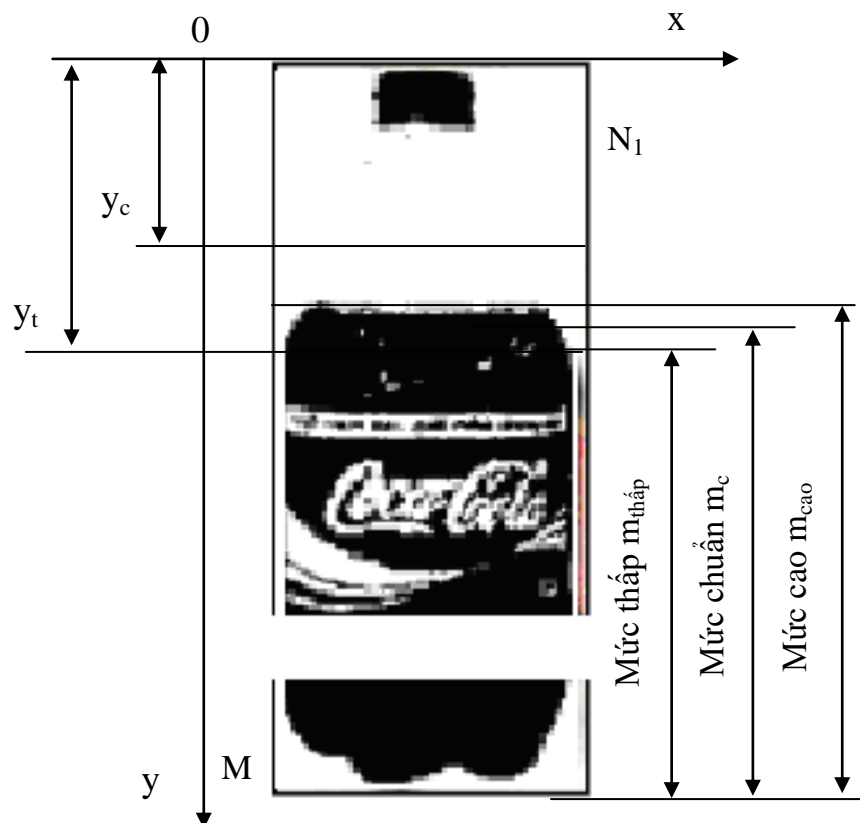
**Hình 2. 9. Thuật toán 2 kiểm tra tồn tại của nắp chai**



## 2.2.2 Thuật toán dùng xử lý ảnh để kiểm tra thể tích của chai

### 2.2.2.1 Phân tích nhiệm vụ

Hình dáng các chai nước khi đã được chiết nước (ngọt) vào có dạng như hình 2.10. Do thời gian bơm nước vào chai không hoàn toàn chính xác nên có thể có một số ít chai bị quá đầy hoặc không được đầy theo như thiết kế. Mặc dù số này là rất ít nhưng không thể đưa chúng vào nhóm sản phẩm đủ quy cách được vì như vậy sẽ làm mất uy tín, thương hiệu sản phẩm. Thuật toán dùng xử lý ảnh để kiểm tra thể tích của một loại chai nước cụ thể dựa trên cơ sở đo chiều cao mức chất lỏng trong chai được thực hiện trên cơ sở phân tích sau:



**Hình 2. 10. Ảnh nhị phân thu được để kiểm tra thể tích của chai**

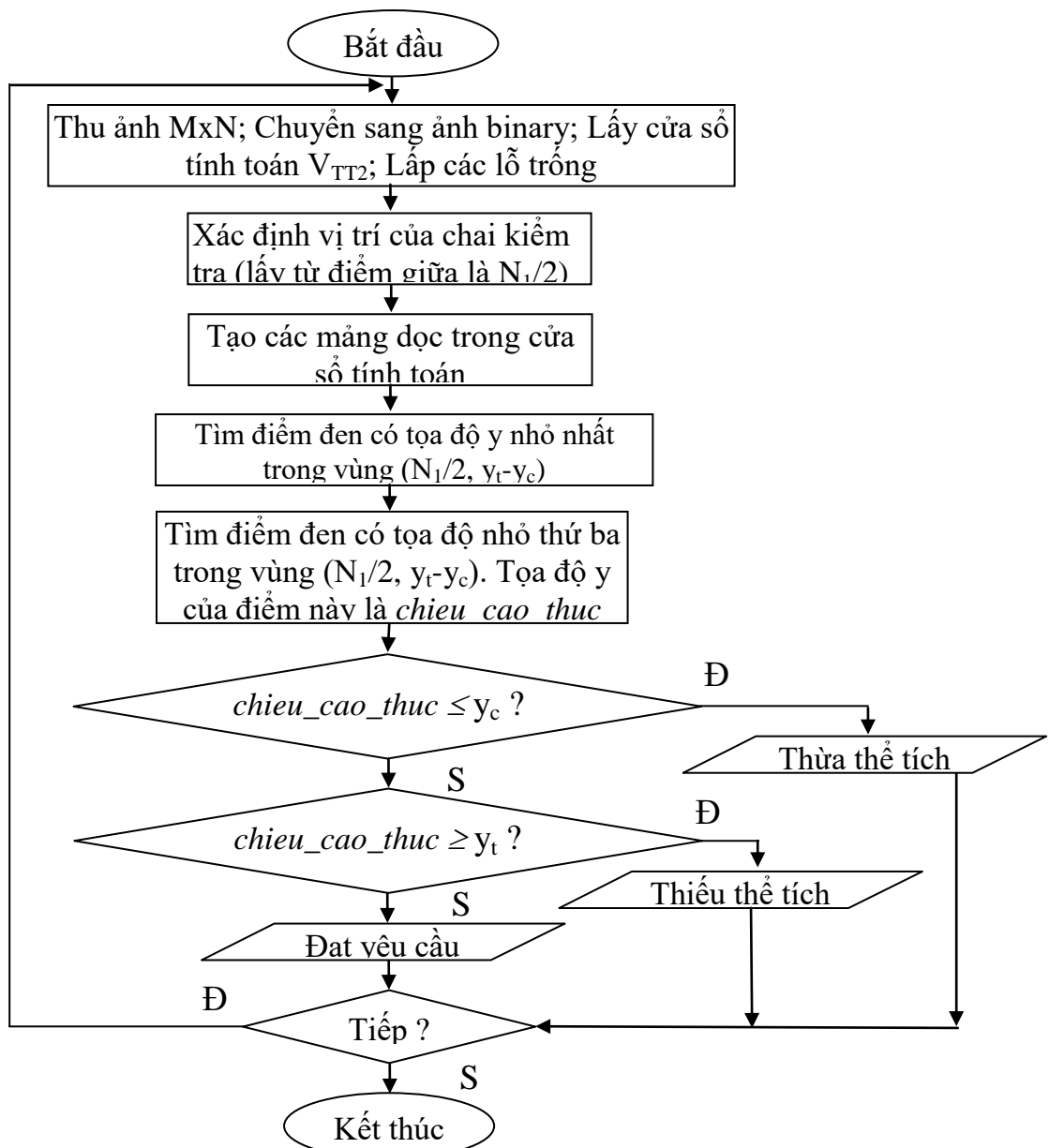
Xem trên hình 2.10 Chai nước coca-cola loại nhỏ với dung tích 390 ml Trên ảnh có độ phân giải 174 x 144 với ống kính 25mm chúng ta chỉnh để sai

lệch này là 2 pixel. Như vậy trên ảnh nếu chiều cao thực tế của một chai nước là  $chieu\_cao\_thuc$  mà chiều cao này thỏa mãn:  $muc\_thap \leq chieucaothuc \leq muc\_cao$  là đạt yêu cầu về thể tích chai. Sai lệch trên ảnh giữa mức chuẩn với các mức cao và mức thấp là 2 pixel.

Vùng ảnh tính toán của chúng ta  $V_{TT2}$  sẽ là vùng có kích thước:

Chiều cao =  $y_t - y_c$  và chiều rộng =  $N_1$ ,  $y_t = m_{thap} + 2$  và  $y_c = m_{cao} - 2$ ;

### 2.2.2.2 Thuật toán thực hiện



**Hình 2. 11. Thuật toán kiểm tra thể tích nước ngọt trong chai**

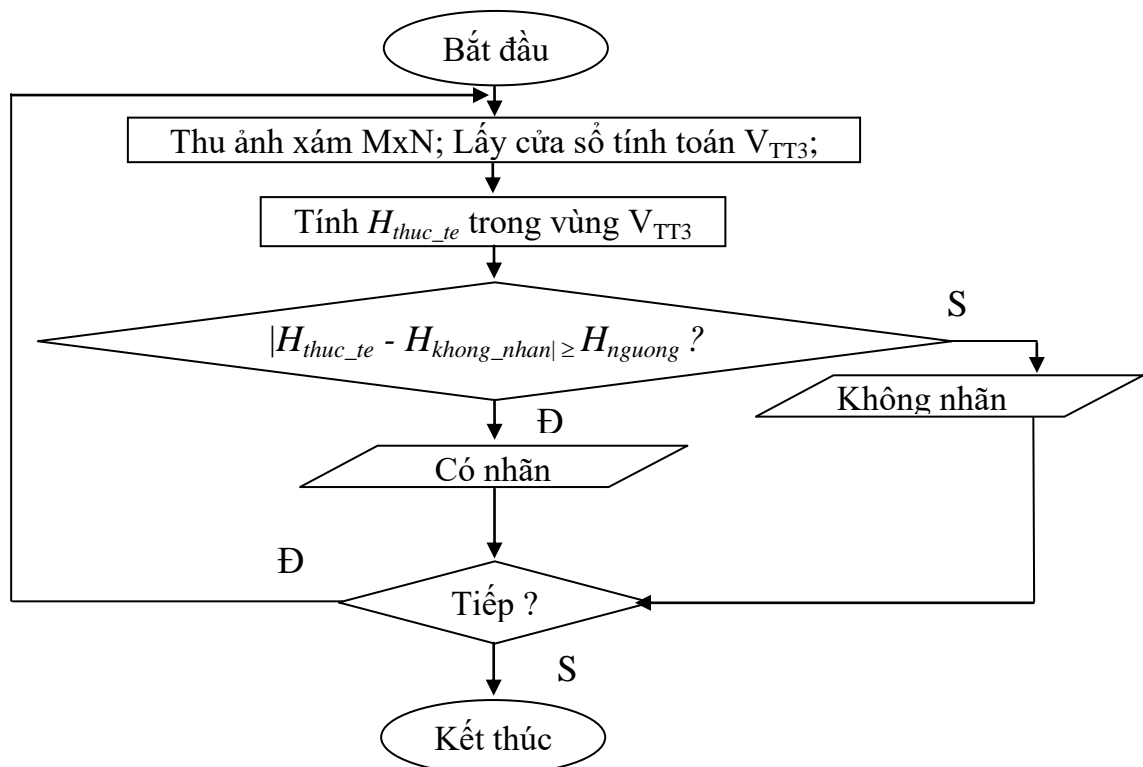
### 2.2.3 Thuật toán dùng xử lý ảnh để kiểm tra nhãn của chai

#### 2.2.3.1 Phân tích nhiệm vụ

Việc xem xét thực hiện nhiệm vụ thứ ba này sẽ thực hiện trên ảnh xám. Như đã nhận xét trong phần 2.1: do các chai trên dây chuyền chuyển động sát nhau, giữa chúng không có khoảng cách nên trên ảnh thu được tổng số điểm đen của vùng tính toán ngoài khu vực dán nhãn là một con số tương đối ổn định.

Ta cũng nhận thấy rằng do có các ký tự, hoa văn trên nhãn nên trên ảnh thực tế khi thu được thì Histogram  $H_{thuc\_te}$  của vùng tính toán  $V_{TT3}$  này sẽ khác với Histogram của chính vùng này khi không có nhãn  $H_{khong\_nhãn}$  ( $H_{khong\_nhãn}$  được tính sẵn từ trước). Chúng ta đặt ra một ngưỡng sai số  $H_{nguong}$  và so sánh nó với giá trị  $|H_{thuc\_te} - H_{khong\_nhãn}|$  để từ đó rút ra kết luận có nhãn hay không.

#### 2.2.3.2 Thuật toán thực hiện

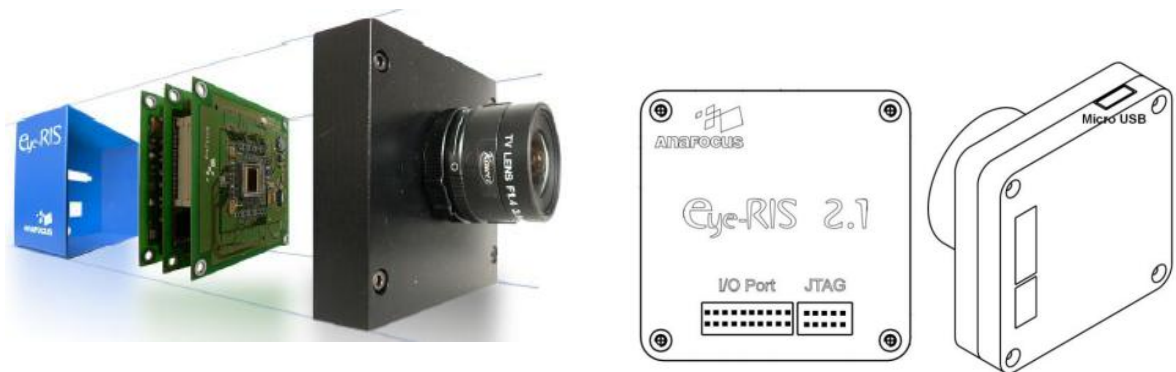


Hình 2. 12. Kiểm tra có nhãn trên vỏ chai hay không

## Chương 3. THỰC NGHIỆM

### 3.1 Thiết bị thu ảnh công nghiệp camera Eye-RIS

#### 3.3.1 Phần cứng



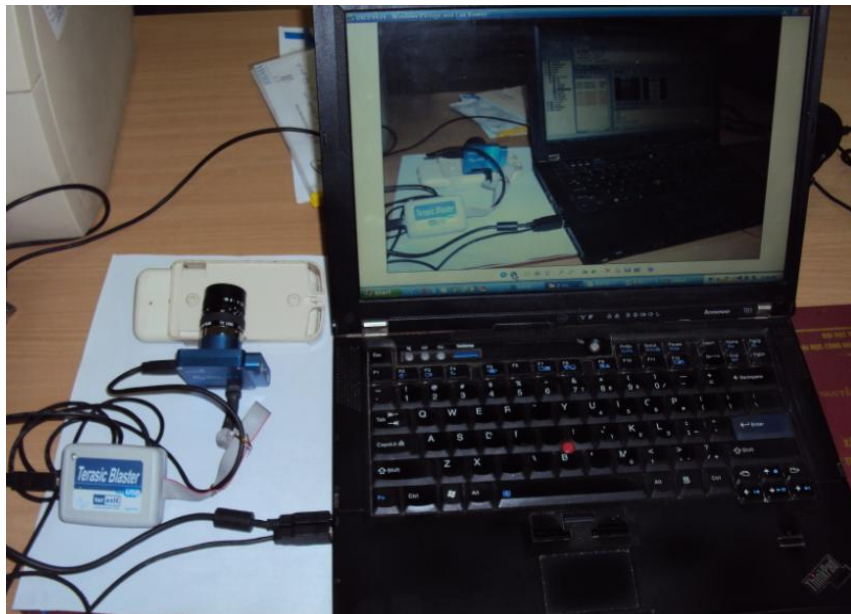
**Hình 3. 1. Camera tốc độ cao Eye- RIS**

Eye-RIS là một camera + máy tính có tốc độ bắt ảnh và xử lý ảnh tốc độ cao đến 10.000fps. Máy tính xử lý ảnh trong Eye-RIS là máy tính tương tự - logic dùng các mảng nơ ron tế bào CNN (Cellular Neural Network). Trong Eye-RIS còn có một máy tính xử lý tuần tự được chế tạo trên công nghệ FPGA nên có thể tái cấu hình, dễ dàng nâng cấp cho các phiên bản mới. Eye-RIS có thể thu và xử lý video hoặc ảnh màu, ảnh xám hoặc ảnh nhị phân theo nguyên lý xử lý song song.

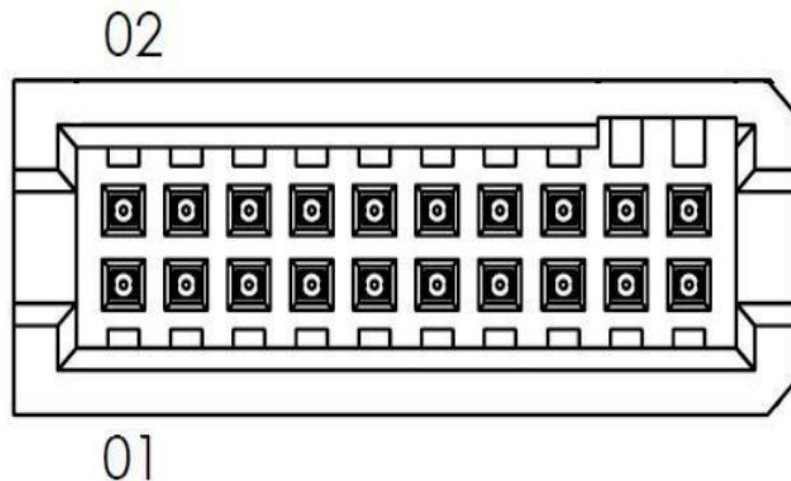
Các đặc điểm kỹ thuật, thông số kỹ thuật:

- + Tốc độ bắt ảnh: 10000fps
- + Độ phân giải: 176 x 144 pixels
- + Optics: C-mount 1/2''
- + Kích thước: 47x47x26.53mm<sup>3</sup>
- + Phạm vi hoạt động:
  - 50dB tích hợp.

- Lên đến 120dB lập trình.
- + Cảm biến đọc: Tích hợp 4ADCs and 4DACs (8-bit at 50MHz) cho ảnh xám.
- + Hệ thống nhúng thông minh cho mỗi điểm ảnh: Lưu trữ đến đến 6 hình ảnh màu xám và 4 hình ảnh nhị phân.
- + Điều khiển và xử lý hình ảnh: Sử dụng chip Altera Nios II 32bit RISC bộ vi xử lý chạy ở 100MHz, 16MB SDRAM cho chương trình và hình ảnh / lưu trữ dữ liệu, EPC flash 8MB.
- + Các cổng vào/ra: USB 2.0, UART...
- + Ống kính, Pentax CCTV Lenses tiêu cự 50mm f1.4. Ống kính có thể thay đổi bằng cách xoay ren để thay.



**Hình 3. 2. Các cổng vào/ ra (I/ O port) trên Eye- RIS V2.1**



**Hình 3. 3.** Đầu nối các chân vào/ra của Eye-RIS V2.1

**Bảng 3. 1.** Chức năng các chân vào/ra

Pin	I/O	Function	Pin	I/O	Function
1		5V	2		5V
3	I/O	GPIO1 / UART CTSn	4	I/O	GPIO0 / UART RxD
5	I/O	GPIO3 / UART RTSn	6	I/O	GPIO2 / UART TxD
7	I/O	GPIO5	8	I/O	GPIO4
9	I/O	GPIO7	10	I/O	GPIO6
11		GND	12		GND
13	I/O	GPIO9	14	I/O	GPIO8
15	I/O	GPIO11	16	I/O	GPIO10
17	I/O	GPIO13 / PWM2	18	I/O	GPIO12 / PWM1
19	I/O	GPIO15 / Falling Edge Interrupt	20	I/O	GPIO14 / Rising Edge Interrupt

### 3.3.2 Phần mềm

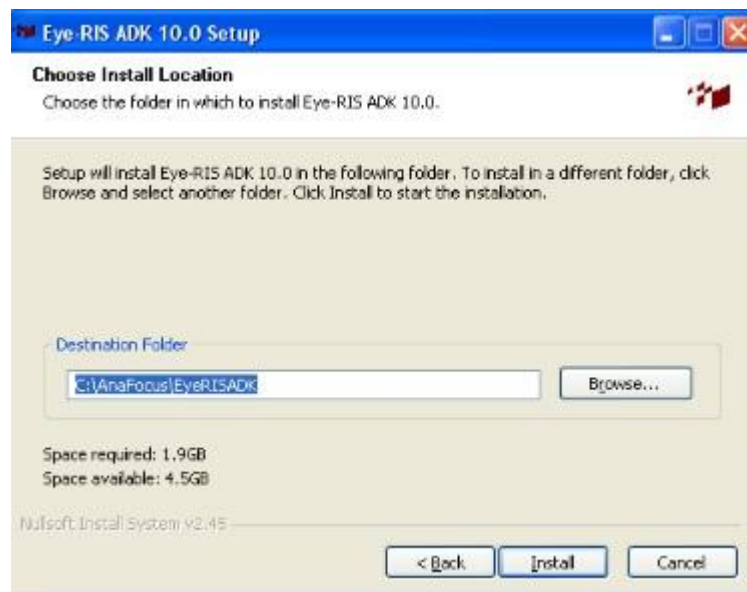
#### 3.3.2.1 Phần mềm điều khiển Eye-RIS ADK 10.2

- ADK Eye-RIS là môi trường phát triển phần mềm được cung cấp với Eye-RIS hệ thống phát triển ứng dụng.

- Các bước cài đặt chương trình:



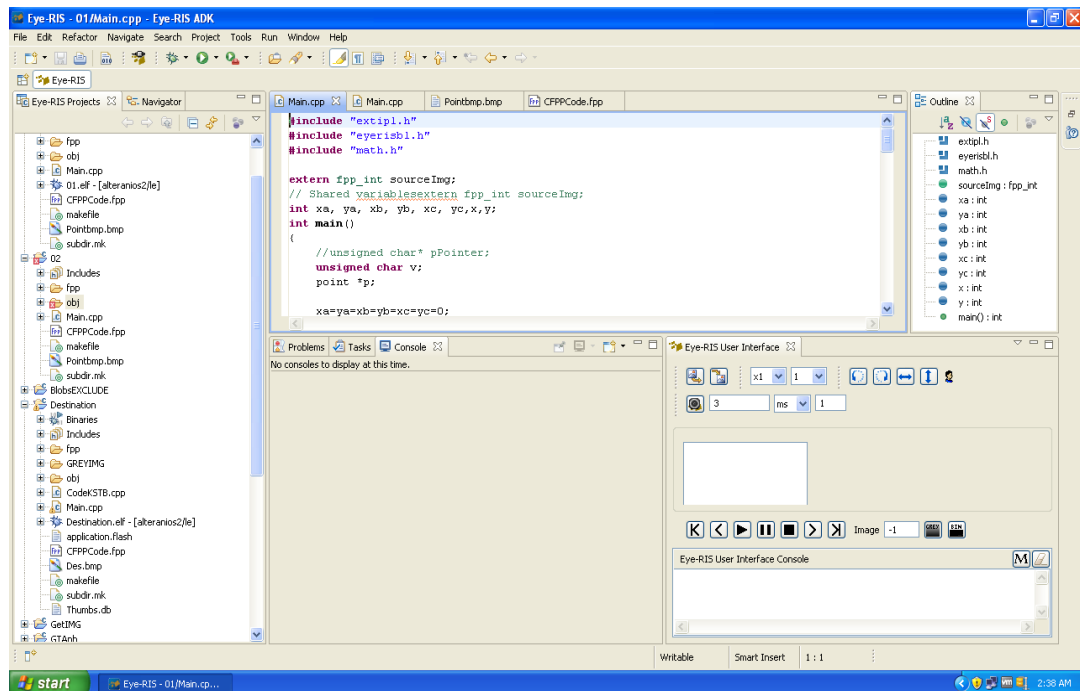
**Hình 3. 4. Cài đặt phần mềm ứng dụng**



**Hình 3. 5. Chọn thư mục cài đặt**



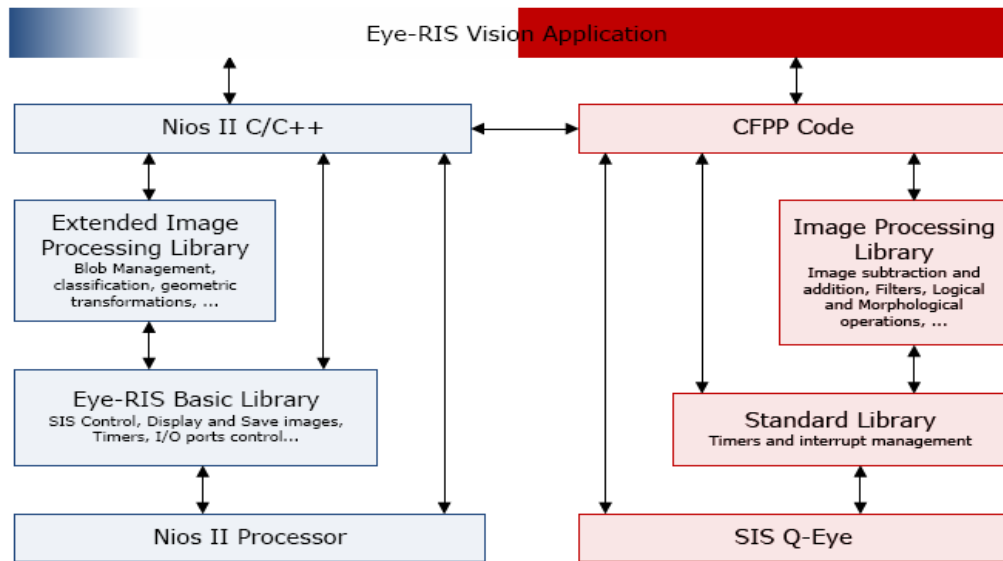
**Hình 3. 6. Hoàn thành quá trình cài đặt**



**Hình 3. 7. Giao diện chính của phần mềm Eye-RIS ADK 10.2**

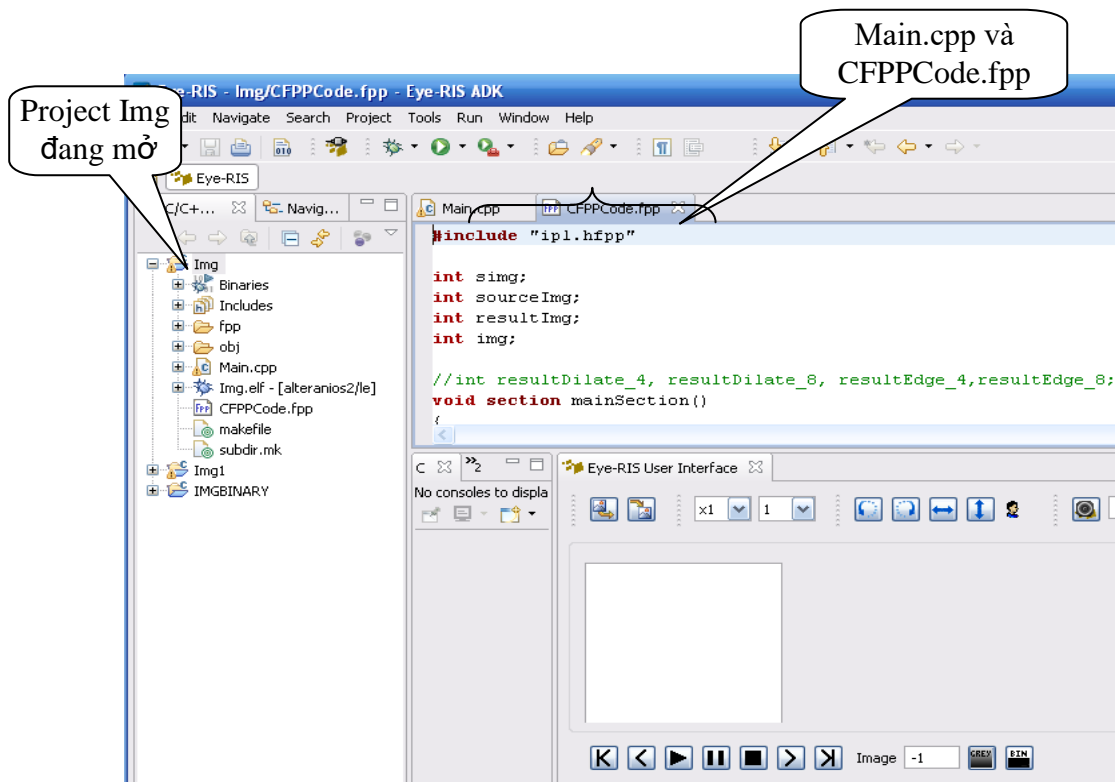
- Lập trình cho Eye-RIS





**Hình 3. 8. Cấu trúc phần mềm lập trình cho Eye-RIS**

Trong Eye-RIS có hai bộ xử lý. Một bộ xử lý trên các mảng CNN sử dụng FPPCode (một loại mã do Anafocus phát triển có cú pháp giống C, C++) và một bộ xử lý nối tiếp. Tương ứng cấu trúc phần mềm lập trình cho Eye-RIS gồm hai phần có liên hệ với nhau khi làm việc. Một Project chạy trên Eye - RIS xây dựng bằng Eye-RIS ADK có nhiều file cấu thành; trong đó hai file quan trọng là Main.cpp và CFPPCode.fpp.



**Hình 3. 9. Chương trình Eye-RIS ADK có hai file Main.cpp và CFPPCode.fpp + Main.cpp**

File Main.cpp chứa hàm main(). Hàm này gọi có các câu lệnh thực hiện tác vụ chính và gọi hàm con và các hàm xử lý song song trên CNN trong file CFPPCode.fpp

```
#include "eyerisbl.h"
```

```
# include .....
```

```
extern fpp_int simg;
```

```
extern fpp_int img;
```

```
int main()
```

```
{
```

```
    // Khai báo các biến .....
```

```
    // Init
```

```
    ...
```

```
while (1)
```

```

        {
// Optical acquisition
...
// CFPP_preProcessing
...
// postProcessing
...
// decisionMaking
...
// sendInformation
...
}
// Final actions
...
}

```

**+ *CFPPCode.fpp***

File *CFPPCode.fpp* chứa các hàm xử lý song song trên CNN. Các hàm này có thể là hàm thu/xử lý song song.

Ví dụ:

```

#include "ipl.hfpp"
    int simg;
    int sourceImg;
    int resultImg;
    int img;
    int resultDilate_4, resultDilate_8, resultEdge_4,resultEdge_8;
void section mainSection()
{

```

```

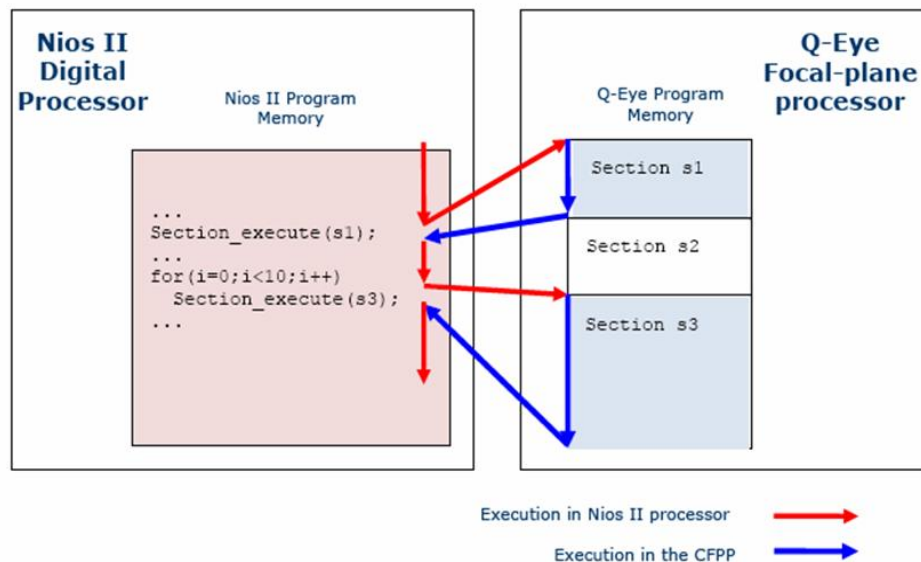
//-----//
// Place your CFPP code here //
//-----//
}
void section Getimg()
{
Sense_acquire(LAM_0, 55, 1);
Move_downloadImage(LAM_0, img, GREY);
Thresh_global(LAM_0, LDM_0, 120);
Logic_not(LDM_0,LDM_1);
Logic_not(LDM_1,LDM_2);
Move_downloadImage(LDM_2, simg, BINARY);
}
void section downloadBlobsImg(int blobsEx, int &resultImg)
{
    Move_loadImage(blobsEx, LDM_0 , BINARY);
    Move_downloadImage(LDM_0, resultImg , BINARY);
}
void section Dilate()
{
// Load the image stored in the page sourceImg of the
// Eye-RIS Image Memory into the LDM_0
Move_loadImage(simg, LDM_0, BINARY); // Tu sigm vao LDM
// Dilate operations
Morph_dilate(LDM_0, LDM_1, CONNECT_4, 1, BINARY_BLACK);
// Results are downloaded to the pages number resultDilate_4
// and resultDilate_8 in the Eye-RIS Image Memory

```

```

Move_downloadImage(LDM_1, resultDilate_4, BINARY);
//Move_downloadImage(LDM_2, resultDilate_8, BINARY);
Move_loadImage(LDM_1, simg, BINARY);
// Get the external borders of the image
//Logic_notAnd (LDM_0, LDM_1, LDM_1);
// Results are downloaded to the pages number resultEdge_4
// and resultEdge_8 in the Eye-RIS Image Memory
Move_downloadImage(LDM_1, resultEdge_4, BINARY);
}

```



**Hình 3. 10. Thực hiện mã trong Eye-RIS**

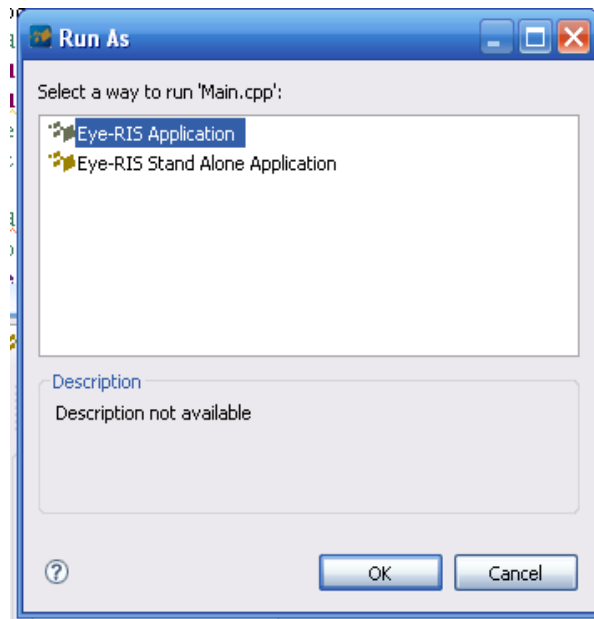
+ Import và Export Project

Dùng để thêm vào hoặc bỏ đi các Project. Ví dụ trên hình 1.14 đang có 3 Project được Import từ " C:\AnaFocus\EyeRISADK\eclipse\workspace". là thư mục chứa các Project ngầm định. Để thực hiện Import/Export Project Trong giao diện EyeRIS ADK vào File\sao đó Import hoặc Export.

### - *Chạy chương trình*

Một chương trình EyeRIS ADK sau khi dịch có thể:

Chạy có giao tiếp với máy tính để hiển thị thông tin, nhận điều khiển từ chương trình, hoặc có thể: Chạy độc lập (standalone)



**Hình 3. 11. Chọn kiểu chạy chương trình**

Tùy theo yêu cầu chúng ta chọn loại nào thì Eye-RIS ADK sẽ nạp mã tương ứng vào Eye-RIS để thực thi. Trong khi phát triển chương trình chúng ta chọn "Eye-RIS Application". Khi đã hoàn chỉnh và muốn chạy độc lập với máy tính chúng ta chọn: " Eye-RIS Stand Alone Application ".

#### **3.3.2.2 Một số hàm thông dụng của Eye-RIS ADK 10.2**

Các hàm của Eye-RIS ADK được chứa trong các thư viện:

\* **Eye - RIS Thư viện cơ bản (EBL)**: gồm một tập hợp các mô-đun được viết bằng C và thực hiện trong bộ vi xử lý Nios II cung cấp các chức năng sau :

Kiểm soát việc thực hiện các mã CFPP, I / O quản lý, gửi và / hoặc nhận hình ảnh đến / từ máy tính hoặc là hiện hoặc lưu trữ tin nhắn đĩa, lỗi in ấn hoặc thông tin vào một giao diện điều khiển, quản lý Timer.

\* **Standard Library (STDL):** Đây là một thư viện nhỏ có chứa một tập hợp các chức năng để kiểm soát các chức năng xử lý của SIS Q- Eye về thời gian, về ngắt và một số thanh ghi, trạng thái.

\* **Các tham khảo khác:**

IPL Eye- RIS v2.1: Các xử lý thu ảnh, nạp ảnh, thực hiện các xử lý ảnh cơ bản trên CNN như phân ngưỡng, tìm biên, lọc, làm mảnh, các phép hình thái học, ...

EIPL Eye- RIS v2.1: Các hàm trích rút đặc điểm ảnh, cấu trúc ảnh,...

Dưới đây chỉ là 1 số hàm hay sử dụng.

**- Thu ảnh: Hàm thu ảnh: *Sense\_acquire***

+ Cú pháp:

```
void Sense_acquire(LAM destination, time exposureTime, int gain);
```

```
void Sense_acquire(LAM destination, time exposureTime, int gain,  
LDM maskImage);
```

+ Diễn tả:

Hàm này thực hiện thu một ảnh xám thấy được trong không gian quang học của Eye-RIS. Có thể đặt tham số thời gian tích hợp, thời gian lộ sáng, độ lợi tích hợp ảnh. Hàm này cũng có thể thu được ảnh nhị phân (hàm thứ hai).

+ Tham số:

Đích (destination): vùng nhớ LAM trong đó ảnh thu được sẽ được lưu trữ.

Thời gian lộ sáng (exposureTime): thời gian tích hợp ảnh tính bằng mili giây.

gain: Độ lợi tích hợp ảnh.

maskImage: vùng nhớ LDM chứa ảnh nhị phân.

+ Giá trị trả về: không.

**- Hàm di chuyển ảnh: *Move\_loadImage***

+ Cú pháp:

```
void Move_loadImage(int source, LAM destination, int loadingMode);
```

```
void Move_loadImage(int source, LAM destination, int loadingMode,  
lrm maskImage);
```

```
void Move_loadImage(int source, LAM destination, int loadingMode,  
int firstRow, int lastRow);
```

```
void Move_loadImage(int source, LDM destination, int loadingMode);
```

```
void Move_loadImage(int source, LDM destination, int loadingMode,  
int firstRow, int lastRow);
```

+ Diễn tả:

**- Di chuyển ảnh giữa các vùng nhớ trong *Eye-RIS***

+ Tham số:

- source: Vùng nhớ Page trong đó ảnh nguồn được lưu giữ.
- destination: các vùng nhớ LAM hoặc LDM ảnh sẽ di chuyển đến.
- loadingMode: kiểu nạp ảnh xám hoặc ảnh nhị phân. Trong trường hợp thứ nhất với ảnh ảnh 1 byte là 1 điểm ảnh, trường hợp 2 1 bit là một điểm ảnh. Hằng số định nghĩa trước GREY và BINARY có thể được sử dụng cho bộ nhớ mở rộng và INTERNAL\_GREY, INTERNAL\_BINARY cho bộ nhớ onchip.

- maskImage: vùng nhớ LDM chứa ảnh nhị phân.

- firstRow: dòng đầu tiên của ảnh nguồn được nạp vào.

- lastRow: dòng cuối cùng của ảnh nguồn được nạp.

+ Giá trị trả về: không

**- Biến đổi thành ảnh nhị phân**

+ Cú pháp:

```
void Thresh_global (LAM source, LDM destination, int th);
```



void Thresh\_global (LDM source, LDM destinationWhite, LDM destinationBlack, int th);

+ Diễn tả:

Phân ngưỡng để biến ảnh xám trong vùng LAM thành ảnh nhị phân trong vùng LDM

+ Tham số:

- source: vùng ảnh xám nguồn
- destination: vùng ảnh nhị phân đích.
- destinationWhite: vùng nhớ LDM trong đó giá trị lớn hơn hoặc bằng ngưỡng th sẽ thành điểm trắng.
- destinationBlack: vùng nhớ LDM trong đó giá trị nhỏ hơn ngưỡng th sẽ thành điểm đen.
- th: giá trị ngưỡng.

+ Giá trị trả về: không

+ Thực hiện các phép logic, hình thái học

Logic: gồm các hàm

void Logic\_not (LDM source, LDM destination);

void Logic\_and (LDM source1, LDM source2, LDM destination);

void Logic\_or (LDM source1, LDM source2, LDM destination);

void Logic\_xor (LDM source1, LDM source2, LDM destination);

void Logic\_nand (LDM source1, LDM source2, LDM destination);

void Logic\_nor (LDM source1, LDM source2, LDM destination);

void Logic\_nxor (LDM source1, LDM source2, LDM destination);

void Logic\_andNot (LDM source1, LDM source2, LDM destination);

void Logic\_notAnd (LDM source1, LDM source2, LDM destination);

void Logic\_orNot (LDM source1, LDM source2, LDM destination);

void Logic\_notOr (LDM source1, LDM source2, LDM destination);

Thực hiện phép logic theo bit giữa vùng nhớ source và lưu kết quả vào vùng destination (logic NOT) hoặc thực hiện các phép AND, OR, XOR, NAND, NOR, NXOR, ANDNOT, NOTAND, ORNOT, NOTOR theo bit của hai ảnh trong hai vùng nhớ source1 và source2 sau đó lưu kết quả vào vùng nhớ destination.

Và 1 số hàm hình thái học:

Morph_erode:	Thực hiện phép giảm số điểm đen
Morph_dilate	Thực hiện phép tăng điểm đen
Morph_open	Phối hợp Erosion và Dialtion $A \bullet B = (A \oplus B) \ominus B$
Morph_close	Phối hợp Erosion và Dialtion $A \circ B = (A \ominus B) \oplus B$
Morph_thin	Làm mảnh ảnh
Morph_thicken	Làm dày ảnh
Morph_centroid	Định tâm ảnh
Morph_skeleton	Tìm xương
Morph_removeSinglePoints	Di chuyển các điểm đơn lẻ

..

Một số hàm khác

Sử dụng UART:

Ví dụ dưới đây đọc ký tự và gửi tin nhắn đến các thiết bị UART bằng cách sử dụng thư viện chuẩn C. Mã này lần đọc ký tự từ UART cho đến khi một 'v' nhận được. Nếu ký tự nhận được là 't', một thông điệp truyền thông này nó được ghi vào UART. Khi 'v' nhận được, vòng lặp kết thúc và xuất hiện một thông báo nói rằng UART đang được đóng.

Nios C code

```
#include "eyerisbl.h"
#include <stdio.h>
#include <stdlib.h>
```

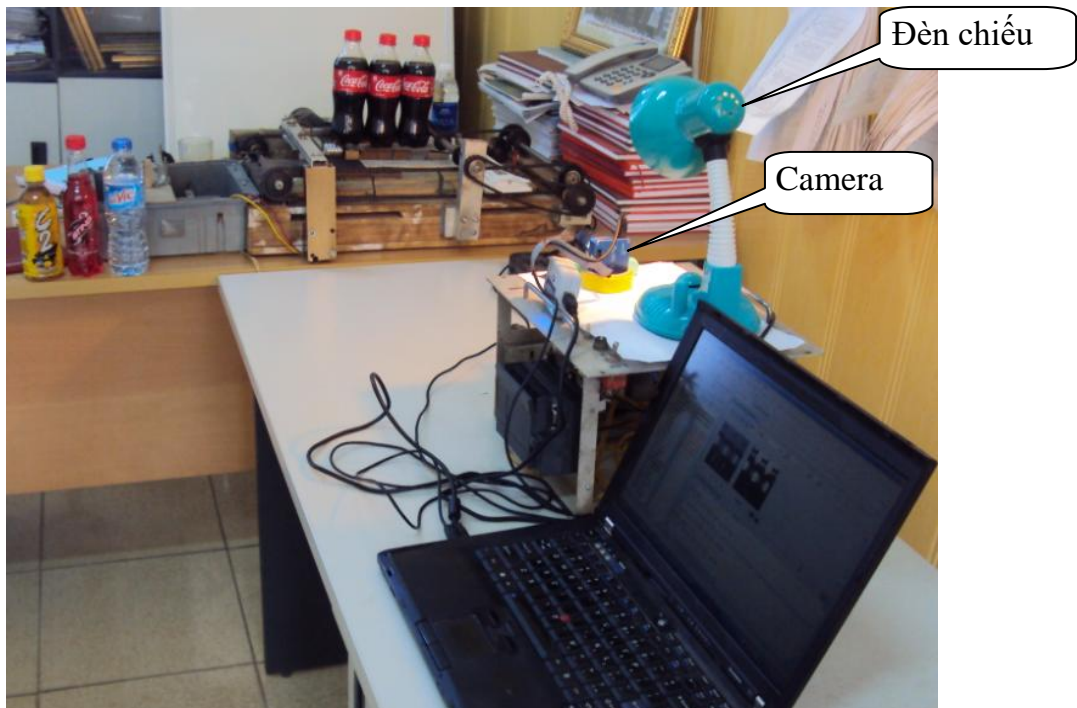
```
int main()
{
    char* msg = "Detected the character 't'.\n");
    FILE* fp;
    char prompt = 0;
    // Enable the UART output to the connector
    UART_enable();
    // Set a new baud rate
    UART_setBaudRate(57600);
    // Open the UART for reading and writing
    fp = fopen(UART, "r+");
    if (fp)
    {
        // Loop until we receive a 'v'
        while (prompt != 'v')
        {
            // Get a character from the UART
            prompt = getc(fp);
            // Print a message if character is 't'
            if (prompt == 't')
            {
                fwrite(msg, strlen(msg), 1, fp);
            }
        }
        fprintf(fp, "Closing the UART.\n"); fclose(fp);
    }
    return 0;}

```

## 3.2 Mô tả dây chuyền nước đóng chai

### 3.2.1 Mô tả thực nghiệm dây chuyền nước đóng chai

Mô hình thực nghiệm có thông số tốc độ như trên hệ thống sản xuất thực và các đối tượng cần kiểm tra thực (các chai nước Coca-cola loại nhỏ có dung tích 390ml). Các chai nước được đặt sát nhau trên mô hình băng tải do hai xích kéo chuyển động với tốc độ 5m/s. Camera Eye-RIS dùng ống kính 25mm đặt cách băng tải 1000mm. Tốc độ bắt ảnh: 6000 ảnh/giây. Ánh sáng chiếu bổ sung vào các chai nước dùng loại đèn 220V/60W đặt tại khoảng cách đến băng tải 1200mm.



**Hình 3. 12.** Thực nghiệm mô hình dây chuyền đóng chai nước



**Hình 3. 13. Chai coca cola trên băng tải**

### 3.2.2 . Bố trí camera, chiếu sáng

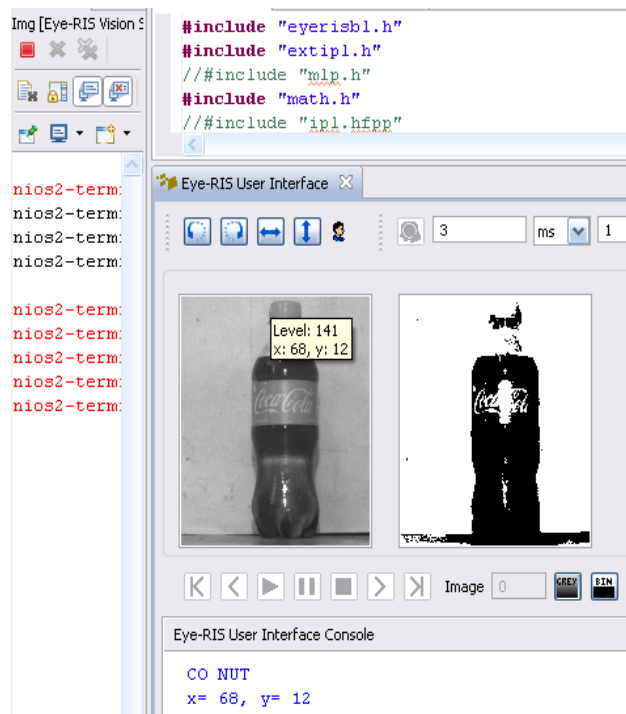
Hệ thống camera đặt cố định chính diện với chai coca-cola chạy trên băng tải đảm bảo quan sát được nút chai, mức nước và nhãn của chai. Khung nhìn của camera bao quát trong tầm nhìn các chai trên nền trắng. Quá trình thu và xử lý ảnh được thực hiện liên tục. Ánh sáng chiếu bổ sung vào các chai nước dùng loại đèn 220V/60W đặt tại khoảng cách đến băng tải 1200mm.



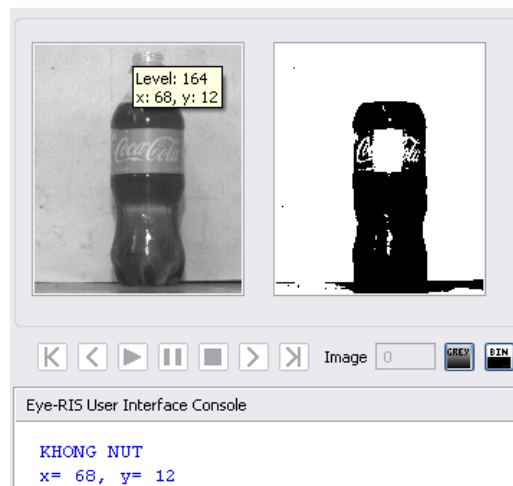
**Hình 3. 14. Chiếu sáng cho đối tượng**

### 3.3 Các kết quả thực nghiệm

Việc tính toán kiểm tra được thực hiện với các trường hợp chai xuất hiện đơn lẻ trong ảnh (các hình 3.4, 3.5, 3.6) và trường hợp các chai xuất hiện liên tục trên băng tải (hình 3.7).

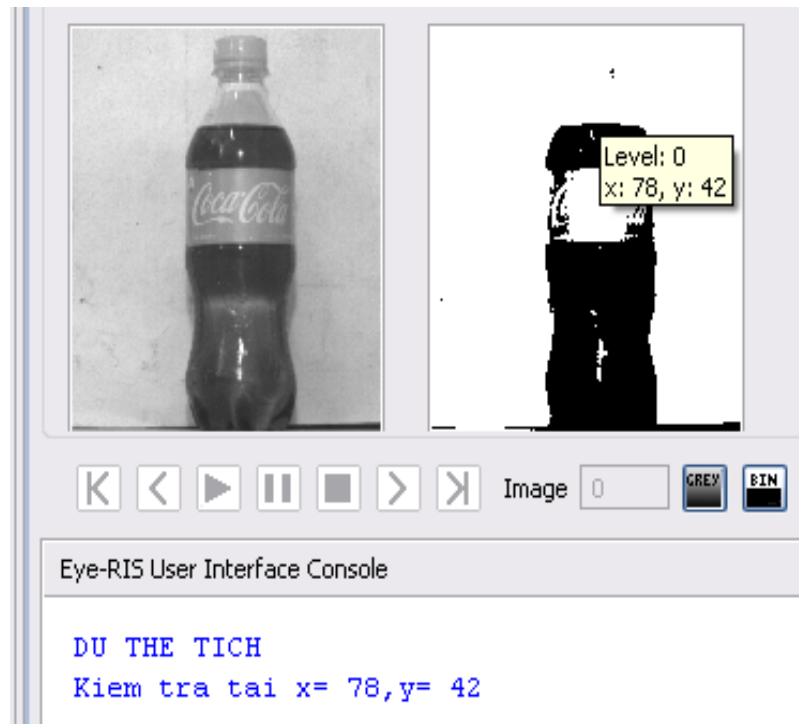


a)

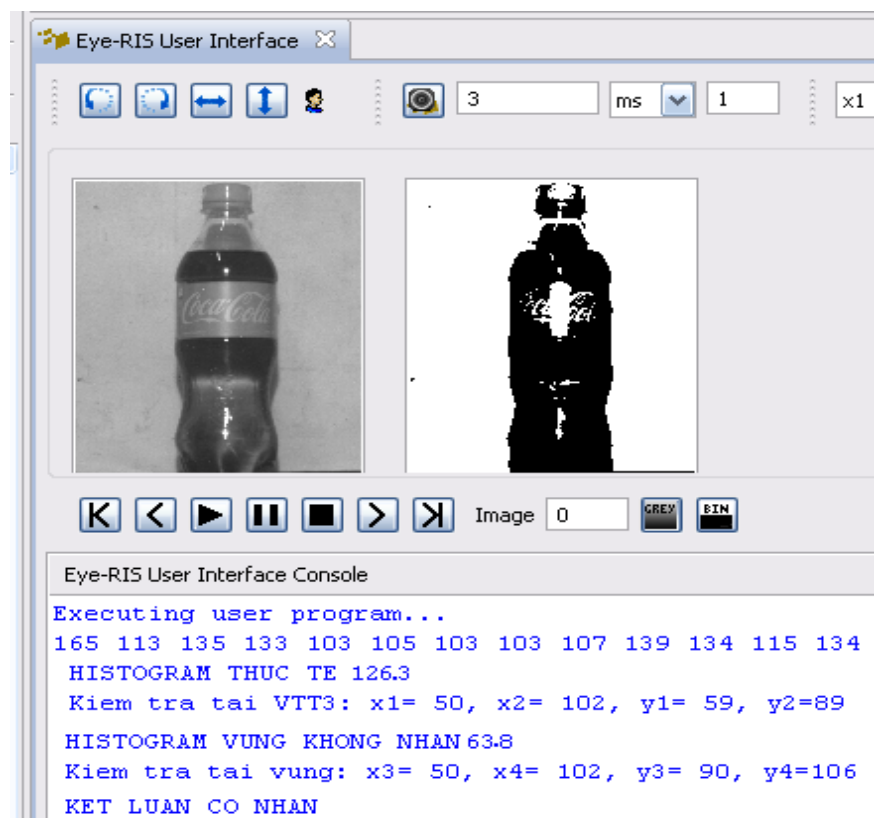


b)

**Hình 3. 15. Ảnh chương trình kiểm tra thể tích: a) chai có nút và b) chai không có nút**



**Hình 3. 16. Kết quả chương trình kiểm tra nhãn**



**Hình 3. 17. Tính Histogram trung bình trong vùng tính toán  $V_{TT3}$**

\* Một số số liệu thực hiện trong thực nghiệm:

- Kiểm tra thể tích nước ngọt: Vị trí chuẩn tại  $y = 42$ .

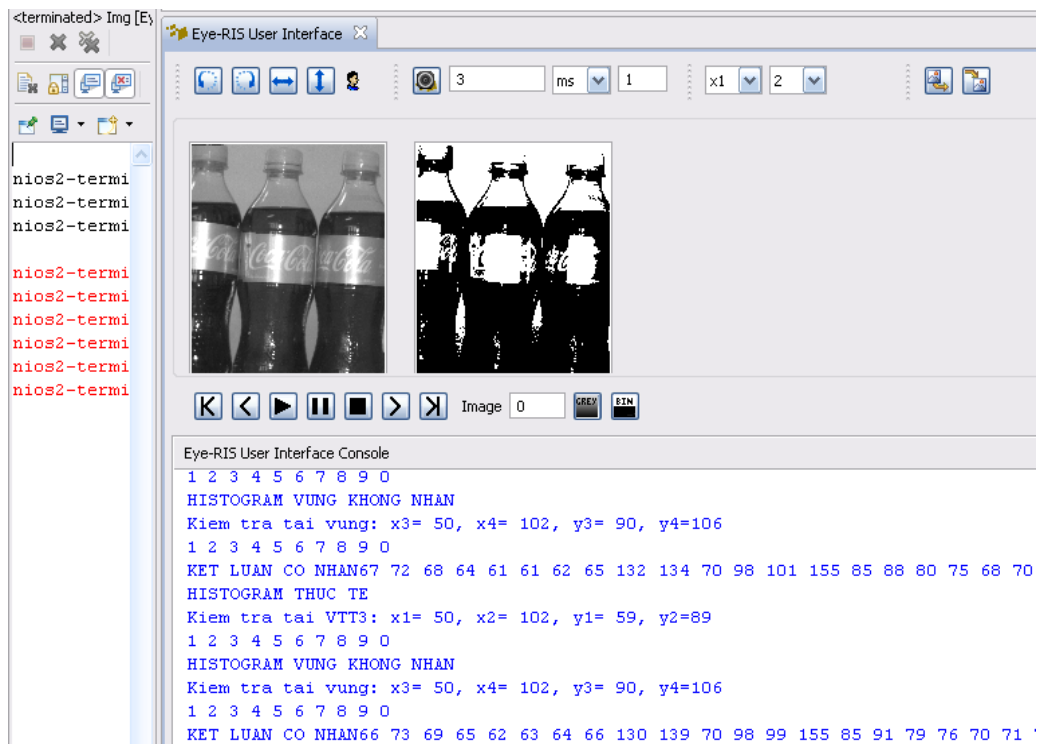
Có nghĩa là nếu mức coca-cola ở  $40 \leq y_T \leq 44$  là đạt yêu cầu ngoài ra là thừa hoặc thiếu thể tích.

**Bảng 3. 2. Kiểm tra thể tích**

TT	Đối tượng	Vị trí thực tế (pixel) $y_T =$	Vị trí thừa thể tích (pixel)	Vị trí thiếu thể tích (pixel)	Kết luận
1.	1	42	40	44	Đạt
2.	2	41	40	44	Đạt
3.	3	45	40	44	thiếu thể tích
4.	...	...	...	...	...

Kiểm tra nhãn

Vùng tính toán kiểm tra nhãn  $V_{TT3}$  có tọa độ:  $x_1 = 50$ ;  $x_2 = 102$ ;  $y_1 = 59$ ;  $y_2 = 89$ ;



**Hình 3. 18. Kiểm tra liên tục trên dây chuyền**



**Bảng 3. 3. Tính Histogram và so sánh giá trị trung bình khi không có nhãn**

<b>T</b>	<b>Đối tượng</b>	<b>Histogram trung bình thực tế tại vùng tính toán</b>	<b>Histogram trung bình khi không nhãn</b>	<b>Kết luận</b>
1.	1	126.3	63.8	Có nhãn
2.	2	126.5	63.2	Có nhãn
3.	3	125.7	63.1	Có nhãn
4.	4	65.4	63.2	không nhãn
5.	...	...	...	...

### 3.4 Nhận xét, đánh giá

Việc ứng dụng công nghệ thông tin đặc biệt là xử lý ảnh vào trong sản xuất công nghiệp để giải quyết những bài toán kiểm tra sản phẩm mà những sản phẩm này khó có thể kiểm tra được bằng các sensor truyền thống đã tạo ra nhiều thuận lợi cho việc thực hiện và cả khâu quản lý sản phẩm. Nếu sử dụng nhân tố con người để kiểm tra các sản phẩm bị lỗi thì sẽ tốn rất nhiều công sức, tiền của và khó có thể kiểm tra hết được sản phẩm lỗi trước khi đưa ra thị trường. Mà hậu quả của việc đưa các sản phẩm lỗi ra thị trường là rất lớn: làm mất uy tín của công ty, nhà máy, doanh số bán hàng sẽ giảm. Việc ứng dụng xử lý ảnh trong công nghiệp còn được ứng dụng trong nhiều mô hình sản xuất khác nữa nếu chúng ta biết cách vận dụng, khai thác các thế mạnh của loại giải pháp công nghệ thông tin này.

## KẾT LUẬN

Xử lý ảnh là một lĩnh vực quan trọng trong công nghệ thông tin. Việc ứng dụng công nghệ thông tin, ứng dụng xử lý ảnh vào trong quá trình sản xuất công nghiệp là vấn đề mới mẻ nhưng rất cần thiết và quan trọng. Chính vì vậy, Luận văn đã sử dụng các camera + máy tính thu ảnh tốc độ cao để có thể dễ dàng xây dựng các chương trình kiểm tra sản phẩm và điều khiển hoạt động một số khâu trong hệ thống sản xuất chai nước ngọt coca-cola như khâu kiểm tra lỗi sản phẩm: Sản phẩm đã đóng nắp hay chưa? Gán nhãn hay chưa và đã đủ thể tích hay chưa?. Các dữ liệu có thể được lưu vào cơ sở dữ liệu của sản phẩm trên máy tính công nghiệp có truyền tin với các camera này. Khả năng này tạo ra việc quản lý chặt chẽ sản phẩm ngay từ trong quá trình sản xuất. Đây cũng là một mô hình phổ biến trong các hệ thống sản xuất tự động hiện nay. Một số công việc đã thực hiện để hoàn thành luận văn:

- + Hệ thống hóa kiến thức về xử lý ảnh số.
- + Tìm hiểu và sử dụng được camera tốc độ cao Eye-RIS.
- + Ứng dụng thu ảnh và kiểm tra online một số đặc điểm của chai nước coca-cola trên mô hình dây chuyền sản xuất.
- + Viết bài báo khoa học về vấn đề nghiên cứu.

Việc bám sát các nhu cầu của các lĩnh vực trong cuộc sống nói chung và trong thực tế sản xuất công nghiệp nói riêng sẽ mở ra việc phát huy tiềm năng của ứng dụng công nghệ thông tin, ứng dụng xử lý ảnh một cách rộng rãi và hiệu quả.

Do thời gian có hạn và khả năng còn nhiều hạn chế nên luận văn khó tránh khỏi những sai sót, rất mong được sự góp ý của thầy cô và bạn bè để luận văn được hoàn thiện hơn.

## TÀI LIỆU THAM KHẢO

### Tài liệu tiếng Việt

- [1]. Phạm Đức Long, Phạm Thượng Cát, “*Xử lý ảnh trong công nghiệp, nhu cầu, thách thức và giải pháp*”, Tạp chí Khoa học và Công nghệ Đại học Thái Nguyên Đại học Thái Nguyên, tập 2, số 45, tr. 121-127, 2008.
- [2]. Đỗ Năng Toàn, Phạm Việt Bình, *Giáo trình xử lý ảnh*, Khoa công nghệ thông tin – Đại học Thái Nguyên, 2007.
- [3]. Lương Mạnh Bá, Nguyễn Thanh Thủy, *Nhập môn xử lý ảnh số*, NXB Khoa học và kỹ thuật, 2002.
- [4]. Nguyễn Kim Sách, *Xử lý ảnh và video số*, NXB Khoa học kỹ thuật, Hà Nội, 1997.

### Tài liệu Tiếng Anh

- [5]. Anil K. Jain, *Fundamental of Digital Image Processing*, Prentice Hall, Engwood cliffs, 1989 .
- [6]. Joannis Pitas, "*Digital Image Processing Algorithms*", Prentice Hall, New York, 1992.
- [7]. Edwin Tjandranegara, Distance Estimation Algorithm for Stereo Pair Images, School of Electrical and Computer Engineering, Purdue University tjandran@ecn.purdue.edu, 2005.
- [8]. Pham Duc Long, Determine the location an object by image processing used for controlling autonomous vehicle, Thai Nguyen University Jurnal of Science and Technology, Volume 116, No 02, pg. 35-39, 2014.