

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN TOÁN ỨNG DỤNG VÀ TIN HỌC
—oOo—



Báo cáo môn học:
ĐỐI SÁNH ẢNH DỰA TRÊN LƯỢC ĐỒ XÁM

Giảng viên hướng dẫn: TS. Vương Mai Phương
Học viên thực hiện: Phùng Anh Hùng–20212497M

Hà Nội, tháng 10 năm 2022

Mục lục

Chương 1: Giới thiệu ảnh số	3
1.1 Ảnh số là gì?	3
1.2 Các đặc trưng của ảnh số	4
Chương 2: Đối sánh ảnh	5
2.1 Thế nào là đối sánh ảnh?	5
2.2 Các phương pháp đối sánh ảnh	5
2.2.1 Harris Detector	6
2.2.2 SIFT Detector	7
2.3 Đối sánh ảnh dựa trên lược đồ xám	9
2.3.1 Cân bằng Histogram	10
2.3.2 Histogram Matching	10
2.3.3 Histogram Intersection (HI)	11
2.4 Gaussian Weighted Histogram Intersection (GWHI) . . .	11
2.5 Colour Ratio Gradients (CRG)	14
2.5.1 Chia ảnh thành các vùng đặc trưng (Segmentation into Texture Regions)	16
2.5.2 Thu thập ảnh (Color Constant Texture Retrieval)	17
2.6 Colour Edge Cooccurrence Histogram (CECH)	17
2.6.1 Cách tính CECH histogram	17
2.6.2 Giảm số màu.	20
2.6.3 So sánh biểu đồ CECH bằng GWHI	24

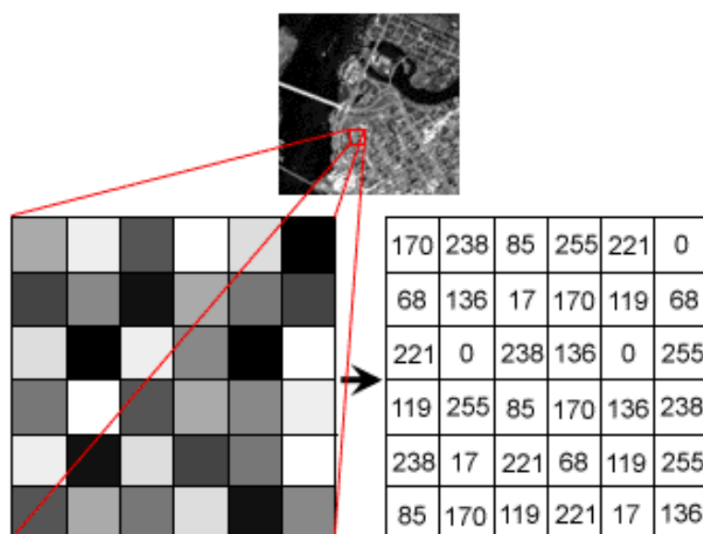
Chương 1

Giới thiệu ảnh số

1.1 Ảnh số là gì?

Ảnh số (Digital Image) là khái niệm chỉ sự biểu diễn trên máy tính của hình ảnh. Trong xử lý ảnh và thị giác máy tính (Computer Vision), các hình ảnh được biểu diễn bởi các điểm ảnh 2 chiều riêng lẻ là pixel. Mỗi pixel có giá trị riêng.

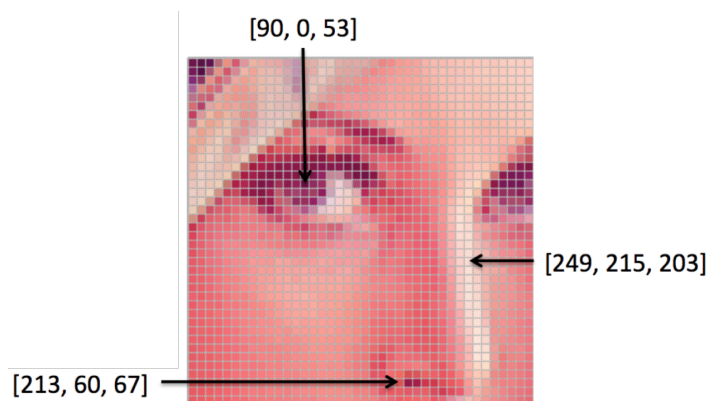
Với ảnh xám (grayscale), mỗi pixel được chia thành cường độ 0 đến 255 với 0 là đen và 255 là trắng. Hàm $f(x,y)$ là giá trị cường độ hình ảnh tại pixel vị trí (x,y) .



Hình 1.1: Ví dụ ảnh xám

Đối với ảnh màu, $f(x, y)$ là một vector với ba giá trị. Ví dụ với ảnh RGB, màu sắc được tạo ra do sự kết hợp ba màu Đỏ, Xanh Lục và Xanh Lam. Do đó, mỗi pixel sẽ được biểu diễn ở dạng vector 1×3 . Ba màu có giá trị nguyên từ 0 đến 255 nên có tổng $256 \times 256 \times 256 = 16.777.216$ màu.

$$f(x, y) = \begin{bmatrix} r(x, y) \\ g(x, y) \\ b(x, y) \end{bmatrix}$$



Hình 1.2: Ví dụ ảnh màu

1.2 Các đặc trưng của ảnh số

Chương 2

Đối sánh ảnh

2.1 Thế nào là đối sánh ảnh?

Đối sánh ảnh (Image Matching) là là một khái niệm quan trọng trong thị giác máy tính và nhận diện vật thể. Những bức ảnh cùng một vật thể có thể được chụp từ các góc độ, ánh sáng, tỷ lệ khác nhau. Do đó, để xác định các ảnh nào là cùng một vật thể, cần có phương pháp để xác định những đặc điểm bất biến của ảnh.

Cách tiếp cận chung của bài toán là tìm các điểm chính (key point) duy nhất hoặc những vị trí đặc biệt nhất trong ảnh. Sau đó, chuẩn hóa nội dung xung quanh key point và tính toán mô tả cục bộ cho nó. Mô tả cục bộ là một vector mô tả key point, từ đó có thể sử dụng để so sánh với các key point ở các ảnh khác.

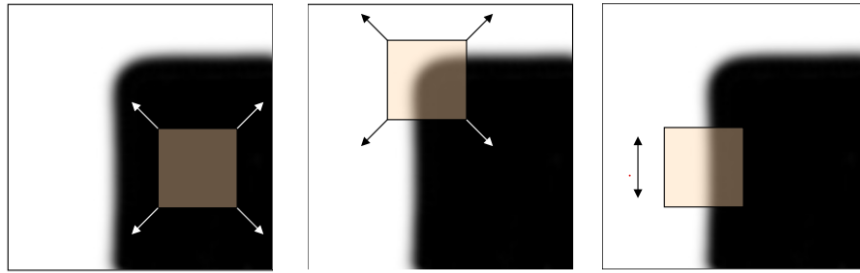
2.2 Các phương pháp đối sánh ảnh

Đến nay, ta đã biết được một số phương pháp đối sánh ảnh như sau:

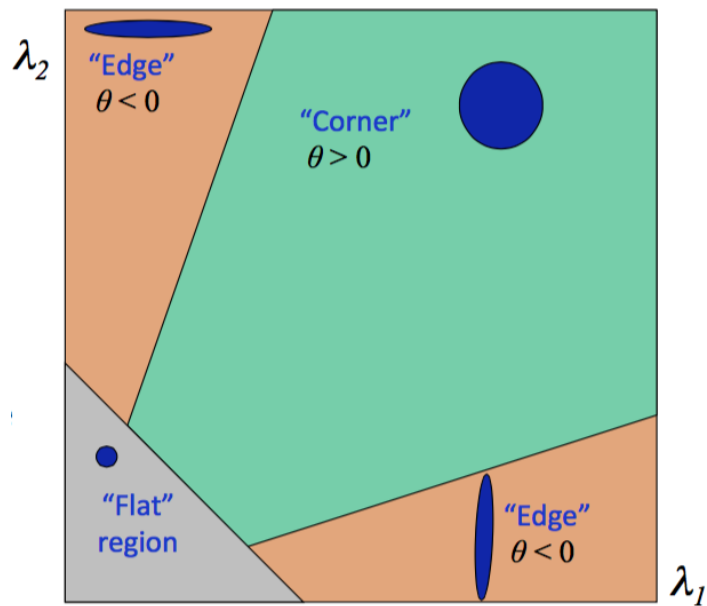
2.2.1 Harris Detector

Các góc (corner) là những key point phổ biến được quan tâm vì đó là nơi có những thay đổi đáng kể ở mọi hướng.

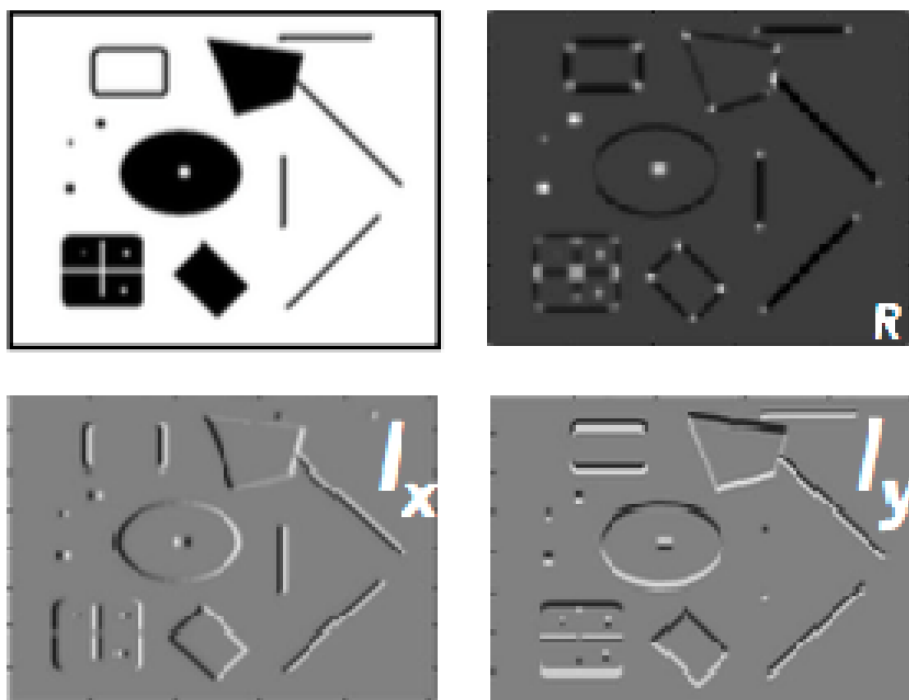
Phương pháp Harris Detector được phát triển để xác định các vùng của ảnh là phần mặt phẳng (Flat), cạnh (Edge) hoặc góc (Corner).



Hình 2.1: Ba vùng mặt phẳng, cạnh và góc



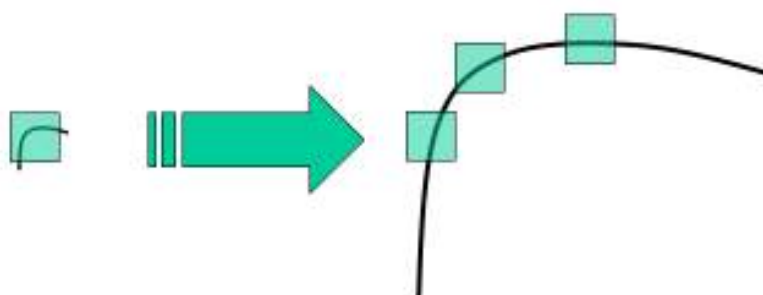
Hình 2.2: Xác định vùng dựa vào λ_1, λ_2



Hình 2.3: Ví dụ tìm keypoint bằng Harris Detector

2.2.2 SIFT Detector

Phương pháp Harris Detector có thể xác định các góc ngay cả khi ảnh bị xoay. Tuy nhiên, nếu ảnh bị thay đổi tỉ lệ (scale), phương pháp có thể không hoạt động tốt.

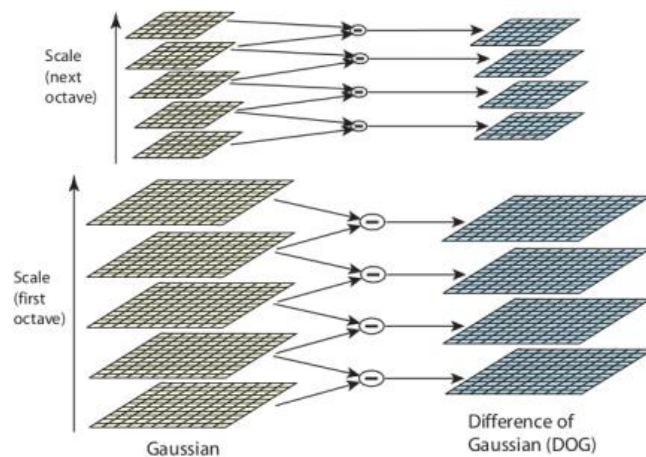


Hình 2.4: Phần góc bị nhận thành cạnh khi thay đổi tỷ lệ ảnh

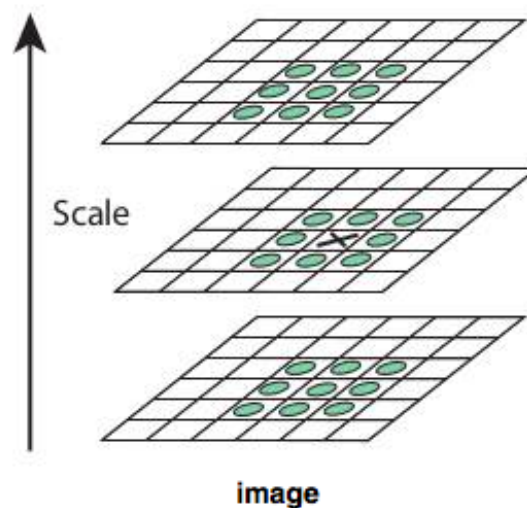
Phương pháp SIFT (Scale-Invariant Feature Transform) tính DoG (Difference of Gaussians) trên từng pixel bằng cách lấy khác biệt Gaus-

sian Blur của hình ảnh với độ lệch chuẩn khác nhau. Với mỗi scale, hình ảnh bị làm mờ bằng Gaussian. Sự khác biệt của các ảnh mờ liền kề được tính là DoG.

Sau khi có DoG, SIFT thực hiện xét từng pixel, so sánh với 8 pixel lân cận, 9 pixel của scale trên và 9 pixel của scale dưới. Nếu pixel có giá trị lớn nhất, nó là key point tiềm năng.



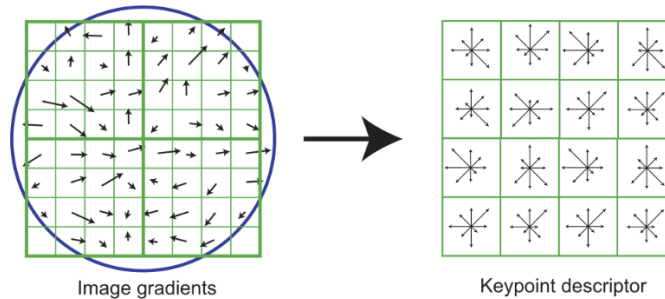
Hình 2.5: Tính DoG của từng tỉ lệ



Hình 2.6: Xét từng pixel trong ảnh

Sau khi tìm được key point, SIFT thực hiện tìm thông tin cục bộ bằng SIFT Descriptor theo các bước sau:

- Lấy gradient ảnh trên mảng 16×16 .
- Xoay các hướng và vị trí Gradient so với hướng của key point.
- Tạo mảng biểu đồ định hướng
- Thêm gradient đã xoay vào biểu đồ với 8 hướng.
- Kết quả là một vector có độ dài 128 đại diện một biểu đồ 4×4 với 8 hướng trên mỗi ô biểu đồ.



Hình 2.7: SIFT Descriptor

2.3 Đối sánh ảnh dựa trên lược đồ xám

Lược đồ xám (Histogram) là biểu đồ tần suất thống kê số lần xuất hiện các mức sáng trong ảnh.

Với một ảnh có n pixel, L mức xám $D = \{0, 1, \dots, L-1\}$. Histogram của ảnh sẽ được xây dựng bởi hàm:

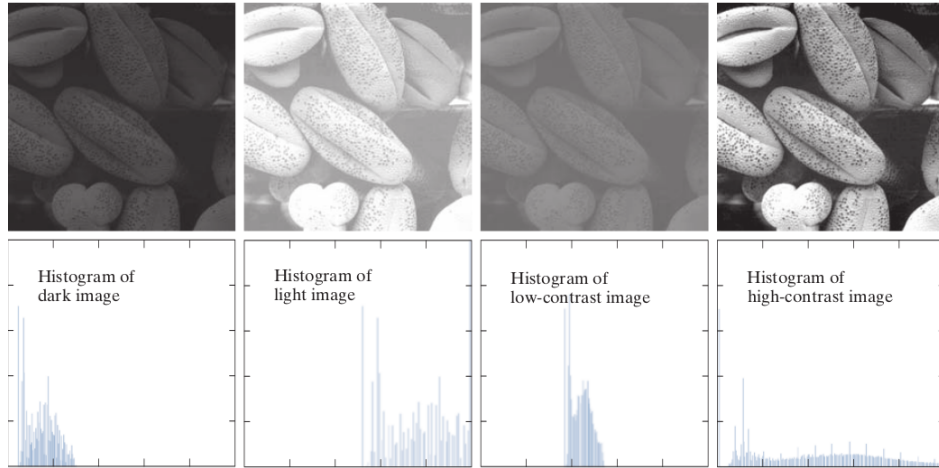
$$f : D \rightarrow N$$

$$f(k) = n_k$$

hoặc dạng chuẩn hóa:

$$f : D \rightarrow [0, 1]$$

$$f(k) = \frac{n_k}{n}$$



Hình 2.8: Ví dụ biểu đồ Histogram

2.3.1 Cân bằng Histogram

Cân bằng Histogram (histogram equalization) là sự điều chỉnh histogram sao cho phân bố giá trị pixel không bị co cụm lại mà được kéo dãn ra.

Xét $r \in [0, L - 1]$, khi đó cân bằng biểu đồ xám là ánh xạ $s = T(r)$ thỏa mãn các điều kiện:

- $0 \leq T(r) \leq 1$
- $T(r)$ là hàm đơn điệu không giảm trong khoảng $0 \leq r \leq L - 1$

Khi đó, hàm $T(r)$ có công thức như sau:

$$s_k = T(r_k) = (L - 1) \sum_{j=0}^k p_r(r_j)$$

trong đó $p_r(r_k) = \frac{n_k}{N}$.

2.3.2 Histogram Matching

Thuật toán Histogram Matching là thuật toán biến đổi ảnh đầu I vào thành một ảnh đầu ra J có Histogram H cho trước.

Ta có r_k và z_k lần lượt là các bậc màu của hai ảnh I và J . Thuật toán Histogram Matching thực hiện xây dựng hàm biến đổi khớp biểu đồ Histogram của I với biểu đồ Histogram của J như sau.

- Xây dựng Histogram của ảnh gốc I .
- Thực hiện tìm các hàm biến đổi cân bằng Histogram $T : r \rightarrow s$ của I và $G : z \rightarrow s$ của J .
- Xây dựng phép biến đổi ngược $G^{-1} : s \rightarrow z$.
- Thu được hàm hợp của T và $G^{-1} : r \rightarrow z$.

2.3.3 Histogram Intersection (HI)

Thuật toán Histogram Intersection là thuật toán đơn giản để tính toán tỷ lệ khớp giữa hai biểu đồ histogram. Ví dụ cho hai histogram của hai ảnh n bậc màu là H_M và H_T . Giao (intersection) HI của hai histogram được tính theo công thức:

$$HI = \sum_{i=1}^n \min(h_M(i), h_T(i)),$$

trong đó $\sum_{i=1}^n h_M(i) = \sum_{i=1}^n h_T(i) = 1$.

Giá trị HI thu được càng cao thì chứng tỏ hai ảnh đầu vào có độ tương đồng cao.

2.4 Gaussian Weighted Histogram Intersection (GWHI)

Trong phương pháp MPM, ta giả định những màu tương đồng với màu đã cho có trọng lượng (weight) ảnh hưởng giống nhau. Điều này chưa phản ánh tốt sự ảnh hưởng của các màu khác nhau. Trong phương pháp GWHI, một hàm Gaussian trọng lượng của khoảng cách

màu (colour distance) được áp dụng vào HI, dùng để mô tả mối quan hệ giữa khoảng cách màu và trọng lượng là:

$$GWHI = \sum_{\vec{c}_i \in C_M} \sum_{\vec{c}_j \in C_T} \min(h_M(\vec{c}_i), h_T(\vec{c}_j)) \cdot w(\|\vec{c}_i - \vec{c}_j\|), \quad (2.1)$$

trong đó

$$w(\|\vec{c}_i - \vec{c}_j\|) = \begin{cases} f(\|\vec{c}_i - \vec{c}_j\|) & \|\vec{c}_i - \vec{c}_j\| \leq BW \\ 0 & other \end{cases} \quad (2.2)$$

Hàm Gaussian $f(x)$ là

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-x^2}{2\sigma^2}\right) \quad \text{với } -\infty \leq x \leq \infty \quad (2.3)$$

Ta có:

$$\int_{-3.3\sigma}^{3.3\sigma} f(x) dx = 99.9\% \int_{-\infty}^{\infty} f(x) dx \quad (2.4)$$

nên $BW = 3.3\sigma$. Với $\sigma = 0.8Th$ tức $BW = 2.64Th$, theo [2] Th là 15. Khoảng cách màu được định nghĩa là sự khác biệt cảm nhận giữa nhau màu với công thức:

$$\|\vec{c}_i - \vec{c}_j\| = \sqrt{(l_1 - l_2)^2 + (u_1 - u_2)^2 + (v_1 - v_2)^2} \quad (2.5)$$

trong đó $\vec{c}_1 = (l_1, u_1, v_1)$, $\vec{c}_2 = (l_2, u_2, v_2)$ là hai màu được biểu diễn trong không gian màu CIE Luv.

Theo [1] và thư viện OpenCV, để chuyển đổi ảnh RGB thành CIELuv, ta cần thực hiện hai bước:

- Chuyển ảnh RGB thành XYZ:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.6)$$

- Chuyển ảnh XYZ thành CIE Luv:

$$L^* = \begin{cases} \left(\frac{29}{3}\right)^3 Y & Y \leq \left(\frac{6}{29}\right)^3 \\ 116Y^{1/3} - 16 & Y > \left(\frac{6}{29}\right)^3 \end{cases} \quad (2.7)$$

$$u^* = 13L^* \cdot (u' - u_n)$$

$$v^* = 13L^* \cdot (v' - u_n)$$

trong đó:

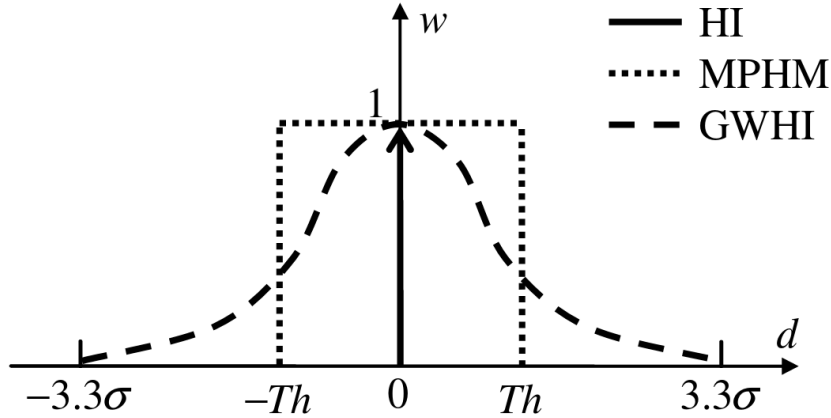
$$u' = \frac{4X}{X + 15Y + 3Z}$$

$$v' = \frac{9Y}{X + 15Y + 3Z} \quad (2.8)$$

$$u_n = 0.19793943$$

$$v_n = 0.46831096$$

khi đó $0 \leq L \leq 100, -134 \leq u \leq 220, -140 \leq v \leq 122$.



Hình 2.9: Quan hệ trọng số và khoảng cách màu trong ba thuật toán histogram

2.5 Colour Ratio Gradients (CRG)

Các phương pháp MPHM, GWHI được nêu ở trên đều nhằm giải quyết vấn đề ảnh nhạy cảm với ánh sáng trong thuật toán histogram-based image matching, chúng không bao gồm bất kì thông tin nào về không gian mà chỉ ghi lại số lượng pixel với số màu nhất định. Phương pháp này không bị ảnh hưởng bởi những thay đổi nhỏ trong góc chụp, tuy nhiên có thể bị kết quả dương tính giả do hai hình ảnh khác nhau có cùng biểu đồ Histogram.

Phương pháp colour ratio gradient (CRG) là phương pháp sử dụng quan tỷ lệ của bộ ba màu RGB trong một vùng lân cận, nhằm mục đích giảm sự ảnh hưởng của vị trí, độ bóng, độ sáng, và một số điều kiện hình ảnh khác.

Tỷ lệ màu ba kênh RGB được định nghĩa dựa vào một lân cận

bằng công thức:

$$\begin{cases} \nabla M_{RG} = \sqrt{\left(\frac{R_1 G_2 - R_2 G_1}{R_2 G_1 + R_1 G_2}\right)^2 + \left(\frac{R_3 G_4 - R_4 G_3}{R_4 G_3 + R_3 G_4}\right)^2} \\ \nabla M_{RB} = \sqrt{\left(\frac{R_1 B_2 - R_2 B_1}{R_2 B_1 + R_1 B_2}\right)^2 + \left(\frac{R_3 B_4 - R_4 B_3}{R_4 B_3 + R_3 B_4}\right)^2} \\ \nabla M_{GB} = \sqrt{\left(\frac{G_1 B_2 - G_2 B_1}{G_2 B_1 + G_1 B_2}\right)^2 + \left(\frac{G_3 B_4 - G_4 B_3}{G_4 B_3 + G_3 B_4}\right)^2} \end{cases}$$

trong đó giá trị màu của bốn điểm $p_i (i = 1, 2, 3, 4)$ được kí hiệu là R_i, B_i, G_i . p_i là các điểm lân cận của p như hình 2.10.

	p_3	
p_1	p	p_2
	p_4	

Hình 2.10: Bốn lân cận dùng để tính tỉ lệ màu gradient của pixel trung tâm \vec{p} .

Thực hiện đếm số lần một gradient của tỉ lệ màu xuất hiện trong ảnh, thu được histogram của ảnh truy vấn Q là \mathcal{H}^Q và ảnh cần xét I_t là \mathcal{H}^{I_t} . Ta có \mathcal{D}_a và \mathcal{D}_x là histogram intersection và histogram matching định nghĩa bởi chuẩn hóa tương quan chéo (normalized cross correlation) là:

$$\mathcal{D}_a(\mathcal{H}^Q, \mathcal{H}^{I_t}) = \frac{\sum_{\vec{k}=1}^N \min\left\{\mathcal{H}^{I_t}(\vec{k}), \mathcal{H}^Q(\vec{k})\right\}}{\sum_{\vec{k}=1}^N \mathcal{H}^Q(\vec{k})} \quad (2.9)$$

$$\mathcal{D}_x(\mathcal{H}^Q, \mathcal{H}^{I_t}) = \frac{\sum_{\vec{k}=1}^N \mathcal{H}^{I_t}(\vec{k}) \mathcal{H}^Q(\vec{k})}{\sum_{\vec{k}=1}^N \left(\mathcal{H}^{I_t}(\vec{k})\right)^2} \quad (2.10)$$

trong đó \vec{k} là bậc màu và N là số bậc màu.

Ta chia thuật toán thành hai bước tách ảnh thành các vùng và

thu nhận ảnh nhằm mục đích giảm số lượng ảnh khớp sai.

2.5.1 Chia ảnh thành các vùng đặc trưng (Segmentation into Texture Regions)

- Khởi tạo: Ảnh đầu vào là ảnh truy vấn Q và ảnh cần xét I_k . Từ gradient tỉ lệ màu, ta thu được histogram tương ứng $\mathcal{H}^Q, \mathcal{H}^{I_k}$.
- Phân tách: Ta thực hiện tách một góc phần tư thành 4 góc phần tư nhỏ khi nó không đủ độ tương thích được tính theo \mathcal{D}_a . Nếu \mathcal{D}_a dưới ngưỡng cho trước, ta coi góc phần tư có kết cấu tương ứng ảnh truy vấn, ngược lại, ta tiếp tục chia nhỏ. Việc chia nhỏ tiếp tục đến khi có kết cấu phù hợp hoặc đạt cỡ pixel nhỏ nhất, ở đây ta đặt là 8×8 .
- Gộp lại: Thực hiện nối các vùng lân cận lại. Ta cho phép các vùng lân cận được gộp vào khi \mathcal{D}_x của chúng với ảnh truy vấn đạt dưới ngưỡng nhất định. Ta bắt đầu từ khối có độ tương đồng cao nhất với ảnh truy vấn và xem xét các khối liền kề, việc xem xét dừng lại khi mọi khối lân cận gộp vào đều cho \mathcal{D}_x vượt ngưỡng.
- Mở rộng vùng: Do có rất nhiều vùng nhỏ ở biên vùng liên kết, cần có phương pháp cho phép các vùng đủ lớn có thể thực hiện mở rộng. Thuật toán chọn những vùng có cỡ đủ lớn, sau đó cho vùng gộp với vùng lân cận phù hợp nhất, đồng thời kết quả gộp có \mathcal{D}_x thỏa mãn độ đo kết cấu cho trước. Quá trình gộp dừng lại khi không còn vùng lớn nào cần gộp.
- Output: Một ảnh gán nhãn xây bằng cách gán nhãn từng pixel dựa vào vùng của pixel. Mỗi vùng có một nhãn j , $1 \leq j \leq$ số vùng cuối.

2.5.2 Thu thập ảnh (Color Constant Texture Retrieval)

Để tiết kiệm thời gian, ta thực hiện theo các bước:

- Bước 1: Lọc sơ các ảnh bằng histogram intersection, chọn ra những ảnh có điểm cao nhất.
- Bước 2: áp dụng thuật toán phân đoạn ảnh (segmentation) ở trên cho các ảnh lọc ra từ bước 1.
- Bước 3: Với mỗi ảnh cần xét, ta chọn vùng có độ tương quan \mathcal{D}_a cao nhất với ảnh gốc để so sánh, sau đó sắp xếp chúng theo \mathcal{D}_x .

2.6 Colour Edge Cooccurrence Histogram (CECH)

2.6.1 Cách tính CECH histogram

Cho ảnh F , ta thực hiện tính CECH histogram của ảnh theo các bước sau:

- Bước 1: Detecting edges
 - Để thu được tập cạnh E của ảnh $F = f(x, y)$, ta dùng các điểm cạnh giả có cường độ gradient mạnh. Gradient của ảnh $f(x, y)$ tại điểm (x, y) được định nghĩa bằng:

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (2.11)$$

độ lớn và hướng của vecto này được, kí hiệu bởi $\nabla f(x, y)$ và

$\theta(x, y)$ là:

$$\nabla f(x, y) = \sqrt{G_x^2 + G_y^2} \quad (2.12)$$

và

$$\theta(x, y) = \tan^{-1} \left(\frac{G_y}{G_x} \right) \quad (2.13)$$

- Hướng của cạnh tại vị trí (x, y) vuông góc với hướng của vecto gradient. Tính gradient của ảnh dựa vào đạo hàm riêng $\frac{\partial f}{\partial x}$ và $\frac{\partial f}{\partial y}$ tại mọi pixel. Trong bài này, toán tử Sobel được sử dụng để lấy đạo hàm cấp 1 tại mỗi điểm.
- Các pixel có cường độ mạnh hơn ngưỡng $TH_G = 30$ cho trước được coi là cạnh và được xem xét. Do đó các tập cạnh có thể mô tả bằng:

$$E(x, y) = \begin{cases} 1 & \nabla f(x, y) \geq TH_G \\ 0 & \text{còn lại} \end{cases} \quad (2.14)$$

- Bước 2: Định vị cặp pixel

- Với điểm cạnh p tại (x, y) , ta lấy hai điểm p_1, p_2 thuộc hai phía của hướng gradient của p và cách p một khoảng d nhất định. Khi đó tọa độ (x_1, y_1) và (x_2, y_2) được tính bằng

$$\begin{cases} x_1 = & x - d \times \cos \theta \\ y_1 = & y + d \times \sin \theta \end{cases} \quad (2.15)$$

và

$$\begin{cases} x_2 = & x + d \times \cos \theta \\ y_2 = & y - d \times \sin \theta \end{cases} \quad (2.16)$$

- Bước 3: Cộng lại

- Kí hiệu giá trị màu ở điểm p_1, p_2 là \vec{c}_1, \vec{c}_2 . Tập màu tương ứng của CECH được tính bằng:

$$CECH(\vec{c}_1, \vec{c}_2, d) = \text{size}(\{(p_1, p_2) | p_1, p_2 \in F \quad (2.17)$$

$$\text{và } \|p_1 - p\| = \|p_2 - p\| = d\}) \quad (2.18)$$

trong đó $\vec{c}_1, \vec{c}_2 \in C$ là tập màu của ảnh.

- Sau khi quét mọi điểm cạnh trên ảnh, ta tổng hợp được các cặp màu giống nhau trong ảnh và tổng hợp chúng vào các bậc màu của biểu đồ CECH.

- Bước 4: Chuẩn hóa CECH histogram với tổng số điểm cạnh của ảnh.

Algorithm 1 Thuật toán xây dựng CECH histogram

Require: Ảnh F , d , TH_G

Ensure: $w, h = \text{shape}(F)$

$\nabla f, \theta$: các ma trận cỡ $w * h$

$CECH = \emptyset$

for i, j in w, h **do**

$$\nabla f(i, j) = \sqrt{G_x^2 + G_y^2}$$

$$\theta(i, j) = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

if $\nabla f(i, j) < TH_G$ **then**

$$x_1 = i - d \times \cos \theta(i, j)$$

$$y_1 = j + d \times \sin \theta(i, j)$$

$$x_2 = i + d \times \cos \theta(i, j)$$

$$y_2 = j - d \times \sin \theta(i, j)$$

$$c_1 = \text{image}(x_1, y_1)$$

$$c_2 = \text{image}(x_1, y_1)$$

$$CECH(c_1, c_2) + = 1$$

end if

end for

2.6.2 Giảm số màu.

Giả sử số màu trong ảnh là N_C . Khi đó số cặp màu pixel có thể xảy ra là $N_C \times N_C$, đây cũng là chiều của biểu đồ CECH khi d cố định. Không mất tính tổng quát, giả sử hai ảnh cần xét có cùng số màu. Nếu tập màu hai ảnh giống hệt nhau, ta chỉ cần thực hiện so sánh từng bậc màu tương ứng. Do đó, số phép toán tối thiểu khi so sánh hai CECH histogram là N_C^2 . Tuy nhiên, nếu tập màu của hai ảnh khác nhau, với mỗi bậc màu trong CECH này, cần so sánh với mọi bậc màu trong CECH còn lại. Do đó, số phép toán là N_C^4 để so sánh tìm sự giống nhau giữa hai biểu đồ CECH.

Với một ảnh 24 bit màu, ta có $N_C = 2^{24}$, rất khó để có thể thực hiện so sánh trong thời gian thực, do vậy ta cần một phương pháp để giảm số màu.

Theo phương pháp của Xiaolin Wu, với một tập màu $\Omega(c_i, c_j]$ ($c_i = (c_{ir}, c_{ig}, c_{ib})$), ta cần chọn c sao cho biểu thức sau nhỏ nhất:

$$E(c_i, c] + E(c, c_j) \quad (2.19)$$

trong đó

$$\begin{aligned} c \in \{c_{jr} \times c_{jg} \times (c_{ib}, c_{jb}]\} \cup \{c_{jr} \times (c_{ig}, c_{jg}] \times c_{jb}\} \\ \cup \{(c_{ir}, c_{jr}] \times c_{jg} \times c_{jb}\} \end{aligned} \quad (2.20)$$

Các công thức, ký hiệu được định nghĩa như sau:

- Một màu c thuộc tập $\Omega(c_i, c_j]$ khi ba giá trị c_r, c_g, c_b lần lượt thuộc tương ứng $(c_{ir}, c_{jr}]$, $(c_{ig}, c_{jg}]$, $(c_{ib}, c_{jb}]$.
- $\omega(c_i, c_j] = \sum_{c \in (c_i, c_j]} P(c)$ với $P(c)$ là tỉ lệ màu c trong ảnh cần xét.
- $E(c_i, c_j] = \sum_{c \in \Omega(c_i, c_j]} c^2 P(c) - \frac{[\sum_{c \in \Omega(c_i, c_j]} c P(c)]^2}{\omega(c_i, c_j]}$

Để rút gọn số phép tính cho công thức, ta áp dụng:

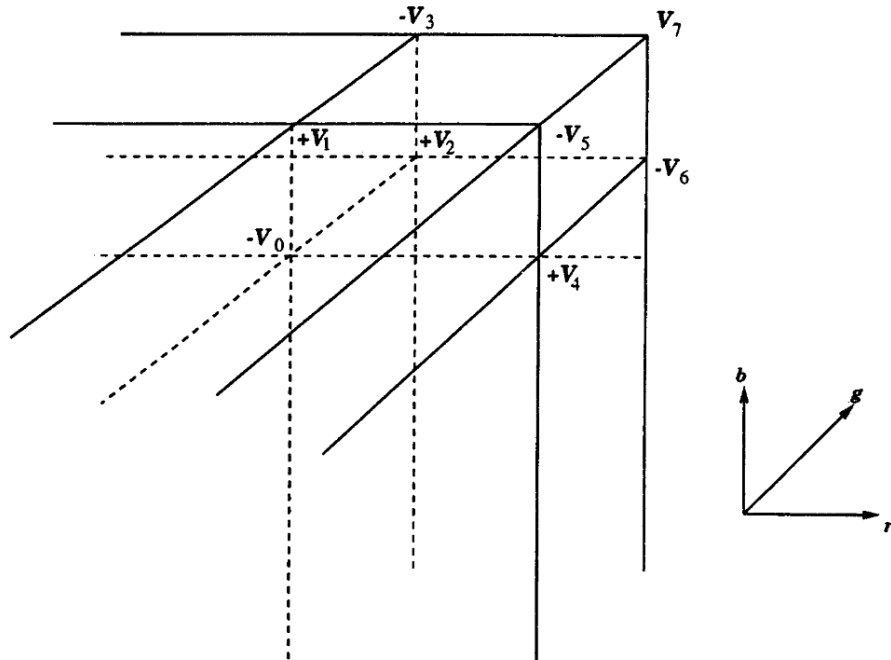
$$\sum_{c \in \Omega(c_i, c_j]} f(c) P(c) = \left[\sum_{c \in \Omega(0, v_7]} - \sum_{c \in \Omega(0, v_6]} - \sum_{c \in \Omega(0, v_5]} + \sum_{c \in \Omega(0, v_4]} \right. \quad (2.21)$$

$$\left. - \sum_{c \in \Omega(0, v_3]} + \sum_{c \in \Omega(0, v_2]} + \sum_{c \in \Omega(0, v_1]} - \sum_{c \in \Omega(0, v_0]} \right] f(c) P(c)$$

trong đó v_i là tám đỉnh của $\Omega(c_i, c_j]$

$$\begin{aligned} v_0 &= (c_{ir}, c_{ig}, c_{ib}), v_4 = (c_{jr}, c_{ig}, c_{ib}) \\ v_1 &= (c_{ir}, c_{ig}, c_{jb}), v_5 = (c_{jr}, c_{ig}, c_{jb}) \\ v_2 &= (c_{ir}, c_{jg}, c_{ib}), v_6 = (c_{jr}, c_{jg}, c_{ib}) \\ v_3 &= (c_{ir}, c_{jg}, c_{jb}), v_7 = (c_{jr}, c_{jg}, c_{jb}) \end{aligned}$$

,vecto 0 là điểm thỏa mãn $\sum_{c \in \Omega(-\infty, 0]} P(c) = 0$.



Hình 2.11: 8 đỉnh của khoảng màu

Thay $f(c)$ bằng c^d , với $c^0 = 1, c^2 = cc^T$, ta thu được:

$$M_d(c_t) = \sum_{c \in \Omega(0, c_t]} c^d P(c) \quad (2.22)$$

Ta thấy 4 trên 8 $M_d(c)$ thuộc công thức (2.21) có giá trị không đổi với mọi c thuộc tập 2.20, từ đó giảm số lần cần tính lại các giá trị biểu thức dạng $\sum_{c \in \Omega(c_i, c_j]} c^d P(c)$.

Như vậy, với mỗi $\Omega(c_i, c_j]$, cần tìm giá trị nhỏ nhất của biểu thức sau:

$$\begin{aligned} E(c_i, c] + E(c, c_j] &= \sum_{c \in \Omega(c_i, c_j]} c^2 P(c) \\ &\quad - \frac{[\sum_{c \in \Omega(c_i, c]} cP(c)]^2}{\omega(c_i, c]} - \frac{[\sum_{c \in \Omega(c, c_j]} cP(c)]^2}{\omega(c, c_j]}, \end{aligned} \quad (2.23)$$

tương ứng với việc tìm giá trị lớn nhất của:

$$\begin{aligned} &\frac{[\sum_{c \in \Omega(c_i, c]} cP(c)]^2}{\omega(c_i, c]} + \frac{[\sum_{c \in \Omega(c, c_j]} cP(c)]^2}{\omega(c, c_j]} \\ &= \frac{[\sum_{c \in \Omega(c_i, c]} cP(c)]^2}{\omega(c_i, c]} + \frac{[\sum_{c \in \Omega(c_i, c_j]} cP(c) - \sum_{c \in \Omega(c_i, c]} cP(c)]^2}{\omega(c_i, c_j] - \omega(c_i, c]}. \end{aligned} \quad (2.24)$$

Algorithm 2 Thuật toán tính $P(c)$

Require: Ảnh F kích thước w, h

Ensure: P là ma trận *zeros* có kích thước $256 \times 256 \times 256$

for i, j in w, h **do**

$r, g, b = F(i, j)$

$P[r, g, b] + = 1$

end for

$P = \frac{P}{w * h}$

Algorithm 3 Thuật toán tính $M_d(c_t)$

Require: $P(c), c_t, d$ **Ensure:** $M_d(c_t) = 0$

```
for  $r, g, b$  in  $c_{tr}, c_{tg}, c_{tb}$  do
     $c = [r, g, b]$ 
     $c_d = [1, c, cc^T]$ 
     $M_d(c_t) += c_d[d] * P(c)$ 
end for
```

Algorithm 4 Thuật toán giảm màu

Require: Ảnh F , số màu $d = [d_r, d_g, d_b]$, histogram $P(c)$ với $c = [c_r, c_g, c_b]$ **Ensure:** $w, h = \text{shape}(F)$

```
convert_rule = [ ]
for  $r, g, b$  in  $d_r, d_g, d_b$ : do
     $c_i = \left\lceil \frac{256}{d_r} * r, \frac{256}{d_g} * g, \frac{256}{d_b} * b \right\rceil$ 
     $c_j = \left\lceil \frac{256}{d_r} * (r + 1), \frac{256}{d_g} * (g + 1), \frac{256}{d_b} * (b + 1) \right\rceil$ 
    max_value = 0, best_c = [0, 0, 0]
    for  $c$  in  $D$ : ( $D$  là tập dựa theo công thức 2.20) do
        value =  $\frac{[\sum_{c \in \Omega(c_i, c]} cP(c)]^2}{\omega(c_i, c)} + \frac{[\sum_{c \in \Omega(c_i, c_j]} cP(c) - \sum_{c \in \Omega(c_i, c]} cP(c)]^2}{\omega(c_i, c_j) - \omega(c_i, c)}$ 
        if value > max_value then
            max_value = value
            best_c =  $c$ 
        end if
    end for
    convert_rule[ $r, g, b$ ] = best_c
end for
for  $x, y$  in  $w, h$  do
     $r = \left\lceil \frac{F(x, y)[r]}{256} * d_r \right\rceil$ 
     $g = \left\lceil \frac{F(x, y)[g]}{256} * d_g \right\rceil$ 
     $b = \left\lceil \frac{F(x, y)[b]}{256} * d_b \right\rceil$ 
     $F(x, y) = \text{convert\_rule}[r, g, b]$ 
end for
```

2.6.3 So sánh biểu đồ CECH bằng GWHI

Để so sánh hai biểu đồ CECH có tập màu tương tự nhau, khởi tạo η bằng 0. Với mỗi $\text{CECH}_M(\vec{c}_i, \vec{c}_j, d)_{|\vec{c}_i, \vec{c}_j \in C_M}$ trong biểu đồ CECH của ảnh cần xét, quét từng $\text{CECH}_T(\vec{c}_k, \vec{c}_l, d)_{|\vec{c}_k, \vec{c}_l \in C_t}$ trong biểu đồ CECH của ảnh gốc khi thực hiện các bước (1) đến (4) đến khi tất cả các cặp màu trong CECH của ảnh cần xét đã được quét:

1. Tính khoảng cách $d_1 = \|\vec{c}_i - \vec{c}_k\|, d_2 = \|\vec{c}_j - \vec{c}_l\|$, chọn $d_C = \max(d_1, d_2)$.
2. Nếu $d_C \leq BW = 52.8??$, phần giao nhau giữa hai biểu đồ CECH được định nghĩa:

$$\text{its} = \min(\text{CECH}_M(\vec{c}_i, \vec{c}_j, d), \text{CECH}_T(\vec{c}_k, \vec{c}_l, d)) \quad (2.25)$$

3. Áp dụng hàm Gaussian của khoảng cách vào its ta có trọng số:

$$\text{its}' = \text{its} \times \exp\left(-\frac{d_C^2}{2\sigma^2}\right) \quad \text{với} \quad -BW \leq d_C \leq BW, \quad (2.26)$$

trong đó σ và BW được nêu lên trong phần trước.

4. Tổng các trọng số ta được: $\eta \leftarrow \eta + \text{its}'$.

η cuối cùng là tỉ lệ đối sánh giữa hai biểu đồ CECH thuộc khoảng $[0, 1]$. η càng lớn, hai ảnh càng giống nhau.

Tài liệu tham khảo

- [1] Hsien-Che Lee and Gaurav Sharma, “Introduction to Color Imaging Science”, *Journal of Electronic Imaging - J ELECTRON IMAGING*, Vol 15, 2006.
- [2] Wenjing Jia, Huaifeng Zhang, Xiangjian He and Qiang Wu, “Gaussian Weighted Histogram Intersection for License Plate Classification”, Vol 3, 2006, 574–577.