BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

---------o0o---------

HIỆU CHỈNH ÁNH SÁNG TRONG ẢNH

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
NGÀNH CÔNG NGHỆ THÔNG TIN

Sinh viên thực hiên: **PHẠM VĂN BÌNH**

Giáo viên hướng dẫn: **PGS.TS ĐỖ NĂNG TOÀN**

Mã số sinh viên: **111230**

Hải Phòng - 2011

MỤC LỤC

DANH MỤC HÌNH VẼ

MỞ ĐẦU

Trong xã hội hiện nay, ảnh số đóng một vai trò quan trọng trong đời sống con người. Ảnh số không chỉ được sử dụng cuộc sống hằng ngày mà nó còn góp phần quan trọng trong việc cung cấp thông tin về vật thể, sự kiện… trong công tác nghiên cứu khoa học. Đối với một bức ảnh, ánh sáng có vai trò quan trọng, ảnh hưởng trực tiếp ảnh hưởng tới chất lượng của bức ảnh. Hiện nay, có rất nhiều phương pháp hiệu chỉnh ánh sáng từ đơn giản như tăng giảm độ sáng, hiệu chỉnh gamma… đến các phương pháp phức tạp hơn như hồi phục màu của vật thể bị ánh sáng chiếu vào gây thay đổi cảm nhận màu sắc… Không chỉ vậy, bài toán còn từ chỉ có một nguồn sáng tới nhiều nguồn sáng, ánh sáng chiếu đều và ánh sáng chiếu không đều… để phục nhu cầu của con người.

Hiệu chỉnh ánh sáng được quan tâm như vậy vì nó có ứng dụng rất lớn trong thực tế. Sau đây là một vài ứng dụng trong thực tế của hiệu chỉnh ánh sáng:

- Hồi phục màu sắc của vật thể chịu tác động của ánh sáng.
- Trong nhận dạng, một số trường hợp khó khăn do ánh sáng gây ra. Hiệu chỉnh ánh sáng có thể giải quyết vấn đề này.
- Tìm kiếm, so sánh ảnh.
- Chức năng tự động hiệu chỉnh ánh sáng trong các máy ảnh số hiện nay.
- Nâng cao chất lượng ánh sáng trong ảnh.

Nội dung đồ án tốt nghiệp gồm:

Chương 1: Nêu khái quái và các khái niệm của xử lý ảnh số và hiệu chỉnh ánh sáng trong ảnh số.

Chương 2: Nêu một số phương pháp và thuật toán hiệu chỉnh ánh sáng trong ảnh số.

Chương 3: Giời thiệu chương trình hiệu chỉnh ánh sáng và chạy thử nghiệm chương trình.

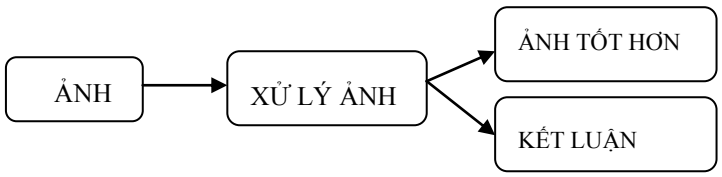
CHƯƠNG 1: KHÁI QUÁT VỀ XỬ LÝ ẢNH VÀ HIỆU CHỈNH ÁNH SÁNG

1.1 KHÁI QUÁT VỀ XỬ LÝ ẢNH

1.1.1 Xử lý ảnh là gì?

Con người thu nhận thông tin qua các giác quan, trong đó thị giác đóng vai trò quan trọng nhất. Những năm trở lại đây với sự phát triển của phần cứng máy tính, xử lý ảnh và đồ hoạ đó phát triển một cách mạnh mẽ và có nhiều ứng dụng trong cuộc sống. Xử lý ảnh và đồ hoạ đóng một vai trò quan trọng trong tương tác người - máy.

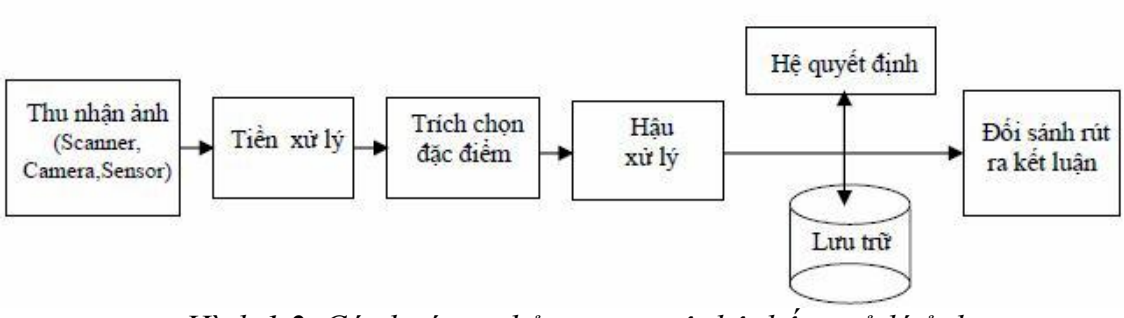
Quá trình xử lý ảnh được xem như là quá trình thao tác ảnh đầu vào nhằm cho ra kết quả mong muốn. Kết quả đầu ra của một quá trình xử lý ảnh có thể là một ảnh "tốt hơn" hoặc một kết luận.



Hình 1.1. Quá trình xử lý ảnh

Ảnh có thể xem là tập hợp các điểm ảnh và mỗi điểm ảnh được xem như là đặc trưng cường độ sáng hay một dấu hiệu nào đó tại một vị trí nào đó của đối tượng trong không gian và nó có thể xem như một hàm n biến $P(c_1, c_2,..., c_n)$. Do đó, ảnh trong xử lý ảnh có thể xem như ảnh n chiều.

Sơ đồ tổng quát của một hệ thống xử lý ảnh:



Hình 1.2. Các bước cơ bản trong một hệ thống xử lý ảnh

- **Thu nhận ảnh (Image acquisition)**

Các thiết bị thu nhận ảnh có hai loại chính ứng với hai loại ảnh thông dụng Raster và Vector. Các thiết bị thu nhận ảnh thông thường Raster là camera. Các thiết bị thu nhận ảnh thông thường Vector là sensor hoặc bộ số hoá (digitalizer) hoặc được chuyển đổi từ ảnh Raster . Các thiết bị thu ảnh thông thường gồm camera cộng với bộ chuyển đổi tương tự số AD (Analog to Digital) hoặc scanner chuyên dụng. Các thiết bị thu nhận ảnh này có thể cho ảnh đen trắng hoặc ảnh màu. Đầu ra của scanner là ảnh ma trận số mà ta quen gọi là bản đồ ảnh (ảnh Bitmap). Bộ số hoá (digitalizer) sẽ tạo ảnh vectơ có hướng. Nhìn chung, các hệ thống thu nhận ảnh thực hiện hai quá trình:

Cảm biến: biến đổi năng lượng quang học (ánh sáng) thành năng lượng điện.

Tổng hợp năng lượng điện thành ảnh.

- **Tiền xử lý (Image processing)**

Tiền xử lý là bước tăng cường ảnh để nâng cao chất lượng ảnh. Do những nguyên nhân khác nhau: có thể do chất lượng thiết bị thu nhận ảnh , do nguồn sáng hay do nhiễu, ảnh có thể bị suy biến. Do vậy cần phải tăng cường và khôi phục lại ảnh để làm nổi bật một số đặc tính chính của ảnh, hay làm cho ảnh gần giống nhất với trạng thái gốc - trạng thái trước khi ảnh bị biến dạng.

- **Trích chọn đặc điểm (Feature extraction)**

Vì lượng thông tin chứa trong ảnh là rất lớn, trong khi đó đa số ứng dụng chỉ cần một số thông tin đặc trưng nào đó, cần có bước trích chọn đặc điểm để giảm lượng thông tin khổng lồ ấy. Các đặc trưng của ảnh thường gồm: mật độ xám, phân bố xác suất, phân bố không gian, biên ảnh.

- **Hậu xử lý**

Nếu lưu trữ ảnh trực tiếp từ các ảnh thô (brut image) theo kiểu bản đồ ảnh đòi hỏi dung lượng bộ nhớ lớn, tốn kém mà nhiều khi không hiệu quả theo quan điểm ứng dụng. Thường người ta không biểu diễn toàn bộ ảnh thô mà tập trung đặc tả các đặc trưng của ảnh như biên ảnh (boundary) hay vùng ảnh (region) . Một số phương pháp biểu diễn thường dùng:

- ✓ Biểu diễn mã loạt dài (Run-Length Code).
- ✓ Biểu diễn mã xích (Chaine -Code).
- ✓ Biểu diễn mã tứ phân (Quad-Tree Code).

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 nhiễu. 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- hệ quyết định được phát huy.

- **Đối sánh rút ra kết luận**

So sánh ảnh sau bước hậu xử lý với mẫu chuẩn hoặc ảnh đã được lưu trữ từ trước, phục vụ cho các mục đích nhận dạng và nội suy ảnh.

1.1.2 Một số vấn đề cơ bản trong xử lý ảnh

1.1.2.1 Một số khái niệm cơ bản

**** Ảnh và 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 hoá về không gian) và lượng hoá thành phần giá trị mà về nguyên tắc bằng mắt thường không 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 - điểm ảnh.

Điểm ảnh được xem như là dấu hiệu hay cường độ sáng tại một toạ độ trong không gian của đối tượng .

Ảnh được xem như 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 điểm ảnh. Người ta thường kí hiệu I(x,y) để chỉ một điểm ảnh. Thường giá trị của n chọn bằng p và bằng 256. Một điểm ảnh có thể lưu trữ trên 1, 4, 8 hay 24 bit .

Về mặt toán học có thể xem ảnh là một hàm hai biến f(x,y) với x, y là các biến tọa độ. Giá trị số ở điểm (x,y) tương ứng với giá trị xám hoặc độ sáng của ảnh (x là các cột, y là các hàng). Giá trị của hàm ảnh f(x,y) được hạn chế trong phạm vi của các số nguyên dương: $0 \leq f(x,y) \leq fmax$. Thông thường đối với ảnh xám, giá trị fmax là 255 (2^8=256) bởi vì mỗi phần tử ảnh được mã hóa bởi một byte. Khi quan tâm đến ảnh màu, ta có thể mô tả màu qua ba hàm số: thành phần màu đỏ qua hàm

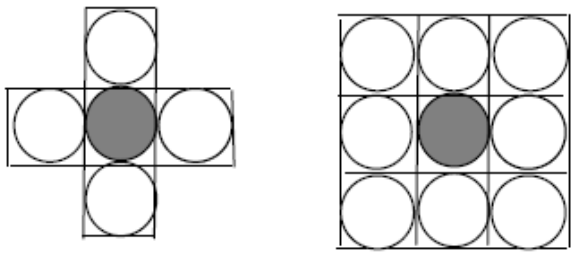
R(x,y), thành phần màu lục qua hàm G(x,y) và thành phần màu lam qua hàm B(x,y).

Số điểm ảnh tạo nên một ảnh gọi là *độ phân giải (resolusion)*. Độ phân giải thường được biểu thị bằng số điểm ảnh theo chiều dọc và chiều ngang của ảnh. Ảnh có độ phân giải càng cao càng rõ nét. Như vậy, ảnh càng to thì càng bị vỡ hạt, độ mịn càng kém. Ảnh có thể được biểu diễn theo mô hình Vector hoặc mô hình Raster:

Mô hình Raster

Đây là mô hình biểu diễn ảnh thông dụng nhất hiện nay. Ảnh được biểu diễn dưới dạng ma trận các điểm ảnh. Tùy theo nhu cầu thực tế mà mỗi điểm ảnh có thể được biểu diễn bởi một hay nhiều bit. Mô hình Raster rất thuận lợi cho hiển thị và in ấn .

Khi xử lý các ảnh Raster, chúng ta quan tâm đến mối quan hệ trong vùng lân cận của các điểm ảnh. Các điểm ảnh có thể xếp hàng trên một lưới (Raster) hình vuông, lưới hình lục giác hoặc theo một cách hoàn toàn ngẫu nhiên với nhau. Cách sắp xếp theo hình vuông là được quan tâm đến nhiều nhất và có hai loại: điểm 4 láng giềng (4 liền kề) hoặc 8 láng giềng (8 liền kề) được minh hoạ như sau:



Hình 1.3. Quan hệ giữa các điểm ảnh

Mô hình Vector

Biểu diễn ảnh ngoài mục đích tiết kiệm không gian lưu trữ, dễ dàng cho hiển thị và in ấn, còn phải đảm bảo dễ dàng trong lựa chọn, sao chép, di chuyển, tìm kiếm…Theo những yêu cầu này, kỹ thuật biểu diễn Vector tỏ ra ưu việt hơn . Trong mô hình Vector người ta sử dụng hướng giữa các Vector của điểm ảnh lân cận để mã hoá và tái tạo hình ảnh ban đầu. Ảnh Vector được thu nhận trực tiếp từ các thiết bị số hóa như Digital hoặc được chuyển đổi từ ảnh Raster thông qua các chương trình số hóa. Công nghệ phần cứng cung cấp những thiết bị xử lý với tốc độ nhanh

và chất lượng cao cho cả đầu vào và ra, nhưng lại chỉ hỗ trợ cho ảnh Raster. Do vậy, những nghiên cứu về biểu diễn Vector đều tập trung chuyển đổi từ ảnh Raster .

Mức xám và lược đồ mức xám

Mức xám (Gray level)

Mức xám là kết quả sự mã hoá tương ứng một cường độ sáng của mỗi điểm ảnh với một giá trị số - kết quả của quá trình lượng hoá. Cách mã hoá kinh điển thường dùng 16, 32 hay 64 mức. Mã hoá 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 hoá bởi 8 bit .

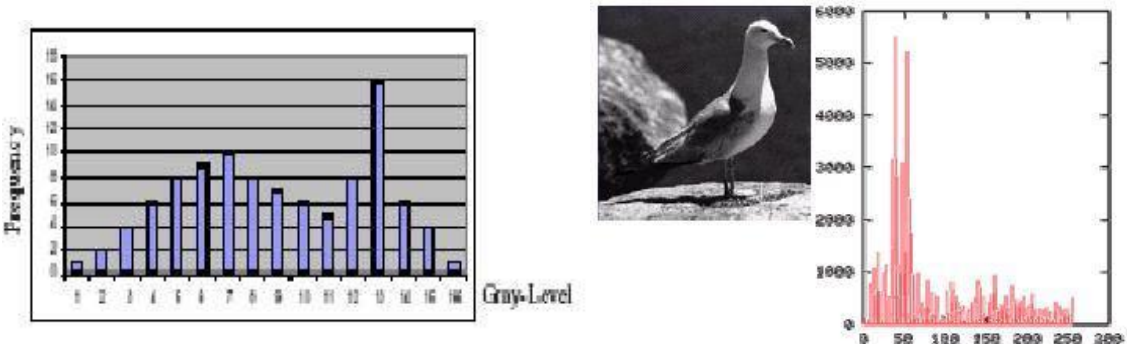
Ảnh có hai mức xám được gọi là *ảnh nhị phân*. Mỗi điểm ảnh của ảnh nhị phân chỉ có thể là 0 hoặc 1. Ảnh có mức xám lớn hơn 2 được gọi là *ảnh đa cấp xám hay ảnh màu*. *Ảnh đen trắng* là ảnh chỉ có hai màu đen và trắng, mức xám ở các điểm ảnh có thể khác nhau.

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 $2^{8x3} = 2^{24}$ 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.

Lược đồ mức xám (Histogram)

Lược đồ mức xám của một ảnh, từ này về sau ta qui ước gọi là *lược đồ xám* hay *biểu đồ tần suất,* là một hàm cung cấp tần suất xuất hiện của mỗi mức xám.

Lược đồ xám được biểu diễn trong hệ tọa độ vuông góc Oxy. Trong hệ tọa độ này, trục hoành biểu diễn cho số mức xám từ 0 đến N, N là số mức xám (256 mức trong trường hợp ảnh xám mà chúng ta 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 đi một chút: trục tung là tỉ lệ số điểm ảnh có cùng mức xám trên tổng số điểm ảnh.

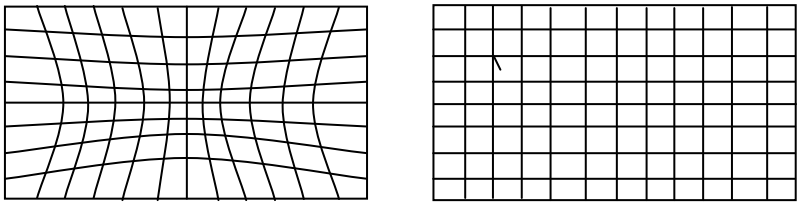


Hình 1.4. Lược đồ xám của ảnh

Lược đồ xám cung cấp rất nhiều thông tin về phân bố mức xám của ảnh. Theo thuật ngữ của xử lý ảnh gọi là tính động của ảnh. Tính động của ảnh cho phép phân tích trong khoảng nào đó phân bố phần lớn các mức xám của ảnh: ảnh rất sáng hay ảnh rất đậm. Nếu ảnh sáng, lược đồ xám nằm bên phải (mức xám cao), còn ảnh đậm lược đồ xám nằm bên trái (mức xám thấp).

1.1.2.2 Nắn chỉnh biến dạng

Ảnh thu nhận thường bị biến dạng do các thiết bị quang học và điện tử.



Ảnh thu nhận **Ảnh mong muốn**

Hình 1.5. Ảnh thu nhận và ảnh mong muốn

Để khắc phục người ta sử dụng các phép chiếu, các phép chiếu thường được xây dựng trên tập các điểm điều khiển.

Giả sử (P_i, P_i') $i = \overline{1,n}$ có n các tập điều khiển

Tìm hàm f: $P_i \mapsto f(P_i)$ sao cho

$$\sum_{i=1}^{n} \left| \left| f(P_i) - P_i' \right| \right|^2 \to \min$$

Giả sử ảnh bị biến đổi chỉ bao gồm: Tịnh tiến, quay, tỷ lệ, biến dạng bậc nhất tuyến tính. Khi đó hàm f có dạng:

$$f(x, y) = (a_1x + b_1y + c_1, a_2x + b_2y + c_2)$$

Ta có:

$$\phi = \sum_{i=1}^{n} (f(Pi) - Pi')^2 = \sum_{i=1}^{n} \left[(a_1x_i + b_1y_i + c_1 - x_i')^2 + (a_2x_i + b_2y_i + c_2 - y_i')^2 \right]$$

Để cho $\phi \to \min$

$$\begin{cases} \dfrac{\partial \phi}{\partial a_1} = 0 \\ \dfrac{\partial \phi}{\partial b_1} = 0 \\ \dfrac{\partial \phi}{\partial c_1} = 0 \end{cases} \Leftrightarrow \begin{cases} \sum_{i=1}^{n} a_1x_i^2 + \sum_{i=1}^{n} b_1x_iy_i + \sum_{i=1}^{n} c_1x_i = \sum_{i=1}^{n} x_ix_i' \\ \sum_{i=1}^{n} a_1x_iy_i + \sum_{i=1}^{n} b_1y_i^2 + \sum_{i=1}^{n} c_1y_i = \sum_{i=1}^{n} y_ix_i' \\ \sum_{i=1}^{n} a_1x_i + \sum_{i=1}^{n} b_1y_i + nc_1 = \sum_{i=1}^{n} x_i' \end{cases}$$

Giải hệ phương trình tuyến tính tìm được a_1, b_1, c_1

Tương tự tìm được a_2, b_2, c_2

$\Rightarrow$ Xác định được hàm f

1.1.2.3 Khử nhiễu

Có 2 loại nhiễu cơ bản trong quá trình thu nhận ảnh

Nhiều hệ thống: là nhiễu có quy luật có thể khử bằng các phép biến đổi

Nhiễu ngẫu nhiên: vết bẩn không rõ nguyên nhân $\to$ khắc phục bằng các phép lọc

1.1.2.4 Chỉnh mức xám

Nhằm khắc phục tính không đồng đều của hệ thống gây ra. Thông thường có 2 hướng tiếp cận:

Giảm số mức xám: Thực hiện bằng cách nhóm các mức xám gần nhau thành một bó. Trường hợp chỉ có 2 mức xám thì chính là chuyển về ảnh đen trắng. Ứng dụng: in ảnh màu ra máy in đen trắng

Tăng số mức xám: Thực hiện nội suy ra các mức xám trung gian bằng kỹ thuật nội suy. Kỹ thuật này nhằm tăng cường độ mịn cho ảnh.

1.1.2.5 Phân tích ảnh

Là khâu quan trọng trong quá trình xử lý ảnh để tiến tới hiểu ảnh. Trong phân tích ảnh việc trích chọn đặc điểm là một bước quan trọng. Các đặc điểm của đối tượng được trích chọn tuỳ theo mục đích nhận dạng trong quá trình xử lý ảnh. Có thể nêu ra một số đặc điểm của ảnh sau đây:

Đặc điểm không gian: Phân bố mức xám, phân bố xác suất, biên độ, điểm uốn v.v..

Đặc điểm biến đổi: Các đặc điểm loại này được trích chọn bằng việc thực hiện lọc vùng (zonal filtering). Các bộ vùng được gọi là "mặt nạ đặc điểm" (feature mask) thường là các khe hẹp với hình dạng khác nhau (chữ nhật, tam giác, cung tròn v.v..)

Đặc điểm biên và đường biên: Đặc trưng cho đường biên của đối tượng và do vậy rất hữu ích trong việc trích trọn các thuộc tính bất biến được dùng khi nhận dạng đối tượng. Các đặc điểm này có thể được trích chọn nhờ toán tử gradient, toán tử la bàn, toán tử Laplace, toán tử "chéo không" (zero crossing) v.v..

Việc trích chọn hiệu quả các đặc điểm giúp cho việc nhận dạng các đối tượng ảnh chính xác, với tốc độ tính toán cao và dung lượng nhớ lưu trữ giảm xuống.

1.1.2.6 Nhận dạng

Nhận dạng tự động (automatic recognition), mô tả đối tượng, phân loại và phân nhóm các mẫu là những vấn đề quan trọng trong thị giác máy, được ứng dụng trong nhiều ngành khoa học khác nhau. Tuy nhiên, một câu hỏi đặt ra là: mẫu (pattern) là gì? Watanabe, một trong những người đi đầu trong lĩnh vực này đã định nghĩa: "Ngược lại với hỗn loạn (chaos), mẫu là một thực thể (entity), được xác định một cách ang áng (vaguely defined) và có thể gán cho nó một tên gọi nào đó". Ví dụ mẫu có thể là ảnh của vân tay, ảnh của một vật nào đó được chụp, một chữ viết, khuôn mặt người hoặc một ký đồ tín hiệu tiếng nói. Khi biết một mẫu nào đó, để nhận dạng hoặc phân loại mẫu đó có thể:

Hoặc phân loại có mẫu (supervised classification), chẳng hạn phân tích phân biệt (discriminant analyis), trong đó mẫu đầu vào được định danh như một thành phần của một lớp đã xác định.

Hoặc phân loại không có mẫu (unsupervised classification hay clustering) trong đó các mẫu được gán vào các lớp khác nhau dựa trên một tiêu chuẩn đồng dạng nào đó. Các lớp này cho đến thời điểm phân loại vẫn chưa biết hay chưa được định danh.

Hệ thống nhận dạng tự động bao gồm ba khâu tương ứng với ba giai đoạn chủ yếu sau đây:

1^o. Thu nhận dữ liệu và tiền xử lý.

2^o. Biểu diễn dữ liệu.

3^o. Nhận dạng, ra quyết định.

Bốn cách tiếp cận khác nhau trong lý thuyết nhận dạng là:

1^o. Đối sánh mẫu dựa trên các đặc trưng được trích chọn.

2^o. Phân loại thống kê.

3^o. Đối sánh cấu trúc.

4^o. Phân loại dựa trên mạng nơ-ron nhân tạo.

Trong các ứng dụng rõ ràng là không thể chỉ dùng có một cách tiếp cận đơn lẻ để phân loại "tối ưu" do vậy cần sử dụng cùng một lúc nhiều phương pháp và cách tiếp cận khác nhau. Do vậy, các phương thức phân loại tổ hợp hay được sử dụng khi nhận dạng và nay đã có những kết quả có triển vọng dựa trên thiết kế các hệ thống lai (hybird system) bao gồm nhiều mô hình kết hợp.

Việc giải quyết bài toán nhận dạng trong những ứng dụng mới, nảy sinh trong cuộc sống không chỉ tạo ra những thách thức về thuật giải, mà còn đặt ra những yêu cầu về tốc độ tính toán. Đặc điểm chung của tất cả những ứng dụng đó là những đặc điểm đặc trưng cần thiết thường là nhiều, không thể do chuyên gia đề xuất, mà phải được trích chọn dựa trên các thủ tục phân tích dữ liệu.

1.1.2.7 Nén ảnh

Nhằm giảm thiểu không gian lưu trữ. Thường được tiến hành theo cả hai cách khuynh hướng là nén có bảo toàn và không bảo toàn thông tin. Nén không bảo toàn thì thường có khả năng nén cao hơn nhưng khả năng phục hồi thì kém hơn. Trên cơ sở hai khuynh hướng, có 4 cách tiếp cận cơ bản trong nén ảnh:

- Nén ảnh thống kê: Kỹ thuật nén này dựa vào việc thống kê tần xuất xuất hiện của giá trị các điểm ảnh, trên cơ sở đó mà có chiến lược mã hóa thích hợp. Một ví dụ điển hình cho phương pháp này là *TIF

- Nén ảnh không gian: Kỹ thuật này dựa vào vị trí không gian của các điểm ảnh để tiến hành mã hóa. Kỹ thuật lợi dụng sự giống nhau của các điểm ảnh trong các vùng gần nhau. Ví dụ cho kỹ thuật này là mã nén *.PCX

- Nén ảnh sử dụng phép biến đổi: Đây là kỹ thuật tiếp cận theo hướng nén không bảo toàn và do vậy, kỹ thuật thường nén hiệu quả hơn. *.JPG chính là tiếp cận theo kỹ thuật nén này.

- Nén ảnh Fractal: Sử dụng tính chất Fractal của các đối tượng ảnh, thể hiện sự lặp lại của các chi tiết. Kỹ thuật nén sẽ tính toán để chỉ cần lưu trữ phần gốc ảnh và quy luật sinh ra ảnh theo nguyên lý Fractal.

1.2 ÁNH SÁNG VÀ HIỆU CHỈNH ÁNH SÁNG TRONG ẢNH

1.2.1 Ánh sáng và màu sắc trong ảnh số là gì?

Như đã giới thiệu ở trên, hình ảnh được số hóa dưới dạng ma trận điểm ảnh. Điểm ảnh được xem như là dấu hiệu hay cường độ sáng tại 1 toạ độ trong không gian của đối tượng. Giá trị có thể có của mỗi điểm ảnh là mức xám, màu sắc của điểm ảnh đó. Và do đó Ánh sáng trong ảnh số cũng chính là màu sắc trong ảnh.

1.2.2 Một số hệ màu

1.2.2.1 Hệ màu RGB

Mắt người có thể phân biệt hàng ngàn màu sắc khác nhau, những con số chính xác hơn vẫn còn đang được bàn cãi nhiều. Ba màu RGB (Red-Green- Blue) mã hóa hệ thống đồ họa sử dụng ba byte $(2^8)^3$ hay khoảng chừng 16 triệu màu phân biệt. Máy tính có thể phân biệt bất kỳ màu gì sau khi được mã hóa, nhưng việc mã hóa có thể không trình bày được những sự khác biệt trong thế giới thực. Mỗi điểm ảnh RGB bao gồm một byte cho màu R, một byte cho màu G và một byte cho màu B.

Việc mã hóa một màu tùy ý trong dãy hiển thị được làm bằng cách tổ hợp ba màu chính. Ví dụ: Red(255,0,0), Green(0,255,0), Blue(0,0,255), Black(0,0,0) Hệ thống màu RGB là một hệ thống màu cộng vào bởi vì mỗi màu được tạo nên bằng cách cộng thêm các phần tử vào màu đen(0,0,0)

Khuôn dạng của không gian màu RGB là định dạng phổ biến nhất của ảnh số, lý do chính là tính tương thích với màn hình hiển thị chính là màn hình vi tính. Tuy nhiên không gian màu RGB có hạn chế lớn nhất là không phù hợp với cách con người cảm nhận về màu sắc. Do đó không phù hợp cho việc ứng dụng vào tìm kiếm ảnh.

1.2.2.2 Hệ màu CMY và CMYK

Hệ thống màu CMY theo mô hình in trên giấy trắng và theo khuôn mẫu trừ từ màu trắng thay vì thêm vào từ màu đen như hệ thống màu RGB.

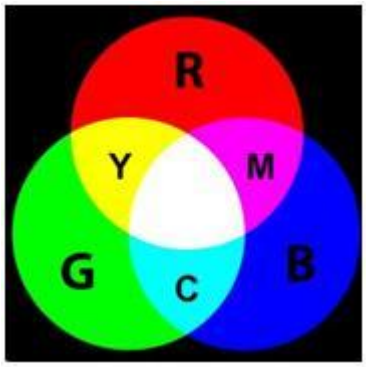
CMY là viết tắt của Cyan-Magenta-Yellow (màu lục lam, màu đỏ tươi, màu vàng), đó là ba màu chính tương ứng với ba màu mực in. Cyan hấp thu sự chiếu sáng của màu đỏ, Magenta hấp thu màu xanh lục, Yellow hấp thu màu xanh dương. Do đó, tạo ra sự phản ánh tương ứng như khi in ảnh được chiếu sáng với ánh sáng trắng. Hệ thống dưới dạng âm tính vì mã hóa theo dạng hấp thụ màu. Có một số mã

hóa như sau: trắng (0,0,0) vì không có ánh sáng trắng được hấp thụ, đen (255,255,255) vì tất cả các thành phần của màu trắng đều được hấp thụ.

Hệ thống màu CMY dường như là một sự đảo ngược của hệ thống màu RGB. Đặc tính của nó là sự đơn giản, ứng dụng nhiều trong thực tế. Tuy nhiên khuyết điểm của nó cũng tương tự như không gian màu RGB, tức là cách mã hóa khác với cách mà con người cảm nhận về màu sắc.

Máy in thường dùng hệ màu CMYK, một "phiên bản mở rộng" của hệ màu CMY. Ba chữ đầu tiên C, M, Y thì các bạn vừa mới được giải thích ở trên. Vậy chữ K là màu gì? Câu trả lời là màu đen, tiếng Anh là Black (chúng ta dùng chữ K mà không dùng chữ B vì chữ B đã được dùng cho màu xanh, Blue, trong hệ RGB). Tại sao chúng ta lại cần màu đen? Chắc chắn nhiều người trong số các chúng ta sẽ thắc mắc là tại sao không trộn ba màu C, M, Y để ra màu đen. Câu trả lời khá đơn giản. Thứ nhất là nếu trộn ba màu này lại thì sẽ rất tốn mực máy in. (Chúng ta để ý máy in thường được dùng để in văn bản trắng đen khá nhiều, nên nếu chúng ta dành riêng một lọ mực màu đen cho những việc như thế này thì hợp lý hơn). Thứ hai là giả sử chúng ta có thử trộn ba màu C, M, Y lại đi nữa thì màu đen mà chúng ta thu được trên thực tế không "đen" cho lắm, nó giống như màu xám đậm hơn.

Vậy thì CMY và RGB liên hệ với nhau như thế nào. Nói một cách ngắn gọn, nếu võng mạc chúng ta tiếp nhận ánh sáng màu R và G cùng một lúc thi chúng ta sẽ thấy màu Y. Tương tự, tiếp nhận R và B cùng lúc sẽ thấy màu M, và G kết hợp với B thì sẽ ra màu C. Hình 2.1 bên dưới sẽ minh họa cho ý này:

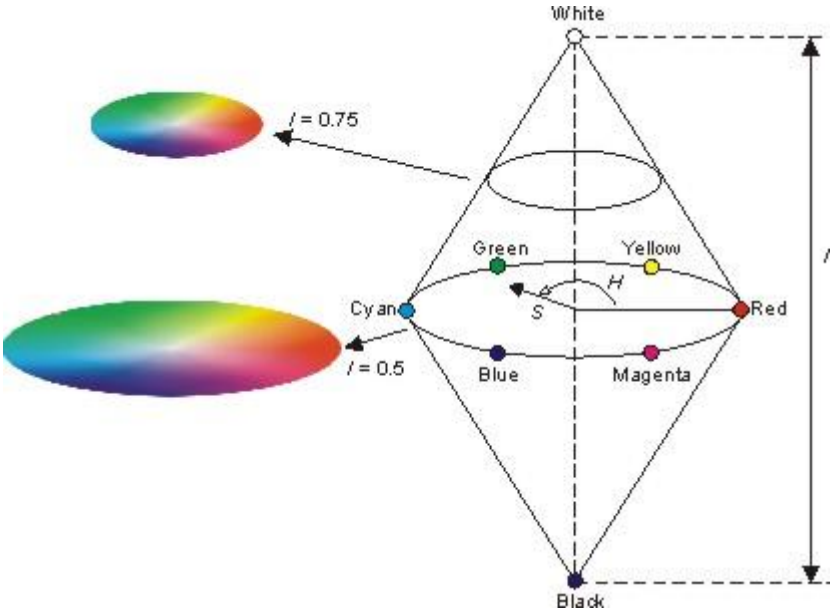


Hình 1.6. Sơ đồ liên hệ giữa không gian màu RGB và CMY

1.2.2.3 Hệ màu HSI

Hệ thống màu HSI mã hóa thông tin màu sắc bằng cách chia giá trị intensity(I) từ hai giá trị được mã hóa thuộc về độ hội tụ của màu - hue(H) và saturation(S).

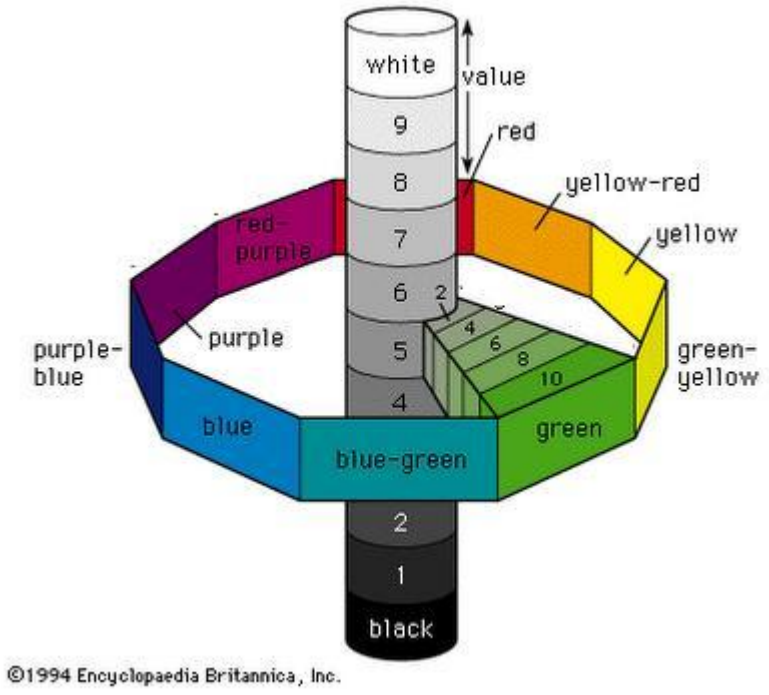
Thành phần không gian màu HSI gồm có ba phần: Hue được định nghĩa có giá trị 0-2Π , mang thông tin về màu sắc. Saturation có giá trị 0-1, mang giá trị về độ thuần khiết của thành phần Hue. Intensity (Value) mang thông tin về độ sáng của điểm ảnh.Ta có thể hình dung không gian màu HSI như là vật hình nón. Với trục chính biểu thị cường độ sáng Intensity. Khoảng cách đến trục biểu thị độ tập chung Saturation. Góc xung quanh trục biểu thị cho sắc màu Hue.



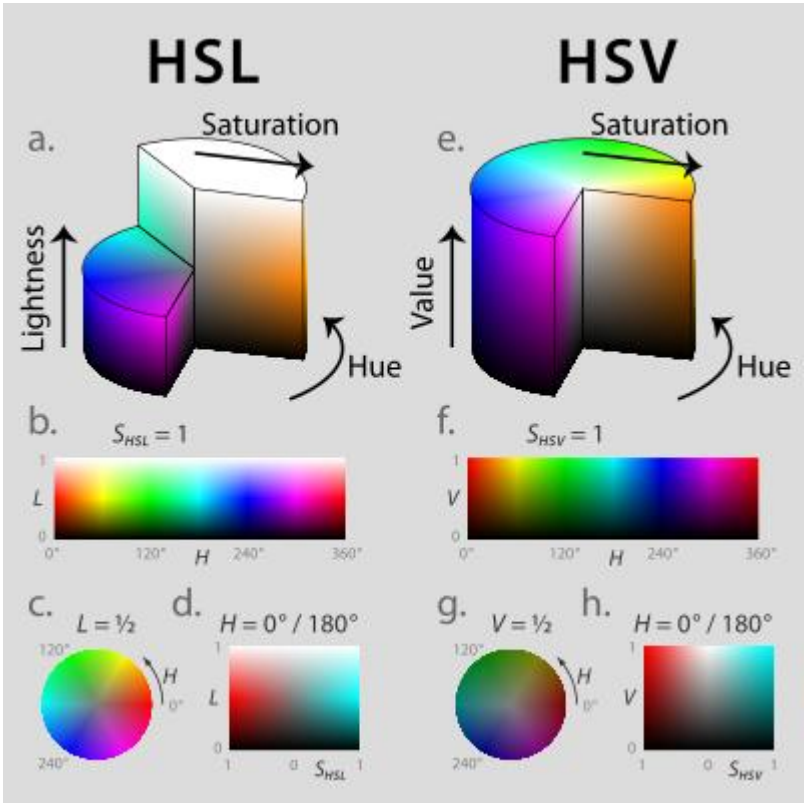
Hình 1.7. Mô hình màu HSI

Đôi khi, hệ thống màu HSI được coi như là hệ thống màu HSV dùng *Value* thay vì *Intensity*, hay HSL(HLS) dùng *Lightness* thay *Intensity*, hay HSB dùng *Brightness* thay *Intensity.*

Hệ thống màu HSI thì thích hợp hơn với một số thiết kế đồ họa bởi vì nó cung cấp sự điều khiển trực tiếp đến ánh sáng và hue. Hệ thống màu HSI cũng hỗ trợ tốt hơn cho những thuật toán xử lý ảnh vì sự tiêu chuẩn hóa về ánh sáng và tập chung vào hai tham số về độ hội tụ màu, và cường độ màu.



Hình 1.8. Mô hình màu HSV



Hình 1.9. So sánh giữa HSL và HSV

Hệ thống màu HSI có sự phân chia rõ rệt giữa ánh sáng và màu sắc.

1.2.3 Hiệu chỉnh ánh sáng trong ảnh

Hiệu chỉnh ánh sáng trong ảnh là một kỹ thuật máy tính nhằm xử lý ánh sáng trong ảnh nhằm đạt một mục đích nào đó của người dùng. Ánh sáng có thể được làm tăng, giảm hoặc loại bỏ tác động của nó lên bức ảnh.

Hiệu chỉnh ánh sáng trong ảnh là một bước quan trọng trong hệ thống xử lý ảnh. Thực hiện hiệu chỉnh ánh sáng tốt có thể tạo ra những lợi thế rất lớn cho các công việc xử lý ảnh khác sau này.

Ảnh chụp cùng một cảnh có thể thay đổi rất nhiều bởi điều kiện ánh sáng và góc chụp của camera. Chẳng hạn như chụp thẳng, chụp nghiêng, chụp ngược sáng… Với điều kiện ánh sáng khác nhau, một số chi tiết trong ảnh sẽ bị biến đổi màu sắc hoặc thậm chí bị mờ.

- Nguồn sáng làm thay đổi màu sắc: ánh sáng chiếu vào một vật thể làm mắt người nhận thấy một phần vật thể hoặc toàn vật thể có màu sắc khác. Ví dụ như ảnh chiếc rèm cửa ở dưới, vùng được ánh sáng chiếu vào có màu khác với các vùng không được chiếu, hoặc chiếu ít.

Hình 1.10. Ánh sáng làm thay đổi màu sắc vật thể

- Ánh sáng quá chói hoặc quá tối: ảnh chụp dưới điều kiện ánh sáng quá sáng hoặc quá tối có thể gây ra ảnh bị mờ, nhiễu, mất chi tiết.

Hình 1.11. Ảnh chụp trong điều kiện ánh sáng tối

CHƯƠNG 2: MỘT SỐ PHƯƠNG PHÁP HIỆU CHỈNH MÀU SẮC VÀ ÁNH SÁNG TRONG ẢNH

2.1 Hiệu chỉnh ánh sáng

Như chúng ta đã biết giá trị tại mỗi điểm ảnh thể hiện cường độ sáng tại điểm đó. Do đó, phương pháp đơn giản nhất để hiệu chỉnh ánh sáng trong ảnh là thay đổi giá trị điểm ảnh của ảnh.

Ý tưởng của phương pháp này là thay đổi một cách đồng đều giá trị tại mỗi điểm ảnh. Phương pháp này được thực hiện bằng cách cộng giá trị mỗi điểm ảnh với một số nguyên nằm trong khoảng [-255, 255].

Giả sử ta có ảnh I với kích thước m × n và một số nguyên c. Khi đó thuật toán được mô tả như sau:

```
for (i = 0; i < m; i + +)
    for (j = 0; j < n; j + +)
        I [i, j] = I [i, j] + c;
```

• Nếu c > 0: ảnh sáng lên.

• Nếu c < 0: ảnh tối đi.

Với ảnh màu ta áp dụng thuật toán trên cho từng kênh màu.

2.2 Hiệu chỉnh độ tương phản

Ý tưởng của thuật toán này là tạo sự thay đổi một cách rõ ràng giữa các điểm ảnh để sau khi thực hiện ta thu được một ảnh với các đối tượng được phân biệt rõ ràng hơn.

Phương pháp này được thực hiện bằng cách nhân giá trị mỗi điểm ảnh với một số nguyên dương

Giả sử ta có ảnh I với kích thước m × n và một số nguyên c. Khi đó thuật toán được mô tả như sau:

```
for (i = 0; i < m; i + +)
    for (j = 0; j < n; j + +)
        I [i, j] = I [i, j] × c;
```

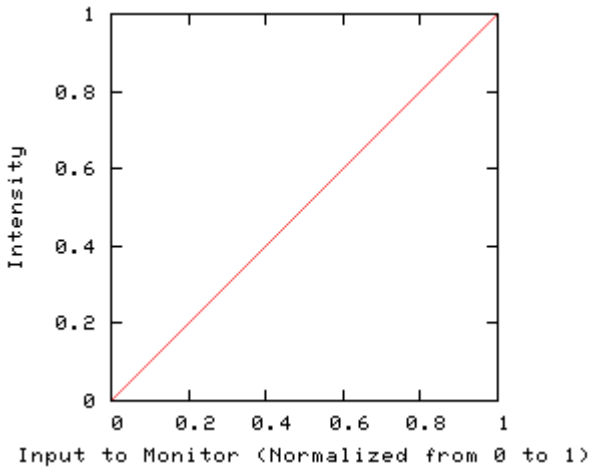
• Nếu c > 1: tăng độ tương phản.

• Nếu c < 1: giảm độ tương phản.

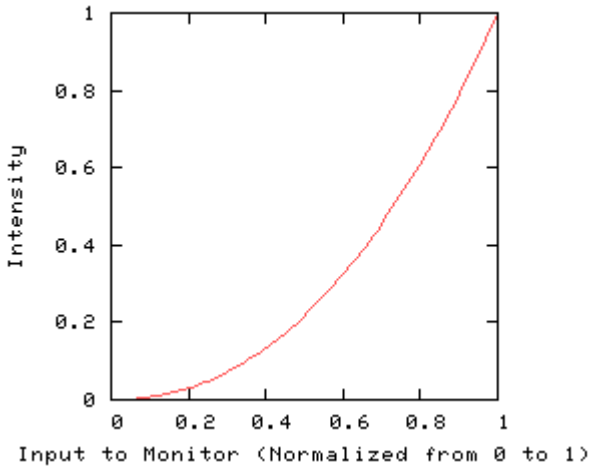
Với ảnh màu ta áp dụng thuật toán trên cho từng kênh màu.

2.3 Hiệu chỉnh gamma

Bản chất của việc hiển thị máy tính là việc đưa hình ảnh dạng dữ liệu ra màn hình (output). Tuy nhiên trong quá trình hiển thị trên màn hình, thiết bị đầu cuối thường gặp 1 vấn đề là độ nhạy sáng (Light Intensity). Hầu như các loại màn hình đều có đặc điểm chung là khi xuất kết quả đều cho ra giá trị là một hàm mũ. Tức là với x là giá trị đầu vào thì khi xuất ra màn hình sẽ là lũy thừa của x. Điều này này làm giảm chất lượng hình ảnh và hình ảnh thường tối hơn bình thường.



Hình 2.1. Giá trị đầu vào màn hình



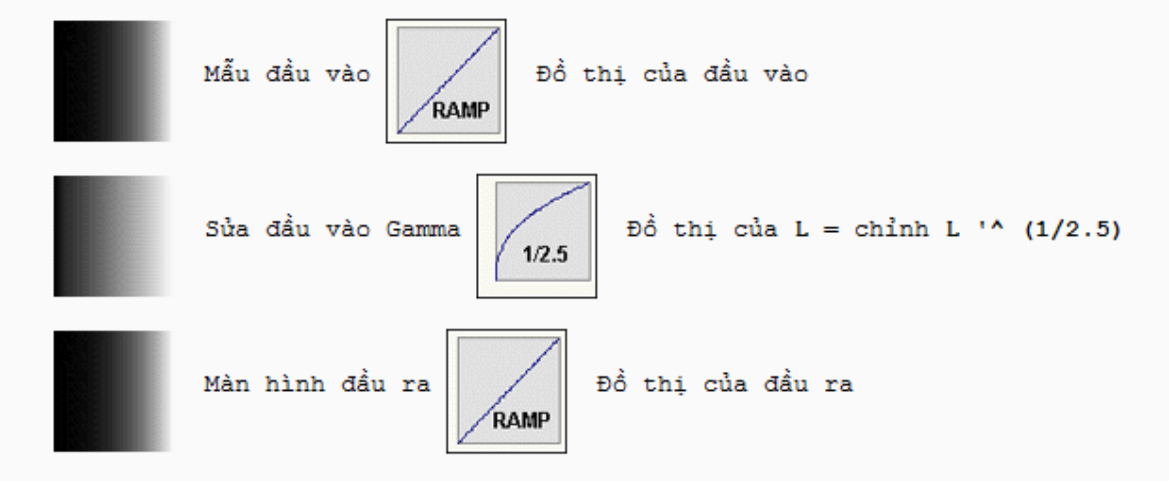
Hình 2.2. Giá trị xuất ra màn hình

Lý do bức ảnh tối hơn là vì mức điện áp của màn hình nằm trong khoảng 0 đến 1. Do đó khi thực hiện hàm mũ sẽ thu được giá trị nhỏ hơn giá trị đầu vào. Ví

dụ giá trị đầu vào là 0.5 khi thực hiện hàm mũ 2.5 sẽ thu được kết quả là 0.177 nhỏ hơn giá trị đầu vào là 0.5

Do đó, để hình ảnh có độ hiển thị trung thực, anh đầu vào sẽ được làm lũy thừa với một số mũ gọi là gamma.

Giả sử màn hình đưa ra kết quả là lũy thừa với số mũ là 2.5 thì ảnh đầu vào khi thực hiện "hiệu chỉnh gamma" sẽ được làm lũy thừa với số mũ γ = 1/2.5.



Hình 2.3. Quá trình hiệu chỉnh gamma

2.3.1 Thuật toán

Trong thuật toán trên, giá trị đầu vào của ảnh nằm trong khoảng [0, 1]. Vậy để áp dụng cho ảnh [0,255] ta áp dụng công thức 2.1:

Color' = Round(255*Pow(Color/255,gamma)+0.5) (2.1)

Ở đây Round là hàm làm tròn, Pow là hàm lũy thừa.

Cho ảnh I có kích thước m × n. Thuật toán được mô tả như sau:

```
for (i = 0; i < m; i + +)
    for (j = 0; j < n; j + +)
        I [i, j] = Round(255*Pow(I [i, j]/255,gamma)+0.5);
```

- Nếu gamma < 1: Ảnh sáng lên
- Nếu gamma > 1: Ảnh tối đi

2.3.2 Cải tiến thuật toán

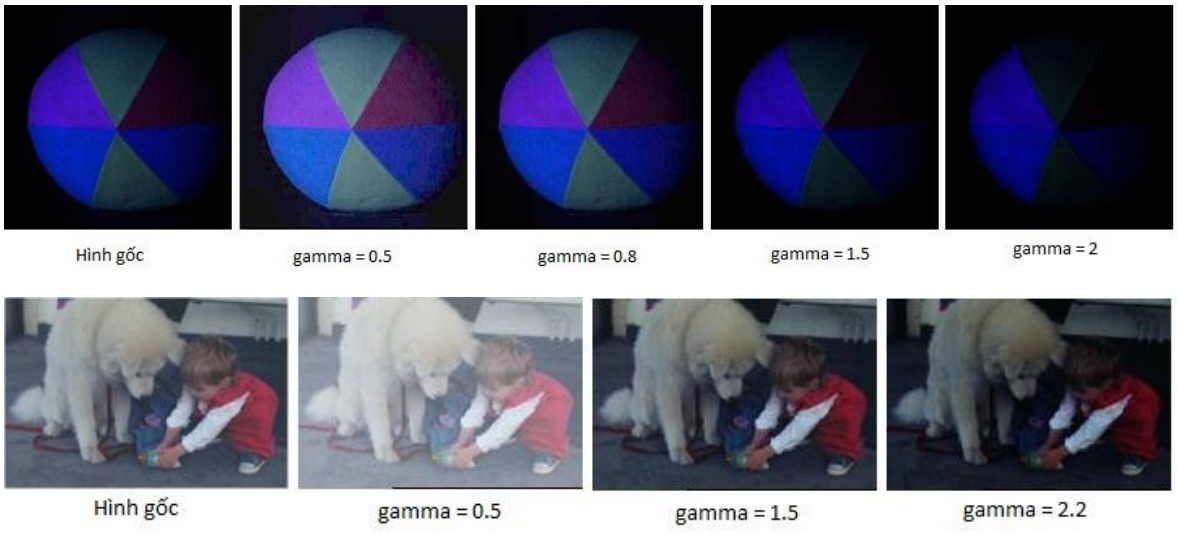
Với thuật toán trên, mỗi lần duyệt 1 điểm ảnh thì hàm lũy thừa và hàm làm tròn lại được thực hiện 1 lần như vậy sẽ làm chậm quá trình thực hiện. Để giải quyết

vấn đề này, ta sẽ tạo 1 mảng lưu sẵn giá trị tính toán cho Color trong khoảng [0,255]. Sau đó áp dụng cho từng điểm ảnh.

```
for(i=0;i<256;i++)
        Color[i] = Round(255*Pow(i/255,gamma)+0.5);
for (i = 0; i < m; i + +)
        for (j = 0; j < n; j + +)
                I [i, j] = Color[I[i, j]];
```

Đối với ảnh màu ta áp dùng thuật toán với từng kênh màu.

2.3.3 Một số kết quả ví dụ



Hình 2.4. Ví dụ về hiệu chỉnh gamma

2.4 Cân bằng màu

Thuật toán cân bằng màu nhằm mục đích để sửa chữa hình ảnh thiếu sáng, hoặc hình ảnh chụp trong ánh sáng nhân tạo hoặc ánh sáng tự nhiên đặc biệt, như hoàng hôn.

Có nhiều thuật toán phức tạp trong việc cân bằng màu sắc hoặc điều chỉnh màu tương phản. Việc thực hiện các thuật toán hiệu chỉnh màu nhiều có thể được đánh giá bằng cách so sánh kết quả của chúng với thuật toán cân bằng màu sắc đơn giản nhất đề xuất ở đây. Các giả định cơ bản thuật toán này là các giá trị cao nhất của R, G, B quan sát thấy trong hình ảnh phải tương ứng với màu trắng, và các giá trị thấp nhất tương ứng với màu tối. Nếu bức ảnh được chụp trong bóng tối, các giá

trị cao nhất có thể được nhỏ hơn đáng kể so với 255. Bằng cách kéo dài các vảy màu sắc, hình ảnh trở nên sáng sủa hơn. Nếu có một màu ánh sáng xung quanh, ví dụ cho ánh sáng điện trong đó R và G chiếm ưu thế, cân bằng màu sắc sẽ tăng cường các kênh B. Do đó, ánh sáng xung quanh sẽ mất màu vàng của nó. Mặc dù nó không nhất thiết phải cải thiện hình ảnh, cân bằng màu sắc đơn giản luôn luôn làm tăng khả năng đọc của nó.

Thuật toán đơn giản là giãn đến mức có thể, các giá trị của ba kênh *Red, Green, Blue* (R, G, B), để chúng chiếm phạm vi tối đa có thể [0, 255]. Cách đơn giản để làm như vậy là để áp dụng một biến đổi affine $ax + b$ cho mỗi kênh, tính toán a và b để các giá trị tối đa trong kênh trở thành 255 và giá trị tối thiểu trở thành 0.

Tuy nhiên, nhiều hình ảnh có chứa một vài điểm ảnh sai mà đã chiếm các giá trị 0 và 255. Như vậy, một hình ảnh đẹp thường cải thiện màu sắc thu được bằng cách *"cắt"* một phần nhỏ của các điểm ảnh với các giá trị cao nhất tới 255 và tỷ lệ phần nhỏ của các điểm ảnh với các giá trị thấp nhất là 0, trước khi áp dụng biến đổi affine. Chú ý rằng độ bão hòa này có thể tạo ra các vùng màu đen hoặc các vùng màu trắng, có thể trông không tự nhiên. Như vậy, tỷ lệ điểm ảnh bão hòa phải được càng nhỏ càng tốt.

2.4.1 Thực hiện

Hình ảnh đầu vào là một mảng của N giá trị số trong khoảng $[min, max]$. Đầu ra là một mảng điều chỉnh của N giá trị đã được thay đổi. Nhiều kênh hình ảnh được xử lý độc lập trên mỗi kênh với cùng một phương pháp.

Chúng ta sẽ thực hiện một sự cân bằng màu sắc trên dữ liệu này mà chúng ta đã bão hòa *s1(%)* điểm ảnh ở phía bên trái của biểu đồ, và *s2(%)* của điểm ảnh ở phía bên phải, sự cân bằng này sẽ bão hòa nhiều nhất là $N \times s1/100$ điểm ảnh ở đầu và $N \times s2/100$ ở phần cuối của biểu đồ. Chúng ta không thể đảm bảo *chính xác* để làm bão hòa $N \times (s1 + s2) / 100$ điểm ảnh bởi vì sự phân bố của các giá trị của điểm ảnh là rời rạc.

2.4.2 Phương pháp phân loại

Gọi V_{min} và V_{max} là các cực trị bão hòa, có thể được xem như là lượng phân phối giá trị của điểm ảnh, ví dụ như 1cm và 99cm(trong 100cm) cho độ bão hòa 2%.

Như vậy, một cách dễ dàng để tính toán V_{min} và V_{max} là để sắp xếp các giá trị điểm ảnh, và chọn cực trị bão hòa từ mảng được sắp xếp. Thuật toán này sẽ được mô tả như sau:

1. **Sắp xếp các giá trị pixel:** Các giá trị ban đầu phải được giữ để chuyển đổi hơn với các hàm affine bị chặn, vậy nên trước hết **N** điểm ảnh phải được sao chép trước khi phân loại.
2. **Chọn cực trị bão hòa** từ các điểm ảnh được sắp xếp với một mức độ bão hòa $s = s1 + s2$ trong $[0, 100]$, chúng ta muốn để làm bão hòa $N \times s/100$ điểm ảnh, vì vậy V_{min} và V_{max} được lấy từ mảng được sắp xếp tại các vị trí $N \times s1 / 100$ và $N \times (1 - s2 / 100) - 1$.
3. **Bão hòa các điểm ảnh**: theo các định nghĩa trước của V_{min} và V_{max}, số lượng điểm ảnh có giá trị thấp hơn so với V_{min} hoặc cao hơn so với V_{max} là nhiều nhất là $N \times s/100$. Các điểm ảnh (trong mảng phân loại ban đầu) được cập nhật cho V_{min} (V_{max}) nếu giá trị của chúng thấp hơn V_{min} (cao hơn so với V_{max}).
4. **Biến đổi affine** hình ảnh được thu nhỏ $[min, max]$ với một sự biến đổi của các giá trị điểm ảnh của hàm:

$$f(x) = (x - V_{min}) \times (max - min) / (V_{max} - V_{min}) + min. \qquad (2.2)$$

2.4.3 Phương pháp biểu đồ(Histogram)

Sắp xếp các giá trị N điểm ảnh đòi hỏi $O(N \log(N))$ hoạt động và một bản sao tạm thời của các N điểm ảnh. Một thực hiện hiệu quả hơn là đạt được bởi một biến thể dựa trên biểu đồ, nhanh hơn (độ phức tạp $O(N)$) và đòi hỏi ít bộ nhớ ($O(max - min)$ so với $O(N)$).

1. **Xây dựng một biểu đồ tích lũy của các giá trị pixel** Các mảng biểu đồ tích lũy có nhãn i chứa số lượng điểm ảnh có giá trị thấp hơn hoặc bằng với i.
2. **Chọn cực trị bão hòa từ biểu đồ** V_{min} là nhãn biểu đồ thấp nhất có giá trị cao hơn so với $N \times s1 / 100$, và số lượng điểm ảnh có giá trị thấp hơn so với V_{min} nhiều nhất là $N \times s1 / 100$. Nếu $s1 = 0$ sau đó V_{min} là nhãn biểu đồ thấp nhất, tức là giá trị tối thiểu pixel của hình ảnh đầu vào. V_{max} là nhãn ngay sau nhãn biểu đồ cao nhất với giá trị thấp hơn hoặc bằng $N \times (1 - S2 / 100)$, và số lượng điểm ảnh có giá trị cao hơn so với V_{max} nhiều nhất là $N \times s2 / 100$. Nếu $s2 = 0$ thì V_{max} là nhãn biểu đồ cao nhất, *tức là* tối đa giá trị pixel của hình ảnh đầu vào.

3. **Bão hòa các điểm ảnh**
4. **Biến đổi affine tương tự như cho phương pháp phân loại.**

2.4.4 Mã giả

Các bước sau đây trình bày cho hình ảnh với các giá trị pixel trong không gian số nguyên 8 bit *(min = 0, max = 255)* với chỉ một kênh màu. Xem các nhận xét sau đây cho độ chính xác cao hơn hình ảnh. Sau đây là việc thực hiện cơ bản, cải tiến có sẵn trong mã nguồn được đề xuất.

image[i] là các giá trị pixel, N là số lượng điểm ảnh, histo là một mảng của 256 số nguyên unsigned, với một kiểu dữ liệu đủ lớn để lưu trữ N , ban đầu chứa giá trị 0. Các chỉ số mảng bắt đầu từ 0.

```
 // Xây dựng histogram tích lũy
for i from 0 to N-1
  histo[image[i]] = histo[image[i]] + 1
for i from 1 to 255
  histo[i] = histo[i] + histo[i - 1]
// Tìm Vmin và Vmax
vmin := 0
while histo[vmin + 1] <= N * s1 / 100
  vmin = vmin + 1
vmax = 255 - 1
while histo[vmax - 1] > N * (1 - s2 / 100)
  vmax = vmax - 1
if vmax < 255 - 1
  vmax = vmax + 1
// Bão hòa điểm ảnh
for i from 0 to N - 1
  if image[i] < vmin
            image[i] = vmin
  if image[i] > vmax
```

```
                image[i] = vmax
        // Tính lại điểm ảnh
        for i from 0 to N-1
          image[i] = (image[i] - vmin) * 255 / (vmax - vmin)
```

2.4.5 Độ chính xác cao hơn

Đối với ảnh 16 bit, phương pháp mảng biểu đồ có thể được sử dụng, và nhu cầu mảng 65,536 (256 Mb trên một hệ thống 32 bit, 512 Mb trên một hệ thống 64 bit, được so sánh với 128 Mb sử dụng cho một hình ảnh 256 × 256). Nhưng việc xác định vmin và vmax sẽ được hưởng lợi của một phương pháp tìm kiếm nhanh hơn, như chia la□m hai đoạn.

Đối với 32 bit giá trị số nguyên pixel, kích thước biểu đồ (4.294.967.296) trở thành một vấn đề và không thể được xử lý đúng trong bộ nhớ. Chúng tôi có thể chuyển sang một quá trình gồm nhiều bước:

Xây dựng một biểu đồ với mảng chứa nhiều hơn một giá trị điểm ảnh duy nhất, như vậy mà kích thước biểu đồ là có hạn (ví dụ mảng 256 giá trị, mỗi một khoảng giá trị pixel);

- Tìm kiếm các mảng chứa vmin và vmax .
- Khởi động lại việc xây dựng biểu đồ và tìm kiếm trên một phân khu của những mảng.

Nếu một độ chính xác chính xác là không cần thiết, những cải tiến mới nhất có thể được bỏ qua.

Đối với dữ liệu dấu chấm động, giá trị điểm ảnh có thể không được sử dụng như một chỉ số mảng, mảng kết hợp và biểu đồ (chỉ dành cho hình ảnh ít) hoặc nhiều bước biểu đồ đã được sử dụng, ví dụ như làm tròn các giá trị dấu chấm động như một bước đầu tiên.

Lưu ý rằng các đề xuất mã giả cũng có thể được sử dụng cho hình ảnh với các giá trị điểm ảnh số nguyên (như sản xuất bởi các thiết bị chụp chung hình ảnh và tìm thấy trong các định dạng hình ảnh thông thường) được lưu trữ là điểm nổi dữ liệu (thường mong muốn cho chế biến hình ảnh), bằng cách chuyển đổi các điểm ảnh giá trị image[i] để tương đương với số nguyên của nó trong khi làm đầy các biểu đồ.

2.4.6 Các trường hợp đặc biệt

Nếu hình ảnh là không đổi (tất cả các điểm ảnh có cùng giá trị *v),* khi đó, theo mô tả việc thực hiện và mã giả, các biểu đồ giá trị là *0* cho các nhãn thấp hơn so với *v,* và *N* cho các nhãn cao hơn hoặc bằng *v,* và sau đó đối với bất kỳ giá trị của *s1* và *s2, V_{min} = v, V_{max} = v.*

Điều này *(V_{min} = V_{max})* cũng có thể xảy ra cho hình ảnh không tương phản, thông thường cho ra hình ảnh với ít hơn *N × s1 / 100* điểm ảnh với các giá trị nhỏ hơn hoặc với nhiều hơn *N × s2 / 100* lớn hơn giá trị trung bình *v*.

Trường hợp đó phải được xử lý bằng cách thiết lập tất cả các điểm ảnh về giá trị *v*.

2.4.7 Ảnh màu

Trong trường hợp hình ảnh màu RGB chúng ta có thể áp dụng các thuật toán độc lập trên mỗi kênh, hoặc áp dụng nó cho cường độ mức xám *(I)* của hình ảnh và sửa đổi các kênh màu tương ứng, chẳng hạn là tỷ lệ R / G / B ban đầu là không đổi.

Trong trường hợp sau, có thể là áp đặt một tỷ lệ tối đa của các điểm ảnh bão hòa *s1(%)* đến *s2(%)* trái và bên phải của biểu đồ màu xám mức độ có thể cho ra một tỷ lệ phần trăm bão hòa cao hơn các điểm ảnh trên một số các kênh màu. Để đảm bảo rằng không có nhiều hơn *s1(%)* điểm ảnh sẽ được bão hòa *min,* cũng không có nhiều *s2(%)* điểm ảnh sẽ được bão hòa *max* trong không ai trong số các kênh, các thuật toán lặp đi lặp lại sau đây được đề xuất (chúng ta xem xét trường hợp của hình ảnh màu 8-bit):

1. Xây dựng các biểu đồ tích lũy của *R, G, B* và *I.*
2. Thiết lập V_{max} = *max* và tìm V_{min} , mức thấp nhất của nhãn biểu đồ *xám (I)* có giá trị cao hơn so với *N × s1 / 100.*
3. Tính toán các giá trị mức xám mới *(I_{out}):* bão hòa các giá trị nhỏ hơn V_{min} hoặc lớn hơn V_{max} và áp dụng một biến đổi affine với phần còn lại của giá trị:

$$I_{out} = (I - V_{min}) \times (max - min) / (V_{max} - V_{min}) + min.$$

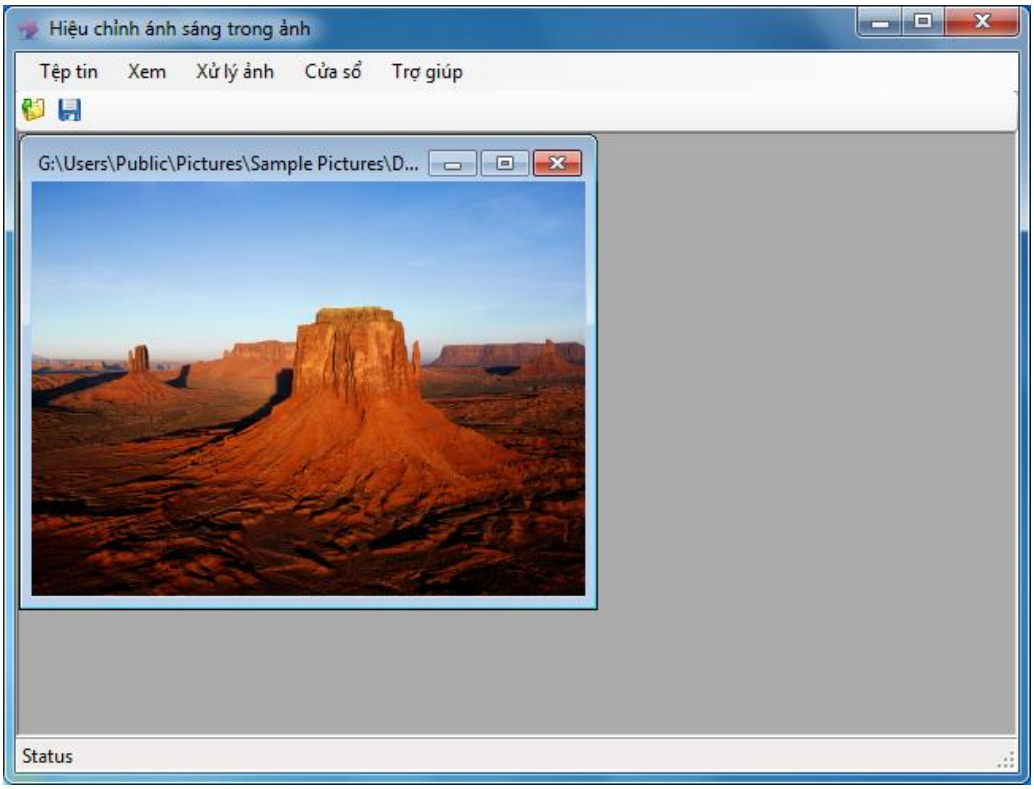
4. Tính toán giá trị mới của các kênh màu: $R_{out} = (I_{out} / I) \times R$, $G_{out} = (I_{out} / I) \times G$, $B_{out} = (I_{out} / I) \times B$
5. Nếu, một số các kênh màu mới, tỷ lệ điểm ảnh bão hòa *min* là lớn hơn *s1(%),* giảm V_{min} *(V_{min} = V_{min} - 1)* và trở về bước 3.

6. Giữ giá trị của V_{min} tìm thấy trong bước trước và tìm $V_{max,}$ nhãn ngay sau mức cao nhất của nhãn biểu đồ *xám (I)* với giá trị thấp hơn hoặc bằng N × (1 - s2 / 100).
7. Tính toán giá trị cấp độ mới màu xám (I_{out}): tương tự như bước 3.
8. Tính toán giá trị mới của các kênh màu: tương tự như bước 4.
9. Nếu, một số các kênh màu mới, tỷ lệ điểm ảnh bão hòa để *max* là lớn hơn *s2(%),* tăng V_{max} $(V_{max} = V_{max} +1)$ và quay trở lại bước 7.

CHƯƠNG 3: CHƯƠNG TRÌNH THỬ NGHIỆM

3.1 Giới thiệu chương trình

Chương trình "hiệu chỉnh ánh sáng trong ảnh" sử dụng các thuật toán nhằm minh họa cho các thuật toán được trình bày trong đồ án. Chương trình được cài đặt bằng ngôn ngữ VB.NET và chạy trên môi trường Windows.



Hình 3.1. Giao diện chính của chương trình.

Chương trình bao gồm 1 cửa sổ chính và các cửa sổ con hiển thị ảnh.

- Cửa sổ chính có chứa menu và thanh công cụ.
- Cửa sổ con hiển thị hình ảnh được mở hoặc ảnh sau khi đã sử lý.

Bên cạnh đó còn một số cửa sổ phụ phục vụ cho việc nhập các tham số cho các thuật toán.

3.2 Các chức năng của chương trình

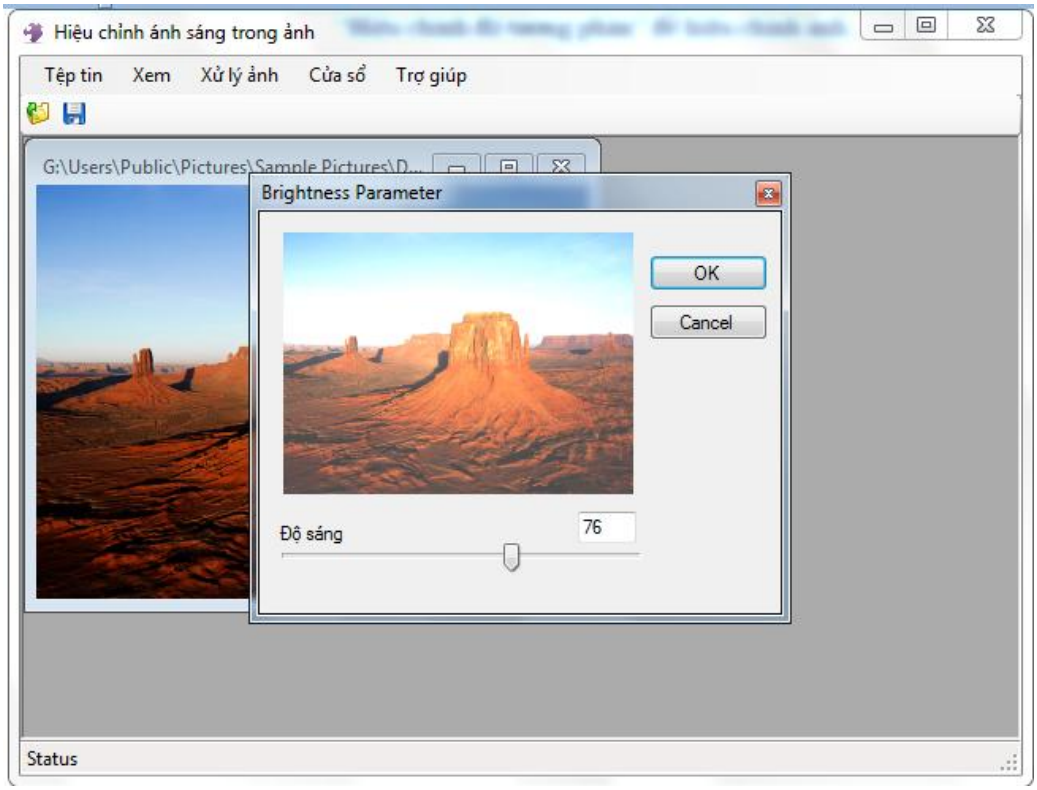
- Nhóm chức năng "**Tệp tin**":
  - Chức năng "**Mở**": Mở file để xử lý.
  - Chức năng "**Đóng**": Đóng file hiện hành.
  - Chức năng "**Thoát**": Thoát khỏi chương trình.

- Chức năng "**Lưu**": Lưu file.

- Nhóm chức năng "**Xử lý ảnh**":
  - Chức năng "**Hiệu chỉnh ánh sáng**": Áp dụng thuật toán "Hiệu chỉnh ánh sáng" cho ảnh.
  - Chức năng "**Hiệu chỉnh độ tương phản**": Áp dụng thuật toán "Hiệu chỉnh độ tương phản" để hiệu chỉnh ảnh.
  - Chức năng "**Hiệu chỉnh gamma**": Áp dụng thuật toán "Hiệu chỉnh gamma" để hiệu chỉnh ảnh.
  - Chức năng "**Cân bằng màu**": Áp dụng thuật toán "Cân bằng màu" để hiệu chỉnh ảnh.
- Nhóm chức năng "**Cửa sổ**": Sắp xếp các cửa sổ hiển thị ảnh và chuyển đổi giữa các cửa sổ.
- Chức năng "**Trợ giúp**": Thông tin về chương trình.

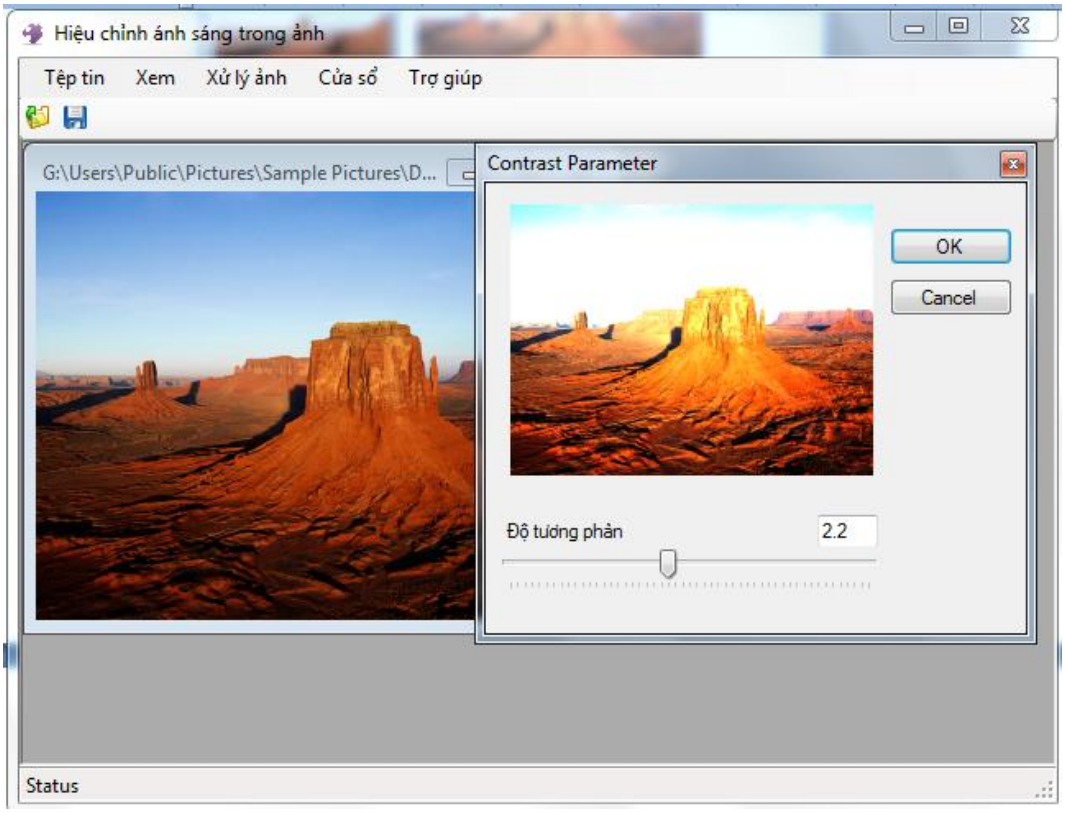
3.3 Ví dụ về nhóm chức năng "Xử lý ảnh"

3.3.1 Chức năng "Hiệu chỉnh ánh sáng"



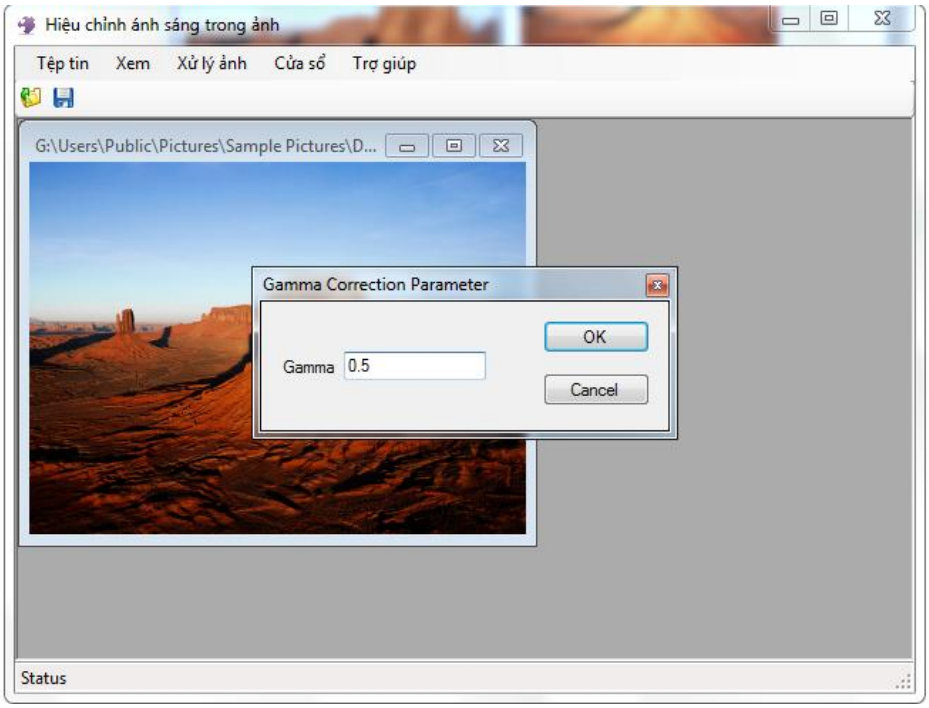
Hình 3.2. Ví dụ về chứ năng "Hiệu chỉnh ánh sáng" với tham số là 76

3.3.2 Chức năng "Hiệu chỉnh độ tương phản"

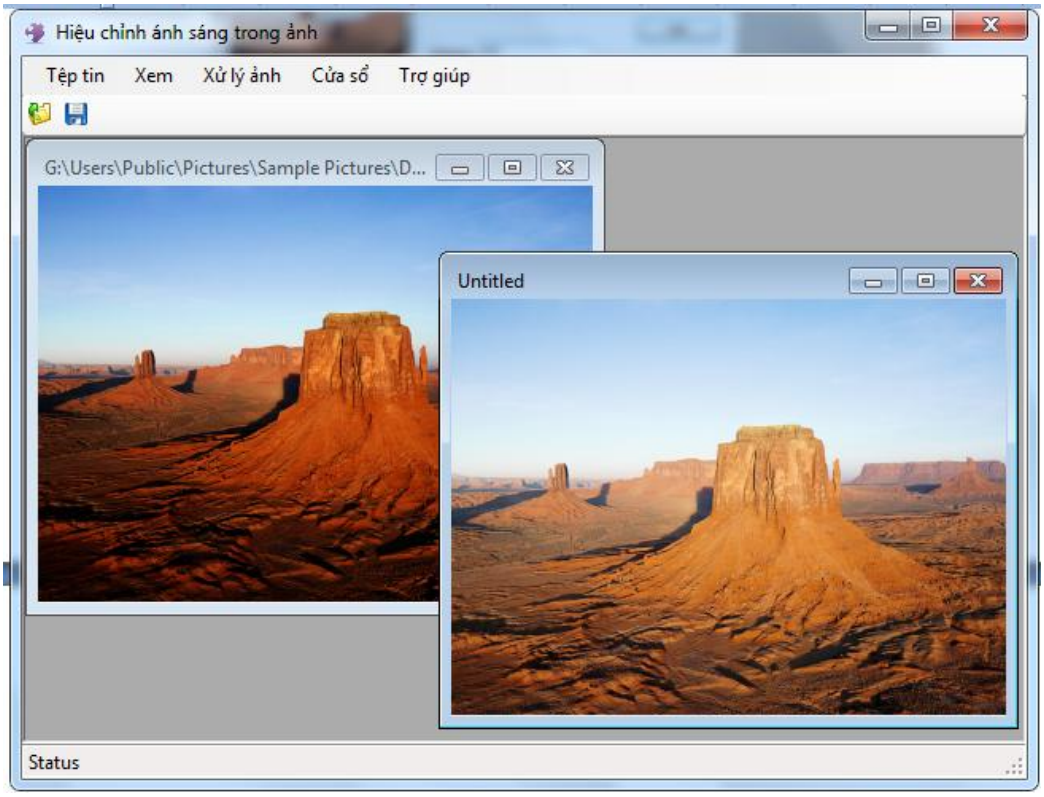


Hình 3.3. Ví dụ chức năng "Hiệu chỉnh độ tương phản" với tham số là 2.2

3.3.3 Chức năng "Hiệu chỉnh gamma"

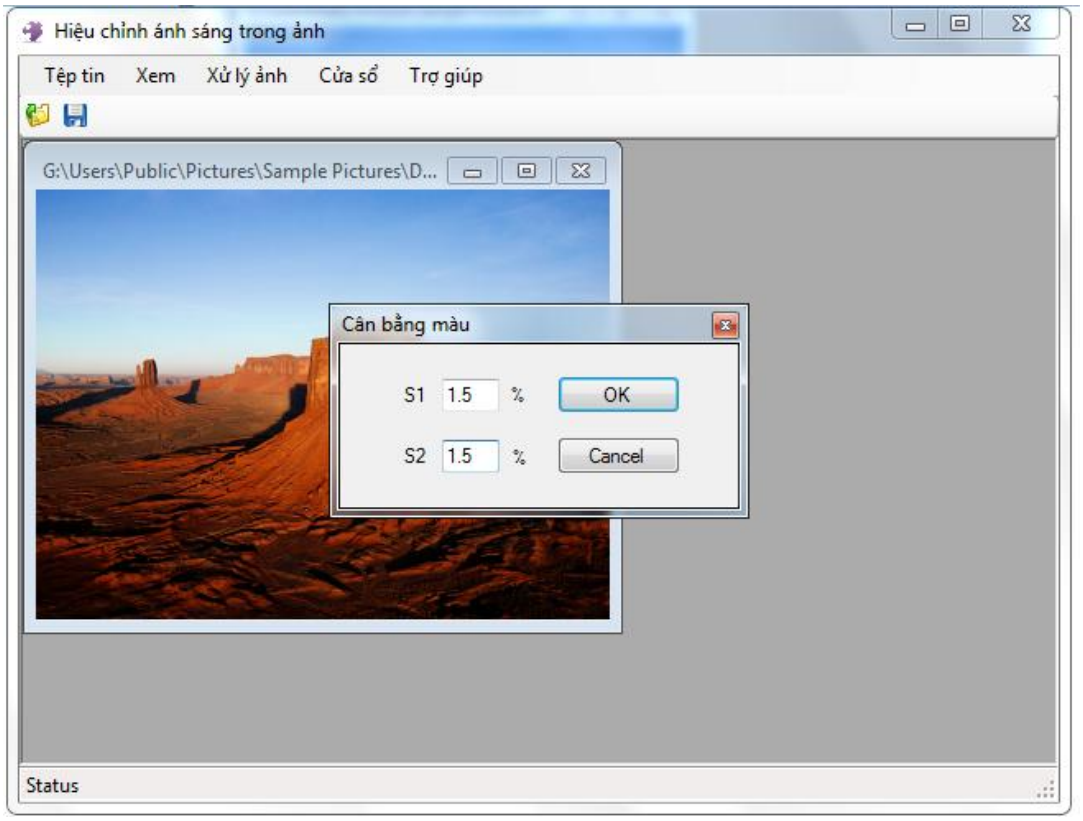


Hình 3.4. Nhập tham số cho chức năng

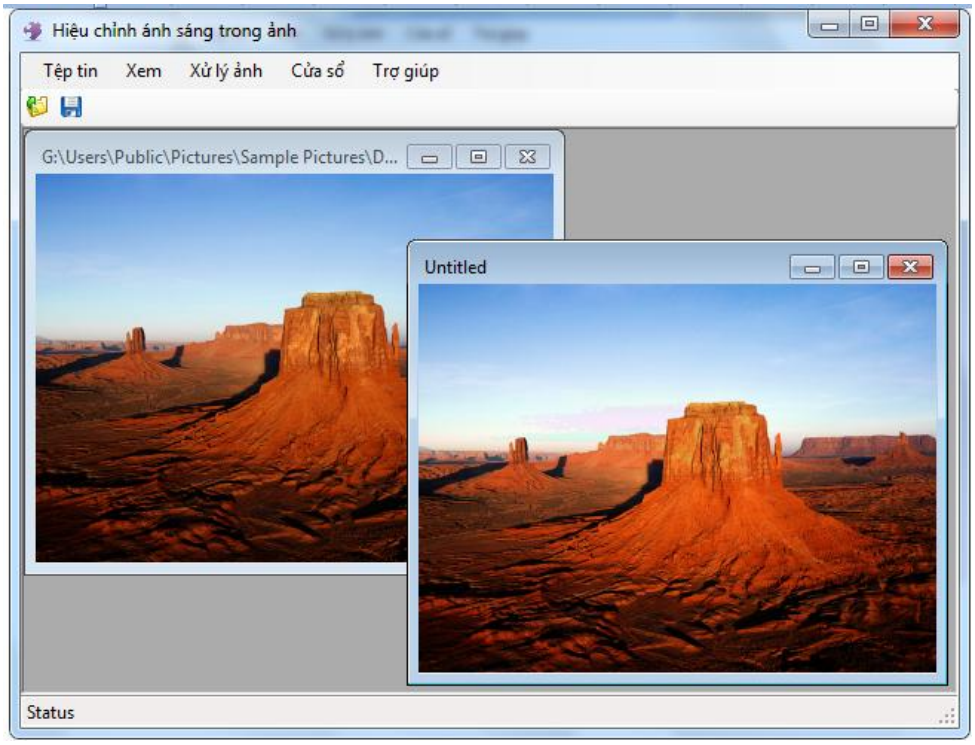


Hình 3.5. Và thu được ảnh kết quả

3.3.4 Chức năng "Cân bằng màu"



Hình 3.6. Nhập tham số đầu vào



Hình 3.7. Và kết quả thu được.

KẾT LUẬN

Ngày nay, hình ảnh có thể coi là một phương tiện truyền thông hết sức hiệu quả vì hình ảnh là ngôn ngữ hết sức trực quan và sinh động giúp việc truyền tải thông tin dễ dàng hơn, hiệu quả hơn. Nhưng để có được sự hiệu quả đó thì hình ảnh phải có bố cục và màu sắc phù hợp để có thể thỏa mãn được người xem. Do đó vấn đề xử lý ảnh nói chung, hiệu chỉnh màu sắc và ánh sáng của ảnh nói riêng là hết sức có ý nghĩa.

Hiệu chỉnh màu sắc và ánh sáng của ảnh là một phần trong chuỗi xử lý ảnh. Nó không những đem lại các kết quả phù hợp với yêu cầu của người dùng mà còn là bước tiền xử lý cho các quá trình xử lý sau của quá trình xử lý ảnh.

Trong đồ án tốt nghiệp này em đã tìm hiểu được một số vấn đề sau:

- Khái quát về xử lý ảnh.
- Một số vấn đề trong xử lý ánh sáng.
- Một số kĩ thuật hiệu chỉnh ánh sáng.
- Cài đặt được chương trình sử dụng các thuật đã nêu trong phần nội dung đồ án

Do hạn chế về mặt thời gian nên đồ án chỉ tìm hiểu được một số ít phương pháp hiệu chỉnh ánh sáng và màu sắc. Do đó hướng phát triển đề tài là còn rất lớn.

TÀI LIỆU THAM KHẢO

Tài liệu Tiếng Việt

[1]. Đỗ Năng Toàn, Phạm Việt Bình , *Giáo trình xử lý ảnh*.

Tài liệu Tiếng Anh

[2]. Ana Belén Petro, Licolas Limare, Jean-Michel Morel , Catalina Sbert, *Simplest Color Balance.*

[3]. Computer Graphics Systems Development Corporaton, *CGSD – Gamma Correction Home Page*.

[4]. Lawrence(*2003*), *Gamma Correction in Computer Graphic.*

PHỤ LỤC

Ảnh thu được sau quá trình số hoá có nhiều loại khác nhau, phụ thuộc vào kỹ thuật số hoá ảnh. Ảnh được chia thành 2 loại: ảnh đen trắng và ảnh màu.

Ảnh thu được có thể lưu trữ trên tệp để phục vụ cho các bước xử lý tiếp theo.

Dưới đây sẽ trình bày một số định dạng ảnh thông dụng hay dùng trong quá trình xử lý ảnh hiện nay.

1. Định dạng ảnh IMG

Ảnh IMG là ảnh đen trắng, phần đầu của ảnh IMG có 16 byte chứa các thông tin cần thiết sau:

+ 6 byte đầu: dùng để đánh dấu định dạng ảnh IMG. Giá trị của 6 byte này viết dưới dạng Hexa: 0x0001 0x0008 0x0001.

+ 2 byte tiếp theo: chứa độ dài mẫu tin. Đó là độ dài của dãy các byte kề liền nhau mà dãy này sẽ được lặp lại một số lần nào đó. Số lần lặp này sẽ được lưu trong byte đếm. Nhiều dãy giống nhau được lưu trong một byte.

+ 4 byte tiếp: mô tả kích cỡ pixel.

+ 2 byte tiếp: số pixel trên một dòng ảnh.

+ 2 byte cuối: số dòng ảnh trong ảnh.

Ảnh IMG được nén theo từng dòng. Mỗi dòng bao gồm các gói (pack).

Các dòng giống nhau cũng được nén thành một gói. Có 4 loại gói sau:

- Loại 1: *Gói các dòng giống nhau.*

Quy cách gói tin này như sau: 0x00 0x00 0xFF Count. Ba byte đầu tiên cho biết số các dãy giống nhau, byte cuối cho biết số các dòng giống nhau.

- Loại 2: *Gói các dãy giống nhau.*

Quy cách gói tin này như sau: 0x00 Count. Byte thứ hai cho biết số các dãy giống nhau được nén trong gói. Độ dài của dãy ghi ở đầu tệp.

- Loại 3: *Dãy các Pixel không giống nhau, không lặp lại và không nén được.*

Quy cách gói tin này như sau: 0x80 Count. Byte thứ hai cho biết độ dài dãy các pixel không giống nhau không nén được.

- Loại 4: *Dãy các Pixel giống nhau.*

Tuỳ theo các bít cao của byte đầu tiên được bật hay tắt. Nếu bít cao được bật (giá trị 1) thì đây là gói nén các byte chỉ gồm bít 0, số các byte được nén được tính bởi 7 bít thấp còn lại. Nếu bít cao tắt (giá trị 0) thì đây là gói nén các byte gồm toàn bít 1. Số các byte được nén được tính bởi 7 bít thấp còn lại.

Các gói tin của file IMG phong phú như vậy là do ảnh IMG là ảnh đen trắng, do vậy chỉ cần 1 bít cho 1 pixel thay vì 4 hoặc 8 như đã nói ở trên. Toàn bộ ảnh chỉ có những điểm sáng và tối tương ứng với giá trị 1 hoặc giá trị 0. Tỷ lệ nén của kiểu định dạng này là khá cao.

2. Định dạng ảnh PCX

Định dạng ảnh PCX là một trong những định dạng ảnh cổ điển nhất. Nó sử dụng phương pháp mã hoá loạt dài RLE (Run – Length – Encoded) để nén dữ liệu ảnh. Quá trình nén và giải nén được thực hiện trên từng dòng ảnh. Thực tế, phương pháp giải nén PCX kém hiệu quả hơn so với kiểu IMG. Tệp PCX gồm 3 phần: đầu tệp (header), dữ liệu ảnh (image data) và bảng màu mở rộng. Header của tệp PCX có kích thước cố định gồm 128 byte và được phân bố như sau: + 1 byte: chỉ ra kiểu định dạng. Nếu là kiểu PCX/PCC thì nó luôn có giá trị là 0Ah.

+ 1 byte: chỉ ra version sử dụng để nén ảnh, có thể có các giá trị sau:

- 0: version 2.5.

- 2: version 2.8 với bảng màu.

- 3: version 2.8 hay 3.0 không có bảng màu.

- 5: version 3.0 có bảng màu.

+ 1 byte: chỉ ra phương pháp mã hoá. Nếu là 0 thì mã hoá theo phương pháp BYTE PACKED, ngược lại là phương pháp RLE.

+ 1 byte: số bít cho một điểm ảnh plane.

+ 1 word: toạ độ góc trái trên của ảnh. Với kiểu PCX nó có giá trị là (0,0), còn PCC thì khác (0,0).

+ 1 word: toạ độ góc phải dưới.

+ 1 word: kích thước bề rộng và bề cao của ảnh.

+ 1 word: số điểm ảnh.

+ 1 word: độ phân giải màn hình.

+ 1 word.

+ 48 byte: chia nó thành 16 nhóm, mỗi nhóm 3 byte. Mỗi nhóm này chứa thông tin về một thanh ghi màu. Như vậy ta có 16 thanh ghi màu.

+ 1 byte: không dùng đến và luôn đặt là 0.

+ 1 byte: số bit plane mà ảnh sử dụng. Với ảnh 16 màu, giá trị này là 4, với ảnh 256 màu (1pixel/8bit) thì số bit plane lại là 1.

+ 1 byte: số bytes cho một dòng quét ảnh.

+ 1 word: kiểu bảng màu.

+ 58 byte: không dùng.

Tóm lại, định dạng ảnh PCX thường được dùng để lưu trữ ảnh vì thao tác đơn giản, cho phép nén và giải nén nhanh. Tuy nhiên, vì cấu trúc của nó cố định, nên trong một số trường hợp nó làm tăng kích thước lưu trữ. Và cũng vì nhược điểm này mà một số ứng dụng lại sử dụng một kiểu định dạng khác mềm dẻo hơn: định dạng TIFF (Targed Image File Format) sẽ mô tả dưới đây.

3. Định dạng ảnh TIFF

Kiểu định dạng TIFF được thiết kế để làm nhẹ bớt các vấn đề liên quan đến việc mở rộng tệp ảnh cố định. Về cấu trúc, nó cũng gồm 3 phần chính:

- Phần Header (IFH)

Có trong tất cả các tệp TIFF và gồm 8 byte:

+ 1 word: chỉ ra kiểu tạo tệp trên máy tính PC hay máy Macintosh. Hai loại này khác nhau rất lớn ở thứ tự các byte lưu trữ trong các số dài 2 hay 4 byte. Nếu trường này có giá trị là 4D4Dh thì đó là ảnh cho máy Macintosh. Nếu trường này có giá trị là 4949h thì đó là ảnh của máy PC.

+ 1 word: version. Từ này luôn có giá trị là 42. Có thể coi đó là đặc trưng của file TIFF vì nó không thay đổi.

+ 2 word: giá trị Offset theo byte tính từ đầu file tới cấu trúc IFD (Image File Directory) là cấu trúc thứ hai của file. Thứ tự các byte ở đây phụ thuộc vào dấu hiệu trường đầu tiên.

- Phần thứ 2 (IFD)

Nó không ở ngay sau cấu trúc IFH mà vị trí của nó được xác định bởi trường Offset trong đầu tệp. có thể có một hay nhiều IFD cùng tồn tại trong file (nếu file có nhiều hơn 1 ảnh).

Một IFD bao gồm:

+ 2 byte: chứa các DE (Directory Entry).

+ 12 byte là các DE xếp liên tiếp. Mỗi DE chiếm 12 byte.

+ 4 byte: chứa Offset trỏ tới IFD tiếp theo. Nếu đây là IFD cuối cùng thì trường này có giá trị 0.

- Phần thứ 3

Các DE. Các DE có độ dài cố định gồm 12 byte và chia làm 4 phần:

+ 2 byte: chỉ ra dấu hiệu mà tệp ảnh đã được xây dựng.

+ 2 byte: kiểu dữ liệu của tham số ảnh. Có 5 kiểu tham số cơ bản:

- 1: BYTE (1 byte)

- 2: ASCII (1 byte)

- 3: SHORT (2 byte).

- 4: LONG (4 byte)

- 5: RATIONAL (8 byte)

+ 4 byte: trường độ dài (bộ đếm) chứa số lượng chỉ mục của kiểu dữ liệu đã chỉ ra. Nó không phải là tổng số byte cần thiết để lưu trữ. Để có số liệu này ta cần nhân số chỉ mục với kiểu dữ liệu đã dùng.

+ 4 byte: đó là Offset tới điểm bắt đầu dữ liệu thực liên quan tới dấu hiệu, tức là dữ liệu liên quan với DE không phải lưu trữ vật lý cùng với nó nằm ở một ví trí nào đó trong file.

Dữ liệu chứa trong tệp thường được tổ chức thành các nhóm dòng (cột) quét của dữ liệu ảnh. Cách tổ chức này làm giảm bộ nhớ cần thiết cho việc đọc tệp. Việc giải nén được thực hiện theo bốn kiểu khác nhau được lưu trữ trong byte dấu hiệu nén.

File ảnh TIFF dùng để giải quyết vấn đề khó mở rộng của file PCX. Tuy nhiên, với cùng một ảnh thì việc dùng file PCX chiếm ít không gian nhớ hơn.