BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

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

XÂY DỰNG CHƯƠNG TRÌNH XÁC THỰC ẢNH SỐ

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: Công nghệ Thông tin

Giáo viên h- ớng dẫn: Th.s Phùng Anh Tuấn

Sinh viên thực hiện: Nguyễn Thị Ngọc

Mã số sinh viên: 121186

HẢI PHÒNG 7/ 2012

LỜI CẢM ƠN

Em xin chân thành cảm ơn thầy giáo Thạc sỹ Phùng Anh Tuấn - giảng viên khoa CNTT - Trường ĐHDL Hải Phòng, người đã trực tiếp hướng dẫn tận tình và tạo mọi điều kiện thuận lợi để em hoàn thành đồ án của mình.

Em cũng xin gửi lời cảm ơn chân thành tới tất cả các thầy cô trong bộ môn Công Nghệ Thông Tin - Trường ĐHDL Hải Phòng cũng như các thầy cô trong trường đã nhiệt tình chỉ dạy và cung cấp những kiến thức quý báu để em có thể hoàn thành tốt đồ án tốt nghiệp này.

Đồng thời em cũng xin cảm ơn tất cả các anh chị trong Văn phòng thành ủy Hải Phòng đã tạo mọi điều kiện tốt nhất cho em trong suốt thời gian làm tốt nghiệp.

Cuối cùng, em xin cảm ơn gia đình và bạn bè luôn tạo điều kiện, động viên và giúp đỡ em trong suốt thời gian học tập, cũng như quá trình nghiên cứu, hoàn thành đồ án này.

Vì thời gian có hạn, kiến thức của bản thân còn nhiều hạn chế cho nên trong đồ án không tránh khỏi những thiếu sót, em rất mong nhận được sự đóng góp ý kiến của tất cả các thầy cô giáo cũng như các bạn để đồ án của em được hoàn thiện hơn.

Em xin chân thành cảm ơn!

Hải Phòng, ngày 06 tháng 07 năm 2012

Sinh viên

Nguyễn Thị Ngọc

Mục lục

DANH MỤC CÁC HÌNH

LỜI MỞ ĐẦU

Có một câu nói nổi tiếng là một hình ảnh trị giá bằng một ngàn từ. Ảnh hưởng của những thông tin từ những bức ảnh là rất lớn, có tác động mạnh mẽ và trực tiếp tới con người. Do vậy ảnh được coi là công cụ biểu diễn và truyền đạt thông tin rất phổ biến và hữu dụng. Với các công nghệ kỹ thuật số hiện đại và sự phổ biến của các phần mềm chỉnh sửa hình ảnh làm cho việc thao tác với ảnh số rất dễ dàng. Kết quả là, có sự tăng nhanh chóng số lượng ảnh số giả mạo trên các phương tiện truyền thông và trên mạng Internet.

Ảnh giả mạo được xem là ảnh không có thật, việc có được ảnh là do sự ngụy tạo bởi các chương trình xử lý ảnh hoặc quá trình thu nhận. Giả mạo ảnh nhằm vào nhiều mục đích trong đó có việc vu cáo, tạo ra các tin giật gân, đánh lừa đối thủ, làm sai lệch chứng cứ phạm tội... Xu hướng này chỉ ra lỗ hổng bảo mật nghiêm trọng và làm giảm độ tin cậy của các hình ảnh kỹ thuật số. Do vậy, kỹ thuật xác minh tính toàn vẹn và tính xác thực của ảnh số đã trở nên rất quan trọng, đặc biệt là khi sử dụng các hình ảnh để làm bằng chứng trong pháp luật, cũng như các tin tức, hay những dữ liệu trong hồ sơ y tế, hoặc tài liệu tài chính. Vì thế xác thực ảnh hay nói cách khác là chứng minh ảnh đó là giả hay thật là vấn đề phải đặt ra ngày càng cấp bách và càng trở nên khó khăn. Việc phát hiện và chống giả mạo ảnh là một chủ đề ngày càng được quan tâm bởi nhiều nhóm nghiên cứu trên thế giới và trong nước.

Lĩnh vực nghiên cứu này có nhiều tiềm năng phát triển trong tương lai gần và dần trở thành một hướng đi mới trong lĩnh vực bảo đảm an toàn thông tin rất hiệu quả. Vì vậy, em đã chọn đề tài ”Xây dựng chương trình xác thực ảnh số ” làm đồ án tốt nghiệp của mình. Nội dung đồ án gồm 3 chương:

- Chương 1 : Trình bày tổng quan về xử lý ảnh và các dạng ảnh giả mạo cơ bản.
- Chương 2: Trình bày các kĩ thuật xác thực ảnh số.
- Chương 3: Xây dựng chương trình thử nghiệm.

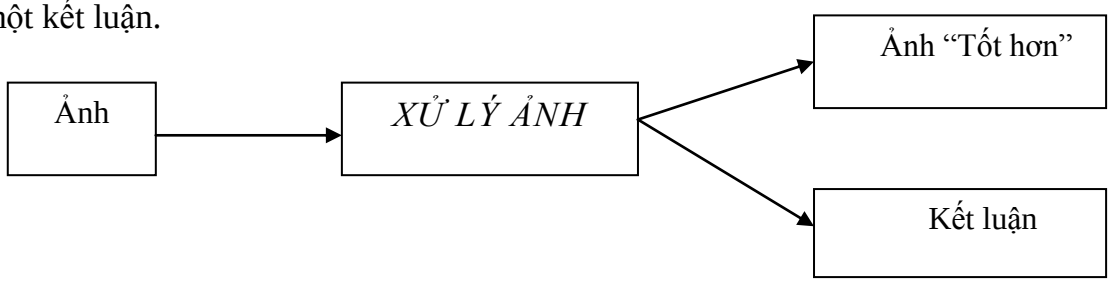
Cuối cùng là phần kết luận và đề xuất hướng nghiên cứu trong tương lai.

Chương 1 . TỔNG QUAN VỀ XỬ LÝ ẢNH VÀ GIẢ MẠO ẢNH

1.1 Xử lý ảnh, các vấn đề cơ bản trong xử lý

1.1.1 Xử lý ảnh là gì?

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: Quá trình xử lý ảnh

1.1.2 Định nghĩa ảnh số (Digital Image)

- Điểm ảnh (Pixel) là một phần tử của ảnh số tại toạ độ (x, y) với độ xám hoặc màu nhất định.
- Mức xám của điểm ảnh là cường độ sáng của nó được gán bằng giá trị số tại điểm đó.
- Ảnh số là tập hợp các điểm ảnh với mức xám phù hợp dùng để mô tả ảnh gần với ảnh thật.
- Phân loại ảnh số:

❖ ***Ảnh xám / ảnh đen trắng (Gray Image)***

Giá trị mỗi điểm ảnh nằm trong dải từ 0 đến 255, nghĩa là cần 8 bits hay 1 byte để biểu diễn mỗi điểm ảnh này.

❖ ***Ảnh nhị phân (Binary Image)***

Giá trị mỗi điểm ảnh là 0 hoặc 1 nghĩa là trắng hoặc đen. Mức 0 ứng với màu sáng, còn mức 1 ứng với màu tối. Trong thực tế khi xử lý trên máy tính thì người ta dùng ảnh xám để biểu diễn ảnh nhị phân.

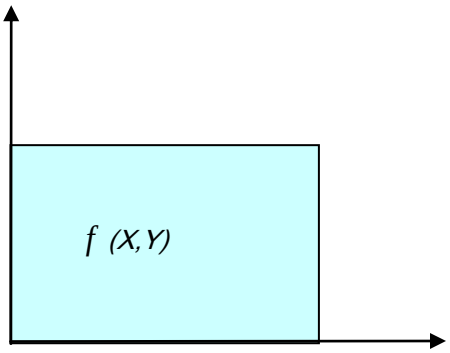
❖ ***Ảnh màu (Color Image)***

Mỗi điểm ảnh có giá trị gồm 3 màu đỏ (R), xanh lục (G) và xanh dương (B), mỗi màu có giá trị từ 0 đến 255, nghĩa là mỗi điểm ảnh cần 24 bits hay 3 bytes để biểu diễn.

1.1.3 Các vấn đề cơ bản trong xử lý ảnh

1.1.3.1 Biểu diễn ảnh

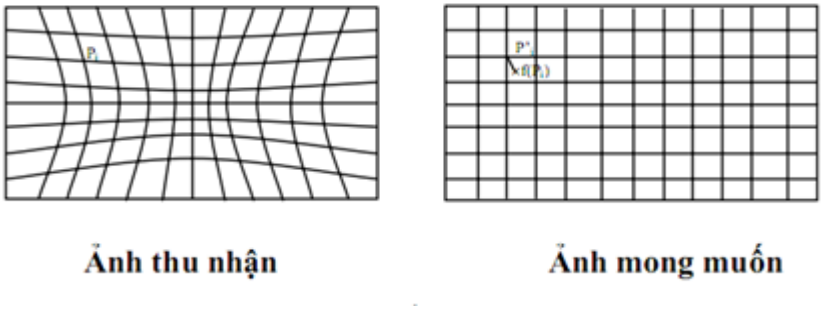
Đối với ảnh đơn giản (ảnh đen trắng) thì ảnh được biểu diễn bằng một hàm cường độ sáng hai chiều $f(x,y)$, trong đó x,y là các giá trị toạ độ không gian và hàm giá trị của f tại một điểm (X,Y) bất kỳ sẽ tỷ lệ với độ sáng hay mức xám của điểm ảnh tại điểm này.



Hình 2: Biểu diễn ảnh bằng hàm $f(X,Y)$

1.1.3.2 Nắn chỉnh biến dạng

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



Hình 3: 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.

1.1.3.3 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 → khắc phục bằng các phép lọc.

1.1.3.4 Nhận dạng ảnh

Nhận dạng ảnh là một quá trình phân hoạch ảnh thành các đối tượng ảnh con, chúng được gán vào từng lớp nhãn để được đối sánh với mẫu và đối sánh theo các quy luật biết trước nào đó.

1.2 Ảnh giả mạo và các dạng giả mạo ảnh cơ bản

1.2.1 Ảnh giả mạo

Ảnh giả mạo được xem là ảnh không có thật, việc có được ảnh là do sự ngụy tạo bởi các chương trình xử lý ảnh hoặc quá trình thu nhận. Giả mạo ảnh nhằm vào nhiều mục đích trong đó có việc vu cáo, tạo ra các tin giật gân, đánh lừa đối thủ, làm sai lệch chứng cứ phạm tội v.v…

Ảnh giả mạo được chia làm hai loại:

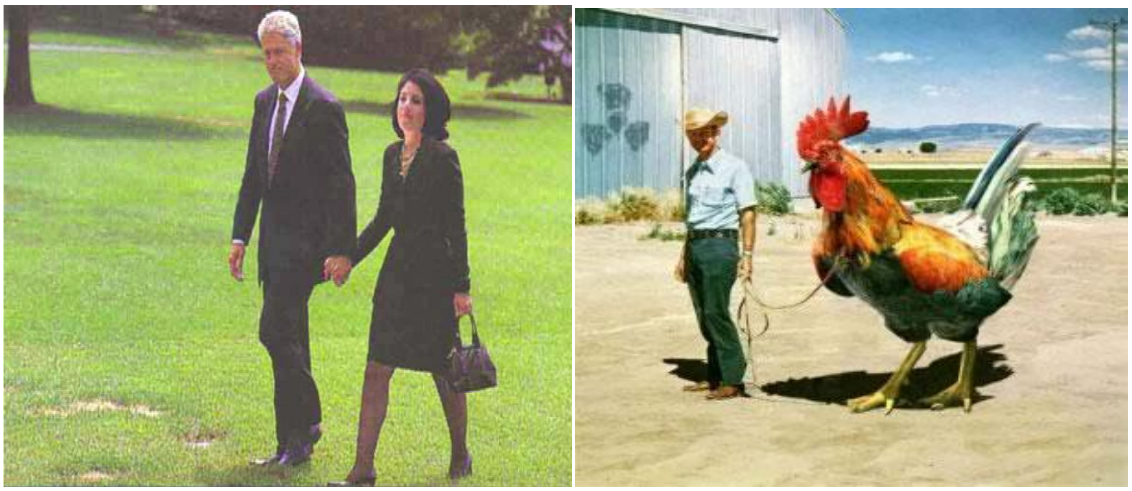
- Thứ nhất, đó là ảnh giả mạo nhưng thật, được dàn dựng một cách có ý đồ sau đó thu nhận ảnh và không thực hiện thao tác chỉnh sửa trực tiếp trên ảnh thu nhận được.
- Thứ hai, ảnh giả mạo được tạo ra từ việc có tác động lên ảnh nhằm thay đổi nội dung và bản chất bức ảnh dựa trên các kỹ thuật xử lý ảnh (cắt, dán, ghép, thêm, bớt, chỉnh sửa).

Trong đề tài nghiên cứu này chỉ quan tâm xác định những bức ảnh giả mạo thuộc loại thứ hai.

1.2.2 Các loại ảnh giả mạo cơ bản

1.2.2.1 Ghép ảnh

Ghép ảnh là dạng giả mạo ảnh số phổ biến nhất. Một ví dụ về ghép ảnh là hình số 4. Hình 4a được ghép từ hai ảnh có cùng tỷ lệ. Rõ ràng là nếu xác định được đây là ảnh thật hay ảnh giả mạo thì cũng chứng minh được mối quan hệ giữa họ. Độ tin cậy của sự giả mạo phụ thuộc vào mức độ phù hợp các thành phần của ảnh về mặt kích thước, tư thế, màu sắc, chất lượng và ánh sáng. Nếu có một cặp ảnh tương thích tốt, được thực hiện bởi một chuyên gia giàu kinh nghiệm thì việc kết hợp hoàn toàn như thật.



a) Ảnh ghép từ hai ảnh riêng rẽ *b) Ảnh ghép từ hai ảnh có thay đổi tỷ lệ*

Hình 4: Ghép ảnh từ hai ảnh riêng rẽ

Một ví dụ khác của dạng giả mạo này là hình 4b. Hình này là ảnh ghép từ hai ảnh có sự thay đổi tỷ lệ. Nếu ảnh này không chứng minh được là giả thì sẽ phải có cách nhìn khác về sự tiến hóa của loài gà?

1.2.2.2 Tăng cường ảnh

Gồm một loạt các phương pháp nhằm hoàn thiện trạng thái quan sát một ảnh, không phải là làm tăng cường lượng thông tin vốn có mà làm nổi bật một số đặc tính của ảnh như: thay đổi độ tương phản, lọc nhiễu, nổi biên, làm trơn biên, tăng cường độ tương phản, điều chỉnh mức xám của ảnh.

Hình 5 gồm một ảnh gốc (góc trên bên trái), và 3 ví dụ về việc tăng cường ảnh: (1)Xe mô tô màu xanh được chuyển thành màu lục lam và xe tải màu đỏ trong nền được chuyển thành màu vàng; (2) Tăng độ tương phản của toàn cảnh làm cho ảnh này giống như được chụp trong một ngày trời nắng; (3) Các xe ôtô đỗ bị làm mờ làm chiều sâu của khung cảnh hẹp hơn... Không giống như ghép ảnh, dạng giả mạo này thường ít sử dụng thao tác nhấp chuột hơn.

Hình 5: Ví dụ về tăng cường ảnh

Ảnh gốc (trên trái) và ảnh được thay đổi màu sắc (trên phải), tăng độ tương phản (dưới trái) và làm mờ nền (dưới phải). Mặc dù loại giả mạo này không thay đổi về hình thức hay ý nghĩa của ảnh (như loại ghép ảnh), nhưng nó vẫn có những ảnh hưởng riêng đến thể hiện của ảnh - ví dụ, các tăng cường ảnh đơn giản có thể làm mờ hay làm tăng quá mức các chi tiết của ảnh, hoặc thay đổi thời gian chụp ảnh.

1.2.2.3 Sao chép và dịch chuyển vùng trên ảnh

Một dạng khác thường thấy nữa của ảnh giả mạo là việc sao chép - dịch chuyển các đối tượng trong ảnh, việc này được xem như là che phủ hoặc xóa đi đối tượng. Hình 6.a là ảnh gốc với hai chiếc ô tô, một xe con và một xe tải. Hình 6.b là ảnh 6.a giả mạo với việc che phủ chiếc xe tải bởi một cành cây cũng lấy từ chính trong ảnh. Trong khi hình 6.c là ảnh gốc với chiếc trực thăng nhỏ còn hình 6.d chính là ảnh gốc 6.c đã được bỏ đi đối tượng là trực thăng. Trong cả hai dạng giả mạo này đều được thực hiện từ một ảnh nên độ tương đồng về ánh sáng và bóng là như nhau. Do đó, bằng mắt thường rất khó xác định.

a) Ảnh gốc

b) Ảnh đã che phủ đối tượng

c) Ảnh gốc

d) Ảnh bỏ đi đối tượng

Hình 6: Ảnh che phủ và bỏ đi đối tượng

a) Ảnh gốc

b) Ảnh bổ sung đối tượng

Hình 7: Ảnh bổ sung đối tượng

Hình 7 thể hiện một dạng khác thường thấy của giả mạo sao chép/di chuyển, đó là việc bổ sung thêm đối tượng. Hình 7.a là ảnh gốc chỉ có một chiếc máy bay trực thăng, nhưng trong hình 7.b đã được bổ sung thêm thành ba chiếc trực thăng ở các

vị trí khác nhau. Các trực thăng này chính là được sao chép từ trực thăng gốc nên góc độ và hướng là giống nhau, do đó rất khó cho việc xác thực.

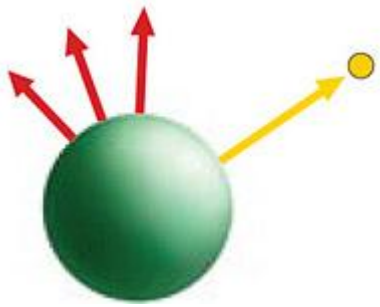
1.2.3 Các cách tiếp cận chính trong xác thực ảnh số

1.2.3.1 Dựa vào hình dạng

Việc phân tích để xác định tính giả mạo có thể dựa vào hình dạng vì việc cắt dán và ghép ảnh thường được thực hiện dựa theo các đường biên, nơi có sự thay đổi không liên tục của cường độ sáng của các điểm ảnh.

1.2.3.2 Dựa vào phân tích nguồn sáng

Tấm ảnh ghép từ nhiều hình ảnh khác nhau sẽ khó có độ thuần nhất về ánh sáng (cường độ chiếu sáng, hướng của ánh sáng…).Ví dụ một quả cầu như hình bên sẽ sáng nhất ở bề mặt có ánh sáng chiếu thẳng góc (hướng của mũi tên vàng), tối nhất ở phía đối diện, các vùng xung quanh nó sẽ sáng với mức độ khác nhau tùy vị trí khuất. Sự phản xạ lại của tia sáng sang không gian hay vật thể xung quanh cũng có mức độ tương ứng.



Để nhận biết hướng của nguồn sáng, bạn phải biết được hướng chiếu sáng trên từng vị trí của bề mặt. Sẽ rất khó nếu nhìn toàn bộ vật thể để xác định nguồn sáng nhưng hãy chú ý đến các đường viền trên bề mặt - nơi hướng ánh sáng vuông góc với bề mặt. Bằng cách đo độ sáng và hướng cùng với một số điểm trên đường viền, các thuật toán có thể xác định được hướng nguồn sáng.

Hình 8: Phát hiện dựa vào hướng chiếu sáng

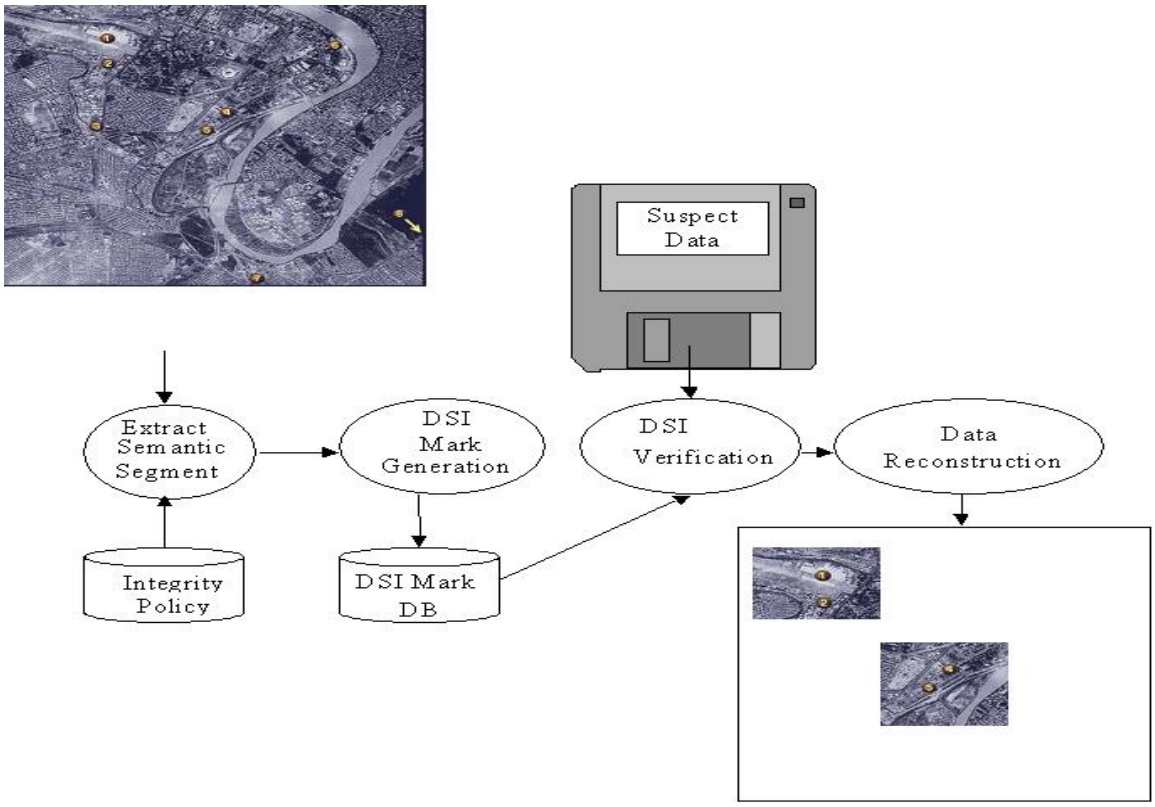
Ví dụ: hình trên là ảnh ghép vì hướng nguồn sáng chiếu vào các viên cảnh sát không tương ứng với những con vịt (xem hướng mũi tên). Việc ghép các ảnh khác nhau hoặc bổ sung thêm đối tượng không phải do sao chép có thể được phát hiện bằng việc phân tích nguồn sáng đối với từng đối tượng, các đối tượng được ghép thường có hướng của nguồn sáng không cùng với các đối tượng trong ảnh gốc.

1.2.3.3 Dựa vào biến đổi màu sắc

Ảnh gốc thu nhận thường được thực hiện bởi một thiết bị. Do tính chất biến đổi của ống kính bao gồm góc độ chụp, độ mở v.v.. nên ảnh thu được thường bị biến dạng theo các tính chất đặc trưng của các nhà sản xuất. Phần ảnh được ghép vào hay bổ sung thường không có sự biến đổi tương đồng về màu sắc ánh sáng.

1.2.3.4 Dựa vào cơ sở dữ liệu

Việc giả mạo ảnh thường dựa vào các ảnh đã có, tức là các ảnh đã được xuất bản bởi một nơi nào đó như: báo chí, trang Web, tạp chí vv... Các ảnh này đã được lưu trữ nên khi xuất hiện một ảnh nghi là giả mạo người ta có thể so sánh các ảnh này với các ảnh gốc trong nguồn ảnh nằm trong cơ sở dữ liệu ảnh.



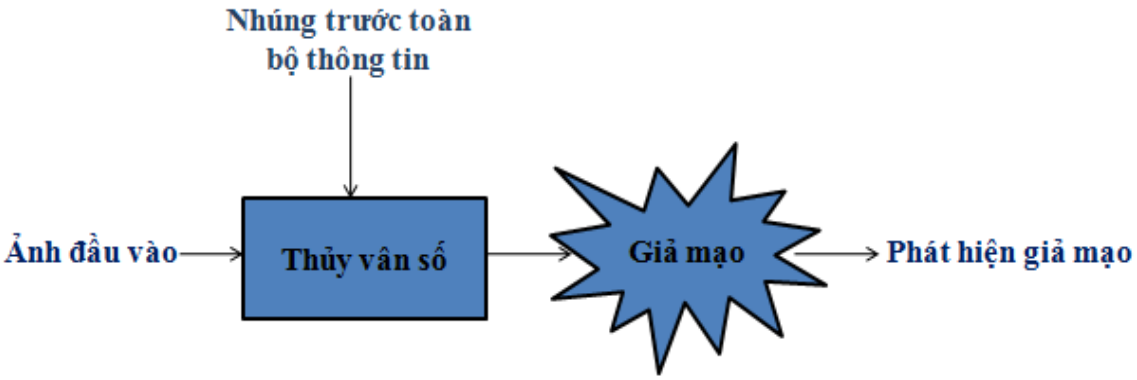
Hình 9: Sơ đồ việc phát hiện giả mạo dựa vào cơ sở dữ liệu.

Chương 2 . MỘT SỐ KỸ THUẬT XÁC THỰC ẢNH SỐ

Có nhiều thuật toán và kỹ thuật để xác thực ảnh số. Nói chung, những kỹ thuật này có thể được chia thành hai nhóm chính: Kỹ thuật chủ động và Kỹ thuật bị động.

Ý tưởng của các kỹ thuật xác thực chủ động là nhúng các thông tin cần thiết vào bức ảnh trước khi phát hành để tránh tình trạng sao chép bất hợp pháp. Dựa vào đó sau này ta có thể xác định được nguồn gốc của bức ảnh. Như vậy kỹ thuật này không hiệu quả lắm trong việc phát hiện giả mạo. Để khắc phục hạn chế này người ta đã nghiên cứu một số kỹ thuật xác thực mà không cần chèn thông tin trước được gọi là kỹ thuật xác thực bị động.

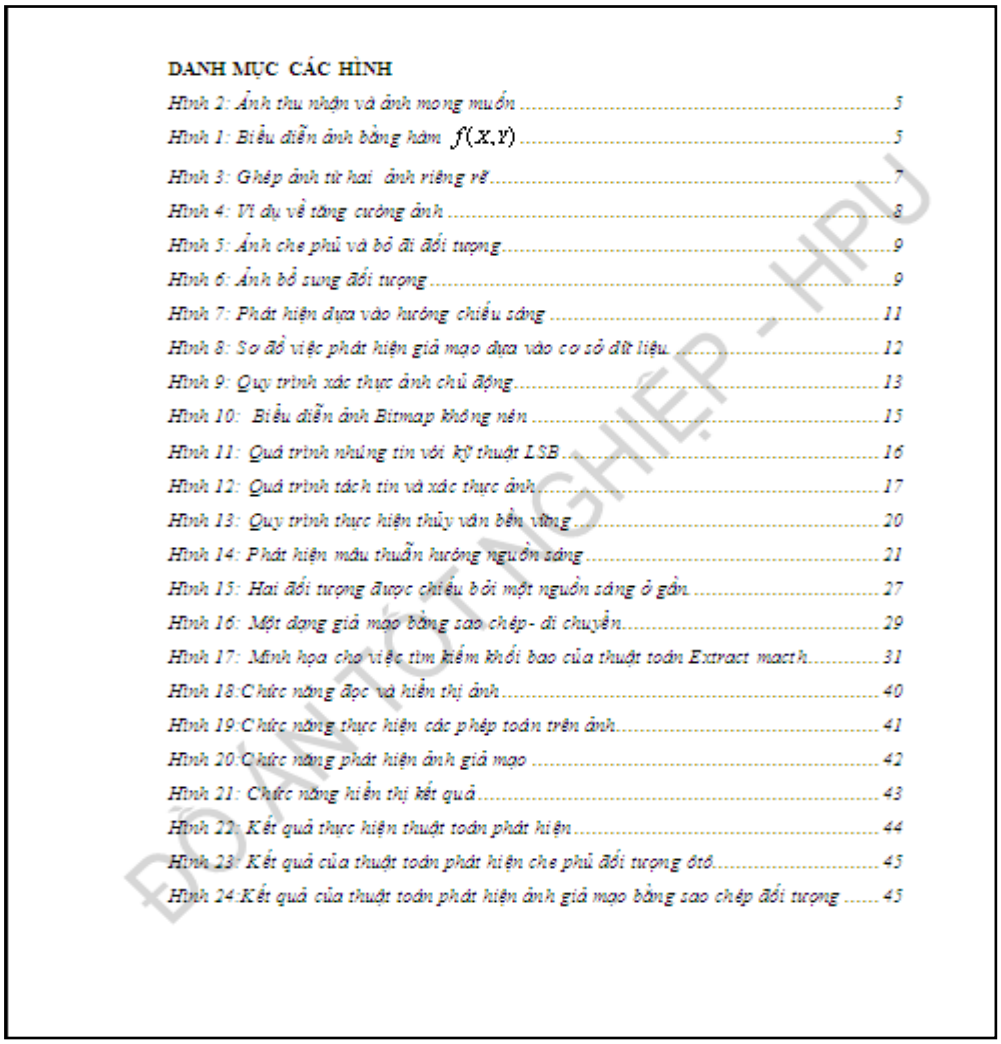
2.1 Các kỹ thuật xác thực ảnh chủ động



Hình 10: Quy trình xác thực ảnh chủ động

❖ Thuỷ vân số (Digital Watermarking) là kỹ thuật nhúng một biểu tượng, chữ ký hay các đánh dấu khác vào trong trong dữ liệu số, như ảnh, âm thanh, video, văn bản... để xác định quyền sở hữu ảnh, chống sự giả mạo và xuyên tạc thông tin.

Ví dụ như trong các tài liệu Word, ta có thể xác định bản quyền bằng cách chọn chức năng thủy vân (Page Layout/ Watermark) và chèn ký tự vào.



DANH MỤC CÁC HÌNH

Hình 11: Ví dụ thủy vân trên tài liệu Word

❖ Một hệ thống thủy vân số bao gồm các thành phần:

1. Thông điệp được nhúng (Message): thường là một chuỗi bits ngắn được dùng để nhúng vào dữ liệu.

2. Dữ liệu phủ (*Cover Data*): Là môi trường nhúng dữ liệu như ảnh, âm thanh, video...

3. Thuật toán nhúng (*Embedding Algorithm*): Thuật toán nhúng thông điệp vào dữ liệu phủ mà không làm thay đổi giá trị sử dụng của dữ liệu phủ.

4. Thuật toán phát hiện thủy vân (*Detection Algorithm*): Thuật toán phát hiện thủy vân và tách chúng khỏi dữ liệu phủ.

5. Dữ liệu đã thủy vân (*Watermarked Data*): Kết quả của quá trình nhúng thông tin vào dữ liệu.

❖ Những nghiên cứu về thủy vân hiện nay chủ yếu tập trung vào vấn đề bảo vệ bản quyền ảnh số với các kỹ thuật:

1. Kỹ thuật thủy vân dễ vỡ (*Fragile WaterMarking*)
2. Kỹ thuật thủy vân bền vững (*Robust WaterMarking*)
3. Nhúng thông tin vào các bit có trọng số thấp (*Least Signification Bits-LSB*)
4. Biến đổi miền không gian ảnh (*Spatial Domain*),
5. Biến đổi miền tần số (*Frequency Domain*),
6. Kỹ thuật trải phổ (*Spread Spectrum*).

Mỗi kỹ thuật đều có những điểm mạnh và điểm yếu riêng nhưng chúng cần thỏa mãn một số tính chất chung, đó là: tính bền vững, tính vô hình, tính khả đảo và thuận nghịch.

2.1.1 Kỹ thuật LSB

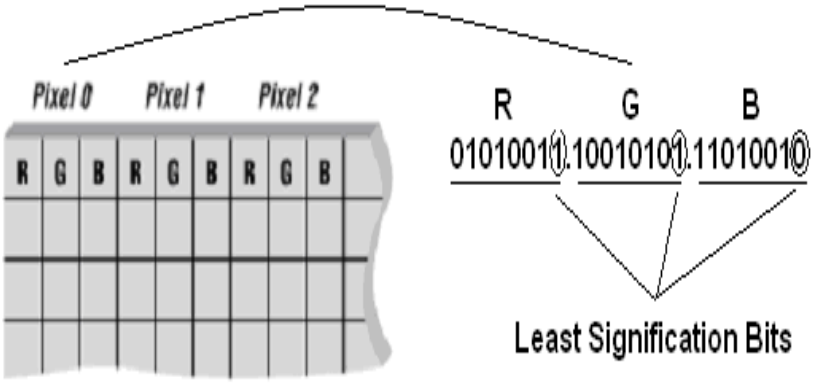
Kỹ thuật nhúng thông tin vào các bit có trọng số thấp là phương pháp đơn giản. Các bit có trọng số thấp có nghĩa là các bit ít quan trọng nhất, các bit đó gần như không có khả năng ảnh hưởng đến việc hiển thị của ảnh. Phương pháp LSB được áp dụng trên các ảnh bitmap không nén và các ảnh có dùng bảng màu. Ý tưởng chính của kỹ thuật này là lấy từng bit của thông điệp cần nhúng rải chúng trên ảnh phủ, bằng cách chèn chúng vào các bit có trọng số thấp.

Kỹ thuật LSB ứng dụng cho việc chống xuyên tạc ảnh phải thỏa mãn tính chất:

- *Tính vô hình*: Mắt thường không thể phát hiện sự thay đổi của ảnh trước khi nhúng thông tin và ảnh sau khi nhúng.
- *Tính không bền vững*: Thông tin nhúng phải dễ bị thay đổi (dễ vỡ) khi có sự tác động làm thay đổi nội dung ảnh dù tác động là nhỏ nhất.
- *Tính phân bố*: Các bít của thông tin nhúng cần được phân bố đều trên ảnh để chống lại sự thay đổi ảnh trên từng vùng.

Trong đồ án này lựa chọn môi trường mang tin là ảnh bitmap không nén 24-bits màu. Với loại ảnh này cho phép nhúng một lượng thông tin đáng kể, việc thực hiện nhúng thông tin vào ảnh dễ thực hiện và khả năng xác thực của thuật toán cũng bảo đảm do mỗi điểm ảnh đều có thể được dùng để lưu trữ các bit giấu. Loại ảnh bitmap

24-bit màu có đặc điểm mỗi điểm ảnh được lưu bởi 24-bits chia thành 3 byte mô tả 3 màu cơ bản là R (red), G (green), B (blue).



Hình 12: Biểu diễn ảnh Bitmap không nén

Trong hình trên biểu diễn ma trận điểm ảnh trong ảnh bitmap mỗi điểm lưu trữ 3 byte (3 x 8 = 24 bit) tương ứng với 3 màu R, G, B mỗi màu có giá trị từ 0 đến 255, ứng với mỗi byte màu thành phần, bit được khoanh tròn gọi là bit có trọng số thấp vì nếu có thay đổi bit đó thì giá trị màu tương ứng chỉ tăng hoặc giảm một đơn vị do đó mắt người rất khó phát hiện sự thay đổi này. Điều này đảm bảo tính vô hình của kỹ thuật thủy vân LSB.

Với kỹ thuật thủy vân LSB trên ảnh 24 bits màu, có thể đánh giá được dung lượng tin được dấu hay khả năng giấu tin:

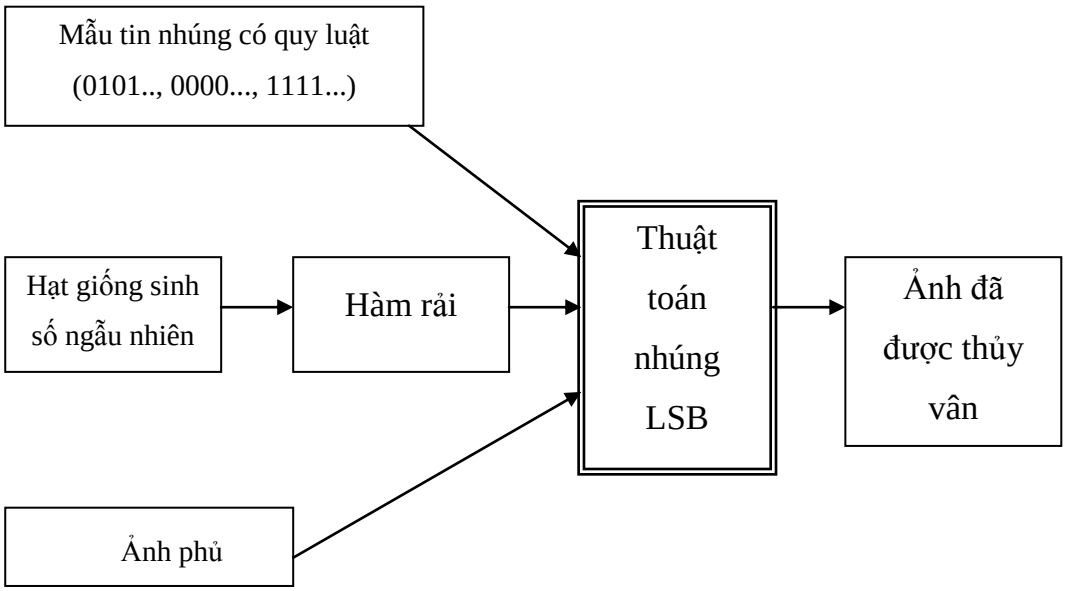
- Nếu giấu 1 bit trọng số thấp nhất của 24 bit màu (bit thứ nhất của màu Blue) ta có: 1/24 (bit ẩn/bit dữ liệu.)
- Nếu giấu 3 bit trọng số thấp của 24 bit màu (3 bit có trọng số thấp tương ứng của màu R, G, B) ta có: 3/24 = 1/8 (bit ẩn/ bit dữ liệu)
- Nếu giấu 6 bit trọng số thấp của 24 bit màu (2 bit thấp nhất của 3 màu tương ứng R, G, B) ta có: 6/24 = 1/4 (bit ẩn/bit dữ liệu).

Như vậy có thể thấy kỹ thuật thủy vân LSB cho phép lượng tin giấu khá lớn tỷ lệ với kích thước ảnh. Ví dụ với ảnh có độ phân giải 800x600 pixel, nếu mỗi điểm ảnh giấu 3 bit thì ta sẽ giấu được: 4320000 bit = 527 (KB), kích thước này tương ứng với 1 đoạn văn khá dài. Việc tính toán trước khả năng giấu tin cho phép chúng ta tạo ra một mẫu tin nhúng tương ứng và trải đều trên bề mặt ảnh phủ.

Vấn đề tiếp theo là phải tạo ra một chuỗi bit nhúng có quy luật để cho phép khi tách thông tin được thủy vân trong ảnh chúng ta có thể phát hiện được sự thay đổi nội dung ảnh nếu chuỗi bit được tách ra phá vỡ quy luật trước khi nhúng nếu không thì ảnh vẫn nguyện vẹn. Độ dài thông điệp nhúng phải là bội số của số bit nhúng trên mỗi điểm ảnh. Ví dụ: $(01)^n$, $(10)^n$, $(0)^n$, $(1)^n$, $n > 1$.

Quá trình nhúng mẫu tin phải tạo ra một phân bố đều trên bề mặt ảnh phủ, điều này được giải quyết bằng một *hàm rải*. Hàm rải là một hàm cần phải chọn các điểm ảnh tương đối ngẫu nhiên. Tuy nhiên tính ngẫu nhiên của thuật toán rải bit nhúng cần có quy luật và được xuất phát từ một hạt giống (số đầu tiên), điều này cho phép thuật toán tách thông tin nhúng cần biết bắt đầu từ đâu. Ví dụ: hàm rải **Random(seed)**, trong đó seed là hạt giống của thuật toán sinh số ngẫu nhiên.

Nhúng tin vào ảnh phủ:

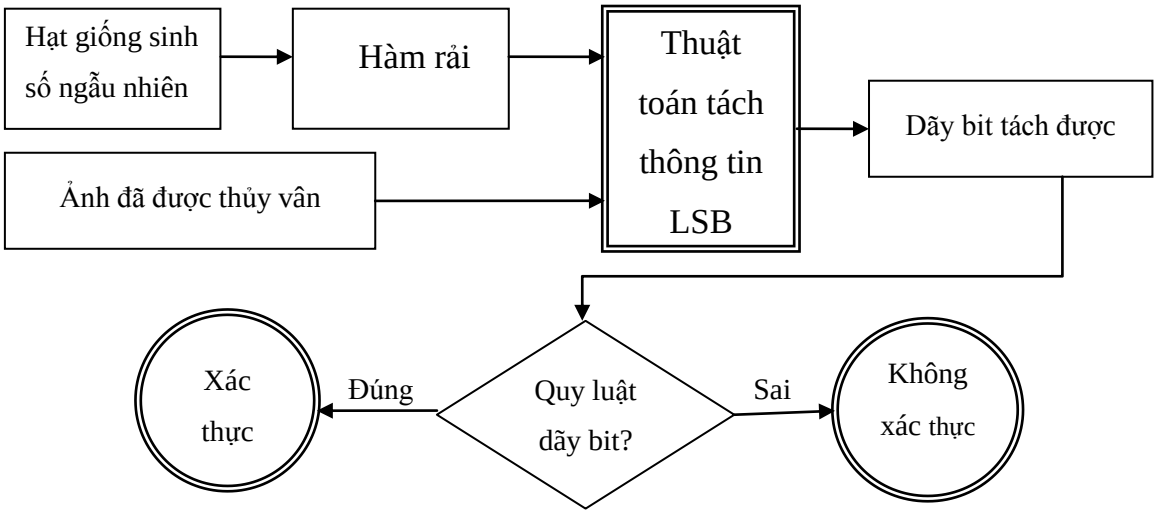


Hình 13: Quá trình nhúng tin với kỹ thuật LSB

a) Thuật toán nhúng thông tin vào ảnh phủ:

1. *Chọn một điểm ảnh ban đầu cho bởi hạt giống (seed)*
2. *Giấu chiều dài mẫu tin vào điểm ảnh đó.*
3. *Duyệt mẫu tin trích 3 bit một:*
   i. *Chọn ngẫu nhiên điểm ảnh chưa dùng (hạt giống seed)*
   ii. *Giấu 3 bit đã chọn vào điểm ảnh đó.*

Tách thông tin nhúng và Xác thực ảnh:



Hình 14: Quá trình tách tin và xác thực ảnh

b) Thuật toán tách tin nhúng và xác thực ảnh:

1. *Chọn điểm ảnh có vị trí bằng giá trị của hạt giống (seed)*
2. *Lấy giá độ dài của dãy bit nhúng từ điểm ảnh đó. L= Độ dài chuỗi bít.*
3. *Khởi tạo i = 0, W ={} chuỗi bít được tách*
4. *Trong khi i < L thực hiện:*
   a) *Chọn ngẫu nhiên điểm ảnh j = Random(seed)*
   b) *Tách 3 bit trọng số thấp tại điểm ảnh j thêm vào W.*
   c) *i = i +3*
5. *Kiểm tra quy luật của chuỗi bít vừa được tách ở bước 4.*
   a) *Nếu W vẫn có quy luật như ban đầu thì ảnh ĐƯỢC XÁC THỰC.*
   b) *Nếu W không có quy luật thì ảnh KHÔNG ĐƯỢC XÁC THỰC.*

Trong phần trên chúng ta đã nghiên cứu kỹ thuật thủy vân số LSB trên ảnh Bitmap không nét 24-bit màu. Và ứng dụng của kỹ thuật này vào việc chống xuyên tạc ảnh. Những năm gần đây có nhiều phương pháp chống xuyên tạc ảnh đã đạt được kết quả tốt nhưng những phương pháp đó đều rất phức tạp. Phương pháp LSB có ưu thế là kỹ thuật khá đơn giản và hiệu quả. Nhưng việc ứng dụng LSB còn nhiều yếu điểm như việc chọn định dạng ảnh phủ, kích thước mẫu tin được thủy vân, độ đo nhiễu...

2.1.2 Kỹ thuật thủy vân bền vững

Thuật toán dưới đây sử dụng kỹ thuật trải phổ trong truyền thông để nhúng thủ vân. Giải tần được sử dụng để chứa tín hiệu thuỷ vân là miền tần số giữa của một khối biến đổi Cosin rời rạc DCT (*Discret Cosne Transformation*) 8×8. Trong đó, các khối DCT 8×8 là những khối ảnh cùng kích thước đã được chọn ra ngẫu nhiên từ ảnh ban đầu và được áp dụng phép biến đổi cosin rời rạc DCT để chuyển sang miền tần số. Mỗi tín hiệu thuỷ vân sẽ được chứa trong một khối.

Mô tả thuật toán

- Input: Một chuỗi các bít thể hiện bản quyền một ảnh
- Output: Một ảnh sau khi thuỷ vân.

Quá trình thủy vân (Watermarking)

- Chia ảnh có kích thước m × n thành (m×n)/64 khối 8×8, mỗi bít sẽ được giấu trong một khối.

- Chọn một khối bất kỳ B và biến đổi DCT khối đó thu được B’

- Chọn hai hệ số ở vị trí bất kỳ trong miền tần số ở giữa của khối DCT, giả sử đó là *b’(i,j)* và *b’(p,q)*. Ta tính: $d = || b'(i,j)|- |b'(p,q)|| \ mod \ a$. Trong đó a là một tham số thoả mãn điều kiện: *a=2(2t+1)*, t là một số nguyên dương.

- Bit s_i sẽ được nhúng sao cho thoả mãn điều kiện sau:

$$\begin{cases} d >= 2t+1 \text{ nếu } s_i = 1 \\ d < 2t+1 \text{ nếu } s_i = 0 \end{cases}$$

- Nếu $d<2t+1$ và $s_i = 1$ thì một trong hai hệ số DCT *b’(i,j)* hoặc *b’(p,q)* có trị tuyệt đối lớn hơn sẽ bị thay đổi để $d>=2t+1$ theo công thức sau:

$$max(|b'(i,j)|, |b'(p,q)|) + (INT(0,75 *a) - d)$$

Với hàm *max(|b’(i,j)|, |b’(p,q)|)* là hàm chọn ra hệ số có trị tuyệt đối lớn hơn, hệ số được chọn sẽ được cộng thêm một lượng là *(INT(0,75 *a) - d)*.

- Tương tự, nếu $d >= 2t+1$ và $s_i = 0$ thì một trong hai hệ số DCT *b’(i,j)* hoặc *b’(p,q)* có trị tuyệt đối lớn hơn sẽ được thay đổi để thoả mãn $d<2t+1$ như sau:

$$max(|b'(i,j)|, |b'(p,q)|) - (d - INT(0,25 *a))$$

Hàm *max(|b’(i,j)|, |b’(p,q)|)* là hàm chọn ra hệ số có trị tuyệt đối lớn hơn, hệ số được chọn sẽ bị trừ đi một lượng là *(d - INT(0,25 *a))*.

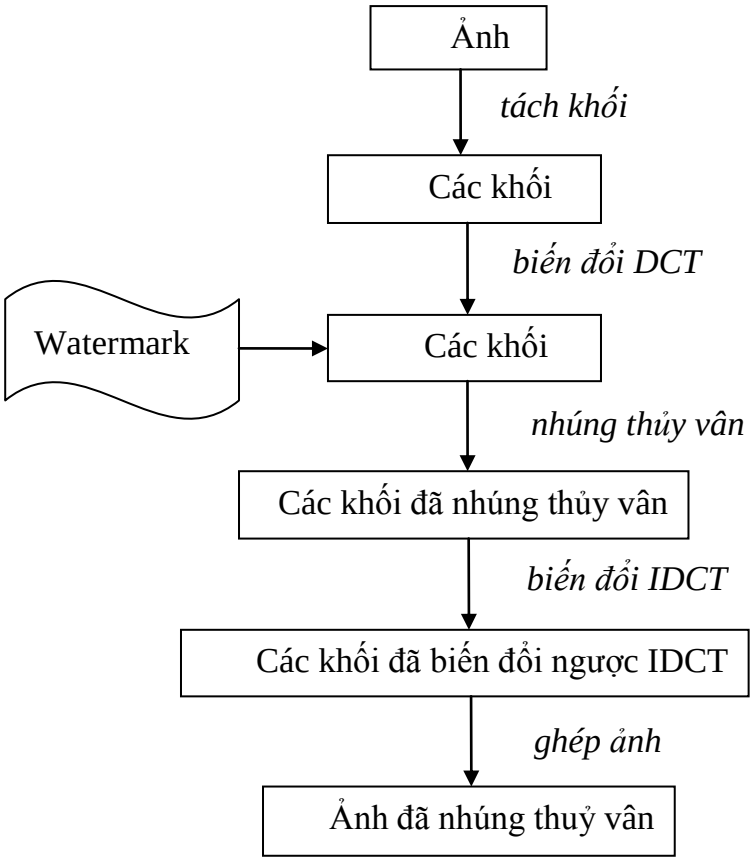
Quá trình giải nhúng để lấy lại thông tin:

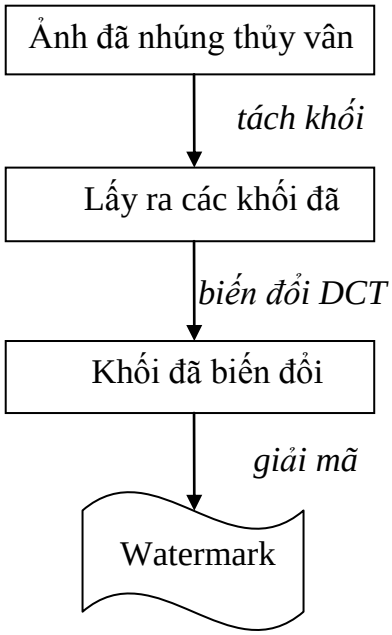
Đọc khối DCT từ ảnh chứa thuỷ vân và vị trí hai hệ số đã biến đổi, tính:

$d = || b'(i,j)|- |b'(p,q)|| \ mod \ a$ với $a = 2(2t+1))$

Nếu $d >= 2t+1$ **thì gán** $s_i = 1$

Nếu $d < 2t+1$ **thì gán** $s_i = 0$





15.1 Quá trình nhúng thủy vân *15.2 Quá trình tách thủy vân*

Hình 15: Quy trình thực hiện thủy vân bền vững

2.2 Các kỹ thuật xác thực ảnh bị động

Không giống như kỹ thuật dựa trên thủy vân và dựa trên chữ ký số, kỹ thuật bị động không cần bất kỳ chữ ký số được tạo ra hoặc thủy vân được nhúng trước. Những kỹ thuật này làm việc dựa trên giả định rằng mặc dù giả mạo kỹ thuật số có thể không để lại manh mối trực quan mà chỉ giả mạo, chúng có thể làm thay đổi cơ bản số liệu thống kê của một hình ảnh. Kỹ thuật này được xem là một hướng đi mới và là một phạm vi đang phát triển nhanh chóng vì không cần bất kỳ thông tin trước về hình ảnh được xác thực hoặc nguồn gốc của nó. Chủ yếu là, cố gắng phân tích từng loại giả mạo một cách riêng biệt (sao chép vùng ảnh, lấy mẫu lại, nén kép JPEG, các mô hình nhiễu trái ngược ...) và phát hiện từng loại riêng biệt.

2.2.1 Phát hiện dựa vào mâu thuẫn hướng nguồn sáng

Khi tạo ra một ảnh giả, ví dụ ghép hai người đứng cạnh nhau, thường khó tương thích về các điều kiện ánh sáng từ các ảnh riêng lẻ. Các đối tượng được ghép thường có hướng nguồn sáng không tương đồng với các đối tượng trong ảnh gốc. Do vậy, sự khác nhau về hướng ánh sáng có thể là một dấu hiệu của giả mạo ảnh số. Ví dụ, hình 16 là một ảnh ghép, ở đó hai người được chụp với ánh sáng ở các góc độ khác nhau.

Hình 16: Phát hiện mâu thuẫn hướng nguồn sáng

Với phạm vi hướng nguồn sáng có thể được ước lượng cho các đối tượng khác nhau trong một ảnh, sự mâu thuẫn trong hướng ánh sáng có thể được sử dụng như bằng chứng của giả mạo ảnh số. Trong phần này sẽ trình bày một số phương pháp ước lượng hướng nguồn sáng và dựa vào đó để phát hiện ảnh giả mạo.

2.2.1.1 Các phương pháp ước lượng hướng nguồn sáng

Phần này trình bày thuật toán ước lượng tự động hướng chiếu của nguồn sáng đối với một ảnh đơn. Thuật toán gồm ba bước. Đầu tiên tìm ra những đường có khả năng là biên khuất với xác suất cao nhất. Sau đó với mỗi đường biên khuất chúng ta sẽ ước lượng véc-tơ chỉ hướng chiếu của nguồn sáng theo mô hình bóng đổ. Cuối cùng các ước lượng đó được đưa vào mô hình mạng Bayes để tìm một ước lượng thích hợp nhất cho hướng chiếu của nguồn sáng. Điều kiện là đối tượng phải có bề mặt Lambertian, đồng thời toàn bộ bề mặt có hệ số phản chiếu là hằng số.

2.2.1.2 Nguồn sáng xa (3-D)

Hướng chuẩn hóa cho việc ước lượng hướng nguồn sáng bắt đầu từ việc xây dựng một số giả thuyết đơn giản:

1) Bề mặt của đối tượng phản xạ ánh sáng đẳng hướng (bề mặt Lambertian)
2) Bề mặt của đối tượng có một hằng số phản xạ
3) Bề mặt được chiếu bởi nguồn sáng điểm ở xa vô hạn
4) Góc giữa bề mặt và hướng của nguồn sáng trong khoảng từ 0 đến 90 độ

Với những giả thuyết như vậy, mật độ ảnh có thể được mô tả bởi:

$$I(x,y) = R\left(\vec{N}(x,y).\vec{L}\right) + A \qquad (1)$$

Với:

+ R là hằng số phản xạ

+ $\vec{L}$ là véctơ 3 chiều chỉ hướng của nguồn sáng

+ $\vec{N}(x,y)$ là véctơ pháp tuyến của bề mặt tại điểm (x,y)

+A là hằng số giới hạn ánh sáng xung quanh

Nếu chúng ta chỉ quan tâm đến hướng của nguồn sáng, thì số hạng hệ số phản xạ, R, có thể được xem là hằng số. Phương trình tuyến tính kết quả đưa ra một ràng buộc đơn trong đó có 4 thành phần chưa biết, 3 thành phần của $\vec{L}$ và A.

Với ít nhất 4 điểm có hệ số phản xạ giống nhau R, và các pháp tuyến bề mặt rõ ràng, $\vec{N}$, hướng nguồn sáng và số hạng ánh sáng nền A có thể được tính bằng việc sử dụng ước lượng bình phương tối thiểu chuẩn.

$$E(\vec{L}, A) = \left\| M \begin{pmatrix} L_x \\ L_y \\ L_z \\ A \end{pmatrix} - \begin{pmatrix} I(x_1, y_1) \\ I(x_2, y_2) \\ \vdots \\ I(x_p, y_p) \end{pmatrix} \right\|^2$$

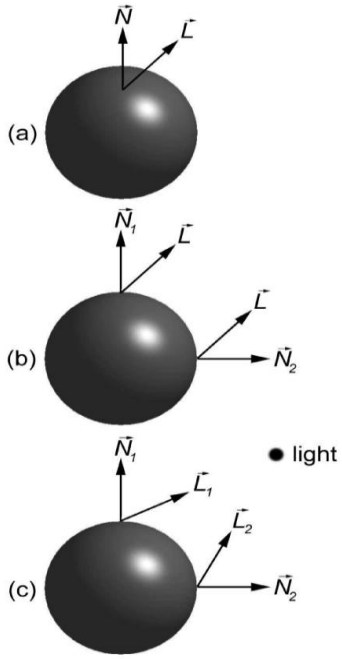
$$= \left\| M\vec{v} - \vec{b} \right\|^2 \qquad (2)$$

Trong đó L_x, L_y, và L_z là các thành phần của hướng nguồn sáng $\vec{L}$, và

$$M = \begin{pmatrix} N_x(x_1, y_1) & N_y(x_1, y_1) & N_z(x_1, y_1) & 1 \\ N_x(x_1, y_1) & N_y(x_1, y_1) & N_z(x_1, y_1) & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ N_x(x_1, y_1) & N_y(x_1, y_1) & N_z(x_1, y_1) & 1 \end{pmatrix} \quad (3)$$

$N_x(x_p,y_p)$, $N_y(x_i,y_i)$, $N_x(x_i,y_i)$ là các thành phần của pháp tuyến bề mặt $\vec{N}$, tại tọa độ (x_i,y_i). Hàm lỗi bậc 2 ở trên được cực tiểu bằng việc lấy vi phân đối với thành phần chưa biết, $\vec{v}$, thiết lập kết quả bằng không, và tìm lời giải cho $\vec{v}$.

$$\vec{v} = (M^T M)^{-1} M^T \vec{b} \quad (4)$$



Trong hình bên (a) nguồn sáng xa (3-D); (b) nguồn sáng xa (2-D); và (c) nguồn sáng cục bộ (2-D). Trong các trường hợp 2-D, thành phần z của pháp tuyến bề mặt ($\vec{N}$,) bằng 0. Không giống nguồn sáng xa, hướng nguồn sáng cục bộ ($\vec{L}$,) khác nhau trên bề mặt của hình cầu.

2.2.1.3 Nguồn sáng xa (2-D)

Phần này trình bày một giải pháp thông minh hơn để ước lượng 2 thành phần của hướng nguồn sáng (L_x và L_y) từ một ảnh đơn. Cách tiếp cận này cung cấp ít thông tin về hướng nguồn sáng, làm cho vấn đề dễ kiểm soát hơn. Tại đường biên của một bề mặt, thành phần z của pháp tuyến bề mặt bằng 0, N_z=0. Do vậy, ta chỉ ước lượng 2 thành phần L_x, L_y của hướng nguồn sáng. Lúc này, hàm lỗi (2) có dạng:

$$E(\vec{L}, A) = \left\| M \begin{pmatrix} L_x \\ L_y \\ A \end{pmatrix} - \begin{pmatrix} I(x_1, y_1) \\ I(x_2, y_2) \\ \vdots \\ I(x_p, y_p) \end{pmatrix} \right\|^2$$

$$= \left\| M\vec{v} - \vec{b} \right\|^2 \qquad (5)$$

Trong đó

$$M = \begin{pmatrix} N_x(x_1, y_1) & N_y(x_1, y_1) & 1 \\ N_x(x_2, y_2) & N_y(x_2, y_2) & 1 \\ \vdots & \vdots & \vdots \\ N_x(x_p, y_p) & N_y(x_p, y_p) & 1 \end{pmatrix} \qquad (6)$$

Như phần trước, hàm lỗi này được cực tiểu hóa sử dụng bình phương tối thiểu chuẩn để cho kết quả giống như (4), nhưng ma trận M có dạng như (6). Trong trường hợp này, giải pháp yêu cầu biết các pháp tuyến bề mặt 2-D từ ít nhất 3 điểm rõ ràng (p≥3) trên một bề mặt có hệ số phản xạ giống nhau.

Làm giảm giả định hệ số phản xạ bất biến:

Trong phần trước ta giả sử hệ số phản xạ trên toàn bộ bề mặt của đối tượng là bất biến, phần này trình bày cách làm giảm giả định hệ số phản xạ bất biến này bằng việc giả sử rằng hệ số phản xạ cho một miếng nối bề mặt cục bộ (không phải toàn bộ bề mặt) là bất biến. Việc này đòi hỏi chúng ta đánh giá các hướng nguồn sáng riêng biệt, $\vec{L}^i$, cho mỗi miếng nối dọc theo bề mặt. Với giả định nguồn sáng xa, sự định hướng các ước lượng này sẽ không thay đổi, nhưng độ lớn của chúng thì có thể thay đổi.

Xét một bề mặt được chia thành n miếng nối, và để đơn giản về mặt ký hiệu, giả sử rằng mỗi miếng gồm p điểm. Hàm lỗi mới được cực tiểu hóa xây dựng bởi sự kết hợp các miếng nối cùng nhau, phiên bản 2-D của ràng buộc (1) là:

$$E_1\left(\vec{L}^1, \dots, \vec{L}^n, A\right) = \left\| M \begin{pmatrix} L_x^1 \\ L_y^1 \\ \vdots \\ L_x^n \\ L_y^n \\ A \end{pmatrix} - \begin{pmatrix} I(x_1^1, y_1^1) \\ \vdots \\ I(x_p^1, y_p^1) \\ \vdots \\ I(x_1^n, y_1^n) \\ \vdots \\ I(x_p^n, x_p^n) \end{pmatrix} \right\|^2$$

$$= \left\| M\vec{v} - \vec{b} \right\|^2 \qquad (7)$$

trong đó

$$M = \begin{pmatrix} N_x(x_1^1, y_1^1) & N_y(x_1^1, y_1^1) & & 0 & 0 & 1 \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ N_x\left(x_p^1, y_p^1\right) & N_y\left(x_p^1, y_p^1\right) & & 0 & 0 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & & N_x(x_1^n, y_1^n) & N_y(x_1^n, y_1^n) & 1 \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ 0 & 0 & & N_x\left(x_p^n, x_p^n\right) & N_y\left(x_p^n, x_p^n\right) & 1 \end{pmatrix}$$

(8)

Như phần trước, hàm lỗi bình phương trên được cực tiểu hóa sử dụng bình phương tối thiểu và cho kết quả như (4). Trong trường hợp này, giải pháp cung cấp n ước lượng về các hướng ánh sáng 2-D, $\vec{\boldsymbol{L}}^{\boldsymbol{1}}, \dots, \vec{\boldsymbol{L}}^{\boldsymbol{n}}$, và một số hạng ánh sáng nền A.

Trong khi sự ước lượng cục bộ về các hướng nguồn sáng cho phép giảm nhẹ giả định hệ số phản xạ bất biến, thì có khả năng cho ra ít kết quả ổn định hơn. Chú ý rằng với giả định về nguồn sáng điểm xa, sự định hướng n hướng ánh sáng sẽ như nhau. Với giả định thêm vào rằng sự thay đổi trong hệ số phản xạ từ miếng nối này đến miếng nối khác là tương đối nhỏ (tức là, sự thay đổi trong độ lớn láng giềng của $\vec{\boldsymbol{L}}^{\boldsymbol{i}}$ là nhỏ), ta có:

$$\boldsymbol{E_2}\left(\vec{\boldsymbol{L}}^{\boldsymbol{1}}, \dots, \vec{\boldsymbol{L}}^{\boldsymbol{n}}\right) = \sum_{\boldsymbol{i=2}}^{\boldsymbol{n}} \left\| \vec{\boldsymbol{L}}^{\boldsymbol{i}}, \dots, \vec{\boldsymbol{L}}^{\boldsymbol{i-1}} \right\|^2 \qquad (9)$$

Phần lỗi thêm vào này cản trở các ước lượng láng giềng khác với ước lượng khác. Hàm lỗi bình phương, E1(.), phương trình(7), được quy định bởi việc kết hợp nó với E2(.) lấy tỷ lệ bởi hệ số λ, cho ra hàm lỗi cuối cùng:

$$\boldsymbol{E}\left(\vec{\boldsymbol{L}}^{\boldsymbol{1}}, \dots, \vec{\boldsymbol{L}}^{\boldsymbol{n}}, \boldsymbol{A}\right) = \boldsymbol{E_1}\left(\vec{\boldsymbol{L}}^{\boldsymbol{1}}, \dots, \vec{\boldsymbol{L}}^{\boldsymbol{n}}, \boldsymbol{A}\right) + \lambda \boldsymbol{E_2}\left(\vec{\boldsymbol{L}}^{\boldsymbol{1}}, \dots, \vec{\boldsymbol{L}}^{\boldsymbol{n}}\right) \qquad (10)$$

Hàm lỗi phối hợp này vẫn có thể được cực tiểu hóa sử dụng phương pháp bình phương tối thiểu. Đầu tiên, hàm lỗi $E_2(.)$ được viết dưới dạng sau:

$$E_2(\vec{v}) = \|C\vec{v}\|^2 \quad (11)$$

trong đó ma trận C kích thước (2n-1 x 2n+1) được cho bởi:

$$C = \begin{pmatrix} -1 & 0 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & \cdots & 0 & 0 & 0 & 0 & 0 \\ \vdots & & & & \ddots & & & & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & -1 & 0 & 1 & 0 \end{pmatrix} \quad (12)$$

với $\vec{v} = (L_x^1 \quad L_y^1 \quad L_x^2 \quad L_y^2 \quad \cdots \quad L_x^n \quad L_y^n \quad A)^T$. Hàm lỗi (10) có dạng:

$$E(\vec{v}) = \left\|M\vec{v} - \vec{b}\right\|^2 + \lambda\|C\vec{v}\|^2 \quad (13)$$

Lấy vi phân hàm lỗi này cho ra:

$$E'(\vec{v}) = 2M^T M\vec{v} - 2M^T\vec{b} + 2\lambda C^T C\vec{v}$$

$$= 2(M^T M + \lambda C^T C)\vec{v} - 2M^T\vec{b} \quad (14)$$

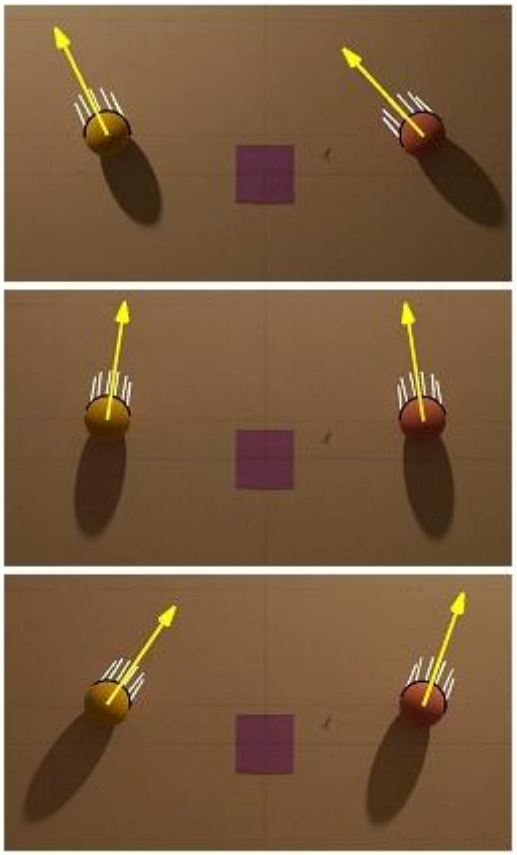
Thiết lập kết quả bằng 0 và đưa ra lời giải cho $\vec{v}$:

$$\vec{v} = (M^T M + \lambda C^T C) + M^T\vec{b} \quad (15)$$

Trong đó dấu + là giả nghịch đảo. Sự ước lượng hướng ánh sáng cuối cùng được tính bằng việc lấy trung bình n ước lượng hướng nguồn sáng kết quả $\vec{L}^1, \ldots, \vec{L}^n$.

2.2.1.4 Nguồn sáng cục bộ(2-D)

Với các phần trên, giả định nguồn sáng xuất phát từ vô tận. Với nguồn sáng bộ phận, các công thức trên trên không còn phù hợp (hình 13.c).



Hình 17: Hai đối tượng được chiếu bởi một nguồn sáng ở gần.

Mô hình cho một nguồn sáng xa, phương trình (1), có thể được viết lại thích hợp với nguồn sáng cục bộ như sau:

$$I(x,y) = R\left(\vec{N}(x,y)\,.\,\vec{L}(x,y)\right) + A \qquad (16)$$

Lúc này hướng của nguồn sáng là một hàm các tọa độ ảnh.

Chúng ta bắt đầu bằng việc thừa nhận hướng nguồn sáng đối với mỗi miếng nối bề mặt cục bộ là không thay đổi. Hướng nguồn sáng cho mỗi miếng nối bề mặt được ước lượng sử dụng phương trình (7). Trong phần trước, ở phương trình (9) khuyến khích các ước lượng láng giềng bằng nhau. Trong trường hợp nguồn sáng cục bộ, người ta mong muốn các hướng của các láng giềng hội tụ đến một điểm đơn gần đó.

$$\boldsymbol{E_2\left(\vec{L}^1, \ldots, \vec{L}^n\right) = \sum_{i=1}^{n} \left\| \vec{C}^i \vec{L}^i \right\|^2} \qquad (17)$$

Như trong phần trước, hàm lỗi cuối cùng được cực tiểu hóa cho bởi:

$$E\left(\vec{L}^1, \dots, \vec{L}^n, A\right) = E_1\left(\vec{L}^1, \dots, \vec{L}^n, A\right) + \lambda E_2\left(\vec{L}^1, \dots, \vec{L}^n\right) \quad (18)$$

Trong đó $E_1(.)$ như phương trình(7), và λ là hệ số tỷ lệ. Không giống như phần trước, hàm lỗi này không thể được cực tiểu hóa theo phép phân tích, mà thay vào đó được cực tiểu hóa sử dụng sự cực tiểu gradient liên hợp lặp. Mặc dù dạng hàm lỗi thể hiện tương tự với dạng hàm trong phần trước nhưng các ma trận C_i phụ thuộc vào ước lượng nguồn sáng $\vec{L}^i$, vì thế cần sự cực tiểu hóa lặp lại.

2.2.1.5 Nhiều nguồn sáng

Trong những phần trên ta giả định rằng chỉ có ánh sáng phát ra từ một nguồn sáng duy nhất chiếu lên vật thể và các nguồn sáng khác coi không đáng kể và được xem xét với hằng số giới hạn ánh sáng xung quanh (A), điều này thường phù hợp với các ảnh ngoài ngoài trời nơi mà mặt trời là nguồn chiếu sáng chủ yếu. tuy nhiên với các ảnh trong nhà giả định này thiếu hợp lý vì đối tượng có thể được chiếu rọi bởi nhiều nguồn sáng.

Ánh sáng có thuộc tính rất đặc biệt đó là tuyến tính. Giả sử có 2 nguồn sáng chiếu lên đối tượng, khi đó hàm mật độ ảnh có dạng:

$$I(x,y) = R\left((\vec{N}(x,y) + \vec{L}_1) + (\vec{N}(x,y).\vec{L}_2)\right) + A$$
$$= R\left(\vec{N}(x,y).(\vec{L}_1 + \vec{L}_2)\right) + A$$
$$= R\left(\vec{N}(x,y).\vec{L}^+\right) + A$$

Trong đó, $\vec{L}^+$là vecto tổng của các vecto riêng lẻ $\vec{L}_1$và $\vec{L}_2$. Chú ý rằng mô hình này có dạng giống như nguồn sáng đơn, công thức (1). Vì vậy, sử dụng cùng cách tiếp cận như trong các phần trước, sẽ cho kết quả là một ước lượng về nguồn sáng ảo - vector tổng của các nguồn sáng riêng lẻ. Điều này mở rộng dễ dàng cho ba hay nhiều nguồn sáng riêng lẻ.

Mặc dù không phổ biến lắm, nhưng có khả năng là tổ hợp các nguồn sáng khác nhau sẽ có tổng nguồn sáng ảo giống nhau, nên cách tiếp cận này không thể phát hiện một mâu thuẫn trong nguồn sáng.

2.2.2 Kỹ thuật phát hiện sao chép – dịch chuyển vùng trên ảnh

2.2.2.1. Giới thiệu dạng giả mạo bằng Sao chép – di chuyển

Đây là một dạng phổ biến của kỹ thuật giả mạo ảnh số. Trong đó một phần của hình ảnh được sao chép và dán vào một phần khác của cùng một hình ảnh thường với ý định để che dấu một đối tượng hoặc một khu vực của hình ảnh. Vì thế, phần được sao chép từ cùng một hình ảnh vào trong bản sao di chuyển giả mạo, các bảng màu, thành phần nhiễu, ánh sáng, và hầu hết các thuộc tính khác sẽ tương thích với phần còn lại của hình ảnh, nó trở nên khó khăn hơn để phát hiện bằng mắt thường. Để gây khó khăn cho việc phát hiện giả mạo, người ta có thể sử dụng nhiễu, làm mờ, thay đổi tương phản vv... định dạng nén JPEG làm cho việc phát hiện thậm chí còn khó khăn hơn nhiều.

Một ví dụ cho loại giả mạo này có thể được nhìn thấy trong hình một nhóm binh lính được nhân đôi che Tổng thống George W. Bush.

Hình 18: Một dạng giả mạo bằng sao chép- di chuyển

Quá trình này có thể được thực hiện mà không có bất kỳ sửa đổi, bổ sung vào những vùng được nhân đôi. Kết quả là, các giả mạo khu vực này sẽ thể hiện các đặc điểm giống như phần còn lại của hình ảnh mà làm cho nó khó để xác định bằng cách sử dụng các công cụ được thiết kế để phát hiện các bất thường trong hình ảnh. Vì vậy, mục tiêu trong việc phát hiện sao chép, di chuyển ảnh bị giả mạo là phát hiện ra hình ảnh các khu vực tương tự hoặc giống nhau.

Do đó, để phát hiện vùng bị sao chép, di chuyển trong ảnh giả mạo, chúng ta cần một kỹ thuật mà có thể phát hiện vùng ảnh xuất nhiều hơn một lần trong ảnh.

Tuy nhiên, việc tìm kiếm những vùng giống nhau có thể sẽ không đủ trong một số trường hợp, vì kẻ giả mạo có thể sử dụng các công cụ chỉnh sửa, thêm nhiễu, hoặc nén hình ảnh. Hơn nữa, có nhiều cách kết hợp trong khu vực sao chép như là xoay nhẹ, thu nhỏ, hoặc xóa mờ mà không làm ảnh hưởng các thông số hình ảnh.

Vì vậy, một kỹ thuật phát hiện giả mạo sao chép-di chuyển tốt có thể phát hiện ra các vùng ảnh trùng lặp dù đã bị sửa đổi hay thêm nhiễu và nén nhỏ.

2.2.2.2. Các kỹ thuật phát hiện sao chép – dịch chuyển vùng trên ảnh

Bất kỳ giả mạo bằng sao chép- di chuyển nào cũng đưa ra một mối tương quan giữa các đoạn hình ảnh gốc và đoạn được dán. Sự tương quan này có thể được sử dụng làm cơ sở để phát hiện thành công của loại giả mạo này. Bởi vì giả mạo có thể sẽ được lưu trong định dạng mất dữ liệu JPEG và vì một công cụ chỉnh sửa có thể được sử dụng hoặc các công cụ xử lý định vị ảnh khác, các phân đoạn có thể không phù hợp tuyệt đối, nhưng cũng chỉ tương đối. Vì thế, chúng ta cần đưa ra các yêu cầu như sau cho thuật toán xác thực:

1. Các thuật toán xác thực cho phép phát hiện tương đối các đoạn ảnh giống nhau.

2. Thời gian làm việc có thể chấp nhận được.

3. Khối phát hiện không phải là những điểm ảnh có sự tương đồng.

Trong phần này sẽ trình bày một số phương pháp phát hiện loại giả mạo này bằng việc so khớp khối.

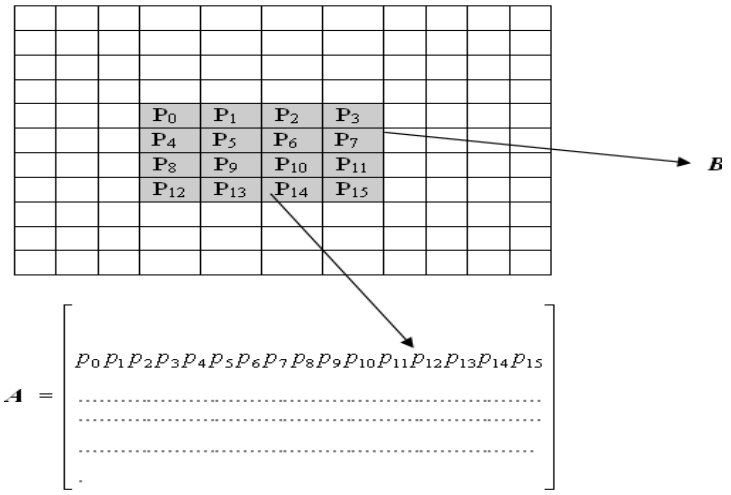
a) So khớp chính xác (Exact Match)

Giả sử bức ảnh có kích thước $M \times N$, với B là kích thước nhỏ nhất của khối bao mà người dùng định nghĩa để đối sánh. với mỗi điểm ảnh ta xác định được một khối bao ma trận $B \times B$ điểm ảnh. Như vậy với bức ảnh $M \times N$ ta xác định được $(M-B+1)\times(N-B+1)$ khối bao. Khối bao này được trượt đi một pixel dọc theo ảnh từ trái sang phải và từ trên xuống dưới. Với mỗi vị trí của khối B x B, các giá trị pixel của khối này được trích ra theo các cột và lưu thành một dòng trong ma trận 2 chiều A với (B×B) cột và $(M-B+1)\times(N-B+1)$dòng. Mỗi dòng tương ứng với một vị trí của khối trượt.

Hai hàng giống nhau trong ma trận A tương đương với 2 khối bao giống nhau trong ảnh. Chúng ta sắp xếp các hàng trong ma trận A theo thứ tự từ điển, yêu cầu

này sẽ được thực hiện trên $MN\log_2(MN)$ bước. Sau đó ta dễ dàng tìm kiếm bằng cách duyệt MN hàng của ma trận đã qua sắp xếp A và tìm kiếm hai hàng giống nhau liên tiếp.

Kết quả thuật toán sẽ tìm kiếm và đưa ra được tập các vùng bao giống nhau là bằng chứng chứng minh ảnh đã bị cắt dán.



Hình 19: Minh họa cho việc tìm kiếm khối bao của thuật toán Exact macth

Hạn chế: Nếu ảnh giả mạo được lưu với định dạng JPEG thì phần lớn các khối đồng nhất sẽ bị mất và thuật toán sẽ phát hiện sai.

b) So khớp bền vững (Robust match)

Ý tưởng chính của thuật toán phát hiện dựa trên so khớp thô tương tự như so khớp chính xác chỉ khác là thuật toán so khớp thô không so sánh và sắp thứ tự dựa trên giá trị pixel trong khối mà dựa trên hệ số DCT. Với mỗi khối, biến đổi DCT được tính, các hệ số DCT được lượng tử hóa và lưu thành một dòng của ma trận A. Ma trận này sẽ có (M-B+1) × (N-B+1) dòng và B×B cột.

Các dòng của A được sắp xếp theo từ điển như trước. Tuy nhiên, phần còn lại của thủ tục thì khác. Vì lúc này giá trị của các hệ số DCT lượng tử hóa được so sánh thay vì biểu diễn điểm nên thuật toán có thể tìm ra quá nhiều khối khớp (các khối khớp sai). Do vậy, thuật toán này hướng về các vị trí chung của mỗi cặp khối tương thích và chỉ chọn ra một cặp khối cụ thể nếu và chỉ nếu có nhiều cặp tương thích khác cùng vị trí như thế (chúng có chung shift-vectơ). Hướng tới mục tiêu này, nếu 2 dòng liên tiếp của ma trận sắp xếp A được tìm thấy thì thuật toán này lưu các vị trí của các khối tương thích trong một danh sách riêng (ví dụ, lưu tọa độ pixel góc trên

trái của một khối) và tăng bộ đếm shift-vectơ C. Cụ thể, lấy (i_1,i_2) và (j_1,j_2) là các vị trí của 2 khối tương thích. Shift-vectơ s giữa 2 khối tương thích này được tính:

$$\boldsymbol{s = (s_1.s_2) = (i_1 - j_1, i_2 - j_2)}$$

Với mỗi cặp khối tương thích, chúng ta tăng bộ đếm shift-vectơ C lên 1:

$$C(s_1, s_2) = C(s_1, s_2) + 1$$

Các vector dịch chuyển được tính và bộ đếm C được tăng lên với mỗi cặp dòng liên tiếp khớp nhau trong ma trận sắp xếp A. Shift-vectơ C được khởi tạo bằng 0 trước khi thuật toán bắt đầu. Kết thúc tiến trình so khớp, bộ đếm C thể hiện tần số của các shift-vectơ . Sau đó, thuật toán này tìm ra tất cả các shift-vectơ s(1), s(2) , …, s(k) sao cho C(s (r)) > T với r = 1, …, K. Các khối khớp nhau đóng góp vào các shift-vectơ này được tô màu giống nhau và được xem như các khối lặp.

c) Phát hiện các vùng lặp dựa vào phép phân tích thành phần chính

Hạn chế của các cách tiếp cận đã nêu là nhạy với các thay đổi nhỏ giữa các vùng lặp ví dụ do tạp nhiễu thêm vào hay kỹ thuật nén mất thông tin. Ở đây miêu tả một thuật toán tiếp theo khắc phục được hạn chế này trong khi vẫn giữ lại tính hiệu quả của nó.

Xét một ảnh cấp xám với N pixels. Một ảnh được lợp bằng việc chồng các khối b pixels ($\sqrt{b} \times \sqrt{b}$ pixels), mỗi khối được giả sử nhỏ hơn đáng kể so với kích thước của các vùng giống nhau được phát hiện. Lấy $\vec{x}_i$, i = 1,…, N_b biểu thị cho các khối này ở dạng vectơ, với $N_b = (\sqrt{N} - \sqrt{b} + 1)^2$. Bây giờ chúng ta xem xét một biểu diễn khác các khối ảnh này dựa trên phép phân tích thành phần chủ yếu (PCA). Giả sử rằng các khối $\vec{x}_i$ có giá trị trung bình là 0, và tính ma trận đồng biến (covariance matrix) như sau:

$$C = \sum_{i=1}^{N_b} \vec{x}_i \, \vec{x}_i^T \qquad (1)$$

Các vectơ riêng $\vec{e}_j$, của ma trận C, có các giá trị riêng tương ứng λ_j, thỏa:

$$C\vec{e}_j = \lambda_j \, \vec{e}_j \qquad (2)$$

định nghĩa các thành phần chủ yếu, trong đó j = 1,…, b và $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_b$. Các vectơ riêng, $\vec{e}_j$, hình thành một cơ sở tuyến tính mới cho mỗi khối ảnh, $\vec{x}_i$:

$$\vec{x}_i = \sum_{j=1}^{b} a_j \, \vec{e}_j \qquad (3)$$

Số chiều của biểu biễn này có thể được giảm một cách đơn giản bằng việc bỏ bớt tổng trong phương trình (3) thành N_t số hạng đầu tiên. Lưu ý rằng phép chiếu lên N_t vectơ riêng đầu tiên của cơ sở PCA cho ra một sự xấp xỉ N_t chiều tốt nhất theo nghĩa các bình phương tối thiểu (nếu phân bố của $\vec{x}_i s$ là phân bố Gaussian đa chiều). Vì vậy, Sự biểu diễn giảm về số chiều cung cấp một khoảng trống thuận tiện để nhận ra các khối giống nhau khi có mặt của tạp nhiễu, vì sự bỏ bớt cơ sở này sẽ bỏ đi các biến đổi cường độ nhỏ.

Thuật toán phát hiện tiến hành như sau. Đầu tiên, làm giảm hơn nữa các biến đổi nhỏ do tạp nhiễu, biểu diễn số chiều giảm của mỗi khối ảnh $\vec{a}_i$, được lượng tử hóa, $\vec{a}_i/Q$ trong đó số nguyên dương Q ám chỉ số lượng các bít lượng tử hóa. Một ma trận N(b×b) được xây dựng sao cho các dòng của nó chứa các hệ số lượng tử này. Lấy ma trận S là kết quả của việc sắp xếp theo từ điển các dòng của ma trận thành thứ tự cột. $\vec{s}_i$ ám chỉ dòng thứ i của ma trận được sắp xếp này, và bộ (x_i, y_i) cho biết các tọa độ ảnh của khối (góc trên trái) tương ứng với $\vec{s}_i$. Tiếp theo xét tất cả các cặp dòng $\vec{s}_i$ và $\vec{s}_j$, khoảng cách dòng của nó, |i - j|, trong ma trận được sắp xếp S nhỏ hơn một ngưỡng cho trước. Trong ảnh, offset của tất cả các cặp như thế được cho bởi:

$$(x_i - x_j, y_i - y_j) \quad if\ x_i - x_j > 0$$
$$(x_j - x_i, y_i - y_j) \quad if\ x_i - x_j < 0$$
$$(0, |y_i - y_j|) \quad if\ x_i = x_j$$

Từ một danh sách tất cả các offset như thế, các vùng lặp trong ảnh được phát hiện bằng việc nhận ra các offset với sự xảy ra cao. Ví dụ, một vùng lặp lớn sẽ gồm nhiều khối nhỏ hơn, mỗi khối trong chúng sẽ xuất hiện trong sự xấp xỉ gần với mỗi khối khác trong ma trận được sắp xếp theo thứ tự từ điển và sẽ có cùng offset. Để tránh các sai sót do các vùng cường độ đồng nhất, độ lớn offset thấp hơn một ngưỡng cho trước phải được lờ đi.

Các kết quả của sự phát hiện này có thể được hình dung bằng việc xây dựng một biểu đồ lặp. Tạo ra một ảnh zero có cùng kích thước như ảnh gốc, và tất cả các điểm trong một vùng lặp được gán một giá trị cấp xám duy nhất. Độ phức tạp của thuật toán này, bị chi phối bởi việc sắp xếp theo thứ tự từ điển, là $O(N_t N \log N)$, trong đó N_t là số chiều của các biểu diễn PCA và N là tổng số pixel của ảnh.

Có ít nhất 2 cách để mở rộng thuật toán này cho các ảnh màu. Cách tiếp cận đơn giản nhất là xử lý độc lập mỗi kênh màu (ví dụ, RGB) để tạo ra 3 biểu đồ lặp. Cách tiếp cận thứ 2 là áp dụng PCA với các khối màu kích thước 3b, và tiến hành theo cách đã miêu tả ở trên.

Chương 3. CHƯƠNG TRÌNH THỬ NGHIỆM

Trong chương này sẽ trình bày phần xây dựng chương trình thử nghiệm kỹ thuật phát hiện ảnh giả mạo trong môi trường lập trình Visual C++. Chương trình áp dụng thuật toán tìm các vùng lặp trong ảnh để phát hiện loại ảnh giả mạo sinh bởi thao tác copy và dịch chuyển vùng trên ảnh.

3.1 Phát biểu bài toán

3.1.1 Phát biểu bài toán

Cho một ảnh kích thước $M \times N$ điểm ảnh. Tìm xem trong ảnh có chứa các vùng lặp giống nhau hay không?

Input: Ảnh kích thước $M \times N$

Output: Tìm xem có các vùng ảnh giống nhau không.

3.1.2 Thuật toán:

Sử dụng thuật toán Exact Match nhằm tìm ra các khối bao giống nhau trên cùng một ảnh, bao gồm các bước sau:

Bước 1: Xác định kích thước cho đoạn cần so khớp. Giả sử đoạn này là hình vuông có kích thước B×B điểm ảnh.

Bước 2: Trượt khối này đi từng điểm ảnh dọc theo ảnh từ góc trên trái xuống góc dưới phải.

Bước 3: Trích các giá trị điểm ảnh của mỗi khối và lưu thành một dòng trong ma trận 2 chiều A.

Bước 4: Sắp xếp ma trận A theo thứ tự tăng dần.

Bước 5: Kiểm tra hai hàng liên tiếp trong mảng lưu sau khi đã sắp xếp, nếu chúng giống nhau thì đưa ra 2 tập khối bao giống nhau tương ứng. Hai dòng đồng nhất trong ma trận A tương ứng với 2 khối ảnh đồng nhất kích thước B×B điểm ảnh.

- **Lưu ý**: Ảnh sử dụng là ảnh có kích thước càng nhỏ thì thời gian phát hiện càng nhanh và kích thước quá lớn thì không thể áp dụng được thuật toán này.

3.2 Phân tích thiết kế chương trình

3.2.1 Phân tích chức năng và thiết kế modul chương trình

Chương trình được xây dựng trên môi trường lập trình VC++ 2008 gồm những chức năng sau:

3.2.1.1 Đọc ảnh và hiển thị ảnh

Việc nạp ảnh từ tệp vào mảng số và đưa ảnh từ mảng số hiển thị ra màn hình là cần thiết cho một chương trình xử lý ảnh nên ta sẽ xây dựng các modul riêng:

- Chứa các khai báo về file ảnh và các thủ tục mở file
- Mở file ảnh
- Đọc bảng màu
- Đọc dòng ảnh vào mảng
- Hiển thị ảnh ra màn hình

3.1.1.2 Các phép toán trên ảnh

a) Biến đổi ảnh

- Biến đổi ảnh âm bản

Bước 1:

- Vào tệp myFunction.h để khai báo tên hàm `convertNegative` ();

Vào tệp myFunction.cpp để mô tả hàm `convertNegative` () với nội dung như sau:

```
fipWinImage *convertNegative(fipWinImage *image)
{
   WORD W=image->getWidth();
   WORD H=image->getHeight();
fipWinImage *result=new fipWinImage(FIT_BITMAP,W,H,
image-> getBitsPerPixel());
   WORD i,j;
   for(i=0;i<W;i++)
         for(j=0;j<H;j++)
         {
               setPixel(result,i,j,getPixel(image,i,j));
               setColor(result,i,j,&getColor(image,i,j));
         }
   result->invert();
   return result;
}
```

Bước 2:

Vào menu tạo chức năng biến đổi ảnh âm bản theo thông tin sau :

`ID_COLORIMAGE_NEGATIVE` , sau đó vào class wizard để tiến hành Add hàm `OnConvertNegative();` (hàm nay sẽ được nằm ở trong lớp SMImageView.cpp)

- Vào lớp SMImageView.cpp để khai báo hàm Onanhamban() theo nội dung sau :

```
void CSMImageView::OnConvertNegative()
{
   CSMImageDoc *pDoc=GetDocument();
   SetCursor(m_cursor);
   pDoc->image=convertNegative(pDoc->image);
   Invalidate();
}
```

Tương tự ta thực hiện với các hàm khác để tạo các chức năng:

- Biến đổi sang ảnh 8bits
- Biến dổi sang ảnh nhị phân
- Nén ảnh
- Giãn ảnh
- Tăng sáng
- Giảm sáng
- Tăng độ tương phản
- Giảm độ tương phản
- Biến đổi ảnh âm bản
- Biến đổi sang ảnh 8bits
- Biến dổi sang ảnh nhị phân

b) Xoay ảnh

- Xoay dọc ảnh
- Xoay ngang ảnh

c) Lọc ảnh

- Lọc thông cao
- Lọc thông thấp

-Lọc trung vị

- Lọc Gaussian

d) Phát hiện biên

- Kỹ thuật phát hiện biên Gradient : dùng mặt nạ Sobel và mặt nạ PreWitt.
- Kỹ thuật phát hiện biên Laplace

3.2.1.3. Phát hiện ảnh giả mạo

Bước 1:

Vào tệp myFunction.h để khai báo tên hàm `ExactMatch` ()

Vào tệp myFunction.cpp để mô tả hàm `ExactMatch` () với nội dung như sau:

```
fipWinImage *ExactMatch(fipWinImage *image,fipWinImage *Region,int
B,COLORREF m_color)
{
	int _width=image->getWidth();
	int _height=image->getHeight();
	fipWinImage *result=new
fipWinImage(FIT_BITMAP,_width,_height,image->getBitsPerPixel());
	fipWinImage *layer=new
fipWinImage(FIT_BITMAP,_width,_height,image->getBitsPerPixel());
	clock_t start,end;
	int i,j;
	BYTE *A;
	int *luuCol,*luuRow;
	for(i=0;i<_width;i++)
		for(j=0;j<_height;j++){
		setPixel(layer,i,j,getPixel(image,i,j));
		setColor(layer,i,j,&getColor(image,i,j));
		setPixel(result,i,j,getPixel(image,i,j));
		setColor(result,i,j,&getColor(image,i,j));
		}
	A=new BYTE[(_width-B+1)*(_height-B+1)*B*B];
	luuCol=new int[(_width-B+1)*(_height-B+1)*B*B];
	luuRow=new int[(_width-B+1)*(_height-B+1)*B*B];
	int k=0;

	start=clock();
	layer->convertTo8Bits();
	for(i=0;i<(_width-B+1);i++)
		for(j=0;j<(_height-B+1);j++){
			for(int a=0;a<B;a++)
				for(int b=0;b<B;b++){
					A[k]=getPixel(layer,i+a,j+b);
					luuCol[k]=i+a;
					luuRow[k]=j+b;
					k++;
				}
	}
	int *luu,count=0;
	luu=new int[(_width-B+1)*(_height-B+1)];
	for(i=0;i<k/(B*B)-1;i++)
		for(j=i;j<k/(B*B);j++)
		{
			if(comparedRows(A,i,j,B*B))
			{
				if(!checkExist(luu,i,count))
					luu[count++]=i;
				if(!checkExist(luu,j,count))
					luu[count++]=j;
			}
		}
	RGBQUAD rg;
	rg.rgbRed=GetRValue(m_color);
```

```
   rg.rgbGreen=GetGValue(m_color);
   rg.rgbBlue=GetBValue(m_color);
   i=0;
   while(i<count)
   {
         for(j=0;j<B*B;j++)
         {

   setColor(result,luuCol[luu[i]*B*B+j],luuRow[luu[i]*B*B+j],&rg);

   setPixel(Region,luuCol[luu[i]*B*B+j],luuRow[luu[i]*B*B+j],255);
         }
         i++;
   }
   end=clock();
 CString s;
   s.Format("Thoi gian %.3f Giay",(double)(end-start)/CLOCKS_PER_SEC);
   AfxMessageBox(s);
   delete A;
   delete luuCol;
   delete luuRow;
   delete luu;
   return result;
}
```

Bước 2:

Vào menu tạo chức năng phát hiện vùng giả mạo theo thông tin sau :

`ID_DETECTION_EXACTMATCH`, sau đó vào class wizard để tiến hành Add hàm `OnExactMatch()` (hàm nay sẽ được nằm ở trong lớp SMImageView.cpp)

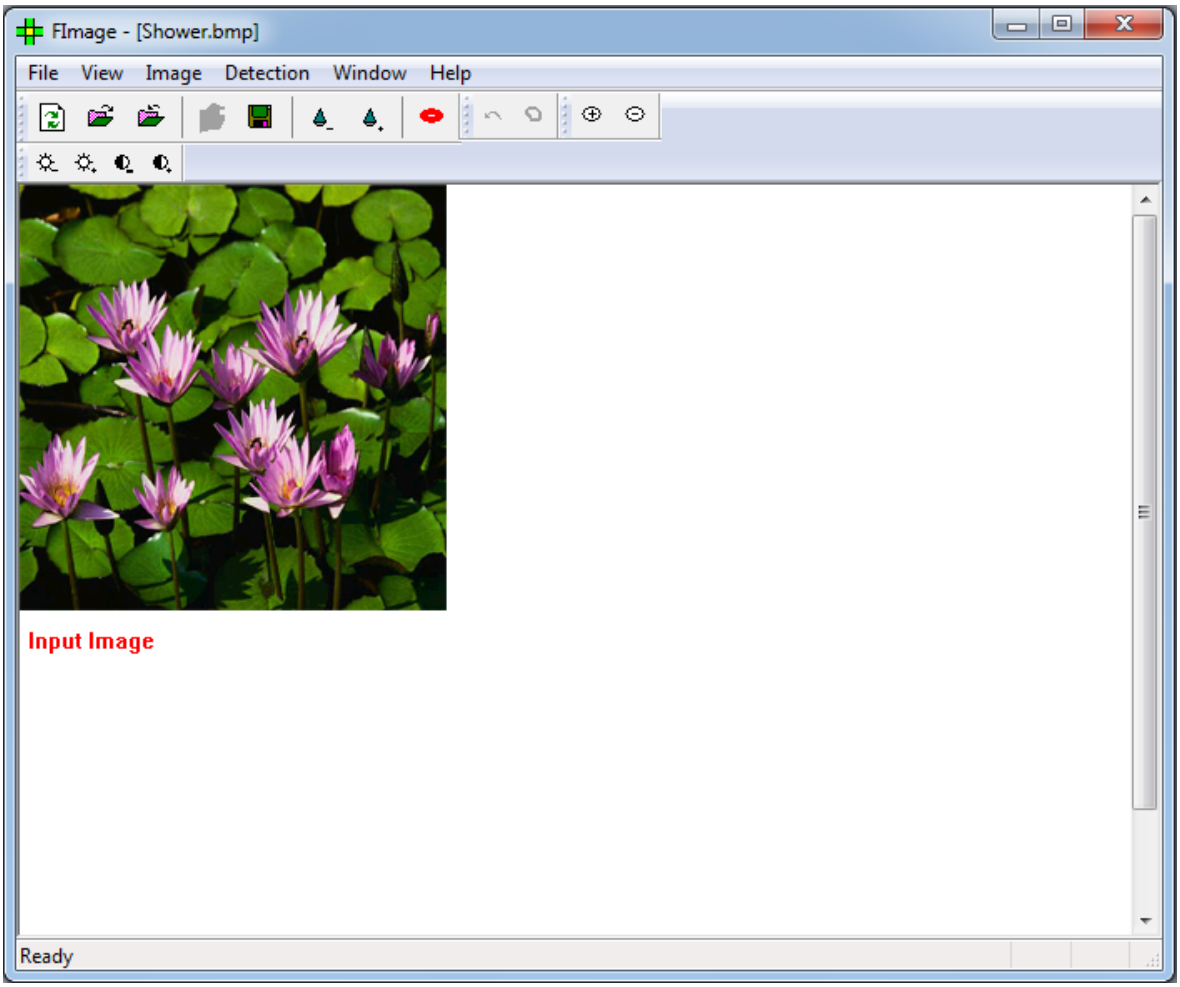
Vào lớp SMImageView.cpp để khai báo hàm `OnExactMatch()` theo nội dung sau:

```
void CSMImageView::OnExactMatch()
{
   CSMImageDoc *pDoc=GetDocument();
   UpdateData(TRUE);
   CReference dlg;
   dlg.choose=1;
   _RegionImg=new fipWinImage(FIT_BITMAP,pDoc->image->getWidth(),pDoc-
>image->getHeight(),8);
   if(dlg.DoModal()==IDOK)
   {
         SearchDraw();
         SetCursor(m_cursor);
         COLORREF color;
         color=RGB(dlg.m_red,dlg.m_green,dlg.m_blue);
         seimage=ExactMatch(pDoc-
>image,_RegionImg,dlg.Threshold,color);
         _UnRegionImg=seimage;
         _activeUp=true;
         _CloseUndo=true;
         Invalidate();
   }
}
```

3.2.2 Một số giao diện của chương trình

3.2.2.1 Giao diện nạp và hiển thị ảnh

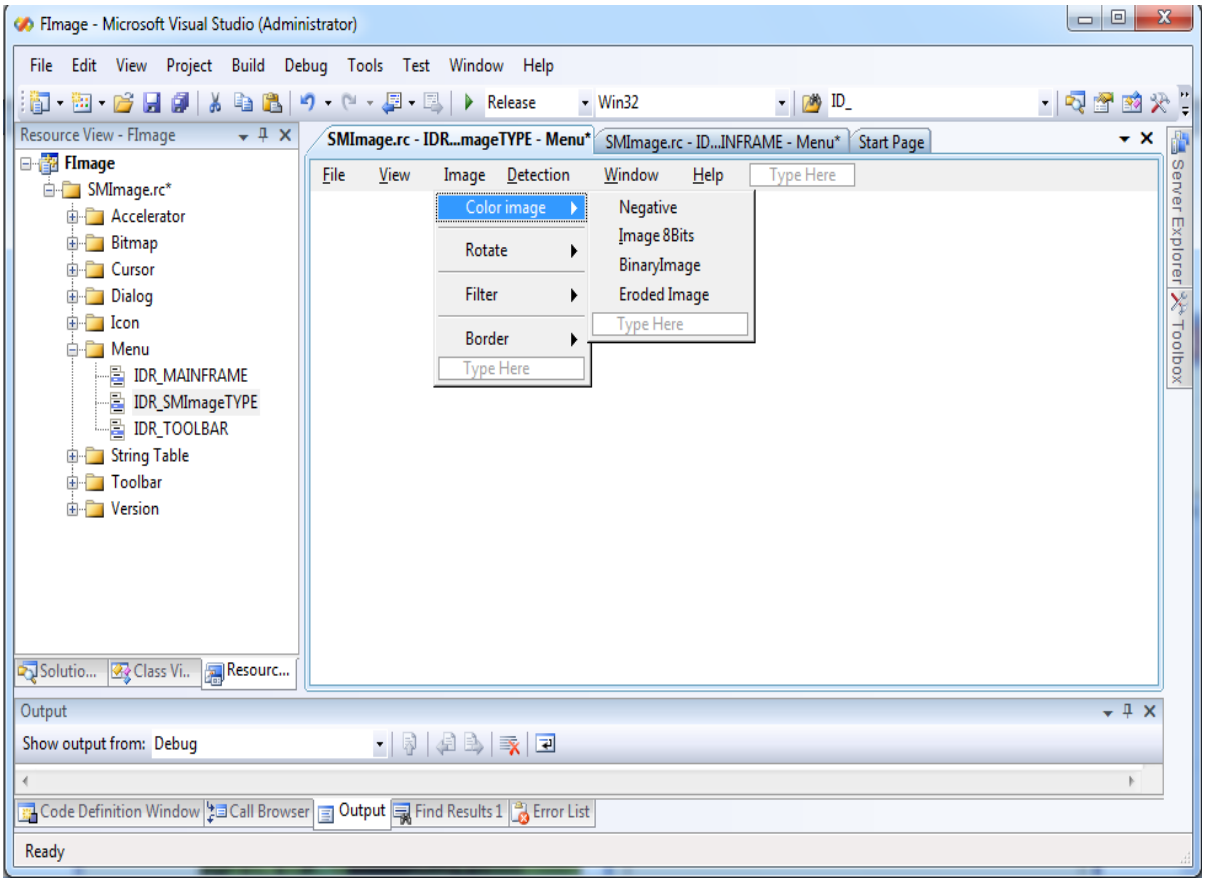
Từ cửa sổ chương trình, người dùng ấn nút Open để file lưu trữ ảnh và lựa chọn mở file ảnh bất kì.



Hình 20: Giao diện hiển thị ảnh

3.2.2.2 Giao diện thực hiện các phép toán trên ảnh

Trong menu Image tạo các chức năng nhỏ thực hiện các phép toán trên ảnh. Người dùng có thể lựa chọn bằng cách nhấn vào các mục đó và có thể undo hành động vừa thực hiện bằng cách ấn nút refresh.

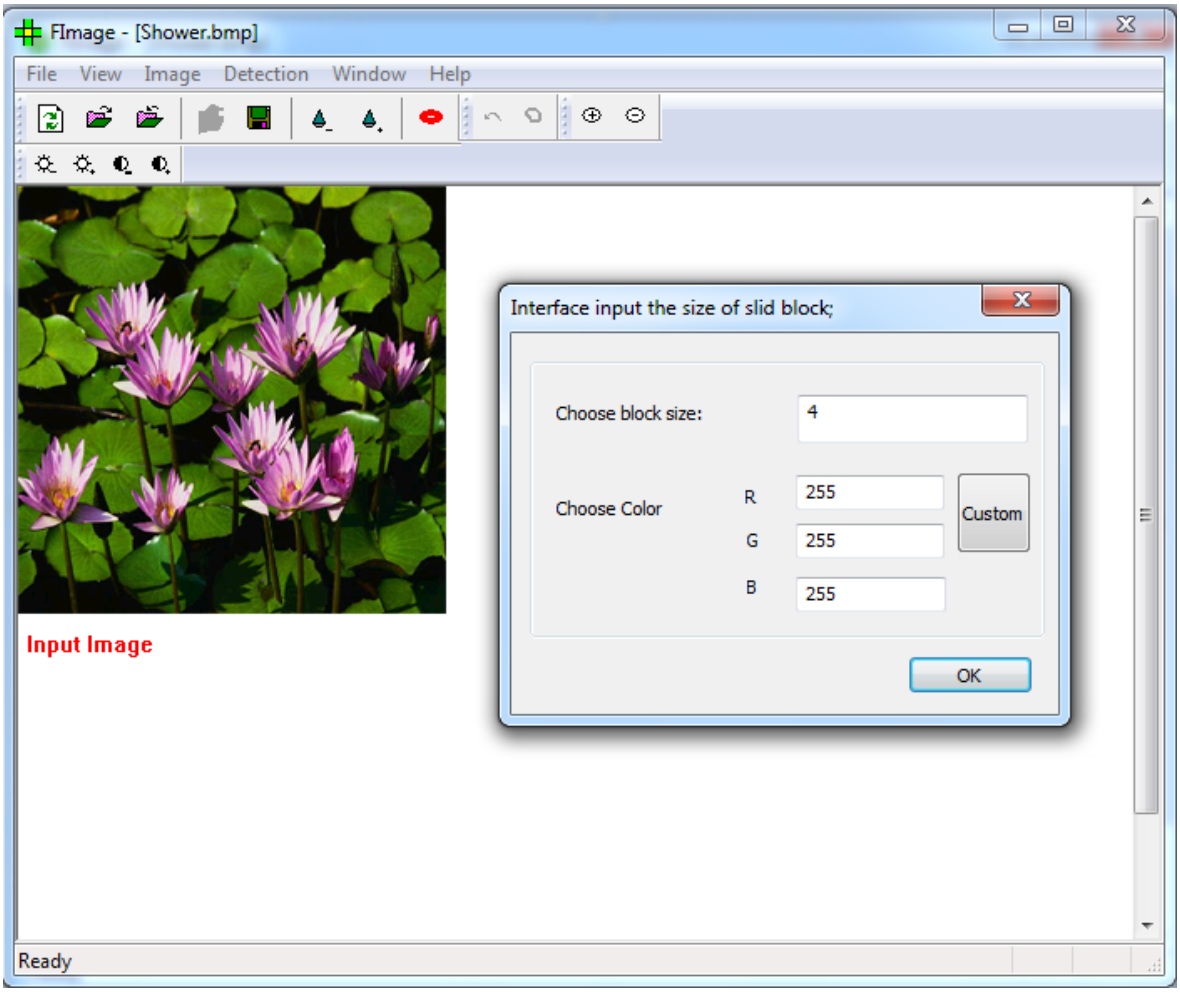


Hình 21:Giao diện thực hiện các phép toán trên ảnh

3.2.2.3 Giao diện phát hiện ảnh giả do sao chép – dịch chuyển vùng

Người dùng lựa chọn ảnh cần kiểm tra từ trong file ảnh của mình, sau đó chọn menu Detection/ Exact Match, rồi thiết lập các thông số:

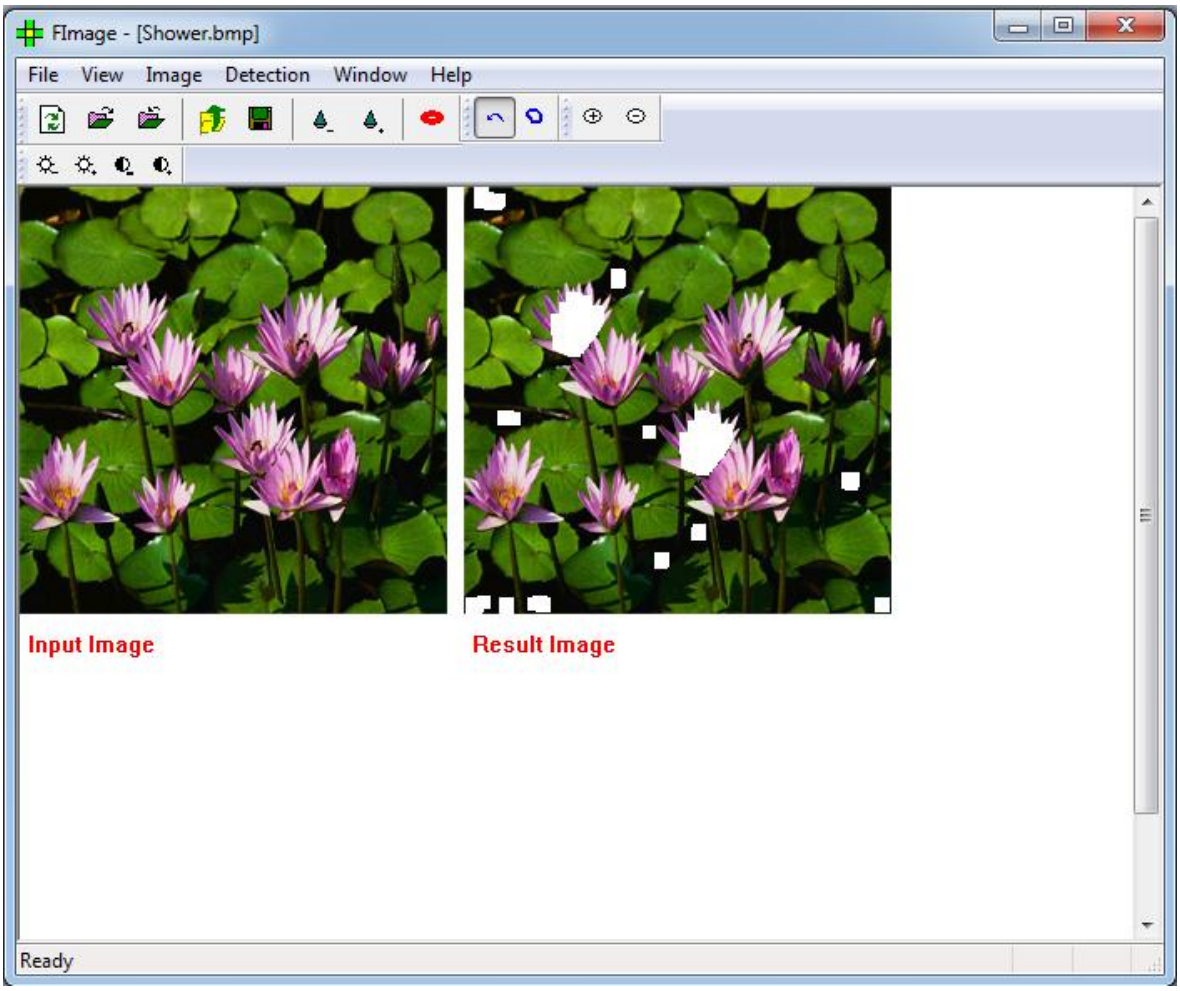
- Chọn kích thước khối bao: 4, 8,16…
- Chọn không gian màu
  Tiếp theo nhấn OK và chương trình sẽ tiến hành tìm kiếm và cho ra kết quả sau đó.



Hình 22:Giao diện phát hiện ảnh giả mạo

3.2.2.3 Giao diện hiển thị kết quả

Tùy vào kích thước ảnh kiểm tra thời gian cho ra kết quả tương ứng, vùng ảnh được tô trắng là vùng giả mạo của bức ảnh.



Hình 23: Giao diện hiển thị kết quả vùng giả mạo

3.3.3 Một số kết quả thực nghiệm

Thực hiện cài đặt thử nghiệm kỹ thuật phát hiện ảnh giả mạo sinh bởi thao tác sao chép và dịch chuyển vùng trên ảnh. Bước đầu với một số kết quả sau:

Hình 24 là ảnh gốc và ảnh giả mạo. Trong ảnh giả mạo người ta đã che khuất chiếc trực thăng bằng một vùng cũng lấy từ ảnh này. Bên dưới là kết quả của thuật toán phát hiện áp dụng cho ảnh giả mạo được lưu với định dạng BMP.

- Kích thước vùng được chọn để so khớp là 8×8 pixels.
- Thời gian chạy chương trình khoảng 40 giây.

(a) ảnh gốc

(b) ảnh giả mạo

(c)

Hình 24: Kết quả thực hiện thuật toán phát hiện che phủ đối tượng máy bay

Hình 25.c và 25.d là các kết quả của việc thực hiện thuật toán phát hiện cho ảnh giả mạo ở hình 25.b với các kích thước vùng được chọn để so khớp là 8×8 pixels và 16×16 tương ứng.

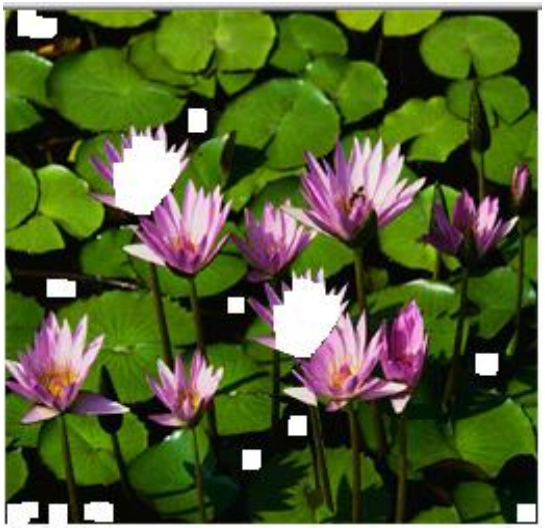
(a) ảnh gốc

(b) ảnh giả mạo

(c) Ảnh kết quả với B=8

(d) Ảnh kết quả với B=16

Hình 25: Kết quả của thuật toán phát hiện che phủ đối tượng ôtô



(a) B = 8

(b) B=16

Hình 26:Kết quả của thuật toán phát hiện ảnh giả mạo bằng sao chép đối tượng

Hình 26 là các kết quả chạy thuật toán phát hiện cho ảnh giả mạo với các kích thước vùng ảnh được chọn để so khớp khác nhau.

Từ thực nghiệm ta thấy việc chọn kích thước vùng ảnh để so khớp có ảnh hưởng nhiều đến kết quả cũng như độ chính xác của thuật toán. Các kích thước vùng so khớp lớn có thể làm cho thuật toán bỏ lỡ một số khối tương thích, còn kích thước vùng quá nhỏ có thể cho ra quá nhiều khối tương thích sai. Vấn đề xác định vùng ảnh như thế nào để có kết quả phát hiện tốt nhất là một vấn đề khó, thông thường chỉ dựa trên các kết quả thực nghiệm.

KẾT LUẬN

Trong khuôn khổ hạn chế, đồ án quan tâm đến một số dạng của giả mạo thuộc loại thứ hai và tập trung vào tìm hiểu các kỹ thuật nhằm phát hiện ảnh giả mạo dựa vào kỹ thuật cắt ghép trên cùng một ảnh, cụ thể đã đạt được một số kết quả sau:

- Trình bày tổng quan các dạng ảnh giả mạo.
- Tìm hiểu các cách tiếp cận chính để phát hiện ảnh giả mạo.
- Nghiên cứu chi tiết kỹ thuật xác thực ảnh bằng phương pháp Exact Match.
- Cài đặt thử nghiệm chương trình dựa trên thuật toán Exact Match.

Kỹ thuật xác thực ảnh bằng phương pháp Exact Match đã giải quyết tốt vấn đề phát hiện sao chép - di chuyển trong ảnh giả mạo. Thuật toán này có khả năng phát hiện đối với các ảnh giả mạo dạng cắt dán từ chính một ảnh. Đây cũng chính là cách thường được các đối tượng sử dụng trong quá trình tạo ảnh số giả.

Ngoài những ưu điểm đã kể trên thì phương pháp vẫn còn nhược điểm là nếu vùng so sánh quá nhỏ và có cùng cường độ sáng vẫn có thể bị nhận dạng là vùng giả mạo đặc biệt thời gian thực hiện thuật toán phụ thuộc vào kích thước ảnh. Nếu ảnh có kích thước quá lớn sẽ khó thực hiện việc phát hiện.

Đề xuất nghiên cứu trong tương lai:

- Hoàn thiện chương trình với các thuật toán khác để đưa ra đánh giá mang tính thực tế hơn.
- Mở rộng hướng nghiên cứu trong phát hiện video giả mạo.

TÀI LIỆU THAM KHẢO

Tiếng Việt

[1] Giáo trình *Nhập môn xử lý ảnh số*, Lương Mạnh Bá, Nguyễn Thanh Thủy.
[2] Giáo trình *Xử lý ảnh*, TS. Đỗ Năng Toàn. TS. Phạm Việt Bình.
[3] Bài giảng môn học *Xử lý ảnh số*, Đại học dân lập Hải Phòng.

Tiếng Anh

[4] *Data hiding and Image a new stego-crypto approach*, Tamilnadu, India, 3/2006

[5], *A newImage Copyright Protection Using Digital Signature of Trading Message and Bar Code watermark*, Ji-Hong Chang, Long-Wen Chang, National Tsing University, 11/2003.

[6] *Visible Watermarking using Verifiable Digital Seal Image*, Huyncheol Park, Kwangjo Kim ,Cryptography and Information Security Oiso, Japan, 3/2001.

[7] *Detection of copy-move forgery in digital images*, J. Fridrich, D. Soukal, In Proceedings of DFRWS, 2003.