

MỤC LỤC

CHƯƠNG 1. TỔNG QUAN KỸ THUẬT GIẤU TIN TRONG ẢNH	3
1.1 Vấn đề giấu tin.....	3
1.2 Mô hình kỹ thuật giấu thông tin cơ bản	3
1.2.1 Quá trình giấu tin	4
1.2.2 Quá trình giải mã	4
1.3 Phân loại giấu tin.....	5
1.3.1 Theo cách thức tác động lên phương tiện.....	6
1.3.2 Theo các mục đích sử dụng.....	7
1.4 Mục đích sử dụng	7
1.4.1 Kỹ thuật giấu thông tin mật(steganography).....	8
1.4.2 Kỹ thuật giấu thông tin theo kiểu đánh dấu(watermarking)	8
1.5 Môi trường giấu tin	8
1.5.1 Giấu tin trong ảnh.....	8
1.5.2 Giấu tin trong audio	9
1.5.3 Giấu tin trong video	9
1.5.4 Giấu tin trong văn bản text	10
CHƯƠNG 2. CẤU TRÚC MỘT SỐ ẢNH ĐẶC TRƯNG	11
2.1 Cấu trúc ảnh bitmap.....	11
2.1.1 BMP File Header.....	11
2.1.2 Bitmap Information (DIB header)	13
2.1.3 Bảng màu (Color Palette).....	14
2.1.4 Dữ liệu ảnh.....	15
2.2 Ảnh xám	16
2.3 Cấu trúc ảnh PNG.....	16
CHƯƠNG 3. KỸ THUẬT GIẤU TIN VỚI DUNG LƯỢNG LỚN	19
3.1 Ý tưởng của thuật toán	19
3.2 Thuật toán.....	19
3.2.1 Giai đoạn giấu tin.....	19
3.2.1.1 Thủ tục nhúng ngang HEm	21
3.2.1.2 Thủ tục nhúng dọc VEm	22
3.2.2 Giai đoạn tách thông điệp	24
3.2.2.1 Thủ tục tách tin dọc VEx	25
3.2.2.2 Thủ tục tách tin ngang HEx.....	26
CHƯƠNG 4. CÀI ĐẶT THỬ NGHIỆM.....	27
4.1 Môi trường thử nghiệm.	27
4.1.1 Tập dữ liệu thử nghiệm.	27

4.2	Đo độ đánh giá PSNR	27
4.3	Một số giao diện chương trình demo	28
4.3.1	Giao diện chính của chương trình (hình 4.3)	28
4.3.2	Giao diện quá trình giấu chuỗi thông tin bất kỳ	30
4.3.3	Giao diện quá trình giấu tệp văn bản:	31
4.3.4	Giao diện tách chuỗi thông tin:	32
4.3.5	Giao diện tách tệp văn bản:	34
4.3.6	giao diện tính psnr	35
4.4	Các module cài đặt	36
4.4.1	Giấu thông tin vào trong ảnh.....	36
4.4.2	Tách thông tin	36
4.4.3	Đọc một tệp văn bản sau đó thực hiện nhúng dữ liệu	36
4.4.4	Tách tệp văn bản và ghi một tệp văn bản.....	37
4.4.5	Đổi một chuỗi kí tự ra một chuỗi nhị phân	37
4.4.6	Đổi một chuỗi nhị phân ra một chuỗi kí tự	37
4.5	Thực nghiệm, đánh giá và so sánh với kỹ thuật DE	38
4.5.1	Giấu trên 10 ảnh chuẩn (hình 4.1)	38
4.5.2	So sánh và đánh giá với kỹ thuật DE	40
	KẾT LUẬN	42
	TÀI LIỆU THAM KHẢO	43

CHƯƠNG 1. TỔNG QUAN KỸ THUẬT GIẤU TIN TRONG ẢNH

1.1 Vấn đề giấu tin

Từ trước đến nay, nhiều phương pháp bảo vệ thông tin đã được đưa ra, trong đó giải pháp dùng mật mã được ứng dụng rộng rãi nhất. Thông tin ban đầu được mã hoá, sau đó sẽ được giải mã nhờ khoá của hệ mã. Đã có rất nhiều hệ mã phức tạp được sử dụng như DES, RSA, NAPSACK..., rất hiệu quả và phổ biến.

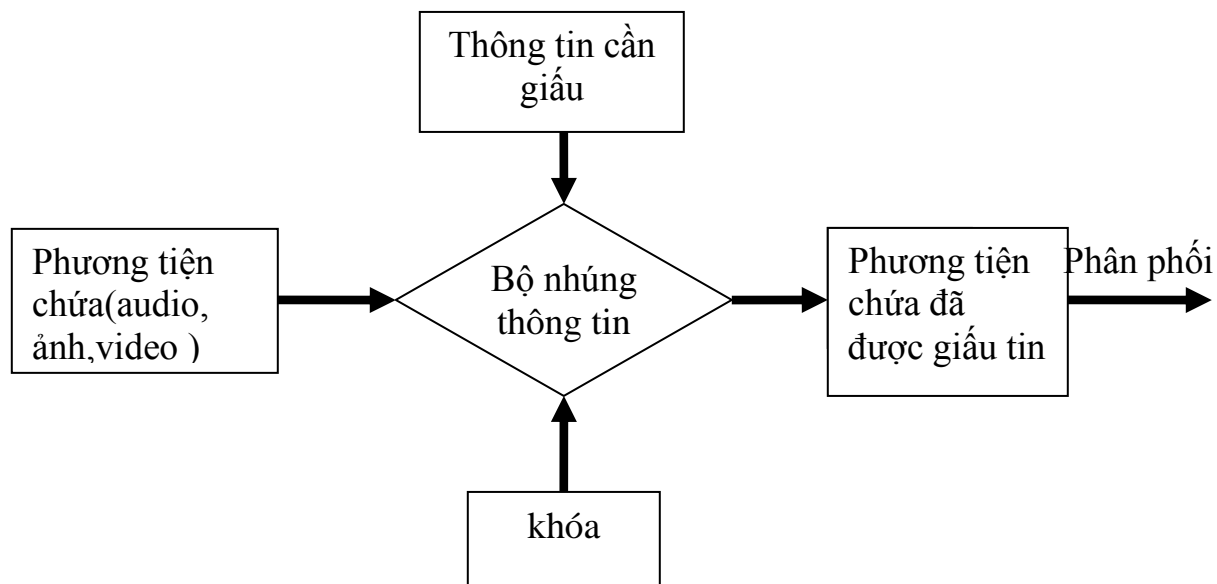
Một phương pháp mới khác đã và đang được nghiên cứu và ứng dụng mạnh mẽ ở nhiều nước trên thế giới, đó là phương pháp giấu tin (DataHiding). Giấu thông tin là kỹ thuật nhúng (embedding) một lượng thông tin số nào đó vào trong một đối tượng dữ liệu số khác. Một trong những yêu cầu cơ bản của giấu tin là đảm bảo tính chất ẩn của thông tin được giấu đồng thời không làm ảnh hưởng đến chất lượng của dữ liệu gốc.

Sự khác biệt chủ yếu giữa mã hoá thông tin và giấu thông tin là mã hoá làm cho các thông tin hiện rõ là nó có được mã hoá hay không, còn với giấu thông tin thì người ta sẽ khó biết được là có thông tin giấu bên trong.

1.2 Mô hình kỹ thuật giấu thông tin cơ bản

Giấu thông tin vào phương tiện chứa và tách lấy thông tin là hai quá trình trái ngược nhau và có thể mô tả qua sơ đồ khối của hệ thống theo hình 1.1 và 1.2

1.2.1 Quá trình giấu tin



Hình 1.1 Lược đồ quá trình giấu tin

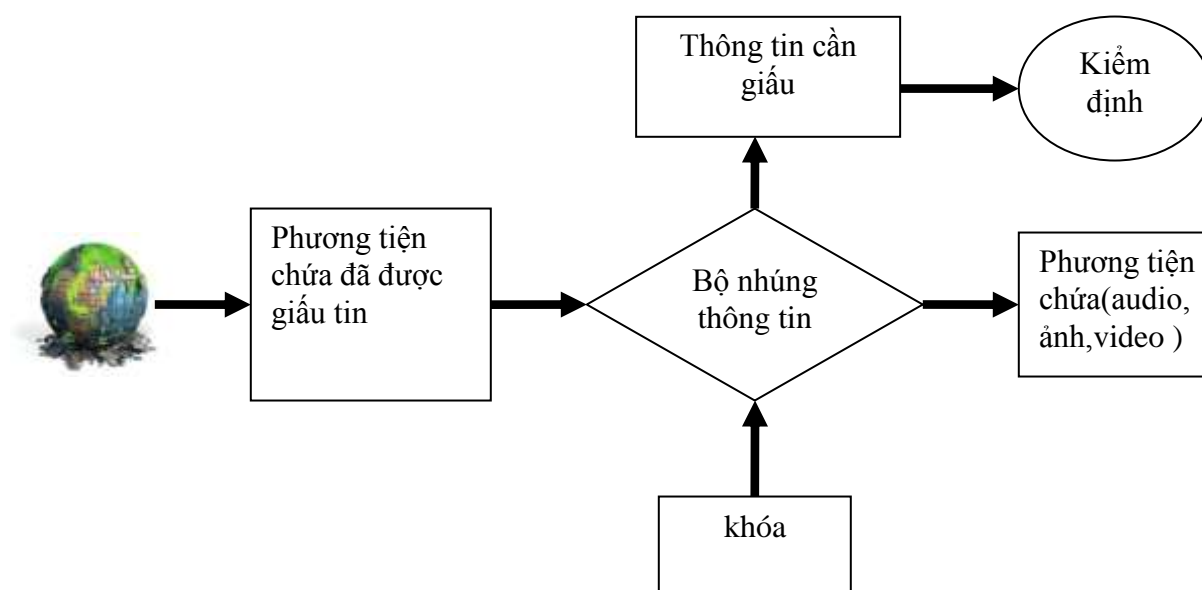
- Thông tin cần giấu tùy theo mục đích của người sử dụng, nó có thể là thông điệp (với các tin bí mật) hay các logo, hình ảnh bản quyền.
- Phương tiện chứa: các file ảnh, text, audio... là môi trường để giấu tin
- Bộ giấu thông tin: là những chương trình thực hiện việc giấu tin
- Đầu ra: là các phương tiện chứa đã có tin giấu trong đó

1.2.2 Quá trình giải mã

Tách thông tin từ các phương tiện chứa đã được giấu tin diễn ra theo trình ngược lại với đầu ra là thông tin đã được giấu vào phương tiện chứa. Phương tiện chứa sau khi tách lấy thông tin có thể được sử dụng, quản lý theo những yêu cầu khác nhau.



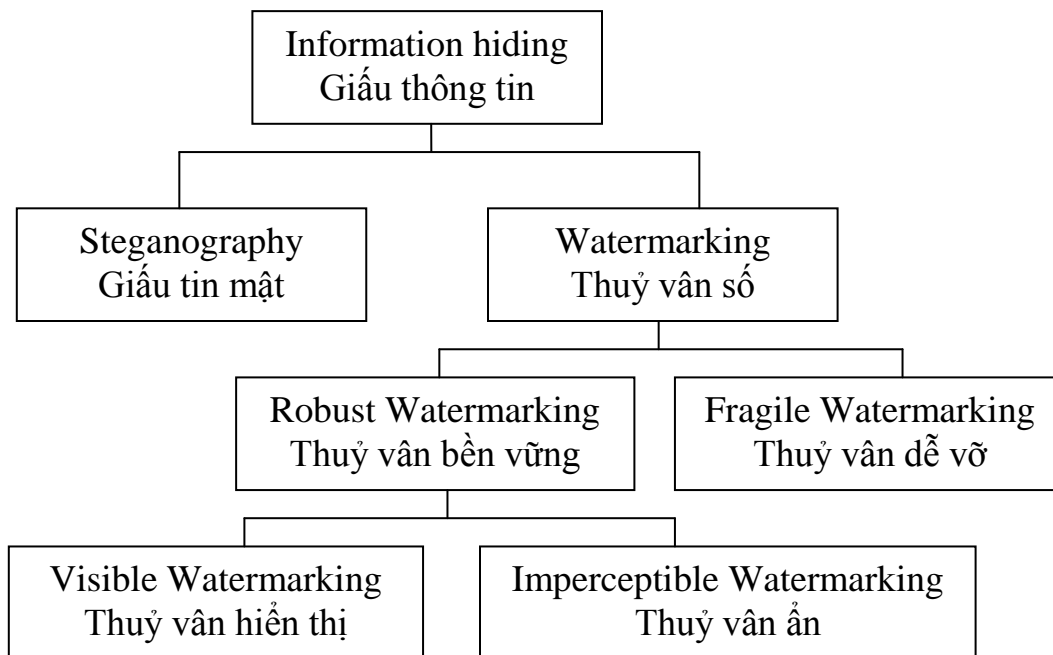
Hình 1.2 chỉ ra các công việc giải mã thông tin đã giấu. Sau khi nhận được đối tượng phương tiện chứa có giấu thông tin, quá trình giải mã được thực hiện thông qua một bộ giải mã ứng với bộ giấu thông tin cùng với khoá của quá trình giấu. Kết quả thu được gồm phương tiện chứa gốc và thông tin đã giấu. Bước tiếp theo thông tin đã giấu sẽ được xử lý kiểm định so sánh với thông tin ban đầu.



Hình 1.2 Lược đồ quá trình giải mã

1.3 Phân loại giấu tin

Do kỹ thuật giấu thông tin số mới được hình thành trong thời gian gần đây nên xu hướng phát triển chưa ổn định. Nhiều phương pháp mới, theo nhiều khía cạnh khác nhau đang và chắc chắn sẽ được đề xuất, bởi vậy một định nghĩa chính xác, một sự đánh giá phân loại rõ ràng chưa thể có được. Sơ đồ phân loại trên hình 1.3 được Fabien A. P. Petitcolas đề xuất năm 1999.



Hình 1.3 Phân loại các kỹ thuật giấu tin

Sơ đồ phân loại này như một bức tranh khái quát về ứng dụng và kỹ thuật giấu thông tin. Dựa trên việc thống kê sắp xếp khoảng 100 công trình đã công bố trên một số tạp chí, cùng với thông tin về tên và tóm tắt nội dung của khoảng 200 công trình đã công bố trên Internet, có thể chia lĩnh vực giấu tin ra làm hai hướng lớn, đó là watermarking và steganography.

1.3.1 Theo cách thức tác động lên phương tiện

Phương pháp chèn dữ liệu: Phương pháp này tìm các vị trí trong file để bị bỏ qua và chèn dữ liệu cần giấu vào đó, cách giấu này không làm ảnh hưởng gì tới sự thể hiện các file dữ liệu ví dụ như được giấu sau các ký tự EOF.

Phương pháp tạo các phương tiện chứa: Từ các thông điệp cần chuyển sẽ tạo ra các phương tiện chứa để phục vụ cho việc truyền thông tin đó, từ phía người nhận dựa trên các phương tiện chứa này sẽ tái tạo lại các thông điệp.

1.3.2 Theo các mục đích sử dụng

Giấu thông tin bí mật: đây là ứng dụng phổ biến nhất từ trước đến nay, đối với giấu thông tin bí mật người ta quan tâm chủ yếu tới các mục tiêu:

- Độ an toàn của giấu tin - khả năng không bị phát hiện của giấu tin.
- Lượng thông tin tối đa có thể giấu trong một phương tiện chứa cụ thể mà vẫn có thể đảm bảo an toàn.
- Độ bí mật của thông tin trong trường hợp giấu tin bị phát hiện.

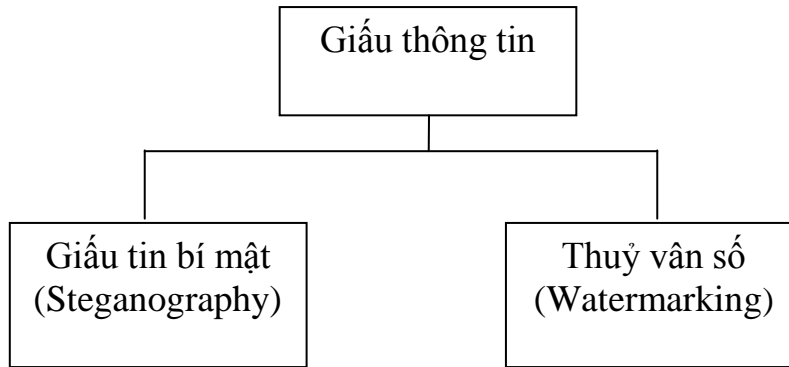
Giấu thông tin bí mật không quan tâm tới nhiều các yêu cầu bền vững của phương tiện chứa, đơn giản là bởi người ta có thể thực hiện việc gửi và nhận nhiều lần một phương tiện chứa đã được giấu tin.

Giấu thông tin thuỷ vân: do yêu cầu bảo vệ bản quyền, xác thực... nên việc giấu tin thuỷ vân có yêu cầu khác với giấu tin bí mật. Yêu cầu đầu tiên là các dấu hiệu thuỷ vân đủ bền vững trước các tấn công vô hình hay cố ý gỡ bỏ nó. Thêm vào đó các dấu hiệu thuỷ vân phải có ảnh hưởng tối thiểu (về mặt cảm nhận) đối với các phương tiện chứa. Như vậy các thông tin cần giấu càng nhỏ càng tốt.

Tuỳ theo các mục đích khác nhau thuỷ vân cũng có các yêu cầu khác nhau.

1.4 Mục đích sử dụng

Bảo mật thông tin bằng giấu tin có hai khía cạnh. Một là bảo mật cho dữ liệu đem giấu (embedded data), chẳng hạn như giấu tin mật: thông tin mật được giấu kỹ trong một đối tượng khác sao cho người khác không phát hiện được (steganography). Hai là bảo mật chính đối tượng được dùng để giấu dữ liệu vào (host data), chẳng hạn như ứng dụng bảo vệ bản quyền, phát hiện xuyên tạc thông tin (watermarking)...



Hình 1.4 Hai lĩnh vực chính của kỹ thuật giấu thông tin.

1.4.1 Kỹ thuật giấu thông tin mật(steganography)

Với mục đích đảm bảo tính an toàn và bảo mật thông tin tập trung vào các kỹ thuật giấu tin để có thể giấu được nhiều thông tin nhất. Thông tin mật được giấu kỹ trong một đối tượng khác sao cho người khác không phát hiện được.

1.4.2 Kỹ thuật giấu thông tin theo kiểu đánh dấu(giấu(watermarking))

Mục đích là để bảo vệ bản quyền của đối tượng chứa thông tin thì lại tập trung đảm bảo một số các yêu cầu như đảm bảo tính bền vững... đây là ứng dụng cơ bản nhất của kỹ thuật thủy vân số.

1.5 Môi trường giấu tin

1.5.1 Giấu tin trong ảnh

Giấu tin trong ảnh hiện đang rất được quan tâm. Nó đóng vai trò hết sức quan trọng trong hầu hết các ứng dụng bảo vệ an toàn thông tin như: nhận thực thông tin, xác định xuyên tạc thông tin, bảo vệ bản quyền tác giả... Thông tin sẽ được giấu cùng với dữ liệu ảnh nhưng chất lượng ảnh ít thay đổi và không ai biết được đằng sau ảnh đó mang những thông tin có ý nghĩa. Ngày nay, khi ảnh số đã được sử dụng rất phổ biến thì giấu thông tin trong ảnh đã đem lại nhiều những ứng dụng quan trọng trên các lĩnh vực trong đời sống xã hội.

Thông tin được giấu một cách vô hình, nó như là cách truyền thông tin mật cho nhau mà người khác không biết được.

1.5.2 Giấu tin trong audio

Khác với kỹ thuật giấu thông tin trong ảnh: phụ thuộc vào hệ thống thị giác của con người – HSV (Human Vision System), kỹ thuật giấu thông tin trong audio lại phụ thuộc vào hệ thống thính giác HAS (Human Auditory System). Bởi vì tai con người rất kém trong việc phát hiện sự khác biệt giữa các giải tần và công suất, có nghĩa là các âm thanh to, cao tần có thể che giấu đi được các âm thanh nhỏ, thấp một cách dễ dàng.

Vấn đề khó khăn đối với giấu tin trong audio là kênh truyền tin, kênh truyền hay băng thông chậm sẽ ảnh hưởng đến chất lượng thông tin sau khi giấu. Giấu thông tin trong audio đòi hỏi yêu cầu rất cao về tính đồng bộ và tính an toàn của thông tin. Các phương pháp giấu tin trong audio thường lợi dụng những điểm yếu trong hệ thống thính giác của con người.

1.5.3 Giấu tin trong video

Cũng giống như giấu thông tin trong ảnh hay trong audio, giấu tin trong video cũng được quan tâm và được phát triển mạnh mẽ cho nhiều ứng dụng như điều khiển truy cập thông tin, nhận thức thông tin, bản quyền tác giả...

Một phương pháp giấu tin trong video được đưa ra bởi Cox là phương pháp phân bố đều. Ý tưởng cơ bản của phương pháp là phân phối tin giấu dàn trải theo tần số của dữ liệu gốc. Nhiều nhà nghiên cứu đã dùng những hàm cosin riêng và những hệ số truyền sóng riêng để thực hiện việc giấu tin. Trong các thuật toán khởi nguồn, thường các kỹ thuật cho phép giấu ảnh vào trong video nhưng thời gian gần đây các kỹ thuật cho phép giấu cả âm thanh và hình ảnh vào video.

1.5.4 Giấu tin trong văn bản text

Cũng giống như giấu thông tin trong ảnh hay trong audio, giấu tin trong video cũng được quan tâm và được phát triển mạnh mẽ cho nhiều ứng dụng như điều khiển truy cập thông tin, nhận thức thông tin, bản quyền tác giả...

Một phương pháp giấu tin trong video được đưa ra bởi Cox là phương pháp phân bố đều. Ý tưởng cơ bản của phương pháp là phân phối tin giấu dàn trải theo tần số của dữ liệu gốc. Nhiều nhà nghiên cứu đã dùng những hàm cosin riêng và những hệ số truyền sóng riêng để thực hiện việc giấu tin. Trong các thuật toán khởi nguồn, thường các kỹ thuật cho phép giấu ảnh vào trong video nhưng thời gian gần đây các kỹ thuật cho phép giấu cả âm thanh và hình ảnh vào video.

CHƯƠNG 2. CẤU TRÚC MỘT SỐ ẢNH ĐẶC TRƯNG

2.1 Cấu trúc ảnh bitmap

Một tập tin BMP điển hình thông thường chứa những khối dữ liệu sau

Bảng 2.1 - Các khối dữ liệu trong một tập tin BMP

BMP File Header	Lưu trữ thông tin tổng hợp về file BMP.
Bitmap Information	Lưu trữ thông tin chi tiết về ảnh bitmap.
Color Palette	Lưu trữ định nghĩa của màu được sử dụng cho bitmap
Bitmap Data	Lưu trữ từng pixel của hình ảnh thực tế.

2.1.1 BMP File Header

Đây là khối bytes ở phần đầu tập tin, sử dụng để định danh tập tin. Ứng dụng đọc khối bytes này để kiểm tra xem đó có đúng là tập tin BMP không và có bị hư hỏng không.

Bảng 2.2 Chi tiết khối bytes tiêu đề tập tin BMP

Offset	Size	Mục đích
0000h	2 bytes	Magic number sử dụng để định nghĩa tập tin BMP: 0x42 0x4D (mã hexa của kí tự B và M). Các mục dưới đây có thể được dùng: <ul style="list-style-type: none">• BM - Windows 3.1x, 95, NT, ... etc• CI - OS/2 Color Icon• CP - OS/2 Color Pointer• IC - OS/2 Icon
0002h	4 bytes	Kích thước của tập tin BMP theo byte.
0006h	2 bytes	Dành riêng, giá trị thực tế phụ thuộc vào ứng dụng tạo ra hình ảnh.
0008h	2 bytes	Dành riêng, giá trị thực tế phụ thuộc vào ứng dụng tạo ra hình ảnh.
000Ah	4 bytes	Offset, địa chỉ bắt đầu các byte dữ liệu ảnh bitmap.

2.1.2 Bitmap Information (DIB header)

Khối bytes này nói cho ứng dụng biết các thông tin chi tiết về hình ảnh, sẽ được sử dụng để hiển thị hình ảnh trên màn hình. Bảng 2.3 miêu tả chi tiết cấu trúc tiêu đề DIB. Tất cả các giá trị được lưu trữ như là unsigned interger, trừ khi lưu ý một cách rõ ràng.

Bảng 2.3 Chi tiết khối bytes thông tin tập tin BMP

Offset	Size	Mục đích
Eh	4	Kích thước của tiêu đề (40 bytes)
12h	4	Chiều rộng bitmap tính bằng pixel (signed interger).
16h	4	Chiều cao bitmap tính bằng pixel (signed interger).
1Ah	2	Số lượng các mặt phẳng màu sắc được sử dụng. Phải được thiết lập bằng 1.
1Ch	2	Số bit trên mỗi pixel, là độ sâu màu của hình ảnh. giá trị điển hình là 1, 4, 8, 16, 24 và 32.
1Eh	4	Phương pháp nén được sử dụng. Xem bảng tiếp theo để có danh sách các giá trị có thể.
22h	4	Kích thước hình ảnh. Đây là kích thước của dữ liệu bitmap (xem bên dưới), và không nên nhầm lẫn với kích thước tập tin.
26h	4	Độ phân giải theo chiều ngang của hình ảnh (signed interger)

2Ah	4	Độ phân giải theo chiều dọc của hình ảnh (signed interger)
2Eh	4	Số lượng màu trong bảng màu.
32h	4	Số lượng các màu sắc quan trọng được sử dụng, hoặc 0 khi màu sắc nào cũng đều là quan trọng, thường bị bỏ qua.

2.1.3 Bảng màu (Color Palette)

Bảng màu xuất hiện trong tập tin BMP trực tiếp sau tiêu đề BMP và tiêu đề DIB. Vì vậy, offset là kích cỡ của tiêu đề BMP cộng với kích thước của tiêu đề DIB.

Có tất cả 2^{24} màu RGB khác nhau, nhưng các loại Bitmap sau:

- 1 bit (2 màu, hoặc chuẩn Windows là trắng-đen)
- 4 bits (16 màu)
- 8 bits (256 màu)

không thể khai thác hết, nên chỉ liệt kê các màu được dùng trong file. Mỗi màu trong bảng màu được mô tả bằng 4 bytes. (BlueByte, GreenByte, RedByte, ReservByte).

Thí dụ: bảng màu loại 1 bit chuẩn Windows có 8 bytes: 0,0,0,0,255,255,255,0 (4 bytes đầu là màu thứ 0; 4 bytes sau là màu thứ 1. Do chỉ có 0 và 1 nên mô tả mỗi điểm ảnh chỉ cần dùng 1 bit).

Tương tự như vậy, bảng màu của file 4 bits có 64 bytes, lần lượt từ màu số 0 đến màu số 15, bảng màu của file 8 bits có 1024 bytes (từ 0 đến 255). Chính vì các màu được liệt kê như vậy nên các màu trong file 1 bit, 4 bits, 8 bits được gọi là Indexed, còn 24 bits – True.

2.1.4 Dữ liệu ảnh

Dữ liệu ảnh được lưu từng điểm cho đến hết hàng ngang (từ trái sang phải), và từng hàng ngang cho đến hết ảnh (từ dưới lên trên).

Đối với mỗi điểm ảnh loại màu Indexed, ta cần 1, 4 hoặc 8 bits để đặc trưng cho điểm đang xét ứng với màu thứ mấy trong bảng màu.

Thí dụ:

Giá trị 0111 (=7) trong loại BMP 4 bits cho biết điểm đó có màu 7 (màu xám theo “chuẩn” Windows). Riêng loại 24 bits, không mô tả màu bằng thứ tự trên bảng màu (nếu liệt kê hết bảng màu của nó thì đã tốn cả Gigabyte bộ nhớ và đĩa), mà người ta liệt kê luôn giá trị RGB của 3 màu thành phần.

Thí dụ:

Trắng = {255,255,255}, Đen = {0,0,0}.

Như vậy, mỗi điểm ảnh loại 1 bit tốn 1/8 bytes (nói cách khác, 1 byte lưu được 8 điểm 1 bit), loại 4 bits - 1/2 byte, loại 8 bits - 1 byte và loại 24 bits - 3 bytes. Tuy nhiên, tính chung cả bức ảnh thì khối data không hoàn toàn tỉ lệ thuận như vậy, mà thường hơi lớn hơn một chút. Lý do chính ở chỗ người ta ngầm quy ước số bytes cần dùng cho 1 hàng ngang phải là bội của 4.

Nếu bạn có ảnh 1x1, 1 bit, thì cũng tốn 66 bytes như ảnh 32x1, 1 bit (54 cho header, 8 cho bảng màu, 4 cho 1 hàng tối thiểu). Và nếu bạn thử xoay bức hình 32x1 (vừa đúng 4 bytes dữ liệu) thành 1x32, sự lãng phí sẽ xuất hiện. Lúc đó, mỗi hàng sẽ lãng phí 31 bits, tổng cộng 32 lần như thế $31 \times 4 \text{ bytes} = 124 \text{ bytes}$.

2.2 Ảnh xám

Đơn vị tế bào của ảnh số là pixel. Tùy theo mỗi định dạng là ảnh màu hay ảnh xám mà từng pixel có thông số khác nhau. Đối với ảnh màu từng pixel sẽ mang thông tin của ba màu cơ bản tạo ra bản màu khả kiến là:

- Đỏ (R)
- Xanh lá (G)
- Xanh biển (B)

[Thomas 1892].

Trong mỗi pixel của ảnh màu, ba màu cơ bản R, G và B được bố trí sát nhau và có cường độ sáng khác nhau. Thông thường, mỗi màu cơ bản được biểu diễn bằng tám bit tương ứng 256 mức độ màu khác nhau. Như vậy mỗi pixel chúng ta sẽ có:

$2^{8 \times 3} = 2^{24}$ màu (khoảng 16.78 triệu màu).

Đối với ảnh xám, thông thường mỗi pixel mang thông tin của 256 mức xám (tương ứng với tám bit) như vậy ảnh xám hoàn toàn có thể tái hiện đầy đủ cấu trúc của một ảnh màu tương ứng thông qua tám mặt phẳng bit theo độ xám.

Trong hầu hết quá trình xử lý ảnh, chúng ta chủ yếu chỉ quan tâm đến cấu trúc của ảnh và bỏ qua ảnh hưởng của yếu tố màu sắc. Do đó bước chuyển từ ảnh màu thành ảnh xám là một công đoạn phổ biến trong các quá trình xử lý ảnh vì nó làm tăng tốc độ xử lý là giảm mức độ phức tạp của các thuật toán trên ảnh.

2.3 Cấu trúc ảnh PNG

Là một dạng hình ảnh sử dụng phương pháp nén dữ liệu mới – không làm mất đi dữ liệu gốc. PNG được tạo ra nhằm cải thiện và thay thế định dạng ảnh GIF với một định dạng hình ảnh không đòi hỏi phải có giấy phép sáng chế sử

dụng. PNG được hỗ trợ bởi thư viện tham chiếu libpng, một thư viện nền độc lập bao gồm các hàm của C để quản lý các hình ảnh PNG.

Những tập tin PNG thường có phần mở rộng là PNG và đã được gán kiểu chuẩn MIME là image/png.

Một tập tin PNG bao gồm 8 – byte kí hiệu (89 50 4E 47 0D 0A 1A) được viết trong hệ thống có cơ số 16, chứa các chữ “PNG” và 2 dấu xuống dòng, ở giữa là xếp theo số lượng của các thành phần, mỗi thành phần đều chứa thông tin về hình ảnh. Cấu trúc dựa trên các thành phần được thiết kế cho phép định dạng PNG có thể tương thích với các phiên bản cũ khi sử dụng. Các “thành phần” trong tập tin.

PNG là cấu trúc như một chuỗi các thành phần, mỗi thành phần chứa kích thước, kiểu, dữ liệu, và mã sửa lỗi CRC ngay trong nó.

Chuỗi được gán tên bằng 4 chữ cái phân biệt chữ hoa chữ thường. Sự phân biệt này giúp bộ giải mã phát hiện bản chất của chuỗi khi nó không nhận dạng được.

Với chữ cái đầu, viết hoa thể hiện chuỗi này là thiết yếu, nếu không thì ít cần thiết hơn ancillary. Chuỗi thiết yếu chứa thông tin cần thiết để đọc được tệp và nếu bộ giải mã không nhận dạng được chuỗi thiết yếu, việc đọc tệp phải được hủy.

Về cơ bản, định dạng PNG đem lại cho ta những ưu thế vượt trội hơn so với các định dạng phổ thông khác hiện nay như JPG, GIF, BMP... Những ưu thế tỏ rõ sức mạnh hơn khi được sử dụng trong môi trường đồ họa web.

- Giảm thiểu dung lượng: Trong tất cả các định dạng ảnh phổ thông hiện nay thì hình ảnh PNG có thể coi là dung lượng nhỏ nhất. Điều này rất quan trọng khi sử dụng PNG trong môi trường web.

- Độ sâu của màu: Ảnh PNG hỗ trợ đến true color 48bit màu. Trong khi đó ảnh gif chỉ ở mức 256 màu.

CHƯƠNG 3. KỸ THUẬT GIẤU TIN VỚI DUNG LƯỢNG LỚN

Kỹ thuật do Mr.P.Mohan Kumar và Dr.K.L.Shunmuganathan phát triển dựa trên phương pháp DE (Difference Expansion do Jun Tian đề xuất năm 2002). Phương pháp được đề xuất làm tăng khả năng nhúng cũng như tăng tính bảo mật của phương pháp DE ban đầu bằng cách sử dụng nhiều tầng nhúng dữ liệu mà vẫn giữ được chất lượng ảnh khả quan.

3.1 Ý tưởng của thuật toán

Thông điệp cần giấu sẽ được chia thành hai chuỗi nhỏ và nhúng theo hai chiều dọc và chiều ngang của các điểm ảnh. Cách thức để chọn điểm ảnh nhúng sẽ đề cập tại phần sau. Giả sử rằng chuỗi thông tin cần giấu đã được mã hóa.

3.2 Thuật toán

3.2.1 Giai đoạn giấu tin

Về cơ bản phương pháp đề xuất là nhúng một bit thông tin b của chuỗi thông tin cần giấu vào một cặp điểm ảnh của ảnh ban đầu O có kích thước $H \times W$ theo thứ tự quét nào đó. Cụ thể, lược đồ giấu tin này bao gồm hai giai đoạn chính: giai đoạn nhúng theo chiều ngang sử dụng thủ tục nhúng ngang HEM và giai đoạn nhúng theo chiều dọc sử dụng thủ tục nhúng dọc VEM.

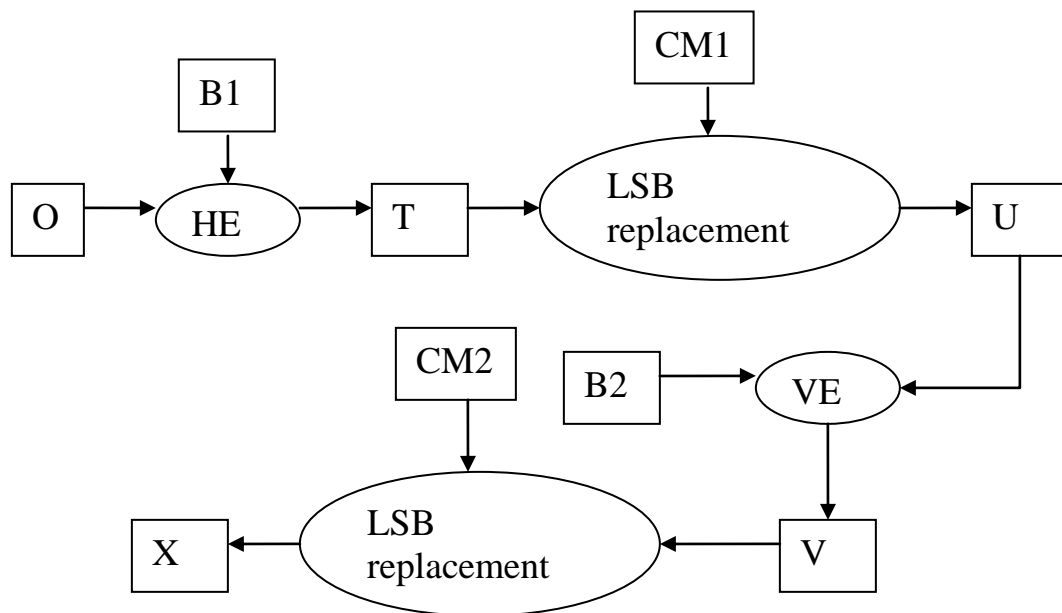
Chuỗi bit mật S có chiều dài là LS được chia làm hai chuỗi nhỏ là $S1$ và $S2$ tương ứng với chiều dài là $LS1$ và $LS2$. Chuỗi bit thông tin $B1$ được sinh ra bằng cách ghép hai chuỗi là chuỗi bit $S1$ và chuỗi bit phụ $A1$ (nghĩa là $B1 = S1 \cup A1$).

Tương tự chuỗi bit thông tin $B2$ được tạo ra bằng cách ghép chuỗi bit mật $S2$ và chuỗi bit phụ $A2$ (nghĩa là $B2 = S2 \cup A2$). Chuỗi $A1$ và $A2$ sẽ được mô tả ở phần sau. Đầu tiên các bit thông tin $B1$ sẽ nhúng theo chiều ngang vào O bằng

thủ tục HEm, để có được ảnh đầu ra T có kích thước là HxW. Sau đó bản đồ định vị của vị trí các pixel HL được sử dụng để giấu tin được nén lại thành CM1 có chiều dài là LC1, sẽ được mô tả sau, được nhúng vào T bằng kỹ thuật LSB – thay thế bit có trọng số thấp nhất. Kỹ thuật thay thế sẽ tạo ra ảnh U có kích thước là HxW.

Tiếp theo chuỗi bit thông tin B2 được nhúng theo chiều dọc vào U bằng cách sử dụng thủ tục VEm, đưa ra ảnh kết quả V, kích thước là HxW. Sau đó bản đồ định vị vị trí VL được nén lại thành CM2 có chiều dài LC2, được miêu tả ở phần sau, được nhúng vào trong V bằng kỹ thuật thay thế LSB đưa ra ảnh stego X kích thước HxW.

Tổng quan của quá trình nhúng được biểu diễn trong sơ đồ hình 3.1.



Hình 3.1 Mô hình tổng quát quá trình nhúng

3.2.1.1 Thủ tục nhúng ngang HEm

Quét các pixel của ảnh O theo chiều ngang từng dòng một (từ trái qua phải từ trên xuống dưới) nhóm các điểm ảnh kề nhau x và y thành một cặp điểm ảnh (x,y) không trùng khớp. Nếu một cặp (x,y) với y có giá trị lẻ thì cặp điểm ảnh (x,y) được định nghĩa là cặp điểm ảnh có thể nhúng ngang. Nếu không thì cặp điểm ảnh (x,y) được định nghĩa không phải là cặp điểm ảnh nhúng ngang. Gọi E1 là tập các cặp điểm ảnh có thể nhúng ngang của O có độ dài là LE1. Những cặp điểm ảnh không nhúng ngang giữ nguyên. Mỗi bit thông tin b trong B1 được nhúng ngang trong mỗi cặp điểm ảnh nhúng ngang (x,y) trong E1 tại một thời điểm bằng cách sử dụng quy tắc nhúng ngang HR.

Quy tắc nhúng ngang HR

Với mỗi cặp điểm ảnh nhúng ngang (x,y) chúng ta áp dụng các quy tắc sau:

HR1: nếu bit thông điệp $b=1$ thì cặp điểm ảnh kết quả (x_0,y_0) được tính bằng $(x_0,y_0)=(x,y)$

HR2: nếu bit thông điệp bit $b=0$ thì cặp điểm ảnh kết quả (x_0,y_0) được tính bằng $(x_0,y_0)=(x,y-1)$

Quy tắc nhúng ngang HR được lặp lại cho mỗi lần nhúng bit thông tin b của B1 trong cặp điểm ảnh gốc (x,y) thuộc E1 của O cho đến khi toàn bộ bit thông tin của chuỗi B1 được nhúng hoàn toàn vào O để được ảnh đầu ra là T. Cần chú ý rằng quy tắc nhúng ngang HR không gây ra vấn đề tràn dưới và tràn trên. Do cặp điểm ảnh sau khi nhúng (x_0,y_0) luôn rơi vào khoảng phạm vi cho phép $[0,255]$.

Chuỗi bit dữ liệu phụ A1 được thành lập như sau:

Bản đồ định vị HL (mô tả phần dưới). HL sẽ được nén lại (gọi là CM1) và có độ dài là LC1. Lấy LC1 các bit có trọng số thấp nhất của điểm ảnh từ vị trí đầu ảnh đến vị trí LC1, tập hợp của tất cả các bit đó là A1 và A1 sẽ được ghép vào sau B1 ($B1=S1 \cup A1$) và A1 được nhúng tiếp vào trong ảnh O bằng thủ tục HEM. Kết thúc ta được ảnh kết quả là ảnh T.

Bản đồ định vị nhúng ngang

Bản đồ định vị HL là một ma trận hai chiều (kích thước $H \times W/2$) ban đầu tất cả các giá trị đều bằng 0. Ta sẽ duyệt ảnh theo từng cặp điểm ảnh (x,y) từ trái qua phải từ trên xuống dưới nếu (x,y) là cặp điểm ảnh nhúng ngang thì giá trị tương ứng trong HL được chuyển thành 1 ta làm tương tự như vậy đến hết tất cả các cặp ảnh.

Nhúng bản đồ định vị

Bản đồ định vị HL sau khi nén lại thành CM1 (sử dụng kỹ thuật nén jpeg2 hoặc kỹ thuật nén carpenter) sẽ được xử lý để tách lấy thông tin mang giá trị. Từ ảnh đầu vào là ảnh T sử dụng kỹ thuật thay thế LSB (bit có trọng số thấp nhất) để nhúng bản đồ định vị, cuối cùng nhúng bộ ký tự 'eom1' để đánh dấu vị trí kết thúc của bản đồ định vị. Ảnh kết quả là ảnh U.

3.2.1.2 Thủ tục nhúng dọc VEm

Ta duyệt từng cặp điểm ảnh (u,v) từ trên xuống dưới từ trái qua phải. Nếu v là giá trị chẵn thì cặp điểm ảnh (u,v) được định nghĩa là cặp điểm ảnh nhúng dọc. Ngược lại cặp điểm ảnh (u,v) được định nghĩa là cặp điểm ảnh không nhúng dọc. Bộ các cặp điểm ảnh nhúng dọc của U là E2 có độ dài là LE2. Những cặp điểm ảnh không thuộc tập hợp trên sẽ không bị thay đổi trong quá trình nhúng dọc. Mỗi bit thông tin b của B2 được nhúng dọc vào các cặp điểm ảnh nhúng dọc

(u,v) của E2 tại một thời điểm bằng cách sử dụng quy tắc nhúng dọc VR được định nghĩa như sau.

Quy tắc nhúng dọc VR

Tại mỗi cặp điểm ảnh nhúng dọc (u,v) chúng ta sẽ áp dụng quy tắc nhúng sau:

VR1: nếu bit thông tin $b=0$ thì cặp điểm ảnh kết quả (u_0,v_0) được tính bằng $(u_0,v_0)=(u,v)$.

VR2: nếu bit thông tin $b=1$ thì cặp điểm ảnh kết quả (u_0,v_0) được tính bằng $(u_0,v_0)=(u,v+1)$.

Quy tắc nhúng dọc VR được áp dụng lặp lại để giấu mỗi bit thông tin b thuộc B2 trong mỗi cặp điểm ảnh (u,v) thuộc E2 của U cho đến khi toàn bộ chuỗi thông tin B2 nhúng vào U để được ảnh V. Cần chú ý rằng đề nhúng dọc VR không gây ra tràn dưới và tràn trên. Do cặp điểm ảnh kết quả (u_0,v_0) nằm trong phạm vi cho phép $[0,255]$. Tương tự như cách tạo ra A1 chuỗi bit dữ liệu phụ A2 được tạo ra như sau.

Bản đồ định vị dọc VL được xác định ngay khi có ảnh U. VL sẽ được nén lại thành CM2 và có độ dài là LC2. Lấy các bit có trọng số thấp nhất (LSB) của điểm ảnh từ vị trí đầu ảnh đến vị trí LC2, tập hợp của tất cả các bit đó là A2 và A2 sẽ được ghép vào sau B2 ($B2=S2 \cup A2$) và A2 được nhúng tiếp vào trong ảnh U bằng thủ tục VEm. Kết thúc ta được ảnh kết quả là ảnh V.

Bản đồ định vị nhúng dọc

Bản đồ định vị VL là một ma trận hai chiều (kích thước $H/2 \times W$) ban đầu tất cả các giá trị đều bằng 0. Ta sẽ duyệt ảnh theo từng cặp điểm ảnh (u,v) từ trên xuống dưới từ trái qua phải nếu (u,v) là cặp điểm ảnh nhúng ngang thì giá trị

tương ứng trong VL được chuyển thành 1 ta làm tương tự như vậy đến hết tất cả các cặp ảnh.

Nhúng bản đồ định vị

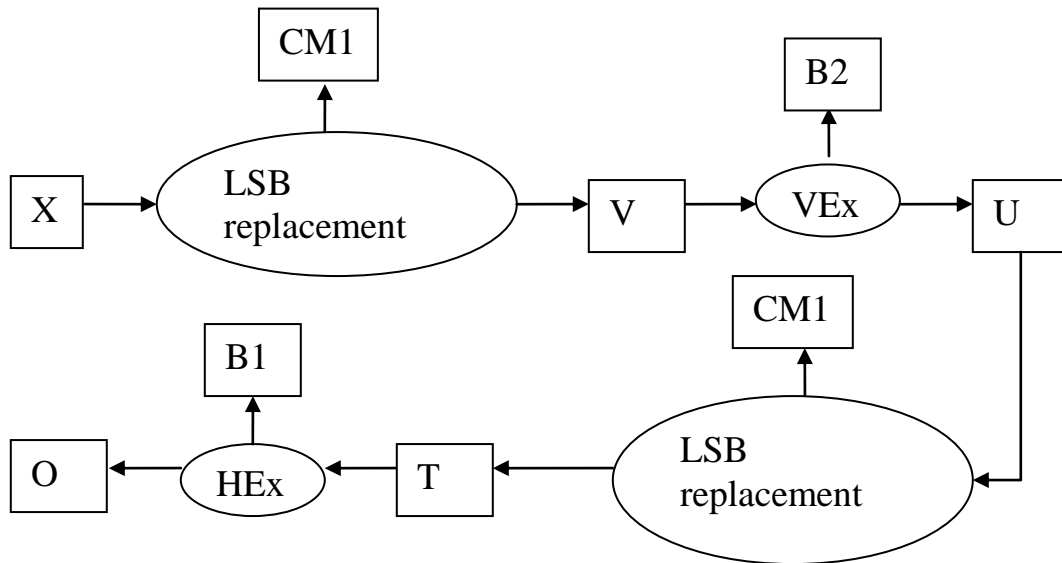
Bản đồ định vị sau khi nén lại CM2 sẽ được xử lý sau này để tách lấy thông tin mang giá trị. Từ ảnh đầu vào V sử dụng kỹ thuật thay thế LSB (thay thế bit có trọng số thấp nhất) để nhúng bản đồ định vị CM2, cuối cùng nhúng bộ ký tự 'eom2' để đánh dấu vị trí kết thúc của bản đồ định vị. Ảnh kết quả là ảnh X.

3.2.2 Giai đoạn tách thông điệp

Giai đoạn tách thông điệp là giai đoạn ngược lại của quá trình nhúng. Giai đoạn này bao gồm hai giai đoạn chính: giai đoạn tách theo chiều dọc sử dụng thủ tục tách tin dọc VEx và giai đoạn tách theo chiều ngang sử dụng thủ tục tách tin ngang HEx. Cụ thể, thứ nhất việc nhúng CM2 được lấy lại bằng cách sử dụng kỹ thuật tách LSBs của LC2 những điểm ảnh đầu tiên trong ảnh X ở nơi nhận (nhờ phát hiện bộ ký tự 'eom2'). Sau khi giải nén CM2 thu lại được bản đồ định vị VL được sử dụng để xác định những cặp điểm ảnh nhúng dọc E2 của X. Tiếp theo A2 được lấy ra từ LC2 những cặp điểm ảnh cuối trong E2 của ảnh X bằng quy tắc giải nén dọc VX. Sau đó những cặp điểm ảnh LC2 đầu tiên của X được thay thế bằng A2 để thu lại ảnh V. Từ ảnh V ta tách B2 đã được nhúng để thu hồi ảnh U bằng cách dùng thủ tục giải nén dọc VX. Sau đó CM1 đã nhúng sẽ được thu hồi bằng kỹ thuật tách LSB của LC1 điểm ảnh đầu tiên trong ảnh U (nhờ phát hiện bộ ký tự kết thúc 'eom1'). Việc giải nén thu lại CM1 nhờ đó xác định những cặp điểm ảnh nhúng ngang thuộc tập E1 của U.

Tiếp theo A1 được tách từ những cặp điểm ảnh LC1 cuối trong tập E1 của U bằng cách sử dụng quy tắc tách ngang Hx. Sau đó những cặp điểm ảnh LC1 đầu tiên của U được thay thế với A1 thu được ảnh T. Từ ảnh T thu hồi B1 đã

nhúng sẽ phục hồi được ảnh O ban đầu bằng cách sử dụng thủ tục tách ngang HEx. LS1 bit đầu tiên của B1 là chuỗi bit mật S1. LS2 bit đầu tiên của B2 là chuỗi bit mật S2. Việc tách chuỗi bit mật S1 và S2 và được ghép nối tạo thành chuỗi bit mật ban đầu $S(S=S1||S2)$. Các tổng quan đưa ra về quá trình tách thông điệp được mô tả trong đồ thị hình 3.2.



Hình 3.2 Mô hình tổng quát quá trình thu hồi thông tin

3.2.2.1 Thủ tục tách tin dọc VEx

Quét ảnh V theo thứ tự giống như quá trình nhúng dọc tạo ra nhóm 2 điểm ảnh kề nhau u_0 và v_0 — hai điểm ảnh nằm trên hai hàng kề nhau trên cùng một cột- trong một cặp điểm ảnh (u_0, v_0) . Việc tách bản đồ định vị sẽ sử dụng để xác định xem một cặp điểm ảnh (u_0, v_0) có thuộc tập E2 (cặp điểm ảnh nhúng dọc) hay không. Sử dụng quy tắc giải nén dọc VX cho các điểm ảnh thuộc tập E2.

Quy tắc tách dọc VX

Nếu v_0 là một giá trị chẵn, thì bit thông tin b trong B2 được xác định là $b=0$ và cặp điểm ảnh (u, v) được phục hồi là $(u, v)=(u_0, v_0)$.

Ngược lại, nếu (u_0, v_0) thuộc E_2 , thì bit thông tin trong B_2 được tách là $b=1$ và cặp điểm ảnh (u, v) được phục hồi là $(u, v) = (u_0, v_0 - 1)$.

Nếu (u_0, v_0) không thuộc E_2 thì không có bit thông tin nào được tách ra. Cặp điểm ảnh (u, v) được khôi phục $(u, v) = (u_0, v_0)$

Ảnh kết quả của thủ tục giải nén dọc là ảnh U .

3.2.2.2 Thủ tục tách tin ngang HEx

Quét ảnh T theo thứ tự giống với quá trình nhúng ngang để tạo nhóm 2 điểm ảnh kề nhau x_0 và y_0 — hai điểm ảnh nằm trên hai cột kề nhau trên cùng một dòng— trong mỗi cặp điểm ảnh (x_0, y_0) . Việc tách được bản đồ định vị được sử dụng để xác định xem một cặp điểm ảnh (x_0, y_0) có thuộc tập E_1 (cặp điểm ảnh nhúng ngang). Sử dụng quy tắc giải nén ngang HX cho các điểm ảnh thuộc tập E_1 .

Quy tắc tách ngang HX

Nếu y_0 là một giá trị lẻ, thì bit thông tin b trong B_1 được xác định là $b=1$ và cặp điểm ảnh (x, y) được phục hồi là $(x, y) = (x_0, y_0)$.

Ngược lại nếu (x_0, y_0) thuộc E_1 , thì bit thông tin b trong B_1 được tách là $b=0$ và cặp điểm ảnh (x, y) được phục hồi là $(x, y) = (x_0, y_0 + 1)$. Nếu (x_0, y_0) không thuộc E_1 thì không có bit thông tin nào được tách ra. Cặp điểm ảnh (x, y) được khôi phục $(x, y) = (x_0, y_0)$

Ảnh kết quả của thủ tục giải nén dọc là ảnh O .

CHƯƠNG 4. CÀI ĐẶT THỬ NGHIỆM

4.1 Môi trường thử nghiệm.

Môi trường cài đặt là chương trình lập trình matlab r2007b. Dựa trên cơ sở phần cứng chip celeron(r) 2.53, ram 256 chạy hệ điều hành window xp service pack 2.

4.1.1 Tập dữ liệu thử nghiệm.

Tập dữ liệu thử nghiệm gồm 10 ảnh chuẩn kích thước 512x512 [6] trong Hình 4.1



Hình 4.1 10 ảnh chuẩn

4.2 Đo độ đánh giá PSNR

Chất lượng ảnh sau khi tin giấu được đánh giá thông qua giá trị của tỷ số PSNR (Peak Signal to Noise Ratio) tỷ số tín hiệu đỉnh trên nhiễu.

Nó được định nghĩa thông qua bình phương trung bình lỗi MSE (mean squared error) cho hai hình ảnh I và K có kích thước HxW:

$$PSNR = 10\log_{10} \frac{255^2}{MSE} \text{ (dB)}$$

Trong đó MSE được định nghĩa:

$$MSE = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W (O_{ij} - X_{ij})^2$$

Khi hai hình ảnh giống hệt nhau, MSE sẽ bằng 0 và PSNR đi đến vô hạn.

4.3 Một số giao diện chương trình demo

4.3.1 Giao diện chính của chương trình (hình 4.2)

Bao gồm các chức năng:

Giấu tin

Giấu chuỗi kí tự: thực hiện giấu một chuỗi thông điệp do người dùng nhập vào.

Giấu tệp văn bản: thực hiện giấu một tệp văn bản do người dùng chọn.

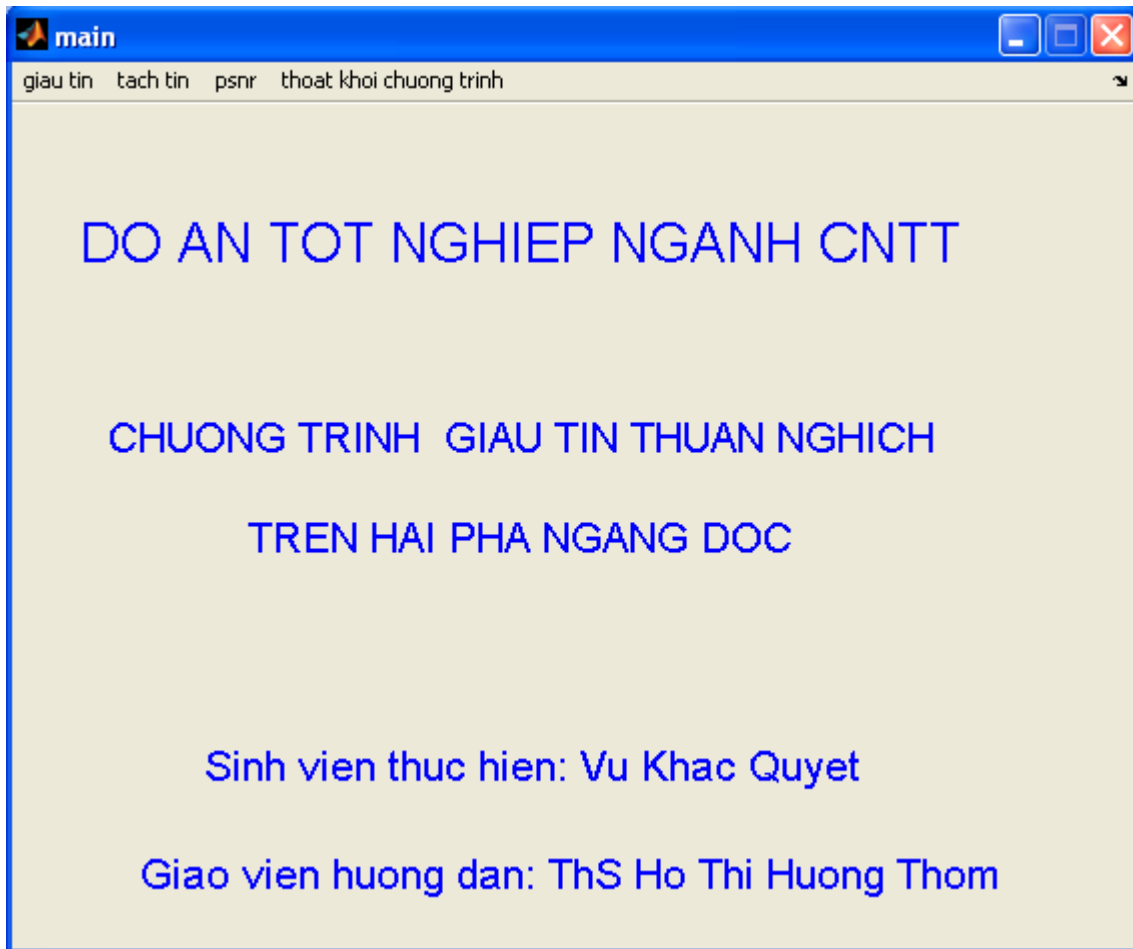
Tách tin:

Tách chuỗi kí tự: thực hiện tách một chuỗi thông điệp từ ảnh đã được giấu bởi chức năng giấu chuỗi kí tự.

Tách tệp văn bản: thực hiện tách một tệp văn bản từ ảnh đã được giấu bởi chức năng giấu tệp văn bản.

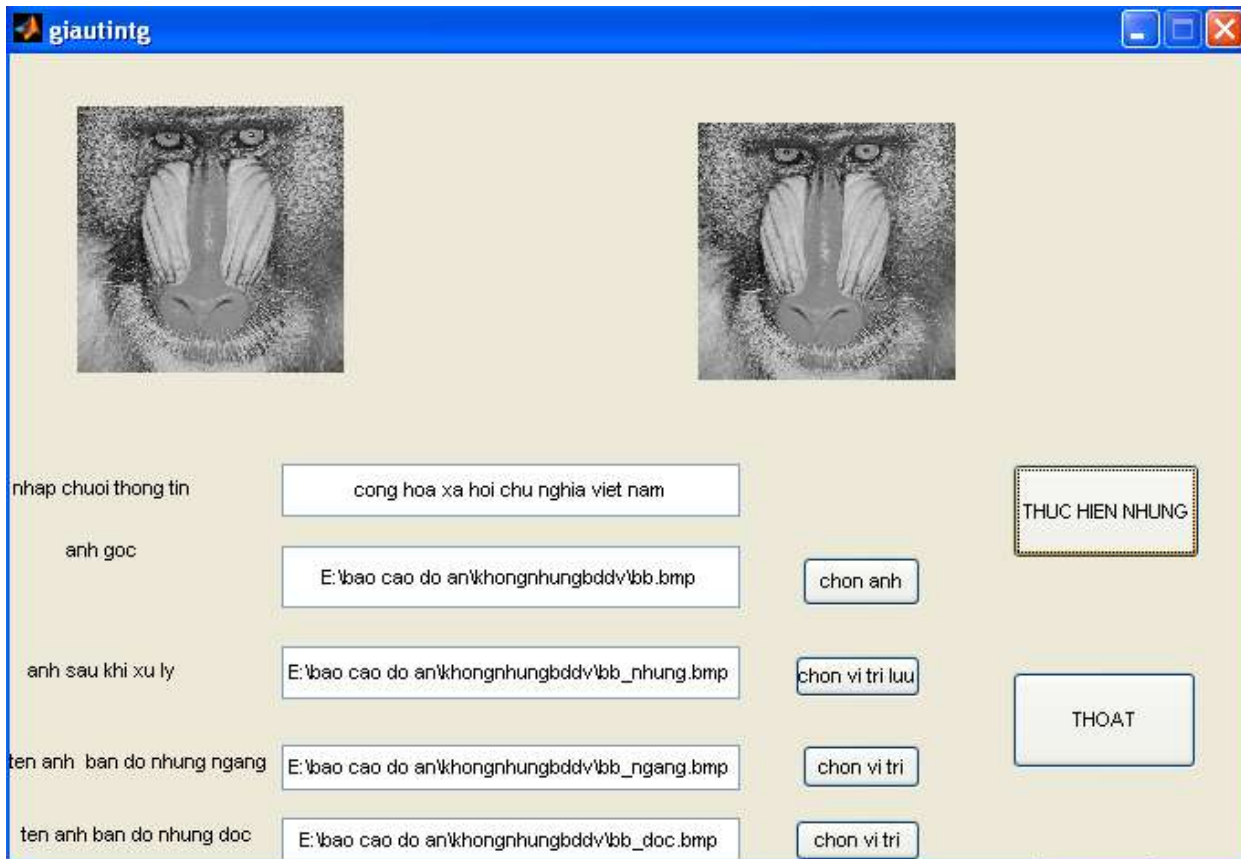
PSNR:

Tính toán PSNR.



Hình 4.2 Giao diện chính của chương trình

4.3.2 Giao diện quá trình giấu chuỗi thông tin bất kỳ



Hình 4.3 Giao diện nhúng chuỗi thông điệp

Ô nhập dữ liệu:

Nhập chuỗi thông tin: Hãy nhập chuỗi thông tin muốn gửi.

Ảnh gốc: Nhấn vào ô chọn ảnh để lựa chọn bức ảnh gốc để làm phương tiện nhúng chuỗi thông tin.

Ảnh xử lý :Nhấn ô chọn vị trí lưu để nhập tên bức ảnh sau khi đã nhúng thông tin.

Tên ảnh bản đồ nhúng ngang:Nhấn ô chọn vị trí để lưu tên bản đồ định vị nhúng ngang.

Tên ảnh bản đồ nhúng dọc: Nhấn ô chọn vị trí để lưu tên bản đồ định vị nhúng dọc.

Nút lệnh

Các nút lệnh bên cạnh các ô nhập dữ liệu có nhiệm vụ mở các cửa sổ để người dùng lựa chọn.

Nút thực hiện nhúng để chương trình thực hiện nhúng thông điệp vào trong ảnh gốc.

Nút thoát có nhiệm vụ thoát khỏi modul nhúng thông điệp

4.3.3 Giao diện quá trình giấu tệp văn bản:



Hình 4.4 Giao diện quá trình giấu tệp văn bản

Ô nhập dữ liệu:

Tệp cần nhúng: tệp văn bản cần giấu.

Ảnh gốc: bức ảnh gốc.

Tên ảnh sau khi nhúng: tên bức ảnh sau khi được nhúng văn bản.

Bản đồ nhúng ngang: tên bản đồ định vị nhúng ngang.

Bản đồ nhúng dọc: tên bản đồ định vị nhúng dọc.

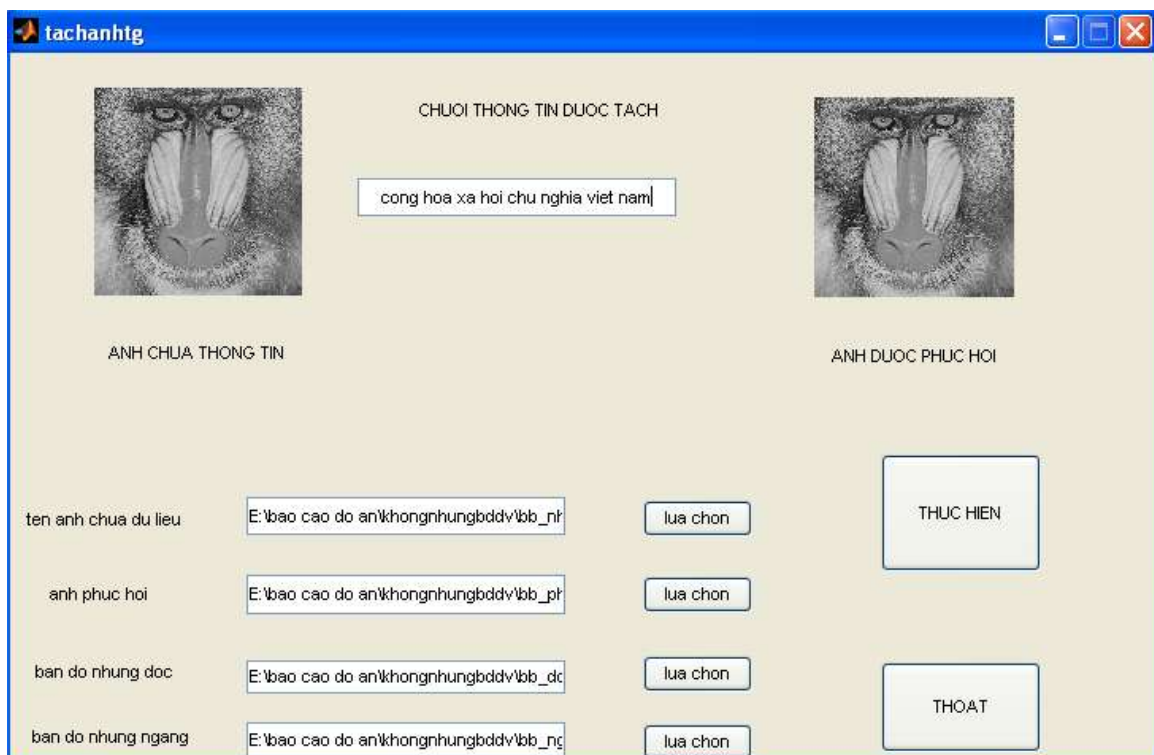
Nút lệnh:

Các nút lệnh bên cạnh các ô nhập dữ liệu có nhiệm vụ mở cửa sổ để người dùng lựa chọn.

Thực hiện: Sẽ thực hiện quá trình giấu tin.

Thoát: Thoát khỏi giao diện giấu tệp.

4.3.4 Giao diện tách chuỗi thông tin:



Hình 4.5 Giao diện tách chuỗi thông tin

Ô nhập dữ liệu:

Tên ảnh chứa dữ liệu: tên bức ảnh chứa thông tin.

Ảnh phục hồi: tên bức ảnh sẽ được lưu sau khi khi phục hồi.

Bản đồ nhúng dọc: tên bản đồ định vị nhúng dọc.

Bản đồ định vị nhúng ngang: tên bản đồ định vị nhúng ngang.

Nút thực hiện:

Các nút lệnh bên cạnh các ô nhập dữ liệu có nhiệm vụ mở các cửa sổ để người dùng lựa chọn.

Nút thực hiện để chương trình tách thông tin từ ảnh gốc.

Nút thoát để thoát khỏi giao diện tách chuỗi thông tin.

Kết quả

chuỗi thông tin tách ra sẽ được trả về ở ô chuỗi thông tin được tách ra.

4.3.5 Giao diện tách tệp văn bản:



Hình 4.6 Giao diện tách tệp văn bản

Ô nhập dữ liệu:

Ảnh chứa thông tin: tên bức ảnh chứa tệp văn bản.

Bản đồ định vị dọc: tên ảnh bản đồ định vị dọc.

Bản đồ định vị ngang: tên ảnh bản đồ định vị ngang.

Ảnh sau khi lấy thông tin: tên bức ảnh sau khi tách thông tin và phục hồi lại nguyên ảnh.

Tệp văn bản: tên văn bản sau khi tách ra khỏi ảnh.

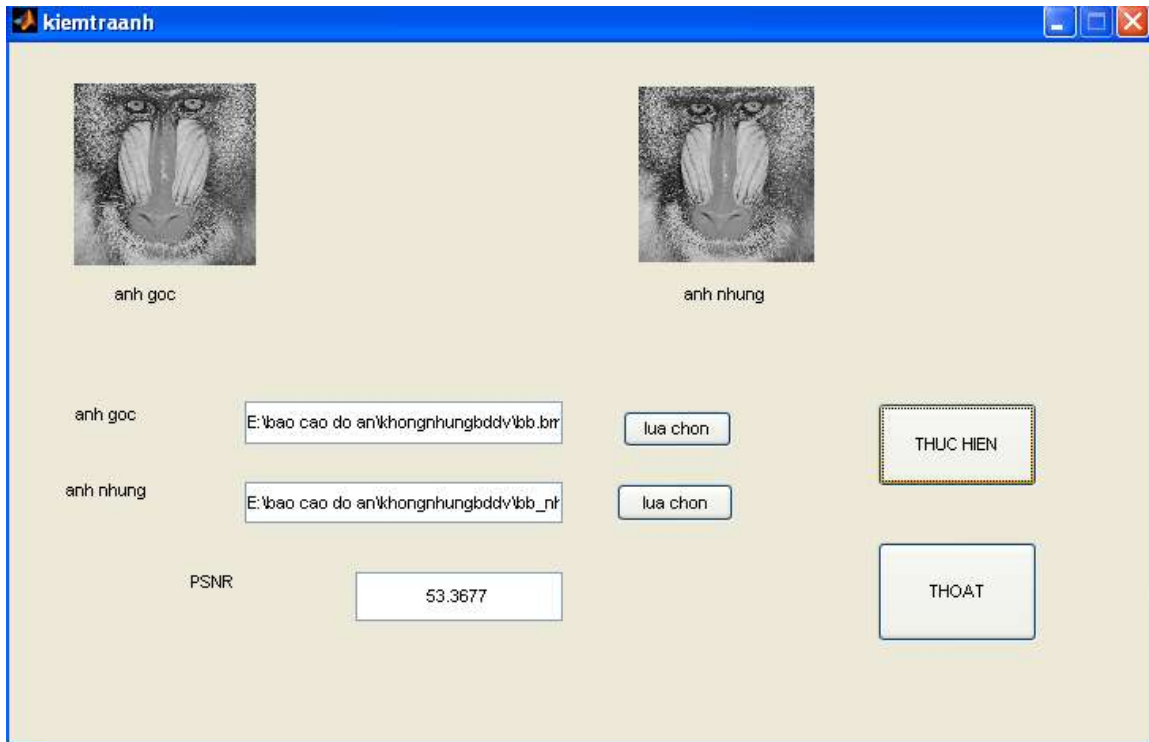
Nút lệnh

Các nút lệnh bên cạnh các ô nhập dữ liệu có nhiệm vụ mở các cửa sổ để người dùng lựa chọn.

Nút thực hiện để chương trình tách văn bản từ ảnh gốc.

Nút thoát để thoát khỏi giao diện tách tệp văn bản.

4.3.6 giao diện tính psnr



Hình 4.7 Giao diện tính psnr

Ô nhập thông tin

Ảnh gốc: tên bức ảnh trước khi thực hiện nhúng thông tin.

Ảnh nhúng: tên bức ảnh sau khi nhúng thông tin.

Nút lệnh:

Các nút lệnh bên cạnh các ô nhập dữ liệu có nhiệm vụ mở các cửa sổ để người dùng lựa chọn.

Nút thực hiện để chương trình tính psnr.

Nút thoát để thoát khỏi giao diện tính psnr.

4.4 Các module cài đặt

4.4.1 Chức năng: Giấu thông tin vào trong ảnh.

Các tham số đầu vào:

- x: chuỗi thông tin đầu vào.
- y: ảnh gốc .
- ten moi: Tên của ảnh kết quả sau khi giấu tin.
- Tenbddvngang: tên của bản đồ định vị nhúng ngang
- Tenbddvdoc: tên của bản đồ định vị nhúng dọc

4.4.2 Chức năng tách thông tin

Tham số đầu vào:

- tenanh: tên ảnh gốc.
- tenanhphuchoi: tên ảnh sau khi lấy thông tin và phục hồi nguyên gốc
- tenbddoc: tên ảnh bản đồ định vị nhúng dọc.
- ten bdnngang: tên ảnh bản đồ định vị nhúng ngang.

Tham số ra: chuỗi thông tin kết quả

4.4.3 Chức năng: Đọc một tệp văn bản sau đó thực hiện nhúng dữ liệu

Tham số đầu vào:

- tenfile: Tên tệp văn bản cần đọc.
- y: bức ảnh gốc.
- tenmoi: tên của bức ảnh sau khi đã nhúng tệp văn bản.
- tenbdngang: tên của ảnh bản đồ định vị nhúng ngang.
- Tenbddoc: tên của ảnh bản đồ định vị nhúng dọc.

Đầu ra là bức ảnh đã được nhúng tệp văn bản

4.4.4 Chức năng: tách tệp văn bản và ghi một tệp văn bản

Tham số đầu vào:

- tenanh: tên ảnh chứa tệp văn bản.
- tenanhphuchoi: tên bức ảnh sau khi tách thông tin và phục hồi
- tenfie: tên của tệp văn bản sau khi được tách ra.
- tenbdoc: tên của ảnh bản đồ định vị nhúng dọc.
- tenbdngang: tên của ảnh bản đồ định vị nhúng ngang.

4.4.5 Chức năng: Đổi một chuỗi kí tự ra một chuỗi nhị phân

Tham số đầu vào:

- s: Chuỗi kí tự cần đổi.

Tham số đầu ra:

- b: Chuỗi bit kết quả.

4.4.6 Chức năng: Đổi một chuỗi nhị phân ra một chuỗi kí tự

Tham số đầu vào:

- b: chuỗi bit cần đổi.

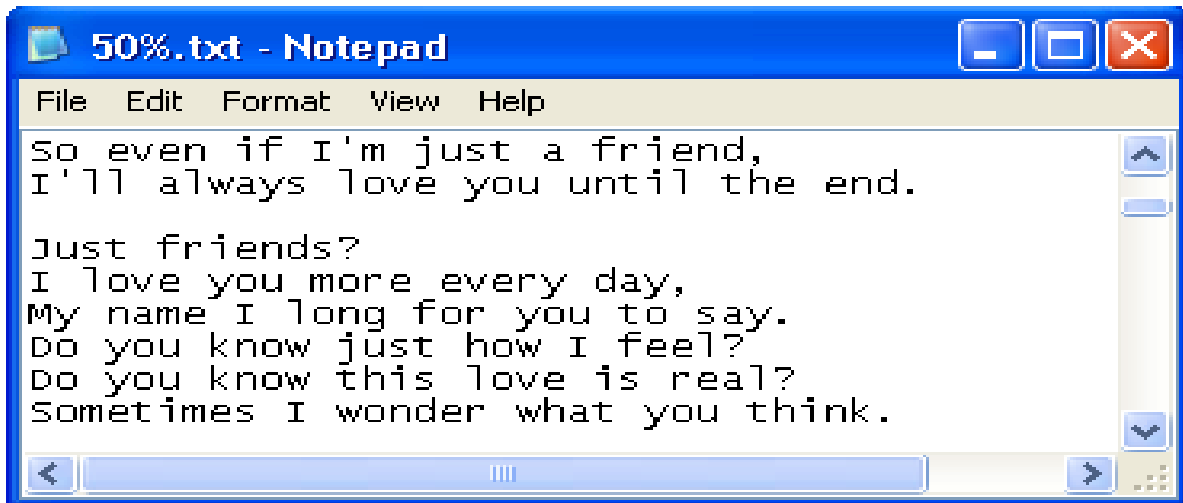
Tham số đầu ra:

- S: chuỗi kí tự kết quả.

4.5 Thực nghiệm, đánh giá và so sánh với kỹ thuật DE

4.5.1 Giấu trên 10 ảnh chuẩn (hình 4.1)

Sử dụng kỹ thuật giấu tin với dung lượng lớn để nhúng tệp txt hình 4.8 .



Hình 4.8 tệp văn bản có dung lượng 67232 bit

Các ảnh trước và sau khi nhúng



Hình 4.9 ảnh trước và sau khi nhúng

4.5.2 So sánh và đánh giá với kỹ thuật DE

Tương tự như vậy ta sử dụng phương pháp DE để nhúng tệp trên trong 10 ảnh chuẩn hình 4.1 ta thu được kết quả như bảng 4.1 kết quả tính psnr.

Bảng 4.1 Kết quả tính psnr

Kiểu nhúng Ảnh	PSNR	
	DE	GIẤU TIN VỚI DUNG LƯỢNG LỚN
Anh1.bmp	37.5255	54.2344
Anh2.bmp	28.3151	53.7832
Anh3.bmp	36.3264	54.426
Anh4.bmp	41.5941	54.4896
Anh5.bmp	36.4263	54.6144
Anh6.bmp	35.0179	54.5821
Anh7.bmp	38.822	54.5245
Anh8.bmp	36.3985	54.5969
Anh9.bmp	36.5056	54.6128
Anh10.bmp	39.2808	54.6258
Trung bình	36.6212	54.44897

Nhận xét:

Qua bảng đánh giá trên ta có thể nhận thấy rằng kỹ thuật giấu tin với dung lượng lớn được đề xuất trong bản báo cáo vượt trội hơn so với phương pháp DE trước đây ($\text{psnr} > 54$) trong khi phương pháp DE cũ $\text{psnr} > 36$.

Và quan trọng hơn sau khi cải tiến kỹ thuật DE sang kỹ thuật giấu tin trên hai pha ngang dọc thì đã không còn xảy ra vấn đề tràn và hiện tượng muối hạt tiêu ở ảnh sau khi nhúng vì điểm ảnh sau khi được mang đi xử lý luôn luôn đảm bảo trong phạm vi $[0 \ 255]$.

Bên cạnh những ưu điểm thì kỹ thuật giấu tin trên hai pha ngang dọc cũng có nhược điểm là không thể tính chính xác được số lượng bit có thể nhúng vào trong ảnh vì pha nhúng dọc phụ thuộc vào pha nhúng ngang trước đó.

KẾT LUẬN

Đồ án đã thực hiện nhiệm vụ:

- Trình bày tổng quan kỹ thuật giấu tin
- Nghiên cứu cấu trúc ảnh BMP, ảnh xám và ảnh PNG
- Nghiên cứu kỹ thuật giấu tin với dung lượng thông điệp lớn

Kỹ thuật giấu tin với dung lượng lớn đã giải quyết rất tốt vấn đề tràn trên tràn dưới và hiện tượng muối hạt tiêu ở ảnh sau khi nhúng thông tin của phương pháp DE, chất lượng ảnh sau khi nhúng tốt tăng khả năng bảo mật dữ liệu, tránh sự nghi ngờ của đối tượng đột nhập. Bên cạnh đó thì thuật toán giấu tin của kỹ thuật giấu tin với dung lượng lớn cũng tương đối đơn giản, dễ hiểu.

Ngoài những ưu điểm đã kể trên thì phương pháp vẫn còn nhược điểm mà một phương pháp DE gặp phải đó là dung lượng thông tin có thể nhúng trong ảnh phụ thuộc rất lớn vào công cụ nén được sử dụng để nén bản đồ định vị. Nếu công cụ nén đạt hiệu suất nén cao thì dung lượng nhúng lớn và ngược lại.

Do còn nhiều hạn chế về hiểu biết nên đề tài này không tránh khỏi những thiếu sót, vì vậy em rất mong nhận được sự đóng góp ý kiến của các thầy cô và các bạn để đồ án được hoàn thiện hơn.

Em xin chân thành cảm ơn!

TÀI LIỆU THAM KHẢO

- [1] Nguyễn Xuân Huy, Trần Quốc Dũng, *Giáo trình giấu tin và thủy vân ảnh*, Trung tâm thông tin tư liệu, TTKHTN-CN 2003
- [2] Đỗ Lâm Hoàng, Đồ án tốt nghiệp, ngành Công nghệ thông tin, năm 2010
- [3] Mặc Như Hiền, Đồ án tốt nghiệp ngành CNTT, năm 2009
- [4] Jun Tian, Reversible Watermarking by Difference Expansion, Multimedia and Security Workshop at ACM Multimedia 2002, Dec-02
- [5] Mr. P. Mohan Kumar & Dr. K. L. Shunmuganathan “A reversible high embedding capacity data hiding technique for hiding secret data in images”
- [6] <http://sipi.usc.edu/database/database.cgi?volume=misc>