MỤC LỤC

CHƯƠNG 1. TỔNG QUAN KĨ THUẬT GIẤU TIN

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.

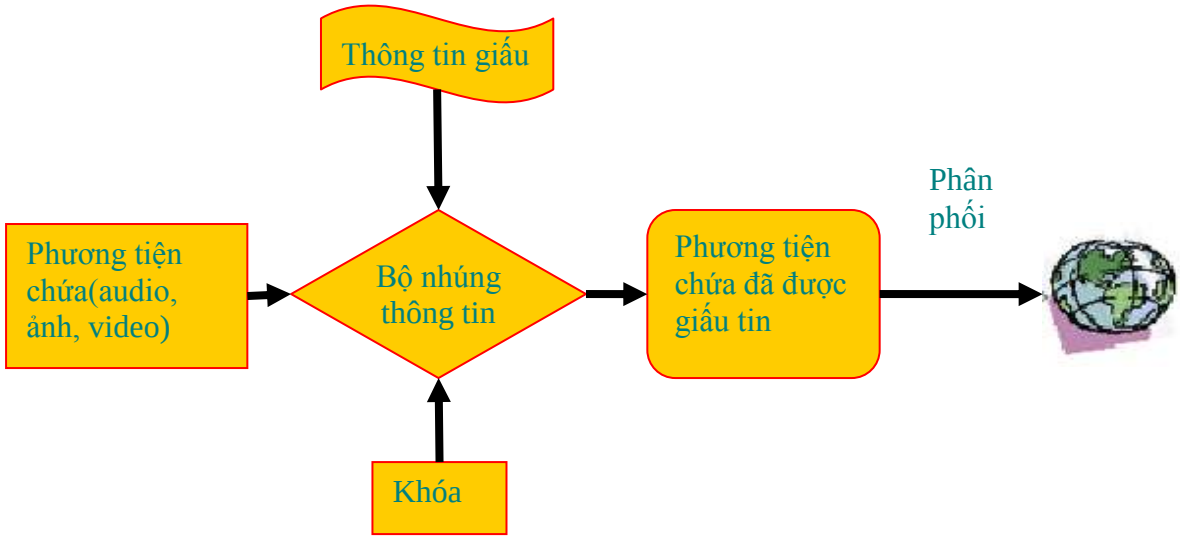
Một phương pháp mới khác đã và đang được nghiên cứu và ứng dụng mạnh mẽ ở nhiều nước trên thế giới, đó là phương pháp giấu tin (DataHiding). *Giấu thông tin là kỹ thuật nhúng (embedding) một lượng thông tin số nào đó vào trong một đối tượng dữ liệu số khác.* Một trong những yêu cầu cơ bản của giấu tin 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 như sau

1.2.1 Quá trình giấu tin

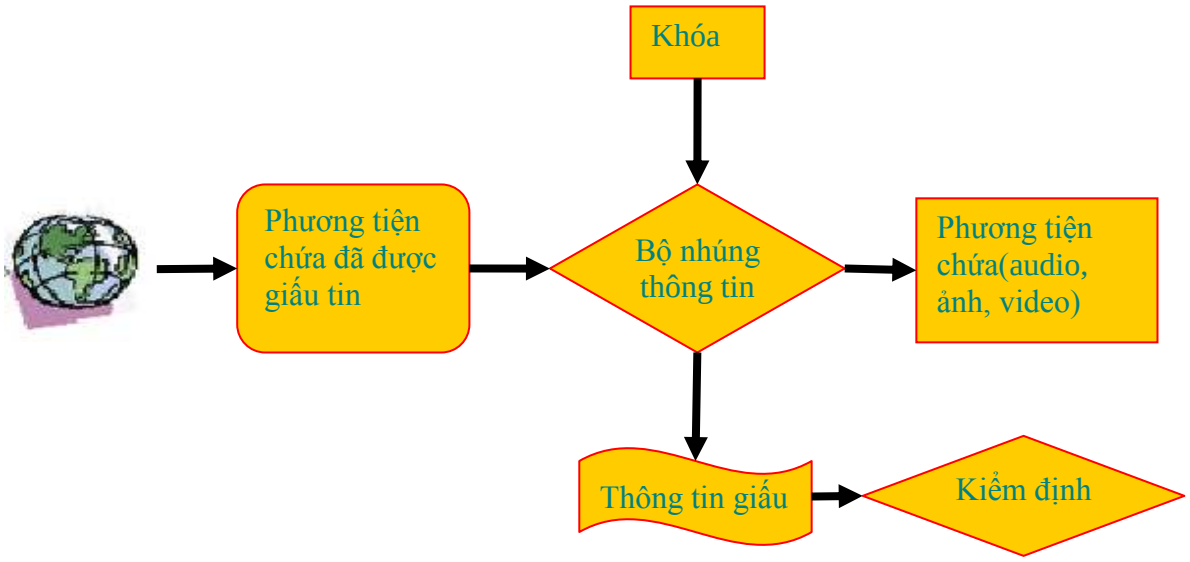


Hình 1.1 - Lược đồ chung cho quá trình giấu tin

- Thông tin cần giấu tuỳ 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 quy 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.

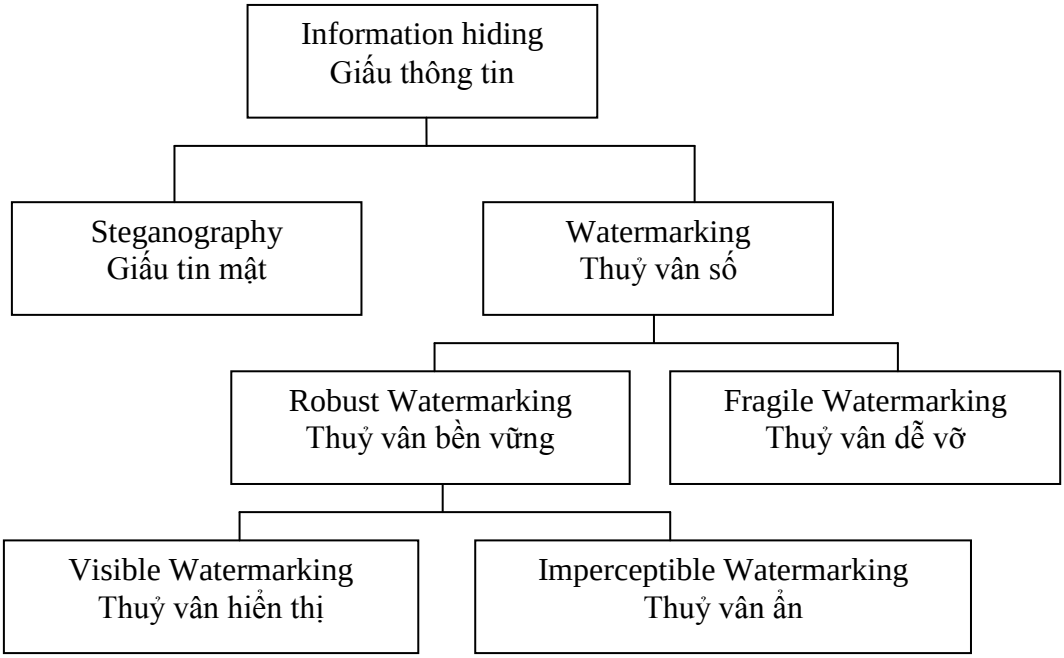


Hình 1.2 - Lược đồ cho quá trình giải mã

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 - Một cách 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 các phương tiện

Phương pháp chèn dữ liệu: Phương pháp này tìm các vị trí trong file dễ 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.

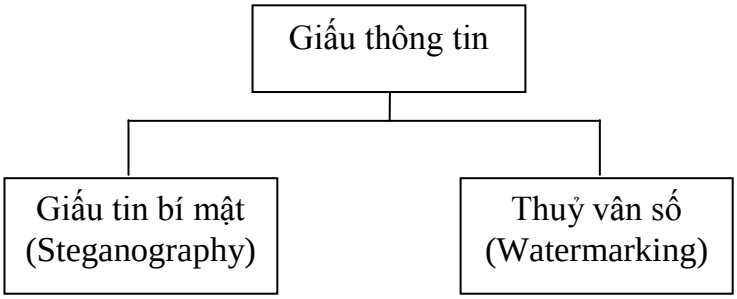
Giấu thông tin thuỷ vân: do yêu cầu bảo vệ bản quyền, xác thực… nên việc giấu tin thuỷ vân có yêu cầu khác với giấu tin bí mật. Yêu cầu đầu tiên là các dấu hiệu thuỷ vân đủ bền vững trước các tấn công vô hình hay cố ý gỡ bỏ nó. Thêm vào đó các dấu hiệu thuỷ vân phải có ảnh hưởng tối thiểu (về mặt cảm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

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 bí 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 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 thuỷ 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 này, 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 thông tin trong văn bản dạng text

Giấu thông tin trong văn bản dạng text thì khó thực hiện hơn do có ít thông tin dư thừa, để làm được điều này người ta phải biết khéo léo khai thác các dư thừa tự nhiên của ngôn ngữ. Một cách khác là tận dụng các định dạng văn bản (mã hoá thông tin vào khoảng cách giữa các từ hay các dòng văn bản).

Kỹ thuật giấu tin đang được áp dụng cho nhiều loại đối tượng chứ không riêng gì dữ liệu đa phương tiện như ảnh, audio, video. Gần đây đã có một số nghiên cứu giấu tin trong cơ sở dữ liệu quan hệ, các gói IP truyền trên mạng chắc chắn sau này còn tiếp tục phát triển tiếp.

CHƯƠNG 2. NGHIÊN CỨU CẤU TRÚC ẢNH BITMAP

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 Infomation	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: • 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:

- 1bit (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 $31 \times 4 bytes = 124 bytes$

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:
$2^{8\times3} = 2^{24}$ 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.

CHƯƠNG 3. NGHIÊN CỨU KỸ THUẬT GIẤU TIN THUẬN NGHỊCH

Giấu tin thuận nghịch là kỹ thuật giấu thông điệp sau khi tách lấy thông điệp ta có thể khôi phục lại xấp xỉ ảnh gốc ban đầu.

3.1 Kỹ thuật giấu tin dựa trên sự khác biệt

Kỹ thuật giấu tin thuận nghịch dựa trên Difference Expansion do Jun Tian đề xuất năm 2002 [5].

3.2 Ý tưởng và thuật toán

Kỹ thuật này nhúng thông điệp cần giấu vào sự khác biệt của cặp giá trị điểm ảnh.

3.2.1 Một số công thức và định nghĩa

a) Một số công thức:

Từ cặp giá trị điểm ảnh (x,y) của ảnh, ta tính giá trị trung bình *l* và sự khác biệt *h* theo công thức:

$$l = \left\lfloor \frac{x+y}{2} \right\rfloor,\ h = x - y \qquad (1)$$

trong đó $\lfloor \rfloor$ là phép toán lấy phần nguyên.

Từ (1) ta biến đổi:

$$x = l + \left\lfloor \frac{h+1}{2} \right\rfloor,\ y = l - \left\lfloor \frac{h}{2} \right\rfloor \qquad (2)$$

Mà giá trị của điểm ảnh trong ảnh xám nằm trong [0,255], ta có:

$$0 \le l + \left\lfloor \frac{h+1}{2} \right\rfloor \le 255 \text{ và } 0 \le l - \left\lfloor \frac{h}{2} \right\rfloor \le 255$$

⇨ Công thức

$$|h| \le \min(2(255-l), 2l+1) \qquad (3)$$

Công thức này được sử dụng để kiểm tra *h* sau khi thay đổi để tránh vấn đề tràn sau khi nhúng vào ảnh.

LSB của *h* sẽ được chọn làm vùng để nhúng bit dữ liệu:

$$h = \left\lfloor \frac{h}{2} \right\rfloor .2 + LSB(h)$$

Với LSB (h) = 0 hoặc 1, để tránh vấn đề tràn, chúng ta chỉ nhúng vào những giá trị *h* có thể thay đổi (changeable).

b) Một số định nghĩa:

Định nghĩa: Cho một cặp giá trị điểm ảnh xám (x,y) chúng ta nói *h* là có thể thay đổi nếu *h* thỏa mãn điều kiện sau:

$$\left| \left\lfloor \frac{h}{2} \right\rfloor .2 + b \right| \le \min(2(255 - l), 2l + 1)$$

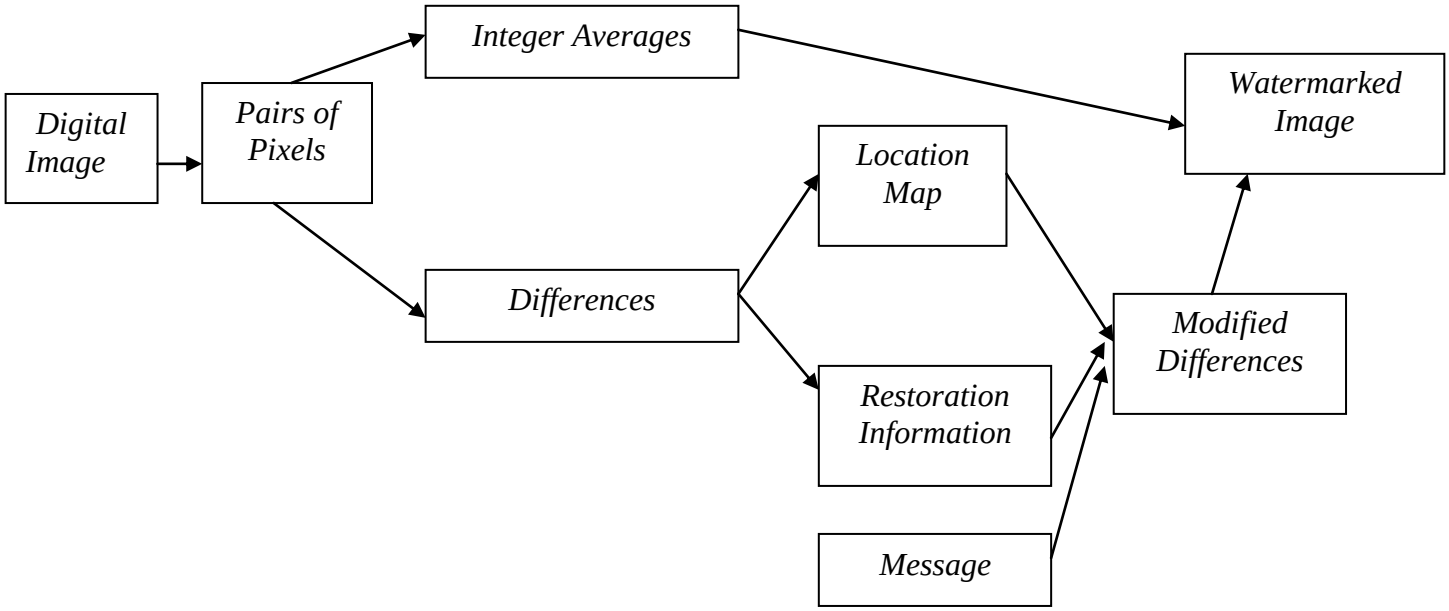
cho cả b=0 và 1.

Định nghĩa: Cho một cặp giá trị điểm ảnh xám (x,y) chúng ta nói *h* là có thể mở rộng nếu *h* thỏa mãn điều kiện sau:

$$|2.h + b| \le \min(2(255 - l), 2l + 1)$$

cho cả b=0 và 1.

3.2.2 Quá trình giấu thông tin



Hình 3.1 - Lược đồ quá trình giấu tin [5].

Các bước thực hiện:

Bước 1:

Chúng ta áp dụng công thức (1) cho mỗi cặp điểm ảnh.

Tiếp theo chúng ta phân loại h thành 5 loại EZ, NZ, EN, CNE và NC:

1. EZ: expandable zeros. Cho tất cả $h \in 0,-1$ mà có thể mở rộng.
2. NZ: not expandable zeros. Cho tất cả $h \in 0,-1$ mà không thể mở rộng.
3. EN: expandable nonzeros. Cho tất cả $h \notin 0,-1$ mà có thể mở rộng.
4. CNE: changeablable, but not expandable. Cho tất cả h $h \notin 0,-1$ mà có thể thay đổi nhưng không thể mở rộng.
5. NC: not changeable. Cho tất cả h $h \notin 0,-1$ mà không thể thay đổi.

Bước 2:

Dựa vào 5 thành phần có được ở bước 1 chúng ta sẽ tạo bản đồ định vị (location map) những cặp mà có *h* thuộc EZ hoặc EN chúng ta sẽ thiết lập giá trị là 1, còn những cặp mà có *h* thuộc NZ, CNE hoặc NC chúng ta sẽ thiết lập giá trị là 0. Bản đồ định vị (bitmap 1bit) sẽ được nén xuống bởi thuật toán nén JBIG2 [8] hoặc mã đoạn dài. Chuỗi bit sau khi nén sẽ được kí hiệu là L.

Chúng ta sẽ thu thập giá trị LSB gốc của *h* trong CNE tạo thành chuỗi bit C, với h=1 hoặc -2 thì LSB sẽ không được thêm vào C.

Với bản đồ định vị L, giá trị LSB gốc C, và chuỗi bit thông điệp P chúng ta kết hợp chúng thành chuỗi bit B:

$$B = L \cup C \cup P$$

Tùy vào *h* mà chúng ta sẽ có những công thức nhúng bit *b* riêng:

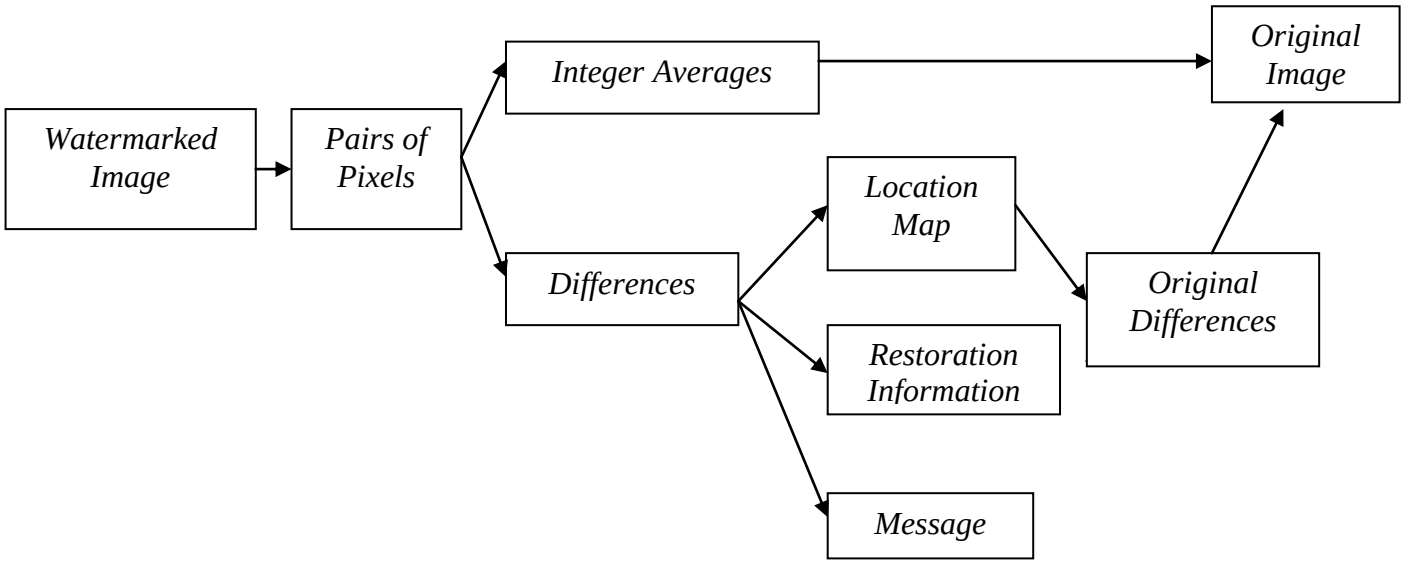
- EZ hoặc EN: $h = 2.h + b$
- CNE: $h = \left\lfloor \frac{h}{2} \right\rfloor .2 + b$
- NZ hoặc NC: không thay đổi giá trị của *h*.

Với *b* lần lượt là những bit thuộc chuỗi bit B.

Bước 3:

Sau khi các bit *b* đã được nhúng hết vào trong *h*. Chúng ta áp dụng công thức (2) để nhúng vào trong ảnh gốc.

3.2.2 Quá trình tách thông tin và khôi phục ảnh gốc



Hình 3.2 - Lược đồ quá trình tách tin [5].

Các bước thực hiện:

Quá trình giải mã ngược lại với quá trình giấu tin.

Bước 1:

Áp dụng công thức (1) cho mỗi cặp giá trị điểm ảnh

Tiếp theo chúng ta chia *h* làm 2 phần C và NC:

1. C: changeable. Cho tất cả *h* có thể thay đổi
2. NC: not changeable. Cho tất cả *h* không thể thay đổi.

Tiếp theo chúng ta thu thập tất cả LSB của *h* trong C sẽ được chuỗi bit B. Từ B chúng ta sẽ có L, C và P. Từ L chúng ta giải mã được bản đồ định vị (location map). Với bản đồ định vị chúng ta sẽ khôi phục lại giá trị *h* ban đầu như sau:

- Nếu $h \in C$ và bản đồ định vị có giá trị là 1, khi đó $h = \left\lfloor \frac{h}{2} \right\rfloor$
- Nếu $h \in C$, bản đồ định vị có giá trị là 0 và $0 \le h \le 1$ khi đó $h = 1$
- Nếu $h \in C$, bản đồ định vị có giá trị là 0 và $-2 \le h \le -1$khi đó$h = -2$
- Nếu $h \in C$, bản đồ định vị có giá trị là 0 và $h \ge 2$ hoặc $h \le -3$khi đó

$$h = \left\lfloor \frac{h}{2} \right\rfloor .2 + b$$

- Nếu $h \in NC$, *h* không thay đổi.

Bước 2:

Sau khi tất cả *h* đã được khôi phục, chúng ta áp dụng công thức (2) để khôi phục lại ảnh gốc.

Kết quả: ảnh gốc được khôi phục, và thu được chuỗi bít thông điệp giấu P.

Nhận xét kỹ thuật:

Kỹ thuật có khả năng giấu cao, trên 95% ảnh. Chất lượng ảnh sau khi giấu tin tốt. Có khả năng khôi phục lại ảnh gốc có độ chính xác cao.

CHƯƠNG 4. CÀI ĐẶT THỬ NGHIỆM

4.1 Môi trường thử nghiệm

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 và 20 ảnh được chụp từ máy ảnh kỹ thuật số và được convert thành ảnh xám 8 bit bởi phần mềm Adobe Photoshop CS2 với nhiều kích cỡ khác nhau Hình 4.2.

Hình 4.1 10 ảnh chuẩn

Hình 4.2 20 chụp bằng máy ảnh kỹ thuật số với nhiều kích cỡ

4.1.2 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 va K có kích thước $m \times n$:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} I[i,j] - K[i,j]^2$$

PSNR được định nghĩa:

$$PSNR = 10.\log_{10}\left(\frac{MAX_I^2}{MSE}\right) = 20.\log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right)$$

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

4.1.3 Một số giao diện chương trình demo

a) Giao diện chính của chương trình (hình 4.3)

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.
- Giấu theo tỷ lệ ảnh: thực hiện giấu với chuỗi bit ngẫu nhiên với kích thước được tính toán theo tỷ lệ % của ảnh (tỷ lệ do người dùng nhập).

Tách tin:

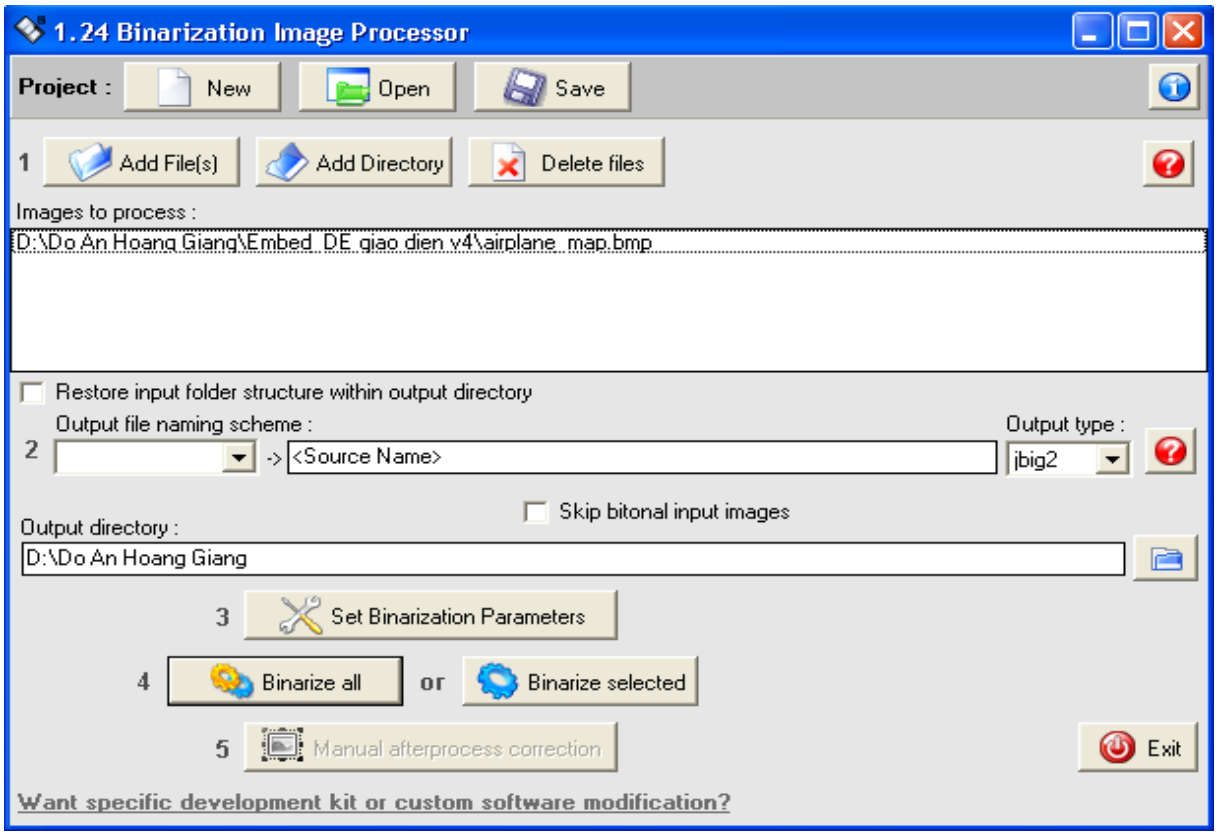
- Tách chuỗi kí tự: thực hiện tách một chuỗi thông điệp từ ảnh đã được giấu bởi chức năng giấu chuỗi kí tự.
- Tách tệp văn bản: thực hiện tách một tệp văn bản từ ảnh đã được giấu bởi chức năng giấu tệp văn bản.

PSNR:

- Tính toán PSNR.

Hình 4.3 Giao diện chính chương trình

b) Giao diện chương trình BIP (Binarization Image Processor) hình 4.4



Hình 4.4 Giao diện chương trình nén ảnh BIP (Binarization Image Processor) [7]

Các chức năng chính:

- Add File (s): tệp ảnh đầu vào quá trình nén
- Output type: lựa chọn định dạng ảnh kết quả
- Output directory: Chọn thư mục lưu ảnh kết quả
- Binarize all: thực hiện quá trình xử lý ảnh

Cách chuyển đổi một định dạng ảnh bằng phần mềm BIP:

Bước 1: Click vào Add File (s) hoặc Add Directory để chọn tệp ảnh cần xử lý. Phần mềm sẽ xử lý tất cả những tệp có trong danh sách Images to process.

Bước 2: Click vào Output type để chọn kiểu (TIFF/PDF/JBIG2/BMP) cho ảnh kết quả.

Bước 3: Click vào Output directory để chọn đường dẫn tới thư mục lưu ảnh kết quả.

Bước 4: Click vào nút Binarize all để xử lý tất cả ảnh có trong danh sách đầu vào hoặc nút Binarize selected để xử lý những ảnh được chọn trong danh sách.

Các định dạng đầu ra của phần mềm BIP:

- TIFF
- PDF
- JBIG2
- BMP (Windows bitmap) 1-bpp
- GIF
- PNG

Các yêu cầu tối thiểu về hệ thống khi cài đặt:

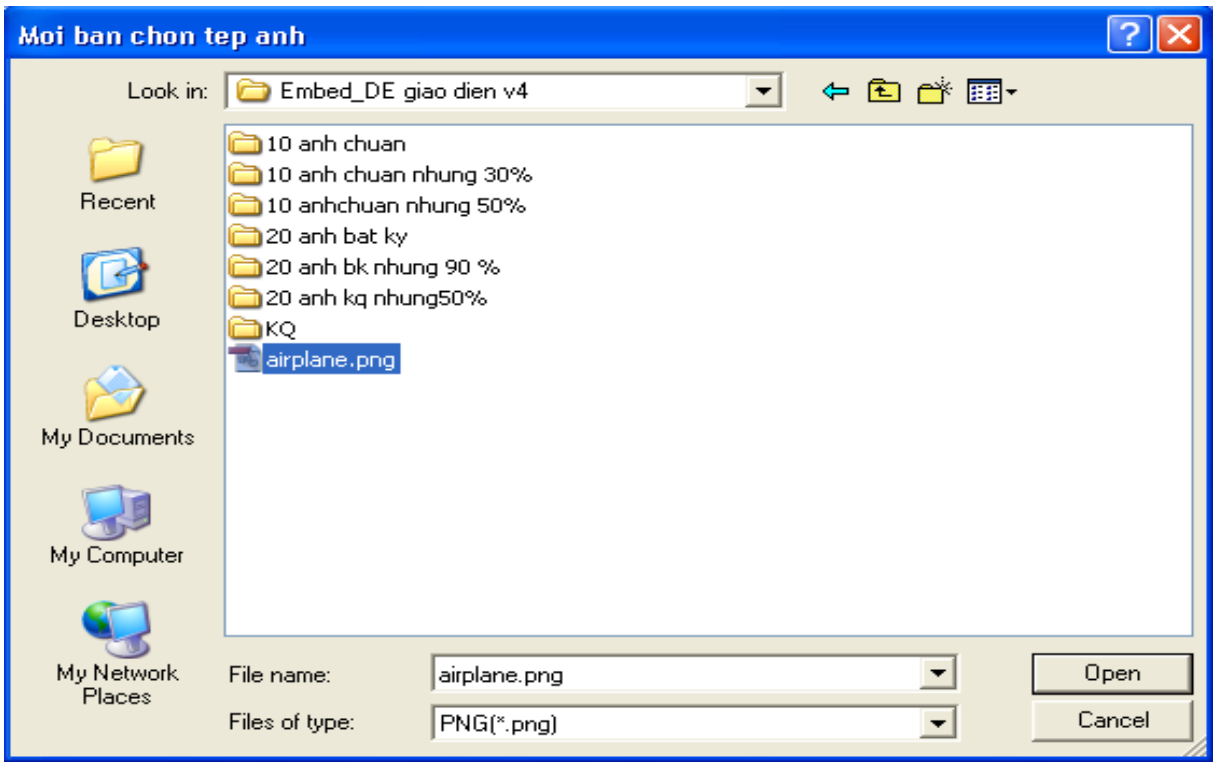
- 128 MB RAM
- ~ 10 MB ổ đĩa trống
- Mode hiển thị 1024x768
- Bộ xử lý Celeron 800MHz
- Windows 98 (Windows XP/Vista)

Thông tin liên hệ:

- Thông tin chung: info@xvel.com
- Đăng ký: orders@xvel.com
- Hỗ trợ khách hàng: support@xvel.com

c)Giao diện quá trình giấu tin bất kỳ

Hình 4.5 Giao diện nhúng thông điệp



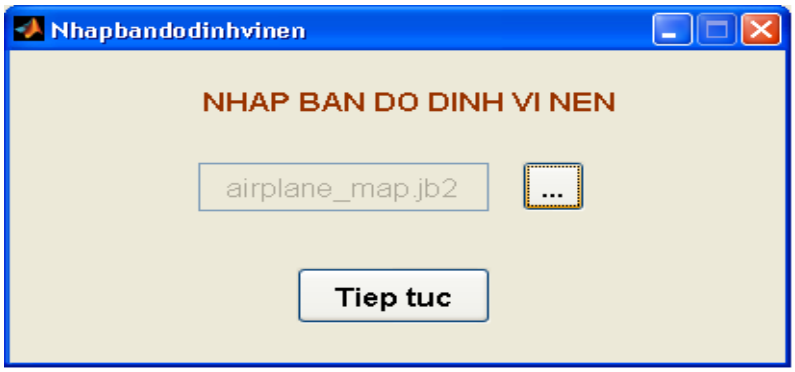
Hình 4.6 Chọn ảnh

Ô nhập dữ liệu:

- Ảnh nhúng: click vào nút bên cạnh chọn ảnh để giấu thông tin.
- Bản đồ: tên ảnh lưu trữ bản đồ định vị, sẽ được nén lại chương trình nén JBIG2 rồi nhúng vào ảnh.
- Thông điệp: nhập vào thông điệp cần giấu.
- Ảnh kết quả: lưu trữ ảnh kết quả sau khi giấu tin.

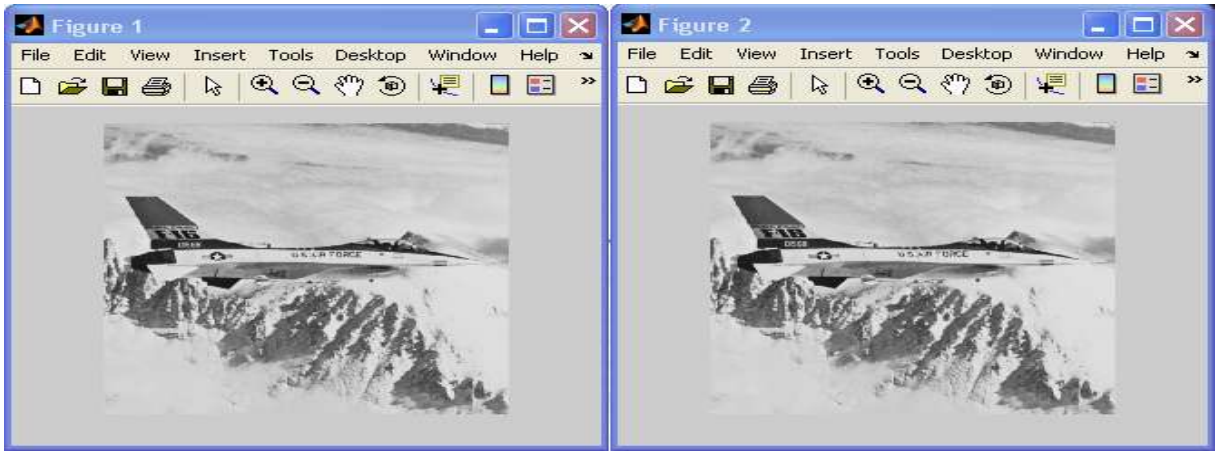
Nút bấm:

- Click vào nút  để chọn ảnh.
- Giấu tin: thực hiện quá trình giấu tin.
- Thoát: thoát khỏi giao diện giấu thông tin.



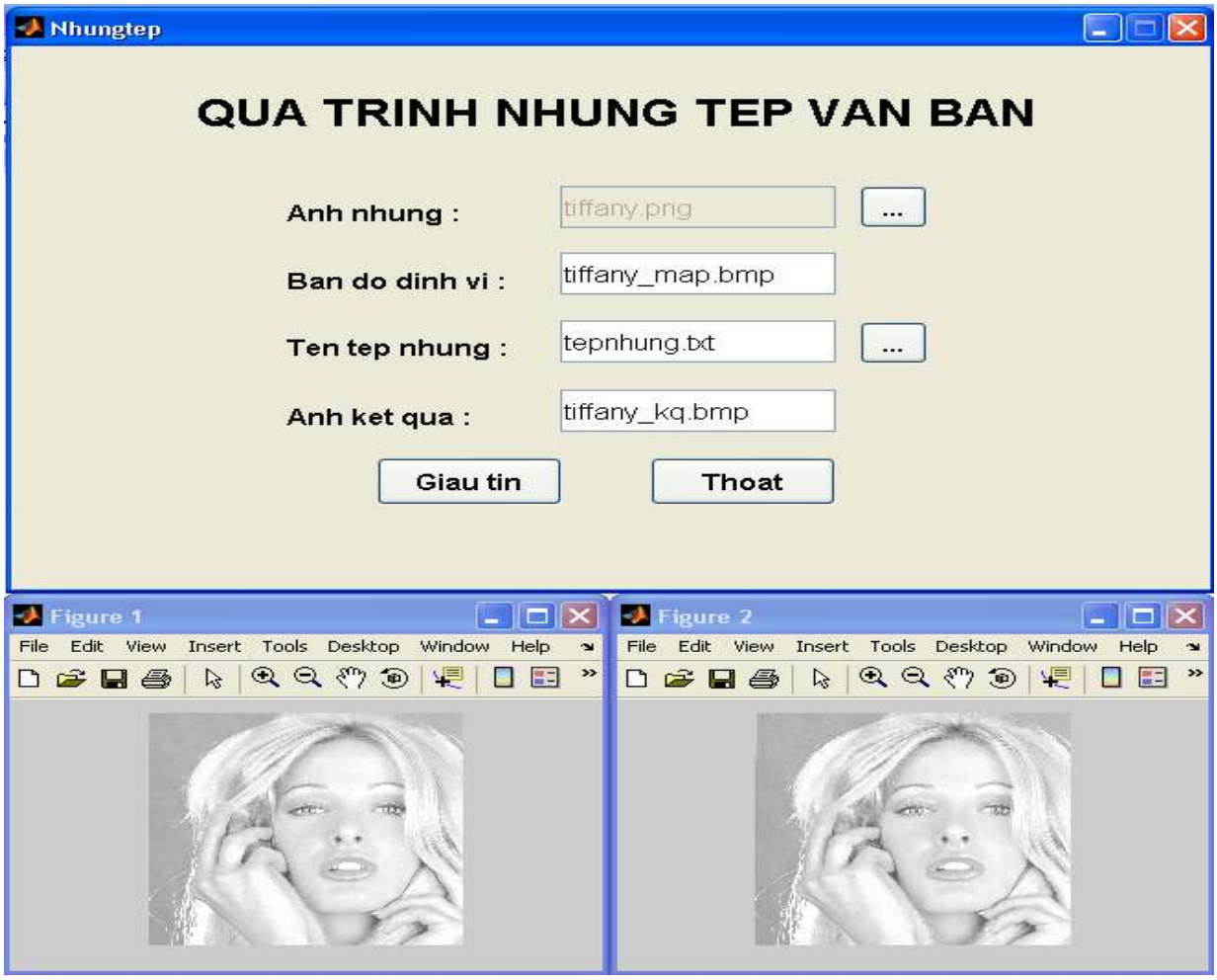
Hình 4.7 Nhập bản đồ định vị nén

Khi thực hiện giấu tin, bản đồ định vị sẽ được tạo với tên trong ô dữ liệu bản đồ. Nén bản đồ định vị trên bởi phần mềm nén BIP (Binarization Image Processor) với định dạng tệp kết quả là jb2 (hình 4.4). Sau đó click vào nút chọn ảnh (hình 4.7) để chọn bản đồ định vị đã nén. Sau đó click vào nút tiếp tục thể tiếp tục thực hiện quá trình giấu tin. Kết quả đạt được thể hiện qua (hình 4.8)



Hình 4.8 Ảnh trước và sau khi giấu tin

d) Giao diện quá trình giấu tệp văn bản:



Hình 4.9 Giao diện quá trình giấu tệp văn bản

Ô nhập dữ liệu:

- Ảnh nhúng: click vào nút bên cạnh chọn ảnh để giấu thông tin.
- Bản đồ định vị: tên ảnh lưu trữ bản đồ định vị, sẽ được nén lại chương trình nén JBIG2 rồi nhúng vào ảnh.
- Tên tệp nhúng: click vào nút bên cạnh để chọn tệp văn bản cần giấu.
- Ảnh kết quả: lưu trữ ảnh kết quả sau khi giấu.

Nút bấm:

- Giấu tin: thực hiện quá trình giấu tin.
- Thoát: thoát khỏi giao diện giấu thông tin.

Khi thực hiện giấu tin, bản đồ định vị sẽ được tạo với tên được nhập trong ô dữ liệu bản đồ. Nén bản đồ định vị trên bởi phần mềm BIP (hình 4.4). Click vào nút chọn ảnh để chọn bản đồ định vị đã nén. Sau đó click vào nút tiếp tục thể tiếp tục thực hiện quá trình giấu tin.

e) Giao diện giấu tin theo tỷ lệ kích thước ảnh

Hình 4.10 Giao diện quá trình giấu theo tỷ lệ ảnh

Giải thích tham số đầu vào:

Ô nhập dữ liệu:

- Ảnh nhúng: click vào nút bên cạnh chọn ảnh để giấu thông tin.
- Bản đồ định vị: tên ảnh lưu trữ bản đồ định vị, sẽ được nén lại chương trình nén JBIG2 rồi nhúng vào ảnh.
- Tỷ lệ nhúng: nhập vào tỷ lệ (0-90 %).
- Ảnh kết quả: lưu trữ ảnh kết quả sau khi giấu tin.

Nút bấm:

- Giấu tin: thực hiện quá trình giấu tin.
- Thoát: thoát khỏi giao diện giấu thông tin.

Khi thực hiện giấu tin, bản đồ định vị sẽ được tạo với tên được nhập trong ô dữ liệu bản đồ. Nén bản đồ định vị trên bởi phần mềm BIP (hình 4.4). Click vào nút chọn ảnh để chọn bản đồ định vị đã nén. Sau đó click vào nút tiếp tục thể tiếp tục thực hiện quá trình giấu tin.

4.2 Các module cài đặt

4.2.1 Chức năng: Giấu thông tin vào trong ảnh.

Các tham số đầu vào:

- digital_image: tên của ảnh sẽ giấu tin lên.
- location_map: tên của ảnh lưu trữ bản đồ định vị.
- watermark_iamge: tên của ảnh kết quả sau khi giấu tin.
- message: nội dung thông điệp giấu vào ảnh.
- dk: thiết lập bằng 1 khi giấu thông tin, giấu tệp. Khác 1 khi giấu theo tỷ lệ ảnh.

4.2.2 Chức năng: Tách thông tin từ trong ảnh và khôi phục lại ảnh ban đầu.

Các tham số đầu vào:

- watermark_image: tên của ảnh sẽ tách tin.
- map_compress: tên của ảnh lưu trữ bản đồ định vị nén (*.jb2).
- result_image: tên ảnh kết quả sau khi khôi phục.

Tham số đầu ra:

- message: lưu thông điệp tách được từ ảnh đầu vào.

4.2.3 Chức năng:Trích xuất phần dữ liệu của ảnh JBIG2.

Tham số đầu vào:

- image_jb2: tên của ảnh jbig2 sẽ được trích phần dữ liệu.

Tham số đầu ra:

- data: lưu trữ phần dữ liệu của ảnh jbig2 đầu vào.

4.2.4 Chức năng: Khôi phục ảnh JBIG2

Các tham số đầu vào:

- data: chứa phần dữ liệu của ảnh JBIG2 cần khôi phục.
- file_name: tên của ảnh JBIG2 sau khi khôi phục.

4.2.5 Chức năng: Đọc một tệp văn bản

Tham số đầu vào:

- filename: tên tệp văn bản cần đọc.

Tham số đầu ra:

- text: nội dung của tệp văn bản đầu vào.

4.2.6 Chức năng: Ghi một tệp văn bản

Tham số đầu vào:

- text: nội dung cần ghi vào tệp văn bản.
- filename: tên tệp văn bản cần ghi vào.

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

Tham số đầu vào:

- String: chuỗi kí tự cần đổi.

Tham số đầu ra:

-Bit: chuỗi bít kết quả.

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

Tham số đầu vào:

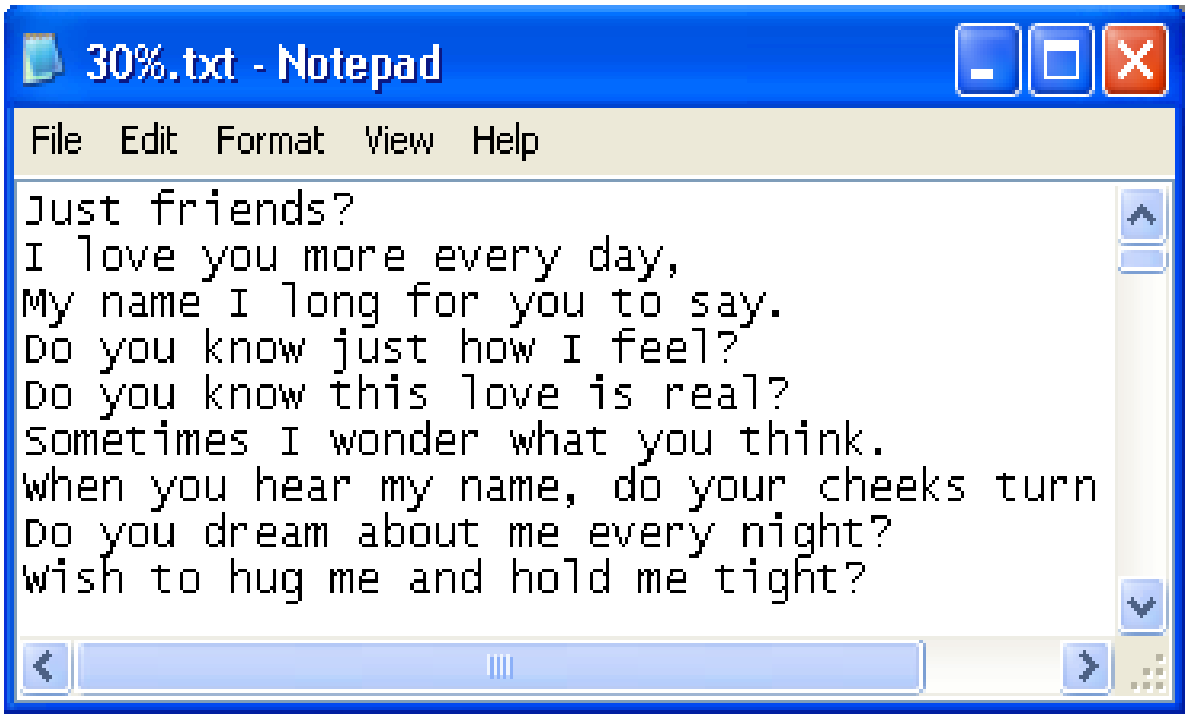
-Bit: chuỗi bít cần đổi.

Tham số đầu ra:

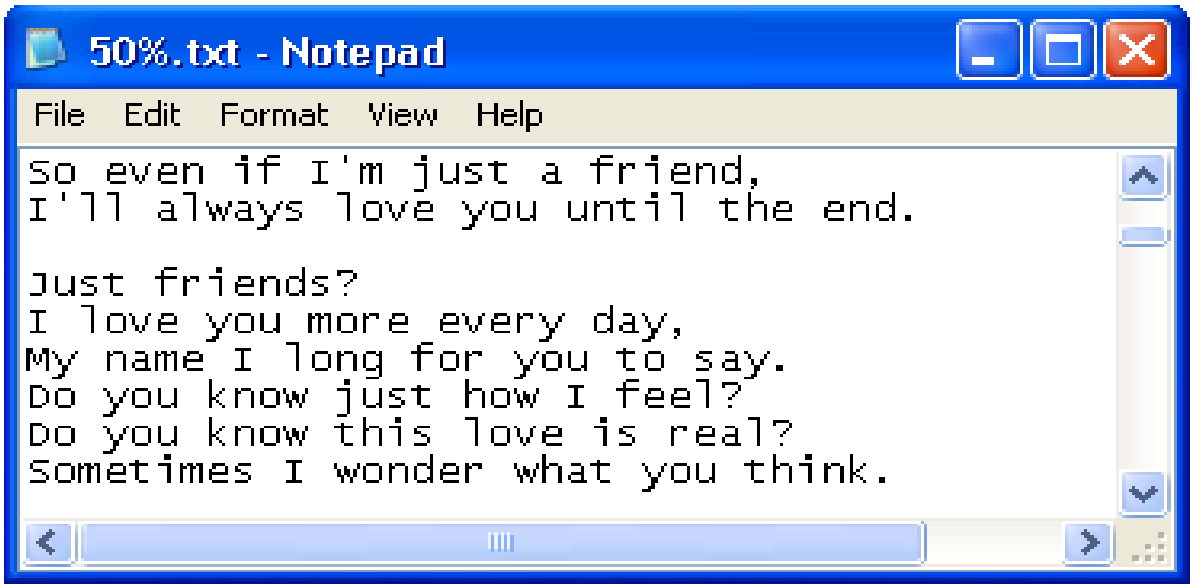
- String: chuỗi kí tự kết quả.

4.3 Thực nghiệm và đánh giá

4.3.1 Giấu trên 10 ảnh chuẩn (hình 4.1)



Hình 4.11 Tệp thông điệp(nội dung 40.320 bit)



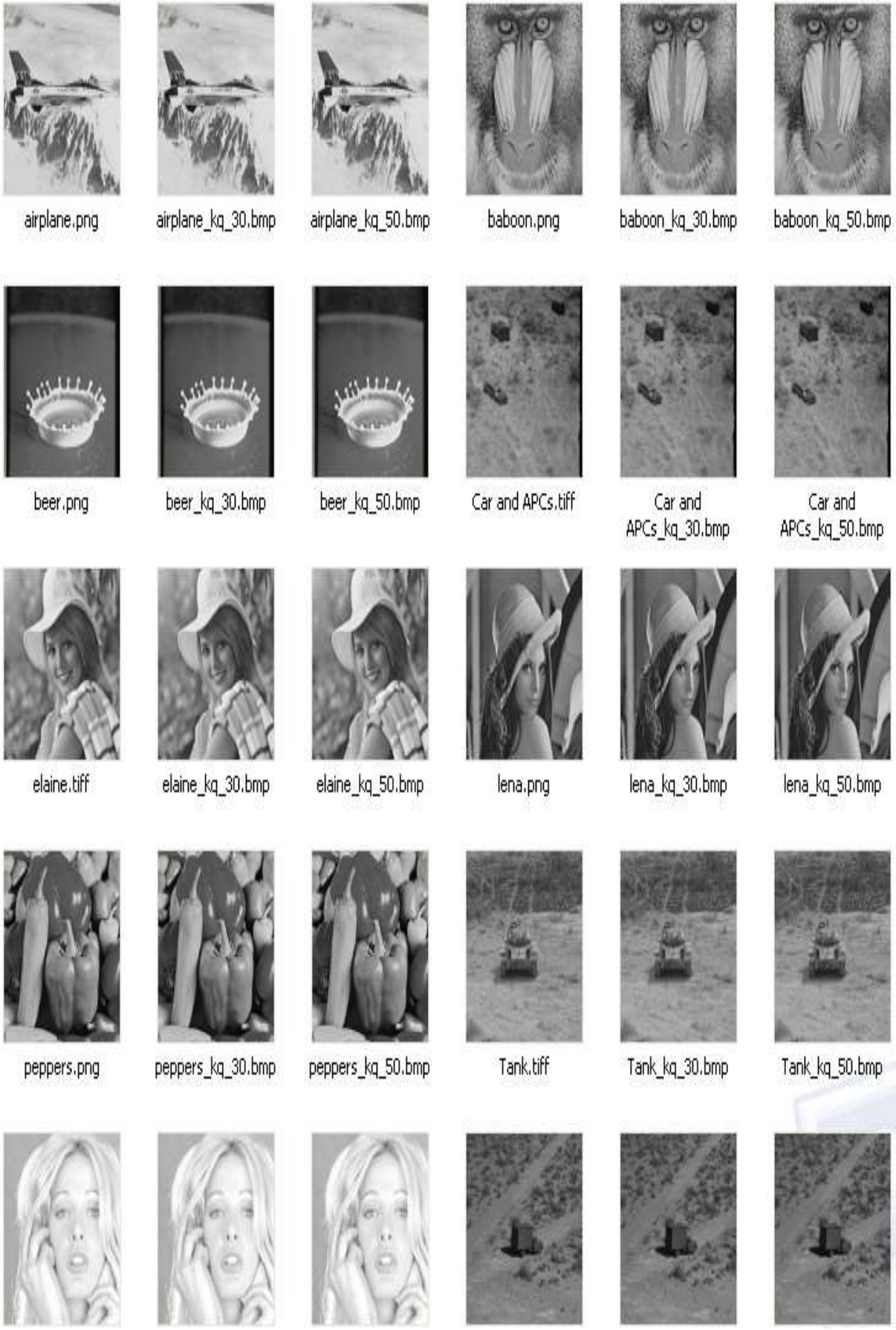
Hình 4.12 Tệp thông điệp (nội dung 67.264 bit)

Sử dụng kỹ thuật Difference Expandable nhúng 2 file text trên vào tập cơ sở dữ liệu ảnh hình 4.1 ta được bảng PSNR 4.1:

Bảng 4.1 Kết quả thực nghiệm trên 10 ảnh chuẩn

Lượng giấu **Ảnh giấu**	**PSNR**	
	40.320 bit(30%)	67.264 bit(50%)
Lena	41.6224	37.5255
Baboon	29.1354	28.3151
Tifany	37.8825	36.3264
Beer	45.7153	41.5941
Airplan	41.9967	36.4263
Peppers	37.8034	35.0179
Car and APCs	40.6679	38.822
Elaine	39.1969	36.3985
Tank	38.5957	36.5056
Truck	41.1357	39.2808
Trung bình	**39.3751**	**36.6212**

Tập ảnh kết quả:



Hình 4.13 Tập ảnh trước và sau khi giấu 30% và 50% kích thước ảnh

4.3.2 Giấu trên 20 ảnh bất kỳ

Hình 4.14 Tập ảnh thử nghiệm gồm 20 ảnh bất kỳ với kích cỡ khác nhau

Bảng 4.2 Bảng kết quả thực nghiệm trên 20 ảnh

Lượng giấu **Ảnh giấu**	**PSNR**	
	50%	90%
01.jpg	49.0099	41.4351
02.jpg	37.4153	35.6182
03.jpg	41.384	39.8677
04.jpg	50.4966	48.6418
05.jpg	42.7052	39.9617
06.jpg	29.0004	28.0265
07.jpg	35.1926	34.5406
08.jpg	40.2189	38.5396
09.jpg	34.001	33.1274
10.jpg	27.505	27.4891
11.jpg	34.5084	30.8999
12.png	31.9743	30.8325
13.jpg	40.2785	37.7396
14.jpg	36.5757	31.6814
15.jpg	34.1363	32.7741
16.jpg	32.1671	31.3052
17.jpg	32.794	29.349
18.jpg	33.3168	31.5951
19.jpg	44.7953	42.1914
20.jpg	40.0799	38.5042
Trung bình	**37.37776**	**35.20601**

Nhận xét:

Kỹ thuật giấu được lượng thông tin lớn, quá trình xử lý nhanh, chất lượng hình ảnh sau khi giấu tin là tốt (PSNR>35).

KẾT LUẬN

Khóa luận đã thực hiện nhiệm vụ:

1. Trình bày tổng quan kỹ thuật giấu tin
2. Nghiên cứu cấu trúc ảnh bitmap
3. Nghiên cứu một số kỹ thuật giấu tin thuận nghịch

Đây là một kiến thức rất hữu ích và cần thiết để có thể khai thác ngày một hiệu quả các thành tựu của tin học. Đó cũng là một lý do để em chọn đề tài này làm đồ án tốt nghiệp, mong muốn giới thiệu và phổ biến những kiến thức rất cơ bản đến người đọc.

Việc kết hợp giấu thông tin và công nghệ thông tin là một vấn đề mới đang được nghiên cứu và phát triển để phục vụ nhiều lĩnh vực khác nhau. Trên thế giới người ta đã nghiên cứu nhiều về vấn đề này.

Kỹ thuật giấu thông tin trong ảnh nói chung và giấu thông tin trong ảnh xám nói riêng là một hướng nghiên cứu chính của kỹ thuật giấu thông tin hiện nay và đã đạt nhiều kết quả khả quan.

Trong đề tài này em đã trình bày một số khái niệm liên quan đến việc che giấu thông tin nói chung và cụ thể là thuật toán giấu thông tin trong ảnh xám nói riêng.

Do còn nhiều hạn chế về thời gian nghiên cứu 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.

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] Trần Thị Thu Hà, Luận văn tốt nghiệp, ngành Công nghệ thông tin, năm 2009

[3] Mặc Như Hiển, Luận văn tốt nghiệp ngành CNTT, năm 2009

[4] Ingemar Cox, Jeffrey Bloom, Matthew Miller, Ton Kalker, Jessica Fridrich, Digital Watermarking and Steganography, Morgan Kaufmann, 2008

[5] Jun Tian, Reversible Watermarking by Difference Expansion, Multimedia and Security Workshop at ACM Multimedia 2002, Dec-02

[6] http://sipi.usc.edu/database/database.cgi?volume=misc

[7] Download phần mềm BIP (Binarization Image Processor) tại địa chỉ http://www.xvel.com/

[8] Hewlett-Packard Company, 11000 Wolfe Road, MS42U0, Cupertino CA 95014, USA. Information technology - coded representation of picture and audio information lossy/lossless coding of bi-level images