

## GIẤU TIN THUẬN NGHỊCH BẰNG DỊCH CHUYỂN LƯU ĐỒ VÀ CÂY TỬ PHÂN

Trần Hoàng Nam, Võ Phước Hưng, Nguyễn Thái Sơn, Huỳnh Văn Thanh

Khoa Kỹ thuật và Công nghệ, Trường Đại học Trà Vinh

tramhoangnam@tvu.edu.vn, hungvo@tvu.edu.vn, thaison@tvu.edu.vn, hvthanh@tvu.edu.vn

**TÓM TẮT:** Bài báo này đề xuất phương pháp giấu tin thuận nghịch dựa trên cây tử phân và dịch chuyển lưu đồ. Trước hết, giải thuật phân đoạn ảnh được thực hiện để phân chia các vùng của ảnh theo mô hình cây tử phân. Sau đó, từ mô hình cây tử phân tìm được, chúng tôi sẽ tiến hành nhúng dữ liệu bí mật vào các nút thích hợp nhất trên cây sao cho khả năng nhúng cao nhất và chất lượng ảnh tốt nhất. Giải pháp này được kết hợp với phương pháp xây dựng và dịch chuyển lưu đồ để giấu tin nhằm đạt được khả năng giấu tin cao và chất lượng ảnh sau mang tin tốt hơn. Kết quả thực nghiệm cho thấy, giải pháp đề xuất làm tăng khả năng nhúng lên khoảng 1,6 lần so với giải pháp trước trong khi chất lượng ảnh vẫn không giảm.

**Từ khóa:** Giấu tin, thuận nghịch, lưu đồ, cây tử phân.

### I. GIỚI THIỆU

Ngày nay, giấu tin trong ảnh số là một trong các lĩnh vực được các nhà khoa học quan tâm rất lớn. Nhiều dữ liệu quan trọng và bí mật được nhúng vào trong ảnh gọi là ảnh mang tin. Tuy nhiên, trong một số lĩnh vực như ảnh quân sự, y khoa, viễn thám,... chỉ cần một sai lệch nhỏ khi trích tin sẽ làm thay đổi ảnh mang tin và dẫn đến những hậu quả vô cùng nghiêm trọng. Chính vì vậy, giấu tin thuận nghịch - phương pháp giấu tin mà ảnh mang tin sẽ được khôi phục hoàn toàn sau khi đã lấy được thông tin mật được phát triển nhanh chóng. Trong đó J. Tian [1] là một trong những người đầu tiên đặt nền móng xây dựng nên phương pháp giấu tin này.

Giấu tin thuận nghịch có thể chia thành 3 phương pháp lớn như sau: mở rộng sự khác biệt (DE - difference expansion) [1]-[4], dịch chuyển lưu đồ (HS - histogram shifting) [5]-[11] và biến đổi số nguyên (IT - integer transformation) [12]-[14]. Phương pháp DE của J. Tian [1] tính toán sự khác biệt giữa từng cặp điểm ảnh và mở rộng nó để nhúng dữ liệu. Phương pháp này có khả năng nhúng đáng kể và ít làm biến dạng ảnh. Tuy nhiên, một bản đồ định vị cần được nhúng thêm vào ảnh gốc để xác định các cặp điểm ảnh đã mở rộng để đảm bảo cho việc phục hồi ảnh gốc. Sau đó, phương pháp DE được Alattar [2] tổng quát hóa với  $n - 1$  bit dữ liệu có thể được nhúng vào một véc tơ có  $n$  điểm ảnh. Năm 2008, Kim và cộng sự [4] đề xuất phương pháp mở rộng sự khác biệt mới với bản đồ định vị có kích thước nhỏ hơn các phương pháp DE trước đó nhờ sử dụng phân phối Laplace. Phương pháp giấu tin bằng dịch chuyển lưu đồ (HS) được Ni và cộng sự đề xuất lần đầu tiên [5]. Trong giấu tin thuận nghịch bằng kỹ thuật này, một lưu đồ (histogram) được xây dựng bằng cách thống kê tần suất xuất hiện của các giá trị điểm ảnh. Sau đó, cột cao nhất của nó (gọi là peak point) sẽ được chọn để giấu tin. Vì đây là giải pháp giấu tin thuận nghịch nên các cột nằm giữa peak point và zero point (cột thấp nhất trong lưu đồ) cần được dịch chuyển theo hướng về bên phải một đơn vị nhằm đảm bảo việc khôi phục ảnh gốc sau khi trích tin. Vì vậy, chiều cao của peak point trong lưu đồ sẽ quyết định khả năng nhúng của phương pháp HS. Năm 2013, Li [9] đề xuất chia ảnh mang tin thành các khối không trùng lặp. Các khối này được sắp xếp theo chiều tăng của giá trị điểm ảnh. Sau đó, giá trị sai khác của điểm ảnh lớn nhất và điểm ảnh lớn thứ hai ở mỗi khối được tính. Nếu sự sai khác này bằng 1 thì khối đó được chọn để nhúng thông tin. Ngược lại, khi sự sai khác lớn hơn 1 thì điểm ảnh lớn nhất trong khối này được mở rộng một đơn vị để đảm bảo tính thuận nghịch. Do đó, khả năng nhúng của phương pháp này còn hạn chế khi giá trị điểm ảnh lớn nhất và lớn thứ hai trong khối có sự khác biệt nhiều hơn 1. Đồng thời cũng kéo theo sự sụt giảm của chất lượng ảnh. Đến năm 2015, Qu và cộng sự [10] tiếp tục cải tiến phương pháp này bằng cách dự báo cho điểm ảnh gốc dựa vào các điểm ảnh lân cận trong mỗi khối. Sau đó, sự khác biệt giữa giá trị điểm ảnh dự báo và điểm ảnh gốc được tính và tiến hành giấu tin thông qua sự khác biệt này cũng bằng dịch chuyển lưu đồ. Phương pháp này làm tăng khả năng nhúng đáng kể đối với các ảnh mà ít có sự khác biệt quá lớn ở các khối. Tuy nhiên, phương pháp này cũng chỉ nhúng được tối đa một bit dữ liệu trên mỗi khối nên khả năng nhúng vẫn còn hạn chế. Năm 2018, nhóm tác giả Nguyễn Kim Sao [15] đề xuất phương pháp giấu tin thuận nghịch bằng cách sử dụng các thuộc tính của ngữ cảnh dự báo để loại bỏ bản đồ định vị. Bằng phương pháp này, nhóm tác giả đã sử dụng phương pháp dự báo hình thoi và nhúng tin theo phương pháp mở rộng sai số dự báo. Cách làm này giúp nhóm tác giả loại bỏ được bản đồ định vị nhưng ngược lại làm tăng thêm các hệ số hỗ trợ nên cũng ảnh hưởng đến khả năng nhúng cũng như chất lượng ảnh. Năm 2019, Fuqiang Di và cộng sự [11] đưa ra phương pháp giấu tin dựa trên phương pháp chia ảnh thành các khối và tìm ra những vùng mịn của ảnh để nhúng tin. Bằng phương pháp này, Fuqiang Di có thể nhúng tối đa hai bit dữ liệu trên mỗi khối bốn điểm ảnh nên khả năng nhúng vẫn chưa cao. Để nâng cao khả năng nhúng tin trong phương pháp của Fu, chúng tôi đề xuất phương pháp giấu tin thuận nghịch dựa trên cây tử phân và dịch chuyển lưu đồ. Ở phương pháp này, một giải thuật được áp dụng để tìm ra các vùng thích hợp nhúng tin và phân cấp các vùng này theo mô hình cây tử phân. Sau đó, từ mô hình cây tử phân tìm được, chúng tôi sẽ tiến hành nhúng dữ liệu vào các nút thích hợp nhất trên cây sao cho đạt khả năng nhúng cao nhất cũng như chất lượng ảnh tốt nhất.

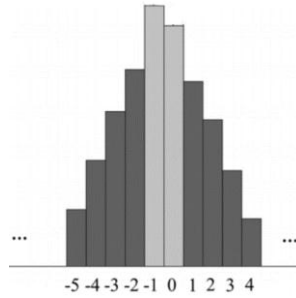
## II. NGHIÊN CỨU LIÊN QUAN

### A. High fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion (PVO)

Ở giải pháp của Li và cộng sự [9], trước hết ảnh gốc  $I$  có kích thước  $H \times W$  được chia thành các khối  $X$  không trùng lặp có cùng kích thước  $h \times w$ . Mỗi khối  $X$  gồm  $h \times w$  điểm ảnh này sẽ được duyệt từ trái sang phải, từ trên xuống dưới tạo thành mảng  $X_i = (x_1, x_2, \dots, x_{h \times w})$ . Tiếp theo  $X_i$  được sắp xếp tạo thành mảng  $X_\partial = (x_{\partial(1)}, x_{\partial(2)}, \dots, x_{\partial(h \times w)})$  với  $\partial$  là song ánh  $\partial: \{1, 2, \dots, h \times w\} \rightarrow \{1, 2, \dots, h \times w\}$  sao cho  $x_{\partial(1)} \leq x_{\partial(2)} \leq \dots \leq x_{\partial(h \times w)}$ ,  $\partial(i) < \partial(j)$  nếu  $x_{\partial(i)} = x_{\partial(j)}$  và  $i < j$ . Sau đó, điểm ảnh (sau khi sắp xếp) có giá trị lớn nhất sẽ được dự báo bởi điểm ảnh có giá trị lớn thứ hai trong mảng, và điểm ảnh có giá trị nhỏ nhất được dự báo bởi điểm ảnh có giá trị nhỏ thứ hai. Cụ thể, sai số dự báo lớn nhất và nhỏ nhất được tính theo công thức (1)

$$\begin{aligned} PE_{max} &= x_{\partial(h \times w)} - x_{\partial(h \times w - 1)} \\ PE_{min} &= x_{\partial(1)} - x_{\partial(2)} \end{aligned} \quad (1)$$

Một lưu đồ thể hiện tần suất xuất hiện của các sai số dự báo được hình thành gọi là lưu đồ sai số dự báo như Hình 1.



Hình 1. Lưu đồ sai số dự báo

Thông tin mật  $s = \{0, 1\}$  có thể được nhúng vào cột cao nhất của lưu đồ này bằng cách điều chỉnh sai số dự báo, thường là điều chỉnh cột có giá trị sai số dự báo bằng 1 hoặc -1. Nếu sai số dự báo bằng không (nằm giữa cột -1 và 1), thì giá trị sai số dự báo sau điều chỉnh không đổi, nếu sai số dự báo lớn hơn 1 hoặc nhỏ hơn -1 thì giá trị sai số dự báo được dịch chuyển một đơn vị sang phải (cộng thêm 1) hoặc sang trái (giảm đi 1) để đảm bảo tính thuận nghịch. Cách điều chỉnh sai số dự báo này được thể hiện ở công thức (2) và (3)

$$\begin{aligned} PE'_{max} &= PE_{max} \text{ nếu } PE_{max} = 0 \\ PE'_{max} &= PE_{max} + s \text{ nếu } PE_{max} = 1 \\ PE'_{max} &= PE_{max} + 1 \text{ nếu } PE_{max} > 1 \end{aligned} \quad (2)$$

$$\begin{aligned} PE'_{min} &= PE_{min} \text{ nếu } PE_{min} = 0 \\ PE'_{min} &= PE_{min} - s \text{ nếu } PE_{min} = -1 \\ PE'_{min} &= PE_{min} + 1 \text{ nếu } PE_{min} < -1 \end{aligned} \quad (3)$$

Như vậy, chỉ có điểm ảnh có giá trị lớn nhất và nhỏ nhất ở mỗi khối  $X$  được nhúng tin hoặc dịch chuyển. Giá trị điểm ảnh của  $X$  sau khi được nhúng tin hoặc dịch chuyển thể hiện qua công thức (4)

$$\begin{aligned} x'_{\partial(h \times w)} &= x_{\partial(h \times w - 1)} + PE'_{max} \\ x'_{\partial(1)} &= x_{\partial(2)} + PE'_{min} \end{aligned} \quad (4)$$

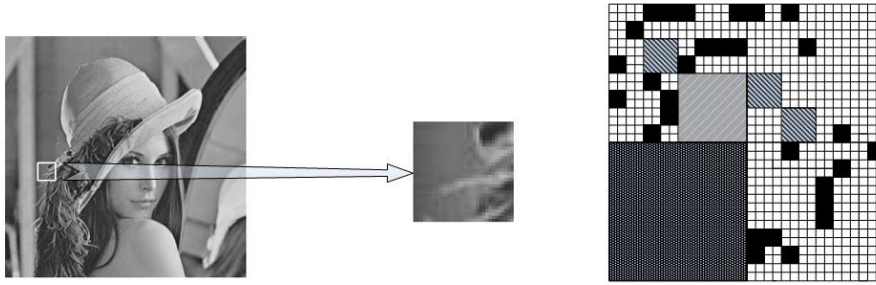
### B. High-fidelity reversible data hiding by Quadtree-based pixel value ordering (QPVO)

Năm 2019, Fuqiang Di và cộng sự đề xuất giải pháp giấu tin dựa trên PVO và cây tứ phân [11]. Trước hết, nhóm tác giả tính độ phức tạp của ảnh gốc để tìm giá trị ngưỡng tối ưu  $T$ . Sau đó, tiến hành phân đoạn ảnh gốc theo ngưỡng  $T$  đó trước khi nhúng dữ liệu. Bằng cách chia ảnh gốc  $I$  thành  $N$  khối không trùng lặp  $\{B_1, B_2, \dots, B_N\}$  có kích thước  $2 \times 2$ , độ phức tạp  $R$  của ảnh gốc  $I$  có kích thước  $2^n \times 2^n$  được tính theo công thức (5) với  $S_{i1}$  và  $S_{i2}$  lần lượt là giá trị điểm ảnh lớn thứ hai và nhỏ thứ hai trong khối  $B_i$ .

$$R = \frac{1}{N} \sum_{i=1}^N (S_{i1} - S_{i2}) \quad (5)$$

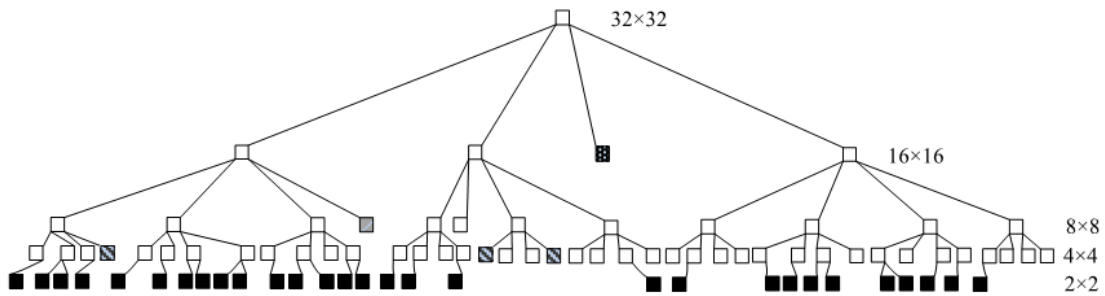
Ngưỡng tối ưu  $T$  trong [11] được xác định bằng  $T = R$ . Kế tiếp, ảnh gốc  $I$  được phân đoạn bằng cách chia  $I$  thành các khối không trùng lặp có kích thước  $2^m \times 2^m$  (trong bài báo này tác giả chọn  $m = 5$ ) và tính toán độ phức

tạp  $r$  của từng khối  $2^m \times 2^m$  theo công thức (5). Nếu  $r > T$ , khối hiện hành được chia thành 4 khối con không trùng lặp  $2^{m-1} \times 2^{m-1}$  và lặp lại công việc phân đoạn tương tự với 4 khối con đó. Ngược lại, nếu  $r \leq T$  thì khối hiện hành được ghi nhận và không phân đoạn nữa.



Hình 2. Một phân đoạn của ảnh Lena trong QPVO

Kế tiếp, các khối con sau khi ghi nhận được sắp xếp theo kích thước từ cao xuống thấp theo mô hình cây tứ phân và tiến hành nhúng dữ liệu mật vào các khối của cây này cho đến khi toàn bộ dữ liệu được nhúng hết.



Hình 3. Cây tứ phân của khối  $32 \times 32$

Với mỗi khối con của cây tứ phân, dữ liệu được nhúng như sau:

Chia khối con thành khối  $2 \times 2$  không trùng lặp và sắp xếp khối con  $(x_1, x_2, x_3, x_4)$  thành  $(x_{p(1)}, x_{p(2)}, x_{p(3)}, x_{p(4)})$  với  $p: \{1, 2, 3, 4\} \rightarrow \{1, 2, 3, 4\}$  là ánh xạ một - một thỏa mãn  $x_{p(1)} \leq x_{p(2)} \leq x_{p(3)} \leq x_{p(4)}$  và  $p(i) < p(j)$  nếu  $x_{p(i)} = x_{p(j)}, i < j$ .

Sai số dự báo được tính theo công thức (6) với  $u = \min(p(3), p(4)), v = \max(p(3), p(4))$  và  $s = \min(p(1), p(2)), t = \max(p(1), p(2))$

$$\begin{aligned} PE_{max} &= x_u - x_v \\ PE_{min} &= x_s - x_t \end{aligned} \tag{6}$$

Giá trị lớn nhất  $x_{p(4)}$  và nhỏ nhất  $x_{p(1)}$  của khối được mở rộng để nhúng dữ liệu theo công thức (7) và (8)

$$x'_{p(4)} = \begin{cases} x_{p(4)} + b & \text{nếu } PE_{max} = 1 \\ x_{p(4)} + 1 & \text{nếu } PE_{max} > 1 \\ x_{p(4)} + b & \text{nếu } PE_{max} = 0 \\ x_{p(4)} + 1 & \text{nếu } PE_{max} < 0 \end{cases} \tag{7}$$

$$x'_{p(1)} = \begin{cases} x_{p(1)} - b & \text{nếu } PE_{max} = 1 \\ x_{p(1)} - 1 & \text{nếu } PE_{max} > 1 \\ x_{p(1)} - b & \text{nếu } PE_{max} = 0 \\ x_{p(1)} - 1 & \text{nếu } PE_{max} < 0 \end{cases} \tag{8}$$

Rõ ràng, ở cả hai giải pháp [9] và [11], dù cách nhúng khác nhau nhưng đều có điểm chung là mỗi khối 4 điểm ảnh thì chỉ nhúng được tối đa 2 bit dữ liệu nên tỉ lệ nhúng chỉ đạt 0,5 bpp do đó khả năng nhúng vẫn còn hạn chế.

### III. GIẢI PHÁP ĐỀ XUẤT

Ở phần này, chúng tôi sẽ trình bày giải pháp đề xuất. Trước hết, một cây tứ phân được tạo thành từ ảnh gốc  $I$  cũng theo cách của Fuqiang Di. Để giải quyết khả năng nhúng hạn chế ở giải pháp PVO cũng như QPVO, các khối lớn nhất của cây sau khi tạo thành sẽ được chia thành các khối có trùng lặp với kích thước  $2 \times 2$  như Hình 4.

$x$	$P_2$
$P_3$	$P_4$

Hình 4. Khối kích thước  $2 \times 2$

#### A. Dự báo giá trị cho điểm ảnh đầu tiên trong mỗi khối

Ở bước này, các điểm ảnh  $P_2, P_3, P_4$  được dùng để dự báo cho điểm ảnh  $x$ . Ba điểm ảnh này tạo thành một véc tơ ngữ cảnh  $P = \{P_2, P_3, P_4\}$  và  $x$  được dự báo bởi giá trị lớn nhất ( $\max(P)$ ) hoặc giá trị nhỏ nhất ( $\min(P)$ ) của véc tơ ngữ cảnh. Điểm ảnh  $x$  sẽ được dự báo trong 2 trường hợp: các giá trị trong  $P$  là khác nhau và các giá trị trong  $P$  giống nhau. Ở trường hợp thứ nhất, nếu  $x$  lớn hơn  $\max(P)$  thì giá trị dự báo của  $x$  chính là  $\max(P)$ , nếu  $x$  nhỏ hơn  $\min(P)$  thì giá trị dự báo của  $x$  chính là  $\min(P)$ , ngược lại thì không dự báo cho  $x$ . Trong trường hợp thứ hai, giả sử  $P_2 = P_3 = P_4 = VP$ , nếu  $VP = 254$  thì giá trị dự báo cho  $x$  là 254, nếu  $VP \neq 254$  và  $x \leq VP$  thì giá trị dự báo là  $VP$ , ngược lại thì không dự báo, nghĩa là khối này được bỏ qua, không dùng để nhúng dữ liệu. Để tránh tình huống vượt ngưỡng hoặc dưới ngưỡng thì các giá trị điểm ảnh được điều chỉnh trước khi dự báo. Cụ thể, nếu giá trị điểm ảnh là 0 hoặc 255 thì được điều chỉnh thành 1 hoặc 254. Những điều chỉnh này sẽ được ghi nhận bởi bản đồ định vị  $LM$ . Giá trị dự báo  $\hat{x}$  cho điểm ảnh  $x$  của mỗi khối được xác định như công thức (9).

$$\hat{x} = \begin{cases} \max(P) & \text{nếu } \max(P) \neq \min(P) \text{ và } x \geq \max(P) \\ \min(P) & \text{nếu } \max(P) \neq \min(P) \text{ và } x \leq \min(P) \\ VP & \text{nếu } \max(P) = \min(P) \text{ và } VP = 254 \\ VP & \text{nếu } \max(P) = \min(P), 1 \leq VP \leq 254 \text{ và } x \leq VP \end{cases} \quad (9)$$

#### B. Thuật toán nhúng tin

+ Đầu vào: ảnh gốc  $I$ , thông tin mật  $B$ .

+ Đầu ra: ảnh mang tin mật  $I'$ .

Bước 1: lưu lại các bit ít quan trọng nhất ( $LSB$ ) của  $24 + \log_2(n \times m)$  điểm ảnh ở dòng cuối cùng của ảnh gốc tạo thành chuỗi  $SLSB$ . Kết hợp chuỗi  $SLSB$  đó với thông tin mật  $B$  để tạo thành dữ liệu nhúng  $S$ .

Bước 2: phân đoạn ảnh gốc theo cấu trúc cây giống như cách của Fuqiang Di và cộng sự.

Bước 3: dựa vào cây thu được ở bước 2, chọn khối thuộc nút con cấp 1 của cây để tiến hành nhúng tin như sau:

- Bước 3.1: chia khối thuộc nút con cấp 1 có kích thước  $n \times m$  thành các khối  $P$  có kích thước  $2 \times 2$ . Trong mỗi khối  $P$  đó, tiến hành dự báo cho điểm ảnh đầu tiên  $x$  bằng 3 điểm ảnh còn lại theo công thức (9).
- Bước 3.2: với mỗi khối mà điểm ảnh dự báo được, tiến hành tính sự sai khác của điểm ảnh gốc với điểm ảnh dự báo bằng công thức (10).

$$e = x - \hat{x} \quad (10)$$

Trong trường hợp các giá trị điểm ảnh của khối là khác nhau, chỉ điểm ảnh có giá trị bằng với điểm ảnh dự báo mới được sử dụng để nhúng dữ liệu. Nếu điểm ảnh được dự báo bởi giá trị lớn nhất của 3 điểm ảnh còn lại thì nó không thay đổi giá trị khi bit thông tin mật là 0 và sẽ tăng 1 khi bit thông tin mật là 1. Ngược lại, giá trị điểm ảnh sẽ giảm 1 hoặc giữ nguyên khi nhúng bit 1 hoặc bit 0 nếu điểm ảnh được dự báo bởi giá trị nhỏ nhất của các điểm ảnh còn lại trong khối. Còn các điểm ảnh mà có giá trị không bằng với điểm ảnh dự báo thì sẽ được dịch chuyển để đảm bảo cho việc phục hồi tin mật và ảnh gốc. Như vậy, giá trị sai khác  $e$  sẽ được mở rộng để nhúng dữ liệu trong trường hợp  $\max(P) \neq \min(P)$  như sau, với  $s = \{0, 1\}$  chính là thông tin mật cần nhúng:

$$e' = \begin{cases} e + s, \text{ nếu } \hat{x} = \max(P), x = \hat{x} \\ e + 1, \text{ nếu } \hat{x} = \max(P), x > \hat{x} \\ e - s, \text{ nếu } \hat{x} = \min(P), x = \hat{x} \\ e - 1, \text{ nếu } \hat{x} = \min(P), x < \hat{x} \end{cases} \quad (11)$$

Trong trường hợp các điểm ảnh của khối là giống nhau ( $\max(P) = \min(P) = VP$ ) thì giá trị sai khác  $e$  sẽ được mở rộng để nhúng dữ liệu như sau:

$$e' = \begin{cases} e - s, & \text{nếu } \hat{x} = VP = 254 \\ e - s, & \text{nếu } \hat{x} = VP, x = \hat{x} \\ e - 1, & \text{nếu } \hat{x} = VP, x < \hat{x} \end{cases} \quad (12)$$

Sau khi nhúng thông tin mật, giá trị điểm ảnh gốc sẽ được điều chỉnh để tạo thành điểm ảnh đã nhúng dữ liệu theo công thức (13).

$$x' = \hat{x} + e' \quad (13)$$

Lặp lại bước 3.2 cho đến khi thông tin mật được nhúng hết hoặc các khối của ảnh gốc đều được nhúng tin hoặc dịch chuyển.

Bước 4: nhúng thông tin bổ trợ cho việc trích tin vào ảnh gốc.

Tiến hành nhúng các thông tin hỗ trợ như ngưỡng phân đoạn ảnh  $T$ , vị trí nhúng cuối cùng và bản đồ định vị vào các  $LSB$  của  $24 + \log_2(n \times m)$  điểm ảnh ở dòng cuối cùng của ảnh gốc nhằm giúp cho người nhận có thể trích tin mật chính xác và phục hồi hoàn toàn ảnh gốc. Cuối cùng, ta thu được ảnh mang tin mật  $I'$ .

### C. Thuật toán rút tin

+ Đầu vào: ảnh mang tin mật  $I'$ .

+ Đầu ra: ảnh phục hồi  $I'' = I$ , thông tin mật  $B$ .

Bước 1: tiến hành trích ngưỡng phân đoạn ảnh  $T$ , vị trí nhúng cuối cùng và bản đồ định vị ở  $LSB$  của  $24 + \log_2(n \times m)$  điểm ảnh ở dòng cuối cùng của ảnh gốc.

Bước 2: phân đoạn ảnh theo cấu trúc cây giống như thuật nhúng tin.

Bước 3: dựa vào cây thu được ở bước 2, chọn khối thuộc nút con cấp 1 của cây để tiến hành trích tin và phục hồi ảnh gốc như sau:

- Bước 3.1: chia khối thuộc nút con cấp 1 có kích thước  $n \times m$  thành các khối  $P'$  có kích thước  $2 \times 2$ . Trong mỗi khối  $P'$  đó, tiến hành dự báo cho điểm ảnh đầu tiên  $x'$  bằng 3 điểm ảnh còn lại theo công thức (9).
- Bước 3.2: với mỗi khối mà điểm ảnh dự báo được, tiến hành tính sự sai khác của điểm ảnh gốc  $x'$  với điểm ảnh dự báo  $\hat{x}'$  bằng công thức (10).

$$e' = x' - \hat{x}' \quad (14)$$

Với khối  $P'$  mà có các giá trị điểm ảnh khác nhau ( $\max(P') \neq \min(P')$ ) thì giá trị dự báo chính là  $\max(P')$  hoặc  $\min(P')$ . Vì vậy, trong trường hợp giá trị dự báo là  $\max(P')$ , nếu sự sai khác  $e'$  là 0 thì sẽ trích được bit thông tin mật là 0 và giá trị điểm ảnh phục hồi chính bằng giá trị điểm ảnh được nhúng tin. Ngược lại, nếu sự sai khác  $e'$  là 1 thì bit thông tin mật là 1 và giá trị điểm ảnh phục hồi bằng giá trị điểm ảnh đã nhúng tin trừ đi 1. Còn khi sự sai khác này lớn hơn 1 thì điểm ảnh đã nhúng được cộng thêm 1 để phục hồi điểm ảnh gốc nhưng nó không mang tin mật. Tương tự vậy, khi giá trị dự báo là  $\min(P')$ , nếu  $e'$  là 0 thì sẽ rút được bit 0 và giá trị điểm ảnh không đổi, nếu  $e'$  là -1 thì rút được bit tin mật là 1 và giá trị điểm ảnh sẽ tăng 1, nếu  $e'$  nhỏ hơn -1 thì giá trị điểm ảnh cũng tăng 1 nhưng không có tin mật. Vì vậy, khi  $\max(P') \neq \min(P')$  ta có công thức trích tin như sau:

$$e'' = e = \begin{cases} e', s = 0 & \text{nếu } \hat{x}' = \max(P'), x' = \hat{x}' \\ e' - 1, s = 1 & \text{nếu } \hat{x}' = \max(P'), x' = \hat{x}' + 1 \\ e' - 1 & \text{nếu } \hat{x}' = \max(P'), x' > \hat{x}' + 1 \\ e', s = 0 & \text{nếu } \hat{x}' = \min(P'), x' = \hat{x}' \\ e' + 1, s = 1 & \text{nếu } \hat{x}' = \min(P'), x' = \hat{x}' - 1 \\ e' + 1 & \text{nếu } \hat{x}' = \min(P'), x' < \hat{x}' - 1 \end{cases} \quad (15)$$

Cũng giống như trường hợp trên, khi  $\max(P') = \min(P') = VP'$  ta có công thức trích tin như sau:

$$e'' = e = \begin{cases} e', s = 0 & \text{nếu } \hat{x}' = VP' = 254, x' = \hat{x}' \\ e' + 1, s = 1 & \text{nếu } \hat{x}' = VP' = 254, x' = \hat{x}' - 1 \\ e', s = 0 & \text{nếu } \hat{x}' = VP', x' = \hat{x}' \\ e' + 1, s = 1 & \text{nếu } \hat{x}' = VP', x' = \hat{x}' - 1 \\ e' + 1 & \text{nếu } \hat{x}' = VP', x' < \hat{x}' - 1 \end{cases} \quad (16)$$

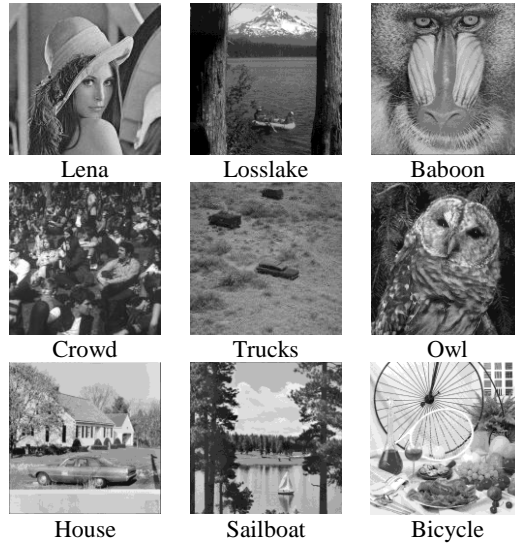
Giá trị điểm ảnh đã nhúng tin sau đó được điều chỉnh để khôi phục thành giá trị điểm ảnh gốc ban đầu bằng công thức (17)

$$x'' = x = \hat{x}' + e'' \quad (17)$$

Lặp lại bước 3.2 cho đến khi tất cả các khối  $P'$  được xử lý, ta thu được ảnh  $I''$ . Bằng cách kết hợp ảnh  $I''$  này và bản đồ định vị  $LM$  đã trích ở bước 1, ta sẽ phục hồi được ảnh gốc  $I$ .

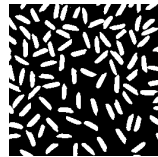
#### IV. KẾT QUẢ THỰC NGHIỆM

Ở phần này, chúng tôi trình bày kết quả thực nghiệm của giải pháp đề xuất. Hiệu quả của giải pháp này cũng được so sánh với giải pháp của Li [9] và Fu [11]. Tập dữ liệu dùng để thực nghiệm cho phương pháp đề xuất là các ảnh 8 bit phổ biến (tải về từ cơ sở dữ liệu ảnh <http://sipi.usc.edu/database/>, <http://decsai.ugr.es/cvg/CG/base.htm>) với kích thước  $512 \times 512$  bao gồm: Lena, Losslake, Baboon, Crowd, Truck, Owl, House, Sailboat, Bicycle như Hình 5.



Hình 5. Tập dữ liệu thực nghiệm

Trong phần thực nghiệm này, chúng tôi chọn  $T = R$  để so sánh khả năng nhúng cũng như chất lượng ảnh với phương pháp [9] và [11]. Chuỗi thông tin mật dùng để nhúng được tạo ra từ ảnh nhị phân Rice có kích thước  $256 \times 256$  như Hình 6.



Hình 6. Ảnh tạo thông tin mật

Bảng 1 thể hiện sự so sánh chất lượng ảnh (PSNR) của 3 giải pháp khi mức nhúng là 10.000 bit. Ở mức nhúng này, cả 3 giải pháp đều thực hiện được trên toàn bộ tập ảnh thực nghiệm do tất cả các ảnh đều có khả năng nhúng trên 10.000 bit. Trong đó, giải pháp đề xuất cho chất lượng ảnh rất cao và tốt hơn giải pháp của Li [9] cũng như giải pháp của Fu [11]. Cụ thể chất lượng ảnh trung bình trên tập ảnh thực nghiệm đạt 61,07 dB so với 58,91 dB trong [9] và 59,84 dB trong [11].

Bảng 1. So sánh giá trị trung bình của PSNR ở mức nhúng 10.000 bit

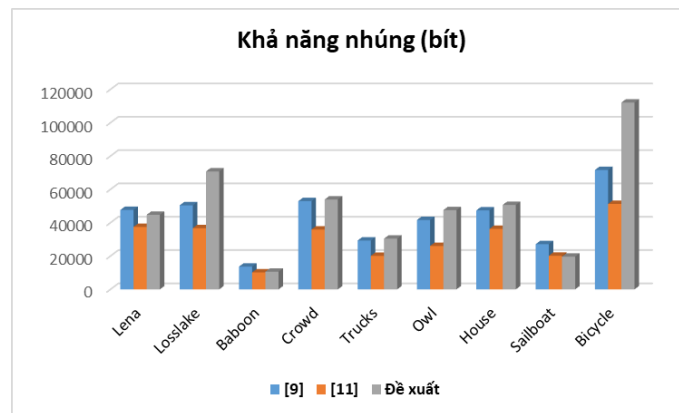
Ảnh	[9]	[11]	Đề xuất
Lena	60,61	60,71	61,16
Losslake	60,24	61,71	64,07
Baboon	52,58	54,38	54,32
Crowd	59,06	60,57	61,43
Trucks	56,74	56,70	57,77
Owl	58,05	58,44	60,65
House	63,50	64,42	65,11
Sailboat	56,92	59,32	59,31
Bicycle	62,52	62,33	65,78
<b>Trung bình</b>	<b>58,91</b>	<b>59,84</b>	<b>61,07</b>

Kết quả thực nghiệm ở Bảng 2 cho thấy, ở cùng mức nhúng 20.000 bit, cả 3 giải pháp có sự biến thiên đôi chút về chất lượng ảnh. Tuy nhiên, phương pháp đề xuất vẫn đạt PSNR trung bình tốt hơn cả hai phương pháp còn lại. Ví dụ ở ảnh Lena thì PSNR của giải pháp [11] là cao nhất, kế đến là giải pháp đề xuất và sau cùng là giải pháp [9]. Nhưng ở các ảnh còn lại như Losslake, Crowd, Trucks thì giải pháp đề xuất lại cho chất lượng ảnh cao nhất, trong khi giải pháp [9] luôn cho chất lượng ảnh thấp nhất.

Bảng 2. So sánh giá trị trung bình của PSNR ở mức nhúng 20.000 bit

Ảnh	[9]	[11]	Đề xuất
Lena	56,65	57,08	56,78
Losslake	57,25	58,45	60,92
Crowd	55,89	57,97	58,63
Trucks	53,52	54,14	56,33
Owl	55,06	55,73	57,69
House	57,37	59,35	62,12
Bicycle	59,12	59,17	62,31
<b>Trung bình</b>	<b>56,41</b>	<b>57,41</b>	<b>59,25</b>

Khả năng nhúng tối đa ở các giải pháp được thể hiện ở Hình 7. Khả năng nhúng cao nhất vẫn là giải pháp đề xuất với lượng tin nhúng trung bình khoảng 48.955 bit. Do không hạn chế vùng nhúng tin mật nên giải pháp [9] đạt khả năng nhúng cao thứ hai (trung bình khoảng 42.425 bit) và cuối cùng là [11] với lượng tin nhúng trung bình rơi vào khoảng 30.478 bit.



Hình 7. So sánh khả năng nhúng tối đa

Do giải pháp đề xuất có khả năng nhúng tối đa cao hơn các giải pháp khác nên chất lượng ảnh sẽ thấp hơn đôi chút so với [11] (trung bình khoảng 0.5dB) nhưng luôn tốt hơn [9] và chất lượng hình ảnh của phương pháp đề xuất ở mức nhúng tối đa luôn trên 53 dB. Tuy nhiên điều này không đáng kể vì khi chất lượng ảnh trên 35 dB thì mắt thường đã không phân biệt được. Mặc dù giải pháp đề xuất cho chất lượng ảnh thấp hơn khoảng 0,5 dB nhưng lại làm tăng khả năng nhúng lên khoảng 1,6 so với [11] nên phương pháp của chúng tôi là tốt hơn. Còn đối với phương pháp [9] thì giải pháp này tốt hơn cả về khả năng nhúng (tăng khoảng 1.2 lần) lẫn chất lượng ảnh. Chất lượng ảnh cụ thể khi nhúng tối đa được thể hiện ở Bảng 3.

Bảng 3. So sánh chất lượng ảnh khi nhúng tối đa

Ảnh	[9]	[11]	Đề xuất
Lena	52,43	54,46	53,39
Losslake	52,53	55,19	54,91
Baboon	51,46	54,24	54,05
Crowd	52,61	55,73	54,89
Trucks	51,87	53,98	54,10
Owl	52,24	54,57	54,30
House	52,42	55,31	54,57
Sailboat	51,80	54,05	54,43
Bicycle	53,35	54,99	54,48
<b>Trung bình</b>	<b>52,30</b>	<b>54,72</b>	<b>54,35</b>

## V. KẾT LUẬN

Trong bài báo này, chúng tôi đề xuất một phương pháp mới về giấu tin thuận nghịch để giấu thông tin mật vào những vùng thích hợp nhất trong ảnh mang tin. Ở giải pháp đề xuất chúng tôi dùng một giải thuật để tìm ra các vùng mịn thích hợp để nhúng tin và phân cấp các vùng này theo mô hình cây tứ phân. Sau đó từ mô hình cây tứ phân tìm được, tiến hành nhúng dữ liệu vào các nút thích hợp nhất trên cây sao cho khả năng nhúng cao nhất và chất lượng ảnh tốt nhất. Giải pháp này kết hợp với phương pháp xây dựng và dịch chuyển lưu đồ để giấu tin nhằm đạt được khả năng giấu tin cao và chất lượng ảnh sau mang tin tốt hơn. Nhờ kết hợp mô hình cây tứ phân, chia khối trùng lặp và dịch chuyển lưu đồ mà thông tin được giấu vào ảnh tăng khoảng 1,6 lần so với phương pháp [11] và khoảng 1,2 lần so với phương pháp [9] trong khi chất lượng ảnh mang tin vẫn rất cao. Kết quả thực nghiệm đã minh chứng phương pháp đề xuất có lượng tin được giấu cao hơn các phương pháp cũ và chất lượng ảnh khi nhúng tối đa luôn duy trì ở mức trên

53 dB. Bên cạnh đó, giải pháp này có thể áp dụng vào việc giấu tin trong các lĩnh vực đặc biệt như ảnh y khoa, bản đồ quân sự vì cũng đảm bảo được tính thuận nghịch.

### TÀI LIỆU THAM KHẢO

- [1] J. Tian, "Reversible data embedding using a difference expansion", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, No. 8, pp. 890-896, Aug. 2003, doi: 10.1109/TCSVT.2003.815962.
- [2] A. M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform", IEEE Transactions on Image Processing, Vol. 13, No. 8, pp. 1147-1156, Aug. 2004, doi: 10.1109/TIP.2004.828418.
- [3] Wei-Liang Tai, Chia-Ming Yeh, and Chin-Chen Chang, "Reversible Data Hiding Based on Histogram Modification of Pixel Differences", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 19, No. 6, pp. 906-910, Jun. 2009, doi: 10.1109/TCSVT.2009.2017409.
- [4] H. J. Kim, V. Sachnev, Y. Q. Shi, J. Nam, and H. Choo, "A Novel Difference Expansion Transform for Reversible Data Embedding", IEEE Transactions on Information Forensics and Security, Vol. 3, No. 3, pp. 456-465, Sep. 2008, doi: 10.1109/TIFS.2008.924600.
- [5] Zhicheng Ni, Y. Q. Shi, N. Ansari, and Wei Su, "Reversible data hiding", in Proceedings of the 2003 International Symposium on Circuits and Systems, 2003. ISCAS '03., Bangkok, Thailand, 2003, Vol. 2, pp. II-912-II-915, doi: 10.1109/ISCAS.2003.1206123.
- [6] S. Lee, Y. Suh, and Y. Ho, "Reversible Image Authentication Based on Watermarking", in 2006 IEEE International Conference on Multimedia and Expo, Toronto, ON, Canada, Jul. 2006, pp. 1321-1324, doi: 10.1109/ICME.2006.262782.
- [7] X. Gao, L. An, X. Li, and D. Tao, "Reversibility improved lossless data hiding", Signal Processing, Vol. 89, No. 10, pp. 2053-2065, Oct. 2009, doi: 10.1016/j.sigpro.2009.04.015.
- [8] V. Sachnev, Hyoung Joong Kim, Jeho Nam, S. Suresh, and Yun Qing Shi, "Reversible Watermarking Algorithm Using Sorting and Prediction", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 19, No. 7, pp. 989-999, Jul. 2009, doi: 10.1109/TCSVT.2009.2020257.
- [9] X. Li, J. Li, B. Li, and B. Yang, "High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion", Signal Processing, Vol. 93, No. 1, pp. 198-205, Jan. 2013, doi: 10.1016/j.sigpro.2012.07.025.
- [10] X. Qu and H. J. Kim, "Pixel-based pixel value ordering predictor for high-fidelity reversible data hiding", Signal Processing, Vol. 111, pp. 249-260, Jun. 2015, doi: 10.1016/j.sigpro.2015.01.002.
- [11] F. Di, M. Zhang, X. Liao, and J. Liu, "High-fidelity reversible data hiding by Quadtree-based pixel value ordering", Multimed Tools Appl, Vol. 78, No. 6, pp. 7125-7141, Mar. 2019, doi: 10.1007/s11042-018-6469-4.
- [12] S. Lee, C. D. Yoo, and T. Kalker, "Reversible Image Watermarking Based on Integer-to-Integer Wavelet Transform", IEEE Trans. Inform. Forensic Secur., Vol. 2, No. 3, pp. 321-330, Sep. 2007, doi: 10.1109/TIFS.2007.905146.
- [13] Xiang Wang, Xiaolong Li, Bin Yang, and Zongming Guo, "Efficient Generalized Integer Transform for Reversible Watermarking", IEEE Signal Process. Lett., Vol. 17, No. 6, pp. 567-570, Jun. 2010, doi: 10.1109/LSP.2010.2046930.
- [14] D. Coltuc, "Low distortion transform for reversible watermarking", IEEE Trans. on Image Process., Vol. 21, No. 1, pp. 412-417, Jan. 2012, doi: 10.1109/TIP.2011.2162424.
- [15] N. K. Sao, Đ. V. Tuấn và P. V. Át, "Giấu tin thuận nghịch sử dụng các thuộc tính của ngữ cảnh dự báo để loại bỏ bản đồ định vị", Res. Dev. Info. Commun. Tech. J., Dec. 2018, doi: 10.32913/rd-ict.vol3.no40.598.

## REVERSIBLE DATA HIDING BASED ON HISTOGRAM SHIFTING AND QUADTREE

**Tram Hoang Nam, Vo Phuc Hung, Huynh Van Thanh, Nguyen Thai Son**

**ABSTRACT:** This paper proposed a reversible data hiding method based on Quadtree and histogram shifting. First, image segmentation algorithm is implemented to segment the areas of the image according to the Quadtree model. Then, we will embed secret data into the most appropriate nodes in the tree so that the highest capacity and best image quality. This scheme is also combined with histogram shifting method to achieve better visual quality and higher hiding capacity than previous methods. Experimental results showed that the proposed method is 1.6 times better than in the embedding capacity, while the image quality remains unchanged in comparison with the previous schemes.

**Keywords:** Data hiding, reversible, histogram, quadtree.