

IOT KẾT HỢP COMPRESSIVE SENSING ỨNG DỤNG CHO HỆ THỐNG GIÁM SÁT

IOT INCORPORATES COMPRESSIVE SENSING APPLICATION FOR SURVEILLANCE SYSTEM

Đào Phương Tùng^{1a}, Võ Ngọc Vinh^{2b}, Nguyễn Thanh Sơn^{2c*}

¹Khoa Công Nghệ, Trường Đại học Công nghệ Đồng Nai

²Trường Đại học Lạc Hồng, Đồng Nai, Việt Nam

^a daophuongtung@dentu.edu.vn, ^b vongocvinh@lhu.edu.vn, ^{c*} nguyenthanson@lhu.edu.vn

TÓM TẮT: Trong những năm gần đây các hệ thống camera giám sát ngày càng nhiều, dữ liệu cần lưu trữ ngày càng lớn. Chính vì vậy việc quá tải cơ sở dữ liệu đang xảy ra trên các ứng dụng lưu trữ. Dữ liệu video giám sát trong các công ty, bệnh viện, trường học nếu được xử lý trước khi lưu trữ và tải lên đám mây sẽ tiết kiệm bộ nhớ đáng kể. Kỹ thuật lấy mẫu nén 'Compressive sensing' như một bước đệm cho kỹ thuật tiền xử lý tín hiệu video, kỹ thuật này cho phép chỉ cần rất ít mẫu ở đầu phát và sử dụng thuật toán nâng cao để khôi phục chính xác dữ liệu tại máy thu.

TỪ KHÓA: IOT, Compressive sensing, Hệ thống giám sát.

ABSTRACT: In recent years, with more and more surveillance camera systems, the data that needs to be stored is getting larger and larger. That's why database overload is happening on hosted applications. Video surveillance data in companies, hospitals, schools, if processed before storing and uploading to the cloud, will save significant memory. 'Compressive sensing' as a stepping stone to video signal preprocessing, which allows very few samples at the playhead and uses advanced algorithms for accurate data recovery at the receiver.

KEYWORDS: IOT, Compressive sensing, Monitoring system.

1. GIỚI THIỆU

Trong những năm gần đây, đã có nhiều nghiên cứu về sự đa dạng của thiết bị cảm biến và kích thước của dữ liệu thu được từ các ứng dụng chụp ảnh y tế đến giám sát video... Vấn đề khó khăn chung đó là quá tải dữ liệu đang xảy ra trên các ứng dụng này. Kỹ thuật 'compressive sensing' (CS) của Candès, Romberg, Tao [1], [2] and Donoho [3] tạo thành một bước đệm cho kỹ thuật tiền xử lý sử dụng các tín hiệu lấy mẫu thưa thớt hiệu quả hơn nhiều so với lý thuyết lấy mẫu theo định lý Nyquist đã được thiết lập [4]. Lý thuyết mới nổi lên gần đây là Compressed Sensing (CS) [5], [6] được coi là giải pháp tốt nhất để giải quyết tắc nghẽn trong công nghệ chuyển đổi tương tự số, CS không chỉ làm giảm các yêu cầu cho các thiết bị tốc độ cao, mà còn tiết kiệm rất nhiều dung lượng lưu trữ, phá vỡ giới hạn của định lý lấy mẫu Nyquist, tốc độ lấy mẫu thấp hơn nhiều trong lý thuyết CS.

Việc lưu trữ và xử lý dữ liệu trong các thiết bị IoT là rất quan trọng. Do các thiết bị IoT có thể bị các cuộc tấn công mạng (ví dụ, ai đó có thể lấy cắp thiết bị và dữ liệu được lưu trữ), hệ thống sẽ dựa vào cơ chế lưu trữ phân tán và bảo mật cao như công nghệ sửa lỗi hiện đại (mã hóa mạng) chia nhỏ và mã hóa dữ liệu trước khi lưu trữ trong các thiết bị khác nhau và thậm chí với các 'bản sao được mã hóa' bổ sung cho độ mạnh của hệ thống. Vì vậy tín hiệu sẽ được bảo mật và cũng có thể lấy lại được dữ liệu đã mất.

Tương lai công nghệ truyền thông IoT trong hệ thống giám sát sử dụng CS có thể trở thành công nghệ sớm được ứng dụng vào thực tiễn. Trong nghiên cứu này, nhóm tác giả đã đưa ra nền tảng lý thuyết xử lý hình ảnh video, mã hóa mạng cơ bản, kết hợp với các thuật toán. Kết quả thu được từ mô hình mô phỏng thực nghiệm cho thấy hệ thống hoạt động có thể đạt được kết quả như mong muốn. Hy vọng kết quả này sẽ làm nền tảng cho những nghiên cứu

tiếp theo trong việc nâng cao, cải thiện khả năng ứng dụng của công nghệ truyền thông IoT trong hệ thống giám sát.

2. ỨNG DỤNG COMPRESSIVE SENSING CHO XỬ LÝ ẢNH

2.1 Cơ sở lý thuyết Compressed sensing

Lý thuyết CS được giới thiệu trong [2], [5] chỉ ra rằng, nếu tín hiệu đại diện thưa thớt trong một số miền biến đổi, nó có thể được lấy mẫu với tốc độ thấp hơn so với tốc độ Nyquist, nhưng vẫn có thể được khôi phục với kỹ thuật tối ưu hóa cao. Miền biến đổi có thể được biểu diễn bằng cơ sở trực giao.

Để xem xét các ý tưởng chính của CS, chúng tôi xét các giá trị thực, độ dài hữu hạn, tín hiệu rời rạc theo thời gian $x \in \mathbb{R}^N$ có thể được biểu thị bằng:

$$x = \sum_{i=1}^N \theta_i \psi_i = \Psi \theta \quad (1)$$

Để khôi phục lại tín hiệu x , chúng tôi không đo lường hoặc mã hóa θ một cách trực tiếp. Thay vào đó, chúng tôi đo lường và mã hóa tập hợp tín hiệu của $y_i = \langle x, \phi_i^T \rangle$ theo các hàm cơ sở thứ ϕ_i , $i \in \{1, 2, \dots, M\}$ Trong đó ϕ_i^T biểu thị chuyển vị ϕ_i và $\langle \cdot, \cdot \rangle$ biểu thị phần tử bên trong.

$$y = \Phi x = \Phi \Psi \theta \quad \text{or} \quad y = \Theta \theta \quad (2)$$

$\Theta = \Phi \Psi$ là ma trận kết hợp và Φ là vectơ cột $M \times N$. Thay vì sử dụng $M \times 1$ mẫu để tìm các hệ số trọng số, có thể sử dụng vector đo mẫu y . Vì số lượng phép đo M nhỏ hơn nhiều so với kích thước tín hiệu N , sự phục hồi của tín hiệu x từ phép đo y là một vấn đề chưa được xác định.

Received: 03, 7, 2021

Accepted: 05, 12, 2021

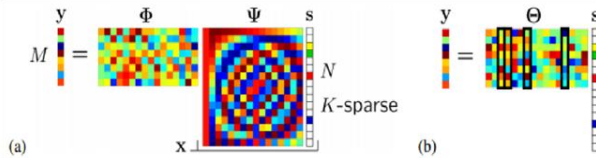
*Corresponding Author: Nguyễn Thanh Sơn

Email: nguyenthanson@lhu.edu.vn

Tuy nhiên, trong CS tín hiệu thưa thớt vẫn có thể được khôi phục lại từ thông qua tối ưu hóa phi tuyến bằng cách khai thác một ưu tiên x trong sự hình thành thưa thớt trên một số cơ sở Ψ . Một số thuật toán khôi phục đã được phát triển, nó bao gồm các kỹ thuật lập trình tuyến tính như Basis Pursuit (BP) [6], [7] hoặc Greedy Algorithms như Orthogonal Matching Pursuit (OMP) [2].

2.2 Compressed sensing ứng dụng cho xử lý ảnh

Các ví dụ thực hiện của CS áp dụng cho dữ liệu hình ảnh có sẵn trên trang web MathWorks [8]. Compressed sensing ứng dụng cho xử lý ảnh đơn đề xuất bởi Baraniuk [5] như sau:



Hình 1. Mô hình Compressive Sensing

(a) Quy trình lấy mẫu và nén tạo ra ma trận đo lường Gaussian ngẫu nhiên Φ và ma trận biến đổi cosin rời rạc (DCT) Ψ . Trên hình này, các vector hệ số s thưa thớt với $K_{sparse} = 4$. (b) Quy trình tạo ra ma trận $\Theta = \Phi \Psi$. Bốn cột được đánh dấu tương ứng với Θ . vector Y là kết hợp tuyến tính của bốn cột này.

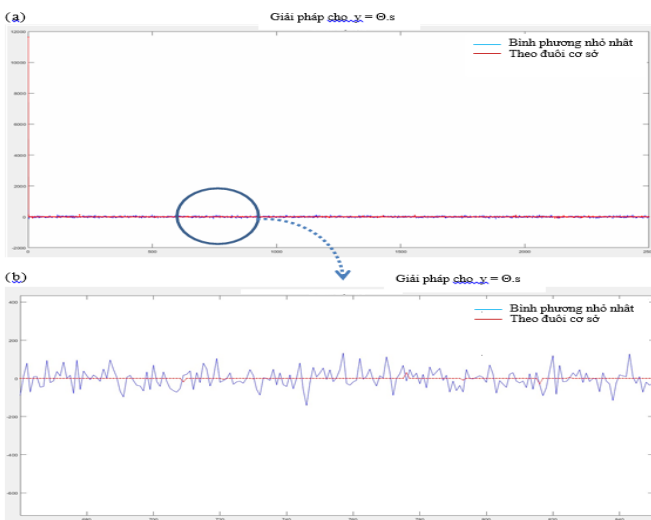
Đầu tiên, một ma trận đo lường Gauss ngẫu nhiên Φ không được tạo ra một cách thích nghi. Điều đó có nghĩa là ma trận $M \times N$ được tạo ra không phụ thuộc vào đầu vào $x \in R_N$. Hình ảnh ban đầu X sau đó được nén bằng cách nhân với Φ . Tín hiệu nén là:

$$y = \Phi \cdot X \in R_M \quad (3)$$

Sau đó, kết quả ma trận $\Theta = \Phi \Psi$ được thiết kế, trong đó Ψ là ma trận biến đổi cosin rời rạc ($N \times N$). Y vector là một tổ hợp tuyến tính của các cột K của các số không trong các vector hệ số thưa thớt.

$$s_2 = \Theta^{-1} \cdot y \quad (4)$$

Cuối cùng, thuật toán khôi phục được áp dụng để giải bài toán sai số nhỏ nhất $y = \Theta \cdot s$



Hình 2. Ước tính sai số nhỏ nhất $y = \Theta \cdot s$

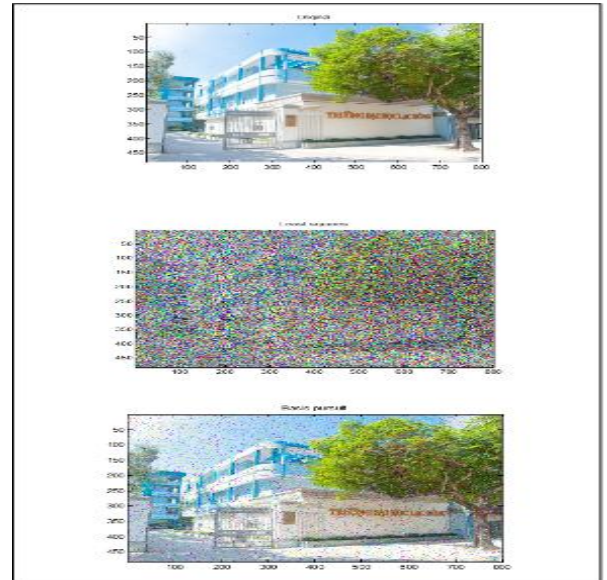
(a) Hiện thị các giải pháp cho $y = \Theta \cdot s$. Giải pháp theo đuổi tái tạo cơ bản trong màu đỏ, giải pháp ước tính sai số nhỏ nhất màu xanh lam. (b) Phóng to hình ảnh sau đó được xây dựng lại bằng lập trình tuyến tính.

2.3 Tính Toán PSNR

Chất lượng hình ảnh được đánh giá dựa vào PSNR và SSIM là tỷ lệ giữa công suất tối đa có thể có của một tín hiệu và độ lớn của nhiễu gây ảnh hưởng đến độ trung thực của ảnh. Do đó, nó có thể được sử dụng để đo lường chất lượng tái tạo để nén ảnh. Trong trường hợp này, tín hiệu là dữ liệu gốc và nhiễu là lỗi. Nó thường được thể hiện trong thang đo decibel logarit, với công thức sau:

$$PSNR = 20 \cdot \log_{10} \left(\frac{MAX_i}{\sqrt{MSE}} \right) \quad (5)$$

$$\text{Hoặc } PSNR = 20 \cdot \log_{10}(MAX_i) - 10 \cdot \log_{10}(MSE) \quad (6)$$



Hình 3. Ví dụ của 1 ảnh ứng dụng CS

Bên trên là hình ảnh gốc của Đại Học Lạc Hồng ($M = 40000$ pixel) bên dưới là hình được phục hồi từ $M = 1000$ phép đo ngẫu nhiên áp dụng CS.

MAX_i là giá trị pixel tối đa có thể có của hình ảnh, khi các pixel được biểu diễn với 8 bit cho mỗi mẫu. MSE - đại diện cho lỗi bình phương trung bình, đo lường sự khác biệt giữa hai hình ảnh, hình ảnh gốc x (b, c) và hình ảnh nén (a, b). MSE đại diện cho trung bình của các ô vuông của "lỗi" giữa hình ảnh gốc và hình ảnh được nén. Lỗi là tổng số mà theo đó giá trị của hình ảnh gốc khác với hình ảnh bị suy giảm. MSE là:

$$MSE = \frac{1}{M \cdot N} \sum_{i=1}^M \sum_{j=1}^N [\hat{x}(i, j) - x(i, j)]^2 \quad (7)$$

M là số hàng pixel trong hình ảnh và i là chỉ số của hàng đó; N là số cột pixel trong hình ảnh và j là chỉ số của cột đó.

2.4 Hạn chế của PSNR VÀ thay thế bằng SSIM

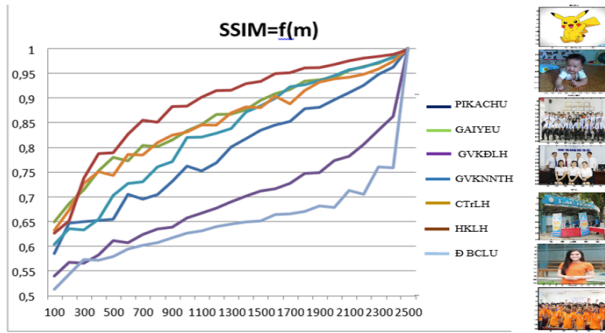
Hạn chế chính của chỉ số MSE / PSNR là nó dựa hoàn toàn vào so sánh số và không xem xét đến các yếu tố sinh học của hệ thống thị giác của con người. Tuy nhiên, đo lường chỉ số cấu trúc tương tự cục bộ (SSIM) lại làm được điều đó.

Như đã giải thích trong [9], [10], phương pháp "chính xác" duy nhất để định lượng chất lượng hình ảnh trực quan cho các ứng dụng mà hình ảnh cuối cùng được con người đánh giá là thông qua đánh giá chủ quan. Nhưng là một quá trình mất thời gian và tốn kém, các nghiên cứu đã được tiến hành để tìm ra một phương pháp đánh giá chất

lượng hình ảnh mục tiêu và phát triển các biện pháp định lượng có thể tự động dự đoán được chất lượng hình ảnh.

SSIM là chỉ số được sử dụng để đo lường sự tương đồng về cấu trúc giữa hai hình ảnh hoặc video kỹ thuật số. Trái với kỹ thuật MSE hoặc PSNR, SSIM không ước tính sai số tuyệt đối nhưng đánh giá tác động trực quan của độ sáng (phép chiếu sáng và phản xạ), độ tương phản và cấu trúc của hình ảnh. SSIM là một sự phát triển của MSE để các lỗi được định lượng theo khả năng hiển thị của chúng. Cách tiếp cận nhạy cảm về lỗi này dựa trên một số giả định và khái quát, tóm tắt bởi Wang và cộng sự [9]:

$$SSIM(x,y) = \frac{(2\mu_x \cdot \mu_y + C_1)}{(\mu_x^2 + \mu_y^2 + C_1)} \frac{(2\sigma_{xy} + C_2)}{(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (8)$$



Hình 4. Các giá trị SSIM cho các hình ảnh khác nhau được xây dựng lại với 100 đến 2500 phép đo

2.5 Thuật toán khôi phục tín hiệu

2.5.1 Phát hiện biên

Trong một hình ảnh, một biên có thể được định nghĩa là một tập hợp các điểm ảnh gần nhau, nơi có sự thay đổi đột ngột cường độ - màu xám hoặc màu sắc. Khi CS lấy mẫu ngẫu nhiên, phát hiện biên có thể được áp dụng sau bước CS để đánh giá việc tái thiết. Nó sẽ xác định nếu CS phục hồi các biên chính xác và một ngưỡng lấy mẫu có thể được thiết lập liên quan đến các kết quả dò tìm biên [10].

Có một số thuật toán phát hiện biên thường dùng như: Sobel, Prewitt, Canny, LoG là những bộ lọc thường được sử dụng.

$$\hat{h} = \arg \min_h \|y - \Theta h\|^2 \quad (9)$$

Ước lượng kênh cổ điển LS có thể thu được từ (8) như

$$\hat{h}_{LS} = \Theta^+ y = \Theta^+ (\Theta h + v) = h + \Theta^+ v \quad (10)$$

Ở đây Θ^+ là nghịch đảo của Θ và được cho bởi

$$\Theta^+ = (\Theta^H \Theta)^{-1} \Theta^H \quad (11)$$

Ở đây $(\cdot)^H$ và $(\cdot)^{-1}$ biểu thị các Hermitian và nghịch đảo ma trận, tương ứng.

2.5.2 Phương pháp khôi phục (BP)

Khi chúng ta xem xét các tín hiệu thưa thớt, quá trình phục hồi CS bao gồm tín hiệu thưa thớt x tạo ra các phép đo bằng cách xác định l_0 "chuẩn" của một vectơ $\|x\|_0$.

$$\hat{x} = \arg \min_{x \in \mathbb{R}^N} \|x\|_0 \quad \text{subject to } y = \Phi x \quad (12)$$

Tuy nhiên, thuật toán này có độ phức tạp tính toán tổ hợp, vì chúng ta phải kiểm tra liệu vectơ đo y có thuộc về khoảng cách của mỗi tập hợp các cột K hay Φ , $K = 1, 2, \dots, N$. Ý tưởng của Basis Pursuit là thay thế vấn đề thưa thớt khó khăn này bằng một vấn đề tối ưu hóa dễ dàng hơn. Một thay thế cho l_0 "chuẩn" được sử dụng trong (12) là sử dụng l_1 các chỉ tiêu, được định nghĩa là:

$$\hat{x} = \arg \min_{x \in \mathbb{R}^N} \|x\|_1 \quad \text{subject to } y = \Phi x \quad (13)$$

Khi l_1 là tiêu chuẩn lồi, (13) có thể được xem như là tiêu chuẩn lồi của (12). Nhờ sự lồi lõm, thuật toán này có thể được thực hiện như một chương trình tuyến tính [2], làm cho đa thức tính toán phức tạp của nó trong chiều dài tín hiệu. Tối ưu hóa (13) có thể được sửa đổi để cho phép nhiễu trong các phép đo $y = \Phi x + n$. Thuật toán này được gọi là denoising (BPDN).

$$\hat{x} = \arg \min_{x \in \mathbb{R}^N} \|x\|_1 \quad \text{subject to } \|y - \Phi x\|_2 \leq \sigma \quad (14)$$

Ở đây σ là ước lượng mức nhiễu trong dữ liệu.

• Định nghĩa thuật toán

Nguyên tắc của BP là tìm một biểu diễn của tín hiệu mà các hệ số l_1 có chỉ tiêu tối thiểu giải quyết vấn đề.

$$\min \|a\|_1 \quad \text{subject to } \Phi a = s \quad (15)$$

Trở thành vấn đề tối ưu hóa lồi, phi bậc hai sao cho một bản dịch thành vấn đề lập trình tuyến tính được thực hiện:

$$\min c^T x \quad \text{subject to } Ax = b, \quad x \geq 0 \quad (16)$$

Ở đây, $x \in \mathbb{R}^m$, $c^T x$ là hàm mục tiêu, $Ax = b$ là tập hợp các ràng buộc bình đẳng, và $x \geq 0$ là một tập hợp các giới hạn. Các vấn đề cơ bản theo đuổi (15) có thể được tái cấu trúc tương đương như một chương trình tuyến tính ở dạng tiêu chuẩn (16) bằng cách thực hiện các bản dịch sau đây:

$m \Leftrightarrow 2p$; $x \Leftrightarrow (u, v)$; $c \Leftrightarrow (l, l)$; $A \Leftrightarrow (\Phi, -\Phi)$; $b \Leftrightarrow s$
Do đó, giải pháp của (15) có thể thu được bằng cách giải một chương trình tuyến tính tương đương [11].

2.5.3 Thuật toán Orthogonal Matching Pursuit (OMP)

Thuật toán (OMP) là một phương pháp kỹ thuật để giải quyết vấn đề xấp xỉ thưa thớt.

So sánh với cơ sở Pursuit, những lợi thế chính của thuật toán này là tốc độ của nó và dễ thực hiện. Do đó, thuật toán OMP có thể được mô tả như bảng 1.

Bảng 1: Thuật toán OMP

Input	Output
- Signal b	- Approximation vector c matrix A and
- Stopping criterion	
Step 1	Start by setting $r_0 = b$, the time $t = 0$ and $V_0 = 0$
Step 2	Let $v_i = i$, where a_i give the solution of $\max < r_i, a_k >$. Where a_k are the row vectors of A
Step 3	Update the set V_t with $v_t : V_t = V_{t-1}$
Step 4	Solve the least-squares problem
Step 5	Calculate the new residual using c
Step 6	Set

Step 7 If the criterion has not been satisfied then return to step 2

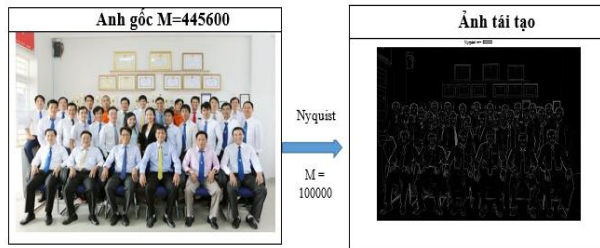
3. Mô phỏng

Trong chương này chúng tôi đưa ra ba mô phỏng để chứng minh tính hiệu quả của Compressive sensing trong xử lý video giám sát.

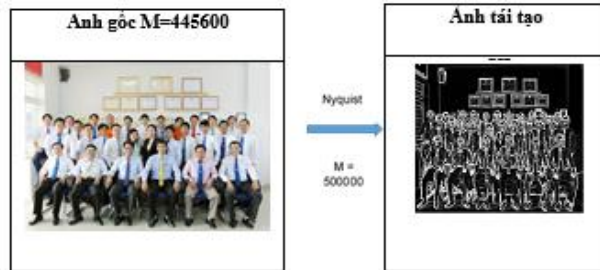
3.1 Mô phỏng thứ nhất

Trong trường hợp này chúng ta tái tạo một hình ảnh từ một camera giám sát trong hai trường hợp: Trường hợp thứ nhất là lấy mẫu theo phương pháp truyền thống Nyquist, trường hợp thứ hai lấy mẫu theo compressive sensing.

Trường hợp thứ nhất ta tái tạo hình ảnh giảng viên khoa Cơ điện - Điện tử Trường Đại học Lạc Hồng lấy mẫu bằng phương pháp Nyquist.



Hình 5. Hình ảnh tái tạo ở $M = 100000$, $SSIM=0.3072$

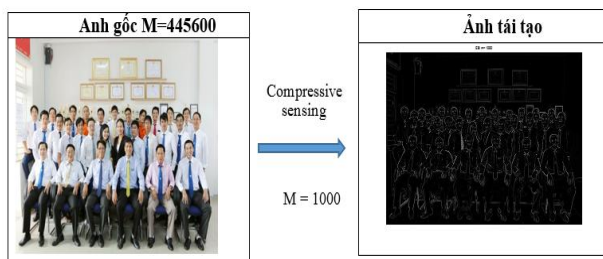


Hình 6. Lấy mẫu bằng phương pháp Nyquist
Hình ảnh tái tạo ở $M = 500000$, $SSIM=0.6656$

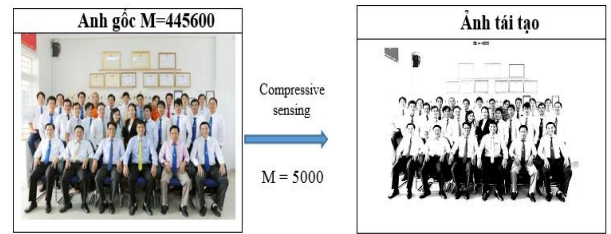


Hình 7. Lấy mẫu bằng phương pháp Nyquist
Hình ảnh tái tạo ở $M = 900000$, $SSIM=0.9740$

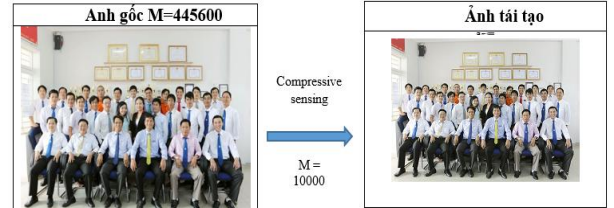
Trường hợp thứ hai ta tái tạo hình ảnh giảng viên khoa Cơ điện – Điện tử Trường Đại học Lạc Hồng lấy mẫu bằng phương pháp Compressive sensing.



Hình 8. Lấy mẫu bằng phương pháp compressive sensing.
Hình ảnh tái tạo ở $M = 100$, $SSIM=0.4033$



Hình 9. Lấy mẫu bằng phương pháp compressive sensing. Hình ảnh tái tạo ở $M = 200$, $SSIM=0.6082$



Hình 10. Lấy mẫu bằng phương pháp compressive sensing.
Hình ảnh tái tạo ở $M = 400$, $SSIM=0.9842$

Từ hai trường hợp trên ta thấy rằng khi sử dụng phương pháp lấy mẫu CS thì số lượng mẫu ít hơn rõ rệt so với Nyquist, cụ thể ở đây là số mẫu để khôi phục ảnh chỉ cần $m = 400$ là có thể khôi phục lại ảnh gốc. Hình ảnh khôi phục được đánh giá qua chỉ số và $SSIM=0.9842$.

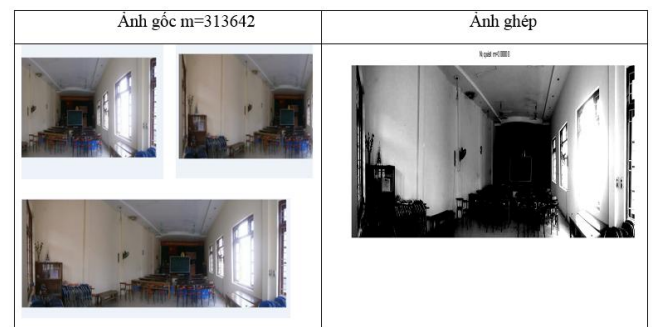
3.2 Mô phỏng thứ hai

Trong mô phỏng này chúng tôi thử nghiệm hai trường hợp, trường hợp thứ nhất truyền một hình ảnh ghép qua mạng lên vùng lưu trữ, trường hợp thứ hai nén hình ảnh ghép bằng CS sau đó mới truyền hình ảnh đó qua mạng lên vùng lưu trữ.

Trường hợp thứ nhất ta truyền một hình ảnh ghép bởi hai hình ảnh từ hai camera giám sát đặt ở hai vị trí khác nhau. Trong trường hợp này chúng ta dùng phương pháp Nyquist để lấy mẫu.



Hình 11. Lấy mẫu bằng phương pháp Nyquist. Ảnh ghép truyền qua mạng với số lượng mẫu $M=10000$. $SSIM=0.3086$



Hình 12. Lấy mẫu bằng phương pháp Nyquist. Ảnh ghép truyền qua mạng với số lượng mẫu $M=300000$. $SSIM=0.6086$

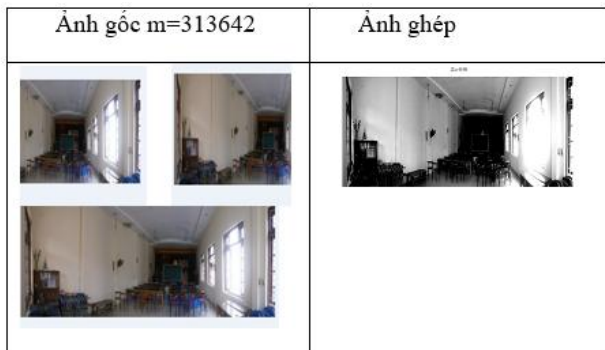


Hình 13. Lấy mẫu bằng phương pháp Nyquist. Ảnh ghép truyền với số lượng mẫu $M=600000$, $SSIM=0.9783$

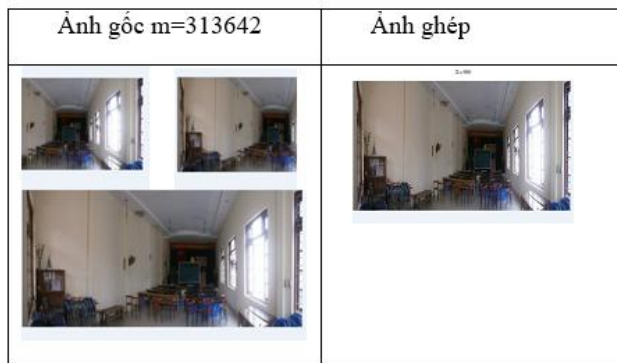
Trường hợp thứ hai ta thực hiện lấy mẫu nén CS cho hình ảnh ghép bởi hai ảnh từ hai camera giám sát đặt hai vị trí khác nhau, sau đó mới gửi hình ảnh này qua mạng lên vùng lưu trữ.



Hình 14. Lấy mẫu bằng phương pháp compressive sensing. Ảnh ghép truyền với số lượng mẫu $M=3000$, $SSIM=0.3756$



Hình 15. Lấy mẫu bằng phương pháp compressive sensing. Ảnh ghép truyền với số lượng mẫu $m=10000$, $SSIM=0.695$



Hình 16. Lấy mẫu bằng phương pháp compressive sensing. Ảnh ghép truyền với số lượng mẫu $m=50000$, $SSIM=0.9855$

Từ hai trường hợp trên ta thấy rằng khi sử dụng CS nén dữ liệu trước khi gửi chúng qua mạng làm giảm rất nhiều mẫu, cụ thể là ở hình 13 $M = 600000$ mẫu xuống còn 50000 mẫu ở hình 16. Đánh giá chất lượng ảnh ở hình 13 $SSIM=0.9783$ cũng thay đổi so với hình 16 là $SSIM=0.9855$.

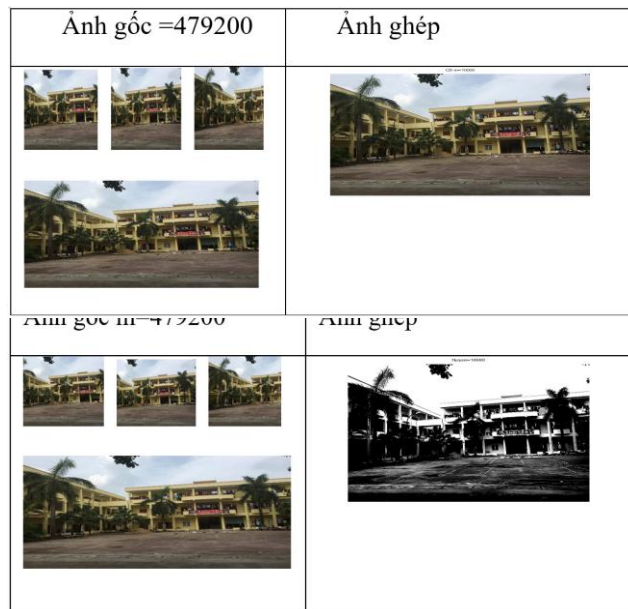
3.3 Mô phỏng thứ ba

Chúng tôi lấy 3 hình ảnh từ 3 camera giám sát đặt các vị trí khác nhau của ký túc xá, sau đó thực hiện ghép 3 hình ảnh lại thành một ảnh toàn cảnh, sau khi ghép lại thành một ảnh toàn cảnh chúng tôi thực hiện gửi chúng qua mạng lên vùng lưu trữ theo hai hình thức khác nhau, một là gửi toàn bộ dữ liệu thu được qua mạng lên vùng lưu trữ theo chuẩn Nyquist, hai là thực hiện lấy mẫu nén compressive sensing trước khi gửi chúng.

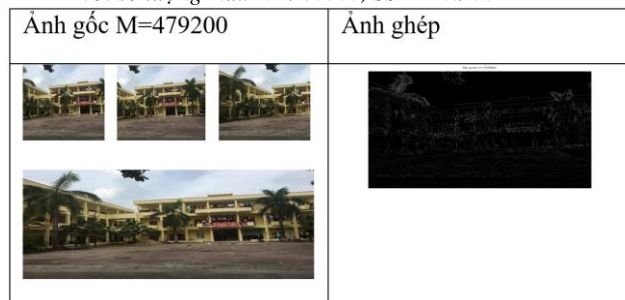
Trường hợp thứ nhất truyền hình ảnh qua mạng lấy mẫu theo phương pháp truyền thống Nyquist.

Hình 17. Lấy mẫu bằng phương pháp Nyquist. Ảnh ghép truyền với số lượng mẫu $m=20000$, $SSIM=0.3640$

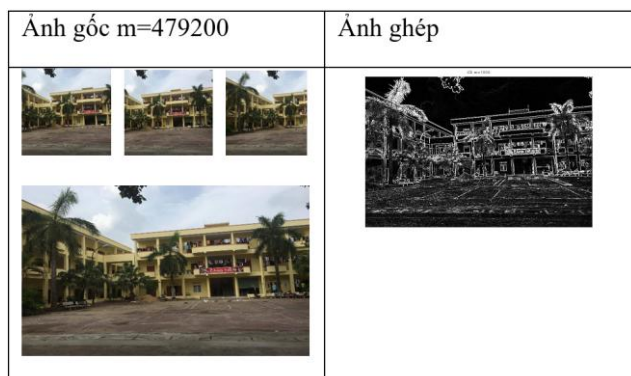
Hình 18. Lấy mẫu bằng phương pháp Nyquist, Ảnh ghép truyền với số lượng mẫu $m=300000$, $SSIM=0.6720$



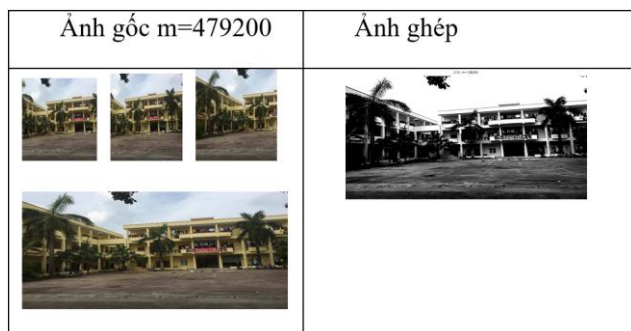
Hình 19. Lấy mẫu bằng phương pháp Nyquist. Ảnh ghép truyền với số lượng mẫu $m=980000$, $SSIM=0.9662$



Trường hợp thứ hai truyền hình ảnh qua mạng lấy mẫu nén theo phương pháp Compressive sensing.



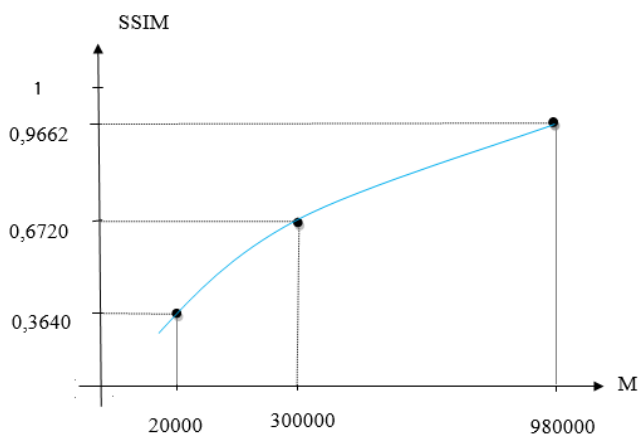
Hình 20. Lấy mẫu bằng phương pháp compressive sensing. Ảnh ghép truyền với số lượng mẫu m=1000, SSIM=0.3591



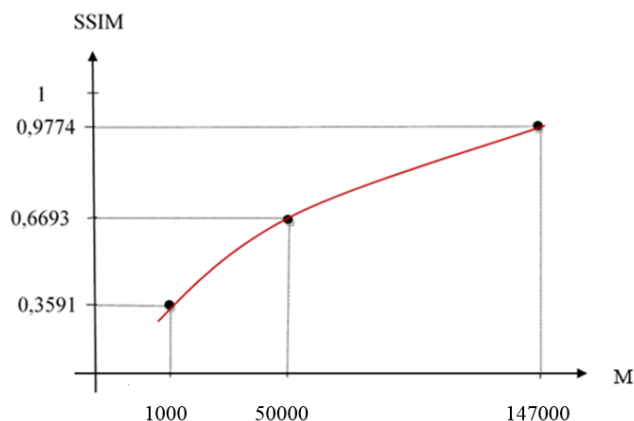
Hình 21: Lấy mẫu bằng phương pháp compressive sensing. Ảnh ghép truyền với số lượng mẫu m=50000, SSIM=0.6693



Hình 22: Lấy mẫu bằng phương pháp compressive sensing. Ảnh ghép truyền với số lượng mẫu m=147000, SSIM=0,9774



Hình 23: Lấy mẫu theo chuẩn Nyquist



Hình 24: Lấy mẫu theo CS

Qua các kết quả của các mô phỏng ta thấy phương pháp lấy mẫu nén CS trước khi truyền có số lượng mẫu ít hơn nhiều so với phương pháp lấy mẫu theo Nyquist, cụ thể ở hai biểu đồ hình vẽ H23 và H24, số lượng mẫu để khôi phục được ảnh ở Nyquist cần M=980000 mẫu, chỉ số SSIM =0,9662. Còn dùng phương pháp CS thì chỉ cần m=147000 là đã đủ để phục hồi được ảnh gốc, chỉ số SSIM cũng lớn hơn, SSIM=0,9774

4. KẾT LUẬN

Tóm lại, công nghệ truyền thông IoT trong hệ thống giám sát kết hợp với việc khai thác lý thuyết mới nổi Compress sensing đã đạt được những thành công nhất định. Dựa trên cả hai phân tích lý thuyết và kết quả thực nghiệm, nó đã được chứng minh rằng lý thuyết của CS là hoàn toàn phù hợp cho truyền dẫn IoT cho hệ thống giám sát.

Đóng góp chính của bài báo này là đề xuất một hệ thống truyền thông IoT trong hệ thống giám sát mới dựa trên CS có tiềm năng giảm đáng kể tỷ lệ lấy mẫu. Bằng cách này, thông tin được xử lý nhanh hơn, giảm tình trạng tắc nghẽn dữ liệu và nhất là nó giảm được dung lượng thông tin cần lưu trữ trong hệ thống giám sát.

5. TÀI LIỆU THAM KHẢO

- [1] Emmanuel Candès, Justin Romberg, Terence Tao. Stable Signal Recovery from Incomplete and Inaccurate
- [2] David L. Donoho. Compressed Sensing, Manuscript, September 2004.
- [3] Emmanuel Candès and Michael Wakin. An Introduction to Compressive Sampling. In IEEE Signal
- [4] Hui Wang, Joyce Liang and C.-C. Jay Kuo. Overview of Robust Video Streaming with Network coding. In Journal of Information Hiding and Multimedia Signal Processing, Volume 1, Issue 1, pages 36-50, January 2010
- [5] R. Baraniuk, Compressive Sensing, Lecture Notes in IEEE Signal Processing Magazine, Volume 24, July 2007.
- [6] G. Davis, S. Mallat, M. Avellaneda. Adaptive greedy approximations. In Constructive Approximation, Volume 13, Issue 1, pages 57-98, March 1997
- [7] Measurements. In Comm. Pure Appl. Math., Volume 59, pages 1207–1223, 2006. doi:10.1002/cpa.20124.
- [7] Duarte, Marco F.; Davenport, Mark A.; Wakin, Michael B.; Baraniuk, Richard G. Sparse signal detection from incoherent projections. In ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings. Vol. 3 2006. 1660651, 2006
- [8] Stuart Gibson. Simple compressed sensing example, May 2013. URL:

<http://www.mathworks.com/matlabcentral/fileexchange/41792-simplecompressed-sensing-example>.

[9] Analysis Of Image Compression Algorithm Using DCT Garg Maneesha Gupta, Dr.Amit Kumar Garg / International Journal of Engineering Research and Applications (IJERA) Vol. 2, Issue 1, Jan-Feb 2012

[10] MathWorks Documentation, Image Processing Toolbox (SSIM, Edge Detection). URL:

<http://fr.mathworks.com/help/images/>

[11] Marco F. Duarte, Mark A. Davenport, Michael B. Wakin, and Richard G. Baraniuk. Sparse signal detection from incoherent projections. In IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), III, 2006.