

ỨNG DỤNG THUẬT TOÁN CHỈNH SỬA VÀ TĂNG ĐỘ PHÂN GIẢI VÉCTƠ CHUYỂN ĐỘNG TRONG KỸ THUẬT TĂNG SỐ KHUNG HÌNH CHO VIDEO

Hoàng Đình Chiến

Trường Đại học Bách khoa, ĐHQG-HCM

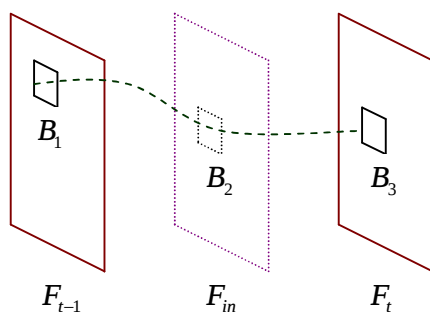
(Bài nhận ngày 06 tháng 03 năm 2009, hoàn chỉnh sửa chữa ngày 25 tháng 07 năm 2009)

TÓM TẮT: Bài báo ứng dụng giải thuật chỉnh sửa và tăng độ phân giải véctor chuyển động nhằm tăng chất lượng khung hình nội suy trong kỹ thuật tăng số khung hình video. Phương pháp ước lượng chuyển động 2 chiều và phương pháp ước lượng theo hình khối được áp dụng để xây dựng khung hình nội suy. Kết quả mô phỏng chứng minh hiệu quả của kỹ thuật chỉnh sửa và tăng độ phân giải véctor trong việc tăng độ trung thực của véctor chuyển động và giảm méo biến khối.

Từ khóa: kỹ thuật tăng số khung hình (FRUC), nội suy khung hình dùng bù chuyển động (MCFI), chỉnh sửa véctor chuyển động, tăng độ phân giải véctor chuyển động

1. GIỚI THIỆU VẤN ĐỀ

Do vấn đề giới hạn băng thông, độ phân giải của video thường bị cắt giảm cả về kích thước khung hình và số khung hình trong một giây trước khi được truyền qua kênh truyền hay trong các ứng dụng nén video. Việc giảm số khung hình trong một giây tương ứng với quá trình giảm độ phân giải về mặt thời gian trong khi giảm kích thước khung hình được xem là giảm độ phân giải về không gian. Tuy nhiên, video với tốc độ khung hình thấp thường có hiện tượng giật khung hình với chuyển động không trơn. Do vậy tại đầu thu hay bộ giải nén, các video thường được tăng tốc độ khung hình trước khi đưa ra màn hình nhằm tăng độ phân giải thời gian của video. Yêu cầu tăng tốc độ khung hình cũng tồn tại trong việc chuyển đổi giữa các định dạng video có tốc độ khung hình khác nhau, ví dụ như từ chuẩn phim ảnh (24, 25 hay 30 khung hình/s) sang chuẩn truyền hình phân giải cao (HDTV) (50-100 khung hình/s). Quá trình tăng tốc độ khung hình (frame rate up-conversion: FRUC) được định nghĩa là một phương pháp hậu xử lý nhằm bù đắp mất mát về độ phân giải thời gian bằng cách tăng tốc độ khung hình.



Hình 1. Mô hình kỹ thuật MCFI

Phương pháp đơn giản nhất để tăng tốc độ khung hình là lặp khung hình nhiều lần hay lấy trung bình giữa khung hình trước (F_{t-1}) và khung hình hiện tại (F_t) để tạo khung hình cần nội

suy (F_{in}). Tuy nhiên cả 2 phương pháp này chỉ nhằm thỏa mãn việc tăng số lượng khung hình, chất lượng của video sau khi tăng khung hình thường bị giật hay có hiện tượng bóng ma, nhất là với các đối tượng có chuyển động nhanh. Kỹ thuật thông dụng và hiệu quả hơn là nội suy khung hình dùng kỹ thuật bù chuyển động (motion compensated frame interpolation: MCFI). Kỹ thuật này xem xét đến sự chuyển động của đối tượng qua các khung hình và cho phép đạt chất lượng khung hình nội suy cao cho cả chuyển động nhanh. Trong Hình 1, giả sử đối tượng B_1 ở khung hình F_{t-1} di chuyển thành đối tượng B_3 ở khung hình F_t , thì đối tượng B_2 ở khung hình cần nội suy F_{in} sẽ được tính toán dựa vào sự chuyển động từ B_1 sang B_3 và giá trị của 2 đối tượng này. Kết quả là khung hình F_{in} được chèn thêm làm tăng gấp đôi tốc độ khung hình.

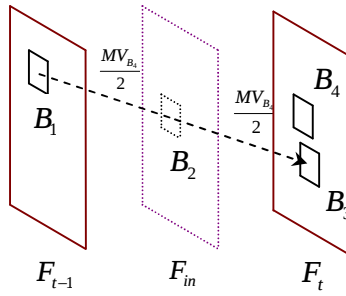
Kỹ thuật MCFI đòi hỏi phải tìm được chuyển động chính xác của đối tượng và công việc này chiếm phần lớn thời gian thực hiện giải thuật. MCFI cũng phải giải quyết các trường hợp không tìm được đối tượng tương ứng của đối tượng B_2 ở khung hình F_{t-1} và F_t (lỗ trống) hay đối tượng B_2 có nhiều đối tượng tương ứng ở khung hình F_{t-1} và F_t (chồng lấn). Ngoài ra MCFI còn gặp phải vấn đề méo khối khi các vectơ của các khối kề cận có độ khác biệt lớn. Bài báo này xem xét 2 giải thuật nâng cao chất lượng MCFI là chỉnh sửa và tăng độ phân giải vectơ nhằm tìm chuyển động thực của đối tượng và giảm méo khối. Bài báo được định dạng như sau. Phần II sẽ trình bày các giải thuật và đặc tính các phương pháp MCFI. Phần III sẽ chi tiết hóa 2 giải thuật được mô phỏng và các phương pháp nhằm nâng cao chất lượng của khung hình nội suy. Kết quả mô phỏng và nhận xét so sánh hiệu quả 2 phương pháp nâng cao chất lượng khung hình nội suy được trình bày ở Phần IV. Cuối cùng Phần V trình bày kết luận và hướng phát triển của đề tài.

2.CÁC PHƯƠNG PHÁP MCFI

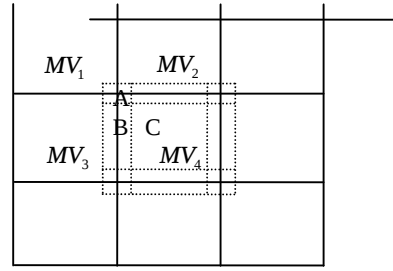
Đối với các ứng dụng nén video, các khung hình thường được chia ra thành các khối với kích thước nhỏ hơn và thông tin chuyển động của từng khối được thể hiện bằng các vectơ chuyển động (motion vector: MV). Kỹ thuật MCFI được chia làm 2 loại tùy vào việc sử dụng vectơ này: kỹ thuật MCFI dùng vectơ chuyển động có sẵn ở bộ giải mã hay kỹ thuật MCFI phải tìm vectơ chuyển động.

2.1.MCFI dùng vectơ chuyển động có sẵn

Giải thuật này được tích hợp vào bộ giải mã và sử dụng các vectơ đã được giải mã hóa trong khối dữ liệu nén. Tuy nhiên các vectơ chuyển động này được tính toán nhằm tối ưu hóa hiệu suất nén dữ liệu tại khối mã hóa, nên đôi khi không phản ánh chuyển động thực tế của đối tượng trong khung hình. Đối tượng B_2 cần nội suy ở khung F_{in} có thể được chọn là khối hay pixel.



Hình 2. Kỹ thuật MCFI cho khối



Hình 3. Méo khối

2.2.MCFI khối

Phương pháp này được nghiên cứu bởi Ha [1], Ai-mei [2][3], Yang [4], Lee [5]. Hình 2 miêu tả giải thuật này. Đối tượng B_2 có kích thước bằng đúng kích thước khối ở bộ mã hóa (thường là 8x8 pixel) và được nội suy theo công thức:

$$F_{in}(B_2) = \frac{1}{2}(F_{t-1}(B_1) + F_t(B_3)) = \frac{1}{2}(F_{t-1}(B_2 - \frac{MV_{B_1}}{2}) + F_t(B_2 + \frac{MV_{B_2}}{2})) \quad (1)$$

trong đó MV_{B_1} là vectơ chuyển động của khối B_1 ở khung hình F_{t-1} có cùng vị trí với B_2 . Giải thuật này hiệu quả cho các video chuyển động dịch chuyển tuyến tính. Đối với các video có chuyển động nhanh $MV_{B_1} \neq MV_{B_2}$ thì giải thuật không cho kết quả chính xác. Ở các khối liền kề nhau có vectơ chuyển động khác nhau thường xảy ra hiện tượng méo khối do sự không liên tục của đối tượng tại đường biên các khối này. Méo khối được làm giảm bằng cách sử dụng bộ lọc trung vị nhằm tính toán vectơ chuyển động thực của các pixel nằm ở đường biên:

$$MV_A = \text{median}(MV_1, MV_2, MV_3, MV_4) \quad (2)$$

$$\text{và } MV_B = (MV_3 + MV_4)/2. \quad (3)$$

Một phương pháp tương tự là bù chuyển động có chồng lấn khối:

$$F_{in}(A) = \frac{1}{8} \sum_{i=1}^4 (F_{n-1}(A - MV_i) + F_n(A + MV_i)), \quad (4)$$

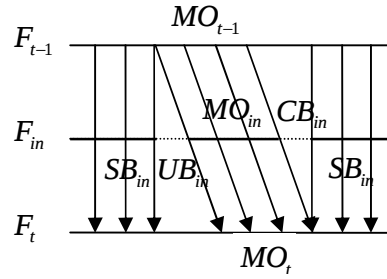
$$F_{in}(B) = \frac{1}{4} \sum_{i=3}^4 (F_{n-1}(A - MV_i) + F_n(A + MV_i)), \quad (5)$$

$$F_{in}(C) = \frac{1}{2} (F_{n-1}(A - MV_4) + F_n(A + MV_4)). \quad (6)$$

Cả 2 phương pháp đều cho kết quả tốt đối với các đối tượng chuyển động không quá khác biệt.

2.3.MCFI cho pixel

Phương pháp này được đề cập bởi Kuo [6], Ishwar [7]. Các vectơ chuyển động được sử dụng nhằm tìm quỹ đạo chuyển động của pixel qua các khung hình như hình 4.



Hình 4. MCFI cho pixel

$$+ \text{Pixel chuyển động: } MO_{in} = \frac{1}{2}(F_{t-1} + F_t) \cdot (7)$$

$$+ \text{Pixel nền không có vectơ đi qua: } UB_{in} = F_t. \quad (8)$$

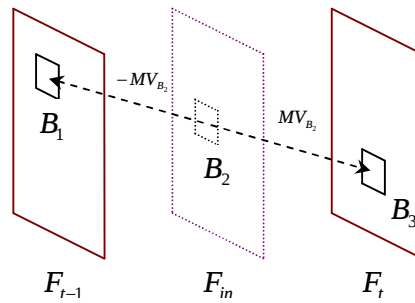
$$+ \text{Pixel nền có vectơ đi qua: } CB_{in} = F_{t-1}. \quad (9)$$

$$+ \text{Pixel nền tĩnh: } MO_{in} = \frac{1}{2}(F_{t-1} + F_t). \quad (10)$$

Phương pháp này không có hiện tượng méo khối nhưng trong một số trường hợp gặp phải vấn đề lỗ trống và chồng lấn. Lỗ trống xảy ra khi một pixel không có vectơ nào đi qua và chồng lấn xảy ra khi một pixel có nhiều vectơ đi qua. Trường hợp lỗ trống có thể được giải quyết bằng cách sử dụng giá trị hoặc vectơ của các pixel xung quanh nhằm nội suy pixel lỗ trống. Trường hợp chồng lấn được giải quyết bằng cách lấy trung bình các giá trị nội suy của pixel chồng lấn.

2.4.MCFI ước lượng vectơ chuyển động

Việc ước lượng vectơ chuyển động mặc dù tốn nhiều thời gian, nhưng cho phép xây dựng giải thuật tách biệt với khối giải mã và có thể tích hợp vào bộ tiền xử lý của màn hình. Thông tin chuyển động từ nhiều khung hình xung quanh có thể được sử dụng thay vì chỉ có thông tin chuyển động theo một chiều như các phương pháp MCFI dùng vectơ có sẵn ở bộ giải mã. Các phương pháp thông dụng là ước lượng vectơ 2 hướng (Choi [8], [9], [10]), dùng mô hình chóp (Jeon [11], Chalidbhongse [12], Thoma [13]) hay sử dụng nguyên lý dòng chảy quang học (Krishnamurthy [14]).



Hình 5. Ước lượng vectơ 2 hướng

2.5. Phương pháp ước lượng vectơ 2 hướng

Vectơ cho mỗi khối của khung hình cần nội suy được ước lượng theo điều kiện 2 hướng

$$F_{in}(B_2) = \frac{1}{2}(F_{t-1}(B_2 - MV_{B_2}) + F_t(B_2 + MV_{B_2})) \quad (11)$$

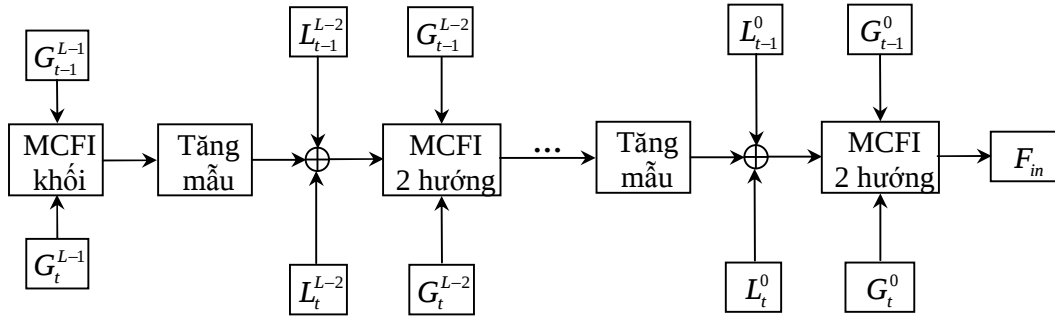
trong đó MV_{B_2} là vectơ cho sai số nhỏ nhất giữa 2 khối tương ứng ở khung hình F_{t-1} và F_t

$$MV_{B_2} = \arg \min_{s \in B_2} \left(\sum_{s \in B_2} |F_{t-1}[s - MV_{B_2}] - F_t[s + MV_{B_2}]| \right). \quad (12)$$

với s là một pixel trong khối B_2 . Phương pháp rất hiệu quả cho các chuyển động dịch chuyển và không gặp phải vấn đề lỗ trống hay chồng lấp, tuy nhiên do là phương pháp MCFI cho khối, nên khung hình được nội suy bị méo khối.

2.6. Phương pháp mô hình chóp

Nguyên lý là dùng phương pháp hình chóp Gaussian và Laplacian cho khung hình ban đầu (G^0) để tìm các bản sao qua bộ lọc thông thấp (bản sao Gaussian G^1) và phần thặng dư (bản sao Laplacian L^1), các bản sao này được giảm mẫu phân nửa và quá trình được lặp lại nhiều lần để tìm các bản sao Gaussian và Laplacian cho các tầng cao hơn [15]. Đối với phương pháp MCFI hình chóp, quá trình nội suy đi theo chiều ngược lại từ bản sao Gaussian và Laplacian tầng cao xuống dần đến các bản sao tầng thấp. Hình 6. minh họa cho quá trình MCFI này.



Hình 6. MCFI mô hình chóp

Ở tầng cao nhất, bản nội suy G_{in}^{L-1} được ước lượng bằng MCFI cho khối như ở phần II.1.1

$$G_{in}^{L-1}(B_2) = \frac{1}{2}(G_{t-1}^{L-1}(B_2 - \frac{MV_{B_4}}{2}) + G_t^{L-1}(B_2 + \frac{MV_{B_4}}{2})). \quad (13)$$

Bản nội suy này được tăng mẫu và cộng với bản sao Laplacian của cấp $L-2$ để có bản nội suy tham chiếu $\hat{G}_{in}^{L-2}$ với nhiều chi tiết hơn

$$\hat{G}_{in}^{L-2}(2B_2) = \text{expand}(G_{in}^{L-1}(B_2)) + \frac{1}{2}(L_{t-1}^{L-2}(2B_2 - MV_{B_4}) + L_t^{L-2}(2B_2 + MV_{B_4})). \quad (14)$$

Bản nội suy tham chiếu này được dùng để ước lượng 2 vectơ chuyển động cho cả 2 chiều $MV_{2B_2,t-1}$ và $MV_{2B_2,t}$ từ bản sao Gaussian G_{t-1}^{L-2} và G_t^{L-2} cho mỗi khối $2B_2$. MCFI 2 chiều được áp dụng để có được bản nội suy cấp $L-2$

$$G_{in}^{L-2}(2B_2) = \frac{1}{2}(G_{t-1}^{L-2}(2B_2 - MV_{2B_2,t-1}) + G_t^{L-2}(2B_2 + MV_{2B_2,t})). \quad (15)$$

Quá trình được lặp lại cho đến tầng thấp nhất để đạt được khung nội suy F_{in} .

Phương pháp này hiệu quả cho cả các chuyển động không tuyến tính. Vì là MCFI 2 chiều có bản tham chiếu nên phương pháp này cho phép ước lượng 2 véc tơ khác nhau cho 2 chiều, do vậy khung nội suy có chất lượng cao hơn so với phương pháp MCFI 2 chiều không có bản tham chiếu như được miêu tả ở phần II.2.1. Tuy nhiên phương pháp này khá phức tạp với nhiều cấp xử lý, gặp phải vấn đề méo khối và phụ thuộc nhiều vào chất lượng của bước nội suy đầu tiên.

3.CHỈNH SỬA VÀ TĂNG ĐỘ PHÂN GIẢI VÉCTƠ CHUYỂN ĐỘNG

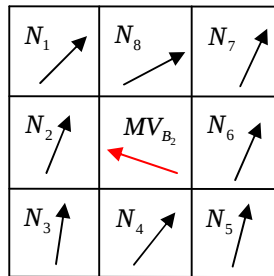
Cả phương pháp MCFI 2 chiều và MCFI hình chóp đều yêu cầu phải ước lượng véc tơ chuyển động. Tuy nhiên các phương pháp ước lượng véc tơ chuyển động lại nhằm tìm kiếm các véc tơ tối thiểu hóa sai số của tính hiệu sai số nhằm đạt mức nén cao nhất, các véc tơ này đôi khi không thể hiện được chuyển động thực của đối tượng. Một số phương pháp nhằm làm tăng tính chính xác của véc tơ chuyển động là chỉnh sửa và tăng độ phân giải của véc tơ chuyển động.

3.1.Chỉnh sửa véc tơ

Phương pháp này sửa đổi các véc tơ không đáng tin cậy bằng cách sử dụng các véc tơ chuyển động xung quanh [16]. Mức tin cậy được tính dựa trên độ tròn của véc tơ chuyển động so với các véc tơ xung quanh

$$S(MV_{B_2}) = \sum_{i=1}^8 (MV_{B_2} - N_i)^2 \quad (16)$$

Véc tơ được xem là không tin cậy khi $S(MV_{B_2}) \geq T$ với T là ngưỡng có giá trị được xác định trước. Hình 7. miêu tả một ví dụ của một véc tơ không tin cậy MV_{B_2} . Các véc tơ không tin cậy sẽ được thay thế bằng véc tơ có độ tròn lớn nhất so với các véc tơ xung quanh.



Hình 7. Véc tơ không tin cậy

$$MV_{B_2}' = \arg \min_{MV_{B_2}} S(MV_{B_2}') = \arg \min_{MV_{B_2}} \sum_{i=1}^8 (MV_{B_2}' - N_i)^2. \quad (17)$$

Ở bài báo này, công thức tính độ tròn ở (16) được thay thế bằng

$$S(MV_{B_2}) = \sum_{i=1}^8 w_i (MV_{B_2} - N_i)^2 \quad (18) \text{ trong đó } w_i = \begin{cases} 1, & \text{nếu } MV_{B_2} \text{ không tin cậy,} \\ 0, & \text{nếu } MV_{B_2} \text{ tin cậy.} \end{cases}$$

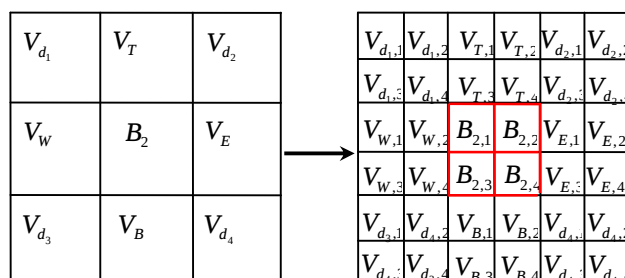
3.2. Tăng độ phân giải vectơ chuyển động

Phương pháp này cho phép nội suy các vectơ cho các khối nhỏ hơn từ vectơ của các vectơ lớn hơn và cho phép giảm độ méo khối [8]. Giả sử giá trị các pixel trong khối không thay đổi đột ngột, các vectơ cho các khối 4*4 pixel $B_{2,i}$ trong Hình 8 được ước lượng nhằm đạt độ trơn tối đa với các vectơ chuyển động xung quanh theo mọi hướng

$$y = y_{North} + y_{South} + y_{East} + y_{West} + y_{Diagonal} + y_{Center} \quad (19)$$

$$\text{trong đó } y_{North} = (v_1 - v_T)^2 + (v_2 - v_T)^2 + (v_3 - v_1)^2 + (v_4 - v_2)^2, \quad (20)$$

$$y_{Diagonal} = (v_1 - v_{d_1})^2 + (v_2 - v_{d_2})^2 + (v_3 - v_{d_3})^2 + (v_4 - v_{d_4})^2. \quad (21)$$



Hình 8. Tăng độ phân giải vectơ chuyển động

4. KẾT QUẢ MÔ PHỎNG VÀ NHẬN XÉT

4.1. MCFI 2 hướng



Hình 9. Khung hình trước



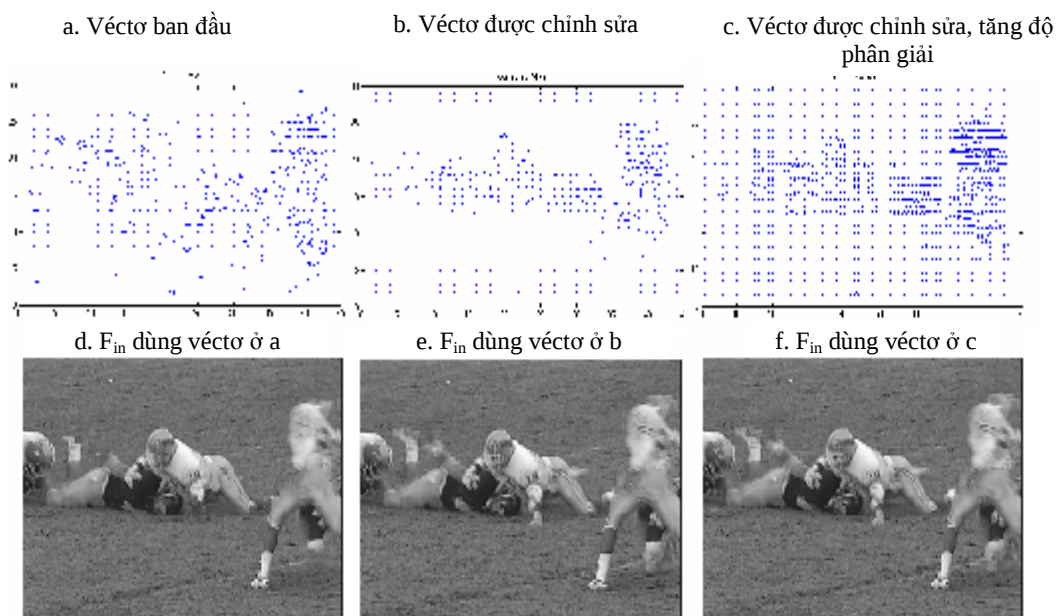
Hình 10. Khung hình hiện tại



Hình 11. Khung hình nội suy không sử dụng giải thuật chỉnh sửa và tăng độ phân giải vectơ



Hình 12. Khung hình nội suy sử dụng giải thuật chỉnh sửa và tăng độ phân giải vectơ

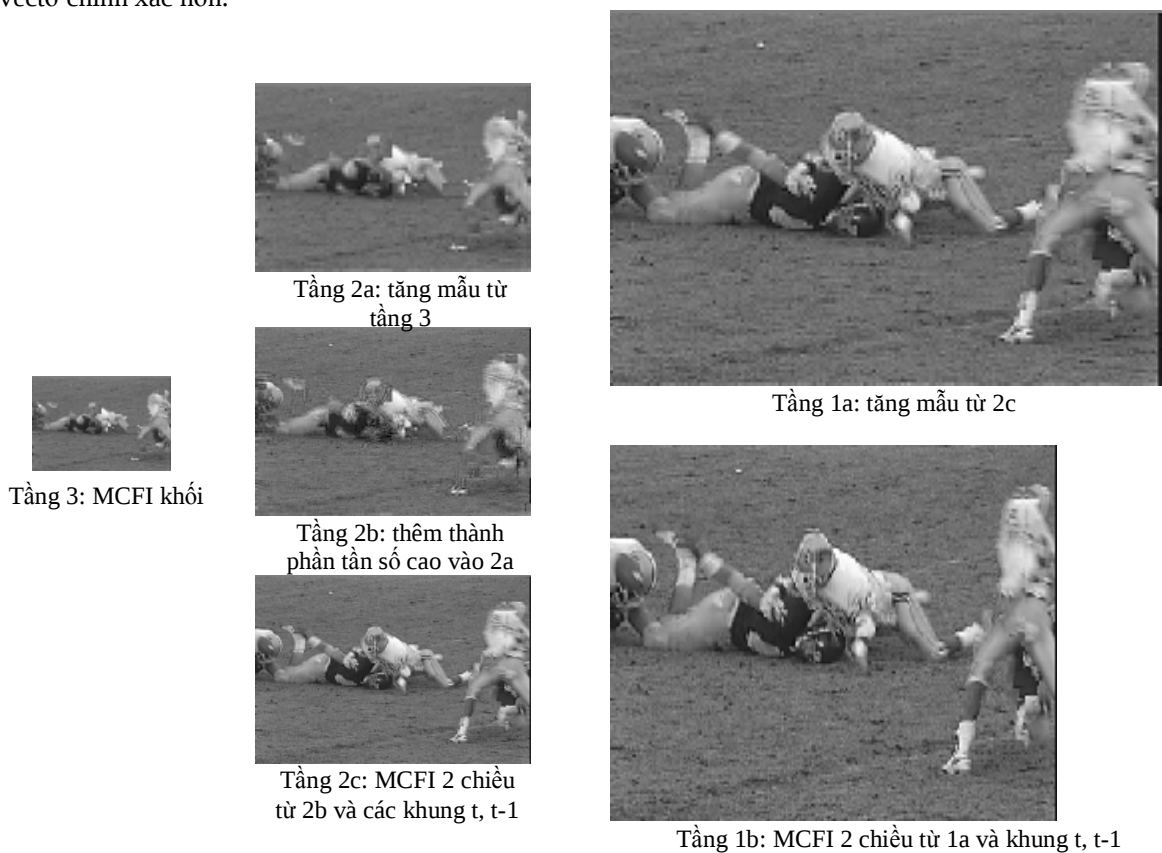


Hình 13. Vectơ và khung hình nội suy minh họa cho giải thuật chỉnh sửa và tăng độ phân giải vectơ

Chương trình mô phỏng được thực hiện trên video với tốc độ 30 khung hình/s. Kỹ thuật MCPI 2 hướng được áp dụng để tăng tốc độ video lên 60 khung hình/s thích hợp cho chuẩn truyền hình phân giải cao. Ngưỡng $T = 50$ để xác định vectơ tin cậy. Kết quả mô phỏng cho thấy kỹ thuật MCPI 2 chiều cho khung hình nội suy chất lượng khá thấp (Hình 11) và có méo biên khối. Khi sử dụng kỹ thuật chỉnh sửa và tăng độ phân giải vectơ thì chất lượng khung hình nội suy được tăng lên rõ rệt (Hình 12) với ít méo biên khối, nhất là ở đối tượng chuyển động nhanh như các cầu thủ bóng bầu dục. Hiệu quả của kỹ thuật chỉnh sửa và tăng độ phân giải vectơ được minh họa riêng biệt ở hình 13. Kỹ thuật chỉnh sửa vectơ thay thế vectơ không tin cậy bằng vectơ có độ trơn lớn nhất với các vectơ xung quanh. Kỹ thuật tăng độ phân giải vectơ cho phép ước lượng vectơ cho các khối 4×4 từ các vectơ khối 8×8 , tăng độ mịn chuyển động và giảm méo biên khối.

4.2.MCFI hình chớp

Các thông số mô phỏng được giữ nguyên như ở phần IV.1. MCFI được thực hiện cho 3 tầng: ở tầng 3 áp dụng MCFI khối, ở các tầng thấp hơn áp dụng MCFI 2 chiều có khung tham chiếu. Ở tầng 2, khung hình tăng mẫu từ tầng trước được cộng với bản Laplacian để có thêm thông tin tần số cao. Khung hình ở từng bước được minh họa ở Hình 14 và kết quả cuối cùng được minh họa ở Hình 17 cho MCFI hình chớp và Hình 18 cho MCFI có sử dụng chỉnh sửa và tăng độ phân giải vectơ. Kết quả cho thấy khi sử dụng kỹ thuật chỉnh sửa và tăng độ phân giải vectơ, chất lượng khung hình nội suy cao hơn và không bị nhòe ở biên các đối tượng dao động. Kết quả này cũng cho thấy hiệu quả phương pháp MCFI hình chớp so với MCFI khối, do MCFI hình chớp ước lượng khung hình tham chiếu từ các tầng trên, cho phép đạt được các vectơ chính xác hơn.



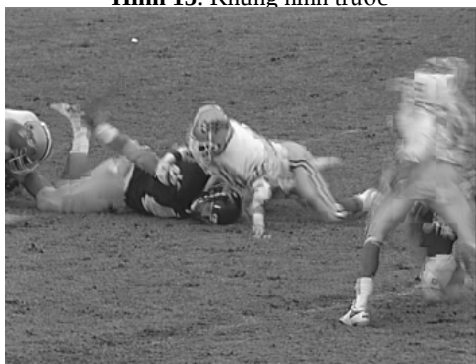
Hình 14. MCFI hình chớp



Hình 15. Khung hình trước



Hình 16. Khung hình hiện tại



Hình 17. Khung hình nội suy dùng MCFI hình chớp



Hình 18. MCFI hình chớp có sử dụng chỉnh sửa và tăng độ phân giải vectơ

5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN BÀI BÁO

Bài báo ứng dụng kỹ thuật chỉnh sửa và tăng độ phân giải vectơ cho phương pháp tăng khung hình. Kết quả mô phỏng cho thấy hiệu quả của việc chỉnh sửa vectơ trong việc loại trừ các vectơ không tin cậy và hiệu quả việc tăng độ phân giải vectơ trong việc giảm méo biên khối và tăng độ mịn chuyển động. Mô phỏng cũng cho thấy MCFI hình chớp cho chất lượng khung hình nội suy tốt hơn MCFI 2 chiều, tuy nhiên MCFI hình chớp có độ phức tạp cao hơn. Đề tài sẽ được phát triển tiếp theo hướng kết hợp với các phương pháp tăng độ phân giải khung hình nhằm giải quyết hoàn thiện bài toán tăng chất lượng khung hình về cả thời gian và không gian. Ngoài ra các phương pháp tìm kiếm vectơ nhanh cũng cần được nghiên cứu nhằm giảm thời gian thực thi. Một hướng phát triển khác là nội suy khung hình từ nhiều khung hình, thay vì chỉ 2 khung hình như ở phương pháp MCFI 2 hướng.

MOTION VECTOR CORRECTION AND REFINEMENT EFFECT IN FRAME UP-RATE CONVERSION

Hoang Dinh Chien

University of Technology, VNU-HCM

ABSTRACT: *The paper applies the motion vector (MV) correction and refinement algorithm to frame up-rate conversion (FRUC) in order to enhance the quality of the interpolated frame. The motion compensated frame interpolation (MCFI) methods using bilateral motion estimation and hierarchical block matching algorithm are used to form the interpolated frame. Simulation results show that MV correction algorithm provides true motion vectors and MV refinement helps reducing the blocking artifact.*

Key words: *frame up-rate conversion (FRUC), motion compensated frame interpolation (MCFI), motion vector correction, motion vector refinement.*

TÀI LIỆU THAM KHẢO

- [1]. Taehyeun Ha, Seongjoo Lee and Jaeseok Kim, *Motion Compensated Frame Interpolation by new Block-based Motion Estimation Algorithm*, IEEE Transactions On Consumer Electronics, Vol. 50, No. 2, May (2004).
- [2]. Ai-Mei Huang and Truong Nguyen, *Motion Vector Processing Based On Residual Energy Information For Motion Compensated Frame Interpolation*, IEEE Conference on Image Processing, (2006).
- [3]. Ai-Mei Huang and Truong Nguyen, *A Novel Motion Compensated Frame Interpolation Based on Block-Merging and Residual Energy*, MMSP (2006).
- [4]. Ya-Ting Yang, Yi-Shin Tung, and Ja-Ling Wu¹, *Quality Enhancement of Frame Rate Up-Converted Video by Adaptive Frame Skip and Reliable Motion Extraction*, IEEE Transactions on Circuits and Systems on Video Technology, (2007).
- [5]. Sung-Hee Lee, Ohjae Kwon, and Rae-Hong Park, *Weighted-Adaptive Motion-Compensated Frame Rate Up-Conversion*, IEEE Transactions on Consumer Electronics, August (2003).
- [6]. Tien-ying Kuo and C.-C. Jay Kuo, *Motion- Compensated Interpolation for Low- Bit-Rate Video Quality Enhancement*, SPIE Conference on Applications of Digital Image Processing XXI, San Diego, California, July (1998).
- [7]. Prakash Ishwar and Pierre Moulin, *On Spatial Adaptation of Motion-Field Smoothness in Video Coding*, IEEE Transactions On Circuits And Systems For Video Technology, Vol. 10, No. 6, September (2000).
- [8]. Byeong-Doo Choi, Jong-Woo Han, Chang-Su Kim, and Sung-Jea Ko, *Motion-Compensated Frame Interpolation Using Bilateral Motion Estimation and Adaptive Overlapped Block Motion Compensation*, IEEE Transactions On Circuits And Systems For Video Technology, Vol. 17, No. 4, April (2007).
- [9]. Byung-Tae Choi, Sung-Hee Lee and Sung-Jea KO, *New Frame Rate Up-conversion Using Bi-directional Motion Estimation*, IEEE Transactions on Consumer Electronics, Vol. 46, No. 3, August (2000).

- [10]. Byeong-Doo Choi, Jong-Woo Han, Chang-Su Kim and Sung-Jea Ko, *Frame Rate Up-Conversion Using Perspective Transform*, IEEE Transactions on Consumer Electronics, Vol. 52, No. 3, August (2006).
- [11]. Bo-Won Jeon, Gun-Il Lee, Sung-Hee Lee and Rae-Hong Park, *Coarse-to-Fine Frame Interpolation for Frame Rate Up-Conversion Using Pyramid Structure*, IEEE Transactions on Consumer Electronics, Vol. 49, No. 3, August (2003).
- [12]. J. Chalidbhongse and C.-C. J. Kuo, *Fast Motion Estimation using Multi-resolution-spatial-Temporal Correlations*, IEEE Trans. Circuits Syst. Video Technol., vol. 7, no. 3, pp. 477-488, June (1997).
- [13]. R. Thoma and M. Bierling, *Motion Compensating Interpolation Considering Covered and Uncovered Background*, Signal Processing: Image Communications, vol. 1, no. 2, pp. 191-212, Oct. (1989).
- [14]. Ravi Krishnamurthy, John W. Woods, and Pierre Moulin, *Frame Interpolation and Bidirectional Prediction of Video Using Compactly Encoded Optical-Flow Fields and Label Fields*, IEEE Transactions On Circuits And Systems For Video Technology, Vol. 9, No. 5, August (1999).
- [15]. P. J. Burt and E. H. Adelson, *The Laplacian Pyramid as a Compact Image Code*, IEEE Trans. Communications, vol. 31, no. 4, pp. 532-540, Apr. (1983).
- [16]. Jiefu Zhai, Keman Yu, Jiang Li, and Shipeng Li, *A Low Complexity Motion Compensated Frame Interpolation Method*, IEEE International Symposium on Circuits and Systems, (2005).