

THUẬT TOÁN CHO PHÉP NÂNG CAO KHẢ NĂNG PHÂN ĐOẠN DỮ LIỆU VIDEO

Lương Xuân Cương⁽¹⁾, Đỗ Trung Tuấn⁽²⁾

⁽¹⁾Học viện Kỹ thuật Quân sự

⁽²⁾Trường Đại học Khoa học Tự nhiên, Đại học Quốc gia Hà Nội

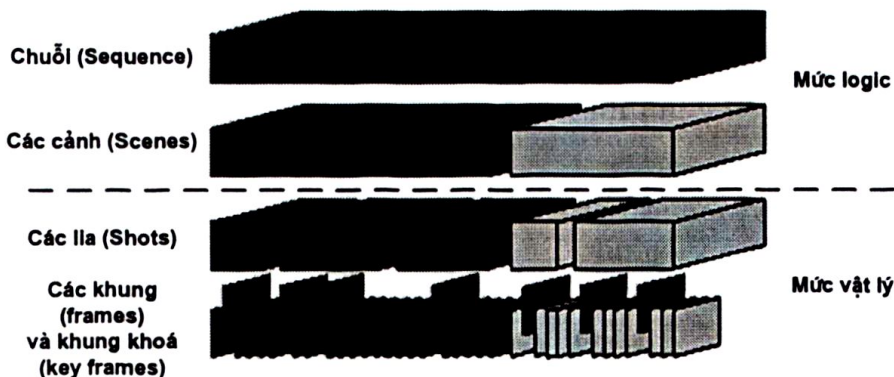
1. Giới thiệu

Hiện nay, bài toán xử lý dữ liệu video đang là vấn đề thời sự và đặt ra nhiều thách thức [3, 7]. Mặc dù được kế thừa khá nhiều kết quả của bài toán xử lý ảnh tĩnh, các kỹ thuật xử lý dữ liệu video còn gặp phải rất nhiều khó khăn. Những nguyên nhân chính là do: Dữ liệu video ngoài đặc tính không gian còn có thêm đặc tính thời gian, có nội dung ngữ nghĩa trừu tượng, kích thước lớn, tốc độ dòng dữ liệu cao...

Một trong những vấn đề quan trọng nhất của bài toán xử lý dữ liệu video là việc lấy mẫu theo thời gian trên cơ sở nội dung của dữ liệu video [1, 5]. Sau đó thông qua việc trích chọn các đặc trưng nhằm xác định những khung video (frame) có ý nghĩa cho những nhiệm vụ khác nhau.

Trong thực tế, việc phân chia dòng video thành các đơn vị cơ sở gọi là các lần lia (lia, shot) được cho là giải pháp hiệu quả nhất. Một lia có thể hiểu là một đơn vị vật lý của dòng video, gồm một chuỗi liên tiếp các khung ứng với một thao tác camera đơn. Việc lấy mẫu có nhiệm vụ chọn gần đúng một khung video đại diện cho mỗi lia (hoặc nhiều hơn một khung tùy theo mức độ phức tạp của nội dung hình ảnh của lia) và các khung đó được gọi là các *khung khóa* (key frame).

Thí dụ một kiểu cấu trúc phân cấp của dòng video như trên hình 1. Trong đó các đơn vị khung và lia là các đơn vị vật lý. Các khái niệm cảnh (scene) và chuỗi (sequence) là các đơn vị logic của dòng video. Một cảnh gồm các lia liên quan về không gian và liên kế về thời gian, cùng mô tả một nội dung ngữ nghĩa hoặc một tình tiết. Trong khi một số các cảnh cùng đưa đến một ý nghĩa, gọi là một chuỗi [1, 3].



Hình 1. Một kiểu cấu trúc phân cấp dòng video

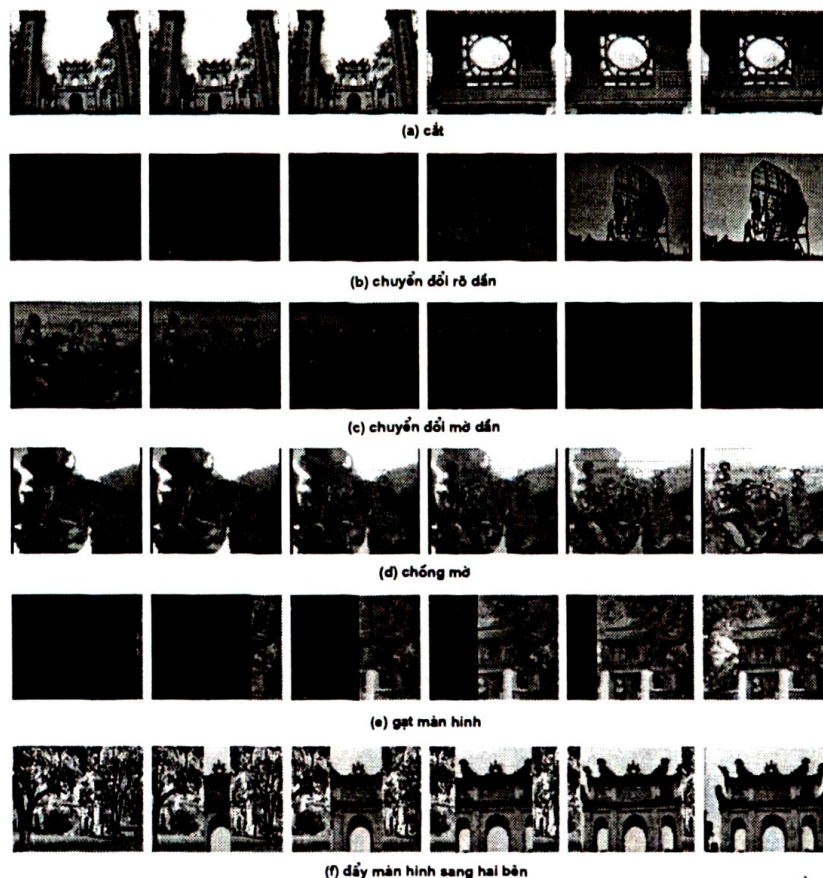
Quá trình phân đoạn dữ liệu video tiến hành phân tích, phát hiện việc chuyển từ lia này sang lia khác, còn gọi là phát hiện ranh giới lia. Và để thực hiện quá trình này một cách tự động luôn là một công việc đầy khó khăn.

Bài báo sẽ đề xuất một kỹ thuật nhằm nâng cao khả năng phân đoạn tự động dữ liệu video trong khi thể hiện lại các file video định dạng AVI. Nội dung các phần tiếp theo: Phần 2 tóm tắt một số thuật toán phân đoạn dữ liệu video dựa trên đặc tính hình ảnh mức thấp. Phần 3 đề xuất thuật toán nâng cao khả năng đã được ứng dụng để xây dựng phần mềm tự động phân đoạn dữ liệu video *PhanDoanVideo06*. Phần 4 đưa ra một số kết quả thực nghiệm trên phần mềm này.

2. Một số kỹ thuật phân đoạn dữ liệu video dựa trên đặc tính hình ảnh mức thấp

2.1. Vấn đề phân đoạn dữ liệu video

Sự chuyển đổi giữa các lia có thể diễn ra đột ngột hoặc từ từ. Thí dụ đơn giản nhất của chuyển đổi đột là một ngắt camera [8]. Trên hình 2a là một ngắt camera. Ngược lại, chuyển đổi từ từ là kết quả của các hiệu ứng camera và/hoặc hiệu ứng biên tập video đặc biệt. Hình 2b - 2f minh họa một số dạng chuyển đổi từ từ thường gặp. Nói chung, quá trình chuyển đổi từ từ thường xảy ra kéo dài hơn và để phát hiện ra nó khó khăn cũng như nhiều sai sót hơn [1].



Hình 2. Minh họa các chuyển đổi giữa các lia video

Hiện đã có một số thuật toán được phát triển để tự động phát hiện ranh giới lia. Bên cạnh các thuật toán so sánh đặc tính hình ảnh mức thấp của video, còn có một số thuật toán so sánh nội dung mức cao (nội dung ngữ nghĩa) của video [1, 5].

Đa số các phương pháp phân đoạn dữ liệu video hiện nay dựa trên phân tích nội dung video mức thấp trên cơ sở so sánh toàn khung. Sự khác nhau giữa chúng là ở: (i) các phép đo được thực hiện như thế nào; (ii) các đặc tính nào được chọn; và (iii) việc chia nhỏ các khung hình như thế nào [1, 3]. Nhiều phương pháp sử dụng đặc tính hình ảnh hay các thông tin về chuyển động trong video để tính toán sự khác nhau giữa các khung. Cách tiếp cận dựa trên các đặc tính hình ảnh có nhược điểm khi gặp phải vấn đề chuyển động của camera và đối tượng. Thuật toán dựa trên chuyển động sẽ phải tính toán rất nhiều bởi vì số lượng các phép so sánh trên các khối ảnh trong các khung là quá lớn. Sau khi đã tính toán được các kết quả về sự sai khác giữa các khung, cần phải sử dụng các giá trị ngưỡng để có thể quyết định về ranh giới lia.

Trong thực tế, kỹ thuật phân đoạn dữ liệu video tự động luôn gặp phải nhiều khó khăn dẫn tới việc phát hiện sai hoặc bỏ sót. Trước hết đề cập đến tính tương quan cao của dòng video nối tiếp. Các khung video liên kế thì gần như giống nhau. Ngoài ra còn có rất nhiều nhiễu có thể gây sai số như sự biến đổi độ sáng đột ngột (chớp đèn flash của camera), các hiệu ứng biên tập video và camera, sự di chuyển của đối tượng và camera... Một yếu tố rất quan trọng khác quyết định nhiều đến kết quả phân đoạn đó là việc lựa chọn giá trị ngưỡng thích hợp trong các phép so sánh [2]. Dưới đây chỉ điểm lại một số thuật toán liên quan đến đặc tính hình ảnh mức thấp của video:

- *So sánh các cặp điểm ảnh (pixel) tương ứng của hai khung liên kế*: Đây là thuật toán đơn giản nhất [1, 2]. Nó so sánh các điểm ảnh tương ứng trong hai khung, xác định có bao nhiêu điểm ảnh đã thay đổi. Thí dụ với ảnh đơn sắc, một điểm ảnh được coi là đã bị thay đổi nếu sự khác nhau giá trị cường độ sáng của nó trong hai khung vượt quá một giá trị ngưỡng.
- *So sánh biểu đồ*: Sử dụng một số biểu đồ biểu diễn các đặc tính hình ảnh mức thấp của video để so sánh các điểm ảnh hoặc vùng tương ứng trong các khung liên kế [1, 2]. Thực tế đã có các phương pháp sử dụng biểu đồ mức xám hoặc sử dụng biểu đồ màu...
- *So sánh dựa trên phát hiện cạnh trong một khung video*: Phương pháp này sử dụng các thuật toán xử lý đường biên và phát hiện cạnh trong một khung hình để so sánh giữa các khung liên kế trong một chuỗi video [4].

2.2. Hướng phát triển của các thuật toán phân đoạn dữ liệu video

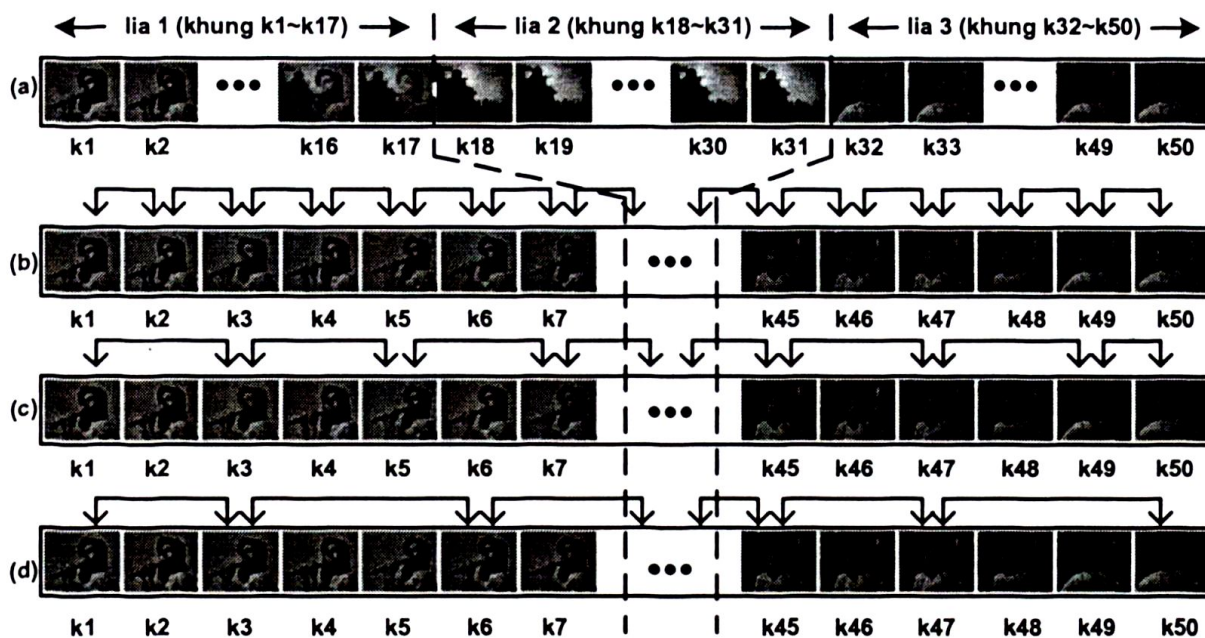
Nói chung, các hướng nghiên cứu về phân đoạn dữ liệu video luôn nhằm vào nâng cao khả năng tự động hóa và tăng độ chính xác của các thuật toán. Các nghiên cứu có thể sẽ tập trung vào những vấn đề sau đây:

- Sử dụng thêm các thông tin khác trong dữ liệu video như âm thanh, văn bản... thay vì chỉ dựa trên một số đặc tính (màu, cường độ sáng, cạnh của đối tượng...) để nâng cao khả năng phân đoạn video [6].
- Xây dựng các thuật toán thích nghi có thể tự động điều chỉnh tham số sao cho phù hợp với các dạng dữ liệu video khác nhau và đưa ra các giá trị tham số tối ưu.
- Kết hợp một số kỹ thuật phân đoạn khác nhau để nâng cao hiệu quả.
- Một hướng đáng chú ý khác là đưa thêm thông tin vào dữ liệu video trong quá trình nén. Các thông tin này sẽ giúp ích cho quá trình phân đoạn dữ liệu sau này.
- Phát triển các thuật toán dựa trên kỹ thuật nhận dạng, trên cơ sở ứng dụng mạng nơ-ron và thị giác máy tính.

3. Thuật toán nâng cao khả năng phân đoạn dữ liệu video

Các thuật toán nêu trên đều thực hiện so sánh ở mức khung giữa các khung liên kế và cho các kết quả chấp nhận được. Chúng quét toàn bộ dòng video và xử lý các khung theo trình tự tuyến tính. Do tính tương quan cao của các khung video trong cùng một lia, cho nên trong trường hợp này rất nhiều phép so sánh thực sự không cần thiết, làm cho các thuật toán phức tạp hơn và khả năng thực hiện gặp nhiều khó khăn, nhất là trong trường hợp kích thước video lớn. Xuất phát từ nhận xét đó, chúng tôi đề xuất phương pháp nhằm cải tiến các thuật toán phân đoạn dữ liệu video bằng cách áp dụng các kỹ thuật nhảy đều và nhảy thích ứng. Thực chất là nhờ đó, có thể bỏ qua việc so sánh một số khung khi chúng ở trong cùng một lia mà không sợ sai sót.

Chuỗi video mẫu dùng để phân tích như trên hình 3a. Giả sử chuỗi video này gồm 50 khung và có 3 lia. Ranh giới của lia 1 và lia 2 nằm giữa các khung k_{17} và k_{18} và chuyển lia này là một chồng mờ; ranh giới của lia 2 và lia 3 nằm giữa các khung k_{31} và k_{32} và chuyển lia này là một cắt lia. Các kỹ thuật phân đoạn dữ liệu video trước đây thực hiện các phép so sánh sự khác nhau giữa các khung liên kế, nếu chúng sử dụng các thuật toán tuyến tính, chúng sẽ quét lần lượt toàn bộ các khung như mô tả trên hình 3b.

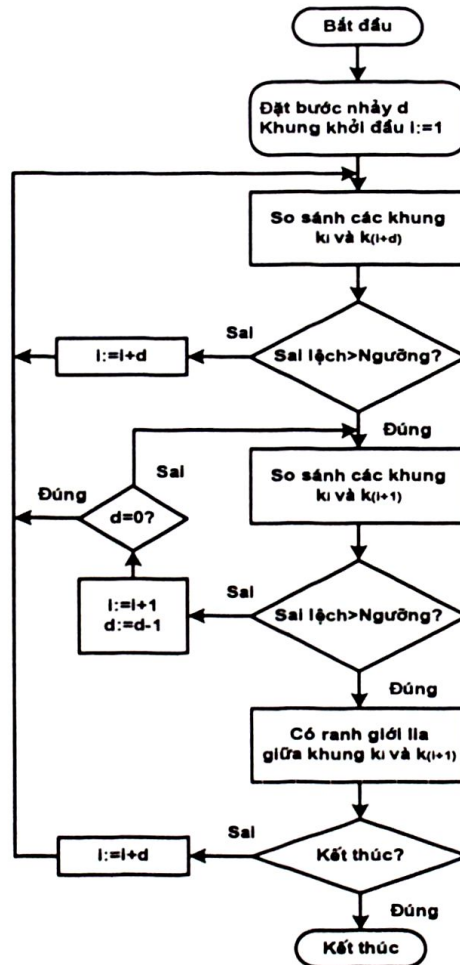


Hình 3. Minh họa phương pháp so sánh không tuyến tính

3.1. Kỹ thuật nhảy đều

Không như các thuật toán tuyến tính, kỹ thuật nhảy đều không thực hiện các phép so sánh giữa các khung liên kế mà thực hiện so sánh cách đều một số khung. Mỗi phép so sánh sẽ được thực hiện cách đều d khung (bước nhảy d). Thí dụ trường hợp $d=2$ (hình 3c). Các phép so sánh thực hiện giữa các khung sau: k_1 với k_{1+d} (k_3); k_3 với k_{3+d} (k_5); k_5 với k_{5+d} (k_7); ...

Quá trình so sánh cứ tiếp tục như vậy. Khi phát hiện ra hai khung k_i và k_{i+2} nằm trong hai lia khác nhau, sẽ tiến hành so sánh khung k_i với k_{i+1} để xác định ranh giới các lia. Nếu khung k_i và k_{i+1} nằm trong hai lia khác nhau, thì ranh giới lia nằm giữa hai khung k_i và k_{i+1} . Ngược lại, nếu khung k_i và k_{i+1} nằm trong cùng một lia, thì chắc chắn ranh giới lia nằm giữa hai khung k_{i+1} và k_{i+2} . Trường hợp mỗi lia có tối thiểu 2 khung sẽ không cần so sánh hai khung k_{i+1} với k_{i+2} . Khi đã xác định được ranh giới lia, trình tự được lặp lại cho lia tiếp theo. Lưu đồ thuật toán của kỹ thuật nhảy đều được thể hiện trên hình 4. Chú ý trong phép kiểm tra để kết thúc vẫn đặt lại số thứ tự khung là $i:=i+d$, vì sau vòng lặp so sánh giữa các khung k_i và k_{i+1} số đếm i tăng lên 1 đơn vị thì ngược lại số đếm d lại giảm 1 đơn vị.



Hình 4. Lưu đồ thuật toán nhảy đều

Kỹ thuật nhảy đều có thể cho phép giảm gần một nửa (với $d = 2$) số phép so sánh so với khi chỉ sử dụng các thuật toán tuyến tính. Có thể tổng quát hoá trong trường hợp bước nhảy d lớn hơn để cho kết quả tốt hơn. Số phép so sánh của quá trình phân đoạn luôn bị giới hạn bởi khả năng xử lý của máy tính. Với các máy tính có tốc độ không cao ý nghĩa của kỹ thuật này càng thể hiện rõ.

3.2. Kỹ thuật nhảy thích ứng

Giá trị bước nhảy d nói trên luôn phụ thuộc vào các dòng video. Trong kỹ thuật nhảy thích ứng (nhảy động), giá trị d được xác định động, biến đổi một cách thích ứng để đảm bảo là tối ưu. Mỗi lần lặp, thuật toán tính lại bước nhảy d cho thích hợp, bằng cách so sánh khung hiện tại với khung vừa được kiểm tra (hình 3d).

Nếu sự giống nhau của phép so sánh này nhiều hơn phép so sánh trước, ở phép so sánh kế tiếp, giá trị d sẽ được tăng lên và ngược lại. Mặt khác nếu phép so sánh hiện tại cho biết 2 khung nằm ở 2 lia khác nhau, sẽ quay lại thực hiện theo xu hướng lùi dần sử dụng các bước nhảy đều.

Khi phát hiện chính xác ranh giới lia, sẽ trở lại tiến trình nhảy động thuận bình thường. Kỹ thuật nhảy động sử dụng bước nhảy thích ứng cho các kết quả khá tốt.

3.3. Các số liệu so sánh

Nếu gọi số phép so sánh cần phải thực hiện là N_s , với một mẫu video có tổng số khung là K , các thuật toán tuyến tính sẽ cần số phép so sánh là: $N_{st} = K - 1$. Theo thí dụ đã nêu, tổng số khung của dòng video $K = 50$ thì $N_{st} = 49$.

Trong trường hợp sử dụng kỹ thuật nhảy đều, $d = 2$ và chỉ có các khung lẻ được so sánh, số phép so sánh cần phải sử dụng là $\frac{K}{2}$. Mặt khác mỗi khi phát hiện 2 khung nằm ở 2 lia khác nhau sẽ phải cần thêm một phép so sánh nữa để xác định chính xác ranh giới lia. Nếu số ranh giới lia của dòng video là r thì số phép so sánh phải thêm vào bằng r . Tổng số phép so sánh trong trường hợp này là: $N_{sd} = \frac{K}{2} + r$. Theo thí dụ đã nêu ($K = 50$; $r = 2$) sẽ tính được $N_{sd} = 27$. Như vậy N_s đã giảm được đáng kể.

Trường hợp sử dụng kỹ thuật nhảy động (bước nhảy thích ứng), việc xác định N_{stu} phức tạp hơn. Trước hết, so sánh giữa các khung k_1 và k_3 ($d=2$). Vì chúng nằm trong cùng một lia, nên tiếp tục so sánh giữa các khung k_3 và k_6 ($d=3$) và sau đó so sánh giữa các khung k_6 và k_{10} ($d=4$)... cứ tiếp tục cho đến khi so sánh giữa các khung k_{15} và k_{21} ($d=6$). Đến đây phát hiện ra 2 khung này nằm trong 2 lia khác nhau. Như vậy bắt đầu trong khoảng từ khung k_{15} đến k_{21} , quay lại sử dụng kỹ thuật nhảy đều với bước nhảy $d = 2$ để xác định được ranh giới lia giữa khung k_{17} và k_{18} . Chỉ cần tổng số 8 phép so sánh cho một lia có 17 khung. Hoàn toàn tương tự, tính được $N_{stu} = 20$. Kỹ thuật nhảy động tỏ ra rất hiệu quả khi số khung của một lia lớn hơn.

4. Phần mềm *PhanDoanVideo06* và một số kết quả thực nghiệm

Phần mềm *PhanDoanVideo06* được xây dựng trên cơ sở áp dụng các kỹ thuật nhảy đều và nhảy thích ứng. Nó có khả năng thu nhận dòng video từ các đầu vào camera hoặc card TV và lưu lại dưới dạng các file AVI hoặc các ảnh bitmap. Tuy nhiên chức năng chính của *PhanDoanVideo06* là thực hiện phân đoạn dữ liệu video từ các file định dạng AVI. Trong quá trình thể hiện lại các file AVI này, nó sẽ tự động phát hiện ranh giới giữa các lia video. Giao diện chính của chương trình như trên hình 5. Phần mềm *PhanDoanVideo06* thực hiện phân đoạn dữ liệu video theo các phương pháp khác nhau để có thể so sánh các kết quả.

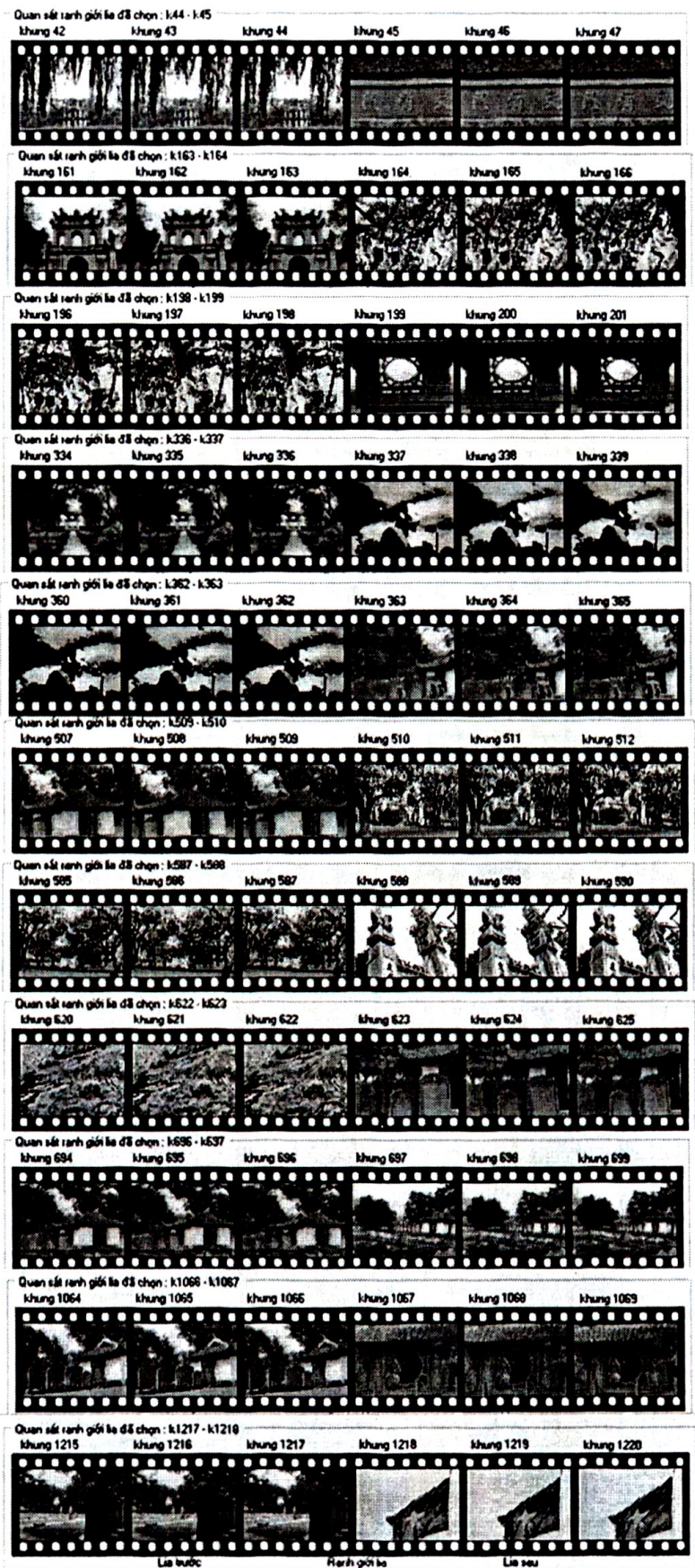


Hình 5. Giao diện chính của chương trình

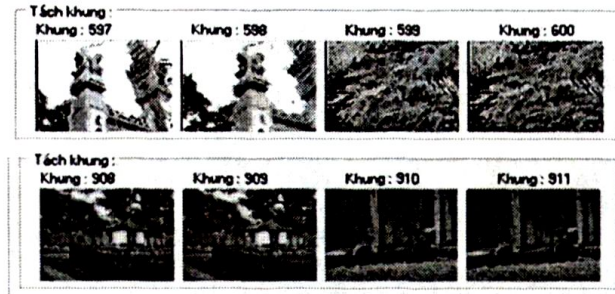
Các kết quả sau đây được thực hiện trên các file video dạng AVI, không nén, kích thước khung hình 160×120, 24 bit màu. Ví dụ file video *HaNoi29_thu.avi* có kích thước 74,4 MB, thực tế gồm có 14 lia với 1301 khung, được chuyển lia theo kiểu cắt lia với 13 cắt lia.

Khi thực hiện phân đoạn bằng phương pháp dựa trên thuật toán tuyến tính đã phát hiện được đúng 11 cắt lia (hình 6). Và đã bỏ sót mất 2 cắt lia. Đó là các cắt lia giữa các khung k_{598} và k_{599} ; k_{909} và k_{910} (hình 7), với thời gian tính toán là 148 giây.

Khi thực hiện phân đoạn dựa trên kỹ thuật nhảy thích ứng cũng đã phát hiện được đúng 11 cắt lia, nhưng phát hiện nhầm thêm 2 chuyển lia (thực sự không có chuyển lia). Đó là ranh giới giữa các khung k_{168} và k_{169} ; k_{192} và k_{193} (hình 8). Kết quả đã bỏ sót mất 2 cắt lia. Đó là các cắt lia giữa các khung k_{198} và k_{199} ; k_{598} và k_{599} . Tuy nhiên thời gian thực hiện trên cùng một file video *HaNoi29_thu.avi* đã rút ngắn được là 13 giây so với phương pháp phân đoạn dựa trên thuật toán tuyến tính.



Hình 6. Kết quả phát hiện lia bằng thuật toán tuyến tính



Hình 7. Hai cắt lia bỏ sót khi phân đoạn dựa trên thuật toán tuyến tính



Hình 8. Hai cắt lia giả phát hiện nhầm khi sử dụng kỹ thuật nhảy động thích ứng

Các kết quả phân đoạn so sánh giữa 2 phương pháp được tóm tắt trong bảng 1.

KẾT QUẢ PHÂN ĐOẠN VIDEO

File video: D:\Multimedia Data\Video\AVI24\Phong au\HaNoi29_thu.AVI
 (1031 khung, 14 lia, kích thước khung 160x120, 24 bit màu)
 Phát hiện được bằng thuật toán tuyến tính (TT):
 Số ranh giới lia (RGL): 11
 Thời gian bắt đầu: 15:44:06
 Thời gian kết thúc: 15:46:34
 Ngưỡng pixel: 20/100
 Ngưỡng block: 100

Phát hiện được bằng phương pháp nhảy động (ND):
 Số ranh giới lia (RGL): 13
 Thời gian bắt đầu: 15:47:57
 Thời gian kết thúc: 15:50:12
 Ngưỡng pixel: 20/100
 Ngưỡng block: 30

Các RGL thực tế		Các RGL phát hiện bằng thuật toán TT		Các RGL phát hiện bằng phương pháp ND		Ghi chú
<i>li</i>	<i>li+1</i>	<i>li</i>	<i>li+1</i>	<i>li</i>	<i>li+1</i>	
44	45	44	45	44	45	
163	164	163	164	163	164	
				168	169	Phát hiện nhầm RGL
				192	193	Phát hiện nhầm RGL
198	199	198	199			RGL bị bỏ sót
336	337	336	337	336	337	
362	363	362	363	362	363	
509	510	509	510	509	510	
587	588	587	588	587	588	
598	599					RGL bị bỏ sót
622	623	622	623	622	623	
696	697	696	697	696	697	
909	910			909	910	RGL bị bỏ sót
1066	1067	1066	1067	1066	1067	
1217	1218	1217	1218	1217	1218	

Bảng 1. Tóm tắt kết quả phân đoạn thực hiện trên file video *HaNoi29_thu.avi*

Phần mềm *PhanDoanVideo06* cũng đã được thử nghiệm trên một số file video ở các thể loại khác nhau (phim, bản tin thời sự, chương trình quảng cáo) và cho kết quả tin cậy. Trong khi số lượng phép tính giảm đáng kể thì độ chính xác là có thể chấp nhận được. Khả năng cải thiện của thuật toán càng thể hiện rõ ràng hơn trong các trường hợp đoạn video sử dụng để phân đoạn có kích thước đủ lớn.

5. Kết luận

Bài báo đã đề cập đến những vấn đề về phân đoạn dữ liệu video, những khó khăn chính sẽ gặp phải. Qua việc tóm tắt một số kỹ thuật phân đoạn hiện nay, chủ yếu là các kỹ thuật dựa trên đặc tính hình ảnh mức thấp của dữ liệu video, bài báo đề xuất một kỹ thuật nhằm nâng cao khả năng cho các thuật toán phân đoạn. Một số kết quả thực nghiệm của phần mềm tự động phân đoạn dữ liệu video *PhanDoanVideo06* do các tác giả xây dựng đã được đưa ra. Đây là sự kết hợp giữa kỹ thuật phân đoạn video đã được các tác giả đề xuất [7] với việc áp dụng các kỹ thuật nhảy đều và nhảy thích ứng. Mặc dù mới chỉ là các kết quả bước đầu nhưng công cụ này có thể tiếp tục phát triển để hỗ trợ đắc lực cho quá trình xây dựng các cơ sở dữ liệu video, một trong những vấn đề thời sự hiện nay.

TÀI LIỆU THAM KHẢO

1. Ahmed K. Elmagarmid, H. Jiang, A. Helal, A. Joshi, and M. Ahmed, *Video Database Systems - Issues, Products and Applications*, Kulwer Academic Publishers, 2001, 126p.
2. Borko Furht, Stephen W. Smoliar, HongJiang Zhang, *Video and Image Processing in Multimedia Systems*, Kluwer Academic Publishers, Boston/Dordrecht/London, 1995, 376p.
3. Đỗ Trung Tuấn, Lương Xuân Cương, Khun Piseth, Nguyễn Văn Tảo, Về Xử lý dữ liệu video. *Tạp chí Khoa học-Khoa học Tự nhiên và Công nghệ*, Đại học Quốc gia Hà Nội. T.XIX, No3, 2003, pp 48-56.
4. J. R. Parker. *Algorithms for Image Processing and Computer Vision*. Wiley Computer Publishing, John Wiley & Sons, Inc, New Yor/Toronto/Singapore, 1997, 417p.
5. JungHwan Oh Kien A. Hua. Efficient and Cost-effective Techniques for Browsing and Indexing Large Video Databases. *Computer Science Program, School of EECS University of Central Florida*, Orlando, 2001, pp 2328-2362.
6. Lương Mạnh Bá, Nguyễn Thanh Thủy, *Nhập môn xử lý ảnh số*, NXB Khoa học Kỹ thuật, Hà Nội, 1998, 175 trang.
7. Lương Xuân Cương, Đỗ Xuân Tiến, Đỗ Trung Tuấn. Một kỹ thuật chỉ số hoá tự động dữ liệu video dựa trên đánh dấu vùng nền, *Tạp chí Khoa học-Khoa học Tự nhiên và Công nghệ*, Đại học Quốc gia Hà Nội. T.XXII, No2, 2006, pp 1-11.
8. Ramesh Jain. Rangachar Kasturi, Brian G. Schunck. *Machine Vision*. McGraw - Hill, Inc, New York, 1995, 549p.