TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

KHOA CÔNG NGHỆ THÔNG TIN

Báo cáo quá trình theo chủ đề

Adaptive Thresholding Using the Integral Image

(Lấy ngưỡng ảnh thích nghi dựa trên tích hợp ảnh)

HV phiên dịch:

Phạm Dương Thành Long

MSSV:

51603190

Tác giả bài báo:

Derek Bradley∗

Carleton University, Canada

derek@derekbradley.ca

Gerhard Roth

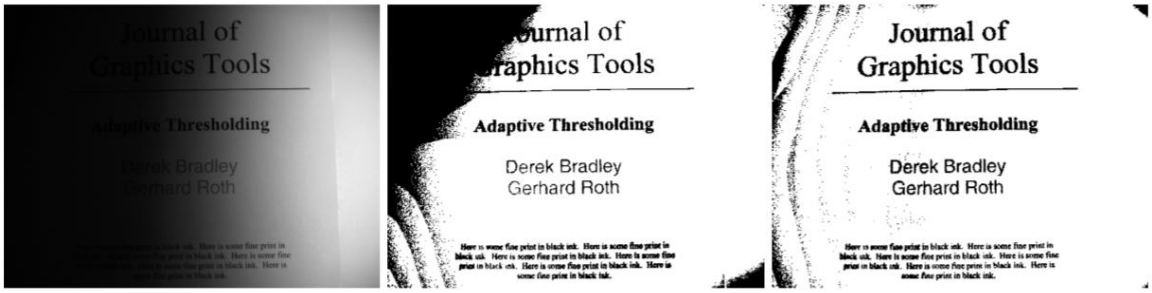
National Research Council of Canada

Gerhard.Roth@nrc-cnrc.gc.

MỤC LỤC

DANH MỤC CÁC HÌNH VẼ VÀ BẢNG BIỂU

I. Tổng quan:



(a) (b) (c)

Hình 1.1 : tập các ảnh được lấy ngưỡng hình ảnh thích nghi theo thời gian thực.

- (a): Ảnh đầu vào.
- (b): Ảnh sử dụng kỹ thuật trước đây của Wellner.
- (c): Ảnh sử dụng kỹ thuật lấy ngưỡng thích nghi trong bài báo trên

1.1 Mục đích

Mục đích chính của việc lấy ngưỡng hình ảnh trong việc sử lý ảnh nhằm giúp phân loại ra các pixel thuộc 2 vùng ***sáng*** *(light) và* ***tối*** *(dark)* trong một bức ảnh. lấy ngưỡng thích nghi là một hình thức lấy ngưỡng có tính đến việc chiếu sáng ở các vị trí như thế nào trong không gian của một bức ảnh.

Phương pháp lấy ngưỡng thích nghi theo thời gian thực được thực hiện bằng cách sử dụng tích hợp ảnh đầu vào. Đây là một phương pháp đơn giản, mạnh mẽ và dễ thực hiện. Kỹ thuật này phù hợp để xử lý các luồng video trực tiếp ở tốc độ khung hình chuẩn theo thời gian thực. Ngoài ra phương pháp trên còn được ứng dụng trong xử lý ảnh tăng cường hay ứng dụng đồ họa máy tính.

1.2 Giới thiệu phương pháp:

Phương pháp lấy ngưỡng dù cơ bản hay nâng cao đều sẽ tập trung vào phân vùng *(segments)* một ảnh đầu vào dựa trên một đặc điểm nhất định trên các pixel trong ảnh đó (ví dụ: giá trị cường độ - *intensity)* Và cho ra kết quả là một ảnh nhị phân đã được phân loại từng pixel dựa trên 2 vùng sáng và tối. và điều này cũng được ứng dụng trong việc chụp ảnh *HDR (High Dynamic Range)*

Tuy nhiên, có thể thấy phương pháp lấy ngưỡng cố định cơ bản chỉ giải quyết được những hình ảnh có độ chiếu sáng cố định và đều này trong thực tế thì thường không tồn tại vì theo một luồng video thì ảnh sáng sẽ luôn thay đổi ở từng góc độ khác nhau do điều kiện ánh sáng tự nhiên.

Để giải quyết vấn đề về sự thay đổi độ sáng trong một bức ảnh thì phương pháp lúc này ta có đó là lấy ngưỡng thích nghi. Mấu chốt của phương pháp này đó chính là

từng pixel trong ảnh sẽ được tính một giá trị ngưỡng khác nhau. Có rất nhiều phương pháp cho việc lấy ngưỡng trên nhưng trong phạm vi bài báo trên sẽ đề cập đến là sử dụng hình ảnh tích hợp. Phương pháp trên được kế thừa và mở rộng từ phương pháp gốc của Wellner *(1993).*

2. Background:

2.1 Real-Time Adaptive Thresholding:

Trong bài báo này, tôi sẽ tập trung vào các việc lấy ngưỡng cho các hình ảnh ngưỡng thích ứng từ một luồng video trực tiếp. Để duy trì hiệu suất thực hiện theo thời gian thực, thuật toán trên phải được ***giới hạn ở một số lần lặp không đổi nhỏ*** qua mỗi hình ảnh. Lấy ngưỡng thường là một nhiệm vụ phụ phục vụ cho một quy trình lớn hơn.

> *Ví dụ: Trong thực tế tăng cường, hình ảnh đầu vào phải được phân vùng để xác định vị trí các điểm đánh dấu đã biết trong bối cảnh được sử dụng để tự động thiết lập tư thế của máy ảnh. Do đó, một kỹ thuật ngưỡng thích ứng đơn giản và nhanh chóng là một công cụ quan trọng cho bài toán lớn.*

2.2 Integral Images:

Một hình ảnh tích hợp - *integral image* (còn được gọi là bảng tổng hợp - *summed-area table*) là một công cụ có thể được sử dụng bất cứ khi nào ta có hàm từ pixel đến số thực ***f(x, y)*** *(ví dụ: cường độ pixel)* và ta muốn tính toán tổng của hàm này trên một vùng hình chữ nhật của hình ảnh.

Mặc khác, nếu ta cần tính tổng trên nhiều cửa sổ hình chữ nhật chồng chéo, ta có thể sử dụng một *integral image* và đạt được số *operations* không đổi trên mỗi hình chữ nhật chỉ với một lượng tiền xử lý tuyến tính.

Để tính toán hình ảnh tích hợp, ta sẽ lưu trữ tại mỗi vị trí $\boldsymbol{I}(\boldsymbol{x},\boldsymbol{y})$, tổng của tất cả các số hạng $\boldsymbol{f}(\boldsymbol{x},\boldsymbol{y})$ ở bên trái và phía trên pixel ***(x, y).*** Điều này được thực hiện trong thời gian tuyến tính bằng cách sử dụng phương trình sau cho mỗi pixel *(tính luôn cả biên của ảnh).*

$$I(x,y) = f(x,y) + I(x-1,y) + I(x,y-1) - I(x-1,y-1). \quad (1)$$

Để mình hoạ ta sẽ qua ví dụ sau:

4	1	2	2
0	4	1	3
3	1	0	4
2	1	3	2

(a)

4	5	7	9
4	9	12	17
7	13	16	25
9	16	22	33

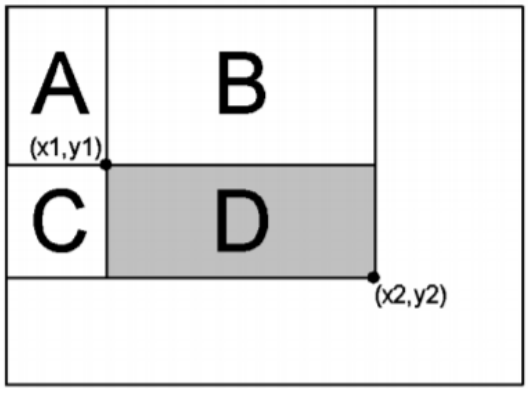
(b)

Hình 2.1: The integral image

- (a): Một đầu vào đơn giản với các giá trị hình ảnh
- (b): Một ảnh đã được tính integral image
- (c): Sử dụng integral image để tính tổng trên hình chữ nhật

Hình 2.1 (a) và (b) minh họa cho việc tính toán của một integral image. Khi ta có integral image, tổng của hàm cho bất kỳ hình chữ nhật nào có góc $upper\_left\ (x1, y1)$ và góc $lower\_right(\ x2, y2)$ có thể được tính trong thời gian không đổi bằng phương trình sau:

$$\sum_{x=x_1}^{x_2}\sum_{y=y_1}^{y_2} f(x,y) = I(x_2,y_2) - I(x_2,y_1-1) - I(x_1-1,y_2) + I(x_1-1,y_1-1). \quad (2)$$



Hình 2.2: Minh họa việc sử dụng integral image để tính tổng trên hình chữ nhật

Hình 2.2 minh họa việc tính toán tổng $\boldsymbol{f(x,y)}$ trên ***hình chữ nhật D*** sử dụng phương trình (2) tương đương với tính toán các tổng trên các hình chữ nhật:

$$(\boldsymbol{A} + \boldsymbol{B} + \boldsymbol{C} + \boldsymbol{D}) - (\boldsymbol{A} + \boldsymbol{B}) - (\boldsymbol{A} + \boldsymbol{C}) + \boldsymbol{A}$$

3. Các vấn đề kỹ thuật:

3.1 Ý tưởng gốc của thuật toán Wellner:

3.1.1 Sơ lược ý tưởng

Kỹ thuật lấy ngưỡng thích nghi trong thuật toán Wellner có thể hiểu là mỗi pixel sẽ được so sánh với mức trung bình của các pixel lân cận xung quoanh nó. Cụ thể, trung bình di chuyển xấp sỉ của các pixel ***s*** cuối cùng đã nhìn thấy sẽ được tính toán trong khi đang duyệt qua một ảnh. Nếu giá trị của pixel hiện tại thấp hơn ***t*** phần trăm so với mức trung bình thì nó được gán màu đen. Ngược lại, nó được gán sang màu trắng.

Phương pháp này hoạt động khá tốt vì nó so sánh một pixel với mức trung bình của các pixel lân cận giúp bảo toàn các đường tương phản cứng *(hard contrast)* và bỏ qua các thay đổi độ dốc mềm *(soft gradient)*. Ưu điểm của phương pháp này là chỉ cần duyệt một lần đi qua hình ảnh. Wellner sử dụng *1/8* chiều rộng hình ảnh cho giá trị *s* và *15* cho giá trị *t*.

3.1.2 Hạn chế:

Tuy nhiên, Hạn chế với phương pháp này đó là nó phụ thuộc vào thứ tự quét của các pixel. Ngoài ra, trung bình di chuyển không phản ánh đúng hoàn toàn các pixel xung quanh ở mỗi bước di chyển vì các mẫu lân cận ***không được phân bố đều theo mọi hướng***. Do đó giải pháp lúc này là kỹ thuật sử dụng hình ảnh tích hợp (và hy sinh một lần lặp bổ sung thông qua hình ảnh), giúp giải quyết được vấn đề trên.

3.2 Kỹ thuật lấy ngưỡng dựa trên việc tích hợp ảnh:

Đây là một kỹ thuật của chúng tôi đơn giản, gọn, dễ lập trình và tạo ra cùng một đầu ra độc lập với trong xử lý hình ảnh. Thay vì tính toán trung bình duyệt qua của pixel s cuối cùng được nhìn thấy nhìn thấy, ta sẽ tính trung bình theo một cửa sổ với kích thước ***s x s*** (pixel) xung quanh mỗi pixel đó.

Đây là cách lấy mức trung bình tốt hơn để so sánh vì nó xem xét các pixel lân cận ở tất cả các mặt. việc tính trung bình được thực hiện trong thời gian tuyến tính bằng cách sử dụng ảnh tích hợp. Ta tính ảnh tích hợp trong lần duyệt đầu tiên qua ảnh đầu vào. Trong lần duyệt thứ hai, ta tính trung bình ***s x s*** bằng cách sử dụng ảnh tích hợp cho mỗi pixel trong thời gian không đổi và sau đó thực hiện so sánh. Nếu giá trị của pixel hiện tại nhỏ hơn t phần trăm so với mức trung bình này thì nó sẽ gán thành màu

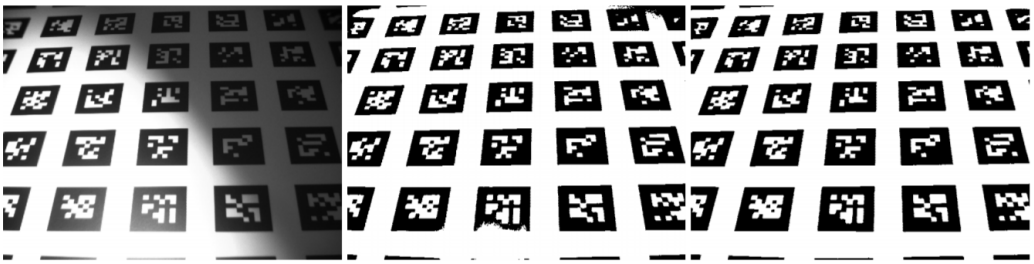
đen, ngược lại, nó được gán thành màu trắng. Mã giả sau sẽ thể hiện kỹ thuật của ta cho ảnh đầu vào, hình ảnh nhị phân đầu ra, chiều rộng hình ảnh *w* và chiều cao hình ảnh *h*.

procedure $AdaptiveThreshold(in, out, w, h)$

```
1:  for i = 0 to w do
2:      sum ← 0
3:      for j = 0 to h do
4:          sum ← sum + in[i, j]
5:          if i = 0 then
6:              intImg[i, j] ← sum
7:          else
8:              intImg[i, j] ← intImg[i − 1, j] + sum
9:          end if
10:     end for
11: end for
12: for i = 0 to w do
13:     for j = 0 to h do
14:         x1 ← i − s/2 {border checking is not shown}
15:         x2 ← i + s/2
16:         y1 ← j − s/2
17:         y2 ← j + s/2
18:         count ← (x2 − x1) × (y2 − y1)
19:         sum ← intImg[x2, y2] − intImg[x2, y1 − 1] − intImg[x1 − 1, y2]
20:         if (in[i, j] × count) ≤ (sum × (100 − t)/100) then
21:             out[i, j] ← 0
22:         else
23:             out[i, j] ← 255
24:         end if
25:     end for
26: end for
```

3.3 Đánh giá và ví dụ:

Phương pháp trên mạng lại tốc độ khá tốt trong việc xử lý video trực tiếp với khung hình trung bình trên 60. Mặc khác, do có thêm 1 lần duyệt nên tốc độ thực tế sẽ chậm hơn so với thực toán gốc cửa wellner khoảng 2.5 lần .Tuy nhiên ta vẫn đạt được tốc độ khung hình thời gian thực khá tốt và phân vùng tốt hơn so với phương pháp cũ.



Hình 3.1: Ví dụ về xử lý đánh dấu ảnh trên thực tế:

- (a): Ảnh đầu vào
- (b): Ảnh sử dụng phương pháp wellner
- (c): Ảnh sử dụng phong pháp cải tiến.

Hình 1.1 và Hình 3.1 của cả phương pháp wellner và sau cải tiến.

- Hình 1.1 minh họa một ví dụ về văn bản với phần bóng trong ảnh rất tối. Phương pháp cải tiến có thể phân vùng tất cả các văn bản trong hình ảnh.
- Hình 3.1 minh họa một ví dụ thực tế ảnh tăng cường với một số điểm đánh dấu phẳng được theo dõi trong suốt luồng video. Trong ví dụ này, kỹ thuật cải tiến cung cấp phân vùng gần hoàn hảo mặc dù có những thay đổi chiếu sáng mạnh trong ảnh. Kỹ thuật Wellner thì thất bại ở sự phản chiếu hình ảnh ở trung tâm của hình ảnh ở phía dưới và phần đổ bóng ở các góc trên cùng.

4. Tổng kết:

Phương pháp lấy ngưỡng thích nghi dựa trên việc tích hợp ảnh là một cách giải quyết các bài toán yêu cầu xử lý các luồng video trên thời gian thực. kỹ thuật trên cũng phù hợp cho các khung cảnh có độ chiếu sáng mạnh hoặc đỗ bóng không đều.

Bên cạnh đó, các sự thay đổi tạm thời trong chiếu sáng cũng được xử lý một cách tự động, đây là đều mà phương pháp lấy ngưỡng toàn cục(global) không giải quyế được. Tuy nhiên, hạn chế chính của phương pháp này là ta phải *xử lý hình ảnh hai lần*. Nhưng đây không phải là một nhược điểm đáng kể với tốc độ xử lý hiện tại khá nhanh và kỹ thuật của này vẫn phù hợp với các ứng dụng yêu cầu thời gian thực.

Số lượng công việc cần xử lý (*processing)* cũng có thể được giảm thiểu bằng cách thực hiện việc lấy ngưỡng tại pixel tại $(\boldsymbol{i} - \frac{\boldsymbol{s}}{\boldsymbol{2}}, \boldsymbol{j} - \frac{\boldsymbol{s}}{\boldsymbol{2}})$ song song với việc tính toán hình ảnh tích hợp tại pixel tại (i, j). Trong trường hợp này, một lượng processing không đổi phải được thực hiện cho số pixel sau:

$$(w \times h) + (\frac{s}{2} \times w) + (\frac{s}{2} \times h) - (\frac{s^2}{4}).$$

Kỹ thuật lấy ngưỡng này cho phép ta giả định rằng: Hình ảnh hiện tại đang xử lý đã phần là chứa các pixel nền - *background pixels (được phân vùng oạn thành màu trắng)* và các *foreground pixels* được phân phối khá nhiều trong hình ảnh. Trong trường hợp thành phần foreground pixels lớn hơn ***s x s*** *(pixel)* , tâm của các thành phần này sẽ được phân loại không chính xác và được nhận biết như là *background*. Tuy nhiên, vấn đề này có thể được giảm thiểu bằng cách sử dụng *kernel* lớn hơn *(nghĩa là tăng kích thước s x s),* Tuy nhiên, các chi tiết tốt trong phân vùng có thể bị mất đi và kích thước kernel phải được đặt phù hợp theo ứng dụng.