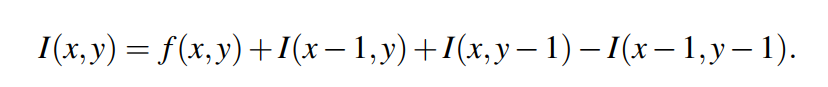
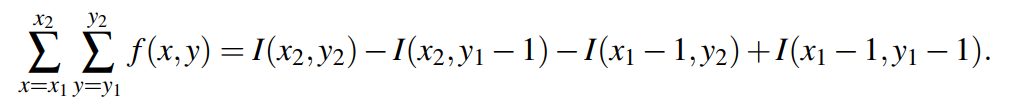
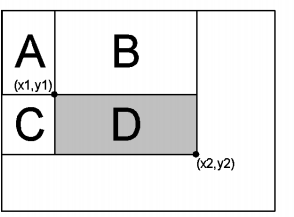
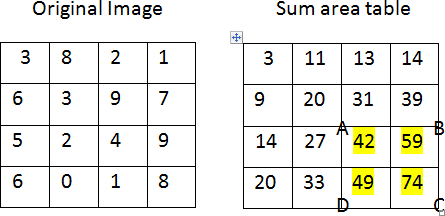
**Adaptive thresholding using the integral image**Định ngưỡng thích ứng bằng cách sử dụng hình ảnh tích phân là một kỹ thuật phù hợp để xử lý các luồng video trực tiếp hoặc dữ liệu thời gian thực, giúp áp dụng cho các ứng dụng tương tác như thực tế được lập luận. Để duy trì hiệu suất thời gian thực, thuật toán ngưỡng phải được giới hạn ở một số lượng nhỏ các lần lặp qua mỗi khung hình. Ý tưởng cốt lõi là sử dụng một ma trận phụ được gọi là hình ảnh tích phân để đánh giá ngưỡng trong khoảng thời gian không đổi.

Hình ảnh tích phân, còn được gọi là bảng tổng diện tích, là một công cụ mạnh mẽ cho phép chúng ta tính tổng trên nhiều vùng chồng chéo trong một số thao tác cố định trên mỗi cửa sổ.

Để tính toán hình ảnh tích phân, chúng ta lưu trữ tại mỗi vị trí, I (x, y), tổng của tất cả các số hạng f (x, y) ở bên trái và bên trên pixel (x, y). Điều này được thực hiện theo thời gian tuyến tính bằng cách sử dụng phương trình sau cho mỗi pixel.

Sau khi chúng ta có được hình ảnh tích phân, tổng của bất kỳ hình chữ nhật nào có góc trên bên trái (𝑥1, 𝑦1) và góc dưới bên phải (𝑥2, 𝑦2) có thể được tính trong thời gian không đổi bằng cách sử dụng công thức dưới đây:





Trái: Hình ảnh đầu vào.

Ở giữa: Hình ảnh tích phân.

Phải: Sử dụng hình tích phân để tính tổng trên hình chữ nhật D.

Chúng ta tính giá trị trung bình của một cửa sổ 𝑠 𝑥 𝑠 pixel tập trung xung quanh mỗi pixel. Đây là mức trung bình tốt hơn để so sánh vì nó xem xét các pixel lân cận trên tất cả các mặt.Đầu tiên chúng ta tính toán hình ảnh tích phân thông qua hình ảnh đầu vào. Thứ hai, chúng ta tính giá trị trung bình 𝑠 𝑥 𝑠 bằng cách sử dụng hình ảnh tích phân cho mỗi pixel trong thời gian không đổi rồi thực hiện phép so sánh. Nếu giá trị của pixel hiện tại là 𝑡 phần trăm nhỏ hơn mức trung bình này thì nó được đặt thành màu đen, ngược lại nó được đặt thành màu trắng. Sau mã giả thể hiện kỹ thuật của chúng tôi đối với hình ảnh đầu vào 𝑖𝑛, hình ảnh nhị phân đầu ra 𝑜𝑢𝑡, chiều rộng hình ảnh 𝑤 và chiều cao hình ảnh ℎ.

**procedure** AdaptiveThreshold(in, out, w, h)  
**for** 𝑖 = 0 to 𝑤 **do**𝑠𝑢𝑚 ← 0  
**for** 𝑗 = 0 to h **do**𝑠𝑢𝑚 ← 𝑠𝑢𝑚 + 𝑖𝑛[𝑖, 𝑗]  
**if** 𝑖 = 0 **then**𝑖𝑛𝑡𝐼𝑚𝑔[𝑖, 𝑗] ← 𝑠𝑢𝑚  
**else**𝑖𝑛𝑡𝐼𝑚𝑔[𝑖, 𝑗] ← 𝑖𝑛𝑡𝐼𝑚𝑔[𝑖 - 1, 𝑗] + 𝑠𝑢𝑚  
**end if  
end for  
end for  
for** 𝑖 = 0 to 𝑤 **do  
for** 𝑗 = 0 to ℎ **do**𝑥1 ← 𝑖 - 𝑠/2 {𝑏𝑜𝑟𝑑𝑒𝑟 𝑐ℎ𝑒𝑐𝑘𝑖𝑛𝑔 𝑖𝑠 𝑛𝑜𝑡 𝑠ℎ𝑜𝑤𝑛}  
𝑥2 ← 𝑖 + 𝑠/2  
𝑦1 ← 𝑗 - 𝑠/2  
𝑦2 ← 𝑗 + 𝑠/2  
𝑐𝑜𝑢𝑛𝑡 ← (𝑥2 - 𝑥1) × (𝑦2 - 𝑦1)  
𝑠𝑢𝑚 ← 𝑖𝑛𝑡𝐼𝑚𝑔[𝑥2, 𝑦2] - 𝑖𝑛𝑡𝐼𝑚𝑔[𝑥2, 𝑦1 - 1] - 𝑖𝑛𝑡𝐼𝑚𝑔[𝑥1 - 1, 𝑦2]  
+ 𝑖𝑛𝑡𝐼𝑚𝑔[𝑥1 - 1, 𝑦1 - 1]  
**if** (𝑖𝑛[𝑖, 𝑗] × 𝑐𝑜𝑢𝑛𝑡) ≤ (𝑠𝑢𝑚 × (100 - 𝑡)/100) **then**𝑜𝑢𝑡[𝑖, 𝑗] ← 0  
**else**𝑜𝑢𝑡[𝑖, 𝑗] ← 255  
**end if  
end for  
end for**

**Code matlab xử lý ngưỡng tích hợp sử dụng ảnh tích phân**

I=imread('./2.jpg');

figure

imshow(I)

I=rgb2gray(I);

I=double(I);

sz=size(I);

h=sz(1);

w=sz(2);

in=I;

s=h/16;

t=20;

for i=1:w

sum =0;

for j=1:h

sum=sum + in(j,i);

if i==1

intImg(j,i)=sum;

else

intImg(j,i)=intImg(j,i-1) + sum;

end

end

end

for i=1:w

for j=1:h

x1=max(i-round(s/2),2);

x2=min(i+round(s/2),w);

y1=max(j-round(s/2),2);

y2=min(j+round(s/2),h);

count =(x2-x1)\*(y2-y1);

sum =intImg(y2,x2)-intImg(y1-1,x2)-intImg(y2,x1-1) + intImg(y1-1,x1-1);

if (in(j,i)\*count)<=(sum \*(100 -t)/100)

out(j,i)=0;

else

out(j,i)=255;

end

end

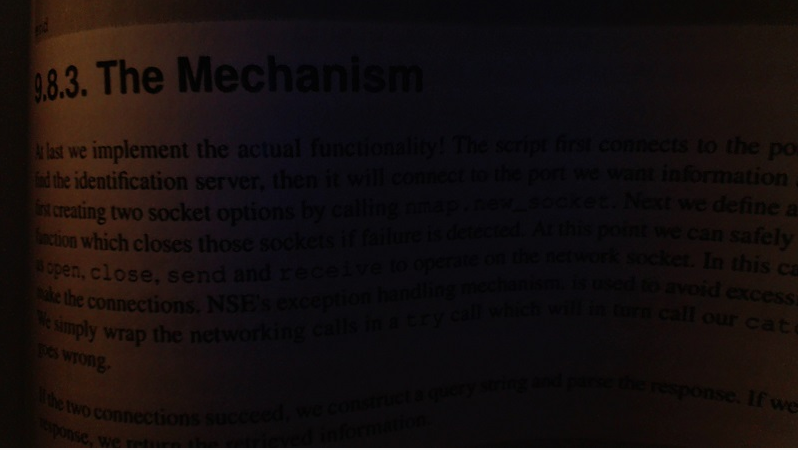
end

figure

imshow(out)

Kết quả kiểm tra:

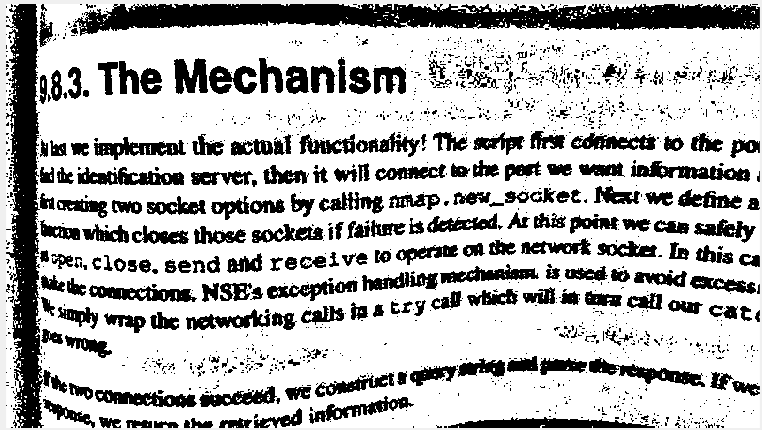
Ảnh gốc:



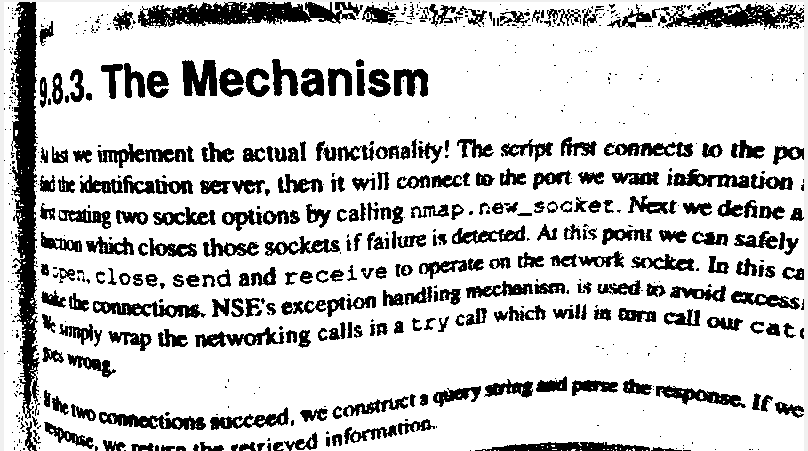
Với s=h/8, t=10

Gía trị s của cửa số s x s để tính giá trị trung bình của cửa sổ s x s tập trung vào mỗi pixel.

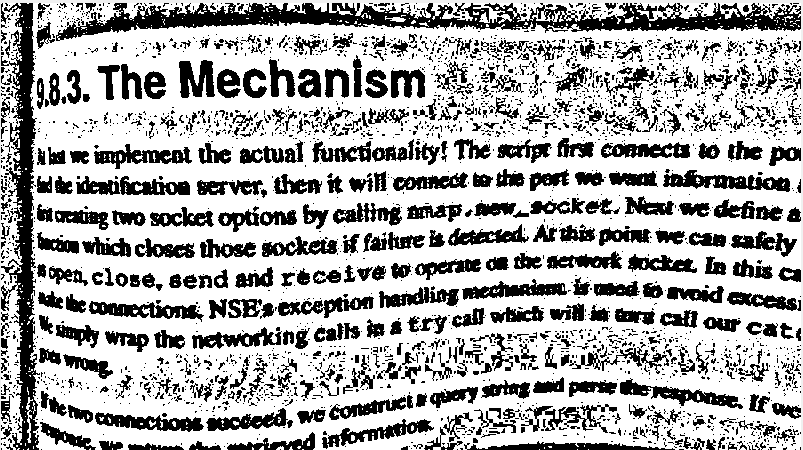
t: tỉ lệ phần trăm so với giá trị trung bình của cửa số s x s cho mỗi pixel.



Với s=h/8, t=20



Với s=h/16, t=10



Ưu nhược điểm:

Ưu điểm:

* Thích hợp xử lý ảnh có ánh sáng thay đổi
* Xử lý trong thời gian thực, có thể tính toán nhanh hơn rất là nhiều so với thông thường.

Nhược điểm:

* Không hoạt động tốt trong trường hợp ảnh có độ sáng cao

Tài liệu tham khảo:

* [1] Derek Bradley and Gerhard Roth, “Adaptive thresholding using the integral image,”  
  Journal of Graphics Tools, Vol. 12, No2, 2007, pp. 13–21.
* [1] Pierre D. Wellner, “Adaptive thresholding for the digitaldesk,” Tech. Rep. EPC-93–  
  110, EuroPARC, 1993