

مقدمه ای بر پردازش تصویر

روش لبه گیری پیشرفته :

1.1: هدف لبه گیری :

اگر لبه ها در تصویر به خوبی مشخص شوند ، تمام اشیا به خوبی موقعیت یابی میشوند و مشخصات مهم شی مانند مساحت، محیط و شکل آن قابل اندازه گیری میشوند. لبه گیری قسمتی از پردازشی هست که به آن segmentation (یا قطعه یابی) میگویند. ناحیه هایی که توسط لبه گیری مشخص میشوند در مراحل بعد با پردازش های دیگر نوع شی ناحیه را مشخص میکنند البته ممکن است این لبه گیری تمام چیزی باشد که در قطعه یابی مورد نیاز است مخصوصا زمانی که اشیا در درون خط ها باشند.

افزایش لبه ها در تصویر ، تضاد لبه ها و زمینه را در تصویر مشخص تر میکند. Edge tracing (یا رد یابی لبه ها) پروسه ای است که مجموعه ی پیکسل لبه ها را ذخیره میکند. این انجام میشود بر روی لبه ها بصورت ساعت گرد یا پاد ساعتگرد.

1.2: دیدگاه سنتی و قضیه :

یکی از عمومی ترین و اصلی ترین تعریف های لبه ideal step edge است .

اولین پیچیدگی که به وجود می آید بخاطر دیجیتالی بودن است . تغییرات در سطح لبه ها ممکن است در تعداد زیادی از پیکسل ها بوجود بیاید . مکان واقعی لبه وسط سرازیری gray level از سطح بالا به پایین یا بر عکس (ramp) است. این در دنیا ریاضی به سادگی صورت میگیرد اما در تصویر رقمی از دنیای واقعی در این سرازیری ناهمواری بوجود می آید.

دومین پیچیدگی وجود خطا در تصویر است (noise) . تأثیرات خطا ، تأثیرات تصادفی هستند و توسط علم آماری قابل تشخیص هستند. تأثیر خطا بر روی تصاویر یک تغییر تصادفی در سطح خاکستری از پیکسلی به پیکسل دیگر بوجود میآورد و دست یابی به خط های هموار و یک ramp ایده آل بر روی لبه ها را غیر ممکن کرده.

مدل های لبه :

در نبود خطا هرگونه تغییری در gray level يك لبه ایجاد میکند. برای مشخص کردن لبه در gray level از فرمول زیر استفاده میکنیم :

$$V = (v_w * a_w + v_b + a_b) / (a_w + a_b)$$

v_w = سطح خاکستری رنگ سفید

v_b = سطح خاکستری رنگ مشکی

a_w = مساحت رنگ سفید

a_b = مساحت رنگ مشکی

خطا :

خطا در بیشتر مواقع با تأثیرش بر روی تصویر قابل شناسایی است و این با يك توزیع احتمالات با يك میانگین خاص و انحراف استاندارد بیان میشود.

در پردازش تصویر دو نوع مهم خطا وجود دارد. یک نوع آن خطا نشان-مستقل (signal-independent) است که مجموعه ای از gray-level ها است . این مجموعه با مجموعه پیکسل ها ی تصویر جمع میشود و تصویر با خطا را تولید میکند. این نوع خطا زمان جابجایی الکترونیکی تصویر از یک مکان به مکان دیگر صورت میگیرد. اگر A تصویر درست باشد و N خطاییست که اتفاق می افتد در اینصورت B تصویر نهایی است : $B=A+N$. خطا تصویر N ممکن است مشخصات آماری داشته باشد اما یک فرضیه عمومی این است که این از یک توزیع عادی با میانگین صفر و مقداری که اندازه گیری شده ، تبعیت میکند و دارای انحراف استاندارد است. این فرضیه به ساخت خطای مصنوعی کمک میکند که این وسیله برای آزمایش کردن الگوریتم لبه گیری است.

(مجموع توان 2 پیکسل ها) $y =$ و تعداد پیکسل ها $k =$ و (gray level مجموع) $x =$

Image :

$$\text{Mean} = x/k \quad \text{و} \quad \text{Standard deviation} = \sqrt{(y-x^2) / (k*k-1)}$$

(مجموع توان 2 پیکسل ها منهای میانگین) $y =$ و تعداد پیکسل ها $k =$ و (mean ها منهای gray level مجموع) $x =$

Noise :

$$\text{Mean} = x/k \quad \text{و} \quad \text{Standard deviation} = \sqrt{(y-x^2) / (k*k-1)}$$

نوع دیگر آن خطای نشان-وابسته (signal-dependent noise) است که در این نوع خطا تابعی از gray level نقطه ی مورد بررسی است . در این حالت پیروی از پروسه ای که تغییر دقیق gray-level را اندازه گیری میکند کار آمد نیست، زیرا وجود خطا باعث ایجاد اشتباه در شناسایی میشود. در اینجا سه روش مهم برای شناسایی لبه وجود دارد.

اولین روش، روش مشتق گیری است که در آن تغییر زیاد شدت مورد بررسی قرار میگیرد.

دومین روش، طراحی برنامه تطبیق قالب است که در اینجا لبه مدل شده مشخصات مختصری از لبه واقعی دارد.

سومین روش، مدل کردن ریاضی وار لبه . این روش پیچیده است ولی بسیار دقیق است.

روش مشتق گیری :

مشتق یعنی میزان تغییرات یک کمیت است و میزان تغییر زیاد میزان gray-level نزدیک لبه است و تغییرات کم آن نزدیک سطح ثابت است. اگر A را تابع تصویر در نظر بگیریم در این صورت گرادین آن میزان تغییرات است که یک بردار 2 بعدی است.

$$A(x,y) \rightarrow \text{grad}A = (A_x, A_y)$$

(البته می دانیم تابع تصویر گسسته است و نمی توانیم مستقیم از گرادین استفاده کنیم به همین دلیل از مقدار های تقریبی زیر استفاده میکنیم :

$$A_{x1} = A(x,y) - A(x-1,y) , A_{y1} = A(x,y) - A(x,y-1) \rightarrow \text{grad}1A = (A_{x1}, A_{y1}) \quad (\text{در نقطه ی } A(x-0.5, y-0.5))$$

$$A_{x2} = A(x+1,y) - A(x-1,y) , A_{y2} = A(x+1,y) - A(x,y-1) \rightarrow \text{grad}2A = (A_{x2}, A_{y2}) \quad (\text{در نقطه ی } A(x,y))$$

.(

گفتیم گرادیان یک بردار 2 بعدی است که طول وتر آن edge response (یا قدرت لبه) است و مقادیر زیر برای محاسبات مهم هستند :

$$Gmag = \sqrt{A_x^2 + A_y^2}, \quad Gdir = \arctan(A_y/A_x)$$

برای یک برنامه لبه گیری میتوان از فرمول زیر برای ارزیابی آن استفاده کرد :

$$E1 = \sum_{i=1,IA} [(1/\alpha * d(i)^2) / \max(IA, II)]$$

تعداد پیکسل‌های پیدا شده : IA

تعداد واقعی پیکسل‌های لب : II

فاصله پیکسل لبه واقعی و لبه پیدا شده : d(i)

مقداری برای مقیاس دادن که برای همه باید ثابت باشد : alpha

این ارزیابی فقط فاصله لبه واقعی و لبه پیدا شده را در نظر میگیرد و کاری به نادرستی مثبت و نادرستی منفی ندارد (نادرستی مثبت : لبه را پیدا کند که وجود نداشته باشد – نادرستی منفی : لبه را پیدا نکند در صورتیکه وجود داشته باشد) .

فرمولی دیگر برای ارزیابی برنامه لبه گیری وجود دارد که با وابستگی لبه محلی کار میکند، در اینجا که محدوده 3×3 پیکسل با مرکزیت پیکسل لبه مورد نظر ، در نظر گرفته میشود ، حال توابع زیر را در نظر میگیریم :

$$L(k) = \{ a(D, Dk) * a(k * \pi/4, D + \pi/4) \text{ اگر همسایه یک پیکسل لبه باشد} \}$$

$$R(k) = \{ a(D, Dk) * a(k * \pi/4, D - \pi/4) \text{ اگر همسایه یک پیکسل لبه باشد} \}$$

$$a(\alpha, \beta) = (\pi - \text{abs}(\alpha - \beta)) / \pi$$

که در اینجا d جهت پیکسل مرکزی است و d0 جهت پیکسل سمت راستی ، d1 جهت پیکسل بالا سمت راستی و .. d7 جهت پیکسل پایین سمت راست را میدهد .

در اینصورت C برابر است با میانگین بیشترین مقدار L(k) یا R(k) در کل تصویر و T برابر است با تقسیم 6 پیکسل دیگر که برای تمام منطقه 3×3 محاسبه میشود .

در اینصورت داریم :

$$E2 = \lambda * C + (1 - \lambda) * T \quad (\lambda = 0.8)$$

با استفاده از این ارزیابی ها نتیجه میگیریم که grad2 اندکی بهتر از grad1 است .

روش لبه گیری با قالب های پایه :

یک قالب برای لبه گیری استفاده از لبه گیری Sobel است که از دو آرایه ی زیر تشکیل شده :

$$S_x = [-1, 0, 1; -2, 0, 2; -1, 0, 1], \quad S_y = [-1, -2, -1; 0, 0, 0; 1, 2, 1]$$

تشکیل شده است . اعداد این آرایه ها وزن هر خانه هستند ، یعنی به ازای هر پیکسل يك فضای 3×3 در نظر گرفته میشود و برای آن S_x و S_y محاسبه میشود ، این مقادیر بجای A_x و A_y استفاده میشود یعنی بجای گرفتن گرادیان. برای هر پیکسل بجای بدست آوردن مقدار $\sqrt{S_x^2 + S_y^2}$ میتوان از $\text{abs}(S_x) + \text{abs}(S_y)$ استفاده کرد که سرعت بیشتری دارد. حال با يك آستانه گیری برای تمام مقادیر بدست آمده و انتخاب بزرگترین ها میتوان لبه ها را شناسایی کرد .

قالب دیگر ، قالب Kirsch است که در آن از آرایه های زیر استفاده میشود :

$$K_i \rightarrow 0 \leq i \leq 7$$

$$K_0 = [-3, -3, 5; -3, 0, 5; -3, -3, 5], K_1 = [-3, 5, 5; -3, 0, 5; -3, -3, -3], K_2 = [5, 5, 5; -3, 0, -3; -3, -3, -3]$$

$$K_3 = [5, 5, -3; 5, 0, -3; -3, -3, -3], K_4 = [5, -3, -3; 5, 0, -3; 5, -3, -3], K_5 = [-3, -3, -3; 5, 0, -3; 5, 5, -3]$$

$$K_6 = [-3, -3, -3; -3, 0, -3; 5, 5, 5], K_7 = [-3, -3, -3; -3, 0, 5; -3, 5, 5]$$

باز هم برای هر پیکسل تصویر 8 ماسک بالا را حساب میکنیم و ماکزیمم را بدست می آوریم (بجای اندازه ی گرادیان) و جهت آن برابر است با $\theta = \pi/4 * I$ باز هم برای بدست آوردن لبه ها از آستانه گیری استفاده میکنیم .

اگر این دو قالب را توسط توابع ذکر شده ارزیابی کنیم میبینیم که قالب لبه گیری Kirsch کمی بهتر از قالب Sobel عمل میکند .

مدلینگ کردن لبه :

مدل carry یکی از پیشرفته ترین روش های لبه گیری که در آن محاسبات پیشرفته ریاضی نقش به سزایی دارد که توضیح آن را در مقالات بعدی قرار خواهم داد.

تهیه کننده : مجید عابدین زاده شهری

Email : majid.microware@gmail.com

Nrm\NalgaYnTel – Translate [2009-08-16]

WWW.NREC.IR