Name: Nguyễn Văn Quý

Class: 22KTMT1

MSSV: 106220232

**PROJECT: Edge-based Segmentation**

* **Edge-based Segmentation là một phương pháp phân đoạn ảnh dựa vào biên (cạnh) – tức là tìm ra ranh giới giữa các vùng khác nhau trong ảnh. Phương pháp này tìm những điểm có sự thay đổi mạnh mẽ (gradient lớn) về cường độ (intensity), và sử dụng chúng để xác định biên (edges) – từ đó phân chia ảnh thành các vùng riêng biệt.**

*Hình 1: các thuật toán của phương pháp Edge-based Segmentation*

**Phát hiện cạnh sử dụng đạo hàm bậc 1 (Sobel, Prewitt, Robert)**

* **Các bước xử lý trong thuật toán Sobel, Prewitt và Robert về cơ bản là giống nhau, chỉ khác nhau ở mặt nạ lọc (kernel/convolution mask).**
* **Các bước thực hiện:**

**+ B1: Chuyển ảnh sang xám (grayscale) nếu ảnh gốc là RGB.**

**+ B2: Sử dụng 2 mặt nạ tích chập (convolution kernels):**

*Hình 2: mặt nạ tích chập thuật toán sobel ( phương ngang và dọc)*

*Hình 3: mặt nạ tích chập thuật toán Prewitt ( phương ngang và dọc)*

= [1 0] = [ 0 -1]

[0 -1] [1 0]

*Hình 3:* **mặt nạ tích chập thuật toán Robert( phương chéo)**

**+ B3: Tính độ lớn gradient tại mỗi điểm ảnh:**

*Hình 4: công thức độ lớn gradient tại mỗi điểm ảnh*

***+* *B4 : Áp ngưỡng (Thresholding)***

*Chọn ngưỡng T, T = Sum(G(x,y))/mxn.*

*Tại mỗi điểm ảnh : mm*

*Nếu G(x,y)≥T → đánh dấu là biên (thường đặt pixel = 255)*

*Ngược lại → pixel = 0*

***Kết quả:***

* ***Robert***

***Code matlab ( thuật toán robert )***

***A screenshot of a computer code

AI-generated content may be incorrect.***

**Kết quả thu được**

* ***Sobel***

***Code matlab ( thuật toán sobel )***

**Kết quả thu được**

* ***Prewitt***

***Code matlab ( thuật toán Prewitt )***

**Kết quả thu được**

**Nhận xét:**

**Robert: Kết quả hơi mờ và nhiễu nhiều, nhiều cạnh mỏng hoặc không rõ**

**Sobel: Cạnh được phát hiện rõ, đậm nét và ổn định hơn, ít nhiễu**

**Prewitt: Cạnh cũng rõ, nhưng mờ hơn Sobel một chút, độ chi tiết thấp hơn một chút**

**Nhược điểm của 3 thuật toán**

* + **Gradient tính toán độ dốc theo một hướng cụ thể (thường là x và y) → Đôi khi chỉ phát hiện tốt biên theo hướng thẳng mà kém với các hướng xiên hoặc cong**
  + **Không xác định rõ ràng vị trí biên như đạo hàm bậc 2**
  + **Không nhấn mạnh được những chi tiết phức tạp như góc hoặc giao nhau**
  + **Sử dụng 2 phép chập => không tuyến tính**
* **=> Giải pháp: Sử dụng bộ phát hiện cạnh dựa trên đạo hàm bậc 2 (ví dụ: LoG, Canny) để xác định vị trí cạnh chính xác hơn**

**Phát hiện cạnh sử dụng đạo hàm bậc 2 (Laplacian of Gauss)**

*Hình 4:* **Đồ thị biểu diễn hàm Gaussian 1 chiều**

**Gaussian**

**Gradient of Gaussian**

**Laplacian of Gaussian**

**Laplacian nhạy cảm với nhiễu hơn gradient do tính chất của đạo hàm bậc hai → thường cần làm mịn ảnh trước bằng Gauss. Giải thích : Đạo hàm bậc hai khuếch đại nhiễu.**

* + **Công thức và ma trận Laplacian:**

**Công thức**

Các bước thực hiện:

* B1: **Đạo hàm bậc hai bằng công thức sai phân rời rạc**
* **B2: Laplacian 2 chiều**

=====è

* **B3: Các Bước thực hiện LoG**

**+ Cách 1: Bước 1: Làm mịn ảnh**

**Trong đó I(x,y) là ảnh xám, G(x,y) là hàm Gauss, và ∗ là phép tích chập.**

**+ Bước 2: Tính Laplacian của ảnh đã làm trơn**

+ **Cách 1: Bước 1: Tìm Laplacian của hàm Gauss**

**+ Bước 2: Chập nó với ảnh I(x,y)**

***Kết quả:***

**Đối với cách 1:**

Sdasd

*Hình 5:* **Kết quả khi thức hiện LoG theo cách 1**

**Đối với cách 2:**

*Hình 6: Kết quả khi thức hiện LoG theo cách 2*

**Nhận xét:**

* + **Phát hiện biên tổng quát tốt hơn, bao gồm cả biên cong, xiên**
  + **Nhấn mạnh tốt các chi tiết phức tạp như góc, điểm giao nhau – điều mà các phương pháp gradient thường bỏ sót.**
  + **Xác định vị trí biên chính xác hơn nhờ sử dụng đạo hàm bậc hai.**
  + **Đường biên sắc nét, mượt mà hơn so với các biên có phần gắt và dễ rời rạc ở Sobel, Prewitt hoặc Robert.**

**Phát hiện cạnh sử dụng đạo hàm bậc 2 (Canny Egde Detector)**

Các bước thực hiện:

**Bước 1: Làm mịn ảnh với hàm Gauss**

**Bước 2: Tính Gradient magtitude và Gradient direction**

**Bước 3: Lấy Laplacian dọc theo hướng của cạnh đó**

**Bước 4: Sử dụng ngưỡng kép (double thresholding) và phân tích kết nối để phát hiện và liên kết các cạnh**

***Kết quả:***

*Hình 7: Kết quả khi thực hiện Canny*

Nhận xét:

* + **Biên của mũ, khuôn mặt, mắt, vai và tóc được phát hiện rõ ràng và liên tục.**
  + **Các cạnh có độ tương phản thấp hơn (như vùng da, bóng đổ) vẫn hiện ra đủ tốt, không bị mất.**
  + **Các đường biên liên kết tốt, đặc biệt là đường viền mũ và đường nét khuôn mặt.**
* **→ Canny là lựa chọn tối ưu cho việc phát hiện biên chính xác, rõ ràng và ổn định.**

**So sánh Canny và LoG:**

A screenshot of a white page

AI-generated content may be incorrect.