

I. Lý thuyết về Canny Edge Detector

1. Tìm hiểu về các bước của thuật toán Canny:

a. Các bước thực hiện thuật toán

Bước 1: Giảm nhiễu (Noise Reduction)

Các cạnh rất nhạy cảm với nhiễu hình ảnh. Thuật toán sử dụng bộ lọc Gaussian để làm mịn ảnh. Phép toán này thường dùng một mặt nạ (kernel) Gaussian để tích chập với ảnh gốc.

Bước 2: Tính toán Gradient (Gradient Calculation)

Xác định cường độ sáng thay đổi nhanh nhất bằng cách sử dụng các toán tử (như Sobel) để tính đạo hàm theo hướng x(Gx) và hướng y(Gy).

- Độ lớn Gradient: $G = \sqrt{G_x^2 + G_y^2}$
- Hướng Gradient: $\theta = \arctan\left(\frac{G_x}{G_y}\right)$

Bước 3: Non-maximum Suppression (Loại bỏ các điểm không cực đại)

Bước này giúp làm "mỏng" các cạnh. Thuật toán sẽ duyệt qua từng điểm ảnh, kiểm tra xem độ lớn gradient tại điểm đó có phải là cực đại trong hướng của gradient hay không. Nếu không, giá trị đó sẽ bị đưa về 0.

Bước 4: Nguõng kép (Double Thresholding)

Để phân loại các cạnh thực sự, ta dùng hai ngưỡng: High và Low.

- Điểm có gradient > High: Cạnh mạnh (chắc chắn là cạnh).
- Điểm có Low < gradient < High: Cạnh yếu (có thể là cạnh).
- Điểm có gradient < Low: Loại bỏ.

Bước 5: Theo dõi cạnh bằng cách liên kết (Edge Tracking by Hysteresis)

Các cạnh yếu sẽ chỉ được giữ lại nếu chúng kết nối trực tiếp với một cạnh mạnh. Điều này giúp loại bỏ các nhiễu nhỏ có cường độ cao nhưng không tạo thành đường nét.

b. So sánh với các thuật toán khác như Sobel, Laplacian.

Sobel: Đơn giản, nhanh nhưng cạnh thường bị dày và rất nhạy cảm với nhiễu. Không có bước lọc nhiễu tích hợp.

Laplacian: Tìm điểm khôn của đạo hàm bậc hai. Rất nhạy với nhiễu và không xác định được hướng cạnh.

Canny: Phức tạp hơn nhưng cho kết quả vượt trội nhờ khả năng lọc nhiễu và làm mỏng cạnh (chỉ dày 1 pixel).

2. Các tham số của thuật toán và ảnh hưởng của chúng:

Tham số	Ý nghĩa	Ảnh hưởng
Sigma (δ)	Độ lệch chuẩn của bộ lọc Gaussian.	δ lớn giúp lọc nhiễu tốt hơn nhưng làm mờ các cạnh nhỏ; δ nhỏ giữ được chi tiết nhưng dễ bị nhiễu.
Low Threshold	Nguồn dưới.	Nếu quá thấp, ảnh sẽ xuất hiện nhiều "cạnh giả" từ nhiễu.
High Threshold	Nguồn trên.	Nếu quá cao, các cạnh thực sự nhưng mờ nhạt sẽ bị mất (đứt đoạn).

3. Ưu điểm và nhược điểm của Canny edge detector. Các ứng dụng thực tế của Canny edge detector.

a. So sánh với các thuật toán khác về độ chính xác, tốc độ, khả năng xử lý nhiễu

Tiêu chí	Canny Edge Detector	Sobel / Prewitt	Laplacian (LoG)
Độ chính xác	Rất cao: Nhờ bước Non-maximum Suppression giúp cạnh chỉ dày 1 pixel.	Trung bình: Cạnh thường bị dày và nhòe, khó xác định vị trí chính xác.	Cao: Xác định tốt các điểm đạo hàm bằng không.
Khả năng xử lý nhiễu	Tốt nhất: Có tích hợp sẵn bộ lọc Gaussian để làm mịn ảnh trước khi xử lý.	Kém: Rất nhạy cảm với nhiễu vì chỉ dùng đạo hàm bậc nhất đơn giản.	Rất kém: Cực kỳ nhạy cảm với nhiễu do sử dụng đạo hàm bậc hai.
Tốc độ xử lý	Chậm: Do phải trải qua nhiều bước tính toán phức tạp (5 bước).	Rất nhanh: Chỉ sử dụng các mặt nạ (kernel) tích chập đơn giản.	Nhanh: Tính toán đơn giản hơn Canny.
Độ phức tạp	Cao, cần tinh chỉnh nhiều tham số (Sigma, nguồn kép).	Thấp, dễ cài đặt.	Trung bình.

b. Trong lĩnh vực nào Canny được sử dụng phổ biến nhất?

Tự động hóa công nghiệp: Kiểm tra lỗi sản phẩm, đo lường kích thước linh kiện trên dây chuyền.

Giao thông thông minh: Hệ thống nhận diện biển số xe, phát hiện làn đường cho xe tự lái.

Y tế: Phân tích ảnh X-quang, MRI để tách biên các cơ quan hoặc khối u.

Nhận diện đối tượng: Là bước tiền xử lý quan trọng để trích xuất đặc trưng hình dạng vật thể.

c. **Ví dụ cụ thể về các ứng dụng.**

Phát hiện làn đường (Lane Detection): Trong xe tự hành, Canny được dùng để tìm ra các vạch kẻ đường trên mặt lộ. Nhờ khả năng lọc nhiễu, nó loại bỏ được các chi tiết thừa như bóng cây hay vết bẩn trên đường để chỉ giữ lại đường biên của vạch kẻ.

Nhận dạng khuôn mặt/vân tay: Sử dụng Canny để trích xuất các đường nét đặc trưng của khuôn mặt hoặc các đường rãnh trên vân tay, phục vụ cho việc đối chiếu dữ liệu sinh trắc học.

Số hóa bản đồ: Trích xuất ranh giới các tòa nhà, con sông từ ảnh vệ tinh để xây dựng bản đồ số hoặc mô hình 3D đô thị.

III. Câu hỏi mở rộng:

1. Làm thế nào để đánh giá chất lượng của các cạnh được phát hiện bởi Canny?

Để đánh giá một thuật toán phát hiện cạnh "tốt", người ta thường dựa trên 3 tiêu chí cốt lõi mà chính John Canny đã đề xuất:

- **Tỷ lệ lỗi thấp (Low error rate):** Thuật toán phải tìm thấy càng nhiều cạnh thực càng tốt và không được phản hồi với các nhiễu không phải là cạnh.
- **Định vị tốt (Good localization):** Khoảng cách giữa điểm ảnh được đánh dấu là cạnh và tâm thực sự của cạnh đó phải là nhỏ nhất.
- **Phản hồi duy nhất (Single response):** Một cạnh thực sự chỉ được đánh dấu bằng một đường biên duy nhất, không được xuất hiện nhiều đường biên song song cho cùng một cạnh (đây là vai trò của bước *Non-maximum suppression*).
- **Định lượng:** Trong nghiên cứu, người ta có thể dùng chỉ số F-measure hoặc so sánh với Ground Truth (ảnh biên chuẩn do con người vẽ) để tính toán độ chính xác.

2. Có những phương pháp nào khác để cải thiện hiệu suất của Canny?

Sử dụng bộ lọc thích nghi (Adaptive Filtering): Thay vì dùng một giá trị Sigma cố định cho toàn ảnh, ta thay đổi Sigma dựa trên độ nhiễu cục bộ của từng vùng ảnh.

Tự động hóa ngưỡng (Auto-Thresholding): Sử dụng thuật toán **Otsu** để tự động tìm ra ngưỡng High và Low dựa trên biểu đồ phân bố cường độ sáng (histogram) của ảnh thay vì nhập thủ công.

Sử dụng toán tử Gradient khác: Thay thế toán tử Sobel bằng toán tử **Scharr** để có độ chính xác cao hơn khi tính toán đạo hàm ở các vùng có biến thiên nhỏ.

Xử lý đa quy mô (Multi-scale): Chạy Canny trên nhiều mức độ phân giải khác nhau của ảnh rồi kết hợp kết quả lại để thu được cả cạnh chi tiết lẫn cạnh tổng thể.

3. Canny có thể được sử dụng để phát hiện các cạnh trong ảnh màu không? Nếu có, thì như thế nào?

Có, Canny hoàn toàn áp dụng được cho ảnh màu, thường theo 2 cách chính:

- **Cách 1 (Phổ biến):** Chuyển ảnh màu (RGB) sang ảnh xám (Grayscale) trước khi áp dụng Canny. Cách này nhanh nhưng có thể làm mất các cạnh tồn tại giữa hai màu có cùng cường độ sáng (độ chói).
- **Cách 2 (Chính xác hơn):** Áp dụng Canny độc lập trên từng kênh màu (R, G, B hoặc Lab). Sau đó, kết hợp kết quả từ 3 kênh bằng phép toán logic **OR** hoặc tính toán toán tử Gradient vector trong không gian màu đa chiều.

4. Làm thế nào để áp dụng Canny cho các video?

Bản chất của video là một chuỗi các khung hình (frames) liên tiếp. Để áp dụng Canny cho video, ta thực hiện:

- **Xử lý theo khung hình:** Duyệt qua từng frame của video, coi mỗi frame là một bức ảnh tĩnh và áp dụng thuật toán Canny như bình thường.
- **Tối ưu hóa thời gian thực:** Vì video yêu cầu tốc độ cao (thường 30 FPS), ta cần giảm độ phân giải frame hoặc sử dụng GPU (như `cv2.cuda.Canny` trong OpenCV) để xử lý mượt mà.
- **Lọc nhiễu theo thời gian (Temporal Filtering):** Để tránh việc các cạnh bị "nháy" giữa các frame, ta có thể kết hợp thông tin từ frame trước đó để làm mịn các đường biên, giúp chúng ổn định hơn qua thời gian.