Họ tên: Nguyễn Hoàng Việt

MSSV: 18021419

Thuật toán Auto-focus

cho camera điện thoại

Nội dung

[I. Tóm tắt 2](#_Toc121142854)

[II. Giới thiệu 2](#_Toc121142855)

[III. Nội dung thuật toán 3](#_Toc121142856)

[1. Luồng xử lý 3](#_Toc121142857)

[2. Chi tiết các bước xử lý 4](#_Toc121142858)

[IV. Kết quả 5](#_Toc121142859)

[V. Kết luận 9](#_Toc121142860)

[VI. Tham khảo 9](#_Toc121142861)

# Tóm tắt

Một trong các phương pháp auto-focus phổ biến được sử dụng cho camera trên điện thoại thông minh ngày nay là phương pháp lấy nét theo độ tương phản. Phương pháp này sẽ điều chỉnh vị trí của ống kính camera và lấy ảnh liên tục trong một khoảng thời gian ngắn nhất định, thực hiện tính toán để tìm ra vị trí của ống kính mà tại đó có thể lấy được ảnh có độ nét cao nhất, rồi di chuyển ống kính về vị trí tương ứng.

Để đo được độ nét của ảnh, điện thoại sử dụng thuật toán để tính toán độ tương phản của ảnh. Thường thì ảnh có độ tương phản cao sẽ có độ nét cao. Thuật toán được sử dụng trong bài viết này có đầu vào là các ảnh được lấy liên tục tại các vị trí ống kính khác nhau với độ mờ-nét khác nhau. Sau đó chuyển lần lượt các ảnh từ miền RGB sang Grayscale, sử dụng bộ lọc cạnh Sobel và tính trung vị để lọc lấy các giá trị điểm ảnh cần thiết để tìm ra ảnh có độ nét cao nhất. Từ đó giúp ống kính điện thoại có thể di chuyển về vị trí chụp ảnh tối ưu.

# Giới thiệu

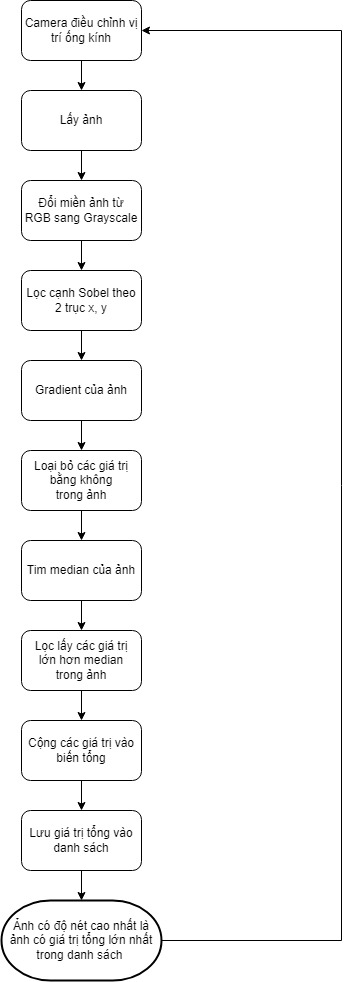
Trong các ứng dụng lấy nét ảnh cho camera, việc xác định vị trí lấy nét tốt nhất thường được thực hiện theo hai cách cơ bản. Phương pháp đầu tiên là AF chủ động, trong đó lấy nét bằng cảm biến hồng ngoại hoặc siêu âm tích hợp vào máy ảnh. Sự phản xạ của sóng hoặc chùm tia được gửi qua các cảm biến trên máy ảnh sẽ được chờ đợi. Sau đó, khi nhận được tín hiệu phản xạ, khoảng cách của đối tượng được tạo ảnh có thể được tìm thấy bằng cách sử dụng mối quan hệ vận tốc-thời gian. Nếu sử dụng ống kính, cảm biến cũng có thể được điều chỉnh ở cấp độ ống kính. Tuy nhiên, AF chủ động đi kèm với chi phí và mức tiêu thụ điện năng cao. Phương pháp thứ hai, AF thụ động, hoàn toàn dựa trên chất lượng của hình ảnh được chụp mà không sử dụng bất kỳ thành phần bổ sung nào. Trong cách tiếp cận thụ động, điều quan trọng là quá trình lấy nét sẽ được thực hiện trên loại hình ảnh nào. Thuật toán lấy nét để đo kích thước trong các bộ phận công nghiệp có thể không mang lại hiệu suất tương tự trong các bức ảnh hàng ngày.

Phương pháp lấy nét ảnh theo độ tương phản cũng thuộc nhóm phương pháp AF thụ động. Nhìn chung, trong một hình ảnh thì mức độ tương phản sẽ cao nhất khi đối tượng được lấy nét tốt (hình ảnh có đường viền sắc nét và màu sắc sống động hơn). Nếu hình ảnh không đủ nét, các màu sẽ có xu hướng mờ vào nhau. Điều này có nghĩa là có mức độ tương phản tổng thể thấp hơn. Trong tính năng tự động lấy nét theo độ tương phản, camera điện thoại sẽ di chuyển ống kính qua lại cho đến khi tìm thấy điểm có độ tương phản tối đa trong khoảng thời gian ngắn nhất định. Ở điểm có độ tương phản tối đa sẽ là điểm giúp hình ảnh được lấy nét tốt nhất.

Thuật toán được sử dụng nhận vào các hình ảnh từ camera khi ống kính đang di chuyển qua lại để tìm điểm chụp thích hợp, chuyển miền ảnh từ RGB sang Grayscale để loại bỏ các thông tin không cần thiết và lấy được các giá trị mức độ sáng của ảnh. Từ đó, các giá trị cạnh được trích xuất sử dụng bộ lọc Sobel theo hai trục x-y. Sau đó, các giá trị cường độ sáng và thông tin cạnh của các hình ảnh tại mỗi vị trí được tính toán bằng phương pháp dựa trên trung vị. Theo kết quả phân tích thống kê, hình ảnh có thông tin cạnh sắc nét nhất, hay nói cách khác là độ tương phản tổng thể cao nhất sẽ được chọn để xác định vị trí tiêu cự tối ưu.

# Nội dung thuật toán

1. Luồng xử lý



1. Chi tiết các bước xử lý

- Bước 1: Lấy ảnh

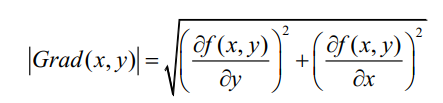
Thuật toán nhận đầu vào là ảnh RGB với độ nét tương ứng với vị trí của ống kính camera đang được điều chỉnh.

- Bước 2: Đổi miền ảnh từ RGB sang Grayscale

Ảnh nhận vào được chuyển từ RGB sang Grayscale giúp lấy được các giá trị mức sáng của ảnh và loại bỏ các thông tin không cần thiết trong quá trình tính toán độ tương phản của ảnh.

- Bước 3: Tính gradient của ảnh

Gradient của ảnh là sự thay đổi giữa các giá trị của điểm ảnh. Sử dụng phương pháp Sobel theo 2 trục x và y rồi tổng hợp chúng lại để nhận được thông tin cạnh của ảnh theo công thức:



Giá trị độ lớn này sẽ giúp phát hiện quá trình chuyển đổi cường độ cao sẽ xảy ra ở các cạnh của các bộ phận trong ảnh.

- Bước 4: Loại bỏ các giá trị bằng không trong ảnh

Sau khi đã có được thông tin cạnh của ảnh dưới dạng mảng giá trị hai chiều, ta đổi mảng đó thành dạng mảng một chiều, sau đó các giá trị bằng không được loại bỏ khỏi mảng để từ đó tìm ra giá trị trung vị của mảng.

- Bước 5: Loại bỏ các giá trị thấp hơn giá trị trung vị của mảng

Sử dụng phép tính trung vị để xem xét giá trị của các phần tử trong mảng, lọc được các thông tin cạnh có ý nghĩa. Các giá trị gradient thấp, không chứa thông tin cạnh quan trọng, được xóa khỏi mảng. Những giá trị này là giá trị không cần thiết. Do đó, các giá trị nhỏ hơn giá trị trung vị sẽ không được đưa vào tính toán.

- Bước 6: Lấy giá trị tổng của mảng

Tất cả các giá trị trong mảng sẽ được tổng cộng lại vào một giá trị tổng chung. Giá trị này giúp nhận biết được độ tương phản hay nói cách khác là độ sắc nét của ảnh. Giá trị tổng càng cao nghĩa là ảnh có độ tương phản càng cao, độ sắc nét càng cao.

- Bước 7: Lưu giá trị tổng của mảng vào danh sách

Giá trị tổng trên được lưu vào một danh sách. Mỗi giá trị trong danh sách sẽ thể hiện độ sắc nét của bức ảnh tương ứng với khoảng cách của tiêu cự ống kính khi ống kính được điều chỉnh di chuyển qua lại. Như vậy, với mỗi lần ống kính di chuyển thì thuật toán sẽ được lặp lại và cho ra danh sách các giá trị cho thấy độ tương phản của ảnh.

- Bước 8: Chọn ra vị trí tiêu cự tối ưu

Sau khi camera điều chỉnh ống kính qua lại trong khoảng thời gian nhất định, thuật toán được lặp lại theo số lần di chuyển của ống kính, ta có được một danh sách kết quả cuối cùng. Giá trị lớn nhất trong danh sách được chỉ định là giá trị mà tại vị trí tiêu cự đó ảnh đạt được độ sắc nét cao nhất. Từ đó camera có thể di chuyển vị trí ống kính tới điểm tương ứng.

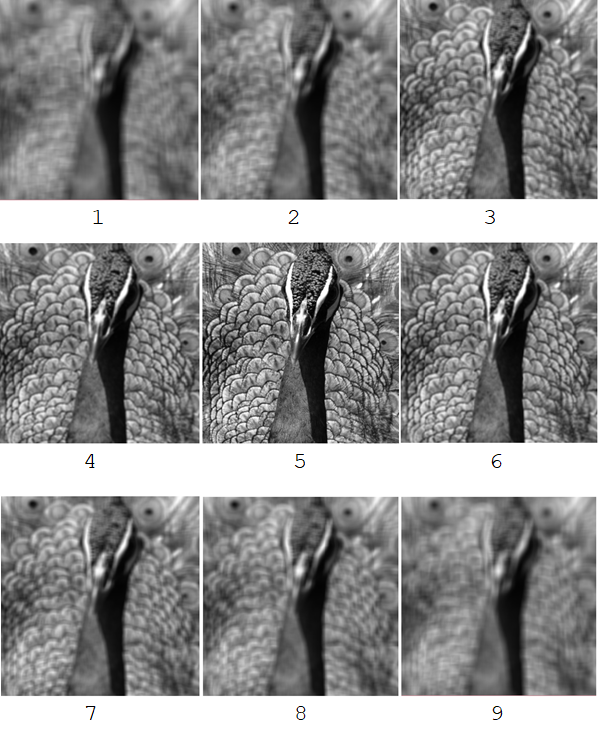
# Kết quả

Thử nghiệm lấy ví dụ sử dụng một ảnh như sau:



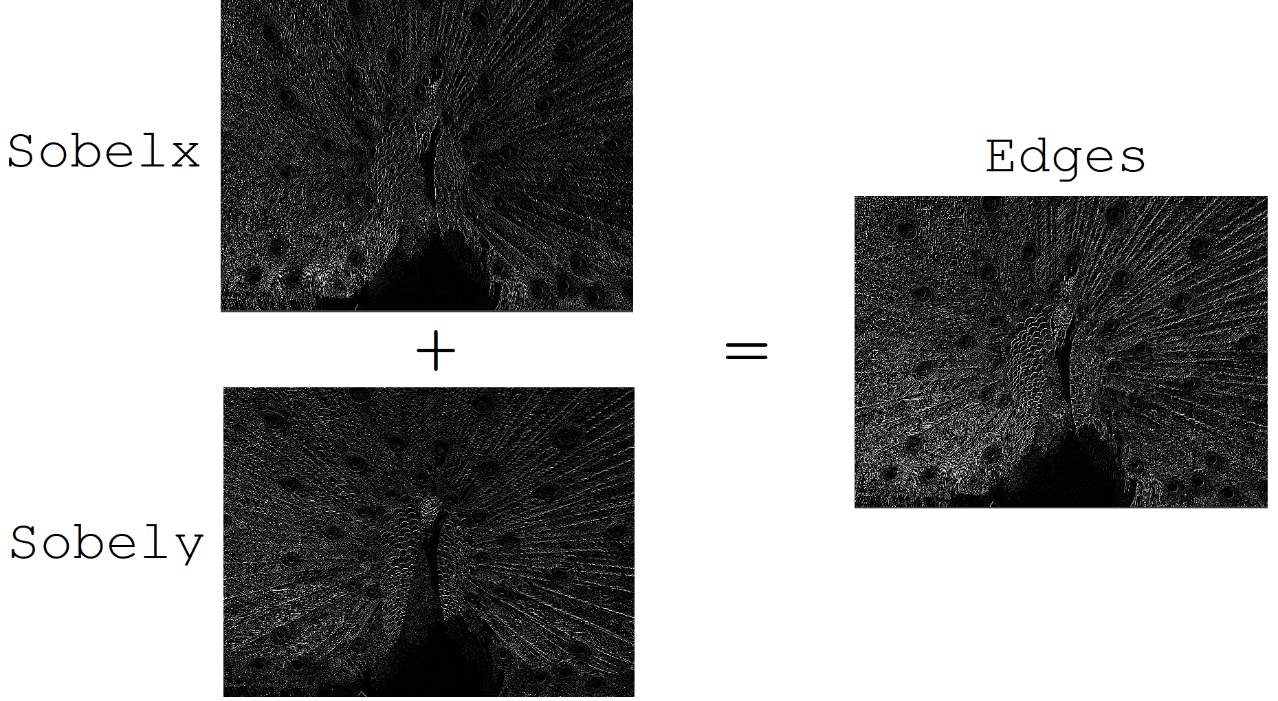
Chương trình sẽ nhận vào một số ảnh, cụ thể ở thử nghiệm này là 9 ảnh như trên với độ sắc nét khác nhau. Ảnh gốc trên trước khi đưa vào chương trình được làm mờ với độ mờ khác nhau để mô phỏng quá trình camera di chuyển qua lại ống kính và cho ra số ảnh tương tự. Bước đầu chương trình sẽ đọc từng ảnh vào và chuyển miền ảnh từ RGB sang miền màu xám.

Dưới đây là hình ảnh minh họa(đã được cắt bớt) của 9 ảnh được đưa vào thuật toán:



Nhìn bằng mắt thường ta có thể thấy ảnh số 5 có độ nét cao nhất, còn ảnh có độ nét thấp nhất có thể là ảnh số 1 hoặc ảnh số 9, mặc dù trong khi tính toán ta chỉ cần quan tâm tới ảnh nét nhất.

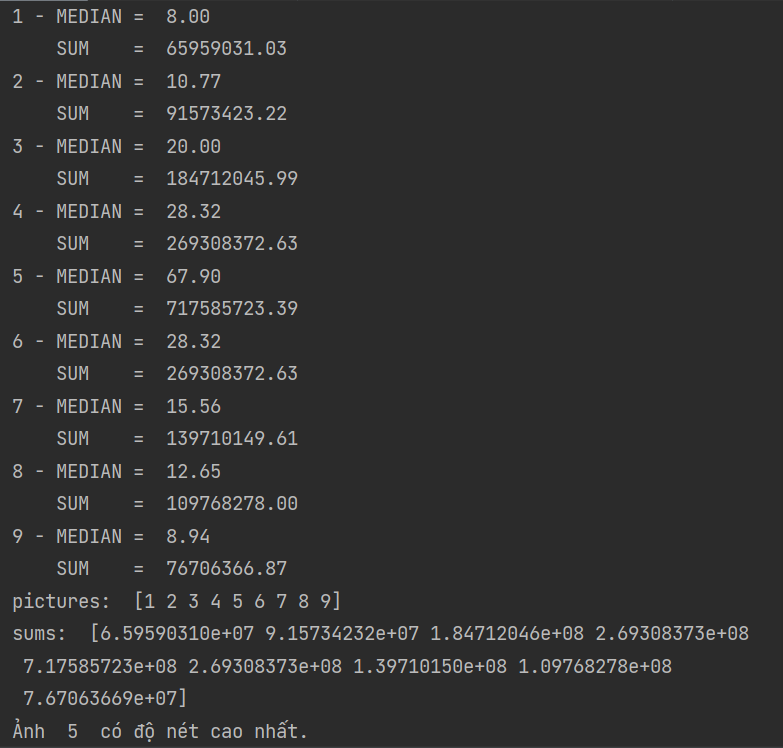
Lấy ví dụ ảnh số 5 sau khi đi qua bộ lọc Sobel như sau:

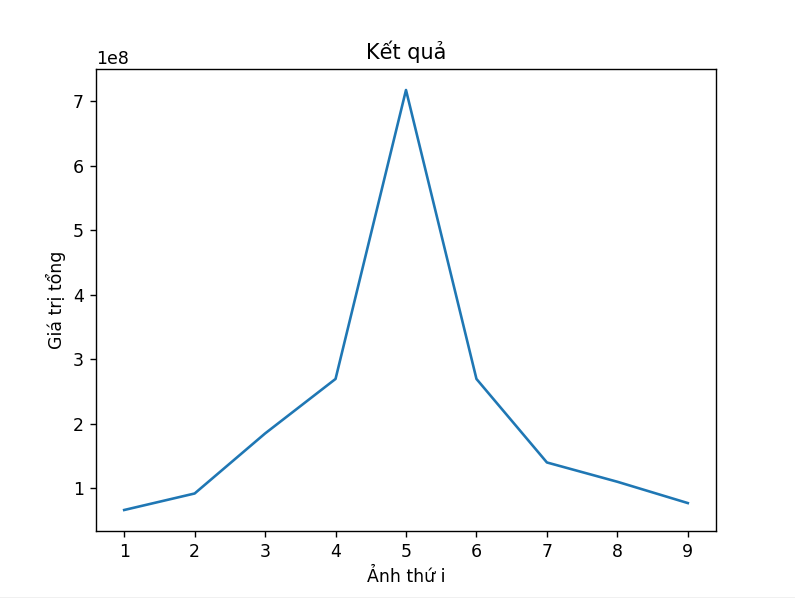


Áp dụng Sobel theo trục x lên ảnh xám sẽ lọc được các cạnh dọc trong ảnh, tương tự theo trục y sẽ lấy được các cạnh nằm ngang. Tổng của hai ảnh là thông tin về tất cả các cạnh của ảnh được tìm thấy.

Ta cũng có thể phát hiện được ở phần xung quanh thân dưới của con công do có bóng râm che phủ nên bộ lọc rất khó để tìm thấy các cạnh trong vùng đó. Đó cũng chính là một trong nhưng khuyết điểm của phương pháp lấy nét theo độ tương phản. Trong điều kiện ánh sáng kém thì độ tương phản của bức ảnh cũng sẽ kém theo dẫn đến sự hoạt động không hiệu quả của thuật toán.

Sau khi lấy chương trình lấy được thông tin cạnh của ảnh, chạy qua các bước tính toán như đã nêu trên thì cho ra kết quả như sau:





Chương trình đưa ra được kết quả chính xác như dự đoán với ảnh số 5 là ảnh có độ nét cao nhất và số liệu so sánh độ tương phản/độ nét của các ảnh khác.

Các thử nghiệm với ảnh khác và việc tráo đổi thứ tự sắp xếp độ nét của các ảnh được đưa vào cũng cho ra kết quả với độ chính xác cao chỉ trừ trường hợp ảnh đầu vào ở điều kiện thiếu ánh sáng khiến cho ảnh có độ tương phản thấp. Tuy nhiên, điểm yếu này cũng có thể được khắc phục sử dụng nguồn ánh sáng bên ngoài như đèn flash camera và/hoặc sử dụng thuật toán cân bằng histogram giúp tăng độ tương phản của ảnh.

Ngoài ra, thử nghiệm trên cũng cho thấy rằng dù bức ảnh thứ 5 có độ nét cao nhất nhưng chương trình vẫn tiếp tục chạy và tính toán hết độ nét của tất cả các ảnh. Điều đó là do camera điện thoại không biết khi nào nó đạt đến điểm tương phản tối đa. Vì lý do này, nó thường sẽ cần phải đi qua tiêu điểm, để nhận ra rằng đó là điểm cho ra độ nét cao nhất, trước khi quay lại điểm đó. Đôi khi ta cũng có thể nhận thấy điều này khi chụp ảnh trên điện thoại thông minh của mình, hình ảnh có thể sẽ trở nên mờ hơn trước khi trở lại rõ hơn.

# Kết luận

Một trong những phương pháp phổ biến trong việc lấy nét ảnh cho camera điện thoại là phương pháp lấy nét dựa theo độ tương phản của ảnh. Thuật toán được viết và thử nghiệm áp dụng phương pháp trên qua việc phân tích thông tin cạnh và giá trị trung vị của ảnh để đưa ra kết quả về độ nét của ảnh, giúp ống kính camera chọn được vị trí chụp ảnh tối ưu.

Theo như kết quả được đưa ra từ các thử nghiệm với thuật toán trên, ta có thể nhận xét rằng thuật toán có độ chính xác cao trong điều kiện chụp ảnh ánh sáng bình thường. Với điều kiện ánh sáng kém thì có thể được hỗ trợ sử dụng thêm đèn ngoài như đèn flash của camera hay thuật toán để tăng độ tương phản của ảnh như cân bằng histogram. Ngoài ra thì tốc độ lấy nét của phương pháp này cũng không quá nhanh do phải tính toán độ nét tại nhiều vị trí ống kính khác nhau để so sánh tìm điểm tốt nhất, nên ta khó có thể chụp được ảnh nét ngay lập tức và ảnh nét của vật đang chuyển động nhanh. Bù lại hệ thống có thể hoạt động đủ tốt trong các trường hợp chụp ảnh thông thường mà không đòi hỏi phần cứng bổ trợ, giúp tiết kiệm chi phí và tiện lợi sử dụng.

# Tham khảo

* “Edge Detection Based Autofocus Algorithm to Detect Accurate Camera Working Distance" Poyraz A.G., Kacmaz M., ISITES'2022.
* Israni, Dippal, Sandip Patel, and Arpita Shah. "Comparison of different techniques of camera autofocusing." Proceedings of First International Conference on Information and Communication Technology for Intelligent Systems: Volume 1. Springer, Cham, 2016.

Code: <https://github.com/NgHoangViet/Contrast-Detection-Autofocus>