

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI BÁO CÁO GIỮA KỲ MÔN
PHƯƠNG PHÁP TÍNH**

PHÂN TÍCH SỐ VÀ XÂY DỰNG ỨNG DỤNG PHÂN ĐOẠN HÌNH ẢNH

Giảng viên hướng dẫn: **CÔ VÕ HOÀNG ANH**

Sinh viên thực hiện: **TÔ VĨNH KHANG - 51800408**

Lớp : 18050203

Khoá : 22

Nhóm : 02

Tổ : 02

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2020

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI BÁO CÁO GIỮA KỲ MÔN
PHƯƠNG PHÁP TÍNH**

PHÂN TÍCH SỐ VÀ XÂY DỰNG ỨNG DỤNG PHÂN ĐOẠN HÌNH ẢNH

Giảng viên hướng dẫn: **CÔ VÕ HOÀNG ANH**

Sinh viên thực hiện: **TÔ VĨNH KHANG - 51800408**

Lớp : 18050203

Khoá : 22

Nhóm : 02

Tổ : 02

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2020

LỜI CẢM ƠN

Em xin chân thành cảm ơn Khoa Công nghệ thông tin và Trường Đại học Tôn Đức Thắng đã tạo điều kiện cho chúng em được học tập trong suốt thời gian qua. Chân thành cảm ơn Cô Võ Hoàng Anh đã hướng dẫn trong quá trình học tập môn Thực hành Phương Pháp Tính. Giúp em có thêm kiến thức lập trình bằng ngôn ngữ MATLAB về các phương pháp nội suy xấp xỉ hàm, tính sai số tương đối,..., hiểu biết khái quát hơn những ứng dụng thú vị và thực tế của môn học này.

BÀI BÁO CÁO GIỮA KỲ NÀY ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Em xin cam đoan đây là sản phẩm của riêng em được sự hướng dẫn của cô Võ Hoàng Anh. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính em thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong bài báo cáo này còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào em xin hoàn toàn chịu trách nhiệm về nội dung bài báo cáo của mình. Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do chung em gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày 07 tháng 05 năm 2020

Tác giả

(ký tên và ghi rõ họ tên)

Tô Vĩnh Khang

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

Phần xác nhận của GV hướng dẫn

Tp. Hồ Chí Minh, ngày tháng năm
(ký và ghi họ tên)

Phần đánh giá của GV chấm bài

Tp. Hồ Chí Minh, ngày tháng năm
(ký và ghi họ tên)

TÓM TẮT

Phân tích ảnh số là một lĩnh vực đang rất được quan tâm. Đã có rất nhiều phương pháp trong xử lý phân đoạn ảnh hiện nay. Và bài báo cáo này tập trung về các phương pháp nội suy , xấp xỉ hàm và quá trình để xây dựng một ứng dụng thực tế cho phân vùng hình ảnh. Ứng dụng được hiện thực bằng cách minh họa việc một máy tính có thể thu thập hình ảnh từ bên ngoài vào và phân tích xử lý dữ liệu từ hình ảnh đó.

MỤC LỤC

MỤC LỤC.....	1
DANH MỤC CÁC HÌNH.....	2
CHƯƠNG 1 – GIỚI THIỆU.....	3
1.1 Phân đoạn ảnh.....	3
1.2 Một số phương pháp phân đoạn ảnh hiện nay.....	3
CHƯƠNG 2 – PHÂN ĐOẠN ẢNH DỰA TRÊN NGƯỠNG.....	4
2.1 Khái quát.....	4
2.2 Phương pháp thực hiện.....	4
CHƯƠNG 3 - THỰC NGHIỆM BẰNG PHẦN MỀM MATLAB.....	5
3.1 Đọc hình ảnh đầu vào và chuyển sang thang độ xám.....	5
3.2 Xấp xỉ hàm và so sánh các sai số tương đối.....	6
3.3 Phân ngưỡng.....	11
3.4 Xuất hình ảnh.....	12
CHƯƠNG 4 - TỔNG KẾT.....	13
4.1 Kết quả đạt được.....	13
4.2 Một số ứng dụng trong phân đoạn ảnh hiện nay.....	13
TÀI LIỆU THAM KHẢO.....	14

DANH MỤC CÁC HÌNH

Hình 3.1 Hình ảnh dưới dạng RGB khi chưa phân đoạn ảnh.....	5
Hình 3.2 Hình ảnh sau khi đã được chuyển đổi sang thang độ xám.....	6
Hình 3.3 Biểu đồ so sánh kết quả hàm polyfit và hàm lagrange.....	8
Hình 3.4 Kết quả trung bình của các sai số tương đối ứng với bậc 3,4,5,6.....	9
Hình 3.5 Biểu đồ so sánh chi tiết về kết quả từng bậc của hàm polyfit.....	10
Hình 3.6 Hình ảnh sau khi đã phân đoạn thành hình ảnh nhị phân.....	12
Hình 4.1 Một số ứng dụng trong việc phân đoạn ảnh hiện nay.....	13

CHƯƠNG 1 – GIỚI THIỆU

1.1 Phân đoạn ảnh

Là quá trình phân tích một tấm ảnh thành nhiều phần để đơn giản hóa hay thay đổi biểu diễn của một tấm ảnh để dễ dàng phân tích. Nó được sử dụng để xác định vị trí các đối tượng, đường biên. Việc phân đoạn ảnh còn là một quá trình gán nhãn cho mỗi điểm ảnh trong một tấm ảnh, các điểm ảnh trong cùng một nhãn sẽ có những đặc tính giống nhau về màu sắc, cường độ hay kết cấu của ảnh. Kết quả của việc phân đoạn ảnh là tập hợp các phân đoạn (có thể là toàn bộ tấm ảnh hoặc tập hợp các đường biên từ tấm ảnh). Các điểm ảnh trong cùng một vùng có đặc tính tương tự nhau về màu sắc, cường độ hoặc kết cấu. Các vùng lân cận thì khác nhau đáng kể về các đặc trưng được nêu.

- *Các giá trị mức xám trong một khoảng (ngưỡng)*
- *Khuynh độ của các giá trị mức xám trong một khoảng (đường biên)*
- *Phân bố thống kê như nhau (kết cấu bề mặt)*

Sau khi áp dụng các thuật toán, ảnh sẽ trở thành ảnh nhị phân. Vùng trên ảnh nhị phân là chuỗi các giá trị 0 và 1.

1.2 Một số phương pháp phân đoạn ảnh hiện nay

- Phân đoạn ảnh dựa trên ngưỡng.
- Phân đoạn ảnh dựa trên đường biên.
- Phân đoạn ảnh dựa trên các miền.
- Phân đoạn dựa trên chuyển động.

CHƯƠNG 2 – PHÂN ĐOẠN ẢNH DỰA TRÊN NGUỖNG

2.1 Khái quát

Các điểm trong ảnh được so sánh với một hằng số định trước gọi là ngưỡng. Kỹ thuật này đặt ngưỡng để hiển thị 2 tông màu trắng đen. Giá trị của ngưỡng sẽ quyết định điểm có được hiển thị hay không. Do vậy ảnh kết quả sẽ mất đi một số chi tiết. Như vậy nghĩa là các điểm ảnh có giá trị lớn hơn hằng số θ sẽ là 1, ngược lại là 0.

2.2 Phương pháp thực hiện

Giả sử tấm ảnh được cung cấp tùy ý H như một ma trận đại diện cho cường độ. Tính tần số của từng cường độ duy nhất trong ma trận H . Các giá trị x, y sử dụng để tìm xấp xỉ hàm đa thức $g(x) = a_0 + a_1x + \dots + a_nx^n$ bằng cách sử dụng các phương pháp nội suy.

- Tính vector nội suy: $y_i^{\wedge} = g(x_i)$
- Tính giá trị đạo hàm thứ hai của vector nội suy được giảm thiểu:

$$j = \arg \min_i \{(|y_i^{\wedge}'|)\}$$

- Tìm giá trị cường độ tối đa L .
- Tính giá trị ngưỡng θ :

$$\theta = \frac{j}{L-1}$$

- Phân đoạn ảnh thành dạng nhị phân:

$$b(x) = \begin{cases} 1 & x > \theta \\ 0 & otherwise \end{cases}$$

- Xuất ma trận nhị phân $b(x)$ ta được ảnh đã phân đoạn.

CHƯƠNG 3 - THỰC NGHIỆM BẰNG PHẦN MỀM MATLAB



Hình 3.1 Hình ảnh dưới dạng RGB khi chưa phân đoạn ảnh

3.1 Đọc hình ảnh đầu vào và chuyển sang thang độ xám

```
% Read image
pic=imread('PictureRGB.png');
% Gray scale
pic=rgb2gray(pic);
% Show image
figure(1)
imshow(pic)
```

Hàm **imread** dùng để đọc file hình ảnh đầu vào mang tên ‘PictureRGB.png’

Sau đó chuyển sang thang độ xám bằng hàm **rgb2gray**. Đối với hình ảnh uint8, đầu ra có thể có tới 256 giá trị độ xám nhất trong ma trận.



Hình 3.2 Hình ảnh sau khi đã được chuyển đổi sang thang độ xám

```
% Find x,y
[y,x] = imhist(pic);
x=x'
y=y'
```

Sử dụng hàm **imhist** để phân tích tấm ảnh thành biểu đồ dữ liệu. Lúc này ta có thể dễ dàng lấy được 2 giá trị x và y. Đối với hình này, ta sẽ thu được 256 giá trị độc nhất.

3.2 Xấp xỉ hàm và so sánh các sai số tương đối

Có rất nhiều phương pháp xấp xỉ hàm như phương pháp nội suy đa thức trực tiếp, Lagrange, Newton, Spline... Ở bài báo cáo này, hàm **polyfit** có sẵn của MATLAB và hàm **lagrange** do em xây dựng sẽ được sử dụng cũng như để so sánh chúng.

Sau khi đã có dữ liệu x,y từ tấm ảnh. Tiến hành đoạn code sau để tính toán xấp xỉ hàm.

Xấp xỉ hàm bằng phương pháp dùng hàm có sẵn polyfit (bậc 3):

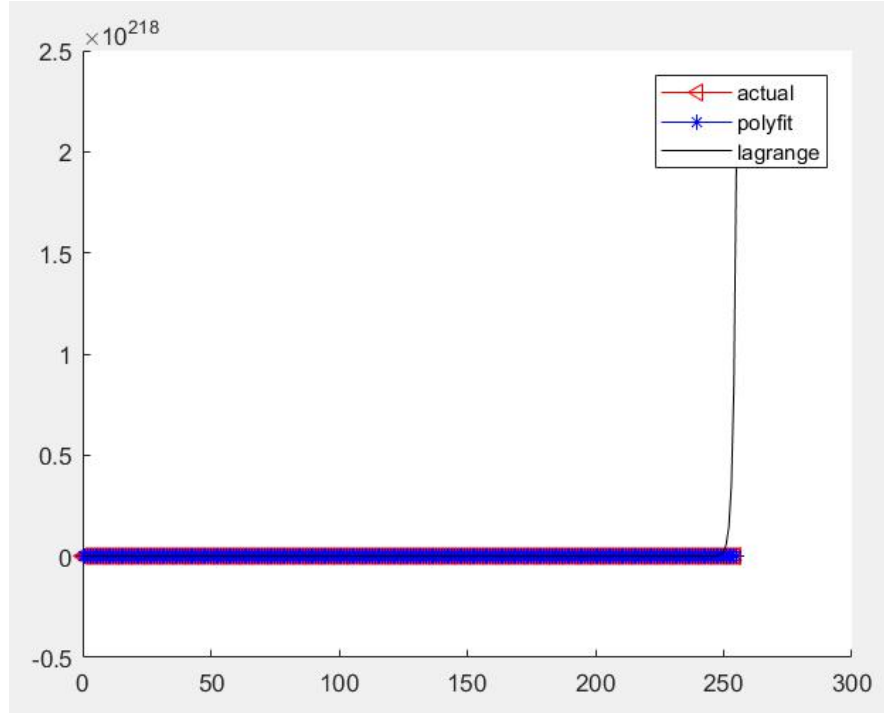
```
% Polynomial function f(x) in the case of using the
built-in function of MATLAB
p=polyfit(x,y,3);
fx1=polyval(p,x)
```

Hàm lagrange được triển khai như sau:

```
function [l,P]=lagrange(x,y)
    l=0;
    n=length(x);
    for i=1:n
        p=1;
        for j=1:n
            if (j~=i)
                c = poly(x(j))/(x(i)-x(j));
                p = conv(p,c);
            end
        end
        P = p*y(i);
        l = l + P;
    end
end
```

Xấp xỉ hàm bằng phương pháp lagrange:

```
% Polynomial function f(x) in the case of using my
function: lagrange
l=lagrange(x,y);
fx2=polyval(l,x)
```



Hình 3.3 Biểu đồ so sánh kết quả hàm polyfit và hàm lagrange

Qua Hình 3.4 có thể thấy hàm polyfit có sẵn của MATLAB có sự vượt trội hơn và chính xác hơn. Do đó, em chọn hàm này để thực hiện cho các phần tiếp theo.

Sau khi đã xác định dùng hàm polyfit, em tiến hành chọn bậc của nó. Đối với yêu cầu đề bài, thay lần lượt bậc $n = 3, 4, 5, 6$ vào hàm polyfit và tính được các sai số tương đối.

```
% Relative Error (n=3)
polywithN3=polyfit(x,y,3);
resultN3=polyval(polywithN3,x);
RelativeErrorN3=abs(y-resultN3/y)
% Relative Error (n=4)
polywithN4=polyfit(x,y,4);
resultN4=polyval(polywithN4,x);
RelativeErrorN4=abs(y-resultN4/y)
% Relative Error (n=5)
polywithN5=polyfit(x,y,5);
resultN5=polyval(polywithN5,x);
RelativeErrorN5=abs(y-resultN5/y)
```

```
% Relative Error (n=6)
polywithN6=polyfit(x,y,6);
resultN6=polyval(polywithN6,x);
RelativeErrorN6=abs(y-resultN6/y)
```

Với: RelativeErrorN3, RelativeErrorN4, RelativeErrorN5, RelativeErrorN6 lần lượt là các mảng chứa giá trị sai số tương đối tương ứng với với các bậc 3,4,5,6. Tiếp theo, tính giá trị trung bình của các sai số tương đối thông qua đoạn hàm được viết như sau:

```
function Result=avgRelativeError(RelativeErrorArray)
    avgError=0;
    n=length(RelativeErrorArray);
    for i=1:n
        avgError=avgError+RelativeErrorArray(i);
    end
    %Calculate the average error
    Result=(avgError/n);
end
```

Xuất kết quả sai số tương đối trung bình:

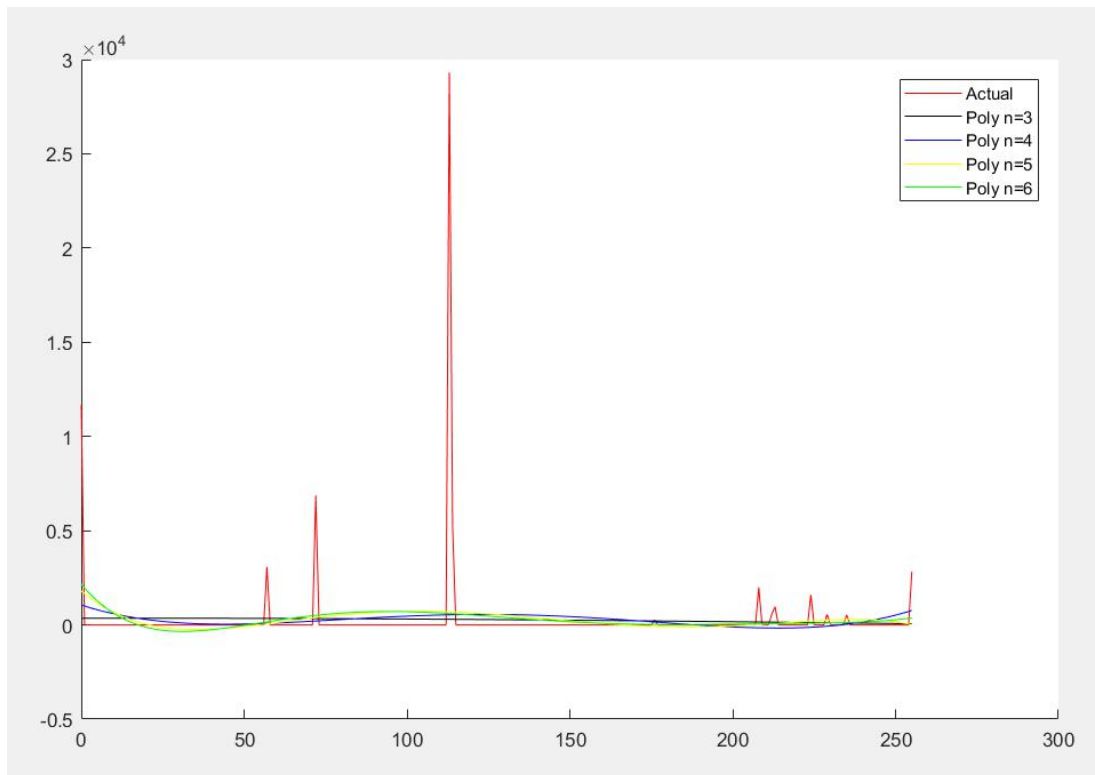
```
% Comparison of the relative errors
% Calculate the average relative error
disp(['Average relative error (degree n=3) : '
num2str(avgRelativeError(RelativeErrorN3))])
disp(['Average relative error (degree n=4) : '
num2str(avgRelativeError(RelativeErrorN4))])
disp(['Average relative error (degree n=5) : '
num2str(avgRelativeError(RelativeErrorN5))])
disp(['Average relative error (degree n=6 ): '
num2str(avgRelativeError(RelativeErrorN6))])
```

```
Average relative error (degree n=3) : 256.0153
Average relative error (degree n=4) : 256.0276
Average relative error (degree n=5) : 256.0395
Average relative error (degree n=6 ) : 256.042
```

Hình 3.4 Kết quả trung bình của các sai số tương đối ứng với bậc 3,4,5,6

Vẽ biểu đồ để kiểm tra , so sánh sự chênh lệch rõ ràng hơn của kết quả vector nội suy y^{\wedge} thu được từ $g(x)$ với y gốc thu được từ tấm ảnh. Dựa vào kết quả thu được có thể thấy bậc $n=3$ là thích hợp để sử dụng cho các bước tiếp theo do sai số trung bình thấp hơn các bậc còn lại.

```
% plot
figure(2)
hold on
plot(x,y, '-r')
plot(x,resultN3, 'k')
plot(x,resultN4, 'b')
plot(x,resultN5, 'y')
plot(x,resultN6, 'g')
legend('Actual', 'Poly n=3', 'Poly n=4', 'Poly
n=5', 'Poly n=6')
hold off
```



Hình 3.5 Biểu đồ so sánh chi tiết về kết quả từng bậc của hàm polyfit

3.3 Phân ngưỡng

Trước khi phân ngưỡng cần tìm giá trị đạo hàm thứ hai của vector nội suy y^{\wedge} được giảm thiểu thông qua đoạn code sau:

```
% y'' of y^ vector from g(x)
pd=abs(diff(polyval(polywithN3,x)));
pd=diff(pd);
% Find the index
IndexJ=find(pd == min(pd(:)))
```

=> Kết quả: $IndexJ = 1$

Tìm giá trị cường độ tối đa:

```
% Find maximum intensity
L = max(pic(:))
```

=> Kết quả: $L = 255$

Tính giá trị ngưỡng:

```
% Find Theta
Theta=IndexJ/L-1
```

=> Kết quả: $Theta = 0$

Phân đoạn ảnh thành dạng nhị phân:

```
% Matrix Binary
for i=1:length(pic)
    for j=1:length(pic)
        if (pic(i,j)/length(x))>Theta
            pic(i,j)=1;
        else
            pic(i,j)=0;
        end
    end
end
pic=pic>Theta;
```

=> Kết quả: Thu được ma trận nhị phân

3.4 Xuất hình ảnh

Sau khi phân ngưỡng và thu được ma trận nhị phân. Xuất hình ảnh để kiểm tra.

```
% Display Binary image  
figure(3)  
imshow(pic)
```



Hình 3.6 Hình ảnh sau khi đã phân đoạn thành hình ảnh nhị phân

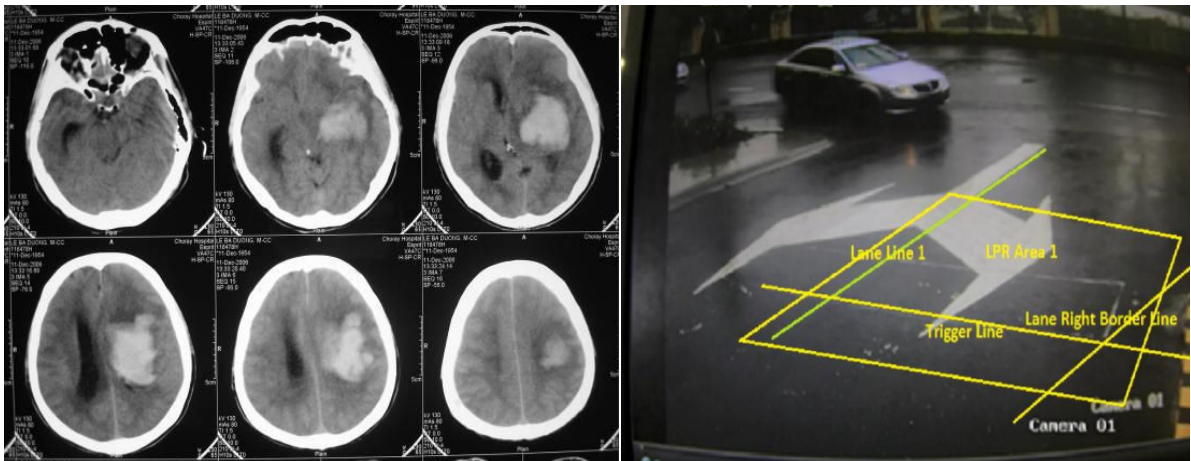
CHƯƠNG 4 - TỔNG KẾT

4.1 Kết quả đạt được

- Hiểu được quá trình nội suy xấp xỉ hàm để phân đoạn ảnh cũng như các lý thuyết xử lý phân đoạn ảnh.
- Thu thập được dữ liệu hình ảnh và xây dựng được ứng dụng phân đoạn ảnh bằng phương pháp nội suy xấp xỉ hàm.
- Tìm hiểu thêm được các kỹ thuật khác hiện nay.

4.2 Một số ứng dụng trong phân đoạn ảnh hiện nay

- Hình ảnh trong y tế (chụp CT, X-Quang để chuẩn đoán bệnh,..)
- Nhận diện đối tượng (khuôn mặt, cử chỉ tay, đếm số lượng người,..)
- Camera an ninh (cảnh báo chuyển động,..)
- Giám sát giao thông (thống kê lưu lượng xe , bãi đỗ,..)



Hình 4.1 Một số ứng dụng trong việc phân đoạn ảnh hiện nay

TÀI LIỆU THAM KHẢO

Tài liệu tiếng Anh

[1] Chen, J., Shao, H., & Hu, C. (2018). Image Segmentation Based on Mathematical Morphological Operator. Colorimetry and Image Processing.

Website

[2] [https://en.wikipedia.org/wiki/Thresholding_\(image_processing\)](https://en.wikipedia.org/wiki/Thresholding_(image_processing))

[3] <http://vuontoanblog.blogspot.com/2012/10/polynomial-interpolation-lagrange.html>

[4] <https://www.mathworks.com/help/images/ref/imshow.html>