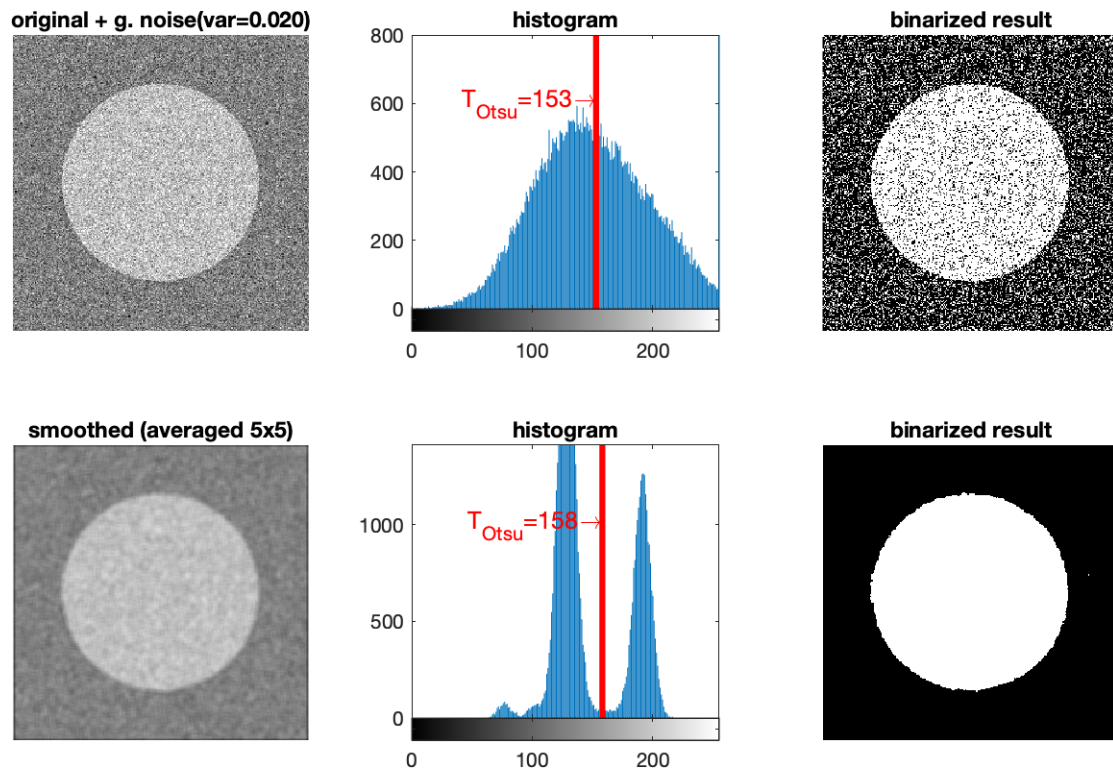


## Zohľadnenie polohy pri prahovaní

Globálne prahovanie nezohľadňuje polohu bodov. Je viacero spôsobov ako polohu bodov zohľadniť a kvalitu segmentácie pomocou prahovania vylepšiť.

Prvým spôsobom je použitie vyhladenie obrázku. Napríklad keď máme zašumený obraz, vyhladenie pomôže potlačiť šum a znovu získať maximá v histograme obrázku. Je dôležité si uvedomiť, že ak vyhladzujeme napr. priemerujúcim filtrom s jadrom 5x5 bodov, tak vlastne využívame okolie 5x5 bodov na vyhladenie. Na druhú stranu vyhladenie nám môže spôsobiť deformáciu hrán objektu, takže treba hľadať kompromis medzi silou vyhladenia a spôsobenou deformáciou. Príklad úspešného vylepšenia globálneho prahovania vyhladením obrazu je na Obr. 1.

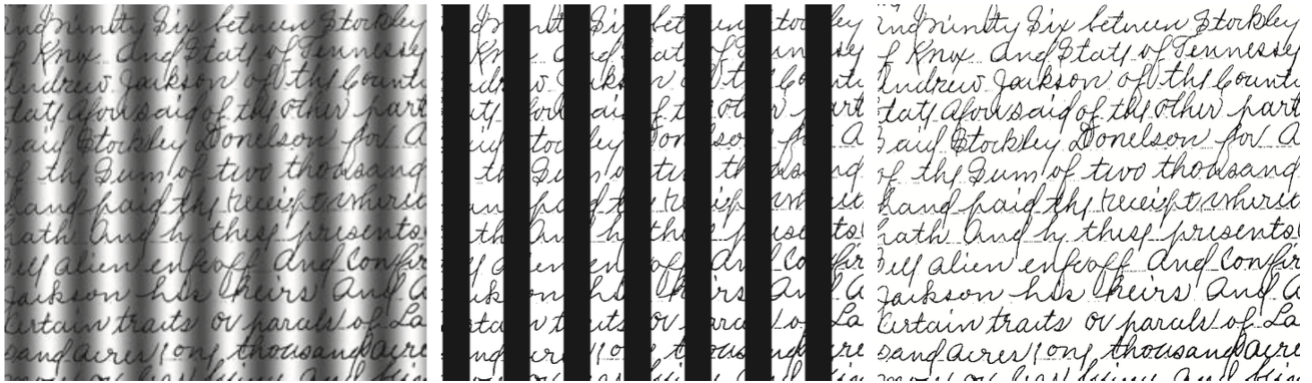


Obr. 1 Príklad úspešného vylepšenia Otsu prahovania vyhladením (V hornom rade obrázkov je vľavo pôvodný zašumený obraz, v strede jeho histogram a v pravo neúspešné Otsu prahovanie. V dolnom rade je vľavo pôvodný zašumený obraz vyhladený priemerujúcim filtrom 5x5, v strede jeho histogram a vpravo úspešné Otsu prahovanie)

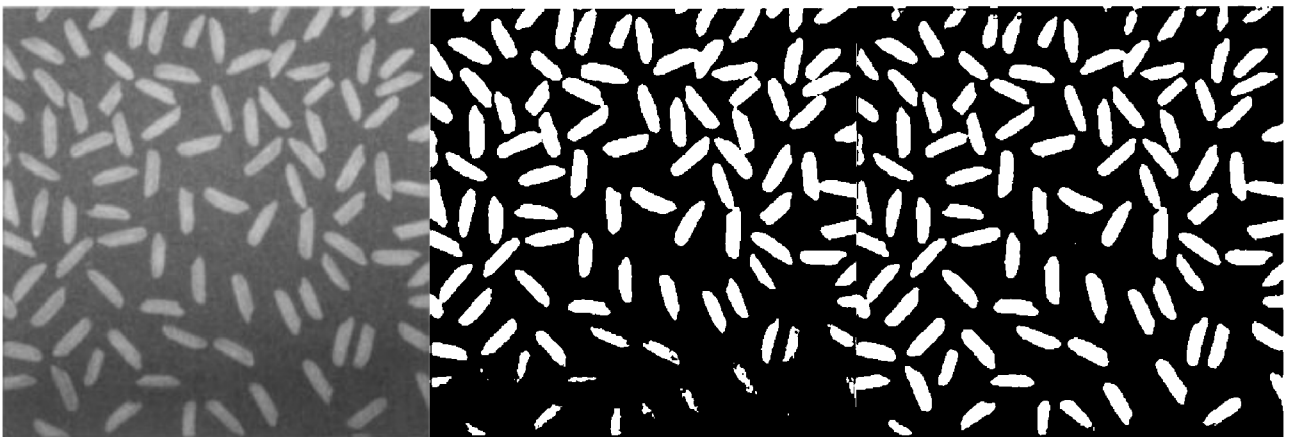
V prípade ak objekt v popredí obsahuje málo bodov oproti bodom v pozadí (t.j. v rámci histogramu je jeho príspevok zanedbateľný) tak vyhladzovanie nemusí pomôcť (prípadne môže viac uškodiť). Výraznou pomôckou môže byť, že pri výpočte histogramu použijeme masku a zoberieme do úvahy iba hodnoty z primerane veľkého okolia objektu v popredí. Masku môžeme získať manuálne či (semi)automaticky. Prípadne zoberieme do úvahy iba body z okolia rozhrania medzi objektom a pozadím (takúto masku nám vie poskytnúť napr. Laplacián).

Šum v obraze, malý objekt v popredí sú len dva z veľkého množstva situácií, kde by pomohlo zohľadnenie okolia pri uskutočnení prahovania. Na Obr. 2 je zobrazená situácia s tieňmi v tvare pruhov po celom obraze, kde je vyslovene žiadúce použiť prahovaciu metódu ktorá zohľadňuje lokálne vlastnosti obrazu. Najčastejšie sa zohľadňuje priemer a rozptyl [1]. V MATLAB prostredí je funkcia **adapt-thresh**, ktorá volí prah adaptívne na základe lokálneho priemeru (mean). Príklad využitia tejto funkcie

pri lokálnom prahovaní obrazu s nerovnakým osvetlením pozadia je uvedený na Obr. 3. Vidíme, že v dolnej časti globálne Otsu prahovanie zlyháva, pri lokálnom sa darí udržiavať štandardnú kvalitu.



Obr. 2 Výhoda lokálneho prístupu pri prahovaní. Vľavo originálny obraz, v strede globálne Otsu prahovanie, vpravo prahovanie pomocou lokálneho priemeru [1]



Obr. 3 Porovnanie globálneho a lokálneho prahovania na príklade obrazu s nerovnomerným jasom. Vľavo originál, v strede Otsu prahovanie, vpravo prahovanie pomocou lokálneho priemeru

## Referencie

- [1] Gonzalez, R., C., Woods, E., W., Digital Image Processing, Global Edition, 4th edition, Pearson 2018, ISBN 10: 1-292-22304-9

## Prílohy

### Zdrojový kód programu, ktorý vytvoril Obr. 1

```
clear all; close all; clc;
N=256
mtx=uint8(128*ones(N,N));
imshow(mtx,[0,255]);
h = drawcircle('Center',[N/2 N/2],"Radius",N/3);
mask = createMask(h);
mtx(mask>0)=128+64;

fig=figure
subplot(2,3,1)
myvar=0.02
mtxn = imnoise(mtx,'gaussian',0,myvar);
imshow(mtxn,[0,255]);
```

```

title(sprintf("original + g. noise(var=%.3f)",myvar))

subplot(2,3,2)
imhist(mtxn);
hold on
[T,EM] = graythresh(mtxn)
myT=T*255;
myYlim=yylim;
plot([myT myT],[myYlim(1) myYlim(2)],"red","LineWidth",3)
text(myT,600,sprintf("T_{Otsu}=%.0d\\rightarrow",myT),...
     'HorizontalAlignment','right','FontSize',12,"Color","red")
title("histogram")

subplot(2,3,3)
BW = imbinarize(mtxn,T);
imshow(BW);
title("binarized result")

subplot(2,3,4)
myFilt=1/25*ones(5,5)
mtxn = imfilter(mtxn,myFilt);
imshow(mtxn,[0,255]);
title(sprintf("smoothed (averaged 5x5)",myvar))

subplot(2,3,5)
imhist(mtxn);
hold on
[T,EM] = graythresh(mtxn)
myT=T*255;
myYlim=yylim;
plot([myT myT],[myYlim(1) myYlim(2)],"red","LineWidth",3)
text(myT,1000,sprintf("T_{Otsu}=%i\\rightarrow",myT),...
     'HorizontalAlignment','right','FontSize',12,"Color","red")
title("histogram")

subplot(2,3,6)
BW = imbinarize(mtxn,T);
imshow(BW);
title("binarized result")

fig.PaperUnits = 'inches';
fig.PaperPosition = [0 0 10 6];
print(fig,"segmOtsuSmooth.png","-dpng");

```

### Zdrojový kód programu, ktorý vytvoril Obr. 3

```

clear all; close all; clc;
I = im2gray(imread('rice.png'));
Totsu = graythresh(I)
BWotsu = imbinarize(I,Totsu);
Tadapt = adaptthresh(I, 0.5);
BWadapt = imbinarize(I,Tadapt);
montage({I,BWotsu,BWadapt},"Size",[1,3]);
exportgraphics(gcf(),'segmLocalAdapt.png')

```