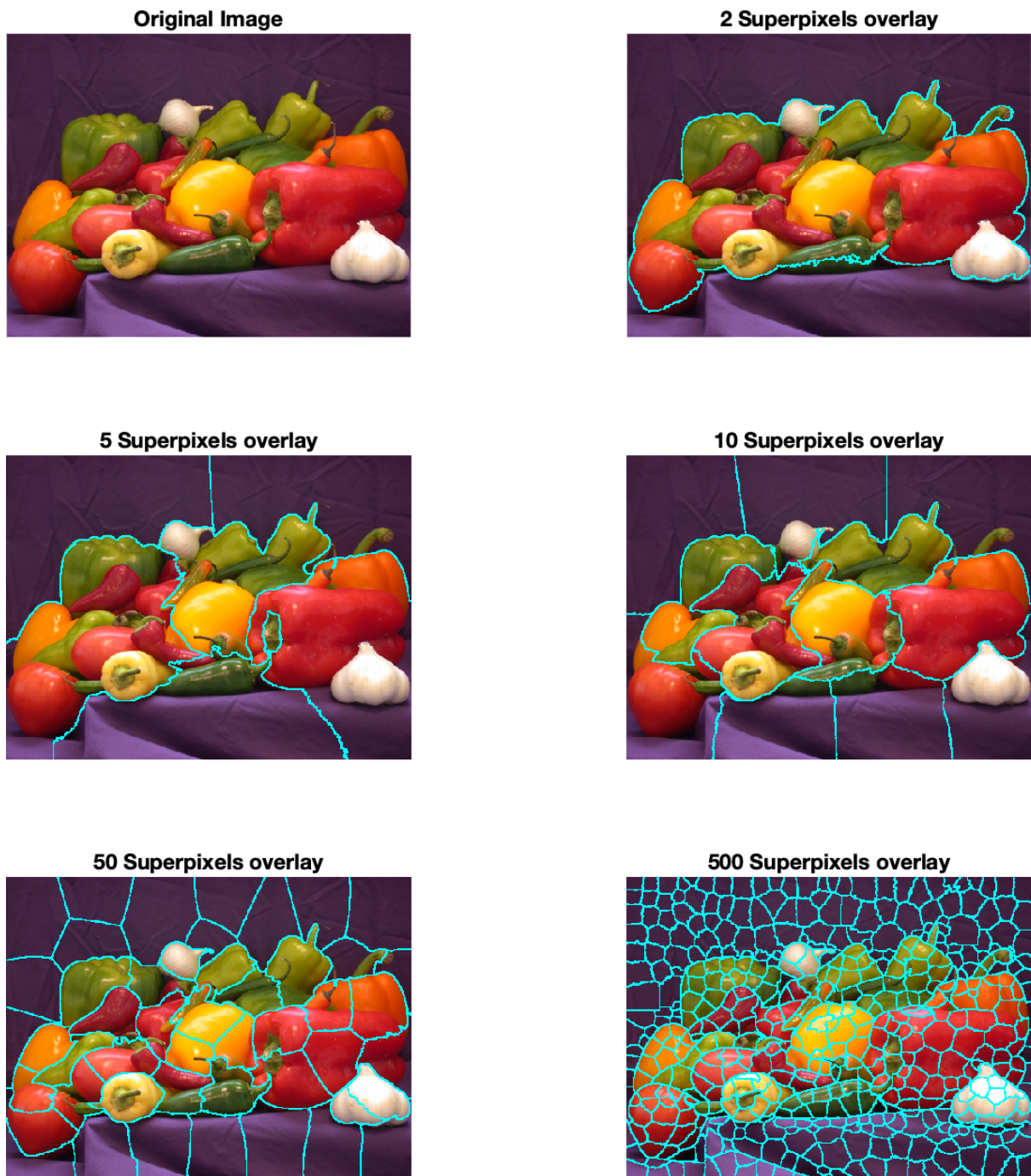


Vybrané segmentačné metódy

V tejto časti si predstavíme 3 vybrané metódy segmentácie: SLIC, activecontour a lazysnapping. Pri SLIC budeme využívať pojem superpixelov a je plne automatická. Activecontour a lazysnapping sú zase metódy, pre ktoré je vhodná asistancia, t.j. približné vyznačenie popredia a pozadia.

Pri segmentácii obrazu sa vo viacerých metódach používa pojem **superpixel**. Superpixelom sa nazýva oblasť bodov v obraze s podobnými vlastnosťami, pričom by mal byť z hľadiska vnímania zmysluplnejší ako jednotlivé body obrazu. Známym algoritmom na delenie obrazu na superpixely je SLIC (Simple Linear Iterative Clustering) [1]. SLIC je modifikáciou algoritmu K-means, ktorý sme mali spomenutý pri segmentácii farebného obrazu ktorý tvoril zhluky na základe podobnosti hodnôt bodov obrazu vo farebnom priestore bez uváženia ich polohy. SLIC rozširuje K-means práve o polohu, takže na miesto hľadania podobnosti v trojrozmernom farebnom priestore hľadáme podobnosti v 5 rozmernom priestore. T.j. súradnice sú (R, G, B, x, y) , resp. v inom farebnom priestore (H, S, V, x, y) a podobne. SLIC algoritmus zároveň zohľadňuje, že v tomto 5 rozmernom priestore sa kombinujú súradnice dvoch typov priestorové a hodnotové, takže oddelene počíta "farebnú" a priestorovú vzdialenosť. Tieto vzdialenosti následne normuje a váhuje do výslednej vzdialenosti. Pomocou váhy je možné určiť dôležitosť farebnej a priestorovej zložky vo výslednej vzdialenosti. V prostredí MATLAB vo funkcii **superpixels** je toto zahrnuté v parametri "Compactness". Príklad tvorby superpixelov je zobrazený na Obr. 1. Vidíme, že pri 2 superpixeloch vynikajúco oddelil popredie od pozadia.



Obr. 1 Príklad tvorby superpixelov SLIC algoritmom

Metóda **activecontour** využíva počiatočnú na hrubo načrtnutú kontúru (hranicu) objektu na popredí a snaží sa ju spresniť. V prostredí MATLAB sú implementované viaceré algoritmy spresňovania. Na farebný obraz sa dá použiť Chan-Vese algoritmus [2], ktorý je založený na evolúcii kriviek (hraníc oblastí) s kritériom minimalizácie metriky založenej na váhovanej energii regiónov a prekryvu s aktuálnou oblasťou. Evolúcia prebieha počas určeného počtu iterácií. Príklad aplikovania a získaného výsledku je na Obr. 2.

Metóda **lazysnapping** spresňuje (snapping) na hrubo načrtnuté pozadie a popredie. Pôvodný článok [3] používal krivky, MATLAB implementácia používa oblasti. Metóda je založená na teórii grafov a rezoch v sieťových grafoch s minimalizáciou energie. Používa superpixely namiesto bodov obrazu. Superpixely sú skupiny bodov s podobnými vlastnosťami. Príklad aplikovania a získaného výsledku je na Obr. 3.

Original Image



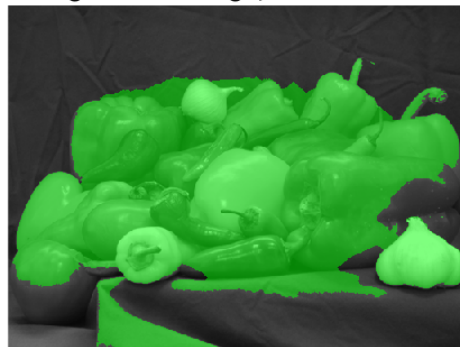
Initial Contour Location



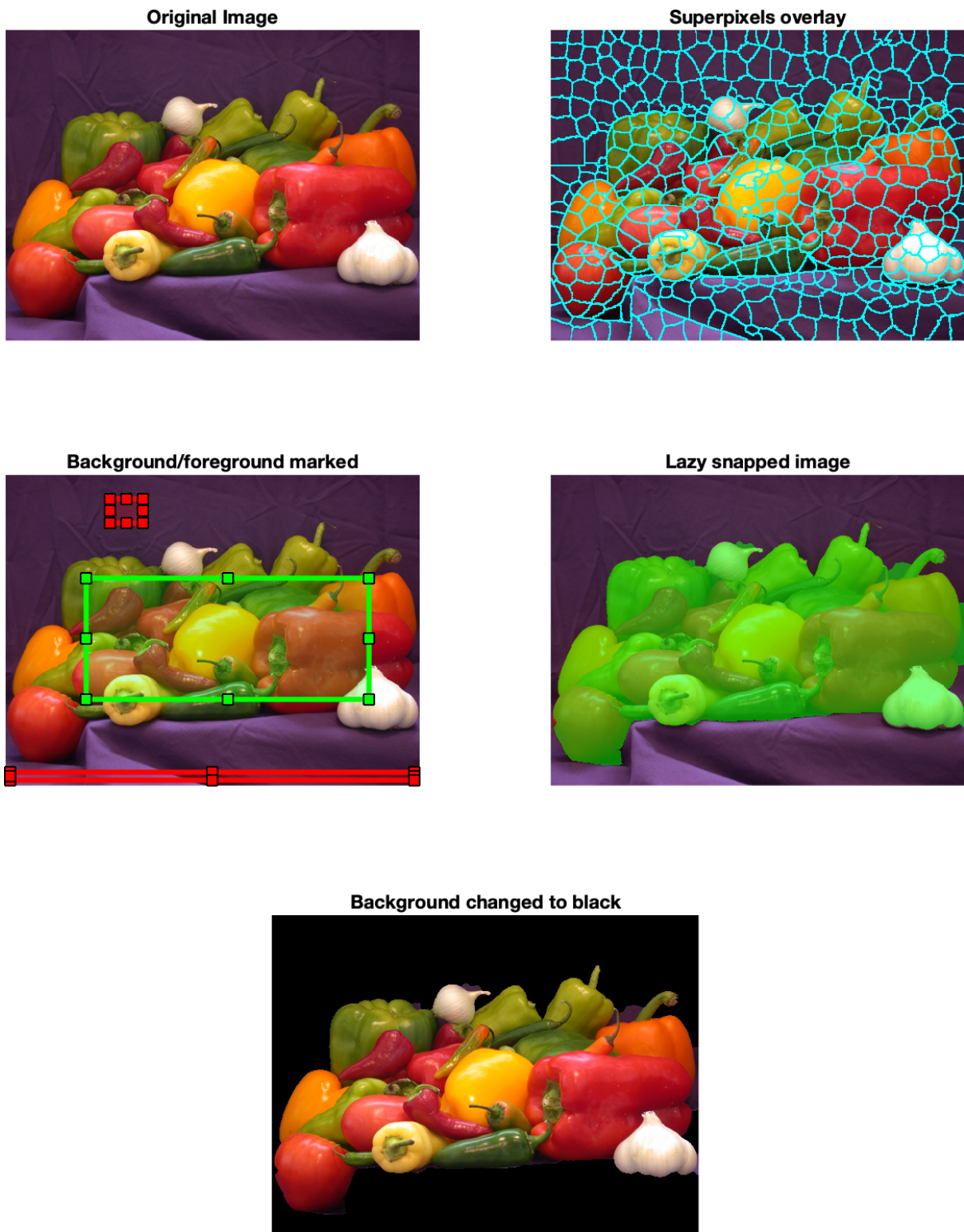
Segmented Image, 100 Iterations



Segmented Image, 300 Iterations



Obr. 1 Príklad segmentácie pomocou metódy activecontour s Chan-Vese algoritmom



Obr. 2 Príklad segmentácie pomocou metódy lazysnapping, na obrázku hore vpravo je označená jedna oblasť v popredí a dve oblasti v pozadí

Referencie

- [1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Susstrunk, SLIC Superpixels Compared to State-of-the-art Superpixel Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 34, Issue 11, pp. 2274-2282, May 2012
- [2] T. F. Chan, L. A. Vese, Active contours without edges. *IEEE Transactions on Image Processing*,

Volume 10, Issue 2, pp. 266-277, 2001.

- [3] Y. Li, S. Jian, C. Tang, H. Shum, Lazy Snapping In Proceedings from the 31st International Conference on Computer Graphics and Interactive Techniques, 2004.

Prílohy

Zdrojový kód programu, ktorý vytvoril Obr. 1

```
clear all; close all; clc;
fig=figure;

subplot(3,2,1)
RGB = imread("peppers.png");
imshow(RGB)
title("Original Image")

N=[2,5,10,50,500];
for idx=1:size(N,2)
    subplot(3,2,idx+1)
    L = superpixels(RGB,N(idx));
    BW = boundarymask(L);
    %imshow(imoverlay(RGB,BW,'cyan'),'InitialMagnification',67)
    imshow(imoverlay(RGB,BW,'cyan'))
    title(sprintf("%u Superpixels overlay",N(idx)));
end

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

Zdrojový kód programu, ktorý vytvoril Obr. 2

```
clear all; close all; clc;
fig=figure;
subplot(2,2,1)
I = im2gray(imread("peppers.png"));
imshow(I)
title("Original Image")

subplot(2,2,2)
mask = zeros(size(I));
mask(25:end-25,25:end-25) = 1;
imshow(mask)
title('Initial Contour Location')

subplot(2,2,3)
bw = activecontour(I,mask);
imshow(labeloverlay(I,bw,'Colormap',[0 1 0]))
title('Segmented Image, 100 Iterations')

subplot(2,2,4)
bw = activecontour(I,mask,300);
imshow(labeloverlay(I,bw,'Colormap',[0 1 0]))
title('Segmented Image, 300 Iterations')

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

Zdrojový kód programu, ktorý vytvoril Obr. 3

```

clear all; close all; clc;
fig=figure;
subplot(3,2,1)
RGB = imread("peppers.png");
imshow(RGB)
title("Original Image")

subplot(3,2,2)
L = superpixels(RGB,500);
BW = boundarymask(L);
%imshow(imoverlay(RGB,BW,'cyan'),'InitialMagnification',67)
imshow(imoverlay(RGB,BW,'cyan'),'InitialMagnification',67)
title("Superpixels overlay")

subplot(3,2,3)
imshow(RGB)
f = drawrectangle(gca,'Position',[100 128 350 150],'Color','g');
foreground = createMask(f,RGB);
b1 = drawrectangle(gca,'Position',[130 30 40 30],'Color','r');
b2 = drawrectangle(gca,'Position',[6 368 500 10],'Color','r');
title("Background/foreground marked")

subplot(3,2,4)
background = createMask(b1,RGB) + createMask(b2,RGB);
BW = lazysnapping(RGB,L,foreground,background);
imshow(labeloverlay(RGB,BW,'Colormap',[0 1 0]))
title("Lazy snapped image")

subplot(3,2,[5:6])
maskedImage = RGB;
maskedImage(repmat(~BW,[1 1 3])) = 0;
imshow(maskedImage)
title("Background changed to black")

fig.PaperUnits = 'inches';
fig.PaperPosition = [0 0 10 12];
print(fig,"segmOtherLazysnapping.png","-dpng");

```