

5 Hranová detekcia

5.1 Teoretický úvod

Špecifickým nástrojom na prácu s obrazmi sú **hranové operátory** – hranová detekcia. Hranová detekcia (operátory) vyhľadáva hranice medzi objektmi v obraze a nepriamo tak definuje samotné objekty. Hranica medzi objektami sa nazýva hrana. Hranou sa nazýva miesto v obraze, kde dochádza k náhlej zmene jasovej úrovne v obraze. Hranová detekcia je založená na označovaní bodov obrazu, ktoré môžu byť súčasťou hrany. Ak hodnota šedej úrovne obrazového bodu je podobná susednému obrazovému bodu, tak existuje veľká pravdepodobnosť, že v tomto bode nie je hrana. Naopak, ak obrazový bod má v susedstve body s výraznejšie odlišnými úrovňami šedej úrovne, môže to reprezentovať hranový bod. Takto nájdené body sú následne zlúčené do čiarových segmentov a čiarové segmenty sú potom zlúčené do hraníc objektov. Pri hranovej detekcii používajú niektoré hranové operátory preddefinované konvolučné masky, ktoré sú najčastejšie založené na diskretných aproximáciách k diferenciálnym operátorom. Diferenciálne operátory zisťujú mieru zmien v jasovej funkcii obrazu. Veľká zmena v jase cez krátku geometrickú vzdialenosť indikuje prítomnosť hrany. Niektoré operátory detekcie hrán dávajú informáciu aj o orientácii hrán, iné poskytujú len informáciu o existencii hrany. Mnohé hranové operátory sú značne citlivé na prítomnosť šumu v obraze. Preto je vhodné pred samotnou hranovou detekciou použiť pedspracovanie obrazu k odstráneniu alebo minimalizovaniu šumových prvkov za použitia rôznych filtrov. Použití hranového detektora s veľkou citlivosťou na zašumený obraz dochádza k označeniu veľkého množstva hranových bodov, ktoré predstavujú príspevok šumu k skutočným hranám a komplikujú určenie hrany respektíve vytvárajú ilúziu prítomnosti ďalších hrán či objektov. Ak použijeme menej citlivý detektor, môže prísť naopak ku strate platných hrán. V takomto prípade treba voliť kompromisné riešenie pri voľbe a nastavení operátora. Pri hranovej detekcii je možné nastaviť vhodné parametre pre hranový operátor. Sú to veľkosť masky hranovej detekcie a prahová úroveň. Väčšia maska je menej citlivá na šum, nižšia prahová úroveň má zase sklon redukovať šum.

5.2 Riešené úlohy

Úloha č. 1:

Na obraz *mri_brain.jpg* aplikujte nasledovné typy hranových detektorov pomocou príkazu *edge* – sobel, prewitt, roberts, log a canny. Jednotlivé výsledky zobrazte v samostatných oknách.

Riešenie:

```
% Použitie funkcie edge na šedotónový obraz
sobel=edge(I, 'sobel');
figure; imshow(sobel);
```



Obr. 5.1 Hranová detekcia – sobel (vľavo) a prewitt (vpravo)

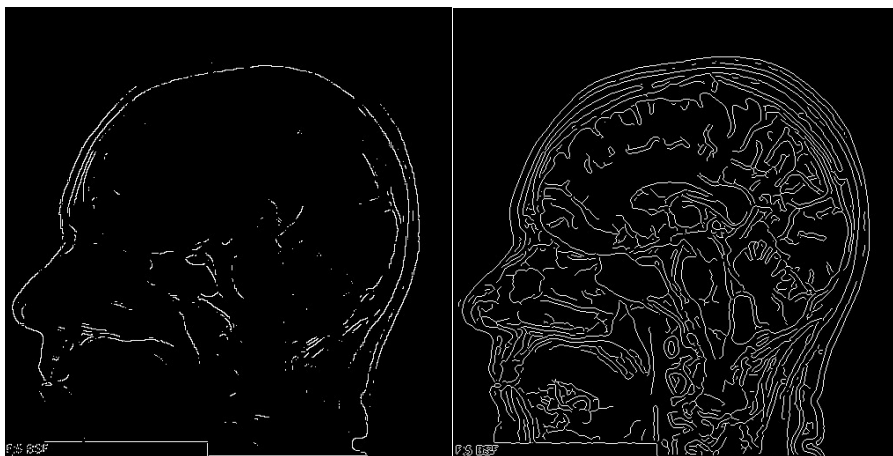
```
prewitt=edge(I, 'prewitt');
figure; imshow(prewitt);

roberts=edge(I, 'roberts');
figure; imshow(roberts);

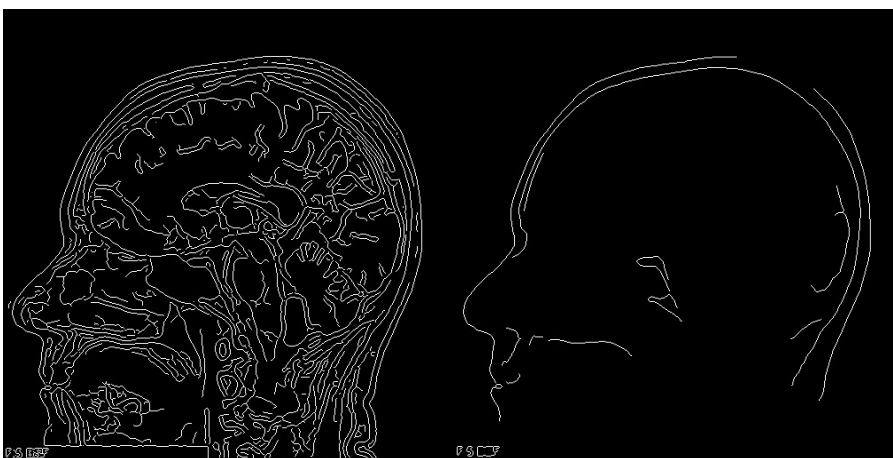
log=edge(I, 'log');
figure; imshow(canny);

canny=edge(I, 'canny');
figure; imshow(canny);

% Vlastnosti hranových operátorov je možné meniť zadáním
% dodatočných parametrov v opačnom prípade sú preddefinované
canny2=edge(I, 'canny', 0.6); % stanovenie zvoleného prahu
figure; imshow(canny2);
```



Obr. 5.2 Hranová detekcia – roberts (vľavo) a log (vpravo)



Obr. 5.3 Hranová detekcia – canny (vľavo) a canny s určeným prahom (vpravo)

Úloha č. 2:

Na obrázok *mri_brain.jpg* aplikujte dve konvolučné masky (horizontálnu a vertikálnu) Prewittovho operátora. Výsledné obrazy (obraz s horizontálnymi hranami a obraz s vertikálnymi hranami) sčítajte. Jednotlivé obrázky zobrazte v jednom okne vedľa seba. Výsledok porovnajte s funkciou *edge(I,'prewitt')*.

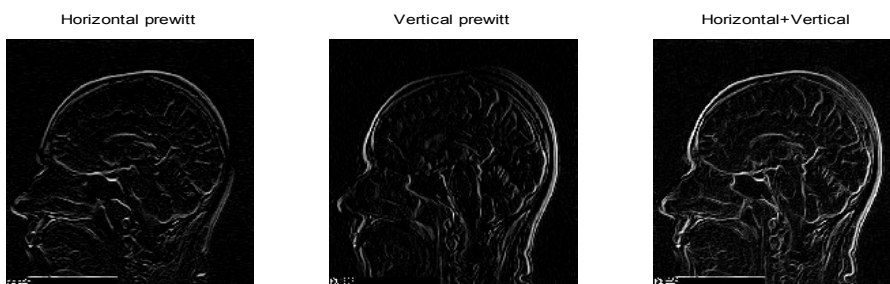
$$prewitt_hor = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}; \quad prewitt_vert = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix};$$

Riešenie:

```
% Vytvorenie operátorov
prewitt_hor=[1 1 1; 0 0 0; -1 -1 -1]; % horizontálny operátor
prewitt_vert=[1 0 -1; 1 0 -1; 1 0 -1]; % vertikálny operátor

% aplikácia operátorov na obraz
horizontal=imfilter(MRI_obraz,prewitt_hor);
vertical=imfilter(MRI_obraz,prewitt_vert);

% kombinácia/sčítanie vzniknutých obrazov
Hor_Vert=imadd(horizontal,vertical);
```



Obr. 5.4 Zobrazenie použitia rôznych operátorov prewitt

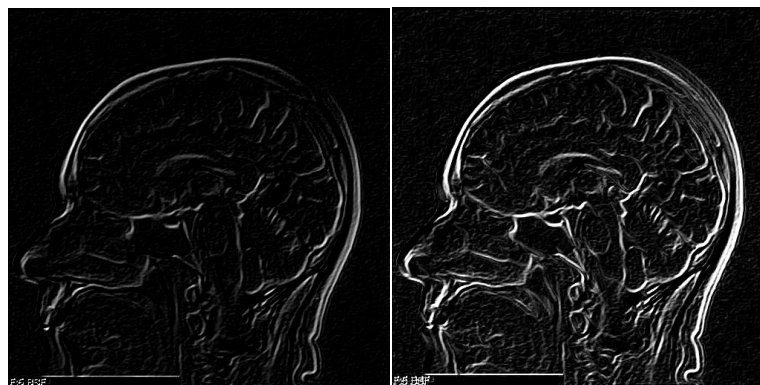
```
% Vytvorenie a použitie diagonálneho operátora
prewitt_diag=[1 1 0;1 0 -1;0 -1 -1];
diagonal=imfilter(MRI_obraz,prewitt_diag);
Hor_Vert_Diag=imadd(Hor_Vert,diagonal);
```

5.3 Neriešené úlohy

Úloha č. 3:

Vytvorte konvolučnú masku diagonálneho Prewittovho operátora a aplikujte ju na obrázok *mri_brain.jpg*. Následne výsledný obraz pričítajte k výslednému obrazu z úlohy č. 2 (súčet horizontálnych a vertikálnych hrán). Výsledok opäť porovnajte s výstupom z funkcie *edge(I,'prewitt')*.

$$prewitt_diag = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{bmatrix}$$



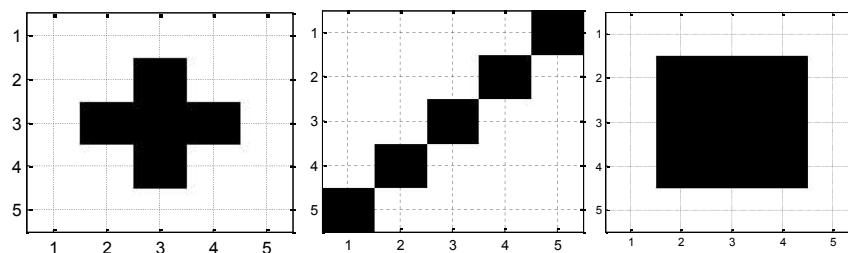
Obr. 5.5 Použitie diagonálnej masky(vľavo) a suma všetkých 3 masiek(vpravo)

7 Morfológické operácie

7.1 Teoretický úvod

Morfologickú transformáciu možno popísať ako interakciu medzi množinou bodov binárneho obrazu a štruktúrnym elementom SE. Štruktúrny element pozostáva taktiež z množiny bodov, ktorá je však menšia ako množina bodov obrazu. SE je väčšinou symetrický a v jeho centre sa nachádza jeden význačný bod X, ktorý sa nazýva reprezentatívny bod.

V programovom prostredí MATLAB je preddefinovaných niekoľko SE (diamond, disk, line, rectangle, square. ...), ktoré možno priamo použiť na morfologické operácie alebo sa SE zdefinuje priamo v kóde podľa požiadaviek. Na Obr. 7.1 sú zobrazené niektoré používané jednoduché SE.



Obr. 7.1 Štruktúrne elementy (definované „čiernymi“ bodmi)

Medzi morfologické operácie patrí:

- Dilatácia
- Erózia
- Otvorenie
- Zatvorenie

Dilatácia objektu štruktúrnym elementom spôsobuje zväčšovanie sa (rozpínanie) daného objektu. Táto operácia sa využíva na „plátanie“ malých dier alebo na spájanie blízkych objektov obrazu do jedného objektu a je založená na Minkowského súčte. Dilatácia prebieha postupným prikladaním SE na obraz a jeho následným posúvaním po obraze. V každej polohe sa kontroluje prekrytie množiny bodov SE a prislúchajúcej množiny bodov obrazu. Ak sa reprezentatívny bod SE prekryje s nulovou hodnotou pixelu na binárnom obraze, nedochádza k žiadnej zmene a SE sa posunie na ďalší pixel. Ak sa však reprezentatívny bod prekryje s jednotkovou hodnotou pixelu, vykoná sa logická operácia

OR na všetkých pixeloch obrazu, ktoré SE pokrýva. To znamená pixely, ktoré SE pokrýva nadobudnú hodnotu jedna a objekt sa rozšíri o pixely, ktoré zmenili hodnotu z 0 na 1.

Erózia objektu štruktúrnym elementom spôsobuje zmenšovanie sa (leptanie) daného objektu obrazu. Táto operácia sa využíva na odstránenie malých výbežkov objektov a na rozdelenie objektov, ktoré sú spojené a nemožno určiť ich hrany z dôvodu ich prekrytia. Erózia je založená na Minkowského rozdiely a prebieha postupným prikladaním SE rovnako ako dilatácia, avšak jedná sa o inverznú operáciu: Ak sa SE pokrýva vo všetkých pozíciách s hodnotu 1 binárneho obrazu, zostane na pixeli pod reprezentatívnym bodom zapísaná hodnota 1 a SE sa iba posunie na ďalšiu pozíciu. Ak sa SE posunie na pixel, na ktorom nespĺňa tento stav, t.j. SE nepokrýva iba hodnoty 1 tak sa zapíše hodnota 0 na pixel pod reprezentatívnym bodom.

Operácia **otvorenia** je kombináciou erózie nasledovanej dilatáciou. Otvorenie sa používa na odstránenie bodov objektu (výbežkov), ktoré sú príliš malé voči štruktúrnemu elementu.

Naproti tomu operácia **zatvorenia** je kombinácia dilatácie nasledovanej eróziou. Používa sa na vyplnenie dier a medzier, bez výraznejšieho posunu hrán na rozdiel od dilatácie.

7.2 Riešené úlohy

Úloha č. 1:

Vytvorte obrazovú maticu rozmeru 5x5 typu *logical*, ktorá obsahuje symetricky umiestnený štvorec 3x3. Pomocou obrazovej konvolúcie vynásobte obraz so štruktúrnym elementom (maskou) typu 3x3 krížik - dilatácia.

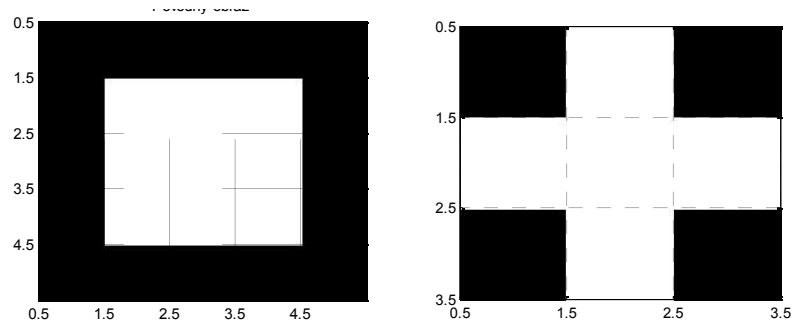
Riešenie:

```
% Vytvorenie obrazovej matice a jej zobrazenie
```

```
I=[0 0 0 0 0;
    0 1 1 1 0;
    0 1 1 1 0;
    0 1 1 1 0;
    0 0 0 0 0];
```

```
I=logical(I); % pretypovanie obrazu na binárny obraz
```

```
figure; imagesc(I); colormap gray;
set(gca, 'XTick', 0.5:1:5, 'YTick', 0.5:1:5); grid on;
```



Obr. 7.2 Obrazová matica (vľavo) a štruktúrny element (vpravo)

```
% vytvorenie štruktúrneho elementu SE
SE=[0 1 0;1 1 1;0 1 0];
% aplikácia SE na obraz I
I_dilatácia=imfilter(I,SE,'conv'); % obrazová konvolúcia

% Zobrazenie dilatovaného obrazu (obr. 7.3 vľavo)
figure; imagesc(I_dilatácia); colormap gray;
set(gca, 'XTick',0.5:1:5, 'YTick',0.5:1:5); grid on;
```

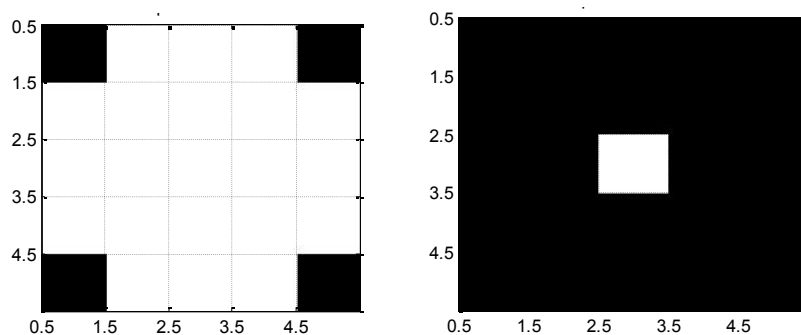
Úloha č. 2:

Pôvodnú obrazovú maticu I z úlohy č. 1 invertujte (*imcomplement*) a znova aplikujte konvolúciu s rovnakou maskou. Výsledný obraz znova invertujte. Výsledok zodpovedá erózii.

Riešenie:

```
% Vytvorenie inverznej obrazovej matice a aplikácia masky
I2=imcomplement(I);
I_erozia=imfilter(I2,maska,'conv');

% Spätné invertovanie obrazu a zobrazenie (obr. 7.3 vpravo)
I_erozia =imcomplement(I_erozia);
figure; imagesc(I_erozia); colormap gray;
set(gca, 'XTick',0.5:1:5, 'YTick',0.5:1:5); grid on;
```



Obr. 7.3 Výstupný obraz z dilatácie (vľavo) a z erózie obrazu (vpravo)

Úloha č. 3:

Aplikujte morfológické operácie - dilatácia, erózia na vytvorený obraz z úlohy č.1 pomocou funkcií *imdilate*, *imerode*. Štruktúrny element vytvorte pomocou funkcie *strel*. Porovnajte výstupy s obr. 7.3. Následne načítajte binárny obraz *domcek.bmp*, na ktorý aplikujte operácie otvorenia a zatvorenia (*imopen*, *imclose*).

Riešenie:

```
% Vytvorenie štruktúrneho elementu SE
maska=[0 1 0;1 1 1;0 1 0];
SE = strel('arbitrary', maska);

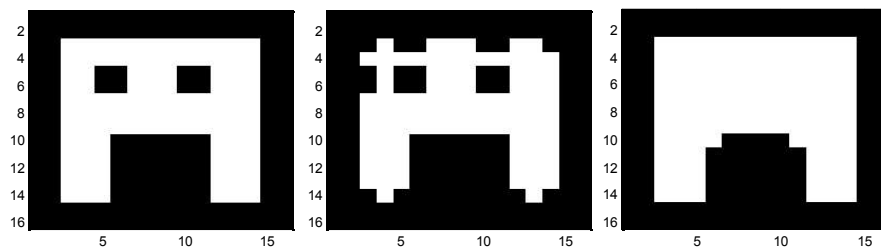
% Dilatácia obrazu
I_dilate=imdilate(I,SE);
figure; imagesc(I_dilate); colormap gray;

% Erózia obrazu
I_erode=imerode(I,SE);
figure; imagesc(I_erode); colormap gray;

% Načítanie obrazu
dom=imread('domcek.bmp');
figure; imagesc(D); colormap gray;

% Aplikácia operácie otvorenia a zatvorenia
otvorenie=imopen(dom,SE);
zatvorenie=imclose(dom,SE);

figure; imagesc(otvorenie);
figure; imagesc(zatvorenie);
```



Obr. 7.4 Aplikácia otvorenia(stred), a zatvorenia(vpravo) na obraz (vľavo)

7.3 Neriešené úlohy**Úloha č. 4:**

Načítajte obraz *rez_gray.jpg* a vysegmentujte z neho stavec. Preved'te obraz na binárny s použitím prahu s hodnotou 144 a pomocou vhodných morfológických operácií zobrazte plný segment prislúchajúci stavcu chrbtice. Násobte obraz so segmentom s originálnym obrazom, pričom z obrazu vytiahnite stavec v úrovniach šedi. Výsledný obraz zobrazte.