

FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY
UNIVERZITY KOMENSKÉHO V BRATISLAVE



DIPLOMOVÁ PRÁCA

Zbierka príkladov z počítačového videnia

Bratislava 2008

Silvia Strungová

Zbierka príkladov z počítačového videnia

DIPLOMOVÁ PRÁCA

Silvia Strungová

UNIVERZITA KOMENSKÉHO BRATISLAVA
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY
KATEDRA APLIKOVANEJ INFORMATIKY

Informatika

Doc. RNDr. Milan Ftáčnik, PhD.

BRATISLAVA 2008

Čestné prehlásenie

Čestne prehlasujem, že som diplomovú prácu Zbierka príkladov z počítačového videnia vypracovala samostatne, s použitím nižšie uvedenej literatúry.

.....

Pod'akovanie

Ďakujem vedúcemu diplomovej práce Doc. RNDr. Milanovi Ftáčnikovi, PhD. za konzultácie a cenné rady. Zároveň ďakujem svojej rodine a priateľom za psychickú podporu.

Abstrakt

STRUNGOVÁ, Silvia: Zbierka príkladov z počítačového videnia. - Univerzita Komenského, Bratislava. Fakulta matematiky, fyziky a informatiky; Katedra aplikovanej informatiky. - Vedúci diplomovej práce: Doc. RNDr. Milan Ftáčnik, PhD. - Bratislava: FMFI UK, 2008, 87 strán.

Cieľom tejto diplomovej práce bolo pripraviť sériu riešených príkladov z pokročilých častí počítačového videnia. Vytvorili sme internetovú zbierku riešených aj neriešených príkladov. Na riešenie sme využili voľne dostupný program ImageJ. Poskytli sme tiež návod, ako sa pomocou neho dajú riešiť príklady.

Keywords: Počítačové videnie, ImageJ, zbierka

Obsah

Úvod	8
1 Prehľad problematiky	9
2.1 Prehľad internetových učebníc	9
2.2 Prehľad internetových zbierok	10
2 Vývoj práce	11
2.1 Prvotná myšlienka	11
2.2 Návrh štruktúry	11
2.3 Tvorba otázok a príkladov	16
2.4 Proces riešenia	18
3 ImageJ	21
3.1 Základná charakteristika	21
3.2 Inštalácia	21
3.3 Prostredie	22
3.4 Ako riešiť príklady s ImageJ	23
4 Stránka	31
4.1 Dizajn	31
4.2 Technológie	32
Záver	33
Zoznam použitej literatúry	34
Prílohy	36
A Zbierka príkladov	37
B Riešenia príkladov	60

Zoznam obrázkov

3.1	Inštalácia	22
3.2	ImageJ	22
3.3	Panel nástrojov	23
3.4	Panel stavov	23
3.5	Panel činností	23
3.6	Nový obrázok	24
3.7	Nekonvexný mnohoúhelník	25
3.8	Konvexný obal mnohoúhelníka	25
3.9	Cannyho hranová detekcia	26
3.10	Menu Plugins	26
3.11	Nové textové okno	27
3.12	Median5	30
4.1	Stránka	31

Úvod

V súčasnosti existuje veľké množstvo literatúry z oblasti počítačového videnia, avšak väčšinou ide o anglickú literatúru. Snahou viacerých diplomových prác na FMFI UK je vytvorenie interaktívnych učebníc a zbierok príkladov publikovaných na internete, ktoré sú pomocou pri výučbe daných predmetov, slúžia na získanie a následné overenie daných vedomostí. Keďže dnešný študent mnohokrát radšej siahne po informáciách na internete, ako po vyhľadávaní v knihách, je pre neho jednoduchšie, ak existuje nejaká internetová učebnica, kde nájde súhrn toho, čo potrebuje a kde si môže svoje vedomosti následne preveriť formou nejakej sady otázok z danej problematiky, prípadne nájsť nejaké vzory riešení príkladov.

Cieľom tejto diplomovej práce je pripraviť sériu riešených príkladov z pokročilých častí počítačového videnia s využitím softvéru ImageJ a publikovať ju na internete. Mala by slúžiť ako pomôcka pri výučbe predmetu Počítačové videnie na Katedre aplikovanej informatiky FMFI UK.

V prvej kapitole hodnotíme stav súčasnej interaktívnej literatúry v oblasti počítačového videnia a spracovania obrazu. Predstavíme doteraz vytvorené interaktívne učebnice a zbierky, taktiež tie, ktoré sú ešte vo vývoji.

V druhej kapitole popisujeme postup, ako sme zbierku tvorili. Ako sme vytvorili štruktúru, rozdelili jednotlivé kapitoly. Ďalej uvádzame, ako sme tvorili otázky, ich počet a rozdelenie a v neposlednom rade ako sme postupovali pri riešení časti príkladov.

Tretia kapitola je venovaná programu ImageJ. Poskytujeme tu základné informácie o programe, zoznámime čitateľa s inštaláciou a prostredím. Jadrom tejto kapitoly je návod, ako použiť program na riešenie príkladov.

Štvrtá kapitola popisuje tvorbu internetovej stránky. Popisujeme samotný dizajn, ale aj techniky, aké sme pri jej tvorbe využili.

Práca obsahuje aj dve prílohy, prvou je samotná zbierka príkladov a druhou sú riešené príklady.

Naším zámerom bolo vytvoriť prehľadnú zbierku príkladov, ktorá bude čitateľovi – študentovi poskytovať možnosť overenia si získaných vedomostí, ale aj možnosť využiť logické uvažovanie pri riešení príkladov.

1 Prehľad problematiky

Počítačové videnie je vedecká disciplína, ktorej snahou je technickými prostriedkami aspoň čiastočne napodobniť ľudské videnie. V súčasnosti existuje v tejto oblasti veľké množstvo literatúry, avšak prevažnú časť tvorí anglická literatúra. Ďalším problémom pre čitateľa – študenta môže byť aj fakt, že existencia veľkého množstva materiálov môže spôsobiť neprehľadnosť a ťažšiu orientáciu sa v problematike.

Keďže jedným zo súčasných trendov je aj informatizácia spoločnosti, existuje viacero diplomových prác na FMFI UK, ktoré spracúvajú danú problematiku vo forme interaktívnych učebníc a zbierok príkladov publikovaných na internete. Tieto sú pomôckou pri výučbe daných predmetov, slúžia na získanie a následné overenie daných vedomostí.

1.1 Prehľad internetových učebníc

Prvým pokusom o vytvorenie slovenskej internetovej učebnice z oblasti počítačovej grafiky je Výučba počítačovej grafiky [5] od Mgr. Juraja Štugela. Je to diplomová práca obhájená v roku 2000, ktorá sa stále vyvíja. Jej cieľom je priblížiť základy počítačovej grafiky pomocou nových metód a využitím nových technológií.

Táto internetová učebnica má 20 kapitol: Úvod, Základné 2D grafické objekty a ich rasterizácia, Orezávanie a prieniky, Krivky a plochy, Spracovanie obrazu I, Spracovanie obrazu II, Transformácie v 2D a 3D, Zobrazovanie a modelovanie 3D objektov, Viditeľnosť objektov, Svetlo a farebné modely, Osvetľovacie modely a tieňovanie, Globálne zobrazovacie modely, Textúry, Fraktály, Kódovanie a programovanie 3D, Grafické formáty, Sieťové multimédiá, Grafické karty a monitory, Normalizované grafické systémy, Záver. Každá z týchto kapitol je rozdelená na podkapitoly, kde autor vysvetľuje základné pojmy formou definícií a príkladov. Učebnicu namiesto klasických obrázkov oživujú interaktívne programy - java aplety. Tieto čitateľovi umožňujú priamo si vyskúšať nadobudnuté vedomosti, ale aj lepšie pochopiť danú tému. Pri každom aplete sa nachádza krátky popis, ako s ním treba pracovať.

Ďalšou takouto prácou je diplomová práca Interaktívna učebnica spracovania obrazu [6] od Mgr. Gábora Blázsovitsa. Táto pozostáva zo siedmich kapitol (Úvod, Predspracovanie, Transformácia obrazu, Segmentácia, Morfológická transformácia, Obnovenie obrazu,

Rozpoznávanie obrazcov), ktoré sú taktiež ešte delené na podkapitoly a oživené Java apleti (spolu je ich 42). Práca obsahuje aj index, je teda možné vyhľadávať aj podľa neznámych pojmov. Súčasťou je aj download a linky na stránky zaoberajúce sa témou počítačovej grafiky.

V roku 2007 obhájila Mgr. Júlia Birošová ďalšiu zo série internetových učebníc Interaktívna učebnica rozpoznávania obrazcov [11]. Učebnica pozostáva z nasledujúcich piatich kapitol: Popis tvaru, Príznakové metódy, Štrukturálne metódy, Iné metódy reprezentácie znalostí a Porozumenie. Jej súčasťou je taktiež množstvo apletov, index a slovník pojmov z rozpoznávania obrazu.

V súčasnosti na podobnom diele pracuje Kristína Vašková, ktorá robí Interaktívnu učebnicu z počítačového videnia. Naša zbierka by mala byť jej doplnením.

1.2 Prehľad internetových zbierok

Čo sa týka oblasti zbierok príkladov, ich počet nie je až taký veľký a teda vzniká priestor na možné rozšírenie.

V roku 2000 obhájila Mgr. Barbora Maťková diplomovú prácu Interaktívne testy v počítačovej grafike [2]. Ide vlastne o doplnenie interaktívnej výučby počítačovej grafiky Mgr. Juraja Štugela [5]. Jej softvér je funkčný online (s prístupom k internetu) aj offline (bez prístupu), obsahuje spolu 504 otázok (na overenie vedomostí získaných zo Štugelovej učebnice), ktoré sú rozdelené v 19-tich kapitolách.

Ďalšom prácou, ktorá je momentálne vo vývoji, je diplomová práca Veroniky Grófovej, Vyučovanie počítačového videnia. Jej cieľom je vytvoriť zbierku riešených a neriešených úloh z počítačového videnia, ktorá by mala dopĺňať učebnicu Gábora Blázsovitsa [6]. Výsledná zbierka by mala byť členená do piatich hlavných kapitol: Digitalizácia a reprezentácia obrazu, Predspracovanie obrazu, Obnovenie obrazu, Segmentácia a Reprezentácia a popis obrazu. Táto práca je svojou formou podobná našej zbierke, t.j. obsahuje riešené i neriešené príklady. Rozdiel medzi týmito prácami je v tom, že my sa snažíme ešte o rozdelenie otázok v rámci kapitoly podľa istej úrovne náročnosti.

2 Vývoj práce

2.1 Prvotná myšlienka

Klasické a osvedčené postupy vzdelávania sa využívajú dodnes s rôznymi obmenami, ktoré ovplyvňuje technický pokrok a ekonomické možnosti spoločnosti. Keďže ale žijeme v dobe neustálych zmien, dobe nových myšlienok, nápadov a hlavne dobe budovania informačnej spoločnosti, je preto nevyhnutné, aby bol tomuto prispôsobený aj spôsob vzdelávania. Ako sme sa dozvedeli v predchádzajúcej kapitole, existuje už viacero takýchto pokusov, ktoré sa ujali a sú využívané študentmi.

Keďže Počítačové videnie je jedným z predmetov na Katedre aplikovanej informatiky FMFI UK, a v tejto oblasti ešte nebola vytvorená žiadna ucelená zbierka príkladov (v slovenskom jazyku), rozhodli sme sa takúto zbierku vytvoriť. Prihliadnuc na dostupnosť a už spomínané budovanie informačnej spoločnosti sme zvolili možnosť umiestnenia zbierky na internete. Prácu sme sa rozhodli doplniť aj vzorovými návodmi a riešeniami, aby si čitateľ - študent mohol svoje výsledky porovnať, prípadne zistiť, čo asi sa od neho vyžaduje. Na riešenie niektorých príkladov sme sa rozhodli použiť voľne dostupný program ImageJ. Čitateľovi dáme presný návod, kde nájsť program, ako ho inštalovať, oboznámime ho s prostredím a návodom, ako pomocou neho môže začať riešiť príklady.

2.2 Návrh štruktúry zbierky

Základom každej učebnice a zbierky je zvoliť si vhodnú štruktúru, aby sme dosiahli prehľadnosť a dobrú orientáciu v texte. Toto bol v prvej časti návrhu štruktúry aj náš problém. Keďže zbierka má dopĺňať učebnicu Kristíny Vaškovej, zvolili sme rozdelenie kapitol podľa nej. Prvotné rozdelenie zbierky teda vyzeralo nasledovne:

1. Dátové štruktúry pre analýzu obrazu
 - 1.1 Úroveň reprezentácie obrazových dát
 - 1.2 Tradičné obrazové dátové štruktúry
 - 1.1.1. Matice
 - 1.1.2. Reťazce

- 1.1.3. Topologické dátové štruktúry
 - 1.1.4. Relačné štruktúry
 - 1.3 Hierarchické dátové štruktúry
 - 1.1.5. Pyramídy
 - 1.1.6. Kvadrantové stromy
 - 1.1.7. Ďalšie pyramídové štruktúry
- 2. Pokročilé techniky predspracovania obrazu
 - 2.1 Zero – crossing
 - 2.2 Škálovanie v spracovaní obrazu
 - 2.3 Parametrické hranové modely
 - 2.4 Hrany v multispektrálnych obrázkoch
 - 2.5 Predspracovanie v adaptívnom okolí
- 3. Pokročilé techniky segmentácie
 - 3.1 Segmentácia založená na hranách
 - 3.1.1 Zisťovanie hranice ako hľadanie grafu
 - 3.1.2 Zisťovanie hranice ako dynamické programovanie
 - 3.1.3 Houghova transformácia
 - 3.1.4 Zisťovanie hranice použitím hraničnej lokálnej informácie
 - 3.1.5 Rekonštrukcia oblastí z hraníc
 - 3.2 Segmentácia založená na oblastiach
 - 3.2.1 Segmentácia povodí
 - 3.2.2 Dodatočné spracovanie narastania oblastí
 - 3.3 Porovnávanie so vzorom
 - 3.3.1 Kritéria porovnávania so vzorom
 - 3.3.2 Kontrola porovnávania so vzorom
 - 3.4 Techniky dodatočného spracovania narastania oblastí
 - 3.4.1 Simultánne hľadanie hranice
 - 3.4.2 Detekcia povrchu

4. Pokročilé techniky reprezentácie tvaru a jej popisu
 - 4.1 Popisy tvaru založené na hranách
 - 4.1.1 Fourierove popisy tvaru
 - 4.1.2 Popis hranice použitím postupností segmentov
 - 4.1.3 B-splajnová reprezentácia
 - 4.1.4 Iné popisy tvaru založené na hranách
 - 4.1.5 Tvarové invarianty
 - 4.2 Popisy tvaru založené na oblastiach
 - 4.2.1 Grafová reprezentácia založená na kostre regiónu
 - 4.2.2 Rozklad oblastí
 - 4.2.3 Graf susedností oblastí
 - 4.3 Triedy tvaru

5. Rozpoznávanie obrazu
 - 5.1 Reprezentácia znalostí
 - 5.2 Neurónové siete
 - 5.2.1 Dopredné siete
 - 5.2.2 Nekomrolované učenie
 - 5.2.3 Hopfieldove neurónové siete
 - 5.3 Rozpoznávanie ako porovnávanie grafov
 - 5.3.1 Grafové a podgrafové izomorfizmy
 - 5.3.2 Grafová podobnosť
 - 5.4 Optimalizačné techniky v rozpoznávaní
 - 5.4.1 Genetické algoritmy
 - 5.4.2 Simulované žíhanie

6. Porozumenie obrazu
 - 6.1 Riadiace stratégie porozumenia obrazu
 - 6.1.1 Paralelné a sekvenčné
 - 6.1.2 Hierarchické riadenie
 - 6.1.3 Riadenie zdola nahor

- 6.1.4 Riadiaca stratégia riadená modelom
- 6.1.5 Kombinované riadiace stratégie
- 6.1.6 Nehierarchické riadenie
- 6.2 Modely aktívnej hranice - hady
- 6.3 Modely rozmiestnenia bodov
- 6.4 Rozpoznávanie obrazcov v porozumení obrazu
 - 6.4.1 Kontextuálna klasifikácia obrazu
- 6.5 Označovanie scén
 - 6.5.1 Diskrétné označovanie
 - 6.5.2 Pravdepodobnostné označovanie
 - 6.5.3 Princíp posúvania obmedzení
- 6.6 Sémantická segmentácia obrazu a porozumenie
 - 6.6.1 Sémantické narastanie oblastí
 - 6.6.2 Genetická interpretácia obrazu
- 6.7 Skryté Markove modely

7. Textúry

- 7.1 Štatistický popis textúr
 - 7.1.1 Metódy založené na priestorovej frekvencii
 - 7.1.2 Matice opakovaného objavenia
 - 7.1.3 Hranová frekvencia
 - 7.1.4 Dĺžka primitíva
 - 7.1.5 Lawove textúrne merania
 - 7.1.6 Popis fraktálnych textúr
 - 7.1.7 Ďalšie štatistické metódy popisu textúr
- 7.2 Metódy syntaktického popisu textúr
 - 7.2.1 Gramatiky tvarových reťazcov
 - 7.2.2 Grafové gramatiky
 - 7.2.3 Zoskupovanie primitív textúr
- 7.3 Metódy hybridného popisu textúr

Naším cieľom bolo spraviť zbierku, kde budú otázky zaradené podľa náročnosti a teda sme sa rozhodli rozdeliť každú kapitolu na dve časti:

A – Otázky

B - Príklady

Do časti A sme zaradili otázky, ktoré boli zamerané na definície, rozdelenia, porovnania... Teda túto časť tvorili otázky, na ktoré by čitateľ, po naštudovaní danej problematiky, nemal mať problém odpovedať. Táto časť má slúžiť na preverenie čitateľovej orientácie v texte, schopnosti zapamätania si čítaného textu.

Do časti B sme zaradili otázky, ktorých riešenie si vyžaduje hlbšie naštudovanie problematiky, podrobné rozobratie problému, preukázanie určitej zručnosti čitateľa, logického zmýšľania, popr. využitie už spomínaného programu ImageJ.

Po aplikovaní tohto rozdelenia, na predchádzajúcu štruktúru práce, sme namiesto prehľadnej zbierky dostali neprehľadnú postupnosť rozdelení, v ktorej bolo problém sa orientovať a nie sa ešte sústrediť na nejaké následné riešenie príkladov. Ďalším problémom takéhoto rozdelenia bol aj fakt, že pomer otázok a príkladov bol dosť nerovnomerný a taktiež sme v niektorých častiach mali problém s vytvorením vhodných príkladov. Preto sme si zvolili vlastnú štruktúru zbierky, ktorá ale obsahovo pokrýva všetky už spomínané kapitoly.

Konečná štruktúra práce je teda nasledovná:

1. Dátové štruktúry pre analýzu obrazu
 - 1.A. Otázky
 - 1.B. Príklady
2. Pokročilé techniky predspracovania obrazu
 - 2.A. Otázky
 - 2.B. Príklady
3. Pokročilé techniky segmentácie
 - 3.A. Otázky
 - 3.B. Príklady
4. Pokročilé techniky reprezentácie tvaru a jej popisu
 - 4.A. Otázky
 - 4.B. Príklady

- 5. Rozpoznávanie obrazu
 - 5.A. Otázky
 - 5.B. Príklady
- 6. Porozumenie obrazu
 - 6.A. Otázky
 - 6.B. Príklady
- 7. Textúry
 - 7.A. Otázky
 - 7.B. Príklady

Podľa nášho názoru je takéto rozdelenie dostatočne prehľadné, pomer otázok a príkladov každej kapitoly je vyvážený a obsahovo zahŕňa toto rozdelenie všetky kapitoly predchádzajúcej štruktúry (v tom istom poradí). Aby náhodou nevznikol problém pri hľadaní sady otázok nejakej podkapitoly, vytvorili sme index kľúčových slov, ktorý vždy odkazuje na prvú otázku týkajúcu sa danej problematiky.

2.3 Tvorba otázok a príkladov

Ako sme už spomenuli v predchádzajúcej kapitole, prvou snahou o vytvorenie internetovej zbierky bola práca Mgr. Barbory Matčkovej [2]. Teda náš prvotný cieľ pri tvorbe otázok bol inšpirovať sa jej prácou. V prvom kroku sme si dôkladne preštudovali jej prácu, ktorá spolu obsahuje 504 otázok. Otázky sú usporiadané v nasledujúcich kapitolách:

- Základné 2D grafické objekty a ich rasterizácia
- Orezávanie a prieniky
- Krivky a plochy
- Spracovanie obrazu
- Transformácie v 2D a 3D
- Zobrazovanie a modelovanie 3D objektov
- Viditeľnosť objektov
- Svetlo a farebné modely
- Osvetľovacie modely a tieň
- Globálne zobrazovacie metódy

Textúry
Grafické formáty
Normalizované grafické systémy
Geometria a Zložitosť geometrických algoritmov

Každá kapitola obsahuje tri úrovne otázok - začiatočník, mierne pokročilý, expert. Ku každej otázke sú štyri možné odpovede, z ktorých si užívateľ vyberá správnu odpoveď. Užívateľ si v úvodnom formulári zvolí kritériá na výber otázok. Na výber sú tri kritériá:

1. kapitoly, z ktorých chce vyberať otázky
2. koľko otázok z každej kapitoly sa má vygenerovať
3. náročnosť otázok.

Podľa týchto kritérií sa následne vygeneruje test, ktorý ak užívateľ vyplní v stanovenom časovom limite, zobrazí sa mu výsledok testu, ak to v časovom limite nestihne, nezodpovedané otázky sa považujú za nesprávne a test sa vyhodnotí.

Po preštudovaní si problematiky pokročilých častí počítačového videnia sme dospeli k názoru, že forma testu by nebola pre našu zbierku vhodná, nakoľko táto oblasť je už o niečo náročnejšia a vyžaduje si väčšinou obsiahlejšie odpovede. A preto sme sa rozhodli, že zvolíme klasickú formu zbierky.

Jedným zo zdrojov, odkiaľ sme čerpali námet na otázky bol [1]. Ďalším, nemenej dôležitým bol [3]. Špeciálne pre kapitoly Rozpoznávanie obrazu a Porozumenie obrazu sme čerpali z [11].

Časť Otázky sme sa snažili vždy zamerať na najdôležitejšiu časť danej problematiky, aby otázky obsahovali definície, základnú charakteristiku konkrétneho problému, zadefinovanie nejakého rozdelenia, jednoduché porovnanie... Tu nebol nejaký väčší problém pri tvorbe, nakoľko táto oblasť je pomerne rozsiahla.

Časť Príklady nám naopak dala viacej námahy pri jej tvorbe. Snažili sme sa príklady konštruovať tak, aby bolo potrebné pri ich riešení využiť danú problematiku, ale nie len formou definície. Väčšinou sa jednalo o vytvorenie nejakého programu, nájdenie nejakého výsledku použitím popísaného algoritmu a pod.

Nie vždy bolo jednoduché nájsť presnú hranicu medzi týmito dvomi časťami, ale dúfame, že

sa nám to aspoň sčasti podarilo.

Celkovo má naša zbierka 293 príkladov, pričom počet v jednotlivých kapitolách je nasledovný:

1. Dátové štruktúry pre analýzu obrazu
 - 1.A. Otázky 17
 - 1.B. Príklady 9
2. Pokročilé techniky predspracovania obrazu
 - 2.A. Otázky 20
 - 2.B. Príklady 11
3. Pokročilé techniky segmentácie
 - 3.A. Otázky 24
 - 3.B. Príklady 14
4. Pokročilé techniky reprezentácie tvaru a jej popisu
 - 4.A. Otázky 27
 - 4.B. Príklady 14
5. Rozpoznávanie obrazu
 - 5.A. Otázky 37
 - 5.B. Príklady 15
6. Porozumenie obrazu
 - 6.A. Otázky 45
 - 6.B. Príklady 20
7. Textúry
 - 7.A. Otázky 30
 - 7.B. Príklady 10

2.4 Proces riešenia

Naším cieľom bolo vytvoriť zbierku riešených príkladov, preto neodmysliteľnou časťou procesu tvorby zbierky bolo zvoliť vhodnú metodiku výberu riešených príkladov. Riešené príklady by mali byť akousi ukážkou, čo sa od čitateľa – študenta očakáva, zároveň

akýmsi návodom, ako riešiť podobné úlohy. A v neposlednom rade by tiež mali byť možnosťou kontroly vlastných riešení.

Prihliadnuc na tieto aspekty a taktiež na počet otázok v jednotlivých kapitolách sme v každej kapitole vyriešili 2 - 5 otázok a 2 - 4 príklady. Počet riešení v jednotlivých kapitolách je nasledovný:

1. Dátové štruktúry pre analýzu obrazu
 - 1.A. Otázky 2
 - 1.B. Príklady 2
2. Pokročilé techniky predspracovania obrazu
 - 2.A. Otázky 3
 - 2.B. Príklady 3
3. Pokročilé techniky segmentácie
 - 3.A. Otázky 3
 - 3.B. Príklady 3
4. Pokročilé techniky reprezentácie tvaru a jej popisu
 - 4.A. Otázky 4
 - 4.B. Príklady 3
5. Rozpoznávanie obrazu
 - 5.A. Otázky 4
 - 5.B. Príklady 4
6. Porozumenie obrazu
 - 6.A. Otázky 5
 - 6.B. Príklady 3
7. Textúry
 - 7.A. Otázky 3
 - 7.B. Príklady 2

Našou snahou pri výbere riešených príkladov bolo pokryť čo najväčšie množstvo typov príkladov, či už išlo o definície, popísanie nejakého problému, praktické riešenia, vytvorenie nejakej gramatiky a pod.

Celkovo sme spolu vyriešili 43 príkladov.

Toto množstvo pokladáme za dostačujúce, vzhľadom na celkový počet otázok zbierky, nakoľko chceme nechať priestor aj pre samotného čitateľa.

Pri riešení príkladov sme čerpali z viacerých zdrojov, podobne ako pri tvorbe samotných príkladov.

Implementačné príklady sme riešili pomocou už spomínaného programu ImageJ, o ktorom bude viac v nasledujúcej kapitole.

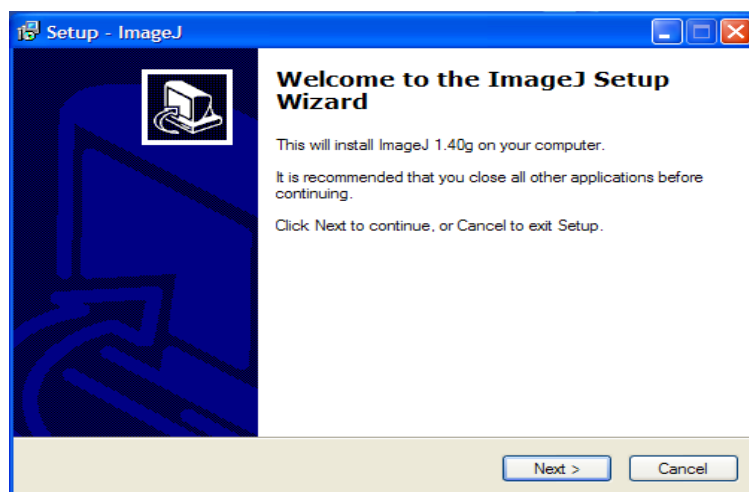
3 ImageJ

3.1 Základná charakteristika

ImageJ je voľne dostupný program na spracovanie obrazu v Jave. Je spustiteľný aj ako online applet, aj ako offline aplikácia pod operačnými systémami Windows, Mac OS, Mac OS X a Linux. ImageJ umožňuje zobrazit', upravovať, analyzovať, spracovať, ukladať a tlačiť 8, 16 a 32-bitové obrazy. Dokáže čítať viacero formátov napr.: TIFF, GIF, JPEG, BMP, DICOM, FITS a tiež nespracované dáta. Podporuje štandardné funkcie spracovania obrazu ako sú úprava kontrastu, ostrenie, vyhladzovanie, detekcia hrán a mediánové filtrovanie. Umožňuje prerátať plochu, pixely, merať vzdialenosti a uhly, ekvalizáciu histogramu. Jeho súčasťou sú aj geometrické transformácie ako zmena mierky (32:1 alebo 1:32), rotácia a prevrátenie obrazu. ImageJ umožňuje otvorenie ľubovoľného počtu okien. Taktiež sú k dispozícii priestorová kalibrácia, kalibrácia hustoty a úrovne šede. Výhodou programu ImageJ je jeho rozšíriteľnosť pomocou tzv. Java pluginov. Pluginy je možné vytvoriť použitím vstavaného editora a Java kompilátora. Tento fakt umožňuje užívateľovi vyriešiť akýkoľvek problém z oblasti spracovania obrazu. ImageJ obsahuje aj viac ako 150 príkladov rôznych pluginov [7].

3.2 Inštalácia

Na stránke [7] v časti Download asi môžeme stiahnuť ImageJ pre rôzne operačné systémy, dokumentáciu i zdrojový kód programu. Na spustenie programu je potrebných 64 MB pamäte, pri práci s veľkými obrázkami je ale lepšie 256 MB a viac. Po spustení ImageJSetup sa nám otvorí inštaláčne okno.

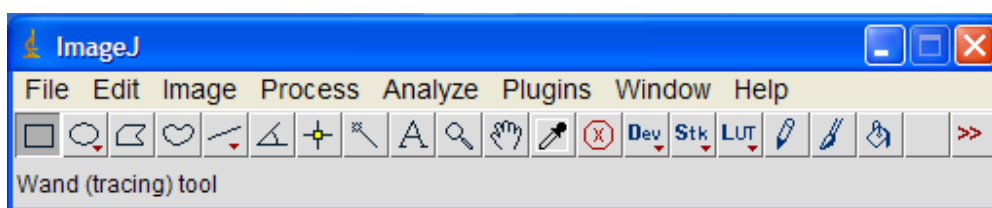


Obr.3.1 Inštalácia

V ďalšom kroku potvrdíme cieľový adresár (poprípade si môžeme zvoliť iný), vyberieme si, či chceme vytvoriť priečinok v menu štart, ikonu na ploche. Po kontrole údajov dokončíme inštaláciu tlačidlom *Install* a nakoniec tlačidlom *Finish*.

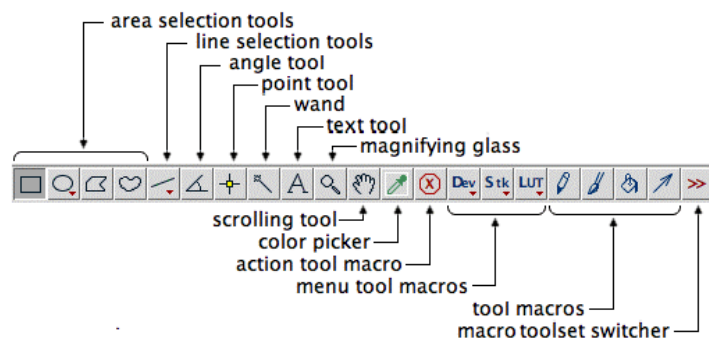
3.3 Prostredie

ImageJ (obr.3.2) obsahuje štyri základné panely: menu, panel nástrojov, panel stavov a panel činností.



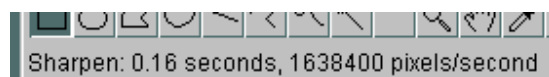
Obr.3.2 ImageJ

V hornej časti okna sa nachádza menu, pod ktorým je panel nástrojov (obr.3.3). Tento obsahuje nástroje na označenie častí obrázku, zväčšovanie a zmenšovanie obrázku, zmenu farby nástrojov kreslenia.



Obr.3.3 Panel nástrojov

Panel stavov (obr.3.4) je na začiatku práce neaktívny, aktivuje sa až po stlačení nejakého nástroja z panelu nástrojov. Zobrazujú sa tu aktuálne súradnice kurzoru a ak máme spustenú nejakú operáciu, zobrazuje čas jej trvania a procesnú rýchlosť.



Obr.3.4 Panel stavov

Posledným zo štyroch vyššie uvedených panelov je panel činností (obr.3.5), ktorý udáva v percentách, koľko z danej činnosti sa už vykonalo. Ak nejaká operácia trvá kratšie ako sekundu, tento sa ani nezobrazí.



Obr.3.5 Panel činností

3.4 Ako riešiť príklady s ImageJ

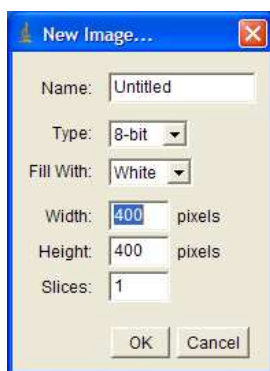
Pri riešení príkladov v programe ImageJ môžeme využiť buď priamo funkcionality tohto programu, môžeme použiť niektorý z doteraz vytvorených pluginov, alebo si môžeme vytvoriť plugin nový.

Prvým príkladom si ukážeme, ako sa dajú využiť funkcie programu ImageJ. Uvažujme príklad z našej zbierky:

4.B.13 Vytvorte program, ktorý nájde konvexný obal jednoduchého mnohoúhelníka.

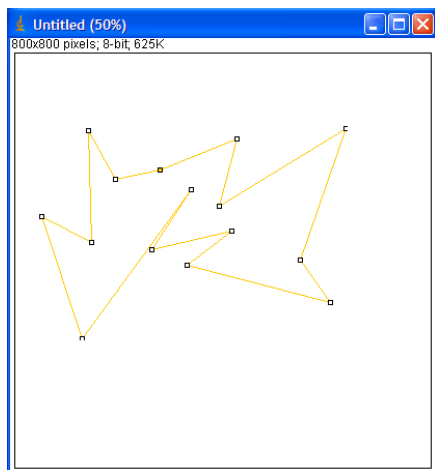
Riešenie:

V prvom kroku sme zistili, že program má už v sebe implementovanú funkciu na hľadanie konvexného obalu mnohoúhelníka. Vytvorili sme si teda nový obrázok (*File -> New -> Image*). Pred vytvorením sa nám otvorilo okno (obr.3.6), kde sme si zvolili názov obrázku, veľkosť, typ...

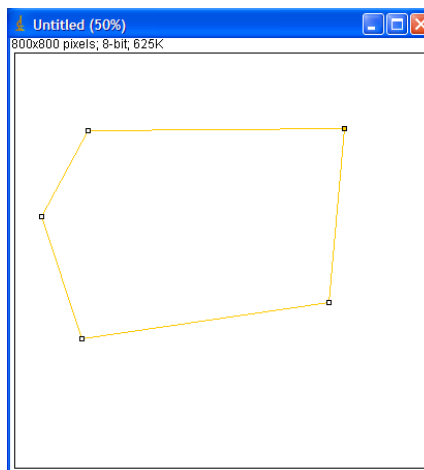


Obr.3.6 Nový obrázok

Ďalej sme využili jeden z nástrojov plošnej sekcie panelu nástrojov, konkrétne *Polygon*. Tento vytvorí nepravidelný tvar výberu, vytvorený zo série čiarových segmentov. Vytvorili sme teda nekonvexný mnohoúhelník (obr.3.7), na ktorý sme následne aplikovali funkciu na hľadanie konvexného obalu mnohoúhelníka (*Edit -> Selection -> Convex Hull*). Výsledný konvexný obal je na obr.3.8 a ako môžeme vidieť, získali sme správny výsledok.



Obr.3.7 Nekonvexný mnohoúhelník



Obr.3.8 Konvexný obal mnohoúhelníka

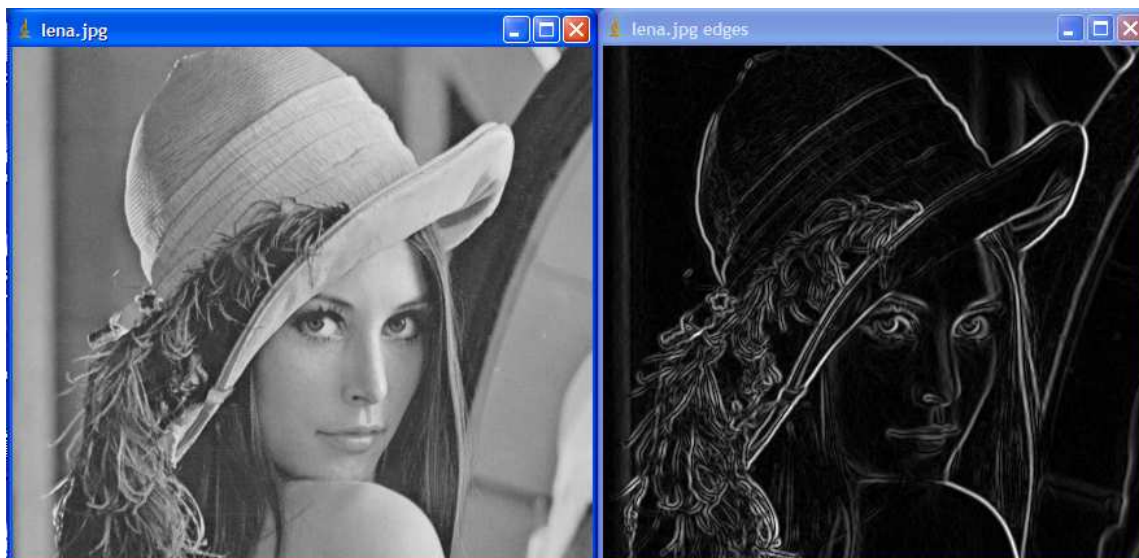
Ďalšou možnosťou je využiť niektorý z už vytvorených pluginov. Veľké množstvo pluginov, alebo odkazov na stránky s rôznymi pluginmi môžeme nájsť na [7], v sekcii Plugins. Sú tu vytvorené rôzne pluginy z oblasti analýzy obrazu, pluginy pre prácu s farbami, rôzne filtre a pod. Zoberme si ďalší príklad z našej zbierky:

2.B.5 Podrobne popíšte Cannyho hranovú detekciu.

Riešenie:

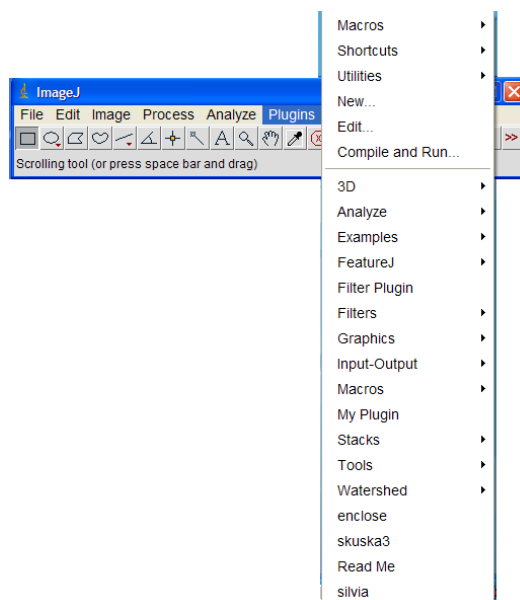
Najlepším popisom daného problému je názorná ukážka, ako sa zmení obraz po aplikovaní danej operácie. Keďže Cannyho hranová detekcia nie je implementovaná v programe, zisťovali sme, či už neexistuje nejaký plugin, ktorý by ju obsahoval. Na [13] sme našli Cannyho hranovú detekciu implementovanú. Do priečinku ImageJ plugins sme si uložili súbory FeatureJ.jar a imagescience.jar. Po vypnutí a následnom zapnutí programu ImageJ sme už mali plugin nainštalovaný.

Otvorili sme si jeden zo vzorových obrázkov (*File -> Open Samples -> Lena*). Keďže ale vstupom pre konkrétnu operáciu je 8-bitový obraz, museli sme najskôr zmeniť typ (*Image -> Typ -> 8-bit*). Následne sme aplikovali operáciu na daný obraz (*Plugins -> FeatureJ -> FJEdges*). Výsledok operácie je znázornený na obr.3.9. (vľavo je pôvodný 8-bitový obraz)



Obr.3.9 Cannyho hranová detekcia

Poslednou možnosťou, ako riešiť príklady, je vytvoriť si vlastný plugin. Táto možnosť už vyžaduje aspoň minimálnu programátorskú skúsenosť. V ďalšej časti ukážeme (na jednoduchom príklade) návod, ako vytvoriť a skompilovať nejaký nový plugin. Najskôr však predstavíme jednotlivé položky menu *Plugins* (obr.3.10).



Obr.3.10 Menu Plugins

Toto menu obsahuje dva základné kódové moduly, ktoré rozširujú funkčnosť ImageJ, makrá a pluginy. Pluginy sú napísané v programovacom jazyku Java a skompilované ako .class súbory, makrá sú napísané v ImageJ Java jazyku a skompilované ako .txt súbory. Pluginy sú rýchlejšie a flexibilnejšie, makrá sú menej náročné na vytvorenie a debugovanie.

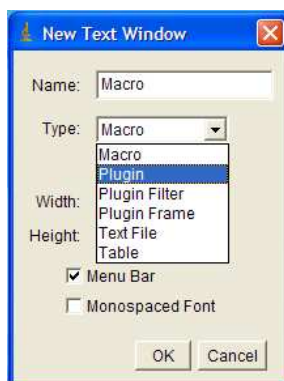
Na prácu s makrami slúži podmenu *Macros*. *Install* slúži na pridanie jedného alebo viacerých už vytvorených makier do menu. *Run* načíta a spustí makro bez jeho otvorenia v ImageJ editore. *Record* zobrazí príkazové okno, kde môžeme zaznamenať postupnosť príkazov programu a následne vytvoriť nové makro. Toto spustíme pomocou *Plugins* -> *Macros* -> *Install*.

Na prácu s pluginmi slúžia *New*, *Edit* a *Compile and Run*. *New* otvorí nové textové okno (obr.3.11) obsahujúce Java zdrojový kód pre jeden z troch typov pluginov podporovaných ImageJ.

PlugIn: Otvára, zachytáva alebo generuje obrázky. Implementuje Plugin rozhranie. Príklady sú na [14].

PlugInFilter: Spracuje aktívny obrázok. Implementuje PlugInFilter rozhrania.

PlugInFrame: Zobrazí neobrázkové okno obsahujúce ovládacie prvky, ako sú napr. tlačidlá.



Obr.3.11 Nové textové okno

Textové okno vytvorené týmto príkazom má dve menu, *File* a *Edit*. Pomocou *Compile and Run* vo *File* menu skompilujeme a spustíme plugin. *Edit* menu neobsahuje Cut/Copy/Paste, ale pomocou klávesových skratiek môžeme tieto funkcie využívať. Názov pluginu musí obsahovať podtržník.

Edit otvára textové okno, ktoré taktiež umožňuje editovať, kompilovať a spúšťať pluginy s tým rozdielom, že nevyžaduje, aby ImageJ bežal na Java Virtual Machine, ktorá obsahuje Java compiler.

Compile and Run kompiluje a spúšťa plugin.

A teraz už vieme všetko na to, aby sme mohli vytvoriť nejaký vlastný plugin. Pre ľahšie pochopenie sme sa rozhodli vytvoriť nejakú jednoduchú operáciu, zvolili sme si operáciu medián (použili sme krížovú masku). Náš postup bol nasledovný:

Vytvorili sme nový plugin (*Plugins* -> *New* -> *PluginFilter*). Tu sme si vytvorili triedu *median_5* a napísali nasledujúci kód:

```
import ij.*;
import ij.process.*;
import ij.gui.*;
import java.awt.*;
import ij.plugin.filter.*;

public class median_5 implements PlugInFilter {
    ImagePlus imp;

    public int setup(String arg, ImagePlus imp) {
        this.imp = imp;
        return DOES_ALL;
    }

    public void run(ImageProcessor ip) {
        if (ip instanceof ByteProcessor)
            median8BitImage(ip);
    }

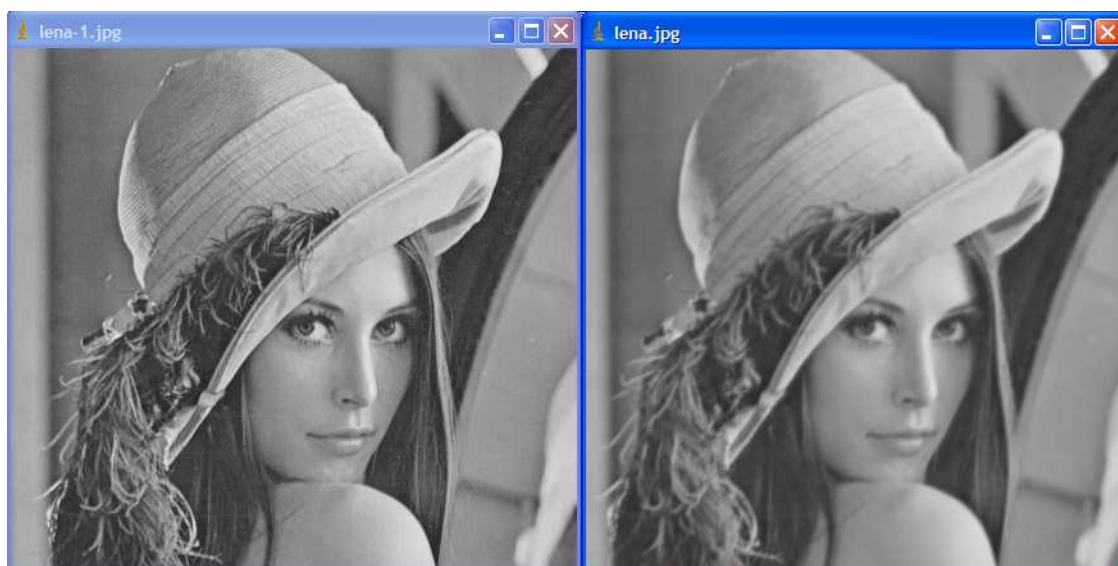
    public void median8BitImage(ImageProcessor ip) {
        byte[] pixels = (byte[])ip.getPixels();
```

```

int width = ip.getWidth();
Rectangle r = ip.getRoi();
int offset, i, a, b, c, d, e, f, g, h, j, k;
for (int y=r.y; y<(r.y+r.height); y++) {
    for (int x=r.x; x<(r.x+r.width); x++) {
        a=ip.getPixel(x, y-1);
        b=ip.getPixel(x, y+1);
        c=ip.getPixel(x-1, y);
        d=ip.getPixel(x+1, y);
        g=ip.getPixel(x, y-2);
        h=ip.getPixel(x, y+2);
        j=ip.getPixel(x-2, y);
        k=ip.getPixel(x+2, y);
        e=ip.getPixel(x, y);
        f=(a+b+c+d+e+g+h+j+k)/9;
        ip.putPixel(x, y, f);
    }
}
}
}

```

Nakoniec sme použili *File -> Compile and Run*. Počas kompilácie sme boli vyzvaní na uloženie, a teda vytvorili sme `median_5.java` súbor v *plugins*. Po vypnutí a následnom zapnutí programu ImageJ sme mali náš plugin už uložený v menu *Plugins*. Následne sme `median5` aplikovali na vzorový obrázok Lena (8-bit). Výsledok je znázornený na obr.3.12 (vľavo je pôvodný 8-bitový obraz).



Obr.3.12 Median5

Ukázali sme teda viacero možností, ako začať riešiť príklady pomocou programu ImageJ. Podali sme návod a veríme, že naše dosiahnuté výsledky budú pre čitateľa – študenta inšpiráciou.

4 Stránka

Neodmysliteľnou súčasťou našej práce je umiestnenie celej zbierky na internete. Umožnili sme tak jej prístupnosť širokej verejnosti.

4.1 Dizajn

V ľavej časti stránky sa nachádza ovládací panel, ktorý slúži na navigáciu v celej zbierke a ktorý pozostáva z trinástich častí (obr.4.1).



Obr.4.1 Stránka

Prvá podáva všeobecné informácie o zbierke, môžete tu nájsť kontakt, kam sa môžete obrátiť v prípade nejasností a pripomienok.

Nasledujúcich sedem častí tvoria samotné kapitoly zbierky, ktoré sú v poradí, o ktorom sme hovorili v druhej kapitole. Pre ľahšiu orientáciu sa v jednotlivých kapitolách zbierky sme vytvorili v každej časti podmenu, ktoré nám vždy zoskupuje po päť otázok.

Ďalšou časťou je ImageJ. Tu podávame informácie o samotnom programe, ako aj o jeho inštalácii. No najdôležitejšou časťou je návod, ako riešiť príklady zbierky pomocou tohto programu.

Časť riešenia obsahuje vyriešené vzorové príklady každej kapitoly, či už ide o definície, popísanie nejakého problému, alebo využitie už spomínaného programu.

Pre ľahšiu orientáciu sme vytvorili register kľúčových slov celej zbierky, ktorý je vždy hypertextovým odkazom na prvú otázku s danou problematikou.

Časť komentáre slúži na rôzne pripomienky návštevníkov, poprípade na diskusiu k jednotlivým otázkam zbierky.

Poslednou časťou je download, kde sa dajú stiahnuť zadania i riešenia príkladov.

4.2 Technológie

Pri tvorbe internetovej stránky sme použili nasledujúce technológie a štandardy: HTML, XHTML, CSS, PHP, JavaScript.

Na zjednotenie celkového vzhľadu stránky, ako aj samotnej tvorby sme využili HTML, XHTML a CSS. Správne zobrazovanie menu nám zabezpečuje JavaScript. PHP sa stará o spracovanie celého kódu stránky. Pri tvorbe ankety sme využili predprogramovaný formulár [16].

HTML (HyperText Markup Language) je značkovací jazyk slúžiaci na tvorbu www stránok. Ide o množinu kódovacích štandardov a pravidiel pre vytváranie stránok. Zodpovednosť za špecifikáciu má W3C (World Wide Web Consortium) [17].

HTML štandard je postupne nahrádzaný štandardom XHTML (Extended HyperText Markup Language), ktorý by mal zjednocovať štruktúru zdrojového kódu stránok. Medzi základné pravidlá kódovania XHTML patria:

- a) XML rozlišuje veľkosť písmen (prvky by mali byť písané malými písmenami)
- b) všetky prvky musia byť ukončené ukončovacím tagom
- c) hodnoty atribútov musia byť v úvodzovkách [18].

Na zjednotenie nastavení vzhľadu stránok sme využili CSS (Cascading Style Sheets). Tieto uľahčujú vytváranie a spracovanie grafického dizajnu stránky z jedného miesta, oddeľujú obsah od vzhľadu.

Záver

Naša práca je ďalším rozšírením série interaktívnych učebníc a zbierok v oblasti spracovania obrazu a počítačového videnia. V tejto práci sa nám podarilo vytvoriť ucelenú zbierku príkladov z oblasti počítačového videnia v slovenskom jazyku, ktorá má byť pomôckou pri výučbe predmetu Počítačové videnie na Katedre aplikovanej informatiky. Ďalším prínosom je vyriešenie vzorových príkladov, ktoré slúžia na kontrolu výsledkov, ale aj ako návod na riešenie a inšpiráciu. Poskytli sme čitateľovi – študentovi podrobný návod, ako začať riešiť príklady pomocou voľne dostupného programu ImageJ. Celú zbierku sme umiestnili na internete [15], čím sme uľahčili jej dostupnosť širokej verejnosti.

Zbierku je možné rozšíriť o ďalšie príklady v závislosti od sylabov daného predmetu.

Zoznam použitej literatúry:

- [1] Sonka, Hlavac, Boyle: Image Processing, Analysis and Machine Vision, 2nd edition, International Thomson Publishing Inc., 1999, USA
- [2] Maťková, B.: Interaktívne testy v počítačovej grafike, Bratislava, 2000, <http://pg.netgraphics.sk>
- [3] Ftáčnik, M.: Homepage, <http://www.sccg.sk/~ftacnik/>
- [4] Ftáčnik, M.: Prednášky z počítačového videnia, 2005, <http://www.sccg.sk/~ftacnik/compvis.htm>
- [5] Štugel, J., Výučba počítačovej grafiky, Bratislava, 2000, <http://www.netgraphics.sk/>
- [6] Blázsovits, G.: Interaktívna učebnica spracovania obrazu, Bratislava, 2003, <http://dip.sccg.sk>
- [7] ImageJ, <http://rsb.info.nih.gov/ij/>
- [8] HTML, <http://www.htmlhelp.com>
- [9] HTML, <http://www.jakpsatweb.cz>
- [10] Obrázky, <http://rsb.info.nih.gov/ij/images/>
- [11] Birošová, J.: Interaktívna učebnica rozpoznávania obrazcov, Bratislava, 2007, <http://www.sprite.edi.fmph.uniba.sk/~julka/>
- [12] <http://w3schools.com>

- [13] <http://www.imagescience.org/meijering/software/featurej/index.html>
- [14] <http://rsb.info.nih.gov/ij/plugins/steps.html>
- [15] <http://strungova.yw.sk/>
- [16] <http://vsevjednom.cz/>
- [17] <http://www.w3.org>
- [18] Stauffer, T.: Tvorba webových stránek pro úplné začátečníky, SoftPress, Praha, 2003.

PRÍLOHY

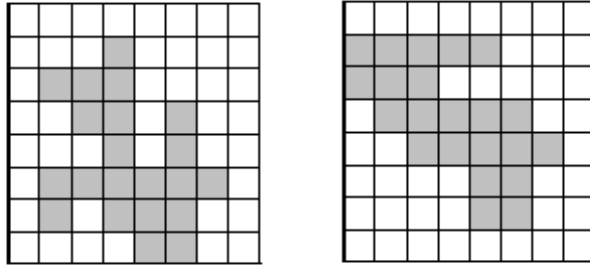
A Zbierka príkladov

1. Dátové štruktúry pre analýzu obrazu

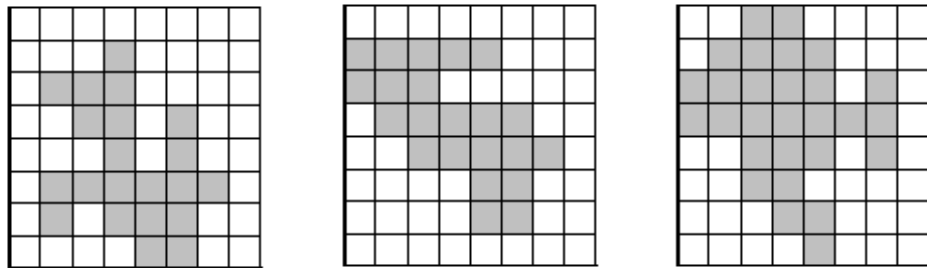
- 1.A.1** Na koľko úrovni a aké vieme rozdeliť reprezentáciu obrazových dát?
- 1.A.2** Ako je to s hranicou medzi jednotlivými reprezentáciami?
- 1.A.3** Ktoré tradičné dátové štruktúry poznáte? Vymenujte aspoň tri.
- 1.A.4** Akými maticami reprezentujeme binárne obrazy?
- 1.A.5** Akými maticami reprezentujeme multispektrálne obrazy?
- 1.A.6** Na čo slúžia reťazce?
- 1.A.7** Pre aké dáta sú vhodné reťazce?
- 1.A.8** Popíšte, ako sa vytvára reťazcový kód.
- 1.A.9** Popíšte, ako sa vytvára RL-kód.
- 1.A.10** Definujte graf susedností oblastí.
- 1.A.11** Popíšte princíp relačných štruktúr.
- 1.A.12** Vymenujte aspoň dve hierarchické dátové štruktúry.
- 1.A.13** Aké pyramidové štruktúry poznáte?
- 1.A.14** Definujte M-pyramídy.
- 1.A.15** Definujte T-pyramídy.
- 1.A.16** Definujte kvadrantové stromy.
- 1.A.17** Popíšte listové kódy.

1.B.1 Charakterizujte jednotlivé úrovně reprezentácie obrazových dát.

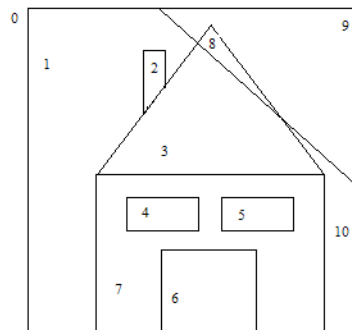
1.B.2 Určite reťazcový kód pre nasledujúce obrázky.



1.B.3 Určite RL - kód pre nasledujúce obrázky



1.B.4 Nakreslite graf susedností pre nasledujúci obrázok.



1.B.5 Implementujte RL-kódovanie pre binárne obrázky.

1.B.6 Implementujte RL-kódovanie tak, aby to bolo vhodné nie len pre binárne obrázky.

1.B.7 Uved'te aspoň dva príklady, na ktorých ukážete výhodu RL-kódu oproti zápisu oblasti pomocou vymenovania jej prvkov. Pre každý príklad uved'te kompresný pomer.

1.B.8 Napíšte program, ktorý počíta T-pyramídu obrazu.

1.B.9 Napíšte program, ktorý odvodzuje reprezentáciu obrazu kvadrantovými stromami.

2. Pokročilé techniky predspracovania obrazu

- 2.A.1 Vymenujte štyri základné typy metód predspracovania a zarad'te metódu Zero-crossing.
- 2.A.2 Popíšte, ako funguje metóda Zero-crossing.
- 2.A.3 Akú úlohu hrá parameter σ pri použití tejto metódy?
- 2.A.4 Popíšte, akým spôsobom vieme vyrátať druhú deriváciu robustnejšie.
- 2.A.5 Vysvetlite význam skratiek LoG a DoG a popíšte, ako ich vieme vypočítať.
- 2.A.6 Vysvetlite, prečo je LoG lepší hranový detektor ako Laplacov.
- 2.A.7 Vymenujte štyri základné typy metód predspracovania a zarad'te metódu škálovania.
- 2.A.8 Vysvetlite myšlienku škálovania v spracovaní obrazu.
- 2.A.9 Kedy, podľa Marrovej tézy, zodpovedá hrana obrazu reálnej hrane?
- 2.A.10 Aký špecifický prístup k optimálnej škáre poznáte?
- 2.A.11 Pre aké hrany je optimálny Cannyho hranový operátor?
- 2.A.12 Aký je rozdiel medzi metódou Zero-crossing a Cannyho hranovou detekciou, čo sa týka orientácie?
- 2.A.13 Na akej myšlienke sú založené parametrické hranové modely?
- 2.A.14 Ako sa nazýva najjednoduchší parametrický hranový model a aké funkcie používa?
- 2.A.15 Popíšte jeden pixel multispektrálneho obrazu.
- 2.A.16 Z akých bodov pozostáva adaptívne okolie?
- 2.A.17 Definujte jadrový obrazový bod.
- 2.A.18 Definujte redundantný jadrový obrazový bod.
- 2.A.19 Na potlačenie akého šumu je vhodná metóda adaptívneho okolia?
- 2.A.20 Uved'te, v čom je použitie adaptívneho okolia výhodnejšie a v čom nevýhodnejšie ako použitie okolia s pevným rozmerom.

- 2.B.1** Vytvorte program počítajúci Zero-crossing. Použite LoG definíciu.
- 2.B.2** Vytvorte program počítajúci Zero-crossing. Použite DoG definíciu.
- 2.B.3** Porovnajzte výsledky predchádzajúcich dvoch príkladov a vysvetlite, prečo sú hranice nesúvislé.
- 2.B.4** Vymenujte tri požiadavky, na ktorých je založené kritérium optimality Cannyho hranového operátora.
- 2.B.5** Podrobne popíšte Cannyho hranovú detekciu.
- 2.B.6** Uved'te bikubický fazetový model a popíšte, ako sa odhadujú parametre c_i .
- 2.B.7** Popíšte, ako môžeme určiť hranu, ak poznáme fazetové parametre pre každý obrazový bod.
- 2.B.8** Existuje viacero spôsobov na detekciu hrany v multispektrálnom obraze. Popíšte aspoň dva z nich.
- 2.B.9** Vysvetlite princíp potlačenia šumu v adaptívnom okolí.
- 2.B.10** Vysvetlite princíp ekvalizácie histogramu v adaptívnom okolí.
- 2.B.11** Vysvetlite princíp zväčšenia kontrastu v adaptívnom okolí.

3. Pokročilé techniky segmentácie

- 3.A.1** Vysvetlite pojmy úplnej a čiastočnej segmentácie.
- 3.A.2** Na aké tri skupiny možno rozdeliť metódy segmentácie?
- 3.A.3** Čo je hlavným segmentačným problémom?
- 3.A.4** Aké tri typy hranice oblasti poznáte? Popíšte ich.
- 3.A.5** V čom spočíva segmentácia založená na hranách a aké sú jej najväčšie problémy?
- 3.A.6** Uvažujte heuristický algoritmus hľadania hrán. Vysvetlite, ako presnosť odhadu $h(x_i)$ ceny cesty z vrcholu x_i do koncového vrcholu ovplyvňuje priebeh hľadania. Uved'te, za akých podmienok je zaručená optimalita algoritmu. Popíšte, ako odhad zhora, odhad zdola a presný odhad ovplyvňuje rýchlosť nájdenia cesty.
- 3.A.7** Aká je hlavná myšlienka princípu optimality?
- 3.A.8** Ako je využitá myšlienka princípu optimality v dynamickom programovaní?
- 3.A.9** Popíšte, na detekciu akých objektov je vhodná Houghova transformácia.
- 3.A.10** S použitím polárnej reprezentácie vysvetlite hlavný princíp Houghovej transformácie na detekciu priamok.
- 3.A.11** Vysvetlite, prečo je polárna reprezentácia
- $$s = x \cos \theta + y \sin \theta$$
- vhodnejšia na detekciu priamok ako klasická smernicová reprezentácia
- $$y = kx + q.$$
- 3.A.12** Vysvetlite, prečo informácia o smere hrán môže zvýšiť rýchlosť Houghovej transformácie.
- 3.A.13** Na čom je založené zisťovanie hranice použitím hraničnej lokálnej informácie?
- 3.A.14** Popíšte, kedy je rekonštrukcia oblastí z hraníc jednoduchá a kedy to nemusí byť úplne triviálne (uved'te dôvod).
- 3.A.15** Aké je kritérium úplnej segmentácie a aké sú podmienky maximálnej homogenity pri segmentácii založenej na oblastiach?
- 3.A.16** Vysvetlite hlavný princíp segmentácie na princípe povodí (watershed).
- 3.A.17** Aké tri základné prístupy segmentácie založenej na hranách existujú?
- 3.A.18** Vysvetlite, prečo má segmentácia na princípe povodí (watershed) tendenciu k presegmentovaniu obrazu.

3.A.19 Uved'te príklady, kedy je vhodné použiť segmentáciu porovnávania so vzorom.

Na čom je založená najlepšia zhoda?

3.A.20 Ako možno definovať kritériá zhody pri porovnávaní so vzorom?

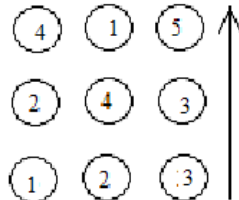
3.A.21 Čo je dôsledkom neoptimálneho nastavenia parametrov u obrazov segmentovaných narastaním oblastí?

3.A.22 Aké dve grafovo - založené metódy detekcie hrán poznáte?

3.A.23 Vysvetlite princíp simultánneho hľadania hranice.

3.A.24 Vysvetlite princíp optimálnej detekcie povrchu.

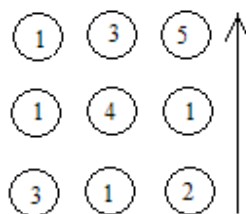
3.B.1 S použitím algoritmu A na hľadanie v grafe, nájdite optimálnu cestu cez graf na nasledujúcom obrázku, kde čísla vo vnútri kruhov určujú cenu vrcholu, pričom uvažujte troch možných nasledovníkov vrcholu. Ukážte všetky kroky hľadania grafu, vrátane dátových štruktúr v každom kroku hľadania.



3.B.2 Predpokladajte optimálnu hranicu zistenú použitím algoritmu A na hľadanie v grafe, rovnomerný graf s ohodnotením každého vrcholu. Uvažujme optimálnu hranicu určenú ako cestu minimálnej ceny. Diskutujte, ako zmeny cien vrcholov uvedené nižšie ovplyvnia výslednú hranicu (či majú alebo nemajú vplyv na optimálnu hranicu) a ako sa zmení počet expandovaných vrcholov (či zostane rovnaký, či sa zmenší alebo zväčší). Uvažujte nasledujúce možnosti:

- Cena všetkých vrcholov sa zvýši o konštantu.
- Cena všetkých vrcholov sa zníži o konštantu, pričom sa žiadna cena nestane zápornou.
- Cena všetkých vrcholov sa zníži o konštantu, pričom sa niektoré ceny stanú zápornými.

3.B.3 S použitím dynamického programovania nájdite optimálnu cestu cez graf na nasledujúcom obrázku, kde čísla vo vnútri kruhov určujú cenu vrcholu, pričom uvažujte troch možných nasledovníkov vrcholu. Ukážte všetky kroky hľadania v grafe, vrátane dátových štruktúr v každom kroku hľadania.



3.B.4 Predpokladajte optimálnu hranicu zistenú použitím prehľadávania grafu založenom na dynamickom programovaní, rovnomerný graf s ohodnotením každého vrcholu. Uvažujme optimálnu hranicu určenú ako cestu minimálnej ceny. Diskutujte,

ako zmeny cien vrcholov uvedené nižšie ovplyvnia výslednú hranicu (či majú alebo nemajú vplyv na optimálnu hranicu) a ako sa zmení počet expandovaných vrcholov (či zostane rovnaký, či sa zmenší alebo zväčší). Uvažujte nasledujúce možnosti:

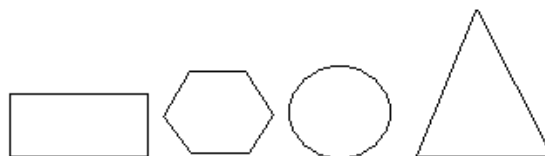
- a) Cena všetkých vrcholov sa zvýši o konštantu.
- b) Cena všetkých vrcholov sa zníži o konštantu, pričom sa žiadna cena nestane zápornou.
- c) Cena všetkých vrcholov sa zníži o konštantu, pričom sa niektoré ceny stanú zápornými.

3.B.5 Popíšte hlavné rozdiely medzi zisťovaním hranice pomocou algoritmu A grafového prehľadávania a dynamického programovania. Vysvetlite, prečo je dynamické programovanie často rýchlejšie ako algoritmus A.

3.B.6 Vysvetlite hlavné konceptuálne prístupy v segmentácii založenej na hranách a segmentácii založenej na oblastiach. Sú tieto dva prístupy duálne? Ak sa segmentácia založená na hranách a založená na oblastiach uplatnia na rovnaké obrazové dáta, budú výsledné segmentácie identické? Uvažujte ideálne bezšumové dáta a skutočné digitálne obrazy z reálneho sveta.

3.B.7 Uvažujte o priamke tvorenej nasledujúcimi zašumenými bodmi v obrazovom priestore: $(0,0)$, $(1,2)$, $(2,4)$, $(3,6)$. Nájdite priamku s použitím Houghovej transformácie. Ukážte svoj postup.

3.B.8 S použitím polárnej reprezentácie priamok nájdite Houghovu transformáciu čiar objektov na nasledujúcom obrázku (ukážte Houghov priestor pre každý objekt).



3.B.9 Popíšte dve možnosti určenia hranice a uveďte, ktorá je kedy výhodnejšia.

3.B.10 Popíšte možnosti rekonštrukcie oblastí z hraníc.

3.B.11 Implementujte segmentáciu na princípe povodí.

3.B.12 Popíšte algoritmus porovnávania so vzorom.

3.B.13 Implementujte algoritmus detekcie povrchu.

3.B.14 Vytvorte jednoduchý príklad demonštrujúci, že opísaná detekcia povrchu môže určiť neoptimálnu plochu.

4. Pokročilé techniky reprezentácie tvaru a jej popisu

4.A.1 Geometrická reprezentácia hranice je založená na viacerých geometrických vlastnostiach popisovanej oblasti. Napíšte aspoň tri takéto vlastnosti.

4.A.2 Definujte

- a) energiu zakrivenia
- b) signatúru
- c) chrbticové rozdelenie.

4.A.3 Metódy popisu tvaru sa delia podľa viacerých kritérií (napr. podľa vstupnej reprezentácie - hranica vs. oblasť). Vymenujte ďalšie možnosti delenia a zarad'te Fourierove popisy tvaru.

4.A.4 Na aké krivky je možné použiť Fourierovu transformáciu?

4.A.5 Definujte Fourierove popisy hranice použitím T_n .

4.A.6 Definujte Fourierove popisy hranice použitím S_n .

4.A.7 Tvar možno reprezentovať ako postupnosť segmentov s určitými vlastnosťami. Popíšte popis hranice použitím postupností segmentov.

4.A.8 Definujte pojmy:

- a) B-splajny
- b) radiaci polygón
- c) bázická funkcia.

4.A.9 Popíšte interpoláciu B-splajnovej krivky.

4.A.10 Vymenujte iné popisy tvaru založené na hranách.

4.A.11 Čo predstavujú tvarové invarianty?

4.A.12 Presne zadefinujte invariant (aj s popisom jednotlivých častí).

4.A.13 Vymenujte niekoľko príkladov invariantov.

4.A.14 Vysvetlite rozdiel medzi lokálnymi a globálnymi invariantmi.

4.A.15 Napíšte dve základné metódy vytvorenia grafu podoblastí.

4.A.16 Jednoduché geometrické objekty používajú geometrické vlastnosti na popis regiónov. Vymenujte tieto vlastnosti.

4.A.17 Graf oblasti je založený na kostre regiónu a prvým krokom je konštrukcia kostry. Vymenujte štyri prístupy ku konštrukcii kostry regiónu.

4.A.18 Určite výhody a nevýhody grafovej reprezentácie založenej na kostre regiónu.

- 4.A.19** Popíšte vytvorenie kostry pomocou stenčovania.
- 4.A.20** Stenčovacie procedúry často používajú na vytvorenie kostry symetrickej transformácii súradníc. Zdefinujte kostru pri tejto transformácii.
- 4.A.21** Popíšte symetrickej transformácii súradníc.
- 4.A.22** Definujte tvarové primitíva.
- 4.A.23** Popíšte tvorbu grafu pri rozklade oblastí.
- 4.A.24** Popíšte princíp popisu tvaru s použitím dekompozície grafu.
- 4.A.25** Čo a ako reprezentuje graf susedností oblastí?
- 4.A.26** Čo reprezentujú tvarové triedy?
- 4.A.27** Čo je široko používanou reprezentáciou variácií tvaru vo vnútri triedy?

- 4.B.1** Napíšte funkciu, ktorá určí popis hranice Fourierovým popisom hranice použitím T_n .
- 4.B.2** Napíšte funkciu, ktorá určí popis hranice Fourierovým popisom hranice použitím S_n .
- 4.B.3** Porovnajte výsledky predchádzajúcich dvoch bodov.
- 4.B.4** Na určenie segmentov sa používajú rôzne metódy, vymenujte ich a každú popíšte.
- 4.B.5** Koľko je nenulových základných funkcií pre $s \in (i, i+1)$? Uveďte ich.
- 4.B.6** Popíšte aspoň dva iné popisy tvaru založené na hranách.
- 4.B.7** Popíšte invariant Podiel priesečníku.
- 4.B.8** Popíšte invariant Systém priamok alebo bodov.
- 4.B.9** Popíšte invarianty odvodené z roviny kónusov.
- 4.B.10** Podrobne popíšte aspoň tri geometrické vlastnosti na popis regiónov.
- 4.B.11** Určite symetrickú transformáciu súradníc kružnice, štvorca, obdĺžnika a trojuholníka.
- 4.B.12** Uveďte príklad, v ktorom je výhodné popísať tvar rozdielnych objektov pomocou kostry.
- 4.B.13** Vytvorte program, ktorý nájde konvexný obal jednoduchého mnohoúhelníka.
- 4.B.14** Pomocou relatívnej polohy (*vľavo od*, *medzi*, *blízko k*) dvoch oblastí sa často vyjadrujú vlastnosti dôležité v procese popisu. Definujte tromi spôsobmi reláciu *vľavo od*.

5. Rozpoznávanie obrazu

- 5.A.1** Na čo sa požíva rozpoznávanie obrazcov?
- 5.A.2** Čo je nevyhnutné pre každé rozpoznávanie? Popíšte hierarchiu.
- 5.A.3** Definujte syntax reprezentácie znalostí.
- 5.A.4** Definujte sémantiku reprezentácie znalostí.
- 5.A.5** Definujte reprezentáciu.
- 5.A.6** Pri reprezentácii znalostí sa využívajú techniky umelej inteligencie. Vymenujte ich.
- 5.A.7** Definujte objekt. Ako nazývame množinu všetkých objektov so spoločnými príznakmi?
- 5.A.8** V čom spočíva štatistické rozpoznávanie obrazcov?
- 5.A.9** Definujte nasledujúce pojmy:
- a) obrazec
 - b) trieda
 - c) klasifikátor
 - d) priestor príznakov.
- 5.A.10** Popíšte hlavné kroky rozpoznávania obrazcov.
- 5.A.11** Definujte:
- a) separabilné triedy
 - b) lineárne separabilné triedy.
- 5.A.12** Definujte neurón a popíšte vstup do neurónu a výstup neurónu.
- 5.A.13** O čom hovorí prenosová funkcia neurónu? Uveďte príklad.
- 5.A.14** Popíšte princíp neurónovej siete.
- 5.A.15** Popíšte tri príklady využitia neurónovej siete.
- 5.A.16** Vysvetlite princíp učenia dopredných sietí.
- 5.A.17** Nakreslite schematický diagram dopredných sietí.
- 5.A.18** Na čo sa používa algoritmus spätného posunu? Vysvetlite jeho hlavné body.
- 5.A.19** Aký je dôvod použitia konštanty rýchlosti pohybu v algoritme spätného posunu?
- 5.A.20** Vysvetlite, v čom spočíva význam samoorganizujúcich sa sietí.
- 5.A.21** Ako sa nazýva najznámejší typ samoorganizujúcich sa sietí?

5.A.22 Nakreslite schematický diagram Hopfieldových neurónových sietí. Diskutujte o rozdieloch v architektúre v porovnaní s doprednými sieťami.

5.A.23 Vysvetlite, ako môžu byť Hopfieldove neurónové siete použité pre rozpoznávanie obrazcov.

5.A.24 Popíšte hlavné kroky syntaktického rozpoznávania.

5.A.25 Formálne definujte:

- a) Graf
- b) Grafový izomorfizmus
- c) Podgrafový izomorfizmus
- d) Dvojnásobný podgrafový izomorfizmus

5.A.26 Popíšte, ako sa rieši podgrafový izomorfizmus.

5.A.27 Popíšte, ako sa dá riešiť dvojnásobný podgrafový izomorfizmus.

5.A.28 Definujte Levenshteinovu vzdialenosť. Vysvetlite jej uplatnenie pri určovaní reťazcovej podobnosti.

5.A.29 V čom spočívajú optimalizačné algoritmy? Aké metódy sú najbežnejším prístupom k optimalizácii?

5.A.30 Vymenujte niekoľko metód (používaných v počítačovom videní) na nájdenie globálneho maxima.

5.A.31 Vysvetlite ideu optimalizácie genetickými algoritmi.

5.A.32 Aké tri operátory sa používajú v genetických algoritmoch?

5.A.33 Aké sú úlohy reprodukcie, kríženia a mutácie v genetických algoritmoch?

5.A.34 S čím v genetických algoritmoch súvisí konvergencia? Ako je to so zostrojením prvotnej populácie?

5.A.35 Vysvetlite ideu optimalizácie založenú na simulovanom žíhaní. Ktoré dve optimalizačné metódy využíva?

5.A.36 Uveďte príklady (z oblasti počítačového videnia), kde sa využíva simulované žíhanie.

5.A.37 Uveďte principiálne, ako by ste mohli využiť optimalizačné algoritmy pre riešenie úlohy rozpoznávania obrazcov.

5.B.1 Popíšte nasledujúce reprezentácie znalostí, uveďte pre každú aspoň jeden príklad.

- a) Popisy (príznamy)
- b) Gramatiky
- c) Predikátová logika
- d) Produkčné pravidlá
- e) Fuzzy logika
- f) Sémantické siete
- g) Rámce, skripty

5.B.2 Porovnajete klasifikačné prístupy použité v štatistickom rozpoznávaní obrazcov a v neurónových sieťach.

5.B.3 Vytvorte program na tréovanie algoritmu spätného posunu a klasifikácie použitím trojvrstvovej doprednej neurónovej siete.

5.B.4 Popíšte funkcionalitu Kohonenových neurónových sietí. Ako môžu byť použité pre nekontrolované rozpoznávanie obrazcov?

5.B.5 Implementujte algoritmus nekontrolovaného učenia Kohonenových príznakových máp.

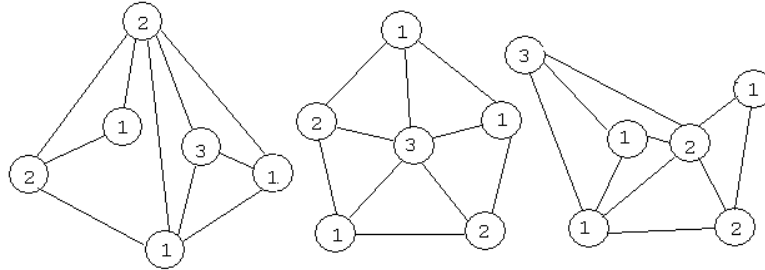
5.B.6 Porovnajete klasifikačné prístupy použité v štatistickom rozpoznávaní obrazcov a v neurónových sieťach.

5.B.7 Implementujte Hopfieldovu neurónovú sieť. Trénujte ju na vzorke čísel 0-9.

5.B.8 Vytvorte gramatiku G , ktorá generuje jazyk $L(G)$ štvorcov ľubovoľnej veľkosti. Primitíva s orientáciou 0° , 90° , 180° a 270° tvoria množinu terminálnych symbolov $V = \{a, b, c, d\}$.

5.B.9 Vytvorte gramatiku G , ktorá generuje jazyk $L(G)$ rovnostranných trojuholníkov. Primitíva s orientáciou 0° , 60° a 120° tvoria množinu terminálnych symbolov $V = \{a, b, c\}$.

5.B.10 Použitím algoritmu grafového izomorfizmu dokážte alebo vyvráťte izomorfizmus grafov na nasledujúcom obrázku.



5.B.11 Určite Levenshteinovu vzdialenosť pre nasledujúce páry reťazcov:

- a) $S_1 = \text{abadcdefacde}$ $S_2 = \text{abadddefacde}$
 b) $S_1 = \text{abadcdefacde}$ $S_3 = \text{abadefaccde}$
 c) $S_1 = \text{abadcdefacde}$ $S_4 = \text{cbadcacdae}$

5.B.12 Použitím optimalizácie genetickými algoritmi určite maximum nasledujúcej funkcie:

$$z(x,y) = 2(1-x)^2 \exp[-x^2 - (y+1)^2] - 5[(x/5) - x^3 - x^5] \exp[-x^2 - y^2] - (1/2) \exp[-(x+1)^2 - y^2]$$

5.B.13 Vymenujte výhody a nevýhody genetických algoritmov a simulovaného žihania v porovnaní optimalizačného prístupu založenom na deriváciách.

5.B.14 V čom spočíva zložitosť rozpoznávania ľudskej reči?

5.B.15 Naznačte spôsob riešenia pre rozpoznávanie izolovaných slov.

6. Porozumenie obrazu

6.A.1 Do ktorej úrovne počítačového videnia by ste zaradili porozumenie obrazu?

Aká je hlavná úloha tejto úrovne?

6.A.2 Vysvetlite rozdiel medzi nižšou a vyššou úrovňou spracovania obrazu.

6.A.3 Vysvetlite, čím je ľudské videnie rozdielne od počítačového. Prečo je problém porozumenia obrazu tak náročný, keď každé malé dieťa 'vie ako na to'?

6.A.4 Vymenujte rôzne stratégie porozumenia obrazu.

6.A.5 Jedno z možných rozdelení stratégií môže byť na paralelné a sekvenčné.

Uvedte, pre ktoré operácie je ktorá stratégia vhodná.

6.A.6 Ďalšie z možných rozdelení stratégií je na hierarchické a nehierarchické.

Rozdeľte množinu hierarchických stratégií.

6.A.7 Vysvetlite hlavné princípy paralelnej stratégie porozumenia obrazu.

6.A.8 Vysvetlite hlavné princípy sekvenčnej stratégie porozumenia obrazu.

6.A.9 Vysvetlite hlavné princípy stratégie hierarchického riadenia porozumenia obrazu.

6.A.10 Vysvetlite hlavné princípy riadiacej stratégie porozumenia obrazu zdola nahor.

6.A.11 Popíšte algoritmus stratégie zdola nahor.

6.A.12 Vysvetlite hlavné princípy riadiacej stratégie porozumenia obrazu zhora nadol.

6.A.13 Porovnajte stratégiu riadenia zdola nahor a zhora nadol.

6.A.14 Vysvetlite hlavné princípy kombinovaných riadiacich stratégií porozumenia obrazu.

6.A.15 Vymenujte štyri zásady, ktoré zahŕňa proces kombinovaných stratégií.

6.A.16 Vysvetlite hlavné princípy nehierarchického riadenia porozumenia obrazu.

6.A.17 Ktorý princíp využíva nehierarchické riadenie na zdieľanie údajov? Na čo slúži a čo obsahuje?

6.A.18 Definujte pojmy:

a) krátkodobá pamäť

b) dlhodobá pamäť

Popíšte rozdiel medzi nimi.

6.A.19 Definujte aktívne kontúry.

- 6.A.20** Napíšte všeobecný výraz pre energiu, ktorá je minimalizovaná počas konvergenzie hada.
- 6.A.21** Napíšte, kde sa využívajú aktívne kontúry.
- 6.A.22** Popíšte metódu narastania hada.
- 6.A.23** Vysvetlite rozdiel medzi hadom a balónom.
- 6.A.24** Vysvetlite, ako možno použiť apriórnu informáciu pri modeloch aktívnej hranice reprezentovaných hadmi.
- 6.A.25** Aký druh informácie môže byť reprezentovaný modelmi rozmiestnení bodov (PDM)?
- 6.A.26** Popíšte techniku PDM.
- 6.A.27** Popíšte, ako sa uplatňujú metódy rozpoznávania obrazu v porozumení.
- 6.A.28** Ako ovplyvní výsledný obraz zašumený vstup pri kontextovej klasifikácii obrazu?
- 6.A.29** Vymenujte niekoľko prístupov kontextovej klasifikácie obrazu.
- 6.A.30** Popíšte algoritmus kontextovej klasifikácie obrazu.
- 6.A.31** Popíšte algoritmus rekurzívnej kontextovej klasifikácie obrazu.
- 6.A.32** V čom spočíva hlavná myšlienka ohodnocovania scény?
- 6.A.33** Ktorú vlastnosť by malo spĺňať ohodnocovanie scény. Uvedte príklady splnenia a nesplnenia tejto vlastnosti.
- 6.A.34** Aké typy ohodnotení scény poznáte? Popíšte a porovnajte ich.
- 6.A.35** Špecifikujte problém ohodnotenia scény.
- 6.A.36** Popíšte algoritmus diskretného ohodnocovania scény.
- 6.A.37** Popíšte algoritmus pravdepodobnostného ohodnocovania scény.
- 6.A.38** Vysvetlite, prečo je diskretné ohodnocovanie scény špecifickým príkladom pravdepodobnostného ohodnocovania.
- 6.A.39** Popíšte algoritmus sémantického spájania oblastí.
- 6.A.40** Popíšte algoritmus genetickej segmentácie a interpretácie obrazu.
- 6.A.41** Definujte Markove modely.
- 6.A.42** Definujte m-usporiadaný Markov model.
- 6.A.43** Definujte skryté Markove modely.
- 6.A.44** Definujte m-usporiadaný skrytý Markov model.

6.A.45 Definujte v súvislosti so skrytými Markovými modelmi:

- a) ohodnotenie
- b) dešifrovanie
- c) učenie.

- 6.B.1** Špecifikujte aplikovateľnosť paralelnej stratégie v rámci procesu porozumenia obrazu.
- 6.B.2** Špecifikujte aplikovateľnosť sekvenčnej stratégie v rámci procesu porozumenia obrazu.
- 6.B.3** Špecifikujte aplikovateľnosť hierarchického riadenia v rámci procesu porozumenia obrazu.
- 6.B.4** Podrobne porovnajte hierarchické riadenie riadené údajmi a riadené modelom.
- 6.B.5** Špecifikujte aplikovateľnosť riadenia zdola nahor v rámci procesu porozumenia obrazu.
- 6.B.6** Uveďte príklady aplikácie porozumenia obrazu z reálneho sveta, v ktorých je použitá riadiaca stratégia zdola nahor.
- 6.B.7** Špecifikujte aplikovateľnosť riadiacej stratégie riadenej modelom v rámci procesu porozumenia obrazu.
- 6.B.8** Uveďte príklady aplikácie porozumenia obrazu z reálneho sveta, v ktorých je použitá riadiaca stratégia zhora nadol.
- 6.B.9** Špecifikujte aplikovateľnosť kombinovaných riadiacich stratégií v rámci procesu porozumenia obrazu.
- 6.B.10** Uveďte príklady aplikácie porozumenia obrazu z reálneho sveta, v ktorých je použitá kombinovaná riadiaca stratégia.
- 6.B.11** Špecifikujte aplikovateľnosť nehierarchického riadenia v rámci procesu porozumenia obrazu.
- 6.B.12** Uveďte príklady aplikácie porozumenia obrazu z reálneho sveta, v ktorých je použité nehierarchické riadenie.
- 6.B.13** Popíšte formálne transformáciu medzi dvomi obrázkami trénovacej množiny.
- 6.B.14** Popíšte kontextovú klasifikáciu na základe Bayesovej minimálnej chyby.
- 6.B.15** Predpokladajme rekurzívny prístup kontextovej klasifikácie obrazu. Po koľkých rekurzívnych krokoch bude informácia obrazu umiestnená na pozícii (10, 62) ovplyvňovať značky na pozícii (9, 50)?
- 6.B.16** Implementujte metódu kontextovej klasifikácie obrazu. Otestujte ju na umelých i reálnych obrazoch.

6.B.17 Implementujte metódu rekurzívnej kontextovej klasifikácie obrazu. Otestujte ju na umelých i reálnych obrazoch.

6.B.18 Vytvorte si nejakú jednoduchú scénu. Označte si objekty, ich unárne a binárne relácie a vytvorte príklad nekonzistentného ohodnotenia scény.

6.B.19 Vytvorte program na interpretáciu obrazu použitím algoritmu na diskkrétne ohodnocovanie scén.

6.B.20 Uved'te niekoľko príkladov aplikácií skrytých Markových modelov.

7. Textúry

- 7.A.1** Čo je to textúra?
- 7.A.2** Čo je to texel?
- 7.A.3** Vysvetlite, ako súvisí popis textúry so škálovaním.
- 7.A.4** Z čoho pozostáva popis textúry?
- 7.A.5** Vysvetlite rozdiel medzi tónom a štruktúrou textúry.
- 7.A.6** Vysvetlite rozdiel medzi jemnými a hrubými textúrami.
- 7.A.7** Vysvetlite rozdiel medzi slabými a silnými textúrami.
- 7.A.8** Aké metódy popisu textúr poznáte?
- 7.A.9** Uvedte hlavné popisy textúr a stratégie rozpoznávania pri použití štatistického a syntaktického prístupu. Pre každý z týchto dvoch všeobecných prístupov uvedte textúrne typy, pre ktoré sa tento prístup hodí a pre ktoré je nevhodný.
- 7.A.10** Pre ktoré metódy rozpoznávania textúr je základom meranie priestorových frekvencií?
- 7.A.11** Na čom je založená metóda matíc opakovaného objavenia?
- 7.A.12** Ako možno formálne reprezentovať nenormalizované frekvencie opakujúceho sa objavenia ako funkcie uhla a vzdialenosti?
- 7.A.13** Napíšte, na kritériách odvodených z akých matíc opakovaného objavenia môže byť založená textúrna klasifikácia.
- 7.A.14** Popíšte popis textúr založený na hranovej frekvencii.
- 7.A.15** Na základe funkcie frekvencie objavenia sa hrán možno vytvoriť viacero charakteristík. Vymenujte a popíšte aspoň štyri z nich.
- 7.A.16** Ako sa počítajú textúrne charakteristiky v prístupe založenom na dĺžke primitíva?
- 7.A.17** Definujte primitívum.
- 7.A.18** Definujte:
 - a) Dôraz na krátke primitíva
 - b) Dôraz na dlhé primitíva
 - c) Šedoúrovňová uniformita
 - d) Uniformita dĺžky primitíva
- 7.A.19** Ako určujú vlastnosti textúry Lawove textúrne merania?

- 7.A.20** Definujte fraktálnu dimenziu a lacunarity a vysvetlite, ako môžu byť tieto miery použité na popis textúr.
- 7.A.21** Vymenujte ďalšie štatistické metódy na popis textúr.
- 7.A.22** Na čom sú založené syntaktické metódy popisu textúr?
- 7.A.23** Hybridný prístup kombinuje štatistické a syntaktické metódy. Čo spôsobuje, že technika je čiastočne syntaktická a čiastočne štatistická?
- 7.A.24** Aké metódy syntaktického popisu textúr poznáte?
- 7.A.25** Na čom je založená metóda grafových gramatík?
- 7.A.26** Popíšte úrovně primitív, ktoré môžeme určiť v hierarchických textúrach.
- 7.A.27** Definujte zoskupovanie primitív v hierarchických textúrach.
- 7.A.28** Vysvetlite, prečo sú deterministické gramatiky príliš obmedzujúce pri popise reálnych textúr.
- 7.A.29** Aké sú výhody zoskupovania primitív? Na aké textúry je tento prístup vhodný?
- 7.A.30** Vymenujte niekoľko oblastí použitia popisu a rozpoznania textúr.

7.B.1 Popíšte štatistické metódy popisu textúr.

7.B.2 Popíšte syntaktické a hybridné metódy popisu textúr.

7.B.3 Popíšte algoritmus na popis textúr pomocou autokorelačnej funkcie.

7.B.4 Popíšte, ako môže byť použitá diskrétna obrazová transformácia na popis textúr.

7.B.5 Vypočítajte matice opakovaného objavenia $P_{45,2}$, $P_{0,1}$, $P_{90,2}$ pre nasledujúci šedoúrovňový obraz.

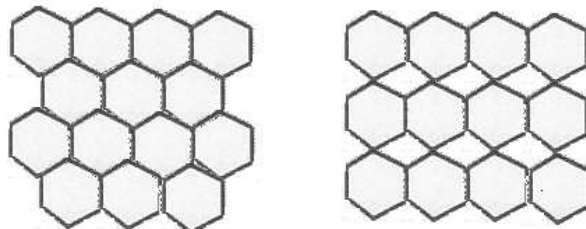
1	1	1	1
1	2	3	1
2	3	2	1
3	1	1	2

7.B.6 Popíšte, ako sa generujú gramatiky tvarových reťazcov.

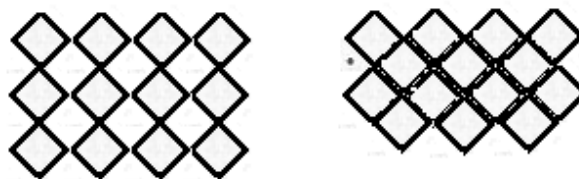
7.B.7 Popíšte konštrukciu rovinného grafu rozloženia primitív a vysvetlite problém klasifikácie textúry.

7.B.8 Popíšte algoritmus zoskupovania primitív textúr.

7.B.9 Vytvorte gramatiku, ktorá akceptuje textúru vľavo a neakceptuje textúru vpravo na nasledujúcom obrázku.



7.B.10 Vytvorte gramatiku, ktorá akceptuje textúru vpravo a neakceptuje textúru vľavo na nasledujúcom obrázku.



B Riešenia príkladov

1.A.1 Na koľko úrovni a aké vieme rozdeliť reprezentáciu obrazových dát?

Reprezentáciu obrazových dát vieme rozdeliť na 4 úrovne:

Ikonické – pozostávajú zo základných obrazových dát, matíc celočíselných hodnôt; sú výstupom predspracovania obrazu

Segmentové - obraz je rozdelený na časti, ktoré pravdepodobne patria tomu istému objektu

Geometrické - obsahuje informácie o 2D a 3D tvaroch

Relačné - vyššia úroveň abstrakcie; sémantické siete alebo rámce

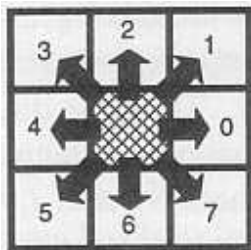
1.A.8 Popíšte, ako sa vytvára reťazcový kód.

Reťazcový kód sa používa na popis hranice objektu. Hranica je definovaná nasledovne:

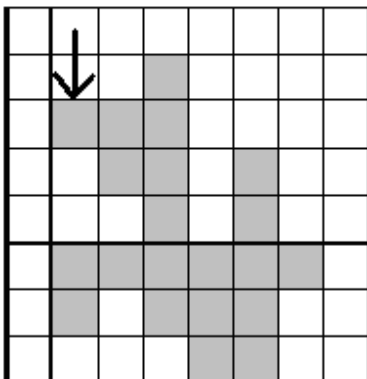
Zvolíme si nejaký referenčný bod a hranicu popíšeme ako postupnosť symbolov podľa preddefinovanej orientácie.

Pr:

Preddefinovaná orientácia

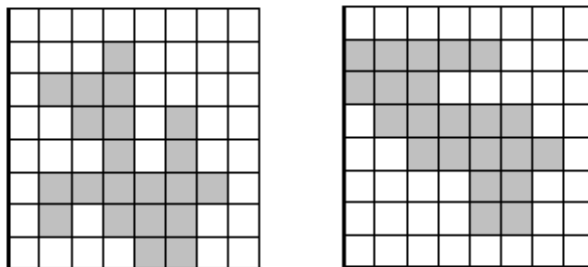


Potom reťazcový kód pre nasledujúci obrázok



bude nasledovný: 016667126756433520133

1.B.2 Určite reťazcový kód pre nasledujúce obrázky.



Reťazcový kód pre prvý obrázok sme skonštruovali v príklade 1.A.8.

Ak použijeme rovnakú preddefinovanú orientáciu, tak reťazcový kód druhého obrázku bude nasledovný (predpokladáme, že referenčný bod, je ľavý horný bod objektu):

0000457007564234332

1.B.7 Uved'te aspoň dva príklady, na ktorých ukážete výhodnosť RL-kódu oproti zápisu oblasti pomocou vymenovania jej prvkov. Pre každý príklad uved'te kompresný pomer.

	0	1	2	3	4	5	6
0							
1		■	■	■	■	■	
2		■	■	■	■	■	
3		■	■	■	■	■	
4		■	■	■	■	■	
5		■	■	■	■	■	
6							

RL – kód: (115)(215)(315)(415)(515) = 19

Vymenovanie: 11121314152122232425313233343541424344455152535455 = 50

Kompresný pomer: 19:50 = 0,38

	0	1	2	3	4	5	6
0							
1							
2							
3							
4							
5							
6							

RL – kód: $(106) = 3$

Vymenovanie: $10111213141516 = 14$

Kompresný pomer: $3:14 = 0,214$

2.A.2 Popíšte, ako funguje metóda Zero-crossing.

Zero-crossing (zistovanie prechodu nulou) druhej derivácie je robustnejšia metóda ako gradientné operátory malej veľkosti – počíta sa ako Laplacián z Gaussiánov (LoG) alebo ako diferenciacia Gaussiánov (DoG).

- prečo druhá derivácia – lebo prechod nulou sa zisťuje ľahšie ako extrém pri prvej derivácii
- postup: vyhladenie obrazu pomocou

$$G(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$$

a následná druhá derivácia

2.A.3 Akú úlohu hrá parameter σ pri použití tejto metódy?

- σ = parameter filtra, je úmerný veľkosti okolia, v ktorom filter pôsobí

2.A.14 Ako sa nazýva najjednoduchší parametrický hranový model a aké funkcie používa?

Parametrické hranové modely sú založené na myšlienke, že diskretnú funkciu intenzity obrazu možno považovať za vzorkovanú a zašumenú aproximáciu určitej spojitej alebo po častiach spojitej funkcie intenzity obrazu. Keďže jej priebeh nie je známy, odhaduje sa zo známej diskretné funkcie a jej vlastností. Používajú sa po častiach spojité funkcie, ktoré sa nazývajú fazety (alebo plôšky) a model sa nazýva fazetový (plôškový) model. Najjednoduchší je plochý fazetový model, ktorý používa konštantné funkcie a okolie pixla je reprezentované funkciami konštantnej intenzity.

2.B.5 Podrobne popíšte Cannyho hranovú detekciu.

- Cannyho hranová detekcia začína konvolúciou obrazu so symetrickým 2D Gaussiánom s parametrom σ (podobne ako pri zisťovaní prechodu nulou).
- Nasleduje odhad smeru normály hrany n pre každý obrazový bod podľa vzorca

$$\mathbf{n} = \frac{\nabla(G * f)}{|\nabla(G * f)|}$$

- Potom urobíme nemaximálne potlačenie hrán, ktoré nájde lokálne maximá kolmé na smer hrany, podľa vzorca

$$\frac{\partial^2}{\partial \mathbf{n}^2} G * f = 0$$

- Ďalej určíme veľkosť hrany ako veľkosť gradientu obrazovej funkcie

$$|G_n * f| = |\nabla(G * f)|$$

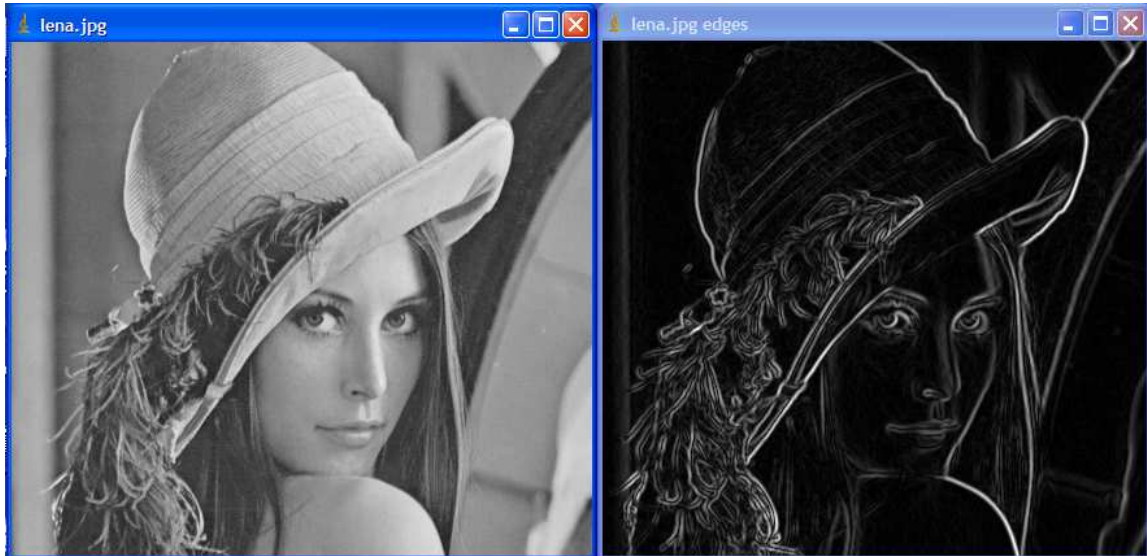
- Nasleduje hysterézne prahovanie (prahovanie odoziev hrany s dvoma prahmi, silná odozva – vracia hranu, nízka odozva spojená s blízkosťou vysokej odozvy môže vrátiť hranu) a syntéza príznakov (zistí sa také sigma Gaussiánu, ktoré dáva odozvu väčšiu ako prahovú, a potom sa zväčšuje sigma a agreguje sa získaná informácia).

- Na rozdiel od Marr-Hildrethovej operátora, ktorý je založený na Laplaciáne a teda nemá orientáciu, tu sa dá robiť aj orientácia.

Najlepším popisáním daného problému je názorná ukážka, ako sa zmení obraz po aplikovaní danej operácie. Keďže Cannyho hranová detekcia nie je implementovaná v programe, zisťovali sme, či už neexistuje nejaký plugin, ktorý by ju obsahoval. Na <http://www.imagescience.org/meijering/software/featurej/index.html> sme našli Cannyho hranovú detekciu implementovanú. Do priečinku ImageJ plugins sme si uložili súbory `FeatureJ.jar` a `imagescience.jar`. Po vypnutí a následnom zapnutí programu ImageJ sme už mali plugin nainštalovaný.

Otvorili sme si jeden zo vzorových obrázkov (*File -> Open Samples -> Lena*). Keďže ale vstupom pre konkrétnu operáciu je 8-bitový obraz, museli sme najskôr zmeniť typ (*Image -> Typ -> 8-bit*). Následne sme aplikovali operáciu na daný obraz (*Plugins -> FeatureJ ->*

FJEdges). Výsledok operácie je znázornený na nasledujúcom obr. (vľavo je pôvodný 8-bitový obraz)



2.B.6 Uved'te bikubický fazetový model a popíšte, ako sa odhadujú parametre c_i .

Bikubický fazetový model:

$$g(i, j) = c_1 + c_2x + c_3y + c_4x^2 + c_5xy + c_6y^2 + c_7x^3 + c_8x^2y + c_9xy^2 + c_{10}y^3,$$

ktorého parametre c_i sa odhadujú z okolia obrazového bodu, pomocou metódy najmenších štvorcov, alebo priamo z okolia rozmeru 5 x 5. Keď už máme fazetové parametre pre každý obrazový bod, hrany môžeme určiť pomocou extrémů prvej derivácie alebo ako prechod nulou druhej derivácie lokálne spojitej fazetovej funkcie. Určenie hrán je potom presnejšie ako pri diferenčných hranových operátoroch, ale platíme za to náročnosťou výpočtu.

3.A.6 Uvažujte heuristický algoritmus hľadania hrán. Vysvetlite, ako presnosť odhadu $h(x_i)$ ceny cesty z vrcholu x_i do koncového vrcholu ovplyvňuje priebeh hľadania. Uved'te, za akých podmienok je zaručená optimalita algoritmu. Popíšte, ako odhad zhora, odhad zdola a presný odhad ovplyvňuje rýchlosť nájdenia cesty.

Odhad ceny cesty z aktuálneho vrcholu n_i do koncového vrcholu n_b má veľký vplyv na priebeh hľadania. Keď sa tento odhad $h^*(n_i)$ presnej ceny $h(n_i)$ neberie do úvahy, teda $h^*(n_i) = 0$, algoritmus neobsahuje žiadnu heuristiku. Vďaka tomuto nájdená cesta bude optimálna podľa použitých kritérií, a teda cesta minimálnej ceny bude vždy nájdená. Použitím heuristiky nájdená cena cesty nebude vždy optimálna, ale prehľadávanie sa môže urýchliť.

Heuristické prehľadávanie môže výrazne urýchliť prehľadávanie grafu, ale heuristika musí spĺňať určité vlastnosti, ak má byť zaručená optimalita. Heuristika musí byť prípustná, to znamená, že $0 \leq h \leq h^*$, kde h^* reprezentuje optimálnu heuristiku.

Keď $h^*(n_i) = 0$, algoritmus nájde cestu s minimálnou cenou

Keď $h(n_i) > h^*(n_i)$, algoritmus môže byť rýchlejší, ale cesta minimálnej ceny nie je zaručená

Keď $h^*(n_i) \leq h(n_i)$, algoritmus nájde cestu s minimálnou cenou práve vtedy, keď

$c(n_p, n_q) \geq h^*(n_p) - h^*(n_q)$, pre nejaké p, q , keď $c(n_p, n_q)$ je presná minimálna cena z n_p do n_q , čo nie je ľahké splniť pre funkciu $f(x)$

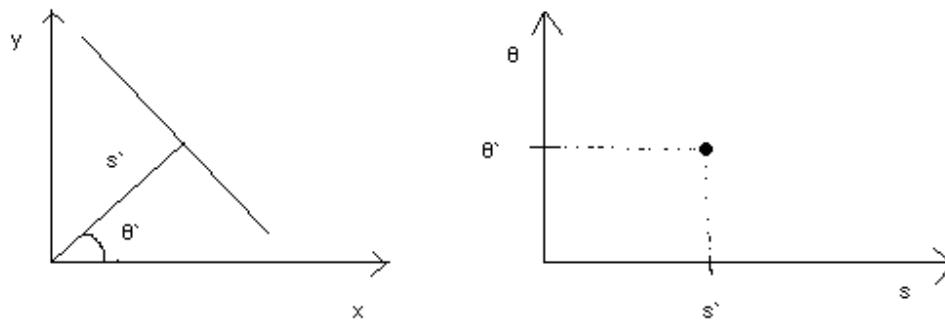
Keď $h^*(n_i) = h(n_i)$, prehľadávanie vždy nájde cestu minimálnej ceny s minimálnym počtom expandovaných vrcholov

Čím lepší je odhad funkcie $h(n)$, tým menší je počet vrcholov, ktoré musia byť expandované.

3.A.10 S použitím polárnej reprezentácie vysvetlite hlavný princíp Houghovej transformácie na detekciu priamok.

Polárna reprezentácia používa namiesto parametrického vyjadrenia priamky polárne súradnice, kde je priamka reprezentovaná ako $s = x \cos\theta + y \sin\theta$. Táto reprezentácia je vhodnejšia preto, lebo pri klasickej smernicovej reprezentácii ($y = kx + q$) vznikajú problémy pri zvislých líniách ($k \rightarrow \infty$).

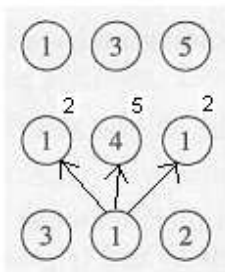
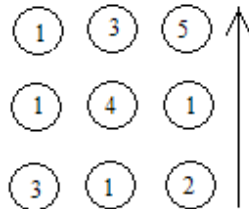
Hlavný princíp Houghovej transformácie s použitím polárnej reprezentácie sa, taktiež ako pri smernicovej reprezentácii, snaží priamku transformovať na jeden bod, ktorý určuje danú priamku. Tento má potom súradnice $[s, \theta]$ (znázornené na obrázku)



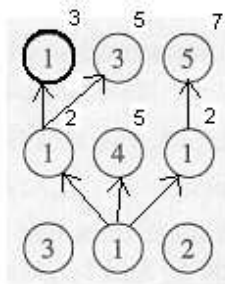
3.A.16 Vysvetlite hlavný princíp segmentácie na princípe povodí (watershed).

Princíp segmentácie metódou povodí je nájdenie lokálnych miním jasu obrazu a priradovanie pixlov týmto minimám na základe (najprudšieho) spádu okolia pixelu. Každé minimum má svoju oblasť (množinu pixelov, ktoré si doňho našli cestu). Pixel sa priradí oblasti ak existuje klesajúca cesta do jej minima (a je najkratšia pre daný pixel).

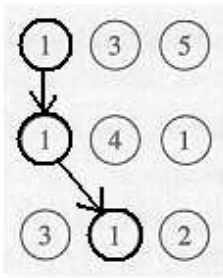
3.B.3 S použitím dynamického programovania nájdite optimálnu cestu cez graf na nasledujúcom obrázku, kde čísla vo vnútri kruhov určujú cenu vrcholu, pričom uvažujte troch možných nasledovníkov vrcholu. Ukážte všetky kroky hľadania v grafe, vrátane dátových štruktúr každom kroku hľadania.



- predstavuje graf po expanzii prvej vrstvy (do každého z vrcholov druhej vrstvy bola optimálna cesta z vrcholu s číslom 1)



- graf po expanzii druhej vrstvy (hrubšie je vyznačený vrchol s najmenšou cenou) – šípkami je vyznačené, z ktorého vrcholu z druhej vrstvy viedla optimálna cesta do každého vrcholu tretej vrstvy (vrcholy boli vždy expandované zľava doprava)



- je vyznačená optimálna cesta (pomocou back-trackingu) – jej cena je 3

3.B.6 Vysvetlite hlavné konceptuálne prístupy v segmentácii založenej na hranách a segmentácii založenej na oblastiach. Sú tieto dva prístupy duálne? Ak sa segmentácia založená na hranách a založená na oblastiach uplatnia na rovnaké obrazové dáta, budú výsledné segmentácie identické? Uvažujte ideálne bezšumové dáta a skutočné digitálne obrázky z reálneho sveta.

Segmentácia založená na hranách:

- je založená na hranách, ktoré boli detekované hranovými operátormi. Lokálne hrany zodpovedajú ostrému prechodu v úrovni šedej, vo farbe, textúre atď.
- najväčším problémom (sú spôsobené šumom alebo nevhodnou informáciou v obraze) je prítomnosť lokálnych hrán v miestach, kde neexistuje globálna hranica a naopak neprítomnosť lokálnych hrán tam, kde globálna hranica existuje.

Prahovanie obrazu hrán – je založené na vytvorení lokálnych hrán, ktoré sa prahujú vhodným prahom

Relaxácia (optimalizácia) hrán – vlastnosti hrany sa posudzujú v kontexte okolitých hrán. Ak existujú dostatočné príznaky existencie hranice, lokálna hrana sa posilňuje a naopak. Pracuje sa s pojmom dôvery hrany (iteratívne sa posudzuje dôvera každej hrany, až kým nekonverguje k 0 alebo k 1)

Ak je definované kritérium optimality, potom na určenie globálne optimálnych hraníc možno použiť heuristické grafové hľadanie alebo dynamické programovanie

Heuristické grafové hľadanie – proces určovania hranice sa transformuje na hľadanie optimálnej cesty v ohodnotenom grafe. Cieľom je nájsť optimálnu hranicu (vzhľadom na optimalizačné kritérium), ktorá spája dva špecifikované obrazové body alebo množiny obrazových bodov, ktoré reprezentujú začiatok a koniec hranice.

Dynamické programovanie – je založené na princípe optimality a reprezentuje efektívnu cestu simultánneho hľadania optimálnych ciest z viacerých začiatočných a koncových bodov. Hľadá optimálnu funkciu, v ktorých nie všetky premenné sú navzájom závislé.

Houghova transformácia – je technika použiteľná vtedy, keď treba detekovať objekty so známym tvarom hranice. Môže detekovať rovné čiary aj krivky, ak sú známe ich analytické vyjadrenia.

Segmentácia založená na oblastiach:

- musí spĺňať kritérium úplnej segmentácie (t.j. oblasť tvoria disjunktné podoblasti, ktorých zjednotením je práve celá oblasť) a podmienky maximálnej homogenity

$$H(R_i) = \text{TRUE, pre každú oblasť } i$$

$$H(R_i \cap R_j) = \text{FALSE pre } i \neq j \text{ a } R_i \text{ susedné s } R_j$$

Spájanie oblastí – začína pri presegmentovanom obraze, ktorý spĺňa podmienku, že každá oblasť je homogénna. Oblasti sa spájajú tak, aby bola splnená podmienka, že maximálnej homogenity. Metódy sa líšia v závislosti od zvolenej počiatočnej segmentácie a kritéria pre spájanie. Výsledná segmentácia závisí aj na poradí, v akom spájame jednotlivé oblasti.

Delenie oblastí – je opačné ku spájaniu oblastí. Začína sa pri podsegmentovanom obraze, ktorý nespĺňa podmienku homogenity oblasti. Potom sa existujúce oblasti postupne delia tak, aby spĺňali základné podmienky.

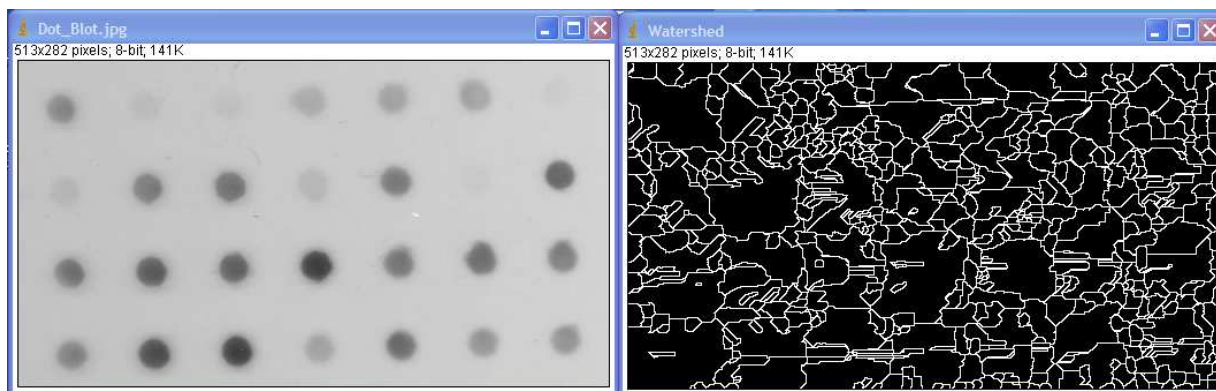
Delenia a spájanie oblastí – kombinácia predchádzajúcich metód. Má výhody oboch prístupov. Techniky tohoto prístupu obyčajne používajú pyramídové obrazové reprezentácie. Počiatočný obrázok nemusí spĺňať ani podmienku maximálnej homogenity, ani to, že každá oblasť musí byť homogénna.

Tieto dva prístupy sú duálne. Každá oblasť môže byť reprezentovaná jej uzavretou hranicou a každá uzavretá hranica popisuje hranicu. Vďaka rozličným vlastnostiam segmentácie založenej na hranách a založenej na oblastiach môžeme očakávať rozličné výsledky pri uplatnení na rovnaké obrazové dáta.

3.B.11 Implementujte segmentáciu na princípe povodí.

Zistili sme, že na <http://rsb.info.nih.gov/ij/plugins/watershed.html> je segmentácia na princípe povodí implementovaná a preto sme ju využili. Do priečinku plugins sme uložili súbor Watershed_Algorithm.jar. Po vypnutí a následnom zapnutí programu ImageJ sme otvorili

obrázok Dot_Blot.jpg a aplikovali sme naň túto operáciu (*Plugins -> Filters -> Watershed Algorithm*). Výsledok je znázornený na nasledujúcom obrázku.



4.A.3 Metódy popisu tvaru sa delia podľa viacerých kritérií (napr. podľa vstupnej reprezentácie - hranica vs. oblasť). Vymenujte ďalšie možnosti delenia a zarad'te Fourierove popisy tvaru.

Ďalšie možnosti delenia sú:

- schopnosť rekonštrukcie objektu - či z popisu tvaru možno zrekonštruovať objekt a do akej miery
- schopnosť určiť neúplné tvary – pri zakrytých objektoch
- lokálny/globálny charakter popisu – globálne vyžadujú úplnú informáciu, lokálne pracujú aj s čiastočnou informáciou
- štatistické alebo syntaktické popisy – súvisí s voľbou rozpoznávacieho algoritmu
- robustnosť popisu - vzhľadom na posunutie, otočenie, škálové transformácie a vzhľadom na rozlíšenie obrazu
- matematické alebo heuristické techniky – matematické sú napr. Fourierove popisy a príkladom heuristickej techniky je pozdĺžnosť

4.A.8 Definujte pojmy:

- a) **B-splajny**
- b) **riadiaci polygón**
- c) **bázická funkcia.**

a) B-splajn – po častiach polynomická krivka, ktorej tvar je úzko spojený s jej riadiacim polygónom

$$S^d(t) = \sum_{i=0}^n V_i N_i^d(t)$$

pričom d = stupeň
 n = m-d-1 (počet b-splajnových bázických funkcií)
 m = počet uzlov

b) Riadiaci polygón je reťazec vrcholov, ktoré určujú polygonálnu reprezentáciu krivku. Keď niektorý z bodov riadiaceho polygónu zmení svoju polohu, zmení sa aj tvar krivky v malom okolí tohto vrcholu.

c) $N_i^d(t)$ sú bázické funkcie, ktorých tvar je daný rádom splajnu.

4.A.15 Napíšte dve základné metódy vytvorenia grafu podoblastí.

Dve základné metódy na vytvorenie grafu podoblastí sú:

Stenčovanie oblastí - vedie ku kostre oblasti, ktorú možno popísať grafom

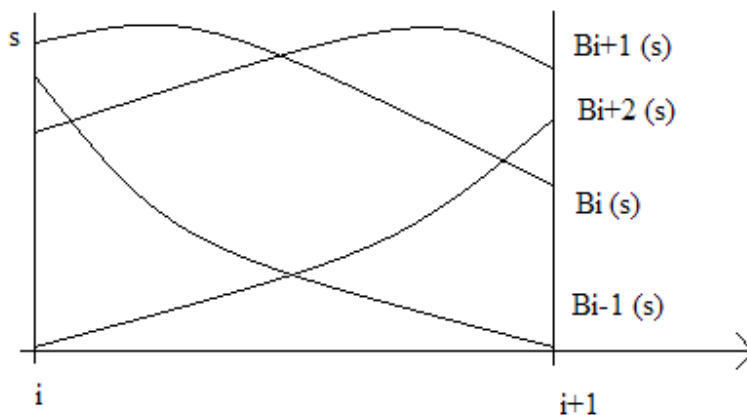
Dekompozícia oblastí – graf sa vytvára na vyššej úrovni

4.A.25 Čo a ako reprezentuje graf susedností oblastí?

Graf susedností oblastí reprezentuje každú oblasť ako vrchol grafu a vrcholy susedných oblastí sú spojené hranou.

4.B.5 Koľko je nenulových bázičkových funkcií pre $s \in (i, i+1)$? Uved'te ich.

Pre daný interval sú iba štyri nenulové bázičkové funkcie:

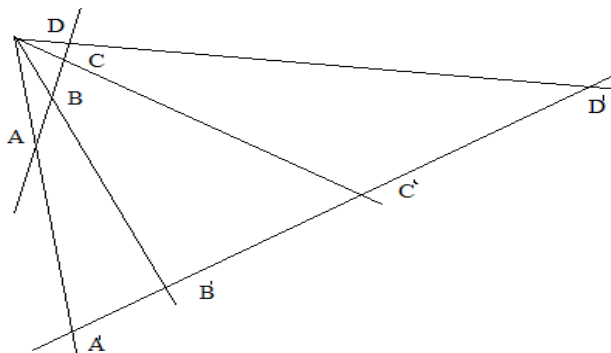


4.B.7 Popíšte invariant Podiel priesečníku.

Tvarové invarianty predstavujú vlastnosti geometrických konfigurácií, ktoré sa nezmenia pri vhodnej skupine transformácií; počítačové videnie sa sústreďuje osobitne na projektívne transformácie.

Podiel priesečníku je klasický invariant projektívnej priamky, pretože priamka sa vždy zobrazí na priamku. Jej ľubovoľné štyri body možno opísať ako invariant podielu priesečníku.

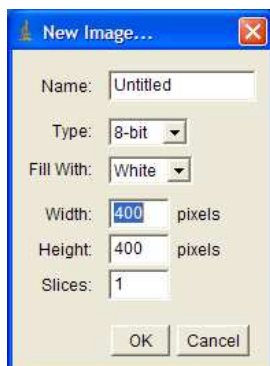
$$I = \frac{(A - C)(B - D)}{(A - D)(B - C)}, \text{ kde } (A - C) \text{ reprezentuje vzdialenosť medzi } A \text{ a } C$$



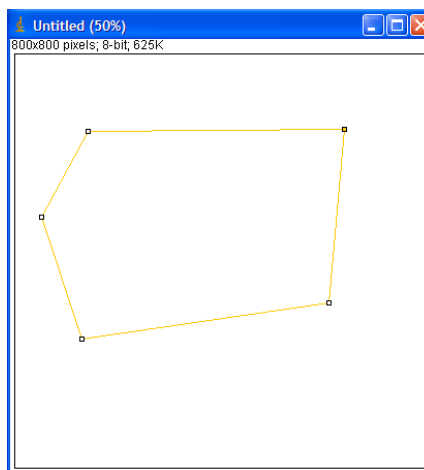
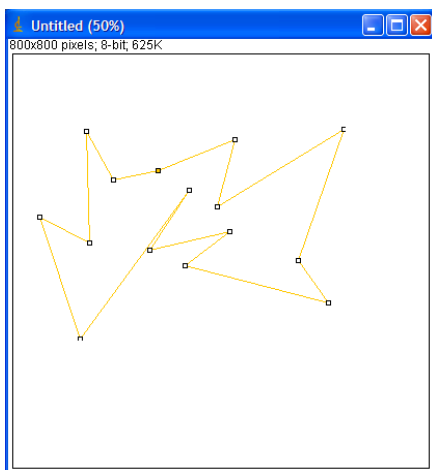
4.B.13 Vytvorte program, ktorý nájde konvexný obal jednoduchého mnohoúhelníka.

V prvom kroku sme zistili, že program má už v sebe implementovanú funkciu na hľadanie konvexného obalu mnohoúhelníka. Vytvorili sme si teda nový obrázok (*File -> New ->*

Image). Pred vytvorením sa nám otvorilo okno (obr.1), kde sme si zvolili názov obrázku, veľkosť, typ...



Ďalej sme využili jeden z nástrojov plošnej sekcie panelu nástrojov, konkrétne *Polygon*. Tento vytvorí nepravidelný tvar výberu, vytvorený zo série čiarových segmentov. Vytvorili sme teda nekonvexný mnohouholník (obr.2), na ktorý sme následne aplikovali funkciu na hľadanie konvexného obalu mnohouholníka (*Edit -> Selection -> Convex Hull*). Výsledný konvexný obal je na obr.3 a ako môžeme vidieť, získali sme správny výsledok.



5.A.9 Definujte nasledujúce pojmy:

- a) obrazec
- b) trieda
- c) klasifikátor
- d) priestor príznakov.

a) obrazec je príznakový vektor $x = (x_1, x_2, \dots, x_n)$, prvok príznakového priestoru

b) trieda je prvok z množiny tried T . Triedy sú podmnožiny objektov so spoločnými príznakmi. Objekt je fyzická jednotka, obvykle reprezentovaná oblasťou v segmentovanom obraze.

c) Klasifikátor je funkcia (zobrazenie) $K: P \rightarrow T$, ktoré priradí obrazcu triedu. Klasifikátor sa stará o zaradenie objektov do tried, nerozhoduje o zaradení do triedy na základe objektu priamo, ale na základe vlastností objektu.

d) Množina všetkých možných obrazcov sa nazýva priestor obrazcov alebo príznakový priestor. Je to kartézsky súčin domén, z ktorých sa vyberajú príznaky. ($P = D_1 \times D_2 \times D_3 \times D_4 \times \dots$).

5.A.24 Popíšte hlavné kroky syntaktického rozpoznávania.

- charakteristický je kvalitatívny popis objektov
- primitíva = elementárne vlastnosti syntakticky popísaných objektov
- abeceda = množina všetkých primitív
- popisný jazyk = množina všetkých slov vytvorených z abecedy, ktoré opisujú objekt z jednej triedy
- gramatika = množina pravidiel, podľa ktorých sa vytvárajú slová nejakého jazyka z prvkov abecedy

Syntaktické rozpoznávanie obrazcov pozostáva z nasledovných krokov:

- definuj primitíva a vzťahy medzi nimi
- zostroj gramatiku pre každú triedu objektov

pre každý objekt vyťahni primitíva a rozpoznaj ich a vzťahy medzi nimi a zostroj slovo reprezentujúce objekt

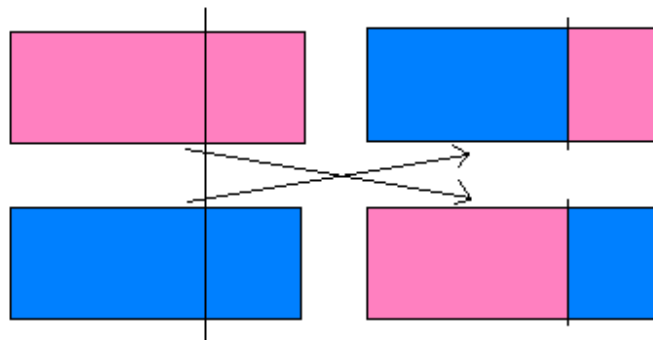
na základe syntaktickej analýzy zaraď predmet do tej triedy, ktorej gramatika ho generuje

- vytvorenie gramatiky si obyčajne vyžaduje výraznú ľudskú interakciu
- automatický proces vytvárania gramatiky na základe príkladov sa nazýva inferencia gramatiky (treba predpokladať pozitívne aj negatívne príklady)
- rozhodnutie, či neznáme slovo môže alebo nemôže byť generované danou gramatikou sa robí syntaktickou analýzou (buď zhora nadol alebo opačne)

5.A.33 Aké sú úlohy reprodukcie, kríženia a mutácie v genetických algoritmoch?

Operátor reprodukcie je zodpovedný za prežitie najschopnejších a smrť ostatných na pravdepodobnostnom princípe. Kopíruje reťazce s najvyšším ohodnotením do ďalšej populácie. Môžu sa kopírovať aj viacnásobne. Celkový počet reťazcov sa nemení, preto každá ďalšia generácia má vyššie priemerné ohodnotenie.

Operátora kríženia je náhodné spojenie častí dvoch reťazcov na vytvorenie dvoch nových reťazcov. Nie všetky reprodukované reťazce podliehajú kríženiu, opäť sa to deje na pravdepodobnostnom princípe, ktorý odhaduje, koľko reťazcov prejde krížením.



Operátor mutácie hrá sekundárnu úlohu. Jeho princípom je z času na čas náhodne zmeniť jeden znak v nejakom reťazci. Môže to byť napr. jeden z tisíca spracovaných bitov.

5.A.37 Uved'te principiálne, ako by ste mohli využiť optimalizačné algoritmy pre riešenie úlohy rozpoznávania obrazcov.

Optimalizačné problémy hľadajú minimalizáciu alebo maximalizáciu nejakej objektívnej funkcie. Návrh tejto funkcie je kľúčovým faktorom fungovania optimalizačných algoritmov. Chceme, aby interpretácia obrazu bola najlepšia.

Pre riešenie úloh rozpoznávania obrazcov by sa tieto algoritmy mohli použiť na nájdenie minima pri pravidle najbližšieho suseda (kritérium minimálnej vzdialenosti), hľadanie parametrov klasifikátora tak, aby sa globálne minimalizovala chyba klasifikácie...

5.B.2 Porovnajete klasifikačné prístupy použité v štatistickom rozpoznávaní obrazcov a v neurónových sieťach.

Štatistické rozpoznávanie : z trénovacej množiny (podmnožina vstupov) sa určia kritéria rozhodovania (pravidlá), tak aby bola minimalizovaná chybová funkcia.

Neurónové siete : neurónová sieť (napr. dopredná) sa natrénuje na trénovacej množine, tak aby správne asociovala vstup so žiadaným výstupom. Taktiež sa snažíme o minimalizáciu chyby klasifikácie.

5.B.9 Vytvorte gramatiku G, ktorá generuje jazyk L(G) rovnostranných trojuholníkov. Primitíva s orientáciou 0°, 60° a 120° tvoria množinu terminálnych symbolov V = {a, b, c}.

Ak primitíva s orientáciou 0°, 60° a 120° tvoria množinu terminálnych symbolov T = {a, b, c}, potom jazyk rovnostranných trojuholníkov je $L(G) = \{a^n b^n c^n / n \geq 1\}$.

Gramatika, ktorá generuje tento jazyk je nasledovná:

$G(N, T, P, S)$, pričom

$N = \{S, A, B\}$ - neterminály

$T = \{a, b, c\}$ - terminály

S – počiatočný neterminál

$P = \{S \rightarrow abc / aAbc,$

$Ab \rightarrow bA,$

$Ac \rightarrow Bbcc,$

$bB \rightarrow Bb,$

$aB \rightarrow aaA / aa\}$ – pravidlá

Ak chcem vytvoriť trojuholník abc, použijem jediné pravidlo $S \rightarrow abc$

Ak chcem vytvoriť trojuholník $a^2b^2c^2$ použijem nasledovné pravidlá:

$S \rightarrow aAbc \rightarrow abAc \rightarrow abBbcc \rightarrow aBbbcc \rightarrow aabbcc$

Takýmto spôsobom viem dostať ľubovoľný trojuholník tvaru $a^n b^n c^n$.

5.B.13 Vymenujte výhody a nevýhody genetických algoritmov a simulovaného žihania v porovnaní optimalizačného prístupu založenom na deriváciách.

výhody

- generické algoritmy – majú väčšiu šancu na nájdenie globálneho optima, nakoľko hľadajú v populácii riešení
- simulované žihanie – kombináciou dvoch optimalizačných princípov (rozdeľuj a panuj a horolezecký algoritmus) sa vyhýba uviaznutiu v lokálnom optime

nevýhody

- problém s tým, kedy zastaviť algoritmus
- nezaručujú nájdenie minima

5.B.15 Naznačte spôsob riešenia pre rozpoznávanie izolovaných slov.

Existuje viacero možností. Napr.:

- rozpoznávanie pomocou klasifikátora využívajúceho metódy dynamického programovania (dáme do súvisu obrazy dvoch slov, ako postupnosti príznakov a vymedzujeme priestor, kadiaľ môže ísť analýza pri hľadaní riešenia)
- štrukturálne metódy rozpoznávania - po akustickej analýze sa robí fonetická analýza a potom sa signál delí na segmenty (najkratšie významné úseky). Hranice segmentov možno charakterizovať zmenou spôsobu artikulácie alebo ostrými zmenami konfigurácie rečového ústrojenstva. Jednotlivým segmentom sa potom priradujú fonetické prvky (fonémy, slabiky) alebo ich triedy (explozívny, sykavky a pod.).

6.A.1 Do ktorej úrovne počítačového videnia by ste zaradili porozumenie obrazu? Aká je hlavná úloha tejto úrovne?

Počítačové videnie môžeme rozdeliť na vyššiu a nižšiu úroveň, pričom porozumenie je v rámci tejto hierarchie tou najvyššou úrovňou.

Hlavným cieľom je dosiahnuť správanie počítača podobné biologickým systémom. Vytvára sa pri ňom v počítači vnútorný model sveta, na ktorý sa pozerá.

Počítaču obvykle chýbajú širšie použiteľné, všeobecné a modifikovateľné znalosti o reálnom svete. Vytvára si vnútorný model, overuje ho a pokiaľ sa zhoduje s realitou, úloha je vyriešená.

6.A.5 Jedno z možných rozdelení stratégií môže byť na paralelné a sekvenčné. Uved'te, pre ktoré operácie je ktorá stratégia vhodná.

Sekvenčné stratégie sú vhodné pre vyššie úrovne spracovania, ktoré používajú vyššiu úroveň abstrakcie a naopak paralelné stratégie je možné použiť takmer pre všetky operácie nižšej úrovne spracovania obrazu.

6.A.24 Vysvetlite, ako možno použiť apriórnu informáciu pri modeloch aktívnej hranice reprezentovaných hadmi.

- apriórna informácia (odhad užívateľa), je špecifikácia počiatočného bodu a približný tvar hranice posúva hada k vhodnému riešeniu. Had sa snaží minimalizovať svoju energiu v obraze, pričom vychádza z daného odhadu užívateľa.

6.A.25 Aký druh informácie môže byť reprezentovaný modelmi rozmiestnení bodov (PDM)?

- informácie o tvare nejakého objektu, alebo triede podobných objektov, ktoré sa nedajú presne popísať kvôli početnosti rôznych variácií tvaru a deformácií (napr. kosti).

- je to technika na popis tvarov objektov, ale môže slúžiť aj na rozpoznanie týchto tvarov v iných obrázkoch. Popisujeme ňou objekty dobre rozpoznateľné ľudským okom, no majúce premenlivý tvar.

6.A.32 V čom spočíva hlavná myšlienka ohodnocovania scény?

Hlavnou myšlienkou ohodnocovania scény je priradiť značku každému objektu obrazu, aby sme mohli celý obrázok dobre popísať. Ide vlastne o zachytenie sémantiky objektov, pričom ohodnotenie scény musí spĺňať určité vlastnosti. Jednou z vlastností je konzistentnosť. Ak má byť ohodnotenie konzistentné, nesmie nastať nejaká zakázaná situácia. Napr. objekt dom je uprostred objektu jazero. Takéto ohodnotenie by bolo nekonzistentné.

6.B.4 Podrobne porovnajte hierarchické riadenie riadené údajmi a riadené modelom.

Riadiaca stratégia riadená údajmi (zdola nahor) je postup od ikonických obrazových dát, cez predspracovanie, ďalej segmentáciu až po porozumenie, ktoré pozostáva z porovnania rozpoznaných objektov (cez metódy rozpoznávania obrazcov) s reálnymi objektami. Táto stratégia je úspešná vtedy, keď sú k dispozícii kvalitné údaje, ktoré poskytujú dobrý vstup do vyššej fázy – napr. dobre osvetlené objekty na priemyselnom páse.

Riadiaca stratégia riadená modelom (zhora nadol) nemá takú ustálenú formu, akú je možné popísať pre predchádzajúcu stratégiu. Hlavným princípom tejto stratégie je vytvoriť vnútorný model a jeho verifikáciu, čiže ide o cieľovo-orientovaný prístup. Ciele na vyššej úrovni sa rozložia na podciele v nižšej úrovni spracovania atď.

6.B.6 Uved'te príklady aplikácie porozumenia obrazu z reálneho sveta, v ktorých je použitá riadiaca stratégia zdola nahor.

Príkladom použitia riadiacej stratégie zdola nahor je napr. Marrova teória porozumenia obrazu, formulovaná koncom 70. rokov.

Cieľom je kvalitatívne a kvantitatívne rekonštruovať 3D geometrické popisy z jedného alebo viac obrazov intenzity jasu, za veľmi slabých predpokladov o objektoch v scéne.

Štyri reprezentácie sú usporiadané spôsobom zdola nahor:

vstupný obraz intenzity jasu

primal sketch (primárny náčrt), reprezentujúci výrazné zmeny (hrany) v obraze v súradniciach pozorovateľa

2,5 D sketch (náčrt), reprezentujúci hĺbku od pozorovateľa a orientáciu plochy

3D reprezentácia, reprezentujúca geometriu objektu v súradniciach objektu

2,5 D náčrt sa získava z primárneho náčrtu technikami, ktoré sa súhrnne nazývajú tvar z X.

3D reprezentácie je veľmi ťažko získať, tento krok nebol zatiaľ vo všeobecnom prípade vyriešený.

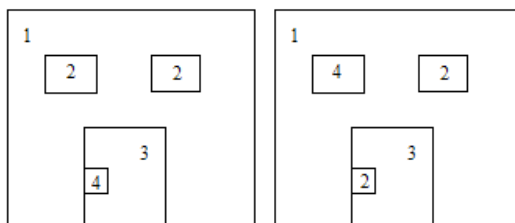
6.B.18 Vytvorte si nejakú jednoduchú scénu. Označte si objekty, ich unárne a binárne relácie a vytvorte príklad nekonzistentného ohodnotenia scény.

Objekty: pozadie, okno, dvere, kľučka

Unárne relácie: Okno je obdĺžnikové. Dvere sú obdĺžnikové. Kľučka je štvorcová.

Binárne relácie: Okno je nad dverami. Kľučka je na dverách. Pozadie susedí s hranicami objektov.

Prvý obrázok predstavuje takúto scénu, a druhý je príkladom nekonzistentného ohodnotenia scény.



7.A.6 Vysvetlite rozdiel medzi jemnými a hrubými textúrami.

Jemné textúry sú charakteristické väčšími priestorovými frekvenciami a skladajú sa z menších primitív, hrubé textúry naopak, sú charakteristické nižšími priestorovými frekvenciami a skladajú sa z väčších primitív.

7.A.9 Uved'te hlavné popisy textúr a stratégie rozpoznávania pri použití štatistického a syntaktického prístupu. Pre každý z týchto dvoch všeobecných prístupov uved'te textúrne typy, pre ktoré sa tento prístup hodí a pre ktoré je nevhodný.

Popis textúry (závisí od škály) pozostáva z:

Tón – popisuje vlastnosti intenzity pixlov primitíva

Štruktúra – odráža priestorové vzťahy medzi primitívami

Metódy popisu textúry:

Štatistické – počítajú rôzne vlastnosti textúry a sú vhodné vtedy, keď veľkosť textúrnych primitív je porovnateľná s veľkosťou pixla

Syntaktické a hybridné (kombinácia štatistických a syntaktických) – sú vhodnejšie pre textúry, ktorých primitíva sa dajú jednoducho určiť a dajú sa popísať ich vlastnosti

Stratégie rozpoznávania pri použití štatistického prístupu:

Štatistické metódy popisu textúry popisujú textúry vo forme vhodnej pre štatistické rozpoznávanie obrazcov. Ako výsledok popisu, každá textúra je popísaná príznakovým vektorom, ktorý reprezentuje bod vo viacrozmernom príznakovom priestore. Cieľom je nájsť deterministické alebo pravdepodobnostné pravidlá rozhodovania o pridelení textúry špecifickej triede.

Stratégie rozpoznávania pri použití syntaktického prístupu:

Syntaktický popis textúr je založený na analógii medzi priestorovým vzťahom primitív a štruktúrou formálnych jazykov.

7.A.22 Na čom sú založené syntaktické metódy popisu textúr?

Syntaktické metódy popisu textúr sú založené na analógii medzi priestorovými vzťahmi textúrnych primitív a štruktúrou formálnych jazykov.

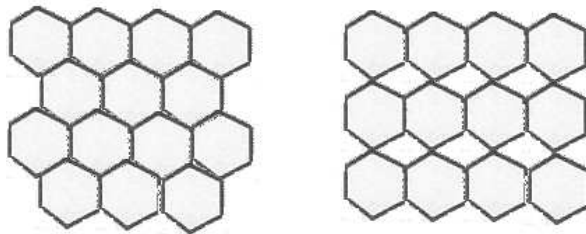
Využívajú myšlienku, že textúra pozostáva z primitív umiestnených v takmer pravidelných vzťahoch. Na popis textúry musia byť určené pravidlá pre popis primitív a ich umiestnenie.

7.B.1 Popíšte štatistické metódy popisu textúr.

Štatistické metódy popisu textúry počítajú rôzne vlastnosti textúry a sú vhodné vtedy, keď veľkosť textúrnych primitív je porovnateľná s veľkosťou pixla.

Metódy štatistického popisu textúr popisujú textúry vo forme vhodnej pre štatistické rozpoznávanie obrazcov. Výsledkom takéhoto popisu je, že každá textúra je popísaná príznakovým vektorom, ktorý reprezentuje bod vo viacrozmernom príznakovom priestore.

7.B.9 Vytvorte gramatiku, ktorá akceptuje textúru vľavo a neakceptuje textúru vpravo na nasledujúcom obrázku.



Gramatika je nasledovná:

$$T = \{ \text{hexagon} \}$$

$$N = \{ \text{hexagon with dot} \}$$

$$S = \text{hexagon with dot}$$

$$R = \begin{aligned} & \text{hexagon with dot} \longrightarrow \text{hexagon} \\ & \text{hexagon with dot} \longrightarrow \text{hexagon} \text{ hexagon} \\ & \text{hexagon with dot} \longrightarrow \text{hexagon} \text{ hexagon} \\ & \text{hexagon with dot} \longrightarrow \text{hexagon} \text{ hexagon} \\ & \vdots \\ & \vdots \\ & \text{hexagon with dot} \longrightarrow \text{hexagon} \text{ hexagon} \end{aligned}$$

pričom rotácia primitíva je reprezentovaná čiernou bodkou.