

Zadanie č.2 (Dúhovky)

Predspracovanie vzoriek a generovanie príznakov

Autor: Bc. Viktor Chovanec

80331

Predmet: Biometria

BIOM



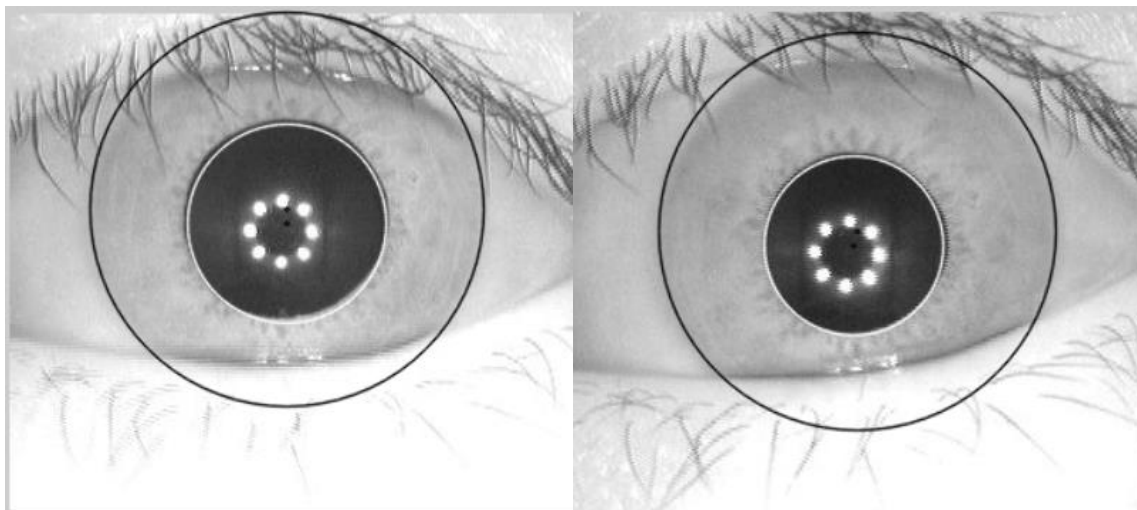
Obsah

Obsah.....	2
1 Úloha č. 1	3

1 Úloha č. 1

Pre vypracovanie zadania sme využili programovací jazyk C++ realizovaný vo vývojovom prostredí CLion. Implementačná časť využívala API rozhranie knižnice OpenCv.

Prvým krokom bolo načítanie dát ohraničujúcich význačné body, ktoré chceme spracovávať. Pre túto úlohu sme využili existujúce riešenie z prvého zadania. Vykonalí sme overenie správnosti načítaných údajov, ktoré sme sa vykreslili. (Obr. 1)

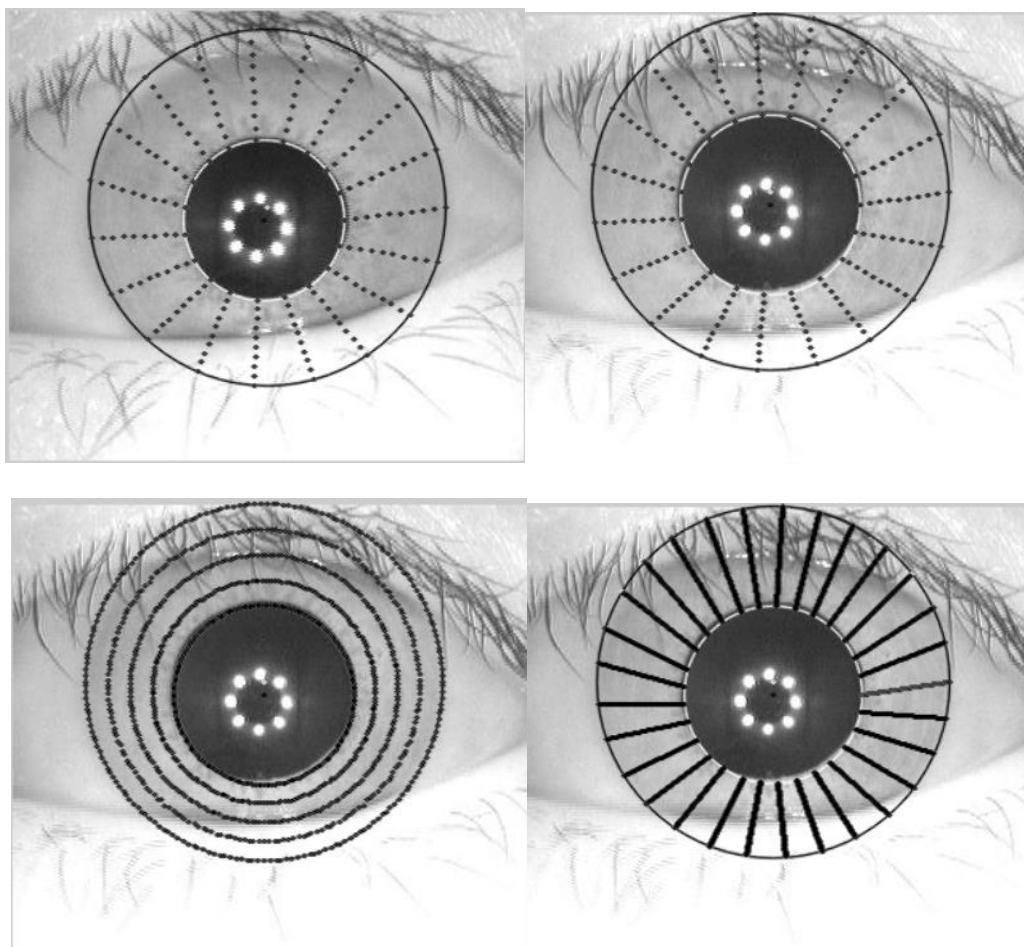


Obr. 1 Overenie správnosti načítaných údajov

Pre konverziu kartézskych súradníc na polárne využívame vlastný algoritmus, ktorý iteratívne prechádza všetky body, ktoré sa nachádzajú na oboch kružniciach, a získava pixely na úseku medzi oboma bodmi. Pre získanie aproximačných bodov oboch kružníc využívame funkciu **ellipse2Poly**. Pracujeme s faktom, že kruh je elipsa, ktorá má oba polomery rovnaké. Takýmto prístupom zabezpečíme dokonalejšie spracovanie v prípade dvoch nesústreďných kružníc.

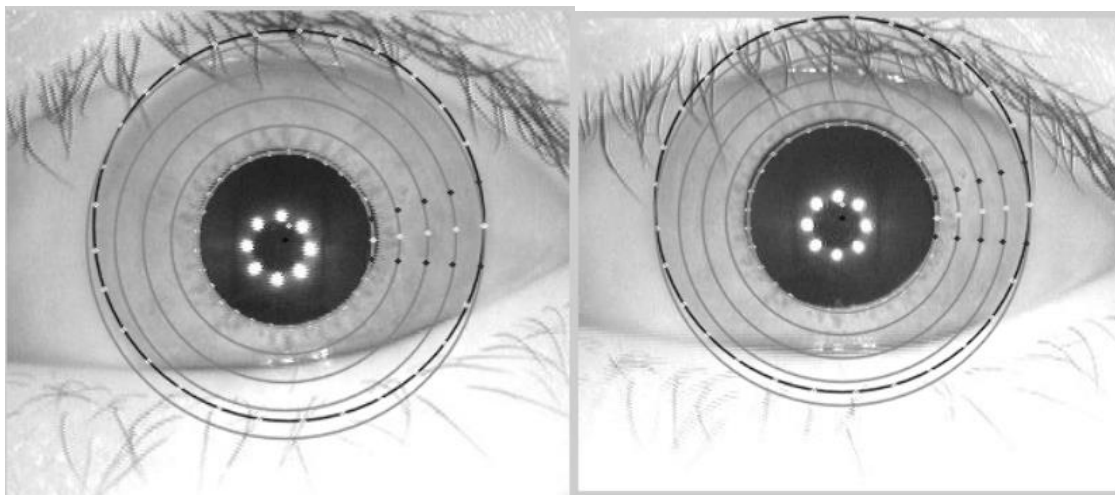
Funkcia slúžiaca na spracovanie a transformáciu dúhoviek umožňuje používateľovi zadať veľkosť požadovaného výstupu z funkcie. Tohto efektu dosahujeme matematickým mapovaním veľkosti požadovaného výstupu na skutočnú veľkosť obrázka.

V praxi toto mapovanie znamená, že pre prípad menšieho požadovaného rozsahu roz distribuuje funkcia požadovaný počet bodov po obraze. Ak zadáme väčšiu výstupnú veľkosť ako je tá skutočná, dochádza k opakovaniu niektorých bodov na obraze. (Obr. 2)



Obr. 2 Ukážky rôznych mapovaní spracovaných veľkostí

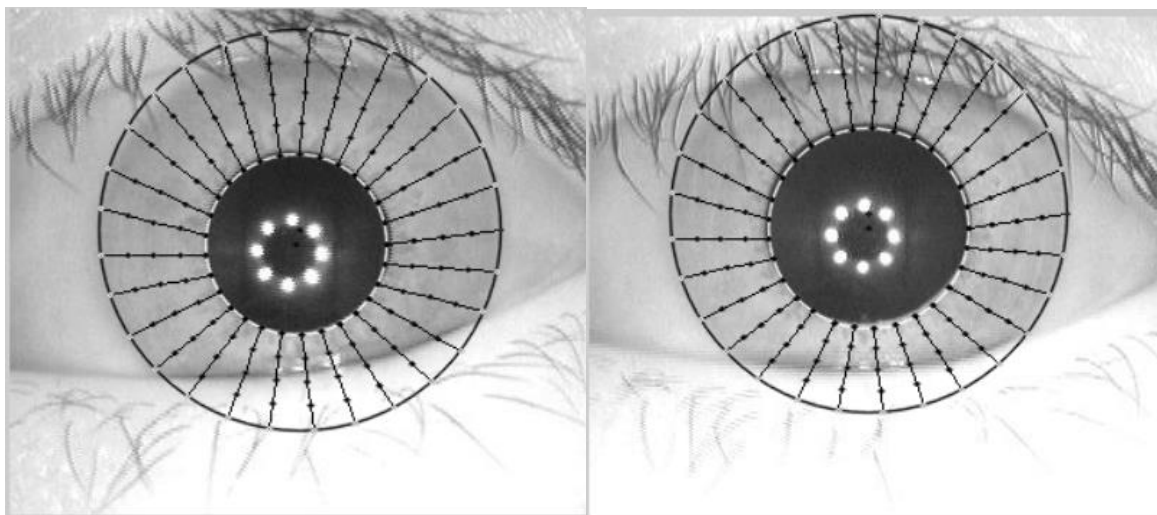
Pokiaľ používateľ zadá menšie výstupné rozlíšenie ako skutočné, dochádza k strate údajov. Náš algoritmus počíta aj s takýmito prípadmi. V prvom kroku dochádza k predspracovaniu všetkých význačných bodov a v druhom iteračnom kroku využívame existencie susedných oblastí. Cieľom tohto spracovania je vytvorenie priemeru pixelov, ktoré sa nachádzajú medzi dvoma susednými priamkami. Je potrebné počítať aj s tým, že tieto pixely sa nachádzajú na kružnici opisujúcej tieto body od stredu. Priemer tejto kružnice popisujúci body si vypočítame ako euklidovú vzdialenosť tohto bodu od stredu. (Obr. 3)



Obr. 3 Kružnice popisujúce trajektóriu požadovaných bodov

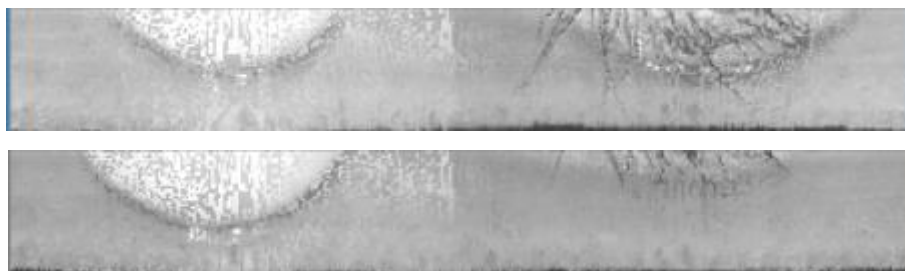
Pre iteráciu týchto bodov využívame rovnaký princíp aproximácie pomocou elipsy. Tieto body však rozdeľujeme na dva intervaly, ktoré predstavujú úsek medzi požadovanými bodmi a mimo nich. Táto iterácia je vyžadovaná z dôvodu, že nevieme zabezpečiť štandardný začiatkový bod iterovania. Výsledným úsekom sa stáva ten s menším počtom bodov. Výsledná hodnota pixelu je následne počítaná ako priemer všetkých pixelov, ktoré sa nachádzajú na tejto trajektórii.

Podobný postup je aplikovaný aj v prípade, kedy máme menší počet vertikálnych bodov. (Obr. 4)



Obr. 4 Aproximácia vertikálnych bodov

Výsledný výsek je následne reprezentovaný ako zhuk týchto významných bodov. (Obr. 5)



Obr. 5 Výsledná aproximácia bodov

Pre vytvorenie masky tohto obrazu využívame trik s použitím dvoch masiek pre horné a spodné viečko. (Obr. 6)



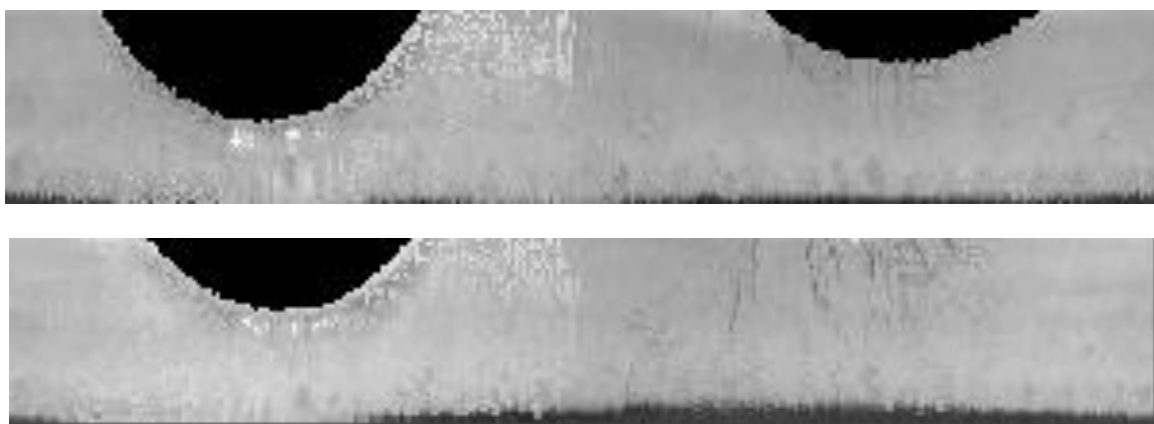
Obr. 6 Masky prikrývania dúhovky s viečkami

Algoritmus sa pre každú prístupovanú pozíciu na pôvodnom obraze pozrie aj na farbu v týchto dvoch maskách. V prípade zistenia šedej farby na akejkoľvek pozícii je tento bod vyznačený ako biely bod na maske. (Obr. 7)

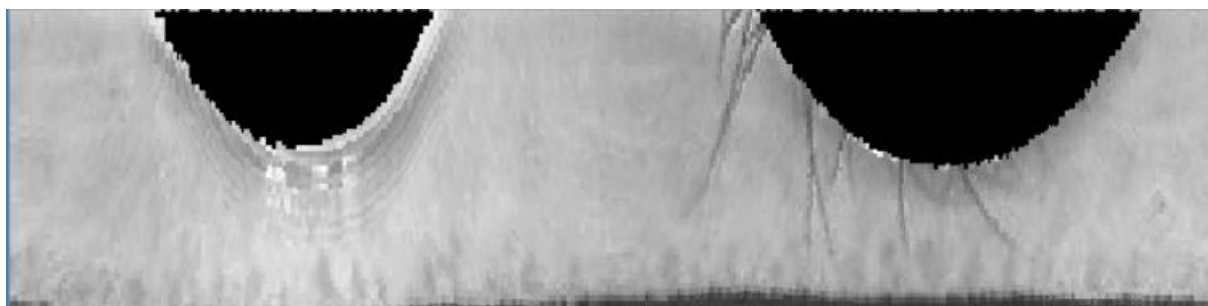


Obr. 7 Masky pre výsledný výrez

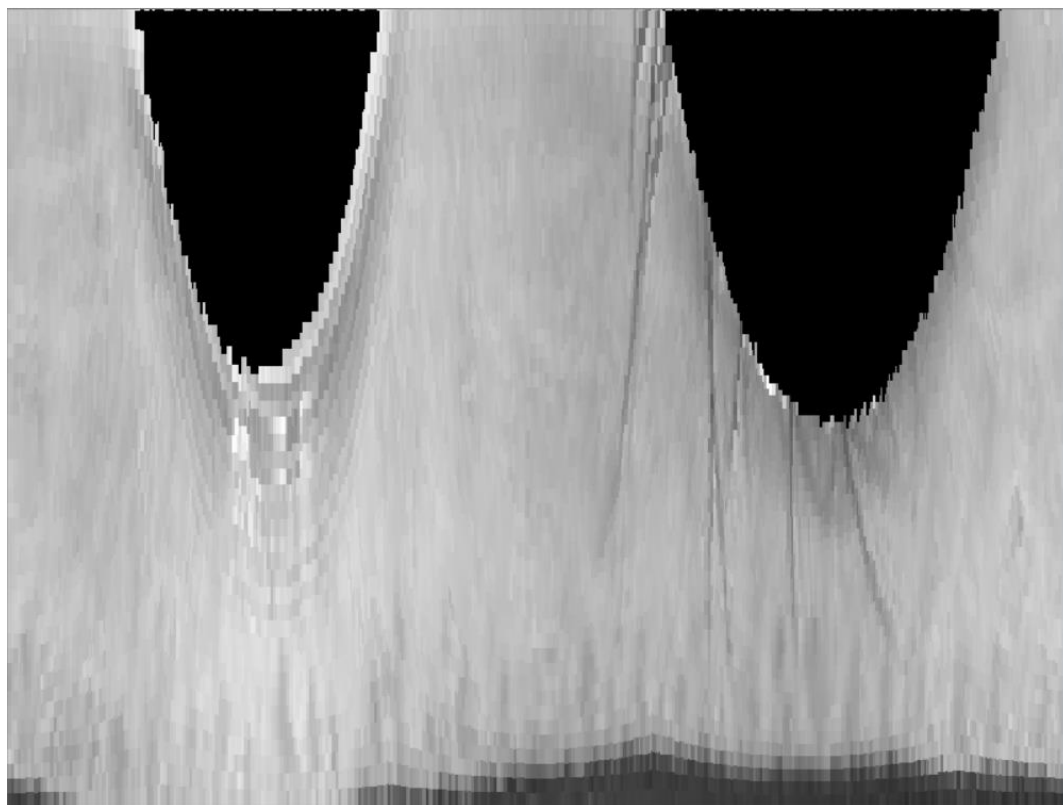
Túto masku následne aplikujeme na aproximačné body (Obr. 5) a získavame finálny výsledok (Obr. 8, Obr. 9, Obr. 10).



Obr. 8 Finálne výrezy 300x50



Obr. 9 Finálny výrez 600x150



Obr. 10 Finálny výrez 800 x 600