



Gépi Látás

GKNB_INTM025

2022/2023/1

Féléves feladat

Hercsel Péter

CSLOP1

Bevezető

A mindennapi életünkben, vásárláskor minden terméken található egy vonalkód amit a kasszánál a vonalkód olvasó lézerrel és a fényvisszaverődés/elnyelés elvén alapuló vonalkód olvasóval leolvassák a vásárolt termék kódját.

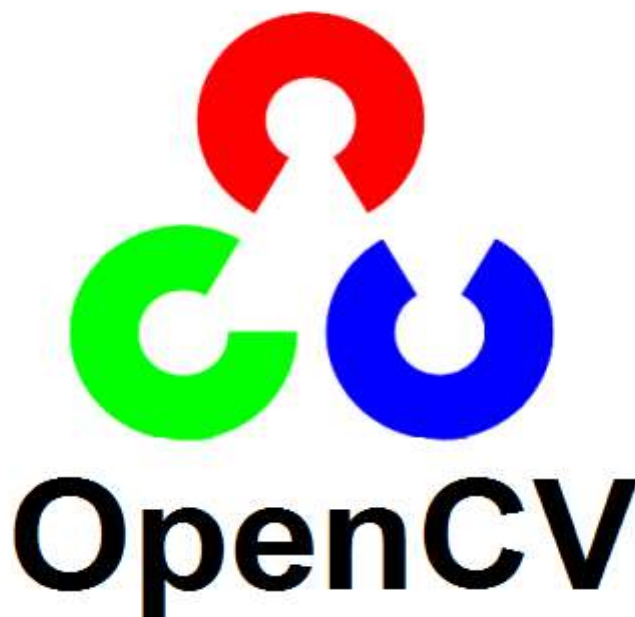
Ehhez a problémához kapcsolódik a feladatom ami az volt hogy megadott képeken, egy algoritmussal a lehető legnagyobb pontossággal megtaláljak egy vonalkódot. A különbség csak annyi, hogy én képeken/videón keresek vonalkódokat. Ehhez a gépi látás különböző technikáit használtam valamint egyéb automatizálási módszereket amivel végtelen számú teszt képet tudtam generálni. Erre azért volt szükség, mivel nem találtam megfelelő számú és minőségű teszt képet amin az algoritmusomat tudtam tesztelni.

A feladat főbb lépései a következőkből állnak:

- Betölteni a képet a megfelelő kódolásban
- Algoritmus lefuttatása
- Kiértékelés
- Statisztika készítése és Vizsgálata
- Algoritmus továbbfejlesztése

A megoldáshoz a Python programozási nyelvet használtam. Azért választottam ezt a nyelvet mert gyorsabban lehet módosítani a programon, valamint haladó szinten ismerem és használok majdnem napi szinten így számomra ez volt a logikus választás. Az alábbi modulok segítségét vettem igénybe a feladat megoldásához:

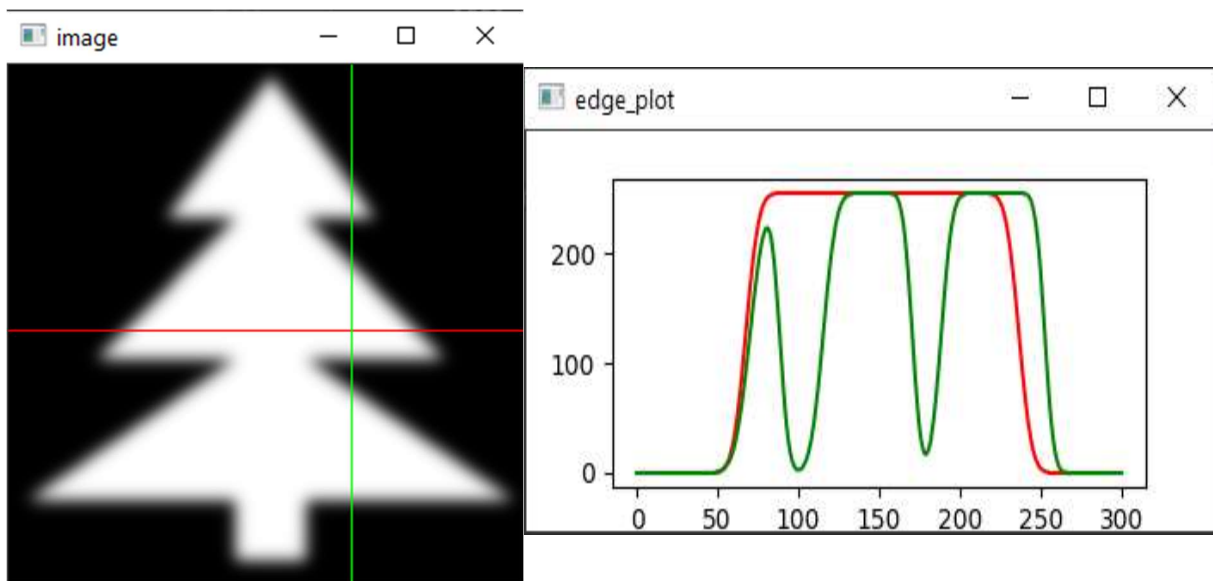
- openCV
- Numpy
- Os
- Random
- Pillow
- Barcode
- Svglib



Elméleti háttér

Éldetektálás:

- Él ott található a képen, ahol a **lokális intenzitáskülönbség nagy**.
- A képfüggvény **elsőrendű deriváltja** egy adott pontban (a gradiensvektor) mutatja a **legnagyobb változás (növekedés) irányát**, ami merőleges az élre (ha van az adott pontban), illetve a **gradiensvektor nagysága (magnitúdója)** jellemzi az **él erősségét**.
- A képfüggvény **második deriváltja** a változás változását mutatja. Ebben az esetben él a **derivált előjelváltásánál (nulla átmeneténél)** fordul elő.
- Digitális képen éleket lehet az **első- vagy másodrendű deriváltak** számításával keresni. Mivel a digitális képek diszkrétnek, ezért a deriváltszámítás is diszkrét közelítés lesz. Jellemzően X- és Y-irányú **parciális deriváltakat közelítünk konvolúciós kernellel**.
- A derivált közelítés nagyon **zajérzékeny**, ezért célszerű előzőleg képsimítást végezni a zaj hatásának csökkentésére. A simító és parciális deriváltat közelítő konvolúciók általában összevonhatók közös kernelbe, így egy lépésen végrehajthatók.
- Egy gyakorlatban jól használható éldetektor további feltételezésekkel is él, ilyen például a **Canny detektor** (simítás, deriválás, nem maximális élek elnyomása, hiszterézis küszöbölés).



Morfológiai szűrés

A matematikai morfológia eszközeivel számos alacsony szintű képfeldolgozó művelet végrehajtható. Most csak a **morfológiai szűréssel** foglalkozunk, amit a **dilatáció** és az **erózió** műveletek egymás utáni végrehajtásával valósíthatunk meg. A dilatáció "hizlal" a képi komponenseken, az erózió pedig erodál, csökkent rajtuk. Bináris és szürkeárnyaltos képekre is értelmezhetők.

A **dilatáció** megfogalmazható úgy, hogy a vizsgált képpontot az adott **környezetében előforduló intenzitások maximális értékével** helyettesítjük. Az **erózió** pedig a **minimális** értékkel.

A környezet alakját definiáló elemet **strukturáló elemnek** nevezik, ami többféle alakot felvehet, ahogyan majd látni fogjuk.

A morfológiai szűrést a dilatáció és erózió egymás utáni végrehajtásával valósíthatjuk meg, például az alábbi szerint.

- Dilatáció
- Erózió

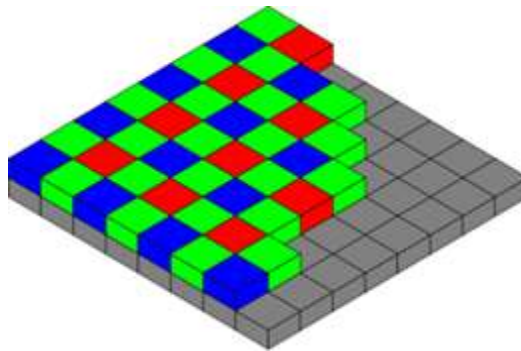


A környezetünkben található szilárd testek anyagával a rá eső fény kölcsönhatásba lép.

A hullámhossztartomány egy része elnyelődik, másik része a tükrözési szabályoknak megfelelően visszaverődik. Így alakulnak ki a tárgyak színei. Látás érzet vagy digitális kép akkor születik, amikor az így megváltozott fénysugarak élőlények látószervében, vagy fényképezőgép érzékelőjén nyelődnek el, ahol fényenergia mérés történik.

Az emberi látás képes színérzékelésre (kromatikus látás). A retinán úgynevezett csapok találhatók, amelyek háromféle típusúak lehetnek, és egy-egy hullámhossztartomány méréséért felelősek (vörös, zöld, kék, lásd az ábrán alul, a ciánkék kivételével). Ezek a fénymérések jutnak idegpályákon az agyba, ahol a színes látásérzet kialakul. Vegyük észre, hogy ez a fajta látás a fény energiájának egy jó részét nem méri. Azt is vegyük észre, hogy egy-egy retina képpontban (=egy csapban) nem a teljes színt látjuk, hanem csak annak egy komponensét. Az egymás melletti retina területeken található eltérő típusú csapok eredményéből áll elő az agyban az egyes pontok színes érzete.

Digitális képalkotás során az érzékelő méri a fényenergiát. A színek előállításához úgynevezett Bayer-szűrőt helyeznek a fény útjába, ami biztosítja, hogy az érzékelő kis mérőeszközeit eltérő hullámhossztartományok érik el (legtöbbször vörös, zöld és kék elrendezést alkalmaznak, de léteznek más szűrő megvalósítások is). A színes RGB képet interpolációval kapjuk a nyers mérési eredményekből, mivel egy-egy képpontban a nyers mérés csak egy komponenst tud figyelembe venni (hasonlóan az emberi látáshoz). Az emberi látáshoz hasonlóan itt is kárba vesz a fényenergia egy jelentős része, ami a szűrő szűkebb tartományán kívül esik.



Az emberi retinán a csapok mellett pálcikák is találhatók. Ezek gyenge fényviszonyok esetén működnek, viszont csak egyféle tartományban mérnek (a fenti ábrán a csapok mellett a ciánkék érzékelési terület), így monokromatikus (szín nélküli) látást eredményeznek.

A digitális érzékelők is képesek monokromatikus érzékelésre, amennyiben a Bayer-szűrőt nem építik bele. Ekkor itt is csak monokromatikus mérésre van lehetőség, színérzékelésre nincs. De a fényenergia nagyobb százaléka tud részt venni a képalkotásban, ami gyenge fényviszonyok között fontos lehet.

A monokromatikus érzékelés tehát egy skalár értékkel, az **intenzitással** jellemezhető. A kisebb energiaszint kevés fényt, vagyis sötétebb régiót, a nagyobb energiaszint több fényt, így világosabb részt jelent. **Értelmezhetjük ezt a fekete (minimális érték) és fehér (maximális érték) közötti szürkeárnyaltos átmenetként.**

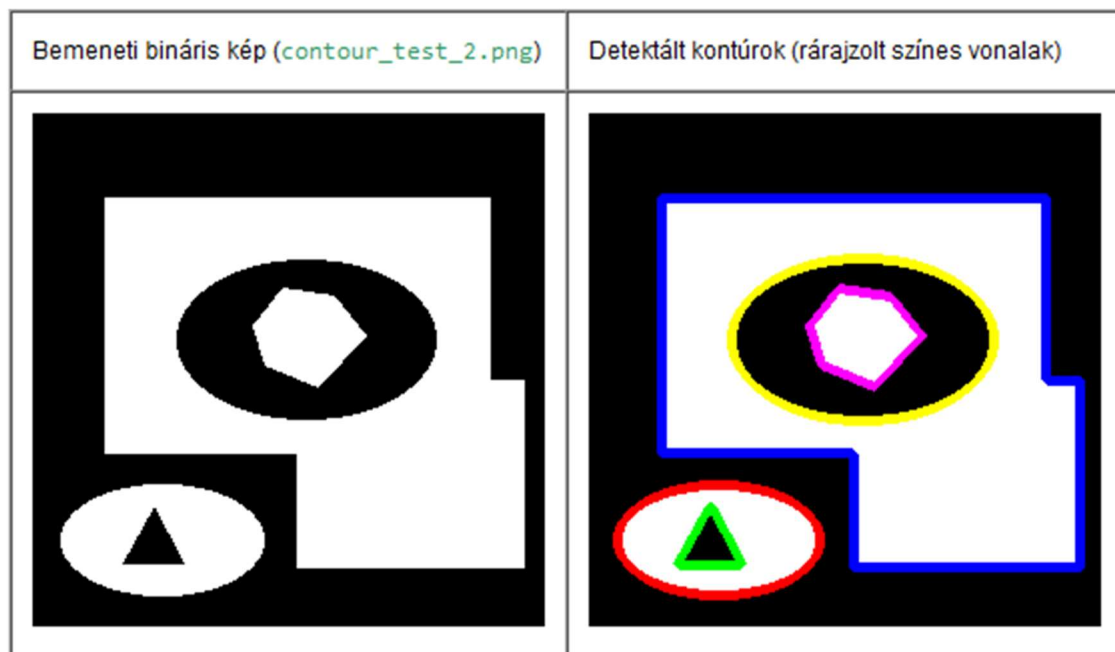
Kontúr detektálás

Az OpenCV lehetőséget biztosít (pl. szegmentálással előállított) **bináris képeken** az objektumok **körvonalának** meghatározására, illetve a **kontúrok képre rajzolására**. Ha szürkeárnyaltos vagy színes képünk van, akkor a bináris bemenet előállításához szükséges előzetesen küszöbölést vagy (bináris) szegmentálást végezni a képen. Ha ezt nem tesszük, akkor hibaüzenetet nem kapunk, viszont **a szürkeárnyaltos bemenetet binárisként kezeli** az OpenCV:

0: háttérpont,
>0: objektumpont.

A bináris képek megjelenítése, és a velük való maszkolási lehetőségek könnyű elérése miatt a jegyzet példáiban a 255 (fehér) intenzitásértéket használjuk az objektumpontokra.

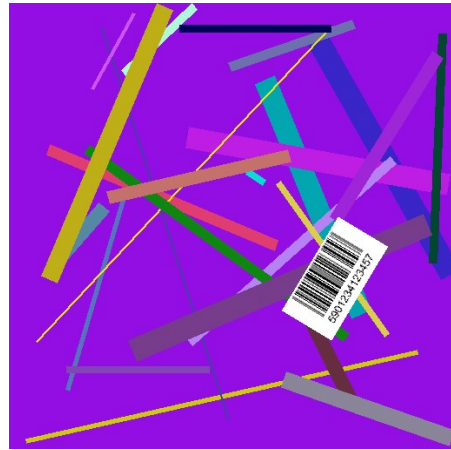
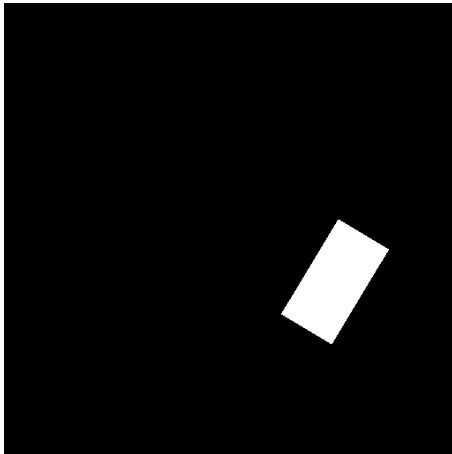
Kontúrpontok azok az objektumpontok lesznek, amelyeknek van háttérpont szomszédja. (Az OpenCV dokumentáció nem részletezi, milyen szomszédsági relációt vesz figyelembe itt.)



A megvalósítás terve és kivitelezése

Főbb lépések:

1, Képek generálása, amin a teszteket lefuttatjuk. Ebben a lépésben egyszerre generáljuk a képeket és a teszteléskor használt maszkot, ahol a generáló algoritmus megmondja hova helyezte el a vonalkódot.

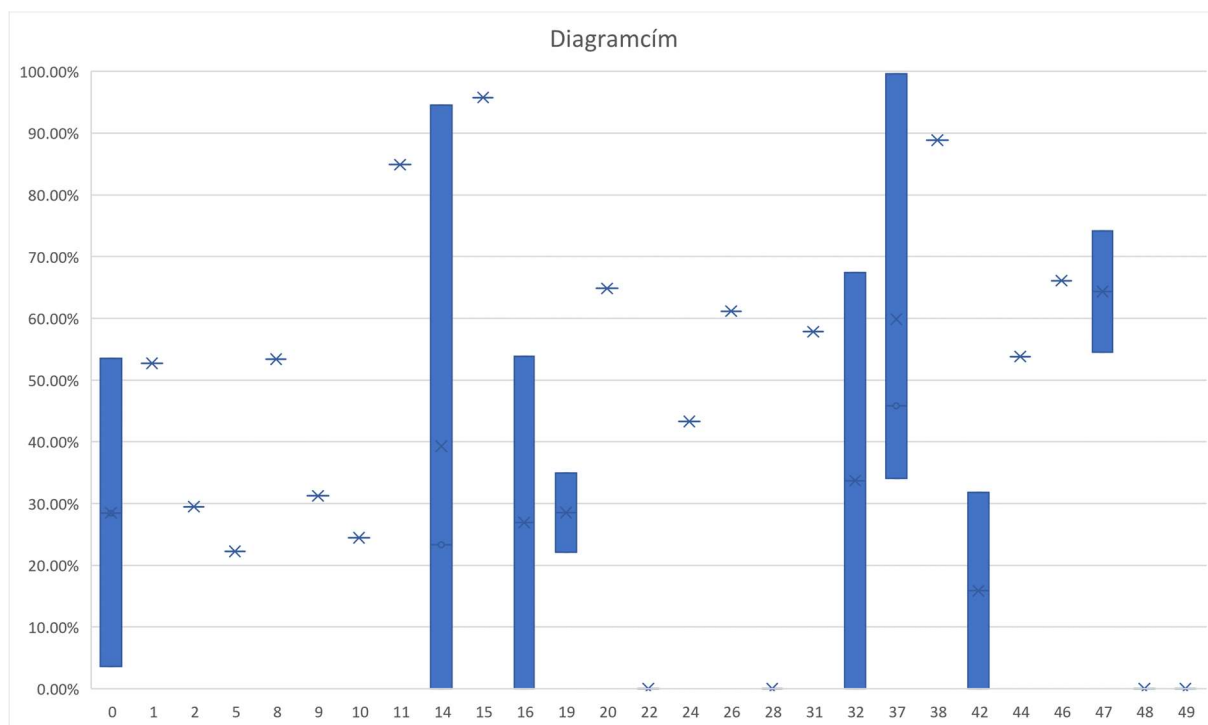


2, Ezen képek beolvasása után a jobb oldali képen elvégzi az algoritmus a lépéseket, ami alapján megpróbálja megtalálni a vonalkódot. Ezek a lépések sorrendben a következők:

- Szürke árnyalatosítás
- Éldetektálás Canny módszerrel
- Vonalak keresése paraméterezett Hough módszerrel
- Vonalak kirajzolása egy fekete képre
- Medián simítás alkalmazása 7-es kernellel
- Morfológus szűrés használata
 - o 10 dilataciós iteráció
 - o 15 eróziós iteráció
- Újabb élek keresése
- Kontúrok megrajzolása
- Kontúrok befogó négyzetének berajzolása
- A bemeneti négyzet és a befogó négyzet összehasonlítása

$$\frac{100 * \sum(BemenetiNégyzet \cap BefogóNégyzet)}{\sum BemenetiNégyzet}$$

Eredmények



Irodalomjegyzék

<https://www.inf.u-szeged.hu/~tanacs/pyocv/>