

Feladatkiírás

A vékonyítás iteratív objektumredukción alapuló vázkijelölő módszer. Az irodalomban ismeretes vékonyító eljárások többsége csak bináris képeken használható. Azonban a szürkeárnyaltos képi objektumok binárisá alakítása információ veszteséggel jár, ami bizonyos torzulásokat okozhat a kinyert vázban. Emiatt gyakran célszerűbb lehet közvetlenül a szürkeárnyaltos képeken elvégezni a vékonyítást. A hallgató feladata néhány ilyen algoritmus implementációja és összehasonlítása tesztképeken.

Tartalmi összefoglaló

- **A téma megnevezése:**

Vékonyító algoritmusok implementálása szürkeárnyaltos képeken.

- **A megadott feladat megfogalmazása:**

A feladat néhány olyan vékonyító algoritmus tanulmányozása, implementációja és összehasonlítása, amelyek szürkeárnyaltos képeken működnek.

- **A megoldási mód:**

Az öt publikált tudományos cikk implementálására hasonló módon jártam el. A képen végig pásztázva kijelölve a határpontokat, ezután pedig a törlésre alkalmas pontokat, melyeket utána töröltem is. Ezzel a módszerrel hajtottam végre a vékonyítást, melynek részleteit a cikkek taglalták. Az eljárás addig folytatódik, amíg van törölhető pont.

- **Alkalmazott eszközök, módszerek:**

Az algoritmusokhoz Python programozási nyelvet használtam, az OpenCV csomag segítségével.

- **Elért eredmények:**

Öt, egymástól lényegesen különböző vékonyító eljárást implementáltam, melyek teszteléséhez több különböző szürkeárnyaltos képet válogattam össze. A teszteredmények alapján az algoritmusok viselkedése összhangban van az irodalomban leírtakkal.

- **Kulcsszavak:**

Képfeldolgozás, vázkijelölés, vékonyítás

Tartalomjegyzék

Feladatkiírás	2
Tartalmi összefoglaló	3
Tartalomjegyzék.....	4
BEVEZETÉS	6
1. A DIGITÁLIS TOPOLOGIA ALAPFOGALMAI	7
1.1. Digitális kép fogalma, jelölések	7
1.2. Összefüggőség, komponensek.....	8
1.3. Topológia megőrző műveletek.....	9
1.4. Egyszerű pont	11
1.5. Szürkeárnyaltos képek.....	11
2. A VÁZKIJEJELÖLÉS	13
2.1. A váz fogalma	13
2.2. Vékonyítás.....	13
2.2.1 Vékonyítás bináris képeken	14
2.2.2 Vékonyítás szürkeárnyaltos képeken	14
3. MEGVALÓSÍTOTT ALGORITMUSOK	17
3.1. Dyer-Rosenfeld-féle algoritmus	17
3.2. Salari-Siy-féle algoritmus	19
3.3. Szürkeárnyaltos karakterek vékonyítása.....	23
3.4. Morfológia alapú vékonyítás	26
3.5. Couprie-Bezerra-Bertrand-féle algoritmus	27
4. IMPLEMENTÁCIÓ.....	30
4.1. Képek beolvasása.....	30
4.2. Képek bejárása és mátrixok	30
4.3. Keresőtábla	30

4.3. Képek elmentése	30
5. EREDMÉNYEK	31
Irodalomjegyzék	36
Nyilatkozat	37
Köszönetnyilvánítás	38

BEVEZETÉS

A digitális képfeldolgozás területén az objektumok egyik alakleíró jellemzője a váz , amely fontos információkkal szolgál több tudományos területen is. Ilyenek például a karakterfelismerés, kromoszómák vizsgálata, ujjlenyomat vizsgálat. A vázkijelölés egyik lehetséges módszere a vékonyítás, mely az alakzat topológiai tulajdonságait megőrizve határozza meg a vázat.

A vékonyító algoritmusok többsége bináris képeket feltételez, tehát a vázkijelöléshez általában a szürkeárnyaltos képek előzetes szegmentálására (például küszöbölésre) van szükség. Azonban a binarizálás többnyire információvesztéssel jár együtt, ezért ezt a műveletet célszerűbb a képfeldolgozási folyamatnak a lehető legkésőbbi fázisában elvégezni. Ebből a megfontolásból az irodalomban többen is javasoltak olyan eljárásokat, amelyek szürkeárnyaltos objektumok vázát határozzák meg.

Az irodalomban a bináris váz fogalmának többféle kiterjesztését is javasolják, ebből adódóan a szürkeárnyaltos képekre értelmezett vékonyító eljárások is lényegesen eltérően működhetnek. Szakdolgozatom keretében ezért több ilyen algoritmust összehasonlítottam és implementáltam.

1. A digitális topológia alapfogalmai

A topológia a matematikának azon részterülete, amelyik az alakzatoknak a folytonos deformációk közben is megmaradó invariáns tulajdonságaival foglalkozik. Az 1.2-1.6 fejezetekben tárgyalt fogalmak a bináris képekre vonatkoznak, ezeknek a fogalmaknak a kiterjesztése szükséges ahhoz, hogy a szürkeárnyaltos képeken is értelmezhetők legyenek.

1.1. Digitális kép fogalma, jelölések

Digitális, vagy bináris képnek nevezzük az olyan (V, m, n, B) képeket, ahol V a képpontok halmaza, $B \subseteq V$ a fekete pontok halmaza, $W = V - B$ a fehér pontok halmaza, m a fekete pontokra érvényes szomszédság, n pedig a fehér pontok szomszédsági relációja.

A digitális képet az $f: V \rightarrow D$ leképezéssel tudjuk megadni, ahol D egy képpontnak adható színintenzitások halmaza. V és D egyaránt véges és diszkrét halmazok [6].

A bináris képeken, ahogy azt a neve is jelzi, csak két intenzitás jelenik meg, ebben az esetben $D = \{0, 1\}$. Ezek a fehér és a fekete intenzitások (0 és az 1).

A szürkeárnyaltos képeken ezzel szemben, $D = \{0, 1, 2, \dots, 255\}$. Az intenzitás halmaz bővítésével tudjuk definiálni a szürkeárnyaltos képeket.

A képfeldolgozásban ahhoz, hogy ilyen alakzatokon transzformációkat tudjunk elvégezni, szükségünk van egy jelölésrendszerre, amellyel hivatkozni tudunk az egyes pixelekre.

A képeken megjelenő képpontokat indexeléssel azonosítjuk. Ezek alapján tudunk később hivatkozni rájuk, a könnyebb elérés érdekében. Indexeléshez a képpont sora alkotja az első, oszlopa pedig a második koordinátát. Ezzel a módszerrel kapunk egy képmátrixot, melynek y tengelye a sort, x tengelye pedig az oszlopot határozza meg. Az origó ilyenkor a bal felső sarokban helyezkedik el. A későbbiekben x és y jelekkel fogok utalni a képpont koordinátáira, ahol M az x -tengely legnagyobb értékét, N pedig az y -tengely legnagyobb értékét jelöli.

A bináris képekhez több alapvető fogalom tartozik, amelyek megértése szükséges a további fejezetek értelmezéséhez. *Objektumpontnak*, vagy *fe fekete komponensnek* nevezzük azokat a képpontokat, melyek egy azonos színű, összefüggő objektumot jelölnek. A véges képen *háttérnek* nevezzük a végtelen fehér komponenst, amelyből csak egy lehet. *Üregnek* nevezzük a véges fehér komponenst, amelyből több is lehet.

Egy P_0 pont 8-szomszédaira a P_1, P_2, \dots, P_8 jelölésekkel hivatkozunk, ahol az egyes szomszédok az 1.1.1 ábrán látható módon helyezkednek el P_0 -hoz képest. A P_i pont intenzitását a $V(P_i)$ -vel jelöljük.

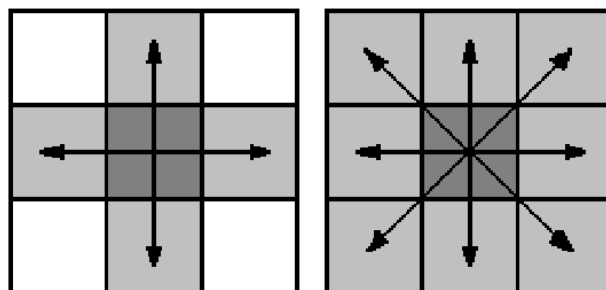
P_4	P_3	P_2
P_5	P_0	P_1
P_6	P_7	P_8

1.1. ábra: Egy P_0 pont 8-szomszédainak jelölése

Útnak azokat az összefüggő képpontok halmazát értjük, melyek úgy kötnek össze két objektumpontot, hogy nincs köztük szakadás (háttérpont). A következő alfejezetben az összefüggőségről lesz szó és arról, hogy hogyan kapcsolódnak egymáshoz a képpontok.

1.2. Összefüggőség, komponensek

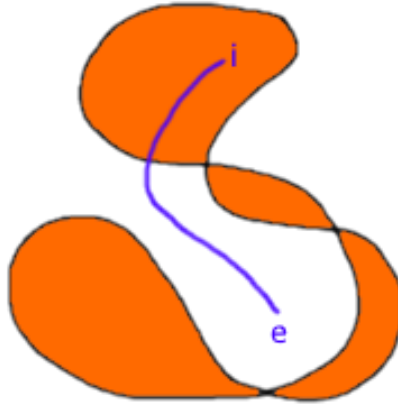
Azt mondjuk, hogy két $p, q \in S$ pixel összefügg az S halmazban, ha létezik egy $p_0 = p, p_1 \dots p_n = q, p_i \in S$ pixelsorozat, amely összeköti p -t és q -t úgy, hogy p_i és p_{i-1} szomszédok. A képpontok között kétféle szomszédsági relációt szokás megkülönböztetni (lásd 1.1 ábra). A 4 szomszédság esetében a p középpontnak azok a szomszédjai, amelyek tőle a négy fő égtájra helyezkednek el. A 8 szomszédság pedig a p középpontnak azok a szomszédjai, amelyek tőle a négy égtájra és átlósan helyezkednek el.



1.2. ábra: 4 szomszédság (balra), 8 szomszédság (jobbra)

Ezek a szomszédok több összefüggő fekete komponenseket határoznak meg, melyek a megadott szomszédság szerint kapcsolódnak egymással.

Jordan tétele szerint, egy egyszerű zárt görbe a sík két diszjunkt részhalmazzra osztja. A belsőre (interior) és a külsőre (exterior) oly módon, hogy az interior egyetlen pontja sem köthető össze az exterior egyetlen pontjával sem a görbe metszése nélkül.



1.3. ábra: Jordán-tétel

A tételnek teljesülnie kell a digitális világban is, így figyelembe kell vennünk mely fekete és mely fehér pontokat minősítünk összekötöttnek.

Ennek függvényében alkalmazzuk a négy, illetve a nyolc szomszédságot a fekete és a fehér pontokra külön-külön. A (8, 4) és a (4, 8) képeken biztosítható a diszkrét Jordán-tétel teljesülése [6].

A fekete objektumok által közrefogott részeket nevezzük üregeknek.

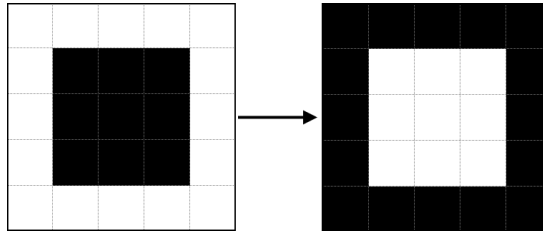
Az objektumon belül elhelyezkedő pontokat, melyek nincsenek szomszédságban egy háttérponttal sem, nevezzük belső pontoknak. Izoláltnak nevezzük azokat a pontokat, melyek önmagukban alkotnak objektumot, összefüggő szomszédok nélkül.

1.3. Topológia megőrző műveletek

A topológia megőrző műveletekre elmondható, hogy az elvégzésük után az alakzatok ugyan azokkal a topológiai tulajdonságokkal rendelkeznek, mint a művelet előtt.

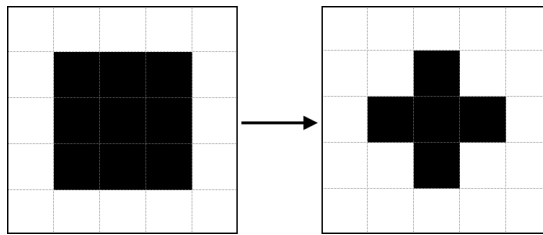
Az ilyen képműveletek a fekete pontok fehérre színezését, és a fehér pontok feketére színezését jelentik. Az ilyen műveleteknek több változata is van.

Az egyik ilyen művelet a negálás, amikor egyszerűen csak átszínezzük a képpontokat az ellenkező intenzitásúra.



1.4. ábra: negálás

Az egyike a leggyakrabban alkalmazott képműveleteknek az redukció, mellyel valamilyen szabály szerint veszünk el fekete pontokat a kiindulási képből, ezzel csökkentve a fekete pontok számát.

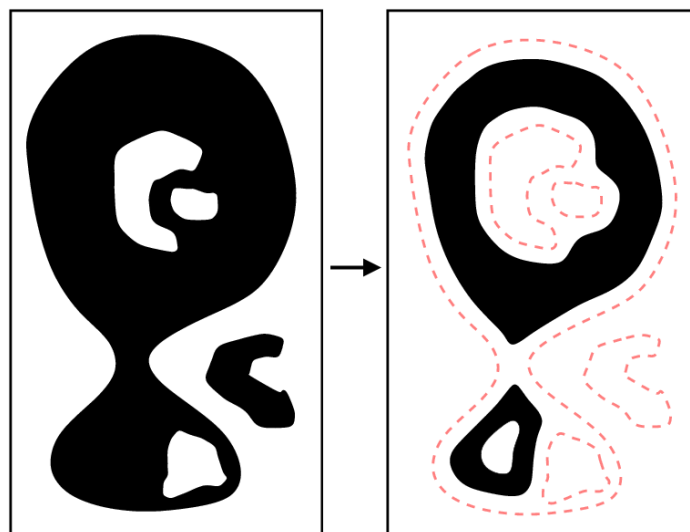


1.5. ábra: redukció

Egy redukció topológia-megőrző, ha az alábbi feltételeknek eleget tesz:

- Nem szakít szét objektumot.
- Nem töröl teljesen objektumot.
- Nem olvaszt össze üreget másik üreggel, sem pedig a háttérrel.
- Nem hoz létre új üreget

Egy ellenpéldát mutat be az 1.10 ábra, ahol minden feltétel meg van szegve.



1.6. ábra: Nem topológia-megőrző redukció [6]

1.4. Egyszerű pont

Az alábbi tétel szerint egy pont egyszerű volta annak 3×3 -as környezete alapján eldönthető, tehát az egyszerűség egy lokális tulajdonság. Az egyszerű pont olyan objektumpont, melynek törlése topológia megőrző redukció.

Tétel:

A $p \in B$ pont akkor és csak akkor egyszerű a $(Z^2, 8, 4, B)$ képen, ha határpont és $C_8(p) = 1$. [6]

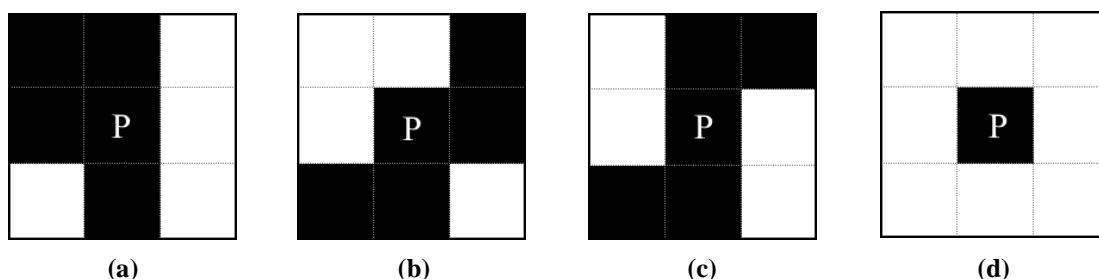
Ahol $C_m(p)$ a fekete m -komponensek száma $N^*_8(p)$ -ben.

Jellemezni az egyszerű pontokat a következőképpen lehet:

A pont határpont, és a 8 szomszédságában, csak egy 8-komponenst tartalmaz.

Másszóval csak azok a pontok számítanak egyszerűeknek, amelyek törlésével az objektum megtartja topológiai tulajdonságát.

Ha egy redukciós képműveletről beszélünk, akkor azok a pontok számítanak egyszerűnek, melyek törlésével nem változik az objektum topológiája, ha pedig egy addíciós képműveletről, akkor pedig azon pontok, melyek hozzáadása érhető el ugyan ez az eredmény.



1.7. ábra: (a) egyszerű (b) egyszerű (c) nem egyszerű (d) nem egyszerű

1.5. Szürkeárnyaltos képek

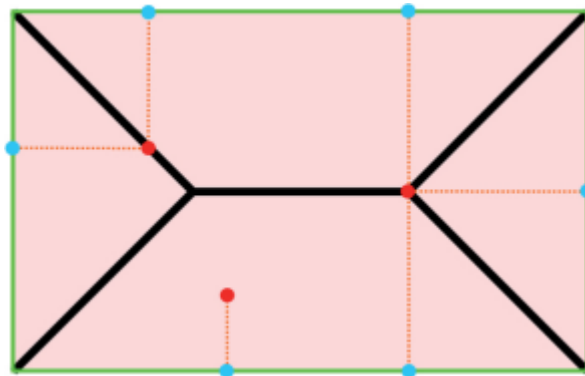
Az eddig említett szabályokat átkonvertálni a szürkeárnyaltos képek világába nem könnyű feladat. A feldolgozott cikkek szinte mindegyike külön módszert alkotott, a saját szabályainak megfogalmazására. Ehhez mérten, kiválogatva a közös tényezőket lehet elmondani a következőket, egy sajátos jelölésrendszerrel:

- A szürkeárnyaltos képek esetében a képpontokhoz intenzitást rendelünk, mely a pont szürkeségét fogja megadni. Ez az érték a 0 és 255 értékek között mozog. Megjegyzés alapján a 255-ös érték fogja a fekete, a 0 pedig a fehér színt jelölni.
- A pixelek egy Z^2 halmaz elemei, mely a sorok és oszlopok 0-tól a végtelenbe haladó számegyenesén kaphatnak értéket.
- A képen egy $F(x, y)$ függvény minden (x, y) koordinátájú pixelhez hozzárendeli a pixel intenzitását.

2. A vázkijelölés

2.1. A váz fogalma

A váz egy gyakran alkalmazott régió-alapú jellemző, mely összegzi az objektumok alakját, formáját. Kinyerni a középtengely-transzformációval tudjuk, amely az objektum minden egyes pontjára megkeresi a hozzá legközelebbre eső határpontokat. Ha az eljárás egy belső pontra egynél több legközelebbi határpontot talál, akkor azt a vázhoz tartozónak, vázpontnak minősíti (2.1 ábra) [9].



2.1. ábra: A váz, középtengely-transzformáció szerint

A vázra jellemzően az mondható el, hogy az objektum közepén helyezkedik el. Vékony, általában egy-két pixel vastagságú és ezek mellett reprezentálni tudja az objektum általános formáját, topológiai szerkezetét.

Egy objektum vázának a meghatározását a bináris világban könnyen el tudjuk képzelni, a szürkeárnyaltos képeken azonban nem egyértelmű, hogy a képpontok intenzitása milyen módon befolyásolja a váz kialakulását. A szakirodalomban nincs konkrét megfogalmazása a szürkeárnyaltos vázaknak, így kénytelenek vagyunk a vékonyítás módszerének leírására hagyatkozni. Itt azonban már találunk hasonlóságokat a feldolgozott cikkekben.

2.2. Vékonyítás

A vékonyítás a vázkijelölés iteratív objektum-redukción alapuló technikája, mely lépésenként az objektum határpontjai közül a törölhető pontokat eltávolítja. Addig fut az eljárás, amíg nem marad törölhető pont az objektumon. Az eljárás eredménye a bemeneti kép objektumának váza. A vékonyítási eljárás szemléletes példája a prérítűz hasonlat, mely során az objektum szélein lévő képpontokat meggyújtjuk, és lépésről lépésre kicsinyítjük az objektumunkat.

Ahol a tűzfrontok találkoznak, ott kioltják egymást. A váz a kioltott képpontokból alkotott objektum lesz.

2.2.1 Vékonyítás bináris képeken

A bináris képek vékonyítására elegendő három feltételt megadni, melyek biztosítják az objektum topológiájának megőrzését.

2. Tétel:

Egy T párhuzamos redukció topológia megőrző a $(8,4)$ képeken, ha:

- T csak egyszerű pontokat töröl;
- ha T a p és q egymással 4-szomszédos pontokat törli, akkor $\{p,q\}$ egyszerű halmaz;
- T nem töröl (teljesen) egység rácsnégyzet által lefedett objektumot.

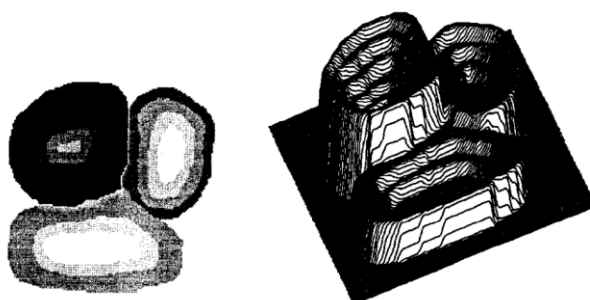
2.2.2 Vékonyítás szürkeárnyaltos képeken

Mivel a szürkeárnyaltos objektumok vázát többféleképpen is értelmezhetjük, így ezen objektumok vékonyítására is többféle megközelítést javasoltak. Az egyik ilyen módszer, ami a feldolgozott cikkekben többször is előfordul az az összefüggőség vizsgálata az egyes törlendő pontok szomszédságában [1]. Ezt az eljárást először A. Rosenfeld és C. R. Dyer alkalmazta a saját szürkeárnyaltos vékonyító algoritmusuk során. Az általuk bevezetett súlyozott összefüggőség koncepciója arra utal, hogy két pont csak akkor számít összefüggőnek, ha az őket összekötő úton nem találunk egyik pontnál sem kisebb intenzitású képpontot (3.2 ábra). Másszóval nem lehet gyengébb értékkel összekötni két pontot, mint a saját intenzitásuk. Ez az összeköttetés szolgál arra, hogy törlés esetén ne válaszunk el egymástól összefüggő képpontokat és lokálisan meg tudjuk állapítani mely képpontok tartoznak egymáshoz és melyek azok, amelyeket szétválaszthatunk.

50	48	40
50	50	42
49	46	43

2.2. ábra: Jobboldali képpontból (42) út a baloldali képpontba (50), ahol a zöld útvonalon nincs a két képpont alá menő érték, a pirosnál viszont van

Erre a kérdésre egy másik megközelítés a terepmodell alkalmazása, amely globálisan közelíti meg a problémát. A szürkeárnyaltos képet úgy kezeli, mint egy háromdimenziós terepmodellt, melyen vannak hegytetők, hegygerincek, üregek és lapos síkságok [7]. A képet kiterítve egy két dimenziós lapra és a képpontok intenzitását magassággal jelölve alakulnak ki ezek az alakzatok. Akárhány alakzat csoportot is alakít ki magának egy-egy cikk, mindegyikkel külön-külön foglalkozik és hajt végre eljárásokat annak érdekében, hogy megtalálja az objektum hegygerincét, ugyanis ez fogja az alakzat vázát alkotni.



2.3. ábra: Terepmodell [7]

Akármelyik módszert is alkalmazzuk, így már meg tudjuk ítélni, mely pontok kerülnek törlésre az adott esetekben. Az azonban, hogy hogy végezzük el a törlést, ismét több módszerrel tudjuk meghatározni. A bináris egyről nullára állító törlés helyett, szürkeárnyaltos képeken három külön módszert alkalmaznak az algoritmusok. A legelső esetben, hasonlóan a binárishoz a képpont intenzitását nullára állítjuk, nem törődve a környezetével.

A második megközelítés szerint a képpontot a háromszor hármass környezetében lévő képpontok közül kiválasztjuk a legkisebbet és azt az értéket adjuk a törölt pontnak, amivel biztonságosabbá tesszük az algoritmusunkat. Így szintén hirtelen nullára válthatjuk a képpontunkat, ha érintkezik nullaintenzitású ponttal, viszont, ha a határpontunk

kiválasztásakor nem csak a nullaintenzitású pontokkal érintkező pontokat jelöljük ki, ezzel akár megoldhatjuk az üregek keletkezését.

A harmadik törlés a legbiztonságosabb, ez eredményezi a leginkább fokozatos intenzitáscsökkentést. Ebben az esetben szintén megvizsgáljuk a pont háromszor hármas környezetét és a vizsgált képpontnál kisebb intenzitással rendelkező képpontok közül választjuk ki a legnagyobb intenzitásút. Így még a határpontok is kaphatnak nullánál nagyobb intenzitású értéket.

Ezeknek a módszereknek a használatával tudjuk a feltételeket átátváltani binárisról szürkeárnyaltosra.

3. Megvalósított algoritmusok

A könnyebb összehasonlíthatóság érdekében mindegyik algoritmusnak a (8,4) képekre működő változatát valósítottam meg.

3.1. Dyer-Rosenfeld-féle algoritmus

Az első szürkeárnyaltos képekre is kiterjedő vékonyító algoritmust 1977-ben javasolták [1]. A cikk bevezetése szerint a szürkeárnyaltos objektumok határpontjait törléssel tudjuk vékonyítani, ha nem szakítjuk szét a lokális összeköttetésben lévő szomszédiaktól, megelőzve az elkülönülő pontok eltűnését és az objektumok szélén lévő pontok zsugorodását. Az előző fejezetben általánosított összeköttetést annyival bővítették ki, hogy belevettek egy R értéket, mely a vizsgált környezet szürkességét határozza meg.

$$R = \max(N_3, N_5, N_0, N_1, N_7) - \min(N_3, N_5, N_0, N_1, N_7) + 1$$

Valamint megadunk egy R' értéket, amely a következő tartományból választhatunk neki értéket:

$$0 \leq R' \leq R$$

Ebben az esetben a törlés alatt, a törlendő pont környezetében lévő pontok minimumára állítását értjük.

Az algoritmus végrehajtásához először a határpontok kijelölésére van szükség. Ezt egy egyszerű pixel bejáró módszerrel tettem meg, mely azokat a pontokat jelöli ki, amelyeknek szomszédjai között van nála kisebb intenzitással rendelkező képpont.

Pszeudokódja:

```
for minden P=(x,y) képpontra
    ha van olyan P-vel 4-szomszédos (x',y') pont, amelyre
        F(x',y') < F(x,y)
    akkor P megjelölése
```

A megjelölt képpontokat elmentjük egy mátrixba, melynek nagysága megegyezik a kép méretével. A határpontok kiválasztásánál fontos megemlíteni, hogy ahhoz, hogy az objektum ne szakadjon szét, egyszerre csak egy oldalról szabad elvégeznünk a vékonyítást. Így az algoritmusunk szekvenciálisan fog működni, oldalanként. A párhuzamosságra törekedve bevezethető az egymás melletti oldalak egyidejű törlése. Erre azért van szükség, mert a kettő

pixel vastagságú objektumok könnyedén eltűnhetnek a párhuzamos megvalósítással, mivel a törlési feltételek nem terjednek ki erre az esetre.

Arra is figyelniünk kell, hogy az egy pixel nagyságú objektumok ne tűnjenek el és az objektum szélén elhelyezkedő képpontok ne menjenek össze. Ennek a megakadályozására szükséges az a végpontfeltétel, amely meggátolja az olyan objektumpontok törlését, mely kettőnél kevesebb szomszédal rendelkezik.

Miután kijelöltük a határpontokat, minden egyes képpontra meg kell vizsgálnunk az alábbi feltételeket:

1. Legalább kettő pontnak az intenzitása nagyobb-egyenlőnek kell lennie $N_0 - R'$ értékénél az alábbi képpontok közül: N_3, N_5, N_1, N_7 (Ez a feltétel egyezik meg azzal a kikötéssel, hogy kettőnél kevesebb szomszédal rendelkező objektumpontok nem törölhetők)
2. Minden párra az N_3, N_5, N_1, N_7 képpontok közül, jelölje m a pár minimumát. Ezután vagy $N_0 < m - R'$, vagy található egy út a két képpont között N_0 érintése nélkül. Az út minden egyes pontjának az intenzitása meg kell haladja, vagy egyenlőnek kell lennie $m - R'$ -vel. (Ez a feltétel felel meg annak a kikötésnek, hogy a képpontokat nem szabad szétszakítanunk a szürke szomszédjaitól)

A két feltételt ellenőrizniük kell minden egyes kijelölt határponton. Amennyiben ezek a feltételek teljesülnek, csak akkor szabad törölnünk a kijelölt képpontot.

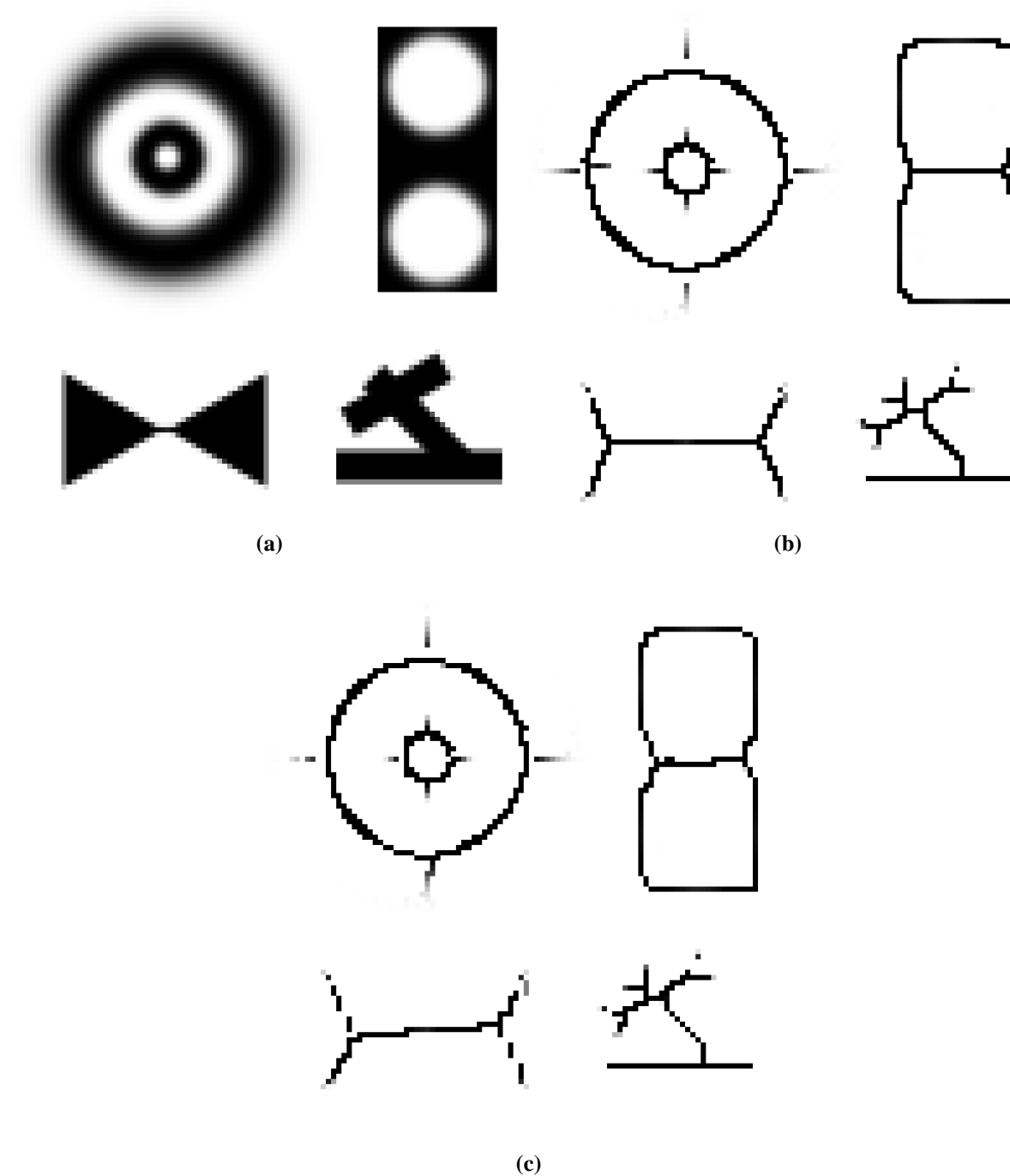
A program abból a szempontból párhuzamos, hogy az egyes határpontok kiválasztása után a törlést nem azonnal hajtjuk végre, hanem a törlésre szintén megjelöljük a pontokat, hasonlóan, mint a határpontok kijelölésénél. Ezek után egyszerre töröljük ki az összes kijelölt pontot.

Az eljárás pszeudokódja:

```
for minden  $P=(x,y)$  képpontra  
    if 1. feltétel és 2. feltétel teljesül:  
        then oldalanként a törölhető pontok megjelölése
```

A cikk megemlíti, ha egyszerre több oldalról történik a vékonyítás, szétszakadhat az összeköttetés. Abban az esetben fordulhat ez elő, ha a középpont kisebb értékkel rendelkezik, mint az összeköttetendő szomszédok és csak a középpont kötné őket össze.

Eredmény:



3.1. ábra: (a) Eredeti kép (b) Dyer-Rosenfeld algoritmus eredménye oldalanként törléssel (c) és a két egymás mellett lévő oldalak törlésével párhuzamos módon

3.2. Salari-Siy-féle algoritmus

A második feldolgozott algoritmust Ezzatollah Salari és Pepe Siy alkotta meg 1984-ben, akik már az előzőleg bemutatott algoritmus ismeretével alkották meg saját megoldásukat [2]. A bináris esetben egzakt módon lehet definiálni a határpontokat, ezzel szemben a

szürkeárnyaltos képek határpontjai nincsenek jól definiálva, mivel a határpontoknál alakulnak ki legtöbbször a halvány átmeneti részek és a zajos hátterek.

Elmondásuk alapján azt tapasztalták, hogy a szürkeárnyaltos képek vékonyítása során a következő módszerek alakultak ki. A képpontok összeköttetését vizsgáló algoritmusok, melyek a lokális összeköttetést biztosítják és a morfológiai transzformációkat alkalmazó megoldások, melyek szintén nem eredményeznek összefüggő vázakat.

Megemlítik a Dyer-Rosenfeld-féle algoritmust, mely szintén nem tartja meg globálisan az összefüggőséget.

Mindezek ismeretében arra jutottak, hogy egy hegygerinc kereső algoritmust fognak megvalósítani. A bemeneti képen minden szürke pont objektumpontnak és minden fehér pont háttérpontnak minősül. Egy bizonyos szint alatt minden pontot háttérpontnak minősítenek, tehát a zajos háttértől megszabadulnak. A küszöbérték nincs meghatározva és véleményük szerint a váz nem is függ ettől az értéktől.

Az algoritmus legfontosabb lépéseit az alábbi állításokra alapozták:

1. A keresett váz a kép hegygerincén helyezkedik el, amire nincs kihatással a nem jól definiálható határpontok összessége.
2. Az összefüggőség globális megtartása.
3. A váznak egy pixel vastagságúnak kell lennie.

Maga az algoritmus két részre osztható. Az előfeldolgozás részére, amely eltünteti az átmeneti határrészt és magára a vékonyításra. A vékonyítás folyamata alatt az alábbi definíciókat alkalmazzák:

1. **Definíció:** *Határpontnak* N_0 akkor minősül, ha a háromszor hármass környezetében intenzitása nem nulla és van legalább egy nullaértékű szomszédja.
2. **Definíció:** *Végpont* N_0 , ha értéke nem nulla és csak az egyik szomszédja rendelkezik nullánál nagyobb értékkel. Ezen felül az az egy nem nullaértékű szomszéd összesen kettő objektumponttal érintkezhet csak.
3. **Definíció:** N_0 *lokális maximumnak* számít, ha mind a nyolc szomszédjánál nagyobb értékkel rendelkezik. (Ezek a pontok vázpontoknak minősülnek)
4. **Definíció:** Az objektum két pontja, P és Q *4-összeköttetésben* vannak akkor és csak akkor, ha van köztük egy 4-út. (a 4-út csak 4-szomszedságot tartalmaz és minden pont nemnullaértékű)
5. **Definíció:** Egy 4-út *erősségét* $S(A)$, $A=(N_0, N_1, N_2...)$ az út legkisebb intenzitású pontja határozza meg.
6. **Definíció:** Két pont *erőssége* az őket összekötő utak legerősebbike határozza meg.

Fontos megjegyezni, hogy N_0 környezetében az N_1, N_3, N_5, N_7 párijainak csupán két lehetséges útvonala lehet, ha nem érintjük az N_0 -t.

1. rész: Előtranszformáció

Ebben a feldolgozási fázisban megpróbáljuk elkülöníteni a háttér az objektumtól, amiben fontos szerepet játszik az alakzat elhelyezkedése. Az úgynevezett *CGDT* (contextual gray distance transform) mátrix definiálásához először meg kell határoznunk a g_1 és a g_2 mátrixokat, melyeknek a minimumára van szükségünk.

$$CGDT(x, y) = \text{minimum}(g_1(x, y), g_2(x, y))$$

Előre haladó raszteres eljárással, a $2 \leq x \leq M - 1, 2 \leq y \leq N - 1$ tartományban:

$$g_1(x, y) = \text{minimum}[(g_1(x - 1, y - 1), g_1(x - 1, y), g_1(x - 1, y + 1), g_1(x, y - 1))] * (ave/max)^2 + Kép(x, y)$$

Hátrafele haladó raszteres eljárással, a $2 \leq x \leq M - 1, 2 \leq y \leq N - 1$ tartományban:

$$g_2(x, y) = \text{minimum}[(g_2(x + 1, y + 1), g_2(x + 1, y), g_2(x + 1, y - 1), g_2(x, y + 1))] * (ave/max)^2 + Kép(x, y)$$

A segédmatrrixok értéke:

$$ave = [Kép(x, y - 1) + Kép(x, y + 1) + Kép(x - 1, y) + Kép(x + 1, y)] / 4$$

$$max = \{a \text{ kép hisztogramjának a legmagasabb pontja, tehát a kép leggyakoribb értéke}\}$$

A korábban látott `for` ciklusokkal könnyen be lehet járni a képet. A maximum értékre nem érzékeny az algoritmus. Miután megkaptuk a *CGDT* mátrixot, azon elkezdhetjük a vékonyító algoritmusunkat.

2. rész: Vékonyítás

Amíg van törölhető pont a képen, addig hajtjuk végre újra a vékonyító algoritmusunkat egy előre haladó raszteres eljárással. A határpontokat azonban itt is meg kell jelölnünk előre. Az algoritmus a törléssel a képpont intenzitását nullára állítja.

A határpontok megjelölése után N_0 törölhető, ha:

1. N_0 határpont
2. N_0 nem lokális maximumpont
3. N_0 nem végpont
4. N_0 törlése nem változtatja meg a szomszédjai összeköttetését
5. N_0 törlése nem gyengíti meg a szomszédjai 4-összeköttetését. A feltétel kielégítése megköveteli, hogy az N_0 ne legyen az N_1, N_3, N_5, N_7 párok egyikének se a legerősebb 4-útján

A szerzők bizonyították a cikkben, hogy egy N_0 pont esetén a fenti kódban (4)-gyel és (5)-tel jelölt feltételek (összefüggőség és összefüggőség erejének megőrzése) teljesülését az alábbi logikai függvénnyel vizsgálhatjuk, amely pontosan akkor ad igaz értéket, ha (4) és (5) teljesül:

$$C(1) \text{ and } C(2) \text{ and } C(3) \text{ and } C(4) \text{ and } L(1) \text{ and } L(2) = \text{true},$$

ahol

$$C(i) = (\min(V(N_{2i-1}), V(N_{(2i+1) \bmod 8})) = 0) \text{ or } (\min(N_{2i-1}, N_{(2i+1) \bmod 8}) > 0 \text{ and } V(N_{2i}) \geq V(N_0))$$

($i \in \{1,2,3,4\}$), és

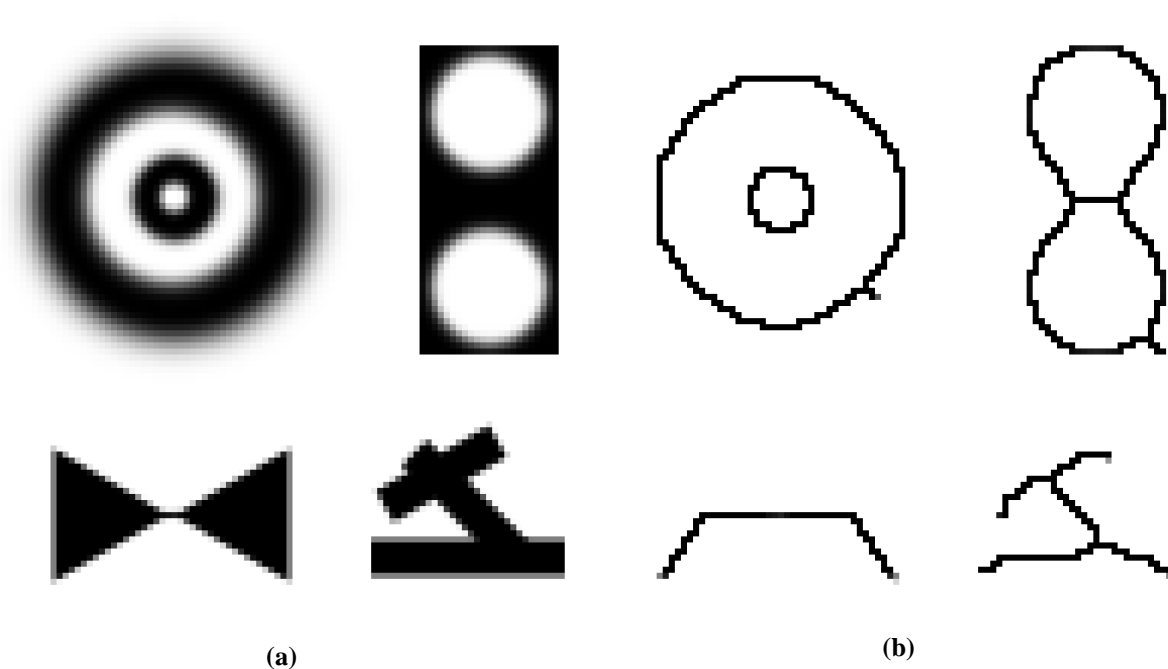
$$L(k) = \text{not } (\min(V(N_{2k-1}), V(N_{(2k+3) \bmod 8})) > 0 \text{ and } V(N_{2k+1}) < V(N_0) \text{ and } V(N_{(2k+5) \bmod 8}) < V(N_0))$$

($k \in \{1,2\}$).

Az algoritmus pszeudokódja:

```
G = {kimeneti mátrix, kezdetben a CGDT}
for minden P=(2 : x - 1, 2 : y - 1) képpontra:
    ha F(x, y) határpont, de nem csúcspont és nem végpont
    és F(x, y) törlése semelyik két 4-szomszédjának, 4-
    összefüggőségét sem befolyásolja
    és F(x, y) törlése semelyik két 4-szomszédjának, 4-
    összefüggőségét sem gyengíti, akkor G képpont értékét a P
    értékére állítjuk
    egyébként G pont értékét nullára állítjuk
```

Eredmények:



3.2. ábra: (a) Eredeti kép (b) Salary-Siy algoritmus eredménye

3.3. Szürkeárnyaltos karakterek vékonyítása

Az előző eljárásoktól eltérően a most következő algoritmus kifejezetten szürkeárnyaltos karakterek felismerésére jött létre 1998-ban [3]. A korábbi cikkek szerint a legjobb megoldás a kínai Hanzi, a koreai Hangul és a japán Kana kézírások felismerésére az alakfelismerő analízis, amely meghatározza milyen módon kell csoportosítani az egyes pontok háromszor hármas környezetében elhelyezkedő elemeket. Ilyenek például a hegygerincek, hegycsúcsok, hágók, stb. Vitatják a korábbi eredményeket, a váz pixelvastagsága miatt és a probléma megoldására hozták létre saját megoldásukat.

A cikk egy pixel fontossági jelölést alkalmaz amivel meg tudja állapítani mely pixelek lesznek nélkülözhetetlenek a váz meghatározásához és a topológia megőrzéséhez.

A vékonyítás első lépéseként meg kell határoznunk az egyes képpontokhoz tartozó *PSI* (*Pixel Superiority Index*) értéket, amely az alárendelt pontok számát jelöli az adott pont háromszor hármas környezetében. Az alárendelt pont a középpontnál kisebb, vagy egyenlő intenzitással rendelkező pontokat jelenti. Ezek alapján a *PSI* értéke nulla és nyolc között mozoghat. A nyolc alárendelt értékkel rendelkező képpontok csúcspontoknak számítanak. Ezek alapján két fontos következtetést vonhatunk le.

1. Ha egy pixel szomszédságában található egy alárendelt vázpont, akkor a vizsgált pont is vázpontnak minősül.

2. Minden vázpontnak a *PSI* értéke meghaladja a 4-et. (Kivételnek számítanak a szakadékhágó pontok, melyeket a szerzők köztudottan lényegtelennek nyilvánítanak)

A vázkijelölés ebből kifolyólag egy kezdetleges váz kiterjedéséből fog állni. A kezdetleges váz a nyolc *PSI* értékkel rendelkező képpontokból fog kiindulni és egészen a négy *PSI* értékig fogjuk bővíteni a vázunkat. Az eljárás során a hamis ágakat el kell távolítani a bemeneti képünkről és így az egy pixel vastagságú vázunk méretét meg tudjuk tartani. Ez a módszer, vékonyítás helyett kiterjedéssel határozza meg a vázat.

A gyakorlatban azonban ez az eljárás hibákkal jár, és a hamis ágak detektálása is időigényes lehet. Ennek elkerülése érdekében egy meghatározott *PSI* értékig fogjuk megadni a kezdetleges vázunkat, majd a kiterjedést csak kisebb értékeken fogjuk végrehajtani. Az írók tapasztalatai alapján a hatnál kisebb, vagy egyenlő *PSI* értékkel szinte mindegyik hamis ágat el lehet tüntetni. Ezek alapján a vázkijelölő algoritmus két részből fog állni:

1. Kezdetleges vázkijelölés

A hatnál nagyobb-egyenlő *PSI* értékkel rendelkező pontokat vesszük csak figyelembe. Ezeknek a pontoknak az összességét jelöljük *S* betűvel. Topológiai fontosság alapján kezdjük el vékonyítani az *S* objektumot úgy, hogy a törlés ne sértse meg a 8-összeköttetést.

Pszeudokódja:

```
PSI = {F méretű mátrix}
for minden P=(x, y) képpontra:
    ha F valamelyik szomszédjának intenzitása kisebb, akkor
        PSI képpont értékét növeljük ugyan azon a helyen, ahol a
        vizsgálatot végeztük
for minden P=(x, y) képpontra:
    ha PSI értéke  $\leq 6$  akkor F értékét a vizsgált helyen
        nullára állítjuk
```

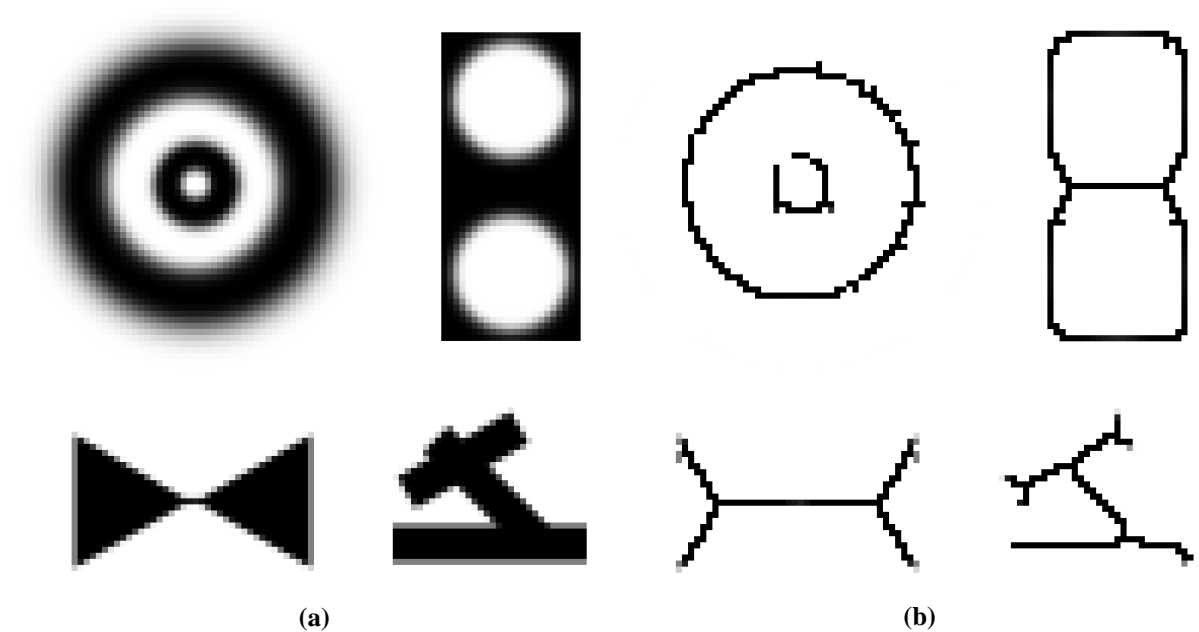
Ezek után a *PSI* érték alapján vékonyítunk a hatos értéktől egészen a nyolc *PSI* értékig. Vékonyító feltételnek egy korábbi cikk feltételeit adták meg, melyek hasonlóak voltak a Salary-Siy-féle algoritmus feltételeihez, ezért azt használtam.

2. Váz kibővítése

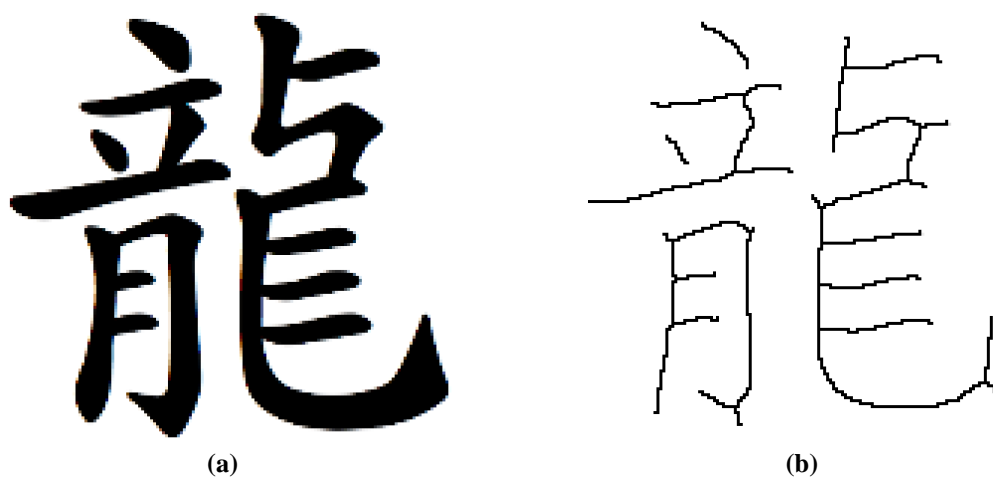
A következő szakaszban a kezdetleges váz leszakadó részeit kell összekötnünk, mivel nincs garantálva, hogy az összeköttetések megmaradnak. Ehhez az 5 *PSI* értékkel rendelkező pontokat fogjuk használni. Minden egyes objektumpontot meg kell vizsgálnunk és egy bejáró

algoritmussal megnéznünk, hogy össze tudjuk-e kötni a különálló objektumokat az 5 *PSI* értékű pontok használatával. Mivel a bejáró algoritmus pszeudokódjával nem szolgált a cikk és a saját megvalósításom időigényesnek bizonyult, ezért a két pixel távolságban lévő objektumpontokat, amelyek között 5 *PSI* értékű képpontok össze tudnak kötni, azokat visszaállítottam az eredeti képről.

Eredmények:



3.3. ábra: (a) Eredeti kép (b) A karakteres vékonyítás eredménye



3.4. ábra: (a) Eredeti kép (b) A karakteres vékonyítás eredménye

3.4. Morfológia alapú vékonyítás

A szürkeárnyaltos képek vázkijelöléséhez nem csak a korábbiakban említett módszereket lehet alkalmazni [4]. 2006-ban létrehoztak egy újfajta megközelítést, morfológiai műveleteket alkalmazva a váz meghatározására. Ezt morfológiai műveletek (erózió, dilatació) egymás utáni alkalmazásával tudjuk megtenni. Ez a módszer már nem érzékeny a zajos háttérre és nem szükséges előtte előfeldolgozó eljárást sem alkalmazni. Az algoritmus a hegygerinceket keresi a bemeneti képen. Minden másik pontban végbemegy a morfológiai eljárás, mely a fennsíkokból hegygerinceket alakít, így megkeresve az objektum vázát.

Ehhez szükségünk van egy hegygerinc detektorra, melyet G jelöl. Ekkor:

$h = \{\text{küszöbérték}\}$

$O_1(x, y) = \{F(x, y) \text{ morfológiai nyitása } 3 \times 3\text{-as szerkesztőelemmel}\}$

$O_2(x, y) = \{F(x, y) \text{ morfológiai nyitása } 5 \times 5\text{-ös szerkesztőelemmel}\}$

$G(x, y) = \{ F(x, y), \text{ ha } F(x, y) - O_1(x, y) > 0 \text{ és } F(x, y) - O_2(x, y) > h, 0 \text{ különben} \}$

Valamint egy összefüggőségi számra, mely azt határozza meg, hogy az $f_1, f_2, f_3, \dots, f_9$ sorozatban hányszor történik váltás -1-ről 1-re. f_l értékét az alábbi módon határozzuk meg:

$(l=1,2,3,4)$

$f_{2l-1} = \{-1, \text{ ha } x_{2l-1} < x_0 \text{ and } (x_{2l-1} < x_{2l} \text{ or } x_{2l-1} < x_{2l+1})$

$1, \text{ ha } x_{2l-1} > x_{2l+1} \text{ and } x_{2l+1} < x_0$

$0 \text{ különben.}\}$

$f_{2l} = \{ 1, \text{ ha } x_{2l} > x_{2l+1} \text{ and } x_{2l+1} < x_0$

$0 \text{ különben.}\}$

Továbbá $f_9 = f_1$. Az (x, y) koordinátájú pixel összefüggőségi számát $C_8(x, y)$ -nal jelöljük.

A vékonyító algoritmus két részre osztható. Magára a vékonyításra és az összefüggőség visszaállítására, viszont a korábbi algoritmusokkal ellentétben ezúttal minden egyes lefutáskor el kell végeznünk mindkét részét, amíg találunk változást a program futtatás előtti és utáni képen.

Minden végrehajtás előtt kiszámoljuk a bemeneti kép morfológiai erózióját háromszor hármasszerkesztőelemmel (E), valamint a hegygerinc detektor eredményét, szintén a bemeneti képen elmentjük (R). Ezeknek a maximumát a $COMP^*$ fogja tárolni, ami a morfológiai transzformáció eredményét tartalmazza.

Pszeudokódja:

amíg van változás:

$E(x, y) = F(x, y)$ morfológiai eróziója 3x3-as szerkesztőelemmel
 $R(x, y) =$ Hegygerinc detektor eredménye $F(x, y)$ -ra
 $COMP^*(x, y) = \max(E(x, y), R(x, y))$

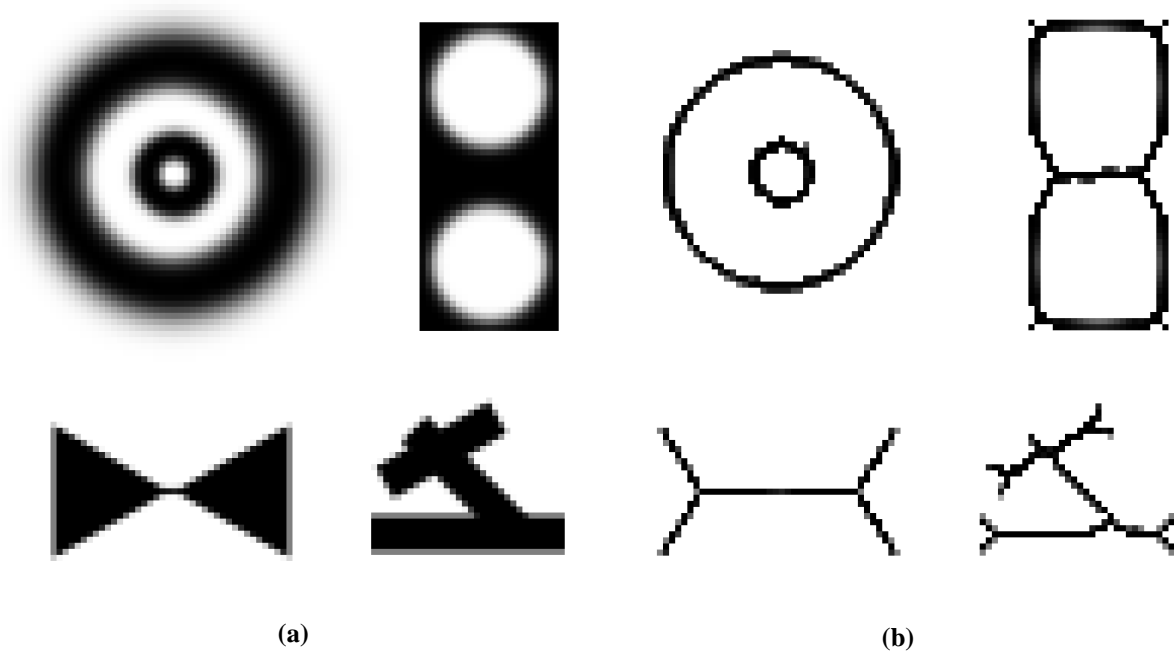
Ezek után még a cikluson belül kiszámoljuk a C_8 mátrixot a bemeneti kép alapján és általa visszaállítjuk az összefüggőséget ott, ahol szétszakadt az objektum.

Pszeudokódja:

$C_8(x, y)$ mátrix kiszámítása a $F(x, y)$ alapján
 for minden $P=(x, 2 : y)$ képpontra:
 ha $R(x, y) = 0$ és $C_8(x, y) \geq 2$,
 akkor $COMP^*(x, y) = F(x, y)$

Végül az F képet egyenlővé tesszük a $COMP^*$ mátrixszal.

Eredmények:



3.5. ábra: (a) Eredeti kép (b) Morfológia alapú vékonyítás eredménye

3.5. Couprie-Bezerra-Bertrand-féle algoritmus

Egy teljesen párhuzamos vékonyító algoritmust hoztak létre 2013-ban, mely visszanyúl a bináris vékonyítás alapjaihoz és az ott alkalmazott eljárásokat bővíti ki [5]. A feldolgozott

cikk kimondja, hogy a párhuzamos algoritmusok minősége meghaladja a szekvenciális elődjeiket, szürkeárnyaltos képekre azonban eddig nem volt egy ilyen általános eljárás sem. A szerzők által leírt algoritmus a bináris képek vékonyításánál leírt 2. tétel elvét követik és terjesztik ki, szürkeárnyaltos képekre.

Mint mindegyik algoritmusnál, itt is először a határpontokat jelöljük ki, tehát a háttérpontokkal érintkező objektumpontokat. Itt azonban a határpontok mellé egy segédmátrixban elmentjük a határpont szomszédjai közül a nála kisebb intenzitással rendelkezők közül a legnagyobb értékűt (L). Erre azért van szükség, mivel a már bemutatott törlések közül az egyik legbiztonságosabbnak mondható esetén erre az értékre kell állítani a törlendő pont intenzitását. Ezek után a törlésre való megjelöléskor minden egyes ponthoz létrehozunk egy bináris ötször-ötös területet, a meglévő intenzitások alapján, amelyen már könnyedén el lehet végezni a 2. tétel vizsgálatait. Az N_0 -nál kisebb intenzitással rendelkező pontokat nullával, a nagyobbakat és az egyenlőket eggyel jelöljük. A törlésre megjelölt pontokat a korábbiakhoz hasonlóan végezzük és az így kapott eljárás egy gyorsan és hatékonyan működő, teljesen párhuzamos algoritmust eredményez.

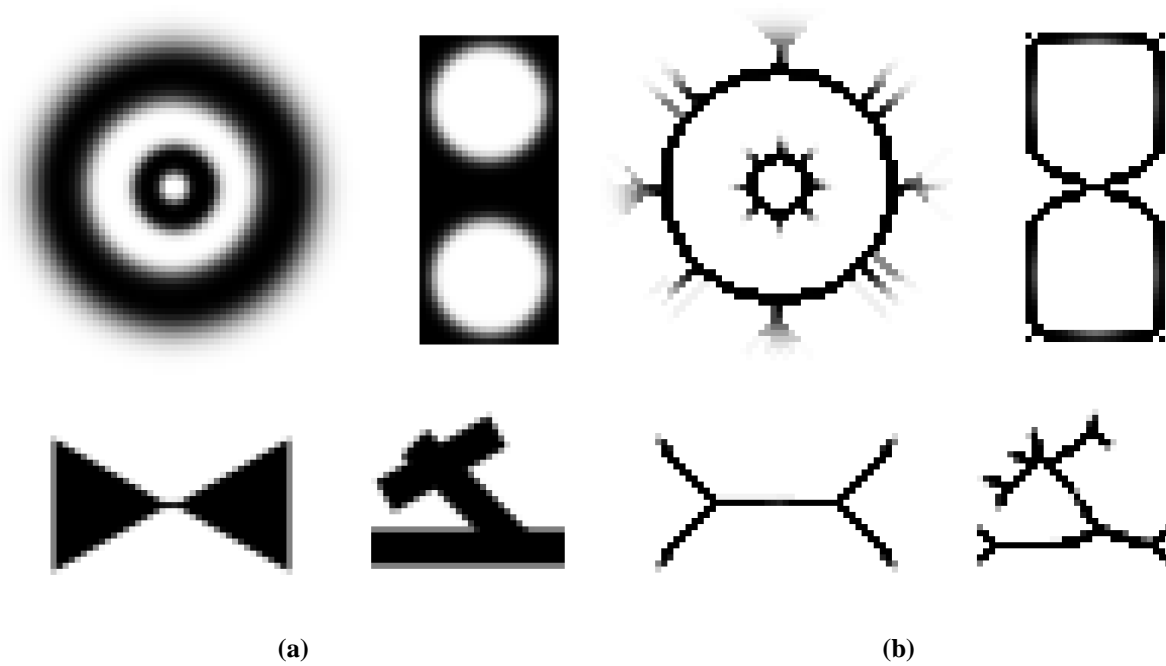
Mivel a bináris állapotokat egy ötször-ötös területen le tudjuk generálni előre, ezért akár fel is tudjuk gyorsítani a vékonyítási folyamatot, egy olvasótábla segítségével.

A leggenerált állapotokon előre elvégezzük a feltételeket és az eredményt (hogyan törölhető-e az N_0 pont a képen) eltároljuk a keresőtáblában. A keresőtábla indexeit a pixel pozícióból nyerjük ki. Minden egyes pozíciót a mátrixunkban lementünk bináris számsorozat elemeiként. A számsorozat egyes értékei megegyeznek az ötször-ötös mátrix értékeivel. Miután megkaptuk a sorozatunkat, tízes számrendszerbe átváltjuk a számot és ez a szám fogja jelölni a pozíció törölhetőségének indexét. Az index számát 4 byte-on mentjük, a törölhetőséget pedig 1 byte-on. Ezek után a beolvasott olvasótáblából minden egyes törölhető pozíció kiolvasható, gyors sorozattá alakítással. Ez a megoldás nagyobb terjedelmű képeknél fogja felgyorsítani a törlési folyamatot.

Pszeudokódja:

```
for minden  $P=(2 : x - 2, 2 : y - 2)$  képpontra:
    ha határpont, akkor a képpont ötször-ötös környezetének
        átalakítása bináris sorozattá, majd tízes számrendszerbeli
        számmá és elmenteni az Index változóba
        ha a keresőtábla Index helyén lévő értéke 1, akkor
            törölhető a pont
```

Eredmények:



3.6. ábra: (a) Eredeti kép (b) Couprie-Bezerra-Bertrand algoritmus eredménye

4. Implementáció

4.1. Képek beolvasása

Ahhoz, hogy a tesztképeket be tudjam olvasni és el tudjam rajtuk végezni az algoritmusokat, a cv2 könyvtárat használtam, amiben egy saját funkció megírásával gyorsan el tudtam végezni a beolvasást.

```
def imreadgray(img):  
    image = cv2.imread(img)  
    img = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)  
    return img
```

Ezután minden egyes képpontot át kellett alakítanom, mivel a cikkekben a fekete pontok rendelkeztek 255-ös értékkel és a fehérek 0-val.

4.2. Képek bejárása és mátrixok

A képek bejárását kettős for ciklussal tudtam megoldani, melyek az oszlopokon és a sorokon haladtak végig egyesével minden pixelen. A beoldasott kép méretét könnyen meg tudtam határozni a kép objektum .shape értékének lekérdezésével. A segédmátrixokat arra használtam, hogy a kép pixeleiről tudjak elmenteni fontos információkat.

```
size = img.shape  
nemegyenlo = True  
for row in range(0, size[0]):  
    for col in range(0, size[1]):  
        img2[row][col] = 0
```

4.3. Keresőtábla

A keresőtábla megvalósításához a file funkciókat alkalmaztam, amelyek lehetővé teszik a bináris adattárolást. A struct könyvtárral tudtam megmondani mennyi bájtot szeretnék adni az index és az értéknek a file-on belül.

```
file = open('lookup_index', 'wb')  
file.write(struct.pack('<l', index))  
file.close()
```

4.4. Képek elmentése

A képek elmentéséhez szintén át kellett állítanom a fekete és a fehér színek értékeit. Ezután a következő paranccsal tudtam a megadott könyvtárba elmenteni a módosított képet:

```
cv2.imwrite('folder/image.png', img)
```

5. Eredmények

A tesztképeken kapott eredményeken megfigyelhető, melyek azok az algoritmusok, amelyek megőrzik a képen látható alakzat topológiáját. A 3. fejezetben bemutatott tesztképekről és az 5. fejezetben bemutatott összehasonlításokról elmondható, hogy bár specifikus esetekben jól működnek a Dyer-Rosenfeld, Salary-Siy és a karakteres vékonyítás algoritmusokról, nem lehet elmondani, hogy topológia-megőrzők lennének a szürkeárnyaltos képeken.

A Dyer-Rosenfeld algoritmus szétszakítja az objektumokat nem törődve azzal, hogy egy, vagy két oldalról egyszerre végezzük a törlést.

E. Salary és P. Siy algoritmusai kiváló eredményeket produkál mindaddig, amíg az objektum lényegesen megkülönböztethető a háttértől. Ellenkező esetben nem nyeri ki a vázat.

A karakter vékonyítás módszerrel az objektumoknak minden belső pontjának az intenzitása meg kell hogy haladjon, a külső határpontok intenzitását, különben szakadások keletkeznek az objektumban. Az utóbbi két algoritmus használata a legjobban karakterek vékonyítására alkalmazható.

A morfológia alapú és a Couprie-Bezerra-Bertrand algoritmus viszont szemmel láthatóan jobb eredményeket produkálnak. Mindkét vékonyítás egy pixelszélességnél vastagabb vázat hoz létre, viszont a bemeneti kép bármilyen háttérrel rendelkezhet, így akár gyengébb minőségű szürkeárnyaltos képeken is alkalmazhatóak.

Kromoszóma:



5.1. ábra: Eredeti kép



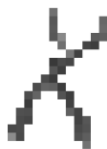
5.2. ábra: Dyer-Rosenfeld



5.3. ábra: Salary-Siy



5.4. ábra: Karakter vékonyítás



5.5. ábra: Morfológia alapú

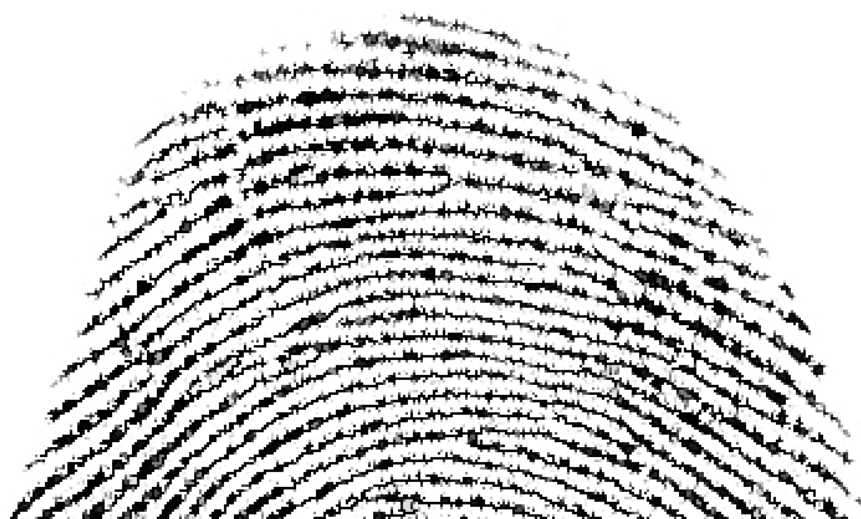


5.6. ábra: Couprie-Bezerra-Bertrand

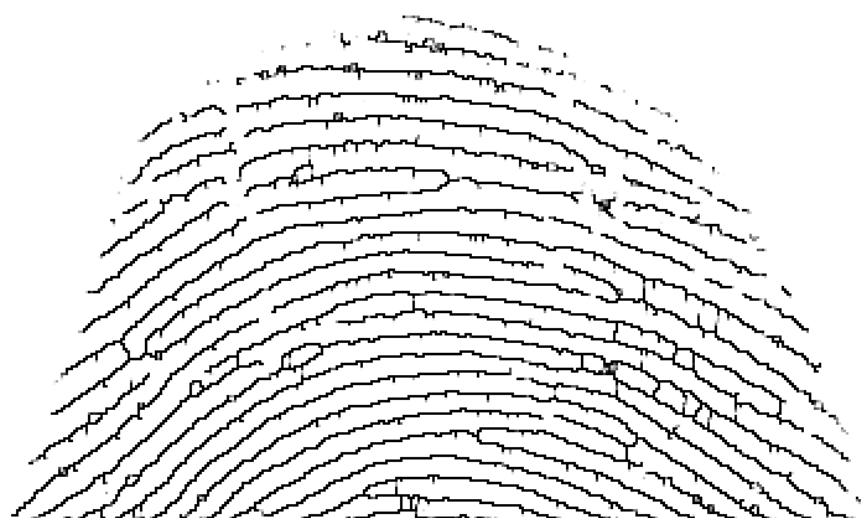
Ujjlenyomat:



5.7. ábra: Eredeti kép



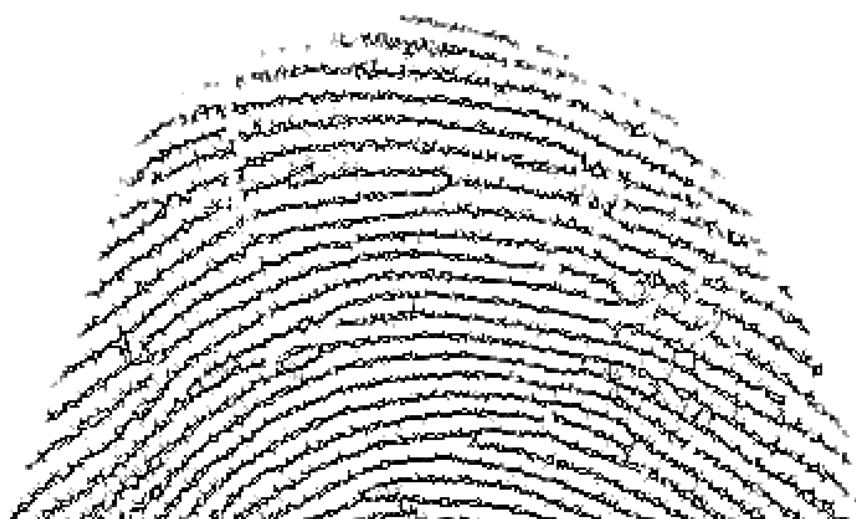
5.8. ábra: Dyer-Rosenfeld



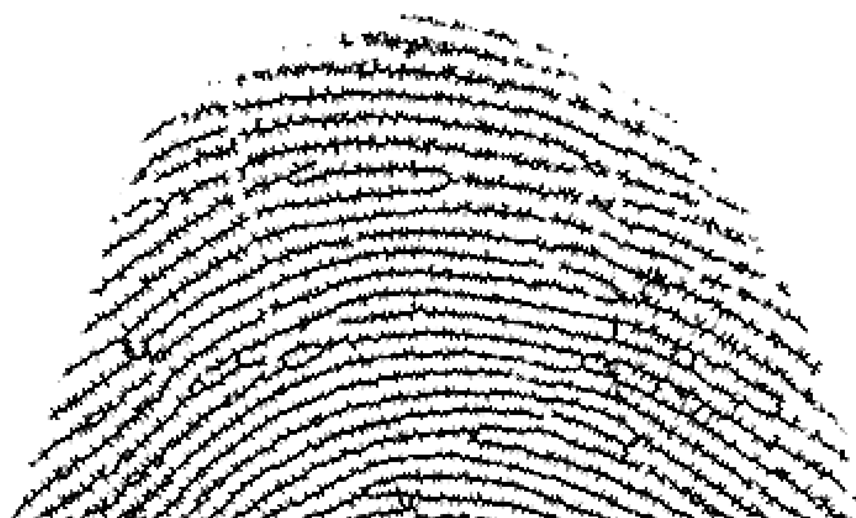
5.9. ábra: Salary-Siy



5.10. ábra: Karakter vékonyítás



5.11. ábra: Morfológia alapú



5.12. ábra: Couprie-Bezerra-Bertrand

Karakterek:

Thinning Algorithms *Thinning Algorithms*

5.13. ábra: Eredeti kép

5.14. ábra: Dyer-Rosenfeld

Thinning Algorithms *Thinning Algorithms*

5.15. ábra: Salary-Siy

5.16. ábra: Karakter vékonyítás

Thinning Algorithms *Thinning Algorithms*

5.17. ábra: Morfológia alapú

5.18. ábra: Couprie-Bezerra-Bertrand

Irodalomjegyzék

1. C. R. Dyer, A. Rosenfeld: Thinning Algorithms for Grayscale Pictures (1977)
2. E. Salari, P. Siy: The Ridge-Seeking Method for Obtaining the Skeleton of Digital Images (1984)
3. K. W. Kang, J. W. Suh and J. H. Kim: Skeletonization of Grayscale Character Images Using Pixel Superiority Index (1998)
4. K. M. Kim, B. Lee, N. S. Choi, G. H. Kang, J. J. Park, and C. Y. Suen: Gray-Scale Thinning Algorithm Using Local Min/Max Operations (2006)
5. M. Couprie, N. Bezerra, G. Bertrand: A Parallel Thinning Algorithm for Grayscale Images (2013)
6. Palágyi Kálmán: Digitális topológia és matematikai morfológia előadás jegyzet. Szegedi Tudományegyetem Természettudományi és Informatikai Kar (2009)
7. C. Arcelli and G. Ramella: Finding grey-skeletons by iterated pixel removal (1995)
8. Németh Gábor: Topológia-megőrző vékonyító algoritmusok tervezése és vázközelítések kvantitatív összehasonlítása (2012)
9. Palágyi Kálmán: Képfeldolgozás haladóknak (egyetemi jegyzet) (2011)

Nyilatkozat

Alulírott Ility Gergő programtervező informatikus BSc szakos hallgató, kijelentem, hogy a dolgozatomat a Szegedi Tudományegyetem, Informatikai Intézet Képfeldolgozás és Számítógépes Grafika Tanszékén készítettem, programtervező informatikus BSc diploma megszerzése érdekében.

Kijelentem, hogy a dolgozatot más szakon korábban nem védtem meg, saját munkám eredménye, és csak a hivatkozott forrásokat (szakirodalom, eszközök, stb.) használtam fel.

Tudomásul veszem, hogy szakdolgozatomat a Szegedi Tudományegyetem Informatikai Intézet könyvtárában, a helyben olvasható könyvek között helyezik el.

Szeged, 2019. május. 18.

Aláírás

Köszönetnyilvánítás

Ezúton szeretnék köszönetet mondani Kardos Péter tanár úrnak, akinek türelme és segítsége nélkül nem tudtam volna elkészíteni a szakdolgozatomat.