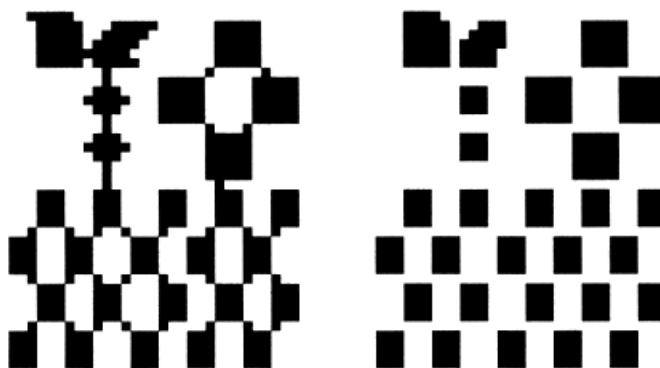


Otvaranje i zatvaranje

Operacija u kojoj eroziju neposredno slijedi dilatacija istim strukturnim elementom naziva se *otvaranje*:

$$O(\mathbf{A}, \mathbf{B}) = D(E(\mathbf{A}, \mathbf{B}), \mathbf{B}).$$

Naziv operacije je opisnog karaktera, jer ova operacija nastoji da proširi male otvore ili ukloni slabo povezane piksele. Slika 156 prikazuje efekat otvaranja jednostavnim strukturnim elementom na kolekciji sitnih objekata u cilju kidanja veza između njih. Otvaranje se može koristiti i za ukljanjanje šuma, Slika 157. U sklopu operacije otvaranja erozija će ukloniti slabo povezane i rubne piksele objekata, dok će dilatacija izvršiti restauraciju rubnih piksela objekata bez restauracije šuma.



Slika 156. [4] Efekat otvaranja kolekcije sitnih objekata

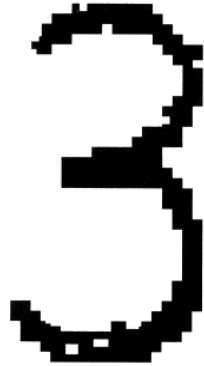


Slika 157. [4] Uklanjanje šuma u obliku crnih izolovanih piksela otvaranjem

Operacija *zatvaranja* je slična operaciji otvaranja, s tim da se dilatacija izvodi prva, a zatim slijedi erozija istim strukturnim elementom:

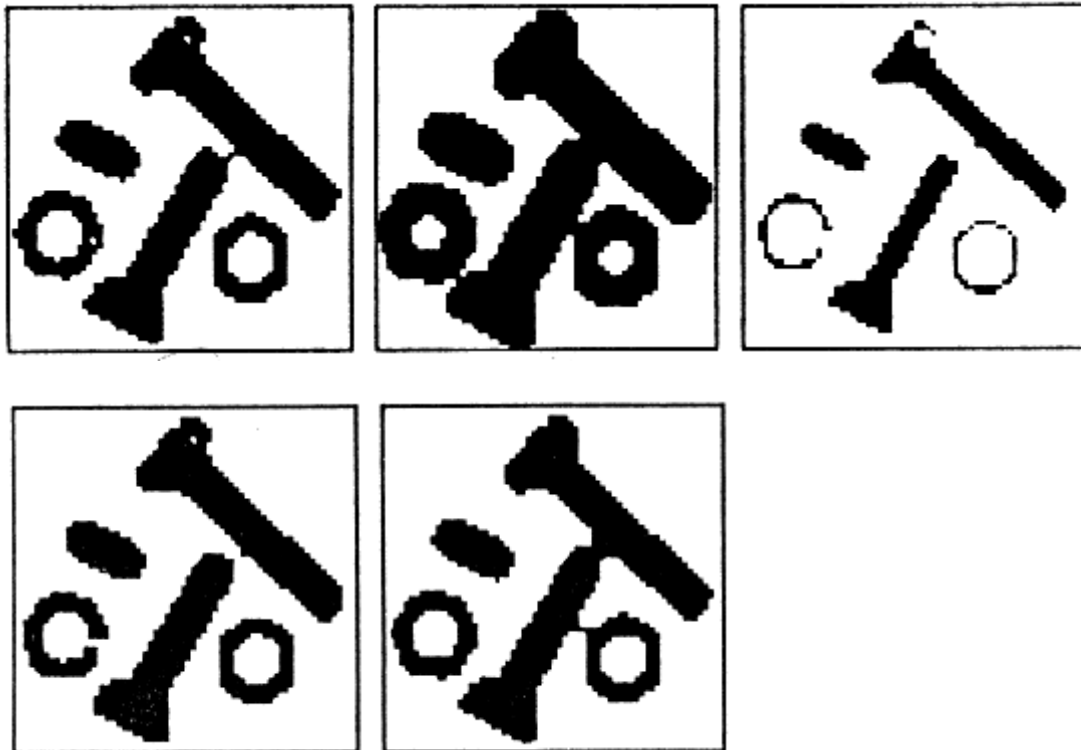
$$C(\mathbf{A}, \mathbf{B}) = E(D(\mathbf{A}, \mathbf{B}), \mathbf{B}).$$

Zatvaranje popunjava male procjepe i izolovane bjeline. Slika 158 prikazuje operaciju zatvaranja na istoj slici na kojoj je već izvedeno otvaranje u cilju eliminisanja šuma u vidu izolovanih crnih piksela. Zatvaranje je ukinulo i izolovane bijele piksele.



Slika 158. Uklanjanje šuma u obliku bijelih izolovanih piksela zatvaranjem

Na Slici 159 je dato poređenje dilatacije, erozije, otvaranja i zatvaranja.



Slika 159. [14] Primjeri morfoloških operacija. Redom: originalna slika, dilatacija, erozija, otvaranje i zatvaranje prostim strukturnim elementom

Osobine otvaranja i zatvaranja

Zatvaranje objekta odgovara otvaranju pozadine istim strukturnim elementom i obrnuto:

$$\begin{aligned}C^c(\mathbf{A}, \mathbf{B}) &= O(\mathbf{A}^c, \mathbf{B}), \\O^c(\mathbf{A}, \mathbf{B}) &= C(\mathbf{A}^c, \mathbf{B}).\end{aligned}$$

Translacija:

$$\begin{aligned}O(\mathbf{A} + x, \mathbf{B}) &= O(\mathbf{A}, \mathbf{B}) + x, \\C(\mathbf{A} + x, \mathbf{B}) &= C(\mathbf{A}, \mathbf{B}) + x.\end{aligned}$$

Za otvaranje strukturnim elementom \mathbf{B} slika \mathbf{A} , \mathbf{A}_1 i \mathbf{A}_2 gdje je $\mathbf{A}_1 \subseteq \mathbf{A}_2$ vrijedi:

$$\begin{aligned}\text{Antiekstenzivnost: } &O(\mathbf{A}, \mathbf{B}) \subseteq \mathbf{A}, \\ \text{Monotoni porast: } &O(\mathbf{A}_1, \mathbf{B}) \subseteq O(\mathbf{A}_2, \mathbf{B}), \\ \text{Idempotentnost: } &O(O(\mathbf{A}, \mathbf{B}), \mathbf{B}) = O(\mathbf{A}, \mathbf{B}).\end{aligned}$$

Za zatvaranje strukturnim elementom \mathbf{B} slika \mathbf{A} , \mathbf{A}_1 i \mathbf{A}_2 gdje je $\mathbf{A}_1 \subseteq \mathbf{A}_2$ vrijedi:

$$\begin{aligned}\text{Ekstenzivnost: } &\mathbf{A} \subseteq C(\mathbf{A}, \mathbf{B}), \\ \text{Monotoni porast: } &C(\mathbf{A}_1, \mathbf{B}) \subseteq C(\mathbf{A}_2, \mathbf{B}), \\ \text{Idempotentnost: } &C(C(\mathbf{A}, \mathbf{B}), \mathbf{B}) = C(\mathbf{A}, \mathbf{B}).\end{aligned}$$

“Hit and Miss” operacija

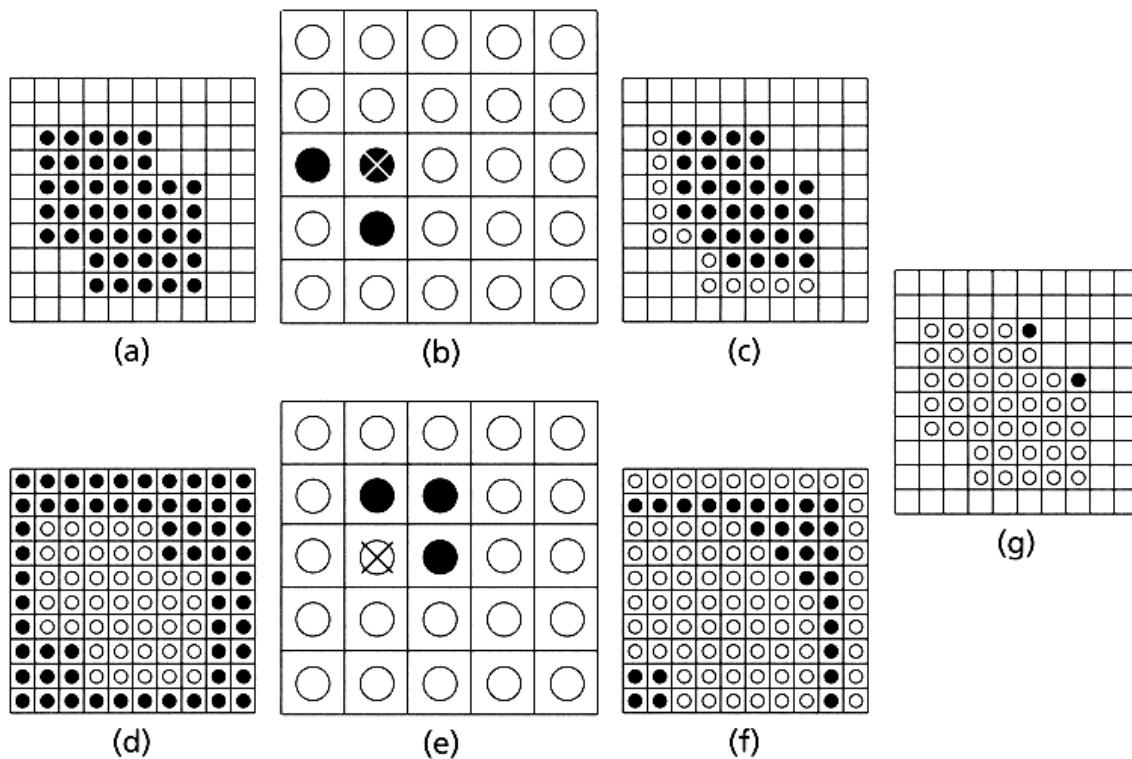
Hit and Miss je morfološki operator dizajniran da locira jednostavne oblike na slici. Zasnovan je na eroziji što je prirodno, jer erozija slike \mathbf{A} strukturnim elementom \mathbf{S} sadrži samo one piksele (lokacije) gdje se \mathbf{S} sadrži u \mathbf{A} , tačnije one lokacije gdje se crni pikseli strukturnog elementa poklapaju sa crnim pikselima slike. To znači da slika nastala erozijom saži i one lokacije gdje se pozadina strukturnog elementa ne poklapa sa pozadinom slike. Ako želimo odrediti lokacije gdje se strukturni element u potpunosti poklapa sa slikom potrebno je odrediti pozicije na koje treba translirati strukturni element tako da se i objekat i pozadina (i crni i bijeli pikseli) strukturnog elementa poklope sa slikom. Poklapanja objekta strukturnog elementa sa slikom nazivamo *hit* i odgovarajuće lokacije se jednostavno dobiju erozijom slike tim strukturnim elementom. Za određivanje gdje se pozadina strukturnog element poklapa sa pozadinom slike potrebno je uraditi eroziju pozadine slike \mathbf{A}^c pozadinom strukturnog elementa \mathbf{S}^c . Umjesto toga, efikasnije je formirati novi strukturni element \mathbf{T} koji sadrži piksele pozadine strukturnog elementa \mathbf{S} . Poklapanje pozadine strukturnog elementa \mathbf{S} sa pozadinom slike \mathbf{A} označeno je kao *miss* i određuje se erozijom pozadine slike strukturnim elementom \mathbf{T} .

Dakle, operacije koja daje lokacije gdje se strukturni element u potpunosti poklapa sa slikom je:

$$\text{HitMiss}(\mathbf{A}, \mathbf{S}, \mathbf{T}) = E(\mathbf{A}, \mathbf{S}) \cap E(\mathbf{A}^c, \mathbf{T}).$$

Hit and Miss operator je morfološki ekvivalent operaciji *template matching*, dobro poznatoj tehnici za prepoznavanje oblika koja je zasnovana na kroskorelaciji.

Na Slici 160 dat je primjer određivanja gornjeg desnog ugla. Strukturni element \mathbf{S} mora da sadrži piksel u uglu, te jedan lijevo i jedan ispod njega. Strukturni element za pozadinu \mathbf{T} nije jednostavni komplement od \mathbf{S} , već sadrži one piksele koji moraju pripadati pozadini da bi piksel označen sa x bio gornji desni ugao. Neophodna je veoma pažljiva selekcija ovih piksela.



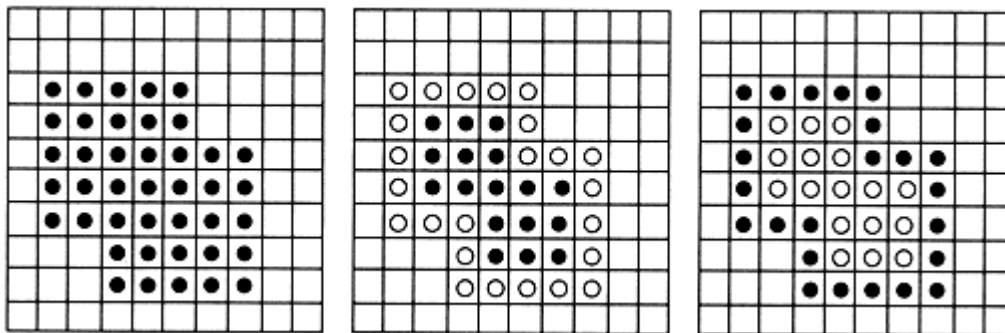
Slika 160. [4] Ilustracija “Hit and Miss” operacije. (a) Slika sa objektom \mathbf{A} . (b) Strukturni element \mathbf{S} . (c) Erozijska $E(\mathbf{A}, \mathbf{S})$. (d) Pozadina \mathbf{A}^c . (e) Strukturni element \mathbf{T} . (f) Erozijska $E(\mathbf{A}^c, \mathbf{T})$. (g) “I” operacija između (e) i (f).

Identifikacija granica regiona

Pikseli na rubovima objekta su oni koji imaju najmanje jedan susjedni piksel koji pripada pozadini. Kako je moguće da to bude bilo koji susjedni element, nije moguće unapred odrediti koji od njih da se traži, tj., nije moguće pronaći odgovarajući strukturni element za eroziju ili dilataciju čijom primjenom bi detektovali rubove. Erozija jednostavnim strukturnim elementom S uklanja baš rubne piksele, te se ta činjenica može iskoristiti za dizajniranje morfološkog operatora za detekciju rubova: erozijom se uklone rubovi pa se tako dobivena slika oduzme od originala. Na taj način zadržavamo samo rubne piksele. Formalno to možemo zapisati sa:

$$\text{Rub} = A - E(A, S).$$

Slike 161 i 162 prikazuju primjere određivanja rubova.



Slika 161. [4] Određivanje rubova jednostavnog objekta



Slika 162. Određivanje rubova složenog objekta

Uslovna dilatacija

Ponekad je potrebno izvršiti dilataciju objekta tako da određeni pikseli ostanu imuni, tj., da se ne dozvoli širenje objekta preko tih piksela. Područje slike izvan kog nije dozvoljena dilatacija se specificira kao posebna *mask* slika A' , a dilatacija počinje od objekta A koji se naziva *seed* (sjeme) i radi se nekim od jednostavnih strukturnih elemenata S_e :

$$D(A, (S_e, A')).$$

Ovaj formalni mehanizam se, zavisno od primjene, naziva različitim imenima: *popunjavanje regiona, rekonstrukcija, propagacija...*

Uslovna dilatacija se često primjenjuje pri segmentaciji slike. Određivanje dobrog praga za segmentaciju gray-scale slike može biti veoma teško, kao što ćemo kasnije vidjeti. Pri previsokom pragu (visok nivo sivila) izdvajamo samo one piksele koji sigurno pripadaju objektu, ali ih je najčešće premalo. Pri niskom pragu će i neki pikseli pozadine biti shvaćeni kao pikseli koji pripadaju objektu. Koristeći uslovnu dilataciju moguće je od dvije loše segmentirane slike dobiti jednu dobro segmentiranu sliku, Slika 163. Slika se prvo segmentira visokim, a zatim niskim pragom. Zatim se vrši uslovna dilatacija prve slike ali ne preko granica druge slike:

$$D(I_{high}, (S_e, I_{low})).$$

Na kraju se primjeni otvaranje za uklanjanje slabo vezanih rubnih piksela.

Implementacija uslovne dilatacije je iterativna. Svaka dilatacija je praćena maskiranjem (presjekom) slikom koja sadrži piksele izvan kojih nije dozvoljena dilatacija:

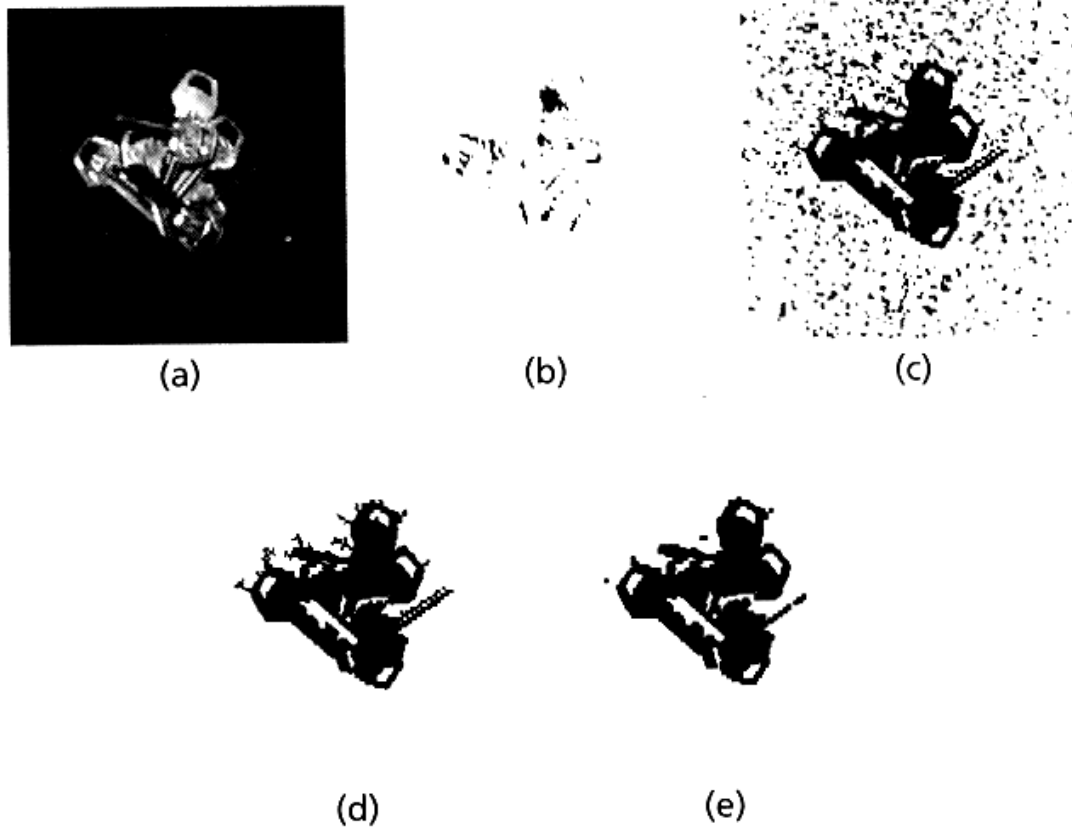
$$A_i = D(A_{i-1}, S_e) \cap A'.$$

Postupak se nastavlja dok ne postane $A_i = A_{i-1}$.

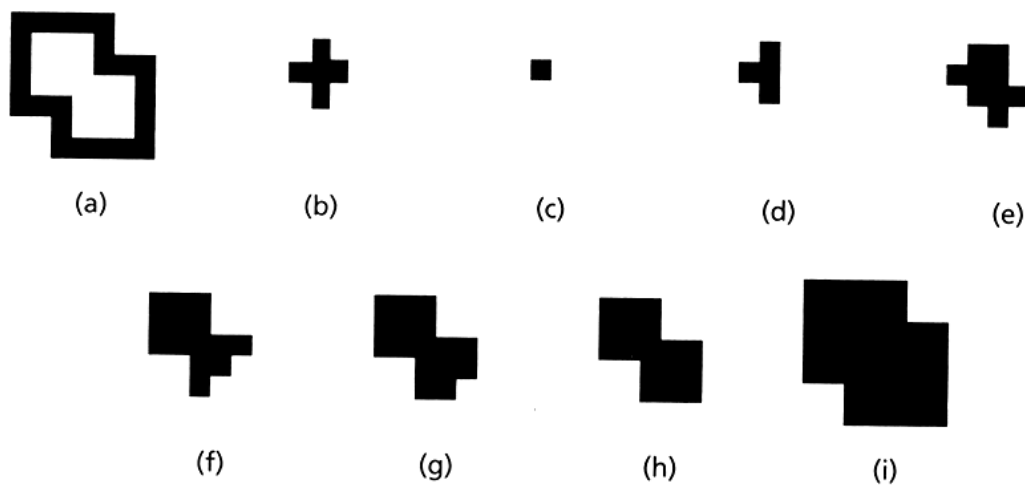
Druga aplikacija uslovne dilatacije je popunjavanje regiona, operacija suprotna izdvajanju rubova. Želimo da popunimo region zadat svojim rubom i jednim pikselom unutar njega, tako da svi pikseli unutar regiona budu crni, i da pri tome ni jedan piksel van granica regiona ne postavimo na crno. Kako se proces uslovne dilatacije zaustavlja kad se objekat proširi do bijelih piksela, za mask sliku se bira komplement slike koja sadrži rub regiona, A^C , a sjeme je bilo koji piksel P unutar regiona. Dilatacija se vrši nekim od jednostavnih strukturnih elemenata. Slika 164 prikazuje primjer popunjavanja regiona, pri čemu je dilatacija vršena strukturnim elementom S_+ u obliku znaka +:

$$Fill = D(P, (S_+, A^C)).$$

Na kraju je urađena unija popunjenog regiona i ruba.



Slika 163. [4] Poboljšanje segmentacije uslovnom dilatacijom. (a) Originalna slika. (b) I_{high} , (c) I_{low} . (d) Uslovna dilatacija. (e) Otvaranje.



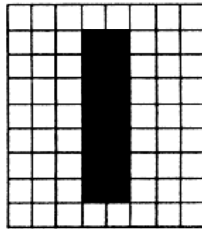
Slika 164. [4] Popunjavanje regiona uslovnom dilatacijom. (a) Objekat A. (b) Strukturni element S_+ . (c) Sjeme P. (d)-(h) Iteracije 1-5. (i) Unija popunjenog regiona i ruba.

Izdvajanje skeleta (esencijalne linije)

Neformalna definicija *skeleta* je linija koja predstavlja objekat sa osobinama:

- jedan piksel debela,
- “provučena” kroz sredinu objekta,
- sačuvala topologiju objekta.

Na osnovu ove definicije nije moguće uvijek odrediti skelet objekta. Na to jasno ukazuje Slika 165, gdje je nemoguće pronaći liniju jedan piksel debelu koja prolazi sredinom objekta. Ipak, postoje mnoge tehnike koje sa određenim uspjehom generišu skelet.



Slika 165. Primjer gdje je nemoguće izdvojiti skelet

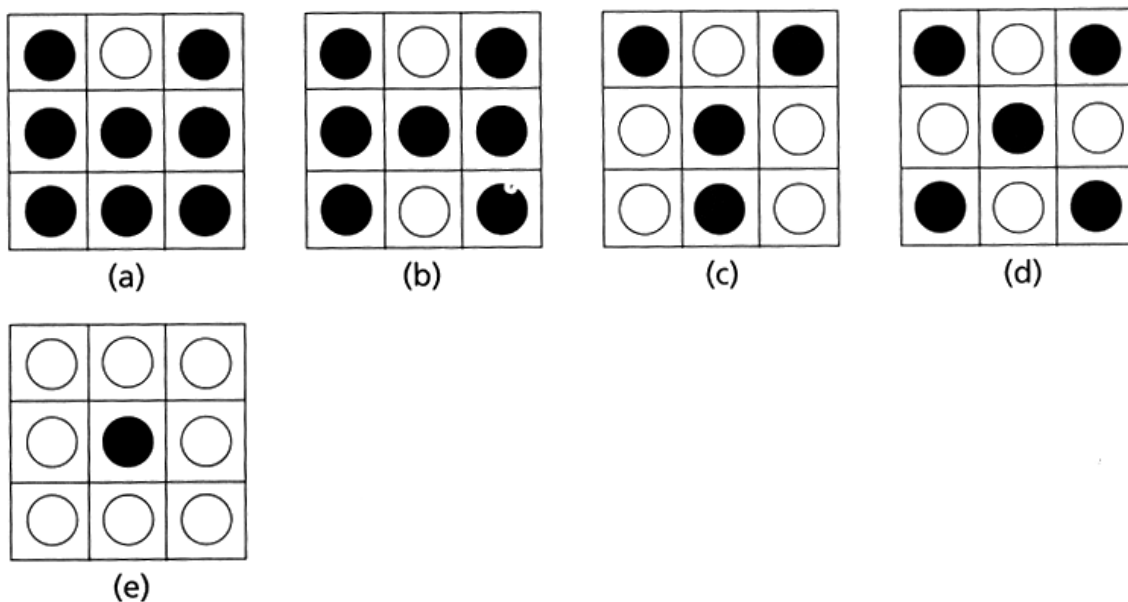
Iterativni metod izdvajanja skeleta

Prije prelaska na sam algoritam objasnimo pojam povezanosti. *Povezanost* je broj koji pokazuje koliko objekata pojedini piksel može povezati. Centralni pikseli na Slici 166 imaju povezanost: (a) 1, (b) 2, (c) 3, (d) 4, (e) 0. Pri određivanju skeleta, pikseli čija je povezanost različita od 1 ne smiju se uklanjati jer bi to promijenilo topologiju objekta. Takođe, ne smije se uklanjati ni *krajnji piksel*. Pod krajnjim pikselom podrazumijevamo onaj piksel koji je vezan samo sa jednim pikselom objekta. U slučaju uklanjanja krajnjih piksela, sve ravne linije i otvorene krive bile bi uklonjene.

Većina algoritama za određivanje skeleta je zasnovana na “ljuštenju” rubnih piksela. Postupak se ponavlja sve dok postoje pikseli koji se mogu ukloniti. Postoji skup pravila koji definiše koji pikseli se smiju uklanjati i često se za implementaciju ovih pravila koriste specijalno dizajnirane maske, Slika 167. Skup pravila se definiše tako da se postupak zaustavlja kad više nema promjena nakon dva uzastopna prolaza kroz sliku.

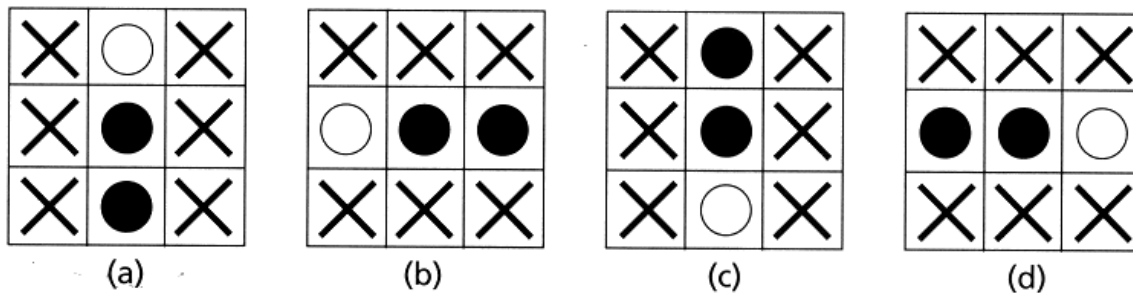
Osnovni algoritam je sljedeći:

1. Pronađi lokaciju piksela gdje se template M1 poklapa sa slikom.
2. Ako centralni piksel nije *krajnji piksel* i ako je njegova *povezanost* =1, označi taj piksel za kasnije brisanje.
3. Ponavljaj korake 1 i 2 za sve lokacije piksela gdje se template M1 poklapa sa slikom.
4. Ponavljaj korake 1-3 i za preostale template M2, M3 i M4.
5. Ako postoje pikseli označeni za brisanje, postavi njihovu vrijednost na “0”.
6. Ako je i jedan piksel obrisani u koraku 5, ponavljaj od koraka 1, inače zaustavi proces.



Slika 166. [4] Ilustracija povezanosti

Slika mora biti skenirana određenim redom za svaki template. Templateom M1 slika se skenira slijeva udesno, odozgo prema dole. Njegova svrha je da pronade piksele uz gornji rub objekta koje je moguće ukloniti. Template M2 služi za markiranje piksela s lijeve strane objekta i on prelazi preko slike odozdo prema gore, slijeva udesno. M3 markira za uklanjanje piksele uz donji rub i pomijera se sdesna ulijevo, odozdo prema gore. Konačno, M4 markira piksele s desne strane, a pomijera se odozgo prema dole, pa sdesna ulijevo. Specifični redosljed primjenjivanja templatea osigurava uklanjanje piksela simetrično sa svih strana.



Slika 167. [4] Maske za označavanje piksela za brisanje pri određivanju skeleta