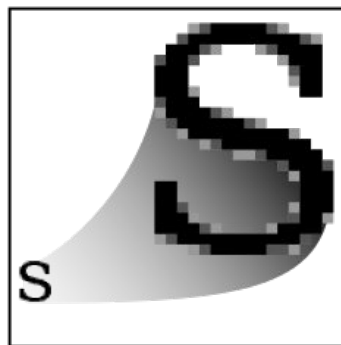


Pretvaranje rasterskih slika u vektorske (SVG)

Projekat iz Soft Computing-a
Vladimir Makarić

Motivacija za vektorsku grafiku



Raster

.jpeg .gif .png



Vector

.svg

Vektorska grafika je pogodna kao medij za prenos vizuelnih podataka iz više razloga:

- Malo zauzeće memorije kada je u pitanju grafički sadržaj poput dijagrama, tehničkih crteža, grafikona, GUI elemenata,...
- Nezavisnost kvaliteta prikaza od rezolucije uređaja. Posledica čuvanja grafičkog sadržaja pomoću matematičkih primitiva je primena invertibilnih transformacija nad slikom bez gubitka informacija ili degradacije sadržaja (rotacija, skaliranje, smicanje,...).

Ovi razlozi imaju još veću težinu u skorije vreme zbog velikog diverziteta uređaja (pogotovo mobilnih) za konzumaciju vizuelnih medija.

Motivacija za pretvaranje rasterske u vektorsku

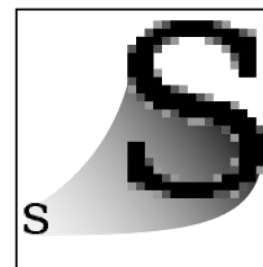
Postoji ogromna količina kvalitetnog grafičkog sadržaja koji je sačuvan u rasterskom obliku, čiji originalni oblik nije dostupan:

- Ilustracije u knjigama
- Tehnički crteži
- Dijagrami
- Karte
- ...

Korisno bi bilo kada bi se geometrijski oblici sa tih rasterskih slika mogli prepoznati, jer bi tada te slike mogle da se transformisu bez gubitaka i time bi se omogućilo njihovo štampanje ili prikaz na medijima visoke rezolucije.

SVG - Scalable Vector Graphics

- Format za čuvanje vektorske grafike razvijen od strane W3C
- Baziran na XML-u
- Podržavaju ga moderni Web pretraživači (Chrome, Firefox, Opera..)

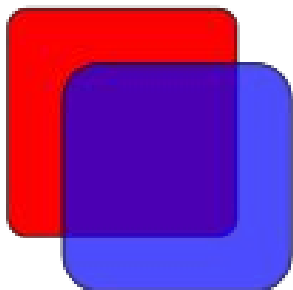


Raster
.jpeg .gif .png



Vector
.svg

Primer:

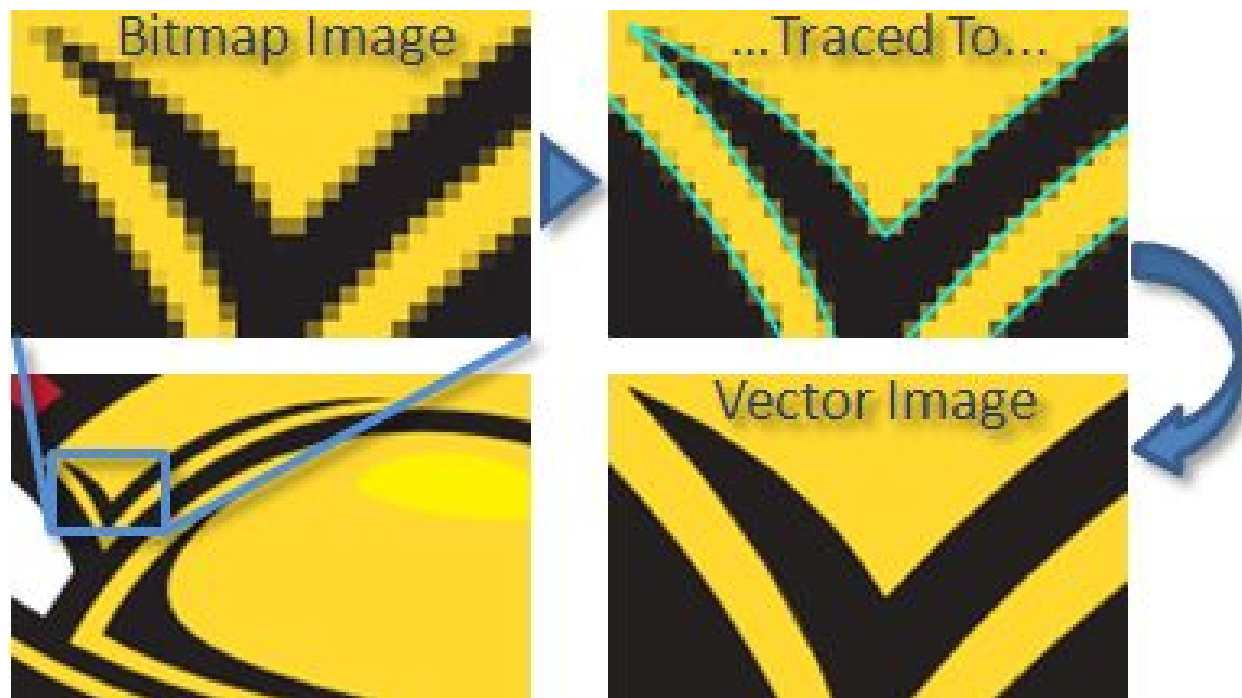


```
<svg xmlns="http://www.w3.org/2000/svg"
width="467" height="462">
  <rect x="80" y="60" width="250" height="250" rx="20"
    style="fill:#ff0000; stroke:#000000;stroke-width:2px;" />

  <rect x="140" y="120" width="250" height="250" rx="40"
    style="fill:#0000ff; stroke:#000000; stroke-width:2px;
    fill-opacity:0.7;" />
</svg>
```

Vektorizacija

Postupak pretvaranja rasterskih slika u vektorske se zove Image Tracing ili vektorizacija.



Postojeća rešenja

Popularan algoritam za vektorizaciju je Potrace (<http://potrace.sourceforge.net/potrace.pdf>).

Zadatak svakog algoritma za vektorizaciju je da napravi dobar trade-off u prepoznavanju ivica i glatkih krivih (uglavnom se koriste Bezierove krive). Ako se prepozna previše ivica, izlazni oblik će izgledati kao poligon i neće biti gladak. Sa druge strane ako se sve aproksimira sa glatkim krivama, izlazni oblik će izgledati previše zaobljeno:

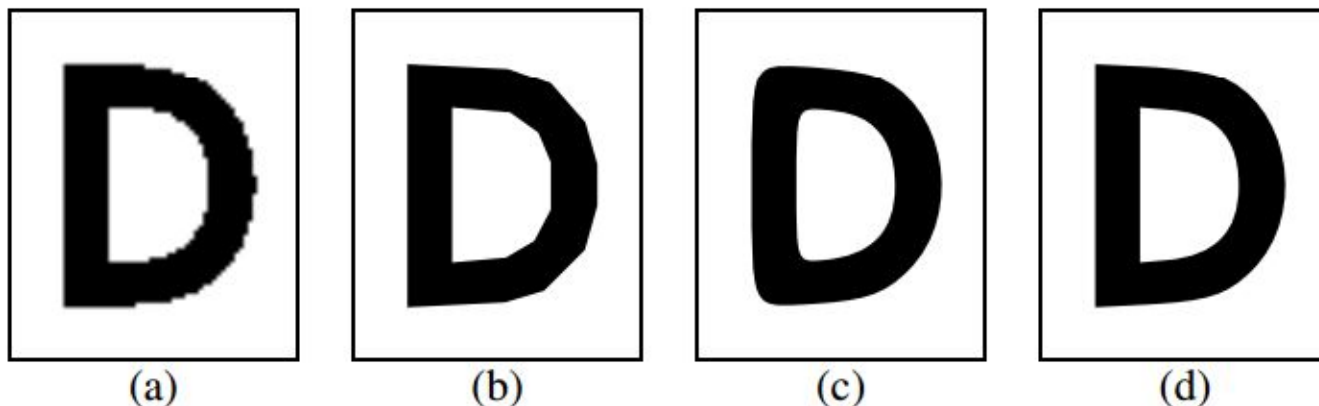


Figure 1: Corner detection. (a) the original bitmap; (b) too many corners; (c) too few corners; (d) good corner detection.

Opseg projekta

Potrace je složen algoritam prvenstveno zbog toga što mora da izračunava zakrivljenost ivica regija na rasterskim slikama i aproksimira ih Bezierovim krivama.

U ovom projektu ja bih se ograničio na rasterske slike bez krivih linija i kontinualnih prelaza izmedju boja, na primer crtežih geometrijskih figura, tehnički crteži, ER dijagrami, geometrijske ilustracije, etc.

Prepoznate vektorske oblike bih sačuvao kao SVG fajl. Takođe bih se ograničio na crno bele ilustracije.

Pošto se ograničavam na pravolinijske oblike, primitive kojima bi opisivao slike su poligoni i duži. Ključna stvar je diskriminacija između duži i pravougaonika.

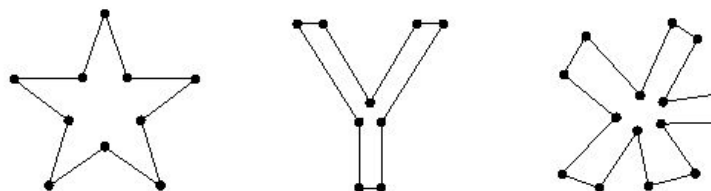
Plan rada

- Od oblika na slici se pomoću OpenCV dobije više poligona koji označavaju regije (konture)
- Problem je što ti poligoni nekad samo predstavljaju duži sa određenom debljinom, iako je vizuelno potpuno isto da li će se primitiva u SVG predstaviti kao poligon ili duž, razlika je u editovanju SVG-a kasnije, duž ima samo 2 kontrolne tačke dok četvorougao ima 4:



Plan rada

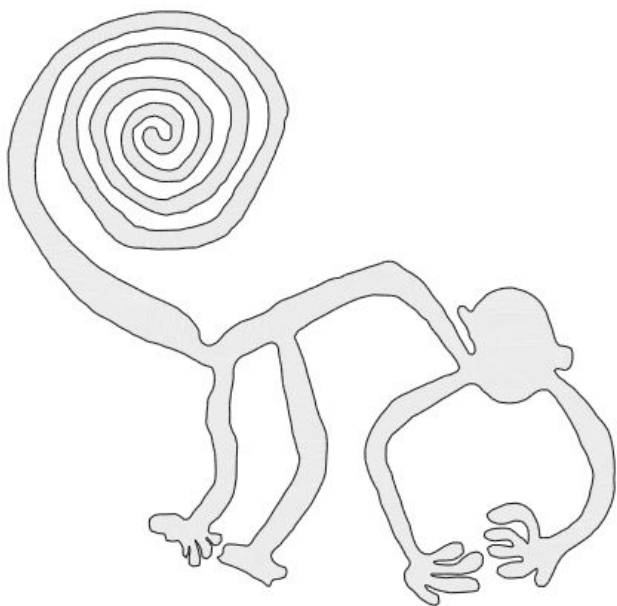
- Problem je što konture mogu biti konkavni poligoni:



- Jedan oblik prepoznat kao konkavan poligon može ustvari biti samo par duži spojenih u jednoj tački, kao Y na slici. Da bi mogao konkavan poligon da se rastavlja na linije potrebno ga je prvo dekomponovati na konveksne poligone, algoritmi koji ovo rade su veoma složeni i za tu potrebu bih koristio neku biblioteku za komputacionu geometriju, poput CGAL (<http://www.cgal.org/>)
- Posle bih konveksne poligone pretvarao u linije tako što bih proverio da li su ti poligoni ustvari obični zarotirani pravougaonici, ili koliko ih blisko aproksimiraju.

Rastavljanje na konveksne poligone

Primer rastavljanja konkavnog na konveksne poligone:



***Minimum Exact Convex
Decomposition***



***Approximate Convex
Decomposition***

