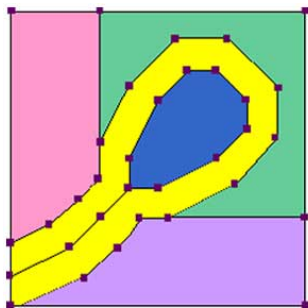


# Računarska grafika

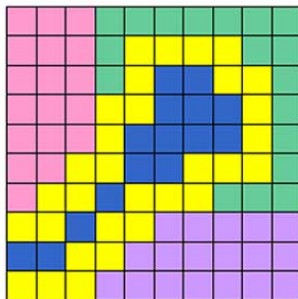
## Rasterizacija

Vesna Marinković

# Vektorski i rasterski sistemi



**Vector**



**Raster**

# Vektorska grafika (eps, svg, ai, drw, ps, cdr, ...)

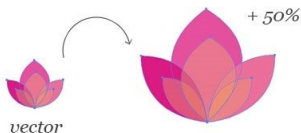
- Slika je predstavljena kao kolekcija geometrijskih figura (tačke, prave, krive, putanje, ...)
- Za svaku figuru čuvaju se njeni parametri (koordinate ključnih tačaka, dimenzija, boja, položaj na slici, ...)

```
<svg width="100"  
height="100">  
  <circle cx="50" cy="50"  
r="40" stroke="green" stroke-  
width="4" fill="yellow" />  
</svg>
```



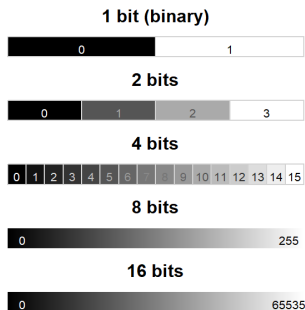
# Vektorska grafika

- Memorija koju slika zauzima zavisi od njenog sadržaja, često je manja od rasterske slike
- Skaliranje vektorske slike je jednostavno i sa savršenom preciznošću
- Vektorska grafika se koristi za fontove, logotipove, štampani materijal velikog formata, slike koje sadrže tekst, ...
- Nije pogodna za predstavljanje fotografija ili fotorealističnih slika
- Za obradu vektorskih slika potreban je poseban softver



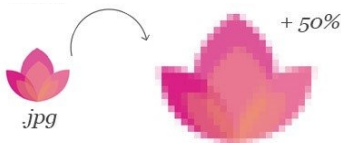
# Osnovni pojmovi rasterske grafike

- Ekranu je pridružena matrica piksela – bitmapa
- **Piksel** – najmanji grafički element rasterske slike sa pridruženim vrednostima intenziteta/boje/transparentnosti/dubine. . .
- **Bitska dubina** – broj bitova  $N$  koji se koristi za predstavljanje piksela
- Svaki piksel ima jednu od  $2^N$  vrednosti



# Rasterska grafika (jpg, png, gif, bmp, tiff, psd, ...)

- Memorija koju slika zauzima zavisi od rezolucije i bitske dubine
- Problemi pri skaliranju slike (nazubljenost, zamućenost)
- Koristi se za fotografije, u veb dizajnu, ...



- Slika se može kompresovati da bi se smanjila veličina datoteke
  - kompresija sa gubicima (*lossy*)
  - kompresija bez gubitaka (*lossless*)

# JPEG vs. PNG vs. GIF

- JPEG (Joint Photographic Expert Group)
  - Podržava 24-bitne boje
  - Dobar izbor za slike visoke rezolucije bez teksta
  - Koristi kompresiju sa gubicima
  - Ne podržava transparentnu pozadinu  
(slike su uvek pravougaonog formata sa punom pozadinom)
  - Manja veličina datoteke (u odnosu na PNG) za sličan kvalitet slike
- PNG (Portable Network Graphic)
  - Podržava 24-bitne boje
  - Koristi kompresiju bez gubitaka
  - Veličina datoteke slika u visokoj rezoluciji je velika
  - Podržava transparentnu pozadinu (pogodna za logoe)
- GIF (Graphics Interchange Format)
  - Podržava samo 8-bitne boje
  - Osnovna prednost u odnosu na JPEG i PNG je da kompresuje digitalne slike u manje datoteke (smanjivanjem broja boja)
  - Podržava transparentnost i kratke animacije male veličine

# JPEG vs. PNG vs. GIF

svojstvo datoteke	JPG	PNG	GIF
može se prikazati na veb strani	Da	Da	Da
kompresija sa gubicima	Da	Ne	Ne
kompresija bez gubitaka	Ne	Da	Da
8-bitne boje	Da	Da	Ne
16-bitne boje	Ne	Da	Ne
transparentnost	Ne	Da	Da
animacije	Ne	Ne	Da



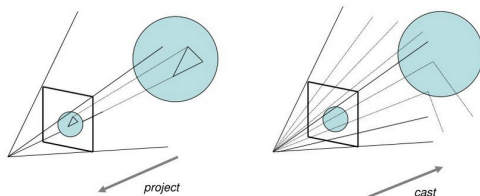
# Dve tehnike prikazivanja objekata na ekranu

## ● Rasterizacija

- elementi scene se projektuju ka kameri
- **za svaku primitivu** (npr. trougao) utvrditi koje piksele treba osvetliti
- efikasna (moguće je prikazati milijarde trouglova u sekundi)
- teže je postići fotorealizam

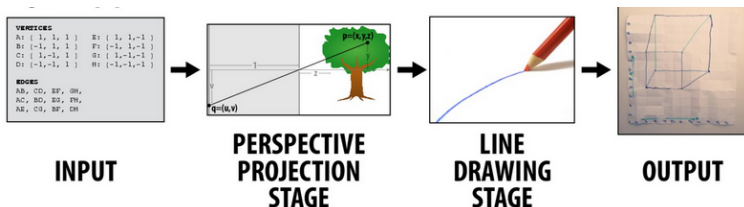
## ● Tehnika praćenja zraka (eng. ray tracing)

- zraci se iz kamere puštaju ka sceni
- **za svaki piksel** utvrditi koje primitive se vide
- u opštem slučaju je sporija
- lakše je postići fotorealizam



# Protočna obrada za generisanje slike 3D objekta

- Proces generisanja slike možemo videti kao **protočnu obradu**
  - **ulaz** – slika koju želimo da nacrtamo
  - **niz etapa** – niz transformacija koje ulaz transformišu u izlaz
  - **izlaz** – finalna slika
- Primer protočne obrade sa prethodnog časa



Veliki deo slajdova je preuzet sa kursa Univerziteta Carnegie Mellone

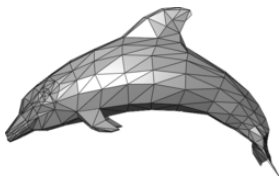
# Protočna obrada za proces rasterizacije

- Generisanje slika u realnom vremenu se zasniva na rasterizaciji
  - **ulaz** – 3D primitive (suštinski trouglovi)
  - **izlaz** – rasterska slika (za svaki piksel njegoja boja, potencijalno dubina, transparentnost, ...)



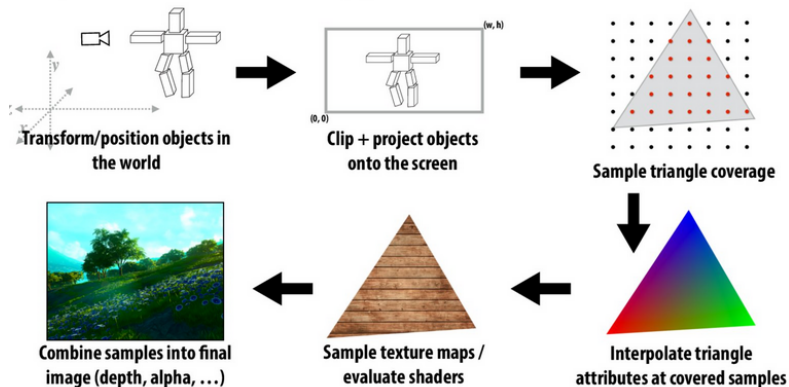
- **Naš cilj:** razumevanje svih etapa u grafičkoj protočnoj obradi

# Zašto trouglovi?



- Najjednostavnija geometrijska figura koja ima površinu
- Sve primitive je moguće predstaviti trouglovima
- Mogu da aproksimiraju proizvoljni oblik
- Uvek leže u ravni, normala je dobro definisana u 3D
- Vrednosti u temenima se lako interpolišu u tačkama u unutrašnjosti
- Kada sve svedemo na trouglove, možemo se fokusirati na razvoj optimizovane protočne obrade za trouglove

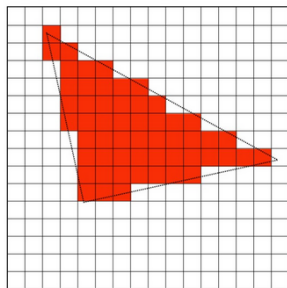
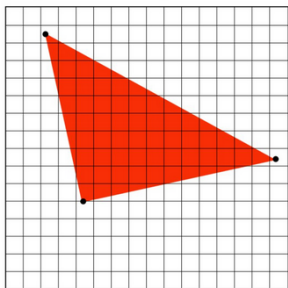
# Gruba skica protočne obrade kod rasterizacije



- Odgovara standardnoj grafičkoj obradi (OpenGL, Direct3D)

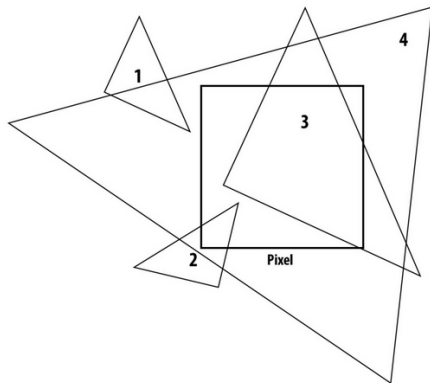
# Računanje prekrivenosti trouglom

- **Pitanje:** koje piksele trougao prekriva?
- **Ulaz:** projektovane pozicije temena trougla  $P_0, P_1, P_2$
- **Izlaz:** skup piksela prekrivenih trouglom



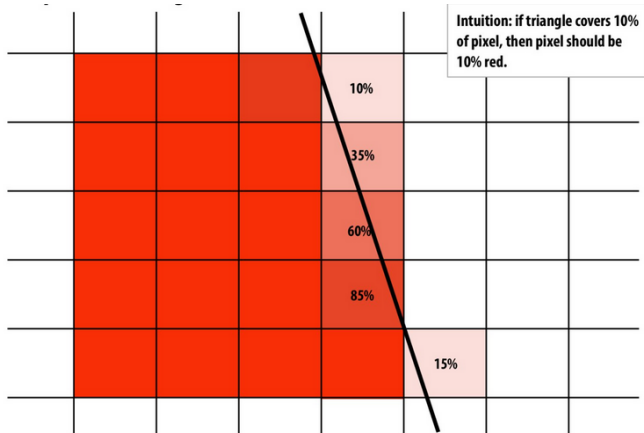
# Šta tačno znači da je piksel prekriven trouglom?

- **Pitanje:** koji od trouglova 1, 2, 3 i 4 prekrivaju dati piksel?



# Šta tačno znači da je piksel prekriven trouglom?

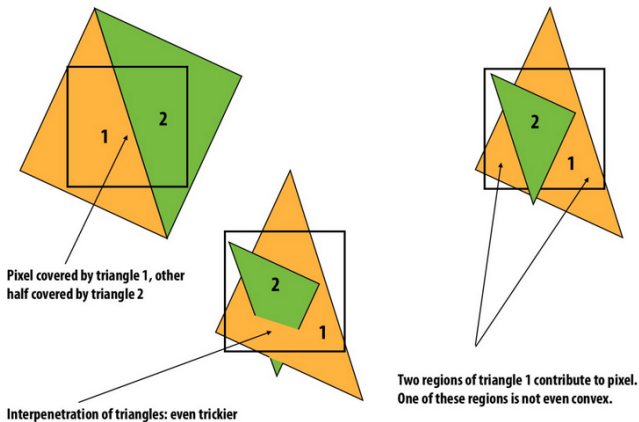
- Jedna mogućnost: izračunati procenat površine piksela prekrivene trouglom i obojiti piksel nijansom koja odgovara tom procentu





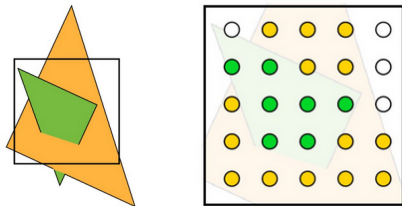
# Šta tačno znači da je piksel prekriven trouglom?

- Prekrivenost postaje problematična kada razmatramo više trouglova



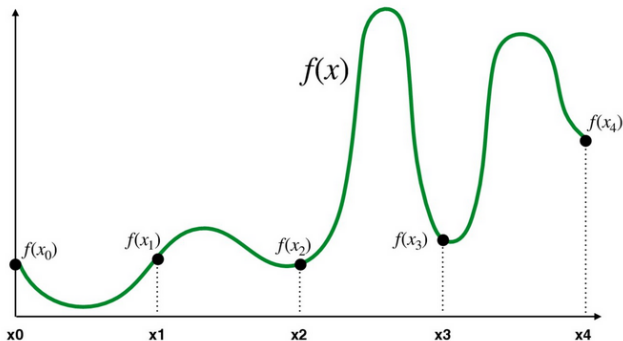
# Prekrivenost putem uzorkovanja

- Realistične scene mogu biti jako složene
- Izračunavanje tačne vrednosti prekrivenosti nije praktično
- Razmatraćemo prekrivenost kao problem uzorkovanja
- Sa dovoljnim brojem tačaka i pametnim izborom lokacija tačaka, možemo dobiti dobru procenu prekrivenosti



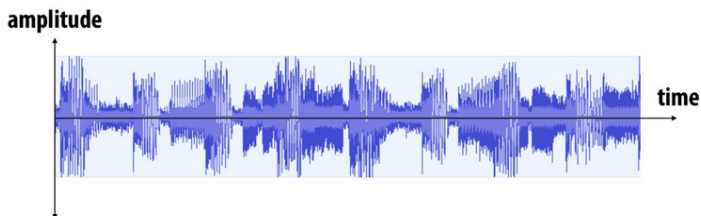
# Uzorkovanje u 1D

- Razmotrimo 1D signal
- Signal je funkcija koja varira u vremenu ili prostoru
- **Uzorkovanje** (eng. sampling) podrazumeva merenje vrednosti signala



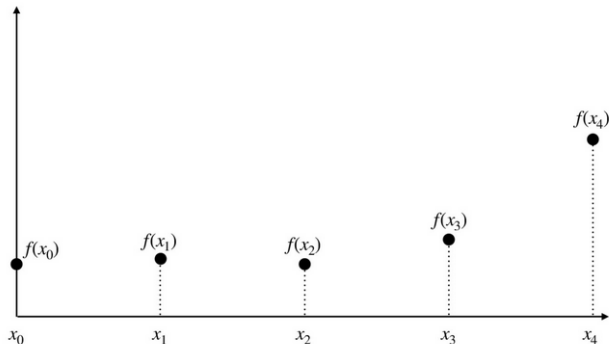
# Primer: Audio fajl

- Čuva uzorke 1D signala
- Najčešće se uzorkuje 44.100 puta u sekundi, tj. na 44.1KHz



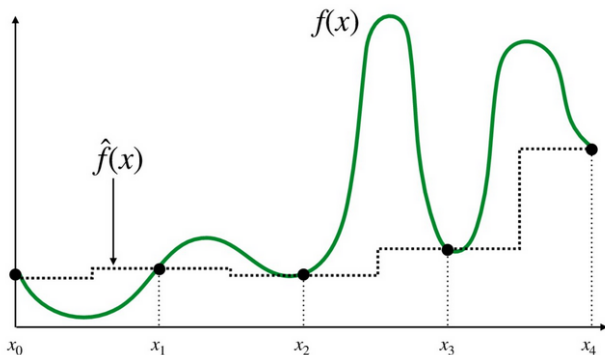
# Rekonstrukcija u 1D

- Proces inverzan uzorkovanju je rekonstrukcija
- Za dati skup uzoraka, kako rekonstruisati originalni signal  $f(x)$ ?



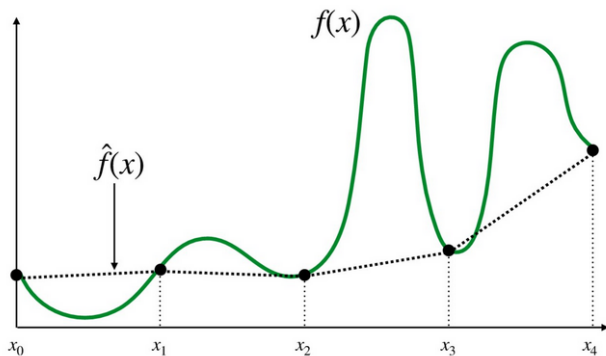
# Deo-po-deo konstantna aproksimacija

- $\hat{f}(x)$  - vrednost uzorka koji je najbliži tački  $x$



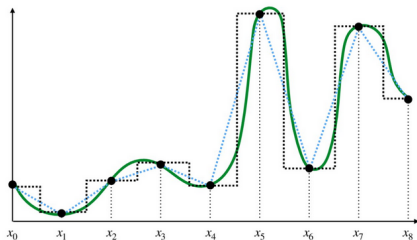
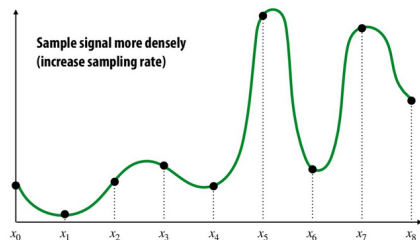
# Deo-po-deo linearna aproksimacija

- $\hat{f}(x)$  - linearna interpolacija između vrednosti dva najbliža uzorka tački  $x$



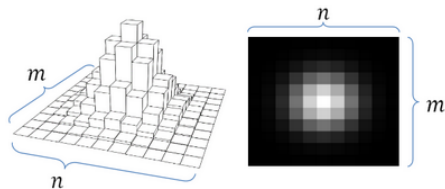
# Kako preciznije predstaviti signal?

- Gušće uzorkovati signal (povećati učestalost uzorkovanja)



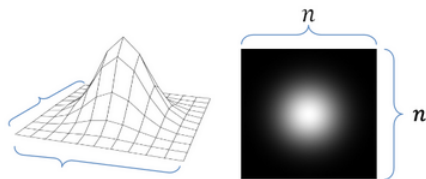


# Slike kao diskretni signali



- Na slike smo navikli da gledamo kao na niz piksela – **diskretni signal**
- Diskretni signali su funkcije  $\mathbb{Z}^2 \rightarrow \mathbb{R}$ , jedna vrednost za jedan par celobrojnih koordinata  $(i, j)$
- Čuvaju se kao niz u memoriji

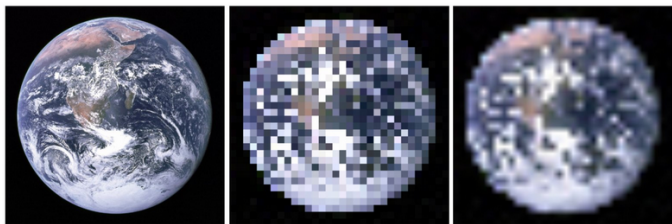
# Slike kao neprekidni signali



- Svaka slika počinje svoj život kao **neprekidni signal**
- Svetlost koja ulazi u kameru je neprekidni signal u ravni 2D slike
- Uzorkovanje predstavlja transformisanje neprekidnog signala u diskretan: fotografisanjem, skeniranjem slike, ...
- Vrednosti koje se čuvaju u diskretnim pikselima slike su uzorci originalnog signala

## Uzorkovanje i rekonstrukcija u 2D

- Stvari se odvijaju slično i u 2D
- Vršiti se uzorkovanje vrednosti na skupu tačaka
- Primenjujemo interpolacioni/rekonstrukcijski filter da bismo aproksimirali sliku



**original**

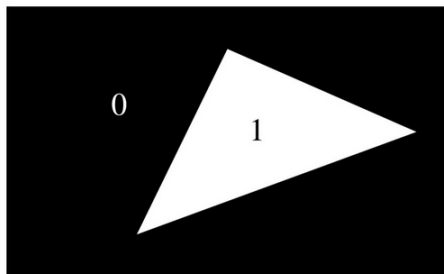
**piecewise constant  
("nearest neighbor")**

**piecewise *bi*-linear**

# Rasterizacija kao uzorkovanje

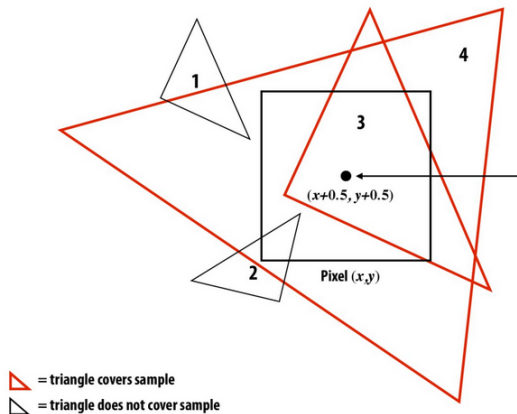
- Koju funkciju treba uzorkovati u slučaju rasterizacije?

$$\text{prekrivenost}(x, y) = \begin{cases} 1, & \text{trougao sadrži tačku } (x, y) \\ 0, & \text{inače} \end{cases}$$



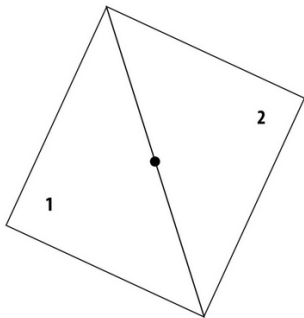
## Uzorkovanje funkcije prekrivenosti

- Jednostavna varijanta rasterizacije se oslanja na uzorkovanje funkcije prekrivenosti
- Kao tačku u kojoj vršimo uzorkovanje biramo centar piksela



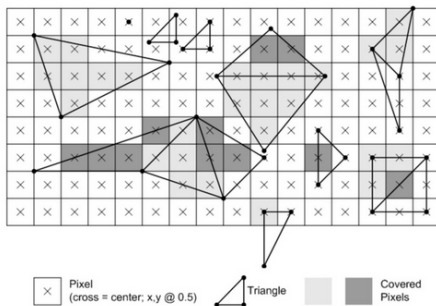
# Granični slučajevi

- Da li je data tačka uzorkovanja prekrivena prvim trouglom?
- Da li je data tačka uzorkovanja prekrivena drugim trouglom?
- Da li je data tačka uzorkovanja prekrivena sa oba trougla?

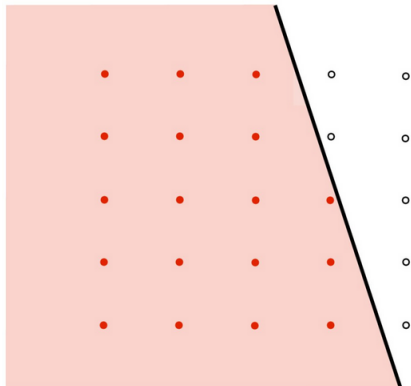


## Kako razrešiti prethodni problem?

- Kada se tačka uzorkovanja nalazi na samoj ivici trougla, klasifikuje se kao unutrašnja ako je u pitanju gornja ivica ili leva ivica
  - gornja ivica – horizontalna ivica koja je iznad ostalih ivica
  - leva ivica – ivica koja nije horizontalna i koja je sa leve strane trougla (trougao može imati dve leve stranice)



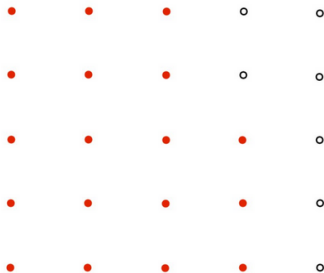
# Rezultat uzorkovanja prekrivenosti trougla





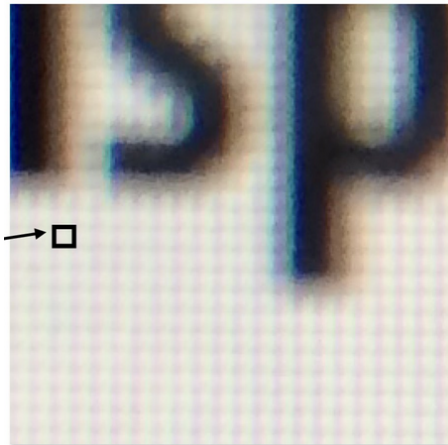
# Rezultat uzorkovanja prekrivenosti trougla

- Kako uzorkovani signal prikazati na ekranu?

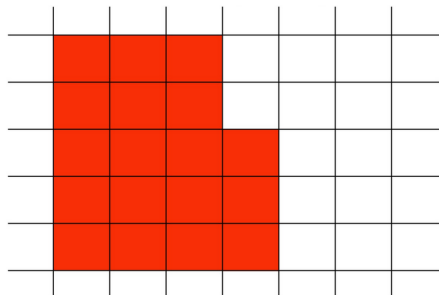
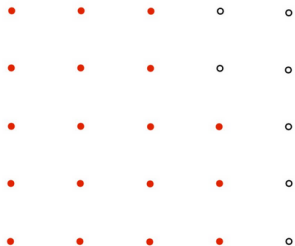


## Pikseli na ekranu

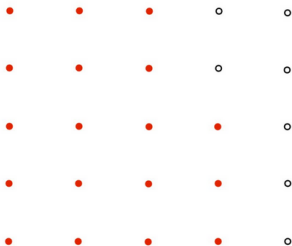
- Svaki uzorak slike koji se šalje ekranu se konvertuje u kvadratić svetlosti odgovarajuće boje



# Aproksimacija prikaza uzorka slike na ekranu

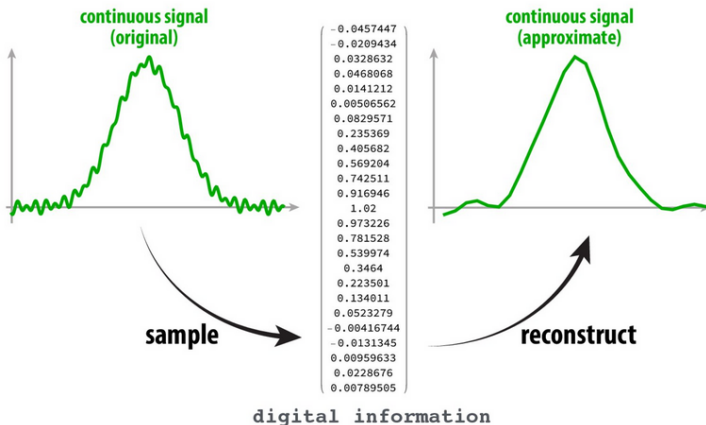


# Prikaz stvarnog signala prekrivenosti trougla



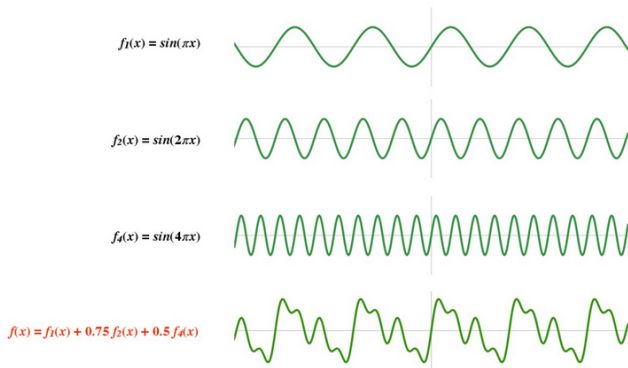
# Uzorkovanje i rekonstrukcija signala

- Cilj: reprodukovati originalni signal što preciznije moguće



# Dekomponovanje 1D signala na frekvencije

- 1D signal se može izraziti kao superpozicija (suma) različitih frekvencija (tj. sinusnih talasa određene frekvencije)
- Sinusni talas je određen amplitudom, frekvencijom i fazom (ofsetom)
- Visoka frekvencija odgovara brznoj promeni, a niska sporoj promeni



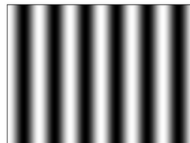
# Sinusni talas u 2D

- Maksimalnu pozitivnu amplitudu označimo belom bojom, a maksimalnu negativnu crnom

Low Frequency



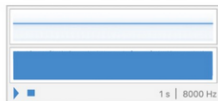
Higher Frequency



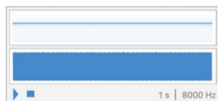
# Aliasing u audio snimku

- Konstantan ton se može dobiti sviranjem sinusoide frekvencije  $\omega$

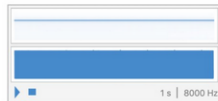
Play[Sin[4000 t], {t, 0, 1}]



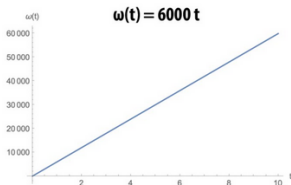
Play[Sin[5000 t], {t, 0, 1}]



Play[Sin[6000 t], {t, 0, 1}]



- Online generator tonova: <https://www.szynalski.com/tone-generator/>
- Šta će se dogoditi ako povećavamo vrednost  $\omega$  tokom vremena?



Play[Sin[ $\omega$  t], {t, 0, 10}]

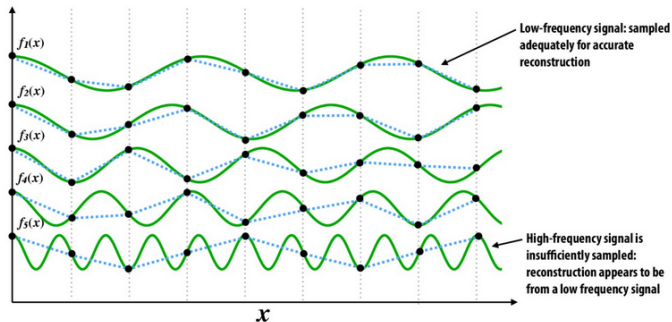


- Zašto se to dogodilo?



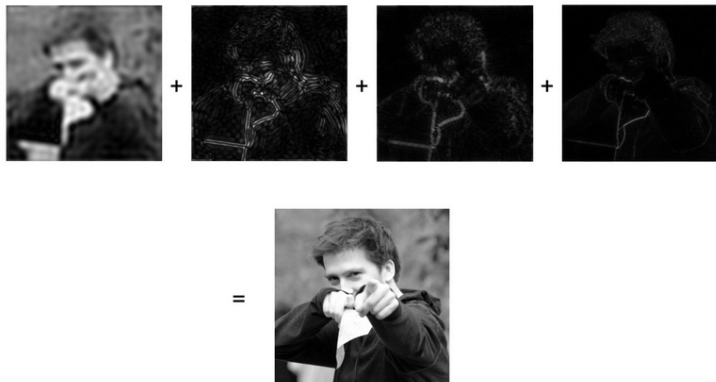
# Aliasing u 1D signalu

- Nedovoljno uzorkovanje signala visoke frekvencije rezultuje aliasingom
- Visoke frekvencije u originalnom signalu deluju kao niske frekvencije nakon rekonstrukcije



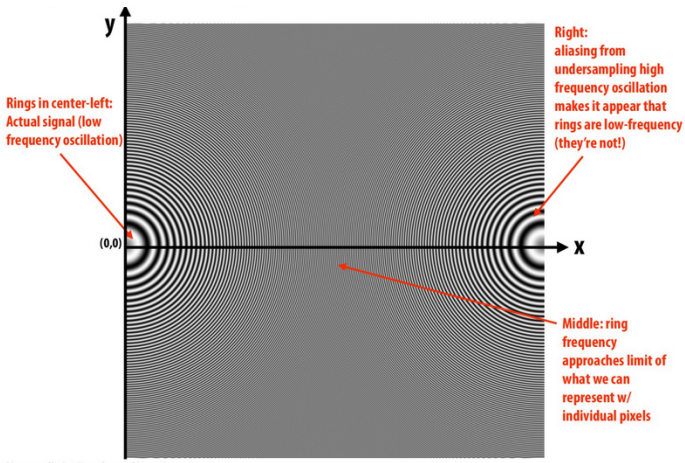
# Dekomponovanje slike na frekvencije

- I slike je moguće dekomponovati u frekvencije
- Dekomponujemo sliku na sumu njenih frekvencija



# Prostorni aliasing

- Razmotrimo funkciju  $\sin(x^2 + y^2)$



# Temporalni aliasing efekt

- Nekad nam se na filmu može učiniti da se gume automobila okreću unazad
- Ovaj efekt naziva se **temporalni aliasing**
- Ilustracija ovog efekta:  
<https://www.youtube.com/watch?v=GuKFup1lGc4&t=15s>

# Najkvist-Šenonova teorema

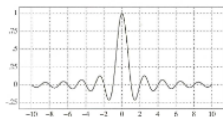
- Razmotrimo signal ograničenog opsega koji nema frekvencije više od nekog praga  $\omega_0$ 
  - primer u 1D: audio-signal iz koga su uklonjene visoke frekvencije
  - primer u 2D: zamućena slika sa nekog od prethodnih slajdova



- Ako signal nema frekvencije više od  $\omega_0$ , on se može **savršeno rekonstruisati** ako se uzorkuje sa učestanošću dvostruko većom od najviše frekvencije  $\omega_0$

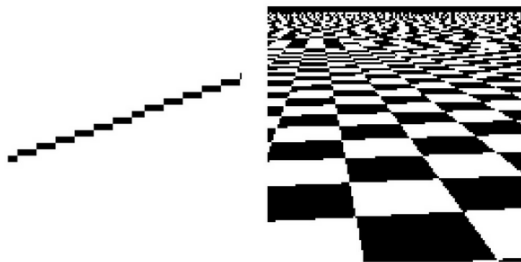
# Prefilterovanje

- Da li možemo uzorkovati sa višom frekvencijom?
  - nije uvek moguće
  - u realnom svetu duži imaju beskonačno visoke frekvencije, ne možemo uzorkovati na dovoljno visokoj frekvenciji
- Koristimo prefilterovanje da se ograniči signal
  - filterovanjem se slika zamućuje
  - prihvatamo zamućenost umesto aliasing efekta
- Umesto deo-po-deo konstantne ili deo-po-deo linearne interpolacije koristi se tzv. **sinc filter**:  $\text{sinc}(x) = \frac{1}{\pi x} \sin(\pi x)$



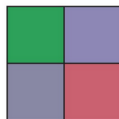
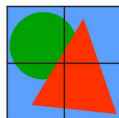
# Aliasing efekt na slikama

- Nesavršeno uzorkovanje + nesavršena rekonstrukcija vodi neprirodnim pojavama na slikama
  - nazubljene linije
  - svetlucanje na slikama kada se animiraju
  - Moire uzorak na oblastima slike koje imaju visoku frekvenciju



## Kako redukovati aliasing efekat?

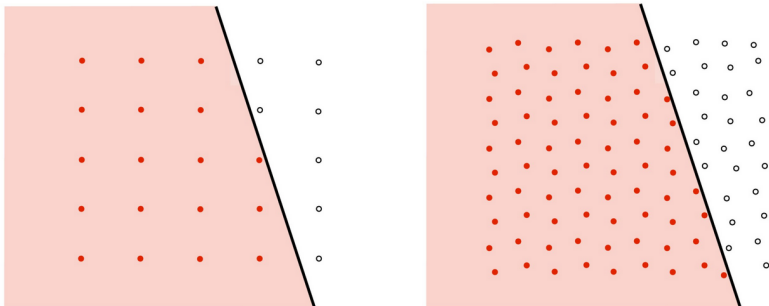
- Aliasing se ne može u potpunosti eliminisati: svaka uzorkovana reprezentacija ne uspeva da uhvati dovoljno visoke frekvencije
- Potrebno je što bolje upariti uzorkovanje i rekonstruisanje
- Ukupno svetlo emitovano od strane piksela treba da bude jednako ukupnom svetlu u neprekidnom signalu
- Vrednost piksela dobijamo integraljenjem ulaznog signala nad pikselom
- Signal funkcije prekrivenosti možemo približno integraliti uzorkovanjem





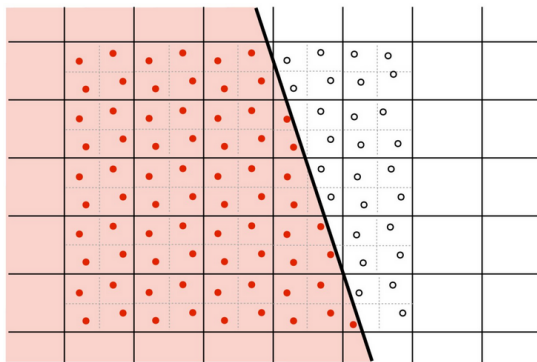
# Kako redukovati aliasing efekt?

- Grešku rekonstrukcije signala možemo smanjiti povećavanjem frekvencije uzimanja uzoraka



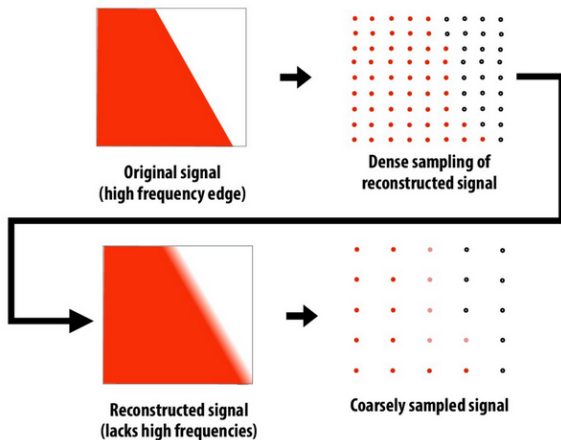
# Kako redukovati aliasing efekat?

- Umesto jednog uzorka po pikselu možemo uzimati više uzoraka po pikselu – **nadsemplovanje**



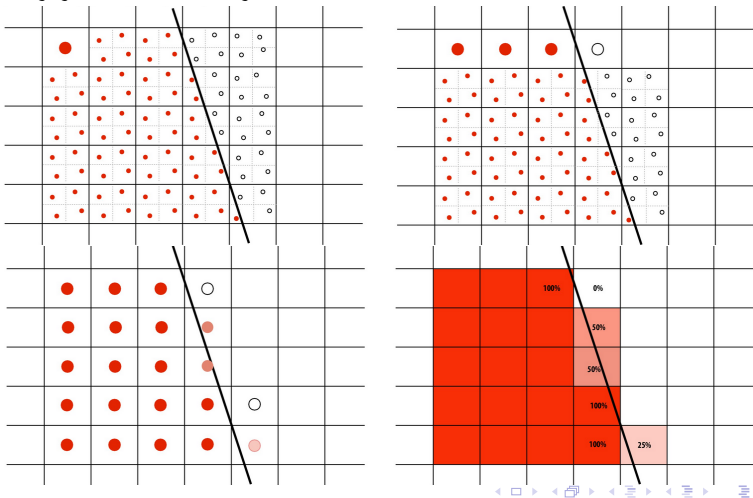
# Ponovno uzorkovanje

- Konverzija iz jedne reprezentacije dobijene uzorkovanjem u drugu

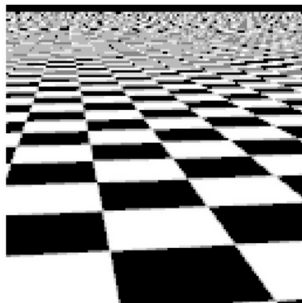


# Ponovno uzorkovanje na rezoluciju ekrana

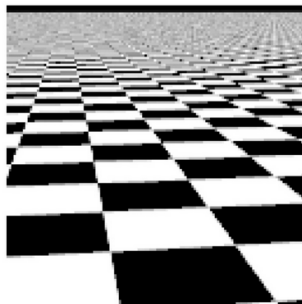
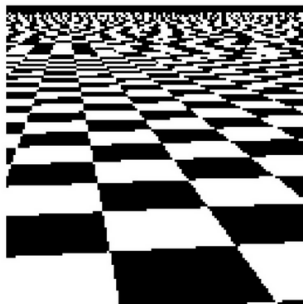
- Ekran prikazuje jednu vrednost uzorka na poziciji jednog piksela
- Primenjujemo filter kutije



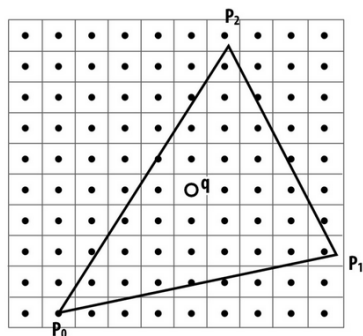
# Uzorkovanje vs. Nadsemplovanje $2 \times 2$



# Uzorkovanje vs. Nadsemplovanje $32 \times 32$

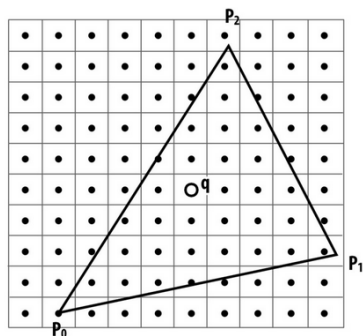


# Ispitivanje da li je tačka u trouglu



- Kako utvrditi da li je tačka  $q$  unutar  $\triangle P_0P_1P_2$ ?

# Ispitivanje da li je tačka u trouglu

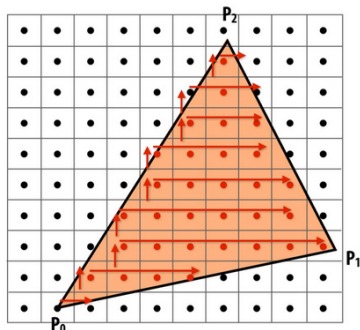


- Kako utvrditi da li je tačka  $q$  unutar  $\triangle P_0P_1P_2$ ?
- Proveravamo da li je tačka  $q$  levo od pravih  $P_0P_1$ ,  $P_1P_2$  i  $P_2P_0$



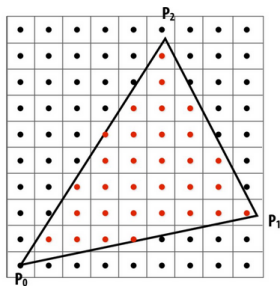
# Tradicionalni pristup rasterizaciji trougla

- Provera da li je tačka levo od prave daje slične vrednosti za bliske tačke, možemo koristiti inkrementalni pristup



## Moderniji pristup rasterizaciji trougla

- Inkrementalni pristup rasterizaciji pretpostavlja serijsko izvršavanje, dok moderni hardver podrazumeva paralelno izvršavanje
- Alternativa: testirati paralelno sve uzorke iz graničnog opsega trougla
- Mogućnost paralelnog izvršavanja nadomešćuje cenu većeg broja testiranja uzoraka koje je potrebno izvršiti
- Moderne grafičke jedinice imaju hardver specijalne namene za efikasno testiranje da li je tačka u trouglu

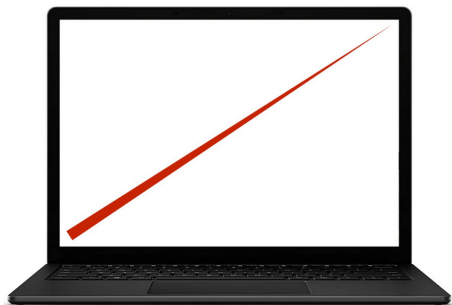


## Moderniji pristup rasterizaciji trougla

- **Pitanje:** u kom slučaju je paralelni pristup veoma neefikasan?

## Moderniji pristup rasterizaciji trougla

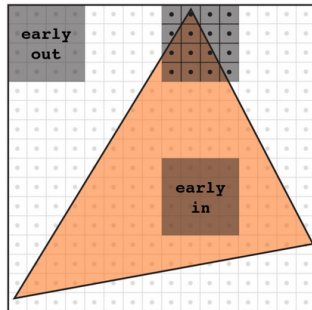
- **Pitanje:** u kom slučaju je paralelni pristup veoma neefikasan?



- Ako imamo dugačak tanki trougao, treba testirati veliki broj uzoraka na pripadnost trouglu, a skoro nijedan neće pripadati

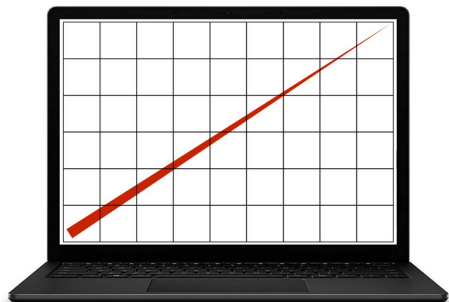
# Hibridni pristup rasterizaciji trougla (popločavanje trougla)

- Granični opseg podeliti na blokove srednje veličine
- Pre provere pojedinačnih piksela, za svaki blok proveriti da li preseca trougao
  - ako blok ne preseca trougao, eliminišemo sve piksele iz bloka
  - ako je čitav blok sadržan unutar trougla, zaključujemo da trougao prekriva sve tačke tog bloka
  - inače, testiramo u paraleli pojedinačno sve tačke iz bloka

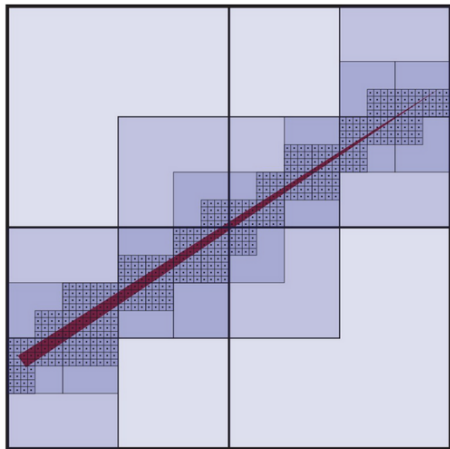


# Hibridni pristup rasterizaciji trougla

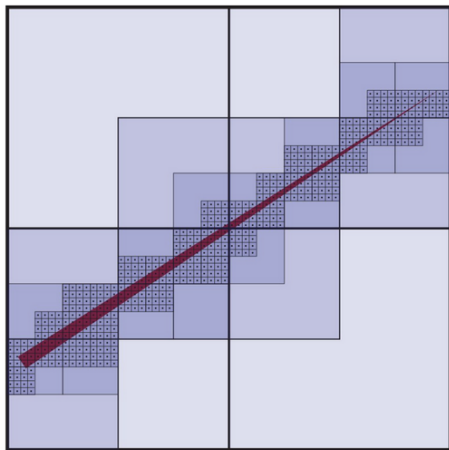
- U slučaju dugačkog uskog trougla, možemo preskočiti veliki broj malih blokova, još uvek je neefikasno
- **Pitanje:** da li možemo još bolje u ovom primeru?



# Hijerarhijski pristup rasterizaciji trougla



# Hijerarhijski pristup rasterizaciji trougla



- **Pitanje:** da li postoji bolji način da nađemo najfinije blokove?
- **Odgovor:** možda i to možemo tehnikom inkrementalnosti



# Zaključak

- Mnoge probleme u grafici možemo posmatrati u terminima **uzorkovanja i rekonstrukcije**
  - **uzorkovanje**: transformisanje neprekidnog signala u digitalni oblik
  - **rekonstrukcija**: transformisanje digitalne informacije u neprekidni signal
  - **aliasing** se javlja kada rekonstruisani signal daje pogrešnu sliku o tome kako je izgledao polazni signal
- **Problem rasterizacije** se može videti kao problem uzorkovanja
  - uzorkujemo funkciju prekrivenosti na mreži piksela
  - rekonstruišemo signal emitovanjem kvadrata svetlosti za svaki piksel
  - aliasing se javlja u vidu nazubljenih ivica, efekta svetlucanja, ...
  - aliasing se može redukovati nadsemplovanjem
- **Rasterizacija trougla** - osnovni gradivni blok grafičke protočne obrade
  - svodi se na tri testa pripadnosti poluravni
  - atomička operacija – treba da bude brza
  - nekoliko pristupa: inkrementalni, paralelni, blokovski, hijerarhijski, ...