|  |  |  |
| --- | --- | --- |
|  | УНИВЕРЗИТЕТ У НОВОМ САДУ  **ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У НОВОМ САДУ** |  |

Milan Đokić

**Interpolacija slike**

Ispitni rad

- OAISDSP2-

Mentor: Dejan Bokan

Novi sad, 2017

**Sadržaj**

[1. Uvod 1](#_Toc485144630)

[2. Interpolacija slike 2](#_Toc485144631)

[2.1 Sample and hold algoritam 2](#_Toc485144632)

[2.2 Bilinearna interpolacija 4](#_Toc485144633)

[2.3 Bikubična interpolacija 7](#_Toc485144634)

[2.4 Rotacija slike 9](#_Toc485144635)

[3. Zaključak 11](#_Toc485144636)

# Uvod

Digitalne slike i video zapisi sadrže veliki broj podataka, čiji obim raste sa napretkom tehnike, što dovodi do prepreka prilikom prenosa multimedijalnog sadržaja, koji mora biti prenet u odgovarajućem roku, kao i bez gubitka kvaliteta. Kako bi se zadovoljili zahtevi kvaliteta, brzine prenosa i prilagođenja prenetih podataka ciljnim sistemima, obradi slike se pridaje sve veći značaj. Ušteda propusnog opsega mreže može se postići kodovanjem slike niske rezolucije na strani enkodera, koja se onda, na strani dekodera, pre samog prikazivanja krajnjem korisniku, uvećava do rezolucije modernih panela.

Povećanje slike do željene visoke rezolucije vrši se nekom od tehnika interpolacije. Jedan od važnih primera rastuće potrebe za interpolacijom je i prikaz TV signala standardne definicije (SD TV) na savremenim panelima koji su mahom veće rezolucije a dosta često i različitih proporcija (prikaz standardnog 4:3 SD signala na 16:9 HD panelu).

Pored navedenih primena interpolacija slike se koristi prilikom uveličavanja slike (*zoom*), izvršenja geometrijskih transformacija slike (kao što je rotiranje), popravljanje-smetnji u slici (engl. *image inpainting*) ili estimacija pokreta sa ne-celobrojnom tačnošcu.

# Interpolacija slike

## Sample and hold algoritam

Ovo je najjednostavniji algoritam u kojem se za interpoliranu vrednost uzima poznata vrednost iz najbliže tačke u osnovnom rasteru.

Iout[p, q] = Iin[(p-1)/Fx + 1], [(q-1)/Fy+1])

p, q – Horizontalni i vertikalni indeks slike

Fx, Fy – Horizontalni i vertikalni faktor skaliranja

Implementacija

Obrada slike se vrši u YUV format, s obzirom da se na ulazu i izlazu očekuje slika u RGB formatu, na početku i na kraju algoritma vrši se konverzija iz jednog u drugi format.

Prvo se izračunaju horizontalni i vertikalni faktor skaliranja slike:

xFactor = newXSize/xSize

yFactor = newYSize/ySize

Vrednost svakog piksela izlazne slike se interpolira sa vrednošću najbližeg piksela ulazne slike.

Iout[i, j] = Iin[round((i-1)/yFactor + 1, round(j-1)/xFactor + 1]

i – horizontalna pozicija piksela izlazne slike

j – vertikalna pozicija piksela izlazne slike



Ulazna slika



Deo Izlazne slike nakon uvećanja 4x4

## Bilinearna interpolacija

Bilinearna interpolacija je nešto kompleksniji algoritam gde se koeficijenti interpolacije računaju na osnovu udaljenosti tačaka iz osnovnog rastera od interpolacione tačke. Osnovna ideja bilinearne interpolacije je da se prvo izvede linearna interpolacija po jednoj dimenziji slike, a potom po drugoj. Za razliku od prethodno opisane tehnike bilinearna interpolacija koristi 4 najbliže vrednosti tačaka, locirane u dijagonalnim pravcima od trenutnog piksela. Bilinearna interpolacija koristi oblast 2x2 poznatih vrednosti piksela koji okružuju nepoznati piksel. Interpolacija se zasniva na usrednjavanju te 4 vrednosti po formuli sledećoj formuli:

Y = (1-a)\*(1-b)\*X(m, n) + (1-a)\*b\*X(m+1, n) + a\*(1-b)\*X(m, n+1) + a\*b\*X(m+1, n+1)

Vrednosti a i b se izračunavaju po sledećoj formuli:

a = ns/Sh – floor(ns/Sh)

b = ms/Sv – floor(ms/Sv)

Gde su:

*ns/ms* – horizontalni/vertikalni indeks piksela u skaliranoj slici (pozicija)

*Sh/Sv* – horizontalni/vertikalni faktor skaliranja

Implementacija

Obrada slike se vrši u YUV format, s obzirom da se na ulazu i izlazu očekuje slika u RGB formatu, na početku i na kraju algoritma vrši se konverzija iz jednog u drugi format.

Prvo se izračunaju horizontalni i vertikalni faktor skaliranja slike:

xFactor = newXSize/xSize

yFactor = newYSize/ySize

Izračuna se horizontalna i vertikalna pozicija piksela u ulaznoj slici:

yPos = (i-1) / yFactor + 1

xPos = (j-1)/xFactor + 1

i – horizontalna pozicija piksela izlazne slike

j – vertikalna pozicija piksela izlazne slike

Izračunaju se pozicije piksela ulazne slike koji okružuju interpolirani piksel izlazne slike:

m = floor(yPos)

n = floor(xPos)

Vrši se interpolacija sa pikselima ulazne slike Iin[m, n], Iin[m, n+1], Iin[m+1, n],

Iin[m+1, n+1].



Ulazna slika



Deo Izlazne slike nakon uvećanja 4x4

## Bikubična interpolacija

Bikubična interpolacija daje značajno bolji kvalitet interpolirane slike u odnosu na *Sample and hold* metodu na račun povećane složenosti izračunavanja. Kod bikubične interpolacije se koeficijenti interpolacije računaju na osnovu udaljenosti tačaka iz osnovnog rastera od interpolacione tačke. Pri interpolaciji koristi se 16 piksela iz originalnog rastera (4 x 4 okolina) i vrednosti piksela koji su bliži interpoliranom pikselu imaju veći uticaj na interpoliranu vrednost.

Osnovna ideja bikubične interpolacije jeste da se prvo uradi kubična (jednodimenziona) interpolacija po jednoj dimenziji slike, a potom po drugoj. Kubična interpolacija predstavlja jednodimenzioni algoritam interpolacije opisan sledećom jednačinom:

w(d) = 3/2\*abs(d^3) – 5/2\*abs(d^2) + 1, abs(d) < 1

w(d) = 3/2\*abs(d^3) – 5/2\*abs(d^2) + 1, 1 <= abs(d) < 2

w(d) = 0 , inače

gde je d udaljenost između interpoliranog piksela i piksela originalnog rastera. Izračunavanje vrednosti interpolacione tačke vrši tako što se nad 4 reda od po 4 tačke poziva kubična interpolacija. Dobijena 4 rezultata smestiti u niz od 4 elementa, i nad njima ponovo pozvati kubičnu interpolaciju.

Implementacija

Obrada slike se vrši u YUV format, s obzirom da se na ulazu i izlazu očekuje slika u RGB formatu, na početku i na kraju algoritma vrši se konverzija iz jednog u drugi format.

Prvo se izračunaju horizontalni i vertikalni faktor skaliranja slike:

xFactor = newXSize/xSize

yFactor = newYSize/ySize

Izračuna se horizontalna i vertikalna pozicija piksela u ulaznoj slici:

yPos = (i-1) / yFactor + 1

xPos = (j-1)/xFactor + 1

i – horizontalna pozicija piksela izlazne slike

j – vertikalna pozicija piksela izlazne slike

Izračunaju se 4x4 pozicije susednih piksela koji okružuju interpolirani piksel.

Zatim se vrše 4 kubične interpolacije susednih piksela po horizontalnoj dimenziji, a nakon toga kubična interpolacija po vertikalnoj dimenziji dobijenih vrednosti iz prethodne faze čiji izlaz predstavlja vrednost piksela izlazne slike.



Ulazna slika



Izlazna slika nakon uvećanja 4x4

## Rotacija slike

Još jedna operacija nad slikama pored promene veličine koja podrazumeva primenu interpolacionih tehnika jeste rotacija slike. Rotacija slike oko piksela sa koordinatama (0, 0) (gornji levi ugao) se na prost način može izvršiti primenom sledeće jednačine:

*X’=X\**cos(θ) +*Y*\*sin*(*θ*)*

*Y’=Y\**cos(θ) -*X*\*sin*(*θ*)*

gde je θ željeni ugao rotacije. Nakon izračunavanja *X’* i *Y’* potrebno je u rezultujuću sliku na koordinate *X* i *Y* upisati vrednosti piksela ulazne slike sa koordinatama *X’* i *Y’.* S obzirom da *X’* i *Y’* najčešće nisu celi brojevi prilikom određivanja vrednosti piksela primenjuje se jedna od tehnika interpolacije.

Rotacija slike oko proizvoljne tačke vrši se upotrebom sledeće jednačine:

*X’ = X \** cos(θ) – *Y \** sin(θ) – m \* cos(θ) + n \* sin(θ) + m

*Y’ = Y \** cos(θ) + *X \** sin(θ) – m \* sin(θ) - n \* cos(θ) + n

gde m i n predstavljaju koordinate tačke oko koje se slika rotira.

Implementacija

Obrada slike se vrši u YUV format, s obzirom da se na ulazu i izlazu očekuje slika u RGB formatu, na početku i na kraju algoritma vrši se konverzija iz jednog u drugi format.

Za svaki izlazni piksel se izračunaju X’ i Y’ koordinate po gore navedenoj formuli.

Nakon toga se vrednost izlaznog piksela računa nekom od metoda interpolacije.

Realizovane su dve varijante rotacije slike, gde se u jednoj vrednost izlaznog piksela računa sample and hold metodom a u drugoj se primenjuje bilinearna interpolacija.



Ulazna slika



Izlazna slika nakon rotacije za 45 stepeni primenom sample and hold interpolacije(levo) i bilinearne interpolacije(desno)

# Zaključak

Složenije metode interpolacije koje vrše interpolaciju sa većim brojem piksela daju znatno bolje rezultate u odnosu na metode koje vrše interpolaciju sa manjim brojem piksela.

Sample and hold metoda uzima najmanje resursa (vremena i memorije) za cenu slabijeg kvaliteta rezultujuće skalirane slike. Rezultat je stepeničasta struktura određenih delova slike.

Sa druge strane bilinearna i bikubična metoda daju bolji kvalitet skalirane slike za cenu povećanja memorijskih i vremenskih zahteva.

Zaključak je da izbor metode interpolacije zavisi od željenog kvaliteta slike sa jedne strane i vremenskih i memorijskih resursa koji su dostupni sa druge strane.