

## Vježba 6.

### Estimacija pokreta u videu

#### Uvod

Vrlo važan dio algoritama za kompresiju videa je estimacija pokreta. Tema ove vježbe je implementacija jedne od najpopularnijih metoda za estimaciju pokreta – uparivanja blokova. Razlog za njenu popularnost je relativno mala složenost i danas se mnogo koristi u H.261, te MPEG-1 i 2 kodecima.

Osnova ideja ovog algoritma je da se vektor pomjeraja piksela na koordinatama  $(n_1, n_2)$  u  $k$ -tom frejmu može odrediti posmatranjem bloka dimenzija  $N_1 \times N_2$  sa centrom u  $(n_1, n_2)$  (referentni blok) i traženjem u  $(k+1)$ -vom frejmu bloka (ciljni blok) koji je najbolje uparen sa referentnim blokom. Razlika između položaja ciljnog i referentnog bloka je traženi vektor pomjeraja. Mogu se koristiti različite mjere uparivanja blokova, a mi ćemo u ovoj vježbi koristiti minimalnu apsolutnu razliku (minimum absolute difference). Apsolutna razlika između blokova definisana je sa:

$$D(d_1, d_2) = \sum_{(n_1, n_2) \in \mathbf{B}} |s(n_1, n_2, k) - s(n_1 + d_1, n_2 + d_2, k + 1)|, \quad (1)$$

gdje je sa  $s(n_1, n_2, k)$  označen intenzitet piksela na koordinatama  $(n_1, n_2)$  u  $k$ -tom frejmu,  $\mathbf{B}$  je posmatrani blok dimenzija  $N_1 \times N_2$ , a  $[d_1, d_2]^T$  je vektor pomjeraja.

Estimacija vektora pomjeraja je data sa:

$$[d_1, d_2]^T = \arg \min_{(d_1, d_2)} D(d_1, d_2). \quad (2)$$

Dakle, za svaki piksel se za svaku vrijednost vektora pomjeraja izračunava apsolutna razlika između blokova prema jednačini (1), a zatim se kao vektor pomjeraja bira onaj vektor za koji je vrijednost apsolutne razlike između blokova minimalna.

Očigledno, ovaj algoritam je izuzetno vremenski zahtjevan i kako bi bila moguća njegova praktična primjena koriste se neke heuristike koje smanjuju vrijeme izvršavanja. Prva heuristika je ograničavanje skupa mogućih vrijednosti vektora pomjeraja na „prozor pretraživanja“:

$$-M_1 \leq d_1 \leq M_1 \quad \text{i} \quad -M_2 \leq d_2 \leq M_2,$$

čiji je centar u pikselu za koji se estimira vektor pomjeraja, a  $M_1$  i  $M_2$  su predefinisani parametri iz skupa cijelih brojeva. Takođe, često se vektor pomjeraja ne estimira u svakom pikselu frejma već u pikselima raspoređenim na rešeci sa fiksnim korakom, tj. estimira se vektor pomjeraja bloka.

#### Napomene

Za vizuelizaciju vektorskog polja korisna je naredba `quiver`.

`quiver(u, v)` crta vektore brzine sa komponentama  $(u, v)$  kao strelice raspoređene na pravilnoj rešeci.

Ishodište koordinatnog sistema na grafiku dobijenom pomoću `quiver` je u donjem lijevom uglu slike. Međutim, kako bi se ovaj grafik mogao uporediti sa polaznim frejmovima videa korisno je ishodište prebaciti u gornji lijevi ugao. Ovo se postiže naredbom `axis ij`.

Matrične i vektorske strukture podataka u MATLAB-u omogućavaju vrlo jasnu implementaciju složenih matematičkih izraza i njihovo efikasno izračunavanje. Na primjer, neka su data dva bloka  $16 \times 16$  piksela predstavljena kao matrice  $A$  i  $B$ . Njihova apsolutna razlika se može izračunati pomoću:

```
D = 0;
for i = 1:16,
    for j = 1:16,
        D = D + abs(A(i,j)-B(i,j));
    end
end
ili pomoću
```

```
D = sum(sum(abs(A - B)));
```

Drugi način koristi MATLAB-ove matrice i vektore i ne zahtijeva upotrebu “for” petlji. Ova reprezentacija je koncizna, čitljiva i laka za debugiranje (u poređenju sa implementacijom koja koristi “for” petlje). Pored toga, matrični pristup je mnogo brži.

## Zadaci

1. Učitati u radni prostor MATLAB-a video klip `mobile.avi`.
2. Izdvojiti frejmove 155 i 160 kao slike. U sledećoj tački će biti estimirano polje 2-D vektora pomjeraja između ova dva frejma. Frejm 155 će biti tekući, a frejm 160 sledeći. Konvertovati frejmove u grayscale slike i u klasu `double`.
3. Implementirati algoritam za estimaciju pokreta uparivanjem blokova tako što ćete tekući frejm izdijeliti na blokove  $16 \times 16$  piksela bez preklapanja i za svaki od tako dobijenih blokova u sledećem frejmu odrediti najbolje uparen blok na osnovu minimalne apsolutne razlike (1). Najbolje uparen blok odredite izračunavanjem apsolutne razlike za sve vektore pomjeraja u prozoru pretraživanja, pri čemu je prozor pretraživanja  $+16$ , odnosno,  $-15$  piksela u horizontalnom i vertikalnom smjeru u odnosu na položaj referentnog bloka. Rezultat ovdje treba da budu dvije matrice u kojima će se nalaziti komponente vektora pomjeraja za svaki od blokova. Kako biste izbjegli granične probleme ne morate određivati vektore pomjeraja za blokove  $16 \times 16$  piksela koji se nalaze na rubu frejma.
4. Grafički prikazati estimirano polje 2-D vektora pomjeraja korištenjem MATLAB naredbe `quiver`.