

Contents

1	Uvod	2
1.1	Općenito o video sekvencama	2
1.1.1	Video fajlovi	2
1.2	Definicija opsega rada	2
1.3	Primjene video interpolacije	2
2	Osnovne tehnike	3
2.1	Duplikacija frejmova	3
2.2	Linearna interpolacija	3
2.3	Područja primjene	3
3	Algoritmi uparivanja blokova	4
3.1	Uvod	4
3.2	Osnovni algoritmi	5
3.3	Adaptive Rood Pattern Search - ARPS	5
3.4	Fazna korelacija	5
4	Uklanjanje grešaka	6
5	Interpolacija frejmova korištenjem optičkog toka	7
6	Paralelno izvršavanje tehnika interpolacije	8
7	Implementacija interpolatora korištenjem OpenCV biblioteke i benchmark testovi	9
8	Zaključak	10

Uvod

1.1 Općenito o video sekvencama

1.1.1 Video fajlovi

1.2 Definicija opsega rada

1.3 Primjene video interpolacije

Osnovne tehnike

2.1 Duplikacija frejmova

2.2 Linearna interpolacija

2.3 Područja primjene

Algoritmi uparivanja blokova

3.1 Uvod

Prva klasa algoritama koje ćemo proučavati kreću od iste osnovne ideje: Podijeliti prvi frejm na blokove, za svaki blok pronaći vektor pomaka iz prvog u drugi frejm, te primijeniti jedan dio tog vektora pomaka na blok. Ako nam je cilj samo kreirati jedan novi frejm između svaka dva postojeća, svaki blok ćemo pomjeriti duž pola izračunatog vektora pomaka. Ako nam je cilj interpolirati dva frejma između dva postojeća, blok ćemo pomjeriti duž jedne trećine vektora pomaka za prvi, i dvije trećine za drugi interpolirani frejm, itd. Cilj slijedećih algoritama jeste uparivanje blokova prvog frejma sa blokom iste veličine u drugom frejmu. Međutim, postoji nekoliko pitanja na koja moramo odgovoriti prije nego što možemo primijeniti ove algoritme:

- Koju veličinu bloka ćemo koristiti?
- Koliki će biti prozor pretrage, odnosno koliko će se svaki blok moći maksimalno pomjeriti između prvog i drugog frejma?
- Koji je kriterij sličnosti dva bloka?
- Kako odrediti uspješnost uparivanja?

U praksi se koriste blokovi veličine 16x16 piksela, te prozor pretrage veličine 30x30 piksela. To znači da pretpostavljamo da se između dva susjedna frejma blokovi neće pomjeriti više od 7 piksela u bilo kojem od 4 kardinalna smjera. To nam daje 225 mogućih lokacija za svaki blok. Naravno, ne postoji definitivna, optimalna veličina bloka ili prozora pretrage za sve slučajeve. Manje blokove je brže uporediti, ali je njihov broj veći, te je veća vjerovatnoća da će dva bloka biti slučajno veoma slična. Veći prozor pretrage

nam omogućava pronalaženje ispravnih vektora pomaka i u slučaju kada se desi pomak veći od 7 piksela, ali značajno povećava vrijeme potrebno za izračunavanje te, slično kao u slučaju blokova, povećanjem prozora pretrage se povećava i vjerojatnoća uparivanja dva bloka koji su slični, ali zapravo ne predstavljaju isti blok. U svim sljedećim algoritmima ćemo koristiti blokove i prozore pretrage navedene veličine.

Sljedeće pitanje se tiče kriterija sličnosti dva bloka. Svaki blok je sastavljen od 256 piksela, koji se sastoje od 3 komponente: crvene, zelene, i plave. Svaka komponenta ima cjelobrojnu jačinu u rasponu od 0 do 255, uključivo. Za upoređivanje blokova se prvo pikseli pretvore u crno-bijele, sa jednom komponentom koja predstavlja jačinu bijele boje piksela. Korišteni kriteriji sličnosti blokova su veoma jednostavni. Jedan je *Mean Absolute Difference (MAD)*, odnosno *srednja apsolutna razlika*. Ova mjera nije ništa drugo nego suma apsolutnih vrijednosti razlika jačina odgovarajućih piksela u blokovima, podijeljena sa veličinom bloka. Drugim riječima, zadana je formulom

$$MAD = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |A_{ij} - B_{ij}|$$

Pri čemu N predstavlja visinu i širinu bloka (u našem slučaju 16), dok A_{ij} i B_{ij} predstavljaju vrijednosti piksela na koordinatama i, j (sa početkom u gornjem lijevom uglu bloka) prvog, odnosno drugog razmatranog bloka.

Druga, veoma slična mjera jeste *Mean Squared Error (MSE)*, odnosno *srednji kvadrat greške*. Umjesto uzimanja apsolutne vrijednosti razlika piksela, ova mjera kvadrira razlike piksela, čime se više kažnjavaju veće razlike. MSE je zadana formulom

$$MSE = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (A_{ij} - B_{ij})^2$$

3.2 Osnovni algoritmi

3.3 Adaptive Rood Pattern Search - ARPS

3.4 Fazna korelacija

Uklanjanje grešaka

Interpolacija frejmova korištenjem optičkog toka

Paralelno izvršavanje tehnika interpolacije

Implementacija interpolatora korištenjem OpenCV biblioteke i benchmark testovi

Zaključak