

Sadržaj

1	Uvod	2
1.1	Video sekvence	2
1.2	Interpolacija u video sekvenci	3
1.3	Primjene video interpolacije	4
2	Osnovne tehnike	5
2.1	Duplikacija frejmova	5
2.2	Linearna interpolacija	6
3	Optički tok	7
4	Popunjavanje rupa	8
5	Implementacija interpolatora korištenjem OpenCV biblioteke i benchmark testovi	9
6	Zaključak	10

Uvod

Cilj ovog rada jeste da objasni osnovne pojmove vezane za video sekvence, te samo neke od brojnih i raznovrsnih algoritama koji se u praksi koriste za video interpolaciju. Objasnit ćemo šta je to tačno video sekvenca, optički tok, interpolacija, framerate, i još mnogo toga. Algoritmi video interpolacije su veoma specijalizovani, te imaju relativno usko područje primjene. Međutim, ipak su izuzetno korisni, te ćemo se sa konkretnim područjima primjene uskoro upoznati. Prije toga bi ipak bilo korisno napraviti kratak uvod u video sekvence, te u to kako ih računari vide.

1.1 Video sekvence

Ljudi svijet oko sebe vide kao sliku koja se kontinuirano mijenja. To jeste, svaki vremenski trenutak je različit od prošlog, koliko god da je blizu. Ljudi i objekti se kreću "glatko", bez potpuno naglih pokreta. Također, slike koje vidimo su prostorno kontinualne, prividno beskonačno detaljne. Računari, s druge strane, moraju slike i video fajlove čuvati u ograničenoj količini memorije, što znači da mora doći do kompromisa. Veća količina detalja zahtijeva više memorije za čuvanje. Svaka slika ima samo konačnu količinu detalja, što znači da se sastoji od konačnog broja piksela. Piksela (od eng. *picture element*) je najmanji djelić slike. Dva najčešća tipa piksela, pri čemu ćemo mi u ovom radu koristiti oba, su 1-komponentni i 3-komponentni (nekada se umjesto termina komponenta koristi termin kanal). Jednokomponentni piksel je djelić slike koji je jednobojan. 1-komponentni pikseli su gotovo isključivo monohromatski, tj. njihova boja može biti crna, bijela, ili neka nijansa sive. 3-komponentni pikseli, s druge strane, se sastoje od 3 jednobojne komponente, koje zovemo podpiksela ili subpiksela. Ako je piksel oblika kvadrata, onda su podpiksela oblika pravougaonika (koji su po pravilu izduženi vertikalno). Na svim modernim ekranima, boje podpiksela su crvena, zelena i

plava. Ovaj raspored ćemo ubuduće zvati RGB (od eng. *Red-Green-Blue*). Drugi tipovi podpiksela također postoje, koji se moraju konvertovati u RGB za prikaz na ekranima. Svaki ekran i svaka slika se sastoje od određenog broja piksela, koji su po pravilu raspoređeni u matricu. Ako na monitoru gledamo sliku koja ima više horizontalnih ili vertikalnih piksela od našeg monitora, onda ne možemo vidjeti sve detalje te slike. Koji detalji će biti izostavljeni zavisi od konkretnog algoritma korištenog za skaliranje slike, o kojima nećemo pričati u ovom radu.

Objasnili smo kako računari rješavaju problem prostorno kontinualnih slika, sada ćemo objasniti video sekvence. Isto kao što smo sliku rastavili u piksele, video ćemo rastaviti u *frejmove*. Frejmovi su pojedinačne sličice (ili okvirovi) od kojih se sastoji video sekvenca (sekvenca upravo zato što je to sekvenca frejmova). Svaki frejm provede određeno vrijeme na ekranu, nakon čega se mijenja narednim frejmom. U pravilu, to vrijeme je za svaki frejm isto (ako je to moguće), da bi video sekvenca koju gledamo izgledala što prirodnije. Broj frejmova koje vidimo svake sekunde (odnosno broj recipročan broju sekundi koji svaki frejm provede na ekranu) se zove *framerate*, a korištena jedinica je *frame per second*, skraćeno *fps*. Tako je maksimalni framerate većine ekrana 60 fps, framerate filmova je gotovo uvijek 24 fps, framerate igrice zavisi od mnogo faktora, ali je u većini slučajeva 30 ili 60 fps.

1.2 Interpolacija u video sekvenci

Sada kada znamo šta je framerate, možemo objasniti video interpolaciju. Naime, interpolacijom video sekvence se dobivaju novi frejmovi između postojećih. Taj novi frejm će biti veoma sličan svojim susjednim, jer u suštini predstavlja "među korak". Izlaz algoritma za video interpolaciju će biti video veoma sličan početnom, sa dodatnim frejmovima koji su generisani iz originalnih. U matematičkom smislu, interpolacija predstavlja izračunavanje novih vrijednosti na osnovu okolnih poznatih, što zvuči veoma slično video interpolaciji. Međutim, zbog svoje prirode, algoritmi interpolacije video sekvenci se potpuno razlikuju od onih korištenih u matematici.

1.3 Primjene video interpolacije

Kao što smo već rekli, izlaz algoritma za video interpolaciju jeste novi video koji je veoma sličan originalnom. Ako želimo, možemo koristiti taj video, koji će zbog svog višeg framerate-a izgledati prirodnije (pod uslovnom da je kvalitetno interpoliran) i glađe. Tako bi sportski sadržaj mogao biti sniman kamerom visoke rezolucije, ali relativno niskog framerate-a, nakon čega bi frejmovi mogli biti interpolirani da bismo dobili video koji je i visoke kvalitete i visokog framerate-a. Također bismo mogli koristiti video interpolaciju za video pozive, gdje je mrežna propusnost često ograničavajući faktor. Možemo snimiti video niskog framerate-a, koji ne zahtijeva visoku mrežnu propusnost, te interpolirati dolazeći video. Naravno, u oba ova slučaja će dobiveni rezultat biti nižeg kvaliteta nego video sniman visokim framerate-om, ali uz kvalitetne algoritme razlika neće biti značajna.

Postoje već projekti koji koriste video interpolaciju za kreiranje *slow-motion* video sekvenci. Umjesto zadržavanje ukupne dužine video sekvencei povećavanja framerate-a, možemo povećati dužinu i zadržati framerate. Jedan primjer bi bio [Butterflow](#).

Osnovne tehnike

Sada ćemo objasniti "trivijalne" algoritme interpolacije: duplikacija frejmova i linearna interpolacija. Kao što ćemo vidjeti, linearna interpolacija bi se čak mogla koristiti samostalno za kreiranje interpoliranih frejmova. Međutim, rezultat bi bio izuzetno mutan. U narednim odjeljcima ćemo objasniti načine rada i područja primjene ovih algoritama.

2.1 Duplikacija frejmova

Kao što samo ime kaže, duplicirani frejm je identičan jednom od njegovih susjednih frejmova (prethodnom ili narednom). Duplicirani frejmovi se koriste u situacijama kada nema smisla koristiti standardne tehnike interpolacije. Naime, neki frejmovi se jednostavno potpuno razlikuju od svojih prethodnika. To se dešava pri promjeni scene ili bilo kakvoj nagloj promjeni slike. Frejm koji se značajno razlikuje od svog prethodnika nazivamo *keyframe*. Ako ćemo između svaka 2 susjedna frejma ubacivati određeni broj interpoliranih frejmova, onda moramo ubaciti nešto i prije keyframe-a. Jednostavno ćemo napraviti onoliko kopija posljednjeg frejma prethodne scene ili novog keyframe-a koliko nam treba interpoliranih frejmova. Ako pokušamo ipak interpolirati novi frejm između dva potpuno različita, dobit ćemo nešto što neće ličiti ni na jedan od dva frejma, te će biti veoma uočljivo gledaocima. Zbog toga je najbolje samo napraviti kopije. Alternativa tom pristupu bi bilo jednostavno preskočiti te frejmove i ostaviti ih kao susjedne. Nedostatak ovog pristupa je potencijalno stvaranje problema oko sinhronizacije ako uz video sekvencu imamo prateći audio zapis ili tekst prijevoda. Ta "rupa" u frejmovima bi značila da će audio zapis i prijevod kasniti za video zapisom. Tako da ćemo mi ipak koristiti duplikaciju prethodnog frejma.

2.2 Linearna interpolacija

Kao što ćemo vidjeti kasnije, tehnike interpolacije koje ćemo mi koristiti neće kreirati cjelokupan frejm. Postojat će velika područja frejma za koje algoritam jednostavno nije dao nikakve podatke. Da bismo izbjegli veoma očigledna područja crnih piksela, moramo koristiti neku drugu tehniku koja će zagarantovano davati podatke o svakom pikselu. A najjednostavnija takva tehnika (osim duplikacije) jeste linearna interpolacija, koja radi na sljedećem principu:

Neka su A i B vektori kolone koji predstavljaju dva piksela. Piksela A je piksel na nekoj poziciji prethodnog, a piksel B je piksel na toj istoj poziciji narednog frejma. Mi sada želimo da vidimo koji će se piksel nalaziti na toj poziciji u interpoliranom frejmu. Realan broj α ($0 < \alpha < 1$) predstavlja relativnu poziciju interpoliranog frejma između prethodnog i narednog. To jeste, ako između ta dva frejma generišemo n novih, a frejm koji želimo generisati je k -ti po redu, tada je:

$$\alpha = \frac{k}{n + 1}$$

Sada je vrijednost piksela C interpoliranog frejma:

$$C = \frac{(1 - \alpha) * A + \alpha * B}{2}$$

Optički tok

Popunjavanje rupa

Implementacija interpolatora korištenjem OpenCV biblioteke i benchmark testovi

Zaključak