Hibridne slike i implementacija brze Furijeove transformacije

Nikola Dimitrijević

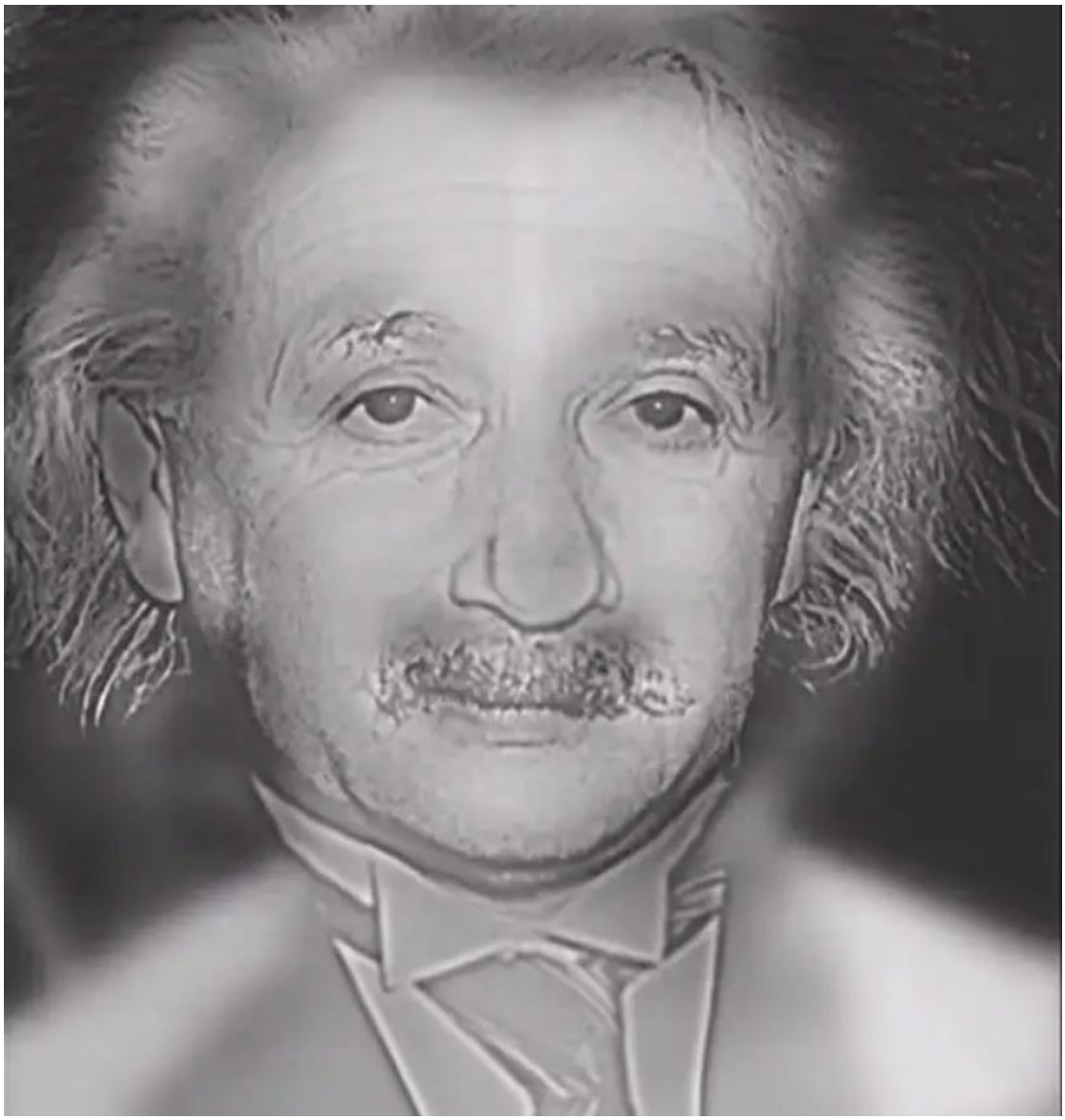
Naučno izračunavanje

Univerzitet u Beogradu, Matematički fakultet

23.06.2019.

Uvod

Spajanjem dve slike na određen način se može dobiti interesantan efekat. Na primeru slike ispod, gledanjem iz blizine se prvo primećuje lik Alberta Ajnštajna. Međutim, ako se posmatrač udalji ili gleda kroz skoro zatvorene oči, onda se može videti lik Merilin Monro. U nastavku teksta će biti opisan način postizanja ovakvog rezultata korišćenjem Furijeovih transformacija, implementacija brze Furijeove transofrmacije, kao i upoređivanje sa postojećom implementacijom unutar nampaj (eng. numpy) biblioteke.



Opis efekta

Ovaj efekat je poznat pod nazivom hibridne slike. Mogu se kombinovati bilo koje dve slike, mada je najbolji efekat kada jedna slika ima više oštrih ivica, dok druga slika ima blaže ivice.

Iluzija je rezultat toga što ljudi ne vide dobro detalje ako su dalje od posmatranog objekta, ili ako im je u perifernom vidu. Tada krupnija svojstva bivaju lakše uočljiva, što omogućava ovakvo ponašanje.

Suština načina generisanja hibridnih slika je da se iz jedne slike uklone niske frekvencije, dok se iz druge uklone visoke frekvencije. Potom se takve slike spoje u jednu sa pogodnim koeficijentima odnosa intenziteta polaznih slika u rezultujućoj. Na taj način su zadržane ivice (odnosno visoke frekvencije) iz jedne slike, a veće komponente (niske frekvencije) su zadržane iz druge slike.

Najpopularniji primer ovog fenomena je spoj Alberta Ajnštajna i Merilin Monro. Umetnici su ovu tehniku koristili još davno. Tako Mona Liza, koju je naslikao Leonardo da Vinči, izgleda kao da se smeška dok se gleda oko nje, ali kada se usmeri pogled u njen osmeh, njega nema.

Iz blizine se vide tužni izrazi lica, a iz daljine redom: srećno, iznenađeno, i dva srećna lica

Iz blizine se vidi “southeast”, a iz daljine “northwest”

Implementacija

Projekat je implementiran korišćenjem programskog jezika Pajton. Korišćene su funkcije iz biblioteke nampaj za Furijeove transformacije slika, a takođe su i implementirane u istom projektu radi upoređivanja performansi.

Funkcija hybridImage ima za argumente matrice dveju slika, dve sigme koje regulišu koliko će se odbaciti ili zadržati frekvencije, parametar odnosa intenziteta slika u rezultatu, kao i da li se koristi bibliotečka implementacija Furijeove transformacije ili implementacija unutar ovog projekta.

Potom se generiše Gausovo zvono čiji centar je u centru pomerene diskretne Furijeove transformacije. Kada se zadržavaju niže frekvencije onda zvono ima najveće vrednosti u centru, a kada se zadržavaju više frekvencije onda su u centru najniže vrednosti. Umesto Gausovog zvona mogao se jednostavno iseći krug ili kvadrat iz frekvencijskog spektra, ali to dovodi do lošijeg kvaliteta zamućivanja. Potom se na slike prijenjuje dvodimenzionalna Furijeova transformacija. Slike reprezentovane u frekvencijskom domenu se pokoordinatno množe matricama Gausovog zvona. Konačno, slike se inverznom Furijeovom transformaciju vraćaju iz frekvencijskog u prostorni domen i spajaju se u jednu sliku množenjem sa koeficijentom odnosa slika i sabiranjem.

Formula za Gausovu funkciju u jednoj dimenziji je

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

Dvodimenzionalna verzija je data sledećom formulom

$$G(x,\ y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

Implementacija brze Furijeove transformacije

Dvodimenzionalna Furijeova transformacija se može izračunati preko jednodimenzionalne tako što se ona primeni na svaki red slike, pa na rezultat toga se primeni još jednom, ali ovaj put na svaku kolonu. Ispostavlja se da je ovakvo korišćenje fft iz nampaj biblioteke brže nego fft2, mada samo za veličine ulaza manje od 10000.

Naivna implementacija diskretne Furijeove transformacije ima kvadratnu složenost. Za ulaz veličine 3000 je vreme izvršavanja 600 milisekundi, dok je za ulaz veličine 4000 vreme izvršavanja 1100 milisekundi. Vidi se da ovakvo rešenje ne skalira dobro za veće slike, pošto već slike dimenzija 100x100 piksela predstavljaju ulaz veličine 10000.

Nampaj biblioteka, iako se koristi u Pajtonu, ima implementirane metode brze Furijeove transformacije u jeziku C (zapravo to je preveden originalni kod FFTPACK, koji je implementiran u Fortranu). Takva implementacija ima složenost O(n logn), a pritom kompajliran C program ima mnogo povoljniji konstantan faktor izvršavanja običnih naredbi nego što je to slučaj u jeziku Pajton.

Prvo bi se pomislilo da je to zbog toga što je Pajton interpretiran jezik, ali to neznatno utiče na ovakve zadatke. Zapravo je razlog u tome što je Pajton dinamički tipiziran jezik, pa za svaku, naizgled jednostavnu operaciju, ima mnogo dodatnog posla. To je upravo motivacija zašto se mnoge metode implementiraju u drugim jezicima, iako se koriste u Pajtonu. Tako i nampaj biblioteka pruža mnogo brže operacije nad nizovima, jer zahteva da svi elementi njegovih nizova ili matrica budu istog tipa. Još jedan razlog je što su nampaj nizovi strukture koje gusto pakuju svoje elemente, dok Pajton nizovi u stvari čuvaju pokazivače na svoje elemente, koji se mogu nalaziti u različitim delovima memorije. Na taj način nampaj nizovi već dobijaju ubrzanje koje je često i reda veličine, a kontinualno smeštanje elemenata omogućava vektorske operacije procesora, tako da se u jednoj instrukciji može izvršiti operacije nad više elemenata nizova.

I pored navedenih mana Pajtona koje utiču na performanse, teško je opravdati loše vreme izvršavanja prethodne implementacije. Glavni problem je kvadratan broj operacija.

Kuli-Tuki algoritam (eng. Cooley-Tooky) je najčešće korišćen algoritam za izračunavanje brze Furijeove transformacije. Ima formu podeli pa vladaj algoritma, pa i takvu složenost, tj. O(n logn). Deli niz na parne i neparne elemente, koje potom rekurzivno obradi, i na kraju ih spoji i sabere sa kompleksnim koeficijentima. Kada se rekurzivno dostigne do dovoljno malog niza, onda se na njega primeni prethodna implementacija, čija loša vremenska složenost ne dolazi do izražaja pošto se primenjuje na nizove male fiksne dužine. Implementacija opisanog algoritma je svakako i dalje sporija od nampajeve verzije, ali za red veličine brža od prethodne kada se meri na nizu dužine 4096. Moguće je menjati parametar od koga zavisi kada algoritam prestaje da deli niz rekurzivno i primenjuje običnu funkciju nad njim. Rezultati su prikazani u tabeli:

	N=64	N=32	N=16	N=8
Milisekundi	17	10	9.5	12.2

Izgradnjom dvodimenzionalne brze Furijeove transformacije od ovakvog algoritma se dobija funkcija koja za 1300 milisekundi izračuna diskretnu Furijeovu transformaciju slike dimenzija 512x512 piksela. To je i dalje sporije od implementacije u programskom jeziku C, ali primenljivo.

Problem sa rekurzivnom Kuli-Tuki implementacijom je to što se u svakom listu rekurzije ponovo konstruiše niz koeficijenata sa kojima se množe elementi u tom listu. Bilo bi mnogo brže kada bi se taj niz samo jednom konstruisao, i još bolje kada bi se sva ta množenja mogla vektorizovati. Zaista i može - funkcija myfft_best je nerekurzivna i paralelizovana implementacija Kuli-Tuki algoritma računanja diskretne Furijeove transformacije.

Dvodimenzionalna Furijeova transformacija koja koristi myfft_best traje 306 milisekundi kada se primeni na sliku dimenzija 512x512, napsram 1300 milisekundi kod prethodne. Nampaj imeplementacija je svakako i dalje pobednik uz vreme izvršavanja od samo 12 milisekundi.

Potrebno je još implementirati inverznu Furijeovu transformaciju, koja će iz frekvencijskog domena vratiti sliku u prostorni domen. Kada postoji imeplementacija Furijeove transformacije, onda je inverziju vrlo lako dobiti. Postoji više načina za to, od kojih je u ovoj implementaciji izabran sledeći. Frekvencijska reprezentacija se konjuguje, na nju se primeni Furijeova transformacija, pa se potom opet konjuguje.

Vršeno je merenje vremena izvršavanja celokupne implementacije. U slučaju korišćenja nampaj fft funkcija, generisanje hibridne slike traje 620 milisekundi, dok korišćenjem implementacije brze Furijeove transformacije u Pajtonu traje 4000 milisekundi.

Sva merenja su vršena korišćenjem timeit magične naredbe u Jupiter svesci. Ona pruža precizne rezultate pošto izvršava svoj argument više puta i pritom daje informaciju o standardnoj devijaciji.

Numpaj metoda allclose se koristi za proveru rezultata imeplementacije, tako što se uporedi svaki element rezultata sa rezultatom bibliotečkih funkcija.

Na samom kraju Jupiter sveske dodat je interaktivan element, tako da se može regulisati odnos intenziteta slika u rezultatu.

Zaključak

Pravljenje hibridnih slika predstavlja interesantnu primenu Furijeove transformacije i ukazuje na specifična svojstva ljudske percepcije. Može se koristiti za pravljenje teksta koji je čitljiv samo na blizinu, prikaz promene scene u vremenu ukrštanjem dve verzije, pravljenje tekstura koje se ne vide na nekoj udaljenosti i promena lica.

Literatura

Hybrid images - Aude Oliva, Antonio Torralba, Philippe. G. Schyns 2006.

Understanding the FFT https://jakevdp.github.io/blog/2013/08/28/understanding-the-fft/

Four Ways to Compute an Inverse FFT Using the Forward FFT Algorithm
https://www.dsprelated.com/showarticle/800.php