

Projektna naloga pri predmetu Računalniški vid v praksi

Kristjan Voje

30/06/2016

1 Uvod

1.1 Predznanje

Obiskujem 3. letnik univerzitetnega programa računalništvo in informatika. Izbrane imam vse tri predmete iz modula Umetna inteligenca: Inteligentni sistemi (1. semester), Umetno zaznavanje (1. semester), Razvoj inteligentnih sistemov (2. semester). Opravljen imam tudi predmet Osnove umetne inteligence.

1.2 Cilji

Rad bi razvil program za zaznavanje gest in gibov z uporabo navadne RGB kamere. Uporabljal bi USB spletno kamero ter vgrajeno spletno kamero. Moj cilj je napraviti program čim bolj robusten in odporen na spremembe okolja, osvetlitve, itd. Program bom uporabil kot alternativni krmilnik za video igre.

2 Potek dela

Projekt je sestavljen iz večih podnalog, zato je temu primerna tudi zgradba. Poskusil sem čim bolj razčleniti podnaloge, tako da vsako opravlja drug razred:

- *detector.cpp*

Ta datoteka vsebuje razred main. Skrbi za osnovne funkcije, kot so zajem videa iz kamere ali datoteke, predvajanje videa, klicanje drugih

objektov za obdelavo videa ter shranjevanje videa. V zanki bere slike, z uporabo Haar Cascade zaznava pozicijo obraza, ter generira binarno matriko pikslov, zaznanih kot koža.

Za lažje spreminjanje nekaterih parametrov, sem ustvaril *detector.conf*. Ta datoteka ima zgradbo: ime_parametra vrednost. Program ob branju *detector.conf* preskoči prvo besedo in prebere vrednost parametra. Vrstni red parametrov je pomemben.

- *color_extractor.cpp*

Razred ColorExtractor prejme barvno sliko (frame), ter iz lokacije obraza razbere barvo kože. Barvo sem iz BGR prostora projeciral v bg prostor tako, da sem vsak piksel normaliziral, ter uporabil le komponenti b in g. Razred vsebuje matriko bg, kamor se shranjujejo glasovi za barvo obraza. Ko je določen prag dosežen, z drsnim oknom poišče najbolj gost del matrike. Središče drsnega okna shrani kot barvni piksel, ki predstavlja kožo.

- *hand_tracker.cpp*

Razred HandTracker sledi poziciji dlani. Za to uporablja binarno kožno matriko ter pozicijo glave. S pozicijo glave razpolovimo iskalni prostor, saj lahko iščemo levo roko levo od obraza in desno roko desno od obraza. Primer iskanja leve roke: iskanje začne spodaj levo. Poišče najbolj levi in najnižji piksel.

Da bi povečal robustnost, posodabljam pozicijo dlani uteženo, z upoštevanjem pozicije v prejšnjem framu. Tako sem zmanjšal skakanje pozicije zaradi šuma.

- *gesture_tracker.cpp*

Razred GestureTracker prejme pozicijo glave ter poziciji leve in desne dlani. Na podlagi teh treh podatkov ugotovi v katerem stanju (pozi) se nahaja uporabnik, ter izvrši potreben ukaz.

Okvir sem razdelil na 3 horizontalne pasove. Prvi se nahaja nad glavo, drugi je pod prvim in zavzema 1/3 višine okvirja, tretji pa je pod drugim. Stanja so določena glede na to, v katerih pasovih se nahajata leva in desna dlan.

GestureTracker opazuje tudi pozicijo glave. Če se višina glave spreminja preko določenega praga, je to prepoznano kot tek. Preden se sproži stanje tek ali ponovno stanje mirovanja, moramo preseči določen prag. Privzeto mora program v 5 zaporednih okvirjih prepoznati dovolj velik premik višine glave, da sproži tek.

GestureTracker poleg zaznavanja ukazov te tudi izvrši. C++ ima na sistemu Windows knjižnico za simulacijo pritiskov tipk. Na Linux podobne rešitve nisem našel, zato sem napisal bash skripto, kateri GestureTracker pošlje tipko kot argument.

- *key_press.sh*

Ta skripta prejema ukaze in simulira pritise tipk miške in tipkovnice. V skripti lahko izberemo, kateremu oknu pošiljamo ukaze. Če nastavimo `simulation=true`, bo program ukaze tipkal v terminal.

3 Rezultati

Uspel sem napisati program, ki uporabniku omogoča preprosto interakcijo z računalnikom preko RGB kamere. Program sem preizkusil z igranjem video igre. Opazil sem, da je zelo pomembna frontalna osvetlitev uporabnika, saj le tako lahko program zazna pravilne barve kože. Dobro je, da je ozadje temnejše od uporabnika. Predmeti v ozadju, ki so bili podobne barve kot koža (lesena omara), so povzročali napačno detekcijo. To sem lahko odpravil z odstranitvijo ozadja.

Interakcija z računalnikom je delovala z zakasnitvijo. Gre za zamik, ki ga detektor potrebuje, da robustno zazna gib. Kljub zamiku je bilo mogoče pošiljati igri ukaze dovolj hitro, da to ni omejevalo igralne izkušnje.

Posnel sem primer uporabe programa: youtube video.

4 Zaključek

Projektna naloga je bila med najzanimivejšimi nalogami na faksu. Naučil sem se osnove programiranja v c++ in navadil sem se na uporabo knjižnice OpenCV. Prav tako sem pridobil podlago za oblikovanje lepe kode. Dobil sem tudi izkušnje iz modularnega razvoja aplikacije, ter testiranja posameznih komponent.