

POROČILO SEMINARSKE NALOGE

Robotika in računalniško zaznavanje

63070327 – Marko Oman

Prepoznavanje objektov

Moj algoritem za prepoznavanje posameznih regij deluje tako, da najprej RGB sliko spremeni v HSV sliko in iz te dobljene slike ven poišče regije. Pretvorba v HSV in nekaj pogojev pri dobljeni sliki deluje kot precej učinkovit »threshold«, tako za sive objekte, kot tudi za vse ostale. Po dobljenih objektih se znebim šuma in manjših regij s širjenjem in krčenjem. Potem dobim slike posameznih objektov iz prve slike s pomočjo vgrajene funkcije »bwlabel«. Ko dobim posamezne slike vseh objektov, na vsaki sliki poiščem določene lastnosti in iz tega ugotovim za kakšne vrste objekta gre. Pri tem vedno pogledam, če je regija slučajno zelo majhna, in če je, jo ne obravnavam. Najprej sem s pomočjo vgrajene funkcije »ind2sub« ugotovil pozicijo objekta. Za barvo regije sem uporabil originalno sliko in jo pomnožil s sliko te regije, da dobim barvo samo v tem območju, kjer se nahaja regija. Potem izračunam vrednosti treh osnovnih barv s pomočjo povprečja. Glede na lastnosti barv določim barvo objekta. Siva barva je, če ima vse tri vrednosti(RGB) približno enake. Za rumeno barvo sem določil nek interval vrednosti po posameznih kanalih. Če pa ni niti siva, niti rumena, potem pa pogledam katera vrednost je največja in ta barva je.

Za obliko »drugo« sem preprosto pogledal površino objekta in, če je površina večja od določene vrednosti je to »drug« objekt. »Drug« objekt je tudi, če je prej določena barva siva. Za pravokotno obliko sem pogledal okroglost regije in, če ima okroglost med 0.3 in 0.6 potem je to pravokotnik. Za okroglo in kvadratno obliko nisem mogel uporabiti okroglosti, ker je preveč podobno in je prevečkrat zamešal kakšne oblike je objekt. Ločitev kroga in kvadrata sem rešil tako, da sem preprosto pogledal razliko med površino samega objekta in površino obsegajočega pravokotnika tega objekta. To je delovalo v večini primerov, vendar če je bil kvadratne oblike objekt obrnjen postrani je še vedno lahko zamešal za krog. Nato sem prej dobljene rezultate s površino združil z lastnostjo, da ima krog površino približno 1.29-krat manjšo od svojega obsegajočega pravokotnika. S preizkušanjem sem postavil določene meje v razliki površin in tako se algoritem zelo malokrat zmoti.

Opis ostalih funkcij:

-dobiObris:Dobimo sliko obrisa podane slike.

-dobiChain: Iz slike dobi približno chain kodo. Približno zato, ker ni narejena za točno chain kodo, vendar samo za to, da ugotovimo ali gre za manhattansko ali evklidsko razdaljo.

-dobiObseg: S pomočjo chain kode izračuna in vrne obseg lika na sliki.

-dobiPovrsino: Izračuna in vrne površino lika na sliki. Preprosto samo prešteje vse piksele na sliki, ki niso ozadje.

-dobiOkroglost: S pomočjo prej naštetih funkcij izračuna in vrne okroglost objekta na podani sliki.

Kazanje robotske roke

Za ugotavljanje na kateri objekt kaže robotska roka sem najprej pripravil vse potrebne matrike za kasnejšo množenje. Predvideval sem, da robotska roka stoji v zgornjem levem kotu. Matriko(0;0;0;1) sem zmnožil z matrikami za translacijo in rotacijo in tako dobil pozicijo kam kaže robotska roka. Nato sem samo še pregledal vse pozicije regij in, če je pozicija roke znotraj obsegajočega pravokotnika regije, robot kaže na to regijo.