

Robottiohjelmoinnin harjoitustyö

Ristinollarobotti

Aaro Salosensaari

aaro.salosensaari@cs.helsinki.fi

Tietojenkäsittelytieteen laitos
Helsingin Yliopisto

Helsinki, 10. tammikuuta 2015

Sisältö

1	Ristinollarobotin kuvaus	2
2	Robotin rakenne	2
2.1	Materiaalit ja tarvikkeet	2
2.2	Rakennusohje	2
3	Ohjelmakoodi	2
3.1	Penbot	2
3.2	BotGame	2
3.2.1	Kuvantunnistuksen toimintaidea	2
4	Testaus	3
5	Rajoitukset ja tulevaisuus	3
5.1	Toteuttamatta jääneet ominaisuudet	3
5.2	Muita puutteita ja rajoitteita:	4
6	Käyttöohjeet	4
6.1	Ohjelmistojen kääntäminen	4
6.1.1	Penbot	4
6.1.2	BotGame	4
6.2	Pelaaminen	4

Tämä on joulun 2014 robottikurssin loppuraportti. Luettavin versio on [pdf](#).

1 Ristinollarobotin kuvaus

Ristinollarobotti on ristinollaa web-kameran avulla pelaava Lego Mindstorms -robotti.

Robotti koostuu varsinaisesta piirtorobotista (Penbot) ja erillisestä tietokoneella (= kannettava tietokone) ajettavasta varsinaisesta peliohjelmasta (BotGame). Ohjelmointikieli on Java ([LeJOS](#)), ja kuvantunnistukseen käytetään [OpenCV](#):n Java-API:ia.

Peliohjelma osaa tunnistaa ulkoisen web-kameran avulla paperille piirretyn pelilaudan ja pelaajien (robotti ja sen vastustaja) sille piirtämät merkit, ja tämän avulla pelata ristinollaa ihmisvastustajaa vastaan. Peliohjelma lähettää tekoälyn valitsemat siirrot Bluetoothin yli piirtorobotille, jolla on valmiit rutiinit ruksin piirtämiseksi kuhunkin ruutuun.

2 Robotin rakenne

2.1 Materiaalit ja tarvikkeet

2.2 Rakennusohje

3 Ohjelmakoodi

3.1 Penbot

3.2 BotGame

3.2.1 Kuvantunnistuksen toimintaidea

Kuvantunnistusmenetelmän pääinspiraationa oli [AI Shackin Sudoku-lukija](#), jota tosin on sovellettu varsin paljon. Menetelmän idea on yleisellä tasolla seuraava:

1. Ensin etsitään peruskuva, johon mahdollisia muutoksia verrataan:
2. Muunnetaan kuva harmaasävykuvaksi.
3. Ruutupaperin ruutujen häivyttämiseksi sumennetaan kuvaa Gauss-sumennoksella, jonka jälkeen tehdään harmaasävykuvasta mustavalkoinen muuttamalla ([adaptive threshold](#)), jolloin kuvaan jää jäljelle vain pääasiassa merkitseviä viivoja ja merkkejä. Tämän 'binäärikuvan' värit käännetään jatkoa varten.
4. Aiemmassa vaiheessa jotkut tärkeätkin ruudukon viivat saattavat 'katketa', joten niitä yritetään palauttaa morfologisella sulkemisella ([morphological closing](#)).
5. Näin käsitellystä kuvasta etsitään [Hough-muunnoksella](#) kaikki viivat.

6. Koska OpenCV:n Hough-rutiini löytää sellaisilla parametreilla joilla varmasti saadaan kaikki *tärkeät* viivat myös *paljon* viivoja jokaista pelilaudan oikeaa viivaa kohti, lähellä toisiaan olevien viivojen parvet yhdistetään yhdeksi viivaksi per parvi (keskiarvo).
7. Yhdistetyistä viivoista etsitään äärimmäiset (tietyn marginaalin puitteissa) vaaka- ja pystyviivat, jotka vastaavat pelilaudan reunoja. Näiden leikkauspisteet (= peliruudukon nurkat) lasketaan.
8. Leikkauspisteiden avulla kuvan perspektiivi korjataan ja se jaetaan 3x3 -ruudukoksi. Kunkin ruudun reunat (jotka sisältävät piirretyn ruudukon viivat) 'leikataan pois' ja (alkutilanteessa tyhjä) sisäalue ('solu') talletetaan.
9. Jokaiselle verrattavalle kuvalle tehdään sama prosessi, ja kuvien vastaavia soluja verrataan toisiinsa. Mikäli jonkin solun histogrammeissa peruskuvan ja verrattavan välillä on suuri ero, todetaan että tähän ruutuun on verrattavassa kuvassa piirretty uusi merkki.

4 Testaus

5 Rajoitukset ja tulevaisuus

5.1 Toteuttamatta jääneet ominaisuudet

Robotti jäi kahdelta osin hieman keskeneräiseksi:

Ensinnäkin, peli ei osaa pelata ristinollaa täysin itsenäisesti, sillä BotGame:n hahmontunnistuskoodi ei osaa hylätä sellaisia webkameran kuvia, joissa pelilaudan ja kameran välissä on este (esimerkiksi ihmispelaajan käsi tai piirtorobotti itse). Ohjelmasta puuttuu myös botin piirtämien merkkien tunnistus kameran kuvasta AI:n omiksi siirroiksi. Tämän vuoksi peli pyytää käyttäjältä vahvistuksen jokaiselle pelilaudalla havaitulle muutokselle esittääkö se botin tai pelaajan tekemää siirtoa.

Toiseksi varsinainen tekoäly puuttuu. Robotti pelaa ristinollaa sääntöjen mukaan, muttei erityisen älykkäästi.

Molemmat ongelmat olisi ollut tarkoitus ratkaista: Ristinollatekoälyn toteutus jonkinlaisella minimax-algoritmilla tai alpha-beta -karsinnalla olisi melko triviaali tehtävä. Kameran eteen tulleen esteen kaltaiset huomattavat muutokset kuvassa puolestaan olisi (ainakin teoriassa) yksinkertaista havaita OpenCV:n avulla.

Esimerkiksi eräs vaihtoehto tähän olisi tarkastella värikuvan histogrammin poikkeamia (pelilautaa esittävään kuvaan nähden) kun laudan päällä on robotin tai käden kaltainen 'ylimääräinen' esine. Oletettavasti histogrammissa nähtäisiin suuri poikkeama verrattuna tilanteeseen, jossa ainoa muutos on peliruutuun ilmestynyt pieni merkki.

Botti-tekoälyn tekemät siirrot vuorostaan olisi luultavasti mahdollista tunnistaa (ja ohittaa kysymättä pelaajan vahvistusta) hieman ohjelmakoodia laajentamalla.

5.2 Muita puutteita ja rajoitteita:

Piirtorobotti ei osaa asemoida itseään pelilautaan nähden. Käyttäjän on sijoitettava robotti ennaltavalittuun asentoon peliruudukkoon nähden (pelilaudan lävistäjän kautta kulkevalle suoralle). Mikäli robotti on hieman vinossa, se myös ajaa hieman sivuun ja pahimmassa tapauksessa piirtää ristejä väärin paikkoihin. Ongelmaa voisi hieman helpottaa laajentamalla nykyistä toiminnallisuutta pienellä kalibrointis-kriptillä (robotti kulkisi edestakaisin ja piirtäisi pisteitä sinne missä se kuvittelee esim. peliruudukon nurkkien olevan; käyttäjä voisi korjata robotin asentoa).

Piirtorobotti nykyisessä muodossaan piirtää kulkemalla edestakaisin ja pyörimällä moottoriakselinsa keskipisteen suhteen; toisin sanoen vaakasuuntainen viiva on kaareva. Tämän vuoksi mahdolliset kuviot käytännössä ovat ruksien ja pisteiden kaltaisia yksinkertaisia kuvioita, joita tämä rajoitus ei haittaa. Ristinollan perinteisen 'nolla'-kuvion piirtäminen olisi nykyisellä rakenteella melko vaikeaa.

Kuvan analysointimenetelmät ovat teoriatasolla yleistettävissä, mutta koodissa käytetyt vakiot, raja-arvot, jne. on löydetty käsin kokeilemalla tietynlaisella kamerakokoonpanolla. Esimerkiksi huomasiin että tutkittavien kuvien resoluution vaihtaminen voi rikkoa nykyisen toiminnallisuuden.

6 Käyttöohjeet

6.1 Ohjelmistojen kääntäminen

6.1.1 Penbot

6.1.2 BotGame

6.2 Pelaaminen