

Huom: Tämän kerran laskuharjoitukset suoritetaan osin pajassa. Katso ohjeet kurssin sivulta.

Tehtävä 1. Hahmontunnistus (2 pistettä)

Lataa kurssin kotisivulta paketti `surf-package.zip`, pura se, siirry syntyneeseen hakemistoon ja suorita komento `ant example`. (Toimii ainakin laitoksen Linux-koneissa.) Tuloksena ruudulle pitäisi ilmestyä luennolla esitetty kasvontunnistusesimerkki. Kuvassa olevat punaiset täplät vastaavat löytyneitä piirteitä ja kuvien väliset vihreät ja siniset viivat vastaavat piirrepareja, jotka ovat niin samankaltaisia, että ne voidaan tulkita samoiksi hahmoiksi.

- a) (1 piste). Kokeile vaihtaa `example`-hakemistossa olevat tiedostot `img1.jpg` ja `img2.jpg` toisiin kuviin ja aja ohjelma uudelleen. Kuvien tulisi olla samankorkuiset.

Kokeile vähintään viittä muuta kuvaparia, joista osassa näkyy sama rakennus, sama henkilö ja/tai sama esine. Kokeile myös kuvapareja, jotka eivät esitä samaa kohdetta. Yritä löytää esimerkkejä, joissa SURF-piirteisiin perustuva tunnistus toimii (eli samaa esittävistä kuvista löytyy useita yhteensopivia piirteitä) ja esimerkkejä, joissa se ei toimi.¹

- b) (1 piste). Valitse yksi (tai useampi) kuva ja muokkaa sitä

- säätämällä valoisuutta, kontrastia tai väritasapainoa,
- kallistamalla kuvaa (rotaatio),
- rajaamalla ja skaalaamalla uudelleen,
- lisäämällä kohinaa (noise) tai sumentamalla (blur).

Voit käyttää haluamaasi kuvankäsittelyohjelmaa (iPhoto, GIMP, Photoshop, jne).

Toimiiko SURF, kun sovellat sitä alkuperäiseen ja muokattuun kuvaan?

Vaihtoehtoisesti (hauskempi tapa): Ota itse kameralla (tai Lumiallasi) kuvapareja, jotka esittävät samaa esinettä, rakennusta tai henkilöä, ja kokeile toimiiko SURF.

Pohdi millaiset muutokset kuvissa ja olosuhteissa aiheuttavat ongelmia SURF:iin perustuvalla kuvantunnistukselle ja millaiset taas eivät.

Tehtävä palautetaan kirjoittamalla lyhyt selostus siitä, mitä huomasit: selitä mielestäsi olennaiset huomiot, joita teit kokeista. Liitä selostukseen muutama kuvaesimerkki.

Palautettavan tiedoston muoto on oltava PDF. Käytä raportissa matalaresoluutioisia kuvia, jotta PDF-tiedoston koko pysyy alle 2 MB. Vaihtoehtoisesti voit tulostaa raportin ja toimittaa sen laskariohjaajille (vaikka pajassa) tai luennoijalle (vaikka luennolla).

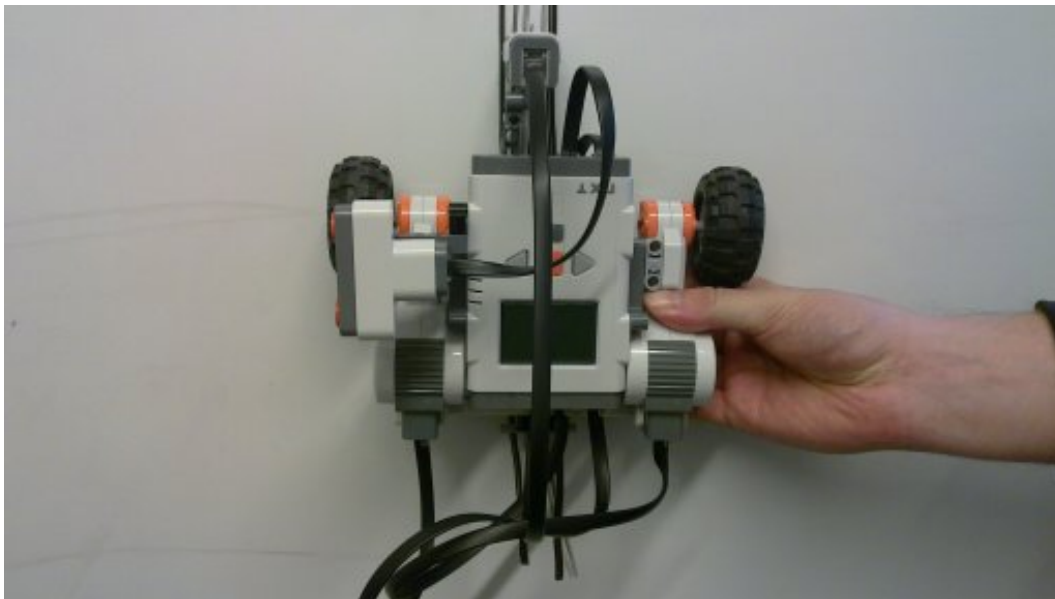
¹Lisävinkki: ohjelma näyttäisi (ehkä) toimivan joskus paremmin, jos `src/com/stromberglabs/jopensurf/SurfCompare.java` -tiedoston riville 66 vaihtaa totuusarvoksi `private boolean mUpright = true`.

Tehtävä 2. Robotiikka.

Tämä tehtävä suoritetaan osin “pajamallia” soveltaen: **Ensimmäinen osa (1 piste) tehdään normaaliin tapaan kotona ja se merkitään tehdyksi laskuharjoitusten aluksi.** Toinen osa (2 pistettä) suoritetaan osallistumalla yhteen tai useampaan pajasessioon. Huomaa että toista osaa on silti syytä valmistella mahdollisimman paljon etukäteen, jotta itse sessiossa saa enemmän irti. Ohjeet pajaan ilmoittautumisesta löytyvät kurssin sivulta (kohdassa *Uutisia*).

Tehtävässä käytetään sitä varten valmiiksi rakennettua Lego Mindstorms -robottia, jonka rakennetta ei ole tarkoitus muuttaa. Robotissa on pyörät, valosensori, joka osoittaa kohti lattiaa pyörien akselista n. 10cm edempänä sekä ultraäänisensori, joka osoittaa suoraan vasemmalle n. 10cm robotin keulasta taaksepäin. Robotin rakenteesta saa paremman käsityksen katsomalla oheista kuvaa (kuva 1).

- a) (1 piste) Toteuta valmista runkoa (`HelloWorld.java`; ks. kurssin sivu) muokkaamalla ohjelma, jonka avulla robotti seuraa lattiaan merkittyä vaaleaa viivaa, kunnes viiva loppuu.
- b) (2 pistettä) Toteuta ohjelma, jonka avulla robotti etsii ultraäänisensorien avulla “taskun” eli esim. oviaukon, jonka koko on vähintään $40\text{cm} \times 40\text{cm}$, minkä löydettyään robotti “pysäköi” taskuun: ajaa taskun ohi, peruuttaa, kääntyy, ajaa taskuun ja pysähtyy siten, että se on kokonaisuudessaan taskussa.



Kuva 1: Tehtävässä käytettävä robotti.