Uvod

- Poudarek na računalniškem vidu, na koncu še nekaj robotike
- reševanje nalog, ki obravnavajo teorijo s predavanj
  - delno skupaj
  - nekaj tudi samostojno
- domače naloge: sprotno preverjanje znanja (+/o/-)
- seminarska naloga:
  - znano do sredine novembra
  - strojni vid
- na vajah bomo uporabljali odprtokodni programski paket Octave

Uvod v Octave

- Skriptni jezik z interaktivnim terminalom
- Primarni namen numerično računanje
- Poudarek na uporabi matrik (več dimenzij)
- Podobnost z Matlab-om
- Komentarji predznačeni s simbolom `%`
- Če ni eksplicitno podano, se rezultat posamezne operacije shrani v spremenljivko `ans`

Spremenljivke, vektorji, matrike

- Spremenljivki a dodelimo vrednost z ukazom `a = <vrednost>`
- Na voljo imamo (navedeni samo najbolj pomembni) primitivne tipe logical, int, double, string ter matrike tipov int in double
- Vektor je matrika dimenzij `1xN`
- Operatorji nad numeričnimi tipi (tudi matrikami): `+, -, *, /, ^`
  - `A + B` ter `A - B` seštej/odštej dve matriki istih dimenzij
  - `A * B` množenje matrik `MxN` in `NxK`, retultat matrika `MxK`
  - `A / B` reši sistem enačb `Ax = B`
  - `A .*` ter `A ./` zmnoži/deli matriki istih dimenzij po elementih
  - `A ^ b` potenca, `A .^ b` potenca po elementih
  - `A'` transponiranje matrike
- Logični operatorji: `<, >, <=, >=, ==, <>, &, |, ~`
  - Delujejo tudi nad matrikami
- Definicija matrik
  - `A = [ 1, 2, 3, 4 ]` - vektor `1x4`
  - `A = [ 1, 2; 3, 4 ]` - matrika `2x2`
  - `A = zeros(2, 2)` - matrika `2x2` - vsi elementi 0
  - `A = ones(2, 2)` - matrika `2x2` - vsi elementi 1
  - `A = rand(2, 2)` - matrika `2x2` - naključne vrednosti med 0 in 1
- Dostop do elementov
  - `A(1, 5)` - element v prvi vrstici, petem stolpcu
  - Drugače kot v večini programskih jezikov! Prvi indeks določa vrstico, drugi stolpec. Indeksi se začnejo z 1.

- `A(1:3, 3:6)` - podmatrika med prvo in tretjo vrstico ter tretjim in šestim stolpcem.
- `A(3:end, :)` - podmatrika med tretjo in zadnjo vrstico ter vsemi stolpci
- `A(end:-1:1, :)` - vrne matriko z obrnjenim vrstnim redom vrstic
- `A(1:2:10, :)` - vrne matriko z vsako drugo vrstico
- `A([1 4 3 2], :)` - vrne matriko z podanim vrstnim redom vrstic
- `A(logical([1 0 0 1]), :)` - vrnem samo vrstice, za katere je ustrezni element podane maske enak 1

Izpis vrednosti

- Če ukaza ne zaključimo s podpičjem, se bo rezultat operacije izpisal na zaslon.
  - a = 1 bo izpisal a = 1
- Poljuben niz lahko izpišemo tudi z ukazom disp
  - `disp(1)` izpiše 1
  - `disp("foo")` izpiše foo
  - `disp(A)` izpiše vsebino spremenljivke A
- Uporabimo lahko tudi funkcijo printf, ki se obnaša tako, kot to poznamo iz C-ja
  - `printf("Vrednost spremenljivke a je %d\n", a)` izpiše vrednost spremenljivke a

Nadzor toka programa

- Octave pozna dve vrsti zank (for, while) ter if in switch stavka.
- For zanka

```
for n = <zaporedje>
      <koda>
end
```

  - Kot zaporedje lahko uporabimo notacijo za zaporedja (`<začetek>:<korak>:<konec>`) ali vektor elementov

- While znaka

```
while <logični pogoj>
      <koda>
end
```

- If stavek

```
if <pogoj>
      <koda>
end
```

  Lahko dodamo tudi else blok:

```
if <pogoj>
      <koda>
else
      <koda>
end
```

- Switch stavek

```
switch <izraz>
      case <možnost1>
            <koda>
      case <možnost2>
            <koda>
      otherwise
            <koda>
end
```

- Za razliko od C-jevskega switch stavka, tu ni potreben ekspliciten izhod iz bloka z ukazom break ali continue (vendar ukaza obstajata za uporabo v zankah).

Funkcije in datoteke

- Najpogosteje bolj kompliciranе programe organiziramo v funkcije, le-te pa pišemo v datoteke, ki se končajo s končnico `.m`
- Definicija funkcije:

```
function [<izhodni parametri>] = <ime funkcije> (<vhodni parametri>)
      <koda>
end
```

- Globalne spremenljivke definiramo z uporabo izraza **`global`** `<ime>`
- Funkcije lahko vračajo več rezultatov. V tem primeru moramo ob klicu pripraviti dovolj spremelnjivk, za sprejem le-teh.

Procesiranje slik

- Matlab slike obravnava kot matrike. Pri tem še enkrat povejmo, da je indeksiranje matrik za prvi dve dimenziji drugačno, kot pri večini drugih programskih jezikov. Slika ima tako dimenzije `<višina>x<širina>x<število kanalov>`.
- Za branje/pisanje slik sta na voljo funkciji `imread` in `imwrite`.
  - `[IMG, MAP, ALPHA] = imread (FILENAME)`
  - `imwrite (IMG, [MAP], FILENAME, FMT, P1, V1, ...)`
- Za prikaz matrike kot slike uporabimo ukaze image, imagesc, imshow.
  - image prikaže sliko, pri čemer vrednosti niso prilagojene prikaznemu razponu
  - imagesc prikaže sliko, pri čemer vrednosti prilagodi prikaznemu razponu
  - imshow prikaže sliko brez koordinatnih osi
  - Pri prikazu neprilagojenih podatkov moramo biti pozorni tudi na tip podatkov. Matirke tipa double so prikazane v rangu 0 do 1, matrike tipa uint8 pa med 0 in 255.
- Črnobele slike so prikazane z uporabo barvnega indeksiranja. Z ukazom `colormap` lahko nastavimo poljubno preslikavo.
  - Nekaj privzetih barvnih shem je gray, jet in bone
  - Funkciji lahko podamo tudi svojo shemo kot matriko velikosti `Nx3`.
- Velikost slike dobimo s splošnim ukazom size
  - `[x y c] = size(A)` - dobimo rezulrat po komponentah
  - `s = size(A)` - dobimo rezultat kot vektor

Dodatna pomoč

- Za pomoč v samem programu uporabi ukaz `help <pojem>`, recimo:
  - `help image` prikaže pomoč o funkciji image

    - `help +` prikaže pomoč o operatorju +
- Uvod v Octave

Vaje

- Branje slike
- Prag nad matriko

```
A = rand(100, 100)
A(A > 0.5) = 1
A(A > 0.9 | A < 0.1) = 0
```

- Primerjava hitrosti (zanke vs. matrične operacije)
    - tic, toc
    - seštevanje matrik