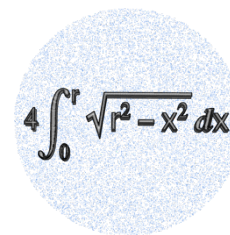


Valokuvien matematiikkaa



Tekijät: Topi Törmä, Oona Laakso, Birgitta Nenonen-Andersson, Heli Virtanen

Kohderyhmä: Yläkoulu / lukio (esimerkiksi MAA11 LOPS 2019 mukaan)

Laadittu vuonna 2021.

Tämä teos on lisensoitu Creative Commons Nimeä 4.0 Kansainvälinen -lisenssillä.
<https://creativecommons.org/licenses/by/4.0/deed.fi>

Esitiedot:

- Ohjelmoinnin osaamisesta on hyötyä

Oppitunnin kesto: 75-90 min

Oppimistavoitteet:

- Miten kuva esitetään matriisina?
- Tutkivan asenteen kehittäminen
- Funktio-käsitteen vahvistaminen

Muut tavoitteet:

- Kiinnostuksen lisääminen matematiikkaa kohtaan.

Sisältö: Harmaasävykuva, värikuva numeroina, valinta: Yhteenlasku ja kertolasku TAI neliöjuurella valokuvan vaalennus

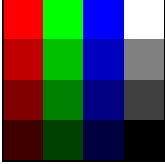
Toteutus:

- Esittely harmaasävykuva + kuvien ja lukujen yhdistely **25min**
 - o Samuli Siltasen Youtube-video Samun tiedepläjäys: valokuva ja neliöjuuri: https://www.youtube.com/watch?v=1sDxtV_V56s (alusta 3min 30s)
 - o Vaihtoehtoinen Video 1: Samun tiedekanavan Youtube-video PechaKucha: Matematiikka, tuo taiteilijan digitaalinen sivellin: <https://www.youtube.com/watch?v=v98KEaYGaqc&t=2s> (alusta n. 1min 20s)
 - o Vaihtoehtoinen Video 2: Samun tiedekanavan Youtube-video Samun tiedepläjäys: Selfie ja bitit: <https://www.youtube.com/watch?v=uiUxgx2QwBU>
 - o Liitteessä 1 olevien RGB-kuvatehtävien ratkaiseminen
 - o Ei tarvita konetta vielä, käsin
- Tavoitteena on käsitellä harmaasävykuvaa ja tehdä niistä histogrammi (tähän ei ole tarkempaa ohjeistusta tai esimerkkitehtävää) **5min**

- Kuvien yhteenlasku ja kertolasku (vaihtoehto 1) **35min**
 - Samun tiedekanavan Youtube-video Samun tekniikkatuokio: Valokuvien yhteenlasku: <https://www.youtube.com/watch?v=EywzmPRLa9M>
- Neliöjuurella valokuvan vaalennus (vaihtoehto 2) **35min**
 - Samuli Siltasen Youtube-video Samun tiedepläjäys: valokuva ja neliöjuuri: https://www.youtube.com/watch?v=1sDxtV_V56s (kohdasta 3min 30s loppuun asti)
- Mietintä mitä operaatioita kuville voi suorittaa ja mitä tapahtuu kuville **10min**
- Vapaamuotoinen kokeileminen jää pois 75min versiosta (**15min**)
 - Vapaamuotoinen kokeileminen, ks. liitteiden 2 ja 3 MATLAB-koodit.

LIITE 1: RGB-KUVATEHTÄVÄT

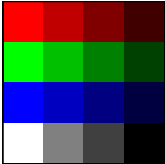
Alla on 12 kuvaa ja niiden pikseleitä vastaavat kolme värikanavaa lukuina taulukoissa. Yhdistä kuvat oikeisiin värikanavataulukoihin. Mieti myös perustellut ratkaisulllesi!

1.  A.

64	25	100	255
80	90	89	70
80	120	150	192
150	128	4	0

200	25	100	200
90	90	60	164
180	120	10	0
20	28	164	19

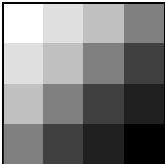
70	255	0	0
70	255	0	200
64	192	128	100
100	28	6	80

2.  B.

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

200	200	200	200
175	175	175	175
150	150	150	150
125	125	125	125

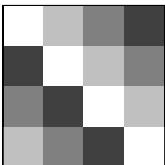
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

3.  C.

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

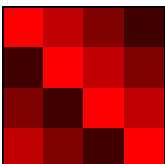
255	200	150	20
64	25	92	228
200	6	255	192
175	90	64	100

4.  D.

255	223	192	128
223	192	128	64
192	128	64	32
128	64	32	0

255	223	192	128
223	192	128	64
192	128	64	32
128	64	32	0

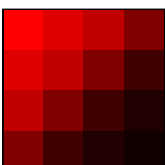
255	223	192	128
223	192	128	64
192	128	64	32
128	64	32	0

5.  E.

255	223	192	128
223	192	128	64
192	128	64	32
128	64	32	16

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

6.  F.

255	192	128	64
64	255	192	128
128	64	255	192
192	128	64	255

255	192	128	64
64	255	192	128
128	64	255	192
192	128	64	255

255	192	128	64
64	255	192	128
128	64	255	192
192	128	64	255

LIITE 2: Lukee kuvan, valitsee sieltä värikanavat, vaihtaa niiden järjestyksen ja tuottaa uuden kuvan. Näyttää alkuperäisen kuvan sekä muokatun kuvan.

```
im = imread('muuri2.png'); % Lataa kuva, täytä tiedostonimi
% Kuvan normalisointi 0-1 välille:
im = double(im);
im = im/max(im(:));
[row,col,ch] = size(im);

% Jaa kuvan värikanavat erikseen:
imr = im(:,:,1);
imb = im(:,:,2);
img = im(:,:,3);

% Alusta + muokkaa värikanavien järjestys:
kuva = zeros(size(im));
kuva(:,:,3) = imr;
kuva(:,:,2) = img;
kuva(:,:,1) = imb;

% Katso kuvaa
figure(1)
subplot(121)
imagesc(im)
subplot(122)
imagesc(kuva)
```

LIITE 3: Lataa kuvan, normalisoi sen (im). Laskee kaksi kuvaa yhteen (im2) ja tämän jälkeen normalisoidaan uudestaan (im3). Lasketaan neliöjuuri (im4). Tehdään osittaisderivaatta kuvalle im3 pystysuunnassa sarake kerrallaan (im5). Tutkitaan pienempää aluetta edellisestä kuvasta (im6). Tutkitaan vielä pienempää aluetta kuvasta (im7). Loppukoodi piirtää kuvat eri tilanteista.

```
im = imread('kala_alk.png'); % Lataa kuva, täytä tiedostonimi
im = double(im);
max(im(:))
[row,col] = size(im)

im2 = im + im;
maksimi = max(im2(:))

im3 = im2/maksimi;
im4 = im3.^(1/2);

im5 = zeros(size(im3));
for iii = 1:(col-1)
    im5(:,iii) = im3(:,iii+1)-im3(:,iii);
end

im6 = im5(500:600,500:600);
im7 = im6(1:10,1:10);

figure(1)
imagesc(im)
axis equal
axis off
colormap gray

figure(2)
imagesc(im2)
axis equal
axis off
colormap gray

figure(3)
subplot(121)
imagesc(im3)
axis equal
axis off
colormap gray
subplot(122)
imagesc(im4)
axis equal
axis off
colormap gray

figure(4)
subplot(121)
imagesc(im3)
axis equal
axis off
colormap gray
subplot(122)
imagesc(im5)
```

```
axis equal
axis off
colormap gray
```

```
figure(5)
subplot(131)
imagesc(im3,[0 1])
axis equal
axis off
colormap gray
subplot(132)
imagesc(im6,[0 1])
axis equal
axis off
colormap gray
subplot(133)
imagesc(im7, [0 1])
axis equal
axis off
colormap gray
```