

12. Geometrisia operaatioita

Affiinimuunnokset

Toteutus Javalla

Epälineaariset muunnokset

Muodonmuutokset

Geometristen muunnosten taustaa

- Muutetaan pikselien paikkoja ja suhteellisia etäisyyksiä
- Syitä:
 - Geometristen vääristymien korjaus, esim. käytettäessä erikoislinssettä
 - Maapallon kaarevuuden korjaus (ilma-/satelliittikuvat)
 - Kuvien täsmäys (registration) päällekkäin (esim. liikkeen havaitseminen; muutokset röntgenkuvissa)
 - Efektit 'taiteellisissa' kuvissa

Yleinen geometrinen muunnos

- Muunnos voidaan ilmaista muodossa

$$x' = T_x(x, y)$$

$$y' = T_y(x, y)$$

ts. pikseli (x, y) siirretään paikkaan (x', y')

- Funktioiden T_x ja T_y muoto määrää muunnostyyppin:
 - *Lineaarinen* (ei termejä x^2 , y^2 , xy , ...)
 - *Epälineaarinen*

Affiinimuunnos

- Lineaarinen:

$$x' = a_0x + a_1y + a_2$$

$$y' = b_0x + b_1y + b_2$$

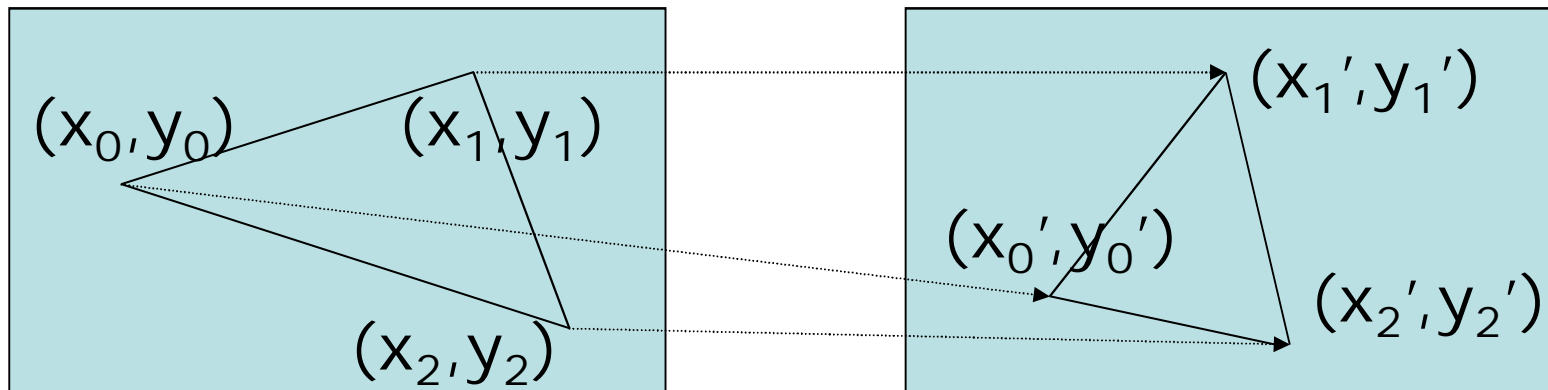
- Esitetään usein matriisitulona; mahdollistaa operaatioiden ketjuttamisen

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_0 & a_1 & a_2 \\ b_0 & b_1 & b_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Kyseessä ns. *homogeeniset koordinaatit*, kolmas dimensio = 1

Affiinimuunnoksen ominaisuuksia

- Suorat viivat pysyvät suorina, yhdensuuntaiset yhdensuuntaisina
- Muunnos tulee määriteltyä kiinnittämällä kolmen alkuperäisen pisteen paikat uudessa kuvassa \rightarrow saadaan 6 yhtälöä, joista ratkaistaan $a_0, a_1, a_2, b_0, b_1, b_2$.



Affiinimuunnoksen erikoistapauksia

- Siirto Δx , Δy :

$$\begin{bmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \\ 0 & 0 & 1 \end{bmatrix}$$

- Skaalaus, kerroin s

$$\begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Rotaatio (kulma θ):

$$\begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Vinoutus kertoimet s_1, s_2

$$\begin{bmatrix} 1 & s_1 & 0 \\ s_2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Muunnosten yhdistäminen

- Kerroinmatriisien matriisitulo
- Esim. rotaatio 45° ja siirto 100 oikealle

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 100 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sqrt{2}/2 & -\sqrt{2}/2 & 0 \\ \sqrt{2}/2 & \sqrt{2}/2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$
$$= \begin{bmatrix} \sqrt{2}/2 & -\sqrt{2}/2 & 100 \\ \sqrt{2}/2 & \sqrt{2}/2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Muunnosvaihtoehdot

- 'Eteenpäin' (forward mapping)
 - Tuloskoordinaatit pyöristetään
 - Tulospikseli ehkä kuvan ulkopuolella
- 'Taaksepäin' (backward mapping)
 - Tarvitaan käänteiskuvaus
 - Tarvitsee laskea vain tuloskuvaan mahtuvat pikselit
 - Jos lähdepikseli ulkopuolella, tulos = 0

Esimerkki taaksepäinmuunnosta

Rotaatiomatr. x Käänteismatr. = Identit.matriisi

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Muuntokaava:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & a_2 \\ -\sin\theta & \cos\theta & b_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

Interpolaatio

- Lähtö-/tulosindeksit eivät aina 1:1
- Vastaava tilanne kuin suurennoksessa/pienennyksessä (vrt. aikais.)
- Vaihtoehdot:
 - Otetaan lähin naapuripikseli
 - Bilineaarinen interpolaatio
 - Bicubic-interpolaatio (ns. 'cubic convolution', 16 x 16 –ympäristössä)
- Bilinear ja bicubic alipäästösuodattimia; bicubicin tulos terävämpi

Affiinimuunnoksen määrittely Java2D:lla

- Luokka *java.awt.geom.AffineTransform*
- Luonti esim.
 - **new** *AffineTransform*(a0, b0, a1, b1, a2, b2)
- Luonti myös 'tehdasmetodeilla':
 - *getTranslateInstance*(dx, dy)
 - *getScaleInstance*(sx, sy)
 - *getRotateInstance*(theta)
 - *getShearInstance*(sx, sy)
- Ei vielä suorita muunnosta

Affiinimuunnoksen suoritus Java2D:llä

- Luodaan *AffineTransformOp*-olio (perii *BufferedImageOp*-luokan).
Luontiparametrit:
 - *AffineTransform*-luokan esiintymä
 - Interpolointitapa
- Muunnoksen suoritus: *BufferedImageOp*-luokan *filter*-metodi
 - Syöttökuva parametrina
 - Tulokuva joko parametrina tai palautettavana arvona

Java-esimerkki

Kääntö d astetta myötäpäivään ja siirto tulokuvan keskelle.

```
public static BufferedImage rotaatio(
    BufferedImage kuva, double d) {
    // Käännös d astetta myötäpäivään; d ensin radiaaneiksi.
    d = d * Math.PI / 180.0;
    AffineTransform rot =
        AffineTransform.getRotateInstance(d);

    // Siirto tulokuvan alueelle
    double siirronPit = kuva.getHeight()*Math.sin(d);
    AffineTransform siirto =
        AffineTransform.getTranslateInstance(siirronPit, 0.0);

    // Muunnosten yhdistäminen
    rot.preConcatenate(siirto);
}
```

Java-esimerkki (jatk.)

```
// Luodaan rotaatio-operaatio-olio
BufferedImageOp rotOp =
    new AffineTransformOp(rot,
        AffineTransformOp.TYPE_BILINEAR);

// Tuloskuva on isompi; lasketaan koko
double s = Math.sin(d); double c = Math.cos(d);
int lev = (int) Math.round(c*kuva.getWidth() + s*kuva.getHeight());
int kork = (int) Math.round(s*kuva.getWidth() + c * kuva.getHeight());
BufferedImage tulos =
    new BufferedImage(lev, kork, kuva.getType());

// Affiiniuunnoksen toteutus
rotOp.filter(kuva, tulos);
return tulos;
} // end of rotaatio
```

Java-esimerkki (jatk.)

Rotaatiometodin tulos ($d=45^\circ$):



Photo Credit: US Fish and Wildlife Service, Archive: Gimp-Savvy

Vääntämisoperaatiot

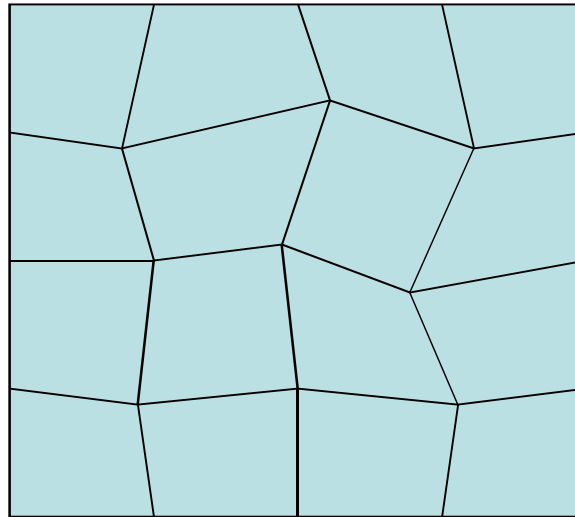
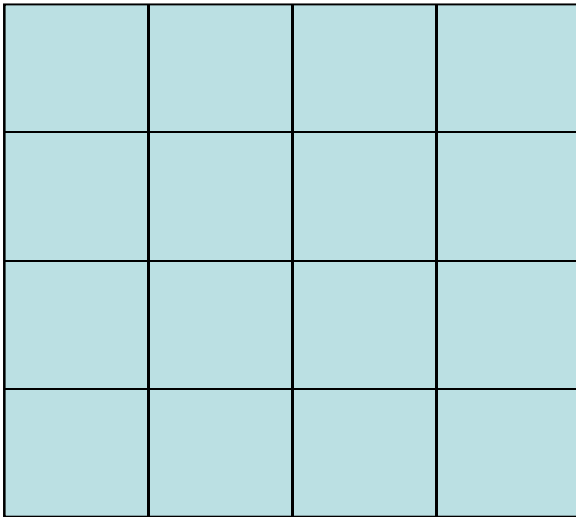
- Ns. warping, esim. neliöllinen:
$$x' = a_0x^2 + a_1y^2 + a_2xy + a_3x + a_4y + a_5$$
$$y' = b_0x^2 + b_1y^2 + b_2xy + b_3x + b_4y + b_5$$
- Määritellään ns. *kontrollipisteiden* vastaavuus lähtö- ja tuloskuvissa:
 - Neliöllinen warp: 6 pistettä
 - Kuutiollinen warp: 10 pistettä
- Sovellus: kuvien kohdistaminen; yleistys isommalle pistejoukolla

Paloittainen vääntäminen

- *Kontrolliristikko*, 4 pistettä/pala
- Bilineaarinen transformaatio:

$$x' = a_0xy + a_1x + a_2y + a_3$$

$$y' = b_0xy + b_1x + b_2y + b_3$$



Muodonmuutos (morphing)

- Muunnetaan kuva inkrementaalisesti tuloskuvaksi; molempien oltava olemassa ennen muodonmuutosta.
- Paloittainen warping & kohdistus
- Lähtö- ja tuloskuvien kontrollipisteet määrää yleensä käyttäjä; myös automaattinen vastinkohtien etsintä mahdollinen.
- Pikselien paikat ja sävyt muuttuvat kussakin vaiheessa tasaisesti.
- Sovelluksia: elokuvat, animaatiot ym.