

Tracking

Bakgrundsmodellering

Expectation Maximization (EM). n = antal sampel (pixlar) k = fördelningar.

Sammanlikheten för intensitet I : $p(I) = \sum_{k=1}^K p(I|\Gamma_k)P(\Gamma_k)$ - Mixture distribution

1. Välj en mixture distribution (tex MOG)

2. E-steget: Compute partial memberships w_{kn} . $\sum_{k=1}^K w_{kn}=1$, dvs pixeln n tillhör alla fördelningar med viss sannolikhet. w_{kn} är en vikt men man kan se den som sannolikhet.

$$\tilde{w}_{kn} = p(I|\Gamma_k)P(\Gamma_k) \quad w_{kn} = \frac{\tilde{w}_{kn}}{\sum_{k=1}^K \tilde{w}_{kn}}$$

$$3. M\text{-steget: } P(\Gamma_k) = \frac{1}{N} \sum_{n=1}^N w_{kn}$$

För Mixture of Gaussians gäller:

$$M_k = \frac{\sum_{n=1}^N w_{kn} I_n}{\sum_{n=1}^N w_{kn}} \quad \sigma_k^2 = \frac{\sum_{n=1}^N w_{kn} (I_n - M_k)^2}{\sum_{n=1}^N w_{kn}}$$

Lucas-Kanade (LK)

Minimerar kostfunktionen: $\mathcal{E} = \sum_{x \in N} \|f_1(x) - f_2(x+d)\|^2$ 'betraktar bara translationer'

Symmetrisk version: $\mathcal{E} = \sum_{x \in N} \|f_1(x+\frac{d}{2}) - f_2(x-\frac{d}{2})\|^2$, (Tomasi)

-Displacement $d = T^{-1}S$

-Structure tensor $T = \int w(x) \nabla_i f_1 \nabla_i f_2^T dx$ $w(x)$ - vikt, specifierar region

-Dissimilarity (dikhet) $S = \int w(x) \nabla_i f_1 (f_2(x) - f_1(x)) dx$, $s = Td$ OBS! $S = \mathcal{E}$!

• Regularised derivatives:

$$\frac{\partial}{\partial x} (J * g) = \frac{\partial}{\partial x} J * g = J * \frac{\partial}{\partial x} g = J * \underbrace{\frac{-x}{\sigma^2} g}_{\text{alltid derivatan av en gaussisk kernel}}$$

Detta ger alltid den exakta derivatan. Om man först filtrar och sedan deriverar filtervaret för man approximation. Dessutom är det mer beräkningstungt.

LK-tracking kan ge subpixel noggrannhet.

Härledning av Lucas-konade

$$\underline{2D} \quad E = \iint_D w(x) (f(x,d) - D(x))^2 dx$$

$$J(x+d) \approx J(x) + d_x \frac{\partial J(x)}{\partial x} + d_y \frac{\partial J(x)}{\partial y} = J(x) + d^T \nabla J(x) \quad [\text{första ordningens Taylor utveckling}]$$

$$Q \approx \int_{\Omega} (J(x) - I(x) + d^T \nabla J(x))^2 dx \quad (\text{om man ignorerar } w)$$

$$\frac{de}{da} \approx 2 \sum_w (J(x) - I(x) + a^T \nabla J(x)) \nabla J(x) dx \quad (\text{derivater f鰎 att hitta minimum})$$

s鰈t derivatan = 0 f鰎 optimum

$$\iint_U (J(x) - I(x)) \nabla J(x) dx = - \iint_U \nabla J(x) d(J(x)) dx \quad (\text{försök bygga ut } d)$$

$$\underbrace{\int (f(x) - I(x)) \nabla J(x) dx}_{= \epsilon - 2 \times 1 \text{ Vektor}} = \underbrace{(-\int_w \nabla f(x) \nabla J^T(x) dx)^T d}_{= T - 2 \times 2 \text{ -matrix}} \Rightarrow Td = \epsilon \quad d = T^{-1} \epsilon$$

$$1D \quad I = \int w(x) \left(3x + \frac{d}{2} \right) - I \left(x - \frac{d}{2} \right)^2 dx$$

$$I(x + \frac{d}{2}) \approx I(x) + \frac{dx}{\partial x} \frac{\partial I(x)}{\partial x} = I(x) + I'(x) (-\frac{d}{2}) \quad \begin{cases} \text{första ordningen} \\ \text{Taylor utveckling} \end{cases}$$

$$\frac{dE}{dx} = 2 \left\{ \left(J(x) + J'(x) \frac{d}{2} - I(x) + I'(x) \frac{d}{2} \right) \left(J'(x) + \frac{I'(x)}{2} \right) \right\} dx$$

$$= \underbrace{\int (J(x) + J'(x) \frac{d}{2} - I(x) + I'(x) \frac{d}{2}) \left(\frac{J(x) + I(x)}{2} \right)^2 dx}_{= 0} = \int (J'(x) + I'(x))^2 d \underbrace{(T_1 + T_2)}_{= T_1 + T_2} = \epsilon$$

Harris - detector

Harris - detector
Hittar pixlar med hörn (sånt som ej blir i 1D) så att de inte
skall påverkas av appertur-problemet vid tracking.

$$C_{\text{Harris}} = \det(T) - k (\text{tr}(T))^2 \quad T - \text{structure tensor}$$

$$\text{char}_T = \det(T - \lambda I) = T_{11}T_{22} - T_{12}^2 = \lambda_1\lambda_2$$

Double angle representation

$$\frac{1}{2} = \left(\frac{T_{11} - T_{22}}{2T_{12}} \right) - \text{local struktur}$$

Motion

Optical flow

BCCE = "Brightness constancy constraint equation"

$$\frac{df}{dx_1} V_1 + \frac{df}{dx_2} V_2 + \frac{df}{dt} = 0 \quad \epsilon = \int_{\Omega} (V(x_1, x_2) \nabla_{2D} f + \frac{df}{dt})^2 dx$$

- Antagande: $\{V(x)\}$ - konstant inom Ω V = motion field
 ∇f = not self-parallel

- Problem: $\begin{cases} \text{Ej unik lösning} \\ \nabla f = 0 \end{cases}$
 - Man antar också att allt displacement beror på 3D-rörelser

Horn & Schunk

$$\epsilon = \int_{\Omega} (V(x_1, x_2) \nabla_{2D} f + \frac{df}{dt})^2 dx + \rho \underbrace{\int_{\Omega} \|\nabla_{2D} V_1\|^2 + \|\nabla_{2D} V_2\|^2 dx}_{\text{smoothness term}} \quad \begin{array}{l} \text{Integration över} \\ \text{hela bilden} \\ \text{"Global metod"} \end{array}$$

- Antagande: kontinuerligt, "smooth", motion field v

Löser problemet med smoothness-termen, ger unik lösning, variationen i v borde också bli minimal

- Problem: Not valid in sharp motion boundaries tex occlusion

Two constraints - optical flow

least square minimization based on T_{3D} & ϵ

1. $\bar{V} = [V_1, V_2]^T$ minimize ϵ that gives $T_{3D}\bar{V} = \bar{e}$

Kan ej hantera aperture problem. Linjer blir i 1D $\Rightarrow T_{3D}$ singular

2. $\bar{V}_{ST} = [r_1, r_2, r_3]^T$ $\|\bar{V}_{ST}\|=1$ ger $T_{3D}\bar{V} = \lambda\bar{V}$ dvs det

V_{ST} som minimerar ϵ måste vara en egenvektor till T_{3D}

minimum blir det minsta egenvärdet. Lite tyngre beräkningar men kan hantera linjer som rör sig ty

de blir i 2D i spatiotemporala rummet.

total least square error based on T_{3D}

$$V_1 = \frac{r_1}{r_3}, V_2 = \frac{r_2}{r_3} \quad (\text{ger } \bar{V} \text{ från } \bar{V}_{ST})$$

Aperture - problemet

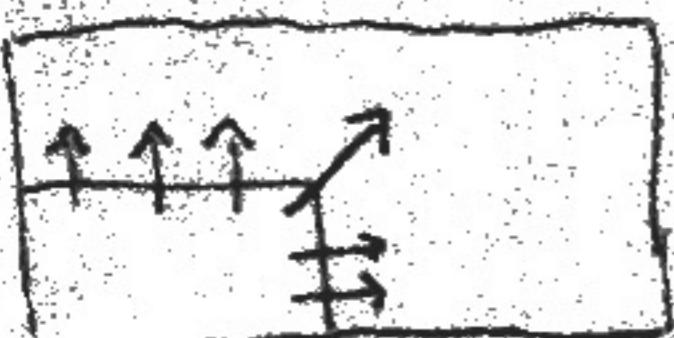
Linjer, kanter ger ej unik lösning på motion. Man vet ej säkert hur de rör sig. (Blir i 1D)

Jämförelser

LK & Block matching

LK

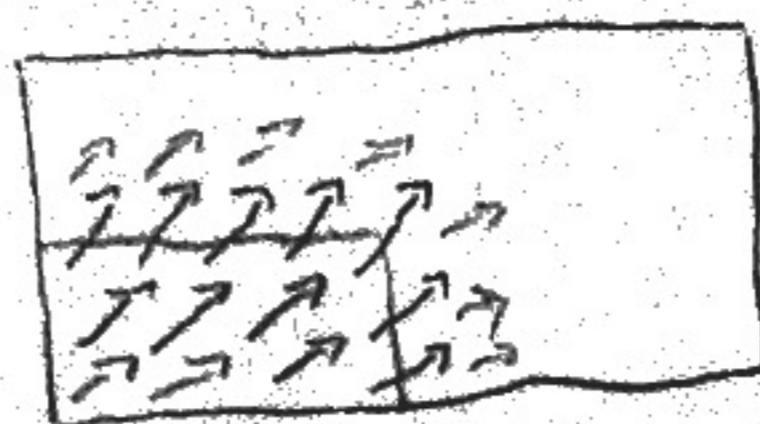
- Ganska råkretsningsgt
- Sub-pixel noggrannhet
- Begränsning i "block"
- Kan inte hantera icke-differensierbara bilder
- LK & HS
- LK väljer kortaste vektorn, dvs o i homogena ytor och rakt upp eller rakt åt sidan i linjer/kanter
- HS väljer den vektor som är mest lik resten av motion field



LK

Block-matching

- Mycket råkretsningsgt
- Enkelt att implementera
- Kräver flera begränsningar, både block-storlek och längd på förflyttning
- Kan hantera icke-differensierbara bilder



HS

Denoising

Anisotropic diffusion

Aven kallad "tensor-driven". Kernel behöver ej vara symmetrisk (men kan vara det). Generellt formen av en ellips längs med strukturernas riktning.

Den kan "smootha" längs med strukturernas riktning.
 $g_s = g_{LP} + \sum_{k=1}^N C_k g_{HP,k}$ $C_k = \text{control tensor}$ $C_k = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \\ c_{31} & c_{32} \end{pmatrix}$ $k = \text{pixlar}$?

g_{LP} - låg pass del som bevarar medelintensiteten i bilden

g_{HP} - beror av lokal struktur (ska bevara kanter och linjer)
 radiell funktion

$$\frac{\partial}{\partial s} L = \frac{1}{2} \operatorname{div}(D \operatorname{grad} L), \quad \text{Antag att } D \text{ varierar längsamt med } x$$

$$\text{Regularization: } \frac{\partial}{\partial s} L \approx \frac{1}{2} \operatorname{tr}(D \nabla^2 L) \approx \frac{1}{2} \operatorname{tr}(D H L)$$

$$L(x,y;s) = (g_s * f)(x,y) \quad f = \text{bilden}, \quad D = \text{diffusions tensor}$$

samma egenvektorer som T men
 andra egenvärden

Inhomogeneous diffusion

Aven kallad "Perona Malik". Alltid symmetrisk kernel.

$$\frac{\partial}{\partial s} L = \frac{1}{2} \mu \nabla^2 L \quad g_s = \frac{1}{2\pi\mu s} e^{-\frac{1}{2\mu s}(x^2+y^2)}$$

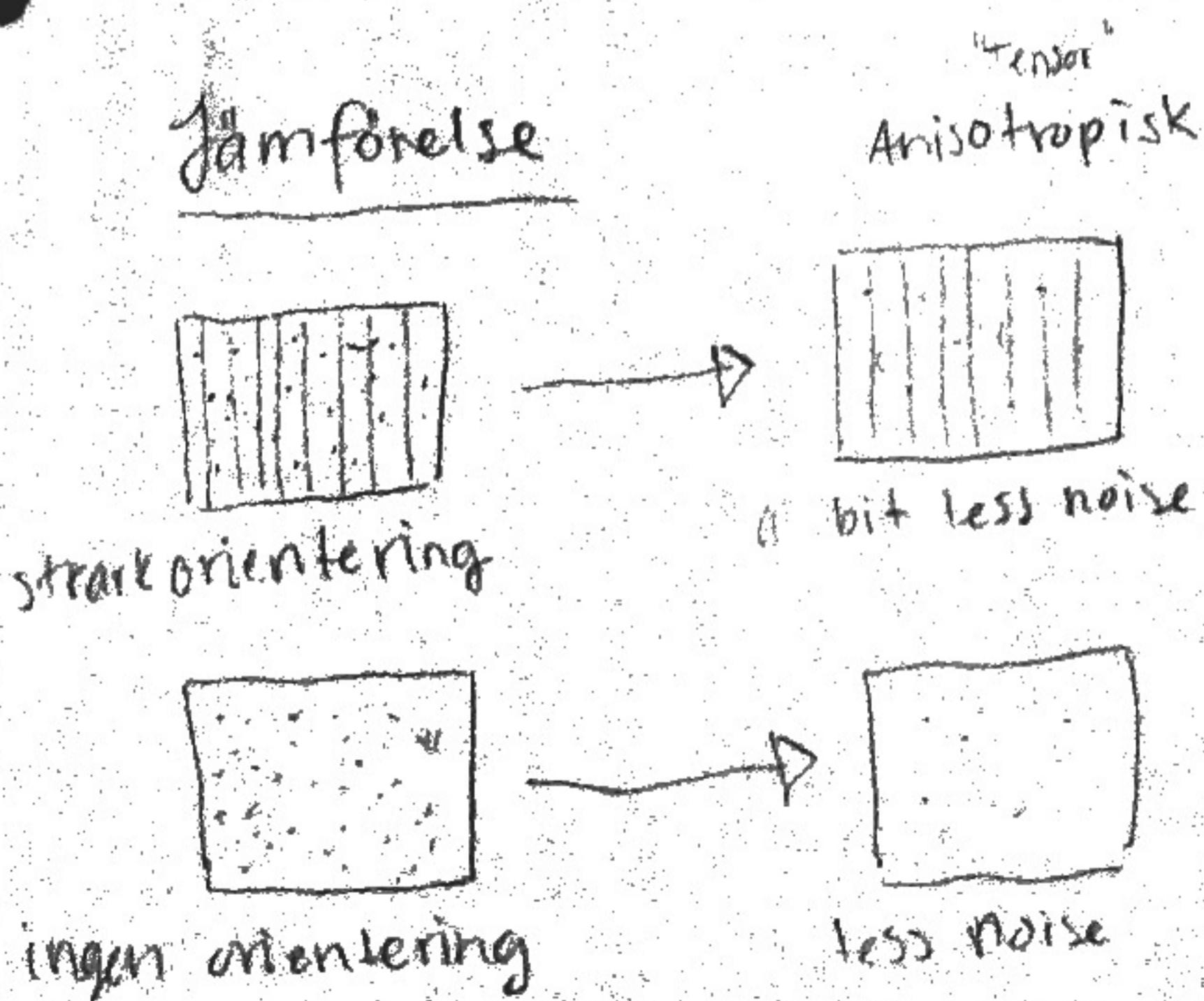
$$\mu(x,y) = \frac{1}{1 + |\nabla f|^2/\lambda^2} - \text{konst}$$

Ex: Strong orientation structure in image \Rightarrow small support \Rightarrow still noisy
 No strong orientation in image \Rightarrow large support \Rightarrow large smoothing \Rightarrow less noise

Pros: Noise effectively removed in flat regions, edges preserved

Cons: Noise preserved close to edges

Jämförelse



"Tensor"
Anisotropisk

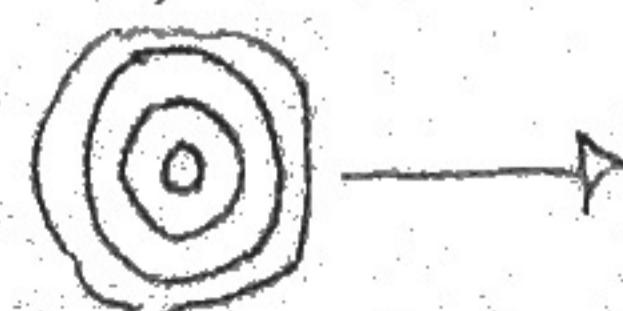
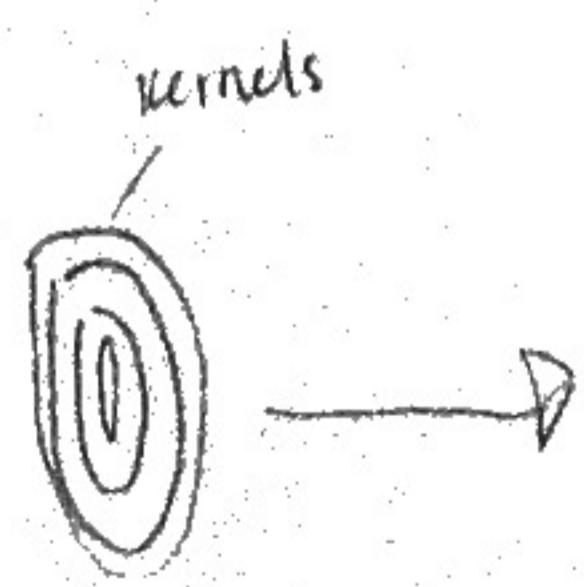
a bit less noise

less noise

Perona-Malik
Inhomogenous

almost not less
noise at all

much less noise



Control tensor

Control tensor C har samma egenvektorer som T_{LP} men ordna egenvärden.
Den beräknas enligt:

1. Löp passfitterna T till T_{LP} $\sum_{k=1}^n \lambda_k \hat{e}_k \hat{e}_k^T$ $\lambda_k \geq \lambda_{k+1}$

2. Normalisera T_{LP} , $T_{LP} = \sum_{k=1}^n \gamma_k \hat{e}_k \hat{e}_k^T$

3. Sätt $C = \sum_{k=1}^n \gamma_k \hat{e}_k \hat{e}_k^T$ $\gamma_k \geq \gamma_{k+1}$, $\gamma_k = \gamma_k(\lambda_1, \dots, \lambda_n)$, $\gamma_n = e^{-\frac{\lambda_n}{2}}$



Euler-Lagrange

$$L_f - \sum dx_i L f x_i = 0 \quad L_i\text{-norm} = |\nabla f|?$$

data term $\rightarrow L_i\text{-term?}$

Multiple View Geometry

RANSAC = Random Sample Consensus - Elimineras artliers.

1. Välj \times antal punkter (randomly) tex 8 för 8-point algor.
2. Uppskatta modell utifrån dessa. Tex en fundamentalmatris F
3. Beräkna antalet punkter som följer modellen, av alla punkter tex med F , testa villkoret $e_1^T F e_2 = 0$ - epipolarvillkoret
4. Sätt de som är tillräckligt nära att passa modellen till inliers, övriga till outliers
5. Repetera tills modellen är tillräckligt bra eller välj bästa modellen av n iterationer

Gold Standard

Man gör icke-linjär optimering över en kamera och 3D-punkter (dvs som BA men bara en kamera).

De resulterande två kamernorna c_1, c_2 ger F -matrisen. Minimerar re-projection error ϵ , som vanligt.

$F \& E$ - matriser

Fundamental matrix beskriver förhållandet mellan två nyer. Essential matrix är normaliserade varianten.

$E = R^T [t]_x$. Dvs beskriver relativ rotation och translation mellan två kamrar. R och t kan fås med singulärvärdes uppdelning av E . $F_{21} X_{22} \sim F_{31} X_{33}$

PnP & BA

Minimerar re-projection error. Dvs 3D-punkternas projection i en bild, deras distans till korresponderande punkt i bilden.

PnP optimisar över en ny kamermatris. BA optimisar över alla kamermatrizen och 3D-punkter.

Minsta n för PnP är 3.

Epipolar constraint

$x_1^T F x_2 = 0$ x_1, x_2 projektorer av samma 3D-punkt

i två utsiktar. Ingen motion. F kan uppskattas.

från 7 eller fler korrespondenser. Tex med 8-punkts algoritmen.

Local Invariant Features

- Detection
- Description
- Matching

Geometric: Frame equivariant to rotations, scalings and skew

Photometric: Descriptor more advanced than direct sampling

SIFT = Scale Invariant Feature Transform

Detect canonical frame directly

• Detection: Image is extended with extra dimension for scaling/blurring

$f(x,y; s) = (f_0 * g(s))(x,y)$ Gaussian blurring kernel

Find points using difference-of-Gaussians pyramid

Rotation invariance: by computing gradient histogram and finding peaks

Affine illumination invariance: by calculating h from gradients

and then normalising descriptor, $\tilde{h} = \frac{h}{\|h\|}$

• Transform SIFT detector to SIFT descriptor to add photometric invariance and add robustness to scale and translation

Alternativ metod MSER.

Transformation

$$\begin{pmatrix} s & 0 & t_1 \\ 0 & s & t_2 \\ 0 & 0 & 1 \end{pmatrix}$$

Rigid (scale+translation)

3 dof

$$\begin{pmatrix} s_1 & s_2 & t_1 \\ -s_2 & s_1 & t_2 \\ 0 & 0 & 1 \end{pmatrix}$$

Similarity

(Rigid+rotation)

4 dof

$$\begin{pmatrix} a_{11} & a_{12} & t_1 \\ a_{21} & a_{22} & t_2 \\ 0 & 0 & 1 \end{pmatrix}$$

Affine

(Similarity + Scew)

6 dof