

Dolžina vektorja:  $\|a\| = \sqrt{\sum_{i=1}^n a_i^2}$ , Skalarni produkt:  $a \cdot b = \sum_{i=1}^n a_i b_i$

## HIPERSPEKTRALNO ZAJEMANJE

Spektralni kot/kosinusna razdalja (p=spekter piksla,  $e_i$ =ciljni spekter):  $\alpha = \arccos(\frac{p \cdot e_i}{\|p\| \|e_i\|})$

**popolno spektralno razmesanje:** 1. Spektri vseh končnih članov v sceni morajo biti znani. 2. Resi sistem  $n$  linearnih enačb za vsak piksel ( $n$ -st. končnih članov,  $e_i$ -i-ti končni član,  $\zeta$ -spekter, p-piksel):

$$\zeta(p) = \alpha_1 \zeta(e_1) + \alpha_2 \zeta(e_2) + \dots + \alpha_n \zeta(e_n)$$

## ODSTRANJEVANJE PEGASTEGA ŠUMA

povprecje:  $\mu = \frac{\sum_i^N x_i}{N}$ , standardni odklon:  $\sigma = \sqrt{\frac{\sum_i^N (x_i - \mu)^2}{N}}$ , varianca:  $\sigma^2 = \frac{\sum_i^N (x_i - \mu)^2}{N}$

**Adit. šum:**  $I(p) = X(p) + N(p)$ , **Mult. šum:**  $I(p) = X(p)N(p)$ , I-šumna slika, X-brezšumna slika, N-šum,  $p = [i, j]$

$$\text{MSE} = \frac{1}{MN} \sum_i \sum_j (I(p) - \hat{X}(p))^2, \text{RMSE} = \sqrt{\text{MSE}}, \text{PSNR} = 20 \log_{10} \left( \frac{\max_p I(p)}{\text{RMSE}} \right) [\text{dB}], \text{SNR} = 10 \log_{10} \frac{\sigma_I^2}{\sigma_X^2} [\text{dB}]$$

$$\text{SNR} = \frac{\mu_I}{\sigma_I} [\text{dB}], \text{S/MSE} = 10 \log_{10} \frac{\sum_i \sum_j \hat{X}(p)^2}{\sum_i \sum_j (I(p) - \hat{X}(p))^2} [\text{dB}]$$

Ocenjevanje nivoja pegastega šuma v slikah SAR:  $\text{ENL} = (\frac{\mu_I}{\sigma_I})^2 = \frac{1}{\sigma_N^2} = L$  (šum ima povprecje 1 ( $\mu_N = 1$ ))

**Adaptivni mediana filter** - 1. Definiramo vecjo masko  $H_1$  in manjšo masko  $H_2$ . 2. Vse piksele manjše od praga  $T$  filtriramo z masko  $H_1$ , preostale pa z masko  $H_2$ . 3. Korak 2 lahko izvedemo 2x.

**Kuanov filter** - predpostavke so: 1. multiplikativni šum:  $I(p) = X(p) + (N(P) - 1)X(p)$ , 2. šum ima povprecje 1.

$$\text{Postopek: } \hat{X}(p) = \mu_I + \frac{\sigma_X^2(I(p) - \mu_I)}{\sigma_X^2 + (\mu_I + \sigma_X^2)/L}, \sigma_X^2 = \frac{L\sigma_I^2 - \mu_I^2}{L+1}, \text{ izredni primer: } \sigma_X^2 < 0, \text{ potem } \hat{X}(p) = \mu_I.$$

$$\text{Leejev (MMSE) filter: } \hat{X}(p) = \mu_I + \frac{\sigma_X^2(I(p) - \mu_I)}{\sigma_X^2 + \mu_I/L} \quad \text{Filter gamma: } \hat{X}(p) = \frac{(\alpha - L - 1)\mu_I + \sqrt{\mu_I(\alpha - L - 1)^2 + 4\alpha L \mu_I I(p)}}{2\alpha}, \alpha = \frac{L+1}{L(\sigma_I/\mu_I)^2 - 1}$$

**Frostov filter:**  $h = e^{-KC_I(t_0)\|t\|}$ ,  $C_I = \frac{\sigma_I}{\mu_I}$  (K - parameter filtra (ponavadi 1),  $t_0$  je lokacija piksla,  $\|t\|$  je razdalja do  $t_0$ ).

**Oddyjev filter (3x3):** if  $m < \alpha \mu_I$  then  $\hat{X}(p) = \mu_I$ , else  $\hat{X}(p) = \frac{\sum_k \sum_l W_{kl} I(k,l)}{\sum_k \sum_l W_{kl}}$

kjer if  $|I(k, l) - I(p)| < m$  then  $W_{kl} = 1$ , else  $W_{kl} = 0$ .  $m = \frac{1}{8} \sum_k \sum_l |I(k, l) - I(p)|$ .  $\alpha$  je podan (npr. 1).

**Filter AFS** - Postopek: 1. Vrednost l.e.s (ploščina) izračunamo za 9 binarnih mask znotraj okna velikosti 5x5. 2.

Izberemo masko z minimalno vrednostjo l.e.s. 3. Filtriramo z nizkim sitom izbrano masko. 4. Dobljeno vrednost priredimo središnjemu pikslu okna, velikosti 5x5.

## BARVNA VZTRAJNOST

**Model za formiranje barvnih vrednosti piksla:**  $C_v(p) = \int_{\lambda} S_v(\lambda) E(\lambda, p) d\lambda$  ( $\lambda$ -valovna dolžina, p - piksel, v-barvni kanal)

Poenostavitev zgornje enačbe:  $C_v(p) = G(p)R_v(p)L_v(p)$  (R-odbojnosc predmetov, L-spekter svetlobe, G-geom. faktor)

**Algoritem WPR:**  $L = \max(C(p))$ , nato:  $O(p) = G(p)R(p) = \frac{C(p)}{L}$

**Algoritem GWA:**  $O(p) = \frac{C(p)}{\overline{C}\varphi}$  ( $\overline{C}$  - povp. vrednost barvnega kanala,  $\varphi$  - faktor povečave(npr. 2))

**Algoritem CCN** - iteriramo: 1.  $O'(p) = \frac{C(p)}{C_R(p) + C_G(p) + C_B(p)}$ , 2.  $O(p) = \frac{O'(p)}{\overline{C}\varphi}$  3.  $C(p) = O(p)$

**Algoritem LSAC:**  $O(p) = \frac{C(p)}{L(p)} = \frac{C(p)}{\overline{C}(p)\varphi}$ ,  $\overline{C}(p) = C(p) * K(p)$ ,  $K(p) = K(x, y) = a^{-1}e^{-\frac{x^2+y^2}{2p^2}}$ ,  $p = 0.093\max(M, N)$  (K-jedro, MxN slika)

**Algoritem RSR:** P mnozic, vsaka vsebuje K pikslov.  $p_{p,k} = [i_{p,k}, j_{p,k}]$ -k-ti piksel p-te mnozice.

$i_{p,k} = i + r\eta \cos(\mu)$ ,  $j_{p,k} = i + r\eta \sin(\mu)$ ,  $n \in U(0, 1)$ ,  $\mu \in U(0, 2\pi)$ ,  $p = [i, j]$ , r-radij razpršene množice.

$$O(p) = \frac{1}{P} \sum_{p=1}^P \frac{C(p)}{\max_k(C(p_{p,k}))}$$

## POSPLOŠENA HOUGHOVA TRANSFORMACIJA

Splošna analitična krivulja:  $F(x, a) = 0$ , kjer x - piksel, a - vektor parametrov krivulje

Postopek HT za iskanje linij oz. premic	Postopek HT za iskanje krožnic, na osnovi robnih pikslov
<p><b>Klasična enačba premice:</b> <math>y = kx + n</math></p> <p><b>Parametrična oblika enačbe:</b> <math>x \cos(\theta) + y \sin(\theta) = r</math></p> <p>Pri <math>\theta</math> izračunamo oddaljenost <math>r</math> za piksel <math>p = [i, j]</math>:  <math>r = i \cos(\theta) + j \sin(\theta)</math> (enacba 2)</p> <p>1. Inicializiraj akumulator A, tj. <math>A(\theta, r) = 0</math></p> <p>2. FOR vsak robni piksel:  Izračunaj enačbo (2)  <math>A(\theta, r) = A(\theta, r) + \Delta A</math></p> <p>3. Poisči lokalne maksimume v akumulatorju A, kar ustreza položajem premic v sliki.</p>	<p><b>Klasična enačba krožnice:</b> <math>(x - a)^2 + (y - b)^2 = r^2</math></p> <p>1. Inicializiraj akumulator A, tj. <math>A(a, b, r) = 0</math></p> <p>2. FOR vsak robni piksel  FOR <math>a = \min_a : \Delta a : \max_x</math>  FOR <math>b = \min_b : \Delta b : \max_y</math>  <math>r = \sqrt{(x - a)^2 + (y - b)^2}</math>  <math>A(a, b, r) = A(a, b, r) + \Delta A</math></p> <p><b>HT elipse, na osnovi robnih pikslov - enačba elipse:</b>  <math>\frac{(x - x_0)^2}{a^2} + \frac{(y - y_0)^2}{b^2} = 1</math> (4 params, 3 FOR-loops),  z dodano rotacijo po x osi pa:  <math>x(t) = x_0 + a \cos(t) \cos(\theta) - b \sin(t) \sin(\theta)</math> in  <math>y(t) = y_0 + a \cos(t) \sin(\theta) + b \sin(t) \cos(\theta)</math> (5 params, 4 loops)</p>

HT za analitične krivulje v sivinskih slikah, na osnovi vseh informacij iz slike robov (jakost + smer gradijenta)	HT za iskanje elips v sivinskih slikah, s pomočjo informacije o smeri gradijenta
<p>Originalna enačba krivulje <math>F(x, a) = 0</math> (enacba 3)  Njen odvod: <math>\frac{dF}{dx}(x, a) = 0</math> (enacba 4)  V odvodu dobimo znan clen <math>\frac{dy}{dx} = \tan[\psi(x) - \frac{\pi}{2}]</math>  Postopek:  1. Inicializiraj akumulator A, tj. <math>A(a) = 0</math>  2. FOR vsak robni piksel      Izračunaj parametre a, za katere velja (3) in (4)  <math>A(a) = A(a) + \Delta A</math>  3. Poisci lokalne maksimume v akumulatorju A.</p>	<p>Enačba elipse: <math>\frac{(x-x_0)^2}{a^2} + \frac{(y-y_0)^2}{b^2} = 1</math> (enacba 5)  <math>X = x - x_0</math> in <math>Y = y - y_0</math>, odvajamo po X  in dobimo <math>\frac{2X}{a^2} + \frac{2Y}{b^2} \frac{dY}{dX}</math>  Vrednost <math>\frac{dY}{dX}</math> poznamo, oznamo <math>\frac{dY}{dX} = \xi</math>  Izračunamo: <math>Y = \pm \sqrt{\frac{b^2}{1+\frac{a^2}{b^2}\xi^2}}</math> in <math>X = \pm \sqrt{\frac{a^2}{1+\frac{b^2}{a^2}\xi^2}}</math>  Dolocimo: <math>y_0 = y \pm \sqrt{\frac{b^2}{1+\frac{a^2}{b^2}\xi^2}}</math> in <math>x_0 = x \pm \sqrt{\frac{a^2}{1+\frac{b^2}{a^2}\xi^2}}</math>  <math>\xi = \tan(\phi - \varphi - \frac{\pi}{2})</math> (enacba 9)  Postopek:  1. Inicializiraj akumulator A,  tj. <math>A(x_0, y_0, \varphi, a, b) = 0</math>  2. FOR vsak robni piksel      FOR <math>a = \min_a : \Delta a : \max_a</math>      FOR <math>b = \min_b : \Delta b : \max_b</math>      FOR <math>\varphi = \min_\varphi : \Delta \varphi : \max_\varphi</math>          Izračunaj enacbo 9          Doloci X in Y po (7), pri cemer doloci          predznak na osnovi opazovanja dY in dX.          Rotiraj X in Y za kot <math>\varphi</math>          <math>x_0 = x + X</math> in <math>y_0 = y + Y</math>          <math>A(x_0, y_0, \varphi, a, b) = A(x_0, y_0, \varphi, a, b) + \Delta A</math>  3. Poisci lokalne maksimume v akumulatorju A, kar ustreza položajem elips v sliki.</p>

Pospositev HT za neanalitische krivulje - vektor parametrov  $a = [y, s, \varphi]$ , kjer  $y = [x_0, y_0]$  - ref. izhodisce za krivuljo,  $\varphi$  - orientacija krivulje,  $s = [s_x, s_y]$  - ortog. skalirna faktorja (v praksi enaka), R-tabela=tabela moznih orientacij robnih pikslov (za opis ref. izhodisca  $y$ ).

Postopek kreiranja R-tabele	Oblika R-tabele
1. Izberi referencno točko y znotraj iskane oblike. 2. FOR vsako točko x iz meje Izračunaj smer gradijenta $\psi(x)$ $\vec{r} = \vec{y} - \vec{x}$ Shrani r v R-tabelo v odvisnosti od $\psi(x)$ .	$i \mid \psi_i \mid R_{\psi_i}$
0   0   $\{r y - r = x \wedge \psi(x) = 0 \wedge x \text{ je na meji}\}$	
1   $\Delta\psi$   $\{r y - r = x \wedge \psi(x) = \Delta\psi \wedge x \text{ je na meji}\}$	
2   $2\Delta\psi$   $\{r y - r = x \wedge \psi(x) = 2\Delta\psi \wedge x \text{ je na meji}\}$	
...   ...   ...	
Lastnosti:	
1. Skaliranje oblike za faktor s: $R'(\psi) = sR(\psi)$ (10)	
2. Rot. za kot $\varphi$ : $R'(\psi) = \text{Rotacija}\{R[(\psi - \varphi) \bmod 2\pi], \varphi\}$ (11)	
3. Sprememba ref. točke iz y v y': $R'(\psi) = R(\psi) + (y - y')$	
Alternativne inkrementalne strategije: $A(a) = A(a) +  \text{grad}(x) $ ali $A(a) = A(a) +  \text{grad}(x)  + \text{konstanta}$	

## MODELI AKTIVNIH KONTUR

Položaj kače:  $v(s) = [x(s), y(s)]$ , kjer je s proporcionalen dolzini loka, tj.  $s \in [0, 1]$ . **Energijski funkcional kace:**  $E_{\text{snake}} = \int_0^1 (E_{\text{int}}(v(s)) + E_{\text{image}}(v(s)) + E_{\text{con}}(v(s))) ds$ , kjer  $E_{\text{int}}$  - interna energija zlepka zaradi upogibanja,  $E_{\text{image}}$  - slikovne sile,  $E_{\text{con}}$  - sile zunanjih omejitev.  $E_{\text{int}} = \frac{1}{2}(\alpha(s)|v_s(s)|^2 + \beta(s)|v_{ss}(s)|^2)$ , kjer  $v_s$  - prvi odvod v po s (clen 1. reda),  $v_{ss}$  - drugi odvod v po s (clen 2. reda),  $\alpha$  in  $\beta$  - utezi (ponavadi  $\alpha = \beta$ ).  $E_{\text{image}} = w_{\text{line}}E_{\text{line}} + w_{\text{edge}}E_{\text{edge}} + w_{\text{term}}E_{\text{term}}$  (13), kjer  $E_{\text{line}} = I(x, y)$  - energ. funkcional, ki pritegne kačo k linijam.  $E_{\text{edge}} = -|\Delta I(x, y)|^2$  - energ. funk., ki pritegne kačo k robovom. Zglajena slika:  $C(x, y) = G_\sigma * I(x, y)$ . Torej je **energ. funk. zaključkov**:  $E_{\text{term}} = \frac{C_{yy}C_x^2 - 2C_{xy}C_xC_y + C_{xx}C_y^2}{(C_x^2 + C_y^2)^{\frac{3}{2}}}$ .  $C_x, C_y, C_{xx}, C_{yy}$  in  $C_{xy}$  so parcialni odvodi. **Diskretizacija energije kače:**  $E_{\text{snake}} = \sum_{i=0}^n E_{\text{int}}(i) + E_{\text{ext}}(i)$ . i-to vozlisce kace:  $v_i = [x_i, y_i] = [x(ih), y(ih)]$ .  $E_{\text{int}}$  postane:  $E_{\text{int}}(i) = \alpha_i \frac{|v_i - v_{i-1}|^2}{2h^2} + \beta_i \frac{|v_{i-1} - 2v_i + v_{i+1}|^2}{2h^4}$ . Naj bo  $F_x(i) = \frac{\partial E_{\text{ext}}}{\partial x_i}$  in  $F_y(i) = \frac{\partial E_{\text{ext}}}{\partial y_i}$ . Dobimo Eulerjevo enacbo:  $\alpha_i(v_i - v_{i-1}) - \alpha_{i+1}(v_{i+1} - v_i) + \beta_{i-1}(v_{i-2} - 2v_{i-1} + v_i) - 2\beta_i(v_{i-1} - 2v_i + v_{i+1}) + \beta_{i+1}(v_i - 2v_{i+1} + v_{i+2}) + (F_x(i), F_y(i)) = 0$ . To zapisemo v matricni notaciji:  $Ax + F_x(x, y) = 0$  in  $Ay + F_y(x, y) = 0$  (x in y sta n-dim. vektorja x in y koord. kače). To resimo kot  $x_t = (A + \gamma I)^{-1}(\gamma x_{t-1} - F_x(x_{t-1}, y_{t-1}))$  in  $y_t = (A + \gamma I)^{-1}(\gamma y_{t-1} - F_y(x_{t-1}, y_{t-1}))$  (enacbi 16).  $\gamma$  je velikost koraka (pogosto 1).

Pentadiagonalna matrika A		Algoritem kac
$\begin{bmatrix} cba0000\dots0ab \\ bcba000\dots00a \\ abcba00\dots000 \\ 0abcba0\dots000 \\ \dots\dots\dots \\ 000000\dots abca \\ 000000\dots abcb \\ 000000\dots abc \\ \end{bmatrix}$	$a = \beta$ $b = -(\alpha + 4/\beta)$ $c = 2\alpha + 6\beta$  $zgornje velja, ce$ $\alpha_i = \alpha$ in $\beta_i = \beta$	1. Vnesi zacetni priblizek (krivuljo) iskanega objekta. 2. Enakomerno razporedi n vozlišč po krivulji. 3. Določi $E_{\text{ext}}$ po enacbi (13). 4. Izračunaj $F_x = \frac{\partial E_{\text{ext}}}{\partial x}$ in $F_y = \frac{\partial E_{\text{ext}}}{\partial y}$ 5. Doloci matriko A 6. WHILE (ni izpolnjen konvergenci pogoj) Doloci nov položaj vozlisc kace v iteraciji t po enacbi (16). $t = t + 1$ Enakomerno razporedi n vozlišč po krivulji.

Iskanje ujemanja v stero slikah - dodatni energijski funkcional:  $E_{\text{stereo}} = (v^L(s) - v^R(s))^2$