

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 enacb za vsak piksel (n -st. koncnih članov, e_i -i-ti koncni član, ς -spekter, p-piksel):

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

ODSTRANJEVANJE PEGASTEŠA 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, $p = [i, j]$

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

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

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

Adaptivni mediana filter - 1. Definiramo vecjo maske H_1 in manjšo masko H_2 . 2. Vse piksele manjse 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.

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

Leejev (MMSE) filter: $\hat{X}(p) = \mu_I + \frac{\sigma_X^2(I(p)-\mu_I)}{\sigma_X^2+\mu_I/L}$ 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 $W_{kl} = 0$. $m = \frac{1}{8} \sum_k \sum_l |I(k,l) - I(p)|$. α 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 središčnemu pikslu okna, velikosti 5x5.

BARVNA VZTRAJNOST

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

Poenostavitev zgornje enačbe: $C_v(p) = G(p)R_v(p)L_v(p)$ (R-odbojnost 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: $O(p) = \frac{C(p)}{\bar{C}_\varphi}$ (\bar{C} - povp. vrednost barvnega kanala, φ - faktor povečave(npr. 2))

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

Algoritem LSAC: $O(p) = \frac{C(p)}{L(p)} = \frac{C(p)}{\bar{C}(p)_\varphi}$, $\bar{C}(p) = C(p) * K(p)$, $K(p) = K(x, y) = a^{-1} e^{-\frac{x^2+y^2}{2p^2}}$, $p = 0.093 max(M, N)$ (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>Klasicna enačba premice: $y = kx + n$</p> <p>Parametrična oblika enačbe: $x cos(\theta) + y sin(\theta) = r$</p> <p>Pri θ izračunamo oddaljenost r za piksel $p = [i, j]$:</p> <p>$r = i cos(\theta) + j sin(\theta)$ (enačba 2)</p> <p>1. Inicializiraj akumulator A, tj. $A(\theta, r) = 0$</p> <p>2. FOR vsak robni piksel:</p> <p> Izračunaj enačbo (2)</p> <p> $A(\theta, r) = A(\theta, r) + \Delta A$</p> <p>3. Poišči lokalne maksimume v akumulatorju A, kar ustreza položajem premic v sliki.</p>	<p>Klasična enačba krožnice: $(x - a)^2 + (y - b)^2 = r^2$</p> <p>1. Inicializiraj akumulator A, tj. $A(a, b, r) = 0$</p> <p>2. FOR vsak robni piksel</p> <p> FOR $a = min_a : \Delta a : max_x$</p> <p> FOR $b = min_b : \Delta b : max_b$</p> <p> $r = \sqrt{(x - a)^2 + (y - b)^2}$</p> <p> $A(a, b, r) = A(a, b, r) + \Delta A$</p> <hr/> <p>HT elipse, na osnovi robnih pikslov - enačba elipse:</p> <p>$\frac{(x-x_0)^2}{a^2} + \frac{(y-y_0)^2}{b^2} = 1$ (4 params, 3 FOR-loops),</p> <p>z dodano rotacijo po x osi pa:</p> <p>$x(t) = x_0 + a cos(t) cos(\theta) - b sint sin(\theta)$ in</p> <p>$y(t) = y_0 + a cos(t) cos(\theta) + b sint sin(\theta)$ (5 params, 4 loops)</p>

HT za analitične krivulje v sivinskih slikah, na osnovi vseh informacij iz slike robov (jakost + smer gradienta)	HT za iskanje elips v sivinskih slikah, s pomočjo informacije o smeri gradienta
<p>Originalna enačba krivulje $F(x, a) = 0$ (enacba 3)</p> <p>Njen odvod: $\frac{dF}{dx}(x, a) = 0$ (enacba 4)</p> <p>V odvodu dobimo znan člen $\frac{dy}{dx} = tg[\psi(x) - \frac{pi}{2}]$</p> <p>Postopek:</p> <ol style="list-style-type: none"> 1. Inicializiraj akumulator A, tj. $A(a) = 0$ 2. FOR vsak robni piksel <p>Izračunaj parametre a, za katere velja (3) in (4)</p> $A(a) = A(a) + \Delta A$ 3. Poiisci lokalne maksimume v akumulatorju A. 	<p>Enačba elipse: $\frac{(x-x_0)^2}{a^2} + \frac{(y-y_0)^2}{b^2} = 1$ (enacba 5)</p> <p>$X = x - x_0$ in $Y = y - y_0$, odvajamo po X</p> <p>in dobimo $\frac{2X}{a^2} + \frac{2Y}{b^2} \frac{dY}{dX}$</p> <p>Vrednost $\frac{dY}{dX}$ poznamo, označimo $\frac{dY}{dX} = \xi$</p> <p>Izračunamo: $Y = \pm \frac{b^2}{\sqrt{1 + \frac{a^2}{b^2} \xi^2}}$ in $X = \pm \frac{a^2}{\sqrt{1 + \frac{b^2}{a^2} \xi^2}}$</p> <p>Dolocimo: $y_0 = y \pm \frac{b^2}{\sqrt{1 + \frac{a^2}{b^2} \xi^2}}$ in $x_0 = x \pm \frac{a^2}{\sqrt{1 + \frac{b^2}{a^2} \xi^2}}$</p> <p>$\xi = tg(\phi - \varphi - \frac{\pi}{2})$ (enacba 9)</p> <p>Postopek:</p> <ol style="list-style-type: none"> 1. Inicializiraj akumulator A, tj. $A(x_0, y_0, \varphi, a, b) = 0$ 2. FOR vsak robni piksel <p>FOR $a = min_a : \Delta a : max_a$</p> <p>FOR $b = min_b : \Delta b : max_b$</p> <p>FOR $\varphi = min_\varphi : \Delta \varphi : max_\varphi$</p> <p>Izračunaj enacbo 9</p> <p>Doloci X in Y po (7), pri cemer doloci predznak na osnovi opazovanja dY in dX.</p> <p>Rotiraj X in Y za kot φ</p> <p>$x_0 = x + X$ in $y_0 = y + Y$</p> <p>$A(x_0, y_0, \varphi, a, b) = A(x_0, y_0, \varphi, a, b) + \Delta A$</p> 3. Poiisci lokalne maksimume v akumulatorju A, kar ustreza položajem elips v sliki.

Posplošitev HT za neanalitične krivulje - vektor parametrov $a = [y, s, \varphi]$, kjer $y = [x_0, y_0]$ - ref. izhodišče za krivuljo, φ - orientacija krivulje, $s = [s_x, s_y]$ - ortog. skalirna faktorja (v praksi enaka), R-tabela=tabela možnih orientacij robnih pikselov (za opis ref. izhodišča y).

Postopek kreiranja R-tabele	Oblika R-tabele															
<div>1. Izberi referenčno točko y znotraj iskane oblike.</div> <div>2. FOR vsako točko x iz meje</div> <div> Izračunaj smer gradienta $\psi(x)$</div> <div> $\vec{r} = \vec{y} - \vec{x}$</div> <div> Shrani r v R-tabelo v odvisnosti od $\psi(x)$.</div> <div>Postopek iskanja poljubnih oblik:</div> <div>1. Inicializiraj akumulator A, tj. $A(y, s, \varphi) = 0$</div> <div>2. FOR vsak robni piksel x</div> <div> FOR $s = min_s : \Delta s : max_s$</div> <div> FOR $\varphi = min_\varphi : \Delta \varphi : max_\varphi$</div> <div> Doloci R' po enacbi (10).</div> <div> Vstavi R' v enacbo (11) in doloci R''.</div> <div> FOR $r \in R''(\psi(x))$</div> <div> $A(x + r, s, \varphi) = A(x + r, s, \varphi) + \Delta A$</div> <div>3. Poiisci lokalne maksimume v akumulatorju A</div>	<table><tr><th>i</th><th>ψ_i</th><th>R_{ψ_i}</th></tr><tr><td>0</td><td>0</td><td>$\{r y - r = x \wedge \psi(x) = 0 \wedge x \text{ je na meji} \}$</td></tr><tr><td>1</td><td>$\Delta\psi$</td><td>$\{r y - r = x \wedge \psi(x) = \Delta\psi \wedge x \text{ je na meji} \}$</td></tr><tr><td>2</td><td>$2\Delta\psi$</td><td>$\{r y - r = x \wedge \psi(x) = 2\Delta\psi \wedge x \text{ je na meji} \}$</td></tr><tr><td>...</td><td>...</td><td>...</td></tr></table> <div>Lastnosti:</div> <div>1. Skaliranje oblike za faktor s: $R'(\psi) = sR(\psi)$ (10)</div> <div>2. Rot. za kot φ: $R'(\psi) = Rotacija\{R[(\psi - \varphi) \bmod 2\pi], \varphi\}$ (11)</div> <div>3. Sprememba ref. točke iz y v y': $R'(\psi) = R(\psi) + (y - y')$</div> <div>Alternativne inkrementalne strategije: $A(a) = A(a) + grad(x)$</div> <div>ali $A(a) = A(a) + grad(x) + konstanta$</div>	i	ψ_i	R_{ψ_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	$2\Delta\psi$	$\{r y - r = x \wedge \psi(x) = 2\Delta\psi \wedge x \text{ je na meji} \}$
i	ψ_i	R_{ψ_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	$2\Delta\psi$	$\{r y - r = x \wedge \psi(x) = 2\Delta\psi \wedge x \text{ je na meji} \}$														
...														

MODELI AKTIVNIH KONTUR

Položaj kače: $v(s) = [x(s), y(s)]$, kjer je s proporcionalen dolžini loka, tj. $s \in [0, 1]$. **Energijski funkcional kace:** $E_{snake} = \int_0^1 (E_{int}(v(s)) + E_{image}(v(s)) + E_{con}(v(s)))ds$, kjer E_{int} - **interna energija zlepka zaradi upogibanja**, E_{image} - **slikovne sile**, E_{con} - **sile zunanjih omejitev**. $E_{int} = \frac{1}{2}(\alpha(s)|v_s(s)|^2 + \beta(s)|v_{ss}(s)|^2)$, kjer v_s - prvi odvod v po s (clen 1. reda), v_{ss} - drugi odvod v po s (clen 2. reda), α in β - uteži (ponavadi $\alpha = \beta$). $E_{image} = w_{line}E_{line} + w_{edge}E_{edge} + w_{term}E_{term}$ (13), kjer $E_{line} = I(x, y)$ - energ. funkcional, ki pritegne kačo k linijam. $E_{edge} = -|\Delta I(x, y)|^2$ - energ. funk., ki pritegne kačo k robovom. Zglajena slika: $C(x, y) = G_\sigma * I(x, y)$. Torej je **energ. funk. zaključkov**: $E_{term} = \frac{C_{yy}C_x^2 - 2C_{xy}C_xC_y + C_{xx}C_y^2}{(C_x^2 + C_y^2)^{\frac{3}{2}}}$. C_x, C_y, C_{xx}, C_{yy} in $C_{x,y}$ so parcialni odvodi. **Diskretizacija energije kače:** $E_{snake} = \sum_{i=0}^n E_{int}(i) + E_{ext}(i)$. i -to vozlišče kace: $v_i = [x_i, y_i] = [x(ih), y(ih)]$. E_{int} postane: $E_{int}(i) = \alpha_i \frac{|v_i - v_{i-1}|^2}{2h^2} + \beta_i \frac{|v_{i-1} - 2v_i + v_{i+1}|^2}{2h^4}$. Naj bo $F_x(i) = \frac{\partial E_{ext}}{\partial x_i}$ in $F_y(i) = \frac{\partial E_{ext}}{\partial y_i}$. 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 matrični 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). γ je velikost koraka (pogosto 1).

Pentadiagonalna matrika A		Algoritem kac
$\begin{bmatrix} cba0000\dots0ab \\ bcb000\dots00a \\ abcba00\dots000 \\ 0abcba0\dots000 \\ \dots\dots\dots \\ 000000\dots abcba \\ 000000\dots0abcb \\ 000000\dots00abc \end{bmatrix}$	<p>$a = \beta$</p> <p>$b = -(\alpha + 4/\beta)$</p> <p>$c = 2\alpha + 6\beta$</p> <p>zgornje velja, ce</p> <p>$\alpha_i = \alpha$ in</p> <p>$\beta_i = \beta$</p>	<ol style="list-style-type: none"> 1. Vnesi zacetni priblizek (krivuljo) iskanega objekta. 2. Enakomerno razporedi n vozlišč po krivulji. 3. Doloci E_{ext} po enacbi (13). 4. Izračunaj $F_x = \frac{\partial E_{ext}}{\partial x}$ in $F_y = \frac{\partial E_{ext}}{\partial y}$ 5. Doloci matriko A 6. WHILE (ni izpolnjen konvergenčni pogoj) <p>Doloci nov položaj vozlišc kace v iteraciji t po enacbi (16).</p> <p>$t = t + 1$</p> <p>Enakomerno razporedi n vozlišc po krivulji.</p>

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