11. Principy počítačového vidění

V této přednášce se zaměříme výhradně na zpracování a rozpoznávání černobílých snímků představovaných maticemi $m \times n$ pixelů.

Nechť každý pixel matice může nabývat k možných hodnot jasové funkce (úrovní šedi) a dále nechť označuje:

$I(u,v)$	aktuální hodnotu jasové funkce pixelu na pozici (u,v),
max	maximální hodnotu jasové funkce,
práh	zvolenou hodnotu jasové funkce.

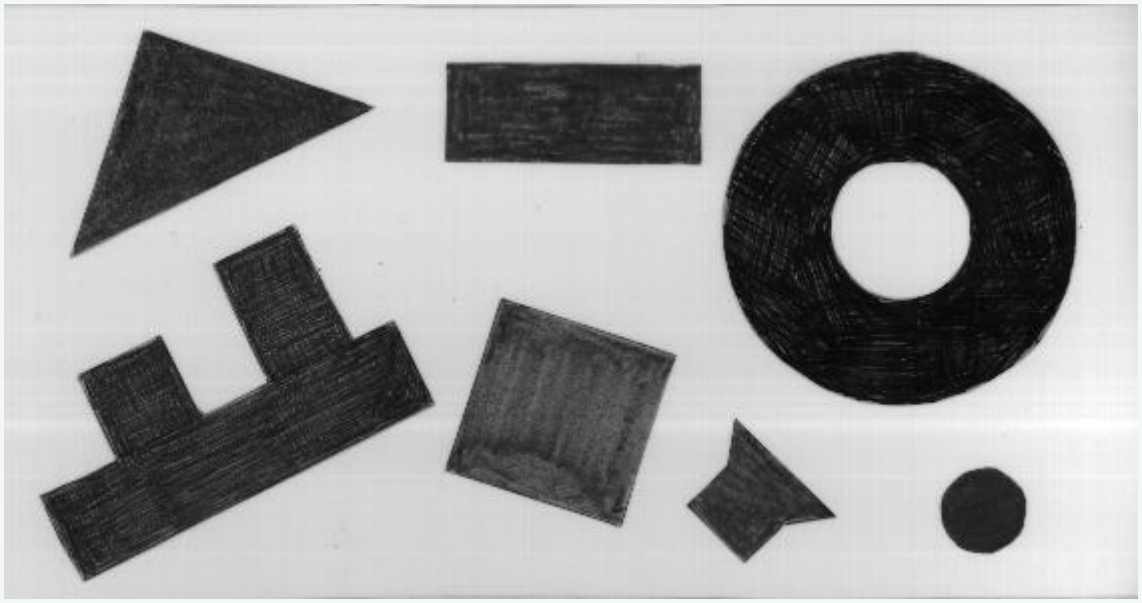
Monadické (jednobodové) transformace:

Inverze: $I(u,v) = max - I(u,v)$

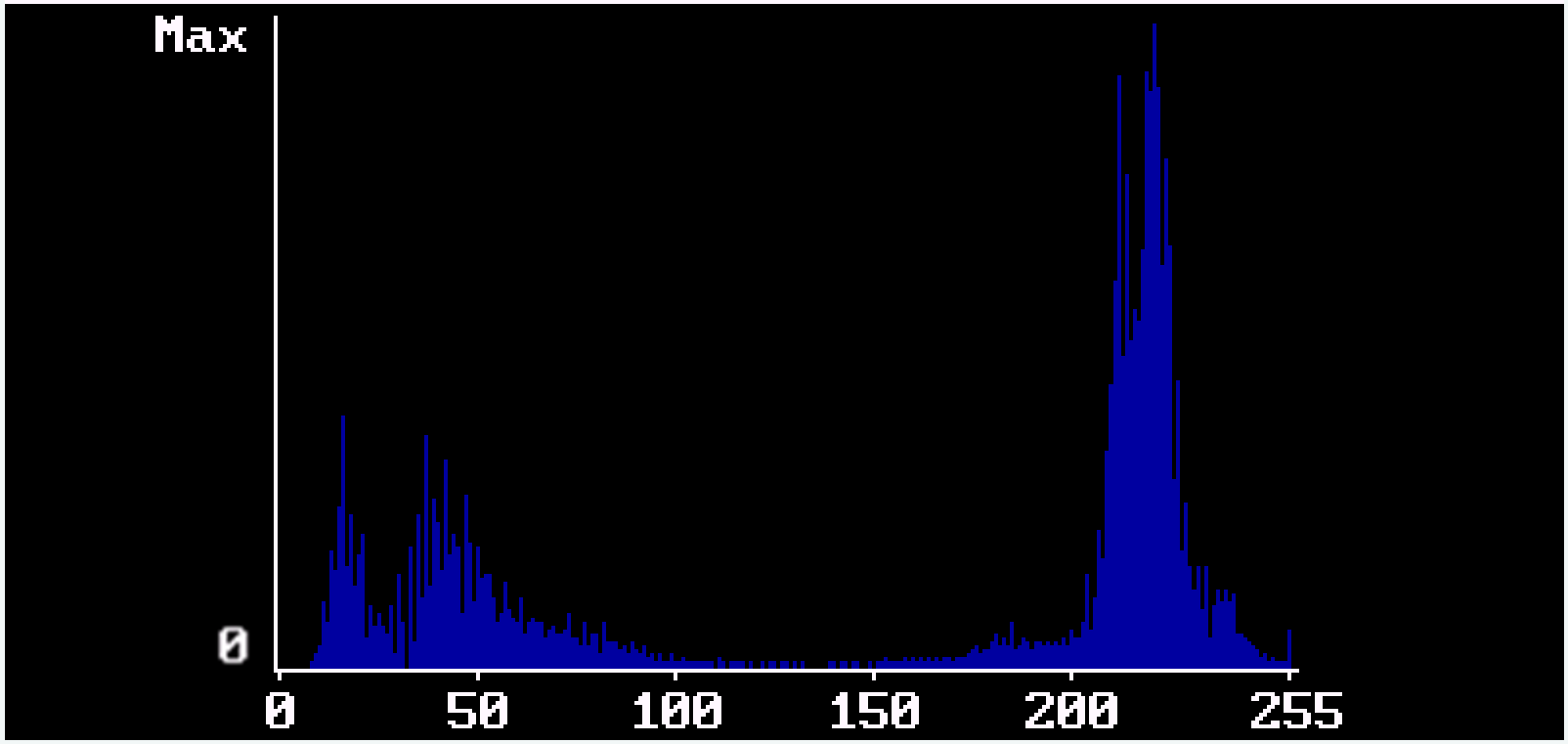
Prahování: $I(u,v) = 0$ *pro* $I(u,v) \leq práh$

$= max$ *pro* $I(u,v) > práh$

Příklad: Snímané 2D objekty



Histogram (počet pixelů jako funkce jasu):



Výsledek po prahování – práh nastaven na hodnotu 128

Dyadické (dvoubodové) transformace:

$$I(u,v) = f(I_1(u,v), I_2(u,v))$$

Součet obrazů (redukce šumu):

$$I(u,v) = f((I_1(u,v) + I_2(u,v))$$

Rozdíl obrazů (detekce změn):

$$I(u,v) = f(I_1(u,v) - I_2(u,v))$$

Funkce f transformuje výslednou hodnotu do intervalu přípustných hodnot jasové funkce (například do intervalu <0, 255>).

Násobení obrazů (*násobení obrazu maskou*):

- korekce nelinearit senzorů:

$$I(u,v) = f((I(u,v)\cdot m(u,v)) + I(u,v))$$

Maska reálných čísel m má stejnou dimenzi jako obraz, funkce f transformuje reálnou hodnotu na nejbližší vyšší celočíselnou hodnotu s omezením maximální hodnoty.

- realizace oken:

$$I(u,v) = I(u,v)\cdot m(u,v)$$

Prvky masky m jsou jedničky (v okně) nebo nuly (mimo okno).

Prostorové transformace (filtry, konvoluce):

$$I(u,v) = \sum_{i=-a}^{b} \sum_{j=-c}^{d} h(i,j) I(u-i, v-j),$$

$$n = a + b + 1, \quad m = c + d + 1, \quad n = m = 3,\ 5,\ \ldots$$

Dolní propusti:

$$h(i,j) \geq 0, \quad \sum h(i,j) = 1$$

Horní propusti:

$$\sum h(i,j) = 0$$

Dále budeme pro jednoduchost uvažovat pouze masky filtrů h o rozměrech 3 x 3, tj.:

$$I(u,v) = \sum_{i=-1}^{1} \sum_{j=-1}^{1} h(i,j) I(u-i, v-j),$$

Dolní propusti se používají především k redukci šumu.

Box filtr (hodnoty $h(i,j)$ jsou stejné):

obraz se šumem

7	7	7	7	7	7
7	7	7	7	7	7
7	7	5	7	7	7
7	7	7	7	7	7
7	7	7	7	7	7

filtr

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

výsledný obraz

x	x	x	x	x	x
x	7	7	7	7	x
x	7	7	7	7	x
x	7	7	7	7	x
x	x	x	x	x	x

Přepočtené hodnoty barevně označených pixelů:

$1/9 * (8 * 7 + 1 * 5) = 61/9 = 6.777 \rightarrow 7$

obraz se šumem

7	7	7	7	7	7
7	7	7	7	7	7
7	7	9	7	7	7
7	7	7	7	7	7
7	7	7	7	7	7

filtr

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

výsledný obraz

x	x	x	x	x	x
x	7	7	7	7	x
x	7	7	7	7	x
x	7	7	7	7	x
x	x	x	x	x	x

Přepočtené hodnoty barevně označených pixelů:

$1/9 * (8 * 7 + 1 * 9) = 65/9 = 7{,}222 \rightarrow 7$

obraz se šumem

7	7	7	7	7	7
7	7	7	7	7	7
7	7	12	7	7	7
7	7	7	7	7	7
7	7	7	7	7	7

filtr

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

výsledný obraz

x	x	x	x	x	x
x	8	8	8	7	x
x	8	8	8	7	x
x	8	8	8	7	x
x	x	x	x	x	x

Přepočtené hodnoty barevně označených pixelů:

$1/9 * (8 * 7 + 1 * 12) = 68/9 = 7{,}555 \rightarrow 8$

Gaussův filtr méně ovlivňuje sousední pixely (lépe koresponduje s dvourozměrnou diskrétní Gaussovou funkcí):

obraz se šumem

7	7	7	7	7	7
7	7	7	7	7	7
7	7	12	7	7	7
7	7	7	7	7	7
7	7	7	7	7	7

filtr

$1/16 *$

1	2	1
2	4	2
1	2	1

výsledný obraz

x	x	x	x	x	x
x	7	8	7	7	x
x	8	8	8	7	x
x	7	8	7	7	x
x	x	x	x	x	x

Přepočtené hodnoty barevně označených pixelů:

$1/16 * (12 * 7 + 4 * 12) = 132/16 = 8{,}25 \rightarrow 8$

$1/16 * (14 * 7 + 2 * 12) = 122/16 = 7{,}625 \rightarrow 8$

$1/16 * (15 * 7 + 1 * 12) = 112/16 = 7{,}0 \rightarrow 7$

Lineární filtry však kromě odstraňování šumu vyhlazují obraz jako celek, tj. potlačují hrany a obrysy objektů na snímku. Proto se k redukci šumu používají častěji nelineární filtry, například mediánový filtr.

Mediánový filtr (seřadí hodnoty v daném okně a původní hodnotu středového pixelu nahradí mediánem):

obraz se šumem

7	7	7	7	7	7
7	6	7	8	7	7
7	7	12	7	7	7
7	7	5	7	7	7
7	7	7	7	7	7

5
6
7
7
7
7
7
8
12

← medián

výsledný obraz

x	x	x	x	x	x
x	7	7	7	7	x
x	7	7	7	7	x
x	7	7	7	7	x
x	x	x	x	x	x

Horní propusti se používají k detekci a zvýrazňování hran (na hranách se totiž výrazně mění hodnoty jasové funkce, a právě horní propusti tyto změny detekují)

Sobelův operátor:

a	***b***	***c***			
d	***e***	***f***			
g	***h***	***i***			

1	0	-1
2	0	-2
1	0	-1

1	2	1
0	0	0
-1	-2	-1

	e				

$$I(\boldsymbol{e}) = (\Delta I_x(\boldsymbol{e})^2 + \Delta I_y(\boldsymbol{e})^2)^{1/2} =$$
$$= (((I(\boldsymbol{a}) + 2I(\boldsymbol{d}) + I(\boldsymbol{g})) - (I(\boldsymbol{c}) + 2I(\boldsymbol{f}) + I(\boldsymbol{i})))^2 +$$
$$+ \quad ((I(\boldsymbol{a}) + 2I(\boldsymbol{b}) + I(\boldsymbol{c})) - (I(\boldsymbol{g}) + 2I(\boldsymbol{h}) + I(\boldsymbol{i})))^2)^{1/2}$$

Pro orientaci hrany procházející pixelem ***e*** zřejmě platí

$$\varphi(\boldsymbol{e}) = \arctan(\Delta I_x(\boldsymbol{e})/\Delta I_y(\boldsymbol{e}))$$

Sobelův operátor (detekce hran):

7	7	7	7	7	7
7	7	7	7	7	7
7	7	7	7	7	7
7	7	7	7	7	7
7	7	7	7	7	7

1	0	-1
2	0	-2
1	0	-1

1	2	1
0	0	0
-1	-2	-1

x	x	x	x	x	x
x	0	0	0	0	x
x	0	0	0	0	x
x	0	0	0	0	x
x	x	x	x	x	x

Pro všechny pixely v obrázku bez hran platí $\Delta I_x = \Delta I_y$, a proto jejich nové hodnoty jsou nulové.

Sobelův operátor (detekce hran):

4	7	7	7	7	7
4	4	7	7	7	7
4	4	4	7	7	7
4	4	4	4	7	7
4	4	4	4	4	7

1	0	-1
2	0	-2
1	0	-1

1	2	1
0	0	0
-1	-2	-1

x	x	x	x	x	x
x	13	13	4	0	x
x	4	13	13	4	x
x	0	4	13	13	x
x	x	x	x	x	x

Přepočtené hodnoty barevně označených pixelů:

$\Delta I_x = -9 \quad \Delta I_y = 9 \quad (\Delta I_x^2 + \Delta I_y^2)^{1/2} = 12{,}73 \rightarrow 13$

$\Delta I_x = -3 \quad \Delta I_y = 3 \quad (\Delta I_x^2 + \Delta I_y^2)^{1/2} = 4{,}24 \rightarrow 4$

$\Delta I_x = 0 \quad \Delta I_y = 0 \quad (\Delta I_x^2 + \Delta I_y^2)^{1/2} = 0 \rightarrow 0$

Úhel hrany ve všech nenulových pixelech $\varphi = \arctan(\Delta I_x / \Delta I_y) = \arctan(-1) = -\pi/4 = -45° \Rightarrow$ hrana je přímka.

Laplaceův operátor (zvýraznění hran):

7	7	7	7	7	7
7	7	7	7	7	7
7	7	7	7	7	7
7	7	7	7	7	7
7	7	7	7	7	7

0	1	0
1	-4	1
0	1	0

x	x	x	x	x	x
x	0	0	0	0	x
x	0	0	0	0	x
x	0	0	0	0	x
x	x	x	x	x	x

13	13	4	0	0	0
4	13	13	4	0	0
0	4	13	13	4	0
0	0	4	13	13	4
0	0	0	4	13	13

0	1	0
1	-4	1
0	1	0

x	x	x	x	x	x
x	-18	-18	10	8	x
x	10	-18	-18	10	x
x	8	10	-18	-18	x
x	x	x	x	x	x

Detekce rohů

Předchozí filtry/operátory byly založeny na skutečnosti, že na hranách se mění intenzita jasové funkce v jednom směru.

Pro detekci rohů může být použit podobný přístup, ale nyní se musí detekovat výrazné změny jasové funkce současně v několika směrech.

Nechť I_x značí první parciální derivaci jasové funkce I ve směru osy x

$$I_x(u,v) = \frac{\partial I}{\partial x}(u,v) \cong \frac{\Delta I_x(u,v)}{\Delta x} = \frac{I(u+1,v) - I(u-1,v)}{2}$$

I_y první parciální derivaci jasové funkce ve směru osy y

$$I_y(u,v) = \frac{\partial I}{\partial y}(u,v) \cong \frac{\Delta I_y(u,v)}{\Delta y} = \frac{I(u,v+1) - I(u,v-1)}{2}$$

a dále nechť

$$\overline{A} = I_x^2, \quad \overline{B} = I_y^2, \quad \overline{C} = I_x I_y$$

Po vyhlazení předchozích tří funkcí (obvykle Gaussovým filtrem) je označíme symboly A, B, C a definujeme strukturální matici

$$M = \begin{bmatrix} A & C \\ C & B \end{bmatrix}$$

jejíž vlastní čísla λ_1 a λ_2 se získají z rovnice

$$\begin{bmatrix} A-\lambda & C \\ C & B-\lambda \end{bmatrix} = 0 \quad \Rightarrow (A-\lambda)(B-\lambda) - C^2 = 0 \quad \Rightarrow$$

$$\lambda^2 - (A+B)\lambda + (AB - C^2) = 0 \quad \Rightarrow$$

$$\lambda_{1,2} = \frac{1}{2}\left(A+B \pm \sqrt{(A+B)^2 - 4(AB - C^2)} \right)$$

Pokud jsou obě hodnoty vlastních čísel $\lambda_{1,2}$ pro pixel (u,v) malé, je oblast okolo tohoto pixelu plochá, pokud je hodnota λ_1 velká a hodnota λ_2 malá (opačně to být nemůže), může pixel (u,v) náležet hraně a pokud jsou obě hodnoty $\lambda_{1,2}$ velké, stává se pixel (u,v) kandidátem na rohový pixel.

Je proto zřejmé, že rozdíl obou vlastních čísel

$$\lambda_1 - \lambda_2 = \sqrt{(A+B)^2 - 4(AB - C^2)}$$

by měl být pro rohové pixely malý.

Harrisův detektor vychází z tohoto předpokladu a pro hledání rohů používá funkci

$$Q = \left(AB - C^2\right) - \alpha\left(A + B\right)^2$$

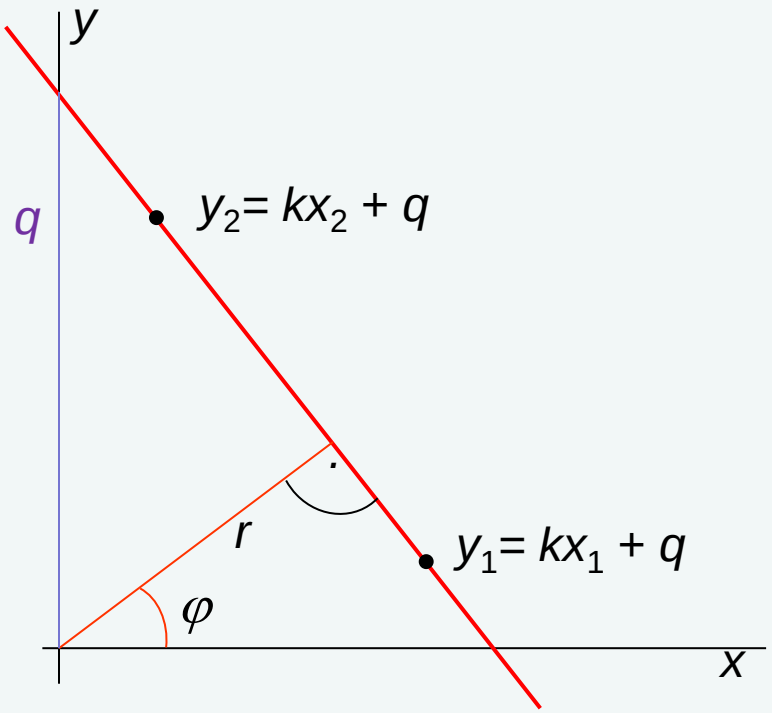
která vrací největší hodnoty právě pro rohové pixely. Koeficient α se doporučuje volit z intervalu (0.04, 0.06) s tím, že čím vyšší je jeho hodnota, tím je detektor méně citlivý.

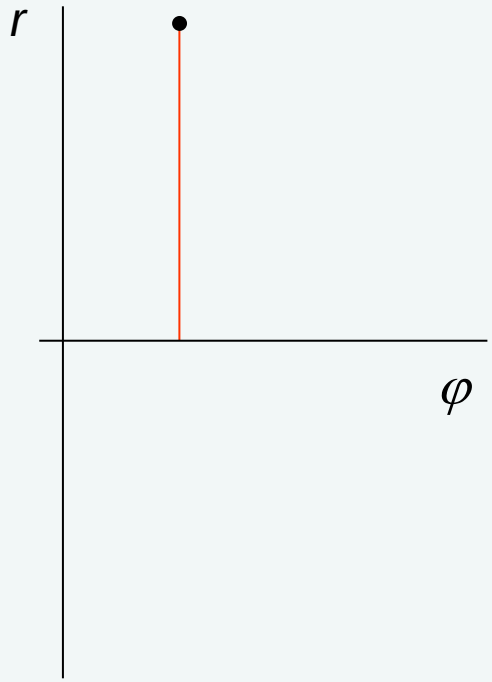
Činnost Harrisova detektoru je následující:

- Každý pixel (u,v) snímku, pro který je hodnota $Q(u,v)$ větší než zvolený práh, je vybrán za kandidáta na rohový pixel.
- Všichni kandidáti na rohový pixel se umístí do množiny kandidátů, ze které se následně vyřazují všechny pixely, které mají nižší hodnocení, než jiné pixely v jejich topologickém sousedství.
- Zbylí kandidáti v množině kandidátů jsou pak považováni za skutečné rohové pixely.

Detekce přímek

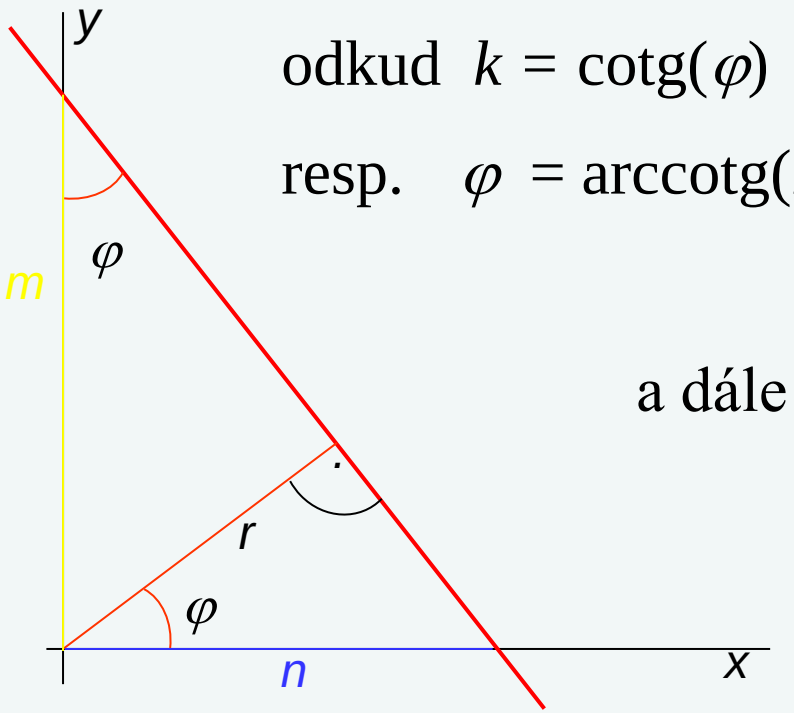
Všechny body přímky v prostoru souřadnic (x,y) musí vyhovovat rovnici $y = kx + q$, zatímco všechny body této přímky v prostoru parametrů (r,φ) představují jediný bod (Houghova transformace).





Platí (resp. lze odvodit) následující vztahy:

$$y = kx + q = -\frac{m}{n}x + m = -\frac{\frac{r}{\sin(\varphi)}}{\frac{r}{\cos(\varphi)}}x + \frac{r}{\sin(\varphi)} = \operatorname{cotg}(\varphi)x + \frac{r}{\sin(\varphi)}$$



odkud $k = \operatorname{cotg}(\varphi)$ a $q = r/\sin(\varphi)$

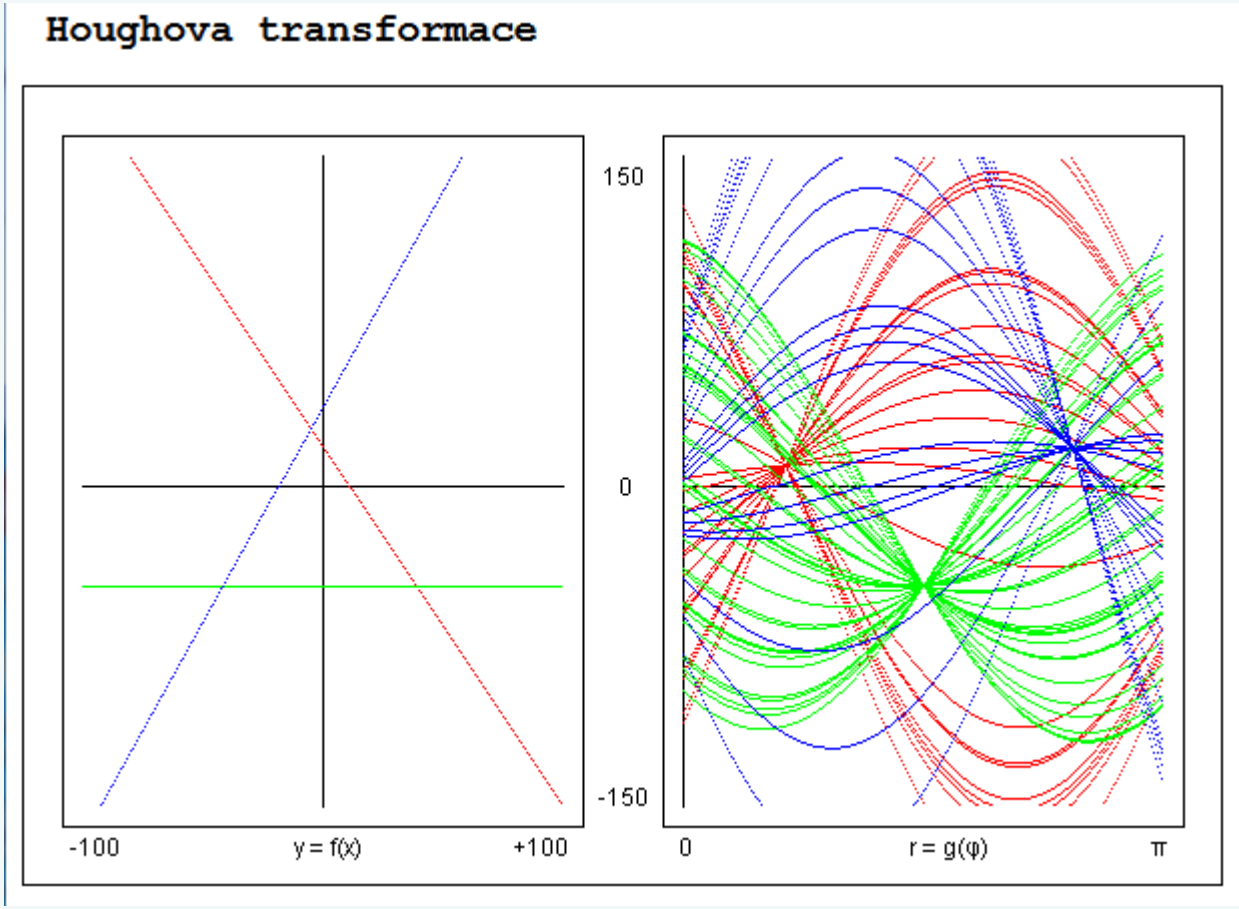
resp. $\varphi = \operatorname{arccotg}(k)$ a $r = q\,\sin(\operatorname{arccotg}(k))$

a dále $y = -\frac{m}{n}x + m \quad \Rightarrow \quad \frac{y}{m} = -\frac{x}{n} + 1 \quad \Rightarrow$

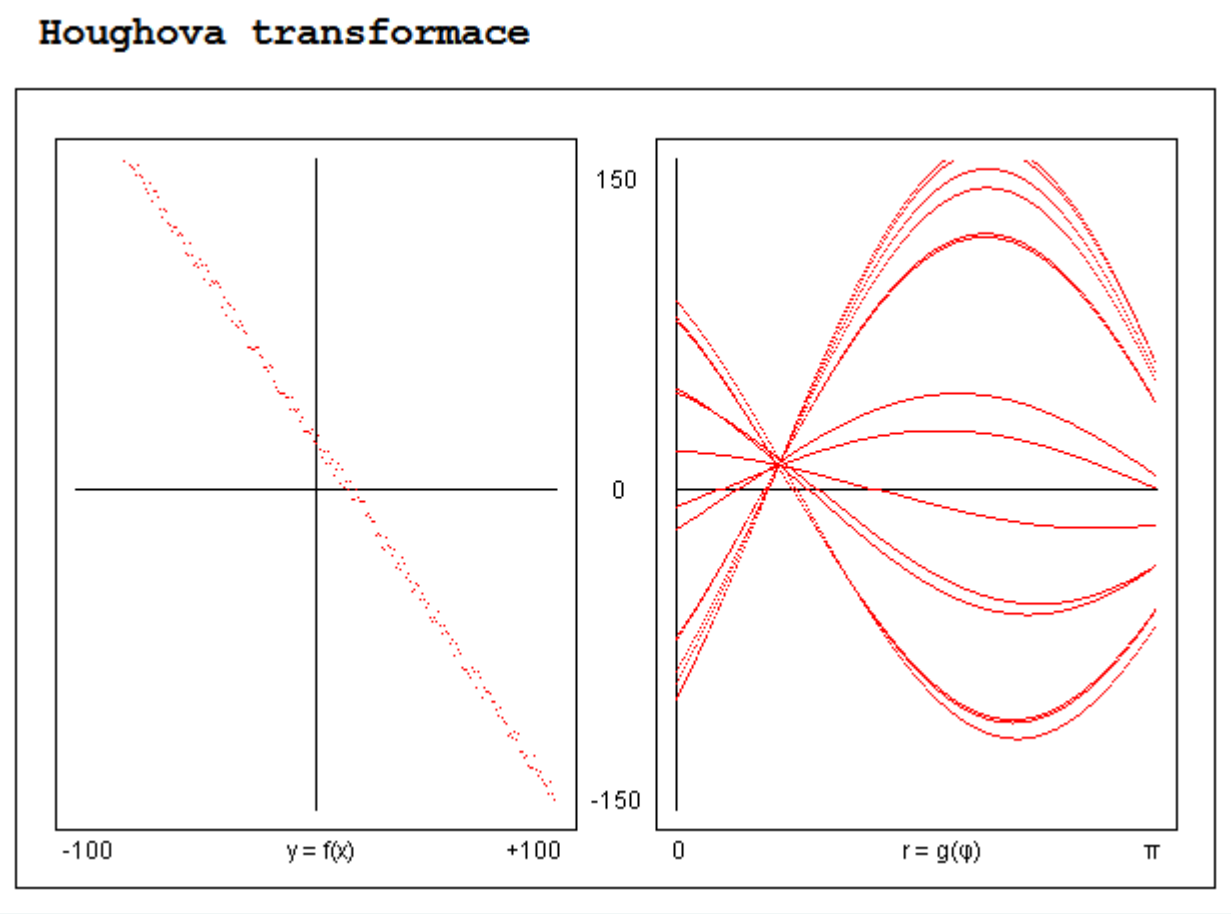
$$y\frac{\sin(\varphi)}{r} + x\frac{\cos(\varphi)}{r} = 1 \quad \Rightarrow$$

$$r = y\sin(\varphi) + x\cos(\varphi)$$

Zobrazení tří přímek v prostoru souřadnic a zobrazení některých jejich bodů v prostoru parametrů (všechny křivky reprezentující různé body jedné přímky se protínají vždy v jednom bodu!).



Zobrazení „rozmazané“ přímky



Algoritmus HoughLines(I)

- Vytvořte dvourozměrné pole střádačů Acc[r,φ] a všechny jeho prvky vynulujte;
- Pro všechny pixely (u,v) snímku I provádějte
  - Je-li (u,v) pixelem hrany pak
    (x,y) = (u - u_c,v - v_c), kde (u_c,v_c) značí střed snímku I
    - Pro $\varphi_i = 0 \ldots \pi$ počítejte $r_i = x\cos(\varphi_i) + y\sin(\varphi_i)$

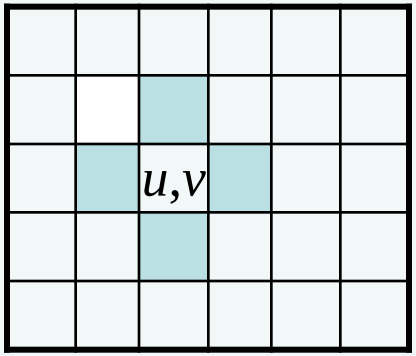
      a inkrementujte hodnoty střádačů Acc[r_i,φ_i]
- Vraťte seznam parametrů (r_i,φ_i) odpovídající nejvýraznějším nalezeným přímkám.

Pozn.: Podobným (ale značně složitějším) způsobem se postupuje při detekci kružnic a elips.

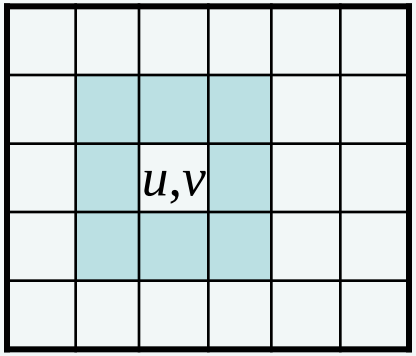
Metrické a topologické vlastnosti snímku

Sousedé pixelu (u,v)

4-okolí ($N_4(u,v)$): $(u,v+1),(u,v-1),(u+1,v),(u-1,v)$



8-okolí ($N_8(u,v)$): 4-okolí + $(u-1,v+1),(u-1,v-1),(u+1,v+1),(u+1,v-1)$



Spojitost pixelů ($V = \{g_i,\ldots,g_m\}$ je množina úrovní šedi použitá pro definici spojitosti):

4-spojitost: Dva pixely (s,t) a (u,v) jsou spojité jestliže

$$((u,v) \in N_4(s,t)) \wedge (I(s,t) \in V) \wedge (I(u,v) \in V)$$

8-spojitost: Dva pixely (s,t) a (u,v) jsou spojité jestliže

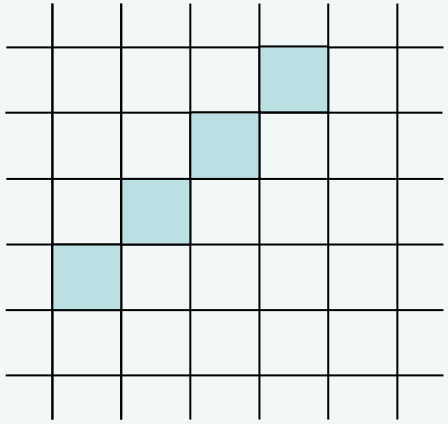
$$((u,v) \in N_8(s,t)) \wedge (I(s,t) \in V) \wedge (I(u,v) \in V)$$

Plocha (oblast):

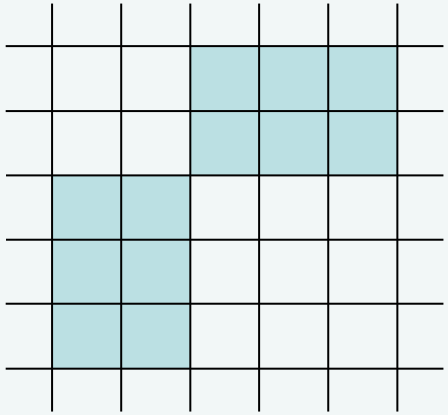
4-spojitou plochu/oblast tvoří dva 4-spojité pixely a dále pak všechny další pixely, které mají v této ploše alespoň jednoho 4-spojitého souseda.

8-spojitou plochu/oblast tvoří dva 8-spojité pixely a dále pak všechny další pixely, které mají v této ploše alespoň jednoho 8-spojitého souseda.

Problémy:



v D_4 nespojitá křivka (přímka) !



v D_8 souvislá plocha !

Hranice plochy (oblasti):

Hraniční pixel 4-spojité plochy má alespoň jednoho 4-souseda uvnitř a alespoň jednoho 4-spojitého souseda vně této plochy.

Hraniční pixel 8-spojité plochy má alespoň jednoho 8-souseda uvnitř a alespoň jednoho 8-spojitého souseda vně této plochy.

Cesta:

4-spojitá cesta z pixelu (s,t) do pixelu (u,v) je posloupnost různých pixelů se souřadnicemi (x_0,y_0), (x_1,y_1),…, (x_n,y_n), pro kterou platí $(x_0,y_0) = (s,t)$, $(x_n,y_n) = (u,v)$ a pro každé $i \in \langle 1,n \rangle$ dále platí, že pixel (x_i,y_i) je 4-spojitý s pixelem (x_{i-1},y_{i-1}).

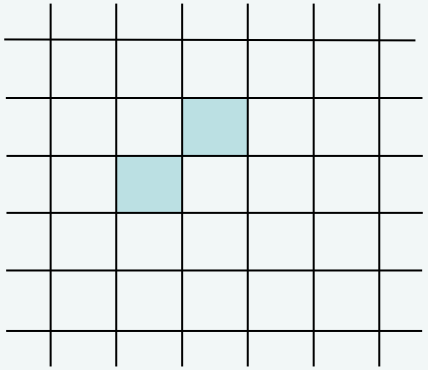
8-spojitá cesta z pixelu (s,t) do pixelu (u,v) je posloupnost různých pixelů se souřadnicemi (x_0,y_0), (x_1,y_1),…, (x_n,y_n), pro kterou platí $(x_0,y_0) = (s,t)$,$(x_n,y_n) = (u,v)$ a pro každé $i \in \langle 1,n \rangle$ dále platí, že pixel (x_i,y_i) je 8-spojitý s pixelem (x_{i-1},y_{i-1}).

Délka cesty: počet pixelů cesty – 1 (t.j. délka cesty = n).

Vzdálenosti pixelů (s,t), (u,v):

$D((s,t),(u,v)) > 0,$
$D((s,t),(u,v)) = D((u,v),(s,t)),$
$D((s,t),(u,v)) \leq D((s,t),(w,z)) + D((w,z),(u,v))$

$D_E((s,t),(u,v)) = Sqrt((s-u)^2 + (t-v)^2)$ (Euklidovská)
$D_4((s,t),(u,v)) = |s-u| + |t-v|$ (4-okolí)
$D_8((s,t),(u,v)) = max(|s-u|,|t-v|)$ (8-okolí)



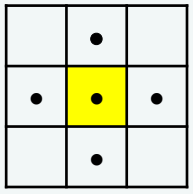
$D_E = \sqrt{2}$
$D_4 = 2$
$D_8 = 1$

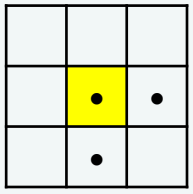
Morfologické operace na dvojúrovňových (binárních) snímcích

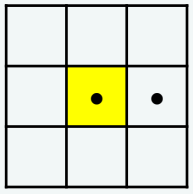
Označme symbolem I množinu všech bodů $\boldsymbol{p}$ snímku, pro které platí $I(\boldsymbol{p}) = 1$ a podobně symbolem H množinu všech bodů $\boldsymbol{q}$ masky, pro které platí $H(\boldsymbol{q}) = 1$. Čtyři používané morfologické operace pak lze popsat takto:

- Dilatace $I \oplus H \equiv \{\boldsymbol{p} + \boldsymbol{q} \mid \text{pro } \boldsymbol{p} \in I \text{ a současně } \boldsymbol{q} \in H\}$
- Erose $I \ominus H \equiv \{\boldsymbol{p} \mid \text{pro } (\boldsymbol{p} + \boldsymbol{q}) \in I \text{ a každé } \boldsymbol{q} \in H\}$
- Otevření $I \circ H = (I \ominus H) \oplus H$
- Uzavření $I \bullet H = (I \oplus H) \ominus H$

Morfologické masky H mohou být symetrické i nesymetrické, vždy však musí být zřejmý tzv. hot spot (výchozí bod masky). Příklady:

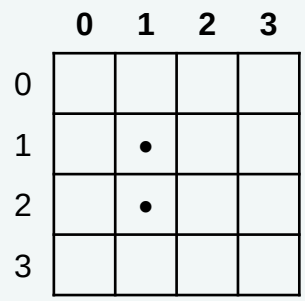
$H =$ 

$H =$ 

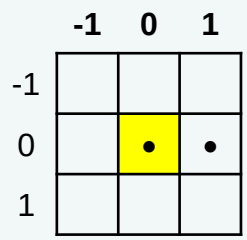
$H =$ 

Příklad dilatace:

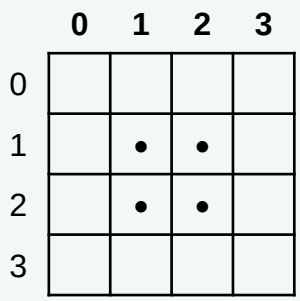
I



H



$I \oplus H$

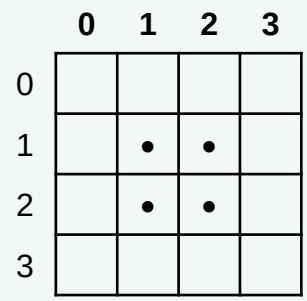


$I \equiv \{(1,1), (2,1)\}, \qquad H \equiv \{(0,0), (0,1)\}$

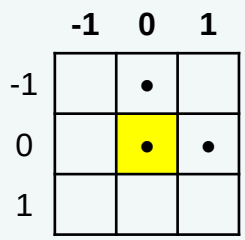
$$I \oplus H = \{(1,1) + (0,0), (1,1) + (0,1), (2,1) + (0,0), (2,1) + (0,1)\} = \{(1,1), (1,2), (2,1), (2,2)\}$$

Příklad erose:

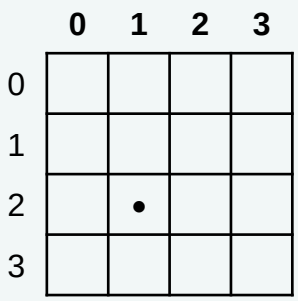
I



H



$I \ominus H$

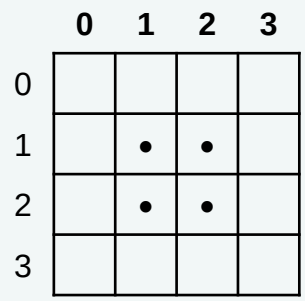


$I \equiv \{(1,1), (1,2), (2,1), (2,2)\}$, $H \equiv \{(-1,0), (0,0), (0,1)\}$

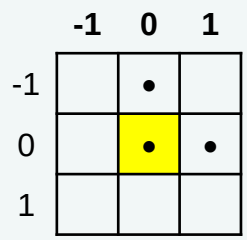
$I \ominus H = \{(2,1)\}$, protože pouze $(2,1) + (-1,0) = (1,1) \in I$ a současně
$(2,1) + (0,0) = (2,1) \in I$ a současně
$(2,1) + (0,1) = (2,2) \in I$.

Příklad otevření:

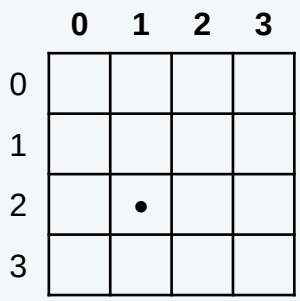
I



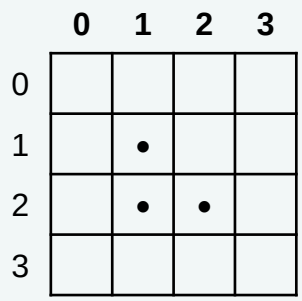
H



$I \ominus H$



$I \circ H$

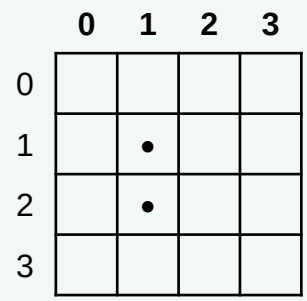


$I \ominus H \equiv \{(2,1)\}, \quad H \equiv \{(-1,0), (0,0), (0,1)\}$

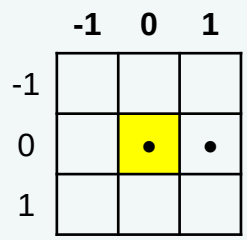
$$I \circ H = (I \ominus H) \oplus H = \{(2,1) + (-1,0), (2,1) + (0,0), (2,1) + (0,1)\} = \{(1,1), (2,1), (2,2)\}$$

Příklad uzavření:

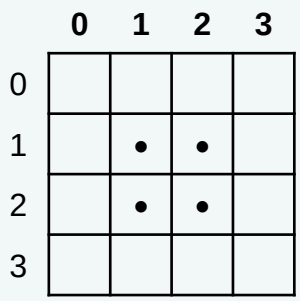
I



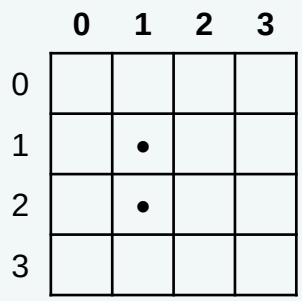
H



$I \oplus H$



$I \bullet H$



$I \oplus H \equiv \{(1,1), (1,2), (2,1), (2,2)\}, \quad H \equiv \{(0,0), (0,1)\}$

$I \bullet H = (I \oplus H) \ominus H =$ $\{(1,1), (2,1)\}$, protože

$(1,1) + (0,0) = (1,1) \in I \oplus H$ and

$(1,1) + (0,1) = (1,2) \in I \oplus H$ and

$(2,1) + (0,0) = (2,1) \in I \oplus H$ and

$(2,1) + (0,1) = (2,2) \in I \oplus H$

Analýza scény z dvojúrovňových (binárních) černobílých snímků

- Dekompozice obrazů
- Rozpoznání obrazů

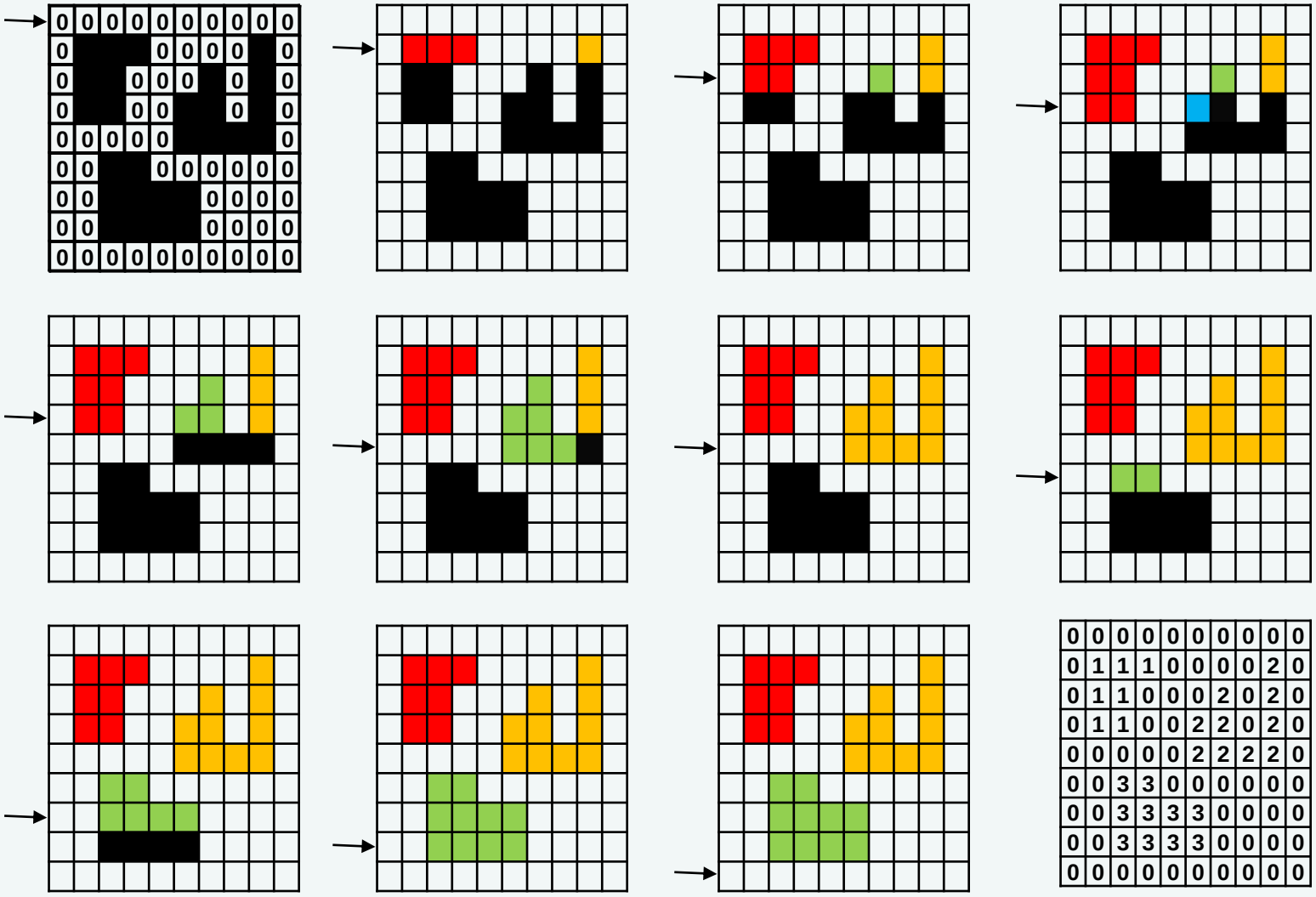
Dekompozice obrazů (tzv. barvení běhů) se provádí takto:

1. Zvolí se výchozí roh snímku, například horní levý a nastaví se index objektu na hodnotu 1.
2. Postupně pro každý řádek (shora dolů)
    a. Prochází se a „přebarvují“ se jednotlivé pixely tohoto řádku (zleva doprava):
        i. Pokud je hodnota aktuálního pixelu nulová a předchozí pixel měl hodnotu větší než nula, tak se inkrementuje index objektu.
        ii. Pokud je hodnota pixelu rozdílná od nuly, tak:
            - Pokud je horním sousedem aktuálního pixelu již obarvený pixel, tak se hodnota aktuálního pixelu a dále

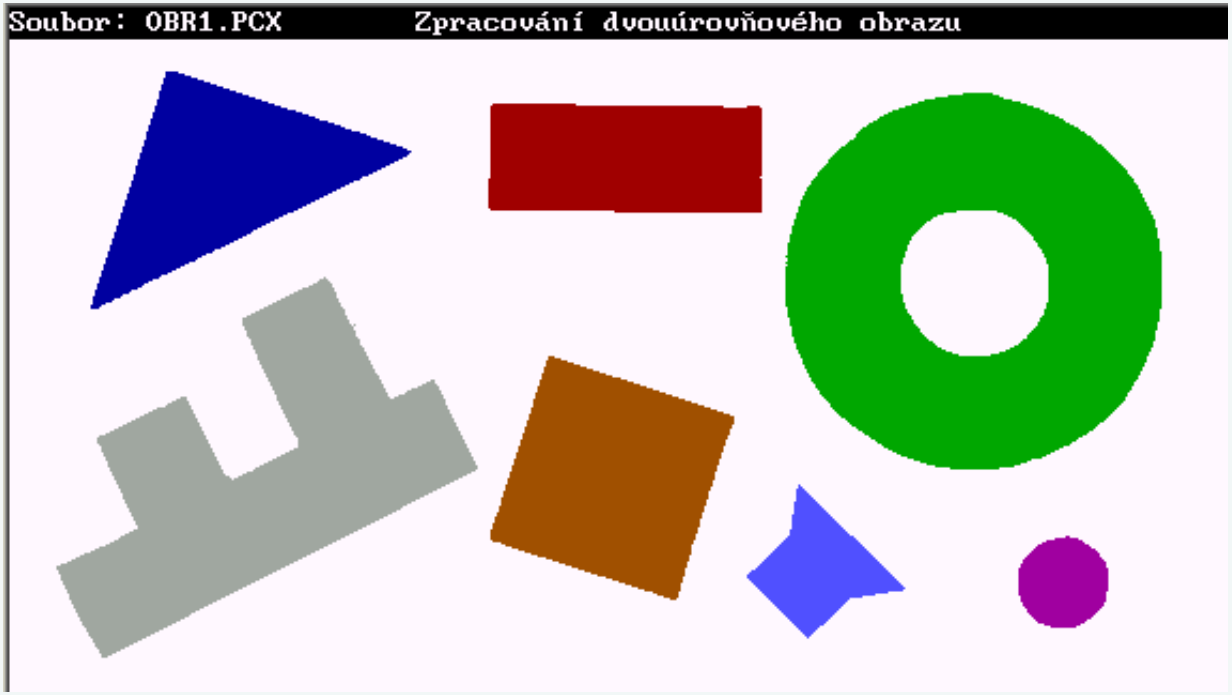
všech pixelů plochy, do které tento aktuální pixel patří, nastaví na hodnotu horního souseda a pokračuje se v procházení řádku (od bodu a.).

- Pokud je hodnota levého souseda aktuálního pixelu rozdílná od nuly, pak se hodnota aktuálního pixelu nastaví na hodnotu tohoto levého souseda a pokračuje se v procházení řádku (od bodu a).
- Pokud je hodnota levého souseda aktuálního pixelu nulová, tak se hodnota aktuálního pixelu nastaví na hodnotu indexu objektu a pokračuje se v procházení řádku (od bodu a).

Dekompozice obrazů:



Obarvené obrazy na snímku po dekompozici



Rozpoznávání obrazů

Příznaky pro rozpoznávání jednotlivých obrazů:

- Plocha (počet pixelů)
- Délka hranice (počet pixelů)
- Počet děr
- Plochy děr (počet pixelů)
- Kompaktnost (délka hranice)^2/plocha
- Minimální a maximální vzdálenosti hranice od těžiště
- Průměrná vzdálenost hranice od těžiště
- Poměr ploch děr k celkové ploše
- Momenty

Obecné momenty:

$$m_{pq} = \sum_x \sum_y x^p y^q I(x,y) \qquad I(x,y) \in \{0,1\}$$

Centrální momenty (jsou invariantní vůči posunutí):

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q I(x,y)$$

$$\bar{x} = \frac{m_{10}}{m_{00}}, \qquad \bar{y} = \frac{m_{01}}{m_{00}},$$

Normalizované centrální momenty (jsou invariantní i vůči změně velikosti):

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\frac{p+q}{2}+1}}$$

RST (Rotate/Scale/Translate) invariantní momenty (jsou invariantní i vůči natočení)

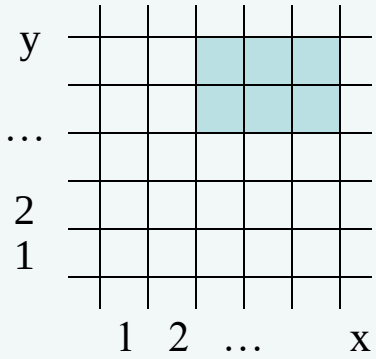
- druhé momenty:

$$\phi_1 = \eta_{20} + \eta_{02}$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$

- pro třetí/smíšené momenty je známo dalších pět vztahů.

Praktický výpočet:



$$m_{00} = \mu_{00} = 6 \qquad \text{(počet pixelů obrazu)}$$

$$\bar{x} = \frac{\sum x}{6} = \frac{24}{6} = 4$$

$$\bar{y} = \frac{\sum y}{6} = \frac{27}{6} = 4.5$$

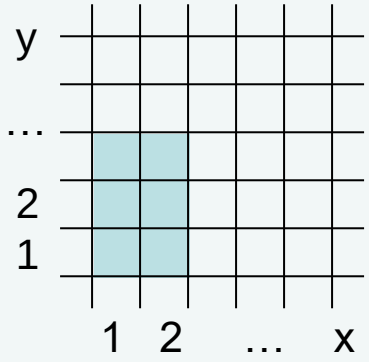
$$\eta_{20} = \frac{\mu_{20}}{\mu_{00}^2} = \frac{\sum (x - \bar{x})^2}{36} = \frac{2*(-1)^2 + 2*(0)^2 + 2*(1)^2}{36} = \frac{1}{9}$$

$$\eta_{02} = \frac{\mu_{02}}{\mu_{00}^2} = \frac{\sum (y - \bar{y})^2}{36} = \frac{3*(-0.5)^2 + 3*(0.5)^2}{36} = \frac{1}{24}$$

$$\eta_{11} = \frac{\mu_{11}}{\mu_{00}^2} = \frac{\sum\sum (x - \bar{x})(y - \bar{y})}{36} = 0$$

$$\phi_1 = \eta_{20} + \eta_{02} = \frac{1}{9} + \frac{1}{24} = \frac{11}{72} = 0.153$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 = \left(\frac{1}{9} - \frac{1}{24}\right)^2 + 0 = 4.82*10^{-3}$$



$$m_{00} = \mu_{00} = 6$$

$$\bar{x} = \frac{\sum x}{6} = \frac{9}{6} = 1.5$$

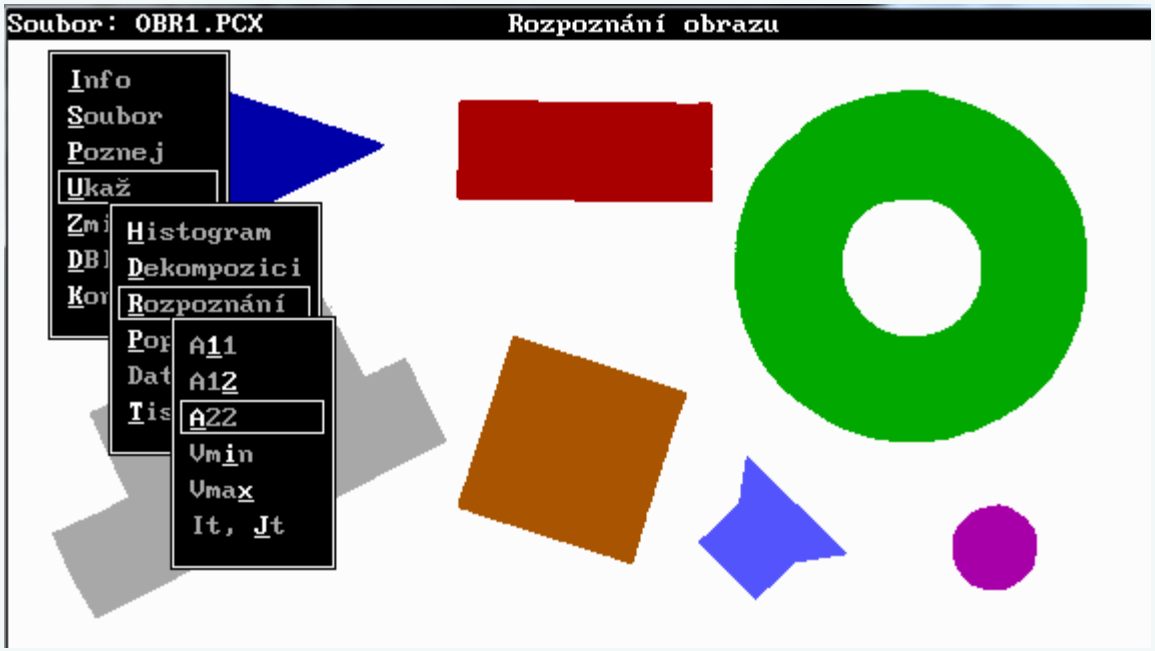
$$\bar{y} = \frac{\sum y}{6} = \frac{12}{6} = 2$$

$$\eta_{20} = \frac{\mu_{20}}{\mu_{00}^2} = \frac{\sum (x - \bar{x})^2}{36} = \frac{3 * (-0.5)^2 + 3 * (0.5)^2}{36} = \frac{1}{24}$$

$$\eta_{02} = \frac{\mu_{02}}{\mu_{00}^2} = \frac{\sum (y - \bar{y})^2}{36} = \frac{2 * (-1)^2 + 2 * (0)^2 + 2 * (1)^2}{36} = \frac{1}{9}$$

$$\eta_{11} = \frac{\mu_{11}}{\mu_{00}^2} = \frac{\sum\sum (x - \bar{x})(y - \bar{y})}{36} = 0$$

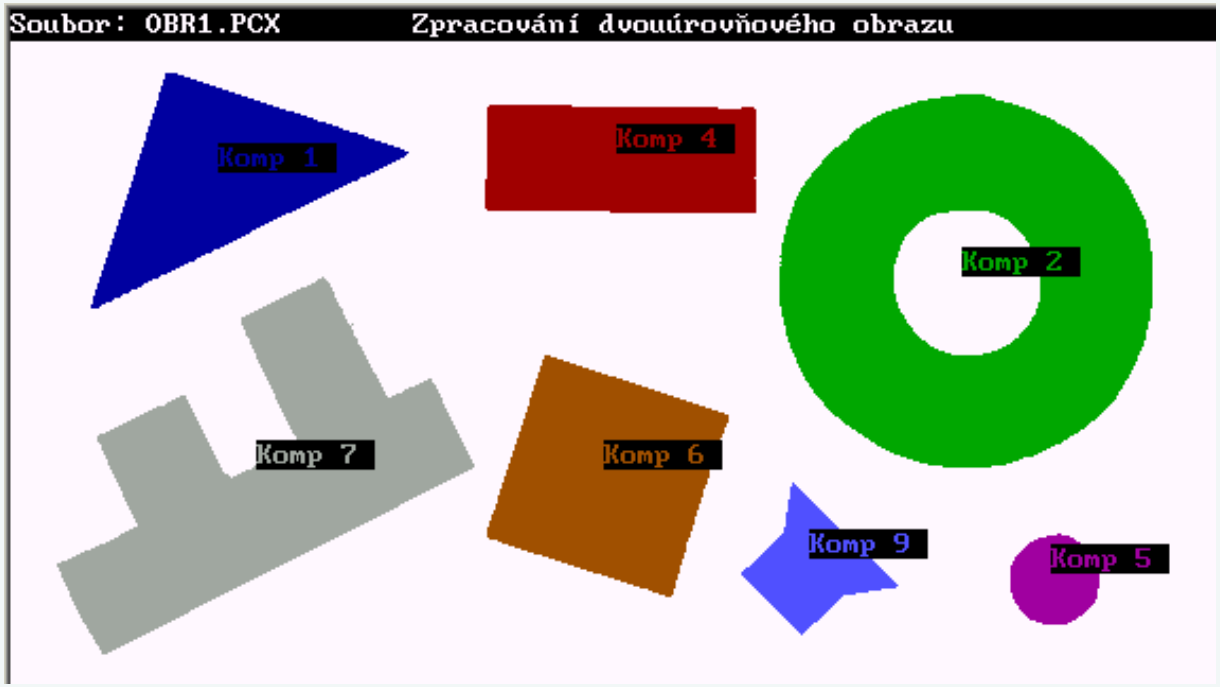
$$\phi_1 = 0.153, \qquad \phi_2 = 4.82 * 10^{-3}$$


Soubor: OBR1.PCX
Rozpoznání obrazu
Info
Soubor
Poznej
Ukaž
Histogram
Dekompozici
Rozpoznání
A11
A12
A22
Vmin
Vmax
It, Jt

Pozn.: RST invariantní charakteristická čísla matice druhých momentů ν_{min} a ν_{max}, která byla použita v ukázkovém programu, a která jsou uvedena na předchozím snímku, se počítala z invariantních momentů ϕ_1 a ϕ_2 takto:

$$\nu_{min} = 0.5 \cdot \phi_1 - 0.5\sqrt{\phi_2}$$

$$\nu_{max} = 0.5 \cdot \phi_1 + 0.5\sqrt{\phi_2}$$




```
Komponenta barvy    je: trojuhelnik
Komponenta barvy    je: mezikruzi
Komponenta barvy    je: obdelnik
Komponenta barvy    je: kolecko
Komponenta barvy    je: ctverec
Komponenta barvy    je: parnik
Komponenta barvy    je: trychtyr
```