

## 10.4 Redukcia zahmlenia

Obraz degradovaný zahmlením môžeme modelovať rovnicou

$$g(n_1, n_2) = f(n_1, n_2) * b(n_1, n_2) \quad (10.22)$$

Je to konvolúcia originálneho obrazu s impulzovou charakteristikou zahmlenia [Lim90].

$b(n_1, n_2)$  je **funkcia zahmlenia** alebo po anglicky "point spread function".

Zahmlenie obrazu môže byť spôsobené napr.

- zlým zaostrením
- pohybom kamery alebo objektu pri expozícii snímky
- atmosférickou turbulenciou



a

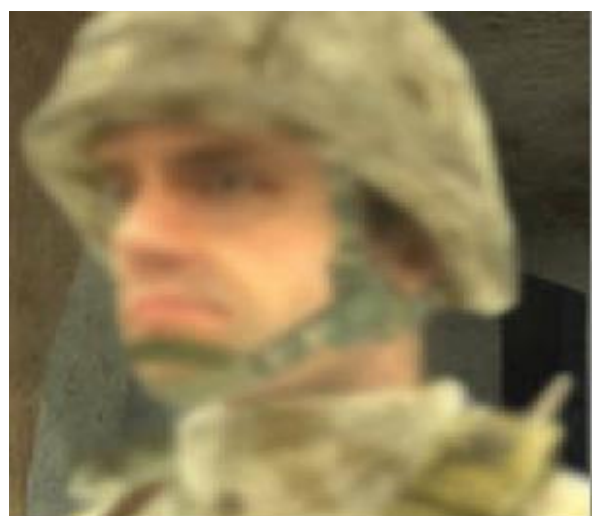


b

**Obr. 10.8** Ilustrácia zahmlenia (a) zahmlenie spôsobené pohybom, (b) po rekonštrukcii inverzným filtrom



a



b

**Obr. 10.9** Ilustrácia zahmlenia (a) zaostrený obraz, (b) nezaostrený obraz

## Postup redukcie zahmlenia - dekonvolúcia

Postup potlačenia zahmlenia na obrazoch môžeme rozdeliť do 2 základných kategórií:

**1. DEKONVOLÚCIA** (spätná konvolúcia) – ak je zahmlenie  $b(n_1, n_2)$  je známa funkcia.

**2. SLEPÁ DEKONVOLÚCIA** – zahmlenie  $b(n_1, n_2)$  je neznáma funkcia

V druhom prípade:

- funkciu zahmlenia najprv odhadneme z dostupných informácií
- následne vykonáme dekonvolúciu

### Zahmlenie

**Ak poznáme originál  $f(n_1, n_2)$  a degradovaný obraz  $g(n_1, n_2)$ , dokážeme odhadnúť zahmlenie  $b(n_1, n_2)$ .**

**Ak  $f(n_1, n_2)$  je Kroneckerov impulz,**

**potom zahmlenie  $b(n_1, n_2)$  je rovné  $g(n_1, n_2)$ , vid' vzťah (10.22).**

Tento model je vhodný napr.

- pri rekonštrukcii hviezdnej oblohy, kde hviezdy, ktoré sú veľmi vzdialené, môžeme považovať za body a reprezentovať ich ako jednotkové impulzy
- pri potlačení zlého zaostrenia mikroskopu, príp. fotoaparátu, kde môžeme nasnímať obraz – jednotkový impulz a z nasnímaného obrazu určiť funkciu zahmlenia

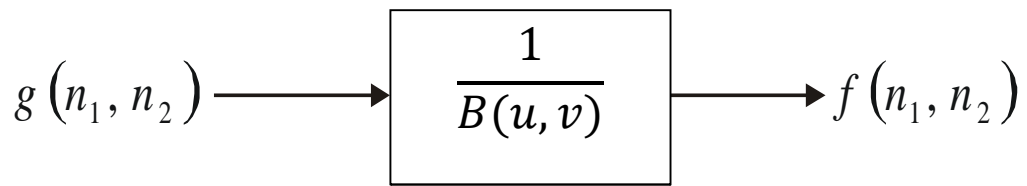
Model degradácie obrazu zahmlením – **konvolúcia** v priestorovej oblasti – zodpovedá **násobeniu** vo frekvenčnej oblasti:

$$G(u, v) = F(u, v) \cdot B(u, v) \quad (10.23)$$

$G(u, v)$  – Fourierova transformácia (FT) degradovaného obrazu

$F(u, v)$  – Fourierova transformácia funkcie originálu

$B(u, v)$  – Fourierova transformácia funkcie degradácie



**Obr. 10.10** Schéma rekonštrukcie obrazu inverznou filtráciou

Ideálne zrekonštruovaný (reštaurovaný) obraz dostaneme ako inverznú Fourierovu transformáciu:

$$f(n_1, n_2) = F^{-1} \left\{ \frac{G(u, v)}{B(u, v)} \right\} \quad (10.24)$$

$B(u, v)$  určíme ako PSF (Point Spread Function) - odozva sústavy na Kroneckerov impulz.

Ak mikroskopom zosnímame bodovú funkciu  $f(x, y)$  (Kroneckerov impulz), potom výstup  $g(x, y)$  je zároveň impulzovou charakteristikou sústavy snímacích šošoviek.

**DEKONVOLÚCIA** sa tiež nazýva **inverzná filtrácia**.

Pri dekonvolúcii zahmleného obrazu vychádzame z predpokladu, že funkcia zahmlenia  $b(n_1, n_2)$  je známa.

Aplikovaním Fourierovej transformácie na vzťah (10.22) dostaneme vzťah (10.23), resp.

$$F(u, v) = \left\{ \frac{G(u, v)}{B(u, v)} \right\} \quad (10.25)$$

Potom prenosová funkcia má tvar

$$H(u, v) = \frac{1}{B(u, v)} - \text{INVERZNÝ FILTER} \quad (10.26)$$

Inverzný filter (10.26) - sústava, ktorá na výstupe získa originál  $f(n_1, n_2)$  z degradovaného obrazu  $g(n_1, n_2)$  je znázornená na Obr.10.10.

**Inverzný filter** je veľmi citlivý na šum.

Ak je  $B(u, v)$  veľmi malé, potom  $\frac{1}{B(u, v)}$  je veľmi veľké. To ale znamená, že aj slabý šum vo frekvenčnej oblasti, kde je  $\frac{1}{B(u, v)}$  veľmi veľké bude **výrazne zosilnený!**

Ako môžeme obmedziť prílišnú citlivosť inverzného filtra na šum?

Skúsime obmedziť frekvenčnú charakteristiku  $\frac{1}{B(u, v)}$  pomocou prahu  $\gamma$  :

$$H(u, v) = \begin{cases} \frac{1}{B(u, v)}, & \frac{1}{|B(u, v)|} < \gamma \\ \gamma \frac{|B(u, v)|}{B(u, v)} & \text{inde} \end{cases} \quad (10.27)$$

### Iteračná metóda implementácie inverzného filtra

Odhad funkcie  $f(n_1, n_2)$  aktualizujeme po každej iterácii.

Nech  $\hat{f}_k(n_1, n_2)$  je signál odhadnutý po  $k$ -tej iterácii.

Ak  $\hat{f}_k(n_1, n_2)$  je dobrý odhad funkcie  $f(n_1, n_2)$ , potom  $\hat{f}_k(n_1, n_2) * b(n_1, n_2)$  je blízke  $g(n_1, n_2)$

Počiatočný odhad a iteračný proces popisujú nasledujúce vzťahy:

$$\hat{f}_0(n_1, n_2) = \delta \cdot g(n_1, n_2) \quad (10.28)$$

Signál odhadnutý po  $k+1$  iteráciách máme odhad  $\hat{f}_{k+1}(n_1, n_2)$  :

$$\hat{f}_{k+1}(n_1, n_2) = \hat{f}_k(n_1, n_2) + \delta \cdot \left( g(n_1, n_2) - \hat{f}_k(n_1, n_2) * b(n_1, n_2) \right) \quad (10.29)$$

kde  $\delta$  je kladná konštanta, ktorá zaručuje konvergenciu iteračného procesu.

$$\hat{f}_k(n_1, n_2) + \delta \cdot \left( g(n_1, n_2) - \hat{f}_k(n_1, n_2) * b(n_1, n_2) \right) \quad (10.30)$$

Vzťah (10.30) obsahuje korekčný člen –súčin váhovacej konštanty  $\delta$  a rozdielu

$$\left( g(n_1, n_2) - \hat{f}_k(n_1, n_2) * b(n_1, n_2) \right) \quad (10.31)$$

Popis procesu vo frekvenčnej oblasti:

Vzťahy (10.28 – 10.29) sa zmenia na

$$\hat{F}_0(\omega_1, \omega_2) = \delta \cdot G(\omega_1, \omega_2) \quad (10.32)$$

$$\hat{F}_{k+1}(\omega_1, \omega_2) = \hat{F}_k(\omega_1, \omega_2) + \delta \cdot \left( G(\omega_1, \omega_2) - \hat{F}_k(\omega_1, \omega_2) B(\omega_1, \omega_2) \right) \quad (10.33)$$

Riešením rekurzcie získame

$$\begin{aligned} \hat{F}_k(\omega_1, \omega_2) &= \delta \cdot G(\omega_1, \omega_2) \left[ 1 + (1 - \delta \cdot B(\omega_1, \omega_2)) + \dots + (1 - \delta \cdot B(\omega_1, \omega_2))^k \right] \\ &= \frac{G(\omega_1, \omega_2)}{B(\omega_1, \omega_2)} \left[ 1 - (1 - \delta \cdot B(\omega_1, \omega_2))^{k+1} \right] \end{aligned} \quad (10.34)$$

Pre  $k \rightarrow \infty$  sa  $\hat{F}_k(\omega_1, \omega_2) \rightarrow G(\omega_1, \omega_2) / B(\omega_1, \omega_2)$ , t.j. blíži sa ku vzťahu pre rekonštrukciu inverzným filtrom za predpokladu, že

$$\left| 1 - \delta \cdot B(\omega_1, \omega_2) \right| < 1 \quad (10.35)$$

– výsledok po  $k$  – iteráciách nie je totožný s inverznou filtráciou

– je odolnejší voči šumu

*Po inverznej filtrácii:*

$$p(n_1, n_2) = IDFT \left[ G(k_1, k_2) \cdot H(k_1, k_2) \right] \quad (10.36)$$

kde  $G(k_1, k_2)$  – DFT  $g(n_1, n_2)$

$H(k_1, k_2)$  získame nasledovne:

$$H(k_1, k_2) = \frac{1}{B(\omega_1, \omega_2)} \Big|_{\omega_1=2\pi k_1/N, \omega_2=2\pi k_2/N} \quad (10.37)$$

Pri DFT a IDFT sú použité veľkosti obrazu 256x256 bodov.

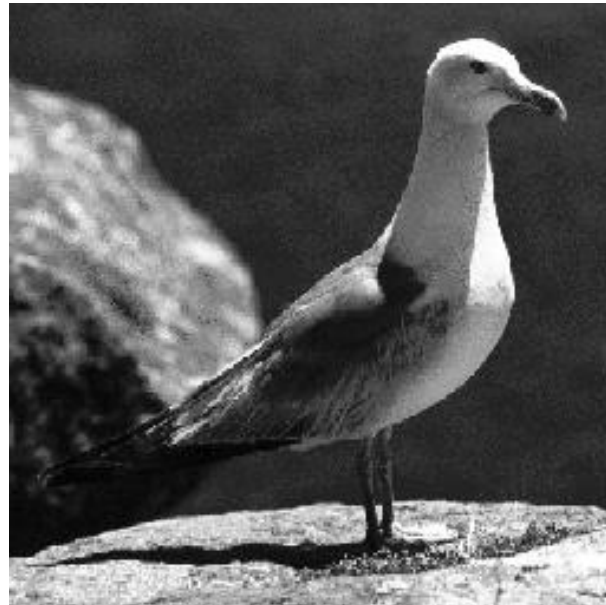
Pri absencii ďalšieho šumu a veľmi nízkych hodnotách  $B(\omega_1, \omega_2)$  je inverzná filtrácia dostatočne účinná aj napriek tomu, že degradovaný obraz  $g(n_1, n_2)$  je ovplyvnený použitím pravouhlého okna.



(a)



(b)



(c)

**Obr. 10.11** Rekonštrukcia obrazu inverznou filtráciou: a) originál 256x256 obrazových bodov, b) zahmlený obraz, c) výsledok po rekonštrukcii [PaPo02].

## SLEPÁ DEKONVOLÚCIA

Ak je funkcia zahmlenia  $b(n_1, n_2)$  neznáma, musíme ju najsôr odhadnúť

Tento proces nazývame **slepá dekonvolúcia** – pokúšame sa o dekonvolúciu signálu  $g(n_1, n_2)$  bez dôkladnej znalosti funkcie zahmlenia  $b(n_1, n_2)$ .

Ak nevieme nič o funkciách  $f(n_1, n_2)$  a  $b(n_1, n_2)$ , potom nie je možné problém slepej dekonvolúcie vyriešiť!

Analógia: Predstavme si, že hľadáme dve čísla, ak je známy len ich súčet.

Potrebuje mať aspoň nejaké informácie o  $f(n_1, n_2)$ , o  $b(n_1, n_2)$  alebo o oboch.

Algoritmy slepej dekonvolúcie sa líšia v závislosti od známej informácie a od spôsobu jej využitia.

Predpokladajme:

1)  $f(n_1, n_2)$  a  $b(n_1, n_2)$  – konečné postupnosti a

2)  $F(z_1, z_2)$  a  $B(z_1, z_2)$  – ich obrazy v Z-rovine

Potom  $f(n_1, n_2)$  môžeme získať z konvolúcie  $g(n_1, n_2) = f(n_1, n_2) * b(n_1, n_2)$

použitím **polynomickej faktorizácie – rozklad na súčin polynómov**  $F(z_1, z_2)$  a  $B(z_1, z_2)$ :

$$G(z_1, z_2) = F(z_1, z_2) \cdot B(z_1, z_2) \quad \text{– obraz konvolúcie v Z-rovine} \quad (10.38)$$

Ak  $f(n_1, n_2)$  a  $b(n_1, n_2)$  sú konečné postupnosti potom  $G(z_1, z_2)$  je 2D-polynóm konečného stupňa s argumentmi  $z_1^{-1}$  a  $z_2^{-1}$ .

Polynómy  $F(z_1, z_2)$  a  $B(z_1, z_2)$  sú zvyčajne nefaktorizovateľné.

Ak je splnený predpoklad, že  $F(z_1, z_2)$  a  $B(z_1, z_2)$  sú nefaktorizovateľné, potom jediné netriviálne faktory  $G(z_1, z_2)$  sú  $F(z_1, z_2)$  a  $B(z_1, z_2)$ .

Na určenie netriviálneho rozkladu  $G(z_1, z_2)$  použijeme polynomicke faktorizačné algoritmy – získame  $F(z_1, z_2)$  (resp.  $f(n_1, n_2)$ )

Algoritmy slepej dekonvolúcie sú – výpočtovo veľmi zložité

– extrémne citlivé na akúkoľvek odchýlku od

predpokladu, že  $G(z_1, z_2) = F(z_1, z_2) \cdot B(z_1, z_2)$ ,

resp.  $g(n_1, n_2) = f(n_1, n_2) * b(n_1, n_2)$

V praxi nemôžeme konvolučný model považovať za presný, napríklad kvôli prítomnosti šumu na pozadí.