

Diskrétní 2D konvoluce

ČVUT FEL v Praze
36ACS

1. prosince 2006

Martin “BruXy” Bruchanov
bruxy@regnet.cz

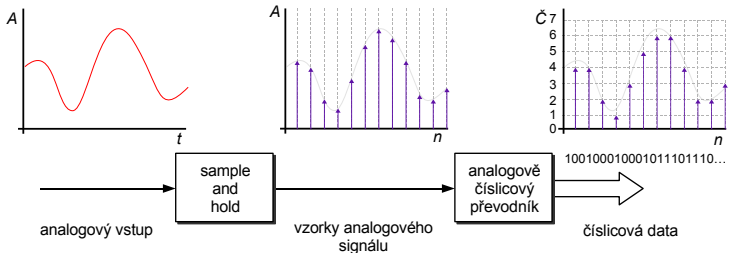
Diracův impuls

- jednotkový impulz, δ -impulz, δ -funkce;
- speciální signál s nulovou šířkou impulzu a nekonečnou amplitudou;
- platí pro něj

$$\delta(t) = \lim_{t \rightarrow 0} f(t) = \begin{cases} \infty, & t = 0 \\ 0, & t \neq 0 \end{cases}, \quad \int_{-\infty}^{+\infty} \delta(t) dt = 1$$

- v DSP technice se definuje δ -funkce $\delta(n)$ jako normalizovaný impulz, který má ve vzorku 0 hotnodu 1 a ostatní vzorky mají hodnotu 0.

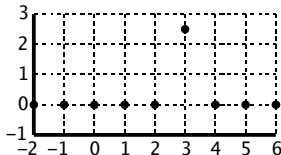
Popis vzorkovaného signálu



- násobení signálu posloupností Diracových impulzů je ekvivalentní vzorkování signálu v okamžicích těchto impulzů
- jakýkoliv impulz může být reprezentován jako posunutá a škálovaná δ -funkce.

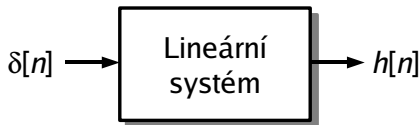
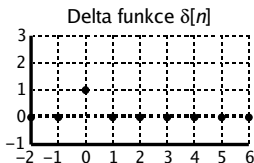
Reprezentace impulsu

Jakýkoliv impuls může být reprezentován jako posunutá a škálované δ -funkce, např.



$$x(n) = 2,5\delta(n - 3)$$

Odezva systému $h(n)$ na jednotkový impulz



Odezvou lineárního systému na Diracův impulz je *impulzní odezva* (váhová funkce) $h(n)$. Konvolucí impulzní odezvy systému a vstupního signálu je výstupní signál systému.

Diskrétní konvoluce

Jak rozumět tomu, jak lin. systém mění vstupní signál $x(n)$ na výstupní $y(n)$?

- Vstupní signál je dekomponován na množinu impulzů,
- každý impulz představuje posunutá a škálovaná δ -funkce.
- Výsledný výstup každého impulzu je škálovaná a posunutá verze impulzní odezvy.

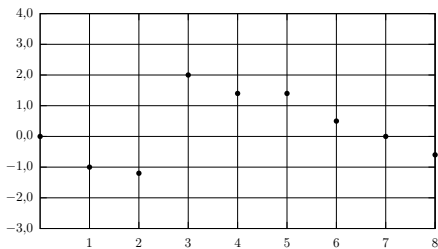
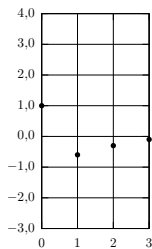
$$x(n) = 2,5\delta(n - 3) \longleftrightarrow y(n) = 2,5h(n - 3)$$

- Znalost impulzní odezvy systému umožňuje stanovit výstup pro libovolný vstupní signál.
- Pokud je lineární systém koncipován jako *filtr*, pak impulzní odezva se nazývá *konvoluční jádro* (*convolution kernel*), nebo maska.

Konvoluce

$$y(n) = x(n) \star h(n) = h(n) \star x(n)$$

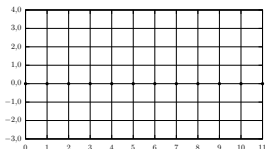
$$y(n) = \sum_{j=0}^{M-1} h(j) \cdot x(i-j)$$

 $x(n)$

 $h(n]$


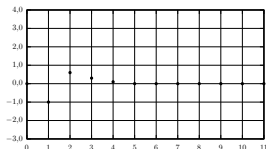
Příklad konvoluce I.

$$x_{0 \rightarrow 5} = \{0,0; -1,0; -1,2; 2,0; 1,4; 1,4\}$$

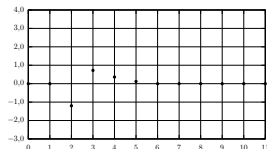
$$x(0)h(n-0)$$



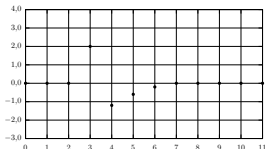
$$x(1)h(n-1)$$



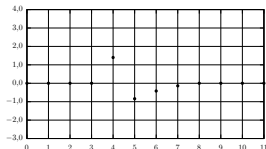
$$x(2)h(n-2)$$



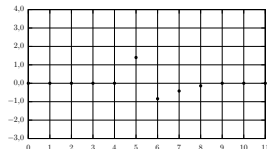
$$x(3)h(n-3)$$



$$x(4)h(n-4)$$



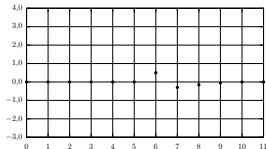
$$x(5)h(n-5)$$



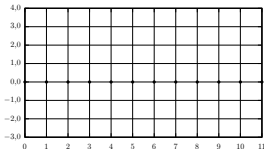
Příklad konvoluce II.

$$x_{6 \rightarrow 8} = \{0,5; 0,0; -0,6\}$$

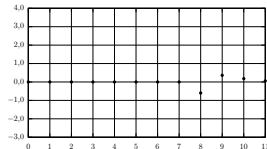
$$x(6)h(n-6)$$



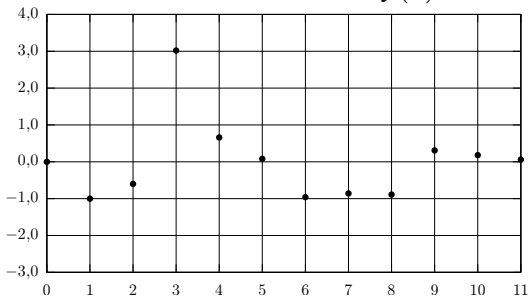
$$x(7)h(n-7)$$



$$x(8)h(n-8)$$



Po sečtení všech složek $y(n)$:



Diskrétní 2D konvoluce

- aplikace v počítačové grafice a pro zpracování obrazu

-

$$y(m, n) = x(m, n) \star h(m, n)$$

-

$$y(m, n) = \sum_{k=0}^{M-1} \left(\sum_{l=0}^{N-1} \left(h(m-k, n-l) \cdot x(k, l) \right) \right)$$

- pomocí konvolučních jader lze definovat filtry provádějící:
 - vyhlazování (filtr dolní propust)
 - doostřování (filtr horní propust)
 - detekce hran (gradientní operátory)

Jak udělat z jednorozměrné 2D

Filtr pro klouzavé průměrování

$$h_1 = \frac{1}{3}(1, 1, 1)$$

Převod do 2D:

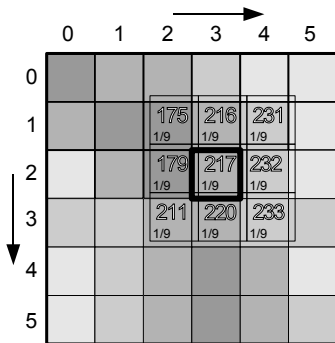
$$H_1 = \frac{1}{3}(1, 1, 1) \star \frac{1}{3} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Binomický filtr

$$h_2 = \frac{1}{16}(1, 4, 6, 4, 1)$$

$$H_2 = \frac{1}{16}(1, 4, 6, 4, 1) \star \frac{1}{16} \begin{pmatrix} 1 \\ 4 \\ 6 \\ 4 \\ 1 \end{pmatrix} = \frac{1}{256} \begin{pmatrix} & 1 & 4 & 6 & 4 & 1 \\ 1 & 1 & 4 & 6 & 4 & 1 \\ 4 & 4 & 16 & 24 & 16 & 4 \\ 6 & 6 & 24 & 36 & 24 & 6 \\ 4 & 4 & 16 & 24 & 16 & 4 \\ 1 & 1 & 4 & 6 & 4 & 1 \end{pmatrix}$$

Výpočet 2D diskrétní konvoluce



$$H_1 = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

$$\begin{aligned} & \frac{1}{9} \cdot 175 + \frac{1}{9} \cdot 216 + \frac{1}{9} \cdot 231 + \\ & \frac{1}{9} \cdot 179 + \frac{1}{9} \cdot 217 + \frac{1}{9} \cdot 232 + \\ & \frac{1}{9} \cdot 211 + \frac{1}{9} \cdot 220 + \frac{1}{9} \cdot 233 = 213 \end{aligned}$$

Postup: Máme vstupní a výstupní bitmapu, konvoluční jádro zpracovává bod po bodu vstup(i, j) a vypočítanou hodnotu zapisuje na odpovídající pozici výstup(i, j).

Kde běžný uživatel využije konvoluční jádra

- Tvorba vlastních filtrů a úpravy obrazu v rastrových editorech: Adobe Photoshop, Paint Shop Pro, GIMP,...
- Dávkové zpracování obrazu pomocí ImageMagick
www.imagemagick.org.
- Podpora v některých vektorových editorech.
- Úprava vektorových elementů ve formátu SVG
<http://www.w3.org/TR/SVG/>.

Grafické editory a konvoluce

Diagram illustrating the configuration of a convolution kernel in a graphical editor (GIMP).

The kernel H is defined as:

$$H = \frac{1}{2} \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{pmatrix} + 128$$

The corresponding GIMP "Konvoluční matice" (Convolution Matrix) dialog box is shown with the following settings:

- Matice:** A 3x5 grid of values. The second row, second column is highlighted with a value of 1,000000. The other values are 0,000000, except for the third row, third column which is -1,000000.
- Dělitel:** 2,0000
- Posun:** 128,00
- Obvod:** ☒ Rozšířit, ☐ Ohýbání, ☐ Ořez
- Kanály:** ☐ Šedá, ☒ Červená, ☒ Zelená, ☒ Modrá, ☐ Alfa
- Buttons:** ☐ Automatické, ☐ Vážená alfa,

Arrows indicate the mapping from the mathematical formula to the dialog box settings: the denominator 2 maps to the "Dělitel" field, the 3x3 kernel matrix maps to the "Matice" grid, and the constant 128 maps to the "Posun" field.

Zadávání konvolučních jader v graf. editoru GIMP.

Vyhlazovací filtry I.



Originál zarušeného snímku.

Vyhlazovací filtry II.



$$H_1 = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Vyhlazovací filtry III.



$$H_2 = \frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

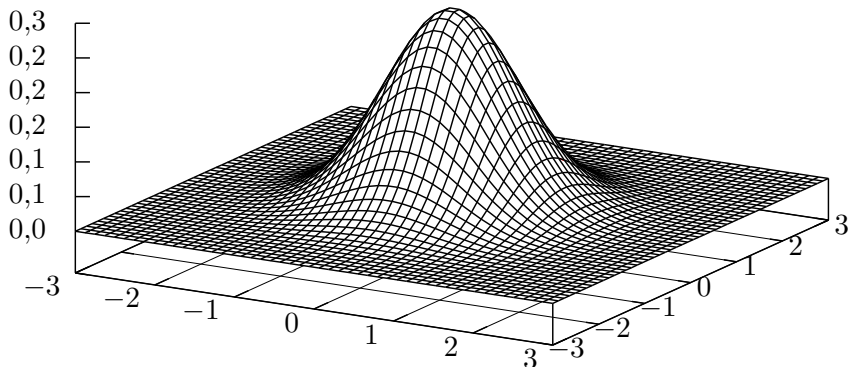
Vyhlazovací filtry IV.



$$H_3 = \frac{1}{256} \begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{pmatrix}$$

Gaussovo rozostření I.

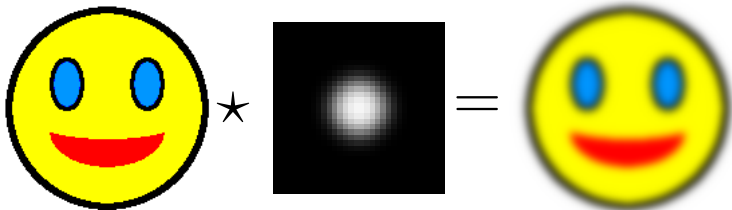
$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



Rozptyl $\sigma = 0,75$.

Gaussovo rozostření II.

$$H_G = \frac{1}{159} \begin{pmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 6 & 4 & 2 \end{pmatrix}$$



Vyhlazovací filtry — závěr

$$H_1 = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, H_2 = \frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}, H_3 = \frac{1}{256} \begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{pmatrix}$$

- Průměrovací filtry mají střední hodnotu všech členů 1, např. pro H_2 je průměr $\frac{1}{16}(1 + 2 + 1 + 2 + 4 + 2 + 1 + 2 + 1) = 1$.
- Nevýhodou průměrovacích filtrů je to, že při vyhlazení dojde na ostrých barevných přechodech k mírnému rozmazání a kvůli tomu utrpí i tenké čáry a další detaily.

Detekce hran

- Hrany se v digitálním obrazu nacházejí v místech, kde se prudce mění jas.
- Vylepšování hran souvisí s doostřováním a dokáže zlepšit vnímání lidského zraku.
- Důležité pro předzpracování obrazu v počítačovém vidění:
 - rozeznávání objektů
 - identifikace – zpracování otisků prstů, tvarů obličeje
 - určování pozice vzhledem ke kameře
 - Optical Character Recognition (OCR)

Princip detekce hran

- Hrana je v obraze dána obrazovým elementem a jeho okolím,
- je určena tím, že se náhle změní hodnota obrazové funkce $f(x, y)$.
- Pro studium změn funkce dvou proměnných se používají parciální derivace a změnu funkce udává její *gradient* ∇ , který určuje směr největšího růstu ψ funkce a strmost tohoto růstu $|\nabla f(x, y)|$:

$$|\nabla f(x, y)| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

- Laplaceův operátor, vychází z 2. derivací:

$$\nabla^2 f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Princip detekce hran, pokračování

- Laplaceův operátor $\nabla^2 f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$
- V diskrétním obraze aproximujeme derivace pomocí diferencí:

$$\frac{\partial^2 f}{\partial x^2} \approx (f(i-1, j) - f(i, j)) - (f(i, j) - f(i+1, j))$$

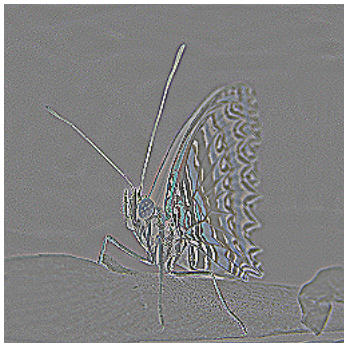
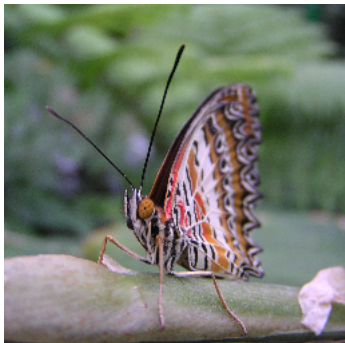
$$\frac{\partial^2 f}{\partial y^2} \approx (f(i, j+1) - f(i, j)) - (f(i, j) - f(i, j-1))$$

$$\frac{\partial^2 f}{\partial x^2} \approx f(i-1, j) - 2f(i, j) + f(i+1, j) \dots (1, -2, 1)$$

$$\frac{\partial^2 f}{\partial y^2} \approx f(i, j+1) - 2f(i, j) + f(i, j-1) \dots \begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix}$$

$$H_4 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad H_5 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Laplacián, filtr typu horní propust



$$H_4 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix} + 128$$

Laplacián je invariantní vůči otočení. Přičtení 128 slouží k posunutí jasu pixelů.

Operátor Prewittové

Příklad operátoru, který aproximuje první derivaci (další možnosti např. Sobel, Kirch, Robinson, ...). Slouží pro odhad gradientu v okolí 3×3 pro osm směrů (ty se získají pootočením matice).

$$H_P = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}, \quad H'_P = \begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{pmatrix}, \quad H''_P = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$

Větší matice s vyšším rozlišením může sloužit k vytváření reliéfu v různých směrech.

Operátor Prewittové, příklad všechny směry



Doostřování

- Ostrost vidění je pro lidské vnímání velmi důležitou vlastností.
- Často se stává, že obraz pro sejmutí kamerou, vyfotografování nebo naskenování nemá moc ostré hrany. Tady nám pomůžou filtry pro ostření, které zvýrazní hrany v obrazu.
- Nevýhoda dále popsaných filtrů je to, že kromě hran zvýrazní v obraze také šum a některé další nechtěné detaily.
- Pro doostřování využijeme Laplacián jehož hodnotu odečteme od hodnoty původního pixelu:

$$\begin{aligned}g(i, j) &= f(i, j) - (f(i-1, j) + f(i+1, j) + \\&\quad + f(i, j+1) + f(i, j-1) - 4f(i, j)) = \\&= 5f(i, j) - f(i-1, j) - f(i+1, j) - f(i, j+1) - f(i, j-1)\end{aligned}$$

$$H_6 = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

Příklad pro programátory

```
int sharpen_filter[3][3]={ {0,-1,0}, {-1,5,-1}, {0,-1,0}};

for(x = 0 ; x < PIX_WIDTH; x++){
    for(y = 0; y < PIX_HEIGHT; y++){
        for(k = 0; k < 3; k++){
            for(l = 0; l < 3; l++){
                y = getpixel (INPUT, (x - 1) + k, (y - 1) + l);
                sum_y += y * sharpen_filter[k][l];
            }
        }
        putpixel(OUTPUT, i, j, sum_y);
    }
}
```

Doostřování, příklady I.



Digitalizovaný videosignál snímku z CCD kamery.

Doostřování, příklady II.



$$H_6 = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

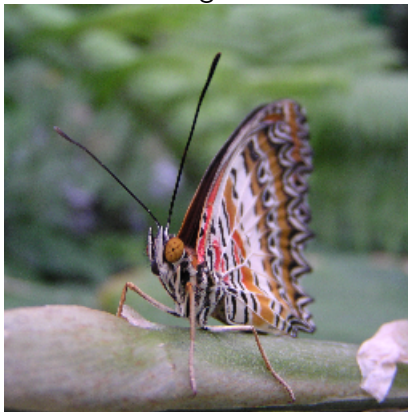
Doostřování, příklady III.



$$H_7 = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

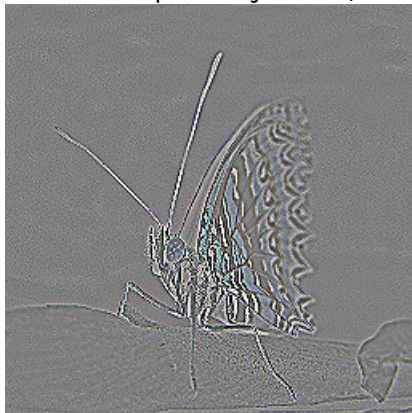
Příklad uživatelského filtru pro vylepšení obrazu

Originál



Příklad uživatelského filtru pro vylepšení obrazu

Detekce hran pomocí jádra $H_4 + 128$



Příklad uživatelského filtru pro vylepšení obrazu

Gaussovo rozostření, rozptyl 5 pixelů



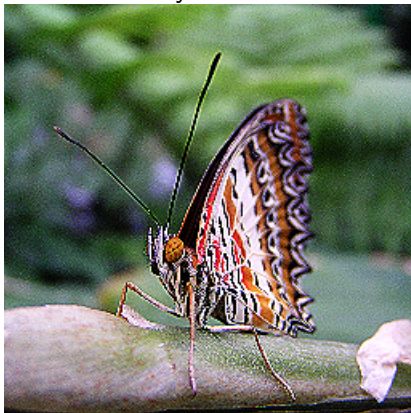
Příklad uživatelského filtru pro vylepšení obrazu



- Sloučení zrnitosti: $L_S = O + D - 128$
- Překrytí: $L_P = L_S \cdot (L_S + (2 \cdot T \cdot (255 - L_S)) / 255) / 255$

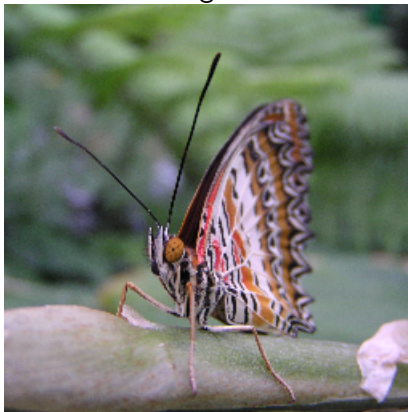
Příklad uživatelského filtru pro vylepšení obrazu

Výsledek



Příklad uživatelského filtru pro vylepšení obrazu

Originál



Reference

- *Smith, Steven W.*: **The Scientists and Engineer's Guide to Digital Signal Processing**. California Technical Publishing, 1999.

<http://www.dspguide.com>

- *Goznales, Rafael C.; Woods, E. Richards*: **Digital Image Processing**. Prentice Hall 2002, 2nd edition.
- *Hlaváč, Václav; Sedláček Miloš*: **Zpracování signálů a obrazů**. Vydavatelství ČVUT 2005.
- **Dokumentace k programu GIMP**

<http://docs.gimp.org>