

Pós-processamento gráfico: Antialiasing recovery no domínio de Fourier

João Paulo Navarro Barbosa
Bacharelado em Matemática e Computação Científica - UFSC
jp_nb95@hotmail.com

Prof. Dr. Leonardo Koller Sacht
Departamento de Matemática - UFSC
leonardo.sacht@ufsc.br

Palavras-chave: Transformada de Fourier, antialiasing, computação gráfica.

Resumo:

Operações não lineares em imagens geram artefatos indesejáveis, principalmente arestas serrilhadas. Mesmo se a imagem de entrada não possuir tais artefatos, o processamento em domínio discreto introduz este problema. Aspectos físicos de uma cena são facilmente danificados por processamentos não lineares, arestas serrilhadas (aliasing), que são um subproduto do processamento digital de imagens, podem ser geradas, por exemplo, pelo processo de binarização de uma imagem, conhecido como processo de threshold - limiar - ou uma simples rotação aplicada em uma cena.

Reconstruções de arestas no âmbito da Computação gráfica possuem duas vertentes de processamento, definidas no domínio espacial [1] ou no domínio da frequência. Estudos e softwares atuais priorizam a filtragem no domínio espacial [1] [2] [3] [4].

Apresenta-se um processo de reconstrução e processamento realizado em todo no domínio da frequência utilizando filtros tradicionais para a acentuação de arestas e reconstrução.

Seja $f(x, y)$ a imagem de entrada discretizada e $h(x, y)$ um operador linear responsável pela filtragem. Tem-se que a filtragem é dada pela convolução de f e h :

$$g : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R} \times \mathbb{R}$$

$$g(x, y) = f(x, y) * h(x, y)$$

Pelo teorema da convolução [7], g no domínio da frequência pode ser descrita como:

$$G : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{C} \times \mathbb{C}$$

$$G(u, v) = F(u, v)H(u, v)$$

onde $F(u, v)$ é a transformada de Fourier de $f(x, y)$, isto é, $F(u, v) = \mathcal{F}\{f(x, y)\}$ e $H(u, v)$ o filtro no domínio da frequência. Note que a transformada de Fourier inversa $\mathcal{F}^{-1}\{G(u, v)\}$ define a imagem filtrada no domínio espacial $g(x, y)$.

O método proposto de *Antialiasing* no domínio da frequência está relacionado com a análise das frequências altas de uma imagem - acentuação de arestas. No entanto, detecção de arestas já possui um grande estudo no domínio espacial, possuindo métodos conhecidos por seus resultados satisfatórios como, por exemplo, Sobel, Roberts, Prewitt e Canny [5].

Por outro lado, na frequência, arestas são dadas por transições muito rápidas realizadas em pequenas regiões da cena. Sabendo disso, torna-se intuitivo que arestas encontram-se na região das frequências altas, pois são representadas por transições abruptas do sinal, que por sua vez estão associados aos componentes de alta frequência do espectro de Fourier. É importante ressaltar que não são apenas as arestas que estão contidas nas frequências altas, mas também regiões com textura predominante. Portanto, aplica-se um filtro *passa alta* para atenuar as frequências baixas, deixando apenas as frequências altas passarem, obtendo-se acentuação das arestas.

Inicialmente esta acentuação das frequências altas será realizada por alguns dos filtros *passa alta* tradicionais, isto é, funções de transferência, dados por [6] [7]:

Ideal(IHPF)	Gaussiano(GHPF)	Butterworth(BHPF)
$H(u, v) = \begin{cases} 0 & D(u, v) \leq D_0 \\ 1 & D(u, v) > D_0 \end{cases}$	$H(u, v) = 1 - e^{(-D^2/2D_0^2)}$	$H(u, v) = \frac{1}{1+[D_0/D(u,v)]^{2n}}$

Com os seguintes parâmetros de entrada dos filtros: $D_0 \in (0, 1]$ é a frequência de corte medida a partir da origem, $D(u, v)$ é a distância do ponto (u, v) até a origem no plano da frequência, e $n \in \mathbb{N}$ é a ordem do filtro Butterworth, onde é necessária uma discussão mais detalhada sobre a qualidade de filtragem de ambos, mas de antemão IHPF realiza uma filtragem muito agressiva gerando artefatos no processamento.

Consequentemente filtros *passa baixa* (H_{LP}) na frequência podem ser escritos em função dos filtros *passa alta* (H_{HP}), com funcionamento inverso, isto é, atenua frequências altas e permite a passagem das frequências baixas tem-se então [6] [7]: $H_{LP} = 1 - H_{HP}$.

Seja $f(x, y)$ a imagem original e $g(x, y)$ a imagem processada com aliasing ambas discretizadas. Considere suas respectivas versões no domínio da frequência dadas pelas transformadas de Fourier: $F(u, v) = \mathcal{F}\{f(x, y)\}$ e $G(u, v) = \mathcal{F}\{g(x, y)\}$ e $H(u, v)$ algum filtro *passa alta* citado.

Procura-se determinar uma maneira de copiar as arestas da imagem original (f) sobrepondo sobre a imagem processada (g), priorizando manter as características originais de g através do processamento no domínio da frequência.

Portanto, através do estudo dos componentes de alta e baixa frequência de F e G respectivamente criaremos uma função de reconstrução $R(u, v)$ dada por:

$$R(u, v) = H(u, v)F(u, v) + [(1 + k) - H(u, v)]G(u, v)$$

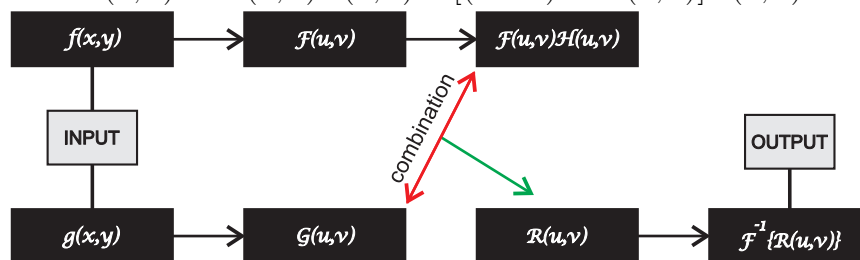


Figura 1: Esquemática teórica do método

A partir das figuras 2 e 3 podemos ver alguns resultados gerados através da função de reconstrução aplicada a uma cena que já sofreu algum tipo de filtragem. Na qual,

cada conjunto de imagens representa a imagem original, processada e restaurada respectivamente, com alguns dos filtros e parâmetros citados.

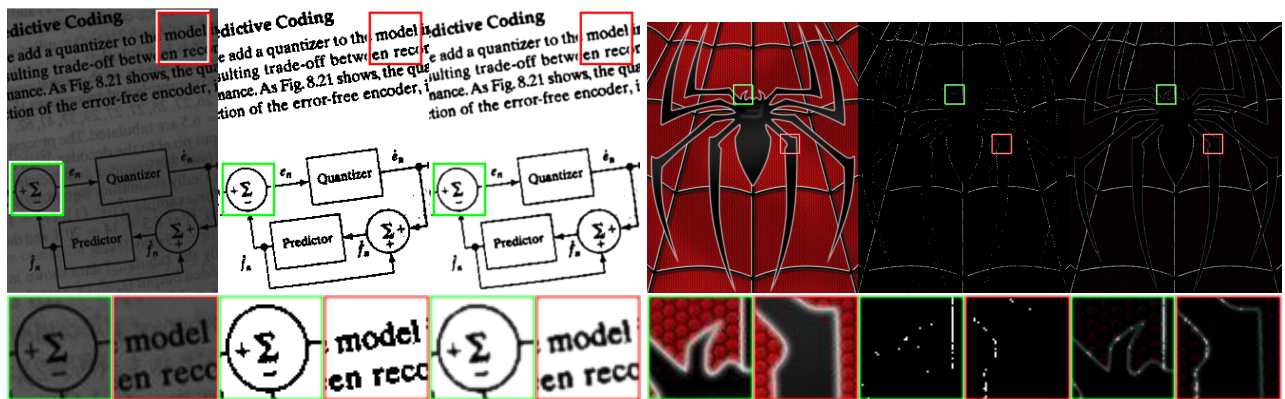


Figura 2: Texto: 1 - Imagem Original; 2 - Processou a imagem original com uma transformação de threshold automática; 3 - Aplicou-se $R(u, v)$ com Gaussiano com $D0 = 0,4$ e $k = 0,1$. Aranha: 1 - Imagem Original; 2 - Processou a imagem original com uma transformação de threshold; 3 - Aplicou-se $R(u, v)$ com Butterworth com $D0 = 0,2$, $n = 1$ e $k = 0,1$



Figura 3: Tucanos: 1 - Imagem Original; 2 - Processou a imagem original com um ajuste de tons de cinza; 3 - Aplicou-se $R(u, v)$ com Gaussiano com $D0 = 0,3$ e $k = 0$. Motoqueiros: 1 - Imagem Original; 2 - Processou a imagem original com o filtro laplaciano normalizado; 3 - Aplicou-se $R(u, v)$ com Gaussiano com $D0 = 0,4$ e $k = 0,013245$

Mais detalhes e resultados sobre o método e implementação no Matlab® (R2016b) estão disponíveis na página do projeto [9].

Referências

- [1] YANG, L.; SANDER, P.V.; LAWRENCE, J.; HOPPE, H. Antialiasing recovery. ACM Transactions on Graphics, 30(3):22, 2011.
- [2] RESHETOV, A. Morphological antialiasing (MLAA). Intel Labs, 2009.
- [3] LOTTES, T. Fast approximation antialiasing (FXAA). NVIDIA whitepaper, 2011.
- [4] NATTERER, M; NEUMANN, S. et al. GIMP - GNU IMAGE MANIPULATION PROGRAM (EUA, Org.). User Support: Manual. Disponível em: <https://docs.gimp.org/en/>.
- [5] GARAGE, W. et al. OPENCV - OPEN SOURCE COMPUTER VISION LIBRARY(Intel Corporation, EUA, Org.). User Support. Disponível em: <http://docs.opencv.org/3.0-beta/opencv2refman.pdf/>.
- [6] GONZALEZ, R.C.; WOODS, R.E. Processamento de Imagens Digitais. 1. ed: Blucher, 2000, v. 1.
- [7] PEDRINI, H.; SCHWARTZ, W.R. Análise de imagens digitais. 1. ed. [S.I.]: Thomson, 2008, v.1.
- [8] BRIGGS, W.L.; HENSON, V.E. The DFT: An Owners' Manual for the Discrete Fourier Transform. Society for Industrial and Applied Mathematics, 1995.
- [9] BARBOSA, J.P.N.; SACTH, L.K. Fourier domain antialiasing recovery: <http://http://mtm.ufsc.br/~leo/aarFourier>