

Flow-Based Image Abstraction

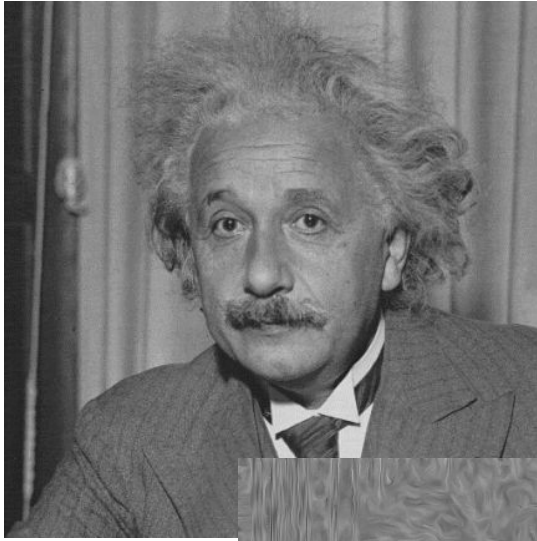
Resultados

Mateus Cordeiro

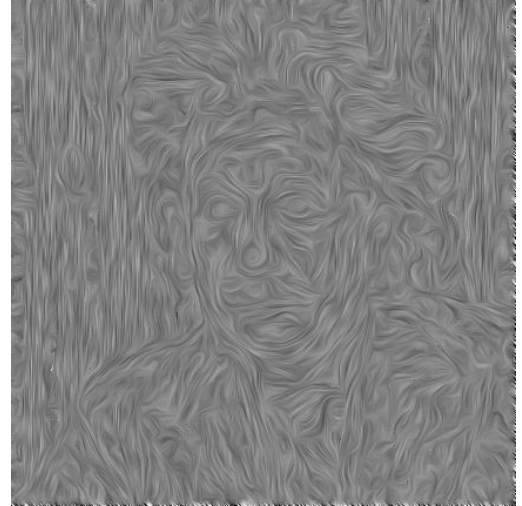
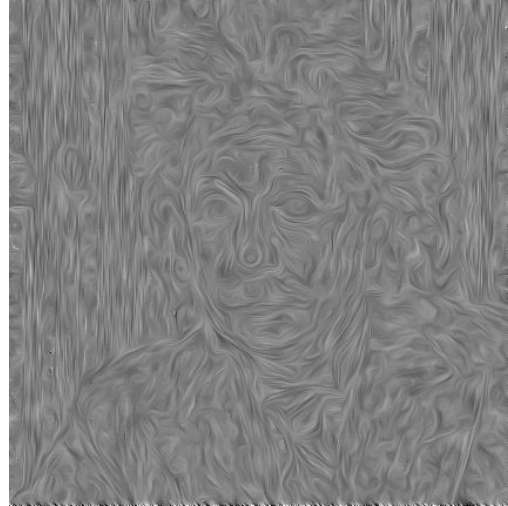
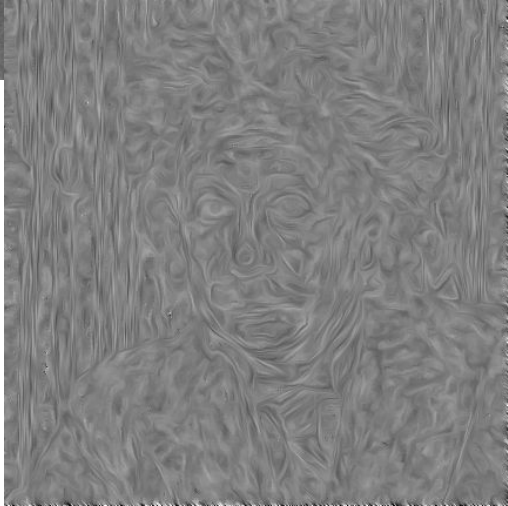
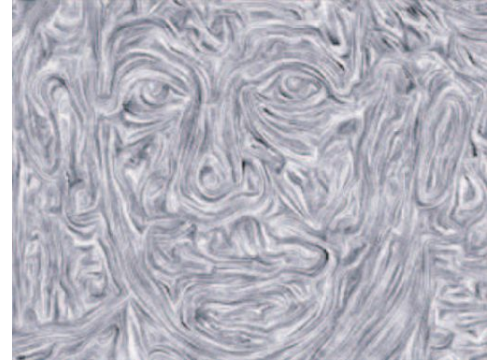
Edge Tangent Flow

- Operador Sobel para recuperação do mapa gradiente da imagem.
- Vetores iniciais são os vetores perpendiculares (sentido anti-horário) ao mapa gradiente.
- ETF Iterativo (3 iterações)
 - Filtro bilateral para refinar o direcionamento dos vetores
 - (sign) Função de sinal para inverter os vetores caso o ângulo entre eles seja maior que 90°.
 - (ws) Função peso que define o kernel radial.
 - (wm) Função peso que leva em consideração o domínio dos vizinhos (gradiente).
 - (wd) Função peso que leva em consideração o alinhamento entre os vetores vizinhos.
- Visualização com line integral convolution [2]

$$\mathbf{t}'(\mathbf{x}) = \frac{1}{k} \iint_{\Omega_\mu} \phi(\mathbf{x}, \mathbf{y}) \mathbf{t}(\mathbf{y}) w_s(\mathbf{x}, \mathbf{y}) w_m(\mathbf{x}, \mathbf{y}) w_d(\mathbf{x}, \mathbf{y}) d\mathbf{y},$$



Artigo:

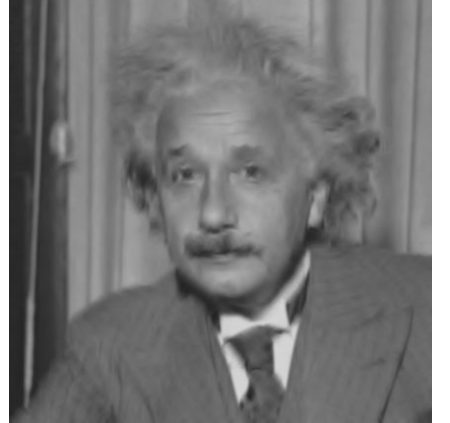
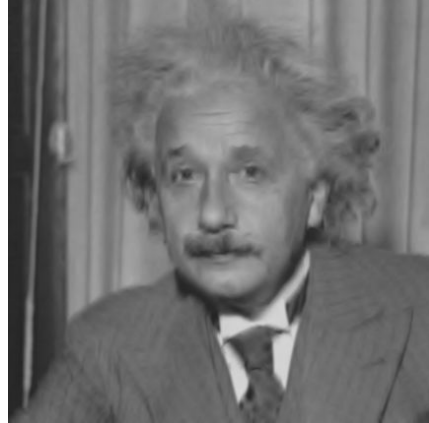
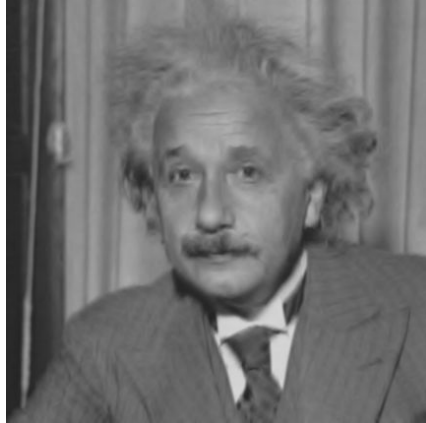
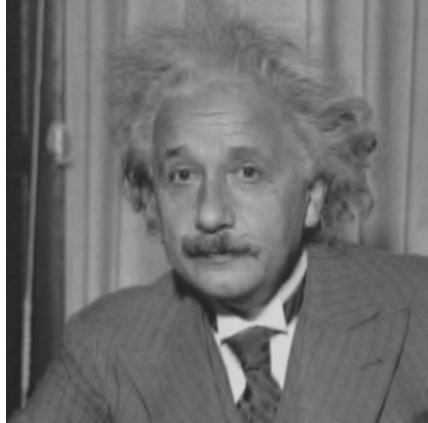
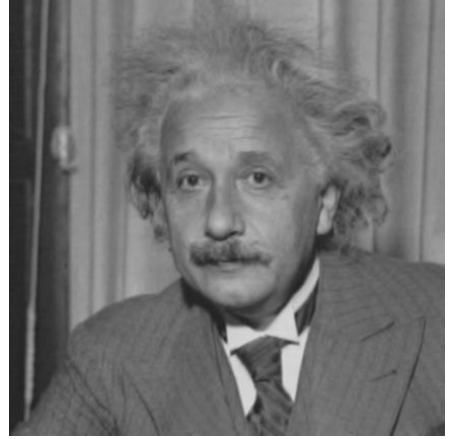
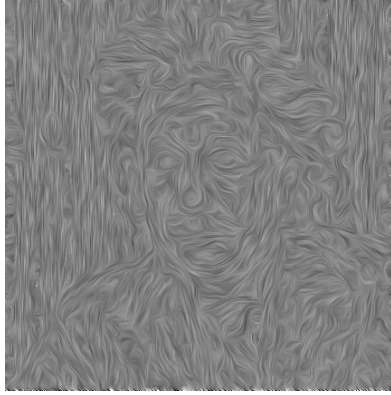
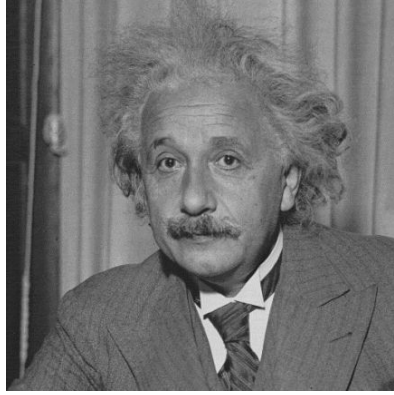


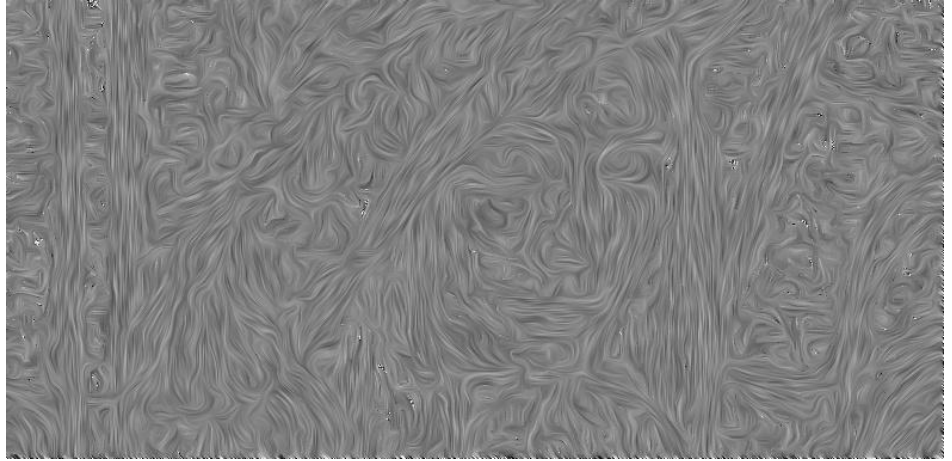
Suavização de regiões

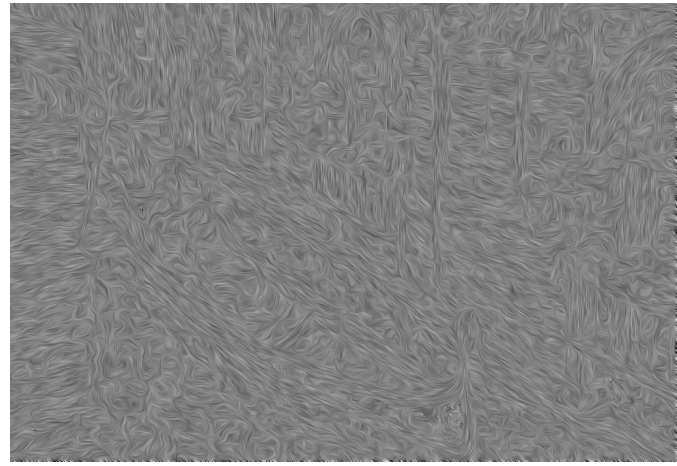
- Aplicação de filtro bilateral levando em consideração o fluxo (FBL).
- Aplicação iterativa (5 iterações).
- Dois filtros:
 - Na direção das bordas (seguindo o fluxo) (C_e) ($\sigma = 2.0$)
 - Na direção do gradiente (perpendicular ao fluxo) (C_g) ($\sigma = 0.3$)
- Composição dos filtros:
 - Função peso gaussiana sob a distribuição espacial (G).
 - Função peso gaussiana sob o domínio (cores) (h).
- Quantização

$$\mathcal{C}_e(\mathbf{x}) = \frac{1}{\nu_e} \int_{-S}^S I(c_{\mathbf{x}}(s)) G_{\sigma_e}(s) h(\mathbf{x}, c_{\mathbf{x}}(s), r_e) ds,$$

$$\mathcal{C}_g(\mathbf{x}) = \frac{1}{\nu_g} \int_{-T}^T I(l_{\mathbf{x}}(t)) G_{\sigma_g}(t) h(\mathbf{x}, l_{\mathbf{x}}(t), r_g) dt,$$







Extração de linhas

- Aplicação de diferença gaussiana seguindo o fluxo (FDoG).
- Aplicação iterativa.
- Estrutura do filtro:
 - Na direção das bordas (seguindo o fluxo) (C_e) ($\sigma_m = 3.0$)
 - Na direção do gradiente (perpendicular ao fluxo) (C_g) ($\sigma_c = 1.0$)

$$\mathcal{H}(\mathbf{x}) = \int_{-S}^S \int_{-T}^T I(l_{\mathbf{x},s}(t)) f(t) G_{\sigma_m}(s) dt ds,$$

- Binarização

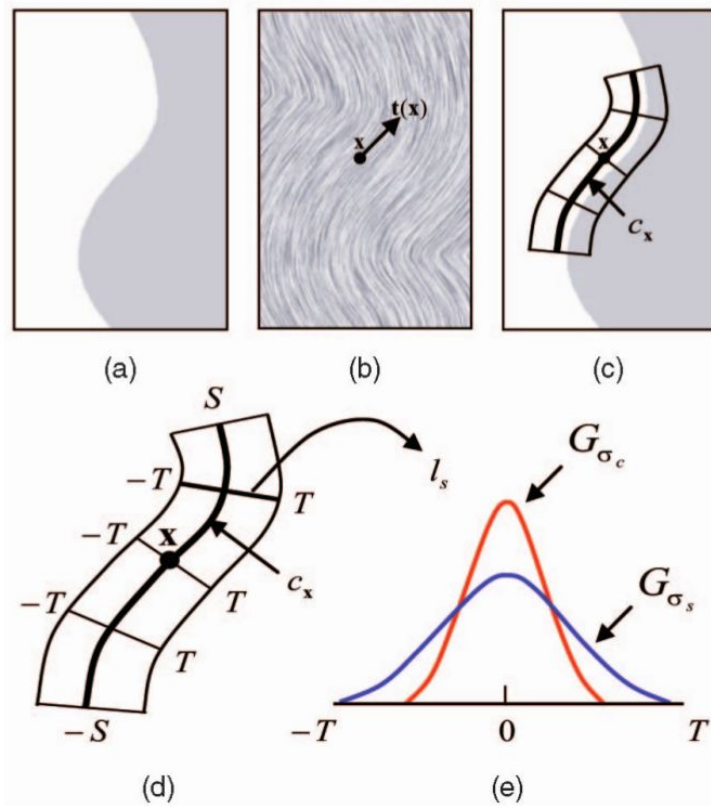


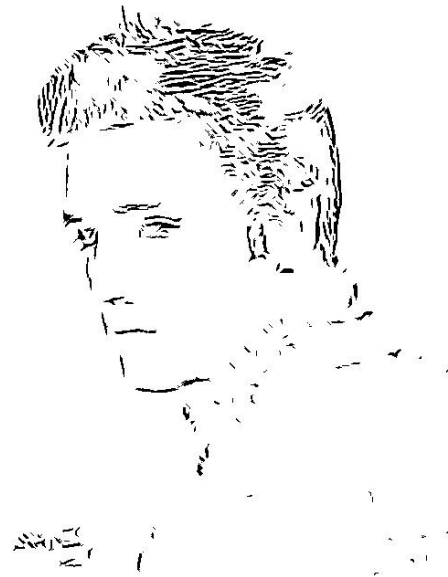
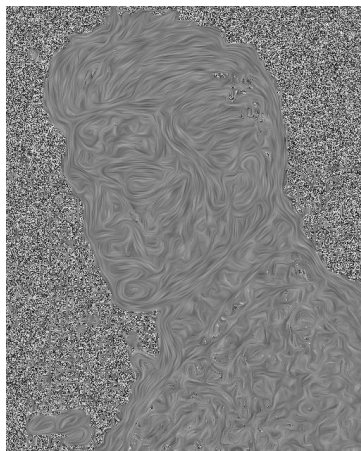
Fig. 6. FDoG filtering. (a) Input. (b) ETF. (c) Kernel at x . (d) Kernel enlarged. (e) Gaussian components for DoG.

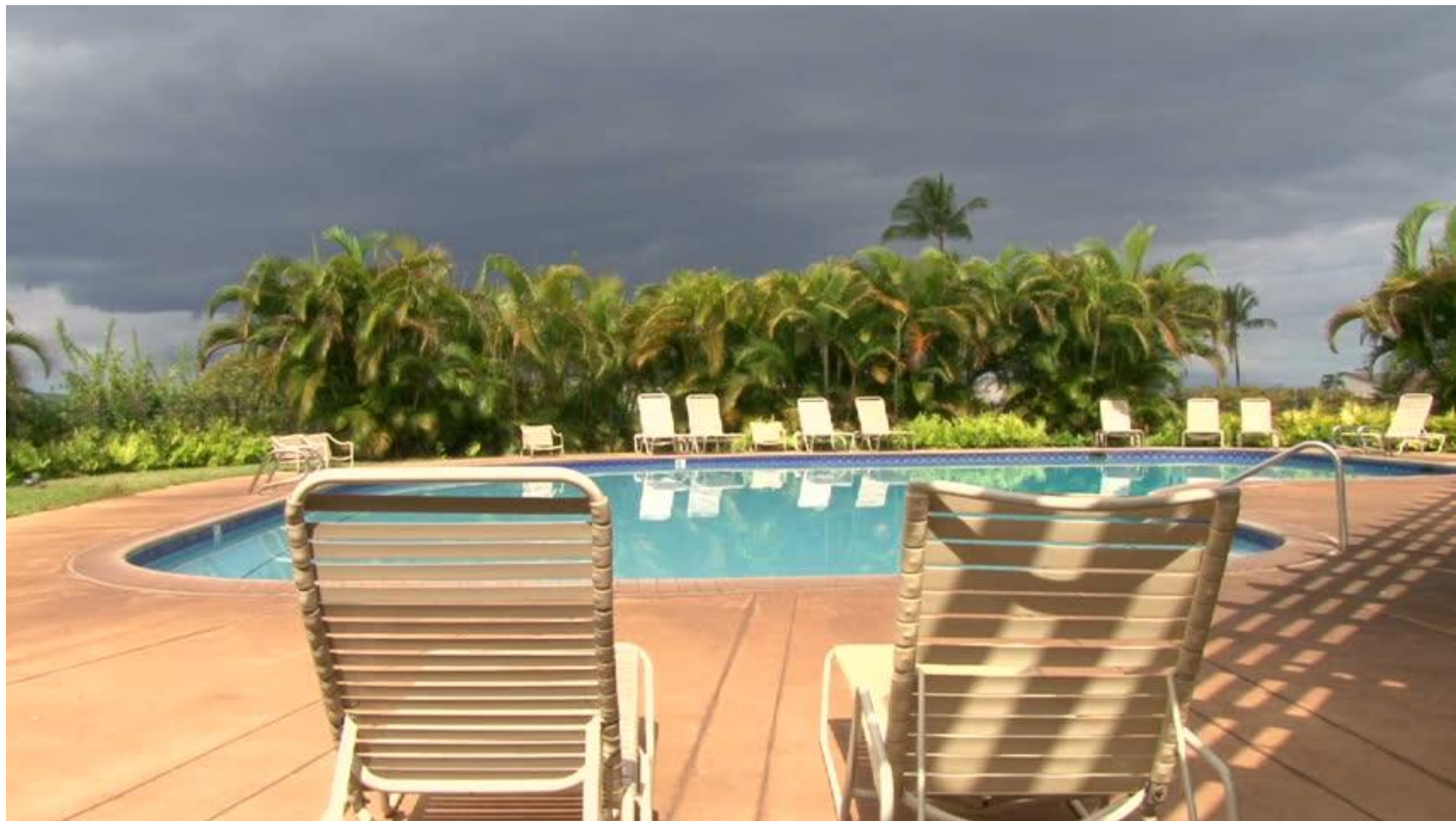
Otimização

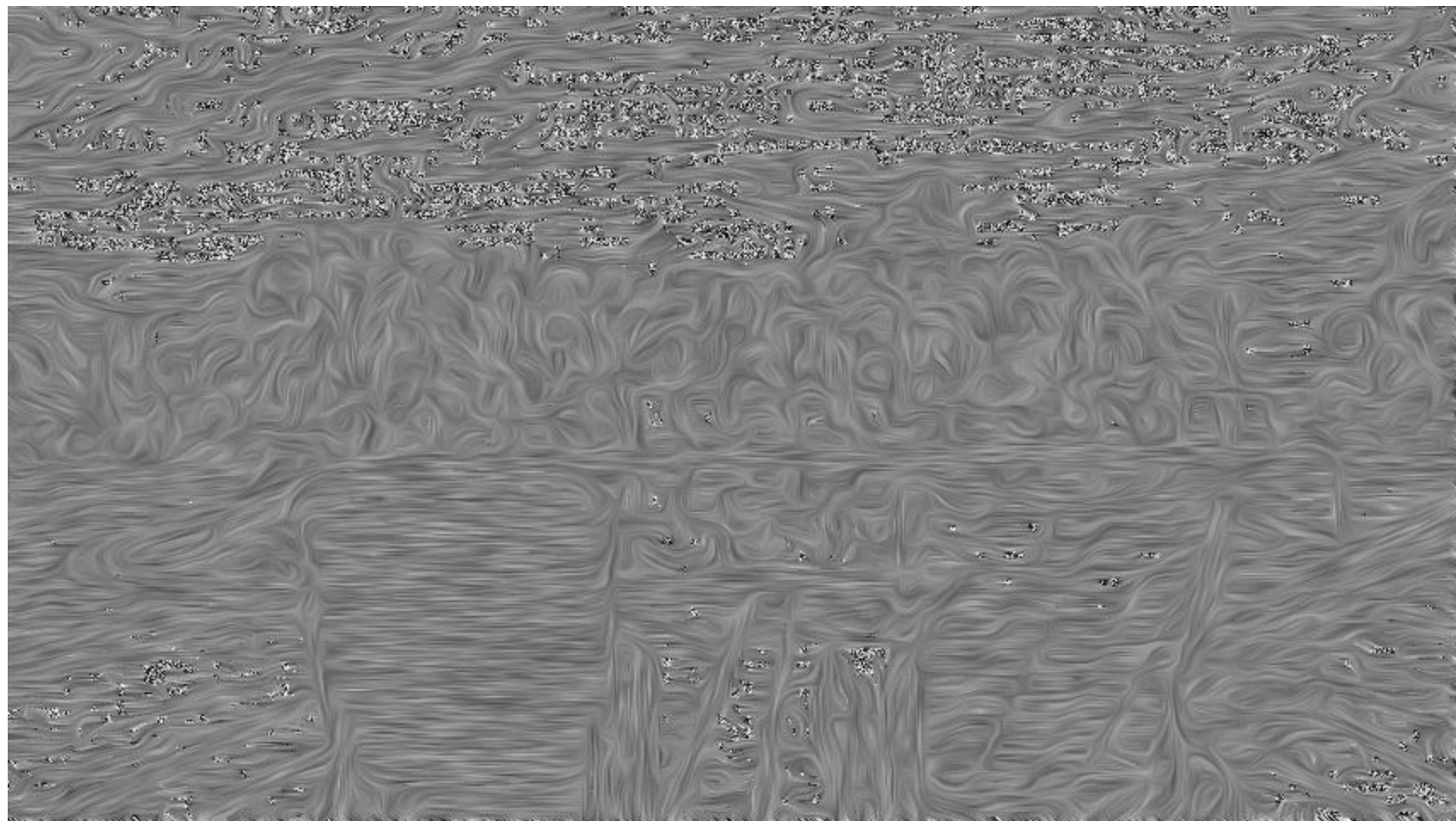
$$\mathcal{H}(\mathbf{x}) = \int_{-S}^S \int_{-T}^T I(l_{\mathbf{x},s}(t)) f(t) G_{\sigma_m}(s) dt ds,$$

$$\mathcal{H}_g(\mathbf{x}) = \int_{-T}^T I(l_{\mathbf{x}}(t)) f(t) dt,$$

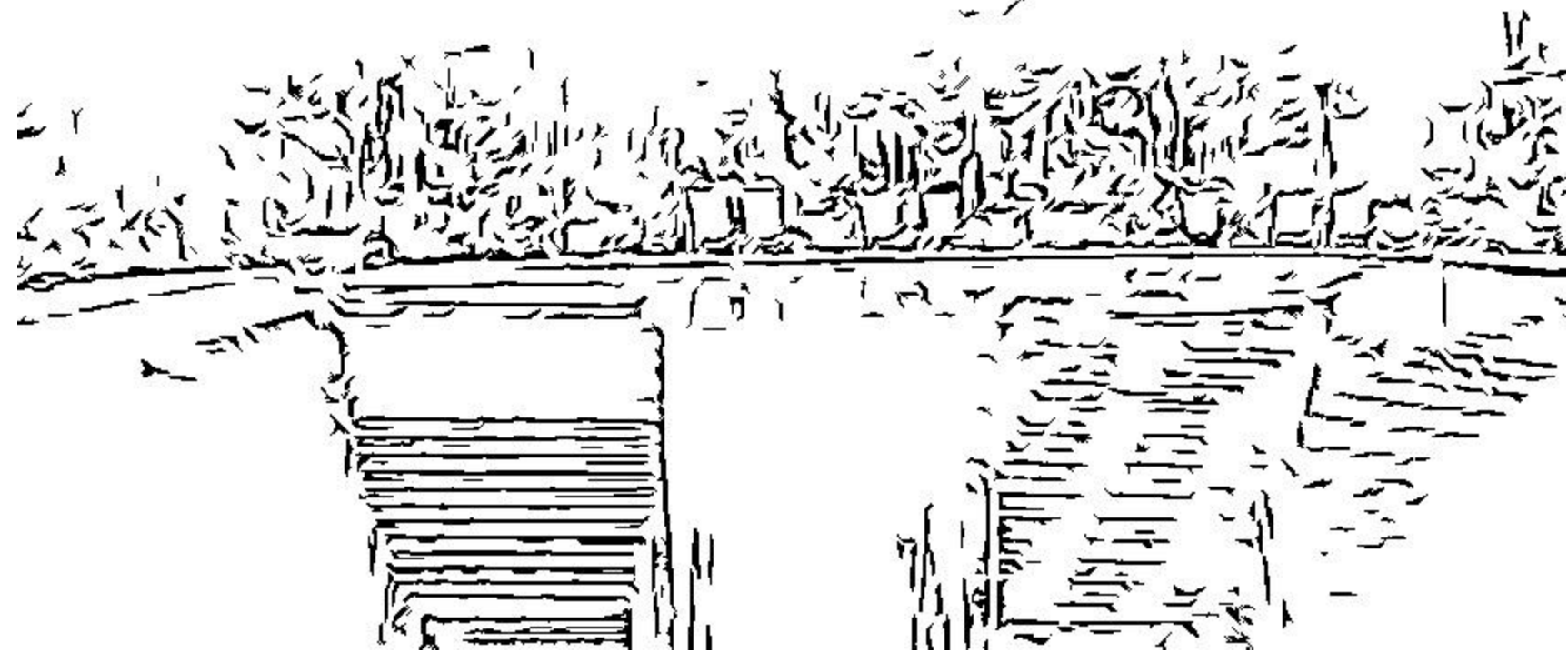
$$\mathcal{H}_e(\mathbf{x}) = \int_{-S}^S G_{\sigma_m}(s) \mathcal{H}_g(c_{\mathbf{x}}(s)) ds,$$













Referências

- H. Kang, S. Lee, C. K. Chui. “Flow-Based Image Abstraction”. IEEE Transactions on Visualization and Computer Graphics, VOL. 15, NO. 1, January/February 2009.
- <http://tatsy.github.io/lime/>

Implementação em:

https://github.com/mateuscgc/image_abstraction