Low-quality Motion Deblurring from a Single Image

Eduardo Stuani
Gustavo Prolla Lacroix

January 2024

1 Introdução

Borramento causado por movimento da câmera é um dos artefatos mais comuns na fotografia. Seja porque não há luz o suficiente para usar uma velocidade do obturador maior, seja porque tomamos muito café no dia, diversas vezes tiramos uma foto e o resultado é desapontante.

Restaurando uma imagem não borrada a partir de uma única fotografia borrada é um problema fundamental na área de imagens digitais. Se assumirmos que o kernel do borramento é *shift-invariant* – isto é, a imagem pode ser transladada e o borramento que ele causa continuará o mesmo – o problema reduz à deconvolução. Mais especificamente, se assumirmos que o kernel é conhecido trata-se de deconvolução não-cega. Caso contrário, trata-se de deconvolução cega, em que é preciso estimar o kernel do borramento e a imagem latente – um problema bem mais difícil.

2 Solução Proposta

Métodos de deconvolução atuais têm um bom desempenho quando a imagem borrada não contém ruído e o kernel de borramento não contém erro. Com isso, tem-se a ideia de que um modelo melhor do ruído e um manejamento explícito dos artefatos gerados a partir de erros na estimação do kernel deve melhorar significativamente o resultado da deconvolução.

Três contribuições são propostas pelo autores [SJA08]. A primeira é um novo modelo do ruído, em que sua distribuição é *espacialmente* randômica. Isso auxilia na distinção dos erros da estimação do ruído dos erros da estimação do kernel. A segunda é uma restrição na suavidade da imagem latente que é imposta onde a imagem observada possui baixo contraste. Isso auxilia a reduzir ruído periódico tanto nas regiões suaves quanto nas regiões texturizadas próximas. A terceira é um algoritmo eficiente de otimização. O algoritmo consiste em iniciar com uma estimativa do kernel imprecisa – por exemplo, uma linha reta ou pontos espalhados – e iterativamente estimar a imagem latente e o kernel de borramento.

2.1 Modelo Probabilístico

O modelo proposto unifica deconvolução cega e não-cega em um único problema de máximo a posteriori (MAP). Podemos modelar o borramento de uma imagem como o processo de convolução

$$I = L \circledast f + n$$

onde I é a imagem observada, L é a imagem latente não-borrada, f é o kernel do borramento e n é o ruído aditivo. Pelo teorema de Bayes, temos que

$$p(L, f|I) \propto p(I|L, f)p(L)p(f) \tag{1}$$

onde $p(I|L, f)$ é a função de verossimilhança, e $p(L)$ e $p(f)$ são as probabilidades a priori da imagem latente e do kernel do borramento.

Verossimilhança $p(I|L, f)$**.** A verossimilhança de uma imagem observada dada a imagem latente e o kernel de borramento é baseada no modelo de convolução $n = L \circledast f - I$. O ruído da imagem n é modelado como um conjunto de variáveis aleatórias independentes e identicamente distribuídas, cada uma seguindo uma distribuição Gaussiana.

Nessa formulação, modelamos a aleatoriedade espacial do ruído restringindo várias ordens de sua derivada. Escrevemos as derivadas parciais de n nas duas direções como $\partial_x n$ e $\partial_y n$, e as computamos utilizando diferença anterior. Como cada n é uma variável aleatória independente e identicamente distribuída seguindo uma distribuição Gaussiana com desvio padrão ζ_0, é provado por [Sim02] que as derivadas parciais também seguem distribuições Gaussianas $N(\partial_x n_i|0, \zeta_1)$ e $N(\partial_y n_i|0, \zeta_1)$ com desvio padrão $\zeta_1 = \sqrt{2} \cdot \zeta_0$.

Na seguinte formulação, utilizamos ∂^* para representar o operador de qualquer derivada parcial e $\kappa(\partial^*)$ para representar a sua ordem.

$$
\begin{aligned}
p(I|L,f) &= \prod_{\partial^* \in \Theta} \prod_i N(\partial^* n_i | 0, \zeta_{\kappa \partial^*}) \\
&= \prod_{\partial^* \in \Theta} \prod_i N(\partial^* I_i | \partial^* I_i^c, \zeta_{\kappa \partial^*})
\end{aligned}
\tag{2}
$$

onde $I_i \in I$ é o valor do pixel, I_i^c é o pixel com coordenada i na imagem reconvolucionada $I^c = L \circledast f$, e Θ é o conjunto de todos os operadores de derivadas parciais que utilizamos, $\Theta = \{\partial_0, \partial_x, \partial_y, \partial_{xx}, \partial_{xy}, \partial_{yy}\}$.

Prior $p(f)$. Modelamos os valores do kernel do borramento seguindo uma distribuição exponencial. O kernel normalmente identifica o caminho percorrido pela câmera ao tirar a foto, portanto possui muitos valores próximos de zero.

$$p(f) = \prod_j e^{-\tau f_j}, f_j \geq 0$$

Onde j indexa os elementos do kernel.

Prior $p(L)$. Construímos $p(L)$ com duas partes: o prior local e o prior global. O objetivo disso é ter um termo que reduz tanto a dificuldade do problema no geral quanto a introdução de ruído periódico.

$$p(L) = p_g(L) p_l(L)$$

Prior Global $p_g(L)$. Pesquisas sobre a estatística de imagens naturais mostram que gradientes de imagens naturais seguem uma distribuição com caudas pesadas [RB05; WF07]. Representamos essa distribuição através de uma função definida por partes

$$
\Phi(x) = \begin{cases} -k|x| & x \leq l_t \\ -(ax^2 + b) & x > l_t \end{cases}
\tag{3}
$$

Setamos $k = 2.7$, $l_t =$, $a = 6.1 \cdot 10^{-4}$, $b = 5.0$. Tendo modelado a densidade logarítmica dos gradientes, a definição do prior global é escrita como

$$p_g(L) \propto \prod_i e^{\Phi(\partial L_i)}$$

Prior Local $p_l(L)$. Utilizamos a imagem borrada para restringir os gradientes da imagem latente de forma a reduzir a introdução de ruído periódico. Borramento por movimento pode ser considerado um processo de suavização. Em uma região localmente suave da imagem borrada – com pixels quase da mesma cor –, a região desborrada correspondente deve também ser suave.

Para formular o prior local, para cada pixel i na imagem borrada I, calculamos o desvio padrão de uma janela de mesmo tamanho que o kernel do borramento centrada nesse pixel. Se o desvio padrão for menor que um certo limiar t, determinamos que esse pixel i pertence à região Ω.

(a) Imagem

(b) Ω do canal vermelho

Figure 1: Cálculo da região Ω utilizando $t = 5$

Para todos os pixels em Ω, restringimos o gradiente da imagem desborrada a ser similar ao da imagem borrada. Definimos que os erros seguem uma distribuição Gaussiana de média zero e desvio padrão σ_1

$$p_l(L) = \prod_{i \in \Omega} N(\partial_x L_i - \partial_x I_i | 0, \sigma_1) N(\partial_y L_i - \partial_y I_i | 0, \sigma_1)$$

2.2 Otimização

O problema de encontrar o máximo a posteriori (MAP) é transformado em um problema de minimização de energia, que minimiza o logaritmo negativo da probabilidade que definimos, $E(L, f) \propto -log(p(L, f|I))$. A partir das definições acima obtém-se o seguinte

$$\begin{aligned} E(L, f) \propto & \left(\sum_{\partial^* \in \Theta} w_{\kappa(\partial^*)} ||\partial^* L \circledast f - \partial^* I||_2^2 \right) \\ & + \lambda_1 ||\Phi(\partial_x L) + \Phi(\partial_y L)||_1 \\ & + \lambda_2 \left(||\partial_x L - \partial_x I||_2^2 \circ M + ||\partial_y L - \partial_y I||_2^2 \circ M \right) + ||f||_1 \end{aligned} \tag{4}$$

Onde $||\cdot||_p$ é o operador de norma-p e $\circ$ é o operador de multiplicação ponto a ponto. Os quatro termos de (4) correspondem aos termos em (1) na mesma ordem. M é uma máscara binária que codifica $p_l(L)$. Para qualquer elemento $m_i \in M$, $m_i = 1$ se o pixel correspondente $I_i \in \Omega$. Os parâmetros $w_{\kappa(\partial^*)}$, λ_1 e λ_2 dos termos de probabilidade:

$$w_{\kappa(\partial^*)} = \frac{1}{\zeta^2_{\kappa(\partial^*)} \tau}, \qquad \lambda_1 = \frac{1}{\tau}, \qquad \lambda_2 = \frac{1}{\sigma_1^2 \tau}$$

Para estimar L e f, portanto, primeiro minimizamos a energia de L, E_L, mantendo f fixo, e depois minimizamos a energia de f, $E(f)$, mantendo L fixo. Isso permite a aplicação de alguns teoremas para que esse cálculo de minimização seja eficiente. A necessidade de um novo método de otimização vem do fato de que tentar minimizar (4) pelo método do gradiente é extremamente lento e não converge bem.

2.2.1 Otimização de L

Aqui, mantemos f fixo e otimizamos L. A energia E de (4) pode ser simplificada para E_L removendo os termos de valor constante.

$$\begin{aligned} E_L = & \left(\sum_{\partial^* \in \Theta} w_{\kappa(\partial^*)} ||\partial^* L \circledast f - \partial^* I||_2^2 \right) \\ & + \lambda_1 ||\Phi(\partial_x L) + \Phi(\partial_y L)||_1 \\ & + \lambda_2 \left(||\partial_x L - \partial_x I||_2^2 \circ M + ||\partial_y L - \partial_y I||_2^2 \circ M \right) \end{aligned} \tag{5}$$

Mesmo simplificada, E_L ainda é difícil de ser otimizada. Os autores proponhem uma mudança de variável e uma técnica de rebalanceamento de parâmetros iterativa. A intenção é separar as convoluções dos outros termos em (5) para que possam ser computadas rapidamente utilizando a transformada de Fourier.

O método consiste em introduzir um par de variáveis auxiliares $\Psi = (\Psi_x, \Psi_y)$ para $\partial L = (\partial_x L, \partial_y L)$, com a condição de que $\Psi \approx \partial L$. Portanto, para cada $(\partial_x L_i, \partial_y L_i) \in \partial L$ existe $(\psi_{x,i}, \psi_{y,i}) \in \Psi$. Transformamos a equação (5) em

$$\begin{aligned} E_L = & \left(\sum_{\partial^* \in \Theta} w_{\kappa(\partial^*)} ||\partial^* L \circledast f - \partial^* I||_2^2 \right) \\ & + \lambda_1 ||\Phi(\Psi_x) + \Phi(\Psi_y)||_1 \\ & + \lambda_2 \left(||\Psi_x - \partial_x I||_2^2 \circ M + ||\Psi_y - \partial_y I||_2^2 \circ M \right) \\ & + \gamma \left(||\Psi_x - \partial_x L||_2^2 + ||\Psi_y - \partial_y L||_2^2 \right) \end{aligned} \tag{6}$$

onde γ é um peso que é iterativamente aumentado, de modo que a condição $\Psi = \partial L$ seja eventualmente alcançada. Setamos $\gamma = 2$ e a cada iteração seu valor é dobrado.

Atualização de Ψ. Com L e $\partial^* L$ fixo, precisamos minimizar

$$\begin{aligned} E'_\Psi = & \lambda_1 ||\Phi(\Psi_x) + \Phi(\Psi_y)||_1 + \lambda_2 ||(\Psi_x - \partial_x I)||_2^2 \circ M \\ & + \lambda_2 ||(\Psi_y - \partial_y I)||_2^2 \circ M + \gamma ||\Psi_x - \partial_x L||_2^2 + \gamma ||\Psi_y - \partial_y L||_2^2 \end{aligned} \tag{7}$$

Podemos decompor Ψ em todos seus elementos $\psi_{i,x}$ e $\psi_{i,y}$ de forma que a equação se torna uma soma de sub-energias, $E'_\Psi = \sum_i \left(E'_{\psi_{i,x}} + E'_{\psi_{i,y}}\right)$, onde cada $E'_{\psi_{i,v}}$, $v \in \{x, y\}$, contém apenas uma variável $\psi_{i,v} \in \Psi_v$. Temos então que

$$E'_{\psi_{i,v}} = \lambda_1|\Phi(\psi_{i,v})| + \lambda_2 m_i(\psi_{i,v} - \partial_v I_i)^2 + \gamma(\psi_{i,v} - \partial_v L_i)^2 \tag{8}$$

Como $\Phi(x)$ é função definida por partes, cujo maior grau é dois, e os outros dois termos também são quadráticos, $E'_{\psi_{i,v}}$ tem valor mínimo quando sua derivada é zero. Portanto, calculamos esse valor para cada uma das partes de $\Phi(x)$ e escolhemos o menor que respeite o limite da concatenação l_t. Assim rapidamente obtemos um mínimo global para Ψ.

Atualização de L. Com Ψ fixo precisamos minimizar

$$\begin{aligned} E'_L = &\left(\sum_{\partial^* \in \Theta} w_{\kappa(\partial^*)}||\partial^* L \circledast f - \partial^* I||_2^2\right) \\ &+ \gamma||\Psi_x - \partial_x L||_2^2 + \gamma||\Psi_y - \partial_y L||_2^2 \end{aligned} \tag{9}$$

Como os principais termos de E'_L envolvem convolução, iremos operar no domínio frequência. Aplicando a transformada de Fourier e utilizando o teorema de Plancherel [Bra99], chegamos que o valor de L que minimiza E'_L é

$$L^* = \mathcal{F}^{-1}\left(\frac{\overline{\mathcal{F}(f)} \circ \mathcal{F}(I) \circ \Delta + \gamma\overline{\mathcal{F}(\partial_x)} \circ \mathcal{F}(\Psi_x) + \gamma\overline{\mathcal{F}(\partial_y)} \circ \mathcal{F}(\Psi_y)}{\overline{\mathcal{F}(f)} \circ \mathcal{F}(f) \circ \Delta + \gamma\overline{\mathcal{F}(\partial_x)} \circ \mathcal{F}(\partial_x) + \gamma\overline{\mathcal{F}(\partial_y)} \circ \mathcal{F}(\partial_y)}\right) \tag{10}$$

onde $\Delta = \sum_{\partial^* \in \Theta} w_{\kappa(\partial^*)}\overline{\mathcal{F}(\partial^*)} \circ \mathcal{F}(\partial^*)$ e $\overline{(\cdot)}$ representa o operador do conjugado. A divisão é feita ponto a ponto.

2.2.2 Otimização de f

Para parte de otimizar o kernel, mantemos L fixo e atualizamos f. A energia de f, $E(f)$, pode ser obtida de (4)

$$E(f) = \left(\sum_{\partial^* \in \Theta} w_{\kappa(\partial^*)}||\partial^* L \otimes f - \partial^* I||_2^2\right) + \|f\|_1 \tag{11}$$

O termo da convolução é quadrático com respeito a f. Dessa forma, podemos escrever (11) como uma multiplicação de matrizes, similarmente como foi feito em [Ara+20]. Além disso, removendo o somatório aplicando os operadores de Θ na multiplicação de $w_{\kappa(\partial^*)}$ com a respectiva derivada parcial ∂^* e a imagem em questão (latente L e degradada I), temos

$$E(f) = \|Af - B\|_2^2 + \|f\|_1 \tag{12}$$

Onde A é uma matriz de Toeplitz duplamente bloqueada, a qual é constituída pelos elementos que dependem da imagem latente L, f é o kernel que desejamos estimar e B é uma matriz na qual os elementos dependem da imagem degradada I. Realizando essa simplificação, a equação (12) pode ser resolvida pela abordagem de [KKB07], a qual minimiza a energia transformando essa otimização em um problema dual e o resolve utilizando o método de ponto interior de Newton. Dessa forma, são obtidos resultados bem próximos do mínimo global.

3 Implementação

Separamos o Algoritmo 1 em 4 funções: calcular o prior local, calcular Ψ, calcular L, e calcular f. As funções funcionam para matrizes 2D, portanto para imagens coloridas, passamos cada canal separadamente. Teoricamente, implementamos o que estava no artigo original, todavia temos dois problemas.

A otimização de L (Seção 2.2.1) não converge. Veja os critérios de parada no algoritmo do artigo original (Algoritmo 1). Entendemos ΔL como sendo a diferença dos valores calculados de L entre uma iteração e outra. Esse valor costuma ser maior que 1 não importa qual iteração (quase sempre $\Delta L > 500$). Nas primeiras iterações L parece ficar mais nítida, porém depois acaba voltando a borrar, eventualmente se tornando uma imagem de cor única. Se não limitarmos o número de iterações, o laço interno no Algoritmo 1 roda 63 vezes, quando γ causa um *integer overflow*.

O método de otimização de f (Seção 2.2.2) no artigo original transforma o problema da equação (12) em um problema dual para então aplicar o método de ponto interior de Newton para resolvê-lo. Tentamos implementar essa transformação do zero e até mesmo utilizar métodos prontos de otimização da biblioteca Scipy, porém não obtivemos resultados satisfatórios. Dessa forma, o caminho que seguimos para resolver a equação (12) foi utilizar o método lsq_linear() do Scipy. A função resolve este seguinte problema:

$$\begin{aligned} \text{minimizar} \quad & 0.5 \cdot \|Af - b\|^2 \\ \text{sujeito a} \quad & lb \leq f \leq ub \end{aligned} \tag{13}$$

Onde A é a matriz de Toeplitz, f é o kernel, b é um vetor proveniente do B original da equação (12), lb é o limite inferior (o qual setamos como 0) e ub é o limite superior (o qual setamos como mais infinito). Entendemos que esta abordagem não é a mais correta porém era a solução que encontramos no momento. Como resultado desta alteração, conseguimos obter apenas uma estimativa aproximada do kernel, não a melhor. Além disso, também é importante mencionar, que temos um parâmetro n_rows o qual manipula o tamanho que será extraído para construção da matriz A. Isso ocorre porque não é possível manter em memória toda matriz A se ela fosse computada a partir de todos os pixels da imagem. Consequentemente também são selecionados os pixels correspondentes para construir o vetor b.

Assim, o algoritmo proposto é descrito por:

Algorithm 1 Image Deblurring algorithm proposed in the original paper

Require: The blurred image I and a kernel estimate
  Compute smooth region Ω with $t = 5$.
  $L \leftarrow I$
  repeat
    repeat
      Update Ψ according to (7)
      Compute L according to (10)
    until $||\Delta L||_2 < 1 \cdot 10^{-5}$ e $||\Delta \Psi||_2 < 1 \cdot 10^{-5}$
    Update f according to (12)
  until $||\Delta f||_2 < 1 \cdot 10^{-5}$

4 Resultados

Como diz no título desse trabalho, os resultados são de baixa qualidade. Isso se deve a dois principais motivos. Primeiro, o fato de nem L nem f convergir com a nossa implementação. Segundo, não estamos fazendo nenhum tipo de tratamento de borda, algo que em métodos que utilizam a transformada de Fourier é fundamental. Além disso, os autores do paper original realizam vários ajustes pós-processamento, que nós não repetimos.

4.1 Deconvolução Não-cega

Para as figuras 2, 3 e 4, fizemos a convolução da imagem original com o kernel e somamos ruído gaussiano. Para gerar a imagem desborrada seguimos o seguinte algoritmo:

Algorithm 2 Nonblind Deconvolution

Require: The blurred image I and kernel f
  Compute smooth region Ω with $t = 5$.
  $L \leftarrow I$
  $i \leftarrow 0$
  while $i < 5$ **do**
    Update Ψ according to (7)
    Compute L according to (10)
    $i \leftarrow i + 1$
  end while
  Choose the best looking image

(a) Desborrada (b) Borrada (c) Original

Figure 2: Esquilo fofinho é borrado por kernel malvado.

(a) Desborrada (b) Borrada (c) Original

Figure 3: Onde eu queria estar agora.

(a) Desborrada (b) Borrada (c) Original

Figure 4: Bola de vidro nos ensina sobre o fenômeno da refração.

4.2 Deconvolução Cega

Para realizar a deconvolução cega, utilizamos seguinte algoritmo:

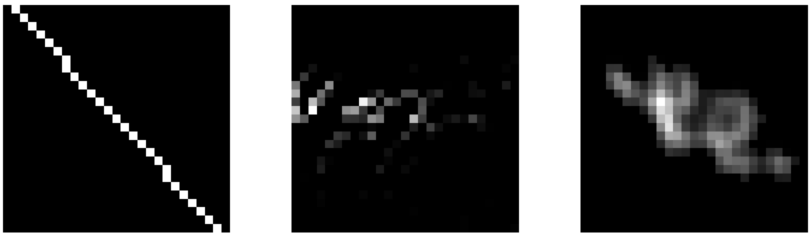
Abaixo se encontram imagens comparando os nossos resultados com os resultados obtidos pelos autores do artigo original [SJA08]:

Algorithm 3 Blind Deconvolution

Require: The blurred image I and a kernel estimate
  Compute smooth region Ω with $t = 5$.
  $L \leftarrow I$
  $i \leftarrow 0$
  while $i < 30$ **do**
    Update Ψ according to (7)
    Compute L according to (10)
    Update f according to (12)
    $i \leftarrow i + 1$
  end while
  Choose the best looking image

(a) Imagem Borrada (b) Nosso resultado (c) [SJA08]

Figure 5: Comparação das imagens



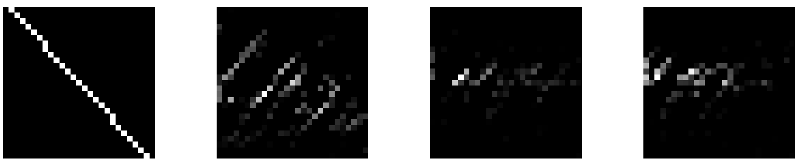
(a) Kernel Inicial (b) Nosso Kernel Final (c) [SJA08]

Figure 6: Comparação dos kernels

O resultado de cada iteração para a obtenção da imagem da Figura 5 e do Kernel da Figura 6, podem ser observados nas Figuras 7 e 8.

(a) Iteração 0 (b) Iteração 1 (c) Iteração 4 (d) Iteração 8 (Imagem Final)

Figure 7: Resultado por iteração da imagem



(a) Iteração 0 (b) Iteração 1 (c) Iteração 4 (d) Iteração 8 (Kernel Final)

Figure 8: Resultado por iteração do kernel


(a) Imagem Borrada


(b) Nosso resultado


(c) [SJA08]


(d) Imagem Borrada


(e) Nosso resultado


(f) [SJA08]


(g) Imagem Borrada


(h) Nosso resultado


(i) [SJA08]

5 Conclusão

Deconvolução cega é um difícil. Ao longo deste trabalho, exploramos os desafios e as soluções propostas para o problema de desborramento de imagens a partir de uma única imagem. A partir da formulação de um modelo probabilístico que unifica os aspectos de deconvolução cega e não cega, implementamos um método que combina diversas técnicas para estimar tanto a imagem latente quanto o kernel de borramento.

No entanto, nossa implementação enfrentou dificuldades significativas na convergência dos métodos de otimização, particularmente na estimativa da imagem latente L e do kernel de borramento f, já que para este último, resolvemos utilizando mínimos quadráticos e não o método de ponto interior, como originalmente proposto pelos autores do artigo. Dessa forma, consideramos que os resultados obtidos não foram satisfatórios como esperávamos, devido a problemas de convergência e estabilidade numérica.

Dito isso, acreditamos que a realização do trabalho foi uma experiência bem construtiva para ambos os integrantes do grupo.

6 Sugestões

Inspirado na última aula dessa cadeira, propronhamos treinar uma rede neural para gerar estimativas do kernel de borramento a partir da imagem observada. O cálculo de f é provavelmente a etapa mais computacionalmente intensiva do algoritmo. Com uma estimativa inicial melhor de kernel, o algoritmo convergiria mais rapidamente, necessitando fazer esse cálculo menos vezes. Além disso, poderíamos adicionar um tratamento de bordas (que não realizamos) para reduzir o ruído periódico da imagem.

Referências

[Ara+20] Alexandre Araujo et al. "On Lipschitz Regularization of Convolutional Layers using Toeplitz Matrix Theory". In: *AAAI Conference on Artificial Intelligence*. 2020. URL: `https://api.semanticscholar.org/CorpusID:226281883`.

[Bra99] Ronald Bracewell. *The Fourier Transform And Its Applications*. McGraw-Hill, 1999.

[KKB07] Kwangmoo Koh, Seung-Jean Kim, and Stephen Boyd. "An Efficient Method for Large-Scale l1-Regularized Convex Loss Minimization". In: Jan. 2007, pp. 223–230. ISBN: 978-0-615-15314-8. DOI: `10.1109/ITA.2007.4357584`.

[RB05] S. Roth and M.J. Black. "Fields of Experts: a framework for learning image priors". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 2. 2005, 860–867 vol. 2. DOI: `10.1109/CVPR.2005.160`.

[Sim02] Marvin Simon. "Probability Distributions Involving Gaussian Random Variables, A Handbook for Engineers and Scientists". In: (2002). DOI: `10.1007/978-0-387-47694-0`.

[SJA08] Qi Shan, Jiaya Jia, and Aseem Agarwala. "High-quality Motion Deblurring from a Single Image". In: *ACM Transactions on Graphics (SIGGRAPH)* (2008).

[WF07] Yair Weiss and William T. Freeman. "What makes a good model of natural images?" In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. 2007, pp. 1–8. DOI: `10.1109/CVPR.2007.383092`.