Proceeding Series of the Brazilian Society of Applied and Computational Mathematics, v. 6, n. 1, 2018.

Trabalho apresentado no XXXVII CNMAC, S.J. dos Campos - SP, 2017.

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

Interpolação de imagens via minimização da função de variação total suavizada

Bruno Schelk[1]
Welington de Oliveira[2]
Instituto de Matemática e Estatística, UERJ, Rio de Janeiro, RJ

1 Introdução

O *problema de interpolação de imagem* – PII – (em inglês *Inpainting Problem*) é um processo de reconstrução de áreas perdidas, indesejadas, corrompidas ou deterioradas de imagens digitais, e envolve a aplicação de algoritmos sofisticados para substituir tais áreas, de modo que seus entornos não se destaquem com o restante da imagem.

Neste trabalho o PII é formulado como um problema de otimização irrestrita cujo objetivo é minimizar a *função de variação total*, uma função convexa mas não-diferenciável que mede a variação ("distância") entre píxeis de uma imagem digital. Para uma imagem colorida com resolução moderada, o problema de otimização não-diferenciável – POND – resultante é de grande porte, podendo ter mais de um milhão de variáveis. PONDs com estas dimensões são, em geral, muito difíceis de serem resolvidos. Métodos computacionais para esta classe de problemas são normalmente lentos e/ou exigem excessiva memória computacional. Com o intuito de aplicar métodos de otimização computacionalmente simples e eficientes, a função de variação total é "suavizada" e o problema de otimização resultante torna-se diferenciável; podendo então ser resolvido pelos métodos do gradiente [1] e gradiente acelerado [2]. A seguir estes métodos são analisados e aplicados ao PII com função de variação total suavizada pela regra de Huber [3].

2 Formulação matemática do problema e algoritmos

Uma imagem digital colorida pode ser vista matematicamente como uma matriz $x \in \Re^{n_1 \times n_2 \times 3}$, na qual sua resolução é definida pelo número de linhas n_1, número de colunas n_2 e escala RGB de três dimensões (*Red, Green* and *Blue*). A imagem x possui então $n_1 n_2$ píxeis $x_{i,j} \in \Re^3$, cada um sendo um vetor de três dimensões com valores que variam de 0 a 255 para definir diversas cores. Seja Λ um conjunto de índices (i, j) de píxeis $x_{i,j}$ de uma área da imagem x que deseja-se restaurar. A função de variação total é definida por

$$TV(x) = \sum_{(i,j)\in\Lambda} \left\| \begin{bmatrix} x_{i+1,j} - x_{i,j} \\ x_{i,j+1} - x_{i,j} \end{bmatrix} \right\|_2 = \sum_{(i,j)\in\Lambda} \sqrt{\|x_{i+1,j} - x_{i,j}\|_2^2 + \|x_{i,j+1} - x_{i,j}\|_2^2}. \quad (1)$$

Cada píxel $x_{i,j}$ da área a ser restaurada é comparado com os píxeis "vizinhos" abaixo $(x_{i+1,j})$ e à sua direta $(x_{i,j+1})$. Espera-se que tais píxeis sejam próximos uns dos outros,

[1]brunoschelk.b@gmail.com
[2]welington@ime.uerj.br

o que significa em termos práticos que píxeis vizinhos possuem cores semelhantes. Note que a função $TV(x)$ é não-diferenciável devido à norma Euclideana, o que dificulta o processo de otimização. Tratando-se do PII, uma alternativa satisfatória é substituir a norma Euclideana pela função diferenciável de perda de Huber [3]: $\Phi(z) = \|z\|_2 - \frac{1}{2}\tau$ se $\|z\|_2 \geq \tau$, e $\Phi(z) = \frac{1}{2\tau}||z||_2^2$ caso contrário, onde $\tau \approx 0$ é um parâmetro positivo dado. Com esta função, o PII é formulado como um problema de otimização convexa diferenciável

$$\min_{x \in \Re^{n_1} \times \Re^{n_2} \times \Re^3} f(x)\,, \quad \text{com} \quad f(x) := \sum_{(i,j)\in\Lambda} \Phi\left(\begin{array}{c} x_{i+1,j} - x_{i,j} \\ x_{i,j+1} - x_{i,j} \end{array}\right). \tag{2}$$

Algoritmos. São analisados dois principais algoritmos para resolver (2): o método do gradiente (MG) e o método do gradiente acelerado (MGA), que geram sequências de pontos (imagens) $\{x^k\}$ que se acumulam em uma solução de (2). A sequência gerada pelo MG tem a seguinte fórmula, com x^0 uma imagem inicial e $\alpha_k > 0$ o tamanho do passo, [1]:

$$x^{k+1} = x^k - \alpha_k \nabla f(x^k)\,, \quad k = 0, 1, \ldots$$

Diferentemente, o MGA gera duas sequências de pontos, iniciadas com $x^0 = y^0$:

$$x^{k+1} = y^k - \alpha_k \nabla f(y^k) \quad \text{e} \quad y^{k+1} = x^{k+1} + \frac{k-1}{k+2}(x^{k+1} - x^k)\,, \quad k = 0, 1, \ldots$$

Resultados. Quando aplicado a imagem inicial x^0 dada pela Figura 1(b) com 786432 variáveis (e valor funcional inicial $f(x^0) = 2597016$), o MG exigiu 358 segundos e forneceu uma estimativa do valor ótimo igual a $f(x_{MG}) = 854156$. O MGA necessitou de 322 segundos e forneceu $f(x_{MGA}) = 184473$. De forma geral, o MG mostrou-se mais rápido nas primeiras iterações, porém fornecendo pouco progresso nas iterações finais, onde o MGA apresentou-se mais rápido. Por esta razão foi testado um terceiro algoritmo denominado MG+MGA, que combina os dois métodos: o MG+MGA executa o MG até o tamanho do passo α_k tornar-se menor que um, e após continua o processo iterativo com o MGA. A imagem restaurada da Figura 1 foi obtida com o algoritmo MG+MGA em 273 segundos e forneceu uma melhor estimativa $f(x_{MG+MGA}) = 182270$ do valor ótimo.

(a) Original (b) Modificada (c) Restaurada

Figura 1: Resultado da aplicação do algoritmo MG+MGA à imagem Lena modificada. Apesar de pequenas imperfeições, a imagem restaurada é visualmente muito semelhante à imagem original. Para obter o mesmo valor funcional fornecido pelo algoritmo MG+MGA, o algoritmo MGA exigiu 306 segundos e 581 iterações. O MG precisou de mais de 14 mil iterações, gastou quase 8300 segundos e forneceu uma estimativa muito inferior.

Referências

[1] M. Solodov e A. Izmailov. Otimização vol. 2 - Métodos Computacionais. IMPA, 2012.

[2] Y. Nesterov: On an approach to the construction of optimal methods of minimization of smooth convex functions. Ekonom. i. Mat. Mettody 24, 509–517, 1988.

[3] P.J. Huber. Robust Estimation of a Location Parameter. Annals of Statistics. 53 (1): 73–101, 1964.

