O que os olhos não veem
Estudo de um algorítimo para detectar imagens digitais manipuladas

André Guaraldo*
Giuliano R. Pinheiro†
Oscar Esgalha‡
Anderson Rocha§

Abstract

Imagens digitais fazem parte do cotidiano de muitas pessoas; raras são as que não possuem acesso à essa tecnologia. Cresce também, em ritmo acelerado, a qualidade de softwares de edição de imagens, bem como as técnicas para a manipulação das mesmas. Em uma realidade na qual uma imagem pode servir de prova em um caso jurídico, é importante haver meios para se identificar imagens fraudulentas afim de garantir a integridade do caso. A evolução da qualidade das fraudes impossibilita essa identificação simplesmente através da observação, mesmo para olhos treinados. É necessário o desenvolvimento de algoritimos para esse fim. O nosso trabalho implementa e testa um algoritimo para identificar imagens alteradas por resampling (i.e. rotação, redimensão) proposto por Popescu [1] e tenta melhorá-lo.

1. Introdução

No passado, quando existiam apenas imagens analógicas, manipular imagens exigia técnicas manuais complexas, muito tempo, cuidado e paciência. O simples ato de remover uma pessoa de uma foto exigia uma grande quantidade de trabalho e de tempo (muitas horas, às vezes dias) na sala escura. Em vista disso, manipulações de imagens eram raras e, usadas principalmente para fins governamentais ou militares. [2]

Nos últimos anos, todavia, computadores e câmeras digitais tornaram-se melhores e mais acessíveis, bem como *softwares* de edição de imagens (*Adobe Photoshop* e *GNU Gimp*[1], por exemplo)[1]. No cenário atual, muitos possuem acesso à imagens digitais e manipulá-las pode ser feito por qualquer um, em poucos minutos. Com um mínimo de técnica em um software de edição, pode-se alterar significativamente uma imagem, de modo que isso não seja perceptível a olho nu.

Figura 1. Fotos originais (esquerda) e imagem manipulada (direita) por Brian Walski.

O uso de tais softwares para fins banais, como correção de iluminação, remoção de olhos vermelhos, entre outros, não interessa à computação forense. Entratanto, quando uma manipulação de imagem tem algum objetivo malicioso, é de grande interesse confirmar a autenticidade da imagem. A facilidade em manipular imagens diminuiu sua credibilidade nos tribunais [3]. Em um caso jurídico no qual a evidência mais forte para inocentar ou criminalizar uma pessoa seja uma imagem, é necessário conseguir separar fraudes de fotos autênticas.

Imagens fraudadas também aparecem na mídia com o intuito de mostrar uma situação sob outra perspectiva, como aconteceu com o fotógrafo Brian Walski em 2003 no jornal *Los Angeles Times* [2]. Na montagem do fotógrafo (ver

*Is with the Institute of Computing, University of Campinas (Unicamp). **Contact**: ra101487@students.ic.unicamp.br
†Is with the Institute of Computing, University of Campinas (Unicamp). **Contact**: ra108579@students.ic.unicamp.br
‡Is with the Institute of Computing, University of Campinas (Unicamp). **Contact**: ra108231@students.ic.unicamp.br
§Is with the Institute of Computing, University of Campinas (Unicamp). **Contact**: anderson.rocha@ic.unicamp.br

[1] Adobe Photoshop e GNU Gimp são softwares registrados com suas respectivas licenças por seus respectivos autores


UM JORNAL A SERVIÇO DO BRASIL ★ ★ ★ WWW.FOLHA.COM.BR

FOLHA DE S.PAULO

DIRETOR DE REDAÇÃO: OTAVIO FRIAS FILHO — DOMINGO, 5 DE ABRIL DE 2009 — EDIÇÃO SÃO PAULO, CONCLUÍDA ÀS 21H00 ★ R$ 4,00

história

CAPTURADO

Ficha de Dilma Rousseff no Dops

Grupo de Dilma planejou sequestro de Delfim Netto

Brasil gasta com 'spread' 2,5 vezes o orçamento da Saúde

Estudo calcula que pessoas físicas e jurídicas pagaram R$ 134,5 bi em 2008

Otan anuncia plano de enviar 5 mil militares ao Afeganistão

Figura 2. Falsa ficha criminal da então chefe de estado Dilma Rousseff no jornal Folha de São Paulo (2009) .

Figura 1) o soldado britânico parece estar orientando um iraquiano com uma criança no colo, mas esse momento nunca aconteceu, a foto que apareceu no jornal foi uma composição de outras duas fotos. Assim que a fraude foi descoberta, o fotógrafo foi demitido [2].

Outro exemplo, é o caso da suposta ficha criminal de Dilma Rousseff, que no dia 5 de abril de 2009 saiu no jornal Folha de São Paulo (ver **Figura 2**). Segundo a ficha, a então chefe de estado teve participação ativa na resistência durante o regime militar brasileiro, planejando roubos e sequestros. Afirmou-se que o documento foi encontrado no arquivo público de São Paulo, portanto, que era autêntico. Entretanto, uma análise forense revelou que a imagem da ficha não foi digitalizada no local, além de possuir características típicas de imagens criadas em computador e a tipografia não possuir características inerentes a uma máquina de escrever, counclui-se que a ficha era falsa [4].

Neste artigo será apresentado um método para separar possíveis fraudes, de imagens autênticas e, no caso da imagem suspeita, apontar as regiões que provavelmente foram adulteradas. O método, originalmente proposto por Popescu [1], identifica alterações através de uma técnica conhecida como *resampling*, na qual uma imagem ou um pedaço de imagem é rotacionado ou redimensionado. Utilizando essa técnica, é possível remover detalhes de uma foto ou adicionar elementos em uma foto vinda da própria imagem.

2. Estado da Arte

Ultimamente, muito esforço vem sendo colocado para resolver o problema de identificação de imagens digitais fraudulentas. Cada método foca em diferentes características que podem ser analisadas afim de autenticar uma imagem. Algumas técnicas tentam separar imagens naturais de imagens geradas por computador; uma delas propõe fazer isso analisando características físicas da imagem, que se mostrou melhor do que analisar se a imagem possui características de desenhos [5].

O método proposto por Fridrich *et al.* [3] detecta imagens alteradas por cópia e colagem, além de revelar onde estão as regiões copiadas. A técnica simples, porém robusta, consiste em comparar blocos de pixels da imagem de dois modos diferentes: no primeiro, compara se os blocos são exatamente iguais, no segundo, utilizando a transformada discreta do cosseno para fazer comparações aproximadas, e assim, tolerar possíveis ruídos ou alterações insignificantes na imagem. Apesar de uma boa precisão, o custo da técnica aumenta muito para imagens com grande quantidade de pixels.

A função resposta de uma câmera é uma função matemática que faz uma aproximação de cores em bordas com contrastes (por exemplo, a borda entre uma árvore verde escuro e um céu azul claro), tal aproximação é necessária devido à limitação da resolução de uma câmera digital, que captura um número limitado de informações da paisagem. É preciso estimar cores intermediárias para bordas afim de uma transição mais realista. Sabendo disso, Lint *et al.* [6] criaram uma técnica que determina funções utilizada para gerar as cores nas bordas da imagem, dividindo-a em blocos de pixels, e compara os resultados entre si ou com uma função resposta conhecida de uma dada câmera. Caso alguma região pareça usar uma função muito diferente pode-se suspeitar de uma montagem. Essa técnica não é muito eficiente em imagens nas quais as bordas não tenham alto contraste.

Popescu [1], em sua tese de doutorado, propõe diversos métodos estatísticos para se identificar se uma imagem é verdadeira. Dentre eles está o método para identificar manipulações por *resampling* no qual este trabalho se baseia. Dentre as outras técnicas abordadas pelo autor, está a detecção de dupla compressão JPEG, que pode identificar se houve uma colagem de duas ou mais imagens JPEGs diferentes, a detecção de regiões duplicadas da imagem (cópia-colagem) e a verificação da interpolação de cores da imagem. Quanto à última, a maioria das câmeras digitais não capturam todos os três canais de cores ao mesmo tempo (vermelho, verde e azul), mas apenas uma cor para cada pixel e depois, através de interpolação dos valores, as cores finais são calculadas. O método proposto consiste em tentar estimar como as cores foram interpoladas e tentar achar aberrações na imagem, isto é, regiões que apresentam outro comportamento de interpolação, que pode significar uma montagem.

3. Solução Proposta

3.1. Resampling

Quando uma imagem é redimensionada ou rotacionada, softwares de edição interpolam a amostra de pixels existentes para fazer uma aproximação para pixels novos, no caso de um aumento de tamanho, ou dos pixels restantes, no caso de diminuir o tamanho. O processo em ambos os casos é generalizado para redimensionar o número de amostras em um sinal por um fator de $\frac{p}{q}$. Seja um sinal unidimensional $x[t]$ com m amostras, a interpolação por um fator de $\frac{p}{q}$ que resulta em um sinal com n amostras ocorre em três passos[1]:

- **Up-sample** cria-se um novo sinal $x_u[t]$ com pm amostras, no qual:

$$x_u[i] = \begin{cases} x[t] & \text{se } i = pt \text{ tal que } t = 1, 2, ..., m \\ 0 & \text{para outros casos} \end{cases}$$

- **Interpolação** convolução de $x_u[t]$ com um filtro passa-baixas h:

$$x_v[t] = x_u[t] \star h[t]$$

- **Down-sample** cria-se um novo sinal $x_d[t]$ com n amostras no qual:

$$x_d[t] = x_v[i] \quad \text{com } i = qt \text{ tal que } t = 1, 2, ..., n$$

O que muda nos algoritimos de resampling que os softwares de edição usam (por exemplo, linear ou bicúbico) é o filtro $h[t]$ usado no passo **Interpolação** [1]. É importante notar que todos os processos feitos para realizar o resampling são lineares, assim, podem ser descritos por sistemas lineares no qual os fatores de interpolação sejam incógnitas. Esse conceito é facilmente estendível para um sinal bidimensional, os passos que o descrevem são os mesmos, correlações periódicas ocorrem de forma análoga.

3.2. O Algoritimo de Esperança e Maximização

Para determinar se um sinal sofreu resampling, será usado o algoritimo de Esperança e Maximização (EM), que é poderoso o suficiente para ao mesmo tempo estimar quais grupos de amostras que estão relacionadas aos vizinhos e como se dá essa relação. Se a relação entre os vizinhos da imagem fosse conhecida, o problema seria trivialmente resolvido, pois bastaria achar amostras que não seguem o padrao de relação, mas na prática tanto os grupos de amostras (pixels) relacionadas quanto o tipo de relação são desconhecidos, logo o EM se encaixa bem no problema [1]. Seja f uma matriz bidimensional cujos valores representam uma imagem em escala cinza, assume-se que os pixels de f pertencem a um de dois modelos:

- M_1, se os pixels são linearmente correlacionados aos vizinhos, isto é são descritos pelo modelo linear:

$$f(x, y) = \sum_{u,v=-N}^{N} \alpha_{u,v} f(x + u, y + v) + n(x, y)$$

  onde os parâmetros do modelo são dados por $\vec{\alpha} = \{\alpha_{u,v} | -N \leq u, v \leq N\}$, N é um número natural que descreve o número de vizinhos usado para gerar as correlações entre os pixels, $\alpha_{0,0} = 0$ e $n(x, y)$ descreve amostras independentes e igualmente distribuídas de uma distribuição gaussiana com média zero e uma variância desconhecida σ^2[1].

- M_2, se os pixels não são correlacionados aos vizinhos, ou seja, se foram gerados por outros processos[1].

O EM é um algoritimo iterativo de 2 passos; no passo E (Esperança) é estimada a probabilidade de cada pixel pertencer a um dos dois modelos, enquanto que no passo M (Maximização) as formas específicas de correlações entre os pixels é estimada. No passo E, a probabilidade de cada pixel pertencer à determinado modelo é calculada usando a regra de *Bayes*:

$$Pr\{f(x, y) \in M_1 | f(x, y)\} = \frac{Pr\{f(x,y)|f(x,y)\in M_1)\}Pr\{f(x,y)\in M_1)\}}{\sum_{i=1}^{2} Pr\{f(x,y)|f(x,y)\in M_i)\}Pr\{f(x,y)\in M_i)\}}$$

Em que as probabilidades $Pr\{f(x, y) \in M_1)\}$ e $Prf(x, y) \in M_2)$ são inicialmente assumidas como $\frac{1}{2}$. A probabilidade de um pixel $f(x, y)$ ser gerado por M_1 é dada por:

$$Pr\{f(x, y)|f(x, y) \in M_1)\} = \frac{1}{\sigma\sqrt{2\pi}} exp\left[-\frac{1}{2\sigma^2}\left(f(x, y) - \sum_{u,v=-N}^{N} \alpha_{u,v} f(x + u, y + v)\right)^2\right]$$

A variância σ^2 da distribuição gaussiana é estimada no passo M, da seguinte forma:

$$\sigma_{n+1} = \sqrt{\left(\frac{\sum_{x,y} w(x, y) r^2(x, y)}{\sum_{x,y} w(x, y)}\right)}$$

Os pixels que não se adequam ao modelo M_1 são assumidos como pertencentes ao modelo M_2, uma distribuição uniforme é assumida para essa probabilidade, ou seja, $Pr\{f(x, y)|f(x, y) \in M_2)\}$ é igual ao inverso do número de valores possíveis para $f(x, y)$, no nosso caso os valores que representam a imagem em escala cinza variam de 0 à 255, logo $Pr\{f(x, y)|f(x, y) \in M_2)\} = \frac{1}{256}$. O passo E necessita de uma estimativa dos coeficientes de $\vec{\alpha}$, os quais são escolhidos aleatoriamente para a iteração inicial. No passo M, uma nova estimativa de $\vec{\alpha}$ é calculada

a cada iteração usando mínimos quadrados ponderados, ou seja, minimizando a seguinte equação:

$$E(\vec{\alpha}) = \sum_{x,y} w(x,y) \left(f(x,y) - \sum_{u,v=-N}^{N} \alpha_{u,v} f(x+u, y+v) \right)^2$$

em que o peso $w(x,y) \equiv Pr\{f(x,y)|f(x,y) \in M_1)\}$ [1]. O algoritimo é finalizado quando a diferença normalizada entre o $\vec{\alpha_n}$ atual e o $\vec{\alpha_{n-1}}$ da última iteração é menor que um determinado ϵ.

4. Experimentos e Discussão

Antes de tentar incrementar a solução para o problema, tentamos simplesmente implementar a solução já existente, proposta por Popescu [1], no qual a nossa iria se basear. Na tese de doutorado do autor, especificamente no capítulo 2, dedicado à detectar fraudes criadas por *resampling*, é explicado a implementação do algoritimo EM para sinais unidimensionais e é apresentado um pseudo-algoritimo, que também trata somente do caso unidimensional. Segundo o autor, sabendo a implementação unidimensional seria simples deduzir a bidimensional. Partimos para várias tentativas frustradas de implementar o algoritimo em MATLAB na sua versão bidimensional. Não foi tão fácil quanto o esperado.

Depois de certo tempo, recorremos à ajuda de dois mestrandos, eles indicaram um artigo com a implementação do algoritimo em MATLAB feita pelo orientador de Popescu, Hany Farid. Essa implementação, tratava também na versão unidimensional, repetiu-se o argumento de que partindo dela a implemntação bidimensional é facilmente dedutível. Depois, os mestrandos compartilharam conosco o código que eles tinham produzido, para o caso bidimensional, também programado em MATLAB. A implementação deles tinha o mesmo problema que a nossa teve depois.

Pesquisando mais um pouco, descobrimos que na mesma tese de doutorado o capítulo 3 abordava outro algoritimo de detecção de fraudes que também utilizava o EM, neste era apresentado o algoritimo bidimensional e o seu respectivo pseudo-código. Seguimos as instruções e implementamos o algoritimo, todavia não conseguimos resultados satisfatórios, os valores não convergiam e o algoritimo falhava ao tentar estimar os mapas de probabilidade da imagem o os coeficientes de relação $\vec{\alpha}$. Tentamos reimplementar diversas vezes o algoritimo, reescrevendo do zero, revisando e testando de várias formas diferentes (usando imagens diferentes e alterando constantes), no entanto não conseguimos atingir os resultados que aparecem na tese de doutorado que estudamos.

Especificamente: tentamos usar imagens aleatórias (geradas dinamicamente), imagens geradas com coeficientes de correlação conhecidos, imagens comuns, imagens redimensionadas e rotacionadas, com interpolação bicúbica (usada pelo autor) e com interpolação linear (a mais simples), tentamos fixar o valor da variância gaussiana σ e também deixá-la ser calculada a cada iteração (feito pelo autor), tentamos diversos valores iniciais para a variância gaussiana σ, além de valores diferentes para a vizinhança N, tentamos formas diferentes de calcular o peso $Pr\{f(x,y)|f(x,y) \in M_1)\}$ e diferente formas para resolver o sistema de equações dos mínimos quadrados ponderados (assumindo $\alpha_{u,v}$ como constante e como variável). Assistimos também a evolução do mapa de probabilidade para cada iteração, confirmando que de fato o mapa não estava convergindo.

Um possível ponto que pode ter causado o defeito de nosso algoritimo é a solução dos mínimos quadrados ponderados. O autor não explica o método explícito que ele usa para resolver o sistema de equações e o desenvolvimento do sistema termina em um sistema confuso. Esse passo é muito importante para o algoritimo, pois com ele se encontra o $\vec{\alpha}$ da iteração seguinte. No pseudo-algoritimo esse passo é simplesmente descrito como a solução do sistema de equações. Interpretamos as palavras dessa parte do artigo com duas abordagens diferentes, entretanto nenhuma delas trouxe um bom resultado.

5. Conclusões

Apesar dos nossos esforços, não conseguimos implementar uma das soluções que queríamos. É interessante destacar que procuramos por outros artigos que usassem essa solução para algo, não encontramos nenhum artigo que o usasse e mostrasse como usá-lo, ou como implementá-lo. O máximo que encontramos foi um artigo de Li *et al.* [7] citando a solução de Popescu [1] e mostrando resultados obtidos pelo algoritimo.

Referências

[1] Alin C. Popescu. *Statistical Tools for Digital Image Forensics*. PhD thesis, DARTMOUTH COLLEGE, 2004. 1, 2, 3, 4

[2] Siome Goldenstein Anderson Rocha. *Atualizações em Informática (JAI)*, chapter CSI: Análise Forense de Documentos Digitais, pages 263–317. Sociedade Brasileira de Computação (SBC), 2010. 1, 2

[3] A Jessica Fridrich, B David Soukal, and A Jan Lukáš. Detection of copy-move forgery in digital images. In *in Proceedings of Digital Forensic Research Workshop*, 2003. 1, 2

[4] T. Boult A. Rocha, W. Scheirer and S. Goldenstein. Vision of the unseen: Current trends and challenges in digital image and video forensics. *ACM Computing surveys*, 2011. to appear. 2

[5] Tian-Tsong Ng, Shih-Fu Chang, Jessie Hsu, Lexing Xie, and Mao-Pei Tsui. Physics-motivated features for distinguishing photographic images and computer graphics. In *Proceedings of the 13th annual ACM international conference on Multimedia*, MULTIMEDIA '05, pages 239–248, New York, NY, USA, 2005. ACM. 2

[6] Zhouchen Lint, Rongrong Wang, Xiaoou Tang, and Heung-Yeung Shum. Detecting doctored images using camera response normality and consistency. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:1087–1092, 2005. 2

[7] Shu ping Li, Zhi Han, Yi zhen Chen, Bo Fu, Chunhui Lu, and Xiaohui Yao. Resampling forgery detection in jpeg-compressed images. In *Image and Signal Processing (CISP), 2010 3rd International Congress on*, volume 3, pages 1166 –1170, oct. 2010. 4