

Otimizando a Consistência de Cores em Coleções de Fotos

Isadora Schwaab

João S. Pastorello

Universidade Federal do Rio Grande do Sul

Instituto de Informática

Porto Alegre, Rio Grande do Sul, Brasil

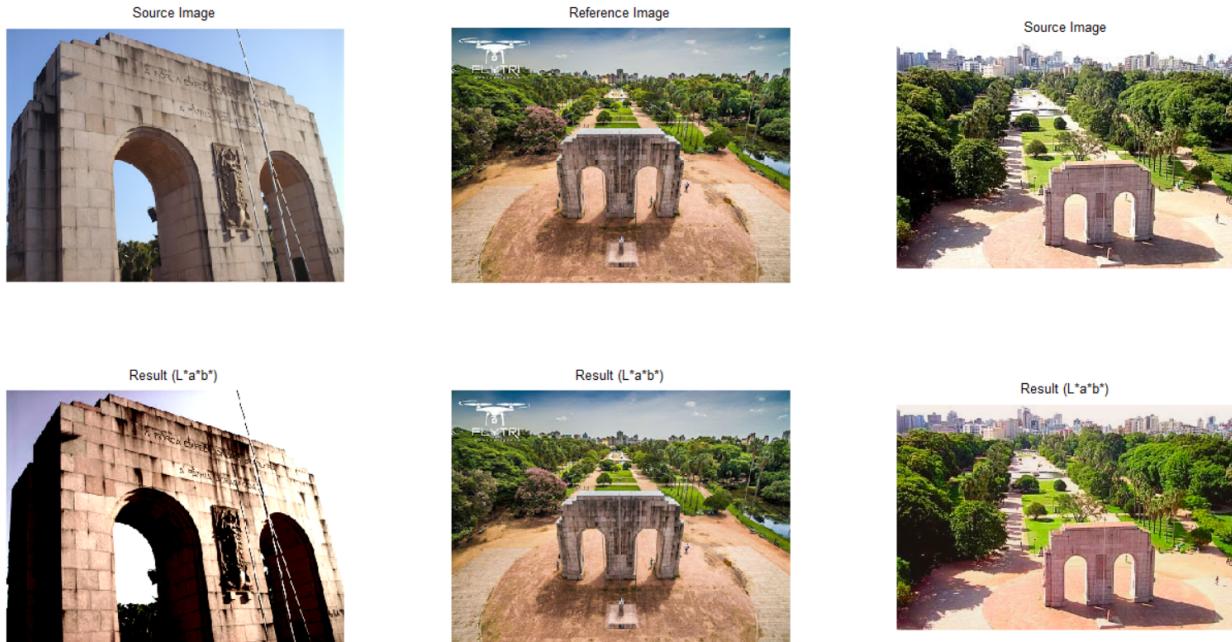


Figura 1: Otimizando automaticamente a consistência de cores de um conjunto de fotos da redenção através da nossa implementação com SURF

1 Introdução

A facilidade com a qual somos capazes de produzir fotografias digitais nos dias atuais representa tanto uma oportunidade quanto um desafio, tendo em vista a quantidade de registros que podemos fazer em uma única ocasião ou evento [HaCohen et al. 2013]. Quanto maior a quantidade de registros e a diversidade de ferramentas utilizadas para capturá-los, mais difícil se torna garantir a consistência de cor e contraste entre diferentes fotografias, ainda que retratem elementos iguais. Ainda assim, a grande maioria das ferramentas de edição disponíveis não oferece ferramentas de ajuste de cor automáticas que levem em conta as similaridades entre objetos e cenas. Alguns softwares comerciais podem aplicar um mesmo conjunto de operações de edição em múltiplas fotos automaticamente, ou transferir eficientemente as cores apenas de uma imagem para outra. No entanto, essas ferramentas não são suficientes para atingir de fato a consistência de aspecto entre conjuntos grandes de imagens. Quanto aos métodos descritos na literatura atual, muitos só podem realizar essa tarefa em conjuntos de fotos estáticas, ou não podem processar eficientemente transformações de cor mais complexas entre pares de fotos.

O método descrito por [HaCohen et al. 2013] busca atingir essa

consistência automaticamente utilizando um algoritmo de identificação de correspondências entre imagens para criar um grafo que representa o nível de semelhança entre um dado conjunto de imagens. Cada aresta desse grafo tem um valor associado que, quanto mais alto, representa uma conexão mais forte entre duas imagens do conjunto. O grafo é usado para encontrar o conjunto de transformações de cor que minimiza uma função de erro quadrático entre o conjunto de imagens, forçando a consistência de cores entre fotos que compartilham conteúdo. Esse método foi utilizado como base para as implementações descritas neste artigo.

2 Metodologia

Algoritmos para a extração de correspondência entre pares de imagens já foram desenvolvidos para a aplicação em diversos contextos, tipicamente relacionados ao alinhamento de imagens, fluxo óptico e combinação de imagens estéreo. Esses casos de uso normalmente operam com imagens de entrada muito similares, em que não há variação significativa no brilho e movimentação local entre as imagens [HaCohen et al. 2011].

Como buscamos encontrar conteúdo compartilhado entre fotografias capturadas com diferentes condições de iluminação e câmeras, é

necessário que o algoritmo de correspondência seja robusto a essas mudanças. Um método que satisfaz essas necessidades é o SURF (*Speeded Up Robust Features*) [Bay et al. 2006]. Extremamente rápido quando comparado a outras soluções, o SURF produz bons resultados mesmo lidando com transformações 3D, borramento e compressão JPEG. Ainda assim, para a solução do problema proposto, o SURF apresenta duas limitações importantes: a incapacidade de lidar com transformações não rígidas e a obtenção de correspondências esparsas. [HaCohen et al. 2011] apresenta o NRDC (*Non-Rigid Dense Correspondence*) e demonstra como a correspondência gerada pelo algoritmo é superior a outros métodos principalmente por superar as duas limitações descritas anteriormente.

Para a propagação de cores entre duas imagens, [Reinhard et al. 2001] descreve um algoritmo usando a média e o desvio padrão de cada canal de ambas as imagens no espaço de cores $L^*a^*b^*$. Para lidar com um conjunto de imagens, [HaCohen et al. 2013] sugere a construção de um grafo em que as arestas conectam fotos que compartilham conteúdo, permitindo que uma função de minimização seja aplicada a todo o grafo. Desse modo, também é possível a propagação indireta, em que características de imagens são propagadas a outras imagens mesmo que não haja correspondência direta entre o par.

3 Implementação

Nossa implementação foi desenvolvida no MATLAB R2014a, e utiliza os algoritmos de SURF ou NRDC para extrair a correspondência entre imagens. Foi utilizada a implementação de referência do NRDC disponibilizada na página do artigo.

A função principal contém parâmetros simples que podem ser modificados pelo usuário para decidir alguns fatores: qual o método de detecção de correspondências a ser usado (SURF ou NRDC), qual o espaço de cores a ser usado (RGB ou $L^*a^*b^*$) e se o resultado deve ser exibido ou exportado. Por padrão, o processamento utiliza SURF + $L^*a^*b^*$, e não salva o resultado no dispositivo, mas exibe a comparação entre as imagens originais e as imagens processadas pelo algoritmo. A pasta de entrada do algoritmo também é definida por um parâmetro interno, que armazena o nome da pasta onde estão as imagens a serem analisadas. Da mesma forma, os índices das imagens de referência, que não devem sofrer alterações, também podem ser ajustados.

A etapa de construção do grafo de correspondências, tanto utilizando SURF quanto NRDC, é feita no script principal, através de duas etapas: extração de características das imagens e construção de uma matriz de adjacência $n \times n$ composta por máscaras binárias, onde n é o número de imagens da entrada e uma máscara de uma linha i , coluna j tem o tamanho da i -ésima imagem, filtrando o conteúdo da imagem i que também está representado em alguma parte da imagem j .

Utilizando o SURF, a primeira etapa extrai descritores dos pontos mais significativos de cada imagem. A segunda etapa constrói duas máscaras binárias para cada par de imagens, colocando-as na matriz de adjacência. Utilizando o NRDC, o processo retorna, para um dado par de imagens com valores de cor normalizados, uma matriz DCF (*Dense Correspondence Field*) que representa um mapeamento dos pixels da primeira imagem para a segunda, e uma matriz de confiança, formada por valores que representam a confiabilidade

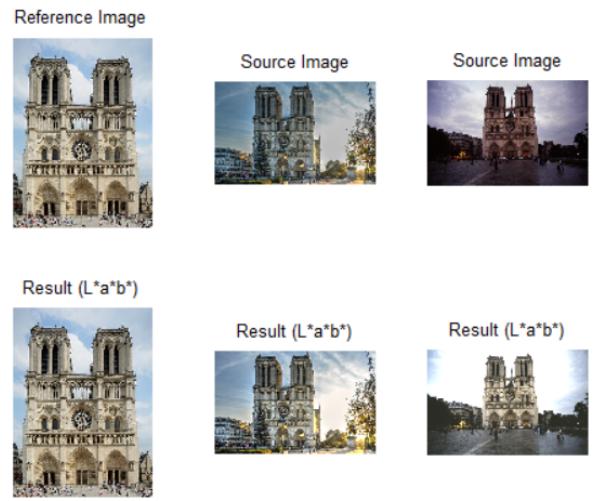


Figura 2: Conjunto NotreDame com SURF + $L^*a^*b^*$

no mapeamento de cada pixel. Esse mapeamento é usado na segunda etapa para construir as máscaras binárias no mesmo formato descrito anteriormente, apenas para os pares de imagens cujo mapeamento tem confiabilidade total maior do que 0.

Já durante a etapa da propagação direta, para cada imagem que não é marcada como imagem de referência, é computada uma média ponderada das médias e dos desvios padrão dos valores em cada canal de cor dos pixels das áreas correspondentes entre essa imagem e as imagens de referência. Além disso, é calculado um valor de peso da correspondência a partir do número de pixels com correspondência para graduar a transformação de cores de acordo com a relevância do conteúdo compartilhado. A média da imagem alvo da transferência de cores é dimensionada por uma média ponderada com a média da imagem que será alterada, usando o peso da correspondência calculado. Essa adaptação do método de [Reinhard et al. 2001] consegue, portanto, transferir as cores com intensidades diferentes para correspondências de níveis diferentes.

4 Resultados

As figuras 1 e 6 mostram casos em que a correspondência fornecida pelo SURF produz bons resultados, o que não acontece nas figuras 2, 4 e 7. Nesse sentido, o uso do NRDC trouxe resultados melhores e mais regulares, como pode ser visto nas figuras 3, 5 e 8. Além disso, o espaço de cor $L^*a^*b^*$ produz resultados mais satisfatórios que o RGB, como o esperado.

5 Conclusões

Foi possível implementar um programa que otimiza a consistência de conjuntos de imagens com relativo sucesso. A utilização do SURF se mostrou inadequada para alguns tipos de imagens, ao contrário do NRDC, que trouxe resultados satisfatórios de forma consistente para diversos conjuntos de imagens. Em alguns casos, a propagação de cores não foi eficaz, principalmente devido à não implementação da propagação indireta. Ainda assim, a adaptação do método de



Figura 3: Conjunto People com SURF + RGB

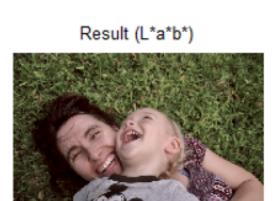
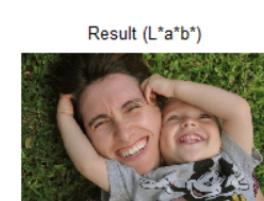
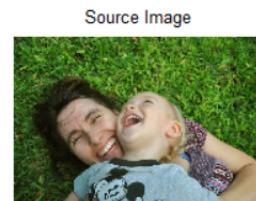


Figura 4: Conjunto People com NRDC + RGB

Figura 5: Conjunto People com SURF + L*a*b*

[Reinhard et al. 2001] se mostrou interessante devido à simplicidade e rapidez.

6 Possíveis Melhorias

Os resultados obtidos mostraram que o SURF não é capaz de computar boas correspondências entre imagens para o propósito de otimizar a consistência de cores, diferentemente do NRDC. No entanto, a implementação do NRDC utilizada é excessivamente lenta para ser computada mesmo em um conjunto de poucas imagens. Desse modo, uma implementação mais eficiente e/ou simplificada do NRDC pode ser considerada para agilizar a construção do grafo. Além disso, estratégias para a classificação de probabilidade de correspondência e para a predição de conexão entre imagens podem acelerar significativamente o processamento sem prejudicar o resultado final de forma relevante [HaCohen et al. 2013].

Além disso, o algoritmo de transferência de cores utilizado não

consegue reproduzir variações complexas como na saturação [HaCohen et al. 2011]. Além disso, nossa adaptação do método de [Reinhard et al. 2001] para um conjunto de imagens considera apenas a propagação direta da aparência, comprometendo a otimização da consistência em conjuntos onde a correspondência entre imagens não é tão presente. A transformação adotada por [HaCohen et al. 2013] suporta a propagação indireta, mas também não lida muito bem com a saturação. Ainda assim, promoveria uma melhoria significativa sobre o método implementado.

Referências

- Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. 2006. SURF: Speeded Up Robust Features. In *European Conference on Computer Vision*.
 Yoav HaCohen, Eli Shechtman, Dan B Goldman, and Dani Lischinski. 2011. Non-Rigid Dense Correspondence with Applications for Image Enhancement. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2011)* 30, 4 (2011), 70:1 – 70:9.

**Figura 7:** Conjunto Room com SURF + $L^*a^*b^*$ **Figura 8:** Conjunto Room com NRDC + $L^*a^*b^*$

- Yoav HaCohen, Eli Shechtman, Dan B Goldman, and Dani Lischinski. 2013. Optimizing Color Consistency in Photo Collections. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2013)* 32, 4 (2013), 85:1 – 85:9.
 Erik Reinhard, Michael Ashikhmin, Bruce Gooch, and Peter Shirley. 2001. Color Transfer between Images. *IEEE Computer Graphics and Applications* 21 (10 2001), 34–41.