

Como separar o trigo do joio?

Ou: Como selecionar a melhor fotografia de um conjunto de fotografias semelhantes

André Alves, Fernando P. Birra, and João M. Lourenço

NOVA Laboratory for Computer Science and Informatics
Departamento de Informática, Faculdade de Ciências e Tecnologia
Universidade NOVA de Lisboa, Portugal

Resumo O advento da fotografia digital está na base de uma clara mudança de paradigma no processo de gestão da fotografia por amadores. Porque tirar mais uma fotografia agora não representa qualquer custo adicional, é frequente tirarem-se múltiplas fotografias ao mesmo sujeito, na expectativa de que uma delas corresponda aos padrões de qualidade desejados, em termos de iluminação, foco e enquadramento. Assumindo que a questão do enquadramento se resolve facilmente recorrendo ao recorte (*crop*) da fotografia, tem-se ainda assim que selecionar qual das várias fotografias bem enquadradas, tecnicamente parecidas em termos de iluminação e foco, vamos guardar (e por oposição quais vamos descartar). A escolha da melhor fotografia com base na observação visual em ecrã de computador é um processo muito pouco preciso e, portanto, gerador de sensações de insegurança que resultam, muitas vezes, na opção de não descartar nenhuma das várias fotografias semelhantes. Neste artigo propomo-nos endereçar a questão de como ajudar um fotógrafo amador a selecionar a melhor fotografia de um conjunto de fotografias semelhantes em termos técnicos (foco e iluminação) e de enquadramento. Este processo é baseado num *workflow* suportado por um pacote de *software*, que com alguma ajuda do utilizador permite ordenar um conjunto de fotografias semelhantes, sendo assim possível escolher aquela que melhor corresponde às expectativas e dando segurança e conforto na eventual eliminação das restantes.

1 Introdução

O registo da luz de uma cena por parte do sensor de uma câmara digital, como por exemplo uma *Digital Single-Lens Reflex*, pode ter resultados bastante diferentes consoante os parâmetros definidos na câmara, sendo assim necessário um certo grau de perícia para dominar este processo. Desta forma, com o objetivo de aumentar a probabilidade de se obter o resultado pretendido, é prática comum entre os fotógrafos amadores tirar várias fotos ao mesmo motivo e em modo automático, na esperança de que uma delas satisfaça os seus requisitos.

Algumas das fotos capturadas podem não corresponder às expectativas. Uma *focagem imprópria*, possivelmente causada por uma má identificação do ponto

de focagem ou por uma má definição da profundidade de campo, pode resultar numa fotografia em que o motivo se encontra desfocado, contribuindo assim para o descontentamento do fotógrafo. Por outro lado, caso a câmara se mova durante o momento da exposição, o resultado será uma fotografia que contém uma *desfocagem geral*, sem qualquer claro centro de interesse. Ainda assim, a desfocagem pode ser causada propositadamente pelo utilizador com o objetivo de dar a impressão de *movimento na cena*. Dado que este efeito requer um certo grau de perícia, nem sempre poderá ser possível obtê-lo numa única tentativa, resultando em várias fotos muito parecidas. A sensibilidade do sensor da câmara perante a luz existente resultará em fotografias com *diferentes níveis de ruído*, sendo que o ruído pode ser subjetivamente considerado um fator degradante para a qualidade da fotografia. No aspeto da cor, esta tem um papel integral não só na percepção visual mas também nas emoções que uma imagem cria no espetador. Para se obter a exposição ideal, o montante correto de luz deve ser capturado. Especialmente quando usa o modo automático da câmara pode ocorrer que o fotógrafo capture imagens com *diferentes níveis de exposição*. Uma captura excessiva de luz dá origem a uma fotografia sobre-exposta, dominada por tons claros. Por outro lado uma captura escassa de luz dá origem a uma fotografia sub-exposta, dominada por tons escuros. Ainda na questão da cor, tem-se que as câmaras digitais tendem a capturar a luz com as suas características primitivas, podendo assim capturar cores que fogem à percepção do sistema visual humano. Como resultado tem-se fotografias com um *elenco de cores irrealista*, normalmente chamadas de fotografias quentes, predominadas por cores avermelhadas ou fotografias frias, predominadas por cores azuladas.

Para além de todos os desafios que o fotógrafo enfrenta no momento da captura da fotografia, este tem de lidar com outros problemas adicionais. O grande desafio ocorre na realidade quando, mais tarde, o fotógrafo transfere o conjunto de fotos semelhantes para o computador na expectativa de escolher a que considera melhor e se sente frustrado pela incapacidade de escolher com segurança quais as fotos a preservar e quais eliminar. É neste processo que este artigo se foca. Os principais desafios na escolha da melhor foto são: como lidar com o *tamanho da amostra*, podendo o número de fotos variar entre um par e dezenas; como lidar com a lentidão resultante de manipular *ficheiros de grande dimensão*; como lidar com o facto de que a *resolução* de uma fotografia tirada com uma DSLR ser normalmente muito maior do que a resolução de um ecrã de computador, obrigando à redução do tamanho da imagem com consequências na qualidade da mesma, ou a um processo de análise de apenas partes da imagem. Desta forma o processo de escolha da melhor foto requer necessariamente que se navegue pelo universo de fotos existentes, podendo-se tornar um processo demorado, tedioso e propenso ao erro, gerador portanto de insegurança e frustração.

Neste artigo apresentamos uma proposta de uma metodologia, acompanhada por uma ferramenta de *software* inovadora, que visa ajudar o utilizar no processo de seleção da melhor foto entre um conjunto de fotos semelhantes. Para tal é feita uma análise aos problemas anteriormente referidos (focagem, desfocagem, estimativa de ruído e análise da cor) em fases separadas, sendo que o utiliza-

dor pode ir sucessivamente rejeitando as fotografias que não preenchem os seus requisitos, até ficar com uma amostra bastante reduzida, tornando assim este processo numa experiência gratificante. A implementação da análise computacional integrada na ferramenta de apoio ao *workflow* de seleção é baseada num conjunto de algoritmos já disponibilizados pelo OpenCV [6].

Este artigo está organizado da seguinte forma: na Secção 2 é apresentado o trabalho relacionado. Na Secção 3 é apresentado o procedimento idealizado para ajudar o utilizador a escolher a melhor foto, nomeadamente o *workflow* que o compõe bem como os desafios levantados por cada uma das suas fases. Na Secção 4 é apresentada a metodologia utilizada para implementação do *workflow*. Na Secção 5 é apresentada uma avaliação sobre os métodos já implementados, sendo assim possível analisar o seu comportamento e antever o impacto da sua utilização pelo utilizador final. Por fim, na Secção 6 é comentado o trabalho realizado e exposto o trabalho futuro.

2 Trabalho Relacionado

Atualmente existem várias técnicas propostas para a análise de imagens segundo diferentes parâmetros, nomeadamente deteção de *focus* [12,9], deteção de *motion blur* [10,16] e deteção de ruído [8,14]. No entanto, segundo as pesquisas realizadas, continua a haver uma escassez de trabalhos que combinem estes métodos com o objetivo de selecionar a melhor foto de um conjunto de imagens semelhantes. Esta secção apresenta e compara dois trabalhos cujo o âmbito está relacionado com o trabalho proposto neste artigo.

2.1 Automatic Photo Selection for Media and Entertainment Applications

Potapova et. al. [15] propuseram um método para a seleção das melhores fotos de um álbum. Numa primeira fase são identificadas fotos de baixa qualidade que não são tidas em conta nos processamentos posteriores. Fotos afetadas por artefactos de compressão, que segundo os autores se assemelham a ruído, são detetadas com recurso a um filtro de *deblocking* e *deringing* [4]. Para ajuste dos parâmetros do filtro é realizada uma análise sobre a tabela de quantização q , de tamanho $3 * 3$, do canal de brilho: $k = 1/9 \sum_{i,j=1}^3 q_{i,j}$.

Segundo as experiências dos autores, caso k tenha um valor superior a 6,5 então a fotografia é fortemente afetada por artefactos de compressão. Imagens de baixo contraste, particularmente afetadas por *backlighting*, são detetadas com recurso ao histograma de brilho da imagem, tendo em conta informação como os valores de tons em sombras, tons médios e o tom máximo do histograma. Estes valores servem como parâmetros para o algoritmo de aprendizagem automática AdaBoost [5] de forma a distinguir imagens de baixo contraste. Para a identificação de imagens desfocadas, cada imagem é convolvida várias vezes com filtros *high-pass* de vários tamanhos sendo que a entropia da variação dos histogramas nas arestas caracteriza o achatamento ou o pico do histograma. Para além disso

é também analisada a diferença entre a versão original da imagem e uma versão suavizada. Caso a diferença seja não seja significativa então a imagem original possivelmente está desfocada.

Após a remoção das fotografias de baixa qualidade, o próximo passo é agrupar as restantes. Para tal é tida em conta a câmara com a qual a foto foi tirada e o momento em que foi tirada. Esta informação é obtida a partir da informação EXIF [7]. Posto isto, os autores assumem que a foto mais apelativa e relevante de cada grupo é a mais saliente. Desta forma, para cada foto é construído um mapa de saliência com base nos mapas de intensidade, orientação e cor, sendo atribuído a cada um destes mapas um peso específico.

Os autores afirmam obter resultados bastante positivos com este método. No entanto não nos foi possível testá-lo visto que não encontramos nenhum produto de *software* com a sua implementação. O método de deteção de fotos desfocadas poderá vir a ser interessante no contexto deste trabalho, possivelmente como um pré-processamento para a deteção de *motion blur*. Pelo contrário, uma fotografia tirada a uma cena cujos objetos tenham cores semelhantes será uma fotografia de baixo contraste, o que não implica necessariamente que seja indesejável. Desta forma, no contexto deste trabalho esse método não será viável. Em relação à deteção de artefactos de compressão, fica em aberto uma análise mais profunda sobre este método. No entanto a nossa primeira opção passará pela tentativa de implementação de um método que permita ter uma estimativa do nível de ruído na imagem.

2.2 Tiling Slideshow

Chu et. al. [2] propuseram um método para gerar *slideshows* audiovisuais em que várias fotos com características semelhantes são exibidas na mesma *frame*. Embora este método não tenha o propósito final de seleccionar a melhor foto, torna-se interessante referenciá-lo visto que um dos procedimentos consta em remover fotos de baixa qualidade. Os autores consideram fotos desfocadas, sub-expostas e sobre-expostas como tendo baixa qualidade. Para a deteção de fotos de baixa qualidade, foi adotado um método baseado em *wavelets* que analisa as características das arestas em diferentes resoluções. Fotos sub-expostas e sobre-expostas são detetadas através do cálculo do número de *pixels* escuros e claros em cada foto. Caso estes valores superem um determinado *threshold* as fotos serão consideradas sub-expostas (*pixels* escuros) ou sobre-expostas (*pixels* claros). Mais uma vez, o facto de uma foto ter baixo contraste não implica necessariamente que tenha baixa qualidade pelo que seguiremos uma abordagem diferente para a deteção de fotografias com um nível de exposição inadequado.

3 Procedimento para escolha da melhor fotografia

O procedimento proposto neste artigo para a escolha da melhor foto tem como base um *workflow*, pelo qual o utilizador é guiado. O utilizador deverá incorporar o resultado de cada fase do *workflow* no seu processo de eliminação seletiva das

fotografias (semelhantes) que mais se afastam dos seus objetivos e expectativas, chegando no final a um conjunto reduzido de fotografias.

3.1 O Workflow

O *workflow* referente a este procedimento é composto por cinco fases, podendo ser analisado na Figura 1.

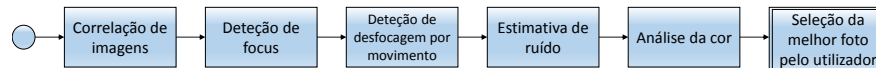


Figura 1: *Workflow* proposto.

A ordem definida para as fases do *workflow* reflete a relevância dos problemas por elas endereçados. Por exemplo, uma má focagem é um fator claramente eliminatório visto que é irreversível, pelo que estas fases se encontram logo no início do *workflow* (apenas precedido pela correlação das imagens que é um pré-requisito para o processamento das mesmas). É possível atenuar ou mesmo eliminar o efeito do ruído nas fotografias, às custas de alguma redução na qualidade da imagem, razão pela qual se considerou a análise do ruído como o terceiro problema mais relevante. Existem técnicas com impacto muito reduzido na qualidade da imagem que permitem alterar o nível de exposição e a temperatura de uma imagem, pelo que este é o último parâmetro a ser analisado no *workflow* proposto. Por fim, cabe ao utilizador escolher a melhor fotografia do conjunto restante.

Correlação de imagens: Com o objetivo de fazer uma análise comparativa entre as imagens do conjunto é necessário correlacionar as mesmas. O resultado deste procedimento permite identificar automaticamente pontos, ou áreas, equivalentes nas duas imagens. Esta fase não requer *input* do utilizador, tendo como propósito possibilitar uma análise comparativa entre imagens nas fases posteriores. Sendo natural que as imagens do conjunto variem em fatores como a profundidade de campo, ruído, distância focal ou orientação, o principal desafio desta fase consta em encontrar-se um método que seja invariante aos fatores mencionados, resultando assim numa correlação de imagens com um grau de precisão elevado. Visto que o conjunto de imagens a processar pode ser vasto, o processo de correlação tem de apresentar reduzidos tempos de processamento. Abordagens possíveis incluem correlacionar todas as imagens entre si, ou utilizar uma imagem de referência e correlacionar as restantes apenas com esta.

Análise das regiões focadas: Nesta fase, o objetivo é apresentar ao utilizador uma máscara sobre as regiões focadas de cada imagem, podendo de seguida o

utilizador eliminar as fotos que não apresentam os motivos de interesse devidamente focados. O principal desafio desta fase prende-se com a apresentação intuitiva, eficiente e precisa da máscara sobre as regiões focadas de cada imagem.

Deteção de desfocagem por movimento: Nesta fase é também apresentada uma máscara sobre as zonas das fotografias que têm uma desfocagem por movimento. No caso de a desfocagem ser resultado de movimento da câmara então a máscara cobrirá toda a superfície da fotografia. Caberá depois ao utilizador decidir se pretende fotografias com desfocagem por movimento ou não, eliminando as que não são pretendidas. A distinção entre desfocagem causada pelo motivo estar fora da profundidade de campo e desfocagem por movimento na cena será o principal desafio a ter em conta nesta fase.

Estimativa de ruído: Nesta fase será apresentado ao utilizador uma estimativa do nível de ruído existente em cada fotografia, podendo o utilizador eliminar as fotografias com um nível de ruído diferente do pretendido. O ruído pode ser visto como um conjunto de pontos alienados, cuja intensidade e cor não estão relacionados com os restantes *pixels*, estando espalhados ao longo da superfície da imagem. A deteção destes pontos ruidosos e consequente estimativa do nível de ruído existente em cada imagem define o principal desafio desta fase.

Análise da cor: Por último, esta fase servirá para distinguir e apresentar ao utilizador o nível de exposição de cada fotografia bem como a temperatura que esta contém. Uma análise correta dos histogramas de tons da cor da imagem poderá servir como ponto de partida para implementar com sucesso esta fase.

Seleção da melhor foto: Após a conclusão dos procedimentos que envolvem uma análise computacional, espera-se que a amostra final de fotografias seja bastante reduzida quando em comparação com a amostra inicial. Das fotografias restantes (no caso de haver mais que uma), cabe ao utilizador escolher a melhor com base no enquadramento e em parâmetros de composição.

4 Desenho do sistema

Tendo descrito cada fase do *workflow* proposto, esta secção tem como propósito apresentar a forma como se procedeu para resolver os desafios identificados na secção anterior. Até ao momento de escrita deste artigo foram implementados os procedimentos de correlação de imagens e de deteção de *focus*. Desta forma, o trabalho descrito incide maioritariamente sobre estes dois procedimentos. Para os restantes procedimentos é apresentado de forma sucinta algum *background* e algumas ideias de como abordar essa fase.

4.1 Correlação de imagens

Uma visão global da implementação seguida para este procedimento pode ser vista na Figura 2.

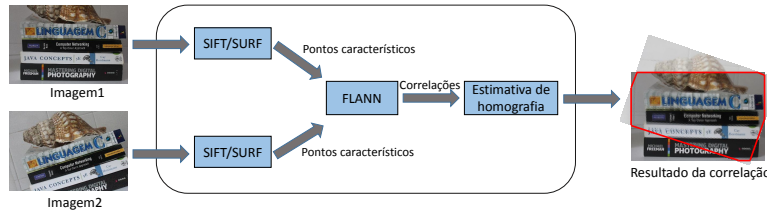


Figura 2: Visão global do procedimento de correlação de imagens.

Os algoritmos SURF [1] e SIFT [11] têm o propósito de detetar pontos característicos, ou seja, pontos distinguíveis em imagens. Ambos têm procedimentos semelhantes: numa primeira fase são detetados pontos característicos em cada imagem. De seguida, é calculado um descritor para cada um destes pontos. Este descritor contém principalmente a orientação e a escala em que o ponto foi detetado. Desta forma estes algoritmos tendem a ser invariantes a mudanças de escala e orientação mas apenas parcialmente invariantes a mudanças de iluminação.

Visto que as imagens podem variar significativamente em termos de iluminação, o nosso primeiro passo constou em normalizar as imagens de forma a torná-las o mais semelhantes possível. Para tal foi utilizada a operação de *Histogram equalization* disponível no OpenCV [6]. A Figura 3 demonstra o resultado da aplicação desta operação sobre uma imagem sub-exposta, sendo que o número de pontos característicos detetados é claramente superior após a operação de normalização. Na mesma figura é também possível ver que o número de pontos característicos detetados, após a operação de normalização, se aproxima do número de pontos característicos detetados numa imagem com a exposição ideal (referência).

Tendo sido calculados os descritores de cada ponto, o próximo passo será emparelhar pontos de interesse respetivos entre imagens diferentes. A biblioteca FLANN [13] contém um conjunto de algoritmos de *neighbour searches*, sendo capaz de escolher o algoritmo mais apropriado consoante o tipo de dados que recebe como entrada. Desta forma é possível encontrar o descritor, numa imagem, que é mais próximo daquele que o FLANN recebe como *input*. Para tal foram testados os métodos `match` e `knnMatch`. O método `match` devolve o descritor do ponto de uma imagem que é mais próximo ao ponto da imagem de referência. O método `knnMatch` devolve os k descritores mais próximos. O principal desafio desta fase consta em minimizar o número de falsos positivos, isto é, correlações de pontos não respetivos. Usando o método `match`, definiu-se um *threshold* global. Para todas as correlações entre pontos foi calculada aquela que tem a menor

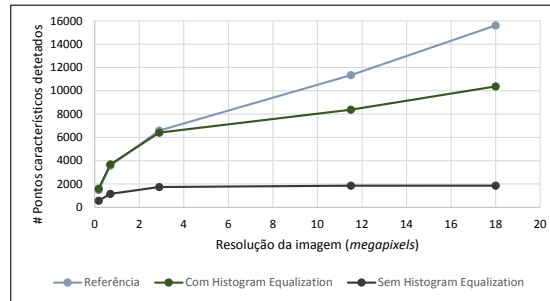


Figura 3: Impacto do *Histogram Equalization* na detecção de pontos característicos.

distância (min_dist). De seguida, para cada uma das restantes distâncias ($dist_i$), caso esse valor esteja abaixo de um *threshold* t ($dist_i < 2 * min_dist$) a correlação é considerada como positiva, caso contrário é negativa. No entanto, tal como apontado por Lowe [11] a definição de um *threshold* global pode não ser a opção mais eficaz visto que alguns descritores são mais discriminatórios que outros. Desta forma Lowe [11] propôs um *ratio test* em que caso o rácio da distância entre o vizinho mais próximo ($v1$) e o segundo vizinho mais próximo ($v2$) seja superior a $1/1,5$ ($v1/v2 > 1,5$) então a correlação é considerada como positiva. Caso contrário é considerada negativa. Este teste permite acima de tudo a não eliminação de *matches* corretos como se pode comprovar pela Figura 4.

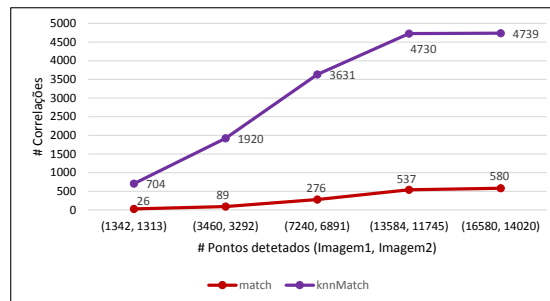


Figura 4: Comparação entre `match` e `knnMatch` no processo de correlação.

De forma a tornar este processo ainda mais eficaz e eliminar eventuais falsos positivos (marcados a vermelho na Figura 5) foi também calculada a matriz de homografia entre as imagens usando o método *random sample consensus* (RANSAC) [3]. O resultado final pode ser visto nas Figuras 6a e 6b.

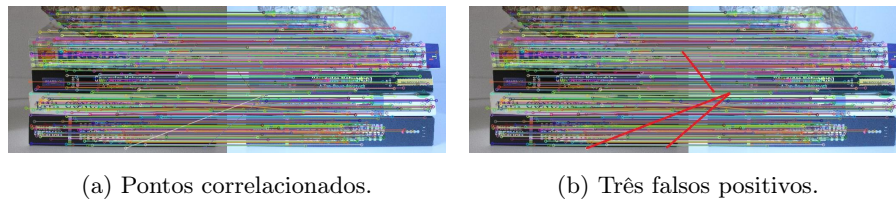


Figura 5: Resultado da aplicação do FLANN.



Figura 6: Resultado final da correlação de imagens após cálculo da matriz de homografia.

De forma a correlacionar todo o conjunto de imagens foi decidido que haverá uma imagem de referência, sobre a qual o utilizador trabalhará, sendo que apenas esta imagem será correlacionada com as restantes. Esta abordagem oferece vantagens acima de tudo em termos de eficiência visto que como exemplo, um caso ilustrativo de dez imagens, implica nove correlações. Já correlacionando todas as imagens entre si implicaria 45 correlações.

4.2 Detecção de *focus*

De forma a analisar as imagens em termos de focagem foram implementadas duas abordagens: a primeira define-se como uma análise global da focagem enquanto que a segunda é uma análise sobre uma região de interesse definida pelo utilizador.

O gradiente da imagem pode ser uma característica importante nesta análise visto que a intensidade da cor de uma imagem focada tende a ser maior do que a de uma imagem desfocada. Desta forma, a metodologia seguida para analisar as imagens em termos de focagem envolve o cálculo do gradiente, nomeadamente usando o filtro Sobel disponível no OpenCV [6].

A **análise global da focagem** tem como objetivo apresentar uma forma rápida e intuitiva de detectar e distinguir regiões focadas nas várias imagens. O algoritmo Sobel é aplicado sobre a versão em tons de cinzento de cada imagem, sendo que as regiões de notória variação de intensidade (ex.: arestas) são detetadas no *output*.

¹ Figuras 7a e 7b tiradas de: <http://teachers.sduhsd.net/delliott/Files-Photo/deep%20vs%20shallow%20DOFchess%20pieces.JPG>

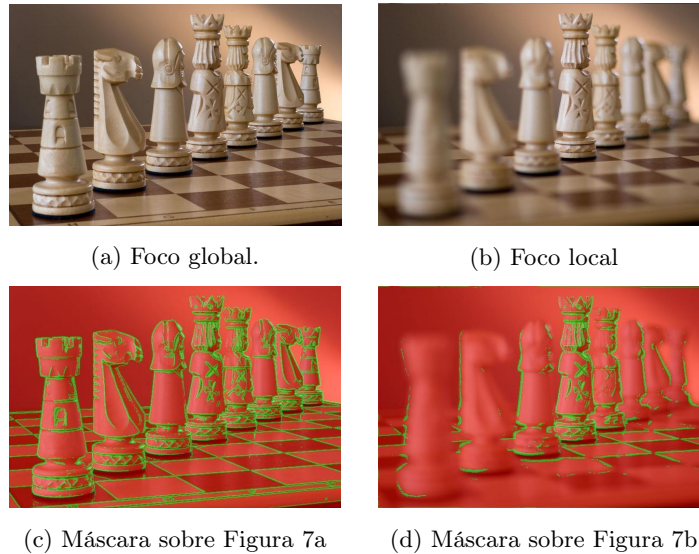


Figura 7: *Pixels* verdes têm uma intensidade superior ao valor do *slider*¹.

Os valores resultantes do algoritmo Sobel são depois normalizados entre 0 e 255 de forma a ser possível oferecer ao utilizador um *slider* que representa o valor da intensidade. Aos *pixels* cujo valor da intensidade é maior que o valor do *slider* é sobreposta uma máscara verde, aos restantes é sobreposta uma máscara vermelha. O efeito pode ser verificado na Figura 7.

Podem ocorrer que as imagens do conjunto sejam bastante semelhantes em termos de focagem, o que torna a abordagem anterior ineficaz. Desta forma é também possível uma **análise sobre uma região de interesse** definida pelo utilizador na imagem de referência. As respetivas regiões nas restantes imagens são calculadas automaticamente via homografia. A cada região de interesse é aplicado o algoritmo Sobel, sendo de seguida calculada a média do valor da intensidade nessa região. A região cujo valor médio de intensidade for mais elevado é tida como sendo a mais focada.

4.3 Detecção de desfocagem por movimento

O gradiente da imagem define mudanças direcionais em termos de intensidade ou cor. Este pode ser visto como um vetor 2D que aponta na direção de maior variação, representando a sua magnitude a taxa dessa variação.

O método proposto em [10] visa distinguir desfocagem causada pelo motivo estar fora da profundidade de campo (*out-of-focus*) e desfocagem causada por movimento na cena (*motion blur*). A metodologia seguida consta em criar uma versão suavizada da imagem, dividida em várias regiões, tendo em conta que:

1. Se uma imagem contiver *motion blur*, as arestas com gradiente perpendicular à direção de movimento não serão suavizadas, tendo o gradiente maior magnitude apenas ao longo de uma direção.
2. Se uma região da imagem estiver desfocada por *out-of-focus*, a magnitude do gradiente é atenuada em todas as direções.

Com base nestas afirmações é construído um histograma direcional para todas as regiões da imagem. Cada entrada do histograma representa uma direção específica sendo o seu valor o número de *pixels* cujo gradiente tem essa direção. Desta forma, quando em face de *motion blur*, o histograma apresenta um pico distintivo na direção de movimento.

O método proposto em [16] é baseado num modelo de duas camadas em que cada pixel I é visto como uma combinação linear de cores de *foreground*, F , e cores de *background*, B , sendo α o peso de cada componente. Este modelo é definido da seguinte forma: $I = \alpha F + (1 - \alpha)B$, onde α pode ter qualquer valor no intervalo $[0, 1]$. Em imagens desfocadas, as cores de *foreground* e as cores de *background* tendem a misturar-se, nomeadamente nas arestas, tomando α valores decimais. Desta forma, é realizada uma análise sobre o modelo do canal α , definido por $\nabla\alpha \cdot b \in \{-1, 1\}$, onde b é um vetor que define a medida de desfocagem na direção vertical e horizontal. Para regiões afetadas por *motion blur*, b tende a ser direcional, logo $\nabla\alpha$ é representado por linhas, enquanto que em situações de *out-of-focus*, $\nabla\alpha$ é espalhado por todas as direções, com um efeito radial.

4.4 Detecção de ruído

O método *filtered-based* [14] tem como ideia base obter-se uma versão suavizada da imagem ruidosa e de seguida subtrair essa versão suavizada à versão original. Isto porque os filtros de suavização tendem a atenuar ou eliminar o ruído existente sendo que a diferença entre versões irá fazer sobressair o ruído.

No método *block-based* [8] a imagem é dividida em vários blocos sendo que o desvio padrão de intensidade do bloco mais suave é considerado como sendo o valor do ruído existente. Nesta abordagem a obtenção de um bloco totalmente suave na imagem poderá ser realizada automaticamente ou com recurso a *input* do utilizador. No caso ilustrativo da Figura 8 encontram-se três regiões homogêneas com diferentes níveis de ruído.

O desvio padrão de intensidade da cor nas Figuras 8a, 8b e 8c é 14.8, 37.8 e 60.3, pelo que estes valores podem servir como um indicador do nível de ruído na imagem.

4.5 Análise da cor

A luminosidade da cor numa imagem pode definir o seu nível de exposição. Uma imagem sub-exposta é dominada por tons escuros, uma imagem sobre-exposta é dominada por tons claros enquanto que uma imagem com um nível de exposição balanceado contém os seus tons de cor espalhados por toda a escala.

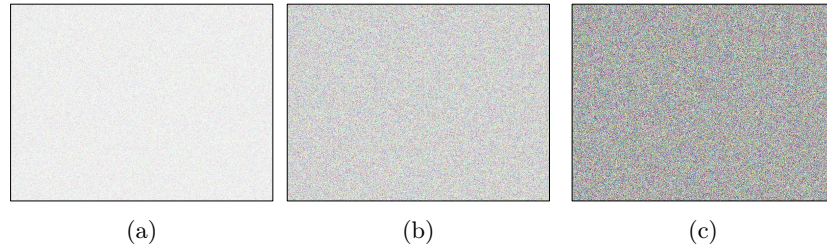


Figura 8: Diferentes níveis de ruído.

Três imagens com diferentes níveis de exposição e os seus respetivos histogramas com a distribuição dos tons de cinzento podem ser vistos na Figura 9.

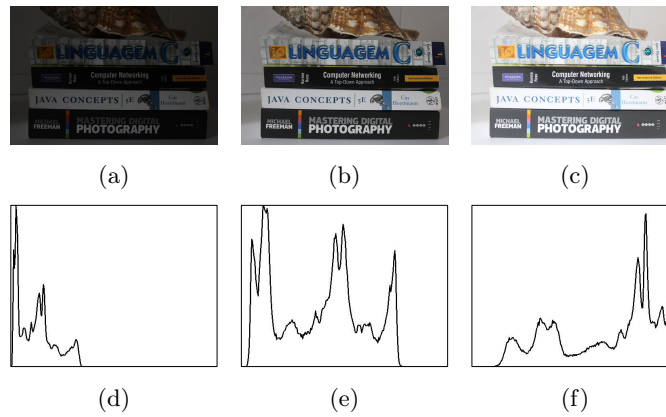


Figura 9: Diferentes níveis de exposição e respetivos histogramas.

As funções que definem os histogramas podem ser relacionadas entre elas. Uma análise comparativa entre histogramas permitirá, em princípio, ordenar as imagens em termos de nível de exposição.

5 Avaliação

Nesta secção são apresentados o protótipo e o desempenho das duas primeiras fases do *workflow* proposto. Todas as experiências foram realizadas num PC Intel(R) Core(TM) i7-4500U CPU @ 1.80GHz 2.40GHz com 8 GB RAM a correr sobre o sistema operativo Windows 8.1 com 64 bits. As imagens utilizadas para realizar as experiências encontram-se disponíveis em <https://github.com/apalves/Sets-of-similar-photos>.

5.1 Protótipo

O desenvolvimento da ferramenta de *software* proposta tem tido como base um conjunto de algoritmos disponibilizados pelo OpenCV [6] usando a linguagem de programação C++. A unidade *highgui* do OpenCV tem sido utilizada como interface gráfica sendo que pela altura da escrita deste artigo a interface gráfica do utilizador ainda não se encontrava implementada.

5.2 Impacto de `contrastThreshold` no tempo de processamento e na detecção de pontos

O método SIFT é definido da seguinte forma:

```
SIFT::SIFT(int nfeatures=0, int nOctaveLayers=3, double  
contrastThreshold=0.04, double edgeThreshold=10, double sigma=1.6)
```

O parâmetro `contrastThreshold` é usado para filtrar pontos dúbios em regiões de baixo contraste. Desta forma, tendo em conta que as imagens podem variar significativamente em termos de iluminação, foram corridos vários testes variando o valor deste parâmetro. O impacto desta variação na detecção de pontos característicos e no tempo de processamento pode ser visto na Figura 10a. Os testes foram realizados com uma resolução de aproximadamente 0,7 *megapixels*.

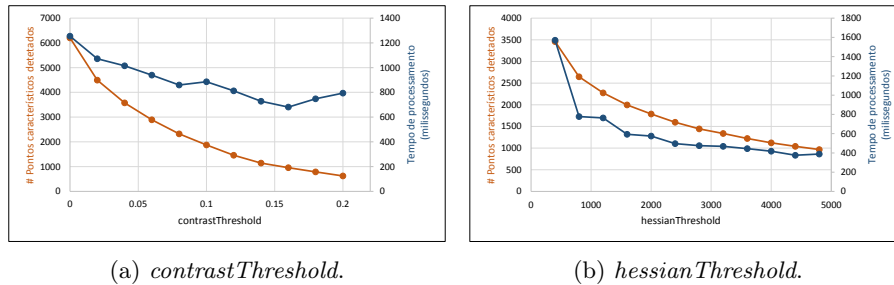


Figura 10: Impacto de `contrastThreshold` e `hessianThreshold` na detecção de pontos característicos e no tempo de processamento.

Naturalmente, há medida que o valor do `contrastThreshold` aumenta menos pontos são detetados sendo que o tempo de processamento também diminui variando entre os 750 e aproximadamente 1200 milissegundos.

5.3 Impacto de `hessianThreshold` no tempo de processamento e na detecção de pontos

O método SURF é definido da seguinte forma:

```
SURF::SURF(double hessianThreshold, int nOctaves=4, int nOctaveLayers=2,  
            bool extended=true, bool upright=false )
```

Seguindo a mesma lógica de raciocínio foram realizados vários testes variando o valor do parâmetro `hessianThreshold`. O impacto desta variação pode ser visto na Figura 10b.

Comparando SIFT e SURF tem-se que o algoritmo SURF é mais rápido sendo que ambos os algoritmos apresentam um número significativamente elevado de pontos característicos detetados.

5.4 Impacto da resolução da imagem no tempo de processamento e na deteção de pontos

O impacto da resolução da imagem na deteção de pontos característicos e no tempo de processamento pode ser visto na Figura 11.

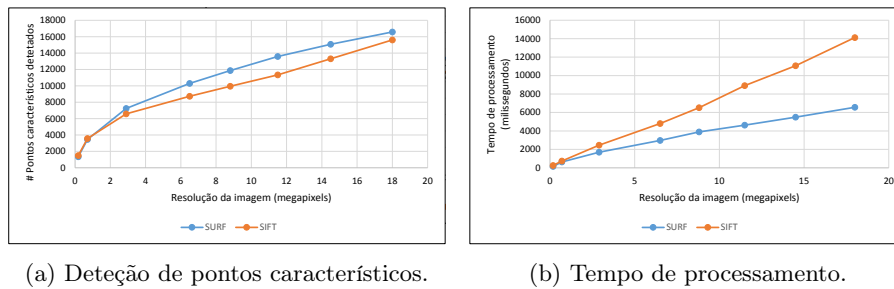


Figura 11: Impacto da resolução da imagem.

O número de pontos característicos detectados tende a ter a mesma ordem de grandeza usando o SIFT ou o SURF nas várias resoluções. A principal razão do SIFT apresentar tempos de processamento mais lentos prende-se com o facto de este processar as imagens em várias escalas de forma a que o resultado se torne invariante a mudanças de escala. Já o algoritmo SURF mantém sempre a mesma resolução da imagem, variando apenas o tamanho do *box filter* que é aplicado à imagem integral. Tendo em conta que a ferramenta proposta é interativa, torna-se importante ter tempos de processamento baixos com o intuito de melhorar a experiência do utilizador. Desta forma, a nossa opção passará, ao que tudo indica, pela utilização do algoritmo SURF.

5.5 Impacto das características da imagem no cálculo do gradiente

O impacto das características da imagem (ruído, nível de exposição, temperatura) no cálculo do gradiente pode ser visto na Figura 12a realizado sobre uma imagem de 18 *megapixels*. O número de pontos verdes (eixo vertical) representa

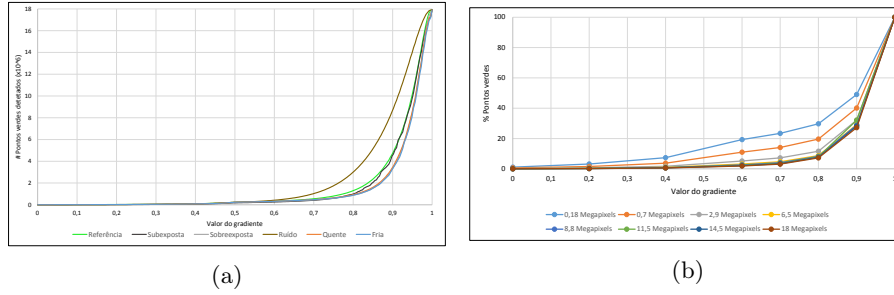


Figura 12: Impacto das características imagem (a) e da resolução (b) no cálculo do gradiente.

o número de *pixels* cujo valor da intensidade é maior que o valor do gradiente, neste gráfico normalizado para valores entre zero e um.

Os resultados são bastante semelhantes excepto quando a imagem tem um nível de ruído considerável, pelo que se conclui que o ruído pode ter uma influência preponderante no cálculo do gradiente e desta forma deve ser estimado de forma a evitar cálculos errados.

5.6 Impacto da resolução da imagem no cálculo do gradiente

O impacto da resolução da imagem no cálculo do gradiente pode ser na Figura 12b. A resolução original da imagem em teste é de 18 *megapixels*, tendo sido reduzida várias vezes de forma a ser possível realizar este teste.

Analisando o gráfico pode-se concluir que em resoluções menores uma maior percentagem de pontos verdes é detetada, isto porque a operação de diminuição da resolução da imagem tende a fazer operações de *sharpening*.

6 Conclusões

Com a utilização da ferramenta de *software* proposta neste artigo espera-se que os resultados da automatização dos fluxos de trabalho no processo de seleção da melhor foto permitam ao utilizador ultrapassar de uma forma simples, precisa e eficaz os desafios apresentados, tornando-se assim a escolha da melhor foto numa experiência gratificante e motivadora.

Em relação às experiências realizadas pretende-se melhorar a parte da correlação de imagens visto que, tal como é apresentado na Figura 11b, este processo tem tempos excessivamente elevados quando em face de imagens de alta resolução. Na deteção de *focus* há também que ter em conta o nível de ruído existente de forma a melhorar a precisão deste procedimento.

O trabalho futuro passa pela implementação das restantes fases do *work-flow* proposto seguido da construção da interface gráfica a ser apresentada ao utilizador. Como resultado final surgirá um produto de *software* que pode ser

standalone ou um *plugin* para um produto de gestão/manipulação de imagens como o Photoshop. Numa fase posterior pretende-se ainda aplicar técnicas de aprendizagem automática que permitam conhecer as preferências do utilizador de forma a simplificar ainda mais o processo de seleção da melhor fotografia.

Agradecimentos. Este trabalho foi parcialmente suportado pela Fundação para a Ciência e Tecnologia através do projeto estratégico PEst-UID/CEC/04516/2013.

Referências

1. H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3):346–359, June 2008.
2. W.-T. Chu, J.-C. Chen, and J.-L. Wu. Tiling slideshow: An audiovisual presentation method for consumer photos. *MultiMedia, IEEE*, 14(3):36–45, July 2007.
3. M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.
4. A. Foi, V. Katkovnik, and K. Egiazarian. Pointwise shape-adaptive DCT for high-quality deblocking of compressed color images. *European Signal Processing Conference*, 16(5):1–17, 2006.
5. Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting, 1995.
6. Itseez. OpenCV. <http://opencv.org/>, 2015. [Online; accessed 15-June-2015].
7. JEITA. *Exchangeable image file format for digital still cameras*. 2002.
8. J. Lee and K. Hoppel. Noise modeling and estimation of remotely-sensed images. In *Geoscience and Remote Sensing Symposium, 1989. IGARSS'89. 12th Canadian Symposium on Remote Sensing., 1989 International*, volume 2, pages 1005–1008, 1989.
9. S. Li and B. Yang. Multifocus image fusion using region segmentation and spatial frequency. *Image Vision Comput.*, 26(7):971–979, July 2008.
10. R. Liu, Z. Li, and J. Jia. Image partial blur detection and classification. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA*, 2008.
11. D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, Nov. 2004.
12. Y. Lu, X. Feng, J. Zhang, R. Wang, K. Zheng, and J. Kong. A multi-focus image fusion based on wavelet and region detection. pages 294–298, 2007.
13. M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Application VISSAPP'09*, pages 331–340. INSTICC Press, 2009.
14. T.-A. Nguyen and M.-C. Hong. Filtering-based noise estimation for denoising the image degraded by gaussian noise. In *Proceedings of the 5th Pacific Rim Conference on Advances in Image and Video Technology - Volume Part II, PSIVT'11*, pages 157–167, Berlin, Heidelberg, 2012. Springer-Verlag.
15. E. Potapova, M. Egorova, and I. Safonov. Automatic photo selection for media and entertainment applications. *GRAPHICON-2009*, pages 117–124, 2009.
16. B. Su, S. Lu, and C. L. Tan. Blurred image region detection and classification. In *Proceedings of the 19th ACM International Conference on Multimedia, MM '11*, pages 1397–1400, New York, NY, USA, 2011. ACM.