Determinação de vértices para correção automática de perspectiva

Fernando Reiszel Pereira¹ Denise Del Re Filippo¹

Resumo: O uso de projetores e quadros interativos está bastante difundido em todas as esferas de ensino, do básico ao superior. Observa-se que a ampla utilização destes recursos na educação, aliada ao uso fortemente disseminado de dispositivos móveis cada vez mais poderosos, vem induzindo no meio discente o registro de aulas e exposições por meio da captura das imagens projetadas em aula. A postura clássica de copiar a aula expositiva em um caderno vem sendo rapidamente substituída pelo registro fotográfico das aulas por dispositivos móveis e notebooks. Sem entrar no mérito dos aspectos didáticos deste procedimento, é fato que este tipo de registro está estabelecido no meio educacional. Por meio de entrevistas com alunos, verificou-se que havia um desejo por parte dos mesmos (principalmente por parte daqueles afastados do centro do quadro) de fotografar os quadros e telas apresentados em aula sem a distorção por perspectiva característica deste tipo de registro. Há vários programas comerciais que corrigem esta distorção, porém a maioria requer um ajuste manual para delimitar a região da imagem que será corrigida. Neste trabalho apresenta-se um novo método para determinar automaticamente a região de interesse da imagem de um quadro a ser corrigida que, aliado a uma sequência de procedimentos e algoritmos de processamento de imagens, possibilita a automação da correção de maneira mais eficiente e precisa. A realização destes algoritmos foi inteiramente realizada na linguagem Processing, o que possibilita seu uso gratuito em diversos sistemas operacionais existentes em dispositivos móveis.

Palavras-chave: Homografia. Perspectiva. Processamento de Imagem.

Abstract: The widespread use of projectors and interactive whiteboards is remarkable in all educational levels, from basic to higher. It is observed that the widespread use of these resources in education, allied with the increasingly use of powerful mobile devices, is leading the students to register lectures and exhibitions by capturing the images projected in class. The classical approach of copying the lecture in a notebook is being rapidly replaced by the photographic record of the classes with mobile devices and notebooks. Without delving into the merits of the didactic aspects of this procedure, it is fact that this type of recording is already established in the educational environment. After interviews with students, it was found that they (especially those not located in front of the projection) desired to shoot frames and screens presented in class without the distortion of perspective characteristic of this type of record. There are several commercial programs that correct this distortion, but most of them require a manual adjustment to delimit the region of the picture to be corrected. In this work we present a new method that automatically determines the region of interest of a whiteboard image to be corrected. This method, allied with a sequence of procedures and image processing algorithms allows a more efficient and precise automatic correction of images. The realization of these algorithms was fully realized in the Processing language, which allows its free use in many existing operating systems on mobile devices.

Keywords: Image Processing. Homography. Perspective.

1 Introdução

O uso disseminado de dispositivos móveis e notebooks tornou o registro fotográfico de apresentações e palestras um procedimento habitual por estudantes dos mais variados graus acadêmicos, que definem quais são as imagens e momentos de uma aula que são de seu interesse. Em uma sala de aula convencional é de se esperar

http://dx.doi.org/10.5335/rbca.v9i2.6105

¹ Escola Superior de Desenho Industrial, ESDI,UERJ – Rio de Janeiro(RJ)- Brasil {freiszel@gmail.com, denise.filippo@gmail.com}

que a captura das imagens apresentadas sofra distorções de perspectiva, notadamente quando o ouvinte está situado nos extremos laterais da sala. Após anos de experiência com alunos da Escola Superior de Desenho Industrial — ESDI observou-se que distorção é tolerável para alguns, porém, no caso de palestras e aulas que tenham diagramas geométricos, desenhos técnicos e imagens gráficas cujo entendimento possa ser prejudicado pela distorção de perspectiva, observou-se que alguns estudantes manifestam a necessidade de ter um registro de imagem sem as distorções.

A Figura 1 mostra dois exemplos de imagens distorcidas, capturadas por uma câmera digital de baixa resolução (640 x 480) durante aulas e suas respectivas correções, executadas automaticamente com a metodologia proposta neste trabalho. A realização dos métodos de tratamento de imagens em plataforma aberta (Processing) e a simplicidade do algoritmo proposto neste trabalho permitem seu uso em diversos ambientes (GNU/Linux, Mac OS X, Windows, Android) e dispositivos móveis (notebooks e smartphones).

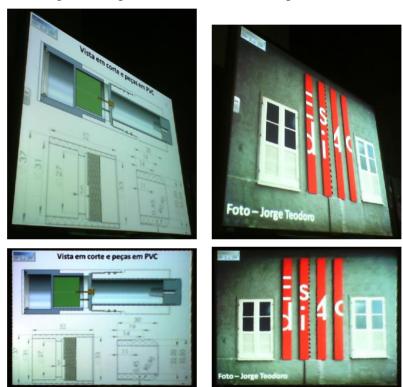


Figura 1: Imagens distorcidas (acima) e corrigidas (abaixo)

A correção de perspectiva vem sendo amplamente utilizada em imagens de aerofotogrametria, como pode ser visto em [1]. Neste trabalho os autores usam características geométricas das imagens (linhas retas e curvas conhecidas, entre outros) para correção das distorções impostas por diferentes posições da câmara usada na captura de imagens. [2]-[4] também descrevem técnicas de correção de distorções para imagens aerofotogramétricas, onde a correção é feita usando-se a imagem de uma mesma cena capturada em diversas vistas, ou, no mínimo, em duas vistas, como demonstrado no algoritmo de correção de perspectiva usando imagens estereoscópicas descrito em [5]. Indo na direção de estruturas de menor escala, a correção de perspectiva é aplicada em processos de digitalização de documentos, como descrito por [6], onde é demonstrado que a correção das distorções acarreta em uma maior capacidade de reconhecimento de caracteres em um texto digitalizado.

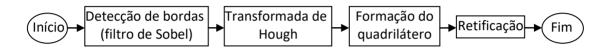
Há uma grande variedade de programas e aplicativos capazes de corrigir as distorções de perspectiva em imagens, dentre os quais se podem citar como exemplo mais representativo o Adobe Photoshop®, cujos recursos pertinentes ao tema são descritos em [7]. Ainda que este e demais programas consigam corrigir as distorções por

perspectiva de maneira plenamente satisfatória, a correção não se dá em tempo real, além de requerer uma interface gráfica com o usuário para que a correção seja feita.

Uma proposta de automatização de correção de imagens fotográficas de quadros brancos usados em salas de aula é apresentada por [8], onde são descritos vários algoritmos de processamento de imagens usados para detectar automaticamente o quadrilátero formado pelo quadro branco. O presente trabalho descreve os procedimentos para automatizar a detecção e correção de perspectiva de imagens projetadas em telas planas e apresenta alterações e novas metodologias de detecção em relação ao publicado em [8].

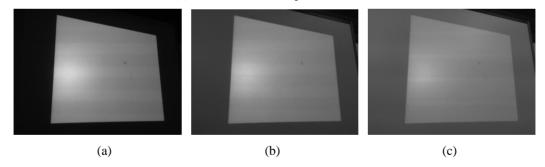
A detecção dos limites formados pelo quadrilátero de um quadro branco ou projeção é o principal obstáculo para a realização de um sistema automatizado de correção de distorções de perspectiva. Os procedimentos descritos em [8] são mostrados no fluxograma da Figura 2, onde a transformada de Hough [9] é usada para a detecção dos limites do quadro ou projeção.

Figura 2: Metodologia de detecção usada por [8].



A detecção do contorno de uma projeção ou de um quadro branco está sujeita às condições de luminosidade do ambiente que podem impor limites na detecção automática do contorno. Estas limitações levaram [8] a apresentar uma alternativa à detecção automática, na qual o usuário estabelecia manualmente o limite do quadro ou tela por meio de uma interface gráfica. Um algoritmo de detecção de contorno deve apresentar boa imunidade a variações de iluminação do ambiente. A Figura 3 mostra a captura de imagem de uma projeção de um retângulo branco sob três condições de iluminação reais encontradas em uma sala de aula: a) no escuro, b) com luzes acesas c) com luzes acesas e janelas abertas.

Figura 3: Projeção de retângulo branco sob três condições reais de iluminação. (a) no escuro, (b) com luzes acesas, (c) com luzes acesas e janelas abertas.



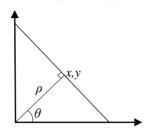
Os algoritmos usados neste trabalho, descritos na seção "Metodologia" possibilitaram a detecção automática do quadrilátero projetado mesmo sob condições de iluminação adversas (figura 3-c).

Em diversos trabalhos sobre reconhecimento de padrões a estimativa das equações das linhas que formam os lados do quadrilátero detectado é feita pela transformada de Hough [9], um método bem conhecido para a detecção de linhas. Ainda que a transformada de Hough tenha sido apresentada pela primeira vez há mais de cinquenta anos, vários trabalhos recentes compilados por [10] mostram a sua versatilidade em várias aplicações que requeiram detecção de linhas em imagens digitalizadas. Na transformada de Hough um ponto (x,y) pode pertencer a um conjunto infinito de linhas parametrizadas por (1).

$$\rho(\theta) = x \cos \theta + y \sin \theta \tag{1}$$

onde ρ é a distância da origem ao ponto mais próximo da linha(expressa em quantidade de pixels) e θ é o ângulo entre o eixo horizontal e o segmento de reta que vai a origem ao ponto mais próximo da linha (expresso em graus), como mostrado na Figura 4.

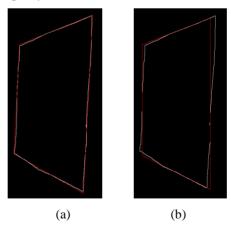
Figura 4: Reta parametrizada por ρ e θ que contém o ponto (x,y).



Dado um ponto (x,y), o conjunto de todas as retas que passam por este ponto correspondem à curvas senoidais no plano (ρ,θ) . Dois ou mais pontos em uma reta darão origem a duas ou mais curvas senoidais que se cruzarão no plano (ρ,θ) . Na transformada de Hough o plano (ρ,θ) pode ser considerado um histograma bidimensional dividido em intervalos discretos de ρ e θ . Um ponto (x,y) contribuirá com um incremento de uma unidade para diversos intervalos de classe discretos (ρ,θ) . Os vários pontos de uma reta contribuirão para uma contagem elevada de um único intervalo de classe dentre todos os outros possíveis.

Em diversos trabalhos relatados por [10], a resolução usada para θ é de 1° e para ρ , um pixel. A princípio pode-se tentar melhorar a precisão das estimativas das retas para obter uma resolução inferior a um pixel reduzindo-se o intervalo de classe para ρ e θ , como abordado por [11], porém limitações causadas por ruídos e desvios de linearidade impõem um limite para além do qual não há melhoria de precisão ou as estimativas resultam em linhas não representativas da realidade, sendo difícil determinar a resolução ótima para ρ e θ , como demonstrado por [12] - [15]. A inacurácia observada por [12] - [15] torna-se ainda mais severa na estimativa de linhas correspondentes ao contorno de uma superfície não-plana (como ondulações no papel de um flipchart, por exemplo). Para exemplificar, a Figura 5 mostra em branco o contorno de um flipchart (determinado com os procedimentos descritos na seção "Metodologia"). Observa-se que ondulações da superfície de papel resultam em um contorno que não pode mais ser representado por linhas retas. As linhas estimadas pela transformada de Hough são mostradas em vermelho. Na Figura 5-a, os intervalos de classe são de 1° para θ e um pixel para ρ e na Figura 5-b, 0,25° para θ e 0,25 pixel para ρ . Torna-se claro pela Figura 5 que a estimativa das linhas feitas com a menor resolução (figura 5-a) é mais fiel à realidade.

Figura 5: Linhas estimadas pela transformada de Hough (em vermelho) para um flipchart. (a) – intervalos de classe de 1º para θ e um pixel para ρ . (b) - intervalos de classe de 0,25º para θ e 0,25 pixel para ρ .



Outro procedimento para determinar os limites de um quadrilátero é o algoritmo conhecido como processador de Harris [16], capaz de detectar vértices em uma imagem usando o coeficiente de auto-correlação de uma pequena região da imagem. Diferentemente da transformada de Hough, o processador de Harris não requer a determinação prévia do contorno do quadro ou projeção e pode ser aplicado diretamente sobre a

imagem. A Figura 6 mostra o resultado do processador de Harris aplicado à projeção mostrada na Figura 3-a. O método identifica as regiões onde os quatro vértices estão localizados, mas ampliando a região do vértice superior direito, observa-se que a determinação exata do local é indefinida, impossibilitando uma resolução menor que um pixel. De fato, a localização pouco precisa de vértices determinados pelo processador de Harris foi estudada por [17] – [19], que apresentaram algumas melhorias no algoritmo, porém sem atingir resoluções inferiores a um pixel.

Figura 6: Processador de Harris aplicado à projeção de retângulo branco e detalhe do vértice do canto superior direito.



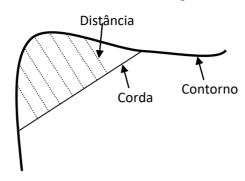
Em processamento de imagens, vértices são considerados "pontos de interesse", porém esta classificação não inclui somente vértices de um polígono, mas também junções, agrupamentos e variações de textura. Vários procedimentos de detecção de vértices são baseados em algoritmos de busca de "pontos de interesse", dentre os quais o de Harris é o mais significativo [20]. Estes algoritmos não requerem uma detecção explícita do contorno de um quadrilátero, sendo mais genéricos, porém impõem uma complexidade desnecessária ao problema de detecção de vértices. Entre os algoritmos de processamento de imagem há uma classe de detectores de vértices cujo procedimento é baseado na determinação prévia do contorno da imagem, sendo computacionalmente mais eficientes. Uma descrição do desempenho destes algoritmos é dada por [21] e [22] onde é mostrado que, a partir do contorno da imagem de um polígono, as regiões correspondentes às maiores curvaturas ao longo do contorno corresponderão aos vértices do polígono. Neste caso a curvatura c em um ponto e0 localizado no contorno é dada por (2), onde e1 o ângulo formado pela reta tangente à e2 e à horizontal, e e3 e o comprimento do arco definido pelo contorno.

$$c = \frac{d\gamma}{ds} \tag{2}$$

Porém a discretização e ruídos inerentes ao processo de captura e digitalização de imagens impõem limites à precisão deste método, como descrito por [23] e [24]. A determinação da curvatura dada por (2) requer o uso de derivadas de segunda ordem, suscetíveis a ruídos. Os mesmos autores também demonstram que qualquer processo de filtragem para redução de ruídos implica no deslocamento da real posição do ponto de interesse da imagem.

Já, o método proposto por [25] e melhorado por [23] é baseado na estimativa das distâncias de uma corda aos pontos do contorno, como exemplificado na Figura 7. Os resultados obtidos por [23] demonstram que a soma dos comprimentos de segmentos de reta entre a corda e os pontos do contorno, ortogonais à corda, resultam em um valor quantitativo representativo da curvatura e extremamente robusto e imune a ruídos. O comprimento da corda pode ser ajustado para uma maior rejeição a ruídos, sacrificando-se com isso a detecção de pequenos detalhes ao longo do contorno.

Figura 7: Distâncias de uma corda aos pontos do contorno.

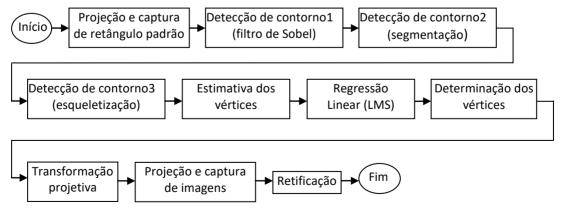


Ainda que o método das distâncias de uma corda aos pontos do contorno seja geometricamente simples, seu custo computacional é alto: à medida que a corda se desloca ao longo do contorno é necessário determinar várias retas ortogonais à corda e suas distâncias ao contorno. O método proposto no presente trabalho baseia-se na determinação da área de um triângulo formado por um ponto no contorno e dois outros pontos equidistantes, também localizados no contorno. É um método tão robusto e simples quanto os métodos apresentados em [23]-[25], porém computacionalmente mais eficiente. Além disso, [23] - [25] fazem uma associação direta entre os valores máximos de curvatura e os vértices de um polígono. Neste trabalho todos os pontos entre os valores de máxima curvatura são usados para a determinação das retas que formam os lados do polígono e o posterior cálculo das interseções das retas conseguiu estimar os vértices com uma precisão superior aos demais métodos.

2 Metodologia

A Figura 2 apresentou o fluxograma dos procedimentos usados em trabalhos similares sobre retificação de imagens. Esta seção descreverá os procedimentos de retificação usados neste trabalho que estão sumarizados no fluxograma da Figura 8. Adicionalmente, é apresentado nesta seção o algoritmo "Área" proposto neste trabalho como alternativa para identificação dos vértices de um polígono. De acordo com o fluxograma, as 4 primeiras etapas consistem na detecção do contorno de um retângulo padrão de cor branca, com dimensões de 800 x 600 pixels, projetado com resolução gráfica de 1024 x 768. A detecção do contorno baseia-se em algoritmos já conhecidos de filtragem, segmentação e esqueletização (ou afinamento, segundo alguns autores). As três próximas etapas são procedimentos de estimativa dos vértices desenvolvidos neste trabalho que formam o algoritmo designado por "Area". O processo de retificação é concluído pelas três etapas finais.

Figura 8: Fluxograma dos procedimentos de retificação usados neste trabalho.



As etapas de detecção de contorno iniciam-se com a aplicação dos processadores de Sobel ou Canny [26] à imagem de um retângulo padrão projetado em uma tela, capturada com uma câmara (Microsoft LifeCam

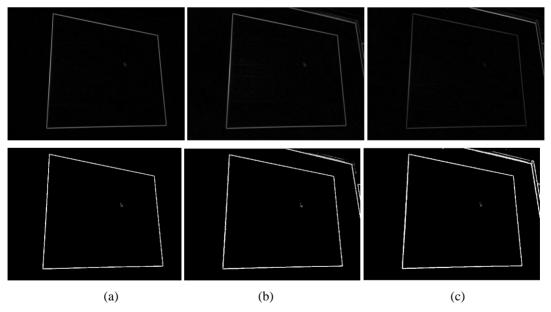
Cinema) com resolução gráfica de 640 x 480 pixels e 256 tons de cinza. Ainda que a câmara e projetor utilizados neste trabalho possam operar com maior resolução gráfica, o método foi propositadamente avaliado com a câmara e projetor operando em baixas resoluções, como encontrado comumente em qualquer sistema de captura e projeção de imagens.

A imagem do retângulo padrão projetado foi capturada sob três condições de iluminação: a) no escuro, b) com luzes acesas c) com luzes acesas e janelas abertas, como mostrado na Figura 3. Foram avaliadas várias combinações dos processadores de Sobel e Canny com diversos algoritmos de segmentação para garantir uma detecção do contorno do retângulo contínua e sem falhas sob todas as condições de iluminação. Existem vários algoritmos de segmentação que permitem automatizar a estimativa do valor de limiar, todos baseados na análise do histograma de tons de cinza da imagem [27]. Dentre estes algoritmos foram avaliados osde Otsu, Media, Entropy e Max_value, descritos por [28]. Porém, o algoritmo que apresentou o melhor desempenho para o caso em questão foi um pouco citado na literatura, conhecido como Métrico e descrito por [29]. Neste algoritmo, dado o histograma de tons de cinza de uma imagem, h(i), onde i representa o valor do tom de cinza, o limiar será o valor de k que minimiza a expressão:

$$\sum_{i=0}^{i=k} h(i) |(i-\mu_1)| + \sum_{i=k+1}^{i=N-1} h(i) |(i-\mu_2)|$$
(3)

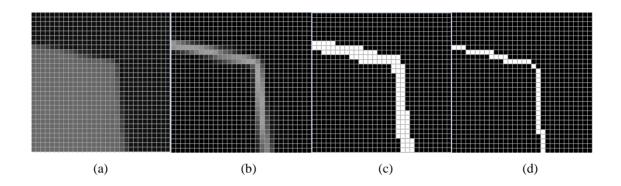
onde N é o número de tons de cinza (256), μ_1 é a média dos valores do brilho dos pixels entre 0 e k e μ_2 é a média dos valores do brilho dos pixels entre k+1 e 255. A Figura 9 mostra o resultado da aplicação do processador de Sobel e do algoritmo de segmentação Métrico às imagens mostradas na figura 3.

Figura 9: Resultado do processador de Sobel (acima) e algoritmo de segmentação Métrico(abaixo) aplicados à projeção de retângulo padrão sob três condições de iluminação: (a) no escuro, (b) com luzes acesas, (c) com luzes acesas e janelas abertas.



A aplicação de um processo de segmentação após o processador de Sobel gera contornos com espessura de alguns pixels. O algoritmo de esqueletização (ou afinamento) proposto por [30] transforma o contorno em uma sequência conectada de pixels, gerando um contorno com espessura de um pixel, como mostrado na Figura 10.

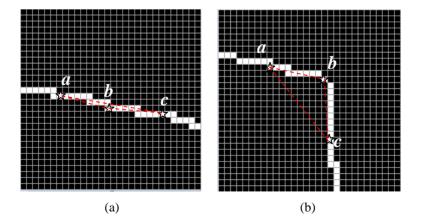
Figura 10: (a) Detalhe do vértice superior direito. (b) Aplicação do processador de Sobel. (c) Segmentação. (d) Esqueletização.



A esqueletização permite que o contorno possa ser tratado como um código de Cadeia (ou código de Freeman [31]). Esta codificação, além de permitir um armazenamento compacto do contorno de imagens, permite o uso de algoritmos eficientes para a detecção de vértices e ângulos. Observa-se que desde o trabalho pioneiro de Freeman, em 1977, até os mais recentes [32], a determinação de vértices com o Código de Cadeia é baseada no cálculo de diferenças de ângulos entre segmentos de reta que excursionam pelo contorno. Em [21] é feita uma compilação extensiva sobre este método, incluindo diversas propostas de melhorias para minimizar ruídos e otimizar a precisão. No entanto ainda que o uso de Código de Cadeia nos trabalhos relatados em [21] reduza o número de pixels sobre os quais o algoritmo é executado, há sempre várias operações com funções trigonométricas, exponenciais e vetoriais, que contribuem para a redução da eficiência computacional dos algoritmos. Já, o método para a identificação dos vértices de um polígono proposto neste trabalho denominado "Área", consiste em calcular a área A de um triângulo formado por três pixels $(a_{xx}a_{y})$, $(b_{xx}b_{y})$ e $(c_{xx}c_{y})$, dada por (4), localizados no contorno e espaçados de N pixels ao longo do contorno. O pixel central $(b_v b_v)$ se desloca ao longo do contorno até retornar à posição inicial, escolhida aleatoriamente. . Desta forma, este método evita as operações trigonométricas e exponenciais que implicam no aumento do tempo de processamento. A Figura 11 ilustra o método em dois casos; com os pixels aproximadamente alinhados e com o pixel central localizado em um vértice, para N = 10.

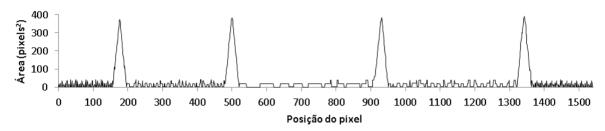
$$A = |a_x(b_y - c_y) + b_x(c_y - a_y) + c_x(a_y - b_y)|$$
(4)

Figura 11: Três pixels localizados no contorno e espaçados de 10 pixels. (a) Ao longo de uma seção aproximadamente reta. (b) Em um vértice do contorno



De maneira análoga ao que ocorre nos trabalhos relatados em [21], um aumento do valor N melhora a rejeição a ruídos, em sacrifício da detecção de pequenos detalhes ao longo do contorno. A Figura 12 mostra a estimativa das áreas para N=20, onde de observa que a detecção dos vértices é clara e bem destacada. Assim como o relatado em [25] - [27] para o caso da detecção por meio do ângulo entre os segmentos de reta, não há como determinar o melhor valor do comprimento dos segmentos, ou de N no caso, sendo que dobrando-se ou reduzindo-se este valor à metade, a estimativa da posição dos vértices varia de no máximo dois pixels na vertical ou horizontal. Esta variação faria com que este método não apresentasse vantagem alguma no que diz respeito à resolução em relação a outros trabalhos, que também relatam o mesmo comportamento.

Figura 12: Área do triângulo formado por três pixels localizados ao longo do contorno, equidistantes de 20 pixels.



A precisão da determinação dos vértices é melhorada com uma etapa posterior, na qual a posição prévia dos vértices determinada pelos quatro valores de pico mostrados na Figura 12 é usada para coletar todos os pixels localizados entre dois vértices. Com isso, cerca de 250 a 400 pixels são usados para determinar, por regressão linear, a reta que representa cada um dos lados do quadrilátero. O cruzamento destas retas fornece um novo valor para os vértices, mais acurado que a estimativa inicial.

Após a determinação final dos vértices do quadrilátero, calcula-se a transformação projetiva que relaciona as posições dos mesmos com os vértices do retângulo padrão (conhecidos). Equipamentos de projeção transferem os pontos X_i de um plano para os pontos x_i localizados em uma tela plana. Todos os raios que ligam os pontos x_i a X_i passam pelo centro de projeção C do projetor. De maneira análoga, todos os pontos x_i na tela são transferidos para os pontos x_i localizados no dispositivo de formação de imagens da câmara, passando pelo centro de projeção C'. De acordo com [33], se todos os pontos forem coplanares, é possível relacionar x_1 , x_2 , x_3 e x_4 com x_1 , x_2 , x_3 e x_4 por meio de uma transformação projetiva x_3 dada por (5).

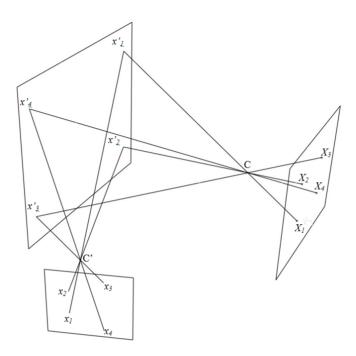
$$X_i = H_{3x3}x_i \tag{5}$$

Observa-se pela Figura 13 que a distorção por perspectiva ocorre em duas etapas, a primeira delas é a distorção causada pelo posicionamento do projetor em relação à tela (geralmente minimizada) e a segunda é a distorção causada pelo posicionamento da câmara em relação à tela, geralmente a mais significativa. É possível compensar simultaneamente as duas distorções por meio de uma única transformação projetiva dada por (5), onde (6) é sua forma expandida e w é um fator de escala.

$$\begin{pmatrix} Xw \\ Yw \\ w \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$
 (6)

Em uma transformação homográfica, bastam quatro correspondências entre os pontos x_i e X_i para resultarem em 8 equações lineares que permitem o cálculo da matriz H, desde que não haja 3 pontos colineares, o que é uma condição sempre atendida pela geometria do caso em questão [34].

Figura 13: Transformação projetiva de pontos X_i do sistema de projeção para pontos x_i da imagem capturada por uma câmara. C é o centro de projeção do dispositivo gerador de imagens e C' é o cento de projeção da câmara. x_i são pontos da tela.



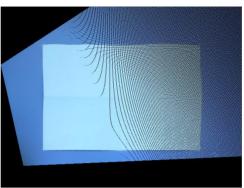
Considerando que os valores de X_i são arbitrários e x_i são determinados pelos procedimentos anteriormente descritos, quatro correspondências entre X_i e x_i resultam em (7).

$$\begin{bmatrix} x_{1} & y_{1} & 1 & 0 & 0 & 0 & -X_{1}x_{1} & -X_{1}y_{1} \\ 0 & 0 & 0 & x_{1} & y_{1} & 1 & -Y_{1}x_{1} & -Y_{1}y_{1} \\ x_{2} & y_{2} & 1 & 0 & 0 & 0 & -X_{2}x_{2} & -X_{2}y_{2} \\ 0 & 0 & 0 & x_{2} & y_{2} & 1 & -Y_{2}x_{2} & -Y_{2}y_{2} \\ x_{3} & y_{3} & 1 & 0 & 0 & 0 & -X_{3}x_{3} & -X_{3}y_{3} \\ 0 & 0 & 0 & x_{3} & y_{3} & 1 & -Y_{3}x_{3} & -Y_{3}y_{3} \\ x_{4} & y_{4} & 1 & 0 & 0 & 0 & -X_{4}x_{4} & -X_{4}y_{4} \\ 0 & 0 & 0 & x_{4} & y_{4} & 1 & -Y_{4}x_{4} & -Y_{4}y_{4} \end{bmatrix} \begin{pmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ \end{pmatrix} \begin{pmatrix} X_{1} \\ Y_{1} \\ X_{2} \\ X_{3} \\ Y_{3} \\ X_{4} \\ Y_{4} \\ \end{pmatrix}$$

$$(7)$$

Após a determinação dos elementos da matriz H qualquer pixel (x,y) de uma imagem pode ser mapeado para sua posição retificada (X,Y) [33]. Neste mapeamento não há uma correspondência biunívoca entre os pixels (x,y) e (X,Y) conforme mostrado na Figura 14, onde observa-se o resultado da retificação de um flipchart branco sobre um fundo azul. A ausência de alguns pixels na imagem retificada é intuitivamente explicada pelo fato das regiões localizadas à direita sofrerem uma expansão. Portanto nem todo o pixel (X,Y) pode ter seu correspondente pixel (x,y). Neste trabalho o valor dos pixels ausentes é obtido pela interpolação dos valores dos pixels vizinhos, onde cada pixel vizinho só é considerado se seu valor for diferente de zero (fundo preto).

Figura 14: Retificação de um flipchart branco sobre fundo azul.



3 Resultados

O algoritmo para identificação dos vértices de um polígono proposto neste trabalho, doravante designado por "Área" será comparado com os quatro métodos descritos nas seções anteriores, denominados por "Corda" ([23] e [25]), "Curvatura" ([21] – [24]), "Hough" ([9] – [15]) e "Harris" ([16] – [20]). No algoritmo "Área" os vértices são determinados em duas etapas: na primeira, os picos resultantes (Figura 12) do cálculo da área de triângulos definidos por três pontos localizados no contorno de um quadrilátero projetado fornecem uma aproximação inicial da localização dos vértices. Na segunda etapa a posição é refinada estimando-se a equação das retas que formam os lados do quadrilátero usando-se uma regressão linear com todos os pontos do contorno localizados entre dois vértices. A posição final dos vértices é obtida pelo cruzamento destas retas. Comparando-se os resultados da Figura 12 com gráficos semelhantes apresentados por [21] e [23] para o método "Corda" e por [31] e [32] para o método "Curvatura", observa-se que o método "Área" apresenta picos bem mais definidos, permitindo a identificação precisa dos vértices de um quadrilátero.

Para comparações mais quantitativas serão avaliados o tempo de processamento para a determinação dos vértices e a fidelidade da imagem retificada com a original. Todos os métodos foram realizados na linguagem Processing e executados em um computador notebook Acer Aspire (Intel Pentium N3700 1.6 GHz 4096 MB).

O tempo de processamento para a determinação dos vértices inclui os procedimentos de carregar a imagem já esqueletizada de um polígono branco projetado, como mostrado na figura 3-a, percorrer o contorno e calcular a área para todos os pontos do contorno. A Tabela 1 mostra o tempo de processamento, em milisegundos, para os diversos algoritmos. Ainda que o algoritmo de Harris não exija que a imagem seja previamente esqueletizada, o mesmo foi incluído na tabela para efeito de comparação.

Tabela 1: Tempo de processamento (ms)

Área	Corda	Curvatura	Harris	Hough
20	36	57	120	570

Observa-se pela tabela que o método proposto requer um tempo de processamento inferior ao dos outros dois métodos semelhantes (Corda e Curvatura). Os outros dois métodos, Harris e notadamente o de Hough, resultaram em tempos bem superiores. Os tempos mostrados na Tabela 1 referem-se à retificação de uma imagem capturada com resolução de 640 x 480 pixels. Testes feitos com resoluções maiores resultaram em tempos de processamento inviáveis para os métodos de Harris e Hough. De fato, estes últimos algoritmos apresentam um tempo de processamento proporcional à N^2 , onde N representa o número de pixels da imagem. Os métodos baseados no percurso ao longo de um contorno aumentam seu tempo de processamento segundo uma dependência linear com N.

O outro critério de comparação, a fidelidade de uma imagem retificada com a original (de referência), indica a precisão com que os vértices do retângulo padrão foram determinados. A fidelidade entre as imagens será determinada pelo coeficiente de correlação bidimensional r_{ii} , definido por (8), onde g(x,y) é o brilho do pixel

em (x,y) na imagem de referência e f(x,y) é o brilho do pixel em (x,y) na imagem retificada transladada de (i,j) pixels. \bar{f} e \bar{g} são os valores médios do brilho das respectivas imagens.

$$r_{ij} = \frac{\sum_{x} \sum_{y} [f(x+i,y+j) - \bar{f}] [g(x,y) - \bar{g}]}{\sqrt{\sum_{x} \sum_{y} [f(x,y) - \bar{f}]^{2} \sum_{x} \sum_{y} [g(x,y) - \bar{g}]^{2}}}$$
(8)

A Tabela 2 mostra na primeira coluna a figura original que foi projetada e na segunda coluna o resultado da correção de perspectiva pelo método deste trabalho ("Área"). As outras colunas apresentam o coeficiente de correlação obtido pelos cinco métodos para cada uma das três figuras. Para estes casos, bem para como outras figuras analisadas, o coeficiente de correlação para o método proposto manteve-se sempre acima dos coeficientes obtidos com outros métodos. Os métodos da "Corda" e "Curvatura" resultam em valores similares, porém os de "Hough" e "Harris" apresentam valores significativamente inferiores.

Referência Retificada Área Corda Curvatura Harris Hough 0,92 0,90 0,83 0,71 0,74 0.76 0.70 0.65 0.59 0.61 0,81 0,75 0.71 0,56 0,63

Tabela 2: Coeficiente de Correlação

4 Conclusões

A correção de perspectiva de imagens digitais é empregada em vários campos da engenharia, sendo um tópico importante na área de processamento de imagens. Recentemente, com o uso cada vez mais disseminado de sistemas de projeção e de dispositivos móveis de captura de imagens, surgiram algumas aplicações na área didática e educacional com o objetivo de corrigir a distorção de imagens capturadas em apresentações e aulas expositivas. Este trabalho apresentou um procedimento para realizar a correção desta distorção e, em particular, um algoritmo para a detecção de vértices de um quadrilátero.

A correção de perspectiva requer a identificação prévia de características geométricas de uma imagem para que possam ser usados como pontos de referência (ou de interesse). Os diversos trabalhos mencionados mostram que esta identificação é baseada, em grande parte dos casos, na busca de pontos de referência (retas, ângulos, texturas, gradientes, entre outros.) cujos algoritmos operam de maneira genérica, não tendo a priori nenhuma geometria a ser buscada, impondo com isso uma complexidade desnecessária. Por outro lado, há outros métodos baseados na detecção explícita do contorno de um quadrilátero, sendo computacionalmente mais eficientes. O algoritmo proposto apresenta um novo método para identificar vértices a partir do Código de Cadeia relatado no trabalho pioneiro de Freeman, cuja determinação de vértices é baseada no cálculo de diferenças de ângulos entre segmentos de reta que excursionam pelo contorno. Este trabalho apresentou um algoritmo simples, já que está baseado num cálculo de área, e eficiente para identificar os vértices de um retângulo padrão projetado ao início de uma apresentação. Com a identificação dos vértices com o algoritmo "Área" proposto obteve-se uma correção de perspectiva computacionalmente mais eficiente e com maior precisão em comparação com outros métodos.

Todos os algoritmos de processamento de imagens descritos foram realizados com a linguagem Processing, que permite a geração de códigos compactos e eficientes para diversos dispositivos de captura de imagens e sistemas operacionais. Uma captura e retificação de uma imagem, mesmo com modos de alta resolução gráfica (4 MPixels) pode ser feita instantaneamente, sem atraso perceptível para o usuário. Mesmo com modos de resolução mais baixos (640 x 480 pixels) a qualidade das imagens retificadas revelou-se plenamente satisfatória para objetivos didáticos.

Os métodos descritos neste artigo serão aplicados em outras áreas de conhecimento, dentre as quais podese citar Arquitetura, Cartografia e Artes Visuais, e os algoritmos aqui propostos podem ser modificados por pessoas que não são profissionais em Computação. Segundo seus criadores, a linguagem Processing foi concebida para facilitar a programação gráfica e interativa, identificada por eles como difícil e frustrante de ser realizada com linguagens como C++ e Java [35]. A motivação para a criação do Processing foi o de possibilitar tanto a criação rápida de protótipos de programas gráficos e interativos quanto a de ensinar programação para alunos de Design, de Artes e demais interessados em programas que fazem uso de desenhos, imagens, vídeos, sons e interação. Desta forma, os algoritmos elaborados em Processing podem ser alterados de maneira mais rápida e imediata por usuários com tal perfil multidisciplinar. Além disso, a linguagem Processing é gratuída e disponível para os sistemas operacionais Windows, Linux, MacOs e Android; possui uma IDE simples e já conta com uma quantidade significativa de bibliotecas também gratuitas fornecidas por uma ampla comunidade de desenvolvedores.

Referências

- [1] Long, T.; Jiao, W.; He, G.; Zhang, Z.; Cheng, B.; Wang, W. A generic framework for image rectification using multiple types of feature, *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 102, pp 161–171, February 2015.
- [2] Zhang, Y.; Hu, B.; Zhang, J. Relative orientation based on multi-features, *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 66 no. 5, pp. 700–707, 2011.
- [3] Long, T.; Jiao, W.; He, G.; Wang, W. Automatic line segment registration using gaussian mixture model and expectation-maximization algorithm, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7 no. 5, pp. 1688–1699, June 2014.
- [4] An, J.; Kim, B. S.; Koo, H. I.; Cho, N. I. Unified framework for automatic image stitching and rectification, *Journal of Electronic Imaging*, vol. 24 no. 3, pp. 701 711, June 2015.
- [5] Pohl, M.; Schaeferling, M.; Kiefer, G.; Petrow, G.; Woitzel, E.; Papenfuß, F. Leveraging polynomial approximation for non-linear image transformations in real time, *Computers and Electrical Engineering* vol. 40, pp. 1146–1157, January 2014.
- [6] Stamatopoulos, N.; Gatos, B.; Pratikakis, I.; Perantonis, S.J. Goal-Oriented Rectification of Camera-Based Document Images, *IEEE Transactions on Image Processing*, vol. 20, no. 4, pp. 910-920, April 2011.
- [7] Evening, M. The Adobe Photoshop Lightroom 3 Book: The Lens Corrections Panel. Adobe Press, 2010.
- [8] Zhang, Z.; He, L. W. Whiteboard scanning and image enhancement, *Digital Signal Processing* vol. 17, pp. 414–432, 2007.
- [9] Hough, P.V.C. A Method and Means for Recognizing Complex Patterns, U.S. Patent 3,069,654, December 18, 1962.
- [10] Mukhopadhyay, P.; Chaudhuri, B.B.; A survey of Hough Transform, *Pattern Recognition*, vol.,48, pp. 993-1010, 2015.
- [11] Veen, T. M.; Groen, F. C. A. Discretization errors in the Hough Transform, *Pattern Recognition*, vol. 14, no. 1, pp. 137–145, 1981.
- [12] Du, S.; Wyk, B. J.; Tu, C.; Zhang, X. An improved Hough Transform neighborhood map for straight line segments, *IEEE Transactions on Image Processing* vol. 19, no.3, pp. 573–585, March, 2010.
- [13] Lee, D.; Park, Y. Discrete Hough transform using line segment representation for line detection, *Optical Engineering*, vol.50, no. 8, pp. 087004-1 087004- 5, August 2011.

- [14] Ji, J.; Chen, G.; Sun, L. A novel Hough transform method for line detection by enhancing accumulator array, *Pattern Recognition Letters*, vol. 32, no. 11, pp.1503-1510, 2011
- [15] Zhao, Y.; Pan, H.; Du, C.; Zheng, Y. Principal direction-based Hough transform for line detection, *Optical Review*, vol. 22, no. 2, pp.224-231, March 2015.
- [16] Harris, C.; Stephens, M. A combined corner and edge detector. In: 4th Alvey Vision Conference, *Proceedings*... pp. 147–151, 1988.
- [17] Ryu, J. B.; Park, H. H.; Log-log scaled Harris corner detector, *Electronics Letters*, vol.46, no.24, pp.1-2, November, 2010.
- [18] Bertrand, M.; Bouchara, F.; Ramdani, S. Estimation of uncertainty for Harris corner detector. In: 2nd International Conference on Image Processing Theory, Tools and Applications, *Proceedings*... pp.249-252, July 2010.
- [19] Yang, H.; Ma, Y.; Yang, H.; Li, M. Parts Shape Recognition Based on Improved Harris Corner Detection Algorithm, *Journal of Computers*, vol.9, no.10, pp.2371-2378, October 2014.
- [20] Mikolajczyk, K.; Schmid, C. Scale & affine invariant interest point detectors, *International Journal of Computer Vision*, vol. 60, no. 1, pp. 63–86, October 2004
- [21] Awrangjeb, M.; Lu, G., Fraiser, C. S. Performance Comparisons of Contour-Based Corner Detectors, *IEEE Transactions on Image Processing*, vol. 21, no. 9, pp. 4167-4179, May 2012.
- [22] Rattarangsi, A.; Chin, R. T. Scale-based detection of corners of planar curves, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 4, pp. 430–449, April 1992.
- [23] Awrangjeb, M.; Lu, G. Robust Image Corner Detection Based on the Chord-to-Point Distance Accumulation Technique, *IEEE Transactions on Multimedia*, vol. 10, no. 6, pp. 1059-1072, October 2008.
- [24] Zhu, P.; Chirlian, P. M. On Critical Point Detection of Digital Shapes, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 737 748, August 1995.
- [25] Phillips, T., Y.; Rosenfeld, A. A method of curve partitioning using arc-chord distance, *Pattern Recognition Letters*, vol. 5, pp. 285–288, April 1987.
- [26] Umbaugh, S. E. Digital image processing and analysis: human and computer vision applications with CVIPtools, 2nd ed. Boca Raton, FL: CRC Press, 2010.
- [27] Glasbey, C.A.; Horgan, G.W. Image Analysis for the Biological Sciences, Wiley, 1th edition, 1995.
- [28] Sezgin, M.; Sankur, B. Survey over image thresholding techniques and quantitative performance evaluation, *Journal of Electronic Imaging*, vol. 13, no.1, pp. 146–165, 2004.
- [29] National Instruments, NI Vision Concepts Manual, National Instruments, 2007.
- [30] Zhang, T. Y.; Suen, C. Y. A fast parallel algorithm for thinning digital patterns, *Communications of the Association of Computing Machinery*, vol. 27, no. 3, pp. 236-239, March 1984.
- [31] Freeman, H.; Davis, L. S. A Corner Finding Algorithm for Chain Coded Curves, *IEEE Transactions in Computing*, vol. 26, pp. 297 303, 1977.
- [32] Naim, N.; Laxmi, V.; Bhadviya, B.; Singh, N. C. Corner Detection using Difference Chain Code as Curvature. In: International MultiConference of Engineers and Computer Scientists, 1. *Proceedings*... pp. 633-638, 2008.
- [33] Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*, Cambridge university press, 2nd Edition, 2004.
- [34] Criminisi, A.; Reid, I.; Zisserman. A. A plane measuring device, *Image and Vision Computing*, vol. 17 no. 8, pp. 625–634, 1999.
- [35] Reas, C.; Fry, B., Getting started with Processing, Ed O'Reilly, 2010.