

Rastreamento Robusto de Olhos Usando Consistência Temporal e Restrições Geométricas

Gabriel M. Araujo^{†‡}, Eduardo A. B. da Silva[†], Felipe M. L. Ribeiro[†], José F. L. de Oliveira[†], Alexandre G. Ciancio[†] e Amir Said[§]

Resumo— Rastreamento robusto em vídeo pode ser realizado integrando detecção e rastreamento. Em um trabalho anterior, utilizamos a consistência temporal do ponto detectado/rastreado para rejeitar detecções inconsistentes e um modelo geométrico para estimar pontos faltantes. Entretanto, se uma detecção ruim tiver consistência temporal, o modelo estimará pontos errados. Propomos uma técnica para avaliar a consistência do modelo geométrico. Tal abordagem é capaz de ajustar o modelo à face do indivíduo e avaliar a qualidade dos pontos estimados. Nos experimentos, utilizamos cinco vídeos de alta definição. Os resultados mostram a capacidade de detectar/rastrear olhos com boa consistência.

Palavras-Chave— Visão Computacional, Rastreamento em Vídeo, Rastreamento de Faces, Rastreamento de Olhos, Detecção de Objetos.

Abstract— Robust video tracking can be achieved by integrating detection and tracking. In a previous work, we have used the temporal consistency of the detected/tracked point to discard unreliable detections and a geometric model to estimate missing points. However, if an unreliable detection has temporal consistency, the model will estimate wrong points. We propose a technique to evaluate the model's consistency. This approach is capable of adjusting the model to the subject's face and evaluate the quality of the estimations. In our experiments, we use five high definition videos. The results have shown the capability of detecting/tracking eyes with good reliability.

Keywords— Computer Vision, Video Tracking, Face Tracking, Eye Tracking, Object Detection.

I. INTRODUÇÃO

Atualmente, temos uma crescente demanda por sistemas móveis. Isso tem acarretado em uma revolução na forma como interagimos com os nossos dispositivos. Cada vez mais, utilizamos a voz, gestos e expressões para controlar nossos aparelhos. Essas duas últimas formas concorrem para o problema de localização e rastreamento de objetos em sequências de vídeo. A detecção e o rastreamento de olhos, por exemplo, tem sido usada para controlar *smartphones* [1]. Em [2], tratamos do problema de detecção e rastreamento robusto de pontos fiduciais nos olhos.

Neste trabalho, a detecção dos pontos de interesse foi feita por um detector baseado em filtros de correlação chamado *Inner Product Detector* (IPD) [3]. A consistência da detecção foi avaliada a partir da combinação da saída do detector com a saída de um rastreador baseado em fluxo óptico muito

conhecido: o Kanade-Lucas (KL) [4]. Assim, as detecções e os rastreamentos com evolução temporal inconsistentes puderam ser eliminados. Entretanto, isso aumenta o número de quadros em que o sistema não fornece uma saída. A solução encontrada foi aplicar restrições geométricas para obter um modelo para a posição relativa entre os pontos rastreados e então estimar os pontos faltantes e/ou que foram eliminados. Todavia, se uma detecção ruim possui consistência temporal, as transformações geométricas podem fornecer estimativas erradas.

Neste trabalho, melhoramos a forma de estimar os pontos faltantes. O método proposto também utiliza restrições geométricas para obter uma transformação que mapeia os pontos corretamente detectados em um quadro anterior no quadro atual. Entretanto, elaboramos uma forma de avaliar não só a qualidade do modelo geométrico obtido, como também dos pontos estimados. Para tanto, em cada quadro, o modelo e os pontos consistentes são utilizados para obter a homografia 2D que mapeia o modelo no quadro atual. Comparando os pontos estimados com as detecções consistentes, podemos avaliar a qualidade do modelo e, portanto, obter melhores estimativas.

O restante deste trabalho está organizado da seguinte forma. Na seção seguinte, apresentamos um breve resumo sobre o IPD. Na Seção III, descrevemos como é feita a integração entre o IPD e o KL através da consistência temporal. Na Seção IV, descrevemos o método de estimação dos pontos faltantes proposto em [2], bem como as melhorias introduzidas neste trabalho. Os procedimentos experimentais, resultados obtidos e discussões estão na Seção V. Na Seção VI, concluímos e apresentamos algumas sugestões para trabalhos futuros.

II. DETECÇÃO DE CARACTERÍSTICAS FACIAIS USANDO O DETECTOR POR PRODUTO INTERNO

Em filtragem de correlação, a detecção de um padrão é dada pela correlação cruzada entre o filtro e um sinal desconhecido [5]. O resultado dessa operação é grande quando o sinal representa o padrão desejado e pequeno caso contrário. Uma das principais vantagens dessa técnica é a robustez a pequenas variações do padrão de interesse. Na etapa de detecção deste trabalho, usamos um detector baseado em filtros de correlação conhecido como *Inner Product Detector* (IPD) [3]. Considere um problema de classificação de duas classes e uma variável aleatória \mathbf{X} , cujas realizações \mathbf{x} pertencem a uma das classes, C_0 ou C_1 . No IPD, o objetivo é obter um detector h_{C_1} capaz de detectar amostras pertencentes a classe C_1 . Idealmente, o produto interno do detector com uma amostra desconhecida \mathbf{x} é igual a 1 se $\mathbf{x} \in C_1$ e zero caso contrário.

[†]PEE/COPPE/DEL/POLI, Universidade Federal do Rio de Janeiro, Cx. P. 68504, Rio de Janeiro – RJ, 21945-970, BRASIL, [‡]Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (CEFET/RJ–UnED–NI), Estrada de Adrianópolis, 1.317, Nova Iguaçu – RJ, 26041-271, Brazil, [§]LG Electronics, USA. E-mails: {gabriel.araujo, eduardo, felipe.ribeiro, jose.oliveira, alexandre.ciancio}@smt.ufrj.br, amir.said@lge.com. Este trabalho foi parcialmente financiado pelo CNPq.

A solução ótima no sentido dos mínimos quadrados para este problema é [3]:

$$\mathbf{h}_{C_1} = \left(\sum_{i=0}^1 p(C_i) \mathbf{R}_{C_i} \right)^{-1} p(C_1) \boldsymbol{\mu}_{C_1}, \quad (1)$$

Onde $\boldsymbol{\mu}_{C_i}$ é a média amostral dos elementos da classe C_i , \mathbf{R}_{C_i} é a matriz da autocorrelação dos elementos da classe C_i e $p(C_i)$ é a probabilidade *a priori* de uma amostra pertencer a classe C_i . Vale ressaltar que a estimativa \mathbf{R}_{C_i} deve ser inversível. Na prática isso significa que o número de amostras da classe C_i para obter o classificador deve ser muito maior que a dimensão do vetor \mathbf{X} .

Em dados do mundo real as classes não são ortogonais entre si, logo o produto interno entre o detector e uma amostra não tende a ser fortemente distribuído entre 0 e 1. Em outras palavras, o classificador dado pela equação (1) é fraco. Entretanto, um classificador forte pode ser obtido pela combinação de classificadores fracos [6]. Neste trabalho, um classificador forte é obtido a partir de uma combinação em cascata de vários classificadores IPD. A saída dessa cascata consiste em um conjunto de pontos com alta probabilidade de pertencer à classe de interesse (no restante deste trabalho iremos nos referir a este conjunto como nuvem de pontos). Para descartar os falsos positivos da nuvem, um pós-processamento deve ser aplicado. Os classificadores IPD usados neste trabalho foram treinados usando a base de dados BioID [7]. Detalhes sobre o IPD e o procedimento de treino podem ser encontrados em [3].

III. INTEGRAÇÃO ENTRE DETECÇÃO E RASTREAMENTO CONSIDERANDO A CONSISTÊNCIA TEMPORAL

O IPD possui três características que merecem ser destacadas. A primeira delas é que o IPD herda dos filtros de correlação a capacidade de tolerar pequenas variações do padrão de interesse. Outro ponto importante é que a função discriminante do IPD consiste essencialmente em avaliar o resultado de um produto interno. Logo, ele é rápido o bastante para ser utilizado em aplicações de tempo real [8]. Como mencionado na Seção II, a saída do detector é uma nuvem de pontos. Logo, a outra vantagem do IPD é que estes pontos tendem a estar agrupados em pequenos grupos que são altamente correlacionados com o padrão desejado [3].

Neste trabalho usamos a saída da cascata de IPDs para alimentar um rastreador. Como rastreador, utilizamos um algoritmo baseado em fluxo óptico muito conhecido, o Kanade-Lucas (KL) [4]. Além disso, usamos a consistência entre a saída do rastreador KL e a saída da cascata de IPDs para descartar detecções inconsistentes. Para tanto, computamos histogramas da nuvem de pontos da seguinte forma:

- 1) Seja j o índice dos quadros na sequência de vídeo. Calcule a mediana vetorial da nuvem de pontos para obter um único ponto y_j .
- 2) Alimente o rastreador com o ponto y_{j-1} detectado em um quadro anterior e rastreie ele no quadro atual para obter a estimativa y'_j .
- 3) Seja x_i ($i = 1, 2, \dots, N$) um ponto de uma nuvem composta por N pontos e d_i a distância entre o ponto rastreado y'_j e o ponto x_i . Compute o histograma das distâncias d_i .

- 4) Analisando o histograma, é possível distinguir uma detecção consistente de uma inconsistente. (veja a descrição abaixo sobre a análise dos histogramas). Se a detecção é consistente, use a mediana vetorial da nuvem y_j como saída e vá para o passo (2). Caso contrário, não forneça um ponto de saída e vá para o passo (1).

Uma vez que a nuvem pode apresentar dois ou mais conjuntos disjuntos, o uso da mediana vetorial garante que o ponto de saída pertence à nuvem. Isso poderia não ocorrer caso usássemos a média (ou centro de massa) da nuvem.

A partir da análise de histogramas típicos, podemos distinguir entre quatro tipos de comportamento, que estão ilustrados nas Figuras 1(a) até 1(d). O histograma da Figura 1(a) é unimodal, que tende a estar relacionado com uma nuvem formada por um único agrupamento próximo do padrão de interesse. Este histograma pode indicar uma detecção consistente. O histograma da Figura 1(b) possui duas modas. De uma maneira geral, uma delas corresponde a um agrupamento próximo do ponto de interesse (vale ressaltar que o histograma pode ter mais de duas modas). Este segundo caso também pode indicar uma detecção consistente. O motivo disso é que podemos analisar o comportamento temporal de cada um destes agrupamentos e então escolher aquele cujo centróide descreve o caminho mais suave (essa análise será discutida com detalhes ainda nesta seção). Os outros dois histogramas típicos que estão ilustrados nas Figuras 1(c) e 1(d) indicam detecções inconsistentes e devem ser rejeitadas. O histograma da Figura 1(c) apresenta poucos pontos aparecendo em várias modas e geralmente está relacionado com pontos isolados devido ao ruído. Já o histograma da Figura 1(d) possui uma aparência uniforme, ou seja, sem um pico discernível. Tal comportamento corresponde a uma nuvem com muitos pontos espalhados.

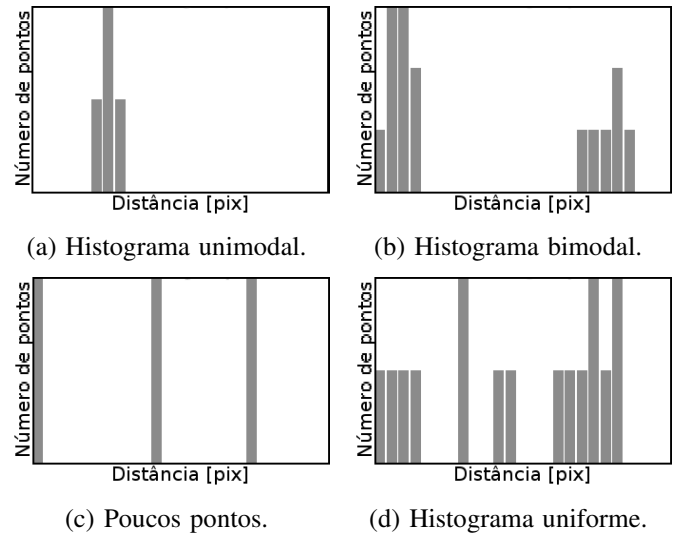


Fig. 1. Histogramas típicos para a detecção do canto interno do olho direito, obtidos na detecção do canto e de uma sequência de vídeo de dificuldade intermediária.

Na análise dos histogramas, descartamos as nuvens de detecção que são inconsistentes. Assim, é possível que tenhamos alguns quadros sem nuvem de pontos. Nos quadros restantes, temos uma sequência temporal de agrupamentos de pontos que tendem a estar próximos do padrão de interesse.

Podemos observar a evolução temporal dos centroides destes agrupamentos, manter aqueles que descrevem um caminho suave e descartar os agrupamentos restantes. Para tanto, usamos a estratégia ilustrada na Figura 2. Os pontos representam os pontos da nuvem, as cruces os centroides dos respectivos agrupamentos e o quadrado representa a saída de um quadro anterior. Os centroides são obtidos pelo emprego do algoritmo de agrupamento hierárquico (Hierarchical Clustering) [9] na nuvem de pontos. No exemplo desta figura, assumimos que foi feita a análise dos histogramas em todos os quadros até o quadro k . O ponto de saída de um quadro anterior ($k-2$, neste caso) é projetado no quadro atual e verificamos se existe algum centroide dentro de um disco em torno do ponto projetado. Se houver, o mais próximo do centro do disco será mantido e os restantes são descartados. O raio do disco depende da sequência em questão. Neste trabalho, testamos valores no intervalo de 1 a 10 pixels. Os melhores resultados foram obtidos usando um disco com raio de 3 pixels para a sequência fácil e 4 pixels para as demais (detalhes sobre as sequências são dados na Seção V-A). Note que o impacto da resolução no tamanho do disco não foi avaliado. Se nenhum disco correspondente às saídas dos últimos 5 quadros contiver algum centróide, então todos os agrupamentos são considerados inconsistentes e descartados.

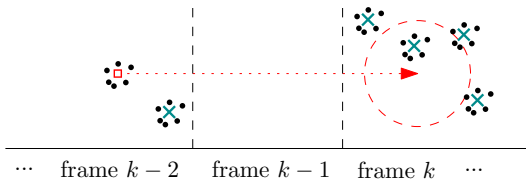


Fig. 2. Evolução temporal da nuvem de pontos após a análise dos histogramas. Os quadros estão separados por linhas tracejadas. A saída de um quadro anterior ($k-2$) é projetada no quadro atual (k). Um disco de quatro pixels em torno do ponto projetado é selecionado. O centroide que estiver dentro deste disco será considerado consistente e os restantes serão descartados. Se o disco contiver mais de um centroide, o mais próximo do centro é escolhido. Note que não há uma saída consistente no quadro $k-1$.

Os centroides dos agrupamentos que permanecerem consistentes após a análise do histograma e a consistência temporal possuem uma alta probabilidade de corresponder ao ponto desejado. Em outras palavras, teremos uma baixa taxa de falsos positivos. Entretanto, muitas saídas são descartadas, o que contribui para um aumento na taxa de falsos negativos. Uma possível solução para este problema foi apresentada em um trabalho prévio [2]. Neste caso, quando não há uma saída consistente, a localização do ponto mais provável é estimada com base na geometria da face. Entretanto se um ponto considerado consistente estiver errado, a estimativa será errada, o que pode causar um aumento na taxa falsos positivos. Neste trabalho, apresentamos uma forma de avaliar a qualidade do modelo geométrico utilizado para estimar os pontos faltantes. Na próxima seção, descrevemos as restrições geométricas utilizadas para estimar os pontos faltantes, bem como a forma de avaliar a consistência do modelo geométrico obtido.

IV. ESTIMAÇÃO DOS PONTOS FALTANTES UTILIZANDO RESTRIÇÕES GEOMÉTRICAS

Neste trabalho, as características de interesse são os quatro cantos dos olhos e as duas pupilas. Considerando que as

faces estão suficientemente longe da câmera, é razoável supor que estes pontos de interesse estão em um plano. Logo, a correspondência destes pontos em dois quadros diferentes pode ser descrita por uma homografia 2D \mathbf{H} [10].

Para obter a homografia \mathbf{H} , supomos ainda que entre dois quadros a face é transladada por $\mathbf{t} = (t_x, t_y)$ e rotacionada por θ em torno de um eixo paralelo ao eixo principal da câmera. Além disso, assumimos uma rotação em torno do eixo ortogonal ao plano horizontal da cena 3D. Esta última pode ser modelada como um escalamento s ao longo da direção horizontal ao plano da câmera.

Usando este modelo, a transformação desejada possui quatro graus de liberdade. Como cada correspondência de pontos entre dois quadros adiciona duas restrições na homografia, precisamos de duas correspondências de pontos para determinar \mathbf{H} . Assumindo que os quatro cantos pertencem a um corpo rígido, podemos utilizar estes pontos para obter a homografia desejada. Logo, poderemos estimar pontos faltantes do quadro atual se no mínimo dois cantos forem consistentemente detectados no quadro atual e todos os cantos forem consistentemente detectados em um quadro anterior.

A Figura 3 ilustra os parâmetros \mathbf{t} , θ e s da homografia \mathbf{H} . O ponto \mathbf{x}_i é um ponto consistentemente detectado em um quadro anterior e o ponto \mathbf{x}'_i é o seu correspondente no quadro atual.

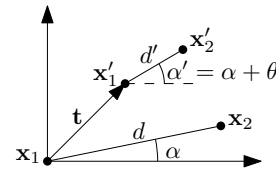


Fig. 3. No modelo utilizado, a relação entre os cantos dos olhos de dois quadros diferentes é dada por uma homografia \mathbf{H} . Colocando a referência em \mathbf{x}_1 , os parâmetros da homografia são $\mathbf{t} = \mathbf{x}'_1$, $\theta = \alpha' - \alpha$ e $s = d'/d$.

Após obter os parâmetros de acordo com a geometria ilustrada na Figura 3, um ponto faltante pode ser estimado usando a seguinte expressão:

$$\mathbf{x}'_i = \mathbf{H}_r(\theta)\mathbf{H}_r(\alpha)\mathbf{H}_s\mathbf{H}_r(-\alpha)\mathbf{x}_i + \mathbf{t}, \quad (2)$$

onde

$$\mathbf{H}_r(\gamma) = \begin{bmatrix} \cos \gamma & -\sin \gamma \\ \sin \gamma & \cos \gamma \end{bmatrix}, \quad \mathbf{H}_s = \begin{bmatrix} s & 0 \\ 0 & 1 \end{bmatrix}. \quad (3)$$

Quando houver mais de duas correspondências, cada par é usado para computar uma homografia diferente. Cada uma das homografias fornece uma estimativa parcial do ponto faltante. Então, a estimativa final é obtida a partir da média das estimativas parciais.

Como as pupilas podem se mover de maneira independente dos cantos dos olhos, nós utilizamos um modelo geométrico diferente para as pupilas. Este segundo modelo está ilustrado na Figura 4, onde a pupila direita é representada pela cruz e a esquerda pelo quadrado. As coordenadas (δ_x, δ_y) da pupila direita em relação ao canto externo do olho direito são as mesmas da pupila esquerda em relação ao canto interno do olho esquerdo. Então, podemos estimar a posição de uma pupila faltante no quadro atual se forem conhecidas: (i) a posição dos cantos dos olhos; (ii) a posição de uma pupila no quadro atual.

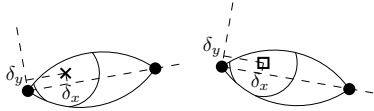


Fig. 4. Modelo geométrico para a localização das pupilas.

A. Avaliação da consistência do modelo geométrico

Uma vez que temos os quatro cantos detectados corretamente em um quadro anterior e no quadro atual, podemos obter seis homografias, uma para cada combinação de duas correspondências. Com cada uma das homografias podemos mapear pontos do modelo \mathbf{x}_i , que correspondem a pontos consistentes no quadro atual \mathbf{x}'_i , e obter uma estimativa $\hat{\mathbf{x}}_i$. Como as homografias não são exatamente iguais, cada ponto estimado $\hat{\mathbf{x}}_i$ é diferente. Então, a consistência do modelo pode ser avaliada da seguinte forma:

- 1) Para cada homografia \mathbf{H}_n , $n = \{1, \dots, 6\}$ obtenha uma estimativa $\hat{\mathbf{x}}_{i,n} = \mathbf{H}_n \mathbf{x}_i$;
- 2) Para cada estimativa, obtenha o erro $e_{i,n} = \|\hat{\mathbf{x}}_{i,n} - \mathbf{x}'_i\|$;
- 3) Calcule o erro global usando $e_H = \sum_i \sum_{n=1}^6 e_{i,n}$;
- 4) Se o erro obtido for menor que qualquer erro anterior, então faça $\mathbf{x}_i = \mathbf{x}'_i$ para atualizar o modelo.

O modelo obtido no algoritmo acima é capaz de se adaptar à geometria da face do indivíduo, o que produz uma estimativa melhor do que em [2]. Quando ocorrem pontos faltantes, podemos utilizar o melhor modelo obtido até o momento para obter as estimativas. Entretanto, é possível que um ponto inconsistente passe pela consistência temporal. Neste caso, mesmo utilizando o melhor modelo possível ($e_H = 0$), a estimativa será errada. Se tivermos três ou quatro pontos faltantes, não há o que fazer, pois não é possível obter a transformação. Com dois pontos faltantes, teremos apenas uma homografia. Com um ponto faltante, teremos três homografias e, portanto, três estimativas. Neste caso, uma estratégia simples é usada para manter a qualidade da estimativa: não use as estimativas caso a variância da posição ultrapasse um dado limiar T_H . Avaliamos empiricamente valores no intervalo de 1 a 5. O valor que apresentou as melhores taxas de acerto na maioria das sequências foi $T_H = 3$. Este valor foi usado para gerar os resultados apresentados na Seção V.

V. PROCEDIMENTOS EXPERIMENTAIS E RESULTADOS

Na Subseção V-A, descrevemos a base de dados utilizada. A métrica utilizada para avaliar o desempenho do nosso método está descrito na Subseção V-B. Na Subseção V-C, apresentamos os resultados obtidos e comparamos com os apresentados em [2]. Além disso, é feita uma discussão acerca destes resultados.

A. Base de dados

Em nossos experimentos, usamos cinco sequências de vídeoconferência de alta definição (1080p) com 300 quadros cada. As sequências possuem quatro indivíduos com diferentes cores de pele, diferentes tipos de plano de fundo, movimento natural da face e oclusão. Além disso, possuem artefatos de compressão. Nós agrupamos as sequências em três classes: “fácil”, “intermediário” e “difícil”. A classe “fácil”, composta por uma sequência, apresenta pouco movimento do indivíduo

e não tem oclusão. A classe “intermediário” possui uma sequência com quantidade moderada de movimento, borramento e também não possui oclusão. Já a classe “difícil”, com três sequências, possui movimentos rápidos, muito borramento, e oclusão parcial ou total da face por uma das mãos. Em cada sequência de vídeo, anotamos manualmente 13 pontos fiduciais faciais (que incluem os cantos dos olhos e as pupilas). As sequências e as anotações manuais estão disponíveis em [11].

Para tratar dos problemas de iluminação da cena e de escala, utilizamos o esquema de pré-processamento proposto por [3]. Este esquema é adequado às nossas sequências porque normaliza o tamanho das faces localizadas pelo algoritmo de Viola-Jones [12] e a iluminação [13].

B. Avaliação

Para medir o desempenho do nosso método, aplicamos uma medida de exatidão baseada na que foi proposta em [14]. A medida consiste na distância entre o rótulo automático e a marcação manual correspondente normalizada pela distância interocular:

$$e = \frac{\|\mathbf{l}_a - \mathbf{l}_m\|}{d_{\text{eyes}}}, \quad (4)$$

onde \mathbf{l}_a é a coordenada do rótulo automático dado pelo método proposto, \mathbf{l}_m é a coordenada da referência (anotação manual) e d_{eyes} é a distância entre as pupilas, também obtida das referências.

Para cada característica facial de cada sequência, nós comparamos a taxa de acerto contra um percentual da distância interocular em que consideramos uma detecção correta. Plotamos a taxa de acerto considerando dois tipos de erro: (i) somente os casos no qual o algoritmo fornece uma saída, levando em conta somente os erros Falsos Positivos (FP), e (ii) todos os casos, incluindo aqueles no qual o algoritmo não fornece um ponto de saída, levando em conta os erros FP e os Falsos Negativos (FN).

C. Resultados e discussões

Devido a restrições de espaço e similaridade encontrada nos resultados de pontos simétricos, serão apresentados apenas os resultados para três pontos de duas sequências, uma do grupo “fácil” (seq18.yuv) e outra do grupo “intermediário” (seq14.yuv). As curvas restantes podem ser visualizadas em [11].

Na Figura 5, os resultados para a sequência do grupo “fácil” estão na esquerda e os resultados para a sequência “intermediária” estão na direita. Para cada ponto fiducial, comparamos o método proposto com o descrito em [2]. Temos duas curvas de acerto em cada caso, uma considerando apenas os falsos FPs como erro e outra considerando também os FNs. Como pode ser visto, o método proposto apresentou melhora em todos os casos apresentados, exceto para o canto interno do olho esquerdo na sequência intermediária no caso em que somente FPs são considerados, para o qual seu desempenho foi apenas cerca de 5% inferior ao método proposto em [2]. Considerando todos os testes realizados (ver [10]), o método proposto obteve, em geral, uma significativa redução tanto no número de FPs quanto de FNs, apresentando uma melhora global em relação a [1].

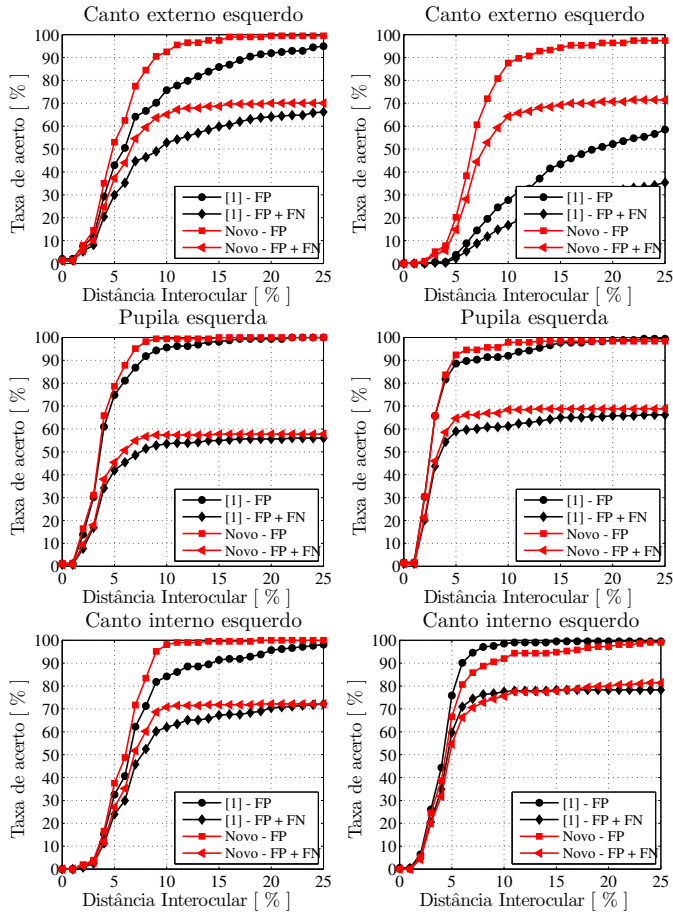


Fig. 5. Esquerda: resultados para uma sequência fácil (seq18.yuv). Direita: resultados para a sequência intermediária (seq14.yuv). As linhas pretas representam os resultados obtidos em [2] enquanto as vermelhas representam os resultados deste trabalho. Nas linhas cheias consideramos apenas os falsos positivos como erro e nas linhas tracejadas consideramos também os falsos negativos.

Os resultados para as sequências difíceis podem ser vistos em [11]. Nesses resultados, temos 18 pontos em 3 sequências (6 em cada uma). Desses pontos, 11 apresentam resultados inconclusivos, visto que a taxa de acerto, considerando FP e FN como erros, é menor que 50% tanto para o método proposto quanto para o descrito em [2]. Para os 7 pontos restantes, o método proposto apresenta um resultado melhor em 4 deles. Isso ocorre porque temos um grande número de quadros com detecções inconsistentes nestes casos, ou seja, o uso da consistência geométrica só traz benefícios em poucos quadros. Por outro lado, as sequências fáceis e intermediárias são mais parecidas com a maioria dos casos práticos. Além disso, os principais métodos conhecidos também tendem a falhar em nossas sequências difíceis (que possuem borramento, movimentos rápidos e oclusão por uma das mãos).

VI. CONCLUSÕES

Neste artigo descreve-se um sistema para a detecção e rastreamento de seis pontos fiduciais na região dos olhos. A principal contribuição deste trabalho é um novo método de consistência geométrica com avaliação da consistência do modelo empregado. Através da técnica proposta é possível reduzir a taxa de falsos negativos recuperando os pontos

perdidos com acurácia razoável. Estimativas errôneas podem ser evitadas avaliando a variância da posição como medida de confiabilidade, tornando-o robustos a erros. Além disso, a baixa complexidade viabiliza o uso em tempo real. O sistema foi implementado em C++ utilizando a biblioteca OpenCV.

Os resultados mostram que o método proposto possui um bom desempenho para as sequências dos grupos fácil e intermediário. Entretanto, ele precisa ser melhorado para lidar com as sequências difíceis, que possuem borramento, movimentos rápidos e oclusão. É importante notar que o método apresentado em [2] com as melhorias descritas neste artigo pode ser utilizado com muitos outros tipos de rastreadores atualmente disponíveis. Além disso, pode ser utilizado com outros tipos de detectores, desde que eles possam ser adaptados para fornecer uma nuvem de pontos na saída ao invés de um único ponto. Trabalhos futuros incluem a aplicação deste método com outros detectores e utilizando outras técnicas de rastreamento, como o filtro de partículas [15] ou o que está em [16].

REFERÊNCIAS

- [1] A. Bleicher, "Rise of eye phones," *Spectrum, IEEE*, vol. 50, pp. 9–10, 2013.
- [2] Gabriel M. Araujo, Eduardo A. B. Silva, Alexandre G. Ciancio, José F. L. de Oliveira, Felipe M. L. Ribeiro, and Amir Said, "Integration of eye detection and tracking in videoconference sequences using temporal consistency and geometrical constraints," in *IEEE International Conference on Image Processing (ICIP)*, 2012.
- [3] Gabriel Araujo, Waldir Sabino, Eduardo da Silva, and Siome Goldenstein, "Facial landmarks detection based on correlation filters," in *International Telecommunications Symposium (ITS)*, 2010.
- [4] Bruce D. Lucas and Takeo Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th international joint conference on Artificial intelligence - Volume 2*, San Francisco, CA, USA, 1981, pp. 674–679, Morgan Kaufmann Publishers Inc.
- [5] B. V. K. Vijaya Kumar, Abhijit Mahalanobis, and Richard D. Juday, *Correlation Pattern Recognition*, Cambridge University Press, New York, NY, USA, 2005.
- [6] Robert E. Schapire, "The strength of weak learnability," *Machine Learning*, vol. 5, no. 2, pp. 197–227, June 1990.
- [7] BioID Technology Research, "The bioid face database," <http://www.bioid.com>, 2001.
- [8] Felipe Moreira Lopes Ribeiro, Gabriel Matos Araujo, Eduardo Antônio Barros da Silva, Jose Fernando Leite de Oliveira, and Siome Klein Goldenstein, "Detecção de pontos fiduciais sobre a face em tempo real," in *XXX Simpósio Brasileiro de Telecomunicações (SBRT)*, 2012.
- [9] Trevor Hastie, Robert Tibshirani, and Jerome Friedman, *The Elements of Statistical Learning*, Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.
- [10] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, second edition, 2004.
- [11] Eye Tracking Research, "Robust eye tracking in high definition videoconference sequences," <http://www.lps.ufrj.br/~biometria/etr/>, 2012.
- [12] Paul A. Viola and Michael J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.
- [13] Xiaoyang Tan and Bill Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," in *Analysis and Modelling of Faces and Gestures*, oct 2007, vol. 4778 of LNCS, pp. 168–182, Springer.
- [14] Oliver Jesorsky, Klaus J. Kirchberg, and Robert Frischholz, "Robust Face Detection Using the Hausdorff Distance," in *AVBPA '01: Proceedings of the Third International Conference on Audio- and Video-Based Biometric Person Authentication*, London, UK, 2001, pp. 90–95, Springer-Verlag.
- [15] Emilio Maggio and Andrea Cavallaro, *Video tracking: theory and practice*, Wiley, 2011.
- [16] Z Kalal, J Matas, and K Mikolajczyk, "Online learning of robust object detectors during unstable tracking," *On-line Learning for Computer Vision Workshop*, 2009.