

Reconhecimento de Propriedade de Componente de Imagem em Jogos

Jorge Luiz dos Santos Ramos Junior, Renan Balbinot

Curso de Bacharelado em Sistemas de Informação – Universidade Tecnológica Federal do Paraná (UTFPR) – Campus de Curitiba
80230-901 – Curitiba – PR– Brasil

tetris.attack@hotmail.com, renandvp@gmail.com

Abstract. *Entertainment games usually utilize mathematical variables in their interactive processes. This is true for the computational golf game “PangYa” in which, among the many involved variables, one of them is wind, which is represented by an arrow that points to a certain angle. Our goal with this research is to develop an algorithm that identifies that angle, given any valid PangYa gameplay screenshot.*

Resumo. *Jogos de entretenimento, usualmente, utilizam variáveis matemáticas em seu processo de interação. No jogo computacional de golfe “PangYa”, esse contexto está presente e, entre as muitas variáveis envolvidas, uma delas é o vento, o qual é representado por uma seta que aponta para um certo ângulo. O objetivo desta pesquisa é desenvolver um algoritmo que identifique esse ângulo a partir de uma imagem de PangYa válida qualquer.*

1. Introdução

Em uma era em que a tecnologia para obtenção e armazenamento de imagens e vídeos aprimora-se continuamente, também há o aumento do interesse em como reconhecer e extrair propriedades desses materiais. Entre os problemas que podem ser resolvidos através de processamento de imagens, estão a medição de energia liberada por descargas de corona [Prasad and Reddy, 2017] e o monitoramento de corais submarinos [Osterloff, Nilssen and Nattkemper, 2016].

O processo pode ser simplificado se existe a possibilidade de tornar constantes algumas condições sob as quais as imagens são obtidas [Gonzalez and Woods, 2008]. No jogo computacional de golfe *PangYa*, podemos restringir a área de interesse e manipular as imagens de saída de modo a facilitar a identificação das propriedades relevantes.

Em *PangYa*, devido ao fato de as variáveis de ambiente permanecerem constantes durante o planejamento e execução de uma tacada, existe a possibilidade de calcular a trajetória da bola com precisão suficiente para acertar o buraco a partir de uma longa distância com grande consistência, possibilitando pontuações muito superiores comparadas às de um jogador que não calcula. Uma dessas variáveis de ambiente é o vento, o qual é representado por uma seta que aponta para um certo ângulo. O objetivo desta pesquisa é que o algoritmo desenvolvido reconheça um ângulo de vento a partir de

qualquer imagem válida. Uma imagem válida é qualquer captura de tela extraída durante a jogabilidade, sendo consideradas algumas restrições.



Figura 1. Captura de tela da jogabilidade de *PangYa*

O algoritmo foi desenvolvido na linguagem *Python* utilizando a biblioteca *OpenCV* para processamento de imagens, e trabalha em sete passos: isolar a parte da imagem que contém a seta representando o vento, converter para o espaço de cores CIELAB, binarizar, eliminar possíveis ruídos com uma máscara, suavizar a seta usando morfologia matemática, obter o contorno e, por fim, traçar a elipse de melhor ajuste para ele. A inclinação da elipse é o ângulo do vento.

Apesar de *PangYa* existir desde 2004 e sua comunidade internacional de jogadores discutir métodos de cálculo de trajetória da tacada com frequência, ainda não há nenhum método ou programa que permita obter o ângulo do vento com ambas rapidez e precisão aceitáveis.

Este artigo está dividido em quatro seções. A primeira contém uma breve introdução sobre o projeto e apresenta o contexto relevante para o entendimento e desenvolvimento do mesmo; a segunda apresenta considerações metodológicas e os cronogramas proposto e executado; apresenta a fundamentação teórica utilizada no algoritmo; a terceira apresenta exemplos de imagens e discorre detalhadamente sobre como é efetuado o processamento; e, por fim, uma seção apresenta a conclusão.

2. Metodologia

2.1 Fontes utilizadas

Este trabalho é fundamentado, principalmente, no conteúdo da obra *Digital Image Processing* [Gonzales and Woods, 2008]. Também pesquisamos por referências nos repositórios acadêmicos *IEEE Xplore* e *ACM Digital Library*, mas nada encontrado foi diretamente utilizado para o resultado final.

Até onde conseguimos pesquisar, este é o primeiro trabalho acadêmico relacionado a *PangYa* - uma busca pela palavra-chave *pangya* em tais repositórios não retornou nenhum resultado. Além disso, pesquisamos por possíveis trabalhos relacionados utilizando combinações das palavras-chave *image processing*, *angle*, *orientation*, *vector*, *edge detection*, *eigenvectors*, *hough transform* mas os artigos mais próximos, em geral, tratavam de análise de objetos em movimento em vídeos, ou tratavam de problemas de natureza diferente e utilizaram técnicas que julgamos não serem adequadas ao nosso contexto.

2.2 Cronograma



Figura 2. Cronograma planejado e cronograma executado

A principal discrepância entre o cronograma planejado e cronograma executado ocorreu na fase de extração de características, devido à tentativa de uso de várias abordagens que acabaram por se revelar não efetivas, possivelmente por falhas dos autores deste artigo.

Discorrer sobre as tentativas com tais abordagens exigiria outro artigo, porém vamos listar algumas que consideramos possuir maior potencial para futuras pesquisas: autovalores e autovetores, detecção da ponta da seta através de análise de padrões de cor, detecção de bordas + detecção de linhas.

3. Fundamentação Teórica

Uma imagem digital é composta por um conjunto de *pixels* que podem ser definidos por suas coordenadas (x,y) e intensidade.

3.1. Espaços de cores

Em imagens coloridas, a intensidade é representada por uma coordenada em um espaço de cores que geralmente possui três dimensões. Para imagens digitais, cada uma dessas dimensões é representada como um canal, cujo tamanho das dimensões horizontal e vertical é o mesmo da imagem e cujo conteúdo é a localização de cada pixel na dimensão [Gonzalez and Woods, 2008].

Apesar de o espaço RGB ser o mais popular para armazenamento de imagens digitais, utilizar espaços de cor diferentes pode ajudar a extrair características em imagens da forma como são percebidas pelo olho humano [Fairchild, 2013].

O algoritmo, após converter a imagem para o espaço de cores CIELAB, extrai o canal *b* e trata a imagem como qualquer outra imagem contendo apenas um canal.

3.2. Binarização

Binarização é o processo de dividir os *pixels* de uma imagem em duas classes, sendo elas geralmente “objetos de interesse” e “resto”. Um método popular para efetuar binarização consiste em comparar o valor de intensidade de cada *pixel* com um limiar, que pode ser ou não definido dinamicamente [Gonzalez and Woods, 2008].

Em nosso algoritmo, utilizamos o algoritmo de Otsu para encontrar o limiar para binarização. O algoritmo tem o objetivo de, considerando uma imagem com apenas um canal, definir automaticamente um limiar para binarização.

Segundo Otsu, o melhor limiar é aquele que, separando os pixels em duas classes, maximiza a variância interclasse. [Otsu, 1979] Em outras palavras, ele procura identificar as duas faixas de intensidades que ocorrem com maior frequência e escolher o ponto que melhor divide as duas.

3.3. Morfologia matemática

No contexto de processamento de imagens, morfologia matemática refere-se a um conjunto de técnicas de processamento baseadas na teoria dos conjuntos. [Gonzalez and Woods, 2008] Duas destas técnicas são a dilatação e a erosão que, em sua forma mais básica, consistem em “engordar” e “afinar” os pixels brancos da imagem, respectivamente.

O algoritmo utiliza uma técnica chamada “abertura”, que consiste em efetuar uma erosão seguida de uma dilatação. Esta técnica tem o resultado de remover pequenos ruídos e suavizar o formato da imagem.

3.4. Ângulo a partir de um triângulo

A partir da aplicação da lei dos cossenos, podemos encontrar qualquer ângulo de um triângulo se sabemos a medida de seus três lados [Maor, 1998].

Setas do vento em *PangYa* podem ser representadas como triângulos, segundo a representação a seguir:



Figura 3. Representação triangular de vento em *PangYa*

Apesar de este método não ser utilizado no algoritmo, ele foi utilizado para obter estimativas com as quais os resultados do algoritmo podem ser comparados. Tais estimativas foram calculadas manualmente.

3.5. Elipse a partir de contorno

Para traçar a elipse de melhor ajuste ao contorno, utilizamos a função *fitEllipse* da biblioteca *OpenCV*. Esta função utiliza o método explicado em *Direct Least Squares Fitting of Ellipses* [Fitzgibbon et al, 1996].

4. Processamento

O processo a seguir considera que a entrada é uma imagem válida. É importante notar que, sendo um jogo computacional onde o jogador possui controle da câmera, é possível manipular o formato da seta até um certo ponto. A imagem a seguir apresenta formas diferentes de representação do mesmo vento, dependendo do estado da câmera.



Figura 4. Representações diferentes do mesmo vento

A representação *a* não é considerada válida para os propósitos deste algoritmo, mas isto não é um problema, pois sempre é possível manipular a câmera para obter representações parecidas com as representações *b* ou *c*.

4.1. Isolamento da região contendo a seta

As capturas de tela em *PangYa* podem ter diferentes tamanhos, dependendo da resolução em que o jogo está sendo executado. Porém, a seta do vento está sempre na mesma posição relativa ao canto inferior direito e é invariante à resolução, portanto podemos extrair um quadrado de dimensões 68x68 *pixels* a partir das coordenadas (*width*-83, *height*-91). Vamos utilizar a captura a seguir para demonstrar o funcionamento do algoritmo.



Figura 5. Figura realçando em laranja a região a ser extraída

4.2. Conversão para o espaço de cores CIELAB

Esta conversão prepara a imagem para ser binarizada no passo a seguir. Todos os canais de todos os espaços de cores disponíveis no *OpenCV* foram testados, e o que conseguiu delimitar com melhor precisão a área da seta no conjunto de testes foi o canal *b* do espaço CIELAB, como indicado na figura a seguir. Uma abordagem “naive” (extrair o canal B do espaço RGB) é mostrada para fins de comparação.



Figura 6. Canal *b* do espaço CIELAB comparado com canal B do espaço RGB

4.3. Binarização

Neste passo é simplesmente aplicada binarização em cima do resultado anterior, usando limiar obtido através do algoritmo de Otsu.

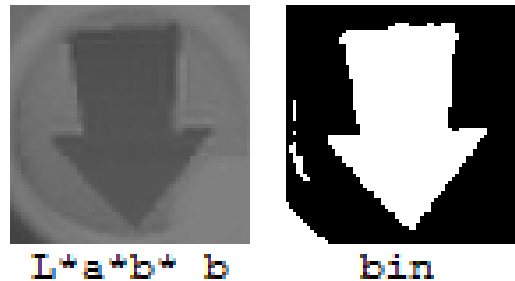


Figura 7. Resultado da binarização usando Otsu

Dependendo da imagem de entrada, ainda vai existir uma quantidade considerável de ruído neste resultado, como neste caso exemplo.

4.4. Aplicação de máscara

Neste passo é aplicada uma máscara com o objetivo de eliminar ruídos que não fazem parte da seta. A máscara é uma imagem pré-determinada que funciona como um simples operador *and*: pixels que não forem brancos nas duas imagens são definidos como 0 na imagem de saída.

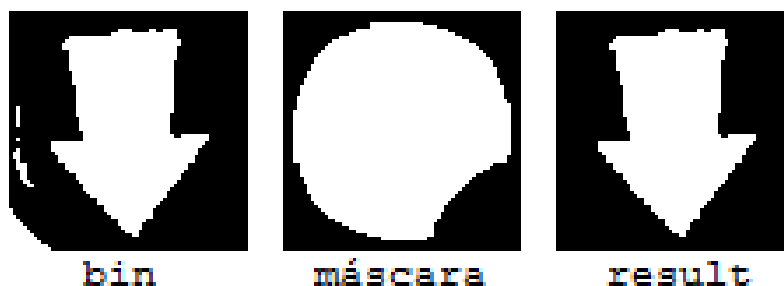


Figura 8. Resultado da aplicação de máscara

4.5. Suavização da seta

Bordas da seta irregulares podem influenciar na definição do contorno, causando mais imprecisão nos resultados. Reduzimos os acidentes no contorno através do uso da abertura morfológica.



Figura 9. Resultado da aplicação de abertura morfológica

4.6. Obtenção do contorno

Os *pixels* de contorno são detectados através da análise de vizinhança-4 de cada *pixel*. Se pelo menos um *pixel* da vizinhança é 0, é um *pixel* de contorno.



Figura 10. Demonstração do contorno sobre o corte original

4.7. Traçado da elipse de melhor ajuste

Determinado o contorno, utilizamos a função *fitEllipse* da biblioteca *OpenCV*. Convenientemente, um dos retornos desta função é o ângulo de rotação, que é o resultado que buscamos.



Figura 11. Demonstração da elipse sobre o corte original

A inclinação desta elipse é de 3.4 graus, enquanto o ângulo estimado pelo método triangular (explicado na seção 3.4) foi 3.9 graus. Um ótimo resultado!

5. Análise dos resultados e conclusão

O algoritmo obteve ótimos resultados em geral, obtendo uma diferença média com relação ao método triangular de 1.1 grau em 28 imagens escolhidas aleatoriamente. Isto inclui casos em que a seta está coberta pelo medidor de velocidade do vento no canto inferior direito.

O pior resultado, considerando a diferença da estimativa do algoritmo para a estimativa do método triangular, foi uma diferença de 5.7 graus. A imagem e os resultados de seu processamento são apresentados a seguir.

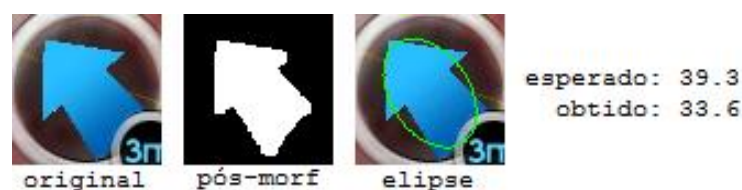


Figura 12. Pior caso de testes para o algoritmo

Uma análise dos casos de diferença de 1.5 grau ou mais mostra que os piores casos para o algoritmo ocorrem quando a seta aponta para a região noroeste, pois frequentemente a parte da seta ocultada pelo canto inferior direito acaba por deixar uma das metades da seta consideravelmente mais “pesada”. Isto resulta em uma deformação na elipse, o que influencia o resultado final.

Os casos de seta apontando para a região sudeste (tendo sua ponta ocultada pelo medidor de vento) apresentaram bons resultados, como estes:



Figura 13. Casos de teste com ponta da seta coberta

A área da seta ocultada nesses casos é relativamente pequena, o que resulta em pouca perda de precisão.

Em geral, o algoritmo obteve resultados acima de nossas expectativas, especialmente para os casos de seta apontando para o sudeste. Trabalhos futuros devem focar em técnicas que melhorem a precisão para os piores casos deste algoritmo.

Referências

- Gonzalez, R. C. and Woods, R. E. Digital Image Processing, Pearson, 3rd edition, 2008.
- Jain, A. K. Fundamentals of Digital Image Processing, Pearson, 1st edition, 1988.
- Fairchild, M. D. Color Appearance Models, Wiley, 3rd edition, 2013.
- Maor, E. Trigonometric Delights, Princeton, 1st edition, 1998.
- Fitzgibbon, A. W., Pilu, M., Fisher, R. B. “Direct Least Squares Fitting of Ellipses”. Disponível: <http://cseweb.ucsd.edu/~mdailey/Face-Coord/ellipse-specific-fitting.pdf>, 1996.
- Otsu, N. “A Threshold Selection Method from Gray-Level Histograms”. Disponível: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4310076>, 1979.
- Osterloff, J., Nilssen, I., Nattkemper, T. W. “Computational Coral Feature Monitoring for the fixed underwater Observatory LoVe”. Disponível: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7761417>, 2016.
- Prasad, D. S., Reddy, B. S. “Digital Image Processing Techniques for Estimating Power Released from the Corona Discharges”. Disponível: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7873462>, Fevereiro/2017.