

ReTiLiSFis - Real Time Light Saber Fight Simulator

Daniel Pacheco de Queiroz e Eduardo Diego Silva
Universidade Federal de Minas Gerais - UFMG
Av. Antônio Carlos, 6627 - Pampulha - Belo Horizonte - MG
{pacheco, edudiego}@dcc.ufmg.br

Resumo

Técnicas de efeitos especiais são cada vez mais empregadas em filmes, mas estes efeitos são raramente utilizados em tempo real, ou seja, no momento em que as cenas são filmadas. Com este trabalho trataremos o problema de criação de efeitos especiais em tempo real, utilizando como tema um efeito especial clássico dos filmes de ficção científica: os sabres de luz.

1. Introdução

Efeitos especiais estão cada vez mais comuns nos filmes atuais, de tão corriqueiros e também por serem bem feitos, muitas vezes não são percebidos. Entretanto, houve uma época na qual estes efeitos eram coisas espantosas e mesmo não sendo primorosamente perfeitos ou tenham empregado grande tecnologia em seu desenvolvimento, causavam grande frisson quando apareciam na tela.

Um dos filmes pioneiros no desenvolvimento de efeitos especiais foi Star Wars (Guerra nas Estrelas) de George Lucas. O primeiro filme, A Ameaça Fantasma, foi lançado em 1977 e era o quarto filme de uma série de seis, que só foi completamente produzida em 2005. Apesar das limitações tecnológicas da época, já no primeiro filme foi possível os fãs experimentarem os efeitos especiais que produziram os primeiros light sabers (sabres de luz) da história, e que se tornaram um dos elementos visuais que mais marcaram os filmes da série.

Este sabre é uma espada utilizada pelos Jedi (guerreiros do bem presentes nos filmes) e pelos Sith (guerreiros do mal), que inicialmente consiste em apenas um punho metálico, mas quando ativada passa a ser uma espada cuja lâmina é feita de luz, sendo extremamente cortante e poderosa.

Este trabalho consiste em criar um simulador de lutas de sabre de luz em tempo real. Diferentemente dos filmes, nos quais as cenas são gravadas com atores utilizando bastões

de cores conhecidas e, posteriormente, são utilizadas técnicas como cromma key para substituir a imagem real pela imagem brilhante do sabre; nosso trabalho visa criar estes efeitos no momento em que a cena é filmada. Não só os efeitos visuais serão gerados, com este trabalho também pretendemos gerar os efeitos sonoros característicos dos sabres, que irão interagir com a cena.

Para tanto serão necessárias técnicas de rastreamento de objetos, substituição da imagem real por uma imagem gerada artificialmente e inserção de sons baseados em características das imagens. Pretende-se com este artigo explicitar os métodos utilizados para a criação deste aplicativo.

2. Referencial Teórico

Um dos principais problemas que devem ser resolvidos para a construção deste aplicativo é o rastreamento eficiente dos objetos de cor conhecida no mundo real. Portanto, foram pesquisados, principalmente, artigos nesta área.

Em [5] é proposto um método para encontrar centros de figuras geométricas de uma mesma cor, o que poderá ser útil no nosso trabalho, pois precisamos encontrar o centro do sabre (que deverá ter uma cor definida na cena real), para podermos inserir a computação gráfica sobre o real.

O método proposto consiste em percorrer a imagem e, ao detectar um pixel da cor especificada, começa a traçar segmentos internos para os pixels da mesma cor na horizontal e na vertical, sendo que o segmento vertical cruza o meio do horizontal. Para percorrer a imagem de maneira eficiente, o sistema salta alguns pixels. Esse processo de gerar segmentos na horizontal e na vertical é chamado de processamento em cruz.

Esse processo em cruz é feito diversas vezes para cada objeto, visto que percorrer a imagem de saltos em saltos permite que esta varredura de pixel caia em vários lugares de um mesmo objeto. Cada processamento em cruz gera um centro provisório do objeto, vários centros provisórios permitem encontrar o centro estimado mais provável do objeto com maior precisão.

Entre os pontos fortes deste método estão a não dependência das imagens anteriores, como ocorre em algumas formas de rastreamento, sua rapidez, gastando aproximadamente 25ms para o processamento, o que permitiria encontrar em tempo real os centros em todas as imagens de um filme de 30 quadros por segundo (33ms de permanência de cada quadro). O outro ponto forte é que ele funciona bem mesmo em imagens degradadas. O ponto fraco é que ele não encontra o centro exato, mas apenas um centro aproximado.

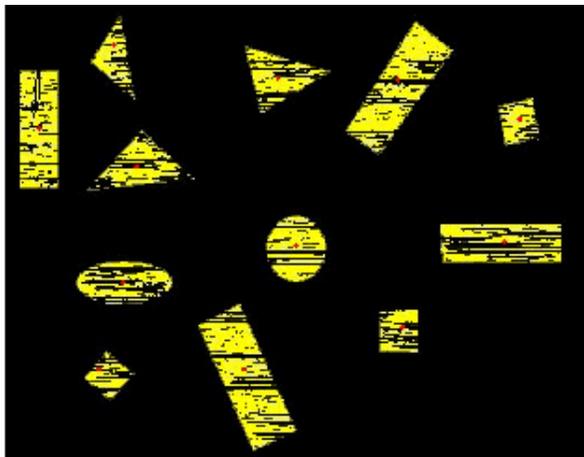


Figura 1. Exemplo do resultado do algoritmo (com ruído).

O BraMBLe [3] é um conhecido rastreador de blobs (objetos com tamanho e cor conhecidos), baseado na teoria matemática da correlação Bayesiana. Foi a primeira implementação com a qual o número de objetos rastreados pode variar durante o processo de rastreamento. Além disso, esse rastreador se apresentou bastante robusto nos testes, mantendo a identificação de objetos mesmo se cruzando, embora falhe em alguns casos. Outro ponto positivo do método é sua baixa utilização da capacidade total de processamento de um computador pessoal.

O algoritmo para rastreamento de mundo fechado em tempo real [7] usa um sistema de informação contextual para rastrear objetos não-rígidos dentro de um limite determinado. Ele usa uma tabela de características que é formada pela tentativa de casamento de cada objeto da cena com uma área da imagem para definir qual é o melhor casamento que contempla todos os objetos ao mesmo tempo. Essas características incluem cor, velocidade e direção dos objetos, bem como o comportamento anterior desse objeto. Eles usam como modelo um quarto de brinquedos de crianças filmado por cima e tentam acompanhar o movimento das crianças e dos objetos no quarto, incluindo entradas e

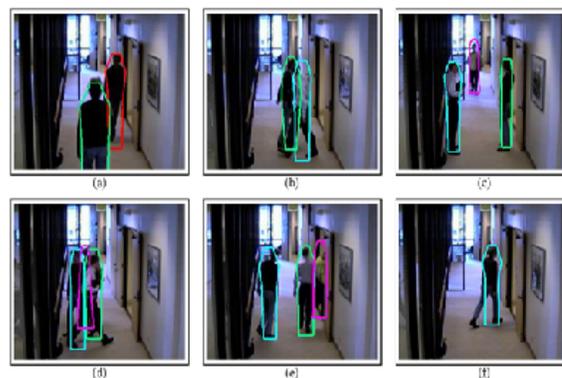


Figura 2. (a) Pessoa A e pessoa B entraram na cena. (b) BraMBLe inferiu corretamente que a pessoa B saiu em direção à câmera, enquanto C entrou pela porta lateral e está andando em frente de A. (c) BraMBLe rastreou A e C apesar da troca de posições, enquanto D se aproximou pelo fim do corredor. (d) A e C trocaram de posição enquanto conversavam com D; note que o BraMBLe trocou incorretamente as identidades de A e C quando as três pessoas ocuparam a mesma região da imagem. (e) BraMBLe inferiu corretamente que D se moveu atrás de C. (f) C e D saíram.

saídas de novos corpos. O casamento é avaliado de forma a minimizar a chance de erros e otimizar a performance do sistema descartando tentativas impossíveis o quanto antes.

Em [6] são revisados um conjunto de abordagens para rastreamento de objetos em tempo real, é apresentada uma nova abordagem baseada em threshold por cor. O artigo explica por que o sistema de cores RGB, comum aos formatos de representação digital, não é apropriado para rastreamento de cores pois é muito sensível à variações de iluminação ambiente. Para esse tipo de aplicação é melhor usar os sistemas de cores HSI ou YUV, onde a o componente da intensidade é tratado em separado da cromaticidade. Como a conversão RGB para HSI é cara em software, eles propõe a conversão do OpenCV [2] de RGB para YUV, que tira proveito do conjunto de instruções MMX dos processadores Intel pra fazer a conversão mais rapidamente. Feita a seleção do objeto a ser rastreado, uma função objetivo dá valores positivos altos ao núcleo da área e negativos baixos à periferia. Isso gera uma máscara de extração do primeiro plano que é aplicada quadro a quadro pelo threshold YUV. Na verdade o sistema equivalente ao YUV no OpenCV é o YCbCr, fato que não é explicitado no artigo, muito provavelmente para não acrescentar informação desnecessária aos leitores.

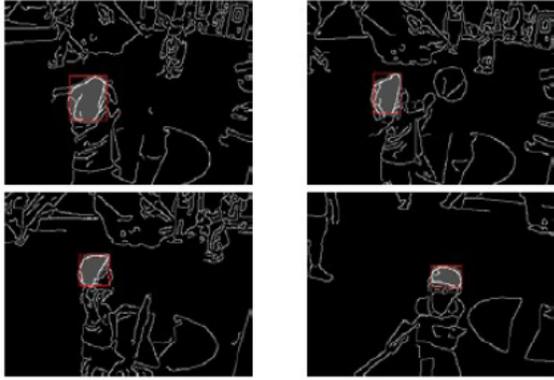


Figura 3. Imagem após a detecção de bordas utilizando Canny e com segmentação por cor.

O método apresentado no artigo se mostrou altamente robusto e eficiente, e está de acordo com a idéia prévia do nosso trabalho de filtrar o objeto por threshold de cor (embora o artigo acrescente métodos para ganho de performance) e será de fundamental importância no desenvolvimento do nosso trabalho.

3. Sistemas de cor

Basicamente, cor é uma parte do espectro eletromagnético cujo comprimento de onda varia de 380 nm a 780 nm. Esta é a parte do espectro a qual os olhos humanos são sensíveis. As cores podem ser descritas de diferentes formas, não somente fisicamente por suas características de comprimento de onda. Um sistema que define como as cores serão apresentadas, modelando as cores reais, é chamado de sistema de cor (também conhecido como espaço de cor).

Como o sistema visual humano ainda não é completamente compreendido, vários modelos de cores têm sido propostos para modelar as características de uma cor. Todos estes modelos são baseados em pelo menos três componentes para descrever uma cor. Em alguns estes componentes são cores primárias e já em outros são características como cromaticidade e luminância, alguns sistemas de cor são:

- **RGB:** O padrão RGB é o formato natural de exibição em monitores. Suas componentes R (vermelho), G (verde) e B (azul) variam de 0 a 255, sendo a tupla (0, 0, 0) a cor preta e a tupla (255, 255, 255) a cor branca. O formato padrão de aquisição em webcams do OpenCV é o formato RGB. O padrão RGB é apropriado para exibição mas pouco consistente quanto à variação de valores e da cor percebida. Uma variação de luminosidade ou saturação no es-

paço RGB varia irregularmente dentro do espaço de cor, sendo necessárias outras formas de representação para capturar as informações de cor mais de acordo com o modelo de visão humana.

- **HSL e HSV:** Tanto HSL quanto HSV descrevem as cores como pontos em um espaço cilíndrico cujo eixo central varia de preto no fundo a branco no topo com cores neutras entre eles, onde o ângulo ao redor do eixo corresponde ao hue (matiz) a distância do eixo corresponde à saturação e a distância no decorrer do eixo corresponde ao valor ou à luminosidade. Como HSL e HSV são transformações lineares sobre o RGB, as cores definidas por (h,s,l) ou (h,s,v) dependem unicamente dos valores de (r, g, b). Ambos os modelos foram definidos formalmente em 1978 por Alvy Ray Smith.
- **XYZ:** O espaço de cores XYZ foi padronizado em 1931 experimentalmente pela CIE, a International Commission on Illumination. O modelo se baseia nos receptores de luz do olho humano, que possuem células cone para detecção de ondas curtas, médias e longas. Assim, a princípio, três parâmetros descrevem a sensação de cor. Os valores tristimulus são as quantidades das 3 cores necessária para passar num teste de cores. Várias fontes de luz de comprimento de onda diferentes podem ser consideradas a mesma cor pelo observador se a sua soma no modelo aditivo resultar no mesmo tristimulus.
- **YCrCb:** É uma família de espaços de cor usada em sistemas de vídeo e fotografia digital. O Y é o componente Luma e os componentes Cb e Cr são a cromaticidade azul e vermelha respectivamente. YCrCb é uma transformação digital do formato YpbPr que por sua vez é derivado do YUV, o formato usado para transmissão de imagens a cores. Muitas vezes quando se fala em YUV no meio digital na verdade se está querendo dizer YCrCb.

YCrCb não é um espaço de cores absoluto, mas apenas uma maneira de codificar a informação RGB real usada para exibição do sinal. Um valor expresso em YCrCb só é compreensível se associado a um padrão de RGB conhecido.

Nossa primeira abordagem adotou o espaço YCrCb, por uma otimização que o OpenCV possui para essa transformação em processadores Intel. No entanto nos demos conta que a filtragem por esse espaço de cores era superada em precisão pela filtragem pelo espaço HSV. Além disso o espaço HSV é o utilizado em diversos softwares de edição de imagens para segmentação, o que conta como bom indicador da sua qualidade.

4. Metodologia

O trabalho será realizado seguindo as etapas de tratamento das imagens definidas a seguir: Primeiramente faremos a conversão do espaço de cor de RGB para HSV, cuja estrutura se assemelha mais ao sistema visual humano e é mais robusta à variação de cor.

Depois disso faremos uma segmentação da imagem baseada na cor do sabre (agora definida pelos 3 parâmetros H, S e V), mantendo na imagem apenas os sabres, que têm uma cor definida.

Após esta etapa, será utilizado um dos métodos de rastreamento pesquisados na revisão bibliográfica ou outro que se adeque melhor à nossa situação. Tendo-se as coordenadas de localização do sabre, será inserido o efeito de cor característico dos sabres de luz. Também através destas coordenadas, serão detectados movimentos mais rápidos dos objetos para que efeitos sonoros possam ser inseridos.

Finalmente, iremos implementar métodos de detecção do choque entre os sabres, para inserção de efeitos sonoros e visuais. Isto deve ser feito através da detecção da interseção dos dois sabres e de variações bruscas na velocidade dos mesmos.

5. Resultados

Todo o método é baseado na API da biblioteca de tratamento de imagens OpenCV. A segmentação HSV foi selecionada após percebermos sua superioridade em termos de qualidade de filtragem com relação ao padrão YCrCb. A filtragem é feita com base em uma imagem derivada da imagem capturada e um threshold com tolerâncias independentes por canal.

Após calibrarmos o sistema para a iluminação ambiente, a imagem selecionada é filtrada para remoção de ruídos. O agrupamento equivalente ao objeto rastreado muito frequentemente apresenta irregularidades, especialmente quando este se encontra em movimento. Para obtermos uma forma mais uniforme a partir dessa filtragem, utilizamos a geração de contorno automática do OpenCV e depois a geração de contorno convexo.

A partir desse contorno convexo é simples detectar se os objetos se encontram em áreas sobrepostas, bastando para isso registrar os pontos dos extremos de cada vetor de pontos e fazer uma comparação entre suas posições: se o ponto mais a esquerda do contorno de um objeto estiver mais a esquerda que o ponto mais a direita do segundo objeto (o que pode ser estendido para todos os extremos opostos dos objetos) podemos detectar a interseção entre eles. Uma interseção acompanhada por uma mudança brusca de velocidade é indicio de impacto entre os objetos.

Para medir a velocidade do objeto, após tê-lo detectado por filtragem de cor, usamos um método que calcula o cen-

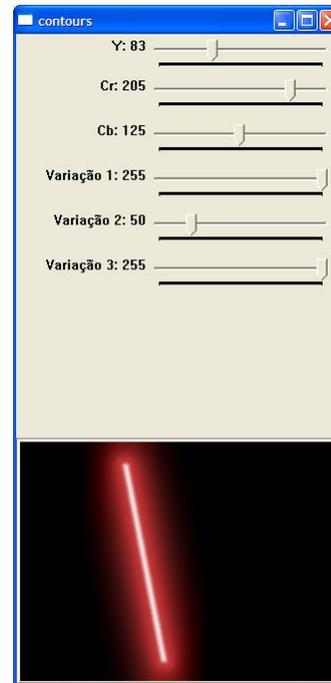


Figura 4. Primeiros testes de desenho de sabre.

tro de massa da imagem segmentada. A comparação com o centro de massa da imagem anterior nos dá a velocidade instantânea do objeto. A maior dificuldade encontrada foi manter a consistência do sistema de filtragem por cores durante condições de movimento do objeto. A maioria das webcams não é capaz de entregar 30 quadros por segundo, fazendo na verdade uma interpolação entre quadros para atingir essa marca. Experimentações com nossa câmera revelaram que sua taxa de frames por segundo é na verdade 10. Isso foi obtido a partir de uma gravação simples de imagem da webcam: a gravação apresentava uma repetição de 3 quadros constantemente.

Atribuímos essa baixa taxa de captura à pequena capacidade de coletar luz da câmera: por ela possuir uma lente pequena (e conseqüentemente menor iris), é preciso que o obturador fique aberto por mais tempo para gerar uma imagem. A imagem gerada possui um efeito de desfoque de movimento.

Um objeto se movimentando rapidamente, como no caso dos sabres, tem sua cor difundida com a cor do plano de fundo. Para contornar esse problema, usamos algumas abordagens: aumentar a tolerância dos componentes HSV, usar uma imagem de diferença absoluta entre 2 frames consecutivos (assim localizando áreas de movimento) para aumentar a tolerância de filtragem nessas áreas e a mais promissora, aproximar a imagem de filtragem da cor procurada.



Figura 5. Primeiros testes com imagem de fundo.



Figura 6. Primeiros testes com dois sabres.

A primeira abordagem, aumentar a tolerância, funciona relativamente bem, mas a partir de certos valores começa a gerar ruídos na detecção e não resolve muito bem o problema do movimento.

A segunda abordagem, usar a imagem de diferença absoluta para aumentar a tolerância nessas áreas, possui o efeito colateral de gerar áreas de detecção onde não há a cor procurada quando há movimento nas mesmas. Além disso a imagem de diferença só fica mais clara nas bordas do movimento. No entanto é um método interessante se usado com um fator de escala pequeno, aumentando apenas um pouco dado um movimento. Notou-se durante a aplicação desse método que o componente H (hue/matiz) é o que mais sofre ruído de um frame para outro, especialmente em áreas escuras. O componente S (saturação) é mais uniforme e o componente V é o mais suave. Resolvemos usar 3 instâncias do componente V por ele ser mais uniforme e preciso quanto à posição do movimento.

A terceira abordagem mostrou-se um pouco mais eficiente que as duas primeiras, mas não deve ser usada em

conjunto com as mesmas sem se tomar cuidado com fatores de escala. A imagem passa a ser somada, canal por canal, à cor alvo, e depois escalada (quanto aos valores de cor) em 50%. Isso gera uma imagem que é uma mescla entre a imagem original e a cor procurada. Com fatores de tolerância baixos o bastante, ela não será detectada pelo filtro de cores. No entanto, quando o sabre se movimentar sobre ela, fazendo a mesclagem de sua cor com a cor do plano de fundo, a imagem cairá dentro dos limites de filtragem, gerando um movimento mais próximo do contínuo.

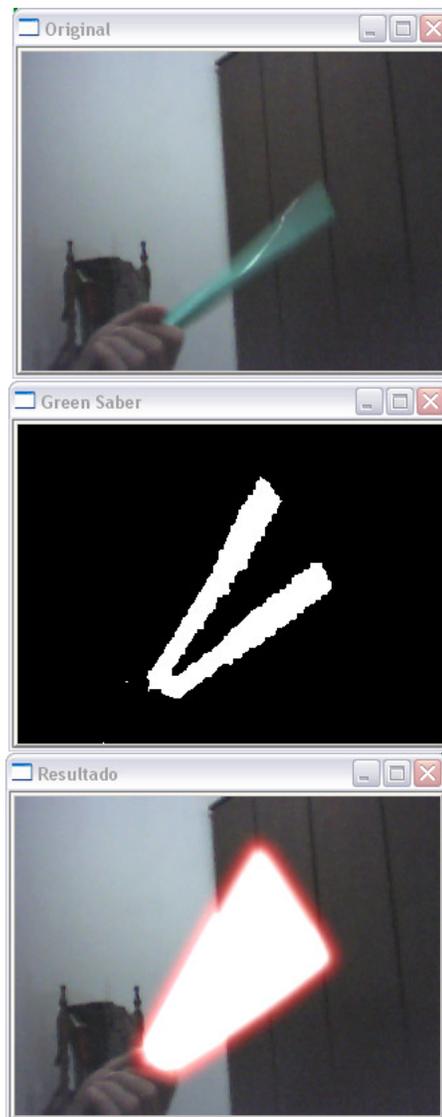


Figura 8. Etapas: a) Imagem original. b) Threshold da imagem. c) Resultado

Essa abordagem é especialmente eficiente quanto à lu-

miniosidade e saturação (canais S e V) e menos eficiente quanto ao matiz (canal H). Acreditamos que uma combinação entre as três técnicas, parametrizadas experimentalmente, poderá gerar o melhor resultado possível dado o hardware disponível.

Referências

- [1] W. R. E. Gonzalez, Rafael C. *Digital Imaging processing*. Editora Edgar Blucher Ltda, São Paulo, 2000.
- [2] Intel. Intel open source computer vision library. Technical report, Intel, <http://www.intel.com/technology/computing/opencv/>, 2008.
- [3] J. M. M. ISARD. Bramble: A bayesian multiple-blob tracker. In *Proceedings of IEEE International Conference on Computer Vision*.
- [4] T. MathWorks. Matlab. Technical report, The MathWorks, <http://www.mathworks.com/>, 2008.
- [5] A. H. R.-C. REINALDO A. C. BIANCHI. O sistema de visão computacional do time futeboli de futebol de robôs. *Anais do Congresso Brasileiro De Automática*, 2000.
- [6] J. S. J. RICHARD Y. D. XU, JOHN G. ALLEN. Robust real-time tracking of non-rigid objects. *NSW*, 2006.
- [7] A. B. S.S. INTILLE, J.W. DAVIS. Real-time closed-world tracking. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.