



ESTUDO SOBRE RASTREAMENTO BASEADO EM
CARACTERÍSTICAS NATURAIS PARA
APLICAÇÕES DE REALIDADE AUMENTADA EM
DISPOSITIVOS MÓVEIS

Proposta de Trabalho de Graduação

Aluno: Guilherme Reis da Rocha Borba Marques (grrbm@cin.ufpe.br)

Orientadora: Veronica Teichrieb (vt@cin.ufpe.br)

Co-orientadores: João Paulo Silva do Monte Lima (jpsml@deinfo.ufrpe.br)
Rafael Alves Roberto (rar3@cin.ufpe.br)

29 de outubro de 2014

1. INTRODUÇÃO

Realidade Aumentada (RA) é a integração de elementos virtuais a visualizações do mundo real (como por exemplo, utilizando a imagem capturada pela câmera) em tempo real. Em aplicações de RA, objetos 3D são inseridos no ambiente real de modo a parecerem fazer parte da cena, se adaptando constantemente a mudanças na mesma. Atualmente, a maior parte das pesquisas com foco em RA está relacionada à utilização de vídeos transmitidos ao vivo, que são processados digitalmente e “ampliados” com inserção de gráficos criados por computador. Para algumas dessas aplicações se faz necessário que o usuário se mova livremente. Outra restrição comum é o uso de dispositivos leves e compactos. Dispositivos móveis modernos, como *smartphones* e *tablets*, conseguem suprir bem essas necessidades e, por esse motivo, projetos de RA com foco em dispositivos móveis tais como os supracitados estão se tornando populares [1].

Apesar do benefício da mobilidade, o desenvolvimento de aplicações de RA para aparelhos móveis é muitas vezes dificultado pelas restrições de memória, processamento e armazenamento desse tipo de aparelho. Isso se deve ao fato de que vários dos algoritmos de rastreamento [2], parte fundamental para as aplicações de RA funcionarem corretamente, possuem custo computacional alto.

Nesse contexto, o trabalho proposto pretende estudar o rastreamento baseado em características naturais em dispositivos móveis. Para poder fazer o registro, isto é, a integração dos objetos virtuais com a cena real de forma que eles fiquem adequadamente alinhados entre si, é necessário fazer um bom rastreamento dos objetos reais. Um meio bastante popular de realizar isto é utilizando marcadores fiduciais planares. Esses são elementos intrusivos adicionados ao ambiente e que possuem um padrão gráfico facilmente reconhecível. Contudo, em muitos casos, utilizar as próprias características naturais do ambiente, sem adicionar elementos novos a ele, é mandatório ou desejável [3]. Esse tipo de rastreamento é mais desafiador de se realizar em dispositivos móveis, pois encontrar e rastrear características reconhecíveis no mundo real é uma tarefa mais complexa.

Com relação ao rastreamento baseado em características naturais, algumas soluções autônomas, ou seja, que executam localmente no dispositivo móvel, foram criadas para reduzir o custo computacional do rastreamento baseado em características naturais. Uma delas é o Studierstube Tracker [4], que começou como uma ferramenta para rastreamento baseado em marcadores, mas recentemente foi atualizada para utilizar características naturais. No Studierstube Tracker não existem alocações de memória em tempo de execução, e a biblioteca Tracker, que realiza o rastreamento, guarda os dados em estruturas amigáveis à *cache*. Com essas melhorias, um *smartphone* médio consegue analisar uma imagem de uma câmera típica em aproximadamente dez milissegundos, o que permite que recursos suficientes

fiquem disponíveis para uso em aplicações com alta exigência de processamento gráfico.

Outra contribuição interessante é a de [5], que conseguiu criar o primeiro sistema de rastreamento baseado em características naturais para telefones capaz de executar numa taxa de trinta quadros por segundo, ou seja, em tempo real. Para isto, examinou-se duas das principais técnicas de casamento de características, o SIFT [6] e o Ferns [7]. As duas abordagens, nas suas formas originalmente publicadas, são impróprias para plataformas embarcadas *low-end* tais como celulares. Alguns aspectos dessas técnicas são inviáveis computacionalmente nos celulares atuais e precisaram ser substituídos por abordagens diferentes, enquanto outros aspectos foram simplificados para funcionar no nível desejado de desempenho, qualidade e consumo de recursos. Com essas modificações, foi possível criar um sistema de rastreamento bastante eficiente.

As primeiras tentativas de realizar RA que utilizavam rastreamento baseado em características naturais com uma abordagem distribuída (cliente-servidor) deixavam todo o processamento no servidor, desde o rastreamento até a renderização. O trabalho de [8] implementou uma solução desse tipo utilizando um PDA, que simplesmente capturava as imagens da cena com a câmera e as enviava para um servidor, que fazia todo o processamento necessário e enviava de volta para o dispositivo os dados a serem renderizados, sobrepostos no vídeo do ambiente.

A pesquisa de [9] demonstra um sistema com apenas 1 segundo de latência. Esse resultado foi obtido de forma inteligente, utilizando o movimento do visor do dispositivo móvel para inferir o interesse do usuário. Uma vez que o usuário repousa o visor sobre uma área do ambiente por um instante, o sistema decide que o usuário se interessou pelo que está vendo, e automaticamente manda aquele quadro para o servidor. Isso é muito mais eficiente do que continuamente enviar quadros para consulta.

2. OBJETIVOS

2.1. OBJETIVO GERAL

O trabalho proposto nesse documento tem por objetivo a investigação de técnicas de rastreamento baseado em características naturais, focando em dispositivos móveis. As soluções de rastreamento desenvolvidas serão avaliadas levando em consideração medidas tais como desempenho, consumo de memória, armazenamento e latência. Latência, no contexto do presente trabalho, é o atraso na exibição do conteúdo virtual casado com o real.

O rastreamento realizado no presente trabalho é baseado em características invariantes locais e é composto de algumas etapas. Primeiramente, são extraídas da imagem da cena características discriminativas e repetíveis [3]. O tipo de característica utilizada neste trabalho é o ponto-chave, que pode ser entendido como um ponto de alto destaque na imagem. A detecção dos objetos é então realizada através do casamento das características extraídas da imagem de consulta com características que foram extraídas antecipadamente de imagens do objeto onde sua pose é conhecida. Uma alternativa para realizar esse casamento é utilizando-se descritores locais, que são vetores de alta dimensão que descrevem a vizinhança em torno da característica. O casamento é feito através da busca pelo vizinho mais próximo baseando-se na distância entre os vetores de alta dimensão. A pose pode ser entendida como as informações de posição e orientação de um objeto em relação ao sistema de coordenadas da câmera. Após o casamento, uma lista de pares de características com propriedades semelhantes será obtida. Porém, muitas dessas características não serão casadas corretamente por vários motivos, como a característica ter sido extraída de uma região do fundo da imagem. Então, é interessante descartar os casamentos que provavelmente são *outliers*, verificando se a razão entre as distâncias do primeiro e segundo vizinhos mais próximos é menor que um limiar. Podemos chamar essa etapa de filtragem. Por último, é calculada a pose do objeto utilizando o algoritmo EPnP [10] juntamente com o algoritmo RANSAC [11] para remoção de *outliers*.

Vão ser implementadas duas soluções, ambas para a plataforma Android. As duas utilizarão OpenCV [12], uma biblioteca de visão computacional que funciona tanto em PCs como em dispositivos móveis.

A primeira solução consiste na paralelização do *pipeline* de rastreamento, que possui quatro fases: extração de características, casamento, filtragem e cálculo de pose. Isso será feito dividindo-se a imagem de entrada da câmera em tantas partes quantas forem os núcleos do processador do dispositivo móvel. Então, cada uma das partes passará pelas fases de extração, casamento e filtragem separadamente, e as saídas serão unidas no final, para servir de entrada para a última parte do *pipeline*, o cálculo de pose. Não será necessário paralelizar o cálculo de pose, pois a biblioteca OpenCV4Android já faz isso internamente, utilizando o método SolvePNPRansac [13].

A biblioteca TBB da Intel [14] será utilizada para paralelizar os passos dessa solução. Essa é uma biblioteca baseada em *templates* na linguagem C++ que auxilia o programador no desenvolvimento de aplicações que utilizam o paralelismo dos múltiplos núcleos do processador.

A segunda solução consiste em transferir a complexidade do rastreamento para um servidor. O cliente será um dispositivo móvel e o servidor será um PC *desktop*, e a comunicação entre os dois acontecerá utilizando uma tecnologia de comunicação sem fio, como a Wi-Fi. A extração das características da imagem da câmera deve gerar como saída um conjunto de pontos-chave e outro de descritores. Essas estruturas serão passadas para o servidor utilizando *sockets*. O servidor então faz todas as etapas do rastreamento, e devolve a pose calculada para o cliente. No caso desta solução, será realizada uma avaliação da latência para verificar a viabilidade deste método.

2.2. OBJETIVOS ESPECÍFICOS

- 1) Estudo de técnicas de rastreamento baseado em características naturais na área de RA móvel;
- 2) Implementação do protótipo do rastreamento baseado em características naturais local, otimizado com paralelismo;
- 3) Implementação do protótipo do rastreamento baseado em características naturais com a abordagem distribuída;
- 4) Realização de estudos de caso para avaliação e comparação das abordagens desenvolvidas;
- 5) Escrita da monografia do Trabalho de Graduação.

3. CRONOGRAMA

Etapa/Fase	Mês				
	Out	Nov	Dez	Jan	Fev
Pesquisa Bibliográfica					
Seleção do material bibliográfico					
Leitura do material selecionado					
Fichamento					
Redação da revisão bibliográfica					
Implementação da Solução Local					
Implementação da Solução Distribuída					
Avaliação e Comparação das Soluções					
Escrita do Trabalho					

REFERÊNCIAS BIBLIOGRÁFICAS

- R. T. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier e B. MacIntyre,
[1] "Recent Advances in Augmented Reality," *IEEE Computer Graphics and Applications*, 21(6), pp. 34-47, Novembro/Dezembro 2001.
- J. P. Lima, F. Simões, L. Figueiredo e J. Kelner, "Model Based Markerless
[2] 3D Tracking applied to Augmented Reality," *Journal on 3D Interactive Systems*, vol. 1, pp. 2-15, 2010.
- J. P. Lima, "Object Detection and Pose Estimation from Rectification of
[3] Natural Features Using Consumer RGB-D Sensors," Tese de Doutorado, Universidade Federal de Pernambuco. Recife, 2014.
- D. Wagner e D. Schmalstieg, "Making Augmented Reality Practical, Part 1,"
[4] *IEEE Computer Graphics and Applications*, 29(3), pp. 12-15, Maio/Junho 2009.
- D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond e D. Schmalstieg, "Real-
[5] Time Detection and Tracking for Augmented Reality on Mobile Phones," *IEEE Transactions on Visualization and Computer Graphics*, 16(3), pp. 355-368, Maio/Junho 2010.
- D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints,"
[6] *International Journal of Computer Vision*, 60(2), pp. 91-110, Novembro 2004.
- M. Ozuysal, P. Fua e V. Lepetit, "Fast Keypoint Recognition in Ten Lines of
[7] Code," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2007.
- W. Pasmán e C. Woodward, "Implementation of an Augmented Reality
[8] System on a PDA," *IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 276-277, 2003.
- D. M. Chen, S. S. Tsai, R. Vedantham, R. Grzeszczuk e B. Girod,
[9] "Streaming Mobile Augmented Reality on Mobile Phones," *IEEE International Symposium on Mixed and Augmented Reality*, pp. 181-182, 2009.
- F. Moreno-Noguer, V. Lepetit e P. Fua, "Accurate Non-Iterative $O(n)$
[10] Solution to the PnP Problem," *IEEE International Conference on Computer Vision*, pp. 1-8, 2007.

M. Fischler e R. Bolles, "Random Sample Consensus: A paradigm for model
[11] fitting with applications to image analysis and automated cartography,"
Communications of the ACM, 24(6), p. 381-395, 1981.

"OpenCV," [Online]. Available: <http://opencv.org/>. [Acesso em 13 Outubro
[12] 2014].

"Camera Calibration and 3D Reconstruction," [Online]. Available:
[13] [http://docs.opencv.org/modules/calib3d/doc/camera_calibration_and_3d_r
econstruction.html](http://docs.opencv.org/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html). [Acesso em 13 Outubro 2014].

"Intel TBB," [Online]. Available: <https://www.threadingbuildingblocks.org/>.
[14] [Acesso em 13 Outubro 2014].

Datas e Assinaturas

Recife, 29 de outubro de 2014.

Veronica Teichrieb (Orientadora)

João Paulo Silva do Monte Lima (Co-orientador)

Rafael Alves Roberto (Co-orientador)

Guilherme Reis da Rocha Borba Marques (Proponente)