2
Trabalhos Relacionados

Neste capítulo discutimos alguns trabalhos relacionados a esta dissertação divididos entre três seções. Primeiramente na Seção 2.1 apresentamos duas pesquisas na área de renderização remota para dispositivos móveis. Na Seção 2.2 abordamos o conceito de jogos em nuvem (*cloud gaming*) e as principais soluções proprietárias disponíveis no mercado. Na Seção 2.3 citamos alguns trabalhos que lidam com o processo de detecção de gestos.

2.1
Renderização remota

O conceito de renderização remota surgiu quando computadores não tinham poder de processamento suficiente para renderizar cenários 3D e necessitavam de compartilhar na rede uma estação de trabalho gráfica. Várias pesquisas foram desenvolvidas nessa área ao longo dos anos com foco em diferentes técnicas de renderização. Nesta proposta optamos por utilizar uma abordagem baseada em imagens, onde o servidor gera imagens da cena renderizada e envia para o cliente que desconhece qualquer informação sobre a geometria da cena.

Seguindo nesta linha de renderização baseada em imagens para dispositivos móveis temos o trabalho de Lamberti et al. (8) que adotou uma abordagem de transmissão de *stream* usando o formato de vídeo MPEG para acelerar a renderização gráfica em PDAs. Lamberti et al. (8) levavam em consideração que comumente dispositivos móveis possuem um hardware dedicado para decodificar *streams* de vídeo. Neste mesmo trabalho, Lamberti et al. (8) apresentaram uma análise dos problemas de latência e como variações na qualidade da compressão de imagens afetavam os resultados obtidos.

Shu Shi (13) propôs um sistema que utiliza uma técnica de *warping* 3D de imagens utilizando várias imagens com profundidade como referência para reduzir a latência da interatividade do usuário. Quando o usuário interage na cena, o cliente, que roda no dispositivo móvel, executa um algoritmo de *warping* 3D das imagens de referência para gerar uma imagem do novo ponto de vista do usuário. Essa imagem é exibida até que uma nova imagem renderizada seja recebida do servidor de renderização. O principal desafio nesta abordagem é selecionar os melhores pontos de vista para serem usados como referência afim de minimizar os artefatos gerados durante o *warping*.

2.2 Plataformas de Jogos em Nuvem

O setor da indústria de jogos tem sido o principal impulsionador no avanço das tecnologias de renderização 3D. Normalmente os jogadores precisam comprar consoles ou placas gráficas poderosas para jogar os jogos mais avançados. Com o surgimento recente dos serviços de *Cloud gaming* (jogos em nuvem), a idéia de renderização remota começou a ser introduzida na indústria de jogos.

OnLive (11) é um exemplo bastante conhecido deste modelo de negócio, onde as cenas dos jogos são extraidas do framebuffer dos servidores de renderização e transmitidas via stream para os jogadores através de redes de banda larga. Um codificador de video é usado para comprimir as imagens e reduzir o consumo de banda. Sem a necessidade de um hardware avançado, os usuários podem jogar em tablets, na TV ou em computadores mais leves. A interação com os jogos pode ser feita através de mouse, teclado ou controle que tem seus sinais de entrada coletados e enviados de volta aos servidores de renderização.

Seguindo o mesmo conceito do OnLive, existem várias outras soluções proprietárias, tais como Gaikai (4), e OTOY (12). Outro serviço interessante é o StreamMyGame (14), um *software* que permite usuários configurarem sua própria máquina como um servidor de jogos e transmitir via stream o jogo para outro computador rodando como cliente.

2.3 Multitoque e detecção de gestos

A popularização dos smartphones e tablets gerou uma demanda muito grande na utilização de telas sensíveis ao toque, aumentando cada vez mais a aceitação desse tipo de interface entre os usuários. Sem a necessidade de um dispositivo intermediário como o mouse, o usuário interage diretamente sobre a informação que vê na tela. Por consequência, os softwares estão tendo que se adaptar rapidamente a essa nova forma de interação. Diversas pesquisas nessa área vêm surgindo com o intuito de facilitar a adoção deste tipo de interação.

Echtler e Klinker (3) propuseram uma arquitetura em camadas capaz de abstrair uma variedade de hardwares multitoque de forma que os mesmos fossem facilmente integrados em softwares com interação multitoque. Visando padronizar como os eventos de toques são obtidos, tratados e interpretados como gestos, a arquitetura propôs a divisão destas tarefas em camadas, permitindo assim que o software seja compatível com diferentes tipos de dispositivos multitoque. A figura 2.1 mostra as camadas da arquitetura proposta.

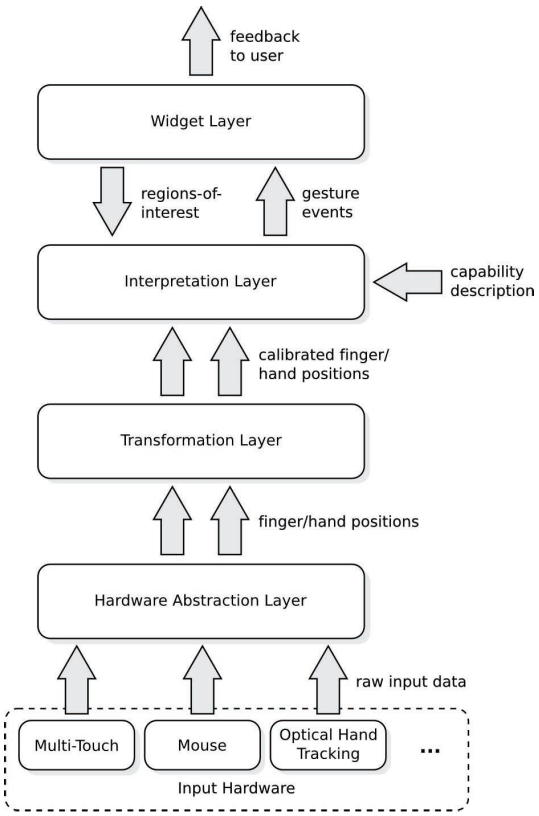

Figura 2.1: Arquitetura proposta por Echtler e Klinker (3) para detecção de gesto.

Alguns protocolos multitoque também surgiram com o intuito de padronizar a transmissão de dados referentes aos toques. TUIO (6) é um destes protocolos que serve de interface entre a aplicação e o dispositivo multi-toque. O TUIO permite uma transmissão de descrições abstratas de superfícies interativas, incluindo não só os eventos de toque como também o estado dos objetos tangíveis sendo manipulados, fornecendo informações de posição, orientação, tamanho, etc.

Algumas plataformas móveis já dispõem de um conjunto padrão de gestos implementados que são fáceis de usar, mas a medida que mais gestos são necessários para enriquecer a interatividade, a complexidade da interpretação de gestos aumenta. Isso se deve à necessidade de tratar os conflitos que podem existir entre os gestos definidos. Para lidar com este tipo de problema Kin et al. (7) sugerem a definição de gestos através de expressões regulares. Essas expressões são definidas por meio de uma espécie de tablatura gráfica inspirada nas tablaturas de notas musicais, onde cada faixa da tablatura representa uma sequência de eventos de um toque.

No framework proposto, optamos por seguir o modelo de arquitetura proposto por Echtler e Klinker (3), dividindo a detecção de gestos em etapas sem se preocupar com a origem dos eventos de toque.