

6 Conclusão e Trabalhos Futuros

Com base nos resultados obtidos, pode-se concluir que o trabalho apresentado atingiu parcialmente seus objetivos.

Os protótipos propostos são capazes de reproduzir a navegação em ambientes virtuais, incluindo a interação com objetos ativos. Apesar da pouca interação apresentada em relação a esses objetos, a arquitetura é extensível e possibilita a inclusão de novos módulos de interação sem muito esforço.

Tanto a implementação de desenho direto em 2D e a implementação 3D demonstraram boa performance em dispositivos de baixa capacidade de processamento .

Mesmo não sendo uma implementação centrada em técnicas de renderização baseada em imagens, o protótipo que utiliza o desenho direto do panorama atinge a maioria dos celulares presentes no mercado, já que utiliza a API básica da plataforma J2ME, e como se mostrou bastante funcional, pode ser utilizada em aplicações comerciais.

Já a implementação com o uso de *image warping*, demonstrou que manipulação de imagens não possui boa performance na plataforma testada.

Todos os protótipos possuem pequenos *footprints* variando de 29 a 38 KB, com a modelagem de um ambiente decomposto em pacotes com menos de 100KB para a aplicação com visualização 3D e de menos de 500KB para as aplicações com visualização 2D e *Warping*. Ela também apresenta uma interface com o usuário intuitiva e a navegação, apesar de ser feita de forma discreta, pode ser feita de forma rápida e não necessita comunicação constante com o servidor, como os trabalhos apresentados por (Bao & Gourley, 2004) e (Thomas,2005) por exemplo, o que é atrativo para usuários finais, além de melhorar a usabilidade.

Por utilizarem uma abordagem baseada na modelagem de panoramas virtuais, assim como outros sistemas com as mesmas características, como o *Quick Time VR* (Chen,1995), os protótipos apresentados possuem um sistema de autoria simples, se comparado com sistemas de IBR com amostragem densa como o *Lumigraph*(Gotler,1996) e o *Lightfield*(Levoy,1996), ou sistemas com ambientes modelados. Mesmo não tendo sido desenvolvido um sistema para

automatizar a construção do descritor das cenas, este é de fácil manipulação por ser baseado em texto.

6.1. Problemas encontrados

Mesmo com os bons resultados quanto à taxa de geração de quadros e da interação com objetos presentes, nos testes com dispositivos reais, as aplicações apresentaram alguns problemas na qualidade de visualização das cenas.

6.1.1. Falta de realismo na visualização dos panoramas na implementação com API 2D e desenho direto das imagens

A implementação com o uso de desenho direto das imagens, por não reproduzir as distorções características de um panorama cilíndrico, apresenta falta de realismo, principalmente para cenas em que o fundo retrata paisagens próximas da câmera virtual.

6.1.2. Baixa qualidade na visualização dos panoramas na implementação com API 3D

A baixa qualidade na visualização do panorama no o protótipo que utiliza API 3D, é devida a uma limitação dos telefones celulares escolhidos para os testes, mas que representam a classe de aparelhos foco desta dissertação.

A abordagem selecionada para a implementação da visualização do panorama foi a de aproximação poligonal, onde, aproveitando as rotinas de renderização de gráficos 3D presentes, a imagem do panorama é mapeada como textura em um modelo que aproxima um cilindro.

A dificuldade encontra-se na implementação interna de tais rotinas que pedem que as texturas tenham dimensões que sejam potências de dois, além, de nos dispositivos reais utilizados nos testes, essas dimensões não podem exceder 256 pixels. Vale ressaltar que esta limitação não é inerente a todos os telefones celulares disponíveis no mercado.

Desta forma, torna-se difícil atingir uma boa qualidade na visualização com apenas uma imagem representando o panorama, pois é necessária sua representação contando com apenas 256 pixels de largura.

Porém, a implementação do M3G(*Mobile Graphics 3D*) possibilita que mais de uma textura seja empregada em um mesmo material. Assim, uma solução para o problema seria, durante o processo de autoria, a divisão da imagem do panorama em imagens, que separadas, precisariam respeitar o limite de 256 pixels. Assim cada uma não deveria perder muita qualidade em sua adaptação. No caso desta solução, o protótipo deve ser modificado incluindo o mapeamento de cada uma dessas imagens.

6.1.3.

Distorções no panorama na implementação com API 3D

A modelagem do cilindro utilizado como base para a visualização do panorama foi feita com base em um modelo gerado pela ferramenta 3D Studio Max, que posteriormente foi exportado para o formato H3T, como já foi explicitado no capítulo 4.

Esta abordagem foi utilizada por facilidade de implementação, mas traz algumas conseqüências.

Primeiro, a quantidade de triângulos que compõem o cilindro poderia ser menor, já que cada face poderia ser representada por um *quad*, ou seja, apenas dois triângulos.

A grande quantidade de triângulos do modelo atual, pode ser a causa do baixo rendimento no Sony Ericsson K750, utilizado nos testes apresentados.

Outra característica é o fato de o “cilindro” apresentar apenas dezoito lados, o que pode apresentar deformidade no panorama dependendo do zoom e imagens empregadas na visualização.

Outro problema encontrado é que como um único modelo é utilizado, as câmeras, reais ou virtuais, utilizadas para a geração dos panoramas, devem ter as mesmas características de lentes e distância focal, para não ocorrerem distorções no panorama. Apesar de provável que uma mesma câmera seja utilizada para a aquisição de todas as imagens necessárias para a representação de um mesmo ambiente, isto limita o processo de autoria, já que esta mesma câmera deve ser empregada na geração de todas as cenas.

A solução é modificar a modelagem do cilindro, tornando-a dinâmica e dependente de parâmetros da imagem a ser utilizada como textura, além de utilizar um menor número de triângulos. Também é necessário permitir que o campo de visão da câmera seja definido a partir destes parâmetros.

6.1.4. Falta de maior realismo nos objetos.

Todos os objetos são representados por *billboards*, tanto através da classe *Sprite*, nas implementações com os métodos de manipulação de imagens 2D, quanto da classe *Sprite3D*, na implementação que faz uso da biblioteca *Mobile 3D Graphics* da plataforma J2ME.

No caso da visualização baseada na API 3D, tais objetos, apesar de apresentarem melhor performance na renderização de cada quadro, não sofrem interferência de nenhuma luz presente no ambiente.

Outro problema encontrado em todos os protótipos, é o fato de os objetos não apresentarem as distorções esperadas quando o ângulo de visão se altera, uma vez que por são *billboards*, *sprites* que sempre estão voltados para câmera.

Uma solução seria a utilização de *3D warping* nas imagens que representam cada objeto. Apesar de a plataforma possibilitar a edição das imagens quadro a quadro, e estas imagens não apresentarem grandes dimensões, a performance deste método pode tornar-se impraticável.

Outra solução para dar um maior realismo a esses objetos, sem o uso de modelos mais complexos, é a utilização de *billboard clouds* (Decoret,2003). Porém, com base nos resultados obtidos com a presença de vários objetos em um mesmo ângulo de visualização do panorama, o uso de muitas camadas podem degradar a performance.

6.2. Trabalhos Futuros

Como trabalhos futuros propõe-se as correções dos problemas apresentados.

Para a implementação 3D, propõe-se:

- Separação da imagem do panorama, para seu mapeamento como múltiplas texturas, evitando a grande perda da qualidade por causa da limitação nas dimensões.

- Modificação da modelagem do cilindro utilizado como base para o panorama, utilizando menos triângulos e gerando-o de forma dinâmica a partir das informações da imagem.

- Modificação dos parâmetros da câmera virtual, também de acordo com as informações de geração da imagem do panorama.

- Maior flexibilização do *framework*, tornando mais genérico, podendo incluir outros tipos de objetos e maior facilidade na inclusão de outros tipos de panorama (e.g esféricos e cúbicos).

Para todas as implementações, propõe-se:

- Representação dos objetos como *billboard clouds* (Decoret,2003) com poucas camadas, com o objetivo de dar maior realismo os objetos presentes no ambiente.

Por fim, com as dificuldades encontradas na aplicação de técnicas de warping de imagens, propõe-se o aumento na gama de dispositivos, verificando quais são as reais limitações da plataforma J2ME no desenvolvimento de aplicações baseadas em técnicas de renderização baseado em imagens.