

William Fernando Toazza di Domenico

*Detecção de colisão e dinâmica de corpos rígidos em
realidade aumentada*

Florianópolis, Santa Catarina

Dezembro de 2012

William Fernando Toazza di Domenico

***Detecção de colisão e dinâmica de corpos rígidos em
realidade aumentada***

Trabalho de conclusão de curso apresentado
como parte dos requisitos para obtenção do grau
de Bacharel em Sistemas de Informação

Orientador:

Prof. Dr. rer.nat. Aldo von Wangenheim

Co-orientador:

M.Sc. Tiago de Holanda Cunha Nobrega

Florianópolis, Santa Catarina

Dezembro de 2012

Resumo

Realidade aumentada é o processo de fazer com que objetos virtuais pareçam existir no mundo real. Na pesquisa corrente deste assunto há grande preocupação com a parte da detecção do mundo real, mas isto não basta para se conseguir uma experiência verossímil. Enquanto houver uma grande discrepância entre os objetos virtuais e o mundo real, a percepção de que eles existem é completamente comprometida.

Este trabalho mostra formas de aumentar esta sensação de existência, aumentando o realismo da aparência e do movimento dos objetos virtuais. Usando técnicas de computação gráfica fornecidas por uma *engine* de renderização como sombras e iluminação, a aparência dos objetos foi perceptivelmente melhorada.

Já o realismo de seu movimento foi melhorado usando detecção de colisão com o mundo real e simulação de dinâmica de corpos rígidos para determinar o movimento causado por estas colisões. O trabalho mostra como estes recursos foram obtidos pela integração de uma engine física.

Sumário

Lista de Figur

1	Introdução	p. 7
1.1	Motivação e justificativa	p. 8
1.2	Objetivos	p. 9
1.2.1	Objetivos gerais	p. 9
1.2.2	Objetivos específicos	p. 10
1.3	Reconhecimento do mundo real	p. 10
1.3.1	Rastreamento	p. 11
1.3.2	Estimação de pose	p. 12
1.3.3	Codificação dos marcadores	p. 14
1.3.4	Tabuleiros	p. 15
1.4	Renderização	p. 15
1.4.1	Grafo de cena	p. 16
1.4.2	Câmara	p. 16
1.4.3	Materiais	p. 17
1.4.4	Sombras	p. 19
1.5	Simulação física	p. 20
1.5.1	<i>Engine</i> física	p. 20
1.5.2	Detecção de colisão	p. 21
1.5.3	Dinâmica de corpos rígidos	p. 22

2	Revisão sistemática da literatura	p. 23
2.1	Protocolo	p. 23
2.2	Detecção do mundo real	p. 24
2.3	Renderização	p. 25
2.4	Simulação física	p. 26
3	Reconhecimento do mundo real	p. 27
3.1	Escolha da biblioteca de realidade aumentada	p. 27
3.2	Calibragem da câmera	p. 28
3.3	Remoção da distorsão da câmera	p. 29
3.4	Detecção dos marcadores	p. 29
3.5	Estabilização da detecção	p. 29
3.6	Limitações	p. 30
4	Renderização dos objetos virtuais	p. 31
4.1	Escolha da <i>engine</i> de renderização	p. 31
4.2	Integração com realidade aumentada	p. 31
4.3	Projeção de sombras	p. 34
4.4	Oclusão de objetos virtuais por reais	p. 35
5	Simulação	p. 36
5.1	Escolha da engine de física	p. 36
5.2	Integração com a engine de renderização	p. 37
5.3	Comparação do comportamento dos objetos antes e após uso da simulação física	p. 40
6	Conclusões	p. 41
6.1	Trabalhos futuros	p. 41

Lista de Figuras

1.1	Exemplos de realidade aumentada em aparelhos portáteis da Nintendo e Sony.	p. 7
1.2	Google Glass, um exemplo de realidade aumentada por sobreposição: os objetos virtuais não interagem com o mundo real.	p. 8
1.3	Sistema de coordenadas local da câmera.	p. 10
1.4	Processo de detecção	p. 11
1.5	Câmera estenopeica	p. 12
1.6	Modelo de câmera estenopeica. Se o plano for invertido para a frente do centro, ele conterà a mesma imagem, mas girada em 180°.	p. 13
1.7	Visão sobre o eixo Y e X, respectivamente, mostrando geometria usada para determinar os valores de x e y do ponto no mundo real, a partir de x e y da imagem.	p. 13
1.8	Código usado nos marcadores	p. 14
1.9	Exemplo de tabuleiro de marcadores	p. 15
1.10	Comportamento de um grafo de cena a medida que nós são adicionados a ele.	p. 16
1.11	Tronco de visão ou, em inglês como é mais conhecido, <i>view frustum</i> e planos da distância máxima e mínima da visão	p. 17
1.12	Diferença da aparência do objeto de acordo com o nível especular, brilho e suavidade de seu material. Fonte: (Junker, 2006)	p. 18
1.13	Você consegue dizer quão alto a estátua de Atena está sobre o chão na figura de cima? E na de baixo? Fonte: (Junker, 2006)	p. 19
3.1	Exemplo de ângulos das fotos usadas para se calibrar a câmera	p. 28
4.1	Sistema de coordenadas da biblioteca de realidade aumentada	p. 32
4.2	Sistema de coordenadas da biblioteca gráfica	p. 33
4.3	Objeto virtual com e sem sombra.	p. 34

4.4	À esquerda o objeto real não oclui o virtual. Já à direita, o objeto real oclui o virtual.	p. 35
5.1	Sistema de coordenadas do mundo físico	p. 38
5.2	Comparação entre interação de objetos com e sem uso de simulação física de seu movimento.	p. 40
6.1	Realidade aumentada em aparelho portátil	p. 42
6.2	Rastreamento cromático	p. 42
6.3	Rastreamento por características naturais	p. 43

1 Introdução

A realidade aumentada funciona como um holograma que só é visível na tela de algum aparelho eletrônico. O vídeo é reconhecido e um objeto virtual é sobreposto à ele de forma a parecer presente no vídeo. Para quem a usa, é possível caminhar ao redor do objeto e visualizá-lo de todos os ângulos como a um objeto real.

É uma variação da realidade virtual que ao invés de imergir o usuário dentro de um ambiente virtual, suplementa o mundo real com objetos virtuais. Ou seja, ao invés de substituir o mundo real por completo, o complementa. (Azuma, 1997)



(a) Nintendo 3DS



(b) PS Vita

Figura 1.1: Exemplos de realidade aumentada em aparelhos portáteis da Nintendo e Sony

Estes objetos virtuais aumentam as informações do mundo real, de onde se cunhou o termo *augmented reality*. Para o autor deste trabalho a tradução mais apropriada do termo seria realidade estendida, no entanto já se encontra como padrão na literatura o uso do termo realidade

aumentada e portanto esta será a tradução usada neste trabalho. Outro termo encontrado para descrever a realidade aumentada é realidade misturada, por misturar objetos virtuais à um ambiente real.

Estes objetos podem estar sobrepostos (figura 1.2) ou em uma composição (figura 1.1) com o mundo real. Neste trabalho a realidade foi aumentada pela composição dos objetos virtuais com o mundo real, de forma que haja interação entre eles e objetos reais.

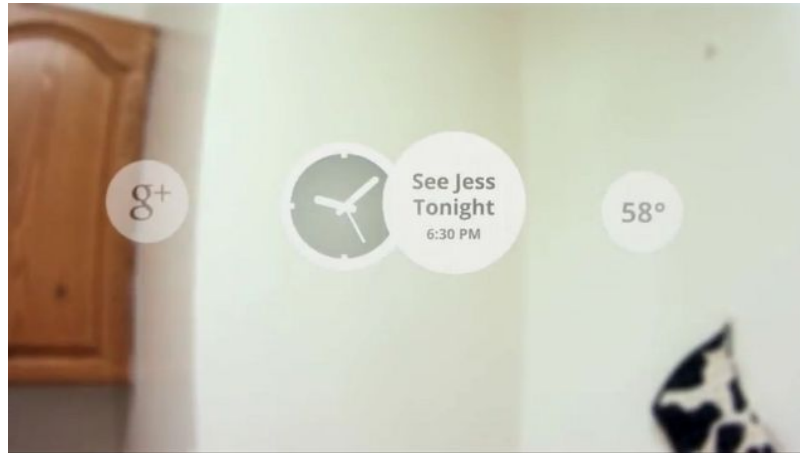


Figura 1.2: Google Glass, um exemplo de realidade aumentada por sobreposição: os objetos virtuais não interagem com o mundo real.

1.1 Motivação e justificativa

Há uma grande quantidade de pesquisa da realidade aumentada buscando a melhoria da detecção do mundo real, no entanto existem outros aspectos a serem considerados para melhorar a experiência dentro dela.

Um deles diz respeito ao realismo dos objetos virtuais. Em grande parte das aplicações que usam realidade aumentada se nota uma grande discrepância entre os objetos virtuais e o mundo real. Isto dificulta sua manipulação porque não há uma boa noção espacial deles. Apesar de apresentar técnicas para melhorar a detecção do mundo real, a principal contribuição deste trabalho é neste aspecto: a melhoria do realismo dos objetos virtuais da realidade aumentada.

O realismo dos objetos virtuais pode ser dividido em três aspectos: aparência, comportamento e conformidade (Ferber, 2003; Chae, Ko, 2008). Este trabalho trata da melhoria do realismo através da aparência e do comportamento.

Através da renderização dos objetos virtuais por uma engine de renderização, se pretende

melhorar a aparência dos objetos, usando recursos como sombras, sombreamento e iluminação. Além disto, esta engine fornece um grafo de cena, que é uma forma apropriada para se representar a relação entre objetos virtuais e reais e serve de referência para todas as representações dos objetos: o mundo físico, gráfico e da detecção

As reações dos objetos virtuais ao estímulo dos objetos detectados pretende ser melhorada com o uso de uma biblioteca de simulação física, que detecta quando os objetos entram em contato e simula seus movimentos de acordo com as forças da física mecânica clássica, como gravidade e atrito.

Durante a revisão da literatura foi encontrado um grande número de publicações sobre uso de engines gráficas e simulação física para realidade aumentada. No entanto nenhum deles explicou em detalhes a forma como fazia a integração entre estas ferramentas.

Este trabalho pretende se diferenciar dos demais oferecendo uma explicação detalhada de como fazer esta integração. Outro diferencial foi a escolha de uma engine de renderização e biblioteca de realidade aumentada não encontrada em nenhuma outra publicação durante a revisão da literatura. Estas bibliotecas foram escolhidas objetivamente por suas capacidades técnicas e são bastante superiores as encontradas nos outros trabalhos.

O aspecto não abordado, a conformidade, é o quanto a aparência do objeto virtual é alterada pelo mundo real da mesma forma que nos objetos reais. Um exemplo que ilustra isto é a iluminação dos objetos pela mesma direção dos objetos reais e portanto projeção de sombras também na mesma direção. Esta característica também diz respeito a aparência, mas como envolve não somente renderização, mas também detecção, ela é resolvida de forma bastante diferente e não foi tratada neste trabalho.

Apesar de não ser o foco do trabalho, formas para melhor a detecção dos objetos reais são apresentadas.

1.2 Objetivos

1.2.1 Objetivos gerais

Este trabalho pretende aumentar o realismo dos objetos virtuais da realidade aumentada e tornar a detecção dos objetos reais mais estável.

1.2.2 Objetivos específicos

- Pesquisa, teste e escolha da biblioteca de realidade aumentada.
- Pesquisa, teste e escolha da engine gráfica, avaliando quão bem ela se integram à biblioteca de realidade aumentada.
- Pesquisa, teste e escolha da engine física, avaliando quão bem ela se integram à biblioteca de realidade aumentada e à engine gráfica.
- Integração da biblioteca de realidade aumentada com a engine gráfica
- Integração da biblioteca de realidade aumentada com a engine física.
- Desenvolvimento de protótipos, ajustando as propriedades físicas dos objetos para aumentar o realismo de seu movimento e testando recursos da engines gráfica que aumentem seu realismo.

1.3 Reconhecimento do mundo real

O reconhecimento do mundo real neste trabalho é feito através de objetos planos facilmente separáveis da imagem do vídeo chamados de marcadores.

Seu reconhecimento envolve três etapas: primeiro a detecção da área dos marcadores, chamada de rastreamento. O rastreamento serve de entrada para a segunda etapa: o cálculo da posição e orientação do marcador. O conjunto das duas coordenadas leva o nome de pose, de onde veio o nome desta etapa: estimacão de pose. A terceira etapa é a identificação do código dos marcadores.

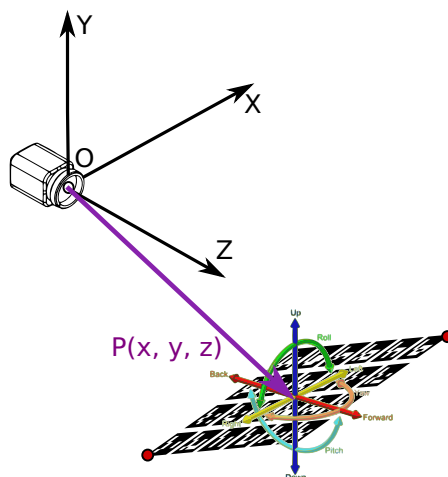


Figura 1.3: Sistema de coordenadas local da câmera

1.3.1 Rastreamento

A etapa de rastreamento começa com a realização da binarização dos píxeis da imagem, processo chamado de *thresholding*, que em português significa limiarização. Esta binarização é feita através da classificação dos píxeis de acordo com um limite. Os píxeis abaixo deste limite são considerados pretos e os acima deste limite brancos. Estes últimos são considerados como as bordas detectadas.

Como após este passo muitas bordas além das dos marcadores - que são as únicas desejadas - são detectadas, todos os passos seguintes envolvem a filtragem destas bordas indesejadas de forma que ao final restem apenas as bordas dos marcadores.

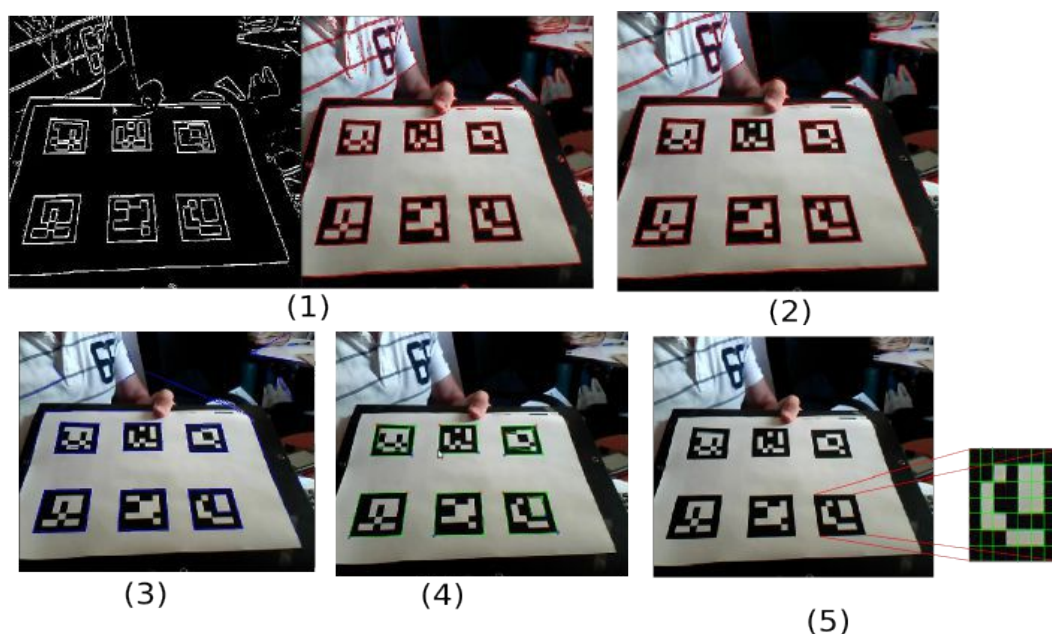


Figura 1.4: Processo de detecção

A remoção das bordas da imagem é feita com os seguintes passos:

1. Remoção das bordas com números pequenos de pontos (imagem 2 da figura 1.4)
2. Aproximação poligonal dos contornos e filtragem dos contornos que não são concavos e não possuem exatamente 4 cantos ao mesmo tempo, ou seja, não são retângulos (imagem 3 da figura 5).
3. Ordenação dos cantos em direção anti-horária.

4. Remoção dos retângulos muito próximos. Isto é necessário porque o threshold adaptativo normalmente detecta uma parte interna no marcador. Após se remover estes retângulos, ainda se consegue manter a maior parte das externas (imagem 4 da figura 5). (Salinas, 2012)

1.3.2 Estimação de pose

O rastreamento envolve apenas a localização do marcador em uma imagem bidimensional. Os pontos da imagem com os cantos dos marcadores são detectados e a partir deles é possível determinar a posição do marcador na imagem.

Uma vez detectado onde os marcadores estão na imagem, esta informação é usada para se estimar as coordenadas do marcador no mundo real. Este mapeamento de pontos bidimensionais da imagem para pontos tridimensionais no mundo real é feito com o uso do modelo de câmera estenopeica.

Modelo de câmera estenopeica

Este modelo é baseado nas cameras obscuras da idade média e em seu sucessores: o daguerreótipo e principalmente a camera estenopeica, herdando seu nome. Esta câmera era uma caixa onde se captava a imagem através de um buraco de alfinete em um dos lados, revelando a imagem em um papel fotossensível do lado oposto da caixa, na parte interna. Estes equipamentos funcionam de forma parecida com as máquinas de hoje, a diferença sendo que as câmeras atuais usam lentes para captar mais luz e filmes negativos ao invés de papéis fotossensíveis.

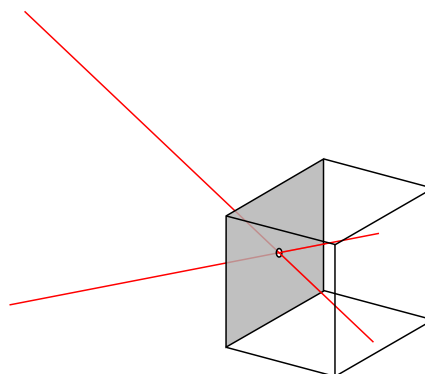


Figura 1.5: Câmera estenopeica

O modelo de câmera estenopeica descreve a geometria envolvida na captação da imagem pela câmera estenopeica e mapeia aonde os pontos tridimensionais do mundo real são projetados

no plano aonde está a fotografia. No caso da estimação de pose, se usa este modelo para fazer o inverso: o ponto bidimensional da imagem é usado para se descobrir aonde o ponto tridimensional no mundo real que ele representa.

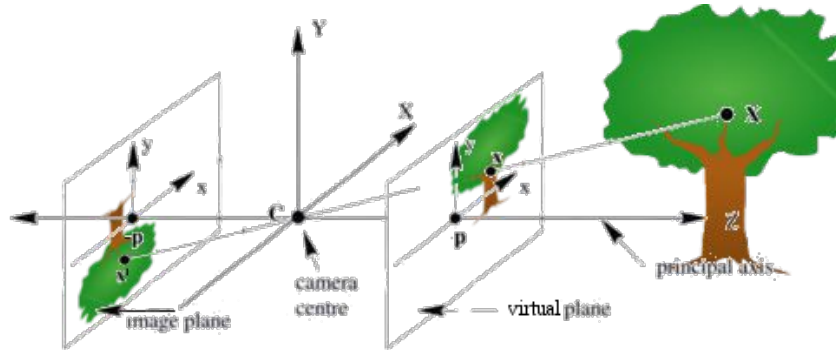


Figura 1.6: Modelo de câmera estenopeica. Se o plano for invertido para a frente do centro, ele conterá a mesma imagem, mas girada em 180°.

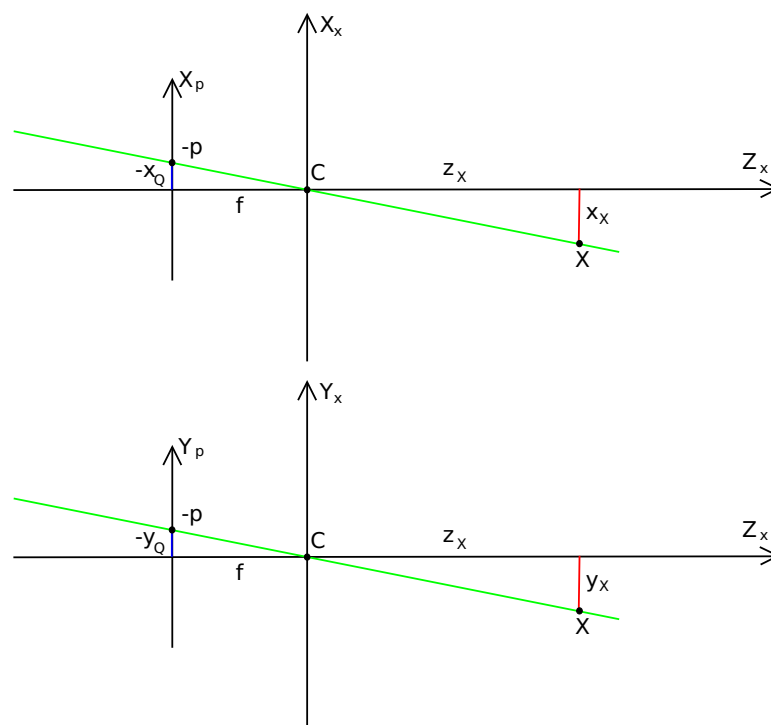


Figura 1.7: Visão sobre o eixo Y e X, respectivamente, mostrando geometria usada para determinar os valores de x e y do ponto no mundo real, a partir de x e y da imagem.

Usando as propriedades de triângulos similares, a partir da figura 1.7 podemos derivar que:

$$\frac{x_q}{f} = \frac{x_x}{z_x} \therefore x_x = \frac{x_q \cdot f}{z_x} \quad \text{e} \quad \frac{y_q}{f} = \frac{y_x}{z_x} \therefore y_x = \frac{y_q \cdot f}{z_x}$$

No entanto, lentes de câmeras costumam distorcer a imagem, fazendo com que os pontos longe do centro fiquem ainda mais longe. Isto faz as bordas da imagem parecerem levemente

torcidas. Como consequência, para se descobrir a projeção de um píxel, é preciso considerar os componentes de distorsão. Existem dois tipos de distorsões: radial e tangencial e estas são representadas pelos parâmetros k_1 , k_2 , p_1 e p_2 . Ao processo de cálculo dos parâmetros intrínsecos da câmera é dado o nome de calibragem da câmera. (Salinas, 2012) Considerando os coeficientes de distorsão c_x e c_y , chegamos a fórmula final:

$$x_{c\grave{a}mera} = \frac{X \cdot f}{z} + c_x \quad y_{c\grave{a}mera} = \frac{y \cdot f}{z} + c_y$$

Uma vez que $x_{c\grave{a}mera}$, $y_{c\grave{a}mera}$, f , c_x e c_y são valores conhecidos, chamados de parâmetros intrínsecos na terminologia específica - pois são inerentes à câmera e constantes -, descoberto o valor do eixo Z do marcador é possível determinar os valores dos eixos Y e Z. Em contraste, às coordenadas do marcador se dá o nome de parâmetros extrínsecos, pois seus valores não são inerentes a câmera e variam de acordo com o movimento tanto da câmera quanto do marcador.

O valor do deslocamento pelo eixo Z do marcador é descoberto através da comparação de seu tamanho real e de seu tamanho na imagem de onde se calcula sua distância da câmera, ou seja: o deslocamento do eixo Z. Feito isto é possível resolver a fórmula.

1.3.3 Codificação dos marcadores

Os marcadores usados são padrões planos e retangulares - como os da figura 2 - e possuem um código interno dado por 5 palavras de 5 bits cada. A codificação emprega um código de Hamming levemente modificado em que cada palavra tem apenas 2 bits de informação dos 5 bits usados. Como consequência, podem ser gerados 1024 códigos identificadores diferentes. Os outros 3 bits são usados para detecção de erro.

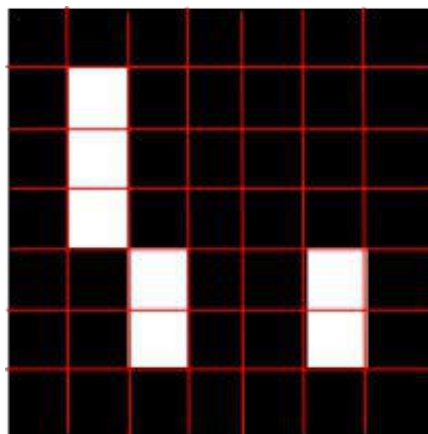


Figura 1.8: Código usado nos marcadore

A principal diferença para o código de Hamming é que o primeiro bit (paridade dos bits 3 e 5) é invertido. Assim, o código 0 (cujo código de Hamming é 00000) se torna 10000 nesta codificação. A idéia por trás disto é prevenir que um retângulo preto seja um marcador válido, com o objetivo de reduzir a probabilidade de falsos positivos em objetos do ambiente. (Salinas, 2012)

1.3.4 Tabuleiros

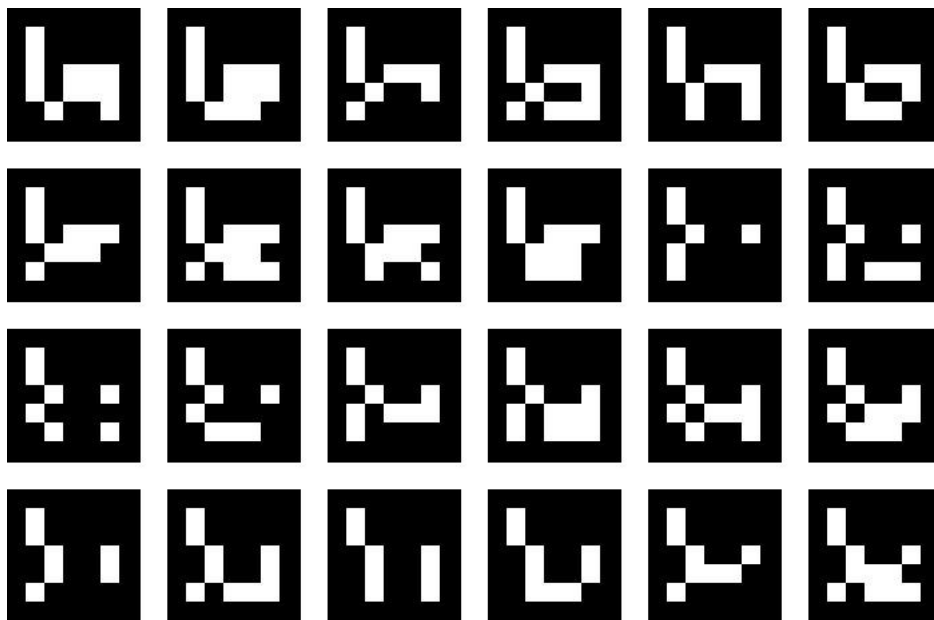


Figura 1.9: Exemplo de tabuleiro de marcadore

Um problema encontrado no uso de marcadores é a perda da detecção se qualquer parte sua, por menor que seja, for tampada. Uma solução para este problema é o uso de vários marcadores engradados, chamados tabuleiros de marcadores.

1.4 Renderização

Renderização é o ato de se desenhar na tela os objetos de uma cena tridimensional. Neste trabalho, este processo foi feito com o uso de uma biblioteca gráfica com alto nível de abstração para desenvolvimento de aplicações 3D em tempo real conhecida pelo seu nome em inglês *rendering engine*, motor de renderização em português (Junker, 2006), neste trabalho ela será referenciada apenas como renderizador.

Por abstrair as operações geométricas necessárias para posicionar e movimentar objetos em uma cena, ela facilita o desenvolvimento de aplicações.

1.4.1 Grafo de cena

O grafo de cena é a forma mais encontrada nos renderizadores atualmente para representar a cena exibida na tela. Os objetos da cena se comportam como nós de um grafo, sendo que nós podem ter filhos, que neste caso acarreta na propagação de comportamentos e estados dos pais para os filhos.

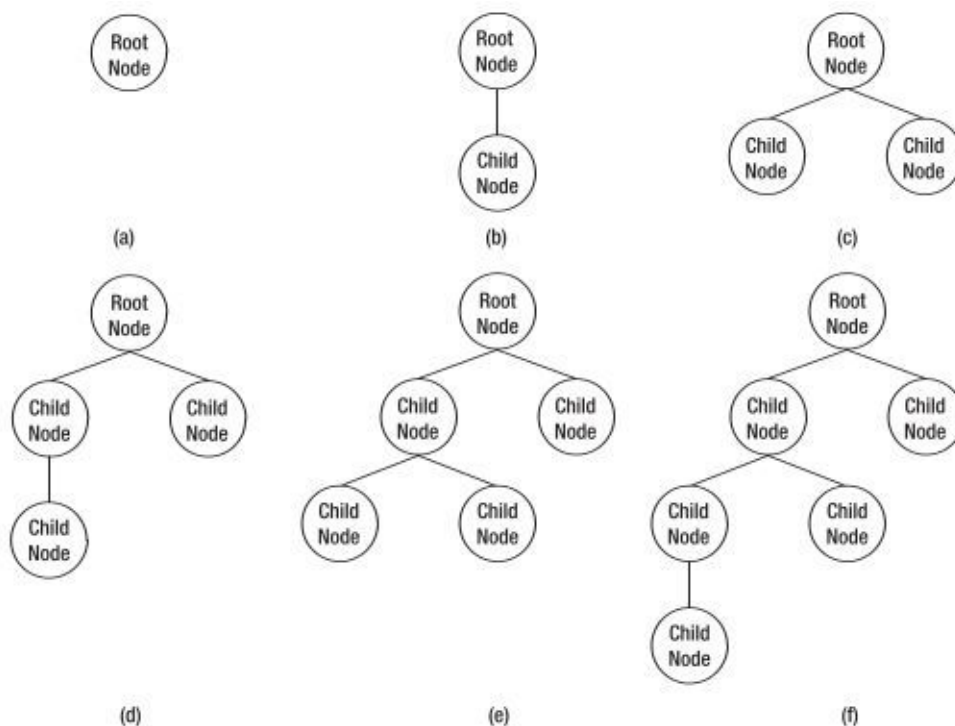


Figura 1.10: Comportamento de um grafo de cena a medida que nós são adicionados a ele

Os nós filhos possuem dois sistemas de coordenadas: um absoluto, em relação à origem da cena e outro relativo, em relação à posição do nó pai. Isto faz com que sempre que o nó pai se movimenta, os nós filhos, filhos de filhos e assim por diante, se movimentem também. Caso o nó pai tenha alguma rotação, os nós filhos rodarão também para manter a mesma posição relativa ao nó pai.

1.4.2 Câmera

A câmera é uma abstração usada para representar o ponto de vista da cena que deve ser mostrado pelo monitor. Ela também é tratada como um nó do grafo de cena, útil em situações

como seguir um personagem pelo cenário: a câmera pode ser um nó filho do personagem na cena e o seguirá automaticamente.

Normalmente ela fica posicionada na cena de forma análoga à uma câmera de um set de filmagens, no entanto, quando usada para realidade aumentada, ela assume uma posição fixa na origem do sistema de coordenadas e apenas os objetos se movem, mesmo quando no mundo real é ela que se move. Isso acontece desta forma por causa de peculiaridades da realidade aumentada que serão explicados em capítulos posteriores do trabalho.

Outra característica da câmera usada na realidade aumentada é sua matriz de projeção. Ela é uma matriz com informações sobre o campo de visão da câmera, como distâncias mínimas e máximas de visão, ângulo de visão e proporção da resolução da tela.

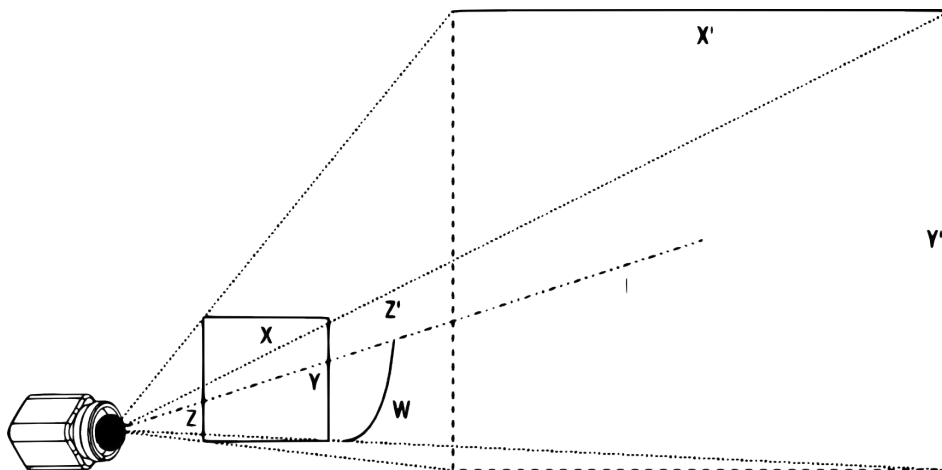


Figura 1.11: Tronco de visão ou, em inglês como é mais conhecido *view frustum* e planos da distância máxima e mínima da visão

1.4.3 Materiais

Os materiais controlam como os objetos serão desenhados na cena. Basicamente toda aparência de um objeto, fora seu formato, é controlado pelo seu material. Os parâmetros do material incluem componentes de cor emissivos, ambiente, difuso e especular. (Streeting, 2010)

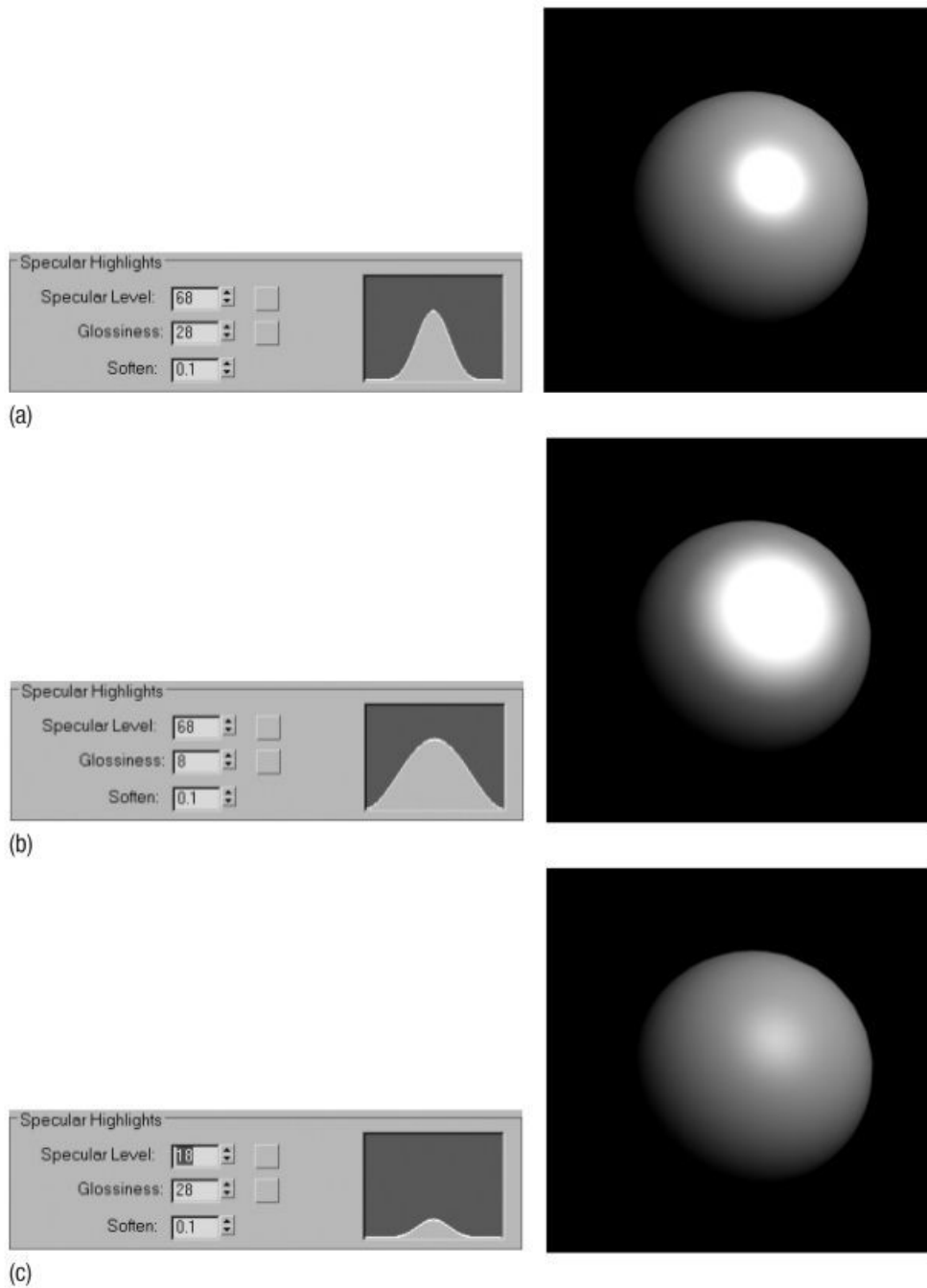


Figura 1.12: Diferença da aparência do objeto de acordo com o nível especular, brilho e suavidade de seu material. Fonte: (Junker, 2006)

A cor emissiva representa luzes emitidas pelo próprio material ao invés de simplesmente refletidas. As cores ambiente, difusa e especular são termos que interagem com as fontes de luz. O brilho é usado para controlar quão definida ou difusa uma luz especular é. O valor alfa é usado para suportar materiais transparentes como uma alternativa a aplicar texturas com um canal alfa. (Eberly, 2007)

1.4.4 Sombras

Sombras são claramente uma parte importante na composição de uma cena verossímil - elas fornecem uma sensação de tangibilidade aos objetos na cena, e ajudam o expectador a entender as relações espaciais entre os objetos (Streeting, 2010).

Essa característica se torna ainda mais importante dentro da realidade aumentada, pois a interação com os objetos virtuais acontece com a manipulação de objetos físicos do mundo real.



Figura 1.13: Você consegue dizer quão alto a estátua de Atena está sobre o chão na figura d cima? E na de baixo? Fonte: Junker, 2006.

Implementações de sombras se enquadram basicamente em duas categorias: “sombras baseadas em estêncil” e “sombras baseadas em texturas”. Esta é a classificação de acordo com o método pelo qual o formato da sombra é gerado. Além disto, existe mais de uma maneira de desenhar a sombra na cena: sombras modulativas, que escurecem a cena nas áreas de sombra, e máscara aditiva de luz, que em contraste cria contribuições de luz nas áreas que não possuem sombra (Streeting, 2010).

1.5 Simulação física

A física é uma ciência ampla que envolve coisas tão distintas quanto quântica e eletromagnetismo. No entanto, normalmente quando se fala de simulação física em computação gráfica - inclusive neste trabalho - nos referimos apenas à mecânica clássica: leis que governam como objetos se movem sob influência da gravidade e outras forças (Millington, 2007).

Esta simulação de movimento dá aos objetos a sensação de serem sólidos, com massa, inércia, elasticidade e fluabilidade (Millington, 2007). É justamente esta sensação que torna o comportamento dos objetos realista e que se busca. Afinal, o que torna a realidade aumentada uma forma interessante de interação com o computador é permitir a interação de objetos presentes no mundo real com objetos virtuais (Azuma, 1997) e quanto mais parecido o movimento dos objetos virtuais for com o de objetos reais, maior será a sensação de que eles coexistem em um mesmo ambiente.

A simulação física neste trabalho foi feita pelo uso de uma biblioteca de simulação física, este tipo de biblioteca é conhecido como motor de física ou, como é mais comum, engine de física. Ela simula o movimento dos objetos através de duas atividades: detecção de colisão e dinâmica de corpos rígidos, que serão explicadas nas subseções a seguir.

1.5.1 *Engine física*

Apesar do uso da física dentro da computação gráfica já acontecer há mais de trinta anos, houve uma mudança distinta nos últimos tempos na maneira como a física é implementada: originalmente cada efeito era especificamente programado, criando um jogo somente com a física necessária para ele. Se o jogo precisasse que flechas percorressem trajetórias, então a equação da trajetória seria programada no jogo. Seria inútil para simular qualquer coisa além da trajetória de flechas, mas seria perfeito nisto.

Para simulações simples - quando o tamanho do código é pequeno e o escopo da física é

bem limitado - isto é adequado, mas quando a complexidade aumenta, se torna difícil se chegar diretamente a um efeito físico verossímil. A dificuldade de fazer o movimento parecer real, combinada com a necessidade do uso dos mesmos efeitos recorrentemente encorajou desenvolvedores a procurar por soluções genéricas que pudessem ser reusadas.

Tecnologias reutilizáveis precisam ser bem gerais: um simulador balístico que lidará apenas com flechas pode ter o comportamento de flechas embutido nele. Se o mesmo código precisar lidar com balas também, então o programa precisa abstrair o comportamento particular de cada projétil e simular a física que todos eles tem em comum. É a isto que se dá o nome de *engine* física: uma peça comum de código que sabe sobre física em geral, mas não está programado para as especificidades do cenário de cada jogo.

Se antes havia um código específico para simular uma flecha, nada mais era preciso para simular a flecha. Já com um engine de física genérico que simula qualquer projétil, quando for preciso simular uma flecha é necessário especificar as características do que precisa ser simulado. São necessários propriedades físicas da flecha, ou bala, ou caixas e assim por diante.

Esta é uma distinção importante, o engine de física é basicamente uma grande calculadora: ele lida com a matemática necessária para simular a física, mas não sabe o que precisa ser simulado. Ele precisa ser alimentado com dados específicos para cada objeto que se deseje representar.

Há duas grandes vantagens de usar um engine de física: primeiro a economia de tempo, já que as funções fornecidas por uma engine abstraem os algoritmos de simulação física. A segunda vantagem é a qualidade.

1.5.2 Detecção de colisão

“Dois corpos diferentes não podem ocupar o mesmo lugar no espaço e ao mesmo tempo” explicou Aristóteles há muito tempo atrás. Grande parte da nossa concepção da realidade ser como é se deve a este princípio: o fato de que podemos ficar em pé e caminhar depende do chão ser impenetrável, por exemplo.

Na simulação física impedir que objetos se penetrem quando entram em contato é fundamental para que objetos interajam entre si de uma maneira realista.

Se nós quisermos que um ambiente simulado se comporte de acordo com o mundo real com respeito à impenetrabilidade de objetos materiais, precisamos incorporar um mecanismo que imponha esta restrição. É justamente a detecção das colisões a questão fundamental na

implementação de restrições de impenetrabilidade na simulação do movimento de objetos.

1.5.3 Dinâmica de corpos rígidos

Alem de impedir que os objetos se penetrem, quando se detecta uma colisão é preciso que o movimento dos objetos seja influenciado por ela de forma plausível. Esta é uma das questões tratadas pela dinâmica de corpos rígidos, sistema usado neste trabalho para a simulação. Este sistema define que os objetos jamais se deformam, mesmo quando na realidade isto aconteceria, como quando se chocam com força um contra o outro.

Esta simplificação diminui o número de parâmetros usado para a simulação do movimento dos objetos enquanto mantem grande realismo, o que é importante quando a simulação é usada em conjunto com a realidade aumentada, já que precisa ser exibida em tempo real e o reconhecimento dos quadros do vídeo feito pela realidade aumentada já é intensivo e demorado.

O estudo do movimento destes objetos rígidos leva o nome de dinâmica, de onde vem o nome dinâmica de corpos rígidos. Usando equações de movimento derivadas das leis de movimento de Newton ou da mecânica Lagrangiana, o movimento do objeto é calculado como uma função de tempo, levando em conta massa, torque e forças externas aplicadas à ele - como gravidade e atrito (Erleben et al., 2005; Ettl, 2006). A dinâmica de um sistema de corpos rígidos é definido pelas suas equações de movimento e a solução delas define como a posição dos objetos muda como uma função de tempo.

2 *Revisão sistemática da literatura*

Normalmente as revisões de literaturas em monografias são feitas de forma improvisada e sem nenhum método. O problema dessa abordagem é que os resultados da revisão podem ser enviesados pelo interesse e expectativas de quem a está realizando. Além disto a revisão feita não poderia ser replicada e verificada por outras pessoas (Kitchenham, 2004).

Para resolver este tipo de problema, este trabalho fará uma revisão da literatura de forma sistemática, baseada em uma metodologia rigorosa e consideravelmente formal proposta por Kitchenham (2004). Este método define uma série de etapas e atividades para identificar, avaliar e interpretar toda pesquisa relevante ao trabalho de forma justa e não de acordo com os interesses do autor.

2.1 **Protocolo**

O protocolo da revisão da literatura é a especificação da forma que se pretende obter os artigos que serão revisados. A primeira parte deste protocolo é a elaboração das perguntas que se gostaria de responder com estes artigos. Neste trabalho estas perguntas são:

1. Qual o estado da arte na realidade aumentada?
2. De que forma é possível melhorar a interação via realidade aumentada?
3. Quais bibliotecas de realidade aumentada estão disponíveis no mercado?
4. Quais técnicas de rastreamento e estimação de pose são mais apropriadas para este trabalho?
5. O que é necessário para integrar simulação física à realidade aumentada?
6. Quais bibliotecas de simulação física e de realidade aumentada já foram integradas?
7. Quais benefícios justificam a integração de simulação física à realidade?

8. O que é necessário para integrar realidade aumentada à uma engine gráfica?
9. Quais engines gráficas e bibliotecas de realidade aumentada já foram integradas?
10. Quais benefícios justificam a integração da realidade aumentada à uma engine gráfica?

A partir destas perguntas foram escolhidos os seguintes critérios de busca:

1. “augmented reality” physics
2. “augmented reality” “physics engine”
3. “augmented reality” “rendering engine”
4. “augmented reality” tracking
5. “augmented reality” library

Usando estes critérios, foram consultadas as bibliotecas de artigos científicos virtuais IEE-EXplore, Google Scholar, Association for Computing Machinery e CiteSeerX. Para a escolha das publicações dentre os resultados encontrados nestas bibliotecas foram usados os seguintes critérios:

1. Publicações que abordem simulação física e realidade aumentada
2. Publicações que abordem uso de engine gráficas em realidade aumentada
3. Publicações que abordem técnicas de computação gráficas dentro da realidade aumentada
4. Publicações que detalhem outros assuntos úteis para este trabalho
5. Número de citações. Este critério não foi usado diretamente, mas como forma de ordenação dos resultados, o que certamente também influencia em sua escolha.

Além dos artigos encontrados nas bibliotecas virtuais, artigos citados que atendam aos critérios de seleção também serão usados.

2.2 Detecção do mundo real

Quanto ao estado da arte, foi encontrada a detecção de imagens pela extração de características na biblioteca SSTT (Sörös et al., 2011), Studierstube (Schmalstieg et al., 2002) e BazAR

(V. Lepetit, Fua, 2004). Na biblioteca MRT (Freeman, Steed, 2004) foi encontrada a detecção de objetos reais tridimensionais, dispensando o uso de marcadores. Na biblioteca PTAM (Klein, Murray, 2007) foi encontrada a estimativa da pose da câmera sem a necessidade de se conhecer previamente qualquer elemento do mundo real, através de rastreamento e mapeamento paralelos. Na biblioteca HandyAR (Lee, Hollerer, 2007) e no trabalho de Liu et al. (2011) foi encontrado o rastreamento de mãos.

Haller (2004), Chae, Ko (2008) e Beaney, MacNamee (2009) usaram a biblioteca ARToolkit para integrar simulação física à realidade aumentada. Pela quantidade de trabalhos a usando, há indícios de que esta biblioteca é apropriada para ser usada para este tipo de atividade. No entanto, há de se considerar que os artigos são relativamente antigos e outras alternativas podem ter surgido desde então. O trabalho mais recente de simulação física em realidade aumentada de Liu et al. (2011) usa ARToolkitPlus.

Haller (2004) fala sobre uma forma muito simples e sagaz de se conseguir detectar objetos reais tridimensionais e com volume em realidade aumentada usando rastreamento de marcadores planos: basta mantê-los fixos em relação a um marcador e definir seu volume manualmente.

2.3 Renderização

Haller (2004) afirma que o uso de sombreado especular e difuso pode ser usado para chamar a atenção ao objeto virtual, que o posicionamento da fonte de luz é um fator chave para se ter bons resultados no uso deste recurso e que usou a técnica de sombreado de Blinn em conjunto a texturas.

O uso de sombras foi mencionado por Haller (2004) e Beaney, MacNamee (2009). Os dois trabalhos concluíram que elas melhoram a percepção espacial dos objetos virtuais e conseqüentemente tornam mais fácil sua manipulação. Haller (2004) preferiu o uso de um algoritmo de volume de sombras ao invés de mapa de sombras. Além disto, propôs quatro tipos diferentes de projeção de sombras na realidade aumentada: de objetos reais em objetos reais, de virtuais em virtuais, de reais em virtuais e finalmente de virtuais em reais e também formas de torná-los mais realistas.

Haller (2004) ainda cita outros recursos de computação gráfica como mapeamento de imperfeições (bump mapping), sombreado de desenho (toon shading) e renderização em forma de estilos de pintura. Todos estes recursos são oferecidos por engines gráficas de forma fácil, e parecem justificar a integração de uma junto à realidade aumentada.

Outro aspecto importante para a renderização dos objetos virtuais é seu realismo funcional, encontrado quando o objeto virtual fornece uma informação visual igual a dos objetos reais. Por exemplo: os objetos são iluminados pela mesma direção e assim suas sombras projetadas da mesma forma (Ferwerda, 2003).

Foi constatado o uso da biblioteca OpenGL por Beaney, MacNamee (2009) e Liu et al. (2011) e da biblioteca OSG por Chae, Ko (2008) para realizar a renderização.

2.4 Simulação física

Para Chae, Ko (2008) o realismo de um objeto virtual pode ser avaliado em relação à sua aparência e comportamento, já (Ferwerda, 2003) propões além destes dois aspectos, o realismo funcional. No entanto, estas duas classificações concordam que o comportamento do objeto, ou seja, a forma que ele se movimenta quando se interage com ele é fundamental para uma experiência satisfatória da realidade aumentada.

Para resolver esta questão, Haller (2004), Chae, Ko (2008), Beaney, MacNamee (2009) e Liu et al. (2011) adotaram bibliotecas de simulação física para simular o comportamento dos objetos virtuais, dando a impressão de que há bastante interesse científico nesta linha de pesquisa. Haller (2004) e Chae, Ko (2008) usaram a biblioteca Open Dynamics Engine, enquanto Beaney, MacNamee (2009) e Liu et al. (2011) usaram a biblioteca Bullet.

Para que seja possível simular os objetos virtuais Chae, Ko (2008) definiram um sistema de coordenadas global através das coordenadas dos marcadores, mas não entraram em detalhes de como fazer isto. Beaney, MacNamee (2009) também menciona que a principal questão para a integração é o uso de um sistema de coordenadas comum entre a simulação física e detecção de realidade aumentada. Liu et al. (2011) afirma que as principais tarefas da simulação física são a detecção de colisão, resolução de restrições e atualizar a representação de mundo para os objetos que se movimentaram.

Um problema encontrado pelo uso da simulação física em realidade aumentada foi o aparecimento de artefatos visuais entre os objetos reais e virtuais, que causa diminuição do realismo dos objetos virtuais (Kim et al., 2011).

3 Reconhecimento do mundo real

3.1 Escolha da biblioteca de realidade aumentada

Existem várias bibliotecas de realidade aumentada disponíveis, entre elas ARToolkit e ARToolkitPlus - as mais usadas - e outras que apresentam recursos do estado da arte da tecnologia, como detecção de características naturais - o caso da Vuforia, localização e mapeamento simultâneos - como a PTAM ou rastreamento de mãos - caso da HandyAR.

Todas estas bibliotecas foram testadas, mas foi constatado que as mais usadas estão abandonadas e seu processo de detecção é arcaico, enquanto as que estão próximas do estado da arte são altamente instáveis e prototípicas.

Além disto, o foco deste trabalho não está na melhora da detecção dos objetos reais, mas na melhora do realismo dos objetos virtuais. É preciso levar em conta que a realidade aumentada usando a técnica de localização e mapeamento simultâneos implementada na PTAM não permite a detecção de objetos que se movem, mas somente de um plano que permanece imóvel. Isto torna impossível o uso de objetos reais para interagir com virtuais, que é uma dos objetivos buscado neste trabalho.

Assim, foi escolhida uma biblioteca chamada ArUco, que entre todas as bibliotecas vistas, era a única baseada em OpenCV, uma biblioteca de visão computacional que é padrão de facto na área, possui uma ampla documentação, muitos tutoriais e grande suporte. Todo o processo de detecção dos marcadores na biblioteca de realidade aumentada é representado por chamadas de métodos à biblioteca OpenCv, isto torna fácil seu entendimento e experimentação, pontos importantes para a pesquisa científica.

Além disto, esta biblioteca é a única que possui métodos que facilitam a integração com a biblioteca gráfica, especificamente a geração de uma matriz de projeção que pode ser usada pela biblioteca de renderização.

3.2 Calibragem da câmera

A biblioteca de visão computacional OpenCv permite a obtenção dos parâmetros intrínsecos à câmera de forma fácil. Só é preciso mostrar imagens de um tabuleiro de xadrez impresso em uma folha e informar suas dimensões.

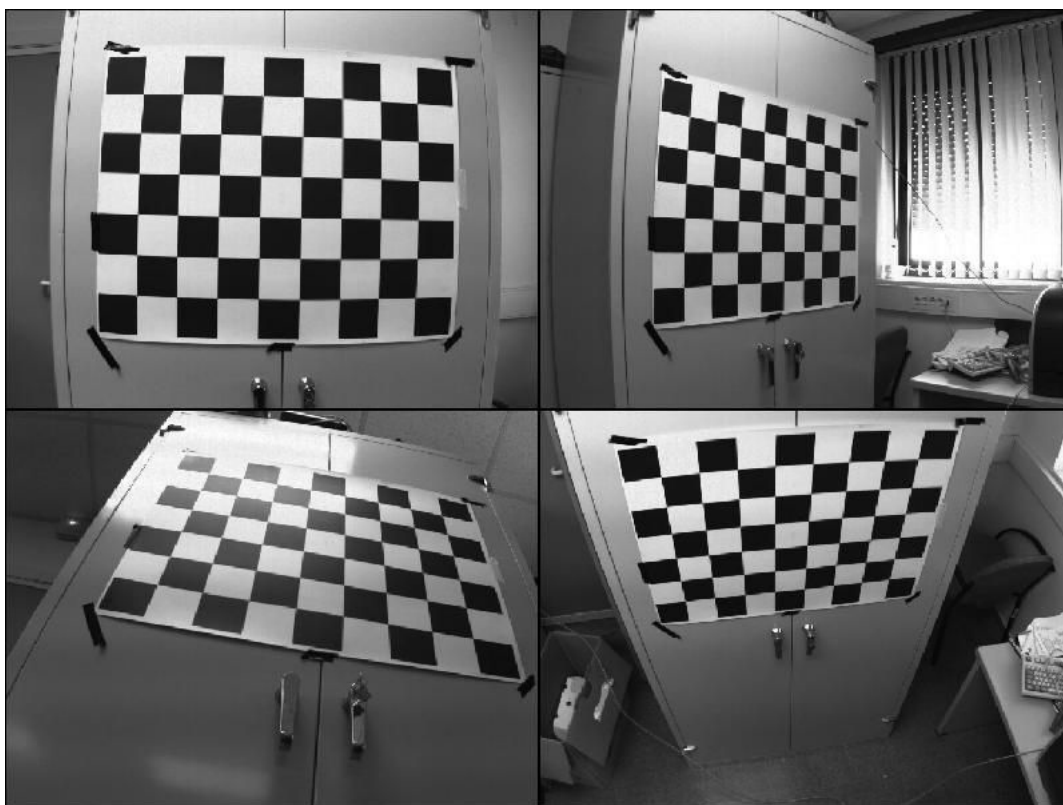


Figura 3.1: Exemplo de ângulos das fotos usadas para se calibrar a câmer

Devem ser tiradas ao menos fotos de cinco ângulos diferentes, uma frontal, uma à esquerda, uma à direita, uma com a câmera mais acima e a última com a câmera mais abaixo. Fotografando o padrão por estes ângulos, é possível estimar a distorsão da câmera com precisão. Na versão 2.2 da OpenCV, encontra-se tal tabuleiro para ser impresso no subdiretório "doc/pattern.pdf".

Uma vez tiradas as fotos com a câmera que será usada, usa-se a aplicação "samples/cpp/calibration.cpp" - fornecida em versões da OpenCV superiores à 2.2 - para calibrar a câmera. É preciso informar o número de cantos do padrão em seus dois eixos, e o tamanho real dos quadrados em metros. O programa é executado da seguinte forma:

Como saída, o programa gera um arquivo com os parâmetros intrínsecos, que pode ser usado pela biblioteca de realidade aumentada ArUco (Salinas, 2012).

3.3 Remoção da distorsão da câmera

O arquivo gerado pela calibragem da câmera é carregado dentro do programa e se obtém uma matriz de visão e uma matriz de distorsão da câmera com sua informação. A matriz de visão contém o campo de visão da câmera e a matriz de distorsão parâmetros para remover a distorsão gerada pela lente da câmera.

A biblioteca de realidade aumentada trazia em seus exemplos um método que criava bordas pretas quando a distorsão era removida da imagem. No entanto, foi descoberto que existia uma forma alternativa que remove a distorsão sem deixar bordas pretas nela. Para isto, é usada um método da biblioteca de visão computacional OpenCV que a partir da matriz da câmera e da matriz de distorsão calcula uma nova matriz de câmera com o maior campo de visão possível em que as bordas pretas não aparecem. Como consequência o campo de visão é um pouco menor e a imagem aproximada um pouco, mas ainda assim isto é muito melhor que uma imagem com bordas pretas.

3.4 Detecção dos marcadores

Foi experimentado a troca do algoritmo de estimação de pose da biblioteca ArUco por um do estado da arte, chamado estimação de pose robusta para alvos planos (Schweighofer, Pinz, 2006). No entanto não foi verificada melhora significativas na detecção e foi decidido manter o algoritmo original.

3.5 Estabilização da detecção

Durante o teste de protótipos, foi percebido o fenômeno da perda da localização dos tabuleiros quando a câmera se move rapidamente. Para tentar resolver este problema, a estratégia usada foi, quando a localização for perdida, manter a última localização detectada durante algum tempo, até que uma nova localização seja detectada. Foi constatado que esta estratégia

melhora bastante o efeito de que o tabuleiro está “piscando” e o melhor valor de tempo para se manter a última localização detectada encontrado foi o de 1 segundo.

3.6 Limitações

Durante o uso dos protótipos foi constatada uma limitação do uso da realidade aumentada que deve ser observadas e evitadas para que a interação não seja comprometida por problemas de reconhecimento do mundo real.

Esta limitação é o uso de marcadores móveis: quando os marcadores se movem em relação à câmera eles simplesmente não são bem detectados e ocorre grande tremulação em seu reconhecimento. Recomenda-se o uso somente de marcadores fixos, que tem objetivo de reconhecer o mundo real e estabelecer o sistema de coordenadas e não se usar marcadores para manipulação de objetos virtuais. A interação deve ocorrer somente entre objetos virtuais.

4 *Renderização*

4.1 *Escolha da engine de renderização*

Apesar de existirem bibliotecas gráficas que já possuem integração com bibliotecas de realidade aumentada, especificamente a OsgART, biblioteca que integra a biblioteca gráfica OSG com a biblioteca de realidade aumentada ARToolkit ou ARToolkitPlus, foi constatado que estas bibliotecas de realidade aumentadas não são as mais interessantes para este trabalho e que existem bibliotecas gráficas melhores que a OSG. Por isto, a biblioteca gráfica foi escolhida baseada em suas qualidade técnicas, documentação, manutenção e comunidade. Através destes critérios foi escolhida uma biblioteca gráfica chamada OGRE.

Outra vantagem desta engine gráfica é que ela possui uma versão que usa OpenGL ES 3 e torna as aplicações desenvolvidas com ela usáveis em aparelhos portáteis sem a necessidade de se modificar o código. Aparelhos portáteis são as plataformas mais apropriadas para o uso da realidade aumentada, pois o aparelho se move junto com o usuário, dando a impressão de que se está usando uma espécie de lente mágicas que mostra objetos escondidos no mundo real e torna a interação mais realista.

4.2 *Integração com realidade aumentada*

O primeiro lugar aonde a informação é atualizada é no mundo da visão computacional. Aqui a imagem da câmera é capturada e através de técnicas de visão computacional, e os marcadores que aparecem na imagem são reconhecidos e sua posição no mundo real encontrada. É importante notar que está havendo uma transformação de uma representação bidimensional, a imagem, para uma tridimensional, que são a posição e orientação dos marcadores. Quando se fala em rotação, é preciso entender que além de estarmos trabalhando com a posição tridimensional, que é dada pelo deslocamento em cada eixo, estamos trabalhando também com a orientação do plano que está sendo identificado no mundo real, que também pode ser representada em termos dos eixos X, Y e Z, daonde a informação sobre a posição e orientação de

um objeto se diz ter seis graus de liberdade para se movimentar. No entanto, a representação usada para a orientação pelo algoritmo de estimação de pose da biblioteca de realidade aumentada é através de quatérnios, que mostra a orientação do objeto dizendo a rotação do objeto em um vetor qualquer e possui várias vantagens sobre outros tipos de representação de orientação.

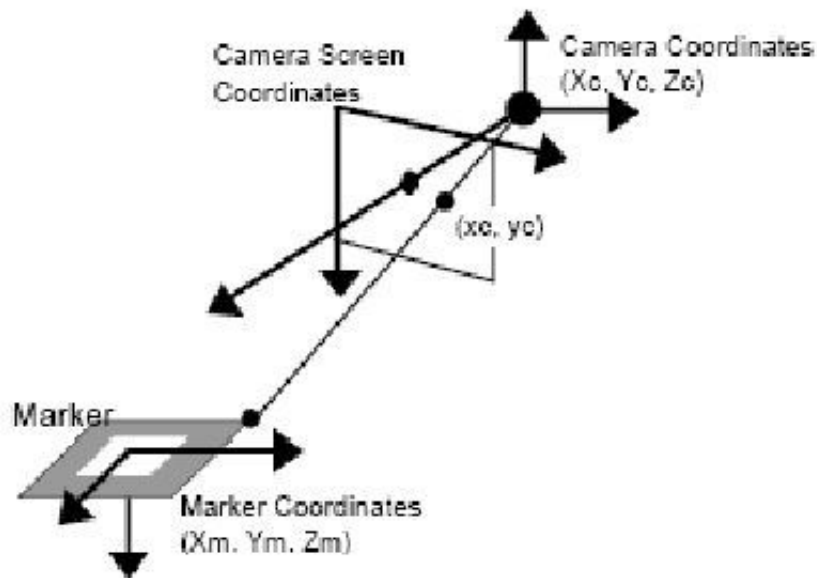


Figura 4.1: Sistema de coordenadas da biblioteca de realidade aumentada

A forma usada neste trabalho para transportar esta representação da posição das técnicas de visão computacional para a representação de cena da engine gráfica foi configurar a cena da engine de forma que seja idêntica à da visão computacional. Assim, basta atualizar as posições da cena da engine gráfica com os mesmos valores da visão computacional. Na representação dos objetos pela visão computacional, são sempre os objetos que se movem ao redor da câmera, mesmo quando no mundo real é a câmera que está sendo movida. Ou seja, a câmera é fixa e nunca muda de posição. Para entender a explicação do posicionamento da câmera no mundo da engine gráfica, é preciso primeiro entender seu sistema de coordenadas, porque a orientação dela será explicada em termos de eixos de um espaço cartesiano.

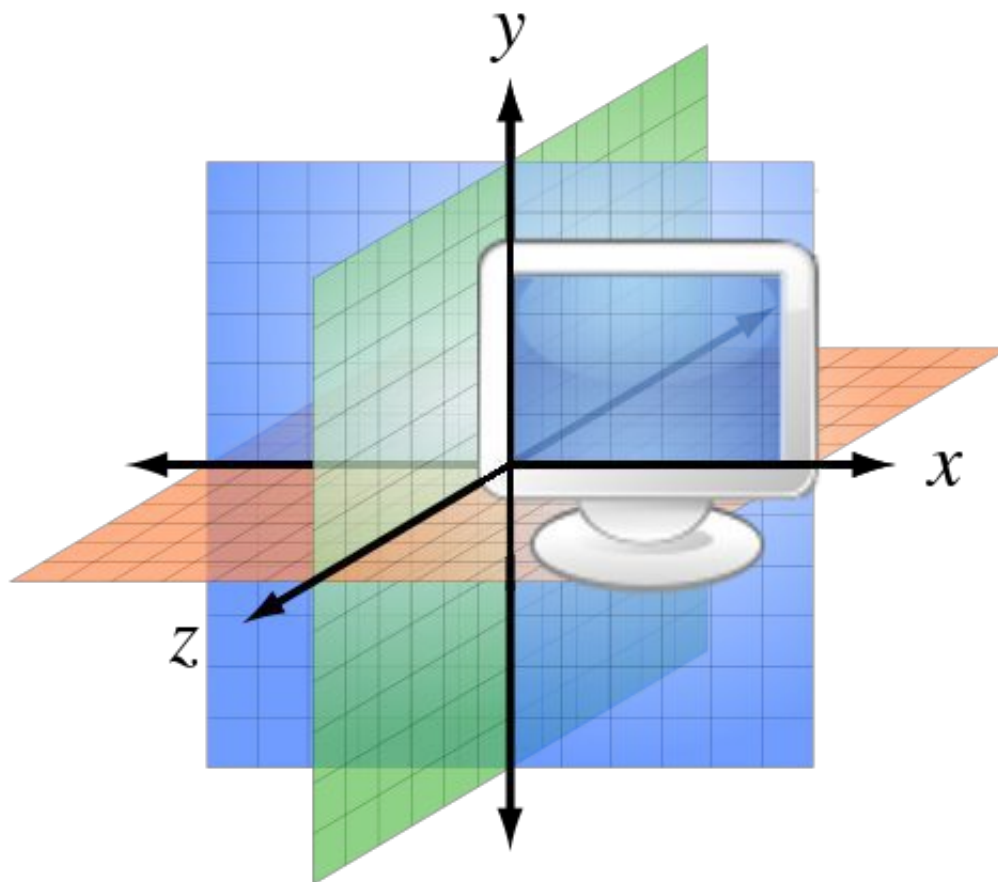


Figure 4.2: Sistema de coordenadas da biblioteca gráfica

Na imagem, a direção positiva é a das setas com as letras dos eixos e a negativa, a das setas sem letras. A câmera deve ser colocada na origem e direcionada de forma que a sua lente aponte na mesma direção que o eixo z, no sentido negativo. Veja que a câmera e sua lente são apenas conceitos para representar objetos virtuais.

Outro problema que precisa ser resolvido é a diferença de escalas dos sistemas de coordenadas da cena da engine gráfica e da biblioteca de realidade aumentada. Felizmente, a biblioteca de realidade aumentada, a partir de suas últimas versões passou a oferecer um método que gera uma matriz de projeção que pode ser usado pelo sistema de coordenadas da engine gráfica para que os pontos e as rotações sejam representadas pelos mesmos valores nos dois sistemas. Além disto, a matriz de visão da câmera da engine precisa ser modificada para uma matriz identidade, uma vez que a ela já se encontra na origem, e seu tipo de projeção de perspectiva para ortográfica.

Desta forma, agora a posição dos marcadores será representada de forma idêntica nos dois sistemas de coordenadas e basta usar a informação dos seis graus de liberdade retornada pelo algoritmo de estimação da pose para representar objetos virtuais com a engine gráfica.

4.3 Projeção de sombras

A sombra é um atributo presente em qualquer objeto real, então naturalmente um objeto virtual com sombra possui um maior realismo. Também deixam o objeto virtual mais integrado ao mundo real, sua sombra interage com o tabuleiro de referência e também com o tabuleiro móvel, como é possível observar na 4.3, aumentando a sensação de sua coexistência no mesmo ambiente físico.

Outra utilidade importante das sombras é ajudar na percepção espacial do objeto virtual. Em experimentos informais, foi constatado que isto torna muito mais fácil determinar precisamente suas posições no mundo real (Beaney, MacNamee, 2009).

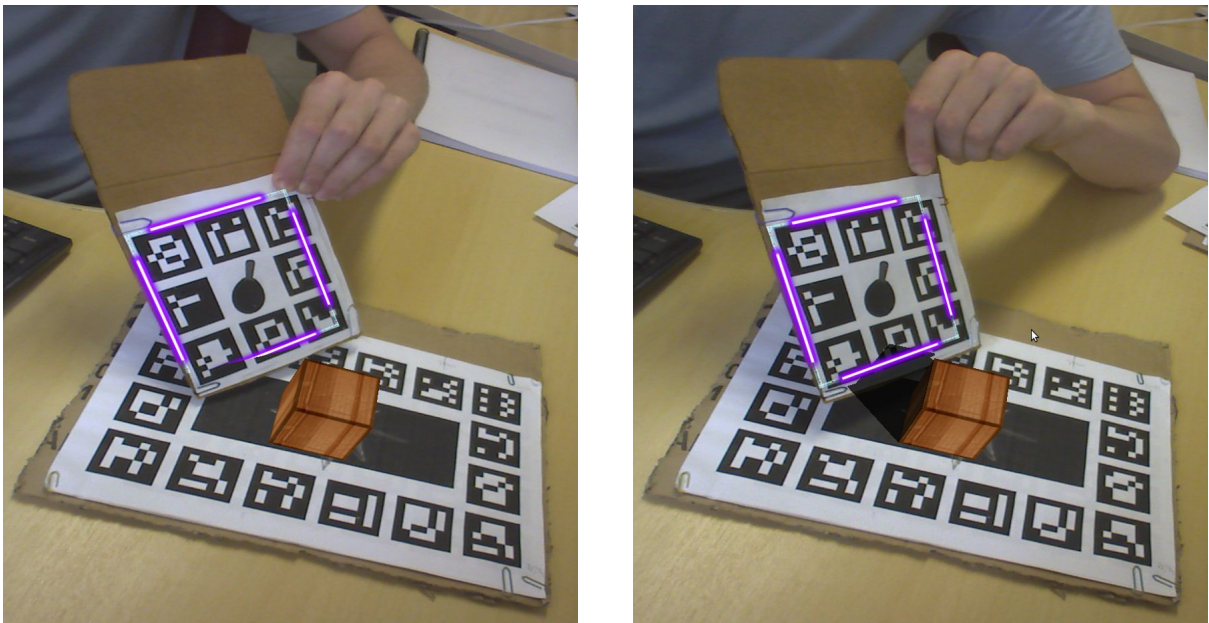


Figure 4.3: Objeto virtual com e sem sombra.

Para que as sombras sejam geradas, é preciso que elas sejam projetadas sobre algum objeto. Colocando sobre o tabuleiro um plano grande o suficiente para alcançar todo o espaço filmado, se cria uma representação para o renderizador do tabuleiro e assim ele consegue fazer com que as sombras sejam projetadas sobre ele e a superfície ao redor onde ele está colocado.

Graças ao uso de um material transparente mas que recebe projeções de sombra nos planos que ficam sobre os tabuleiros, é alcançada a ilusão de que as sombras estão realmente sendo

projetadas sobre os objetos reais que estão sendo filmados. O tipo de sombra usado foi o baseado em estencil e modulativo, por ter sido o único que funcionou corretamente com a realidade aumentada.

4.4 Oclusão de objetos virtuais por reais

Como dito acima, são associados planos com materiais transparentes, mas que recebem projeção de sombra aos marcadores móveis. Este material também tem a propriedade de fazer com que quando eles sejam colocados na frente de objetos virtuais da biblioteca gráfica, eles os tapem, aumentando bastante a sensação de que os objetos reais e virtuais coexistem no mesmo mundo.

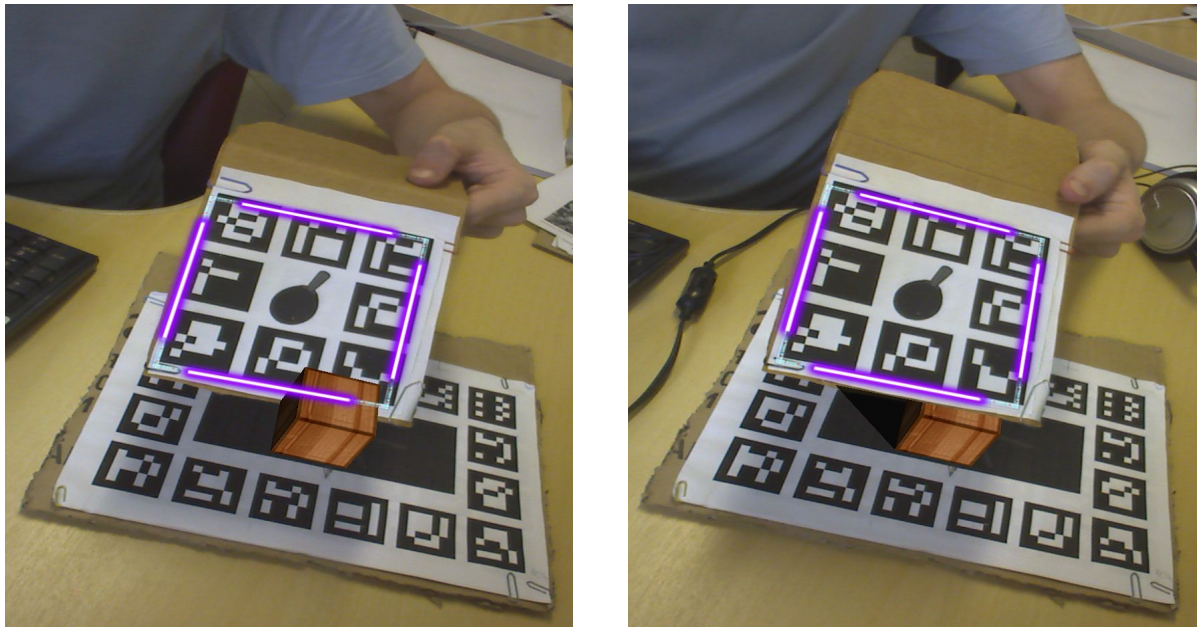


Figure 4.4: À esquerda o objeto real não oclui o virtual. Já à direita, o objeto real oclui o virtual.

5 *Simulação*

A principal contribuição deste trabalho para o aumento do realismo da realidade aumentada é fazer com que os movimentos dos objetos virtuais sejam similares ao dos objetos do mundo real. Para isto foi feita uma simulação que aplica leis da mecânica clássica para calcular o movimento deste objetos, tornando possível estabelecer como será seu movimento dinamicamente para cada ocasião e não com movimentos pré-estabelecidos.

Existem três representações de mundos que precisam ser sincronizadas para serem integradas. Existe a representação do mundo do ponto de vista da visão computacional, a representação do mundo da simulação física e a representação de grafo de cena da engine gráfica.

Cada representação possui diferenças significativas e precisa de todo um processo de transformação para que a informação seja representada de forma correta em três lugares diferentes. Este capítulo explicará como a transformação das coordenadas entre o mundo da simulação física e da renderização pode ser feito.

5.1 Escolha da engine de física

A escolha da engine de física foi feita levando em conta atividade da comunidade, abertura do código, documentação, manutenção de código e qualidades técnicas da engine. As opções consideradas foram Bullet, ODE e PhysX. A engine PhysX possui código fechado, enquanto a ODE não possui seu código atualizado desde 2009 na época em que foi feita a avaliação das engines, além de não ser muito bem documentada. Enquanto isso, a Bullet possui código aberto, atualizado, grande atividade da comunidade e extensa documentação. Por estes motivos, a Bullet foi a engine de física escolhida.

5.2 Integração com a engine de renderização

A abordagem usada para o sistema de coordenadas do mundo físico é radicalmente diferente da usada para a do mundo gráfico. Como o mundo físico age sobre forças representadas por vetores, principalmente da gravidade, foi constatado que é muito mais simples usar um sistema de coordenadas similar ao conceito do nosso mundo real, aonde o a gravidade empurra as coisas para baixo, ou seja, na direção negativa do eixo vertical Y, parando quando chega ao chão, o plano do eixo horizontal X e transversal Z.

Assim, precisamos saber o que é o plano XZ no mundo real e para isso usamos a posição e orientação de um tabuleiro de marcadores que nunca deve ser movido ou girado, pois ele servirá como referência do que é o chão, já que não é possível saber quando a câmera foi movida ou o o tabuleiro.

É importante que durante o uso do aplicativo, a posição do tabuleiro nunca seja perdida, pois a simulação física não poderia ser feita, assim é recomendado o uso de um tabuleiro grande como referência, como assim sua posição é melhor detectada. Assim, o mundo físico tem sua origem no centro do tabuleiro, o eixo vertical X e transversal Z paralelos ao tabuleiro e o eixo vertical Y perpendicular a ele.

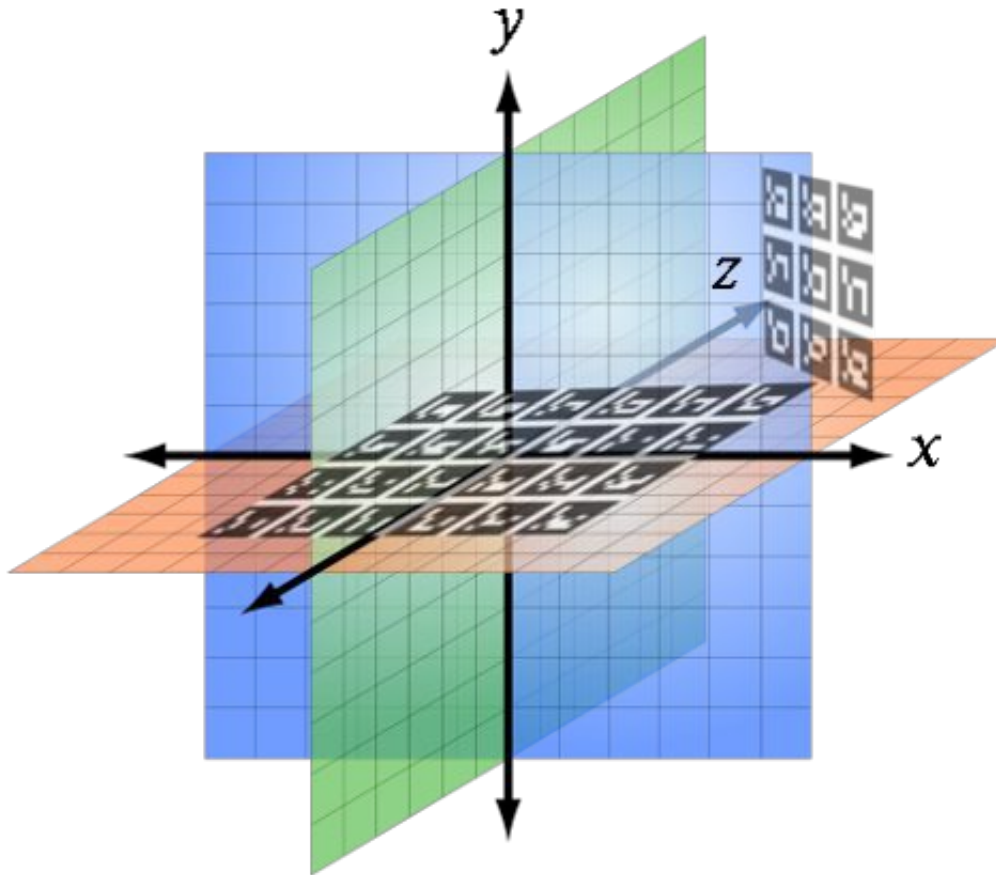


Figura 5.1: Sistema de coordenadas do mundo físico

Felizmente cada nó do grafo de cena já possui um sistema de coordenadas que tem como origem seu centro e, caso seja um plano, perpendicular eixo vertical Y e em cima do plano dos eixos X e Z, exatamente a mesma que a da simulação física. A conversão das coordenadas é feita apenas chamando um método do objeto do nó do grafo de cena para transformar as coordenadas de globais, que são as do mundo gráfico, para locais, que são as do objeto e conseqüentemente do mundo físico, usado para atualizar a posição dos marcadores do mundo gráfico no mundo físico e outro método para transformá-las de locais para globais, para atualizar a posição dos objetos virtuais do mundo físico no mundo gráfico.

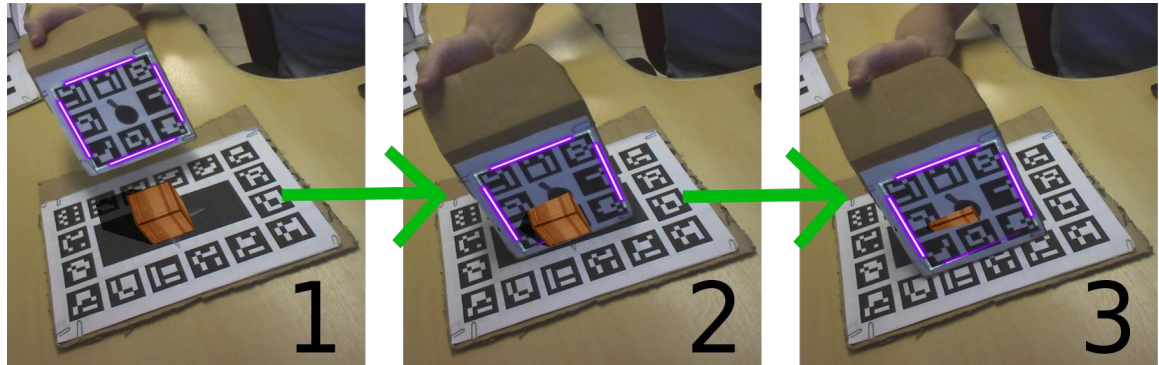
Aqui abre-se um parentêsis para chamar a atenção a como o uso das ferramentas certas tornam muito mais fáceis a tradução de uma teoria conceitual para um protótipo operacional. Para se transformar os sistemas de coordenadas é preciso usar apenas um método da biblioteca gráfica e para fazer o transporte das posições dos objetos virtuais após a simulação física de seus movimentos no mundo físico para o mundo gráfico, não é necessário chamar o método uma

vez para cada objeto que está sendo simulado, mas apenas associar um estado de movimento a cada objeto, durante a fase de criação dos objetos, antes de se entrar no ciclo principal de atualização. Assim, é preciso apenas chamar um método, que faz o mundo físico ser atualizado na quantidade de tempo que passou desde a vez anterior em que ele foi atualizado e todo o processo de atualização dos objetos do mundo gráfico são feitos automaticamente.

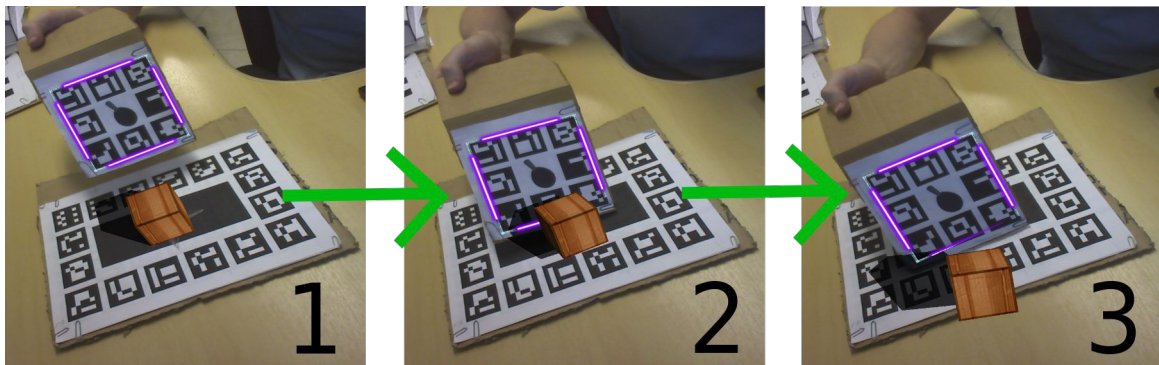
Outras vantagens do uso deste padrão de projeto é o aumento do desempenho por objetos que não se movem no mundo físico não serem atualizados no mundo gráfico e interpolação de movimento automática (Coumans, 2012).

Outra transformação necessário é na escala do sistema de coordenadas. O simulador físico não trabalha bem com número pequenos, recomenda-se que eles variem de 0,05 para uma bolinha de gude a 10 para um caminhão (Johnson, 2008). No entanto os valores do mundo da visão computacional estão na casa dos milésimos, e provavelmente são medidas em metros. Empiricamente se constatou que o dimensionamento que traz melhores resultados é o em três ordens de magnitude, ou seja, aumentar os valores em 100 vezes. Seria como transformar uma medida de metros em centímetros.

5.3 Comparação do comportamento dos objetos antes e após uso da simulação física



(a) Sem simulação física



(b) Com simulação física

Figura 5.2: Comparação entre interação de objetos com e sem uso de simulação física de seu movimento.

6 *Conclusões*

Foi chegada a conclusão que os objetivos propostos por este trabalho foram largamente cumpridos. A escolha de uma biblioteca gráfica para a apresentação dos objetos virtuais se mostrou muito acertada, pois seu alto nível de abstração tornou a integração com a realidade aumentada e a simulação física bastante mais simples.

Além disto os recursos gráficos que ela oferece, como sombras, sombreamento, luzes e uso de materiais são fáceis de serem usados e realmente deram aos objetos uma impressão maior de estarem presentes no mundo real. Outra possibilidade aberta pelo uso da biblioteca gráfica, é o uso da aplicação em aparelhos portáteis, que também aumentam a sensação dos objetos virtuais realmente existirem.

A simulação física também foi bem sucedida e o movimento dos objetos virtuais obedece às leis da física newtoniana e às colisões com objetos reais. Isto torna o comportamento dos objetos reais e virtuais muito mais previsível e realista, e por isto a interação com eles bem mais intuitiva.

Colateralmente ao foco nos objetos virtuais buscado neste trabalho, também se conseguiu melhorar a detecção dos marcadores com o uso de duas técnicas simples: o uso de tabuleiros e a recuperação da última posição detectado do marcador quando esta é perdida.

6.1 **Trabalhos futuros**

Como trabalhos futuros, o mais iminente é a portabilização para aparelhos portáteis. Estes são a plataforma mais interessante e indicada para o uso de realidade aumentada. Isto acontece principalmetne por dois motivos: a câmara se move junto com o usuário e a tela está atrás da câmara. Isto da a sensação do uso de uma lente “mágica” que mostra objetos que não podem ser vistos sem ela.



Figura 6.1: Realidade aumentada em aparelho portáti

Outro trabalho futuro é o rastreamento por filtro cromático: que usa uma cor ao invés de código para detectar objetos na imagem. Este tipo de rastreamento possibilita o uso de objetos mais naturais, além de tornar seu reconhecimento mais estável.

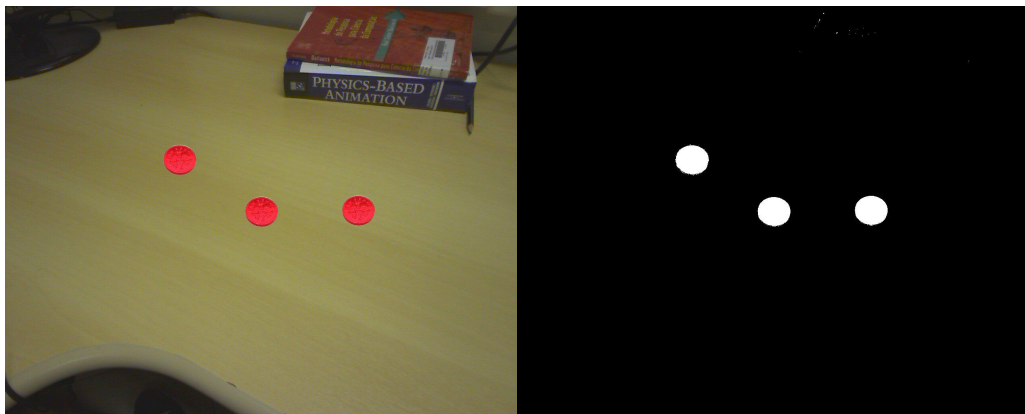


Figura 6.2: Rastreamento cromáti

Também há a possibilidade de rastreamento por características naturais, que é o atual estado da arte da tecnologia de realidade aumentada. Este tipo de rastreamento possibilita o rastreamento de qualquer objeto, usando suas características e as detectando posteriormente no vídeo.



Figura 6.3: Rastreamento por características natura

Referências Bibliográficas

Azuma Ronald. A Survey of Augmented Reality. Presence. 1997. 6, 4. 355–38

Beaney D., MacNamee B. Forked! A demonstration of physics realism in augmented reality. Mixed and Augmented Reality, 2009. ISMAR 2009. 8th IEEE International Symposium on. 2009. 171–172.

Chae Changhun, Ko Kwanghee. Introduction of Physics Simulation in Augmented Reality. Ubiquitous Virtual Reality, 2008. ISUVR 2008. International Symposium on. July 2008. 37–40.

Coumans Erwin. Bullet 2.80 Physics SDK Manual. 2012.

Eberly D.H. 3D game engine design: a practical approach to real-time computer graphics. 2007.

Erleben K., Sporning J., Henriksen K., Dohlmann H. Physics-based animation. 2005.

Ettlin A. Rigid body dynamics simulation for robot motion planning. 2006.

Ferwerda J. A. Three varieties of realism in computer graphics. SPIE Human Vision and Electronic Imaging '03. 2003. 290–297.

Freeman R., Steed A. Mixed Reality Toolkit. MSc VIVE Final Year Project Report, Department of Computer Science, University College London. 2004.

Haller Michael. Photorealism or/and Non-Photorealism in Augmented Reality. 2004.

Johnson Paul. Lista de e-mail da bullet. March 2008.

Junker G. Pro OGRE 3D programming. 2006.

Kim S., Kim Y., Lee S.H. On Visual Artifacts of Physics Simulation in Augmented Reality Environment. Ubiquitous Virtual Reality (ISUVR), 2011 International Symposium on. 2011. 25–28.

Kitchenham B. Procedures for performing systematic reviews. 2004.

- Klein G., Murray D.* Parallel tracking and mapping for small AR workspaces. Mixed a Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on. 2007. 225–234.
- Lee T., Hollerer T.* Handy AR: Markerless inspection of augmented reality objects using fingertip tracking. Wearable Computers, 2007 11th IEEE International Symposium on. 20083–90.
- Liu D.S.M., Yung C.H., Chung C.H.* A Physics-Based Augmented Reality Jenga Stacking Game. Proceedings of the 2011 Workshop on Digital Media and Digital Content Management. 2011. 1–8.
- Millington I.* Game Physics Engine Development. 2007. (The Morgan Kaufmann Series in Interactive 3D Technology).
- Salinas Rafael Muñoz.* ArUco: a minimal library for Augmented Reality applications based on OpenCv. June 2012.
- Schmalstieg D., Fuhrmann A., Hesina G., Szalavári Z., Encarnação L.M., Gervautz M., Purgathofer W.* The studierstube augmented reality project. Presence: Teleoperators and Virtual Environments. 2002. 11, 1. 33–54.
- Schweighofer G., Pinz A.* Robust pose estimation from a planar target. Pattern Analysis and Machine Intelligence, IEEE Transactions on. 2006. 28, 12. 2024–2030.
- Sörös Gabor, Rautek Peter, Seichter Hartmut, Gröller Eduard.* Augmented Visualization with Natural Feature Tracking. 10th ACM International Conference on Mobile and Ubiquitous Multimedia. Beijing, China: ACM, 12/2011 2011.
- Streeting Steve.* OGRE Manual v1.7 (Cthugha). 2010.
- V. Lepetit J. Pilet, Fua P.* Point Matching as a Classification Problem for Fast and Robust Object Pose Estimation. Conference on Computer Vision and Pattern Recognition, Washington, DC. June 2004.